



Munich Personal RePEc Archive

Random Variables, Their Properties, and Deviational Ellipses: In Map Point and Excel, v 4.0

Goodwin, Roger L

6 December 2014

Online at <https://mpra.ub.uni-muenchen.de/64442/>

MPRA Paper No. 64442, posted 18 May 2015 19:11 UTC

RANDOM VARIABLES, THEIR PROPERTIES, AND DEVIATIONAL ELLIPSES

RANDOM VARIABLES, THEIR PROPERTIES, AND DEVIATIONAL ELLIPSES

In Map Point and Excel, v 4.0

Roger L. Goodwin

Copyright ©2015 by Roger L. Goodwin, All rights reserved.

Published by Roger L Goodwin, West Virginia.

*To my parents and in
memory of Winston*

CONTENTS IN BRIEF

1	Introduction to VBA	1
2	Introduction to Map Point	13
3	Mathematics Review	23
4	Geo-Coding	35
5	Standard Deviatonal Ellipse	45
6	The Exponential Ellipse	59
7	The Weibull Ellipse	83
8	Spherical Statistics	117

CONTENTS

List of Figures	xiii
List of Tables	xix
Foreword	xxi
Preface	xxiii
Acknowledgments	xxvii
Acronyms	xxix
1 Introduction to VBA	1
1.1 The Development Environment	1
1.2 Variables	3
1.3 Arrays and Records	4
1.4 Branching	5
1.5 Loops	6
1.6 Setting Properties	7
1.7 The Excel Examples	8
2 Introduction to Map Point	13
	ix

2.1	Handling Data	14
2.2	Plotting the Mean Center	14
2.3	Visualizing the Survey Data	15
2.4	Drawing Axes	19
2.5	Drawing Ellipse Boundaries	19
2.6	Programmer Notes	21
3	Mathematics Review	23
3.1	Notation	23
3.2	Derivatives	24
3.3	Integrals	25
3.4	Quadratic Equation	26
3.5	Ellipse Equation	26
3.6	Trigonometry Functions and Conversion Factors	26
3.7	Modulo Arithmetic	27
3.8	Weighted Data	28
3.9	Area	30
3.10	Eccentricity	30
3.11	Axes Length	31
3.12	Rotation	32
3.13	Base 10 Logarithm	33
4	Geo-Coding	35
4.1	Kentucky June Area Survey	37
4.2	US Crime Statistics	39
4.3	OECD Countries	40
4.4	Coordinate Systems	42
5	Standard Deviational Ellipse	45
5.1	A Weighted Regression Model	46
5.2	Mean Latitude and Mean Longitude	47
5.3	VBA	48
5.4	Standard Deviational Ellipse	48
5.5	VBA	49
5.5.1	Weighted Mean Center	50
5.6	Ellipse Properties	51
5.7	VBA	52
5.8	Kentucky Example	54

5.9	Crime Example	55
5.10	GDP Example	56
5.11	Exercises	58
6	The Exponential Ellipse	59
6.1	Mean Latitude	59
6.2	Mean Longitude	62
6.3	Joint Distribution	63
6.4	Exponential Ellipse	65
6.4.1	Mean Center	65
6.5	VBA	65
6.5.1	Ellipse	68
6.6	Axis Rotation VBA Code	68
6.7	Ellipse Properties	68
6.8	VBA	70
6.9	Kentucky Example	73
6.10	Crime Example	73
6.11	GDP Example	75
6.12	Comparison to SDE	76
6.13	Exercises	80
7	The Weibull Ellipse	83
7.1	Latitude Mean Center	83
7.2	Longitude Mean Center	86
7.3	VBA	89
7.4	Joint Distribution	91
7.5	VBA	94
7.6	Weibull Ellipse	97
7.6.1	Ellipse Angle	97
7.7	Ellipse Properties	98
7.8	VBA	99
7.9	Kentucky Example	103
7.10	Crime Example	106
7.11	GDP Example	106
7.12	Axes Length Comparison	108
7.13	Sample Distribution Fitting	108
7.14	VBA	111
7.15	Exercises	115

8	Spherical Statistics	117
8.1	Introduction	117
8.2	Concepts	119
8.3	X -Axis Mean Center and Resultant Length	120
8.4	Y and Z -Axis Mean Centers	121
8.4.1	Spherical Variance	121
8.5	VBA	122
8.6	Test and Characterizations	128
8.7	VBA	128
8.7.1	Finding the Eigenvalues	131
8.8	VBA	132
8.8.1	Finding the Eigenvectors	134
8.9	VBA	134
8.10	Kentucky Example	139
8.11	Crime Example	142
8.12	GDP Example	144
A	SDE VBA Drivers	147
B	Exp VBA Drivers	155
C	Weibull VBA Drivers	161
D	Spherical VBA Drivers	170
E	References	173
	Index	179

LIST OF FIGURES

1.1	<i>This figure shows Excel without the Developer menu.</i>	2
1.2	<i>This figure shows Excel Options dialog box. Excel uses the dialog box for adding the Developer menu for VBA programming.</i>	2
1.3	<i>This figure shows Excel with the Developer menu at the top of the screen.</i>	3
1.4	<i>This figure shows the Excel workbook. A data worksheet at the bottom named KY 2003 has been circled. The worksheet named STATS has been circled. This application always saves the results to the STATS worksheet.</i>	10
1.5	<i>This figure shows the Excel spreadsheet with data. For this application to run properly, the first row can contain any column names the reader wishes. Column 1 must contain the latitude observations. Column 2 must contain the longitude observations. Column 3 must contain the random variable of interest. It is not advisable to put any observations in the remaining columns as this application may over-write them.</i>	11
2.1	<i>This figure shows the Map Point dialog box for finding a point on the Earth using the latitude and longitude.</i>	14

2.2	<i>This figure shows the Map Point dialog box for browsing Excel files, which have the coordinates and the random variable.</i>	15
2.3	<i>This figure shows the Map Point dialog box that lists the Excel spreadsheets in the Excel workbook.</i>	15
2.4	<i>This figure shows the Map Point dialog box containing the spreadsheet column names and the data.</i>	16
2.5	<i>This figure shows the Map Point Data mapping wizard.</i>	17
2.6	<i>This figure shows the Map Point choices for the pushpin colors.</i>	17
2.7	<i>This figure shows the Map Point choice for the column chart.</i>	17
2.8	<i>This figure shows the Map Point choices for scaling the data and change the label text.</i>	18
2.9	<i>This figure shows the Microsoft Word shapes for drawing an ellipse.</i>	19
2.10	<i>This figure shows the Microsoft Office dialog box for formatting a shape.</i>	20
3.1	<i>This figure shows the Options dialog box in Map Point. We use it to convert the latitude and longitude coordinates to decimal degrees.</i>	27
3.2	<i>This figure shows the Options dialog box in Google Earth. We use it to convert the latitude and longitude coordinates to decimal degrees.</i>	28
4.1	<i>This figure shows Google Earth. It shows how to geo-code a county with-in a state.</i>	36
4.2	<i>This figure shows the Excel spreadsheet of the Kentucky data. The spreadsheet has not been geo-coded. It still contains the regional data.</i>	38
4.3	<i>This figure shows the Excel spreadsheet of the crime data. The spreadsheet has not been geo-coded and still contains the regional data, two years of data in the same spreadsheet, merged columns, and percentages.</i>	39
4.4	<i>This figure shows the Excel spreadsheet of the Gross Domestic Product data. The spreadsheet has not been geo-coded.</i>	41
4.5	<i>This figure shows the Earth using Map Point. We identify points on the Earth using latitude and longitude.</i>	43

5.1	<i>This figure shows the Kentucky data from 2003 and the standard deviational ellipse. It shows a sketch of the standard deviational ellipse with Ohio County at the center. The semi-major axis is approximately 184 miles long and the semi-minor axis is approximately 52 miles long. We rotated the major axis roughly 70° from the Y-axis in an imaginary Cartesian coordinate system.</i>	54
5.2	<i>This figure shows the standard deviational ellipse of violent crime in the US for the year 2007. It also shows the values of the violent crime using a bar graph. The ellipse is rather large compared to the area in the survey.</i>	56
5.3	<i>This figure shows the plot of the mean centers of global GDP on a map. As the years progress, the mean center measurements move closer to the Chinese coastline.</i>	57
6.1	<i>This figure shows the probability plot of the weighted latitude transformation. The exponential distribution fits the weighted latitude data since it shows a straight line.</i>	60
6.2	<i>This figure shows the probability plot of the weighted longitude transformation. The exponential distribution fits the absolute value of the weighted longitude data since it shows a straight line. It was necessary to take absolute values since longitudinal data is negative.</i>	60
6.3	<i>This figure shows sketches of both the standard deviational ellipse and the exponential deviational ellipse from the Kentucky data for the year 2003. The exponential ellipse is the smaller of the two ellipses. It covers one-third the area of the standard deviational ellipse, yet one-half of the same points.</i>	72
6.4	<i>This figure shows a raised bar graph of the weights in their respective counties for the Kentucky data for the year 2003. The majority of the weighted data comes from the lower-left corner of the state of Kentucky. The majority of the un-weighted data comes from the center of the state and the right-hand side.</i>	73
6.5	<i>This figure shows both the exponential deviational ellipse and the standard deviational ellipse for the violent crime data in the US for 2007. Both have the same center of gravity. The exponential ellipse is much smaller than the standard deviational ellipse.</i>	74
7.1	<i>This figure shows both the Weibull deviational ellipse for the Kentucky data for the year 2003. The zoom level is 200 miles above the earth. The standard deviational ellipse and the exponential deviational ellipse were too large to fit on the map. It has the same center of gravity as the other ellipses. The Weibull ellipse is much smaller than the other two.</i>	105

7.2	<i>This figure shows both the Weibull deviational ellipse for the Crime data for the year 2007. The zoom level is 550 miles above the earth. The standard deviational ellipse was too large to fit on the map. It has the same center of gravity as the other ellipses. The Weibull ellipse is much smaller than the other two.</i>	107
8.1	<i>This figure shows a map of the U.S. relative to the Equator (represented by the black line).</i>	119
8.2	<i>This figure shows the diagram of a triangle with the quantities for the longitude θ, n_0, and the resultant length R.</i>	123
8.3	<i>This figure shows the VBA environment for Module 4. It also highlights the driver subroutine <code>modal_stats()</code>.</i>	123
8.4	<i>This figure shows the STATS spreadsheet with the output from the subroutines <code>main_Modal()</code> and <code>modal_stats()</code>.</i>	124
8.5	<i>This figure shows the VBA environment for Module 5. It also highlights the driver subroutine <code>main_Modal()</code>.</i>	135
8.6	<i>This figure shows the histogram for the Kentucky 2003 data.</i>	140
8.7	<i>This figure shows the Kentucky 2003 converted data.</i>	140
8.8	<i>This figure shows the histogram for the Violent Crime 2008 data. Three possible modes appear at California, Texas, and Florida.</i>	143
8.9	<i>This figure shows the 2008 US Crime converted data.</i>	143
8.10	<i>This figure shows the histogram for the GDP 2008 data. The U.S. accounts for approximately 25% of the total GDP and China accounts for approximately 14%.</i>	145
8.11	<i>This figure shows the 2008 Gross Domestic Product converted data.</i>	145
A.1	<i>In Excel, double click on Module 1 in the left-hand pane circled in red to display the VBA code for the Standard Deviational Ellipse. The subroutine called <code>SDE()</code> runs the programs to calculate the statistics in this chapter.</i>	148
A.2	<i>This figure shows the VBA environment for Module 1. It also highlights the driver subroutine <code>SDE()</code>.</i>	149
A.3	<i>This figure shows the STATS worksheet and the results from the <code>SDE()</code> subroutine.</i>	150
B.1	<i>This figure shows the VBA environment for Module 2. It also highlights the driver subroutine <code>EXPellipse()</code>.</i>	157

B.2	<i>This figure shows the STATS worksheet and the results from the EXPellipse() subroutine.</i>	158
B.3	<i>Double click on Module 2 to display the VBA code for the Exponential Deviational Ellipse in Excel. The subroutine called EXPellipse() runs the programs to calculate the statistics in this chapter.</i>	160
C.1	<i>This figure shows the VBA environment for Module 3. It also highlights the driver subroutine RunWeibull().</i>	162
C.2	<i>This figure shows the STATS worksheet and the results from the Weibull() subroutine.</i>	163
C.3	<i>Double click on Module 3 to display the VBA code for the Secant code and the Weibull Deviational Ellipse in Excel. The subroutines called Secant1(), Secant2(), and RunWeibull() run the programs to calculate the statistics in this chapter. As the reader may have guessed, calculating the statistics in this chapter is more complex than those in previous chapters.</i>	167

LIST OF TABLES

5.1	Results on Corn acreage in Kentucky	54
5.2	Results on Corn acreage in Kentucky	55
5.3	Results on Violent Crime in the U.S.	55
5.4	Results on Violent Crime in the U.S.	56
5.5	Results on Gross Domestic Product of OECD Countries	57
5.6	Results on Gross Domestic Product of OECD Countries	58
6.1	Results for the Exponential Model for Kentucky	71
6.2	Results for the Exponential Model for Kentucky	72
6.3	Results for the Exponential Model on Violent Crime in the U.S.	74
6.4	Results for the Exponential Model on Violent Crime in the U.S.	74
6.5	Results for the Exponential Model on Gross Domestic Product	75
6.6	Results for the Exponential Model on Gross Domestic Product	75
6.7	Relationship of angles of rotation for large x'_i s.	80

6.8	Relationship of angles of rotation for small x'_i 's.	80
7.1	Results for the Weibull Model for Kentucky	104
7.2	Results for the Weibull Model for Kentucky	104
7.3	Results for the Weibull Model on Violent Crime in the U.S.	106
7.4	Results for the Weibull Model on Violent Crime in the U.S.	106
7.5	Results for the Weibull Model on Gross Domestic Product	107
7.6	Results for the Weibull Model on Gross Domestic Product	107
8.1	Interpretation of the Eigenvalues	131
8.2	Eigenvalues and Eigenvectors for the 2003 Kentucky Data	141
8.3	Results for the Spherical Model for Kentucky	142
8.4	Eigenvalues and Eigenvectors for the 2008 Violent Crime Data	144
8.5	Results for the Spherical Model for 2008 Crime	144
8.6	Eigenvalues and Eigenvectors for the 2008 Gross Domestic Product	146
8.7	Results for the Spherical Model for GPD	146

FOREWORD

This book is a practical reference guide accompanied with an Excel Workbook. This book gives an elementary introduction of the weighted standard deviational ellipse. This book also presents the computational aspects of the weighted exponential distributions as well. For the examples given, calculations are performed using VBA for Excel. This book makes comparisons (and shows the computations via VBA for Excel) using the likelihood functions with spatial data of the weighted ellipses. Lastly, the book covers spherical statistics. Throughout the text, the reader can see how to perform these difficult calculations and learn to adapt the code for his research.

PREFACE

Scientific articles exist since the 1960's for the use of an area frames for collecting and interpreting survey data in the United States, and in the 1990's in Europe and Africa. Cartographers divide units of land from satellite imagery into enumerable segments. Statisticians assign the segments to defined strata. Typically, crop estimation is performed using imagery analyzes, field surveys, and mail surveys. The following list of articles includes both practical results and theoretical results typically found in the literature on area frame surveys.

[Pratt, Bird, Taylor, and Carter (52)] wrote a paper that covers the topics of planning and implementing an area frame survey in Nigeria and choosing the estimation procedures. They were careful to choose their satellite imagery and coordinate it with the fieldwork. In Europe and Africa, Statisticians use regression to classify the satellite images from field data collection and from image classification. Image classification without field data collection does tend to lead to failure due to the following reasons: 1) intercropping practices, 2) fallow/cultivated continuum and 3) small, irregular field structure. The authors more generally state those reasons as "spectral confusion." The authors state that the best time to perform their fieldwork is shortly after harvest because it would be easiest to distinguish harvested rice fields from wet, green swamp grasslands. [page 70] calculates the sample size in terms of segments according to several criteria: 1) the size of the study size area, 2) the need to maximize the area covered, 3) the speed to cover the area during enumeration, 4)

reduce locational error within the segment during field mapping. The authors obtain a 4.8% sampling fraction by choosing forty-nine segments (500×500 meters with a sampling fraction of 5%) out of 75. Three estimators of the data collected are presented: 1) direct expansion of the survey data, 2) pixel count, and 3) regression estimator. The authors recommend the regression estimator, which take into account both the irrigation mapped by enumerators on the ground in each segment and the satellite image classification. This is because it was highly dependent on the classifier and gave a narrower range of possible classifications. They used a ratio of the direct expansion variance and the regression model to prove this mathematically.

[Kelly (32), Chhikara and Deng (6), Faulkenberry and Garoui, (15)] discuss area-frame data collection and estimation in the United States. [Kelly (32)] collects and summarizes survey data in a production environment. [Chhikara and Deng (6)] consider the problem of using the area frame and rotating segments amongst years. On page 926, they develop an ANOVA model to capture the stratum means per year and the segment mean effects for a particular random variable. The author also rotates and overlaps segments amongst years via a simulation study that shows the optimal rotation of segments should be 40% to 60%. [Faulkenberry and Garoui, (15)] discuss the topic of estimating population totals and variances from an area survey frame. The authors point out that because of the association of a farm with more than one segment does not lead to cluster sampling. Based on four classifications, the authors show that the Horvitz-Thompson estimator for totals is probably the best choice of estimators. It is a good estimator when the probability of selection π_k is proportional with the random variables y'_k 's. Depending on the number of farms or the number of known segments, four classifications arise. The author assigns an estimator to each class that will work well, and then takes expectations and variances.

Although most of the authors so far do present maps with their articles and their statistical estimation methods, they lack discussing the latitude and longitude of their random variables. The remainder of this section will discuss the history of associating random variables with the latitude and longitude (i.e. the placement on a map).

[Ebdon, (13), Chapter 7, 1985] discusses spatial statistics and several measures. The author gives interpretations to the concepts of the mean center and standard deviational ellipse. For instance, the mean center can be thought of the "center of gravity" of the distribution of the given points on a map. He defines the standard deviational ellipse as the "spread of points" about the mean center. Without using modern software, the author has an interesting way to draw the ellipse. It involves plotting the deviation for each point parallel to the rotated axes and fitting the ellipse [Ebdon, (13), p. 137].

The author Lefever first used the standard deviational ellipse in 1926 to study the geographic position of social problems [Lefever, (40)]. The author argues that a uniform distribution will not fit well for location problems of any kind because ecological and sociological problems occur more frequently on a select area of the map than other areas. The author shows the correspondence between calculating the standard deviation between the mean center of the points and each location; and then showing its geometric relationship on a map. An example is given. In the conclusion, the author states four characteristics of a unit of locations:

1. The center of the system determined by the distances from the central point (versus the extreme unit locations).
2. The direction or trend of the system given by the angle θ_m of the axis of maximum standard deviational variation value. This is the line of best fit for the entire system of unit locations.
3. The concentration of the system (or dispersion) shown in terms of a standard deviational ellipse.
4. The relative concentration given by the ratio of observations within the ellipse compared to the entire population expressed in area units.

The author gives no references in the paper. To summarize, the article presents the mean and standard deviation of a set of numbers — longitude and latitude. So, who came up with the idea of weighting the longitude and latitude data?

The weighted standard deviational ellipse appears in the literature in 1971 in [Yuill, (67)]. The author begins with a discussion of the work by Lefever and the comments by Furtley in 1927. The paper proceeds to defend using the ellipse for geographic applications. The author derives the formulas on pages 30-31 for the weighted standard deviational ellipse. It is on page 32 that the weighted mean center is introduced (a new notion in the literature). The author weights the latitude and longitudinal observations using the random variable. The author gives the following computations for comparing ellipses.

1. The enclosed area within the ellipse.
2. The number of points enclosed within the ellipse (more is better).
3. The shape of the ellipse measured by its eccentricity .

The shape of the ellipse determines the distribution of the points . Points concentrated at the pole of the ellipse (or circle) should have a non-uniform distribution while those scattered from the pole will have a uniform distribution . Finally, the author applies the concepts to several sets of data. References appear in the footnotes throughout the paper.

So, where did the weighted mean center originate? Yuill's paper stated the mean center, but did not derive it. Books and research articles have methods on the weighted least squares dating back to the 1970's. The weight assigned to each longitude and latitude pair is between zero and one; additionally, the weights sum to one. This is due to the formula, not due to the nature of the application (e.g. ecological versus sociological problem). Other authors [Magee (42), and Woolson and Clarke (65)] assign least squares weights according to the sample design. Magee assigns weights based on sample selection probabilities. Woolson and Clarke assign weights to compensate for missing data. [Rubin, (55)] models the leverage values as weights in the regression model. This in turn shows which data points in the data set exert more influence over the others. [Lee and Wong, (36), pages 38-39] assigns weights

according to the distribution of the phenomenon. If a city's population is the random variable of interest, then the city's population is the weight for the latitude and longitude. We apply weights in the context of [Lee and Wong, (36)] in this book.

Finally, to extend the concepts of the weighted mean center and the standard deviational ellipse, at times, we prove that the data has a particular distribution. By transforming the data and plotting the points $[t, F(t)]$ where $F(t)$ is a known cumulative distribution function, one should obtain a straight regression line [Lee, (35)] as long as the data has that distribution. The author covers data transformations for distributions with a location parameter, a scale parameter and a shape parameter. Knowing the distribution of the data is advantageous. Using maximum likelihood estimators, we can compare likelihood functions, [Lee (35), page 227]. Generally, we choose the likelihood function with the largest value.

ACKNOWLEDGMENTS

Map Point, Excel, and MS Word are products of Microsoft Corporation

Google Earth is a product of Google, Inc.

R. L. G.

ACRONYMS

GDP	Gross Domestic Product
OECD	Organization for Economic Cooperation and Development
SDE	Standard Deviational Ellipse
VBA	Visual Basic for Applications

CHAPTER 1

INTRODUCTION TO VBA

1.1 The Development Environment

The calculations in this book are difficult at times. Having a programming environment becomes advantageous. A programming environment provides the flexibility to calculate statistics and likelihood functions. This textbook uses the Visual Basic Application (VBA) for Excel. It is a programming environment based on the Visual Basic programming language. The VBA Development environment does not automatically appear as a menu option in Excel. See Figure 1.1. The user must make this option visible by following these steps:

1. Click on File | Options | Customize Ribbon. The dialog box in Figure 1.2 will appear.
2. From the Choose Commands drop-down list, select Main Tabs.
3. Highlight Developer.
4. Click on the Add>> button in the middle of the screen.

2 INTRODUCTION TO VBA

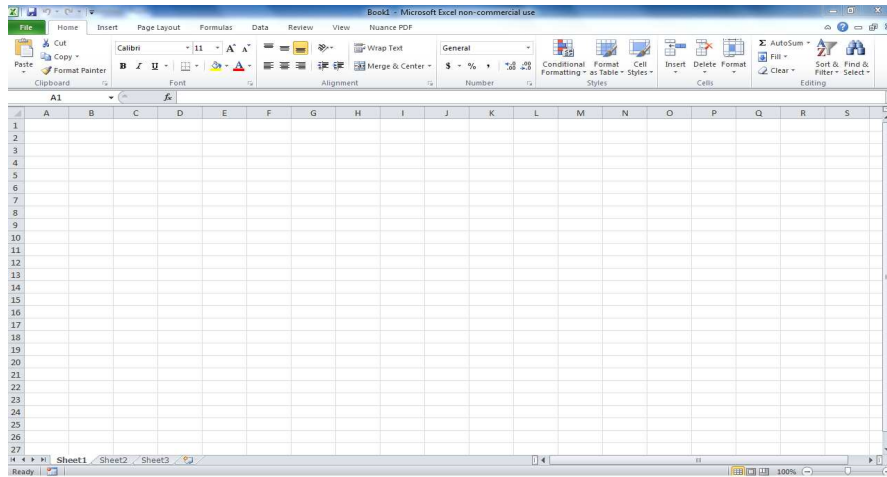


Figure 1.1 This figure shows Excel without the Developer menu.

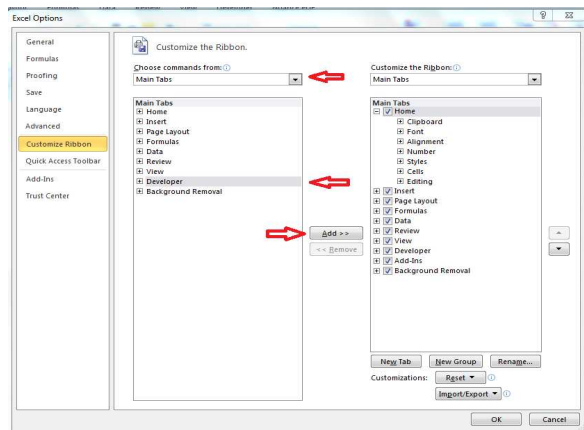


Figure 1.2 This figure shows Excel Options dialog box. Excel uses the dialog box for adding the Developer menu for VBA programming.

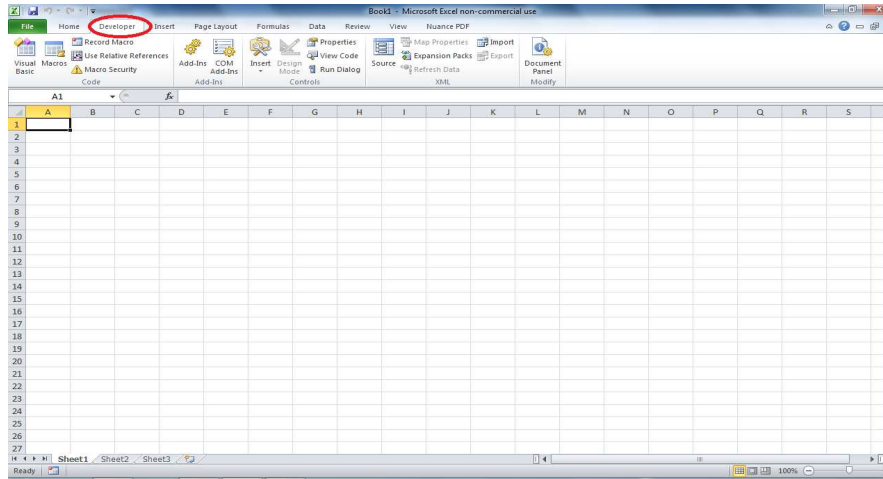


Figure 1.3 This figure shows Excel with the Developer menu at the top of the screen.

The **Developer** menu option will appear somewhere along the top of the screen. See Figure 1.3.

1.2 Variables

To declare variables explicitly in VBA for Excel, use the **Dim** statement. Excel supports the following basic data types:

- Integer numbers — **Dim X As Integer** The **Integer** data type holds integer variables stored as 2-byte whole numbers in the range of -32,768 to 32,767.
- Real numbers — **Dim Y As Double**. The **Real** data type holds double-precision floating-point numbers as 64-bit numbers in the range of -1.79769313486231E308 to -4.94065645841247E-324 for negative values and 4.94065645841247E-324 to 1.79769313486232E308 for positive values.
- String characters — **Dim S As String**. The **String** data can include letters, numbers, spaces, and punctuation. The **String** data type can store fixed-length strings ranging in length from 0 to approximately 63,000 characters.
- Date — **Dim BirthDay As Date**. Excel allows the **Date** data type to store dates and times as a real number. **Date** variables are stored as 64-bit (8-byte) numbers. The value to the left of the decimal represents a date, and the value to the right of the decimal represents a time. Excel considers the date expression to be any expression such as a date, including date literals, numbers that look like dates, strings that look like dates, and dates returned from functions. A date expression is limited to numbers or strings, in any combination, that can represent a date from January 1, 100 to December 31, 9999.

Dates are stored as part of a real number. Values to the left of the decimal represent the date; values to the right of the decimal represent the time. Negative numbers represent dates prior to December 30, 1899.

A date can be any sequence of characters with a valid format surrounded by number signs (#). Valid formats include the date format specified by the locale settings for your code or the universal date format. For example, use #12/31/92# in the VBA editor when explicitly referring to a date.

- **Currency — Dim Q As Currency.** The **Currency** data type has a range of -922,337,203,685,477.5808 to 922,337,203,685,477.5807. Use this data type for calculations involving money and for fixed-point calculations where accuracy is particularly important. In the VBA Editor, use the "@" sign when referring to currency.
- **Long integer — Dim R As Long.** The **Long** data type is a 4-byte integer ranging in value from -2,147,483,648 to 2,147,483,647. In the VBA Editor, use the "&" symbol when referring to long integers.
- **Logical — Dim L As Boolean.** The **Boolean** data type has only two possible values, True (-1) or False (0).

Variables defined inside a subroutine are visible only inside that subroutine. Variables defined at the module level are visible to the subroutines defined in that module.

1.3 Arrays and Records

We can build on the basic data types listed in Section 1.2. Consider arrays and records.

- **One dimensional array — Dim Name_List(1 To 10) As String.** This array defines a set of sequentially indexed elements having the data type **String**. Each element of the array has a unique identifying index number. Changes made to one element of an array do not affect the other elements.
- **Two dimensional array — Dim AList(1 To 5, 1 To 10) As Double.** This array defines a matrix of indexed elements having the numeric data type **Double**.
- **A user defined record —** Excel allows the definition of the user-defined record using the **TYPE-END-TYPE** statement. To define a list of customer address records, use the following VBA code:

```

1 3  [Type Customers
      [First_Name, Last_Name As String
        Address As String
        City As String
        State As String
        Zip_Code As Integer
        Phone As Integer
      ]
    ]
2  [End Type

```

```

2  [Dim List As Customers

```

Do not use the **Dim** inside the **TYPE-END-TYPE** statement. Use the dot "." notation to reference the items in the record. For example in the VBA Editor, to reference the customer phone number, use the code:

```

1  [List.Phone = 3223223

```

1.4 Branching

VBA has an **IF-THEN-ELSE** statement for conditionally executing code. The syntax is as follow:

```

1  [
2  [ If <condition> Then
      VBA statements
      Else
3  [ VBA statements
      End If

```

The **IF-THEN-ELSE** construct must appear in a subroutine. It cannot appear as open code in a module. Aside from variable definitions and subroutine declarations, this is true for the majority of VBA programming. There can only be one **ELSE** statement in any **IF-END-IF** construct. An alternative to the above **IF-THEN-ELSE** construct is the **IF-ELSE-IF** construct.

```

1  [
2  [ If <condition> Then
    VBA statements
3  [ ElseIf <condition> Then
    VBA statements
4  [ ElseIf <condition> Then
    VBA statements
5  [ Else
    VBA statements
    End If

```

If the user has a short, single VBA statement for any of the **ELSEIF** statements, it is not advisable to put it on the same line after the **THEN**. It usually creates a run-time error even though the syntax looks correct.

1.5 Loops

The two loops covered in this section are the **FOR-NEXT** loop and the **WHILE-WEND** loop. To execute a set of VBA statements a given number of times in a given sequence, the **FOR-NEXT** loop has the following syntax:

```

1  [ For <counter> = start To end
    VBA statements
    Next <counter>

```

As an example, we can generate two digits in a phone number.

```

1  [ multiplier = 1
    List.phone = 0
    [ For i = 1 To 7
      List.phone = List.phone + i*multiplier
      multiplier = multiplier * 10
    ] Next i

```

The value of `List.phone` is 28 because $\frac{n(n+1)}{2} = \frac{7(8)}{2} = 28$. The only purpose of the example is to show the syntax of the **FOR-NEXT** loop.

The **WHILE-WEND** loop has the following syntax.

```

1  [ While <condition>
    VBA statements
    Wend

```

It is more likely to program infinite loops with the **WHILE-WEND** loop than with the **FOR-NEXT** loop. It is important to initialize the condition variable(s) before entering the loop and to update the condition variable(s) inside the loop.

1.6 Setting Properties

Both the Excel spreadsheet and the VBA environment allow the user to change the cell contents properties to bold, italic, underline, strikethrough, and the font size. Except for the font size, most of these properties are Boolean valued.

- Object.Bold — Boolean
- Object.Italic — Boolean
- Object.Size — Integer
- Object.StrikeThrough — Boolean
- Object.Underline — Boolean

In our case, the VBA object is a cell in a spreadsheet. We use the following code to reference the properties of the cell.

```
1 [ActiveSheet.Cells(1, 3).Font.Bold = True
   ActiveSheet.Cells(1, 3).Font.Italic = True
   ActiveSheet.Cells(1, 3).Font.Size = 25
   ActiveSheet.Cells(1, 3).Font.Strikethrough = True
   ActiveSheet.Cells(1, 3).Font.Underline = True
```

`Cells(1, 3)` references row 1, column C in the last spreadsheet viewed before entering the VBA environment. Alternatively, we could have substituted the `ActiveSheet` with a specific spreadsheet name such as `Worksheets("Sheet1")`. We have hard-coded `Sheet1`. `Sheet1` will always be the spreadsheet referenced.

When updating a set of cells in a spreadsheet, it is convenient to use the **WITH-END-WITH** statement. It can save some typing and make the code easier to read. For example, we can re-write the VBA code for the font updates as follow:

```
1 [With ActiveSheet
   .Cells(1, 3).Font.Bold = True
   .Cells(1, 3).Font.Italic = True
   .Cells(1, 3).Font.Size = 25
   .Cells(1, 3).Font.Strikethrough = True
   .Cells(1, 3).Font.Underline = True
   End With
```

The **WITH-END-WITH** statement is convenient to use in complicated calculations as well.

1.7 The Excel Examples

The Excel spreadsheet and the Excel VBA environment contain many of the same functions. Some common syntax and conventions for the worksheet follows.

1. Columns in an Excel worksheet always begin with a letter.
2. Rows in an Excel worksheet always begin with a number.
3. The three default names of the worksheets in an Excel workbook are `Sheet1`, `Sheet2`, and `Sheet3`. Microsoft Corporation capitalized the "S" in the word "Sheet." In the VBA for Excel environment, upper and lower case counts.
4. Character and numeric data can ordinarily be copy and pasted into the worksheet cells.
5. Precede numeric data with leading zeros with the single quote " ' " to retain those leading zeros. Entering the single quote in front of the data is a manual operation. Changing the column to the TEXT format usually causes other problems later on.
6. Formulas begin with an equal sign. Some useful formulas used in this application include:
 - `mod(number, divisor)` — where the argument `number` can either be reference to a cell or a hard coded number such as 180 or 360; and the argument `divisor` can either be a cell reference or a hard coded number such as 180 or 360.
 - `average(range)` — where the argument `range` is a range of cells such as `A2..A91`.
 - `sum(range)` — where the argument `range` is a range of cells such as `C2..C39`.

In this application, we keep the data and results in the worksheets. We use VBA to perform the calculations. Some common conventions in VBA for Excel follow.

1. Declare global variables at the top of a module.
2. VBA program code must appear inside a subroutine. Some useful, recurring VBA statements used in this application include:
 - `Selection.Rows.Count` — Visual Basic applies the `selection` expression to a range of data in the active spreadsheet. It returns a count of the number of rows. Since we included the first row, the count n is larger by one. This application adjusts for this.

- `ActiveSheet.Cells(row, col).Value` — The `activesheet` object references the data in the spreadsheet highlighted before entering the VBA editor. This entire statement allows reading the data.
- `Worksheets("name").Cells(row, col).Value` — The `worksheets` object references the spreadsheet named NAME. In the application name = stats. This entire statement allows writing data to a spreadsheet other than the active spreadsheet.
- `For-Next` — This is the common looping structure used for calculating the sums, the areas, the probability distributions, eccentricities, and so on.
- `While-Wend` This application uses this looping in Module 3 in the Secant algorithms because the termination condition required more knowledge that the number of observations in the data set.
- VBA numeric calculations — The Visual Basic for Excel numeric calculations are similar in style to those of other programming language statements.

Some useful, recurring VBA functions used in this application include:

- `log(number)` — This function returns the natural logarithm of the given number. The user must write his own code for other base logarithms.
- `exp(number)` — This function returns the exponential function of the given number. e is approximately 2.718282.
- `worksheetfunction.pi()` — This function returns the value of π .

3. Modules are a collection of related subroutines and variables.
4. Declare local variables inside a subroutine.
5. Variable names are not case sensitive. For all declared variables, the VBA editor will change the upper and lower case spelling.

In this application, the results are stored in the spreadsheet called STATS. See Figure 1.4. The data are stored in the remaining spreadsheets. The first three columns of the data must be in a particular order. See Figure 1.5.

1. Latitude x_i .
2. Longitude y_i .
3. Random variable of interest w_i .

Further, select the data (and only the data) so that the subroutines can identify the beginning and ending rows. To get into the VBA editor, follow these steps.

1. Click on the `Developer` ribbon.

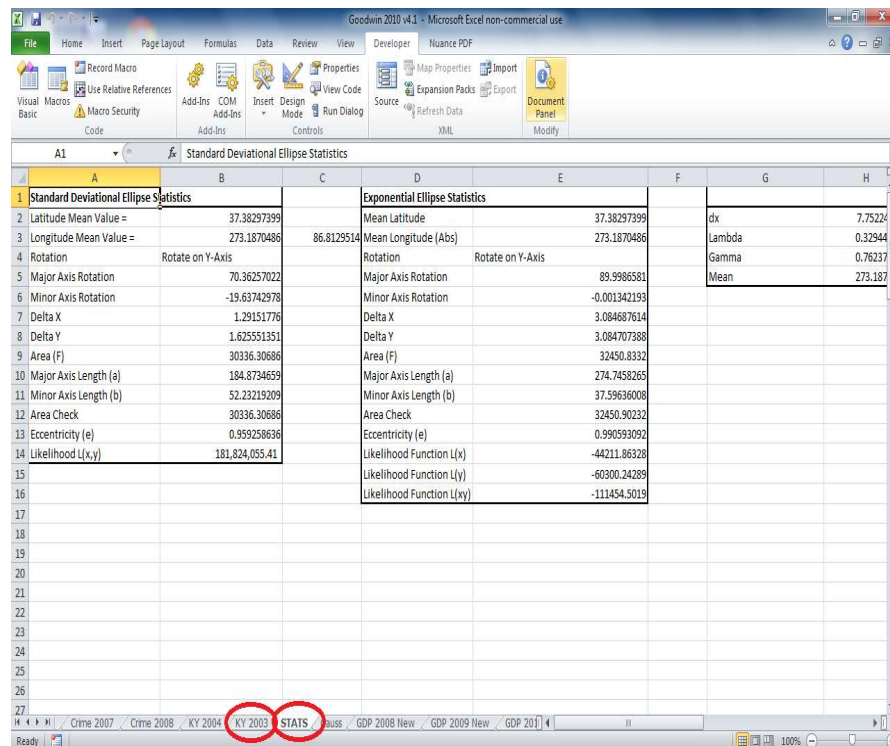


Figure 1.4 This figure shows the Excel workbook. A data worksheet at the bottom named KY 2003 has been circled. The worksheet named STATS has been circled. This application always saves the results to the STATS worksheet.

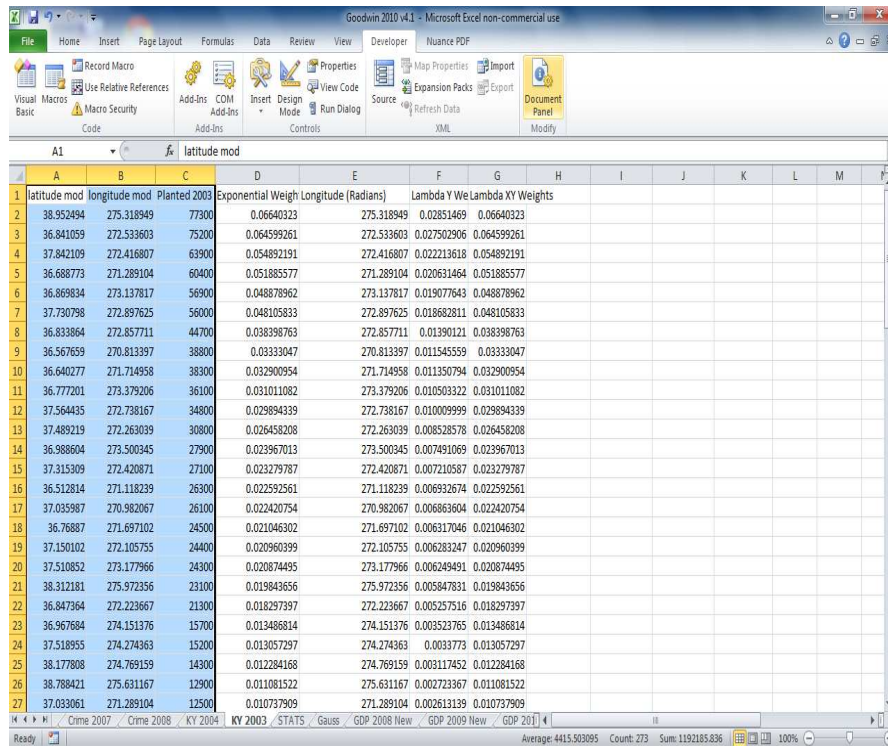


Figure 1.5 This figure shows the Excel spreadsheet with data. For this application to run properly, the first row can contain any column names the reader wishes. Column 1 must contain the latitude observations. Column 2 must contain the longitude observations. Column 3 must contain the random variable of interest. It is not advisable to put any observations in the remaining columns as this application may over-write them.

2. Click on the `Visual Basic` button on the left side.
3. For a new module, click on the `Insert` menu item and select `Module`.

A blank VBA Editor screen will appear. Just like any other programming language, either you know the syntax or you do not. VBA for Excel is similar in style to the programming language Basic. However, the objects tend to be centered around the Excel workbook — cells, spreadsheets, columns, rows, and so on.

CHAPTER 2

INTRODUCTION TO MAP POINT

Map Point is a software package that maps points on the Earth. The user provides the latitude and longitude, and Map Point plots the point. The points in this textbook are either geo-coded observations from Google Earth or calculations such as the mean center (also called the center of gravity) or the semi-major and semi-minor axis lengths of an ellipse. It is possible to enter, say a county name into Map Point. Map Point will identify the county on the Earth. However, it will not give the user the latitude and longitude coordinates. This is why we must use an alternate software product.

There are several useful concepts when using Map Point:

1. Plotting the mean center on the Earth.
2. Plotting and graphing the entire survey of points on the Earth.
3. Drawing lines with an exact length in miles (or kilometers).
4. Saving work.
5. Copying and pasting images to other software applications.

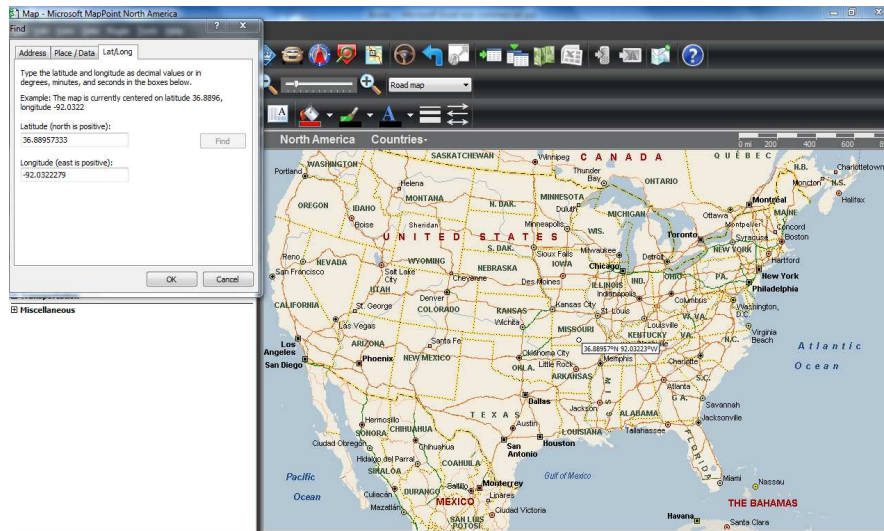


Figure 2.1 This figure shows the Map Point dialog box for finding a point on the Earth using the latitude and longitude.

Map Point does allow us to draw ellipses. However, it does not allow us to rotate them. Because of this, we must draw ellipses outside of Map Point. We use Map Point to draw the semi-major axis and the semi-minor axes.

2.1 Handling Data

Map Point has two ways to enter data:

1. Manually one data point at a time using `Tools | Find | Lat\Long`.
2. Import an Excel spreadsheet using `Data | Data import wizard`.

When the user enters data manually, one data point at a time, Map Point requests the latitude and the longitude. When the user imports multiple data points via an Excel spreadsheet, Map Point requires column names with "latitude" and "longitude." In addition, Map Point expects the user to plot a third variable called a "data" variable or an analysis variable along with the latitude and longitude. Naming the third variable is at the user's discretion. More notes follow on importing data in Section 2.3.

2.2 Plotting the Mean Center

To plot the mean center (a single data point), select `Tools | Find`. The dialog box on the left side in Figure 2.1 will appear. Upon selecting the `Lat/Long` option

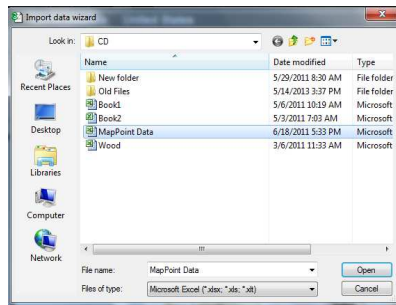


Figure 2.2 This figure shows the Map Point dialog box for browsing Excel files, which have the coordinates and the random variable.

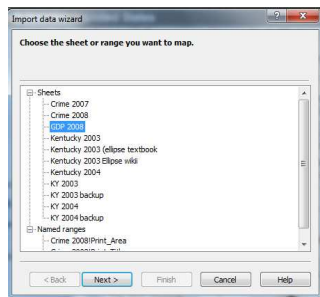


Figure 2.3 This figure shows the Map Point dialog box that lists the Excel spreadsheets in the Excel workbook.

at the top of the dialog box, the user types in the coordinates and clicks on the Find button. Map Point will locate that latitude and longitude on the Earth and identify those coordinates with a box.

2.3 Visualizing the Survey Data

Sometimes the user may want to visualize the survey data (i.e. many data points) and the ellipse together (i.e. drawing). Alternatively, maybe, the user may want to visualize the plotted points on the Earth with the random measurements (i.e. graphing). Map Point does provide the software tool to plot or graph an entire data set onto the Earth.

2.3.0.1 Plotting To plot the data, first prepare an Excel file with the latitude, longitude, and the random variable of interest. To plot the data set, select Data | Import data wizard.

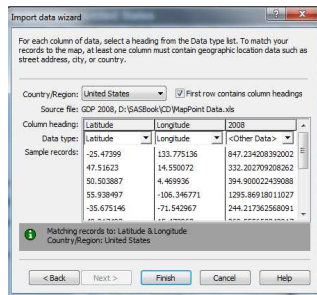


Figure 2.4 This figure shows the Map Point dialog box containing the spreadsheet column names and the data.

- The dialog box in Figure 2.2 allows the user to browse the hard drive and select the Excel file that contains the survey data.
- Map Point displays the dialog box in Figure 2.3 showing the spreadsheet names in the Excel workbook. Select one of the spreadsheets.
- Click the **Next** button in the Import Data Wizard. The screen in Figure 2.4 appears showing the columns and the data in the spreadsheet.
- Click the **Finish** button.
- The Data mapping wizard in Figure 2.5 appears. There are some choices here. Do we want to use shaded circles, sized circles or pushpins to mark the points on the Earth? The remaining options pertain to graphing the data. We will discuss graphing later. Let us choose pushpins since they stand out the best.
- Click on the **Pushpin** button.
- Click on the **Next** button at the bottom of the dialog box. The dialog box in Figure 2.6 appears. The user can change the color of the pushpins here, if desired. Map Point automatically recognized the column names `latitude` and `longitude`. This is important because the user would have had to identify these had this not been the case.
- Click on the **Finish** button at the bottom of the dialog box. Map Point plots the data in the spreadsheet on the Earth. Plotted data on the Earth are shown throughout the textbook.

2.3.0.2 Graphing To graph the data, first prepare an Excel file with the survey data. Save the Excel file to the hard drive. Select `Data | Import data wizard`.

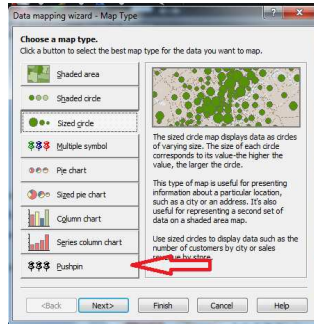


Figure 2.5 This figure shows the Map Point Data mapping wizard.

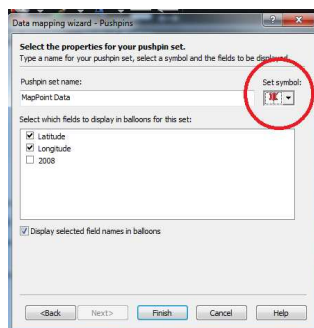


Figure 2.6 This figure shows the Map Point choices for the pushpin colors.



Figure 2.7 This figure shows the Map Point choice for the column chart.

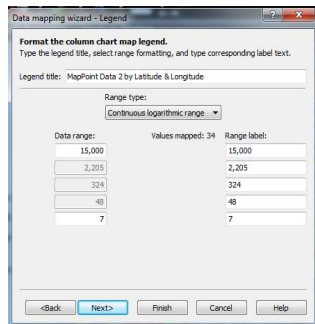


Figure 2.8 This figure shows the Map Point choices for scaling the data and change the label text.

- The dialog box in Figure 2.2 allows the user to browse the hard drive and select the Excel file that contains the data to analyze.
- Map Point displays the dialog box in Figure 2.3 showing the spreadsheet names in the Excel workbook. Select one of the spreadsheets.
- Click the **Next** button in the Import Data Wizard. The screen in Figure 2.4 appears which shows the column names and the data.
- Click the **Finish** button.
- The Data mapping wizard in Figure 2.5 appears. There are some choices here. Do we want our data displayed as a pie chart, sized pie chart, column chart, or as a series column chart? The remaining options pertain to plotting the data. We discussed plotting data in the previous section.
- Click on the **Column chart** button.
- Click on the **Next** button at the bottom of the dialog box. The dialog box in Figure 2.7 appears. Map Point automatically recognized the column names `latitude` and `longitude`. The user must confirm to Map Point to graph the column name `2008`.
- Click the check box next to the field `2008` to confirm graphing this field.
- Click the **Next** button. The dialog box in Figure 2.8 appears. This lets the user scale the data if needed. It also allows the user to change the label text.
- Click on the **Finish** button at the bottom of the dialog box. Map Point displays a graph on the Earth of the data. Short bars represent small values and tall bars represent large values. Map Point places the bars on the Earth using the latitude and longitude coordinates.

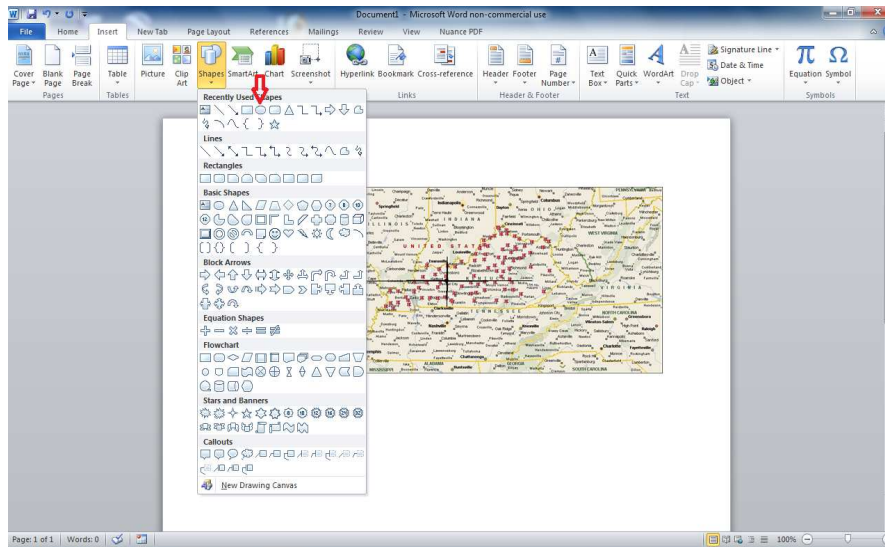


Figure 2.9 This figure shows the Microsoft Word shapes for drawing an ellipse.

2.4 Drawing Axes

Before drawing an axis, it may be necessary to zoom into the survey area. To draw an axis for an ellipse which has an exact length measured in miles or kilometers, select **Tools | Measure distance**. To toggle between miles and kilometers, select **Tools | Options** and the dialog box in Figure 3.1 will appear. The section on **Units** changes the unit of measure.

- Single click on the label box for the mean center. This will anchor the ruler. Map Point displays the unit of measure and number of measurements at the right side of the line.
- Drag the mouse the length required.
- Single click. This will anchor the right end of the line.
- Hit the <Esc> key to stop drawing lines.

2.5 Drawing Ellipse Boundaries

Given that we drew the major axis and the minor axis, and we identified the mean center of the ellipse, it is a simple task to draw the ellipse using MS Word. The steps are as follow:

- Draw the axes described in Section 2.4 using an imaginary *X* and *Y*-axes laid on top of the Earth.

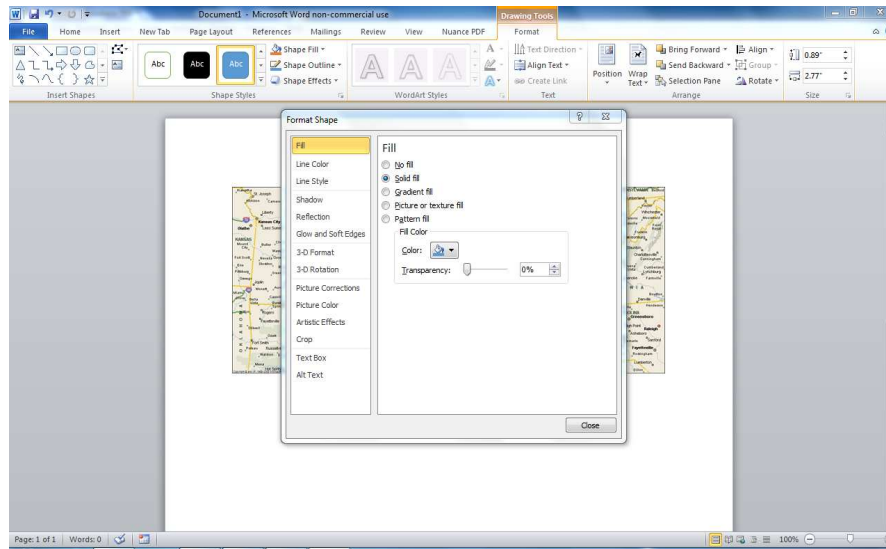


Figure 2.10 This figure shows the Microsoft Office dialog box for formatting a shape.

- Note the axis of rotation (e.g. X or Y) from the statistical computations.
- Copy and paste the image from Map Point into MS Word. The imaginary X and Y -axes form 90° angles. Half of 90° is 45° , and so on. If the rotation is from the Y -axis, then begin counting from there.
- In MS Word, select `Insert` | `Shapes`. The shapes in Figure 2.9 will appear.
- Select the fifth shape from the top, `oval`. Left click and right click around the axes lengths. The oval will be dark filled.
- Right click the oval. Left click the oval and select `Format shape`. The dialog box in Figure 2.10 will appear.
- Click on the `No fill` radio button.
- Click on `Close`.

We drew the ellipse following the steps above. We need to rotate the ellipse. Suppose we want to rotate the ellipse from the Y -axis 70° . Then, follow these steps:

- Click on one of the edges of the ellipse to highlight it.
- A green circle appears at the top of the ellipse. This is for rotating the ellipse. Single click on the green circle.

- Drag the mouse to obtain a 70° angle between the Y -axis and the semi-major axis. Certainly you know where 90° and 45° are at. This is not difficult.
- Select `File | Save As` to save the ellipse.

To avoid the appearance of *cross hairs* in any published graphs, the user can alternatively identify the ends of the major and minor axes using pushpins or some other symbol. When in Map Point, delete any lines drawn when using the ruler.

2.6 Programmer Notes

- Microsoft Corporation did not provide Map Point with a development environment as they did with Excel and MS Word. We need to perform repetitive research manually. The one advantage is that the registered version of Map Point does accept Excel files for plotting and graphing data.
- Not being able to rotate ellipses in Map Point is a draw back. This is just another reason to leave Map Point for another software product.
- Map Point does not geo-code data *and* return the coordinates to the user for computational purposes. This is just another reason to leave Map Point for another software product.

CHAPTER 3

MATHEMATICS REVIEW

3.1 Notation

We discuss some common notation used throughout this book here.

Notation	Meaning
x	Subscript denoting latitude.
y	Subscript denoting longitude.
t	A dummy variable used in integration. Usually denotes the latitude.
u	A dummy variable used in integration. Usually denotes the longitude.
w	Subscript denoting the weight variable.
γ_x	Shape parameter to the Weibull distribution for the latitude.
γ_y	Shape parameter to the Weibull distribution for the longitude.
λ_x	Scale parameter to the Exponential distribution and the Weibull distribution for the latitude. Generally, there is not implied equality.
λ_y	Scale parameter to the Exponential distribution and the Weibull distribution for the longitude. Generally, there is no implied equality.
k	The iteration number in the Secant algorithm.
n	Denotes the sample size .
ϕ_i	Represents the observed values of the latitude in spherical statistics.
θ_i	Represents the observed values of the longitude in spherical statistics.
R	Called the resultant length in spherical statistics.
$\bar{l}_0, \bar{m}_0, \bar{n}_0$	Called the mean direction of cosines in spherical statistics.
S^*	Denotes the spherical variance.
T	Denotes the matrix T . The diagonal elements sum to the sample size n .
B	Denotes the matrix B . The diagonal elements sum to $2n$.

Generally, the set $\{x_1, x_2, \dots, x_n\}$ represents the observed values of the latitude . The set $\{y_1, y_2, \dots, y_n\}$ represents the observed values of the longitude . The set $\{w_1, w_2, \dots, w_n\}$ represents the observed values of the random variable at those corresponding latitude and longitude. Using these three sets of observed numbers (w_i, x_i, y_i) , we estimate parameters from known distributions.

3.2 Derivatives

Some common derivatives used in this text are given next. Using the same notation as in most calculus books and the CRC Standard Math Tables, u represents a function of x , a and n represent fixed real numbers.

Example 1:

$$\frac{d(a)}{dx} = 0.$$

Example 2:

$$\frac{d \log_e n}{dx} = 0.$$

Example 3:

$$\frac{d \log_e u}{dx} = \frac{1}{u} \frac{du}{dx}.$$

Example 4:

$$\frac{d e^u}{dx} = e^u \frac{du}{dx}.$$

Take the derivative of $f(w, \lambda, x) = w \frac{1}{\lambda} e^{-\frac{x}{\lambda}}$ with respect to x .

$$\frac{d w \frac{1}{\lambda} e^{-\frac{x}{\lambda}}}{dx} = -w \left(\frac{1}{\lambda} \right)^2 e^{-\frac{x}{\lambda}}$$

Example 5:

$$\frac{d x^a}{da} = x^a \log x.$$

We are not asking for the derivative with respect to x in this example.

3.3 Integrals

Some common integrals used in this text are given in this section.

Example 6:

$$\int_0^x a \, du = ax.$$

Example 7:

$$\int_0^a \ln n \, dx = x \ln n \Big|_0^a = a \ln n.$$

Example 8:

$$\int \frac{1}{x} \, dx = \log x$$

Example 9:

$$\int e^{ax} = e^{ax}/a$$

In this textbook, sometimes the symbol $\exp\{\}$ represents the exponential function instead of e .

3.4 Quadratic Equation

Most of the material in this section appears in CRC's Standard Math Tables. A quadratic equation has the form $ax^2 + bx + c = 0$. Then, the roots are

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

If a , b , and c are real, then

1. If $b^2 - 4ac$ is positive, then the roots are real and unequal.
2. If $b^2 - 4ac$ is zero, then the roots are real and equal.
3. If $b^2 - 4ac$ is negative, then the roots are imaginary and unequal.

The roots to the quadratic equation give the equation for the standard deviational ellipse. Cases (1) and (2) have more practicable importance than (3). We will discuss the standard deviational ellipse and other ellipses in later Chapters.

3.5 Ellipse Equation

The elliptical formula appears in Equation (3.1).

$$\frac{(x - h)^2}{a^2} + \frac{(y - k)^2}{b^2} = 1, \quad a > b > 0 \quad (3.1)$$

It may be necessary to complete the quadratic equations in the numerators to obtain an ellipse.

3.6 Trigonometry Functions and Conversion Factors

In Excel, the $\text{Atn}()$ function is used to obtain the argument x from the tangent function $\text{Tan}(x)$. The function takes degrees as input, but returns radians as output. Given that, it is convenient to know the following conversion factors:

- To convert radians to degrees,

$$1 \text{ radian} = \frac{180}{\pi} = 57.2957795 \text{ degrees.}$$

- To convert degrees to radians,

$$1 \text{ degree} = \frac{\pi}{180} = 0.0174532925 \text{ radians.}$$

3.7 Modulo Arithmetic

Modulo arithmetic arises when we convert spherical data to directional data. The spherical data comes from the observed latitude and longitude for a random event. To apply some statistical techniques, we need to convert the longitude to a circular reference system, which results in directional data. To accomplish this, we use modulo arithmetic.

To convert the latitude coordinates x_i to the circular reference system x'_i we use the general rule:

$$x'_i = \text{mod}(x_i, 180). \quad (3.2)$$

To convert the summary statistic for the mean latitude \bar{x} back to the original coordinate system, we use the general rule:

$$\bar{x}' = \begin{cases} \bar{x}, & \text{if } \bar{x} \leq 90^\circ. \\ -\text{mod}(180, \bar{x}), & \text{if } \bar{x} > 90^\circ. \end{cases} \quad (3.3)$$

The reader should not worry about how to calculate \bar{x} in this Chapter. Realize that when samples with mixed signs for the latitude and longitude arise, we must have a mechanism to determine the sign of the summary statistics.

Example 10: The coordinates of Australia are $(x_i, y_i) = (25^\circ 16' 27.84 \text{ South}, 133^\circ 46' 30.49 \text{ East})$. The decimal degree coordinates are $(x_i, y_i) = (-25.47399, 133.775136)$.

To convert the coordinates to decimal degrees in Map Point or Google Earth:

- Select Tools | Options.
- If using Map Point, the dialog box in Figure 3.1 will appear. If using Google Earth, the dialog box in Figure 3.2 will appear.
- Click on the radion button `Decimal degrees`.

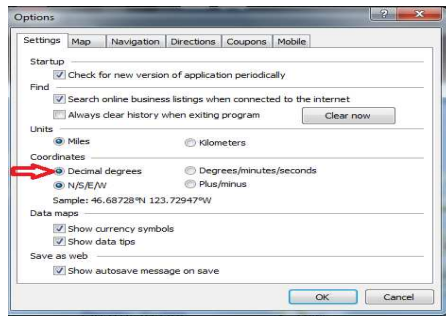


Figure 3.1 This figure shows the Options dialog box in Map Point. We use it to convert the latitude and longitude coordinates to decimal degrees.

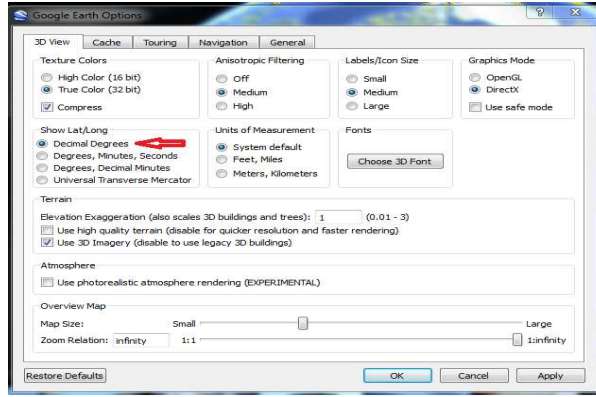


Figure 3.2 This figure shows the Options dialog box in Google Earth. We use it to convert the latitude and longitude coordinates to decimal degrees.

The latitude is negative. The longitude is positive. We convert the latitude using modulo arithmetic. The new latitude observation becomes $x'_i = \text{mod}(-25.47399, 180) = 154.52601$. For computation purposes, we use the coordinates $(x_i, y_i) = (154.52601, 133.775136)$.

To convert the longitude coordinates y_i to the circular reference system y'_i we use the general rule:

$$y'_i = \text{mod}(y_i, 360). \quad (3.4)$$

To convert the summary statistic for the mean longitude \bar{y} back to the original coordinate system, we use the general rule:

$$\bar{y}' = \begin{cases} \bar{y}, & \text{if } \bar{y} \leq 180^\circ. \\ -\text{mod}(360, \bar{y}), & \text{if } \bar{y} > 180^\circ. \end{cases} \quad (3.5)$$

Example 11: The coordinates of Canada are $(x_i, y_i) = (55^\circ 56' 18.59 \text{ North}, 106^\circ 20' 48.37 \text{ West})$. The decimal degree coordinates are $(x_i, y_i) = (55.938497, -106.346771)$. The longitude is negative. The latitude is positive. We convert the longitude using modulo arithmetic. The new longitude becomes $y'_i = \text{mod}(-106.346771, 360) = 253.653229$. For computation purposes, we use the coordinates $(x_i, y_i) = (55.938497, 253.653229)$.

3.8 Weighted Data

Most of the chapters deal with weighted data. What is the motivation for weighting the latitude and longitude observations? Consider the case of un-weighted observations. For example, the Bureau of Justice Statistics (BJS) conducts a survey to measure violent crime in the United States. Hypothetically, let us say that the lowest level of geography reported by the BJS in the survey is at the state level. To find the center of gravity of violent crime, we take the average of the latitude observations

and the average of the longitude observations. Averages are un-biased estimators. To say that each state contributes equally to the center of gravity is misleading. Some states have a high number of incidents of crime while others have a low number of incidents of crime. We should reflect this fact in the estimator. The literature on weighted data typically multiplies the latitude and longitude by a weight. In this case, the weight is the number of incidents of violent crime divided by the sum of the number of incidents of violent crime for the entire U.S. This ensures the weights are between zero and one. It ensures those states with a high number of incidents of violent crime will have a correspondingly proportional effect on the center of gravity — those states will draw the mean center of gravity to themselves. Vice-versa, those states with a low number of incidents of violent crime will have the mean center of gravity pulled away from them.

Throughout the text book, the examples uses weighted data. Equation (3.6) shows the weighting scheme for observation i out of n observations.

$$w'_i = \frac{w_i}{\sum_{i=1}^n w_i} \quad (3.6)$$

```

1  Sub Set_Weights(sum_weight)
    'this procedure calculates the weights between 0 and 1 in column D
    'columns A, B, and C must be highlighted
    n = Selection.Rows.Count
    sum_weight = 0
    For i = 2 To n
        For j = 2 To 4
            With ActiveSheet
                sum_weight = sum_weight + .Cells(i, j).Value
            End With
        Next j
    Next i
    For i = 2 To n
        For j = 5 To 6
            With ActiveSheet
                .Cells(i, j).Value = .Cells(i, 3).Value / sum_weight
            End With
        Next j
    Next i
End Sub

```

Comments in VBA for Excel begin with a single tick mark `'''`. Loop 2 shows the code for summing the weight column in an Excel spreadsheet. In this case, Column C (referred to as 3) and rows 2 thru n are being summed. The Excel provided function `Selection.Rows.Count` counts the number of rows highlighted in the active spreadsheet. It initializes the value of n . Loop 3 shows the code for the final calculation of the weights. It divides each observation w_i by the sum $\sum_{i=1}^n w_i$ and stores the results in Column D. We do not weight observation i equal to 1 (i.e. row 1 in the active spreadsheet) because it is a label.

Upon running this subroutine, column D will always sum to one. The use of the `WITH-END-WITH` statement is optional. It is a short-cut to not having to keep typing in the Excel object — in this case, the spreadsheet last referenced "Activsheet."

3.9 Area

Consider the simple formula for calculating the area of an ellipse (or a circle). Let a be the length of the semi-major axis. Let b be the length of the semi-minor axis. If $a > b$, then we have an ellipse. If $a = b$, then we have a circle. A function to calculate the area of an ellipse is given next. The parameters to the function are the semi-major axis length a and the semi-minor axis length b . Equation (3.7) gives the formula for the area of an ellipse.

$$f = \pi ab. \quad (3.7)$$

```

1  [Function Area(a, b)
    'a= semi-major axis length
    'b= semi-minor axis length
    2  [Area = WorksheetFunction.Pi * a * b
    End Function

```

Line 2 calculates the area of the ellipse using the formula in Equation (3.7). The value of the calculation is returned via the function name `Area`.

3.10 Eccentricity

Equation (3.8) gives the eccentricity of an ellipse (or circle) where a is the length of the semi-major axis and b is the length of the semi-minor axis, $a \geq b$.

$$e = \frac{\sqrt{a^2 - b^2}}{a} \quad (3.8)$$

```

1  [Function Eccentricity(a, b)
    'a= semi-major axis length
    'b= semi-minor axis length
    [If a >= b Then
    2  Eccentricity = Sqr(a ^ 2 - b ^ 2) / a 'eccentricity
    Else
    Eccentricity = Sqr(b ^ 2 - a ^ 2) / b 'eccentricity
    End If
    End Function

```

Some defensive programming has been incorporated into the `Eccentricity()` function. If a is the semi-major axis, then Equation (3.8) works fine. If the user forgets the variable definitions, then the function would return an error message if only one formula was programmed. If $a < b$, the roles of a and b are switched in Equation (3.8). The IF-THEN-ELSE statement (2) demonstrates the defensive programming concept.

3.11 Axes Length

Equation (3.9) gives the formula for the semi-major axis length a . Equation (3.10) gives the formula for the semi-minor axis length b .

$$a = \sqrt{\frac{y}{n} + \frac{2(xy)^2}{n \left[-(x-y) + \sqrt{(x-y)^2 + 4(xy)^2} \right]}} \quad (3.9)$$

$$b = \sqrt{\frac{x}{n} + \frac{2(xy)^2}{n \left[-(x-y) + \sqrt{(x-y)^2 + 4(xy)^2} \right]}} \quad (3.10)$$

Calculating the values x , y , and xy are complicated sums depending on the probability distribution. The details on calculating the sums will be covered in later Chapters. For now, assume that the variables x , y , and xy are given.

```

Sub Axes_Length(m, x, y, xy, a, b)
  'm= number of observations
  'x= sum of diffs from mean center squared x direction (input)
  'y= sum of diffs from mean center squared y direction (input)
  'xy= diff sums from mean centers x and y direction (input)
  'a= semi-major axis (output)
  'b= semi-minor axis (output)
1  If x >= y Then
    a = Sqr(y / m + 2 * (xy) ^ 2 / (m * (-1 * (x - y) + Sqr((x - y) ^ 2 + 4 * xy ^ 2))))
    b = Sqr(x / m - 2 * (xy) ^ 2 / (m * (-1 * (x - y) + Sqr((x - y) ^ 2 + 4 * xy ^ 2))))
2  Else
    a = Sqr(x / m + 2 * (xy) ^ 2 / (m * (-1 * (y - x) + Sqr((y - x) ^ 2 + 4 * xy ^ 2))))
    b = Sqr(y / m - 2 * (xy) ^ 2 / (m * (-1 * (y - x) + Sqr((y - x) ^ 2 + 4 * xy ^ 2))))
  End If
End Sub

```

Given the quantities x , y , xy , the subroutine `Axes_Length()` returns the length of the semi-major axis a and the length of the semi-minor axis b .

3.12 Rotation

The subroutine `Angle_of_Rotation()` calculates the angle of rotation θ in Equation (5.10). It also gives the axis of rotation. The first `FOR-NEXT` loop (12) calculates the individual sums in Equation (5.10) and stores the values in the variables `x`, `y`, and `xy`. The next four VBA statements calculate the angle of rotation with the plus and minus sign accounted for. The angle of rotation θ is stored in the variables `theta1` and `theta2`. The `IF-THEN-ELSE` statement (13) determines the axis of rotation using the values `theta1` and `theta2`. If `theta1 > theta2` then the axis of rotation is the Y-axis because the square root term is positive. Otherwise, the square root term is negative (or equal) and the axis of rotation is the X-axis. The `WITH-END-WITH` statement (14) puts the results onto the spreadsheet called `STATS`.

```

Sub Angle_of_Rotation(x, y, xy, note, atheta, itheta)
'calculate the rotation from the axes
'x= sum of diffs from mean center squared x direction (input)
'y= sum of diffs from mean center squared y direction (input)
'xy= diff sums from mean centers x and y direction (input)
'note= states direction of rotation (output)
'atheta= angle of rotation in the x direction (output)
'itheta= angle of rotation in the y direction (output)

tan_theta1 = -1 * (x - y) / (2 * xy) + Sqr((x - y) ^ 2 + 4 * xy ^ 2) / (2 * xy)
tan_theta2 = -1 * (x - y) / (2 * xy) - Sqr((x - y) ^ 2 + 4 * xy ^ 2) / (2 * xy)
1 theta1 = Atn(tan_theta1) * 180 / WorksheetFunction.Pi()
  theta2 = Atn(tan_theta2) * 180 / WorksheetFunction.Pi()
    If theta1 > theta2 Then
      note = "Rotate on Y-Axis"
      atheta = theta1
      itheta = theta2
    2 Else
      note = "Rotate on X-Axis"
      atheta = theta2
      itheta = theta1
    End If
End Sub

```

3.13 Base 10 Logarithm

The VBA logarithm function has a natural exponential base. VBA for Excel does not provide a means to change the base of the logarithm. If the programmer wishes to change the base, then the programmer must write a function for that particular logarithm with that particular base. The static function `Log10(X)` returns the logarithm of the number X with the base 10.

```

1  [ Static Function Log10(X)
    [ Log10 = Log(X) / Log(10#)
    [ End Function

```


CHAPTER 4

GEO-CODING

The data used in this book comes from many sources. The majority of the data resides on government web sites today. It would have been nice if the data were prepared before downloading it. The data provided (free) did have the random variables of interest. However, the data did not have the latitude and longitude observations. Thus, the data must be *geo-coded* before performing any analyzes in this textbook.

Existing textbook authors use either outdated software or expensive software to calculate their statistics. Outdated software such as FORTRAN 77 and 10 Statement Fortran has been around since at least the 1960's. Arc GIS and Map Soft are modern Geographic Software Information systems that can geo-code data; however, products such as those are expensive. We can always geo-code the data free on Google Earth — but this becomes time consuming. Academic software products do exist such as *CrimeStat*. It performs many statistical calculations including weighted calculations. That one comes with an extensive software manual, too. Map Point, a Microsoft product, is fairly inexpensive. The user can save work to the hard drive. The user can copy and paste maps into other Microsoft products to draw the ellipse and to rotate it. [Hammond (26)] has a thirty-one page printed "Student Notebook Atlas" of the world. The atlas has the land masses enclosed in a box with the latitude and longitude references on the sides of the box. The Hammond Atlas is an inexpensive

product, too. It may be difficult to get the same precise coordinates that the software products give.

It is a trade-off among data precision, time, cost, and scope of the research as to what software one chooses. Data is data — of course, obtain it in the most pristine condition possible before performing the research.

Do we need the software or data provider to geo-code the data and calculate the statistics or just geo-code the data? In this case, we just want the data geo-coded. We will write the code for calculating the statistics for the various ellipses in Visual Basic for Applications (VBA) in Excel 2010. This application is further accessible on Android tablets and smart phones using *Quick Office*.

Most of the geo-coding in this textbook was done using Google Earth. Begin by downloading the software at <http://www.google.com/earth/index.html>. Google Earth periodically updates their images of the Earth. This causes slight differences in latitude and longitude readings. As long as the data is geo-coded under one or another satellite images, it does not alter the survey data or the statistical techniques presented in this textbook. Figure 4.1 shows Google Earth. As an example, for the June Area Survey in Kentucky, the user types-in the county and state in the upper-left corner of Google Earth. Google Earth returns the decimal degrees of the county and state in the lower-right corner.

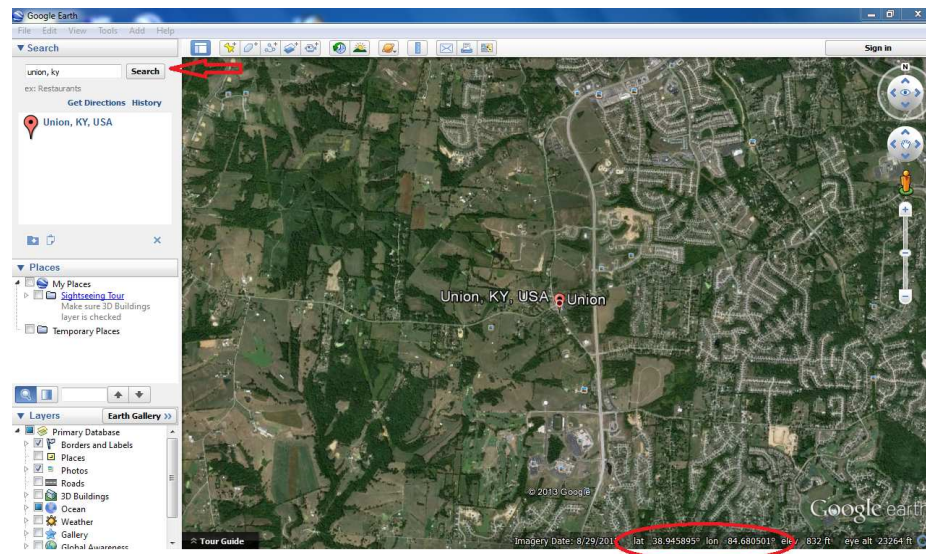


Figure 4.1 This figure shows Google Earth. It shows how to geo-code a county within a state.

Similarly, the user can type in a state name to obtain the geo-coding or the user can type in a country name such as Canada to obtain the geo-coding. It is important not to zoom into or out of the Earth when collecting these coordinates. Otherwise,

the altitude (which is not being collected) will change. This in turn will change the latitude and longitude observations.

4.1 Kentucky June Area Survey

This is a survey conducted by the US Department of Agriculture (USDA). Within the USDA, the National Agricultural Statistics Service (NASS) supervises and oversees the data collection and reporting responsibilities for this survey. Plenty of data is available and downloadable that is ready to use with little or no formatting issues.

Experience says the Area Frame Section at USDA / NASS uses a stratified survey design to obtain population totals, standard errors, and CVs. The June Area Survey is conducted every year to measure, among other things, corn acreage, cotton acreage, soybean acreage, three seasons of wheat acreage, NOL cattle, and the number of farms in the United States. Each state has a set of land use strata specific to it. Within each strata, replicates are assigned to sub-strata. A sample of land called a segment is finally sampled. Since a minimum of two replicates is assigned to the sub-strata, the minimum number of segments per strata is two (assuming one sub-strata). Segment sizes range from 1/10th of a square mile to 1 square mile (640 acres per mile). Often, multiple sub-strata are assigned to each stratum. A complete, comprehensive set of designs for the United States can be found in the Area Frame Design Books.

[Kelly (32)] describes the June Area Survey conducted by the USDA / National Agricultural Statistics Service. That paper describes techniques intended to identify differences in variation, not to visualize them. Given that the spatial data is available and the computations described so far, it is possible to visualize the variation in the data. The author draws the sample from a sample of land called a segment. Segment sizes range from 1/10th of a square mile to 1 square mile (640 acres per mile). Enumerators completely cover each segment during the survey. Cartographers and survey researchers know every segments center point coordinates. Instead of using the simplistic weighted estimators for the total crop acreage where the weights are simply the total number of segments divided by the number of sampled segments, the alternative weights will be defined using spatial data. We will define alternative weights using spatial data, instead of using the simplistic weighted estimators for the total crop acreage where the weights are simply the total number of segments divided by the number of sampled segments. From these alternative weights, we can derive what are called what are called *weighted mean centers* and *weighted standard deviational ellipses*.

We obtain the data in this book by augmenting the NASS data available to the public with Google Earth¹ coordinates for each county. Data consumers can easily download the crop data. Figure 4.2 shows the data downloaded from the NASS web site. Regional level observations have arrows identifying them. The yellow arrows point to the following formatting issues for us.

¹The web site is www.earthgoogle.com. Decimal coordinates are preferred since we can more easily integrate them than degrees, minutes, and seconds.

- A. The arrow points to non-geographical data. For analytical purposes, we delete non-geographical observations.
- B. The arrow points to the sum of a region. For analytical purposes, we delete regional observations.

Obtaining the latitude and longitude for the counties does take time. Given that, our sample will be the county-level data in Kentucky for corn for grain. We are not so much interested in the quantity of acreage of corn grown in June for the state. *We are interested in the location that corresponds to that single quantity for the state.* Hence, we augment the data with latitude and longitude coordinates.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
1	Commodity	Practice	Year	State	cofips	StFips	District	county	CommCode	PracCode	Planted	All Purposes	Harvested	Yield	Production
2	Corn For Grain	Total For Crop	2003	Kentucky	Ballard	21	10	7	11199199	9	26100	25500	128	326400	
3	Corn For Grain	Total For Crop	2003	Kentucky	Calloway	21	10	35	11199199	9	38300	37600	112	421120	
4	Corn For Grain	Total For Crop	2003	Kentucky	Carlisle	21	10	39	11199199	9	23100	22700	141	320070	
5	Corn For Grain	Total For Crop	2003	Kentucky	Fulton	21	10	75	11199199	9	26300	24200	144	348480	
6	Corn For Grain	Total For Crop	2003	Kentucky	Graves	21	10	83	11199199	9	60400	59400	131	778140	
7	Corn For Grain	Total For Crop	2003	Kentucky	Hickman	21	10	105	11199199	9	38800	37000	138	510600	
8	Corn For Grain	Total For Crop	2003	Kentucky	Livingston	21	10	139	11199199	9	7000	6600	114	75240	
9	Corn For Grain	Total For Crop	2003	Kentucky	Lyon	21	10	143	11199199	9	6000	5700	132	75240	
10	Corn For Grain	Total For Crop	2003	Kentucky	McCracken	21	10	145	11199199	9	12500	12000	117	140400	
11	Corn For Grain	Total For Crop	2003	Kentucky	Marshall	21	10	157	11199199	9	9300	8700	100	87000	
12	Corn For Grain	Total For Crop	2003	Kentucky	Trigg	21	10	221	11199199	9	21300	21100	126	265860	
13	Corn For Grain	Total For Crop	2003	Kentucky	D10 Purchase	21	10	999	11199199	9	269100	260500	128.5	3348550	
14	Corn For Grain	Total For Crop	2003	Kentucky	Caldwell	21	20	33	11199199	9	24400	23700	120	284400	
15	Corn For Grain	Total For Crop	2003	Kentucky	Christian	21	20	47	11199199	9	75200	73700	144	1061280	
16	Corn For Grain	Total For Crop	2003	Kentucky	Crittenden	21	20	55	11199199	9	9000	8400	124	104160	
17	Corn For Grain	Total For Crop	2003	Kentucky	Davess	21	20	59	11199199	9	56000	54700	139	760330	
18	Corn For Grain	Total For Crop	2003	Kentucky	Hancock	21	20	91	11199199	9	6500	6500	118	76700	
19	Corn For Grain	Total For Crop	2003	Kentucky	Henderson	21	20	101	11199199	9	63900	61800	140	865200	
20	Corn For Grain	Total For Crop	2003	Kentucky	Hopkins	21	20	107	11199199	9	27100	26900	135	363150	
21	Corn For Grain	Total For Crop	2003	Kentucky	Logan	21	20	141	11199199	9	56900	55000	168	924000	
22	Corn For Grain	Total For Crop	2003	Kentucky	McLean	21	20	149	11199199	9	34800	34500	153	527850	
23	Corn For Grain	Total For Crop	2003	Kentucky	Muhlenberg	21	20	177	11199199	9	12100	11900	123	146370	
24	Corn For Grain	Total For Crop	2003	Kentucky	Ohio	21	20	183	11199199	9	24300	24000	141	338400	
25	Corn For Grain	Total For Crop	2003	Kentucky	Simpson	21	20	213	11199199	9	36100	35100	163	572130	
26	Corn For Grain	Total For Crop	2003	Kentucky	Todd	21	20	219	11199199	9	44700	43200	151	652320	
27	Corn For Grain	Total For Crop	2003	Kentucky	Union	21	20	225	11199199	9	77300	75300	139	1046670	
28	Corn For Grain	Total For Crop	2003	Kentucky	Webster	21	20	233	11199199	9	30800	30400	147	446880	
29	Corn For Grain	Total For Crop	2003	Kentucky	D20 Midwestern	21	20	999	11199199	9	579100	565100	144.6	8169840	
30	Corn For Grain	Total For Crop	2003	Kentucky	Adair	21	30	1	11199199	9	7800	4500	83	37350	
31	Corn For Grain	Total For Crop	2003	Kentucky	Allen	21	30	3	11199199	9	3000	2700	147	39690	
32	Corn For Grain	Total For Crop	2003	Kentucky	Barren	21	30	9	11199199	9	15700	9500	135	128250	

Figure 4.2 This figure shows the Excel spreadsheet of the Kentucky data. The spreadsheet has not been geo-coded. It still contains the regional data.

The data lacks latitude and longitude observations. These observations are obtainable from Google Earth. Since, the geography attached to the observations from USDA are at the county level, the most precision we can hope for from Google Earth is the latitude and longitude of the counties in the reported data for the given counties.

4.2 US Crime Statistics

The Department of Justice established the Bureau of Justice Statistics (BJS) on December 27, 1979, under the Justice Systems Improvement Act of 1979 as an amendment to the Omnibus Crime Control and Safe Streets Act of 1968. According to the Bureau of Justice Statistics's website, their mission is to collect, analyze, publish, and disseminate information on crime, criminal offenders, victims of crime, and the operation of justice systems at all levels of government.

This data appears on the Fed Stats web site. Figure 4.3 shows the downloaded data from the Department of Justice. Examining the left column (Column A), some of the rows are summarized to the regional level. The data provider merged the headings across the top. The data include Puerto Rico and Hawaii. These states and islands are disconnected from the US. We are interested in the main-land part of the United States.

	A	B	C	D	E	F	G	H	I	J	K	L	M
				Violent crime		Murder and negligent manslaughter		Forcible rape		Robbery		Aggravated assault	
				Number	Rate per 100,000	Number	Rate per 100,000	Number	Rate per 100,000	Number	Rate per 100,000	Number	Rate per 100,000
4	Area	Year	Population ¹	Number	Rate per 100,000	Number	Rate per 100,000	Number	Rate per 100,000	Number	Rate per 100,000	Number	Rate per 100,000
7	United States Total ^{2,3,4}	2007	301,621,157	1,408,337	466.9	16,929	5.6	90,427	30.0	445,125	147.6	855,856	283.8
8		2008	304,059,724	1,382,012	454.5	16,272	5.4	89,000	29.3	441,855	145.3	834,885	274.6
9		Percent change		-1.9	-2.7	-3.9	-4.7	-1.6	-2.4	-0.7	-1.5	-2.5	-3.2
10	Northeast	2007	54,680,626	203,632	372.4	2,261	4.1	10,821	19.8	75,326	137.8	115,224	210.7
11		2008	54,924,779	203,654	370.8	2,293	4.2	10,981	20.0	76,048	138.5	114,332	208.2
12		Percent change		*	-0.4	+1.4	+1.0	+1.5	+1.0	+1.0	+0.5	-0.8	-1.2
13	New England	2007	14,264,185	43,334	303.8	357	2.5	3,395	23.8	12,225	85.7	27,357	191.8
14		2008	14,303,542	46,682	326.4	380	2.7	3,580	25.0	12,696	88.8	30,026	209.9
15		Percent change		+7.7	+7.4	+6.4	+6.1	+5.4	+5.2	+3.9	+3.6	+9.8	+9.5
16	Connecticut	2007	3,502,309	8,965	256.0	106	3.0	658	18.8	3,607	103.0	4,594	131.2
17		2008	3,501,252	10,427	297.8	123	3.5	674	19.3	3,907	111.6	5,723	163.5
18		Percent change		+16.3	+16.3	+16.0	+16.1	+2.4	+2.5	+8.3	+8.3	+24.6	+24.6
19	Maine	2007	1,317,207	1,554	118.0	21	1.6	391	29.7	349	26.5	793	60.2
20		2008	1,316,456	1,547	117.5	31	2.4	375	28.5	333	25.3	808	61.4
21		Percent change		-0.5	-0.4	+47.6	+47.7	-4.1	-4.0	-4.6	-4.5	+1.9	+1.9
22	Massachusetts	2007	6,449,755	27,832	431.5	184	2.9	1,634	25.3	7,006	108.6	19,008	294.7
23		2008	6,497,967	29,174	449.0	167	2.6	1,736	26.7	7,069	108.8	20,202	310.9

Figure 4.3 This figure shows the Excel spreadsheet of the crime data. The spreadsheet has not been geo-coded and still contains the regional data, two years of data in the same spreadsheet, merged columns, and percentages.

To address cleaning the data from this spreadsheet will take more work than that for the Kentucky spreadsheet. This data contains merged cells, regional data and two years of data in the same spreadsheet. In addition, we are not interested in the percentages. We must contend with some of the problems indicated by the yellow arrows. Obviously, the data provider created this spreadsheet for presentation purposes, not analytical. The yellow arrows point to the following formatting issues for us.

- A. The arrow points to the sum of the U.S. totals for the U.S. This is the first clue that other summary data exists in the spreadsheet. We need to delete summary data.

- B. The arrow points to multiple years of data in the same spreadsheet. We need to separate this into two different spreadsheets for the algorithms in the VBA code to work properly.

- C. The arrow points to a percentage change. Although useful for a presentation, we do not need the percentage change for any analyzes in this book. We need to delete percentage change data.

4.3 OECD Countries

The Organization for Economic Cooperation and Development (OECD) has its headquarters in Paris, France; the organization produces 250 publications per year. Europe created OECD as a successor to the Marshall Plan for the reconstruction of Europe after World War II. OECD countries are committed to democracy and the market economy. This data comes from the *Organization for Cooperation and Economic Development* website at www.oecd.org. Obviously, some very large economies are missing from the data. Non-OPEC countries such as India is missing. Most OPEC countries (Saudi Arabia, Egypt, Iran, Libya, etc.) are missing. This data does provide a global aspect for the computations in this textbook. We must perform additional pre-computations on the latitude and longitude before performing any proposed calculations.

We need knowledge of the bounds on the latitude and longitude before modeling this data set. In prior data sets, the longitude was consistently negative. We can simply change the sign to positive, and then carry out certain computations. This is not the case with the OECD data. It has mixed signs for both the latitude and the longitude. See Figure 4.4. An understanding of the Prime Meridian and the Equator measurements helps. Very little formatting is necessary compared to the US Crime data to use it "as-is."

Consider the following mapping for the latitude $[-90^\circ, 90^\circ]$.

A	B	F	G	H	I	J
1	Economics: Key tables from OECD - ISSN 2074-384x - © OECD 2011					
2	Gross domestic product in US dollars					
3	At current prices and current PPPs					
4	Billions of US dollars					
5		2006	2007	2008	2009	2010
6	Australia	774.3	828.2	847.2	876.5 e	901.5 e
7	Austria	299.9	313.8	332.2	324.7	334.5
8	Belgium	360.1	378.1	394.9	391.8	405.9
9	Canada	1 200.5	1 263.0	1 295.9	1 275.6	1 331.3 e
10	Chile	214.2	230.7	244.2	239.2	253.8
11	Czech Republic	229.5	253.7	269.6	268.2	274.1 e
12	Denmark	195.9	206.0	216.9	208.1	215.5
13	Estonia	25.7	28.5	29.2	26.6	28.3
14	Finland	174.3	191.2	200.8	188.1	196.7
15	France	2 000.2	2 123.9	2 195.7	2 173.3	2 232.7
16	Germany	2 776.8	2 930.5	3 052.5	2 975.3	3 108.4
17	Greece	302.1	316.2	338.0	330.7	316.7
18	Hungary	184.6	192.9	207.8	203.3	210.4
19	Iceland	10.9	11.6	12.5	11.7	11.5
20	Ireland	180.1	197.7	189.5	176.8	175.1
21	Israel (1)	176.1	190.9	202.3	205.8	218.0 e
22	Italy	1 781.5	1 893.9	1 990.5	1 951.5	1 995.2
23	Japan	4 071.4	4 290.2	4 322.9	4 135.2 e	4 330.3 e
24	Korea	1 173.0	1 269.1	1 306.4	1 321.0	1 423.7 e
25	Luxembourg	37.1	40.6	43.8	42.2	45.9
26	Mexico	1 432.2	1 540.9	1 629.6	1 540.4	1 647.2 e

Figure 4.4 This figure shows the Excel spreadsheet of the Gross Domestic Product data. The spreadsheet has not been geo-coded.

Latitude	$\text{mod}(x_i, 180)$
10	10
20	20
30	30
40	40
50	50
60	60
70	70
80	80
90	90
0	0
-10	170
-20	160
-40	140
-50	130
-60	120
-80	100

For negative values of the latitude, the conversion has unique, positive latitude values. We can do the same for the longitude using modulo 360. When we interpret the results, the mean latitude and longitude must be converted back to the original coordinate system.

Longitude	$\text{mod}(y_i, 360)$
10	10
40	40
80	80
120	120
160	160
180	180
-10	350
-40	320
-80	280
-120	240
-160	200

4.4 Coordinate Systems

The Cartesian coordinate system is the classical coordinate system taught in most textbooks. In two dimensions, we have an X -axis and a Y -axis. This is a single pole coordinate system with $(0, 0)$ at the center. In three dimensions, we have the additional Z -axis — still a single pole coordinate system.

We can alternatively define the function $\mathbf{r} = \mathbf{i}x + \mathbf{j}y$, where \mathbf{i} and \mathbf{j} are vectors defined as:

$$\mathbf{i} = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

and

$$\mathbf{j} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

In the spherical statistics literature, a one-pole coordinate system appears most often because these coordinate systems have the most practical applications and interpretations. Suppose we are measuring the hands on a clock, a roulette wheel, an experiment where a person in a room must identify the source of a sound, and so on. All of these examples have one common theme in that they have one pole. The center of the clock is the pole. The center of the roulette wheel is the pole. The person's head is the pole. [Leong and Carlile (38)] describe the *hoop* coordinate system, which has two dimensions and one pole. They mention that data is restricted to observations lying on the sphere as opposed to measuring distance (i.e. the sound experiment).



Figure 4.5 This figure shows the Earth using Map Point. We identify points on the Earth using latitude and longitude.

Exact formulas exist for converting from the polar coordinate system to the Cartesian coordinate system (and vice-versa). For most problems of practicable importance from these two coordinate systems, we restrict the space to one-dimension or two-dimensions so that it becomes a *circular coordinate system*. Otherwise, interpreting the results becomes difficult.

Another set of practicable problems use latitude and longitude coordinates. We identify points on the Earth using latitude and longitude. See Figure 4.5. We look-up the latitude and longitude coordinates for our data. This is called *geo-coding*. The latitude ranges from $(-90^\circ, 90^\circ)$. The longitude ranges from $(-180^\circ, 180^\circ)$. We can always convert geographic data to circular data. The mean results can be converted back to the original coordinate system while preserving the sign. We must interpret the results such as the mean center, which is also called the *center of gravity*. Using the calculations from the spherical statistics literature, these calculations strive to project the data onto an X , Y and Z -axis as opposed to the Earth. Using the calculations from the geography and statistics literature, the center of gravity will always be in the survey area. Some careful forethought goes into what you expect from spherical statistics.

Many times the data in this textbook comes from several sources. The random variable w_i augments with a pair of coordinates called latitude x_i and longitude y_i for each observation i . We omit the altitude from the analysis. Using special software, we plot the random variable on a map using the pair (x_i, y_i) . To make calculations simple, we measure the latitude and longitude pairs (x_i, y_i) in *degrees*, then *decimal minutes and seconds* as opposed to degrees, minutes, and seconds. Having knowledge of the Cartesian coordinate system for plotting points does help to understand the subtle differences between mapping and plotting.

When drawing an ellipse on a map, knowledge of the polar coordinate system is most useful. We derive several pieces of information:

- The center of the ellipse.
- The semi-major and the semi-minor axis lengths.

- The angle of rotation from an imaginary X or Y -axis.

We have enough information to draw an ellipse and to rotate the ellipse from the imaginary axis. The imaginary axis (or axes) is simply the X and Y -axes from the Cartesian coordinate system super-imposed onto a map using specialty software. Chapter 2 discusses some of the specialty software available.

- The software comes with a ruler to draw lines. The software can measure lines in miles or kilometers. We use the ruler to draw the semi-major and semi-minor axes.
- The user must use another software product to draw and rotate the ellipse.

CHAPTER 5

STANDARD DEVIATIONAL ELLIPSE

Textbooks on geo-statistics cover the weighted mean center and the dispersion of the data, [Davis, (12), p. 25]. The author defines the weighted mean center used in Section 5. The author also defines the accompanying standard distance based on Pythagoras' theorem for a set of coordinates (x_i, y_i) as

$$\sqrt{\frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 + \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2}. \quad (5.1)$$

The author objectively compares the standard distance of the dispersion of different point patterns. to be compared objectively. He gives as an example the number of clothing shops versus butchers or grocers. The standard distance is a single number and always results in a circle around the mean center to show the dispersion. [Gregory (22), and Lee (35)] discuss data transformations and regression estimators. To derive the estimators for the weighted mean centers, we can use a general linear means model. [Johnston, (31), p. 130-34] discusses the relationship between the standard deviational ellipse and the correlation coefficient in linear regression modeling. The author provides details to graphical representations in lieu of proofs.

We will introduce an extension to the standard deviational ellipse and use the above three criteria given in [Yuill, (67)] to compare the results. This book will consider these two spatial estimators:

1. The weighted mean center — shows the spatial mean of a random variable.
2. The weighted standard deviational ellipse — shows the spatial spread of a set of point locations for a random variable.

The *mean center* is simply the usual sample mean found in the statistics literature. For the *weighted mean center* found in the geo-statistics literature, the formulas and their distributional properties are neither too obvious nor intuitive. We will discuss those concepts next.

5.1 A Weighted Regression Model

Consider the weighted regression model $y_i = \alpha + \beta x_i + \epsilon_i$ where $\epsilon_i \sim N(0, \frac{1}{\sqrt{w_i}}\sigma)$ where w_i represents the weighted observations. [Lee (35), page 236] gives the following estimators.

The parameter estimator for α appears in Equation (5.2).

$$\hat{\alpha} = \bar{y} - \hat{\beta}\bar{x}. \quad (5.2)$$

The parameter estimator for β appears in Equation (5.3).

$$\hat{\beta} = \frac{\sum_{i=1}^n w_i(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n w_i(x_i - \bar{x})^2} = \frac{\sum_{i=1}^n w_i x_i y_i - \frac{(\sum_{i=1}^n w_i x_i)(\sum_{i=1}^n w_i y_i)}{\sum_{i=1}^n w_i}}{\sum_{i=1}^n w_i x_i^2 - \frac{(\sum_{i=1}^n w_i x_i)^2}{\sum_{i=1}^n w_i}} \quad (5.3)$$

where the averages are estimated using Equations (5.4) and (5.5).

$$\bar{x} = \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i} \quad (5.4)$$

$$\bar{y} = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i} \quad (5.5)$$

Under the normal distribution, the weighted residual sums of squares for σ^2 appears in Equation (5.6).

$$\sum_{i=1}^n \hat{\epsilon}_i^2 = \sum_{i=1}^n w_i (y_i - \hat{\alpha} - \hat{\beta}x_i)^2. \quad (5.6)$$

Equation (5.7) shows the weighted likelihood function for the regression model. It has a chi-square distribution.

$$L = \prod_{i=1}^n \left(\frac{1}{2\pi\hat{\epsilon}^2} \right)^{\frac{1}{2}} \exp \left\{ - \sum_{i=1}^n \frac{w_i(y_i - \hat{\alpha} - \hat{\beta}x_i)^2}{2\hat{\epsilon}^2} \right\} \quad (5.7)$$

The natural logarithm of the weighted likelihood function appears in Equation (5.8).

$$\ln L = \ln \left[\prod_{i=1}^n \left(\frac{1}{2\pi\hat{\epsilon}^2} \right)^{\frac{1}{2}} \right] - \frac{1}{2} \sum_{i=1}^n \frac{w_i(y_i - \hat{\alpha} - \hat{\beta}x_i)^2}{\hat{\epsilon}^2}. \quad (5.8)$$

Equation (5.8) is based on the Akaike information criterion. The entire equation is retained for comparison purposes in Section 7.13.

5.2 Mean Latitude and Mean Longitude

Let $(w_i x_i, w_i y_i)$ equal to a weighted, paired observation. Let w_i equal to the weight for the sample total of a random variable on observation i . Let x_i equal to the latitude of the i^{th} observation such that $0 \leq x_i \leq 90$. Let y_i equal to the longitude of the i^{th} observation such that $-180 \leq y_i \leq 180$.

$$(\bar{x}, \bar{y}) = \left(\frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i}, \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i} \right). \quad (5.9)$$

Equation (5.9) is called the *center of gravity*.

5.3 VBA

```

Sub SDEmean_centers()
'calculate the mean center for the latitude and the longitude

mean_latitude = 0
mean_longitude = 0
n = Selection.Rows.Count

  For i = 2 To n
    With ActiveSheet
      mean_latitude = mean_latitude + .Cells(i, 1).Value * .Cells(i, 3).Value
      mean_longitude = mean_longitude + .Cells(i, 2).Value * .Cells(i, 3).Value
    End With
  Next

mean_latitude = mean_latitude / sum_weight
mean_longitude = mean_longitude / sum_weight

'user must convert latitude and longitude back from circular coordinates
'put the values in STATS worksheet

End Sub

```

To calculate the mean center of gravity in Equation (5.9), the weights must be set on the active spreadsheet first. The subroutine named `SDEmean_centers()` calculates the weighed mean center of the standard deviational ellipse. The `FOR-NEXT` loop (10) multiplies the latitude and longitude observations by the weights. It sums the observations and puts the sums into the variables `mean_latitude` and `mean_longitude`. The `WITH-END-WITH` statements (11) puts the results into the output spreadsheet called `STATS`. The cell reference is always row, then column. For example, the code `.Cell(3, 2)` refers to row 3 and column 2. Column 2 refers to column B.

5.4 Standard Deviational Ellipse

To calculate the standard deviational ellipse, one must substitute the latitude and longitude coordinate pairs as follow.

$$x'_i = x_i - \bar{x}, \quad y'_i = y_i - \bar{y}.$$

[Yuill (67)] calculates the angle of rotation for the *weighted standard deviational ellipse* as

$$\tan \theta = -\frac{\left(\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i\right)}{2 \sum_{i=1}^n x_i' y_i' w_i} \pm \frac{\sqrt{\left(\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i\right)^2 + 4 \left(\sum_{i=1}^n x_i' y_i' w_i\right)^2}}{2 \sum_{i=1}^n x_i' y_i' w_i} \quad (5.10)$$

Observe that Equation (5.10) gives the roots to the quadratic equation where the constants

$$a = \sum_{i=1}^n x_i' y_i' w_i, \quad b = \sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i, \quad \text{and} \quad c = 1.$$

The `Arctan()` function is applied to the right-hand-side of Equation (5.10) to obtain the angle of rotation θ . The value for θ is substituted into Equations (5.11) and (5.12) to calculate the standard error terms along the axes. The standard error terms for the semi-major axis and semi-minor axis are given by Equations (5.11) and (5.12).

$$\delta_x = \sqrt{\frac{\sum_{i=1}^n (y_i' \sin \theta - x_i' \cos \theta)^2 w_i}{\sum_{i=1}^n w_i}} \quad (5.11)$$

$$\delta_y = \sqrt{\frac{\sum_{i=1}^n (y_i' \cos \theta - x_i' \sin \theta)^2 w_i}{\sum_{i=1}^n w_i}} \quad (5.12)$$

The advantages of these two estimators are that they allow trends in the data to be visualized. The mean center is pulled closer together with counties with larger weights w_i using Equation (5.9). Section 5 will show how Equation 5.9 is derived using a general linear means model and a linear regression model. Section 5.8 will show how Equation (5.10) can be extended to the exponential model.

5.5 VBA

The subroutine `SDEdeltas()` calculates the standard error terms along the axes in Equations (5.11) and (5.12). The first `FOR-NEXT` loop (15) calculates the sums and stores the values in variables `delta_x` and `delta_y`. After the `FOR-NEXT` loop (15), the two VBA statements divide by the total sum of the weights and take the square weights of the sums. The final values of the standard error terms along the axes are stored in the variables `delta_x` and `delta_y`. The `WITH-END-WITH` statements (16) save the results to the output spreadsheet called `STATS`.


```

Sub SDEdeltas()
'calculate the deltas
n = Selection.Rows.Count
delta_x = 0
delta_y = 0

    For i = 2 To n
        With ActiveSheet
            delta_x = delta_x + _
                ((.Cells (i, 2).Value - mean_Longitude) * Sin (atheta / (180 / WorksheetFunction.Pi() )) - _
                (.Cells (i, 1).Value - mean_Latitude) * Cos (atheta / (180 / WorksheetFunction.Pi() ))) ^ 2 * _
                .Cells (i, 3). Value
            delta_y = delta_y + _
                ((.Cells (i, 2).Value - mean_Longitude) * Cos (itheta / (180 / WorksheetFunction.Pi() )) - _
                (.Cells (i, 1).Value - mean_Latitude) * Sin (itheta / (180 / WorksheetFunction.Pi() ))) ^ 2 * _
                .Cells (i, 3).Value
        End With
    Next

delta_x = Sqr (delta_x / sum_weight)
delta_y = Sqr (delta_y / sum_weight)

End Sub

```

5.5.1 Weighted Mean Center

Although [Yuill, (67)] did provide a derivation for the weighted ellipse, the author did not provide the derivation for the mean centers. We will cover the derivation next. Let the variable W represent the random variable of interest. We wish to give those latitude observations with the largest values of W the most weight, and those latitude values with the smaller values of W the lesser weights. Consider the general linear means model

$$\text{model } x_i \quad y_i = \mu_x + \mu_y + \frac{\epsilon_x}{\sqrt{w_i}} + \frac{\epsilon_y}{\sqrt{w_i}}$$

where w_i equals to the weight for the sample total of a random variable on observation i ; x_i equals to the latitude of the i^{th} observation such that $0 \leq x_i \leq 90$; y_i equals to the longitude of the i^{th} observation such that $-180 \leq y_i \leq 180$; μ_x represents the weighted mean response of the latitude responses; and μ_y represents

the weighted mean response of the longitude responses. The general linear means model weights the error terms by w_i .

Consider the first model (latitude). We wish to minimize the weighted sum of squares. Our error terms are the difference between the known random values and the predicted values (i.e. the epsilons).

$$e_x = \sqrt{w_i}(x_i - \mu_x).$$

We square e_x and sum to obtain the weighted sum of squares.

$$e_x^2 = \sum_{i=1}^n w_i(x_i - \mu_x)^2.$$

The w_i 's and x_i 's are known observations. We estimate μ_x using the w_i 's and x_i 's observations. We take the derivative of e_x with respect to μ_x ; set it to zero; and solve for μ_x .

$$\begin{aligned} \frac{d e_x}{d \mu_x} &= -2 \sum_{i=1}^n w_i(x_i - \mu_x) = 0 \Rightarrow \\ \sum_{i=1}^n w_i x_i - \mu_x \sum_{i=1}^n w_i &= 0 \Rightarrow \\ \sum_{i=1}^n w_i x_i &= \mu_x \sum_{i=1}^n w_i \Rightarrow \\ \hat{\mu}_x &= \frac{\sum_{i=1}^n w_i x_i}{\sum_{i=1}^n w_i} = \bar{x}. \end{aligned} \tag{5.13}$$

A similar derivation holds for the longitude e_y by substituting y_i for x_i and μ_y for μ_x above.

$$\hat{\mu}_y = \frac{\sum_{i=1}^n w_i y_i}{\sum_{i=1}^n w_i} = \bar{y}. \tag{5.14}$$

The quantities (\bar{x}, \bar{y}) are the weighted mean center for the random variable W .

5.6 Ellipse Properties

The quantity a_1 in Equation (5.15) gives the length of the semi-major axis under the standard deviational ellipse.

$$a_1^2 = \frac{1}{n} \sum_{i=1}^n y_i'^2 w_i = \tag{5.15}$$

$$+ \frac{1}{n} \frac{2 (\sum_{i=1}^n x'_i y'_i w_i)^2}{- (\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i) + \sqrt{(\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i)^2 + 4 (\sum_{i=1}^n x'_i y'_i w_i)^2}}$$

where $x'_i = x_i - \bar{x}$, $0 \leq x_i \leq 90$ and $y'_i = y_i - \bar{y}$, $-180 \leq y_i \leq 180$, $w_i \geq 0$. Both \bar{x} and \bar{y} are weighted means. Equation (5.15) is subject to the constraint $\sum_{i=1}^n x_i'^2 w_i > \sum_{i=1}^n y_i'^2 w_i$.

The quantity b_1 in Equation (5.16) gives the formula for the semi-minor axis length for the weighted standard deviational ellipse.

$$b_1^2 = \frac{1}{n} \sum_{i=1}^n x_i'^2 w_i \quad (5.16)$$

$$- \frac{1}{n} \frac{2 (\sum_{i=1}^n x'_i y'_i w_i)^2}{- (\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i) + \sqrt{(\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i)^2 + 4 (\sum_{i=1}^n x'_i y'_i w_i)^2}}$$

There are two formulas for the area. We use one formula to check the other.

$$F = \quad (5.17)$$

$$\frac{\pi}{n} \sqrt{\sum_{i=1}^n (x_i - \bar{x})^2 w_i \sum_{i=1}^n (y_i - \bar{y})^2 w_i - \sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) w_i}.$$

The second formula for the area uses the axes lengths. $F = \pi a_1 b_1$ gives the area where a_1 is the semi-major axis length and b_1 is the semi-minor axis length.

Most calculus books give the *eccentricity* for an ellipse (or circle). It is based on the axes lengths with the major axis a_1 being the longest.

$$e = \frac{\sqrt{a_1^2 - b_1^2}}{a_1} \quad (5.18)$$

$0 \leq e \leq 1$ binds the eccentricity e . It denotes the degree from circularity. When e is close to 1, it means that the shape of the ellipse is very elliptical and very close to a straight line. When the eccentricity e is close to 0, it means that the shape of the ellipse is close to a circle.

The *concentration* is a count of the number of data points that fall within (or on) the ellipse. We calculate this quantity as a percentage relative to the total number data points.

5.7 VBA

We use the function `Area()` to calculate the area of the standard deviational ellipse. It calculates the area using the formula $F = \pi a_1 b_1$. We can write special code

to calculate the area using Equation (5.17). It does need the sums $\sum x_i^2$, $\sum y_i^2$, and $\sum x_i y_i$ from the subroutine `Angle_of_Rotation()`. If $\sum_{i=1}^n x_i'^2 w_i \geq \sum_{i=1}^n y_i'^2 w_i$, then the lengths must be calculated using the two equations:

$$a_1^2 - \frac{1}{n} \sum_{i=1}^n y_i'^2 w_i =$$

$$+ \frac{1}{n} \frac{2 \left(\sum_{i=1}^n x_i' y_i' w_i \right)^2}{\left(\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i \right) + \sqrt{\left(\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i \right)^2 + 4 \left(\sum_{i=1}^n x_i' y_i' w_i \right)^2}}$$

and

$$b_1^2 - \frac{1}{n} \sum_{i=1}^n x_i'^2 w_i =$$

$$- \frac{1}{n} \frac{2 \left(\sum_{i=1}^n x_i' y_i' w_i \right)^2}{\left(\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i \right) + \sqrt{\left(\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i \right)^2 + 4 \left(\sum_{i=1}^n x_i' y_i' w_i \right)^2}}$$

If $\sum_{i=1}^n x_i'^2 w_i < \sum_{i=1}^n y_i'^2 w_i$, then the lengths must be calculated using the two equations:

$$a_1^2 - \frac{1}{n} \sum_{i=1}^n x_i'^2 w_i =$$

$$+ \frac{1}{n} \frac{2 \left(\sum_{i=1}^n x_i' y_i' w_i \right)^2}{\left(\sum_{i=1}^n y_i'^2 w_i - \sum_{i=1}^n x_i'^2 w_i \right) + \sqrt{\left(\sum_{i=1}^n y_i'^2 w_i - \sum_{i=1}^n x_i'^2 w_i \right)^2 + 4 \left(\sum_{i=1}^n x_i' y_i' w_i \right)^2}}$$

and

$$b_1^2 - \frac{1}{n} \sum_{i=1}^n y_i'^2 w_i =$$

$$- \frac{1}{n} \frac{2 \left(\sum_{i=1}^n x_i' y_i' w_i \right)^2}{\left(\sum_{i=1}^n y_i'^2 w_i - \sum_{i=1}^n x_i'^2 w_i \right) + \sqrt{\left(\sum_{i=1}^n y_i'^2 w_i - \sum_{i=1}^n x_i'^2 w_i \right)^2 + 4 \left(\sum_{i=1}^n x_i' y_i' w_i \right)^2}}$$

This ensures we do not take the square root of a negative number in the denominator.

Avoiding taking the square root of a negative number, if $a_1 > b_1$, then a_1 is the semi-major axis and b_1 is the semi-minor axis. Otherwise, if $a_1 < b_1$, then a_1 is the semi-minor axis and b_1 is the semi-major axis.

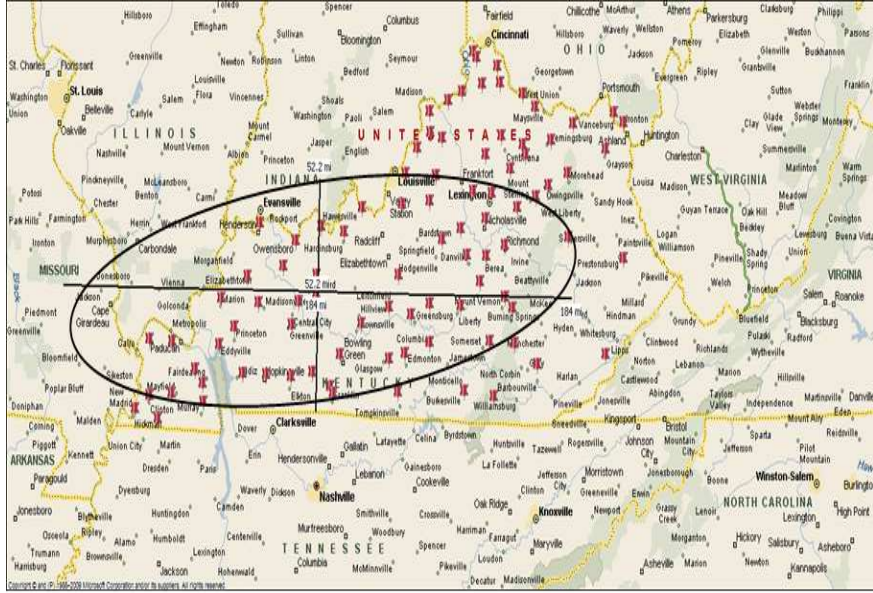


Figure 5.1 This figure shows the Kentucky data from 2003 and the standard deviational ellipse. It shows a sketch of the standard deviational ellipse with Ohio County at the center. The semi-major axis is approximately 184 miles long and the semi-minor axis is approximately 52 miles long. We rotated the major axis roughly 70° from the Y-axis in an imaginary Cartesian coordinate system.

Table 5.1 Results on Corn acreage in Kentucky

2003	
Center	(37.38297399, -86.8129514)
Axes	$a = 184.8734659$, $b = 52.23219209$
Area	30,336.30686 sq. mi.
Standard Deviations	$\delta_x = 1.29151776$, $\delta_y = 1.625551351$
Rotation	Y-Axis
Orientation	$\theta_x = 70.36257022^\circ$, $\theta_y = -19.63742978^\circ$
Eccentricity	0.959258636
Concentration	55/90

5.8 Kentucky Example

Tables 5.1 and 5.2 summarize the results from the weighted standard deviational ellipse. Using Equation (5.10), we calculate the angle of rotation as $\theta = 70.4^\circ$, the semi-major axis length as $a = 184$ miles and the semi-minor axis length as $b = 52$ miles (using 2003 data as an example). With the semi-major and semi-minor axis

Table 5.2 Results on Corn acreage in Kentucky

2004	
Center	(37.40588805, -86.8066601)
Axes	$a = 184.632769$, $b = 53.02565716$
Area	30,757.05259 sq. mi.
Standard Deviations	$\delta_x = 1.28529402$, $\delta_y = 1.613981198$
Rotation	Y-Axis
Orientation	$\theta_x = 70.42188382^\circ$, $\theta_y = -19.57811618^\circ$
Eccentricity	0.957872058

lengths, θ , and the weighted mean centers from Section 5, we can draw the standard deviational ellipse in Figure 5.1 using the procedure outlined in [Lee and Wong (36)]. It shows Ohio County as the center.

Given the summary tables, there is sufficient information to draw the ellipse. The information needed includes (\bar{x}, \bar{y}) , the angle of rotation from the given axis, and the lengths of the major and minor axes. The program does not draw the ellipse or calculate the percentage of concentration.

5.9 Crime Example

The example in this section comes from violent crime in the U.S. for 2007 and 2008. Tables 5.3 and 5.4 summarize the results for the weighted standard deviational ellipse. The data includes the continental U.S.

Table 5.3 Results on Violent Crime in the U.S.

2007	
Center	(36.8766, -92.1131260)
Axes	$a = 2,604.517043$, $b = 821.0805658$
Area	6,718,353.465 sq. mi.
Standard Deviations	$\delta_x = 4.857083084$, $\delta_y = 4.857028283$
Rotation	X-Axis
Orientation	$\theta_x = 0.045206827^\circ$, $\theta_y = -89.95479317^\circ$
Eccentricity	0.949007823

Figure 5.2 shows the standard deviational ellipse for the violent crime in the US for 2007. The ellipse covers most of the United States. We omitted Puerto Rico and Hawaii from the analysis. A histogram of the data shows the three highest crime states as California (13.6%), Texas (8.7%), and Florida (9.4%). These three states stretch from the west coast to the east coast and account for almost 32% (one-third)

Table 5.4 Results on Violent Crime in the U.S.

2008	
Center	(36.88957333, -92.0322279)
Axes	$a = 2,577.512836$, $b = 809.2827681$
Area	6,553,163.484 sq. mi.
Standard Deviations	$\delta_x = 15.39212752$, $\delta_y = 15.39222494$
Rotation	Y-Axis
Orientation	$\theta_x = 89.89264601^\circ$, $\theta_y = -0.107353986^\circ$
Eccentricity	0.949430707



Figure 5.2 This figure shows the standard deviational ellipse of violent crime in the US for the year 2007. It also shows the values of the violent crime using a bar graph. The ellipse is rather large compared to the area in the survey.

of the violent crime. This could explain why the weighted, standard deviational ellipse covers so much area.

5.10 GDP Example

This section shows the weighted standard deviational ellipse and the statistics for the Gross Domestic Product of OECD Countries data set. Since this is a global data set, the quantities tend to be enormous compared to the other two examples. Tables 5.5 and 5.6 summarize the standard deviational ellipse for the GDP data for 2008 and 2009. We see that the ellipses are highly eccentric. The total area contracted from 2008 to 2009. This is the period where the Great Recession took place. The mean center changed an entire degree in both the latitude and longitude directions.

The GDP data for 2010 gives the center of gravity as (45.94186507, -60.0743298). The GDP data for 2006 gives the center of gravity as (46.02643851, -48.8800190) and for 2007 as (45.94290023, -52.4580413). As we can see, the trend shows that



Figure 5.3 This figure shows the plot of the mean centers of global GDP on a map. As the years progress, the mean center measurements move closer to the Chinese coastline.

Table 5.5 Results on Gross Domestic Product of OECD Countries
2008

Center	(44.95865069, 173.1929)
Axes	$a = 4,308.807649$, $b = 695.7350122$
Area	9,417,829.832 sq. mi.
Standard Deviations	$\delta_x = 20.69202965$, $\delta_y = 19.93139537$
Rotation	X-Axis
Orientation	$\theta_x = 1.307691476^\circ$, $\theta_y = -88.69230852^\circ$
Eccentricity	0.986877931

the global GDP is moving away from the U.S. and toward the China coastline before and after the Great Recession.

Table 5.6 Results on Gross Domestic Product of OECD Countries

2009	
Center	(45.10440685, 172.9595)
Axes	$a = 4, 259.806334$, $b = 700.0081681$
Area	9,367,912.709 sq. mi.
Standard Deviations	$\delta_x = 21.04761702$, $\delta_y = 20.27737248$
Rotation	X-Axis
Orientation	$\theta_x = 1.328357537$, $\theta_y = -88.67164246$
Eccentricity	0.986405646

5.11 Exercises

1. Show that if there are no predictor variables in Table 3.1 and $w_i = \frac{1}{n}$, then the total sum of squares equals to the standard distance formula. HINT: $\hat{x}_i = \bar{x}$ and $\hat{y} = \bar{y}$.

Solution: If $\hat{x}_i = \bar{x}$ and $\hat{y}_i = \bar{y}$, then the sum of squares due to the model equals to zero.

$$\sum_{i=1}^n (\hat{x}_i - \bar{x})^2 + \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 = \sum_{i=1}^n (\bar{x} - \bar{x})^2 + \sum_{i=1}^n (\bar{y} - \bar{y})^2 = 0$$

Then, the total sums of squares in Table ?? due to error,

$$\sum_{i=1}^n w_i (x_i - \bar{x})^2 + \sum_{i=1}^n w_i (y_i - \bar{y})^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 + \frac{1}{n} \sum_{i=1}^n (y_i - \bar{y})^2.$$

The standard deviation is the square root of the sums of squares which gives Equation (5.1).

2. Derive Equation (5.17) using Equations (5.11), (5.12) and $F = \pi ab$.
3. What are two ways to obtain un-weighted estimates and statistics from the given Excel spreadsheets? **Soluton:**
 - Put $\frac{1}{n}$ into column 3.
 - Copy and paste the VBA code to a new module, modify it to perform un-weighted calculations.
4. Show that the linear regression least squares sums of squares due to error does not equal to the standard deviational ellipse error.

CHAPTER 6

THE EXPONENTIAL ELLIPSE

Most of the theory in this section follows the text in [Lee, (35)]. [Lee, (35), pages 164-165] demonstrates how to prove that a set of data has a particular distribution using graph paper. In the absence of graph paper, modern graphing software will suffice. The technique involves transforming the data to a linear regression model and plotting the transformed data. This section deviates from the traditional linear regression literature in that the data are not directly observed, then modeled.

6.1 Mean Latitude

We consider fitting the weighted exponential distribution to the data. If the fit is a success, then the result is a linear relationship between the random variable and the cumulative distribution function. If the fit fails to result in a linear relationship, then we can add a location parameter and a shape parameter to the distribution. Before we can transform the data, we need to find the maximum likelihood estimator for the scale parameter of the weighted exponential distribution. We summarize the latitude data using the maximum likelihood estimator.

The exponential distribution function considered for the latitude is

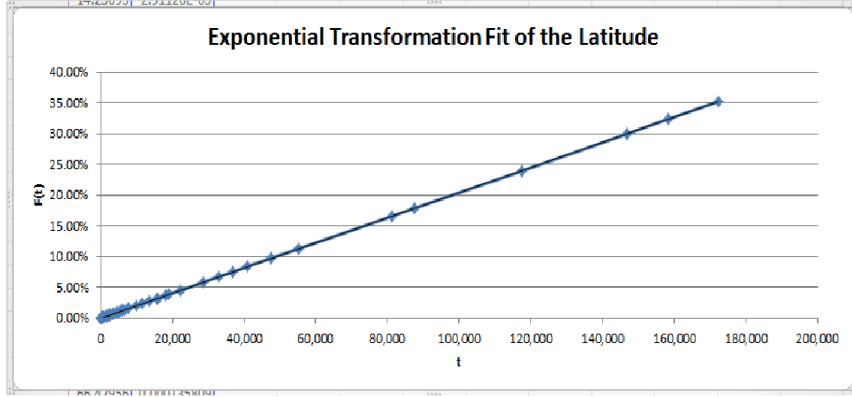


Figure 6.1 This figure shows the probability plot of the weighted latitude transformation. The exponential distribution fits the weighted latitude data since it shows a straight line.

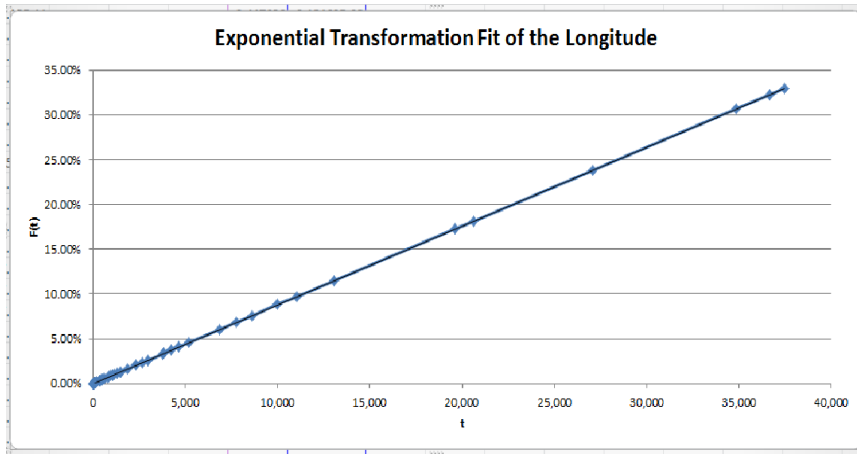


Figure 6.2 This figure shows the probability plot of the weighted longitude transformation. The exponential distribution fits the absolute value of the weighted longitude data since it shows a straight line. It was necessary to take absolute values since longitudinal data is negative.

$$f_x(w_i, x_i) = \lambda_x w_i e^{-\lambda_x x_i}, \quad 0 \leq x_i \leq 180, \quad w_i \geq 0, \quad (6.1)$$

where λ_x is the scale parameter. We estimate λ_x using the sample data. Equation (6.1) is a probability density function only if $\sum_{i=1}^n w_i = 1$. Why is this? Because the cumulative distribution function is

$$\begin{aligned} F_x(w, x) &= \int_0^t f(w, x) dx = \int_0^t \lambda_x w e^{-\lambda_x x} dx = \\ &= -w e^{-\lambda_x x} \Big|_0^t = w - w e^{-\lambda_x t} = F_t(t_x). \end{aligned}$$

1. As $t \rightarrow 0$, $F_t \rightarrow 0$.
2. As $t \rightarrow 180$, $F_t \rightarrow w$.
3. F_t is right continuous.

Thus, for F_t to be a true cumulative distribution function, $w = \sum_{i=1}^n w_i = 1$ is a necessary condition. We let each observation be weighted by $w_i = \frac{w'_i}{\sum_{i=1}^n w'_i}$ where w'_i is the observed value of the random variable with the latitude x_i .

The maximum likelihood estimator for the parameter λ_x occurs at the first derivative of the likelihood function $L = \prod_{i=1}^n f_x(w, x_i)$ taken with respect to λ_x .

$$L = \prod_{i=1}^n f_x(w, x_i) = \prod_{i=1}^n \lambda_x w \exp\{-\lambda_x x_i\} = (\lambda_x w)^n \exp\left\{-\lambda_x \sum_{i=1}^n x_i\right\},$$

$$\log L = n \log(\lambda_x w) - \lambda_x \sum_{i=1}^n x_i, \quad (6.2)$$

$$\frac{d \log L}{d \lambda_x} = \frac{n}{\lambda_x w} - \sum_{i=1}^n x_i = 0 \Rightarrow$$

$$\hat{\lambda}_x = \frac{n}{w \sum_{i=1}^n x_i}. \quad (6.3)$$

The maximum likelihood estimator appears in Equation (6.3).

As in [Lee, pages 162-164, (35)], we wish to fit a linear relationship by solving the following equality:

$$e^{-\lambda_x w t_x} = 1 - F_t(t_x),$$

$$\lambda_x w t_x = \log_e \left[\frac{1}{1 - F_t(t_x)} \right],$$

$$\bar{t}_x = \frac{1}{\lambda_x} \log_e \left[\frac{1}{1 - F_t(t_x)} \right] / w. \quad (6.4)$$

Using Equation (6.4), we plot our transformed data points $[t_{(x,i)}, F(t_{x,i})]$, to obtain the graph in Figure 6.1 on page 60. As one can see, it shows a linear relation that means the weighted latitude is exponentially distributed. We repeat the same procedure for the longitude.

For the joint distribution, we do not need to prove linearity. We do need the relationship between the two transformed variables, though.

6.2 Mean Longitude

A similar set of equations follow for the longitude from the latitude. The parameter estimator is different from that for the latitude since the data can be negative and the sample space for exponential distribution only allows for positive numbers (including zero). The exponential distribution function considered for the longitude is

$$f_y(w_i, y_i) = \lambda_y w_i e^{-\lambda_y y_i}, 0 \leq y_i \leq 360, w_i > 0. \quad (6.5)$$

We derive the cumulative distribution function as follow.

$$\begin{aligned} F_y(w, y) &= \int_0^u f(w, y) dy = \int_0^u \lambda_y w e^{-\lambda_y y} dy = \\ &= -w e^{-\lambda_y y} \Big|_0^u = w - w e^{-\lambda_y u} = F_u(u_y). \end{aligned}$$

1. As $u \rightarrow 0$, $F_u \rightarrow 0$.
2. As $u \rightarrow 360$, $F_u \rightarrow w$.
3. F_u is right continuous.

Thus, for F_u to be a true cumulative distribution function, $w = \sum_{i=1}^n w_i = 1$ is a necessary condition. We let each observation be weighted by $w_i = \frac{w'_i}{\sum_{i=1}^n w'_i}$ where w'_i is the observed value of the random variable with the longitude y_i .

We obtain the maximum likelihood estimator for λ_y for the longitude by taking the derivative of the likelihood function $L = \prod_{i=1}^n f(w, y_i)$ with respect to λ_y ; setting it to zero; and solving for λ_y .

$$\begin{aligned} L &= \prod_{i=1}^n f(w, y_i) = \prod_{i=1}^n \lambda_y w \exp \{-\lambda_y y_i\} = (\lambda_y w)^n \exp \left\{ -\lambda_y \sum_{i=1}^n y_i \right\} \\ \log L &= n \log(\lambda_y w) - \lambda_y \sum_{i=1}^n y_i, \end{aligned} \quad (6.6)$$

$$\frac{d \log L}{d \lambda_y} = \frac{n}{\lambda_y w} - \sum_{i=1}^n y_i = 0.$$

This results in Equation (6.7).

$$\hat{\lambda}_y = \frac{n}{w \sum_{i=1}^n y_i}. \quad (6.7)$$

We wish to see if we can obtain a linear relationship using the exponential distribution. We transform the data as follow:

$$\begin{aligned}
e^{-\lambda_y w u_y} &= 1 - F_u(u_y), \\
\lambda_y w u_y &= \log_e \left[\frac{1}{1 - F_u(u_y)} \right], \\
\bar{u}_y &= \frac{1}{\lambda_y} \log_e \left[\frac{1}{1 - F_u(u_y)} \right] / w.
\end{aligned} \tag{6.8}$$

Using Equation (6.8), we plot our transformed data points $[u_{(y,i)}, F(u_{y,i})]$, to obtain the graph in Figure 6.2 on page 60. As one can see, it shows a linear relation that means the weighted longitude is exponentially distributed.

6.3 Joint Distribution

Finally, the joint distribution between two weighted exponential models needs to be determined. Given the two probability density functions in Equations (6.1) and (6.5), the joint likelihood function is

$$\begin{aligned}
L &= \prod_{i=1}^n f_x(w, x_i) f_y(w, y_i) = \\
&\prod_{i=1}^n \lambda_x w \exp \{-\lambda_x x_i\} \lambda_y w \exp \{-\lambda_y y_i\}.
\end{aligned}$$

Let $\lambda_x = \lambda_y = \lambda_{xy}$. Then, the joint likelihood function becomes,

$$L(w, x_i, y_i) = \prod_{i=1}^n \lambda_{xy}^2 w^2 \exp \{-\lambda_{xy}(x_i + y_i)\}.$$

Next, we find the maximum likelihood estimator of λ_{xy} .

$$\log L(w, x_i, y_i) = 2n \log(\lambda_{xy} w) - \lambda_{xy} \sum_{i=1}^n (x_i + y_i). \tag{6.9}$$

$$\frac{d \log L}{d \lambda_{xy}} = \frac{2n}{\lambda_{xy} w} - \sum_{i=1}^n (x_i + y_i) \Rightarrow$$

$$\hat{\lambda}_{xy} = \frac{2n}{w \sum_{i=1}^n (x_i + y_i)}. \tag{6.10}$$

Equation (6.10) gives the maximum likelihood estimator for λ_{xy} for the joint distribution of the latitude and longitude. The joint probability density function of the latitude and longitude is

$$f_{xy}(w_i, x_i, y_i) = \lambda_{xy}^2 w_i^2 \exp \{-\lambda_{xy}(x_i + y_i)\}.$$

We need to find the cumulative distribution function $F_{xy}(w_i, x_i, y_i)$ for the joint distribution, next. We need the cumulative distribution function in the *exponential* standard deviational ellipse formula.

$$\begin{aligned} F_{xy}(w, x, y) &= \\ \int_0^t \int_0^u (w\lambda_{xy})^2 \exp \{-\lambda_{xy}(x + y)\} dy dx &= \\ \int_0^t -w^2 \lambda_{xy} \exp \{-\lambda_{xy}(x + y)\} \Big|_0^u dx &= \\ \int_0^t -\lambda_{xy} w^2 \exp \{-\lambda_{xy}(x + u)\} + \lambda_{xy} w^2 \exp \{-\lambda_{xy}x\} dx &= \\ w^2 \exp \{-\lambda_{xy}(x + u)\} - w^2 \exp \{-\lambda_{xy}x\} \Big|_0^t &= \\ w^2 \exp \{-\lambda_{xy}(t + u)\} - w^2 \exp \{-\lambda_{xy}t\} - w^2 \exp \{-\lambda_{xy}u\} + w^2 & \quad (6.11) \\ &= F_{tu}(w, t, u) \end{aligned}$$

Is Equation (6.11) a valid cumulative distribution function?

1. As $t \rightarrow 0$, and $u \rightarrow 0$, then $F_{tu} \rightarrow 0$.
2. As $t \rightarrow 180$, and $u \rightarrow 360$, then $F_{tu} \rightarrow w$.
3. F_{tu} is right continuous.

This is a valid cumulative distribution function only if the weights w sum to 1 as $t \rightarrow 180$ and $u \rightarrow 360$. We choose the weights

$$w'_i = \sqrt{\frac{w_i}{\sum_{i=1}^n w_i}}$$

where w_i equals to the observed value of the random variable. Then it is a trivial fact that

$$w^2 = \sum_{i=1}^n w_i'^2 = \sum_{i=1}^n \frac{w_i}{\sum_{i=1}^n w_i} = 1.$$

6.4 Exponential Ellipse

We really do not know how the ellipse in Section 5 got its name. The sums come from the method of least squares in regression. Both the general linear means model and the linear regression model assume the random variable has a normal distribution. The normal distribution has a location parameter typically named the mean μ and a scale parameter typically named the variance σ^2 . The term really does not apply to the exponential distribution since it does not have a location parameter. This Section will derive the *exponentially distributed, weighted, deviational ellipse*.

6.4.1 Mean Center

The weighted mean estimates (\bar{x}, \bar{y}) under the general linear means model in Section 5 and the exponential distribution are exactly the same — except possibly the sign on the longitude. Equation (6.12) gives the relationship for the exponential mean center for the latitude. Equation (6.13) gives the relationship for the exponential mean center for the longitude.

$$\frac{n}{\widehat{\lambda}_x} = \bar{x}. \quad (6.12)$$

$$\frac{n}{\widehat{\lambda}_y} = \bar{y}. \quad (6.13)$$

The exponential ellipse uses the following formulas.

$$t_x = \sum_{i=1}^n \frac{1}{\widehat{\lambda}_x} \log_e \left[\frac{1}{1 - F_t(t_i)} \right] / w_i. \quad (6.14)$$

$$u_y = \sum_{i=1}^n \frac{1}{\widehat{\lambda}_y} \log_e \left[\frac{1}{1 - F_u(u_i)} \right] / w_i. \quad (6.15)$$

$$t_{xy} = \sum_{i=1}^n \frac{1}{\widehat{\lambda}_{xy}} \log_e \left[\frac{1}{1 - F_{tu}(t_i, u_i)} \right] / w_i. \quad (6.16)$$

The two ellipses differ from the sums of squares between the original data $(w_i x_i, w_i y_i)$ and the transformed data $(t_{(x,i)}, u_{(y,i)})$. Equation (6.16) gives the mean joint distribution estimate between the weighted latitude and weighted longitude estimates. The estimators for λ_x , λ_y , and λ_{xy} appear in Equations (6.3), (6.7) and (6.10).

6.5 VBA

The subroutine `EXPparameters()` (4) calculates the parameter estimates $\widehat{\lambda}_x$, $\widehat{\lambda}_y$, and $\widehat{\lambda}_{xy}$ using Equations (6.3), (6.7), and (6.10). It also calculates the mean latitude and the mean longitude (\bar{x}, \bar{y}) using Equations (6.12) and (6.13). The `FOR-NEXT` loop (13) calculates the sums for the maximum likelihood estimates. The

VBA statements that follow the `FOR-NEXT` loop (13) calculate the final values for the parameter estimates and store them in the variables `lambda_x` for the latitude, `lambda_y` for the longitude, and `lambda_xy` for the joint distribution. We need the parameter estimates to calculate the distribution functions.

The `WITH-END-WITH` statements (14) store the mean latitude and the mean longitude in the `STATS` spreadsheet. Descriptive labels accompany the weighted means. In addition, we insert a heading for the Exponential statistics at the top of column D of the `STATS` spreadsheet.

The subroutine `EXPdistributions()` (5) calculates the exponential distributions for the latitude using Equation (6.1) and the longitude using Equation (6.5). The subroutine calculates the cumulative joint distribution function in Equation (6.11) and stores the value in the variable `F_xy`. The `FOR-NEXT` loop (15) calculates the sums from Equations (6.14), (6.15), and (6.16) and stores the values in the variables `t_x`, `t_y`, and `t_xy`, respectively. This `FOR-NEXT` loop also calculates the log likelihood functions in Equations (6.2), (6.6), and (6.9). It stores those values in the variables `L_x`, `L_y`, and `L_xy`, respectively.

```

4 13 26 Sub EXPparameters()
      n = Selection.Rows.Count
      wx = 0
      wy = 0
      For i = 2 To n
        With ActiveSheet
          wx = wx + .Cells(i, 1).Value * .Cells(i, 4).Value
          wy = wy + .Cells(i, 5).Value * .Cells(i, 4).Value
        End With
      Next
      lambda_x = (n - 1) / wx
      lambda_y = (n - 1) / wy
      lambda_xy = 2 * (n - 1) / (wx + wy)
End Sub

```

```

Sub EXPdistributions()
    'calculate the cumulative distribution functions for the latitude, longitude, and the joint distribution
    'calculate the values t_x, t_y, and t_xy, tx^2, ty^2, and t_xy^2

    n = Selection.Rows.Count
    t_x = 0
    t_y = 0
    t_xy = 0
    t_xsq = 0
    t_ysq = 0
    t_xysq = 0
    L_x = 0 'used for the log L calculation
    L_y = 0 'used for the log L calculation
    L_xy = 0 'used for the log L calculation

    For i = 2 To n
        With ActiveSheet
            f_x = .Cells(i, 4).Value - .Cells(i, 4).Value * EXP(-1 * lambda_x * .Cells(i, 1).Value)
            F_y = .Cells(i, 4).Value - .Cells(i, 4).Value * EXP(-1 * lambda_y * .Cells(i, 5).Value)
            { F_xy = .Cells(i, 4).Value * EXP(-1 * lambda_xy * (.Cells(i, 1).Value + .Cells(i, 5).Value)) - _
              .Cells(i, 4).Value * EXP(-1 * lambda_xy * .Cells(i, 1)) - _
              .Cells(i, 4).Value * EXP(-1 * lambda_xy * .Cells(i, 5)) + .Cells(i, 4).Value
            t_x = t_x + 1 / lambda_x * Log(1 / (1 - f_x)) / .Cells(i, 4).Value
            t_y = t_y + 1 / lambda_y * Log(1 / (1 - F_y)) / .Cells(i, 4).Value
            t_xy = t_xy + 1 / lambda_xy * Log(1 / (1 - F_xy)) / .Cells(i, 4).Value
            t_xsq = t_xsq + ((n - 1) / lambda_x * Log(1 / (1 - f_x)) / .Cells(i, 4).Value) ^ 2
            t_ysq = t_ysq + ((n - 1) / lambda_y * Log(1 / (1 - F_y)) / .Cells(i, 4).Value) ^ 2
            t_xysq = t_xysq + ((n - 1) / lambda_xy * Log(1 / (1 - F_xy)) / .Cells(i, 4).Value) ^ 2
            L_x = L_x + (n - 1) * Log(lambda_x * .Cells(i, 4).Value) - lambda_x * .Cells(i, 1).Value
            L_y = L_y + (n - 1) * Log(lambda_y * .Cells(i, 4).Value) - lambda_y * .Cells(i, 5).Value
            { L_xy = L_xy + 2 * (n - 1) * Log(lambda_xy * .Cells(i, 4).Value) - lambda_xy * _
              (.Cells(i, 1).Value + .Cells(i, 5).Value)
            End With
        Next
    End Sub

```

6.5.1 Ellipse

Equation (7) in [Yuill, (67)] is general enough to obtain a weighted, exponential standard deviational ellipse. We need to substitute the original sums of squares with the new sums of squares and re-calculate the major and minor axes. These axes lengths will determine the total area from which we can compare the area and number of weighted points inside the ellipse. Additionally, we can compare the standard deviations and the eccentricity of the ellipses.

Using equations (6.14), (6.15), and (6.16), we obtain our *exponentially distributed, weighted, deviational ellipse* in Equation (6.17).

$$\tan \theta = -\frac{(\sum_{i=1}^n t_{x,i}^2 - \sum_{i=1}^n u_{y,i}^2)}{2 \sum_{i=1}^n t_{x,y,i}} \pm \frac{\sqrt{(\sum_{i=1}^n t_{x,i}^2 - \sum_{i=1}^n u_{y,i}^2)^2 + 4 (\sum_{i=1}^n t_{x,y,i})^2}}{2 \sum_{i=1}^n t_{x,y,i}} \quad (6.17)$$

6.6 Axis Rotation VBA Code

The subroutine `EXPthetas()` (6) calculates the angle of rotation θ for the weighted exponential ellipse. This subroutine uses sums calculated previously in the `EXPdistributions()` subroutine. Equation (6.17) can have two values, depending on the plus or minus sign. The `EXPthetas()` subroutine calculates both values of θ and stores them in the variables `theta1` and `theta2`. We use the VBA function `Atn` to obtain $\arctan(\theta)$.

The `IF-THEN-ELSE` statement (16) determines if the rotation is about the Y-axis or the X-axis. The two values of θ from Equation (6.17) determine the axis of rotation. If the sum $\sum_{i=1}^n t_{x,i}^2 < \sum_{i=1}^n u_{y,i}^2$, then the axis of rotation is about the Y-axis. Otherwise, if $\sum_{i=1}^n t_{x,i}^2 \geq \sum_{i=1}^n u_{y,i}^2$, then the rotation is about the X-axis.

The `WITH-END-WITH` statements (17) put the results onto the output spreadsheet `STATS`. We write descriptive labels in column D (column 4) and the actual statistics in column E (column 5).

6.7 Ellipse Properties

Equation (6.18) gives the formula for calculating the deviations along the X axis.

$$\delta_x = \sum_{i=1}^n (u_{y,i} \sin \theta_a - t_{x,i} \cos \theta_a)^2 w_i \quad (6.18)$$

where the subscript a denotes the major axis for the angle of rotation θ . Equation (6.19) gives the formula for calculating the deviations along the Y axis.

$$\delta_y = \sum_{i=1}^n (u_{y,i} \cos \theta_b - t_{x,i} \sin \theta_b)^2 w_i \quad (6.19)$$

where the subscript b denotes the minor axis for the angle of rotation θ .

Equation (6.20) gives the formula for calculating the major axis.

$$a^2 - \frac{\sum_{i=1}^n u_{y,i}^2}{n} = \quad (6.20)$$

$$+ \frac{2 \left(\sum_{i=1}^n t_{x,y,i} \right)^2}{n \left(- \left(\sum_{i=1}^n t_{x,i}^2 - \sum_{i=1}^n u_{y,i}^2 \right) + \sqrt{\left(\sum_{i=1}^n t_{x,i}^2 - \sum_{i=1}^n u_{y,i}^2 \right)^2 + 4 \left(\sum_{i=1}^n t_{x,y,i} \right)^2} \right)}$$

subject to

$$\sum_{i=1}^n t_{x,i}^2 \leq \sum_{i=1}^n u_{y,i}^2.$$

Equation (6.21) gives the formula for calculating the minor axis.

$$b^2 - \frac{\sum_{i=1}^n u_{y,i}^2}{n} = \quad (6.21)$$

$$- \frac{2 \left(\sum_{i=1}^n t_{x,y,i} \right)^2}{n \left(- \left(\sum_{i=1}^n t_{x,i}^2 - \sum_{i=1}^n u_{y,i}^2 \right) + \sqrt{\left(\sum_{i=1}^n t_{x,i}^2 - \sum_{i=1}^n u_{y,i}^2 \right)^2 + 4 \left(\sum_{i=1}^n t_{x,y,i} \right)^2} \right)}$$

subject to

$$\sum_{i=1}^n t_{x,i}^2 \geq \sum_{i=1}^n u_{y,i}^2.$$

The difference between Equations (6.20) and (6.21) is the positive and negative sign after the first term. Since square roots are involved in the formulae, one must ensure that the conditions are true.

Equation (6.22) gives the formula for calculating the area.

$$F = \frac{\pi}{n} \sqrt{\sum_{i=1}^n t_{x,i}^2 \sum_{i=1}^n u_{y,i}^2 - \left(\sum_{i=1}^n t_{x,y,i} \right)^2} \quad (6.22)$$

It can be verified that $F = \pi ab$. The last equation is for the eccentricity of the ellipse. Equation (6.23) gives the formula for the eccentricity.

$$e = \frac{\sqrt{a^2 - b^2}}{a} \quad (6.23)$$

where Equation (6.20) gives the major axis length a and Equation (6.21) gives the minor axis length b .

6.8 VBA

The subroutine `EXPdeviations()` (7) calculates the deviations on the X-axis and the Y-axis. The `FOR-NEXT` loop (18) calculates the values t_x and u_y using Equations (6.14) and (6.15), respectively. In addition to t_x and u_y , using the two values for θ we previously obtained using the plus and minus sign, we can calculate the sums for the deviations on the X-axis and Y-axis using Equations (6.18) and (6.19). We store the final values in the variables `delta_x` and `delta_y`. The source code after the `FOR-NEXT` loop calculates the standard errors on the X-axis and Y-axis. We simply divide by the sum of the weights and then take the square root of the quantity. The final values for the standard errors in Equations (6.18) and (6.19) reside in the variables `error_x` and `error_y`.

The `WITH-END-WITH` statement (19) saves the standard errors to the `STATS` spreadsheet in column E (column 5). We put descriptive labels with the output in column D (column 4).

The `EXPstats()` subroutine calculates the major and minor axes lengths, the area, and the eccentricity. One of the first VBA statements calculates the area of the exponential ellipse before the `IF-THEN-ELSE` statement using Equation (6.22) and saves the value in the variable `f`. The `IF-THEN-ELSE` statement (20) calculates the semi-major axis length a in Equation (6.20) and the semi-minor axis length b in Equation (6.21). Should the value $\sum_{i=1}^n t_{x,i}^2 < \sum_{i=1}^n u_{y,i}^2$, then we reverse the roles of the sums to calculate the lengths of the semi-major and semi-minor axes. This approach avoids the square root of a negative number in the denominator.

The second `IF-THEN-ELSE` statement (21) calculates the eccentricity. If the axis lengths $a \geq b$, then a is the semi-major axis. The code uses Equation (6.23) to calculate the eccentricity. If $b > a$, then b is assumed to be the semi-major axis and Equation (6.23) is applied appropriately.

The `WITH-END-WITH` statement (2) saves the remaining exponential statistics to the `STATS` spreadsheet in column E (column 5). We add appropriate labels in column D (column 4).

Table 6.1 Results for the Exponential Model for Kentucky

	2003
Center	(37.38297399, -86.8129514)
Axes	$a = 273.2320657$, $b = 37.38920285$
Area	32,094.2879 sq. mi.
Standard Deviations	$\delta_x = 3.050790055$, $\delta_y = 3.050809397$
Rotation	Y-Axis
Orientation	$\theta_x = 89.99867285^\circ$, $\theta_y = -0.001327443^\circ$
Eccentricity	0.990593099
Concentration	27/90

```

Sub EXPdeviations()
'calculate the deviations on the axes

n = Selection.Rows.Count
delta_x = 0
delta_y = 0

    For i = 2 To n
        With ActiveSheet
            f_x = .Cells(i, 4).Value - .Cells(i, 4).Value * EXP(-1 * lambda_x * .Cells(i, 1).Value)
            F_y = .Cells(i, 4).Value - .Cells(i, 4).Value * EXP(-1 * lambda_y * .Cells(i, 5).Value)
            t_x_ = 1 / lambda_x * Log(1 / (1 - f_x)) / .Cells(i, 4).Value
            t_y_ = 1 / lambda_y * Log(1 / (1 - F_y)) / .Cells(i, 4).Value
            delta_x = delta_x + (t_y_ * Sin(atheta / 57.2957795) - t_x_ * Cos(atheta / 57.2957795)) ^ 2 * .Cells(i, 3).Value
            delta_y = delta_y + (t_y_ * Cos(itheta / 57.2957795) - t_x_ * Sin(itheta / 57.2957795)) ^ 2 * .Cells(i, 3).Value
        End With
    Next

error_x = Sqr(delta_x / sum_weight) 'standard errors on the x axis
error_y = Sqr(delta_y / sum_weight) 'standard errors on the y axis

End Sub

```

Table 6.2 Results for the Exponential Model for Kentucky

2004	
Center	(37.40588805, -86.8066601)
Axes	$a = 273.2391008$, $b = 37.4120928$
Area	32,114.76312 sq. mi.
Standard Deviations	$\delta_x = 2.987069658$, $\delta_y = 2.987088203$
Rotation	Y-Axis
Orientation	$\theta_x = 89.99870116^\circ$, $\theta_y = -0.001299137^\circ$
Eccentricity	0.99058201

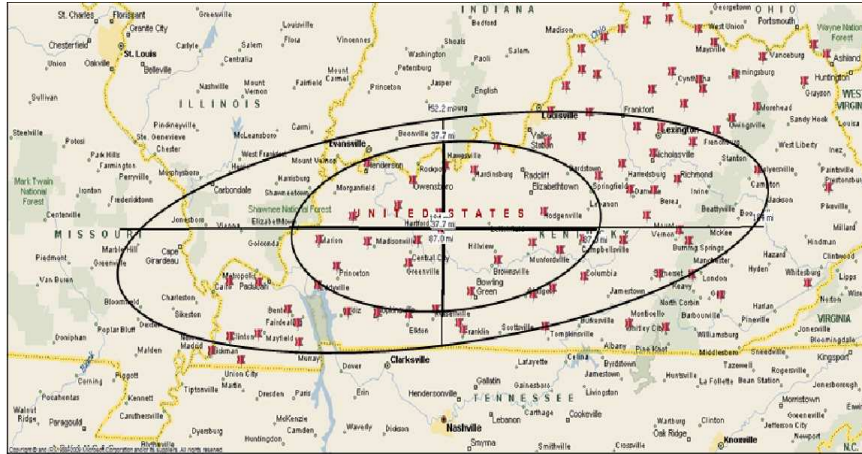


Figure 6.3 This figure shows sketches of both the standard deviational ellipse and the exponential deviational ellipse from the Kentucky data for the year 2003. The exponential ellipse is the smaller of the two ellipses. It covers one-third the area of the standard deviational ellipse, yet one-half of the same points.

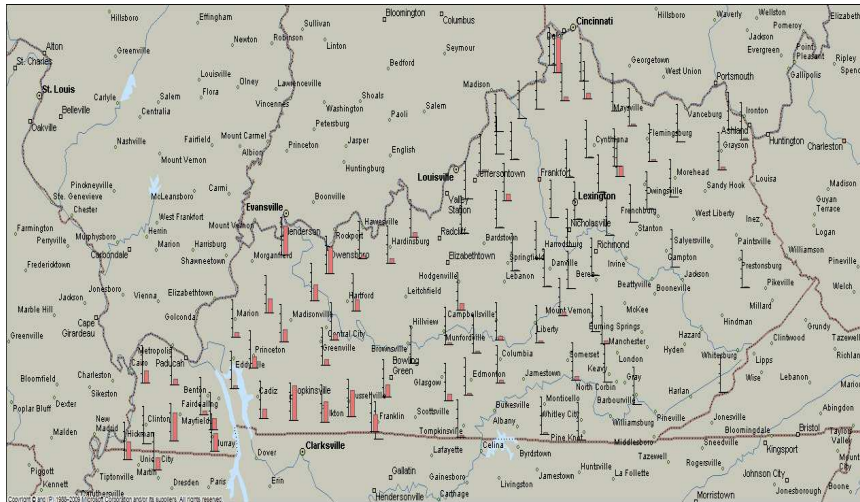


Figure 6.4 This figure shows a raised bar graph of the weights in their respective counties for the Kentucky data for the year 2003. The majority of the weighted data comes from the lower-left corner of the state of Kentucky. The majority of the un-weighted data comes from the center of the state and the right-hand side.

6.9 Kentucky Example

We compare the weighted, standard deviational ellipse Tables 5.1 on page 54 to the exponential ellipse Tables 6.1 and 6.2. It shows that the area of the ellipses have been significantly reduced (within years). It shows that the exponential ellipse is contained inside the standard deviational ellipse.

Figure 6.3 on page 72 shows both the standard deviational ellipse and the exponential deviational ellipse using 2003 Kentucky data as an example. The gain in knowing the distribution of the data is in the reduction of the standard deviations and the reduction of area. What did the two exponential ellipses identify that the standard deviational ellipse did not? The bar graph in Figure 6.4 shows the weighted data in Kentucky. Looking at Figure 6.4, it found those land segments with the largest weights that are closest to those with smaller weights.

6.10 Crime Example

The example in this section comes from violent crime data in the U.S. for 2007 and 2008. Tables 6.3 and 6.3 summarize the results for the weighted exponential deviational ellipse for violent crime in the U.S. for 2007 and 2008. The data includes the continental U.S.

Figure 6.5 on page 74 shows the exponential ellipse inside of the standard deviational ellipse. The center of gravity is the same as that for the standard deviational



Figure 6.5 This figure shows both the exponential deviational ellipse and the standard deviational ellipse for the violent crime data in the US for 2007. Both have the same center of gravity. The exponential ellipse is much smaller than the standard deviational ellipse.

Table 6.3 Results for the Exponential Model on Violent Crime in the U.S.

2007	
Center	(36.8766, -92.1131260)
Axes	$a = 268.0294408$, $b = 36.89621172$
Area	31,068.06111 sq. mi.
Standard Deviations	$\delta_x = 5.511172335$, $\delta_y = 5.511237311$
Rotation	Y-Axis
Orientation	$\theta_x = 89.99754656^\circ$, $\theta_y = -0.002453737^\circ$
Eccentricity	0.990479917

Table 6.4 Results for the Exponential Model on Violent Crime in the U.S.

2008	
Center	(36.88957333, -92.0322279)
Axes	$a = 268.1090262$, $b = 36.90903466$
Area	31,088.08671 sq. mi.
Standard Deviations	$\delta_x = 5.51140013$, $\delta_y = 5.511465094$
Rotation	Y-Axis
Orientation	$\theta_x = 89.99754726^\circ$, $\theta_y = -0.002453032^\circ$
Eccentricity	0.990478949

Table 6.5 Results for the Exponential Model on Gross Domestic Product

2008	
Center	(44.95865069, 173.19287)
Axes	$a = 162.1154426$, $b = 45.08062313$
Area	22,959.59217 sq. mi.
Standard Deviations	$\delta_x = 5.203392563$, $\delta_y = 5.203732811$
Rotation	Y-Axis
Orientation	$\theta_x = 89.99301352^\circ$, $\theta_y = -0.006986772^\circ$
Eccentricity	0.960558702

Table 6.6 Results for the Exponential Model on Gross Domestic Product

2009	
Center	(45.10440685, 172.9595353)
Axes	$a = 161.9197005$, $b = 45.22752649$
Area	23,006.59785 sq. mi.
Standard Deviations	$\delta_x = 5.198792719$, $\delta_y = 5.199135181$
Rotation	Y-Axis
Orientation	$\theta_x = 89.99299395$, $\theta_y = -0.007006348$
Eccentricity	0.960197864

ellipse (36.89957, -92.03222). There is a tremendous difference in the size of the two different ellipses.

6.11 GDP Example

. This section shows the weighted exponential ellipse and the statistics for the Gross Domestic Product of OECD Countries data set. Since this is a global data set, the quantities tend to be enormous compared to the other two examples. Tables 6.5 and 6.6 summarize the standard deviational ellipse for the 2008 and 2009 GDP data. We see that the ellipses are highly eccentric. The total area contracted from 2008 to 2009. This is the period where the Great Recession took place. The mean center changed an entire degree in both the latitude and longitude directions. Where are the mean centers located? In 2008, the center of gravity for the OECD countries was in the Pacific Ocean between the United States and China. In 2009, the center of gravity moved away from the United States coastline and closer to the Chinese coastline. See Figure 5.3.

6.12 Comparison to SDE

We will derive proof that shows the weighted deviational ellipse always has a smaller area than the standard deviational ellipse. We base the proof on showing that both the semi-major axis and the semi-minor axis are shorter than those of the standard deviational ellipse are.

Semi-major Axes Length Equation (6.24) gives the length of the semi-major axis under the standard deviational ellipse by the quantity a_1 .

$$a_1^2 - \frac{1}{n} \sum_{i=1}^n y_i'^2 w_i = \quad (6.24)$$

$$+ \frac{1}{n} \frac{2 \left(\sum_{i=1}^n x_i' y_i' w_i \right)^2}{\left(\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i \right) + \sqrt{\left(\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i \right)^2 + 4 \left(\sum_{i=1}^n x_i' y_i' w_i \right)^2}}$$

where $x_i' = x_i - \bar{x}$, $0 \leq x_i \leq 90$ and $y_i' = y_i - \bar{y}$, $-180 \leq y_i \leq 180$, $w_i \geq 0$. Both \bar{x} and \bar{y} are weighted means. Equation (6.24) is subject to the constraint $\sum_{i=1}^n x_i'^2 w_i > \sum_{i=1}^n y_i'^2 w_i$. Equation (6.25) gives the length of the semi-major axis under the weighted, exponential deviational ellipse.

$$a_2^2 - \frac{1}{n} \sum_{i=1}^n t_{y,i}^2 = \quad (6.25)$$

$$+ \frac{1}{n} \frac{2 \left(\sum_{i=1}^n t_{x,y,i} \right)^2}{\left(\sum_{i=1}^n t_{x,i}^2 - \sum_{i=1}^n t_{y,i}^2 \right) + \sqrt{\left(\sum_{i=1}^n t_{x,i}^2 - \sum_{i=1}^n t_{y,i}^2 \right)^2 + 4 \left(\sum_{i=1}^n t_{x,y,i} \right)^2}}$$

where

$$t_{x,i}^2 = \sum_{i=1}^n \left(\frac{1}{\hat{\lambda}_x} \log_e \left[\frac{1}{(1 - F_x(t_i))} \right] / w_i' \right)^2,$$

$$\hat{\lambda}_x = \frac{n}{\sum_{i=1}^n w_i' x_i}, \quad w_i' = \frac{w_i}{\sum_{i=1}^n w_i}, \quad 0 \leq x_i \leq 180.$$

$$t_{y,i}^2 = \sum_{i=1}^n \left(\frac{1}{\hat{\lambda}_y} \log_e \left[\frac{1}{(1 - F_y(t_i))} \right] / w_i' \right)^2,$$

$$\hat{\lambda}_y = \frac{n}{\sum_{i=1}^n w_i' y_i}, \quad w_i' = \frac{w_i}{\sum_{i=1}^n w_i}, \quad 0 \leq y_i \leq 360.$$

$$t_{x,y} = \sum_{i=1}^n \left(\frac{1}{\hat{\lambda}_{xy}} \log_e \left[\frac{1}{(1 - F_{xy}(t_i, u_i))} \right] / w_i' \right),$$

$$\hat{\lambda}_{xy} = \frac{2n}{\sum_{i=1}^n w'_i x_i + w'_i y_i}, \quad w'_i = \frac{w_i}{\sum_{i=1}^n w_i},$$

$$0 \leq x_i \leq 180, \quad 0 \leq y_i \leq 360.$$

Under both models, $-90 \leq \bar{x} \leq 90$ and $-180 \leq \bar{y} \leq 180$. Now that the lengths of the semi-major axes have been defined under the weighted standard deviational ellipse and the weighted exponential deviational ellipse, it will be shown that the sums of squares for the latitude, longitude, and jointly are smaller for the exponential ellipse. If that is proven, then $n a_2^2 \leq n a_1^2$.

Using substitution, the sum from the standard deviational ellipse

$$n \sum_{i=1}^n x_i'^2 w_i = n \sum_{i=1}^n (x_i - \bar{x})^2 w_i$$

where \bar{x} is a weighted mean is positive definite. Using the relationship between $\hat{\lambda}_x$ and \bar{x} in Equation (6.12), we can rewrite the sum from the exponential ellipse as

$$n t_{x,i}^2 = n \sum_{i=1}^n \left(\frac{1}{\hat{\lambda}_x} \log_e \left[\frac{1}{(1 - F_x(t_i))} \right] / w'_i \right)^2 =$$

$$\frac{1}{n} \sum_{i=1}^n \left(\bar{x} \log_e \left[\frac{1}{(1 - F_x(t_i))} \right] / \frac{w_i}{\sum_{i=1}^n w_i} \right)^2$$

We make two observations.

1. In the standard ellipse, the sum $\sum_{i=1}^n (x_i - \bar{x})^2$ is positive definite.
2. In the exponential ellipse, an approximate formula for the \log_e is $\log_e = (x - 1) - \frac{(x-1)^2}{2} + \frac{(x-1)^3}{3} - \frac{(x-1)^4}{4} + \dots$, $0 < x \leq 2$.

So,

$$\sum_{i=1}^n (x_i - \bar{x})^2 w_i = \sum_{i=1}^n (x_i^2 - 2x_i \bar{x} + \bar{x}^2) w_i >$$

$$\frac{1}{n} \sum_{i=1}^n \left(\bar{x} \left[\left(\frac{1}{(1 - F_x(t_i))} - 1 \right) - \frac{\left(\frac{1}{(1 - F_x(t_i))} - 1 \right)^2}{2} + \right. \right.$$

$$\left. \frac{\left(\frac{1}{(1 - F_x(t_i))} - 1 \right)^3}{3} - \right.$$

$$\left. \frac{\left(\frac{1}{(1 - F_x(t_i))} - 1 \right)^4}{4} + \dots \right] \left(\frac{\sum_{i=1}^n w_i}{w_i} \right) \right)^2 =$$

$$\begin{aligned} & \frac{1}{n} \sum_{i=1}^n \left(\bar{x} \log_e \left[\frac{1}{(1 - F_x(t_i))} \right] \left(\frac{1}{w'_i} \right) \right)^2 \\ &= \sum_{i=1}^n \left(\frac{1}{\bar{\lambda}_x} \log_e \left[\frac{1}{(1 - F_x(t_i))} \right] / w'_i \right)^2. \end{aligned}$$

Hence, we can conclude that the sums $n \sum_{i=1}^n x_i'^2 w_i > n t_{x,i}^2$. There are two more sets of sums to prove are smaller — those associated with the weighted longitude and the weighted joint distribution. The sums associated with the weighted longitude follow the same argument as above. Simply replace x with y for the standard deviational ellipse and replace x with y for the exponential ellipse. Then it follows that $n \sum_{i=1}^n y_i'^2 w_i > n u_{y,i}^2$.

We wish to establish the relationship between $(\sum_{i=1}^n x'_i y'_i w_i)^2$ and $t_{x,y}^2$. Which is larger?

$$\begin{aligned} \left(\sum_{i=1}^n x'_i y'_i w_i \right)^2 &= \left(\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y}) w_i \right)^2 > t_{x,y}^2 = \\ & \left(\sum_{i=1}^n \left(\frac{\bar{x} + \bar{y}}{2n} \left[\left(\frac{1}{(1 - F_{tu}(t_i, u_i))} - 1 \right) - \right. \right. \right. \\ & \quad \left. \frac{\left(\frac{1}{(1 - F_{tu}(t_i, u_i))} - 1 \right)^2}{2} + \frac{\left(\frac{1}{(1 - F_{tu}(t_i, u_i))} - 1 \right)^3}{3} - \right. \\ & \quad \left. \left. \left. \frac{\left(\frac{1}{(1 - F_{tu}(t_i, u_i))} - 1 \right)^4}{4} + \dots \right] \left(\frac{\sum_{i=1}^n w_i}{w_i} \right) \right) \right)^2 = \\ & \left(\sum_{i=1}^n \left(\frac{1}{\bar{\lambda}_{xy}} \log_e \left[\frac{1}{(1 - F_{tu}(t_i, u_i))} \right] / w'_i \right) \right)^2 \end{aligned}$$

We can make similar observations as before. The bounds on \log_e are $0 < \frac{1}{(1 - F_{tu}(t_i, u_i))} - 1 \leq 2$ and the sums $(\sum_{i=1}^n x'_i y'_i w_i)^2$ and $(t_{x,y})^2$ are positive definite.

We have shown that the three sums $t_{x,i}^2$, $u_{y,i}^2$ and $t_{x,y}^2$ are smaller than their respective sums in the standard deviational ellipse $\sum_{i=1}^n x_i'^2 w_i$, $\sum_{i=1}^n y_i'^2 w_i$, and $(\sum_{i=1}^n x'_i y'_i w_i)^2$. From this, we can conclude that the semi-major axis length of the exponential ellipse a_2 is smaller than that of the standard deviational ellipse a_1 .

Semi-Minor Axes Length The sums of squares in this section are the same as those in the previous section. The formulas for the semi-minor axes are slightly different than those for the semi-major axes. The quantity b_1 in Equation (6.26) gives the formula for the semi-minor axis length for the weighted standard deviational ellipse.

$$b_1^2 - \frac{1}{n} \sum_{i=1}^n x_i'^2 w_i = \quad (6.26)$$

$$-\frac{1}{n} \frac{2 \left(\sum_{i=1}^n x_i' y_i' w_i \right)^2}{\left(\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i \right) + \sqrt{\left(\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i \right)^2 + 4 \left(\sum_{i=1}^n x_i' y_i' w_i \right)^2}}$$

Equation (6.27) gives the length b_2 of the semi-minor axis under the weighted, exponential deviational ellipse.

$$b_2^2 - \frac{1}{n} \sum_{i=1}^n t_{x,i}^2 = \quad (6.27)$$

$$-\frac{1}{n} \frac{2 \left(\sum_{i=1}^n t_{x,y,i} \right)^2}{\left(\sum_{i=1}^n t_{x,i}^2 - \sum_{i=1}^n u_{y,i}^2 \right) + \sqrt{\left(\sum_{i=1}^n t_{x,i}^2 - \sum_{i=1}^n u_{y,i}^2 \right)^2 + 4 \left(\sum_{i=1}^n t_{x,y,i} \right)^2}}$$

Given the arguments in the previous Section, the following three inequalities of the sums hold true.

1. $n \sum_{i=1}^n x_i'^2 w_i > n t_{x,i}^2$.
2. $n \sum_{i=1}^n y_i'^2 w_i > n u_{y,i}^2$.
3. $\left(\sum_{i=1}^n x_i' y_i' w_i \right)^2 > t_{x,y}^2$.

We can conclude that the semi-minor axis length of the exponential ellipse b_2 is smaller than that of the standard deviational ellipse b_1 , $b_2 < b_1$.

Axes Summary We showed that both the semi-major axis length and semi-minor axis length of the exponential ellipse are smaller than those of the standard deviational ellipse. Given this, the area of the exponential ellipse is always smaller.

Given that both share the same mean center, then next obvious question is that, does the weighted standard deviational ellipse always encompass the weighted exponential deviational ellipse? Obviously, in the numeric example given, the answer is yes. Under what conditions do the standard deviational ellipses encompass the weighted exponential deviational ellipses? If the axes relationship holds true, $a_1 \leq b_2$, then the standard deviational ellipse encompasses the exponential ellipse regardless of the angle of rotation θ . Let θ_1 be the expression for the angle of rotation for the standard deviational ellipse given in Equation (6.28).

$$\tan \theta_1 = -\frac{\left(\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i\right)}{2 \sum_{i=1}^n x_i' y_i' w_i} \pm \frac{\sqrt{\left(\sum_{i=1}^n x_i'^2 w_i - \sum_{i=1}^n y_i'^2 w_i\right)^2 + 4 \left(\sum_{i=1}^n x_i' y_i' w_i\right)^2}}{2 \sum_{i=1}^n x_i' y_i' w_i}, \quad (6.28)$$

$-1 \leq \tan \theta_1 \leq 1$. Let θ_2 be the angle of rotation for the exponential ellipse given in Equation (6.29).

$$\tan \theta_2 = -\frac{\left(\sum_{i=1}^n t_{x,i}^2 - \sum_{i=1}^n u_{y,i}^2\right)}{2 \sum_{i=1}^n t_{x,y,i}} \pm \frac{\sqrt{\left(\sum_{i=1}^n t_{x,i}^2 - \sum_{i=1}^n u_{y,i}^2\right)^2 + 4 \left(\sum_{i=1}^n t_{x,y,i}\right)^2}}{2 \sum_{i=1}^n t_{x,y,i}}, \quad (6.29)$$

$-1 \leq \tan \theta_2 \leq 1$. Tables 6.7 and 6.8 summarize the relationship between θ_1 and θ_2 for different values of the plus/minus sign in the middle of θ_1 and θ_2 .

Table 6.7 Relationship of angles of rotation for large x_i' s.

Bounds	Plus/Minus	$\tan()$	Angle	Inference
$\sum_{i=1}^n x_i'^2 w_i > \sum_{i=1}^n y_i'^2 w_i$	+	Positive	$\theta_1 > 0$	$0 < \theta_1 < 90$
$\sum_{i=1}^n t_{x,i}^2 > \sum_{i=1}^n u_{y,i}^2$	+	Positive	$\theta_2 > 0$	$0 < \theta_2 < \theta_1 < 90$
$\sum_{i=1}^n x_i'^2 w_i < \sum_{i=1}^n y_i'^2 w_i$	-	Negative	$\theta_1 < 0$	$0 < \theta_1 < 90$
$\sum_{i=1}^n t_{x,i}^2 < \sum_{i=1}^n u_{y,i}^2$	-	Negative	$\theta_2 < 0$	$0 < \theta_1 < \theta_2 < 90$

Table 6.8 Relationship of angles of rotation for small x_i' s.

Bounds	Plus/Minus	$\tan()$	Angle	Inference
$\sum_{i=1}^n x_i'^2 w_i < \sum_{i=1}^n y_i'^2 w_i$	+	Positive	$\theta_1 > 0$	$0 < \theta_1 < 90$
$\sum_{i=1}^n t_{x,i}^2 < \sum_{i=1}^n u_{y,i}^2$	+	Positive	$\theta_2 > 0$	$0 < \theta_2 < \theta_1 < 90$
$\sum_{i=1}^n x_i'^2 w_i > \sum_{i=1}^n y_i'^2 w_i$	-	Negative	$\theta_1 < 0$	$0 < \theta_1 < 90$
$\sum_{i=1}^n t_{x,i}^2 > \sum_{i=1}^n u_{y,i}^2$	-	Negative	$\theta_2 < 0$	$0 < \theta_1 < \theta_2 < 90$

Upon examining Tables 6.7 and 6.8, we would expect that for extreme values of θ_1 that the exponential ellipse would fall inside the standard deviational ellipse.

6.13 Exercises

1. Consider the pdf $f(y) = \lambda e^{-\lambda_y y/w}$.

- (a) Derive the mle for λ_y .
 (b) Find the cdf. Show that it is a true cdf for $\sum_{i=1}^n w_i = 1$. Solution:

$$f(y) = \lambda e^{-\lambda y/w} \Rightarrow$$

$$F(y) = \int_0^u \lambda e^{-\lambda y/w} dy = w e^{-\lambda y/w} \Big|_0^u = w - w e^{-\lambda u/w}$$

As $u \rightarrow \infty$, $F(y) \rightarrow w$. So, for $F(y)$ to be a cdf, $w = \sum_{i=1}^n w_i = 1$.

2. Consider the pdf $f = \lambda e^{-\lambda_y w y}$.
 (a) Derive the mle for λ_y .
 (b) Find the cdf. Show that it is not necessarily a cdf for $\sum_{i=1}^n w_i = 1$.
 3. Programming Exercise. Verify the cdfs $F(x)$, $F(y)$, and $F(x, y)$ sum to 1. This should take no more than three or four lines of additional VBA code.
 4. Explicitly show that

$$\sum_{i=1}^n \frac{w_i}{\sum_{i=1}^n w_i} = 1.$$

Solution: Let $w = \sum_{i=1}^n w_i$, $w_i \geq 0$, $\forall i$. Then,

$$\sum_{i=1}^n \frac{w_i}{\sum_{i=1}^n w_i} = \frac{1}{w} \sum_{i=1}^n w_i = \frac{w}{w} = 1.$$

5. Show that

$$\frac{n}{\widehat{\lambda}_x} = \bar{x} \text{ in Chapter 5.}$$

Solution: Let $w = \sum_{i=1}^n \frac{w_i}{\sum_{i=1}^n w_i}$, $w_i \geq 0$, $\forall i$. Then,

$$\widehat{\lambda}_x = \frac{n}{w \sum_{i=1}^n x_i} = \frac{n}{\left(\sum_{i=1}^n \frac{w_i x_i}{\sum_{i=1}^n w_i} \right)} \Rightarrow$$

$$\frac{n}{\widehat{\lambda}_x} = \sum_{i=1}^n \frac{w_i x_i}{\sum_{i=1}^n w_i} = \bar{x}, \quad 0 \leq x_i \leq 180, \quad -90 \leq \bar{x} \leq 90.$$

6. Show that

$$\frac{n}{\widehat{\lambda}_y} = \bar{y} \text{ in Chapter 5.}$$

Solution: This is very similar to Question 2, but the bounds are different. Let $w = \frac{w_i}{\sum_{i=1}^n w_i}$, $w_i \geq 0, \forall i$. Then,

$$\hat{\lambda}_y = \frac{n}{w \sum_{i=1}^n y_i} = \frac{n}{\left(\sum_{i=1}^n \frac{w_i y_i}{\sum_{i=1}^n w_i} \right)} \Rightarrow$$

$$\frac{n}{\hat{\lambda}_y} = \sum_{i=1}^n \frac{w_i y_i}{\sum_{i=1}^n w_i} = \bar{y},$$

$$0 \leq y_i \leq 360, \quad -180 \leq \bar{y} \leq 180.$$

7. Derive Equation (6.22) using Equations (6.18), (6.19) and $F = \pi ab$.

CHAPTER 7

THE WEIBULL ELLIPSE

7.1 Latitude Mean Center

Equation (7.1) gives the weighted probability density function for the latitude.

$$f_x(w_i, x_i) = \gamma_x (w_i \lambda_x)^{\gamma_x} (x_i)^{\gamma_x - 1} \exp \{ - (\lambda_x x_i)^{\gamma_x} \}, \quad (7.1)$$

$$0 \leq x_i \leq 180, \lambda_x > 0, \gamma_x > 0.$$

We need the cumulative distribution function $F_x(w, x_i)$. Therefore, we integrate the probability density function $f_x(w, x_i)$.

$$\begin{aligned} F_x(w, x_i) &= \int_0^t f(w, x) dx = \\ &= \int_0^t \gamma_x (w \lambda_x)^{\gamma_x} (x)^{\gamma_x - 1} \exp \{ - (\lambda_x x)^{\gamma_x} \} dx = \\ &= -w^{\gamma_x} \exp \{ - (\lambda_x x)^{\gamma_x} \} \Big|_0^t = w^{\gamma_x} - w^{\gamma_x} \exp \{ - (\lambda_x t)^{\gamma_x} \} = \end{aligned}$$

$$F_t(w, t).$$

The properties of this cumulative distribution function are:

1. As $t \rightarrow 0$, $F_t \rightarrow 0$.
2. As $t \rightarrow 180$, $F_t \rightarrow w^{\gamma_x}$.
3. F_t is right continuous.

As $t \rightarrow 180$, $F_t \rightarrow w^{\gamma_x} \Rightarrow w^{\gamma_x}$ must sum to 1. We choose

$$w'_i = \left(\frac{w_i}{\sum_{i=1}^n w_i} \right)^{1/\gamma_y}$$

where w_i is the observed value of the random variable for the latitude.

The parameter estimation for the exponential ellipse was straightforward because there was only one. The Weibull ellipse has two parameters and we have to solve for these parameters simultaneously. The procedure is similar. The maximum likelihood estimator for λ_x occurs at the first derivative of $L = \prod_{i=1}^n f_x(w, x_i)$ taken with respect with λ_x . The maximum likelihood estimator for γ_x occurs at the first derivative of L taken with respect with γ_x . x_1, x_2, \dots, x_n is the random sample. Equation (7.2) gives the likelihood function.

$$L(w, x_i) = \gamma_x^n (w\lambda_x)^{n\gamma_x} \left(\prod_{i=1}^n x_i \right)^{\gamma_x - 1} \exp \left\{ -\lambda_x^{\gamma_x} \sum_{i=1}^n x_i^{\gamma_x} \right\} \quad (7.2)$$

(x_1, x_2, \dots, x_n) is the minimum set of sufficient statistics. Equation (7.3) gives the logarithm of the likelihood function.

$$\log L = n \log \gamma_x + n\gamma_x \log(w\lambda_x) + \sum_{i=1}^n (\gamma_x - 1) \log x_i - \quad (7.3)$$

$$\lambda_x^{\gamma_x} \sum_{i=1}^n x_i^{\gamma_x}.$$

Taking the usual derivatives with respect to the parameters, $\frac{d \log L}{d \lambda_x} = 0$, and $\frac{d \log L}{d \gamma_x} = 0$ yields the following two equations.

$$\frac{d \log L}{d \lambda_x} = 0 \Rightarrow$$

$$\frac{n\gamma_x}{w\lambda_x} - \gamma_x \lambda_x^{\gamma_x - 1} \sum_{i=1}^n x_i^{\gamma_x} = 0$$

$$n\gamma_x - w\gamma_x \lambda_x^{\gamma_x} \sum_{i=1}^n x_i^{\gamma_x} = 0$$

$$n - w\lambda_x^{\gamma_x} \sum_{i=1}^n x_i^{\gamma_x} = 0 \quad (7.4)$$

and

$$\begin{aligned} \frac{d \log L}{d \gamma_x} = 0 \Rightarrow \\ \frac{n}{\gamma_x} + n \log(w\lambda_x) + \sum_{i=1}^n \log x_i - \\ \lambda_x^{\gamma_x} \sum_{i=1}^n x_i^{\gamma_x} (\log \lambda_x + \log x_i) = 0 = L'(w, x_i). \end{aligned} \quad (7.5)$$

Solving Equation (7.4) for λ_x yields

$$\lambda_x = \left(\frac{n}{w \sum x_i^{\gamma_x}} \right)^{\frac{1}{\gamma_x}}. \quad (7.6)$$

We must solve for γ_x using numerical algorithms which use Equations (7.3), (7.5), (7.6). Using [Lee 35, Appendix A], we apply the Newton-Raphson method. Instead of working with the Jacobian matrix of derivatives, we substitute Equation (7.6) into Equation (7.5).

Then we wish to iterate through the following expression using the variable k .

$$\gamma_{(x,k+1)} = \hat{\gamma}_{(x,k)} - \frac{L(w, x_i)}{L'(w, x_i)}.$$

An alternative to using the Newton-Raphson method is to apply the Secant method. This only involves using Equation (7.3). We apply it as follow:

$$\gamma_{(k+1)} = \quad (7.7)$$

$$\gamma_{(k)} - L \left[\hat{\lambda}_{(k)}, \gamma_{(k)} \right] \left(\frac{\gamma_{(k)} - \gamma_{(k-1)}}{L \left[\hat{\lambda}_{(k)}, \gamma_{(k)} \right] - L \left[\hat{\lambda}_{(k-1)}, \gamma_{(k-1)} \right]} \right)$$

We need to add an additional subscript x to denote the latitude to identify it from the longitude y . So, the equation for the Secant method becomes:

$$\begin{aligned} \gamma_{(x,k+1)} = \gamma_{(x,k)} - \\ L \left[\hat{\lambda}_{(x,k)}, \gamma_{(x,k)} \right] \left(\frac{\gamma_{(x,k)} - \gamma_{(x,k-1)}}{L \left[\hat{\lambda}_{(x,k)}, \gamma_{(x,k)} \right] - L \left[\hat{\lambda}_{(x,k-1)}, \gamma_{(x,k-1)} \right]} \right) \end{aligned} \quad (7.8)$$

where k denotes the iteration.

With the Secant method, we must choose two initial values for γ_x instead of one. The reader will find that working with either method is difficult to work with for

finding the value of λ_x . Knowledge of the functions L , $\log L$, and the derivatives are paramount. The Secant method is the easiest way out of these derivations and initialization. However, more initial values are required on the reader's part for the Secant method.

The relationship of the parameter estimate $\hat{\lambda}_x$ in Equations (7.6) and (7.8) to the weighted mean center is

$$\bar{x} = \frac{n}{\hat{\lambda}_x} \quad (7.9)$$

Notice that it depends on the range of λ_x . This textbook provides Excel 2010 Visual Basic for Applications (VBA) code to calculate the weighted means based these derivations. We must state the initial values of γ_x in addition to the maximum number of iterations. Some trial and error is required to obtain valid results. Other authors' recommendations include setting the two initial values of γ_x fairly far apart to obtain convergence. Both values must be greater than one. If the reader is in doubt about the number of iterations, then try 1; then 2; then 3; and so on. It only took seven tries to obtain valid results in the Kentucky example.

7.2 Longitude Mean Center

The longitude data can range from -180° to 180° . The Weibull distribution does not allow negative numbers. However, if all of the longitude observations are negative, then it is simple to change the sign to positive to perform parameter estimation. Afterwards, we can change the sign on the mean center. If this is not the case, some data are negative, and some are positive, then convert the observations to circular coordinate system where all of the observations are positive.

Equation (7.10) gives the weighted probability density function for the longitude.

$$f_y(w_i, y_i) = \gamma_y (w_i \lambda_y)^{\gamma_y} (y_i)^{\gamma_y - 1} \exp \{ - (\lambda_y y_i)^{\gamma_y} \} \quad (7.10)$$

$$0 \leq y_i \leq 360, \lambda_y > 0, \gamma_y > 0.$$

We need the cumulative distribution function $F_y(w, y_i)$. Therefore, we integrate the probability density function $f_y(w, y_i)$.

$$\begin{aligned} F_y(w, y) &= \int_0^y f(w, y) dy = \\ &= \int_0^y \gamma_y (w \lambda_y)^{\gamma_y} (y_i)^{\gamma_y - 1} \exp \{ - (\lambda_y y_i)^{\gamma_y} \} dy = \\ &= -w^{\gamma_y} \exp \{ - (\lambda_y y)^{\gamma_y} \} \Big|_0^y = w^{\gamma_y} - w^{\gamma_y} \exp \{ - (\lambda_y y)^{\gamma_y} \} = \\ &= F_u(w, u). \end{aligned}$$

The properties of this cumulative distribution function are:

1. As $u \rightarrow 0$, $F_u \rightarrow 0$.
2. As $u \rightarrow 360$, $F_u \rightarrow w^{\gamma_y}$.
3. F_u is right continuous.

As $u \rightarrow 360$, $F_u \rightarrow w^{\gamma_y} \Rightarrow w^{\gamma_y}$ must sum to 1. We choose

$$w'_i = \left(\frac{w_i}{\sum_{i=1}^n w_i} \right)^{1/\gamma_y}$$

where w_i is the observed value of the random variable for the longitude.

The parameter estimation for the exponential ellipse was straightforward because there was only one. The Weibull ellipse has two parameters and we have to solve for the parameters simultaneously. The procedure is similar. The maximum likelihood estimator for λ_y occurs at the first derivative of $L = \prod_{i=1}^n f_y(w, y_i)$ taken with respect with λ_y . The maximum likelihood estimator for γ_y occurs at the first derivative of L taken with respect with γ_y . y_1, y_2, \dots, y_n is the random sample. Equation (7.11) gives the likelihood function.

$$L(w, y_i) = \gamma_y^n (w \lambda_y)^{n \gamma_y} \left(\prod_{i=1}^n y_i \right)^{\gamma_y - 1} \exp \left\{ -\lambda_y^{\gamma_y} \sum_{i=1}^n y_i^{\gamma_y} \right\} \quad (7.11)$$

(y_1, y_2, \dots, y_n) is the minimum set of sufficient statistics.

Equation (7.12) gives the logarithm of the likelihood function.

$$\log L = n \log \gamma_y + n \gamma_y \log(w \lambda_y) + \sum_{i=1}^n (\gamma_y - 1) \log y_i - \lambda_y^{\gamma_y} \sum_{i=1}^n y_i^{\gamma_y}. \quad (7.12)$$

Taking the usual derivatives with respect to the parameters, $\frac{d \log L}{d \lambda_y} = 0$, and $\frac{d \log L}{d \gamma_y} = 0$ yields the following two equations.

$$\begin{aligned} \frac{d \log L}{d \lambda_y} &= 0 \Rightarrow \\ \frac{n \gamma_y}{w \lambda_y} - \gamma_y \lambda_y^{\gamma_y - 1} \sum_{i=1}^n y_i^{\gamma_y} &= 0 \\ n \gamma_y - w \gamma_y \lambda_y^{\gamma_y} \sum_{i=1}^n y_i^{\gamma_y} &= 0 \\ n - w \lambda_y^{\gamma_y} \sum_{i=1}^n y_i^{\gamma_y} &= 0 \end{aligned} \quad (7.13)$$

and

$$\begin{aligned} \frac{d \log L}{d \gamma_y} &= 0 \Rightarrow \\ \frac{n}{\gamma_y} + n \log(w \lambda_y) + \sum_{i=1}^n \log y_i - \\ \lambda_y^{\gamma_y} \sum_{i=1}^n y_i^{\gamma_y} (\log \lambda_y + \log y_i) &= 0 = L'(w, y_i). \end{aligned} \quad (7.14)$$

Solving Equation (7.13) for λ_y yields

$$\lambda_y = \left(\frac{n}{w \sum y_i^{\gamma_y}} \right)^{\frac{1}{\gamma_y}}. \quad (7.15)$$

γ_y must be solved numerically using Equations (7.12), (7.14), (7.15). Using [Lee 35, Appendix A], we apply the Newton-Raphson method. Instead of working with the Jacobian matrix of derivatives, we substitute Equation (7.15) into Equation (7.14).

Then we wish to iterate through the following expression using the variable k .

$$\gamma_{(y,k+1)} = \hat{\gamma}_{(y,k)} - \frac{L(w, y_i)}{L'(w, y_i)}.$$

An alternative to using the Newton-Raphson method is to apply the Secant method. This only involves using Equation (7.12). We apply it as follow:

$$\begin{aligned} \gamma_{(k+1)} &= \gamma_{(k)} - \log L \left[\hat{\lambda}_{(k)}, \gamma_{(k)} \right] \\ &\left(\frac{\gamma_{(k)} - \gamma_{(k-1)}}{\log L \left[\hat{\lambda}_{(k)}, \gamma_{(k)} \right] - \log L \left[\hat{\lambda}_{(k-1)}, \gamma_{(k-1)} \right]} \right) \end{aligned} \quad (7.16)$$

We need to add an additional subscript y to denote the longitude to identify it from the latitude y . So, the equation for the Secant method becomes:

$$\begin{aligned} \gamma_{(y,k+1)} &= \gamma_{(y,k)} - \log L \left[\hat{\lambda}_{(y,k)}, \gamma_{(y,k)} \right] \\ &\left(\frac{\gamma_{(y,k)} - \gamma_{(y,k-1)}}{\log L \left[\hat{\lambda}_{(y,k)}, \gamma_{(y,k)} \right] - \log L \left[\hat{\lambda}_{(y,k-1)}, \gamma_{(y,k-1)} \right]} \right) \end{aligned} \quad (7.17)$$

where k denotes the iteration.

With the Secant method, we must choose two initial values for γ_y instead of one. The reader will find that working with either method is difficult to work with for finding the value of λ_y . Knowledge of the functions L , $\log L$, and the derivatives are paramount. The Secant method is the easiest way out of these derivations and initialization. However, more initial values are required on the reader's part for the Secant method.

Equations (7.15) and (7.8) give the relationship of the parameter estimate $\hat{\lambda}_y$ to the weighted mean center.

$$\bar{y} = \frac{n}{\hat{\lambda}_y} \quad (7.18)$$

Notice that the relationship depends on the range of λ_y if we do not use circular coordinates. If we use circular coordinates, then we can remove the absolute value operator on the mean center \bar{y} .

7.3 VBA

The `secant1()` subroutine (2) calculates the $\hat{\gamma}_x$ and $\hat{\gamma}_y$ estimates for the latitude and longitude using Equations (7.8) and (7.17). The `WHILE-WEND` loop (12) has two termination conditions:

1. The loop can terminate when it reaches a specified number of iterations. The user sets this condition with a parameter to the subroutine.
2. The loop can terminate when the parameter estimate for the latitude $\hat{\gamma}_x$ converges on two consecutive iterations to within 10^{-6} . The same concept applies to the parameter estimate for the longitude $\hat{\gamma}_y$. The user can not set this condition with a parameter to the subroutine.

The first `FOR-NEXT` loop (31) calculates the sum of the weights. In the Weibull model, the weights are dependent on the parameter estimates $\hat{\gamma}_x$, $\hat{\gamma}_y$, and $\hat{\gamma}_{xy}$. The second `FOR-NEXT` loop (32) initializes the observation weights on the active spreadsheet. The final values of the observation weights cannot be determined yet because we do not have the final values for $\hat{\gamma}_x$ and $\hat{\gamma}_y$.

The `FOR-NEXT` loop (33) calculates the sums needed for the parameter estimates $\hat{\lambda}_x$ and $\hat{\lambda}_y$. The VBA statements following this loop calculate the parameter estimate $\hat{\lambda}_x$ or $\hat{\lambda}_y$ and store the result in the variable `lambda2` and `lambda1`. This reflects the two iterations, not that the subroutine simultaneously solves for both the latitude and the longitude.

The remaining VBA statements implement Equation (7.8) for the latitude and Equation (7.17) for the longitude. More specifically, the `WHILE-END` loop terminates when $dx = |L_{x_k} - L_{x_{k-1}}| \leq 10^{-6} |L_{x_k}|$.

The `WITH-END-WITH` statement (13) saves the parameter estimates $\hat{\gamma}_x$, or $\hat{\gamma}_y$ to the spreadsheet `STATS`. It saves the parameter estimate $\hat{\lambda}_x$ or $\hat{\lambda}_y$. Additionally, the subroutine saves the variables `dx` and `condition` for debugging purposes.

```

2  Sub secant1(iterations, variable, Gamma1, gamma2)
    j = 1
    dx = 1 'trivial value
    condition = 0 'trivial value

```



```

2 12 25 While (j <= iterations) And (dx > condition)
    With ActiveSheet
        'set the weights
        n = Selection.Rows.Count
        sum_weight1 = 0
        sum_weight2 = 0
        For i = 2 To n
            31 sum_weight1 = sum_weight1 + (.Cells(i, 3).Value)
            sum_weight2 = sum_weight2 + (.Cells(i, 3).Value)
        Next
        .Cells(1, 4).Value = "Lambda 1 Weights"
        .Cells(1, 6).Value = "Lambda 2 Weights"
        For i = 2 To n
            32 .Cells(i, 4).Value = ((.Cells(i, 3).Value) / sum_weight1) ^ (1 / Gamma1) 'previous iteration
            .Cells(i, 6).Value = ((.Cells(i, 3).Value) / sum_weight2) ^ (1 / gamma2) 'next iteration
        Next
        'end of setting the weights
        sumw_x2 = 0
        sumw_x1 = 0
        sum_x2 = 0
        sum_x1 = 0
        sum_logx = 0
        For i = 2 To n
            33 sumw_x2 = sumw_x2 + (.Cells(i, 6) ^ gamma2 * .Cells(i, variable).Value) ^ gamma2 'weighted
            sumw_x1 = sumw_x1 + (.Cells(i, 4) ^ Gamma1 * .Cells(i, variable).Value) ^ Gamma1 'weighted
            sum_x2 = sum_x2 + .Cells(i, variable).Value ^ gamma2 'unweighted
            sum_x1 = sum_x1 + .Cells(i, variable).Value ^ Gamma1 'unweighted
            sum_logx = sum_logx + Log(Abs(.Cells(i, variable).Value)) 'unweighted
        Next
        lambda2 = ((n - 1) / sumw_x2) ^ (1 / gamma2) 'weighted
        lambda1 = ((n - 1) / sumw_x1) ^ (1 / Gamma1) 'weighted
        L_x = 0
        L_X_Minus = 0
        {
            L_x = L_x + (n - 1) * Log(gamma2) + (n - 1) * gamma2 * Log(sum_weight2 * lambda2) + _
            (gamma2 - 1) * sum_logx - lambda2 ^ gamma2 * sum_x2
            L_X_Minus = L_X_Minus + (n - 1) * Log(Gamma1) + (n - 1) * Gamma1 * Log(sum_weight1 * lambda1) + _
            (Gamma1 - 1) * sum_logx - lambda1 ^ Gamma1 * sum_x1
        }
        gamma3 = gamma2 - L_x * (gamma2 - Gamma1) / (L_x - L_X_Minus)
        Gamma1 = gamma2 'correct assignment
        gamma2 = Abs(gamma3) 'correct assignment, must be positive
        dx = Abs(L_x - L_X_Minus)
        condition = 10 ^ (-6) * Abs(L_x)
        j = j + 1
    End With
Wend
End Sub

```

7.4 Joint Distribution

The joint distribution between two weighted Weibull models need to be determined. Equations (7.1) and (7.10) gives the two probability density functions for the joint likelihood function L .

$$L = \prod_{i=1}^n f_x(w, x_i) f_y(w, y_i) =$$

$$\prod_{i=1}^n \gamma_x (w \lambda_x)^{\gamma_x} (x_i)^{\gamma_x - 1} \exp \{ - (\lambda_x x_i)^{\gamma_x} \}$$

$$\gamma_y (w \lambda_y)^{\gamma_y} (y_i)^{\gamma_y - 1} \exp \{ - (\lambda_y y_i)^{\gamma_y} \}.$$

Let $\lambda_x = \lambda_y = \lambda_{xy}$, and let $\gamma_x = \gamma_y = \gamma_{xy}$. Then, the joint likelihood function becomes,

$$L(w, x_i, y_i) =$$

$$\prod_{i=1}^n \gamma_{xy} (w \lambda_{xy})^{\gamma_{xy}} (x_i)^{\gamma_{xy} - 1} \exp \{ - (\lambda_{xy} x_i)^{\gamma_{xy}} \}$$

$$\gamma_{xy} (w \lambda_{xy})^{\gamma_{xy}} (y_i)^{\gamma_{xy} - 1} \exp \{ - (\lambda_{xy} y_i)^{\gamma_{xy}} \} =$$

$$\prod_{i=1}^n \gamma_{xy}^2 (w \lambda_{xy})^{2\gamma_{xy}} (x_i y_i)^{\gamma_{xy} - 1} \exp \{ - (\lambda_{xy} x_i)^{\gamma_{xy}} - (\lambda_{xy} y_i)^{\gamma_{xy}} \},$$

$$\lambda_{xy} > 0, \quad \gamma_{xy} > 0.$$

Next, we find the maximum likelihood estimator of λ_{xy} .

$$\log L(w, x_i, y_i) =$$

$$2 \sum_{i=1}^n \log \gamma_{xy} + 2n \gamma_{xy} \log(w \lambda_{xy}) + (\gamma_{xy} - 1) \sum_{i=1}^n \log x_i y_i - \quad (7.19)$$

$$\sum_{i=1}^n (\lambda_{xy} x_i)^{\gamma_{xy}} - \sum_{i=1}^n (\lambda_{xy} y_i)^{\gamma_{xy}}.$$

$$\frac{d \log L}{d \lambda_{xy}} = 0 + \frac{2n \gamma_{xy}}{w \lambda_{xy}} + 0 + 0 -$$

$$\gamma_{xy} \sum_{i=1}^n (\lambda_{xy} x_i)^{\gamma_{xy} - 1} - \gamma_{xy} \sum_{i=1}^n (\lambda_{xy} y_i)^{\gamma_{xy} - 1} = 0,$$

$$2n\gamma_{xy} - w\lambda_{xy}^{\gamma_{xy}}\gamma_{xy} \sum_{i=1}^n (x_i)^{\gamma_{xy}} - w\lambda_{xy}^{\gamma_{xy}}\gamma_{xy} \sum_{i=1}^n (y_i)^{\gamma_{xy}} = 0$$

Solving for λ_{xy} yields the following expressions.

$$\begin{aligned} -w\gamma_{xy}\lambda_{xy}^{\gamma_{xy}} \sum_{i=1}^n x_i^{\gamma_{xy}} - w\gamma_{xy}\lambda_{xy}^{\gamma_{xy}} \sum_{i=1}^n y_i^{\gamma_{xy}} &= -2n\gamma_{xy} = \\ \lambda_{xy}^{\gamma_{xy}} \left[w \left(\sum_{i=1}^n x_i^{\gamma_{xy}} + \sum_{i=1}^n y_i^{\gamma_{xy}} \right) \right] &= 2n \\ \hat{\lambda}_{xy} = 2n / \left[w \left(\sum_{i=1}^n x_i^{\gamma_{xy}} + \sum_{i=1}^n y_i^{\gamma_{xy}} \right) \right]^{1/(\gamma_{xy})} &. \end{aligned} \quad (7.20)$$

Equation (7.20) gives the maximum likelihood estimator for λ_{xy} for the joint distribution of the latitude and longitude. Using the Secant algorithm, we find the maximum likelihood estimates for λ_{xy} and γ_{xy} as before. We apply the Secant algorithm to Equation (7.21) to find $\hat{\gamma}_{xy}$.

$$\gamma_{(xy,k+1)} = \gamma_{(xy,k)} - \log L \left[\hat{\lambda}_{(xy,k)}, \gamma_{(xy,k)} \right] \quad (7.21)$$

$$\left(\frac{\gamma_{(xy,k)} - \gamma_{(xy,k-1)}}{\log L \left[\hat{\lambda}_{(xy,k)}, \gamma_{(xy,k)} \right] - \log L \left[\hat{\lambda}_{(xy,k-1)}, \gamma_{(xy,k-1)} \right]} \right)$$

where k denotes the iteration, Equation (7.19) gives $\log L$, and Equation (7.20) gives $\hat{\lambda}_{xy}$.

The joint probability density function of the latitude and longitude is

$$\begin{aligned} f_{xy}(w_i, x_i, y_i) &= \\ (\gamma_{xy})^2 (w_i \lambda_{xy})^{2\gamma_{xy}} (x_i y_i)^{\gamma_{xy}-1} \exp \{ -(\lambda_{xy} x_i)^{\gamma_{xy}} - (\lambda_{xy} y_i)^{\gamma_{xy}} \}. \end{aligned}$$

We need to find the cumulative distribution function $F_{xy}(w_i, x_i, y_i)$ for the joint distribution, next. We need the joint distribution function so that we can derive the *Weibull* standard deviational ellipse.

$$\begin{aligned} F_{xy}(w, x, y) &= \\ \int_0^t \int_0^u (\gamma_{xy})^2 (w \lambda_{xy})^{2\gamma_{xy}} (xy)^{\gamma_{xy}-1} \times \\ \exp \{ -(\lambda_{xy} x)^{\gamma_{xy}} - (\lambda_{xy} y)^{\gamma_{xy}} \} dy dx &= \end{aligned}$$

$$\begin{aligned}
& \int_0^t -(\gamma_{xy}) (w \lambda_{xy})^{\gamma_{xy}} (x)^{\gamma_{xy}-1} \times \\
& \exp \left\{ -(\lambda_{xy} x)^{\gamma_{xy}} - (\lambda_{xy} y)^{\gamma_{xy}} \right\} \Big|_0^u dx = \\
& \int_0^t (\gamma_{xy}) (w \lambda_{xy})^{\gamma_{xy}} (x)^{\gamma_{xy}-1} \exp \left\{ -(\lambda_{xy} x)^{\gamma_{xy}} - (\lambda_{xy} u)^{\gamma_{xy}} \right\} - \\
& (\gamma_{xy}) (w \lambda_{xy})^{\gamma_{xy}} (x)^{\gamma_{xy}-1} \exp \left\{ -(\lambda_{xy} x)^{\gamma_{xy}} \right\} dx = \\
& w^{\gamma_{xy}} \exp \left\{ -(\lambda_{xy} x)^{\gamma_{xy}} - (\lambda_{xy} u)^{\gamma_{xy}} \right\} - \\
& w^{\gamma_{xy}} \exp \left\{ -(\lambda_{xy} x)^{\gamma_{xy}} \right\} \Big|_0^t = \\
& w^{\gamma_{xy}} \exp \left\{ -(\lambda_{xy} t)^{\gamma_{xy}} - (\lambda_{xy} u)^{\gamma_{xy}} \right\} - \\
& w^{\gamma_{xy}} \exp \left\{ -(\lambda_{xy} t)^{\gamma_{xy}} \right\} - w^{\gamma_{xy}} \exp \left\{ -(\lambda_{xy} u)^{\gamma_{xy}} \right\} + w^{\gamma_{xy}} \\
& = F_{tu}(w, t, u)
\end{aligned} \tag{7.22}$$

Is Equation (7.22) a valid cumulative distribution function?

1. As $t \rightarrow 0$ and $u \rightarrow 0$, $F_{tu} \rightarrow 0$.
2. As $t \rightarrow 180$ and $u \rightarrow 360$, $F_{tu} \rightarrow w^{\gamma_{xy}}$.
3. F_{tu} is right continuous.

This is a valid cumulative distribution function only if the weights $w^{\gamma_{xy}}$ sum to 1 as $t \rightarrow 180$ and $u \rightarrow 360$. We choose the weights

$$w'_i = \left(\frac{w_i}{\sum_{i=1}^n w_i} \right)^{1/\gamma_{xy}}$$

where w_i is the observed value of the random variable. It does not change for latitude and longitude. Then it is a trivial fact that

$$w^{\gamma_{xy}} = \sum_{i=1}^n w_i^{\gamma_{xy}} = \sum_{i=1}^n \frac{w_i}{\sum_{i=1}^n w_i} = 1.$$

7.5 VBA

The `secant2()` subroutine (3) calculates the γ_{xy} estimate for the joint distribution between the latitude and longitude using Equation (7.21). The looping structures are similar to the `secant1()` subroutine. The `WHILE-WEND` loop (14) has two termination conditions:

1. The loop can terminate when it reaches a specified number of iterations. The user sets this condition with a parameter to the subroutine.
2. The loop can terminate when the parameter estimate for the joint distribution between the latitude and longitude, $\hat{\gamma}_{xy}$, converges on two consecutive iterations to within 10^{-6} . The user can not set this condition with a parameter to the subroutine.

The first `FOR-NEXT` loop (34) calculates the sum of the weights. The weights are dependent on the parameter estimate $\hat{\gamma}_{xy}$. The second `FOR-NEXT` loop (35) initializes the observation weights on the active spreadsheet. The final values of the observation weights cannot be determined yet because we do not have the final value for $\hat{\gamma}_{xy}$.

The `FOR-NEXT` loop (35) calculates the initial values of the weights and saves them in the active spreadsheet in column D and column F.

The next two `FOR-NEXT` loops (36, 37) calculate sums required to find $\hat{\lambda}_{xy}$ and $\hat{\gamma}_{xy}$. The two VBA statements between the `FOR-NEXT` loops (36, 37) calculate the mean for the joint distribution for two iterations and save the values in the variables `lambda2` and `lambda1`.

The `FOR-NEXT` loop (38) calculates the sums for two iterations of the likelihood function L . The VBA statements following this loop actually calculate the value $\hat{\gamma}_{xy}$. The `WHILE-END` loop terminates when $dx = |L_{xy_k} - L_{xy_{k-1}}| \leq 10^{-6}|L_{xy_k}|$.

The `WITH-END-WITH` statement (15) saves the parameter estimates $\hat{\gamma}_{xy}$ and $\hat{\lambda}_{xy}$ to the spreadsheet `STATS`. The subroutine saves the variables `dx` and `condition` for debugging purposes.

```

Sub secant2(iterations, Gamma1, gamma2)
j = 1
dx = 1 'trivial value
condition = 0 'trivial value

    While (j <= iterations) And (dx > condition)
        With ActiveSheet
            'set the weights
            n = Selection.Rows.Count
            sum_weight1 = 0
            sum_weight2 = 0

            For i = 2 To n
34         sum_weight1 = sum_weight1 + (.Cells(i, 3).Value)
            sum_weight2 = sum_weight2 + (.Cells(i, 3).Value)
            Next

            .Cells(1, 4).Value = "Lambda 1 Weights"
            .Cells(1, 6).Value = "Lambda 2 Weights"

            For i = 2 To n
35         .Cells(i, 4).Value = ((.Cells(i, 3).Value) / sum_weight1) ^ (1 / Gamma1) 'previous iteration
            .Cells(i, 6).Value = ((.Cells(i, 3).Value) / sum_weight2) ^ (1 / gamma2) 'next iteration
            Next

            'end of setting the weights
            sumw_x2 = 0
            sumw_y2 = 0
            sum_x1 = 0
            sum_y1 = 0
            log_gamma1 = 0
            log_gamma2 = 0
            log_xy1 = 0
            log_xy2 = 0
            sum_lambdax2 = 0
            sum_lambdax1 = 0
            sum_lambday2 = 0
            sum_lambday1 = 0
            sumw_lambda2 = 0
            sumw_lambda1 = 0

```

```

'calculate the sums used in the Secant Method
  For i = 2 To n
    sumw_x2 = sumw_x2 + (.Cells(i, 6).Value ^ gamma2 * .Cells(i, 1).Value) ^ gamma2 'weighted
    sumw_y2 = sumw_y2 + (.Cells(i, 6).Value ^ gamma2 * Abs(.Cells(i, 2).Value)) ^ gamma2 'weighted
    sumw_x1 = sumw_x1 + (.Cells(i, 4).Value ^ Gamma1 * .Cells(i, 1).Value) ^ Gamma1 'weighted
    sumw_y1 = sumw_y1 + (.Cells(i, 4).Value ^ Gamma1 * Abs(.Cells(i, 2).Value)) ^ Gamma1 'weighted
36 log_gamma2 = log_gamma2 + 2 * Log(gamma2)
    log_gamma1 = log_gamma1 + 2 * Log(Gamma1)
    log_xy2 = log_xy2 + (gamma2 - 1) * Log(.Cells(i, 1).Value * Abs(.Cells(i, 2).Value))
    log_xy1 = log_xy1 + (Gamma1 - 1) * Log(.Cells(i, 1).Value * Abs(.Cells(i, 2).Value))
  Next

lambda2 = (2 * (n - 1) / (sumw_x2 + sumw_y2)) ^ (1 / gamma2) 'weighted
lambda1 = (2 * (n - 1) / (sumw_x1 + sumw_y1)) ^ (1 / Gamma1) 'weighted
'calculate the sums involving lambda

  For i = 2 To n
    sum_lambdax2 = sum_lambdax2 + (lambda2 * .Cells(i, 1).Value) ^ gamma2
    sum_lambdax1 = sum_lambdax1 + (lambda1 * .Cells(i, 1).Value) ^ Gamma1
    sum_lambday2 = sum_lambday2 + (lambda2 * Abs(.Cells(i, 2).Value)) ^ gamma2
    sum_lambday1 = sum_lambday1 + (lambda1 * Abs(.Cells(i, 2).Value)) ^ Gamma1
14 26 37 sumw_lambda2 = sumw_lambda2 + 2 * (n - 1) * gamma2 * Log(.Cells(i, 6).Value * lambda2)
    sumw_lambda1 = sumw_lambda1 + 2 * (n - 1) * Gamma1 * Log(.Cells(i, 4).Value * lambda1)
  Next

'estimate gamma using the Secant Method
L_x = 0
L_X_Minus = 0

  For i = 2 To n
38 L_x = L_x + log_gamma2 + sumw_lambda2 + log_xy2 - sum_lambdax2 - sum_lambday2
    L_X_Minus = L_X_Minus + log_gamma1 + sumw_lambda1 + log_xy1 - sum_lambdax1 - sum_lambday1
  Next

gamma3 = gamma2 - L_x * (gamma2 - Gamma1) / (L_x - L_X_Minus)
Gamma1 = gamma2 'correct assignment
gamma2 = Abs(gamma3) 'correct assignment, must be positive
dx = Abs(L_x - L_X_Minus)
condition = 10 ^ (-6) * Abs(L_x)
j = j + 1
End With
Wend
End Sub

```

7.6 Weibull Ellipse

This Section will derive the *Weibull distributed, weighted, deviational ellipse*. The weighted mean estimates (\bar{x}, \bar{y}) under the general linear means model in Section 5 and the Weibull distribution are exactly the same — except possibly the sign on the longitude only if γ_x and γ_y both equal to 1. Generally, this is not the case. The weighted mean estimates (\bar{x}, \bar{y}) under the general linear means model and the Weibull distribution do not equal each other.

Graphically, if the data has a straight line with the following transformation, then it comes from a Weibull distribution.

$$\log t = \log(1/\lambda) + (1/\gamma) \log \left\{ \log_e \left[\frac{1}{1 - F} \right] \right\}$$

[Lee (35), p 167] shows a sheet of graph paper for estimating the parameters for the Weibull distribution. It is possible to Xerox that page, whiteout her example, and plot your data. That is what this author did in graduate school. For small data sets, this may seem plausible at first. It is best to run the VBA code to estimate the parameters.

The Weibull ellipse uses the following formulas.

$$\log t_x = \sum_{i=1}^n \log \left[\frac{1}{\hat{\lambda}_x} \right] + \left[\frac{1}{\hat{\gamma}_x} \right] \log \left\{ \log_e \left[\frac{1}{1 - F_t(t_i)} \right] \right\}. \quad (7.23)$$

$$\log u_y = \sum_{i=1}^n \log \left[\frac{1}{\hat{\lambda}_y} \right] + \left[\frac{1}{\hat{\gamma}_y} \right] \log \left\{ \log_e \left[\frac{1}{1 - F_u(u_i)} \right] \right\}. \quad (7.24)$$

$$\log t_{xy} = \sum_{i=1}^n \log \left[\frac{1}{\hat{\lambda}_{xy}} \right] + \left[\frac{1}{\hat{\gamma}_{xy}} \right] \log \left\{ \log_e \left[\frac{1}{1 - F_{tu}(t_i, u_i)} \right] \right\}. \quad (7.25)$$

The two ellipses differ (Exponential and Weibull) differ between the sums of squares of the original data $(w_i x_i, w_i y_i)$ and the transformed data $(\log t_{(x,i)}, \log u_{(y,i)})$.

Equation (7.25) gives the mean joint distribution estimate between the weighted latitude and weighted longitude estimates. The estimators for γ_x , γ_y , and γ_{xy} appear in Equations (7.8), (7.17) and (7.21). We had to perform Secant iterations to find these. The λ 's are already weighted. Therefore, the term w_i is absent from the formulas. The estimators for λ_x , λ_y , and λ_{xy} appear in Equations (7.6), (7.15), and (7.20). These estimators had closed forms *given their respective values for gamma*. Thus, we did not need to perform additional Secant iterations for these values.

7.6.1 Ellipse Angle

Equation (7) in [Yuill, (67)] is general enough to obtain a weighted, Weibull standard deviational ellipse. We need to substitute the original sums of squares with the new

sums of squares and re-calculate the major and minor axes. These axes lengths will determine the total area from which we can compare the area and number of weighted points inside the ellipse. Additionally, we can compare the standard deviations and the eccentricity of the ellipses.

Using equations (7.23), (7.24), and (7.25), we get our *Weibull distributed, weighted, deviational ellipse* in Equation (7.26).

$$\tan \theta = -\frac{(\sum_{i=1}^n \log t_{x,i}^2 - \sum_{i=1}^n \log u_{y,i}^2)}{2 \sum_{i=1}^n \log t_{x,y,i}} \pm \frac{\sqrt{(\sum_{i=1}^n \log t_{x,i}^2 - \sum_{i=1}^n \log u_{y,i}^2)^2 + 4 (\sum_{i=1}^n \log t_{x,y,i})^2}}{2 \sum_{i=1}^n \log t_{x,y,i}} \quad (7.26)$$

This textbook provides an Excel 2010 Visual Basic for Applications (VBA) program that calculates the weighted, Weibull ellipse. In addition to the angles of rotation, it calculates the mean center, the area, the standard deviations, and the eccentricity.

7.7 Ellipse Properties

Equation (7.27) gives the formula for the deviations on the X axis.

$$\delta_x = \sum_{i=1}^n [\log_e(u_{y,i}) \sin \theta_a - \log_e(t_{x,i}) \cos \theta_a]^2 w_i^{\gamma_x} \quad (7.27)$$

where the subscript a denotes the angle of rotation θ about the major axis. Equation (7.28) gives the formula for the deviations on the Y axis.

$$\delta_y = \sum_{i=1}^n [\log_e(u_{y,i}) \cos \theta_b - \log_e(t_{x,i}) \sin \theta_b]^2 w_i^{\gamma_y} \quad (7.28)$$

where the subscript b denotes the angle of rotation θ about the minor axis. Equation (7.27) raises the weights w_i to the power γ_x . Thus, the weights in Equation (7.28) differ from those in Equation (7.27) because Equation (7.28) raises the weights to the power of γ_y .

Equation (7.29) gives the formula for calculating the semi-major axis length a .

$$a^2 = \frac{\sum_{i=1}^n \log u_{y,i}^2}{n} = \frac{2 (\sum_{i=1}^n \log t_{x,y,i})^2}{n[-1(\sum_{i=1}^n \log t_{x,i}^2 - \sum_{i=1}^n \log u_{y,i}^2) + \sqrt{(\sum_{i=1}^n \log t_{x,i}^2 - \sum_{i=1}^n \log u_{y,i}^2)^2 + 4 (\sum_{i=1}^n \log t_{x,y,i})^2}]} \quad (7.29)$$

Equation (7.30) gives the formula for calculating the semi-minor axis length b .

$$b^2 - \frac{\sum_{i=1}^n \log u_{y,i}^2}{n} = \quad (7.30)$$

$$\frac{2 \left(\sum_{i=1}^n \log t_{x,y,i} \right)^2}{n \left[-1 \left(\sum_{i=1}^n \log t_{x,i}^2 - \sum_{i=1}^n \log u_{y,i}^2 \right) + \sqrt{\left(\sum_{i=1}^n \log t_{x,i}^2 - \sum_{i=1}^n \log u_{y,i}^2 \right)^2 + 4 \left(\sum_{i=1}^n \log t_{x,y,i} \right)^2} \right]}$$

subject to the condition

$$\sum_{i=1}^n \log t_{x,i}^2 \geq \sum_{i=1}^n \log u_{y,i}^2.$$

Equation (7.31) gives the formula for calculating the area F .

$$F = \frac{\pi}{n} \sqrt{\left(\sum_{i=1}^n \log t_{x,i}^2 \right) \left(\sum_{i=1}^n \log u_{y,i}^2 \right) - \left(\sum_{i=1}^n \log t_{x,y,i} \right)^2} \quad (7.31)$$

We can always use the area can always use the formula $F = \pi ab$ to check the area. Finally, Equation (7.32) gives the eccentricity e .

$$e = \frac{\sqrt{a^2 - b^2}}{a} \quad (7.32)$$

where Equation (7.29) gives the semi-major axis length a and Equation (7.30) gives the semi-minor axis length b .

7.8 VBA

The function `Log10(X)` (4) calculates and returns the logarithm with the base 10 of the variable `X`. By default, VBA calculates natural logarithms. The programmer must write his own function for non-natural logarithms.

The `Weibull()` subroutine (5) calculates the remaining measures for the Weibull ellipse. The first two `FOR-NEXT` loops (16, 18) calculate the weights and save them to columns `D`, `F`, and `G` in the active spreadsheet. The source code that estimates $\hat{\lambda}_x$, $\hat{\lambda}_y$, $\hat{\lambda}_{xy}$, $\hat{\gamma}_x$, $\hat{\gamma}_y$, and $\hat{\gamma}_{xy}$ need these values initialized. The subroutines `secant1()` and `secant2()` only needed two iterations of a single weight initialized.

The `FOR-NEXT` loop (19) calculates the *sums* used in Equation (7.26), the axes lengths, the area, and the eccentricity.

The `IF-THEN-ELSE` statement (20) determines which axis to rotate the ellipse. Since θ from Equation (7.26) contains a plus and minus sign, both values are calculated. If $\sum_{i=1}^n \log t_{x,i}^2 > \sum_{i=1}^n \log u_{y,i}^2$, then rotate the ellipse about the Y-axis. Otherwise, rotate the ellipse about the X-axis.

The `FOR-NEXT` loop (21) calculates the *sums* used for the standard errors and the area. We use Equation (7.27) to calculate the sums for the standard error for the X-axis. We use Equation (7.28) to calculate the sums for the standard error for the Y-axis. The VBA statements following the `FOR-NEXT` loop finish the calculations and save the results in the variables `error_x` and `error_y`. Finally, the program calculates the area, F , in Equation (7.31) and saves it in the variable `f`.

The `IF-THEN-ELSE` statement (22) determines the major and minor axis lengths. If $\sum_{i=1}^n \log t_{x,i}^2 \geq \sum_{i=1}^n \log u_{y,i}^2$, then we can calculate the semi-major axis length directly with Equation (7.29) and the semi-minor axis length with Equation (7.30). Otherwise, we must reverse the roles of the sums in the two equations.

We calculate the eccentricity inside the `IF-THEN` statement (23). If the axis length $a \geq b$, then the formula in Equation (7.32) can be used directly. Otherwise, we reverse the roles of a and b so that we do not take the square root of a negative number.

The set of VBA statements inside the `WITH-END-WITH` (24) save the results to the spreadsheet called `STATS` in column `L` (column 12). We write appropriate text documentation to column `K` (column 11).

```

Sub Weibull(gamma_x, gamma_y, gamma_xy, lambda_x, lambda_y, lambda_xy)
'set the weights
n = Selection.Rows.Count
m = n - 1
sum_weight1 = 0
sum_weight2 = 0
sum_weight3 = 0
    For i = 2 To n
        With ActiveSheet
            sum_weight1 = sum_weight1 + (.Cells(i, 3).Value) 'single x
16 27 sum_weight2 = sum_weight2 + (.Cells(i, 3).Value) 'single y
            sum_weight3 = sum_weight3 + (.Cells(i, 3).Value) 'joint xy
        End With
    Next
    With ActiveSheet
        .Cells(1, 4).Value = "Lambda X Weights"
17 .Cells(1, 6).Value = "Lambda Y Weights"
        .Cells(1, 7).Value = "Lambda XY Weights"
    End With
5    For i = 2 To n
        With ActiveSheet
            .Cells(i, 4).Value = ((.Cells(i, 3).Value) / sum_weight1) ^ (1 / gamma_x) 'lambda_x
18 28 .Cells(i, 6).Value = ((.Cells(i, 3).Value) / sum_weight2) ^ (1 / gamma_y) 'lambda_y
            .Cells(i, 7).Value = ((.Cells(i, 3).Value) / sum_weight3) ^ (1 / gamma_xy) 'lambda_xy
        End With
    Next
'set the weights
f_x = 0
F_y = 0
F_xy = 0
L_xy = 0 'used for the log L calculation
logt_x = 0
logt_y = 0
logt_xy = 0
logt_xsqr = 0
logt_ysqr = 0
logt_xysqr = 0

```

```

5
19
29
For i = 2 To n
    With ActiveSheet
        {
            f_x = .Cells(i, 4).Value ^ gamma_x - .Cells(i, 4).Value ^ gamma_x * _
            EXP(-1 * lambda_x * (.Cells(i, 1).Value) ^ gamma_x)
            F_y = .Cells(i, 6).Value ^ gamma_y - .Cells(i, 6).Value * _
            EXP(-1 * lambda_y * Abs(.Cells(i, 2).Value) ^ gamma_y)
            F_xy = .Cells(i, 7).Value ^ gamma_xy * EXP(-1 * lambda_xy * .Cells(i, 1).Value ^ _
            gamma_xy - lambda_xy * Abs(.Cells(i, 2).Value) ^ gamma_xy) - _
            .Cells(i, 7).Value ^ gamma_xy * EXP(-1 * lambda_xy * .Cells(i, 1).Value ^ gamma_xy) - _
            .Cells(i, 7).Value ^ gamma_xy * EXP(-1 * lambda_xy * Abs(.Cells(i, 1).Value) ^ gamma_xy) + _
            .Cells(i, 7).Value ^ gamma_xy
            logt_x = logt_x + (Log10(1 / lambda_x) + (1 / gamma_x) * _
            Log10(Log(1 / (1 - f_x)))) * .Cells(i, 4).Value ^ gamma_x
            logt_y = logt_y + (Log10(1 / lambda_y) + (1 / gamma_y) * _
            Log10(Log(1 / (1 - F_y)))) * .Cells(i, 6).Value ^ gamma_y
            logt_xy = logt_xy + (Log10(1 / lambda_xy) + (1 / gamma_xy) * _
            Log10(Log(1 / (1 - F_xy)))) * .Cells(i, 7).Value ^ gamma_xy
            logt_xsqr = logt_xsqr + ((m * (Log10(1 / lambda_x) + (1 / gamma_x) * _
            Log10(Log(1 / (1 - f_x)))) * .Cells(i, 4).Value ^ gamma_x)) ^ 2
            logt_ysqr = logt_ysqr + ((m * (Log10(1 / lambda_y) + (1 / gamma_y) * _
            Log10(Log(1 / (1 - F_y)))) * .Cells(i, 6).Value ^ gamma_y)) ^ 2
            logt_xysqr = logt_xysqr + ((m * (Log10(1 / lambda_xy) + (1 / gamma_xy) * _
            Log10(Log(1 / (1 - F_xy)))) * .Cells(i, 7).Value ^ gamma_xy)) ^ 2
            L_xy = L_xy + 2 * Log(gamma_xy) + 2 * (n - 1) * gamma_xy * Log(lambda_xy * _
            .Cells(i, 4).Value) + (gamma_xy - 1) * (Log(.Cells(i, 1).Value * .Cells(i, 2).Value)) - _
            (lambda_xy * .Cells(i, 1).Value) ^ (gamma_xy) - (lambda_xy * .Cells(i, 2).Value) ^ (gamma_xy)
        }
    End With
Next
    'Calculate angles of rotation
    Call Angle_of_Rotation(logt_xsqr, logt_ysqr, logt_xy, note, atheta, itheta)

```

```

'calculate the deltas
delta_x = 0
delta_y = 0
eF_x = 0
eF_y = 0
  For i = 2 To n
    With ActiveSheet
      { eF_x = .Cells(i, 4).Value ^ gamma_x - .Cells(i, 4).Value ^ gamma_x * _
        EXP(-1 * lambda_x * (.Cells(i, 1).Value))
      { eF_y = .Cells(i, 6).Value ^ gamma_y - .Cells(i, 6).Value ^ gamma_y * _
        EXP(-1 * lambda_y * Abs(.Cells(i, 2).Value))
    21 30 elogt_x = Log10(1 / lambda_x) + (1 / gamma_x) * Log10(Log(1 / (1 - eF_x)))
    elogt_y = Log10(1 / lambda_y) + (1 / gamma_y) * Log10(Log(1 / (1 - eF_y)))
    5   { delta_x = delta_x + (elogt_y * Sin(atheta / 57.2957795) - elogt_x * _
      { Cos(atheta / 57.2957795)) ^ 2 * .Cells(i, 4).Value ^ gamma_x
      { delta_y = delta_y + (elogt_y * Cos(itheta / 57.2957795) - elogt_x * _
      { Sin(itheta / 57.2957795)) ^ 2 * .Cells(i, 6).Value ^ gamma_y
    End With
  Next
error_x = Sqr(delta_x)
error_y = Sqr(delta_y)
'calculate the area, eccentricity, and axes lengths
f = Area(m, logt_xsqr, logt_ysqr, logt_xy)
Call Axes.Length(m, logt_xsqr, logt_ysqr, logt_xy, a, b)
e = Eccentricity(a, b)
Area2 = Area.Check(a, b)
End Sub

```

7.9 Kentucky Example

This textbook provides two Secant programs because the likelihood functions for the joint distribution and the individual distributions are different. The code for the two individual distributions is general enough so that it will work on both the latitude and longitude data.

The input parameters to the Secant algorithm are as follow.

```

Call secant1(variable=1, iterations = 10, gamma1=1,
              gamma2=5)

```

Table 7.1 Results for the Weibull Model for Kentucky

2003	
Center	$(37.38297005, -86.81297917)$
Axes	$a = 3.749083127, b = 2.34002055$
Area	27.56097734 sq. mi.
Standard Deviations	$\delta_x = 1.774503013, \delta_y = 1.783855806$
Rotation	X-Axis
Orientation	$\theta_x = 89.89609928^\circ, \theta_y = -0.103900704^\circ$
Eccentricity	0.781298076

Table 7.2 Results for the Weibull Model for Kentucky

2004	
Center	$(37.40588282, -86.80669635)$
Axes	$a = 3.842076104, b = 2.392385246$
Area	28.87665946 sq. mi.
Standard Deviations	$\delta_x = 1.78033221, \delta_y = 1.789024565$
Rotation	X-Axis
Orientation	$\theta_x = 89.90357593^\circ, \theta_y = -0.096424046^\circ$
Eccentricity	0.782476324

- VARIABLE — either the column number to the latitude or the longitude. Be sure to use circular coordinates.
- ITERATIONS — the maximum number of iterations to perform.
- GAMMA1 — the first guess for γ_x or γ_y . Be consistent when working with latitude and longitude.
- GAMMA2 — the second guess for γ_x or γ_y . Again, be consistent when working with latitude and longitude.

The input parameters to the Secant algorithm for the joint distribution are very similar to those for the individual distributions.

```
Call secant2(latitude =1, longitude =5,
            iterations = 4, gamma1=1, gamma2=5);
```

The input parameters to the Weibull program are as follow.

This textbook contains the VBA code for the likelihood functions in Section ?? . The Secant programs will return the values for $\hat{\lambda}_x$, $\hat{\lambda}_y$, and $\hat{\lambda}_{xy}$. They will also give the difference between the last two iterations for convergence analysis.

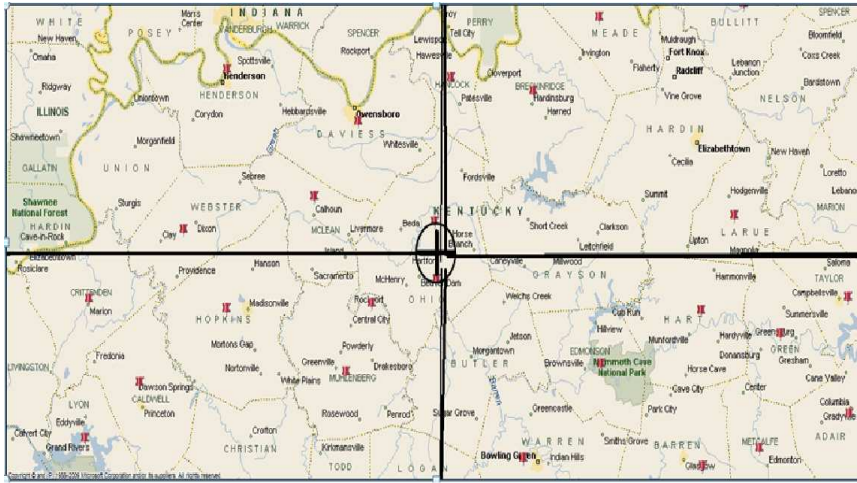


Figure 7.1 This figure shows both the Weibull deviational ellipse for the Kentucky data for the year 2003. The zoom level is 200 miles above the earth. The standard deviational ellipse and the exponential deviational ellipse were too large to fit on the map. It has the same center of gravity as the other ellipses. The Weibull ellipse is much smaller than the other two.

It took much trial and error to find the number of iterations and to set the initial parameter estimates for the Secant method. Figure 7.1 shows a map of the Weibull deviational ellipse for the Kentucky data for the year 2003. Tables 7.1 and 7.2 show the Weibull ellipse statistics for the Kentucky example for the years 2003 and 2004.

Here are some tips on whether the Secant algorithm found credible parameter estimates.

1. The literature states to halt the Secant algorithm when two iterations of the algorithm results in a parameter difference of $|\log L(\lambda, \gamma, k) - \log L(\lambda, \gamma, k - 1)| \leq \epsilon |\log L(\lambda, \gamma, k)|$. Arbitrarily choose $\epsilon = 10^{-6}$.
2. Choose two initial values for γ that are not too close to each other.
3. The weighted mean values (\bar{x}, \bar{y}) for the latitude and longitude are functions of the parameter estimates $\hat{\lambda}_x$ and $\hat{\lambda}_y$. The values (\bar{x}, \bar{y}) should be within the range of the sample data. This is a hard, must have rule.
4. When running the Secant programs, begin the iterations slowly, working up to more iterations until convergence occurs. According to the literature, performing too many iterations may indeed cause the two functions $\log L(\lambda, \gamma, k)$ and $\log L(\lambda, \gamma, k - 1)$ to converge to division by zero.

Should the reader get an error message while running the VBA code, a number of reasons can cause this.

1. Mouse over the variables `L_X` and `L_X_Minus`. Check if the two values are the same. If so, set the number of iterations to a lower number.

Table 7.3 Results for the Weibull Model on Violent Crime in the U.S.

2007

Center	(36.87659982, -92.11312648)
Axes	$a = 2.684221693$, $b = 1.421680712$
Area	11.98865122 sq. mi.
Standard Deviations	$\delta_x = 1.305334214$, $\delta_y = 1.320237824$
Rotation	X-Axis
Orientation	$\theta_x = 89.79107974^\circ$, $\theta_y = -0.208920244^\circ$
Eccentricity	0.848220318

Table 7.4 Results for the Weibull Model on Violent Crime in the U.S.

2008

Center	(36.88957315, -92.03222761)
Axes	$a = 2.653413305$, $b = 1.407222952$
Area	11.73053137 sq. mi.
Standard Deviations	$\delta_x = 1.306416749$, $\delta_y = 1.322062007$
Rotation	X-Axis
Orientation	$\theta_x = 89.78095302^\circ$, $\theta_y = -0.219046959^\circ$
Eccentricity	0.847782278

2. We must highlight the cells with the data in the spreadsheet before going into the VBA Editor.

7.10 Crime Example

The example in this section comes from violent crime in the U.S. for 2007 and 2008. Tables 7.3 and 7.4 summarize the results for the weighted standard deviational ellipse for violent crime in the U.S. for the years 2007 and 2008. The data includes the continental U.S. Figure 7.2 shows a graph of both the Weibull ellipse and the exponential ellipse for the Violent Crime data in the U.S. for the year 2007.

7.11 GDP Example

Tables 7.5 and 7.6 summarize the Weibull ellipse for the GDP data for 2008 and 2009. We notice that the mean center is not *exactly the same* for the exponential ellipse and the standard deviational ellipse. This is because the values for the parameters γ_x and γ_y did not equal to exactly 1. Therefore, we can find some variation there.

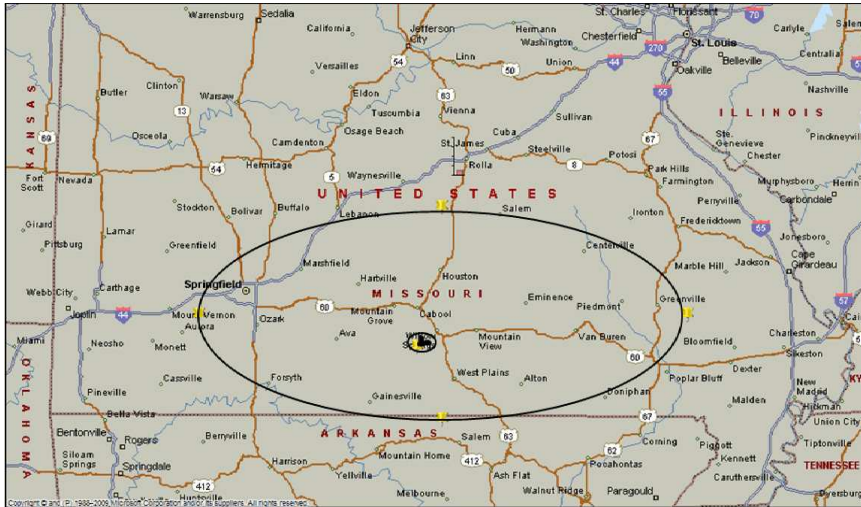


Figure 7.2 This figure shows both the Weibull deviational ellipse for the Crime data for the year 2007. The zoom level is 550 miles above the earth. The standard deviational ellipse was too large to fit on the map. It has the same center of gravity as the other ellipses. The Weibull ellipse is much smaller than the other two.

Table 7.5 Results for the Weibull Model on Gross Domestic Product
2008

Center	(44.95865049, 173.1927484)
Axes	$a = 1.876920973$, $b = 1.48828381$
Area	8.777 sq. mi.
Standard Deviations	$\delta_x = 1.5549$, $\delta_y = 1.5947$
Rotation	X-Axis
Orientation	$\theta_x = 89.28^\circ$, $\theta_y = -0.72^\circ$
Eccentricity	0.61

Table 7.6 Results for the Weibull Model on Gross Domestic Product
2009

Center	(45.10440668, 172.959405)
Axes	$a = 1.872381589$, $b = 1.493600132$
Area	8.786 sq. mi.
Standard Deviations	$\delta_x = 1.5588$, $\delta_y = 1.5997$
Rotation	X-Axis
Orientation	$\theta_x = 89.26^\circ$, $\theta_y = -0.74^\circ$
Eccentricity	0.60

7.12 Axes Length Comparison

To compare the lengths of the axes to those of the exponential ellipse, we use the relationships of the mean centers. When the parameters $\gamma_x = 1$ and $\gamma_y = 1$, then the mean center (\bar{x}, \bar{y}) are the same for both ellipses. The sums of squares are exactly the same also.

1. $0 < \gamma_x < 1$ (Powers): The exponential ellipse is smaller.

2. $\gamma_x > 1$ (Roots): The Weibull ellipse is smaller.

7.13 Sample Distribution Fitting

Three underlying distributions have been presented, the normal distribution (the weighted standard deviational ellipse), the exponential distribution (the weighted exponential deviational ellipse), and the Weibull distribution (the weighted Weibull deviational ellipse). We presented three data sets usually over multiple years, the USDA Kentucky corn example, the Department of Justice violent crime example, and OECD gross domestic product example. The examples expand from a local to a global perspective. We can use graphs to show the distribution of the data. What happens when all of the graphs show a straight line? What is the distribution of the sample?

We can continue to use the likelihood functions derived in previous sections. Since we already calculated the parameters for the samples, it is an easy task to calculate the likelihood function. Using the maximum likelihood theory, the distribution with the maximum value of the likelihood function L (and $\log L$) is the better fit.

Example	Model	$2k - 2\ln L(w, x, y)$	Comment
KY 2003 Corn Acres	SDE	-11,436.74	Min AIC
KY 2003 Corn Acres	Exp	222,911.00	
KY 2003 Corn Acres	Weibull	274,909.42	
KY 2004 Corn Acres	SDE	-2,110.16	Min AIC
KY 2004 Corn Acres	Exp	235,242.64	
KY 2004 Corn Acres	Weibull	291,173.28	
<hr/>			
US Crime 2007	SDE	5,697.60	Min AIC
US Crime 2007	Exp	64,790.55	
US Crime 2007	Weibull	78,359.93	
US Crime 2008	SDE	5,707.33	Min AIC
US Crime 2008	Exp	64,623.60	
US Crime 2008	Weibull	78,111.50	
<hr/>			
Global GDP 2008	SDE	4,976.84	Min AIC
Global GDP 2008	Exp	39,282.18	
Global GDP 2008	Weibull	47,388.74	
Global GDP 2009	SDE	4,906.12	Min AIC
Global GDP 2009	Exp	39,391.99	
Global GDP 2009	Weibull	47,560.40	

Given the following summary tables, for each of the data sets, the standard devotional distribution usually fits the sample data best.

When comparing the two probability models, the exponential model consistently outperforms the Weibull model. Adding the shape parameter using the Weibull distribution did not help in this particular analysis.

Example	Parameter Estimates (Rounded)
KY 2003 Corn Acres	$\hat{\alpha} = -1147830.91, \hat{\beta} = 1087263.36$
KY 2004 Corn Acres	$\hat{\alpha} = -1030547.49, \hat{\beta} = 1020882.76$
KY 2003 Corn Acres	$\hat{\lambda}_x = 2.40751, \hat{\lambda}_y = 0.32944, \hat{\lambda}_{xy} = 0.57958$
KY 2004 Corn Acres	$\hat{\lambda}_x = 2.45951, \hat{\lambda}_y = 0.33676, \hat{\lambda}_{xy} = 0.59240$
KY 2003 Corn Acres	$\hat{\lambda}_x = 2.40751, \hat{\lambda}_y = 0.32944, \hat{\lambda}_{xy} = 0.57958$
	$\hat{\gamma}_x = 0.76857, \hat{\gamma}_y = 0.76237, \hat{\gamma}_{xy} = 1.00000$
KY 2004 Corn Acres	$\hat{\lambda}_x = 2.45951, \hat{\lambda}_y = 0.33675, \hat{\lambda}_{xy} = 0.59240$
	$\hat{\gamma}_x = 0.76640, \hat{\gamma}_y = 0.76018, \hat{\gamma}_{xy} = 1.00000$
US Crime 2007	$\hat{\alpha} = 42344.54, \hat{\beta} = 35730.76$
US Crime 2008	$\hat{\alpha} = 41952.14, \hat{\beta} = 35152.29$
US Crime 2007	$\hat{\lambda}_x = 1.32875, \hat{\lambda}_y = 0.18291, \hat{\lambda}_{xy} = 0.32156$
US Crime 2008	$\hat{\lambda}_x = 1.38289, \hat{\lambda}_y = 0.18286, \hat{\lambda}_{xy} = 0.32146$
US Crime 2007	$\hat{\lambda}_x = 1.32876, \hat{\lambda}_y = 0.18291, \hat{\lambda}_{xy} = 0.32156$
	$\hat{\gamma}_x = 0.765003, \hat{\gamma}_y = 0.74781, \hat{\gamma}_{xy} = 1.00000$
US Crime 2008	$\hat{\lambda}_x = 1.32829, \hat{\lambda}_y = 0.18286, \hat{\lambda}_{xy} = 0.32146$
	$\hat{\gamma}_x = 0.765406, \hat{\gamma}_y = 0.75763, \hat{\gamma}_{xy} = 1.00000$
Global GDP 2008	$\hat{\alpha} = 44620.24, \hat{\beta} = 13643.50$
Global GDP 2009	$\hat{\alpha} = 44258.72, \hat{\beta} = 13570.98$
Global GDP 2008	$\hat{\lambda}_x = 0.84687, \hat{\lambda}_y = 0.25678, \hat{\lambda}_{xy} = 0.39406$
Global GDP 2009	$\hat{\lambda}_x = 0.84915, \hat{\lambda}_y = 0.25969, \hat{\lambda}_{xy} = 0.39774$
Global GDP 2008	$\hat{\lambda}_x = 0.84687, \hat{\lambda}_y = 0.25677, \hat{\lambda}_{xy} = 0.39405$
	$\hat{\gamma}_x = 0.77847, \hat{\gamma}_y = 0.76382, \hat{\gamma}_{xy} = 1.00000$
Global GDP 2009	$\hat{\lambda}_x = 0.84915, \hat{\lambda}_y = 0.25969, \hat{\lambda}_{xy} = 0.39773$
	$\hat{\gamma}_x = 0.77787, \hat{\gamma}_y = 0.76407, \hat{\gamma}_{xy} = 1.00000$

A word of caution about interpreting the parameters in the above table. For example, the parameters under the SDE models for the Violent Crime data survey and the Global GDP data survey have approximately the same north-south parameter estimate $\hat{\alpha}$, but a vastly different east-west parameter estimate $\hat{\beta}$. Does this mean that the United States capital (money) went east-ward, while the crime stayed in the U.S? No, because the two data surveys intentionally represent two different levels of geography to illustrate the calculations.

7.14 VBA

```

Sub SDE_Likelihood()
Dim sum_latitude, sum_longitude As Double 'Z^Tw
Dim alpha, beta As Double 'parameter estimates

'calculate the mean center for the latitude and the longitude

mean_latitude = 0
mean_longitude = 0
n = Selection.Rows.Count

    For i = 2 To n
        With ActiveSheet
            mean_latitude = mean_latitude + .Cells(i, 1).Value * .Cells(i, 3).Value
            mean_longitude = mean_longitude + .Cells(i, 2).Value * .Cells(i, 3).Value
        End With
    Next

    mean_latitude = mean_latitude / sum_weight
    mean_longitude = mean_longitude / sum_weight

'calculate Z^TZ and store it in the active spreadsheet

    For j = 1 To 2
        For i = 2 To n
            With ActiveSheet
                If j = 1 Then
                    .Cells(j, j + 20).Value = .Cells(j, j + 20).Value + _
                        (.Cells(i, j).Value - mean_latitude) * (.Cells(i, j).Value - mean_latitude)
                Else
                    .Cells(j, j + 20).Value = .Cells(j, j + 20).Value + _
                        (.Cells(i, j).Value - mean_longitude) * (.Cells(i, j).Value - mean_longitude)
                End If
            End With
        Next 'i
    Next 'j

```

```

[ For i = 2 To n
22   [ With ActiveSheet
37     .Cells(2, 21).Value = .Cells(2, 21).Value + (.Cells(i, 1).Value - mean_Latitude) * _
      (.Cells(i, 2).Value - mean_Longitude)
     End With
  Next ' off diagonals

  [ With ActiveSheet
    .Cells(1, 22).Value = .Cells(2, 21).Value
    'keep a copy for checks

    .Cells(9, 21).Value = .Cells(1, 21).Value
    .Cells(10, 21).Value = .Cells(2, 21).Value
    .Cells(9, 22).Value = .Cells(1, 22).Value
    .Cells(10, 22).Value = .Cells(2, 22).Value
23
    'initialize the identity matrix

    .Cells(1, 23).Value= 1
    .Cells(2, 24).Value= 1
    .Cells(1, 24).Value= 0
    .Cells(2, 23).Value= 0
  End With

  'perform gaussian elimination to find the inverse of Z^TZ

  [ m = 24
24   Do While m >= 21
38     [ With ActiveSheet
        .Cells(1, m).Value = .Cells(1, m).Value / .Cells(1, 21).Value
        .Cells(2, m).Value = .Cells(2, m).Value / .Cells(2, 21).Value
        m = m - 1
      End With
    Loop

```

```

'r1 - r2 ----> r2

    m = 24
    Do While m >= 21
25      With ActiveSheet
39        .Cells(2, m).Value = .Cells(1, m).Value - .Cells(2, m).Value
        m = m - 1
      End With
    Loop

'r2 / a_22 ----> r2

    m = 24
    Do While m >= 21
26      With ActiveSheet
40        .Cells(2, m).Value = .Cells(2, m).Value / .Cells(2, 22).Value
        m = m - 1
      End With
    Loop

7 'a_12 r2 - r1 ----> r1

    m = 24
    Do While m >= 21
27      With ActiveSheet
41        .Cells(1, m).Value = .Cells(1, 22).Value * .Cells(2, m).Value - .Cells(1, m).Value
        m = m - 1
      End With
    Loop

'-1*r1 ----> r1

    m = 24
    Do While m >= 21
28      With ActiveSheet
42        .Cells(1, m).Value = -1 * .Cells(1, m).Value
        m = m - 1
      End With
    Loop

```



```

'end of gaussian elimination. (Z^TZ)^-1 resides in (23,1), (24,1), (23,2), (24,2)
'(Z^TZ) resides in (21,9), (22,9), (21,10), (22,10)

'calculate Z^Tw
sum_latitude = 0
sum_longitude = 0
n = Selection.Rows.Count

    For i = 2 To n
        With ActiveSheet
            sum_latitude = sum_latitude + .Cells(i, 1).Value * .Cells(i, 3).Value 'w_i
            sum_longitude = sum_longitude + .Cells(i, 2).Value * .Cells(i, 3).Value 'w_i
        End With
    Next

'calculate alpha and beta

    With ActiveSheet
        alpha = .Cells(1, 23).Value * sum_latitude + .Cells(1, 24).Value * sum_longitude
        beta = .Cells(2, 23).Value * sum_latitude + .Cells(2, 24).Value * sum_longitude
    End With

'calculate log L
L_reg = 0
n = Selection.Rows.Count

    For i = 2 To n
        With ActiveSheet
            L_reg = L_reg + (.Cells(i, 3).Value - alpha * (.Cells(i, 1).Value - mean_latitude) -
            beta * (.Cells(i, 2).Value - mean_longitude))
        End With
    Next

    With Worksheets("STATS")
        .Cells(14, 1).Value = "Likelihood L(x,y)"
        .Cells(14, 2).Value = L_reg
    End With
End Sub

```

7.15 Exercises

1. Show that $\sum_{i=1}^n w_i^{\gamma_y} = 1$ where

$$w_i = \left(\frac{w'_i}{\sum_{i=1}^n w'_i} \right)^{1/\gamma_y} = 1$$

Solution: Let $\sum_{i=1}^n w'_i = w$. Then,

$$w_i = \left(\frac{w'_i}{\sum_{i=1}^n w'_i} \right)^{1/\gamma_y} = \left(\frac{w'_i}{w} \right)^{1/\gamma_y} \Rightarrow \sum_{i=1}^n w_i^{\gamma_y} = \sum_{i=1}^n \frac{w'_i}{w} = \frac{w}{w} = 1.$$

2. Show that

$$\frac{n}{\hat{\lambda}_x} = \bar{x} \text{ in Chapter 5.}$$

Solution:

$$\hat{\lambda} = \left(\frac{n}{w \sum_{i=1}^n x_i^{\gamma_x}} \right)^{1/\gamma_x}, \quad 0 \leq x_i \leq 180.$$

Let $w = \frac{\sum_{i=1}^n w_i}{\sum_{i=1}^n w_i}.$

$$\hat{\lambda} = \left(\frac{n}{\sum_{i=1}^n \left(\frac{w_i x_i^{\gamma_x}}{\sum_{i=1}^n w_i} \right)} \right)^{1/\gamma_x}$$

3. Show that

$$\frac{n}{\hat{\lambda}_y} = \bar{y} \text{ in Chapter 5.}$$

4. For what value of γ_x and γ_y in the Weibull ellipse makes it equal to the exponential ellipse [special case]? Show this algebraically.

CHAPTER 8

SPHERICAL STATISTICS

8.1 Introduction

So far, we have used the latitude and longitude coordinate system found in most geography literature. The statistical estimates gave meaningful results (e.g., the mean center landed in the survey area, the expected standard errors followed a known distribution, and two methods for calculating the area matched). In this Chapter, we introduce the concept of spherical statistics. For these analyses to work, we must convert the data from latitude and longitude coordinates to the Cartesian coordinate system (i.e. having an X , Y , and Z -axes each extending to infinity with a single pole at the point $(0, 0, 0)$).

This Chapter gives another view of the approach to spatial analysis. The spherical statistics may not match the other Chapter statistics, exactly. These are presently because they give additional information such as the "shape" of the data. The derivations and calculations of the spherical mean centers and standard deviations may not align well with previous Chapters. We follow mathematical and statistical theory carefully.

These calculations are unweighted, which means that we are analyzing the latitude and longitude observations in the absense of the random variable. Given this, multiple

year survey statistics give the same results since the latitudes and longitudes have been augmented to the data sets. New information not attainable before, will be presented.

This Section covers the multiple ellipses from a single population. The data shows more than one value with a high a frequency (probability) occurring in the survey. In directional statistics, we define the *mode* as the maximum a value of a given distribution and we define the *anti-mode* as the minimum of the distribution. For example, if we construct a histogram of a distribution for categorical ranges of the sample data and count the number of observations that fall into each category, then the category with the largest number of observations is the mode. The category with the smallest number of observations is the anti-mode.

The literature on multi-modal ellipses divides the techniques for measuring the ellipses into several broad categories:

- Multiple variable statistics.
- Multivariate statistics.
- Multi-modal statistics.

Multiple variable statistics includes such techniques as linear regression with one random variable and multiple explanatory variables [B. L. Bowerman and R. T. O'Connell, (2)]. We estimate the parameters using least squares. On the other hand, multivariate statistics linear regression with two or more random variables and can include one or more explanatory variables for each random variable [R. Khattree and D. N. Naik, (33)]. There may be correlation among the random variables. [R. G. Petersen (50); E. T. Lee, (35)] discuss more general analysis of variance models (ANOVA models) and specifically the distributions discussed in preceding Chapters for the univariate cases. [R. Khattree and D. N. Naik, (33)] give analogous multivariate analysis of variance models (MANOVA models).

Multimodal statistics is a property of a particular distribution albeit a univariate distribution or a multivariate distribution. [L. Cobb, P. Koppstein, and N. H. Chen (7)] contains a discussion on multimodal distributions for the exponential distribution, the normal distribution, and the beta distribution. The paper also contains a test for multimodality [(7, page 128]. Although the paper seems relevant, after some correspondence with L. Cobb, the paper does not discuss random variables or random distributions. [Mardia, (45)] contains extensive discussions on univariate and multivariate sampled distributions. Of interest in this text is [Mardia, (45), page 221], which shows how to calculate the mean center of gravity, using sines and cosines. The author has a simple test for multimodality [Mardia (45), page 209]. The author gives other, advanced tests also.

[D. J. Steffensmeier, E. A. Allan, M. D. Harer, and C. Streifel, (59)] provide an in-depth example of crime data linking crime type and age. This article discusses two distinct issues.

- Committing crime as a person ages (supported by Sociological studies).

- Shift in the age distribution by the type of crime (supported by the data).

The authors rely on percentages and quantiles throughout the paper for statistical comparisons. To show multimodality, the authors show the shift in the age distribution using quantiles of committing a specific type of crime over a period from 1940 to 1980. They also provide the usual skewness and kurtosis calculations for the shape of the distributions as well as chi-square tests.

8.2 Concepts

Directional statistics models probability densities and angles. In prior chapters, our data came in the triplet (x_i, y_i, w_i) for the latitude, the longitude, and the random variable and mapped onto the Earth's surface. In this chapter, the data comes from the sample of size n of points on a sphere $P(\phi_i, \theta_i)$. The latitude ϕ_i is the vertical line running north and south parallel of the Prime Meridian and has the range $[-90^\circ, 90^\circ]$. The longitude θ_i is the horizontal line running east and west parallel of the Equator and has the range $[-180^\circ, 180^\circ]$. We define the Equator as having latitude of 0° . We define the Prime Meridian as having longitude of 0° . We will need to convert decimal degrees into radians often in this Chapter because the calculations rely heavily on trigonometry functions. [Shelby (57), pages 389-392] gives an excellent discussion of projecting coordinates from the Earth onto a circle.

How do we convert our latitude and longitude data to directional data? The Equator divides the Earth in half into the Northern and Southern Hemispheres each having the same mass. The latitude of the Equator is $\phi^\circ = 0^\circ$.



Figure 8.1 This figure shows a map of the U.S. relative to the Equator (represented by the black line).

To obtain a scatter diagram in two dimensions on the surface of a sphere, use a projection on the sphere. Consider the polar coordinates (ρ, ψ) where ρ is the polar

distance and ψ is the polar angle. [Mardia, (45)] defines the *stereographic equal-angle projection* as $\rho = \tan\left(\frac{\theta}{2}\right)$, and $\psi = \phi$ where $0 < \rho < 1$ and $0 < \psi < 2\pi$. [Mardia, (45)] calls the plot a *Wulff-net* or *stereographic-net*. It is an "area preserving" plot compared to other plots.

Steps to mimic the previous chapters:

1. Convert the spatial data to circular data.
2. Calculate the quantities $(\bar{l}_0, \bar{m}_0, \bar{n}_0)$. These calculations result in radians when using Excel 2010 spreadsheets.
3. Convert the results back to degrees from radians.
4. Convert the mean latitude and mean longitude back to spatial coordinates.

8.3 X-Axis Mean Center and Resultant Length

To find the mean direction North-South using directional statistics, we first calculate the cosines in Equation (8.1).

$$n_0 = \sum_{i=1}^n n_i = \sum_{i=1}^n \cos \phi_i. \quad (8.1)$$

where $P(\phi_i, \theta_i)$ is a point on the Earth; ϕ_i equals to the latitude and θ_i equals to the longitude for observation i . To find the average of n_0 , we divide by the quantity R to obtain

$$\bar{X} = \bar{n}_o = \frac{1}{n} n_o \quad (8.2)$$

where we define R next.

$$\bar{R} = \sqrt{\left(\frac{1}{n} \sum_{i=1}^n l_i\right)^2 + \left(\frac{1}{n} \sum_{i=1}^n m_i\right)^2 + \left(\frac{1}{n} \sum_{i=1}^n n_i\right)^2}, \quad 0 < \bar{R} \leq 1 \quad (8.3)$$

where n is the sample size and l_i , m_i , and n_i are defined in this Section and the next. For now it suffices to say that R is between 0 and 1.

The diagram in Figure 8.2 shows the triangle with the longitude θ . It has a right angle (90°), the angle θ identified, the hypotenuse R and the two sides n_0 and m_0 . We will define m_0 in Section 8.4.

Equation (8.4) gives the weighted cosines and average latitude using the random variable W .

$$\bar{X}_w = \bar{n}_{w_0} = \sum_{i=1}^n \frac{w_i n_i}{\sum_{i=1}^n w_i} = \sum_{i=1}^n \frac{w_i \cos \phi_i}{\sum_{i=1}^n w_i}. \quad (8.4)$$

Accordingly, Equation (8.5) gives the weighted resultant length \bar{R} .

$$\bar{R}_w = \sqrt{\left(\sum_{i=1}^n \frac{w_i l_i}{\sum_{i=1}^n w_i}\right)^2 + \left(\sum_{i=1}^n \frac{w_i m_i}{\sum_{i=1}^n w_i}\right)^2 + \left(\sum_{i=1}^n \frac{w_i n_i}{\sum_{i=1}^n w_i}\right)^2}, \quad 0 < \bar{R}_w \leq 1 \quad (8.5)$$

8.4 Y and Z-Axis Mean Centers

To find the means in the Y and Z -axis directions using directional statistics, we need both the latitude (ϕ_i) and the longitude (θ_i) observations for the points $P(\phi_i, \theta_i)$ on the Earth. We calculate the sums for the Y -axis in Equation (8.6) and the sums for the Z -axis in Equation (8.7).

$$l_0 = \sum_{i=1}^n l_i = \sum_{i=1}^n \sin \phi_i \cos \theta_i, \quad (8.6)$$

$$m_0 = \sum_{i=1}^n m_i = \sum_{i=1}^n \sin \phi_i \sin \theta_i. \quad (8.7)$$

$P(\phi_i, \theta_i)$ is a point on the Earth. ϕ_i is called the latitude and θ_i is called the longitude for observation i . To find the average of l_0 and m_0 , we need the mean resultant length R , which was defined in Section 8.3. For now, it suffices to say that, Equations (8.8) and (8.9) gives the means for l_0 and m_0 .

$$\bar{Y} = \bar{l}_0 = \frac{1}{n} l_0. \quad (8.8)$$

$$\bar{Z} = \bar{m}_0 = \frac{1}{n} m_0. \quad (8.9)$$

Equations (8.10) and (8.11) give the weighted averages for the Y and Z axes.

$$\bar{Y}_w = \bar{l}_{w_0} = \sum_{i=1}^n \frac{w_i \sin \phi_i \cos \theta_i}{\sum_{i=1}^n w_i}. \quad (8.10)$$

$$\bar{Z}_w = \bar{m}_{w_0} = \sum_{i=1}^n \frac{w_i \sin \phi_i \sin \theta_i}{\sum_{i=1}^n w_i}. \quad (8.11)$$

8.4.1 Spherical Variance

[Mardia and Jupp, (46)] give two definitions for the sample spherical variance. The first sample spherical variance is defined in Equation (8.12). This definition minimizes a matrix of the arithmetic mean of the Euclidean distances.

$$S^* = 2(1 - \bar{R}), \quad 0 < S^* < 1 \quad (8.12)$$

where R is the resultant length. For small S^* , the data is uniform. There is no preferred direction. For large S^* , the data is directional and may be clustered or may not be clustered. When $\bar{R} \approx 1$, the data are heavily concentrated. \bar{R} measures clustering around the mean direction.

A second measure of the sample spherical variance is given in Equation (8.13). This definition is an analogue from a derivation of the matrix T . The matrix T will be used later.

$$S^* = 1 - \bar{R}^2, 0 < S^* < 1. \quad (8.13)$$

8.5 VBA

Figure 8.3 shows the VBA programming environment for `Module 4` and the driver subroutine `modal_stats()`.

`Module 4` contains the following subroutines:

- `SinCosS()` — This subroutine calculates the sines and cosines of the latitude and longitude. It also calculates the observations l_i , m_i , and n_i . The subroutine stores the values in the active worksheet in columns `F` thru `K`. The number of rows depends on the length of the input data set.

The `WITH-END-WITH` statement (4) writes headings across the columns of the active spreadsheet.

The second `FOR-NEXT` loop (6) converts the latitude and longitude observations from degrees to radians. The `FOR-NEXT` loop calculates the sines and cosines of the latitude and longitude observations and stores those values in columns `F`, `G`, and `H`. The `FOR-NEXT` loop calculates the observation values l_i , m_i , and n_i , $i = 1, 2, \dots, n$ and stores them in columns `I`, `J`, and `K`.

- `Calculate_B()` — This subroutine calculates the matrix B . The upper left corner of the matrix is stored in cell `O13`. The lower right corner of the matrix is stored in cell `Q15`.
- `modal_stats()` — This subroutine calls the `SinCosS()` subroutine. It calculates the matrix T and the spherical statistics.

The `FOR-NEXT` loop (9) calculates the sums l_0 , m_0 , and n_0 using Equations (8.6), (8.7), and (8.1).

The VBA statement immediately following the `FOR-NEXT` loop calculates the resultant length R in Equation (8.3) and saves the value in the variable `R_length`. Following this VBA statement, we calculate the spherical variance S^* using Equation (8.12) and we save the value in the variable `S`.

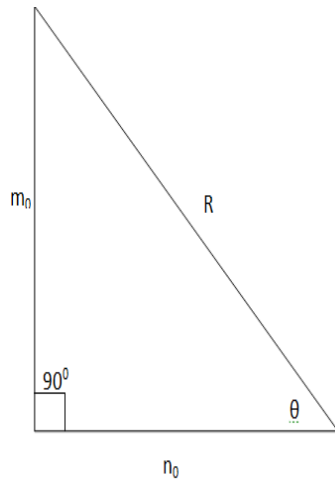


Figure 8.2 This figure shows the diagram of a triangle with the quantities for the longitude θ , n_0 , and the resultant length R .

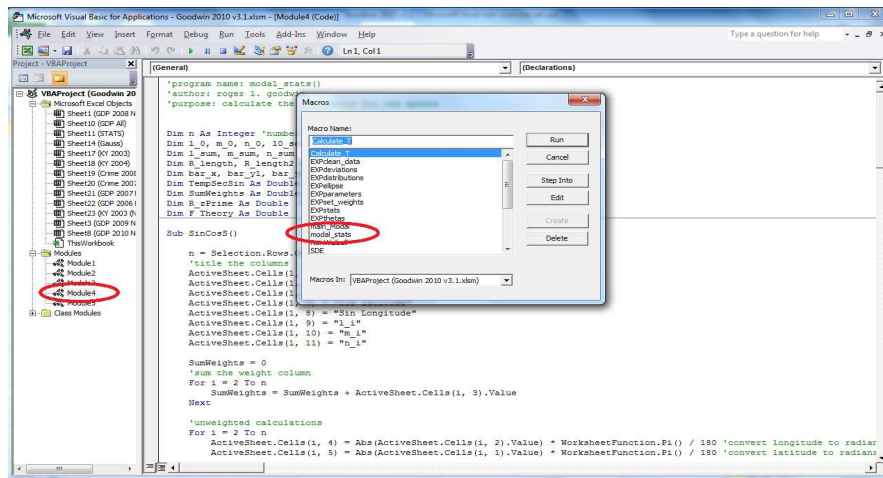


Figure 8.3 This figure shows the VBA environment for Module 4. It also highlights the driver subroutine `modal_stats()`.

	N	O	P	Q	R	S	T	U	V	W
Modal Ellipse Statistics										
Mean X		45.35963106								
Mean Y		2.701085798								
Mean Z		-34.87836501								
l_0 (degrees)		4082.366795								
m_0 (degrees)		243.0977218								
n_0 (degrees)		-3139.052851								
Spherical Variance		0.000461194								
Resultant Length (R)		0.999769376								
							lambda1	lambda2	lambda3	
							3.728833	6.216179	29.0549876	39
Matrix B		33.58777584	-3.351383372	43.36764038						
		-3.351383372	89.77014071	2.59054283						
		43.36764038	2.59054283	56.64208343						
			180 -trace(B)							
			90 -trace(T)							
Matrix T										
		56.41222416	3.351383372	-43.36764038	v1				HINT: Choose v3 = 1	
		3.351383372	0.229859286	-2.59054283	v2	=	0.00663	v1	0.73478842	
		-43.36764038	-2.59054283	33.35791655	v3			v3	0.5733343	
									1	
							Check			
Eigenvalue		0.006629744					0.004871	0.004871		0.012388514
Newton Iterations		6					0.003801	0.003801		0.007276926
							0.00663	0.00663		

Figure 8.4 This figure shows the STATS spreadsheet with the output from the subroutines `main_Modal()` and `modal_stats()`.

Following the calculation of the spherical variance, we calculate the means \bar{X} , \bar{Y} , and \bar{Z} using Equations (8.2), (8.8), and (8.9). We save the results to the variables `l_sum`, `m_sum`, and `n_sum`. Finally, we apply the arcsin or arccos function to the sums and we convert the results back to degrees from radians.

The WITH-END-WITH statement (10) writes the statistical results to column O and the descriptive labels to column N. The upper-left corner of matrix T is cell O19. The lower-right corner of matrix T is cell Q17.

The spherical statistics calculated by the subroutine `modal_stats()` include the following:

- Mean in the X -axis direction \bar{X} .
- Mean in the Y -axis direction \bar{Y} .
- Mean in the Z -axis direction \bar{Z} .
- l_0 — one of three measures for the mean direction of cosines.
- m_0 — one of three measures for the mean direction of cosines.
- n_0 — one of three measures for the mean direction of cosines.
- Spherical variance S^* .
- Resultant length R .

Figure 8.4 shows the output statistics from the subroutine `modal_stats()`.

```
1 [ 'program name: modal_stats()  
  'author: roger l. goodwin  
  'purpose: calculate the statistics for the sphere  
  Dim n As Integer 'number of observations  
  Dim l_0, m_0, n_0 As Double  
  Dim l_sum, m_sum, n_sum As Double  
  Dim R_length As Double  
  Dim bar_x, bar_y1, bar_y2 As Double  
  Dim TempSecSin As Double  
  Dim X, Y, Z As Double
```

```

Sub SinCosS()
n = Selection.Rows.Count
'title the columns
    With ActiveSheet
        .Cells(1, 4).Value = "Latitude (Radians)"
        .Cells(1, 5).Value = "Longitude (Radians)"
        .Cells(1, 6).Value = "Sin Longitude"
        .Cells(1, 7).Value = "Cos Longitude"
        .Cells(1, 8).Value = "Sin Latitude"
        .Cells(1, 9).Value = "X_i"
        .Cells(1, 10).Value = "Y_i"
        .Cells(1, 11).Value = "Z_i"
    End With
X = 0
Y = 0
Z = 0
'unweighted calculations
    For i = 2 To n
        With ActiveSheet
            .Cells(i, 4).Value = .Cells(i, 1).Value * WorksheetFunction.Pi() / 180 'convert latitude to radians
            .Cells(i, 5).Value = .Cells(i, 2).Value * WorksheetFunction.Pi() / 180 'convert longitude to radians
            .Cells(i, 6).Value = Sin(.Cells(i, 5).Value) 'sin of longitude in radians
            .Cells(i, 7).Value = Cos(.Cells(i, 5).Value) 'cos of longitude in radians
            .Cells(i, 8).Value = Sin(.Cells(i, 4).Value) 'sin of latitude in radians
            .Cells(i, 9).Value = Cos(.Cells(i, 4).Value) * X_i
            .Cells(i, 10).Value = .Cells(i, 8) * .Cells(i, 7) * Y_i
            .Cells(i, 11).Value = .Cells(i, 8) * .Cells(i, 6) * Z_i
        End With
    Next
End Sub

```

```

Sub modal_stats()
Call SinCosS
n = Selection.Rows.Count
l_0 = 0
m_0 = 0
n_0 = 0
R_length = 0
    For i = 2 To n
        With ActiveSheet
            l_0 = l_0 + .Cells(i, 9).Value
13 31 m_0 = m_0 + .Cells(i, 10).Value
            n_0 = n_0 + .Cells(i, 11).Value
        End With
    Next
l_sum = l_0 / (n - 1)
m_sum = m_0 / (n - 1)
n_sum = n_0 / (n - 1)
R_length = Sqr(l_sum ^ 2 + m_sum ^ 2 + n_sum ^ 2)
S = (1 - R_length ^ 2) 'spherical variance
'put the stats in an excel worksheet
3 'put the stats in an excel worksheet
    With Worksheets("STATS")
        .Cells(1, 14).Value = "Modal Ellipse Statistics"
        .Cells(2, 14).Value = "Mean Latitude"
        .Cells(3, 14).Value = "Mean Longitude 1 (+/-)"
        .Cells(4, 14).Value = "Mean Longitude 2 (+/-)"
        .Cells(5, 14).Value = "l_0 (degrees)"
        .Cells(6, 14).Value = "m_0 (degrees)"
        .Cells(7, 14).Value = "n_0 (degrees)"
        .Cells(8, 14).Value = "Spherical Variance"
        .Cells(9, 14).Value = "Resultant Length (R)"
14 .Cells(18, 14).Value = "Matrix T"
        .Cells(2, 15).Value = l_sum * 180 / WorksheetFunction.Pi()
        .Cells(3, 15).Value = m_sum * 180 / WorksheetFunction.Pi()
        .Cells(4, 15).Value = n_sum * 180 / WorksheetFunction.Pi()
        .Cells(5, 15).Value = l_0 * 180 / WorksheetFunction.Pi()
        .Cells(6, 15).Value = m_0 * 180 / WorksheetFunction.Pi()
        .Cells(7, 15).Value = n_0 * 180 / WorksheetFunction.Pi()
        .Cells(8, 15).Value = S
        .Cells(9, 15).Value = R_length
    End With
End Sub

```

8.6 Test and Characterizations

The spherical distributions can have one of three visual characteristics:

- Uniform: The points have a uniformly dispersed pattern and have no "preferred" direction.
- Bimodal: The points have a clustered pattern about the ends of an axis (e.g. the north and south poles). The axis joins the two modes as well as the two anti-modes.
- Girdle: The points have a dispersed, elliptic pattern. The mode(s) are on a plane.

To detect which distribution we have, define the matrix B as:

$$B = nI - T = \begin{bmatrix} n - \sum_{i=1}^n l_i^2 & -\sum_{i=1}^n l_i m_i & -\sum_{i=1}^n l_i n_i \\ -\sum_{i=1}^n l_i m_i & n - \sum_{i=1}^n m_i^2 & -\sum_{i=1}^n m_i n_i \\ -\sum_{i=1}^n l_i n_i & -\sum_{i=1}^n m_i n_i & n - \sum_{i=1}^n n_i^2 \end{bmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \quad (8.14)$$

We find the eigenvalues of matrix B to determine the visual shape of the data. We define the vector \underline{b}_1 as the first row times the vector of unknowns:

$$\underline{b}_1 = \begin{bmatrix} n - \sum_{i=1}^n l_i^2 & -\sum_{i=1}^n l_i m_i & -\sum_{i=1}^n l_i n_i \\ -\sum_{i=1}^n l_i m_i & n - \sum_{i=1}^n m_i^2 & -\sum_{i=1}^n m_i n_i \\ -\sum_{i=1}^n l_i n_i & -\sum_{i=1}^n m_i n_i & n - \sum_{i=1}^n n_i^2 \end{bmatrix} \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} = \begin{pmatrix} b_1(n - \sum_{i=1}^n l_i^2) \\ -b_2 \sum_{i=1}^n l_i m_i \\ -b_3 \sum_{i=1}^n l_i n_i \end{pmatrix}$$

Vector \underline{b}_2 corresponds to row 2 and vector \underline{b}_3 corresponds to row 3.

8.7 VBA

The `Calculate_B()` subroutine (2) calculates the matrix B in Equation (8.14). The `FOR-NEXT` loop (7) uses previously calculated results stored on the active spreadsheet in columns I, J, and K to calculate the sums in Equation (8.14). The `WITH-END-WITH` statement (8) saves the matrix B to the `STATS` spreadsheet. The upper-left corner of the matrix is cell O13. The lower-right corner of the matrix is cell Q15.

```

Sub Calculate_B()
n = Selection.Rows.Count
l_0 = 0
m_0 = 0
n_0 = 0
l0_sqrd = 0
m0_sqrd = 0
n0_sqrd = 0
lm_sum = 0
ln_sum = 0
mn_sum = 0
R_length = 0
  For i = 2 To n
    With ActiveSheet
      l_0 = l_0 + .Cells(i, 9).Value
      m_0 = m_0 + .Cells(i, 10).Value
      n_0 = n_0 + .Cells(i, 11).Value
      'calculate the matrix B
      l0_sqrd = l0_sqrd + (.Cells(i, 9).Value) ^ 2
      m0_sqrd = m0_sqrd + (.Cells(i, 10).Value) ^ 2
      n0_sqrd = n0_sqrd + (.Cells(i, 11).Value) ^ 2
      lm_sum = lm_sum + .Cells(i, 9).Value * .Cells(i, 10).Value
      ln_sum = ln_sum + .Cells(i, 9).Value * .Cells(i, 11).Value
      mn_sum = mn_sum + .Cells(i, 10).Value * .Cells(i, 11).Value
    End With
  Next
  With Worksheets("STATS")
    .Cells(13, 14).Value = "Matrix B"
    .Cells(13, 15).Value = (n - 1) - l0_sqrd
    .Cells(13, 16).Value = -1 * lm_sum
    .Cells(13, 17).Value = -1 * ln_sum
    .Cells(14, 16).Value = (n - 1) - m0_sqrd
    .Cells(14, 15).Value = -1 * lm_sum
    .Cells(14, 17).Value = -1 * mn_sum
    .Cells(15, 17).Value = (n - 1) - n0_sqrd
    .Cells(15, 15).Value = -1 * ln_sum
    .Cells(15, 16).Value = -1 * mn_sum
  End With
End Sub

```


Finding the eigenvalues of B involves two steps:

1. Find the roots λ to the determinant $\text{Det}(B - \lambda I)$ where the matrix I is the 3×3 identity matrix. There may be at most three unique roots to the determinant.
2. For each eigenvalue λ_j , solve the system $(B - \lambda_j I)\underline{v} = \underline{0}$.

Equation (8.15) gives the determinant for $B - \lambda_j I$.

$$\begin{aligned} \text{Det}(B - \lambda_j I) = & \quad (8.15) \\ & \left(n - \sum_{i=1}^n l_i^2 - \lambda_j \right) \left(n - \sum_{i=1}^n m_i^2 - \lambda_j \right) \left(n - \sum_{i=1}^n n_i^2 - \lambda_j \right) - \\ & \sum_{i=1}^n l_i m_i \sum_{i=1}^n m_i n_i \sum_{i=1}^n l_i n_i - \sum_{i=1}^n l_i n_i \sum_{i=1}^n l_i m_i \sum_{i=1}^n m_i n_i + \\ & \sum_{i=1}^n l_i n_i \sum_{i=1}^n l_i n_i \left(n - \sum_{i=1}^n m_i^2 - \lambda_j \right) + \\ & \sum_{i=1}^n m_i n_i \sum_{i=1}^n m_i n_i \left(n - \sum_{i=1}^n l_i^2 - \lambda_j \right) + \\ & \sum_{i=1}^n l_i m_i \sum_{i=1}^n l_i m_i \left(n - \sum_{i=1}^n n_i^2 - \lambda_j \right). \end{aligned}$$

Arrange the eigenvalues λ_1 , λ_2 , and λ_3 in ascending order such that $0 \leq \lambda_1 \leq \lambda_2 \leq \lambda_3 \leq n$ where n is the sample size and $\sum_{i=1}^3 \lambda_i = n$. The trace of matrix B sums to $2n$. As in [Mardia (45), page 223-224], we interpret the diagonal elements of B and the eigenvectors.

β_i , $i = 1, 2, 3$ equal to the diagonal elements of B where $\beta_1 = b_{11}$, $\beta_2 = b_{22}$, and $\beta_3 = b_{33}$. Working with the matrix T , we define the following eigenvalues and eigenvectors.

$$\lambda_i = n - \beta_i, \quad \underline{t}_i = \underline{b}_i, \quad i = 1, 2, 3, \quad \lambda_1 + \lambda_2 + \lambda_3 = n.$$

The terms large and small are subjective. The reader may ask whether there is a statistical test to differentiate between the values for λ_i ? [Mardia (45), pages 276-283] shows such a test statistic. Let the null hypothesis be $H_0 : \lambda_1 = \lambda_2 = \lambda_3 = \frac{n}{3}$, which states that the eigenvalues have equal values. The test statistic is S_U

$$S_U = \frac{15}{2n} \sum_{i=1}^3 \left(\lambda_i - \frac{1}{3}n \right)^2 \sim \chi^2(5). \quad (8.16)$$

Equation (8.16) is asymptotically a chi-square distribution with five degrees of freedom. For large n , the chi-square test works best. For sufficiently large sample sizes, this should not be an issue. S_U tests for uniform λ'_i s. Other statistical tests test for girdle and rotational symmetry using the λ'_i s.

Table 8.1 Interpretation of the Eigenvalues

Eigenvalues	Condition	Distribution	Build
$\lambda_1 \approx \lambda_2 \approx \lambda_3$		Uniform	Axes having no orientation
Small λ_1, λ_2 . Large λ_3	$\lambda_1 \neq \lambda_2$	Unimodal for large R . Bimodal otherwise.	If unimodal, the concentration at end of the eigenvector \underline{t}_3 . Otherwise, concentration at both ends of the eigenvector \underline{t}_3 .
Small λ_1, λ_2 . Large λ_3	$\lambda_1 \approx \lambda_2$	Unimodal for large R . Bipolar otherwise.	Rotational symmetry about \underline{t}_3 .
Small λ_1 . Large λ_2 , and λ_3 .	$\lambda_2 \neq \lambda_3$	Girdle	Girdle plane spanned by the the eigenvectors \underline{t}_2 and \underline{t}_3 .
Small λ_1 . Large λ_2 , and λ_3 .	$\lambda_2 \approx \lambda_3$	Symmetric girdle	Rotational symmetry about the eigenvector \underline{t}_1 .

8.7.1 Finding the Eigenvalues

This is a simple Netwon algorithm for finding the eigenvalues λ_j , $j = 1, 2, 3$. The Gaussian elimination algorithm must be run to find the eigenvectors for each λ_j . This subroutine must be run mutiple times. This routine needs to find the three values in the *determinant of the matrix B* :

$$\begin{bmatrix} a - \lambda & d & e \\ f & b - \lambda & g \\ h & i & c - \lambda \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

where the constants a, b, c, d, e, f, g, h , and i are given. The cubic equation must be solved for λ :

$$(a - \lambda)(b - \lambda)(c - \lambda) + dgh + efi - he(b - \lambda) - ig(a - \lambda) - fd(c - \lambda). \quad (8.17)$$

The derivative of Equation (8.17) with respect to λ is given in Equation (8.18).

$$-(b - \lambda)(c - \lambda) - (a - \lambda)(c - \lambda) - (a - \lambda)(b - \lambda) + eh + ig + fd. \quad (8.18)$$

[Cheney and Kincaid, (5), pages 83-90] give an introduction and an algorithm for Newton's method. We wish to apply the method to find the eigenvalue λ_j , $j = 1, 2$ or 3. The recursive definition of Netwon's method is

$$\lambda_{k+1} = \lambda_k - \frac{f(\lambda_k)}{f'(\lambda_k)}$$

where $k = 0, 1, 2, \dots$ represents the iteration (also called the index in the sequence). We use Equation (8.17) to calculate the function $f(\lambda_k)$. We use Equation (8.18) to calculate the function $f'(\lambda_k)$.

8.8 VBA

The function `fx()` (5) calculates the determinant of the matrix T using Equation (8.14) and (8.17). $T = nI - B$ where I is the 3×3 identity matrix. The function `gx()` (6) calculates the derivative of the determinant of the matrix T using Equations (8.14) and (8.18). The subroutine `Newton()` (7) applies the Newton algorithm to find the eigenvalue λ_j , $j = 1, 2$ or 3 . We will discuss the VBA code for calculating the matrix T in Section 8.8.1.

```

8  Function fx(Lambda)
   'the determinate of the matrix T
   Dim a, b, c, d, e, g As Double
   'matrix T is symmetrix. not all of it needs to be read in.
   With Worksheets("STATS")
      a = .Cells(19, 15).Value 'sum l_i^2
      d = .Cells(19, 16).Value 'sum l_i m_i
      e = .Cells(19, 17).Value 'sum l_i n_i
      b = .Cells(20, 16).Value 'sum m_i^2
26  g = .Cells(20, 17).Value 'sum m_i n_i
      c = .Cells(21, 17).Value 'sum n_i^2
      f = d
      i = g
      h = e
   End With
   fx = (a - Lambda) * (b - Lambda) * (c - Lambda) + d * g * h + e * f * i - _
   h * e * (b - Lambda) - i * g * (a - Lambda) - f * d * (c - Lambda)
End Function

```

```

Function gx(Lambda)
'the derivative of the determinate of matrix T
Dim a, b, c, d, e, g As Double
    With Worksheets("STATS")
        a = .Cells(19, 15).Value 'sum l_i^2
        d = .Cells(19, 16).Value 'sum l_i m_i
        e = .Cells(19, 17).Value 'sum l_i n_i
        b = .Cells(20, 16).Value 'sum m_i^2
9 27 g = .Cells(20, 17).Value 'sum m_i n_i
        c = .Cells(21, 17).Value 'sum n_i^2
        f = d
        i = g
        h = e
    End With
    gx = -1 * (b - Lambda) * (c - Lambda) - (a - Lambda) * (c - Lambda) -
    (a - Lambda) * (b - Lambda) + e * h + i * g + f * d
End Function

```

```

Sub Newton(Lambda)
Dim lambda0 As Double 'old value of lambda
Dim diff As Double 'termination condition
Dim i As Integer
diff = 1
i = 1
f_x = fx(Lambda)
10 While (diff > 0.0000000001)
    lambda0 = Lambda
    Lambda = Lambda - f_x / gx(Lambda)
28 f_x = fx(Lambda)
    diff = Abs(lambda0 - Lambda)
    i = i + 1
Wend
29 With Worksheets("STATS")
    .Cells(23, 14).Value = "Eigenvalue"
    .Cells(24, 14).Value = "Newton Iterations"
    .Cells(23, 15).Value = Lambda
    .Cells(24, 15).Value = i
    .Cells(20, 20).Value = Lambda
End With
End Sub

```

8.8.1 Finding the Eigenvectors

This is a Gaussian elimination algorithm that performs row reduction on a 3×3 matrix. This subroutine only needs to be run once for each λ_j , $j = 1, 2, 3$. There is no trial and error as there was in finding the eigenvalues. [Grossman, (23), pages 277-281] gives the theory behind eigenvalues and eigenvectors and the strategy for finding each. Keep in mind that we are only solving a 3×3 matrix in Spherical Statistics and the code given reflects this simplicity. Setting the initial value for v_3 to back solve the system of equations is the reader's prerogative as stated in [Grossman, (23)]. There are an infinite number of solutions to the equations.

8.9 VBA

The statistics associated with the spherical distribution are complicated. They usually begin with finding the some statistic of a 3×3 matrix from either B or T . This Section describes the VBA code in Module 4 and Module 5. It is best

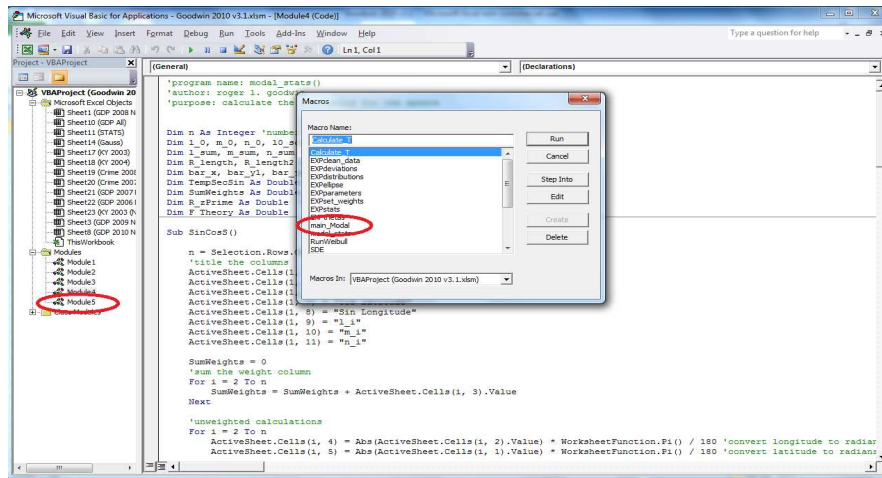


Figure 8.5 This figure shows the VBA environment for Module 5. It also highlights the driver subroutine `main_modal()`.

to start with Module 5. Figure 8.5 shows a screen capture of the VBA environment for Module 5. The subroutines in Module 5 calculate the matrix T and find the eigenvalues to matrix T . Since there are an infinite number of eigenvectors, some interaction and verification on the reader's part is required when finding the eigenvectors. Module 5 contains the following six subroutines:

1. `main_modal()` — This is a driver subroutine.
2. `Calculate_B()` — This subroutine calculates the matrix B from which the diagonal elements $\hat{\beta}_1$, $\hat{\beta}_2$ and $\hat{\beta}_3$ are determined. We can re-use the subroutine in Module 5 for calculating this matrix.
3. `Calculate_T()` — This subroutine calculates the matrix T . The upper left-hand corner of the matrix resides in cell O19 in the `STATS` spreadsheet and cell Q21 in the lower-right corner.

The FOR-NEXT loop (10) calculates the matrix T using Equation (8.14) where $T = nI - B$. The WITH-END-WITH statement (11) saves the matrix T below matrix B in the `STATS` spreadsheet. The upper-left corner is cell O19 and the lower-right corner is cell Q21. The remaining code writes descriptive labels to the `STATS` spreadsheet for the eigenvectors in columns R and V.

4. `Gauss(Lambda)` — This subroutine performs row elimination on matrix T given the eigenvalue `Lambda`. This finds a possible eigenvector and puts it into cells V19, V20, and V21. The value in cell V21 will always be equal to 1. This allows the reader to scale the other values in the vector up or down (usually be a factor of 10).

The `FOR-NEXT` loop (12) reads the matrix T into the array `T_prime`. Since T will always be a 3×3 matrix, the code in the `Gauss()` subroutine performs row reduction as one would do on paper.

5. `fx(Lambda)` — This subroutine finds the determinant of matrix T given the eigenvalue `Lambda`.
6. `gx(Lambda)` — This subroutine finds the derivative of the determinant of matrix T given the eigenvalue `Lambda`.
7. `Newton(Lambda)` — This subroutine finds the eigenvalues of matrix T using the subroutines `fx()` and `gx()`. It puts the eigenvalue in cell `T20`.

When trying to find the eigenvalues and the eigenvectors, the reader only needs to run the `main_Modal()` subroutine and ensure that it contains the proper calls and parameters to the `Newton()` and `Gauss()` subroutines. Ordering is important. The subroutines in Module 5, particularly the `Newton()` subroutine, must be run before the `Gauss()` can be run. The statements in the `main_Modal()` subroutine clearly show the ordering of the other subroutine calls.

```

Sub Calculate_T()
n = Selection.Rows.Count
l_0 = 0
m_0 = 0
n_0 = 0
l0_sqrd = 0
m0_sqrd = 0
n0_sqrd = 0
lm_sum = 0
ln_sum = 0
mn_sum = 0
R_length = 0
  For i = 2 To n
    With ActiveSheet
      l_0 = l_0 + .Cells(i, 9).Value
      m_0 = m_0 + .Cells(i, 10).Value
      n_0 = n_0 + .Cells(i, 11).Value
      'calculate the matrix T
      l0_sqrd = l0_sqrd + (.Cells(i, 9).Value) ^ 2
      m0_sqrd = m0_sqrd + (.Cells(i, 10).Value) ^ 2
      n0_sqrd = n0_sqrd + (.Cells(i, 11).Value) ^ 2
      lm_sum = lm_sum + .Cells(i, 9).Value * .Cells(i, 10).Value
      ln_sum = ln_sum + .Cells(i, 9).Value * .Cells(i, 11).Value
      mn_sum = mn_sum + .Cells(i, 10).Value * .Cells(i, 11).Value
    End With
  Next
  With Worksheets("STATS")
    .Cells(19, 15).Value = l0_sqrd
    .Cells(19, 16).Value = lm_sum
    .Cells(19, 17).Value = ln_sum
    .Cells(20, 16).Value = m0_sqrd
    .Cells(20, 15).Value = lm_sum
    .Cells(20, 17).Value = mn_sum
    .Cells(21, 17).Value = n0_sqrd
    .Cells(21, 15).Value = ln_sum
    .Cells(21, 16).Value = mn_sum
    .Cells(19, 18).Value = "v1"
    .Cells(20, 18).Value = "v2"
    .Cells(21, 18).Value = "v3"
    .Cells(20, 19).Value = "="
    '(20,20) reserved for Lambda
    .Cells(19, 21).Value = "v1"
    .Cells(20, 21).Value = "v2"
    .Cells(21, 21).Value = "v3"
  End With
End Sub

```



```

Sub Gauss(Lambda)
'this subroutine performs simple row elimination as in Gaussian elimination
Dim T_Prime(1 To 3, 1 To 3) As Double
n = 3
'initialize the data structures A and B; the results get saved to X
  For i = 1 To n
    With Worksheets("STATS")
      For j = 1 To n
        T_Prime(i, j) = .Cells(i + 18, j + 14).Value
        If i = j Then
          T_Prime(i, j) = .Cells(i + 18, j + 14).Value - Lambda
        End If
      Next
    End With
  Next
'row reduction
T_Prime(2, 2) = T_Prime(2, 2) / T_Prime(2, 1)
T_Prime(2, 3) = T_Prime(2, 3) / T_Prime(2, 1)
T_Prime(2, 1) = 1
T_Prime(3, 2) = T_Prime(3, 2) / T_Prime(3, 1)
T_Prime(3, 3) = T_Prime(3, 3) / T_Prime(3, 1)
T_Prime(3, 1) = 1
'subtract row 1 from rows 2 and 3
  For i = 1 To 3
    T_Prime(2, i) = T_Prime(2, i) * T_Prime(1, 1)
    T_Prime(3, i) = T_Prime(3, i) * T_Prime(1, 1)
  Next
  For i = 1 To 3
    T_Prime(2, i) = T_Prime(2, i) - T_Prime(1, i)
    T_Prime(3, i) = T_Prime(3, i) - T_Prime(1, i)
  Next
'end of subtraction
T_Prime(2, 3) = T_Prime(2, 3) / T_Prime(2, 2)
T_Prime(3, 3) = T_Prime(3, 3) / T_Prime(3, 2)
T_Prime(2, 2) = 1
T_Prime(3, 2) = 1

```

```

'subtract row 2 from 3
  For i = 1 To 3
23   T_Prime(3, i) = T_Prime(3, i) - T_Prime(2, i)
  Next
'reduce row 1
T_Prime(1, 1) = T_Prime(1, 1) / T_Prime(1, 2)
T_Prime(1, 3) = T_Prime(1, 3) / T_Prime(1, 2)
T_Prime(1, 2) = 1
'subtract row 2 and row 1 to give row 1
  For i = 1 To 3
24   T_Prime(1, i) = T_Prime(1, i) - T_Prime(2, i)
  Next
7  'end of subtraction and row reductions
  With Worksheets("STATS")
    .Cells(26, 14).Value = "T minus Lambda (I) row reduced"
    For i = 1 To n
      For j = 1 To n
25   35   37   .Cells(25 + i, 14 + j).Value = T_Prime(i, j)
      Next 'j
    Next 'i
    .Cells(30, 15).Value = "The reader will have to determine the eigenvector from here."
  End With
End Sub

```

8.10 Kentucky Example

Figure 8.6 shows the histogram for the Kentucky 2003 data. There does not appear to be any one or two observations with a disproportional weight. This data looks fairly random. Figure 8.7 shows the Kentucky 2003 converted data. The Y_i and Z_i observations have been interchanged in the graph. This is because the Z -axis points directly at the reader. Had the data not been interchanged, only one circle would have been shown. The 91 data markers, shown as circles, would lay on top of each other since the data is fairly linear.

We run the `Newton()` subroutine in Excel to calculate the eigenvalues. Since the equation of the determinant of B is triadic, we expect to find three roots to the determinant. The `Newton()` subroutine will find two of the three. The third λ_3 can be deduced by the equation $\lambda_3 = n - \lambda_1 - \lambda_2$ where λ_1 and λ_2 are known eigenvalues that have been found using the `Newton()` subroutine. Using a value close to n is a good guess for λ_3 . Here, there is no implied ordering between λ_1 , λ_2 , and λ_3 . Table 8.2 gives the eigenvalues and eigenvectors for the Kentucky data.

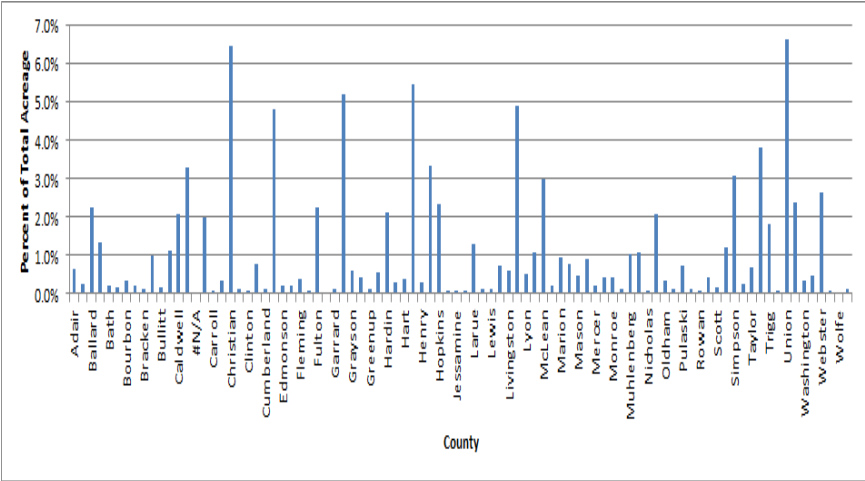


Figure 8.6 This figure shows the histogram for the Kentucky 2003 data.

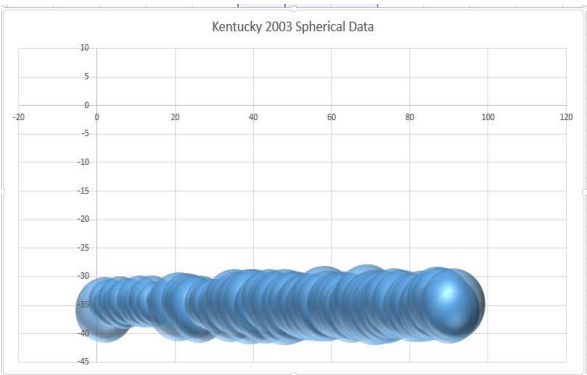


Figure 8.7 This figure shows the Kentucky 2003 converted data.

Matrix B for the 2003 Kentucky data is as follow:

$$B = \begin{pmatrix} 33.58777584 & -3.351383372 & 43.36764038 \\ -3.351383372 & 89.77014071 & 2.59054283 \\ 43.36764038 & 2.59054283 & 56.64208345 \end{pmatrix}$$

The reader can verify that the diagonal elements sum to $2n = 180$. Matrix T for the 2003 Kentucky data is as follow:

$$T = \begin{pmatrix} 56.41222416 & 3.351383372 & -43.36764038 \\ 3.351383372 & 0.229859286 & -2.59054283 \\ -43.36764038 & -2.59054283 & 33.35791655 \end{pmatrix}$$

Table 8.2 Eigenvalues and Eigenvectors for the 2003 Kentucky Data

	Value	Eigenvector \underline{t}'_i
λ_1	0.006629744	(0.734788415, 0.573334304, 1.0)
λ_2	0.034874073	(0.944911442, -2.955186475, 1.0)
λ_3	89.95849618	(-1.300507805, -0.077445098, 1.0)

The reader can verify that the property $T\underline{v}^T = \lambda_i\underline{v}^T$ where the vector \underline{v} is the eigenvector corresponding to the eigenvalue λ_i for $i = 1, 2, 3$. Just looking at the magnitude of the values of λ_1 , λ_2 , and λ_3 we can conclude that the eigenvalues are not equal. Hence, at a minimum the data is directional. We will calculate the test statistic S_U for illustrative purposes.

$$S_U = \frac{15}{2(90)} \sum_{i=1}^3 \left(\lambda_i - \frac{1}{3}(90) \right)^2 =$$

$$\frac{15}{180} ((0.006629744 - 30)^2 + (0.034874073 - 30)^2 +$$

$$(89.95849618 - 30)^2) = 449.378.$$

The cut-off value is $\chi^2(5) = 11.07$ at the 95% confidence level and $\chi^2(5) = 15.09$ at the 99% confidence level. Since $449 > 15.09$, we reject the hypothesis that the eigenvalues are equally valued. Thus, the data is directional. We need the resultant length R to determine the visual distribution of the points. This Section covers the derivation of R .

To obtain the results in Table 8.3, we ran the `modal_stats()` routine. It does not have any input parameters, but does assume that the first three columns in the spreadsheet have the following values:

- Column 1 contains the latitude values.

Table 8.3 Results for the Spherical Model for Kentucky

2003	
Center	(45.35963106, 2.701085798, -34.87836501)
(l_0, m_0, n_0)	(4082.366795, 243.0977218, -3139.052851)
Resultant Length	$R = 0.999769376$
Sample Size	$n = 90$
Data Type	Directional
Data Shape	Bipolar
Spherical Variance	$S = 0.000461194$

- Column 2 contains the longitude values.
- Column 3 contains the weight values (random variable). This is not used in the computations.

In Table 8.2, we notice we have two small eigenvalues and one large eigenvalue. According to Table 8.1, these observations have a bipolar distribution because λ_3 is large; λ_1 and λ_2 are small. The concentration of the spherical data is at both ends of the vector \underline{t}_3 . We knew the data originated from a bipolar coordinate system with a north and south pole. The vector \underline{t}_3 in radians is $\underline{t}_3 = (-1.300507805, -0.077445098, 1.0)$.

8.11 Crime Example

We look at the histogram in Figure 8.8 and notice three states with approximately the same mode — California, Texas, and Florida. California accounts for approximately 13.6% of the total violent crime, Texas 8.7%, and Florida 9.4%. These three observations weigh the calculations disproportionately. One is on the west coast, one in the south west, and the other in the south. Figure 8.9 shows a graph of the converted data. The data is symmetric. One ball is separated from the rest of the data.

We need to determine the visual distribution of the data. To accomplish this, we first find the eigenvalues of matrix T (uses the 2007 data). The matrices B and T are as follow:

$$B = \begin{pmatrix} 19.82660121 & 0.299933647 & 22.9389611 \\ 0.299933647 & 47.65668036 & -0.234303397 \\ 22.9389611 & -0.234303397 & 30.51671843 \end{pmatrix}$$

$$T = \begin{pmatrix} 29.17339879 & -0.299933647 & -22.9389611 \\ -0.299933647 & 1.343319644 & 0.234303397 \\ -22.9389611 & 0.234303397 & 18.48328157 \end{pmatrix}$$

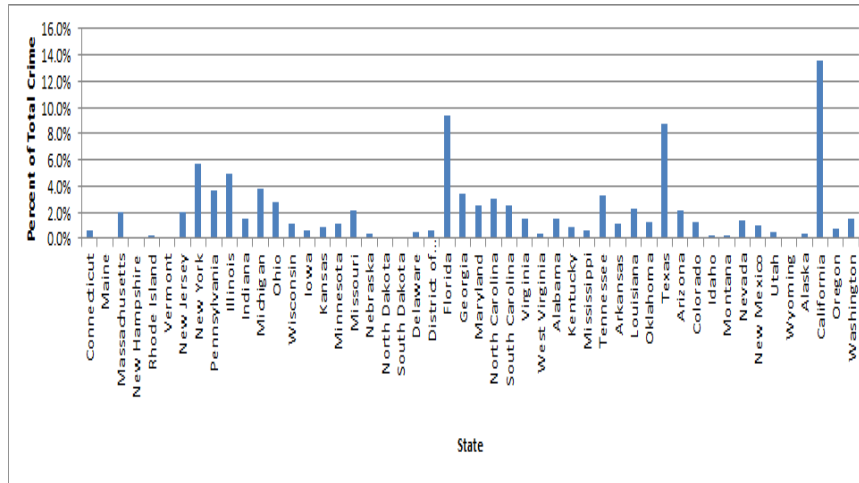


Figure 8.8 This figure shows the histogram for the Violent Crime 2008 data. Three possible modes appear at California, Texas, and Florida.

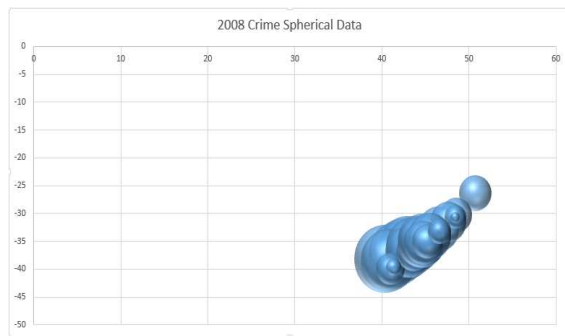


Figure 8.9 This figure shows the 2008 US Crime converted data.

Table 8.4 gives the eigenvalues and eigenvectors.

We test the hypothesis that the eigenvalues are equal versus they are not equal using Equation (8.16) using the test statistic S_U .

$$\begin{aligned}
 S_U &= \frac{15}{2(49)} \sum_{i=1}^3 \left(\lambda_i - \frac{1}{3}(49) \right)^2 = \\
 &\frac{15}{98} [(0.27486963 - 16.33)^2 + (1.340181802 - 16.33)^2 + \\
 &\quad (47.38494857 - 16.33)^2] = 221.46.
 \end{aligned}$$

Table 8.4 Eigenvalues and Eigenvectors for the 2008 Vilonent Crime Data

Diagonal	Value	Eigenvector \underline{t}_i'
λ_1	0.27486963	(0.793812871, 0.003545129, 1.0)
λ_2	1.340181802	(-0.649158203, -136.7206467, 1.0)
λ_3	47.38494857	(-1.259802117, 0.013295803, 1.0)

Table 8.5 Results for the Spherical Model for 2008 Crime

2008	
Center	(44.11484743, -0.535389408, -35.03421925)
(l_0, m_0, n_0)	(2161.627524, -26.23408097, -1716.676743)
Resultant Length	$R = 0.983257676$
Sample Size	$n = 48$
Data Type	Directional
Data Shape	Unimodal
Spherical Variance	$S = 0.033204342$

We reject the null hypothesis $H_0 : \lambda_i = \frac{n}{3}, i = 1, 2, 3$ since $221.46 > 15.09$ at the 99% confidence level. We also reject the hypothesis that the 2008 crime data is uniform. The usual statistics for the sphere are given in Table 8.5.

Next, we determine the data shape by examining the λ_i' s. Using the λ_i' s in Table 8.4 and the interpretation in Table 8.1, we most likely have a unimodal distribution because λ_3 is large; λ_1 and λ_2 are small; R is close to 1. Since this is a unimodal distribution, the concentration of the data is at one end of the vector \underline{t}_3 where $\underline{t}_3 = (-1.259802117, 0.013295803, 1.0)$.

8.12 GDP Example

We begin by looking at the histogram of the observations in Figure 8.10. Two bars appear longer than the rest. These indicate modes in the distribution of the data. The United States accounts for approximately 25% of the total GDP and China accounts for approximately 14%. This will influence the shape of the data. Figure 8.11 shows the converted data. The data is spread out into the eight quadrants of the X , Y , and Z planes.

The matrices B and T for the GDP 2008 data is

$$B = \begin{pmatrix} 19.35173834 & -10.96157833 & -2.888240385 \\ -10.96157833 & 23.90143271 & -1.953298019 \\ -2.888240385 & -1.953298019 & 34.74682895 \end{pmatrix}$$

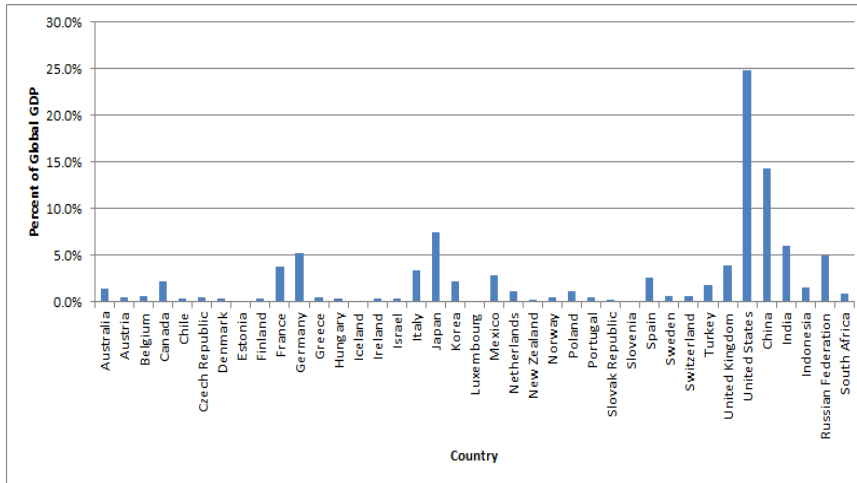


Figure 8.10 This figure shows the histogram for the GDP 2008 data. The U.S. accounts for approximately 25% of the total GDP and China accounts for approximately 14%.

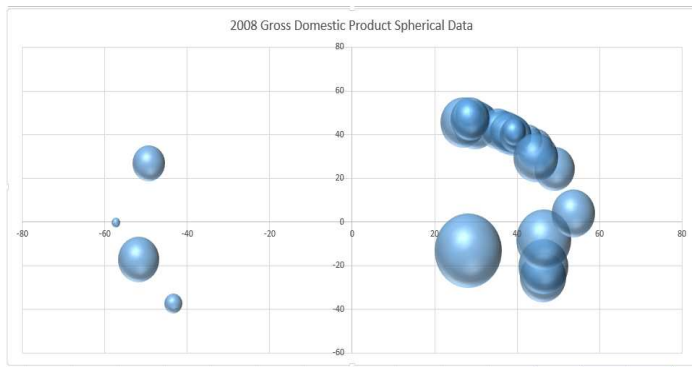


Figure 8.11 This figure shows the 2008 Gross Domestic Product converted data.

The diagonal of B sums to $2n = 2 \times 39 = 78$.

$$T = \begin{pmatrix} 19.64826166 & 10.96157833 & 2.888240385 \\ 10.96157833 & 15.09856729 & 1.953298019 \\ 2.888240385 & 1.953298019 & 4.253171052 \end{pmatrix}$$

The diagonal of T sums to $n = 39$. We test the hypothesis that the eigenvalues are equal versus they are not equal using Equation (8.16) using the test statistic S_U .

Table 8.6 Eigenvalues and Eigenvectors for the 2008 Gross Domestic Product

Diagonal	Value	Eigenvector \underline{t}_i'
λ_1	3.728832776	(-0.187814653, 0.00927436, 1.0)
λ_2	6.216179462	(5.008334449, -6.400592894, 1.0)
λ_3	29.05498776	(5.546422255, 4.496198733, 1.0)

Table 8.7 Results for the Spherical Model for GPD

2008	
Center	(27.14815799, 24.99879834, 6.46609507)
(l_0, m_0, n_0)	(1058.778161, 974.9531353, 252.1777077)
Resultant Length	$R = 0.653921742$
Sample Size	$n = 39$
Data Type	Directional
Data Shape	Bimodal
Spherical Variance	$S = 0.572386356$

$$S_U = \frac{15}{2(39)} \sum_{i=1}^3 \left(\lambda_i - \frac{1}{3}(39) \right)^2 =$$

$$\frac{15}{78} [(3.728832776 - 13)^2 + (6.216179462 - 13)^2 +$$

$$(29.05498776 - 13)^2] = 74.95.$$

We reject the null hypothesis $H_0 : \lambda_i = \frac{n}{3}, i = 1, 2, 3$ since $74.95 > 15.09$ at the 99% confidence level. We also reject the hypothesis that the 2008 crime data is uniform. The usual statistics for the sphere are given in Table 8.7. Table 8.6 shows the eigenvalues and eigenvectors for the 2008 GDP example.

Since the eigenvalue λ_3 is larger relative to λ_2 and λ_1 , and since R is not close to 1, this may be a bimodal distribution. The concentration is at both ends of the vector $\underline{t}_3 = (5.546422255, 4.496198733, 1.0)$.

APPENDIX A

SDE VBA DRIVERS

The code for calculating the weighted standard deviational ellipse resides in Module 1 in the Visual Basic Editor . Only highlight the observations themselves using the <shift> ↓ key combination, not the entire column by clicking on the column headings. The subroutine `Set_weights()` sets the weights on the spreadsheet and puts them in the fourth column. It is important that the reader only use columns 1 – 3 for his or her data. The VBA code in Module 1 using the `SDE()` subroutine calculates the following statistics.

- The weighted mean center (\bar{x}, \bar{y}) .
- The area F .
- The semi-major a and semi-minor b axes lengths.
- The error terms on the axes δ_x , and δ_y .
- The angle of rotation on the axes θ_x , and θ_y .
- The eccentricity of the ellipse e .

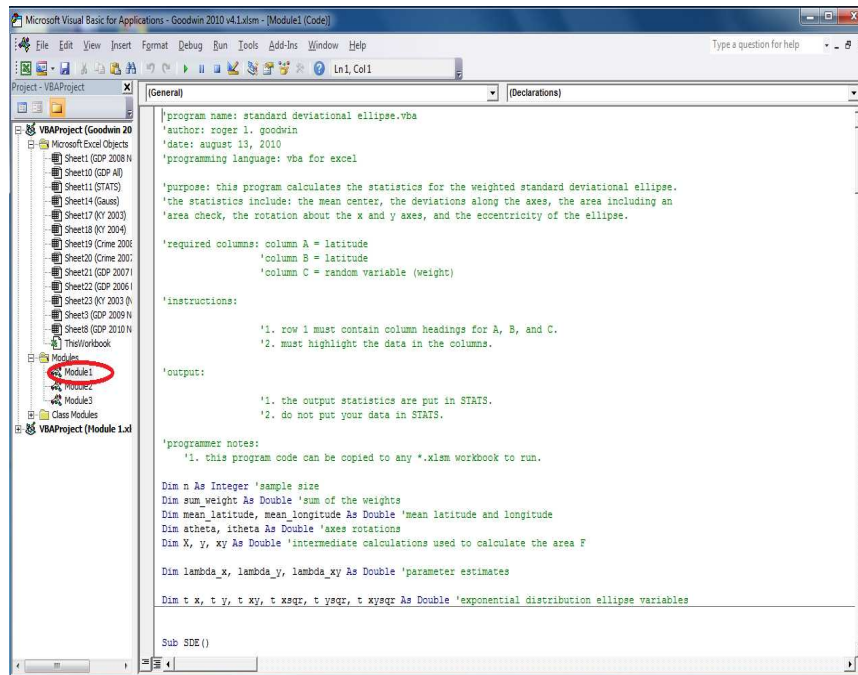


Figure A.1 In Excel, double click on Module 1 in the left-hand pane circled in red to display the VBA code for the Standard Deviational Ellipse. The subroutine called SDE() runs the programs to calculate the statistics in this chapter.

To run the SDE code, follow these steps. Select and highlight the data.

1. Click in the spreadsheet tab that contains the data you wish to analyze.
2. Ensure that the first three columns of data (and only the data) are highlighted (selected). If not, follow these steps:
 - (a) Click on cell A1.
 - (b) While holding down the <SHIFT> key, arrow over to Column C. The headings of the data should be highlighted and selected.
 - (c) While still holding down the <SHIFT> key, hold down the <CTRL> key.
 - (d) Touch the down arrow key once. This should take you to the bottom of the data.

Figure A.2 shows the VBA environment for Module 1. To enter the VBA environment and run the standard deviation ellipse code:

1. Click on the Developer ribbon.

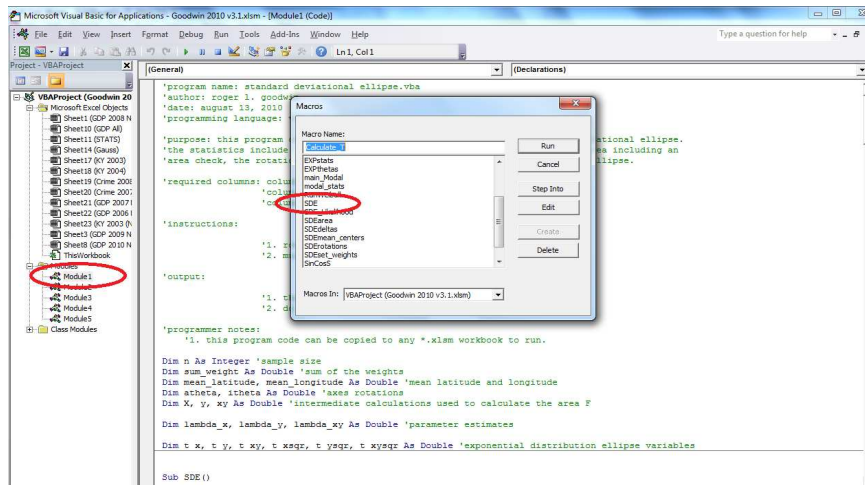


Figure A.2 This figure shows the VBA environment for Module 1. It also highlights the driver subroutine `SDE()`.

2. Click on the Visual Basic button on the left side.
3. Click on Module 1 on the left-side pane.

This module has seven subroutines.

- `SDE()` — This subroutine is the driver that calls the other subroutines.
- `Set_weights()` — This subroutine calculates the weights and saves them to the active data spreadsheet.
- `SDEmean_centers()` — This subroutine calculates the mean latitude, \bar{x} , and the mean longitude, \bar{y} .
- `Angle_of_Rotation()` — This subroutine calculates major axis rotation and minor axis rotation. The sum of the absolute value of both will always sum to 90° .
- `SDEdeltas()` — This subroutine calculates the δ_x for the latitude and the δ_y for the longitude.
- `Area()` — This subroutine calculates the major axis length, minor axis length, the area, and the eccentricity.
- `SDELikelihood()` — This subroutine calculates the log likelihood function for the weighted regression model.

The `SDE()` subroutine (1) drives the standard deviatonal ellipse subroutines. The weighted standard deviatonal ellipse is one of the simplest concepts to implement, and thus, does not require input from the user. A *circular coordinate system*

NETWORKDAYS ✕ ✓ <i>f_x</i> =MOD(360,B3)		
	A	B
1	Standard Deviational Ellipse Statistics	
2	Latitude Mean Value =	37.38297399
3	Longitude Mean Value =	273.1870486
4	Rotation	Rotate on Y-Axis
5	Major Axis Rotation	70.36257022
6	Minor Axis Rotation	-19.63742978
7	Delta X	1.29151776
8	Delta Y	1.625551351
9	Area (F)	30336.30686
10	Major Axis Length (a)	184.8734659
11	Minor Axis Length (b)	52.23219209
12	Area Check	30336.30686
13	Eccentricity (e)	0.959258636
14	Likelihood L(x,y)	181,824,055.41
15		
16		

Figure A.3 This figure shows the `STATS` worksheet and the results from the `SDE()` subroutine.

is not required, either. The `SDE()` routine will return the same results with either data set.

To run the `SDE()` subroutine, make sure you are in the upper-left corner of the VBA Editor (Line 1, Column 1).

1. Select the menu option `Run`.
2. Select the sub-menu item `Run Sub/UserForm`.

The subroutines called in the `SDE` subroutine run quickly. As stated several times, the results will appear in the `STATS` spreadsheet.

Upon clicking on the `STATS` tab, the `SDE()` results appear in columns A and B. See Figure A.3. The `SDE` statistics and their locations in the `STATS` spreadsheet are as follow:

```

'program name: standard deviational ellipse.vba
'author: roger l. goodwin
'date: august 13, 2010
'programming language: vba for excel

'purpose: this program calculates the statistics for
' the weighted standard deviational ellipse.
' the statistics include: the mean center, the
' deviations along the axes, the area including an
' area check, the rotation about the x and y axes,
' and the eccentricity of the ellipse.

'required columns: column A = latitude
'column B = longitude
'column C = random variable (weight)

'instructions:

'1. row 1 must contain column
' headings for A, B, and C.
'2. must highlight the data in the
' columns.

'output:
'1. the output statistics are put
' in STATS.
'2. do not put your data in STATS.

'programmer notes:
'1. this program code can be copied to any *.xlsm
' workbook to run.

Dim n As Integer 'sample size
Dim sum_weight As Double 'sum of the weights

```

```

' mean latitude and longitude
Dim mean_Latitude, mean_Longitude As Double
' axes rotations
Dim atheta, itheta As Double
' intermediate calculations used to calculate
' the area F
Dim X, y, xy As Double
' parameter estimates
Dim lambda_x, lambda_y, lambda_xy As Double
' exponential distribution ellipse variables
Dim t_x, t_y, t_xy, t_xsqr, t_ysqr, t_xysqr As Double

' run the routines that calculate the standard
' deviational ellipse statistics

```

```

1 Sub SDE()
  Call Set_weights
  Call SDEmean_centers
  Call Angle_of_Rotation
  Call SDEdeltas
  Call SDEarea
  Call SDE_Likelihood
End Sub

```

- The weighted mean center from Equation (5.9), (\bar{x}, \bar{y}) , \bar{x} appears in cell B2, and \bar{y} appears in cell B3 .
- The axis of rotation in Equation (5.10) appears in cell B4 .
- The major axis angle of rotation θ appears in cell B5 .
- The minor axis angle of rotation θ appears in cell B6 .
- The standard error from Equation (5.11) in the latitude direction x appears in cell B7 .
- The standard error from Equation (5.12) in the longitude direction y appears in cell B8 .
- The area F of the weighted standard deviational ellipse from Equation (5.17) appears in cell B9 .
- The semi-major axis length a from Equation (5.15) appears in cell B10 .

- The semi-minor axis length b from Equation (5.16) appears in cell B11 .
- The area check using the formula $F = \pi ab$ appears in cell B12 .
- The eccentricity e of the standard deviational ellipse from Equation (5.18) appears in cell B13 .
- The logarithm of the likelihood function of the regression model from Equation (??) appears in cell B14 . This will be discussed in Section 7.13.

This application does not write in column C . Use column C to transform the mean center coordinates, if needed using Equation (3.3) or (3.5). See Figure A.3.

Implementation Issue 1: The implementation of the modulo function `mod()` differs in the VBA application versus the Excel spreadsheet. The VBA function returns an integer. The spreadsheet function returns a real number. Because of this difference, it is the responsibility of the user to convert the latitude and the longitude observations to circular coordinates on the input data set. This applies to every data set. It becomes more important for our global GNP data. If in doubt, it is never wrong to apply the formulas in Section 3.7 to the input coordinates and the output statistics for the mean center.

Implementation Issue 2: The implementation of the modulo function `mod()` differs in the VBA application versus the Excel spreadsheet. The VBA function returns an integer. The spreadsheet function returns a real number. Because of this difference, it is the responsibility of the user to convert the mean latitude and the mean longitude from circular coordinates to the original coordinate system.

APPENDIX B

EXP VBA DRIVERS

It is in `Module 2` that this textbook provides an Excel 2010 VBA program to calculate the statistics that accompany the weighted, exponential deviational ellipse. The VBA code calculates the following statistics.

```

'program name: EXPellipse.vba
'author: roger l. goodwin
'define the exponential deviational ellipse routines

Dim wx, wy As Double
Dim lambda_x, lambda_y, lambda_xy As Double 'parameter estimates
Dim t_x, t_y, t_xy, t_xsqr, t_ysqr, t_xysqr As Double 'intermediate calculations
Dim sum_weight 'sum of the weights
Dim atheta, itheta As Double 'angles of rotation
Dim L_x, L_y, L_xy As Double 'log likelihood functions
Dim n As Integer

Sub EXPellipse()
    Call EXPclean_data
    Call Set_weights
    Call EXPparameters
1 Call EXPdistributions
    Call EXPthetas
    Call EXPdeviations
    Call EXPstats
End Sub

```

To run the exponential ellipse code, follow these steps.

1. Click in the spreadsheet tab that contains the data you wish to analyze.
2. Select and highlight the data first. Follow these steps to select and highlight the data.
 - (a) Click on cell A1 .
 - (b) While holding down the <SHIFT> key, arrow over to column C . The headings of the data should be highlighted and selected.
 - (c) While still holding down the <SHIFT> key, hold down the <CTRL> key.
 - (d) Touch the down arrow key once. This should take you to the bottom of the data.

Figure B.1 shows the VBA environment for Module 2 . To enter the VBA environment and run the exponential ellipse code:

1. Click on the Developer ribbon.

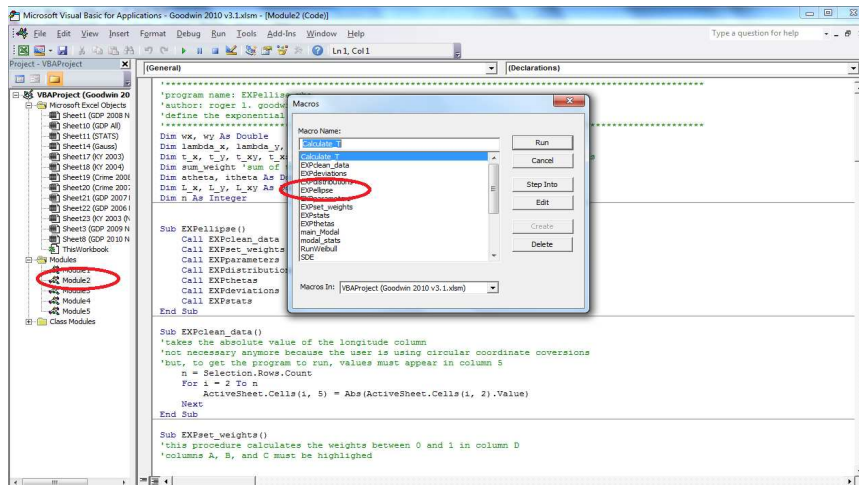


Figure B.1 *This figure shows the VBA environment for Module 2. It also highlights the driver subroutine `EXPellipse()`.*

2. Click on the **Visual Basic** button on the left side.
3. Click on **Module 2** on the left-side pane.

This module has eight subroutines.

- `EXPclean_data()` — Not used.
- `Set_weights()` — This subroutine calculates the weights and saves them to the active data spreadsheet.
- `EXPparameters()` — This subroutine calculates the parameter estimates $\widehat{\lambda}_x$, $\widehat{\lambda}_y$, and $\widehat{\lambda}_{xy}$.
- `EXPdistributions()` — This subroutine calculates a variety of sums. It also calculates the log likelihood function. Other subroutines use these sums.
- `EXPthetas()` — This subroutine calculates major axis rotation and minor axis rotation. The sum of the absolute value of both will always sum to 90° .
- `EXPdeviations()` — This subroutine calculates the δ_x for the latitude and the δ_y for the longitude.
- `EXPstats()` — This subroutine calculates the major axis length, minor axis length, the area, and the eccentricity. It also writes the log likelihood value to the STATS spreadsheet.

The `EXPellipse()` subroutine drives the exponential deviational ellipse subroutines. The weighted exponential deviational ellipse is another one of the simplest

D	E	F
Exponential Ellipse Statistics		
Mean Latitude	37.38297399	
Mean Longitude (Abs)	273.1870486	=MOD(360,E3)
Rotation	Rotate on Y-Axis	
Major Axis Rotation	89.9986581	
Minor Axis Rotation	-0.001342193	
Delta X	3.084687614	
Delta Y	3.084707388	
Area (F)	32450.8332	
Major Axis Length (a)	274.7458265	
Minor Axis Length (b)	37.59636008	
Area Check	32450.90232	
Eccentricity (e)	0.990593092	
Likelihood Function L(x)	-44211.86328	
Likelihood Function L(y)	-60300.24289	
Likelihood Function L(xy)	-111454.5019	

Figure B.2 This figure shows the STATS worksheet and the results from the EXPellipse() subroutine.

concepts to implement, and thus, does not require input from the user. A circular coordinate system is required.

To run the EXPellipse() subroutine, make sure you are in the upper-left corner of the VBA Editor (Line 1, Column 1).

1. Select the menu option Run .
2. Select the sub-menu item Run Sub/UserForm .

The subroutines called in the EXPellipse() subroutine run quickly. As previously stated, the results will appear in the STATS spreadsheet.

Upon clicking on the STATS tab, the EXPellipse() results appear in columns D and E. See Figure B.2. The EXPellipse() statistics and their locations in the STATS spreadsheet are as follow:

- For the weighted mean center (\bar{x}, \bar{y}) , \bar{x} appears in cell E2, and \bar{y} appears in cell E3.
- The axis of rotation appears in cell E4.
- The major axis angle of rotation θ appears in cell E5.
- The minor axis angle of rotation θ appears in cell E6.
- The standard deviation in the latitude direction x appears in cell E7.

- The standard deviation in the longitude direction y appears in cell E8 .
- The area F of the weighted standard deviational ellipse appears in cell E9 .
- The semi-major axis length a appears in cell E10 .
- The semi-minor axis length b appears in cell E11 .
- The area check using the formula $F = \pi ab$ appears in cell E12 .
- The eccentricity e of the standard deviational ellipse appears in cell E13 .
- The logarithm of the likelihood function of the regression model for the latitude x appears in cell E14 .
- The logarithm of the likelihood function of the regression model for the longitude y appears in cell E15 .
- The logarithm of the likelihood function of the regression model for the joint distribution of the latitude and longitude (x, y) appears in cell E16 .

This application does not write in column F . If either of the mean center coordinates need to be transformed via Equation (3.3) or (3.5), then the reader can use column F for those calculations. See Figure B.2. Notice that the two values for the area (with-in the exponential ellipse statistics are equivalent). Also notice that the mean center for the exponential ellipse and for the standard deviational ellipse are the same.

- The weighted mean center (\bar{x}, \bar{y}) .
- The area.
- The semi-major a and semi-minor b axes lengths.
- The error terms on the axes δ_x , and δ_y .
- The angle of rotation on the axes θ_x , and θ_y .
- The eccentricity of the ellipse e .

Should the reader wish to calculate un-weighted exponential deviational ellipses, set the weight column to 1.0.

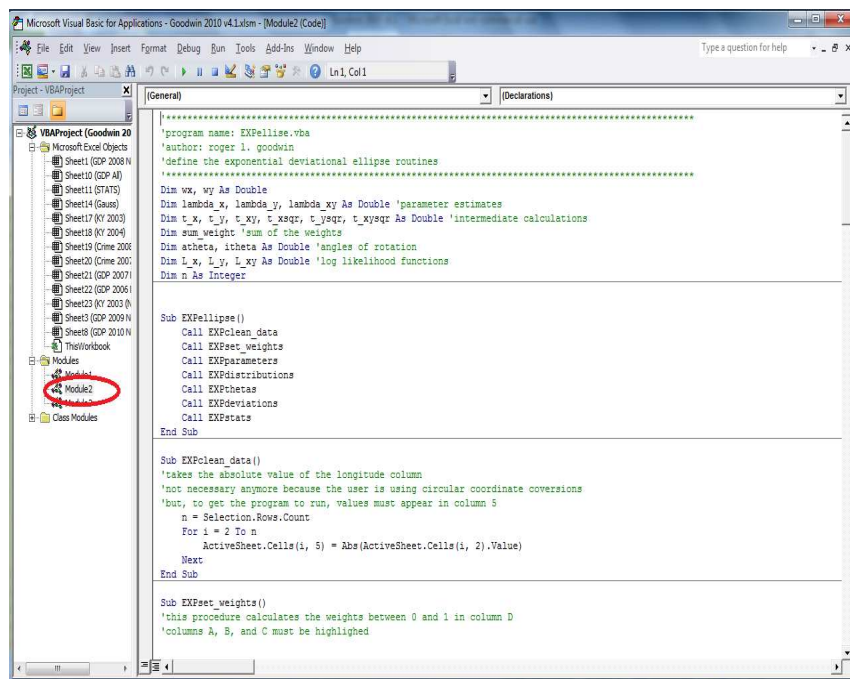


Figure B.3 Double click on Module 2 to display the VBA code for the Exponential Deviational Ellipse in Excel. The subroutine called EXPellipse() runs the programs to calculate the statistics in this chapter.

APPENDIX C

WEIBULL VBA DRIVERS

Running the Weibull ellipse takes considerable more interaction and time than that for the previous two ellipses. The secant algorithm finds three parameters. Two initial guesses for each parameter must accompany each run. Two secant algorithms have been programmed — one for the individual, weighted variables (x_i, w_i) and (y_i, w_i) called `Secant1()` and a second algorithm for the joint distribution of (x_i, y_i, w_i) called `Secant2()`. This was necessary because the likelihood functions are different.

Figure C.1 shows the VBA environment for `Module 3`. This module has five subroutines.

- `RunWeibull()` — This subroutine is the driver that calls the other subroutines.
- `Secant1()` — This subroutine calculates the point estimates for the latitude $\hat{\lambda}_x$ and $\hat{\gamma}_x$. It can also be used to calculate the point estimates for the longitude $\hat{\lambda}_y$, and $\hat{\gamma}_y$.

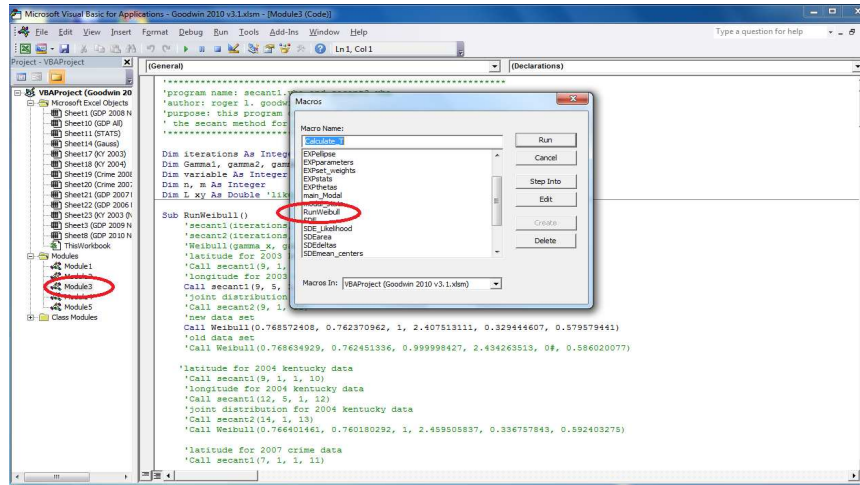


Figure C.1 This figure shows the VBA environment for Module 3. It also highlights the driver subroutine `RunWeibull()`.

- `Secant2()` — This subroutine calculates the point estimates for the joint distribution $\hat{\lambda}_{xy}$ and $\hat{\gamma}_{xy}$.
- `Log10(X)` — This subroutine calculates the logarithm of the given number X using base 10.
- `Weibull()` — This subroutine calculates all of the statistics for the Weibull ellipse. These statistics include the mean center (\bar{x}, \bar{y}) , the axis of rotation, the rotations θ_x and θ_y , the standard deviations on the axes δ_x and δ_y , the major and minor axes lengths a and b , the area F , the eccentricity e , and the log likelihood function $\log L_{xy}$.

The `Secant1()` algorithm has the following parameters:

- `Iterations` — This parameter is a termination condition. This is the maximum number of iterations that the secant algorithm should perform.
- `Variable` — This parameter is the column number of the variable (not the name in row 1). This will be either column 1 or column 2.
- `Gamma1` — This parameter is the first initial guess at γ_x or γ_y .
- `Gamma2` — This parameter is the second initial guess at γ_x or γ_y . Note that `Gamma1 < Gamma2`.

The `Secant1()` algorithm returns the values $\hat{\gamma}_x$ and $\hat{\lambda}_x$, or $\hat{\gamma}_y$ and $\hat{\lambda}_y$, depending on if `Variable = 1` or `Variable = 2`. All four point estimates have to be saved or recorded where they will not be over-written.

J	K	L	M
	Weibull Ellipse Statistics		
	Mean Latitude	37.38297399	
	Mean Longitude (Abs)	273.1870490	
	Rotation	Rotate on X-Axis	
	Major Axis Rotation	89.89385427	
	Minor Axis Rotation	-0.10614571	
	Delta X	1.77010417	
	Delta Y	1.779642002	
	Area (F)	27.11384518	
	Major Axis Length (a)	3.720699854	
	Minor Axis Length (b)	2.319618704	
	Area Check	27.11384518	
	Eccentricity (e)	0.781874284	
	Likelihood Function L(x,y)	-137452.7083	

Figure C.2 This figure shows the STATS worksheet and the results from the Weibull() subroutine.

The Secant2() algorithm has the following parameters:

- Iterations — This parameter is a termination condition. This is the maximum number of iterations that the secant algorithm should perform.
- Gamma1 — This parameter is the first initial guess at γ_x or γ_y .
- Gamma2 — This parameter is the second initial guess at γ_x or γ_y . Note that $\text{Gamma1} < \text{Gamma2}$.

Since the Secant2() subroutine returns the values $\hat{\gamma}_{xy}$ and $\hat{\lambda}_{xy}$, there is no need to specify the latitude or longitude variable. Save the point estimates so that they will not be over-written.

After finding the six point estimates of the parameters, run the Weibull() subroutine.

- $\hat{\gamma}_x$ — The point estimate for the latitude for the shape parameter.
- $\hat{\gamma}_y$ — The point estimate for the longitude for the shape parameter.
- $\hat{\gamma}_{xy}$ — The point estimate for the joint distribution for the shape parameter.
- $\hat{\lambda}_x$ — The point estimate for the latitude for the scale parameter.
- $\hat{\lambda}_y$ — The point estimate for the longitude for the scale parameter.
- $\hat{\lambda}_{xy}$ — The point estimate for the joint distribution for the scale parameter.

Upon clicking on the STATS tab, the `Weibull()` results appear in columns K and L. See Figure C.2. The `Weibull()` statistics and their locations in the STATS spreadsheet are as follow:

- For the weighted mean center (\bar{x}, \bar{y}) , \bar{x} appears in cell L2, and \bar{y} appears in cell L3.
- The axis of rotation appears in cell L4.
- The major axis angle of rotation θ appears in cell L5.
- The minor axis angle of rotation θ appears in cell L6.
- The standard deviation in the latitude direction x appears in cell L7.
- The standard deviation in the longitude direction y appears in cell L8.
- The area F of the weighted standard deviational ellipse appears in cell L9.
- The semi-major axis length a appears in cell L10.
- The semi-minor axis length b appears in cell L11.
- The area check using the formula $F = \pi ab$ appears in cell L12.
- The eccentricity e of the standard deviational ellipse appears in cell L13.
- The logarithm of the likelihood function of the regression model for the joint distribution of the latitude and longitude (x, y) appears in cell L14.

```

'program name: secant1.vba and secant2.vba
'author: roger l. goodwin
'purpose: this program calculates the parameter estimates using
' the secant method for a distribution with a single variable.
Dim iterations As Integer
Dim Gamma1, gamma2, gamma3 As Double
Dim variable As Integer
Dim n, m As Integer
Dim L_xy As Double 'likelihood function

```

```

1 Sub RunWeibull()
  'documentation:
  'secant1(iterations, variable, Gamma1, gamma2)
  'secant2(iterations, Gamma1, gamma2)
  'Weibull(gamma_x, gamma_y, gamma_xy, lambda_x, lambda_y, lambda_xy)

  'latitude for 2003 kentucky data
  'Call secant1(9, 1, 1, 10)
  'longitude for 2003 kentucky data
  Call secant1(9, 5, 1, 10)
  'joint distribution for 2003 kentucky data
6 'Call secant2(9, 1, 11)
  'new data set
  Call Weibull(0.768572408, 0.762370962, 1, 2.407513111, 0.329444607, 0.579579441)
  'old data set
1 'Call Weibull(0.768634929, 0.762451336, 0.999998427, 2.434263513, 0#, 0.586020077)

  'latitude for 2004 kentucky data
  'Call secant1(9, 1, 1, 10)
  'longitude for 2004 kentucky data
7 'Call secant1(12, 5, 1, 12)
  'joint distribution for 2004 kentucky data
  'Call secant2(14, 1, 13)
  'Call Weibull(0.766401461, 0.760180292, 1, 2.459505837, 0.336757843, 0.592403275)

  'latitude for 2007 crime data
  'Call secant1(7, 1, 1, 11)
  'longitude for 2007 crime data
8 'Call secant1(9, 5, 1, 14)
  'joint distribution for 2007 crime data
  'Call secant2(10, 1, 17)
  'Call Weibull(0.765003479, 0.747808984, 1, 1.328755906, 0.182913031, 0.321560844)

```

```

1  [ 'latitude for 2008 crime data
    [ 'Call secant1(10, 1, 1, 13)
      'longitude for 2008 crime data
    9 [ 'Call secant1(15, 5, 1, 17)
      'joint distribution for 2008 crime data
      [ 'Call secant2(10, 1, 17)
        [ 'Call Weibull(0.765405522, 0.757628441, 1, 1.328288608, 0.18285781, 0.321461829)

    [ 'latitude for 2008 GDP data
      [ 'Call secant1(4, 1, 1, 10)
        'longitude for 2008 GDP data
      [ 'Call secant1(10, 2, 1, 16)
    10 [ 'joint distribution for GDP data
        [ 'Call secant2(12, 1, 18)
          [ 'Weibull(gamma_x, gamma_y, gamma_xy, lambda_x, lambda_y, lambda_xy)
            [ 'Call Weibull(0.778466889, 0.763823411, 1, 0.846866662, 0.256765441, 0.394055395)

    [ 'latitude for 2009 GDP data
      [ 'Call secant1(4, 1, 1, 10)
        'longitude for 2008 GDP data
      [ 'Call secant1(10, 2, 1, 16)
    11 [ 'joint distribution for GDP data
        [ 'Call secant2(12, 1, 18)
          [ 'Weibull(gamma_x, gamma_y, gamma_xy, lambda_x, lambda_y, lambda_xy)
            [ 'Call Weibull(0.777870853, 0.764074484, 1, 0.849145918, 0.259687836, 0.397738372)

    [End Sub

```

To obtain the results in this Section, this textbook provides three VBA programs for Excel 2010. Two of the programs perform the Secant estimation for the parameters γ_x , γ_y , and γ_{xy} . The other program calculates the usual statistics associated with an ellipse such as the mean center, the area and so on. The program `main` in Module 3 calculates the parameters λ_x , λ_y , γ_x and γ_y . The program `main2` in Module 3 calculates the parameters λ_{xy} and γ_{xy} . Note the values from these two programs. We must input them into the program `Weibull` in Module 3.

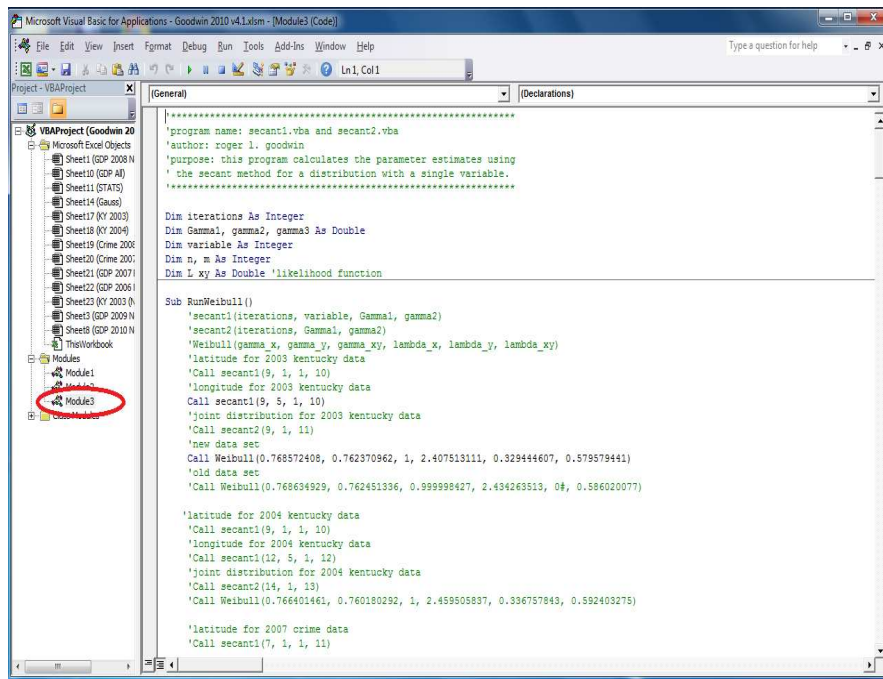


Figure C.3 Double click on Module 3 to display the VBA code for the Secant code and the Weibull Devotional Ellipse in Excel. The subroutines called Secant1(), Secant2(), and RunWeibull() run the programs to calculate the statistics in this chapter. As the reader may have guessed, calculating the statistics in this chapter is more complex than those in previous chapters.

APPENDIX D

SPHERICAL VBA DRIVERS

```

'program names: newton(), gauss()
'author: roger l. goodwin
'purpose: calculate the eigenvalues and the matrix T for the sphere
Dim Lambda As Double '1 to 3 possible eigenvalues
    Sub main_Modal()
        'syntax
        'Sub Newton(Lambda)
        'Sub Gauss(Lambda)
        'sub Calculation_T()
        Call Calculate_B
        Call Calculate_T
        'kentucky 2003
        'lambda_1
        'Call Newton(0)
4      'Call Gauss(0.006629744)
        'lambda_2
        'Call Newton(20)
15     'Call Gauss(0.034874073)
        'lambda 3
        'Call Newton(90)
        'Call Gauss(89.95849618)
        'kentucky 2004
        'lambda_2
        'Call Newton(0)
        'Call Gauss(7.53622608147269E-03)
        'lambda_1
        'Call Newton(2)
        'Call Gauss(8.25032435835292E-02)
        'final step — round the chosen shape to an integer for interpretation
        'Call Gauss(92)

```

```

4 15  'crime 2008
      'lambda_1
      'Call Newton(1)
      'Call Gauss(1.340181802)
      'lambda_2
      'Call Newton(30)
      'Call Gauss(0.27486963)
      'lambda_3
      'Call Newton(50)
      'Call Gauss(47.38494857)
      'final step — round the chosen shape to an integer for interpretation
      'Call Gauss(100)
      'testing
      'Call Newton(30)
      'Call Gauss(8.58033582016791E-02)
      'GDP 2008
      'lambda_1
      'Call Newton(0)
      'Call Gauss(3.728832776)
      'lambda_2
      'Call Newton(10)
      'Call Gauss(6.216179462)
      'lambda_3
      Call Newton(39)
      Call Gauss(29.05498776)
      End Sub

```


APPENDIX E

REFERENCES

1. R. Beran, "Exponential Models for Directional Data," *The Annals of Statistics*, vol. 7, no. 6 (Nov., 1979), p. 1162-1178.
2. B. L. Bowerman and R. T. O'Connell, *Linear Statistical Models, an Applied Approach*, Duxbury Press, 1990.
3. G. Brassard and P. Bratley (1988), *Algorithmics: Theory and Practice*, Prentice Hall, NJ.
4. M. B. Brown and A. B. Forsythe (1974), "Robust Tests for the Equality of Variances," *Journal of the American Statistical Association*, vol. 69, no. 346, p. 364-367.
5. W. Cheney and D. Kincaid (1985), *Numerical Mathematics and Computing*, Wadsworth, Inc., Belmont, California.
6. R. S. Chhikara and L. Deng, "Using Multiyear Rotation Design Sampling in Agricultural Surveys," *Journal of the American Statistical Association*, Vol. 87, No. 420 (Dec., 1992), pp. 924-932.

7. L. Cobb, P. Keppstein, and N. H. Chen, "Estimation and Moment Recursion Relations for Multimodal Distributions of the Exponential Family," *Journal of the American Statistical Association*, Vol 78, No. 381 (Mar., 1983), pp. 124-130.
8. W. J. Conover (1999), *Practicing Non-Parametric Statistics, Third Edition* John Wiley & Sons, Inc., New York, New York.
9. B. R. Crain, "Estimation of Distributions Using Orthogonal Expansions," *The Annals of Statistics*, vol. 2, no. 3 (May, 1974), p. 454-463.
10. B. R. Crain, "More on Estimation of Distributions Using Orthogonal Expansions," *Journal of the American Statistical Association*, vol. 71, no. 355 (Sep., 1976), p. 741-745.
11. N. Cressie, *Statistics for Spatial Data (Revised Edition)*, John Wiley & Sons, Inc., New York, 1993.
12. P. Davis, *Data Description and Presentation*, Oxford University Press, Ely House, London, 1974.
13. D. Ebdon (1985), *Statistics in Geography, Second Edition*, Basil Blackwell, New York, NY.
14. J. Fagan, "Generating Implied Ratio Edits," unpublished Census manuscript. October 1999.
15. G. D. Faulkenberry and A. Garoui, "Estimating a Population Total Using an Area Frame," *Journal of the American Statistical Association*, Vol. 86, No. 414 (Jun., 1991), pp. 445-449.
16. I. P. Fellegi and D. Holt, "A Systematic Approach to Automatic Edit and Imputation," *The Journal of the American Statistical Association*, No. 71, 1976.
17. L. M. Friedman, C. D. Furberg and D. L. DeMets (1996), *Fundamentals of Clinical Trials, Third Edition*, Mosby-Year Book, Inc., St. Louis, Missouri.
18. J. Gong, "Clarifying the Standard Deviation Ellipse," *Geographical Analysis*, Vol. 34, No. 2 (April 2002), pp. 155-167.
19. M. Garcia and R. Goodwin, "Developing SAS Software for Generating a Complete Set of Ratio Edits," *Statistical Research Division, US Census Bureau, Research Report Series*, Statistics # 2002-06, November 8, 2002.
20. R. Goodwin, M. Garcia and K. Thompson, "The GenBounds Software for Generating a Complete Set of Ratio Edits: %Implied User's Guide," *Statistical Research Division Report SS-Statistics # 2002-01*.

21. B. Greenberg and T. Petkunas, "SPEER (Structured Program for Economic Editing and Referrals)," *Proceedings of the Section on Survey Research Methods*, American Statistical Association, 1990.
22. S. Gregory, *Statistical Methods and the Geographer, Second Edition*, Longmans, Green, and Co. Ltd, London and Harlow, 1969.
23. S. I. Grossman (1984), *Elementary Linear Algebra*, Wadsworth Publishing Company, Belmont, California.
24. J. Hajek and Z. Sidak (1967), *Theory of Rank Tests*, Academic Press, New York, New York.
25. J. Hajek, Z. Sidak and P. K. Sen (1999), *Theory of Rank Tests*, Academic Press, San Diego, California.
26. Hammond, *Student Notebook Atlas*, Hammond World Atlas Corporation, 2010.
27. F. S. Hillier and G. J. Lieberman (1990), *Introduction to Operations Research*, McGraw-Hill Publishing Company, New York, New York.
28. R. V. Hogg and A. T. Craig (1995), *Introduction to Mathematical Statistics, Fifth Edition* Prentice Hall, Upper Saddle River, New Jersey.
29. R. V. Hogg and E. A. Tanis (1993), *Probability and Statistical Inference*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
30. M. Hollander and D. A. Wolfe (1999), *Nonparametric Statistical Methods, Second Edition*, John Wiley & Sons, Inc., New York, New York.
31. R. J. Johnston, *Multivariate Statistical Analysis in Geography*, Longman Group Limited, Longman, Inc., New York, 1978.
32. B. W. Kelly, "Probability Sampling in Collecting Farm Data," *Journal of Farm Economics*, Vol. 45, No. 5, Dec. 1963, pp. 1515-1520.
33. R. Khattree and D. N. Naik, *Applied Multivariate Statistics with SAS Software*, SAS Institute, Inc., Cary, North Carolina, 1995.
34. A. R. Kokan and S. Khan, "Optimum Allocation in Multivariate Surveys: An Analytical Solution," *Journal of the Royal Statistical Society, Series B (Methodological)*, Vol. 29, No. 1, 1967, pp. 115-125.
35. E.T. Lee, *Statistical Methods for Survival Data Analysis, Second Edition*, John Wiley & Sons, Inc., New York, 1992.
36. J. Lee and D. W. S. Wong, *Statistical Analysis with Arc View GIS*, John Wiley & Sons, Inc., New York, 2001.

37. E. L. Lehmann and H. J. M. D'Abrera (1975), *Nonparametrics, Statistical Methods Based on Ranks*, Holden-Day, Inc., Oakland, California.
38. P. Leong and S. Carlile, "Methods for Spherical Data Analysis and Visualization," *Journal of Neuroscience Methods*, vol. 80, 1998, pp. 191200.
39. P. A. W. Lewis (1966), "A Computer Program for the Statistical Analysis of Series of Events," *IBM Systems Journal*, vol. 5, no. 4.
40. D. W. Lefever, "Measuring Geographic Concentration by Means of the Standard Deviation Ellipse," *The American Journal of Sociology*, Vol. 32, No. 1. Jul., 1926, pp. 88-94.
41. R. V. Lenth, "Some Practical Guidelines for Effective Sample Size Determination," *The American Statistician*, Vol. 55, No. 3 (Aug., 2001), pp. 187-193.
42. L. Magee, "Improving Survey-Weighted Least Squares Regression," *Journal of the Royal Statistical Society, Series B (Statistical Methodology)*, Blackwell Publishing for the Royal Statistical Society, Vol. 60, No. 1, 1998, pp 115-126.
43. K. V. Mardia, "Statistics of Directional Data," *Journal of the Royal Statistical Society, Series B (Methodological)*, Vol. 37, No. 3 (1975), p. 349-393.
44. mardia2
45. K. V. Mardia, *Statistics of Directional Data*, Academic Press, New York, New York, 1972, pp. 52-58.
46. K. V. Mardia and P. E. Jupp, *Directional Statistics*, John Wiley & Sons, West Sussex, England, 2000.
47. A. Matei and Y. Tille, "Maximal and Minimal Sample Co-Ordination," *Sankhya: The Indian Journal of Statistics (2003)*, Vol. 67, No. 3 (Aug., 2005), pp. 590-612.
48. A. M. Mood (1954), "On the Asymptotic Efficiency of Certain Nonparametric Two-Sample Tests," *The Annals of Mathematical Statistics*, vol. 25, no. 3, p. 514-522.
49. L. E. Moses (1963), "Rank Tests for Dispersion," *The Annals of Mathematical Statistics*, vol 34, no. 3, , p. 973-983.
50. R. G. Petersen, *Design and Analysis of Experiments* Marcel Dekker, Inc., New York, New York, 1985.
51. P. W. Porter, "What is the Point of Minimum Aggregate Travel?," *Annals of the Association of American Geographers*, Vol. 53, No. 2, Taylor & Francis, Ltd. on behalf of the Association of American Geographers, Jun., 1963, pp. 224-232.

52. N. D. Pratt, A. C. Bird, J. C. Taylor, and R. C. Carter, "Estimating Areas of Land Under Small-Scale Irrigation Using Satellite Imagery and Ground Data for a Study in N.E. Nigeria," *The Geographical Journal*, Vol. 163, No. 1 (Mar., 1997), pp. 65-77.
53. M. L. Puri (1964), "Asymptotic Efficiency of a Class of c -sample Tests," *The Annals of Mathematical Statistics*, vol 35, no. 1, p. 102 - 121.
54. V. F. Rickey and P. M. Tuchinsky, "An Application of Geography to Mathematics: History of the Integral of the Secant," *Mathematics Magazine*, Vol. 53, No. 3 (May, 1980), pp. 162-166.
55. D. B. Rubin, "Composite Points in Weighted Least Squares Regressions," *Technometrics*, Vol. 22, No. 3, Aug. 1980, pp. 343-348.
56. C. E. Sarndal, B. Swensson and J. Wretman (1999), *Model Assisted Survey Sampling*, Springer-Verlag New York, Inc., New York, New York.
57. B. Shelby, "Girdle Distributions on the Sphere," *Biometrika Trust*, Vol. 51, No. 3/4, Dec. 1964, pp. 381-392.
58. F. Solomon (1987), *Probability and Stochastic Processes*, Prentice-Hall, Inc., Englewood Cliffs, New Jersey.
59. D. J. Steffensmeier, E. A. Allan, M. D. Harer and C. Streifel, "Age and the Distribution of Crime," *The American Journal of Sociology*, Vol. 94, No. 4 (Jan., 1989), pp. 803-831.
60. K. J. Thompson and R. Sigman, "Statistical Methods for Developing Ratio Edit Tolerances for Economic Data," *Journal of Official Statistics*, 2000.
61. G. B. Thomas and R. L. Finney, *Calculus and Analytic Geometry, Fifth Edition*, Addison-Wesley Publishing Company, Reading Massachusetts, 1979.
62. W. Winkler and L. Draper, "The SPEER Edit System," *Proceedings of the Conference of European Statisticians*, Section on Statistical Editing, UNECE, 1997.
63. C. A. Wolf and D. A. Sumner, "Are Farm Size Distributions Bimodal? Evidence from Kernel Density Estimates of Dairy Farm Size Distributions," *American Journal of Agricultural Economics*, vol. 83, no. 1 (Feb., 2001), p. 77-88.
64. A. Wood, "A Bimodal Distribution on the Sphere," *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, Vol. 31, No. 1 (1982), pp. 52-58.
65. R. F. Woolson and W. R. Clarke, "Analysis of Categorical Incomplete Longitudinal Data," *Journal of the Royal Statistical Society, Series A (General)*,

- Blackwell Publishing for the Royal Statistical Society, Vol. 147, No. 1, 1984, pp 87-99.
66. J. Yu, Z. Yang and X. Zhang (2002), "A Class of Nonlinear Stochastic Volatility Models and its Implications on Pricing Currency Options," www.buseco.monash.edu.au/depts/ebs/pubs/wpapers/2002/wp17-02.pdf.
 67. R. S. Yuill, "The Standard Deviation Ellipse: An Updated Tool for Spatial Description," *Geografiska Annaler, Series B, Human Geography*, Vol 53, No. 1, Blackwell Publishing on behalf of the Swedish Society for Anthropology and Geography, 1971, pp. 28-39.

Index

Arctan function, 49
R, 120
Developer, 1
Dim statement, 3

weighted mean center, 37

Active.Cells, 9
altitude, 37
angle of rotation, 44, 49, 54, 68, 69, 79, 98, 147, 159
Angle_of_Rotation(), 149
anti-mode, 118, 128
approximate formula, 77
Arc GIS, 35
area, xxv, 52, 69, 72, 73, 98, 99, 147, 152, 153, 159, 164
area frame, xxiii
area frame survey, xxiii
area preserving plot, 120
Area(), 149
area, check, 99
area, total, 68
array, 4
arrays, 4
asymptotically, 130

average, 8, 120
averages, 29
axes lengths, 68, 98
axes relationship, 79
axis, 19
axis length, semi-major, 43
axis length, semi-minor, 43
axis of rotation, 20, 152, 158, 164
axis, semi-major, 14
axis, semi-minor, 14

bar graph, 73
bars, 18
beginning row, 9
bimodal, 128
browse, 16, 18

Calculate_B(), 122, 135
Calculate_T(), 135
Cartesian coordinate system, 42, 43
cell, 7
cell contents, 7
cell, properties, 7
cental point, xxv
center of gravity, xxiv, 29, 43, 47, 73, 74, 105, 107

- center of system, xxv
- chi-square distribution, 130
- circle, xxv, 45, 52
- circular coordinate system, 86, 150
- circular coordinates, 89
- circular data, 43, 120
- closed form, 97
- clustered, 122
- clustered pattern, 128
- column chart, 18
- column names, 18
- columns, 8
- common syntax, 8
- concentration, 52
- concentration of the system, xxv
- condition variable, 7
- continental U.S., 73
- conventions, 8
- convergence, 86, 105
- convergence analysis, 104
- coordinate pairs, 48
- coordinates, 13, 15, 36
- copying and pasting, 13
- correlation, 118
- correlation coefficient, 45
- county, 36
- county level, 38
- county-level, 38
- credible parameter estimates, 105
- crime data, 73, 74
- Crimestat, 35
- cross hairs, 21
- cubic equation, 131
- cumulative distribution function, xxvi, 60, 61, 86, 92, 93
- cumulative distribution function, joint, 64
- currency, 4

- data collection, xxiv
- Data mapping wizard, 16, 18
- data transformation, xxvi, 45
- data types, 4
- Data wizard, 16
- data, character, 8
- data, graph it, 16
- data, numeric, 8
- data, scale, 18
- data, shape, 117, 144
- data, transform, 59, 62
- data, transforming, 59
- date, 3, 4
- decimal degrees, 36
- default names, 8
- degrees, 120

- derivatives, 24, 86
- determinant, 130, 131, 139
- Developer ribbon, 9, 148, 156
- development environment, 21
- deviation, xxiv
- deviations, 68, 98
- diagonal elements, 130, 141
- direction, xxv
- directional data, 119
- directional statistics, 118–120
- directional, data, 122, 141
- dispersed pattern, 128
- dispersion, 45
- disproportional weight, 139
- distance, 42
- distribution, 59
- distribution of the data, 73
- distribution of the points, xxv
- distribution, sample, 108
- distribution, underlying, 108
- double, 3
- draw ellipses, 14
- drawing lines, 13

- Earth, 13, 15, 16, 19, 43
- eccentricity, xxv, 52, 68, 69, 98, 147, 153, 159, 164
- eigenvalue, 130
- eigenvalues, 128, 130, 135, 143, 146
- eigenvectors, 130, 135, 143, 146
- ellipse, xxv, 52, 68, 69
- ellipse, center, 43
- ellipse, draw, 19, 44
- ellipse, multi-modal, 118
- ellipse, rotate, 20, 44
- elliptic pattern, 128
- elliptical formula, 26
- ELSEIF, 6
- ending row, 9
- Equator, 119
- error messages, reasons, 105
- error term, 147
- error terms, 51, 159
- estimates, 162
- estimator, 29, 49
- Excel, 3
- Excel 2010, VBA, 36
- Excel workbook, 16
- exp, 9
- EXPclean_data(), 157
- EXPdeviations(), 157
- EXPdistributions(), 157
- EXPellipse(), 158
- exponential deviational ellipse, 72–74

- Exponential distribution, 24
- exponential distribution, 62, 65
- exponential distribution function, 59, 62
- exponential ellipse, 65, 72–74, 77–80, 84, 87, 108
- exponential ellipse code, 156
- exponential mean center, 65
- exponential model, 49
- exponentially distributed, 61, 63
- exponentially distributed, weighted, deviational ellipse, 65
- EXPparameters(), 157
- EXPstats(), 157
- EXPthetas(), 157
- extreme values, 80

- first derivative, 84, 87
- font, 7
- font size, 7
- FOR-NEXT, 6, 7
- format shape, 20
- formulas, 8
- FORTRAN, 35
- fx(Lambda), 136

- Gamma1, 162
- Gamma2, 162
- Gauss(Lambda), 135
- Gaussian elimination, 131, 134
- general linear means model, 49, 50, 65
- geo-code, 13, 35
- geo-coding, 43
- geo-statistics, 45
- geographic data, 43
- girdle, 128, 130
- girdle distribution, 142
- global variables, 8
- Google Earth, 13, 35, 36
- graph, 63
- graph paper, 59
- graphing, 13
- graphs, 108
- gx(Lambda), 136

- histogram, 118, 139, 144
- hoop coordinate system, 42

- IF-ELSE-IF, 5
- IF-THEN-ELSE, 5
- image classification, xxiii
- imaginary axis, 44
- Import data wizard, 15, 16, 18
- inital values, 85
- initial guess, 163
- initial guesses, 161
- initial value, 105, 134
- initial values, 86, 88
- initialization, 86, 88
- input parameters, Weibull, 104
- integrals, 25
- iteger numbers, 3
- iteration, 24, 85, 92
- Iterations, 162
- iterations, 86, 105, 163
- iterations, number of, 86

- Jacobian matrix, 85, 88
- joint density function, 91
- joint distribution, 61, 63, 91, 92, 159, 163, 164
- joint likelihood function, 63
- joint probability density function, 63, 92
- June Area Survey, 36, 37

- kilometers, 19
- kurtosis, 119

- label text, 18
- land segments, 73
- largest weights, 73
- latitude, xxiv, 9, 24, 43, 59, 62, 63, 65, 77, 83, 92, 103, 105, 119, 141, 159
- latitude observation, 50
- least squares, 65, 118
- length, 13, 19
- likelihood function, 61, 62, 84, 104, 108, 153, 159, 164
- likelihood function, Weibull, 84, 87
- likelihood functions, 161
- linear means model, 97
- linear regression, 118
- linear regression model, 49, 59, 65
- linear relation, 61, 63
- linear relationship, 59, 61, 62
- local variables, 9
- location parameter, 59, 65
- location problem, xxiv
- log, 9
- Log10(X), 162
- logical, 4
- long integer, 4
- longitude, xxiv, 9, 24, 43, 51, 62, 63, 65, 77, 87, 92, 103, 105, 119, 142, 159
- longitude data, 86
- longitude observations, 86
- loop, For-Next, 9
- loop, While-Wend, 9

- main.Modal(), 135
- major axis, 68, 69, 98, 152

- major axis length, 69
- major axis of rotation, 158, 164
- map, 43, 44
- Map Point, 13, 18, 20
- Map Soft, 35
- matrix B , 128, 141, 144
- matrix T , 130, 135, 141, 142, 144
- maximum likelihood estimates, 92
- maximum likelihood estimator, 59, 61–63, 84, 87, 91, 92
- maximum likelihood theory, 108
- mean \bar{X} , 124
- mean \bar{Y} , 124
- mean \bar{Z} , 124
- mean center, xxiv, 13, 19, 45, 46, 49, 50, 79, 86, 89, 98, 106, 117, 159
- mean center coordinates, 159
- mean center, transform, 153
- mean direction, 120
- mean direction of cosines, 124
- mean joint distribution, 65
- mean joint distribution estimate, 97
- mean latitude, 120, 121
- mean longitude, 120
- measure lines, 44
- menu, Insert, 12
- merged cells, 39
- miles, 19
- minor axis, 69, 98, 152
- minor axis length, 69
- minor axis of rotation, 158, 164
- mod, 8
- modal_stats(), 122, 141
- mode, 118, 128
- modes, 144
- module, 4
- Module 1, 149
- Module 2, 157
- Module 3, 161, 166
- Module 4, 122, 134
- Module 5, 134, 136
- modules, 9
- MS Word, 20
- multimodal statistics, 118

- Newton algorithm, 131
- Newton's method, 131
- Newton() subroutine, 139
- Newton(Lambda), 136
- Newton-Raphson method, 85, 88
- no fill, 20
- non-uniform distribution, xxv
- normal distribution, 65
- Northern Hemisphere, 119

- null hypothesis, 144, 146
- number of iterations, 86
- numerical algorithms, 85

- observation, 62
- observed value, 64, 84
- one pole, 42
- original data, 97
- output statistics, 124
- oval, 20

- parameter, 24, 61
- parameter estimate, 84, 86, 87, 89
- parameter estimates, 105
- parameter estimation, 86
- parameter estimator, 62
- parameters, 84, 108, 136, 162, 163
- percentage, 119
- pie chart, 18
- plane, 128
- plot, 14, 61, 63
- plotting, 13
- plotting the transformed data, 59
- point estimates, 163
- point pattern, 45
- points, 43
- points, distribution, 141
- polar coordinate, 120
- polar coordinate system, 43
- positive definite, 77, 78
- predicted values, 51
- preferred direction, no, 122
- Prime Meridian, 119
- probability density function, 60, 63, 83, 86, 91
- program code, 8
- properties, 7
- pushpins, 16
- Pythagoras' theorem, 45

- quadratic equation, 26
- quadratic equation, roots, 26, 49
- quantile, 119
- quantity, 38

- radians, 120
- random, 139
- random sample, 84, 87
- random variable, xxiv, 9, 24, 35, 43, 50, 51, 61, 62, 64, 65, 87, 93, 142
- random variable w_i , 84
- range, 89
- real number, 4
- real numbers, 3
- record, 4

- records, 4
- regional data, 39
- regional level, 37, 39
- regression, 65
- regression estimator, 45
- regression line, xxvi
- regression model, 159
- relationship, 65, 77, 78
- relative concentration, xxv
- resultant length, 121, 122, 124
- roots, 130
- rotate, 14
- rotational symmetry, 130
- row reduction, 134
- ruler, 44
- ruler, anchor, 19
- RunWeibull(), 161

- sample data, 60
- sample size, 24, 120
- sample space, 62
- satellite images, 36
- saving work, 13
- scale parameter, 24, 60, 65, 163
- scatter diagram, 119
- SDE code, 148
- SDE statistics, 150
- SDE(), 149
- SDE.Likelihood(), 149
- SDEdeltas(), 149
- Secant algorithm, 92, 161, 163
- Secant estimation, 166
- Secant iterations, 97
- Secant method, 85, 86, 88
- Secant programs, 103
- Secant1(), 161, 162
- Secant2(), 162, 163
- segment, xxiv
- Selection.Rows.Count, 8
- semi-major axes, 77, 79
- semi-major axis, 44, 49, 76
- semi-major axis length, 51, 54, 78, 79, 98, 99, 147, 152, 159, 164
- semi-minor axes, 79
- semi-minor axis, 44, 49, 79
- semi-minor axis length, 52, 54, 79, 98, 99, 147, 153, 159, 164
- series column chart, 18
- Set_weights(), 149, 157
- shaded circles, 16
- shape parameter, 59, 163
- shapes, 20
- SinCosS(), 122
- single pole, 42

- sized circles, 16
- sized pie chart, 18
- skewness, 119
- smaller weights, 73
- Southern Hemisphere, 119
- spatial analysis, 117
- spatial data, 37, 120
- spatial statistics, xxiv
- specialty software, 44
- spherical distribution, 128, 134
- spherical statistics, 42, 43, 117, 124, 134
- spherical variance, 121, 124
- spreadsheet, 39
- standard deviation, xxiv, 117, 152, 158, 159, 164
- standard deviational ellipse, xxiv, 45, 51, 55, 72–74, 76–80, 92, 105, 107, 148
- standard deviational ellipse formula, 64
- standard deviations, 68, 73, 98
- standard distance, 45
- standard ellipse, 77
- standard error terms, 49
- state name, 36
- STATS, 9, 150, 158, 164
- stereographic-net, 120
- straight line, 60, 97
- string characters, 3
- subroutine, 4
- sufficient statistics, 84, 87
- sum, 8
- sums of squares, 65, 68, 77, 79, 98, 108
- survey, 37
- survey area, 19, 43
- survey data, 13

- termination condition, 163
- test statistic, 141, 143, 145
- THEN, 6
- time, 3
- trace, 130
- transformation, weighted latitude, 60
- transformed data, 61, 65, 97
- transformed variable, 61
- trend of the system, xxv
- triadic, 139
- trial and error, 86, 134
- two dimensions, 42
- TYPE-END-TYPE, 4

- uniform, 122, 128, 130, 146
- uniform distribution, xxiv, xxv
- units, 19
- unweighted exponential deviational ellipse, 159
- unweighted, calculations, 117

- Variable, 162
- VBA code, 86
- VBA Development environment, 1
- VBA Editor, 12, 150
- VBA editor, 4
- VBA environment, 8, 135, 148
- VBA functions, 9
- VBA numeric calculations, 9
- VBA program, 155
- VBA programming environment, 122
- VBA statements, 8
- VBA, SDE(), 147
- VBA, Set_weights(), 147
- vector \underline{b}_i , 128
- vectors, 42
- violent crime, 55, 142
- violent crime, survey, 28
- visible, 4
- Visual Basic button, 12, 157
- Visual Basic Editor, 147, 155
- Visual Basic for Applications (VBA), 36
- Visual Basic ribbon, 149

- Weibull, 166
- Weibull deviational ellipse, 97, 105, 107
- Weibull distribution, 24, 86, 97
- Weibull distribution, sufficient statistics, 84, 87
- Weibull ellipse, 84, 87, 105–108, 161
- Weibull models, weighted, 91
- Weibull(), 162, 164
- weight, 24, 29, 50, 142, 147
- weight column, 159
- weight, disproportionately, 142
- weighted data, 28, 73
- weighted ellipse, 50
- weighted exponential deviational ellipse, 73, 76, 77, 79
- weighted exponential distribution, 59
- weighted exponential model, 63
- weighted exponential standard deviational ellipse, 68
- weighted joint distribution, 78
- weighted latitude, 60, 61, 65
- weighted longitude, 63, 65, 78
- weighted mean, 76, 77
- weighted mean center, xxv, 45, 46, 51, 89, 147, 152, 158, 159, 164
- weighted mean estimate, 65
- weighted mean response, 50
- weighted means estimate, 97
- weighted points, 68
- weighted standard deviational ellipse, xxv, 37, 46, 49, 77, 79, 147
- weighted sum of squares, 51
- weighted Weibull models, 91
- weighted, Weibull ellipse, 98
- weighted, Weibull standard deviational ellipse, 97
- weights, 93
- WHILE-WEND, 6, 7
- whole numbers, 3
- WITH-END-WITH, 7
- worksheet, 8
- worksheetfunction.pi(), 9
- Worksheets().Cells().Value, 9
- Wulff-net, 120