Random Variables, Their Properties, and Deviational Ellipses: In Map Point and Excel, v 4.3

Goodwin, Roger L

1 January 2015
RANDOM VARIABLES, THEIR PROPERTIES, AND DEVIATIONAL ELLIPSES
RANDOM VARIABLES, THEIR PROPERTIES, AND DEVIATIONAL ELLIPSES
In Map Point and Excel, v 4.3

Roger L. Goodwin
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 Deviational Ellipse ................................. 45  
6 The Exponential Ellipse ....................................... 65  
7 The Weibull Ellipse ............................................ 89  
8 Spherical Statistics ............................................. 119
2.1 User 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 Standard Deviational Ellipse 47
  5.3.1 Weighted Mean Center 48
5.4 Ellipse Properties 49
5.5 Kentucky Example 51
5.6 Crime Example 53
5.7 GDP Example 53
5.8 Exercises 54
5.9 VBA Code 56

6 The Exponential Ellipse 65
  6.1 Mean Latitude 65
  6.2 Mean Longitude 68
  6.3 Joint Distribution 69
  6.4 Exponential Ellipse 71
    6.4.1 Mean Center 71
    6.4.2 Ellipse 71
  6.5 Axis Rotation 72
  6.6 Ellipse Properties 72
  6.7 Kentucky Example 73
  6.8 Crime Example 75
  6.9 GDP Example 77
  6.10 Comparison to SDE 78
  6.11 Exercises 82
  6.12 VBA Code 85

7 The Weibull Ellipse 89
  7.1 Latitude Mean Center 89
  7.2 Longitude Mean Center 92
  7.3 Joint Distribution 95
  7.4 Weibull Ellipse 98
    7.4.1 Ellipse Angle 98
  7.5 Ellipse Properties 99
  7.6 Kentucky Example 100
  7.7 Crime Example 103
  7.8 GDP Example 103
  7.9 Axes Length Comparison 105
  7.10 Sample Distribution Fitting 105
  7.11 Exercises 107
  7.12 VBA Code 109

8 Spherical Statistics 119
  8.1 Introduction 119
  8.2 Concepts 121
  8.3 X-Axis Mean Center and Resultant Length 122
  8.4 Y and Z-Axis Mean Centers 123
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>This figure shows Excel without the Developer menu.</td>
<td>2</td>
</tr>
<tr>
<td>1.2</td>
<td>This figure shows Excel Options dialog box. Excel uses the dialog box for adding the Developer menu for VBA programming.</td>
<td>2</td>
</tr>
<tr>
<td>1.3</td>
<td>This figure shows Excel with the Developer menu at the top of the screen.</td>
<td>3</td>
</tr>
<tr>
<td>1.4</td>
<td>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.</td>
<td>10</td>
</tr>
<tr>
<td>1.5</td>
<td>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.</td>
<td>11</td>
</tr>
<tr>
<td>2.1</td>
<td>This figure shows the Map Point dialog box for finding a point on the Earth using the latitude and longitude.</td>
<td>14</td>
</tr>
</tbody>
</table>
LIST OF FIGURES

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

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

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

2.5 This figure shows the Map Point Data mapping wizard.

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

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

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

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

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

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.

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.

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

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.

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.

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

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

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.

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.

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.

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.

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.

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.

6.5 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.

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.
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.

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

8.2 This figure shows the diagram of a triangle with the quantities for the longitude $\theta$, $n_0$, and the resultant length $R$.

8.3 This figure shows the histogram for the Kentucky 2003 data.

8.4 This figure shows the Kentucky 2003 converted data.

8.5 This figure shows the histogram for the Violent Crime 2008 data. Three possible modes appear at California, Texas, and Florida.

8.6 This figure shows the 2008 US Crime converted data.

8.7 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%.

8.8 This figure shows the 2008 Gross Domestic Product converted data.

A.1 In Excel, double click on the module named Ch4StandardDeviationalEllipse on the left-hand pane circled in red to display the VBA code for the Standard Deviational Ellipse. The subroutine called `Standard_Dev_Ellipse()` runs the programs to calculate the statistics in Chapter 5.

A.2 This figure shows the VBA environment for the module named Ch4StandardDeviationalEllipse. It also highlights the driver subroutine `Standard_Dev_Ellipse()`.

A.3 This figure shows the STATS worksheet and the results from the `Standard_Dev_Ellipse()` subroutine.

B.1 This figure shows the VBA environment for the module Ch5ExponentialDeviationalEllipse. It also highlights the driver subroutine `Exponential_Dev_Ellipse()`.

B.2 This figure shows the STATS worksheet and the results from the `Exponential_Dev_Ellipse()` subroutine.
C.1 This figure shows the VBA environment for the module Ch6 Weibull Deviational Ellipse. It also highlights the driver subroutine Weibull_Dev_Ellipse(). 158

C.2 This figure shows the STATS worksheet and the results from the Weibull() subroutine. 160

D.1 This figure shows the VBA environment for the module Ch7NewtonGaussRoutines. It also highlights the driver subroutine NewtonGauss(). 166
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Results on Corn acreage in Kentucky</td>
<td>51</td>
</tr>
<tr>
<td>5.2</td>
<td>Results on Corn acreage in Kentucky</td>
<td>52</td>
</tr>
<tr>
<td>5.3</td>
<td>Results on Violent Crime in the U.S.</td>
<td>53</td>
</tr>
<tr>
<td>5.4</td>
<td>Results on Violent Crime in the U.S.</td>
<td>53</td>
</tr>
<tr>
<td>5.5</td>
<td>Results on Gross Domestic Product of OECD Countries</td>
<td>54</td>
</tr>
<tr>
<td>5.6</td>
<td>Results on Gross Domestic Product of OECD Countries</td>
<td>55</td>
</tr>
<tr>
<td>6.1</td>
<td>Results for the Exponential Model for Kentucky</td>
<td>74</td>
</tr>
<tr>
<td>6.2</td>
<td>Results for the Exponential Model for Kentucky</td>
<td>74</td>
</tr>
<tr>
<td>6.3</td>
<td>Results for the Exponential Model on Violent Crime in the U.S.</td>
<td>75</td>
</tr>
<tr>
<td>6.4</td>
<td>Results for the Exponential Model on Violent Crime in the U.S.</td>
<td>76</td>
</tr>
<tr>
<td>6.5</td>
<td>Results for the Exponential Model on Gross Domestic Product</td>
<td>77</td>
</tr>
<tr>
<td>6.6</td>
<td>Results for the Exponential Model on Gross Domestic Product</td>
<td>77</td>
</tr>
<tr>
<td>6.7</td>
<td>Relationship of angles of rotation for large $x_i's.$</td>
<td>82</td>
</tr>
</tbody>
</table>

xix
<table>
<thead>
<tr>
<th>Table</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.8</td>
<td>Relationship of angles of rotation for small $x_i's$.</td>
<td>82</td>
</tr>
<tr>
<td>7.1</td>
<td>Results for the Weibull Model for Kentucky</td>
<td>101</td>
</tr>
<tr>
<td>7.2</td>
<td>Results for the Weibull Model for Kentucky</td>
<td>101</td>
</tr>
<tr>
<td>7.3</td>
<td>Results for the Weibull Model on Violent Crime in the U.S.</td>
<td>103</td>
</tr>
<tr>
<td>7.4</td>
<td>Results for the Weibull Model on Violent Crime in the U.S.</td>
<td>103</td>
</tr>
<tr>
<td>7.5</td>
<td>Results for the Weibull Model on Gross Domestic Product</td>
<td>104</td>
</tr>
<tr>
<td>7.6</td>
<td>Results for the Weibull Model on Gross Domestic Product</td>
<td>104</td>
</tr>
<tr>
<td>8.1</td>
<td>Interpretation of the Eigenvalues</td>
<td>126</td>
</tr>
<tr>
<td>8.2</td>
<td>Eigenvalues and Eigenvectors for the 2003 Kentucky Data</td>
<td>129</td>
</tr>
<tr>
<td>8.3</td>
<td>Results for the Spherical Model for Kentucky</td>
<td>130</td>
</tr>
<tr>
<td>8.4</td>
<td>Eigenvalues and Eigenvectors for the 2008 Violent Crime Data</td>
<td>132</td>
</tr>
<tr>
<td>8.5</td>
<td>Results for the Spherical Model for 2008 Crime</td>
<td>132</td>
</tr>
<tr>
<td>8.6</td>
<td>Eigenvalues and Eigenvectors for the 2008 Gross Domestic Product</td>
<td>134</td>
</tr>
<tr>
<td>8.7</td>
<td>Results for the Spherical Model for GPD</td>
<td>135</td>
</tr>
</tbody>
</table>
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.
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 ob-
tain 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) regres-
sion 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)] con-
sider 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 \( \pi_k \)
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 associat-
ing random variables with the latitude and longitude (i.e. the placement on a map).

[Edon, (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
[Edon, (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 $\theta_m$ 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 Furtey 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.
<table>
<thead>
<tr>
<th>ACRONYMS</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GDP</td>
<td>Gross Domestic Product</td>
</tr>
<tr>
<td>OECD</td>
<td>Organization for Economic Cooperation and Development</td>
</tr>
<tr>
<td>SDE</td>
<td>Standard Deviational Ellipse</td>
</tr>
<tr>
<td>VBA</td>
<td>Visual Basic for Applications</td>
</tr>
</tbody>
</table>
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.
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.
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 dimensiona1 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:
### Type Customers

```
        Type Customers
            First_Name, Last_Name As String
            Address As String
            City As String
            State As String
            Zip_Code As Integer
            Phone As Integer
        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:

```
        List.Phone = 3223223
```

### 1.4 Branching

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

```
        2 If <condition> Then
            VBA statements
        Else
            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 by one **ELSE** statement in any **IF-END-IF** construct. An alternative to the above **IF-THEN-ELSE** construct is the **IF-ELSE-IF** construct.
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:

```vba
For <counter> = start To end
    VBA statements
Next <counter>
```

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

```vba
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.

```vba
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.

```vba
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:

```vba
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 $\pi$.

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.
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.
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 User 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. Notes on entering data manually appear in Section 2.2. 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...
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.
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.
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 2.8  This figure shows the Map Point choices for scaling the data and change the label text.
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.
Simulation of a Damped Pendulum

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 drawback. 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.
3.1 Notation

We discuss some common notation used throughout this book here.
### Notation and Meaning

<table>
<thead>
<tr>
<th>Notation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>Subscript denoting latitude.</td>
</tr>
<tr>
<td>$y$</td>
<td>Subscript denoting longitude.</td>
</tr>
<tr>
<td>$t$</td>
<td>A dummy variable used in integration. Usually denotes the latitude.</td>
</tr>
<tr>
<td>$u$</td>
<td>A dummy variable used in integration. Usually denotes the longitude.</td>
</tr>
<tr>
<td>$w$</td>
<td>Subscript denoting the weight variable.</td>
</tr>
<tr>
<td>$\gamma_x$</td>
<td>Shape parameter to the Weibull distribution for the latitude.</td>
</tr>
<tr>
<td>$\gamma_y$</td>
<td>Shape parameter to the Weibull distribution for the longitude.</td>
</tr>
<tr>
<td>$\lambda_x$</td>
<td>Scale parameter to the Exponential distribution and the Weibull distribution for the latitude. Generally, there is no implied equality.</td>
</tr>
<tr>
<td>$\lambda_y$</td>
<td>Scale parameter to the Exponential distribution and the Weibull distribution for the longitude. Generally, there is no implied equality.</td>
</tr>
<tr>
<td>$k$</td>
<td>The iteration number in the Secant algorithm.</td>
</tr>
<tr>
<td>$n$</td>
<td>Denotes the sample size.</td>
</tr>
<tr>
<td>$\phi_i$</td>
<td>Represents the observed values of the latitude in spherical statistics.</td>
</tr>
<tr>
<td>$\theta_i$</td>
<td>Represents the observed values of the longitude in spherical statistics.</td>
</tr>
<tr>
<td>$R$</td>
<td>Called the resultant length in spherical statistics.</td>
</tr>
<tr>
<td>$\hat{\theta}_0$, $\hat{m}_0$, $\hat{a}_0$</td>
<td>Called the mean direction of cosines in spherical statistics.</td>
</tr>
<tr>
<td>$S^*$</td>
<td>Denotes the spherical variance.</td>
</tr>
<tr>
<td>$T$</td>
<td>Denotes the matrix $T$. The diagonal elements sum to the sample size $n$.</td>
</tr>
<tr>
<td>$B$</td>
<td>Denotes the matrix $B$. The diagonal elements sum to $2n$.</td>
</tr>
</tbody>
</table>

The set $\{x_1, x_2, ..., x_n\}$ represents the observed values of the latitude. The set $\{y_1, y_2, ..., y_n\}$ represents the observed values of the longitude. The set $\{w_1, w_2, ..., w_n\}$ 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 \( f(w, \lambda, x) = w \frac{1}{\lambda} e^{-x} \lambda \) 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: \[ \int_0^a \ln n \, dx = x \ln n \bigg|_0^a = a \ln n. \]

Example 8: \[ \int \frac{1}{x} \, dx = \log x \]

Example 9: \[ \int e^{ax} \, dx = \frac{e^{ax}}{a} \]

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).

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

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:

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

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

$$\bar{x}' = \begin{cases} 
\bar{x}, & \text{if } \bar{x} \leq 90^\circ. \\
-\text{mod}(180, \bar{x}), & \text{if } \bar{x} > 90^\circ.
\end{cases}$$  \hfill (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 radio button **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:

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

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

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

Example 11: The coordinates of Canada are \((x_i, y_i) = (55° 56' 18.59 \text{ North}, 106° 20' 48.37 \text{ West})\). 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. Vise-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 use weighted data. Equation (3.6) shows the weighting scheme for observation \( i \) out of \( n \) observations.

\[
w_i' = \frac{w_i}{\sum_{i=1}^{n} w_i}
\]

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
        With ActiveSheet
            sum_weight = sum_weight + .Cells(i, 3).Value
        End With
    Next
    For i = 2 To n
        With ActiveSheet
            .Cells(i, 4).Value = .Cells(i, 3).Value / sum_weight
        End With
    Next
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 \texttt{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 "Activesheet."

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.

\[ f = \pi ab. \tag{3.7} \]

```
Function Area(a, b)
    'a= semi-major axis length
    'b= semi-minor axis length
    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 \texttt{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, $a \geq b$.

\[ e = \frac{\sqrt{a^2 - b^2}}{a} \tag{3.8} \]

```
Function Eccentricity(a, b)
    'a= semi-major axis length
    'b= semi-minor axis length
    If a >= b Then
        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 \texttt{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 \texttt{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 \( b \).

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

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.

```vbnet
Sub AxesLength(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)
    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))))
    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 \texttt{AxesLength()} 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 $\theta$ 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 $\theta$ 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`.

```vba
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)
    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
    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 \( \text{Log10}(X) \) returns the logarithm of the number \( X \) with the base 10.

\[
\begin{align*}
\text{Static Function} & \quad \text{Log10}(X) \\
1 & \quad \text{Log10} = \frac{\log(X)}{\log(10)} \\
\text{End Function}
\end{align*}
\]
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.

![Google Earth](image)

**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. Their website is http://www.nass.usda.gov/index.asp. 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 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 Earth coordinates for each county. Data consumers can easily download the crop data. Figure 4.2 shows the data downloaded from the NASS web.

---

1 The web site is www.earthgoogle.com. Decimal coordinates are preferred since we can more easily integrate them than degrees, minutes, and seconds.
Regional level observations have arrows identifying them. The yellow arrows point to the following formatting issues for us.

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.

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 Statistic’s website http://www.bjs.gov/, 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.

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.
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 are 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 \([-90^\circ, 90^\circ]\).
Figure 4.4 This figure shows the Excel spreadsheet of the Gross Domestic Product data. The spreadsheet has not been geo-coded.

<table>
<thead>
<tr>
<th>Latitude</th>
<th>( \text{mod}(x, 180) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>20</td>
</tr>
<tr>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>60</td>
<td>60</td>
</tr>
<tr>
<td>70</td>
<td>70</td>
</tr>
<tr>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>90</td>
<td>90</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>−10</td>
<td>170</td>
</tr>
<tr>
<td>−20</td>
<td>160</td>
</tr>
<tr>
<td>−30</td>
<td>140</td>
</tr>
<tr>
<td>−50</td>
<td>130</td>
</tr>
<tr>
<td>−60</td>
<td>120</td>
</tr>
<tr>
<td>−80</td>
<td>100</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>Longitude</th>
<th>mod(y, 360)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>80</td>
<td>80</td>
</tr>
<tr>
<td>120</td>
<td>120</td>
</tr>
<tr>
<td>160</td>
<td>160</td>
</tr>
<tr>
<td>180</td>
<td>180</td>
</tr>
<tr>
<td>-10</td>
<td>350</td>
</tr>
<tr>
<td>-40</td>
<td>320</td>
</tr>
<tr>
<td>-80</td>
<td>280</td>
</tr>
<tr>
<td>-120</td>
<td>240</td>
</tr>
<tr>
<td>-160</td>
<td>200</td>
</tr>
</tbody>
</table>

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 $r = \mathbf{i}x + \mathbf{j}y$, 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).
Exact formulas exist for converting from the polar coordinate system to the Cartesian coordinate system (and vise-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

\[
\left( \frac{1}{n} \sum_{i=1}^{n} (x_i - \bar{x})^2 + \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^2 \right)^{\frac{1}{2}}.
\]

The author objectively compares the standard distance of the dispersion of different point patterns. 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 \( \alpha \) appears in Equation (5.2).

\[
\hat{\alpha} = \bar{y} - \hat{\beta} \bar{x}.
\] (5.2)

The parameter estimator for \( \beta \) appears in Equation (5.3).

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

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

\[
\bar{x} = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \tag{5.4}
\]

\[
\bar{y} = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i} \tag{5.5}
\]

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

\[
\sum_{i=1}^{n} \epsilon_i^2 = \sum_{i=1}^{n} w_i (y_i - \hat{\alpha} - \hat{\beta} x_i)^2. \tag{5.6}
\]

Equation (5.7) shows the weighted likelihood function for the regression model. It has a chi-square distribution.
\[ L = \prod_{i=1}^{n} \left( \frac{1}{2\pi \epsilon^2} \right)^{\frac{1}{2}} \exp \left\{ -\frac{1}{2} \sum_{i=1}^{n} w_i(x_i - \hat{\alpha} - \hat{\beta} x_i)^2 \right\} \]  

(5.7)

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

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

(5.8)

Equation (5.8) is based on the Akaike information criterion. The entire equation is retained for comparison purposes in Section 7.10.

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 \(-180 \leq y_i \leq 180\).

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

(5.9)

Equation (5.9) is called the center of gravity.

5.3 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

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

\[ \frac{2 \sum_{i=1}^{n} x_i' y_i' w_i}{2 \sum_{i=1}^{n} x_i' y_i' w_i} \]  

(5.10)

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

\[ a = \sum_{i=1}^{n} x_i' y_i' w_i, \quad \text{and} \quad c = 1. \]
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).

\[ \delta_x = \sqrt{\frac{\sum_{i=1}^{n} (y'_i \sin \theta - x'_i \cos \theta)^2 w_i}{\sum_{i=1}^{n} w_i}} \]  
(5.11)

\[ \delta_y = \sqrt{\frac{\sum_{i=1}^{n} (y'_i \cos \theta - x'_i \sin \theta)^2 w_i}{\sum_{i=1}^{n} w_i}} \]  
(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 5.5 will show how Equation (5.10) can be extended to the exponential model.

### 5.3.1 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

\[
\text{model } x_i, y_i = \mu_x + \mu_y + \frac{\varepsilon_x}{\sqrt{w_i}} + \frac{\varepsilon_y}{\sqrt{w_i}}.
\]

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 \( -180 \leq y_i \leq 180 \); \( \mu_x \) represents the weighted mean response of the latitude responses; and \( \mu_y \) 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).

\[ \varepsilon_x = \sqrt{w_i} (x_i - \mu_x). \]

We square \( \varepsilon_x \) and sum to obtain the weighted sum of squares.

\[ \varepsilon_x^2 = \sum_{i=1}^{n} w_i (x_i - \mu_x)^2. \]
The $w_i'$s and $x_i'$s are known observations. We estimate $\mu_x$ using the $w_i'$s and $x_i'$s observations. We take the derivative of $e_x$ with respect to $\mu_x$; set it to zero; and solve for $\mu_x$.

\[
\frac{de_x}{d\mu_x} = -2 \sum_{i=1}^{n} w_i (x_i - \mu_x) = 0 \Rightarrow \\
\sum_{i=1}^{n} w_i x_i - \mu_x \sum_{i=1}^{n} w_i = 0 \Rightarrow \\
\sum_{i=1}^{n} w_i x_i = \mu_x \sum_{i=1}^{n} w_i \Rightarrow \\
\hat{\mu}_x = \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} = \bar{x}. \quad (5.13)
\]

A similar derivation holds for the longitude $e_y$ by substituting $y_i$ for $x_i$ and $\mu_y$ for $\mu_x$ above.

\[
\hat{\mu}_y = \frac{\sum_{i=1}^{n} w_i y_i}{\sum_{i=1}^{n} w_i} = \bar{y}. \quad (5.14)
\]

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

### 5.4 Ellipse Properties

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

\[
a_1^2 = \frac{1}{n} \sum_{i=1}^{n} y_i'^2 w_i 
= 1 - \frac{1}{n} \sum_{i=1}^{n} y_i'^2 w_i + \frac{2 (\sum_{i=1}^{n} x_i' y_i' w_i)^2}{\left(\sum_{i=1}^{n} x_i'^2 w_i - \sum_{i=1}^{n} y_i'^2 w_i\right)^2 + 4 \left(\sum_{i=1}^{n} x_i' y_i' w_i\right)^2} 
\]

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 (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.

\[
b_1^2 = \frac{1}{n} \sum_{i=1}^{n} x_i'^2 w_i = \quad (5.16)
\]
\[
\frac{1}{n} - \left( \sum_{i=1}^{n} x_i^2 w_i - \sum_{i=1}^{n} y_i^2 w_i \right) + \sqrt{\left( \sum_{i=1}^{n} x_i^2 w_i - \sum_{i=1}^{n} y_i^2 w_i \right)^2 + 4 \left( \sum_{i=1}^{n} x_i y_i w_i \right)^2} \]

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

\[ F = \frac{\pi}{n} \sum_{i=1}^{n} (x_i - x)\bar{x} w_i - \sum_{i=1}^{n} (x_i - x)(y_i - y) w_i. \]

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.

\[ e = \frac{\sqrt{a_1^2 - b_1^2}}{a_1} \]

\( 0 \leq e \leq 1 \) 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.

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:

\[ a_1^2 - \frac{1}{n} \sum_{i=1}^{n} y_i^2 w_i = \]

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

and

\[ b_1^2 - \frac{1}{n} \sum_{i=1}^{n} x_i^2 w_i = \]

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

\[
a_1^2 - \frac{1}{n} \sum_{i=1}^{n} x_i^2 w_i = \frac{1}{n} \left( \sum_{i=1}^{n} y_i^2 w_i - \sum_{i=1}^{n} x_i^2 w_i \right) + \frac{2 \left( \sum_{i=1}^{n} x_i y_i w_i \right)^2}{\left( \sum_{i=1}^{n} y_i^2 w_i - \sum_{i=1}^{n} x_i^2 w_i \right)^2 + 4 \left( \sum_{i=1}^{n} x_i y_i w_i \right)^2}
\]

and

\[
b_1^2 - \frac{1}{n} \sum_{i=1}^{n} y_i^2 w_i = \frac{1}{n} \left( \sum_{i=1}^{n} y_i^2 w_i - \sum_{i=1}^{n} x_i^2 w_i \right) + \frac{2 \left( \sum_{i=1}^{n} x_i y_i w_i \right)^2}{\left( \sum_{i=1}^{n} y_i^2 w_i - \sum_{i=1}^{n} x_i^2 w_i \right)^2 + 4 \left( \sum_{i=1}^{n} x_i y_i w_i \right)^2}
\]

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.

### 5.5 Kentucky Example

<table>
<thead>
<tr>
<th>Table 5.1</th>
<th>Results on Corn acreage in Kentucky</th>
</tr>
</thead>
<tbody>
<tr>
<td>2003</td>
<td></td>
</tr>
<tr>
<td>Center</td>
<td>(37.38297399, -86.8129514)</td>
</tr>
<tr>
<td>Axes</td>
<td>( a = 184.8734659, \ b = 52.23219209 )</td>
</tr>
<tr>
<td>Area</td>
<td>30,336.30686 sq. mi.</td>
</tr>
<tr>
<td>Standard Deviations</td>
<td>( \delta_x = 1.29151776, \ \delta_y = 1.625551351 )</td>
</tr>
<tr>
<td>Rotation</td>
<td>Y-Axis</td>
</tr>
<tr>
<td>Orientation</td>
<td>( \theta_x = 70.36257022^\circ, \ \theta_y = -19.63742978^\circ )</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.959258636</td>
</tr>
<tr>
<td>Concentration</td>
<td>55/90</td>
</tr>
</tbody>
</table>

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
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 5.2 Results on Corn acreage in Kentucky

<table>
<thead>
<tr>
<th></th>
<th>2004</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
<td>(37.40588805, -86.8066601)</td>
</tr>
<tr>
<td>Axes</td>
<td>$a = 184.632769, b = 53.02565716$</td>
</tr>
<tr>
<td>Area</td>
<td>30,757.05259 sq. mi.</td>
</tr>
<tr>
<td>Standard Deviations</td>
<td>$\delta_x = 1.28520402, \delta_y = 1.613981198$</td>
</tr>
<tr>
<td>Rotation</td>
<td>Y-Axis</td>
</tr>
<tr>
<td>Orientation</td>
<td>$\theta_x = 70.42188382^\circ, \theta_y = -19.57811618^\circ$</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.957872058</td>
</tr>
</tbody>
</table>

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.
5.6 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>
<thead>
<tr>
<th>Table 5.3</th>
<th>Results on Violent Crime in the U.S.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2007</td>
<td></td>
</tr>
<tr>
<td>Center</td>
<td>(36.8766, -92.1131260)</td>
</tr>
<tr>
<td>Axes</td>
<td>$a = 2,604.517043$, $b = 821.0805658$</td>
</tr>
<tr>
<td>Area</td>
<td>6,718,353.465 sq. mi.</td>
</tr>
<tr>
<td>Standard Deviations</td>
<td>$\delta_x = 4.857083084$, $\delta_y = 4.857028283$</td>
</tr>
<tr>
<td>Rotation</td>
<td>X-Axis</td>
</tr>
<tr>
<td>Orientation</td>
<td>$\theta_x = 0.045206827^\circ$, $\theta_y = -89.95479317^\circ$</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.949007823</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 5.4</th>
<th>Results on Violent Crime in the U.S.</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td></td>
</tr>
<tr>
<td>Center</td>
<td>(36.88957333, -92.0322279)</td>
</tr>
<tr>
<td>Axes</td>
<td>$a = 2,577.512836$, $b = 809.2827681$</td>
</tr>
<tr>
<td>Area</td>
<td>6,553,163.484 sq. mi.</td>
</tr>
<tr>
<td>Standard Deviations</td>
<td>$\delta_x = 15.39212752$, $\delta_y = 15.39222404$</td>
</tr>
<tr>
<td>Rotation</td>
<td>Y-Axis</td>
</tr>
<tr>
<td>Orientation</td>
<td>$\theta_x = 89.89264601^\circ$, $\theta_y = -0.107353986^\circ$</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.949430202</td>
</tr>
</tbody>
</table>

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

5.7 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
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.

Table 5.5  Results on Gross Domestic Product of OECD Countries

<table>
<thead>
<tr>
<th>Year</th>
<th>Center</th>
<th>Axes</th>
<th>Area</th>
<th>Standard Deviations</th>
<th>Rotation</th>
<th>Orientation</th>
<th>Eccentricity</th>
</tr>
</thead>
<tbody>
<tr>
<td>2008</td>
<td>(44.95865069, 173.1929)</td>
<td>$\alpha = 4.308.807649$, $b = 695.7350122$</td>
<td>9,417,829.832 sq. mi.</td>
<td>$\delta_x = 20.69202965$, $\delta_y = 19.93139537$</td>
<td>X-Axis</td>
<td>$\theta_x = 1.307691476^\circ$, $\theta_y = -88.69230852^\circ$</td>
<td>0.986877931</td>
</tr>
</tbody>
</table>

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

5.8 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: $\bar{x}_i = \bar{x}$ and $\bar{y} = \bar{y}$. 
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 5.6  Results on Gross Domestic Product of OECD Countries

<table>
<thead>
<tr>
<th></th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
<td>(45.10440685, 172.9595)</td>
</tr>
<tr>
<td>Axes</td>
<td>$\alpha = 459.806334, \beta = 700.0081681$</td>
</tr>
<tr>
<td>Area</td>
<td>9,367,912.709 sq. mi.</td>
</tr>
<tr>
<td>Standard Deviations</td>
<td>$\delta_x = 21.04761702, \delta_y = 20.27737248$</td>
</tr>
<tr>
<td>Rotation</td>
<td>X-Axis</td>
</tr>
<tr>
<td>Orientation</td>
<td>$\theta_x = 1.328357537, \theta_y = -88.67164246$</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.986405646</td>
</tr>
</tbody>
</table>

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? Solution:
   - 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.

5.9 VBA Code

To calculate the weights \( w_i, i = 1, 2, \ldots n \), we first create the VBA subroutine called \texttt{SDE\_Stats(AVar)}. This subroutine does not require any input parameters. The first \texttt{FOR-NEXT} loop (2) calculates the sum of the total weights \( \sum_{i=1}^{n} w_i \). The second \texttt{FOR-NEXT} loop (3) calculates the observation weights

\[
w_i' = \frac{w_i}{\sum_{i=1}^{n} w_i}, \quad 0 \leq w_i' \leq 1.
\]

The observation weights are stored in column D of the active spreadsheet. Upon running this subroutine, column D will always sum to one. The use of the \texttt{WITH-END-WITH} statement is optional. It is a shortcut to not having to keep typing in the Excel object — in this case, the spreadsheet last referenced "Activesheet."

To calculate the mean center of gravity in Equation (5.9), the weights must be set on the active spreadsheet first. The subroutine named \texttt{SDE\_Stats(AVar)} calculates the weighted mean center of the standard deviational ellipse. The \texttt{FOR-NEXT} loop (2) multiplies the latitude and longitude observations by the weights. It sums the observations and puts the sums into the variables \texttt{mean\_latitude} and \texttt{mean\_longitude}.

The cell reference is always row, then column. For example, the code \texttt{.Cell(3,2)} refers to row 3 and column 2. Column 2 refers to column B.

The subroutine \texttt{Angle\_of\_Rotation()} calculates the angle of rotation \( \theta \) in Equation (5.10). It also gives the axis of rotation. The first \texttt{FOR-NEXT} loop (3) 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 \(\theta\) is stored in the variables \(\text{thetal}\) and \(\text{theta2}\). The IF-THEN-ELSE statement (13) determines the axis of rotation using the values \(\text{thetal}\) and \(\text{theta2}\). If \(\text{thetal} > \text{theta2}\) 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.

The subroutine \(\text{SDE} \_\text{Stats(AVar)}\) calculates the standard error terms along the axes in Equations (5.11) and (5.12). The first FOR-NEXT loop (4) calculates the sums and stores the values in variables \(\text{delta}_x\) and \(\text{delta}_y\). After the FOR-NEXT loop (4), 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 \(\text{delta}_x\) and \(\text{delta}_y\). The WITH-END-WITH statements (16) save the results to the output spreadsheet called STATS.

The subroutine \(\text{Eccentricity()}\) calculates the eccentricity using Equation (5.18). 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.

We use the function \(\text{Area()}\) to calculate the area of the standard deviational ellipse. It calculates the area using the formual \(F = \pi a_1 b_1\). We can write special code to calculate the area using Equation (5.17). It does need the sums \(x, y,\) and \(xy\) from the subroutine \(\text{Angle\_of\_Rotation()}\). 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:

\[
a_1^2 = \frac{1}{n} \sum_{i=1}^{n} y_i^2 w_i + \frac{2 (\sum_{i=1}^{n} x_i y_i w_i)^2}{n (\sum_{i=1}^{n} x_i^2 w_i - \sum_{i=1}^{n} y_i^2 w_i) + \sqrt{(\sum_{i=1}^{n} x_i^2 w_i - \sum_{i=1}^{n} y_i^2 w_i)^2 + 4 (\sum_{i=1}^{n} x_i y_i w_i)^2}}
\]

and

\[
b_1^2 = \frac{1}{n} \sum_{i=1}^{n} x_i^2 w_i = \frac{1}{n} \sum_{i=1}^{n} x_i^2 w_i - \frac{2 (\sum_{i=1}^{n} x_i y_i w_i)^2}{n (\sum_{i=1}^{n} x_i^2 w_i - \sum_{i=1}^{n} y_i^2 w_i) + \sqrt{(\sum_{i=1}^{n} x_i^2 w_i - \sum_{i=1}^{n} y_i^2 w_i)^2 + 4 (\sum_{i=1}^{n} x_i y_i w_i)^2}}
\]

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:
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.

The subroutine \texttt{SDE\_Likelihood( VAR)} calculates the log likelihood function of the regression model from Equation (5.8) of the weighted standard deviational ellipse for comparison to other mathematical models later. The subroutine \texttt{Put\_SDE\_Output \_In \_Spreadsheet(VAR)} outputs the variables and labels of the standard deviational ellipse to the spreadsheet called \texttt{STATS}.

Finally, we output the area, the axes lengths, and the eccentricity to the spreadsheet called \texttt{STATS} using the WITH\texttt{-}\_END\texttt{-}\_WITH statement (18).
Sub SDE_Stats(Var)
' 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
  mean_latitude = mean_latitude + ActiveSheet.Cells(i, 1).Value * ActiveSheet.Cells(i, 3).Value
  mean_longitude = mean_longitude + ActiveSheet.Cells(i, 2).Value * ActiveSheet.Cells(i, 3).Value
Next
mean_latitude = mean_latitude / sum_weight
mean_longitude = mean_longitude / sum_weight
' calculate the rotation from the axes
x = 0
y = 0
xy = 0
For i = 2 To n
  x = x + (ActiveSheet.Cells(i, 1).Value - mean_latitude)^2 * ActiveSheet.Cells(i, 3).Value
  y = y + (ActiveSheet.Cells(i, 2).Value - mean_longitude)^2 * ActiveSheet.Cells(i, 3).Value
      ActiveSheet.Cells(i, 3).Value
Next
Call Angle_of_Rotation(x, y, xy, note, atheta, itheta)
' calculate the deltas
delta_x = 0
delta_y = 0
For i = 2 To n
  delta_x = delta_x + ((ActiveSheet.Cells(i, 2).Value - mean_longitude) * Sin(atheta / (180 / WorksheetFunction.Pi())) - _
                     (ActiveSheet.Cells(i, 1).Value - mean_latitude) * Cos(atheta / (180 / WorksheetFunction.Pi())))^2 * ActiveSheet.Cells(i, 3).Value
  delta_y = delta_y + ((ActiveSheet.Cells(i, 2).Value - mean_longitude) * Cos(itheta / (180 / WorksheetFunction.Pi())))^2 * ActiveSheet.Cells(i, 3).Value
Next
delta_x = Sqr(delta_x / sum_weight)
delta_y = Sqr(delta_y / sum_weight)
' calculate the area, axes lengths, and eccentricity
m = n - 1
f = Area(m, x, y, xy) 'area
Call Axes_Len(m, x, y, xy, a, b)
e = Eccentricity(a, b)
area_check2 = Area_Check(a, b)
End Sub
Sub SDE_Likelihood(Var)
Dim sum_latitude, sum_longitude As Double
 Dim alpha, beta As Double
 '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
    mean_latitude = mean_latitude + ActiveSheet.Cells(i, 1).Value * ActiveSheet.Cells(i, 3).Value
    mean_longitude = mean_longitude + ActiveSheet.Cells(i, 2).Value * ActiveSheet.Cells(i, 3).Value
 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
        If j = 1 Then
        Else
        End If
    Next 'i
 Next 'j
 For i = 2 To n
    (ActiveSheet.Cells(i, 1).Value - mean_latitude) * (ActiveSheet.Cells(i, 2).Value - mean_longitude)
 Next 'off diagonals
```vba
'keep a copy for checks
'initialize the identity matrix
ActiveSheet.Cells(1, 23).Value = 1
ActiveSheet.Cells(2, 24).Value = 1
ActiveSheet.Cells(1, 24).Value = 0
ActiveSheet.Cells(2, 23).Value = 0
'perform guassian elimination to find the inverse of Z^-TZ
m = 24
  Do While m >= 21
    ActiveSheet.Cells(1, m).Value = ActiveSheet.Cells(1, m).Value / ActiveSheet.Cells(1, 21).Value
    m = m - 1
  Loop
'r1 - r2 ---> r2
m = 24
  Do While m >= 21
    m = m - 1
  Loop
'r2 / a22 ---> r2
m = 24
  Do While m >= 21
    m = m - 1
  Loop
'a12 r2 - r1 ---> r1
m = 24
  Do While m >= 21
    ActiveSheet.Cells(1, m).Value = ActiveSheet.Cells(1, 22).Value * ActiveSheet.Cells(2, m).Value -
    ActiveSheet.Cells(1, m).Value
    m = m - 1
  Loop
```
STANDARD DEVIATIONAL ELLIPSE

\[
\begin{bmatrix}
-1 & r_1 \\
\end{bmatrix} \rightarrow r_1
\]

\[m = 24\]

\[\textbf{Do While } m >= 21\]

\[\text{ActiveSheet.Cells(1, m).Value} = -1 * \text{ActiveSheet.Cells(1, m).Value}\]

\[m = m - 1\]

\[\textbf{Loop}\]

'(end of gaussian elimination. \((Z^\top T Z)^{-1}\) resides in (23,1), (24,1), (23,2), (24,2)

'(\(Z^\top T Z\) resides in (21,9), (22,9), (21,10), (22,10)

'calculate \(Z^\top Tw\)

\[\text{sum_latitude} = 0\]

\[\text{sum_longitude} = 0\]

\[n = \text{Selection.Rows.Count}\]

\[\textbf{For } i = 2 \text{ To } n\]

\[\text{sum_latitude} = \text{sum_latitude} + \text{ActiveSheet.Cells(i, 1).Value} * \text{ActiveSheet.Cells(i, 3).Value} \text{ 'w_i'}\]

\[\text{sum_longitude} = \text{sum_longitude} + \text{ActiveSheet.Cells(i, 2).Value} * \text{ActiveSheet.Cells(i, 3).Value} \text{ 'w_i'}\]

\[\textbf{Next}\]

'(calculate alpha and beta

\[\text{alpha} = \text{ActiveSheet.Cells(1, 23).Value} * \text{sum_latitude} + \text{ActiveSheet.Cells(1, 24).Value} * \text{sum_longitude}\]

\[\text{beta} = \text{ActiveSheet.Cells(2, 23).Value} * \text{sum_latitude} + \text{ActiveSheet.Cells(2, 24).Value} * \text{sum_longitude}\]

'(calculate log L

\[L_{\text{reg}} = 0\]

\[n = \text{Selection.Rows.Count}\]

\[\textbf{For } i = 2 \text{ To } n\]

\[L_{\text{reg}} = L_{\text{reg}} + (\text{ActiveSheet.Cells(i, 3).Value} - \text{alpha} * (\text{ActiveSheet.Cells(i, 1).Value} - \text{mean_latitude}) - \\
\]

\[\text{beta} * (\text{ActiveSheet.Cells(i, 2).Value} - \text{mean_longitude})\]

\[\textbf{Next}\]

\[\textbf{End Sub}\]
Sub Put_SDEOutput_In_Spreadsheet(Var)
    ' put the values in STATS spreadsheet
    With Worksheets("STATS")
        .Cells(1, 1).Value = "Center of Gravity Statistics"
        .Cells(2, 1).Value = "Latitude Mean Value"
        .Cells(3, 1).Value = "Longitude Mean Value"
        .Cells(2, 2).Value = mean_latitude
        .Cells(3, 2).Value = mean_longitude
        ' rotation from axes
        .Cells(4, 1).Value = "Rotation"
        .Cells(4, 2).Value = note
        .Cells(5, 1).Value = "Major Axis Rotation"
        .Cells(5, 2).Value = atheta
        .Cells(6, 1).Value = "Minor Axis Rotation"
        .Cells(6, 2).Value = itheta
        ' deltas along the axes
        .Cells(7, 1).Value = "Delta X"
        .Cells(7, 2).Value = delta_x
        .Cells(8, 1).Value = "Delta Y"
        .Cells(8, 2).Value = delta_y
        ' area, axis length, eccentricity
        .Cells(9, 1).Value = "Area (F)"
        .Cells(9, 2).Value = f
        .Cells(10, 1).Value = "Major Axis Length (a)"
        .Cells(10, 2).Value = a
        .Cells(11, 1).Value = "Minor Axis Length (b)"
        .Cells(11, 2).Value = b
        .Cells(12, 1).Value = "Area Check"
        .Cells(12, 2).Value = area_check2
        .Cells(13, 1).Value = "Eccentricity (e)"
        .Cells(13, 2).Value = e
        ' likelihood function
        .Cells(14, 1).Value = "Likelihood L(x,y)"
        .Cells(14, 2).Value = L_reg
    End With
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 \( \lambda_x \) is the scale parameter. We estimate \( \lambda_x \) 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

\[
F_x (w, x) = \int_0^t f(w, x) \, dx = \int_0^t \lambda_x w e^{-\lambda_x x} \, dx = \left[ -w e^{-\lambda_x x} \right]_0^t = w - we^{-\lambda_x t} = F(t_x).
\]
1. As \( t \to 0 \), \( F_t \to 0 \).

2. As \( t \to 180 \), \( F_t \to w \).

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 \( \lambda_x \) occurs at the first derivative of the likelihood function \( L = \prod_{i=1}^{n} f_x(w, x_i) \) taken with respect to \( \lambda_x \).

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

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

\[
\frac{d \log L}{d \lambda_x} = \frac{n}{\lambda_x w} - \sum_{i=1}^{n} x_i = 0 \Rightarrow \\
\hat{\lambda}_x = \frac{n}{w \sum_{i=1}^{n} x_i}, \quad (6.3)
\]

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 66. 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

\[ f_y(w_i, y_i) = \lambda_y w_i e^{-\lambda_y y_i}, \quad 0 \leq y_i \leq 360, \quad w_i > 0. \]  \hspace{1cm} (6.5)

We derive the cumulative distribution function as follow.

\[ F_y(w, y) = \int_0^y f(w, y) \, dy = \int_0^y \lambda_y w e^{-\lambda_y y} \, dy = -w e^{-\lambda_y y} \bigg|_0^u = w - w e^{-\lambda_y u} = F_u(u_y). \]

1. As \( u \to 0 \), \( F_u \to 0 \).
2. As \( u \to 360 \), \( F_u \to w \).
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 \( \lambda_y \) for the longitude by taking the derivative of the likelihood function \( L = \prod_{i=1}^n f(w, y_i) \) with respect to \( \lambda_y \); setting it to zero; and solving for \( \lambda_y \).

\[ L = \prod_{i=1}^n f(w, y_i) = \prod_{i=1}^n \lambda_y w \exp\left\{-\lambda_y y_i\right\} = (\lambda_y w)^n \exp\left\{-\lambda_y \sum_{i=1}^n y_i\right\} \]

\[ \log L = n \log(\lambda_y w) - \lambda_y \sum_{i=1}^n y_i, \]  \hspace{1cm} (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).

\[ \hat{\lambda}_y = \frac{n}{w \sum_{i=1}^n y_i}. \]  \hspace{1cm} (6.7)

We wish to see if we can obtain a linear relationship using the exponential distribution. We transform the data as follow:
\[ e^{-\lambda_y w u_y} = 1 - F_w(u_y), \]
\[ \lambda_y w u_y = \log_e \left[ \frac{1}{1 - F_w(u_y)} \right], \]
\[ \bar{u}_y = \frac{1}{\lambda_y} \log_e \left[ \frac{1}{1 - F_w(u_y)} \right]. \]  

(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 66. 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

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

\[ \prod_{i=1}^{n} \lambda_x w \exp \{-\lambda_x x_i\} \lambda_y w \exp \{-\lambda_y y_i\}. \]

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 \( \lambda_{xy} \).

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

(6.9)

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

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

(6.10)

Equation (6.10) gives the maximum likelihood estimator for \( \lambda_{xy} \) 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 \( F_{xy}(w, x, y) \) for the joint distribution, next. We need the cumulative distribution function in the exponential standard deviational ellipse formula.

\[ F_{xy}(w, x, y) = \int_0^t \int_0^u (w\lambda_{xy})^2 \exp \{ -\lambda_{xy}(x + y) \} \, dy \, dx = \]

\[ \int_0^t -w^2 \lambda_{xy} \exp \{ -\lambda_{xy}(x + y) \} \bigg|_0^u \, dx = \]

\[ \int_0^t -\lambda_{xy} w^2 \exp \{ -\lambda_{xy}(x + u) \} + \lambda_{xy} w^2 \exp \{ -\lambda_{xy} x \} \, dx = \]

\[ w^2 \exp \{ -\lambda_{xy}(x + u) \} - w^2 \exp \{ -\lambda_{xy} x \} \bigg|_0^t = \]

\[ w^2 \exp \{ -\lambda_{xy}(t + u) \} - w^2 \exp \{ -\lambda_{xy} t \} - w^2 \exp \{ -\lambda_{xy} u \} + w^2 = F_{tu}(w, t, u) \] (6.11)

Is Equation (6.11) a valid cumulative distribution function?

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

This is a valid cumulative distribution function only if the weights \( w \) sum to 1 as \( t \to 180 \) and \( u \to 360 \). 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 \( \mu \) and a scale parameter typically named the variance \( \sigma^2 \). The term really does not apply to the exponential distribution since it does not have a location parameter. This Section will derive the \textit{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.

\[
\frac{n}{\lambda_x} = \bar{x}. \tag{6.12}
\]

\[
\frac{n}{\lambda_y} = \bar{y}. \tag{6.13}
\]

The exponential ellipse uses the following formulas.

\[
t_x = \sum_{i=1}^{n} \frac{1}{\lambda_x} \log_e \left[ \frac{1}{1 - F_t(t_i)} \right] / w_i. \tag{6.14}
\]

\[
u_y = \sum_{i=1}^{n} \frac{1}{\lambda_y} \log_e \left[ \frac{1}{1 - F_u(u_i)} \right] / w_i. \tag{6.15}
\]

\[
t_{xy} = \sum_{i=1}^{n} \frac{1}{\lambda_{xy}} \log_e \left[ \frac{1}{1 - F_t(t_i, u_i)} \right] / w_i. \tag{6.16}
\]

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 \(\lambda_x, \lambda_y\), and \(\lambda_{xy}\) appear in Equations (6.3), (6.7) and (6.10).

6.4.2 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).

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

(6.17)

### 6.5 Axis Rotation

The two values of \(\theta\) 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.

### 6.6 Ellipse Properties

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

\[
\delta_x = \sum_{i=1}^{n} \left(u_{y,i} \sin \theta_a - t_{x,i} \cos \theta_a\right)^2 w_i
\]

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

\[
\delta_y = \sum_{i=1}^{n} \left(u_{y,i} \cos \theta_b - t_{x,i} \sin \theta_b\right)^2 w_i
\]

where the subscript \(b\) denotes the minor axis for the angle of rotation \(\theta\).

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

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

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.
\[
b^2 - \frac{\sum_{i=1}^{n} u_{y,i}^2}{n} = \frac{2 (\sum_{i=1}^{n} t_{x,y,i})^2}{n}\left(-\frac{\left(\sum_{i=1}^{n} t_{x,i}^2 - \sum_{i=1}^{n} u_{y,i}^2\right) + \sqrt{\left(\sum_{i=1}^{n} t_{x,i}^2 - \sum_{i=1}^{n} u_{y,i}^2\right)^2 + 4 \left(\sum_{i=1}^{n} t_{x,y,i}\right)^2}}{\left(\sum_{i=1}^{n} t_{x,i}^2 - \sum_{i=1}^{n} u_{y,i}^2\right)}\right)
\]
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.
\[
F = \frac{\pi}{n} \sqrt{\sum_{i=1}^{n} t_{x,i}^2 \sum_{i=1}^{n} u_{y,i}^2 - \left(\sum_{i=1}^{n} t_{x,y,i}\right)^2}
\]
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.
\[
e = \frac{\sqrt{a^2 - b^2}}{a}
\]
where Equation (6.20) gives the major axis length \( a \) and Equation (6.21) gives the minor axis length \( b \).

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.

If the axis lengths \( a \geq b \), then \( a \) is the semi-major axis. If \( b > a \), then \( b \) is assumed to be the semi-major axis and Equation (6.23) is applied appropriately.

### 6.7 Kentucky Example

We compare the weighted, standard deviational ellipse Tables 5.1 on page 51 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 74 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...
Table 6.1  Results for the Exponential Model for Kentucky

<table>
<thead>
<tr>
<th></th>
<th>2003</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
<td>(37.38297399, −86.8129514)</td>
</tr>
<tr>
<td>Axes</td>
<td>( a = 273.2320657, b = 37.38920285 )</td>
</tr>
<tr>
<td>Area</td>
<td>32,094.2879 sq. mi.</td>
</tr>
<tr>
<td>Standard Deviations</td>
<td>( \delta_x = 3.050700055, \delta_y = 3.050809397 )</td>
</tr>
<tr>
<td>Rotation</td>
<td>Y-Axis</td>
</tr>
<tr>
<td>Orientation</td>
<td>( \theta_x = 89.99867285^\circ, \theta_y = -0.00132743^\circ )</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.9905930999</td>
</tr>
<tr>
<td>Concentration</td>
<td>27/90</td>
</tr>
</tbody>
</table>

Table 6.2  Results for the Exponential Model for Kentucky

<table>
<thead>
<tr>
<th></th>
<th>2004</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
<td>(37.40588805, −86.8066601)</td>
</tr>
<tr>
<td>Axes</td>
<td>( a = 273.2391008, b = 37.4120928 )</td>
</tr>
<tr>
<td>Area</td>
<td>32,114.76312 sq. mi.</td>
</tr>
<tr>
<td>Standard Deviations</td>
<td>( \delta_x = 2.987069658, \delta_y = 2.987088203 )</td>
</tr>
<tr>
<td>Rotation</td>
<td>Y-Axis</td>
</tr>
<tr>
<td>Orientation</td>
<td>( \theta_x = 89.99870116^\circ, \theta_y = -0.001299137^\circ )</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.99058201</td>
</tr>
</tbody>
</table>

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.
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.

6.8 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  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 6.4  Results for the Exponential Model on Violent Crime in the U.S.

<table>
<thead>
<tr>
<th>2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Axes</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Area</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Standard Deviations</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Rotation</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Orientation</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Eccentricity</td>
</tr>
</tbody>
</table>

Center: (36.88957333, −92.0322279)

Axes: \( a = 268.1090262, b = 36.90903466 \)

Area: 31.088.08671 sq. mi.

Standard Deviations: \( \delta_x = 5.5114\,0013, \delta_y = 5.5114\,65094 \)

Rotation: Y-Axis

Orientation: \( \theta_x = 89.99754726^\circ, \theta_y = −0.002453032^\circ \)

Eccentricity: 0.990478949
Figure 6.5 on page 76 shows the exponential ellipse inside of the standard deviational ellipse. The center of gravity is the same as that for the standard deviational ellipse (36.89957, −92.03222). There is a tremendous difference in the size of the two different ellipses.

### 6.9 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 6.5 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.

<table>
<thead>
<tr>
<th>Table 6.5</th>
<th>Results for the Exponential Model on Gross Domestic Product</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2008</strong></td>
<td></td>
</tr>
<tr>
<td>Center</td>
<td>(44.95865069, 173.19287)</td>
</tr>
<tr>
<td>Axes</td>
<td>(a = 162.1154426, b = 45.08062313)</td>
</tr>
<tr>
<td>Area</td>
<td>22.959.59217 sq. mi.</td>
</tr>
<tr>
<td>Standard Deviations</td>
<td>(\delta_x = 5.203392563, \delta_y = 5.203732811)</td>
</tr>
<tr>
<td>Rotation</td>
<td>Y-Axis</td>
</tr>
<tr>
<td>Orientation</td>
<td>(\theta_x = 89.99301352^\circ, \theta_y = -0.006986772^\circ)</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.960558702</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 6.6</th>
<th>Results for the Exponential Model on Gross Domestic Product</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2009</strong></td>
<td></td>
</tr>
<tr>
<td>Center</td>
<td>(45.10440685, 172.9595353)</td>
</tr>
<tr>
<td>Axes</td>
<td>(a = 161.9197005, b = 45.22752649)</td>
</tr>
<tr>
<td>Area</td>
<td>23.006.59785 sq. mi.</td>
</tr>
<tr>
<td>Standard Deviations</td>
<td>(\delta_x = 5.198792719, \delta_y = 5.199135181)</td>
</tr>
<tr>
<td>Rotation</td>
<td>Y-Axis</td>
</tr>
<tr>
<td>Orientation</td>
<td>(\theta_x = 89.99290395, \theta_y = -0.007006348)</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.960197864</td>
</tr>
</tbody>
</table>
6.10 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 $a_1$.

\[
a_1^2 = \frac{1}{n} \sum_{i=1}^{n} y_i^2 w_i = \tag{6.24}
\]

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

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.

\[
a_2^2 = \frac{1}{n} \sum_{i=1}^{n} t_{y,i}^2 = \tag{6.25}
\]

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

where

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

\[
\lambda_x = \sum_{i=1}^{n} \frac{w_i}{w_i x_i}, \quad w_i = \frac{w_i}{\sum_{i=1}^{n} w_i}, \quad 0 \leq x_i \leq 180.
\]

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

\[
\lambda_y = \sum_{i=1}^{n} \frac{w_i}{w_i y_i}, \quad w_i = \frac{w_i}{\sum_{i=1}^{n} w_i}, \quad 0 \leq y_i \leq 360.
\]

\[
t_{x,y} = \sum_{i=1}^{n} \left( \frac{1}{\lambda_{xy}} \log_e \left[ \frac{1}{1 - F_{xy}(t_i, u_i)} \right] / w_i \right)^2,
\]
\[ \bar{x}_{xy} = \frac{2n}{\sum_{i=1}^{n} w'_{ix} + w'_{iy}}, \quad w'_{i} = \frac{w_{i}}{\sum_{i=1}^{n} w_{i}}, \]

\[ 0 \leq x_{i} \leq 180, \quad 0 \leq y_{i} \leq 360. \]

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 \( n_{a}^{2} \leq n_{a}^{1} \).

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 \( \beta \lambda x \) and \( \bar{x} \) in Equation (6.12), we can rewrite the sum from the exponential ellipse as

\[ nt_{x,i}^{2} = n \sum_{i=1}^{n} \left( \frac{1}{\bar{x}} \log_{e} \left[ \frac{1}{(1 - F_{x}(t_{i}))} \right] / w'_{i} \right)^{2} = \]

\[ \frac{1}{n} \sum_{i=1}^{n} \left( \bar{x} \log_{e} \left[ \frac{1}{1 - F_{x}(t_{i})} \right] / \sum_{i=1}^{n} w_{i} \right)^{2} \]

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

\[ \log_{e} = (x - 1) - \frac{(x-1)^{2}}{2} - \frac{(x-1)^{3}}{3} - \frac{(x-1)^{4}}{4} + \cdots, \quad 0 < x \leq 2. \]

So,

\[ \sum_{i=1}^{n} (x_{i} - \bar{x})^{2} w_{i} = \sum_{i=1}^{n} (x_{i}^{2} - 2x_{i}\bar{x} + \bar{x}^{2}) w_{i} > \]

\[ \frac{1}{n} \sum_{i=1}^{n} \left( \bar{x} \left[ \frac{1}{(1 - F_{x}(t_{i}))} - 1 \right] - \frac{1}{2} \left( \frac{1}{(1 - F_{x}(t_{i}))} - 1 \right)^{2} + \right. \]

\[ \left. \frac{1}{3} \left( \frac{1}{(1 - F_{x}(t_{i}))} - 1 \right)^{3} \right. \]

\[ \frac{1}{4} \left( \frac{1}{(1 - F_{x}(t_{i}))} - 1 \right)^{4} + \cdots \left( \sum_{i=1}^{n} w_{i} \right)^{2} = \]
\[
\frac{1}{n} \sum_{i=1}^{n} \left( \bar{x} \log_e \left[ \frac{1}{1 - F_x(t_i)} \right] \left( \frac{1}{w_i} \right)^2 \right) \\
= \sum_{i=1}^{n} \left( \frac{1}{\lambda_x} \log_e \left[ \frac{1}{1 - F_x(t_i)} \right] \right) \left( \frac{1}{w_i} \right)^2.
\]

Hence, we can conclude that the sums \( n \sum_{i=1}^{n} x_i^2 w_i > nt_{x,i}^2 \). 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 \( n \sum_{i=1}^{n} y_i^2 w_i > nu_{y,i}^2 \).

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

\[
\left( \sum_{i=1}^{n} x_i y_i w_i \right)^2 = \left( \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y}) w_i \right)^2 > t_{x,y}^2 =
\]

\[
= \left( \sum_{i=1}^{n} \left( \frac{\bar{x} + \bar{y}}{2} \right) \left[ \frac{1}{(1 - F_{tu}(t_i, u_i))} - 1 \right] \right)^2 + \left( \sum_{i=1}^{n} \left( \frac{1}{(1 - F_{tu}(t_i, u_i))} - 1 \right)^3 \right)^2 + \left( \sum_{i=1}^{n} \left( \frac{1}{(1 - F_{tu}(t_i, u_i))} - 1 \right)^4 \right)^2 + \ldots
\]

\[
= \left( \sum_{i=1}^{n} \left( \frac{1}{\lambda_{xy}} \log_e \left[ \frac{1}{1 - F_{tu}(t_i, u_i)} \right] \right) \right)^2 \\
\leq 2 \text{ and the sums } \left( \sum_{i=1}^{n} x_i y_i w_i \right)^2 \text{ and } (t_{x,y})^2 \text{ are positive definite.}
\]

We have shown that the three sums \( t_{x,i}^2, u_{x,i}^2, \) 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.

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

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

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

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

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

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

1. $n \sum_{i=1}^{n} x_i^2 w_i > nt_{x,i}^2$.
2. $n \sum_{i=1}^{n} y_i^2 w_i > nu_{y,i}^2$.
3. $(\sum_{i=1}^{n} x_i y_i w_i)^2 > t_{x,y,i}^2$.

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 $\theta$. Let $\theta_1$ be the expression for the angle of rotation for the standard deviational ellipse given in Equation (6.28).
\[
\tan \theta_1 = \pm \left( \frac{\sum_{i=1}^{n} x_i^2 w_i - \sum_{i=1}^{n} y_i^2 w_i}{2 \sum_{i=1}^{n} x_i y_i w_i} \right) \tag{6.28}
\]

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

\[-1 \leq \tan \theta_1 \leq 1. \] 

Let \( \theta_2 \) be the angle of rotation for the exponential ellipse given in Equation (6.29).

\[
\tan \theta_2 = \pm \left( \frac{\sum_{i=1}^{n} t_{x,i}^2 - \sum_{i=1}^{n} u_{y,i}^2}{2 \sum_{i=1}^{n} t_{x,y,i}} \right) \tag{6.29}
\]

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

\[-1 \leq \tan \theta_2 \leq 1. \]

Tables 6.7 and 6.8 summarize the relationship between \( \theta_1 \) and \( \theta_2 \) for different values of the plus/minus sign in the middle of \( \theta_1 \) and \( \theta_2 \).

**Table 6.7** Relationship of angles of rotation for large \( x_i \)’s.

<table>
<thead>
<tr>
<th>Bounds</th>
<th>Plus/Minus</th>
<th>( \tan() )</th>
<th>Angle</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sum_{i=1}^{n} x_i^2 w_i &gt; \sum_{i=1}^{n} y_i^2 w_i )</td>
<td>+</td>
<td>Positive</td>
<td>( \theta_1 &gt; 0 )</td>
<td>( 0 &lt; \theta_1 &lt; 90 )</td>
</tr>
<tr>
<td>( \sum_{i=1}^{n} t_{x,i}^2 &gt; \sum_{i=1}^{n} u_{y,i}^2 )</td>
<td>+</td>
<td>Positive</td>
<td>( \theta_2 &gt; 0 )</td>
<td>( 0 &lt; \theta_2 &lt; \theta_1 &lt; 90 )</td>
</tr>
<tr>
<td>( \sum_{i=1}^{n} x_i^2 w_i &gt; \sum_{i=1}^{n} y_i^2 w_i )</td>
<td>–</td>
<td>Negative</td>
<td>( \theta_1 &lt; 0 )</td>
<td>( 0 &lt; \theta_1 &lt; 90 )</td>
</tr>
<tr>
<td>( \sum_{i=1}^{n} t_{x,i}^2 &gt; \sum_{i=1}^{n} u_{y,i}^2 )</td>
<td>–</td>
<td>Negative</td>
<td>( \theta_2 &lt; 0 )</td>
<td>( 0 &lt; \theta_1 &lt; \theta_2 &lt; 90 )</td>
</tr>
</tbody>
</table>

**Table 6.8** Relationship of angles of rotation for small \( x_i \)’s.

<table>
<thead>
<tr>
<th>Bounds</th>
<th>Plus/Minus</th>
<th>( \tan() )</th>
<th>Angle</th>
<th>Inference</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sum_{i=1}^{n} x_i^2 w_i &lt; \sum_{i=1}^{n} y_i^2 w_i )</td>
<td>+</td>
<td>Positive</td>
<td>( \theta_1 &gt; 0 )</td>
<td>( 0 &lt; \theta_1 &lt; 90 )</td>
</tr>
<tr>
<td>( \sum_{i=1}^{n} t_{x,i}^2 &lt; \sum_{i=1}^{n} u_{y,i}^2 )</td>
<td>+</td>
<td>Positive</td>
<td>( \theta_2 &gt; 0 )</td>
<td>( 0 &lt; \theta_2 &lt; \theta_1 &lt; 90 )</td>
</tr>
<tr>
<td>( \sum_{i=1}^{n} x_i^2 w_i &lt; \sum_{i=1}^{n} y_i^2 w_i )</td>
<td>–</td>
<td>Negative</td>
<td>( \theta_1 &lt; 0 )</td>
<td>( 0 &lt; \theta_1 &lt; 90 )</td>
</tr>
<tr>
<td>( \sum_{i=1}^{n} t_{x,i}^2 &lt; \sum_{i=1}^{n} u_{y,i}^2 )</td>
<td>–</td>
<td>Negative</td>
<td>( \theta_2 &lt; 0 )</td>
<td>( 0 &lt; \theta_1 &lt; \theta_2 &lt; 90 )</td>
</tr>
</tbody>
</table>

Upon examining Tables 6.7 and 6.8, we would expect that for extreme values of \( \theta_1 \) that the exponential ellipse would fall inside the standard deviational ellipse.

**6.11 Exercises**

1. Consider the pdf \( f(y) = \lambda e^{-\lambda y / w} \).
(a) Derive the mle for $\lambda_y$.
(b) Find the cdf. Show that it is a true cdf for $\sum_{i=1}^n w_i = 1$. Solution:

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

$$F(y) = \int_0^y \lambda e^{-\lambda y/w} dy = we^{-\lambda wy} \bigg|_0^y = w - we^{-\lambda wy}$$

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

2. Consider the pdf $f = \lambda e^{-\lambda w y}$.
(a) Derive the mle for $\lambda_y$.
(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

$$\frac{n}{\lambda_x} = \bar{x}$$ in Chapter 5.

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

$$\hat{\lambda}_x = \frac{n}{w \sum_{i=1}^n x_i} = \frac{n}{w \sum_{i=1}^n \frac{w_i x_i}{w_i}} \Rightarrow$$

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

6. Show that

$$\frac{n}{\lambda_y} = \bar{y}$$ 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,

\[
\hat{\lambda}_y = \frac{n}{w \sum_{i=1}^{n} y_i} = \frac{n}{\left( \sum_{i=1}^{n} \frac{w_i y_i}{\sum_{i=1}^{n} w_i} \right)} \Rightarrow \\
\frac{n}{\lambda_y} = \sum_{i=1}^{n} \frac{w_i y_i}{\sum_{i=1}^{n} w_i} = \bar{y}, \\
0 \leq y_i \leq 360, \quad -180 \leq \bar{y} \leq 180.
\]

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

The subroutine \texttt{EXP\_Stats(AVar)} (1) calculates the parameter estimates \( \hat{\lambda}_x, \hat{\lambda}_y, \) and \( \hat{\lambda}_{xy} \) 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 \texttt{FOR-NEXT} loop (3) calculates the sums for the maximum likelihood estimates. The VBA statements that follow the \texttt{FOR-NEXT} loop (3) calculate the final values for the parameter estimates and store them in the variables \( \text{lambda}_x \) for the latitude, \( \text{lambda}_y \) for the longitude, and \( \text{lambda}_{xy} \) for the joint distribution. We need the parameter estimates to calculate the distribution functions.

The subroutine \texttt{EXP\_Stats(AVar)} (1) 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 \texttt{FOR-NEXT} loop (4) 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 \texttt{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.

The subroutine \texttt{EXP\_Stats(AVar)} (1) calculates the deviations on the X-axis and the Y-axis. The \texttt{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 \( \theta \) we previous 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 \( \text{delta}_x \) and \( \text{delta}_y \). The source code after the \texttt{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 \( \text{error}_x \) and \( \text{error}_y \).

The \texttt{EXP\_Stats(AVar)} 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 \texttt{IF-THEN-ELSE} statement using Equation (6.22) and saves the value in the variable \( f \). The \texttt{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 \( \sum_{i=1}^{n} t^2_{x,i} < \sum_{i=1}^{n} u^2_{y,i} \) 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 \texttt{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 \texttt{WITH-END-WITH} statement (7) saves the remaining exponential statistics to the \texttt{STATS} spreadsheet in column \texttt{E} (column 5). We add appropriate labels in column \texttt{D} (column 4).
Sub EXP_Stats(Var)
    n = Selection.Rows.Count
    For i = 2 To n
        ActiveSheet.Cells(i, 5) = ActiveSheet.Cells(i, 2).Value 'copy values to column E
    Next
    wx = 0
    wy = 0
    For i = 2 To n
        wx = wx + ActiveSheet.Cells(i, 1).Value * ActiveSheet.Cells(i, 4).Value
        wy = wy + ActiveSheet.Cells(i, 5).Value * ActiveSheet.Cells(i, 4).Value
    Next
    lambda_x = (n - 1) / wx
    lambda_y = (n - 1) / wy
    lambda_xy = 2 * (n - 1) / (wx + wy)

'calculate the cumulative distribution functions for the latitude, longitude, and the joint distribution
'calculate the values t_x, t_y, and t_xy, t_x^2, t_y^2, and t_xy^2
    t_x = 0
    t_y = 0
    t_xy = 0
    t_xsq = 0
    t_ysq = 0
    t_xysq = 0

1

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
    l_x = ActiveSheet.Cells(i, 4).Value / ActiveSheet.Cells(i, 4).Value * EXP(-1 * lambda_x * ActiveSheet.Cells(i, 4).Value)
    l_y = ActiveSheet.Cells(i, 4).Value / ActiveSheet.Cells(i, 4).Value * EXP(-1 * lambda_y * ActiveSheet.Cells(i, 5).Value)

2

ActiveSheet.Cells(i, 4).Value = EXP(-1 * lambda_xy * ActiveSheet.Cells(i, 5).Value) - 1

3

ActiveSheet.Cells(i, 4).Value = l_x * l_y * l_xy

4

ActiveSheet.Cells(i, 4).Value = l_x + l_y + l_xy

Next
'calculate the angle of rotations about the axes
Call AngleOfRotation(txsqr, tysqr, txy, note, atheta, itheta)
'calculate the deviations on the axes
deltax = 0
deltay = 0
For i = 2 To n
    txy = ActiveSheet.Cells(i, 4).Value * EXP(-1 * lambday * ActiveSheet.Cells(i, 5).Value)
    lambdax = 1 / lambdax * Log(1 / (1 - tx)) / ActiveSheet.Cells(i, 4).Value
    lambday = 1 / lambday * Log(1 / (1 - txy)) / ActiveSheet.Cells(i, 4).Value
    deltax = deltax + lambday * Sin(atheta / 57.2957795) - lambdax * Cos(atheta / 57.2957795)
    deltay = deltay + lambday * Sin(itheta / 57.2957795) - lambdax * Cos(itheta / 57.2957795)
Next
errorx = Sqr(deltax / sum_weight) 'standard errors on the x axis
errorey = Sqr(deltay / sum_weight) 'standard errors on the y axis
'calculate the area and eccentricity
f = Area(n - 1, txsqr, tysqr, txy)
Call AxesLength(n, txsqr, tysqr, lambdax, lambday, a, b)
e = Eccentricity(a, b)
area_check2 = AreaCheck(a, b)
End Sub

Sub PutEXPOutputInSpreadsheet(Var)
    With ActiveSheet("STATS")
        .Cells(1, 4).Value = "Exponential Ellipse Statistics"
        .Cells(2, 4).Value = "Mean Latitude"
        .Cells(3, 4).Value = "Mean Longitude"
        .Cells(2, 5).Value = (n - 1) / lambdax
        .Cells(3, 5).Value = (n - 1) / lambday
        .Cells(4, 4).Value = "Rotation"
        .Cells(5, 4).Value = "Major Axis Rotation"
        .Cells(6, 4).Value = "Minor Axis Rotation"
        .Cells(4, 5).Value = note
        .Cells(5, 5).Value = atheta
        .Cells(6, 5).Value = itheta
    End With
'the deltas
.Cells(7, 4).Value = "Delta X"
.Cells(8, 4).Value = "Delta Y"
.Cells(7, 5).Value = error_x
.Cells(8, 5).Value = error_y
'area, axes lengths, eccentricity, likelihood function
.Cells(9, 4).Value = "Area (F)"
.Cells(10, 4).Value = "Major Axis Length (a)"
.Cells(11, 4).Value = "Minor Axis Length (b)"
.Cells(12, 4).Value = "Area Check"
.Cells(13, 4).Value = "Eccentricity (e)"
.Cells(14, 4).Value = "Likelihood Function L(x)"
.Cells(15, 4).Value = "Likelihood Function L(y)"
.Cells(16, 4).Value = "Likelihood Function L(xy)"
.Cells(9, 5).Value = f
.Cells(10, 5).Value = a
.Cells(11, 5).Value = b
.Cells(12, 5).Value = area_check2
.Cells(13, 5).Value = e
.Cells(14, 5).Value = L_x
.Cells(15, 5).Value = L_y
.Cells(16, 5).Value = L_xy
End With
CHAPTER 7

THE WEIBULL ELLIPSE

7.1 Latitude Mean Center

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

\[
f_x(w_i, x_i) = \gamma_x (w_i \lambda_x)^{\gamma_x} (x_i)^{\gamma_x-1} \exp \{ - (\lambda_x x_i)^{\gamma_x} \},
\]

(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 \( f_x(w, x_i) \).

\[
F_x(w, x_i) = \int_0^t f(w, x) \, dx =
\]

\[
\int_0^t \gamma_x (w \lambda_x)^{\gamma_x} (x_i)^{\gamma_x-1} \exp \{ - (\lambda_x x_i)^{\gamma_x} \} \, dx =
\]

\[
-w^{\gamma_x} \exp \{ - (\lambda_x x)^{\gamma_x} \} \bigg|_0^t = w^{\gamma_x} - w^{\gamma_x} \exp \{- (\lambda_x t)^{\gamma_x} \} =
\]
The properties of this cumulative distribution function are:

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

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

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

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 $\lambda_x$ occurs at the first derivative of $L = \prod_{i=1}^{n} f_x(w, x_i)$ taken with respect with $\lambda_x$. The maximum likelihood estimator for $\gamma_x$ occurs at the first derivative of $L$ taken with respect with $\gamma_x$. $x_1, x_2, \ldots, x_n$ is the random sample. Equation (7.2) gives the likelihood function.

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

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

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

$$\lambda_x^{\gamma_x} \sum_{i=1}^{n} x_i^{\gamma_x}. \quad (7.3)$$

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.

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

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

$$n \gamma_x - w \gamma_x \lambda_x^{\gamma_x} \sum_{i=1}^{n} x_i^{\gamma_x} = 0$$
\[ n - w \lambda_x^x \sum_{i=1}^{n} x_i^{y_x} = 0 \]  \hspace{1cm} (7.4)

and

\[ \frac{d \log L}{d \gamma_x} = 0 \Rightarrow \]

\[ \frac{n}{\gamma_x} + n \log(w \lambda_x) + \sum_{i=1}^{n} \log x_i - \]

\[ \lambda_x^x \sum_{i=1}^{n} x_i^{y_x} (\log \lambda_x + \log x_i) = 0 = L'(w, x_i). \]

Solving Equation (7.4) for \( \lambda_x \) yields

\[ \lambda_x = \left( \frac{n}{w \sum x_i^{y_x}} \right)^{\frac{1}{\gamma_x}}. \]  \hspace{1cm} (7.6)

We must solve for \( \gamma_x \) 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 \( k \).

\[ \gamma(x,k+1) = \gamma(x,k) - \frac{L(w,x_i)}{L'(w,x_i)}. \]  \hspace{1cm} (7.7)

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:

\[ \gamma(k+1) = \]

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

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:

\[ \gamma(x,k+1) = \gamma(x,k) - \]

\[ L \left[ \hat{\lambda}(x,k), \gamma(x,k) \right] \left( \frac{\gamma(x,k) - \gamma(x,k-1)}{L \left[ \hat{\lambda}(x,k), \gamma(x,k) \right] - L \left[ \hat{\lambda}(x,k-1), \gamma(x,k-1) \right]} \right) \]  \hspace{1cm} (7.8)

where \( k \) denotes the iteration.

With the Secant method, we must choose two initial values for \( \gamma_x \) instead of one. The reader will find that working with either method is difficult to work with for
finding the value of $\lambda_x$. 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

$$\bar{x} = \frac{n}{\lambda_x} \quad (7.9)$$

Notice that it depends on the range of $\lambda_x$. 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 $\gamma_x$ 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 $\gamma_x$ 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 $-180^\circ$ to $180^\circ$. 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.

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

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

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

$$F_y(w, y) = \int_0^y f(w, y) \, dy =$$

$$\int_0^y \gamma_y (w \lambda_y)^{\gamma_y} (y_i)^{\gamma_y - 1} \exp \left\{ - \left( \lambda_y y_i \right)^{\gamma_y} \right\} \, dy =$$

$$-w^{\gamma_y} \exp \left\{ - \left( \lambda_y y \right)^{\gamma_y} \right\} \bigg|_0^w = w^{\gamma_y} - w^{\gamma_y} \exp \left\{ - \left( \lambda_y u \right)^{\gamma_y} \right\} =$$

$$F_u(w, u).$$

The properties of this cumulative distribution function are:
1. As $u \to 0$, $F_u \to 0$.

2. As $u \to 360$, $F_u \to w^{\gamma_y}$.

3. $F_u$ is right continuous.

As $u \to 360$, $F_u \to w^{\gamma_y} \Rightarrow w^{\gamma_y}$ 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 $\lambda_y$ occurs at the first derivative of $L = \prod_{i=1}^{n} f(y_i)$ taken with respect with $\lambda_y$. The maximum likelihood estimator for $\gamma_y$ occurs at the first derivative of $L$ taken with respect with $\gamma_y$. $y_1, y_2, \ldots, y_n$ is the random sample. Equation (7.11) gives the likelihood function.

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

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

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

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

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.

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

$$\frac{n \gamma_y}{w \lambda_y} - \gamma_y \lambda_y^{\gamma_y - 1} \sum_{i=1}^{n} y_i^{\gamma_y} = 0$$

$$n \gamma_y - w^{\gamma_y} \lambda_y^{\gamma_y} \sum_{i=1}^{n} y_i^{\gamma_y} = 0$$

$$n - w \lambda_y^{\gamma_y} \sum_{i=1}^{n} y_i^{\gamma_y} = 0 \quad (7.13)$$
and

\[ \frac{d \log L}{d \gamma_y} = 0 \Rightarrow \]

\[ \frac{n}{\gamma_y} + n \log(w \lambda_y) + \sum_{i=1}^{n} \log y_i - \lambda_y \sum_{i=1}^{n} y_i^{\gamma_y} (\log \lambda_y + \log y_i) = 0 = L'(w, y_i). \]  

(7.14)

Solving Equation (7.13) for \( \lambda_y \) yields

\[ \lambda_y = \left( \frac{\sum_{i=1}^{n} y_i^{\gamma_y}}{w} \right)^{1/\gamma_y}. \]  

(7.15)

\( \gamma_y \) 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 \( k \).

\[ \gamma_{(y,k+1)} = \gamma_{(y,k)} - \frac{L(w, y_i)}{L'(w, y_i)}. \]

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:

\[ \gamma_{(k+1)} = \gamma_{(k)} - \frac{L \left[ \lambda_{(k)}, \gamma_{(k)} \right]}{ \log L \left[ \lambda_{(k)}, \gamma_{(k)} \right] - \log L \left[ \lambda_{(k-1)}, \gamma_{(k-1)} \right] } \]  

(7.16)

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

\[ \gamma_{(y,k+1)} = \gamma_{(y,k)} - \frac{\log L \left[ \lambda_{(y,k)}, \gamma_{(y,k)} \right]}{ \log L \left[ \lambda_{(y,k)}, \gamma_{(y,k)} \right] - \log L \left[ \lambda_{(y,k-1)}, \gamma_{(y,k-1)} \right] } \]  

(7.17)

where \( k \) denotes the iteration.

With the Secant method, we must choose two initial values for \( \gamma_y \) instead of one. The reader will find that working with either method is difficult to work with for finding the value of \( \lambda_y \). 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.

$$\bar{y} = \frac{n}{\lambda_y}$$  \hspace{1cm} (7.18)

Notice that the relationship depends on the range of $\lambda_y$ 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 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 $\lambda_{xy}$.

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

$$2 \sum_{i=1}^{n} \log \gamma_{xy} + 2n\gamma_{xy} \log(w\lambda_{xy}) + (\gamma_{xy} - 1) \sum_{i=1}^{n} \log x_i y_i -$$  \hspace{1cm} (7.19)
\[
\sum_{i=1}^{n} (\lambda_{xy} x_i)^{\gamma_{xy}} - \sum_{i=1}^{n} (\lambda_{xy} y_i)^{\gamma_{xy}}.
\]

\[
\frac{d \log L}{d \lambda_{xy}} = 0 + 2n \gamma_{xy} \frac{w \lambda_{xy}}{u} = 0 + 0 - \gamma_{xy} \sum_{i=1}^{n} (\lambda_{xy} x_i)^{\gamma_{xy} - 1} - \gamma_{xy} \sum_{i=1}^{n} (\lambda_{xy} y_i)^{\gamma_{xy} - 1} = 0,
\]

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

Solving for \(\lambda_{xy}\) yields the following expressions.

\[
-w \gamma_{xy} \lambda_{xy} \sum_{i=1}^{n} x_i^{\gamma_{xy}} - w \gamma_{xy} \lambda_{xy} \sum_{i=1}^{n} y_i^{\gamma_{xy}} = -2n \gamma_{xy} =
\]

\[
\lambda_{xy}^{\gamma_{xy}} \left[ w \left( \sum_{i=1}^{n} x_i^{\gamma_{xy}} + \sum_{i=1}^{n} y_i^{\gamma_{xy}} \right) \right] = 2n
\]

\[
\lambda_{xy} = 2n \left/ \left[ w \left( \sum_{i=1}^{n} x_i^{\gamma_{xy}} + \sum_{i=1}^{n} y_i^{\gamma_{xy}} \right) \right]^{1/(\gamma_{xy})} \right.. \tag{7.20}
\]

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

\[
\gamma(xy,k+1) = \gamma(xy,k) - \log L \left[ \lambda(xy,k), \gamma(xy,k) \right] \tag{7.21}
\]

\[
\left( \frac{\gamma(xy,k) - \gamma(xy,k-1)}{\log L \left[ \lambda(xy,k), \gamma(xy,k) \right] - \log L \left[ \lambda(xy,k-1), \gamma(xy,k-1) \right]} \right)
\]

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

\[
f_{xy}(w_i, x_i, y_i) =
\]

\[
(\gamma_{xy})^2 \left( w_i \lambda_{xy} \gamma_{xy} (x_i y_i)^{\gamma_{xy} - 1} \right) \exp \left\{ - (\lambda_{xy} x_i)^{\gamma_{xy}} - (\lambda_{xy} y_i)^{\gamma_{xy}} \right\}.
\]

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**.
$F_{xy}(w, x, y) = \int_0^t \int_0^u (\gamma_{xy})^2 (w \lambda_{xy})^{2\gamma_{xy}} (x y)^{\gamma_{xy} - 1} \times \exp \left\{ -(\lambda_{xy} x)^{\gamma_{xy}} - (\lambda_{xy} y)^{\gamma_{xy}} \right\} dy \, dx = \int_0^t -(\gamma_{xy}) (w \lambda_{xy})^{\gamma_{xy}} (x)^{\gamma_{xy} - 1} \times \exp \left\{ -(\lambda_{xy} x)^{\gamma_{xy}} - (\lambda_{xy} y)^{\gamma_{xy}} \right\} \bigg|_0^u \, dx = \int_0^t (\gamma_{xy}) (w \lambda_{xy})^{\gamma_{xy}} (x)^{\gamma_{xy} - 1} \exp \left\{ -(\lambda_{xy} x)^{\gamma_{xy}} - (\lambda_{xy} y)^{\gamma_{xy}} \right\} - (\gamma_{xy}) (w \lambda_{xy})^{\gamma_{xy}} (x)^{\gamma_{xy} - 1} \exp \left\{ -(\lambda_{xy} x)^{\gamma_{xy}} \right\} \, dx = w^{\gamma_{xy}} \exp \left\{ -(\lambda_{xy} x)^{\gamma_{xy}} - (\lambda_{xy} y)^{\gamma_{xy}} \right\} - w^{\gamma_{xy}} \exp \left\{ -(\lambda_{xy} x)^{\gamma_{xy}} \right\} \bigg|_0^t = w^{\gamma_{xy}} \exp \left\{ -(\lambda_{xy} x)^{\gamma_{xy}} - (\lambda_{xy} y)^{\gamma_{xy}} \right\} - w^{\gamma_{xy}} \exp \left\{ -(\lambda_{xy} x)^{\gamma_{xy}} \right\} - w^{\gamma_{xy}} \exp \left\{ -(\lambda_{xy} y)^{\gamma_{xy}} \right\} + w^{\gamma_{xy}} = F_{tu}(w, t, u)$

Is Equation (7.22) a valid cumulative distribution function?

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

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

$$w^{\gamma_{xy}} = \sum_{i=1}^n w'_i^{\gamma_{xy}} = \sum_{i=1}^n \frac{w_i}{\sum_{i=1}^n w_i} = 1.$$
7.4 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 \(\gamma_x\) and \(\gamma_y\) 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.

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

[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.

\[
\log t_x = \sum_{i=1}^{n} \log \left[ \frac{1}{\lambda_x} \right] + \frac{1}{\gamma_x} \log \left[ \frac{1}{1 - F_x(t_i)} \right]. \tag{7.23}
\]

\[
\log u_y = \sum_{i=1}^{n} \log \left[ \frac{1}{\lambda_y} \right] + \frac{1}{\gamma_y} \log \left[ \frac{1}{1 - F_u(u_i)} \right]. \tag{7.24}
\]

\[
\log t_{xy} = \sum_{i=1}^{n} \log \left[ \frac{1}{\lambda_{xy}} \right] + \frac{1}{\gamma_{xy}} \log \left[ \frac{1}{1 - F_{tu}(t_i, u_i)} \right]. \tag{7.25}
\]

The two ellipses differ (Exponential and Weibull) differ between the sums of squares of the original data \((w_ix, w_iy)\) 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 the weighted longitude estimates. The estimators for \(\gamma_x\), \(\gamma_y\), and \(\gamma_{xy}\) appear in Equations (7.8), (7.17) and (7.21). We had to perform Secant iterations to find these. The \(\lambda's\) are already weighted. Therefore, the term \(w_i\) is absent from the formulas. The estimators for \(\lambda_x\), \(\lambda_y\), and \(\lambda_{xy}\) 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.

7.4.1 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,
devotional ellipse in Equation (7.26).

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

(7.26)

This textbook provides an Excel 2010 Visual Basic for Applications (VBA) pro-
gram that calculates the weighted, Weibull ellipse. In addition to the angles of rota-
tion, it calculates the mean center, the area, the standard deviations, and the eccen-
tricity.

7.5 Ellipse Properties

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

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

(7.27)

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

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

(7.28)

where the subscript \(b\) denotes the angle of rotation \(\theta\) about the minor axis. Equation
(7.27) raises the weights \(w_i\) to the power \(\gamma_x\). 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 \(\gamma_y\).

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

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

(7.29)

Equation (7.30) gives the formula for calculating the semi-minor axis length \(b\).
\[ b^2 - \frac{\sum_{i=1}^{n} \log u_{y,i}^2}{n} = \] (7.30)

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

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 \( F \).

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

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

\[
e = \frac{\sqrt{a^2 - b^2}}{a}
\]

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

7.6 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)

- 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 \( \gamma_x \) or \( \gamma_y \). Be consistent when working with latitude and longitude.
Table 7.1  Results for the Weibull Model for Kentucky

<table>
<thead>
<tr>
<th>2003</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
</tr>
<tr>
<td>Axes</td>
</tr>
<tr>
<td>Area</td>
</tr>
<tr>
<td>Standard Deviations</td>
</tr>
<tr>
<td>Rotation</td>
</tr>
<tr>
<td>Orientation</td>
</tr>
<tr>
<td>Eccentricity</td>
</tr>
</tbody>
</table>

Table 7.2  Results for the Weibull Model for Kentucky

<table>
<thead>
<tr>
<th>2004</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
</tr>
<tr>
<td>Axes</td>
</tr>
<tr>
<td>Area</td>
</tr>
<tr>
<td>Standard Deviations</td>
</tr>
<tr>
<td>Rotation</td>
</tr>
<tr>
<td>Orientation</td>
</tr>
<tr>
<td>Eccentricity</td>
</tr>
</tbody>
</table>

- GAMMA2 — the second guess for γx or γy. 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 \( \lambda_x \), \( \lambda_y \), and \( \lambda_{xy} \). They will also give the difference between the last two iterations for convergence analysis.

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 $|\log L(\lambda, \gamma, k) - \log L(\lambda, \gamma, k - 1)| \leq \varepsilon |\log L(\lambda, \gamma, k)|$. Arbitrarily choose $\varepsilon = 10^{-6}$.

2. Choose two initial values for $\gamma$ 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 $L_XMinus$. Check if the two values are the same. If so, set the number of iterations to a lower number.

2. We must highlight the cells with the data in the spreadsheet before going into the VBA Editor.
7.7 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.

7.8 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 $\gamma_x$ and $\gamma_y$ 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

<table>
<thead>
<tr>
<th></th>
<th>2008</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
<td>(44.95865049, 173.1927484)</td>
</tr>
<tr>
<td>Axes</td>
<td>$a = 1.876920973, b = 1.48828381$</td>
</tr>
<tr>
<td>Area</td>
<td>8.777 sq. mi.</td>
</tr>
<tr>
<td>Standard Deviations</td>
<td>$\delta_x = 1.5549, \delta_y = 1.5947$</td>
</tr>
<tr>
<td>Rotation</td>
<td>X-Axis</td>
</tr>
<tr>
<td>Orientation</td>
<td>$\theta_x = 89.28^\circ, \theta_y = -0.72^\circ$</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.61</td>
</tr>
</tbody>
</table>

Table 7.6  Results for the Weibull Model on Gross Domestic Product

<table>
<thead>
<tr>
<th></th>
<th>2009</th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
<td>(45.10440668, 172.959405)</td>
</tr>
<tr>
<td>Axes</td>
<td>$a = 1.872381589, b = 1.493600132$</td>
</tr>
<tr>
<td>Area</td>
<td>8.786 sq. mi.</td>
</tr>
<tr>
<td>Standard Deviations</td>
<td>$\delta_x = 1.5588, \delta_y = 1.5997$</td>
</tr>
<tr>
<td>Rotation</td>
<td>X-Axis</td>
</tr>
<tr>
<td>Orientation</td>
<td>$\theta_x = 89.26^\circ, \theta_y = -0.74$</td>
</tr>
<tr>
<td>Eccentricity</td>
<td>0.60</td>
</tr>
</tbody>
</table>
7.9 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.

7.10 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.
Given the following summary tables, for each of the data sets, the standard deviational 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 | $\alpha = -1147830.91$, $\beta = 1087263.36$
KY 2004 Corn Acres | $\alpha = -1030547.49$, $\beta = 1020882.76$
KY 2003 Corn Acres | $\tilde{\alpha} = 2.40751$, $\tilde{\beta} = 0.32944$, $\tilde{\lambda}_{xy} = 0.57958$
KY 2004 Corn Acres | $\tilde{\alpha} = 2.45951$, $\tilde{\beta} = 0.33676$, $\tilde{\lambda}_{xy} = 0.59240$
KY 2003 Corn Acres | $\tilde{\lambda}_{x} = 2.40751$, $\tilde{\lambda}_{y} = 0.32944$, $\tilde{\lambda}_{xy} = 0.57958$
KY 2004 Corn Acres | $\tilde{\lambda}_{x} = 2.45951$, $\tilde{\lambda}_{y} = 0.33676$, $\tilde{\lambda}_{xy} = 0.59240$

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 $\alpha$, but a vastly different east-west parameter estimate $\beta$. Does this mean that the United States capital (money) went eastward, 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.

7.11 Exercises

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

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

\[
  \frac{n}{\lambda_x} = \bar{x} \text{ in Chapter 5.}
\]

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 \( w = \sum_{i=1}^{n} w_i' \).

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

3. Show that 

\[
  \frac{n}{\lambda_y} = \bar{y} \text{ in Chapter 5.}
\]

4. For what value of \( \gamma_x \) and \( \gamma_y \) in the Weibull ellipse makes it equal to the exponential ellipse [special case]? Show this algebraically.
7.12 VBA Code

The function \( \text{Log10}(X) \) 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 \texttt{Weibull()} subroutine (15) calculates the remaining measures for the Weibull ellipse. The first two \texttt{FOR-NEXT} loops (16, 17) 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 \texttt{secant1()} and \texttt{secant2()} only needed two iterations of a single weight initialized.

The \texttt{FOR-NEXT} loop (19) calculates the sums used in Equation (7.26), the axes lengths, the area, and the eccentricity.

The subroutine \texttt{Angle of Rotation()} determines which axis to rotate the ellipse. Since \( \theta \) from Equation (7.26) contains a plus and minus sign, both values are calculated. If \( \sum_{i=1}^{n} \log t_{x,i}^2 \geq \sum_{i=1}^{n} \log u_{y,i}^2 \), then rotate the ellipse about the \( Y \)-axis. Otherwise, rotate the ellipse about the \( X \)-axis.

The subroutine \texttt{Area()} calculates the sums used for the area. The program calculates the area, \( F \), in Equation (7.31) and saves it in the variable \( f \).

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 \texttt{FOR-NEXT} loop finish the calculations and save the results in the variables \texttt{error_x} and \texttt{error_y}.

The subroutine \texttt{Axes Length()} 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 using the subroutine \texttt{Eccentricity()}. If the axis length \( a \geq b \), then the formula in Equation (7.32) can be used directly used. 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 \texttt{WITH-END-WITH} (20) save the results to the spreadsheet called \texttt{STATS} in column L (column 12). We write appropriate text documentation to column K (column 11).
Sub secant1( iterations, variable, Gamma1, gamma2)
    j = 1
    dx = 1 'trivial value
    condition = 0 'trivial value
    While (j <= iterations) And (dx > condition)
        'set the weights
        n = Selection.Rows.Count
        sum_weight1 = 0
        sum_weight2 = 0
        For i = 2 To n
            sum_weight1 = sum_weight1 + (ActiveSheet.Cells(i, 3).Value)
            sum_weight2 = sum_weight2 + (ActiveSheet.Cells(i, 3).Value)
        Next
        ActiveSheet.Cells(1, 4).Value = "Lambda 1 Weights"
        ActiveSheet.Cells(1, 6).Value = "Lambda 2 Weights"
        For i = 2 To n
            ActiveSheet.Cells(i, 4).Value = ((ActiveSheet.Cells(i, 3).Value) / sum_weight1) ^ (1 / Gamma1) 'previous iteration
            ActiveSheet.Cells(i, 6).Value = ((ActiveSheet.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
            sumw_x2 = sumw_x2 + (ActiveSheet.Cells(i, 6) ^ gamma2 * ActiveSheet.Cells(i, variable).Value) ^ gamma2 'weighted
            sumw_x1 = sumw_x1 + (ActiveSheet.Cells(i, 4) ^ Gamma1 * ActiveSheet.Cells(i, variable).Value) ^ Gamma1 'weighted
            sum_x2 = sum_x2 + ActiveSheet.Cells(i, variable).Value ^ gamma2 'unweighted
            sum_x1 = sum_x1 + ActiveSheet.Cells(i, variable).Value ^ Gamma1 'unweighted
            sum_logx = sum_logx + Log(Abs(ActiveSheet.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 * (gamma2 - Gamma1) / (L - L Minus)
Gamma1 = gamma2 'correct assignment
gamma2 = Abs(gamma3) 'correct assignment, must be positive
dx = Abs(L - L Minus)
condition = 10 ^ (-6) * Abs(L)
j = j + 1
Wend

With Worksheets("STATS")
.Cells (1, 7).Value = "Secant Algorithm"
.Cells (2, 7).Value = "dx"
.Cells (2, 8).Value = dx
.Cells (2, 9).Value = condition
.Cells (3, 7).Value = "Lambda"
.Cells (3, 8).Value = lambda2
.Cells (4, 7).Value = "Gamma"
.Cells (4, 8).Value = gamma2
.Cells (5, 7).Value = "Mean"
.Cells (5, 8).Value = (n - 1) / lambda2
End With
End Sub
Sub secant2(iterations, Gamma1, gamma2)
    j = 1
    dx = 1 'trivial value
    condition = 0 'trivial value
    While (j <= iterations) And (dx > condition)
    'set the weights
    n = Selection.Rows.Count
    sum_weight1 = 0
    sum_weight2 = 0
    For i = 2 To n
        sum_weight1 = sum_weight1 + (ActiveSheet.Cells(i, 3).Value )
        sum_weight2 = sum_weight2 + (ActiveSheet.Cells(i, 3).Value )
    Next
    ActiveSheet.Cells(1, 4).Value = "Lambda 1 Weights"
    ActiveSheet.Cells(1, 6).Value = "Lambda 2 Weights"
    For i = 2 To n
        ActiveSheet.Cells(i, 4).Value = ((ActiveSheet.Cells(i, 3).Value ) / sum_weight1) ^ (1 / Gamma1) 'previous iteration
        ActiveSheet.Cells(i, 6).Value = ((ActiveSheet.Cells(i, 3).Value ) / sum_weight2) ^ (1 / gamma2) 'next iteration
    Next
    'end of setting the weights
    sumw_x2 = 0
    sumw_y2 = 0
    sumw_x1 = 0
    sumw_y1 = 0
    log_gamma1 = 0
    log_gamma2 = 0
    log_xy1 = 0
    log_xy2 = 0
    sum_lambda_x2 = 0
    sum_lambda_y2 = 0
    sum_lambda_x1 = 0
    sum_lambda_y1 = 0
    sumw_lambda2 = 0
    sumw_lambda1 = 0
    'calculate the sums used in the Secant Method
\[ \text{For } i = 2 \text{ To } n \\
\text{sum}\_\text{w}_x^2 = \text{sum}\_\text{w}_x^2 + (\text{ActiveSheet.Cells}(i, 6).Value \wedge \text{gamma}_2) \wedge \text{gamma}_2 \] \\
\text{sum}\_\text{w}_y^2 = \text{sum}\_\text{w}_y^2 + (\text{ActiveSheet.Cells}(i, 6).Value \wedge \text{gamma}_2) \wedge \text{gamma}_2 \\
\text{sum}_x = \text{sum}_x + (\text{ActiveSheet.Cells}(i, 4).Value \wedge \text{Gamma}_1) \wedge \text{Gamma}_1 \\
\text{sum}_y = \text{sum}_y + (\text{ActiveSheet.Cells}(i, 4).Value \wedge \text{Gamma}_1) \wedge \text{Gamma}_1 \\
\text{log}\_\text{gamma}_2 = \text{log}\_\text{gamma}_2 + 2 \cdot \text{Log}(\text{gamma}_2) \\
\text{log}\_\text{gamma}_1 = \text{log}\_\text{gamma}_1 + 2 \cdot \text{Log}(\text{Gamma}_1) \\
\text{log}_xy = \text{log}_xy + (\text{gamma}_2 - 1) \cdot \text{Log}(\text{ActiveSheet.Cells}(i, 1).Value \wedge \ABS(\text{ActiveSheet.Cells}(i, 2).Value)) \\
\text{log}_xy = \text{log}_xy + (\text{Gamma}_1 - 1) \cdot \text{Log}(\text{ActiveSheet.Cells}(i, 1).Value \wedge \ABS(\text{ActiveSheet.Cells}(i, 2).Value)) \\
\text{Next} \\
\text{lambda}_2 = (2 \cdot (n - 1) / (\text{sum}_x^2 + \text{sum}_y^2)) \wedge (1 / \text{gamma}_2) \] \\
\text{lambda}_1 = (2 \cdot (n - 1) / (\text{sum}_x + \text{sum}_y)) \wedge (1 / \text{Gamma}_1) \\
\text{calculate the sums involving lambda} \\
\text{For } i = 2 \text{ To } n \\
\text{sum}\_\text{lambda}_x^2 = \text{sum}\_\text{lambda}_x^2 + (\text{lambda}_2 \wedge \text{ActiveSheet.Cells}(i, 1).Value) \wedge \text{gamma}_2 \\
\text{sum}\_\text{lambda}_x = \text{sum}\_\text{lambda}_x + (\text{lambda}_1 \wedge \text{ActiveSheet.Cells}(i, 1).Value) \wedge \text{Gamma}_1 \\
\text{sum}\_\text{lambda}_y^2 = \text{sum}\_\text{lambda}_y^2 + (\text{lambda}_2 \wedge \text{ActiveSheet.Cells}(i, 2).Value) \wedge \text{gamma}_2 \\
\text{sum}\_\text{lambda}_y = \text{sum}\_\text{lambda}_y + (\text{lambda}_1 \wedge \text{ActiveSheet.Cells}(i, 2).Value) \wedge \text{Gamma}_1 \\
\text{sum}_\text{lambda}_2 = \text{sum}_\text{lambda}_2 + 2 \cdot (n - 1) \cdot \text{gamma}_2 \cdot \text{Log}(\text{ActiveSheet.Cells}(i, 6).Value \wedge \text{lambda}_2) \\
\text{sum}_\text{lambda}_1 = \text{sum}_\text{lambda}_1 + 2 \cdot (n - 1) \cdot \text{Gamma}_1 \cdot \text{Log}(\text{ActiveSheet.Cells}(i, 4).Value \wedge \text{lambda}_1) \\
\text{Next} \\
\text{‘estimate gamma using the Secant Method} \\
L_x = 0 \\
L_X\_\text{Minus} = 0 \\
\text{For } i = 2 \text{ To } n \\
L_x = L_x + \text{log}\_\text{gamma}_2 + \text{sum}_\text{lambda}_2 + \text{log}_xy - \text{sum}_\text{lambda}_x^2 - \text{sum}_\text{lambda}_y^2 \\
L_X\_\text{Minus} = L_X\_\text{Minus} + \text{log}\_\text{gamma}_1 + \text{sum}_\text{lambda}_1 + \text{log}_xy - \text{sum}_\text{lambda}_x - \text{sum}_\text{lambda}_y \\
\text{Next} \\
\text{gamma}_3 = \text{gamma}_2 - L_x \cdot (\text{gamma}_2 - \text{Gamma}_1) / (L_x - L_X\_\text{Minus}) \\
\text{Gamma}_1 = \text{gamma}_2 \ ‘\text{correct assignment} \\
\text{gamma}_2 = \text{ABS}(\text{gamma}_3) \ ‘\text{correct assignment, must be positive} \\
dx = \text{ABS}(L_x - L_X\_\text{Minus}) \\
\text{condition} = 10 \cdot (-6) \cdot \text{ABS}(L_x) \\
j = j + 1 \\
\text{Wend}
With Worksheets("STATS")
.Cells(1, 7).Value = "Secant Algorithm"
.Cells(2, 7).Value = "dx"
.Cells(2, 8).Value = dx
.Cells(2, 9).Value = condition
.Cells(3, 7).Value = "Lambda XY"
.Cells(3, 8).Value = lambda2
.Cells(4, 7).Value = "Gamma XY"
.Cells(4, 8).Value = gamma2
.Cells(5, 7).Value = "Mean XY"
.Cells(5, 8).Value = (n - 1) / lambda2
End With

Sub Weibull(gammaₓ, gammaᵧ, gammaₓᵧ, lambdaₓ, lambdaᵧ, lambdaₓᵧ)
' set the weights
n = Selection.Rows.Count
m = n - 1
sum_weight1 = 0
sum_weight2 = 0
sum_weight3 = 0
For i = 2 To n
    sum_weight1 = sum_weight1 + (ActiveSheet.Cells(i, 3).Value) ' single x
    sum_weight2 = sum_weight2 + (ActiveSheet.Cells(i, 3).Value) ' single y
    sum_weight3 = sum_weight3 + (ActiveSheet.Cells(i, 3).Value) ' joint xy
Next
ActiveSheet.Cells(1, 4).Value = "Lambda X Weights"
ActiveSheet.Cells(1, 6).Value = "Lambda Y Weights"
ActiveSheet.Cells(1, 7).Value = "Lambda XY Weights"
For i = 2 To n
    ActiveSheet.Cells(i, 4).Value = (ActiveSheet.Cells(i, 3).Value) / sum_weight1) ^ (1 / gammaₓ) ' lambdaₓ
    ActiveSheet.Cells(i, 6).Value = (ActiveSheet.Cells(i, 3).Value) / sum_weight2) ^ (1 / gammaᵧ) ' lambdaᵧ
    ActiveSheet.Cells(i, 7).Value = (ActiveSheet.Cells(i, 3).Value) / sum_weight3) ^ (1 / gammaₓᵧ) ' lambdaₓᵧ
Next
' end of setting the weights
fₓ = 0
Fₓ = 0
Fₓᵧ = 0
Lₓᵧ = 0 ' used for the log L calculation
logtₓ = 0
logtᵧ = 0
logtₓᵧ = 0
logtₓₓᵧ = 0
logtₓᵧᵧ = 0
logtₓₓᵧᵧ = 0
For i = 2 To n
  \( f_x = \text{ActiveSheet.Cells(i, 4).Value}^\gamma_x - \text{ActiveSheet.Cells(i, 4).Value}^\gamma_x * \exp(-1 * \lambda_x * (\text{ActiveSheet.Cells(i, 1).Value})^\gamma_x) \)
  \( F_x = \text{ActiveSheet.Cells(i, 6).Value}^\gamma_x - \text{ActiveSheet.Cells(i, 6).Value}^\gamma_x * \exp(-1 * \lambda_x * (-\text{ActiveSheet.Cells(i, 2).Value})^\gamma_x) \)
  \( F_{xy} = \text{ActiveSheet.Cells(i, 7).Value}^\gamma_{xy} - \text{ActiveSheet.Cells(i, 7).Value}^\gamma_{xy} * \exp(-1 * \lambda_{xy} * ((\text{ActiveSheet.Cells(i, 1).Value})^\gamma_{xy}) - \lambda_{xy} * (\text{ActiveSheet.Cells(i, 2).Value})^\gamma_{xy}) \)
  \( \log t_x = \log t_x + (\log(1 / (1 - f_x))) \)
  \( \log t_y = \log t_y + (\log(1 / (1 - F_y))) \)
  \( \log t_{xy} = \log t_{xy} + (\log(1 / (1 - F_{xy}))) \)
  \( \log t_{xsqr} = \log t_{xsqr} + ((m * (\log(1 / \lambda_x) + (1 / \gamma_x) * \log(1 / (1 - f_x)))) \)
  \( \log t_{ysqr} = \log t_{ysqr} + ((m * (\log(1 / \lambda_y) + (1 / \gamma_y) * \log(1 / (1 - F_y)))) \)
  \( \log t_{xysqr} = \log t_{xysqr} + ((m * (\log(1 / \lambda_{xy}) + (1 / \gamma_{xy}) * \log(1 / (1 - F_{xy})))) \)
  \( \log t_{xsq} = \log t_{xsq} + ((m * (\log(1 / \lambda_x) + (1 / \gamma_x) * \log(1 / (1 - f_x)))) \)
  \( \log t_{ysq} = \log t_{ysq} + ((m * (\log(1 / \lambda_y) + (1 / \gamma_y) * \log(1 / (1 - F_y)))) \)
  \( \log t_{xysq} = \log t_{xysq} + ((m * (\log(1 / \lambda_{xy}) + (1 / \gamma_{xy}) * \log(1 / (1 - F_{xy})))) \)
  \( \log t_{xsqr} = \log t_{xsqr} + ((m * (\log(1 / \lambda_x) + (1 / \gamma_x) * \log(1 / (1 - f_x)))) \)
  \( \log t_{ysqr} = \log t_{ysqr} + ((m * (\log(1 / \lambda_y) + (1 / \gamma_y) * \log(1 / (1 - F_y)))) \)
  \( \log t_{xysqr} = \log t_{xysqr} + ((m * (\log(1 / \lambda_{xy}) + (1 / \gamma_{xy}) * \log(1 / (1 - F_{xy})))) \)
  \( L_{xy} = L_{xy} + 2 * \log(\gamma_{xy}) + 2 * (n - 1) * \gamma_{xy} * \log(\lambda_{xy}) \)
  \( \text{Call Angle_of_Rotation}(\log t_{xsqr}, \log t_{ysqr}, \log t_{xy}, \text{note, atheta, itheta}) \)
  \( \text{call the deltas} \)
  \( \delta_{\gamma_x} = 0 \)
  \( \delta_{\gamma_y} = 0 \)
  \( eF_{\gamma_x} = 0 \)
  \( eF_{\gamma_y} = 0 \)
Next
For $i = 2$ To $n$

eFx = ActiveSheet.Cells(i, 4).Value ^ gamma_x - ActiveSheet.Cells(i, 4).Value ^ gamma_x * EXP(-1 * lambda_x * (ActiveSheet.Cells(i, 1).Value))
eFy = ActiveSheet.Cells(i, 6).Value ^ gamma_y - ActiveSheet.Cells(i, 6).Value ^ gamma_y * EXP(-1 * lambda_y * Abs(ActiveSheet.Cells(i, 2).Value))
elogt_x = Log10(1 / lambda_x) + (1 / gamma_x) * Log10(Log(1 / (1 - eFx)))
elogt_y = Log10(1 / lambda_y) + (1 / gamma_y) * Log10(Log(1 / (1 - eFy)))
delta_x = delta_x + (elogt_y * Sin(atheta / 57.2957795) - elogt_x * Cos(atheta / 57.2957795)) ^ 2 * _
ActiveSheet.Cells(i, 4).Value ^ gamma_x
delta_y = delta_y + (elogt_y * Cos(itheta / 57.2957795) - elogt_x * Sin(itheta / 57.2957795)) ^ 2 * _
ActiveSheet.Cells(i, 6).Value ^ gamma_y
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)
'write the stats to the STATS spreadsheet

With Worksheets("STATS")
.Cells(1, 11).Value = "Weibull Ellipse Statistics"
.Cells(2, 11).Value = "Mean Latitude"
.Cells(3, 11).Value = "Mean Longitude"
.Cells(2, 12).Value = (n - 1) / lambda_x
.Cells(3, 12).Value = (n - 1) / lambda_y
.Cells(4, 11).Value = "Rotation"
.Cells(5, 11).Value = "Major Axis Rotation"
.Cells(6, 11).Value = "Minor Axis Rotation"
.Cells(4, 12).Value = note
.Cells(5, 12).Value = atheta
.Cells(6, 12).Value = itheta
.Cells(7, 11).Value = "Delta X"
.Cells(8, 11).Value = "Delta Y"
.Cells(7, 12).Value = error_x
.Cells(8, 12).Value = error_y
.Cells(9, 11).Value = "Area (F)"
.Cells(9, 12).Value = f
.Cells(10, 11).Value = "Major Axis Length (a)"
.Cells(11, 11).Value = "Minor Axis Length (b)"
.Cells(10, 12).Value = a
.Cells(11, 12).Value = b
.Cells(12, 11).Value = "Area Check"
.Cells(13, 11).Value = "Eccentricity (e)"
.Cells(14, 11).Value = "Likelihood Function L(x,y)"
.Cells(12, 12).Value = Area2
.Cells(13, 12).Value = e
.Cells(14, 12).Value = L_xy
End With
End Sub
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 absence 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 multi-variate 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 sins 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 \(\phi_i\) is the vertical line running north and south parallel of the Prime Meridian and has the range \([-90^\circ, 90^\circ]\). The longitude \(\theta_i\) 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 \(0^\circ\). We define the Prime Meridian as having longitude of \(0^\circ\). 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](image.png)

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 \((\rho, \psi)\) where \(\rho\) is the polar
distance and $\psi$ 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 $(\bar{\ell}_0, \bar{m}_0, \bar{n}_0)$. 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).

$$
n_0 = \sum_{i=1}^{n} n_i = \sum_{i=1}^{n} \cos \phi_i.
$$

(8.1)

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

$$
\bar{X} = \bar{n}_0 = \frac{1}{n} n_0
$$

(8.2)

where we define $R$ next.

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

(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 $\theta$. It has a right angle (90°), the angle $\theta$ identified, the hypotenuse $\bar{R}$ 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 $W$.

$$
\bar{X}_w = \bar{n}_{w0} = \sum_{i=1}^{n} \frac{w_i m_i}{\sum_{i=1}^{n} w_i} = \sum_{i=1}^{n} \frac{w_i \cos \phi_i}{\sum_{i=1}^{n} w_i}.
$$

(8.4)

Accordingly, Equation (8.5) gives the weighted resultant length $\bar{R}$. 
\[ R_w = \sqrt{\left( \frac{\sum_{i=1}^{n} w_i l_i}{\sum_{i=1}^{n} w_i} \right)^2 + \left( \frac{\sum_{i=1}^{n} w_i m_i}{\sum_{i=1}^{n} w_i} \right)^2 + \left( \frac{\sum_{i=1}^{n} w_i n_i}{\sum_{i=1}^{n} w_i} \right)^2}, \quad 0 < R_w \leq 1 \] (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 \((\phi_i)\) and the longitude \((\theta_i)\) 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).

\[
l_0 = \sum_{i=1}^{n} l_i = \sum_{i=1}^{n} \sin \phi_i \cos \theta_i, \tag{8.6}
\]

\[
m_0 = \sum_{i=1}^{n} m_i = \sum_{i=1}^{n} \sin \phi_i \sin \theta_i. \tag{8.7}
\]

\(P(\phi_i, \theta_i)\) is a point on the Earth. \(\phi_i\) is called the latitude and \(\theta_i\) 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 \(m_0\).

\[
\bar{Y} = \bar{l}_0 = \frac{1}{n} l_0. \tag{8.8}
\]

\[
\bar{Z} = \bar{m}_0 = \frac{1}{n} m_0. \tag{8.9}
\]

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

\[
\bar{Y}_w = \bar{l}_{w_0} = \frac{\sum_{i=1}^{n} w_i \sin \phi_i \cos \theta_i}{\sum_{i=1}^{n} w_i}. \tag{8.10}
\]

\[
\bar{Z}_w = \bar{m}_{w_0} = \frac{\sum_{i=1}^{n} w_i \sin \phi_i \sin \theta_i}{\sum_{i=1}^{n} w_i}. \tag{8.11}
\]

#### 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.

\[
S^* = 2(1 - \bar{R}), \quad 0 < S^* < 1 \tag{8.12}
\]
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.

$$S^* = 1 - \bar{R}^2, \quad 0 < S^* < 1.$$  

(8.13)

8.5 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:

$$B = nI - T =
\begin{bmatrix}
    n - \sum_{i=1}^{n} l_i^2 & -\sum_{i=1}^{n} l_i m_i & -\sum_{i=1}^{n} l_i n_i \\
    -\sum_{i=1}^{n} l_i m_i & n - \sum_{i=1}^{n} m_i^2 & -\sum_{i=1}^{n} m_i n_i \\
    -\sum_{i=1}^{n} l_i n_i & -\sum_{i=1}^{n} m_i n_i & n - \sum_{i=1}^{n} n_i^2
\end{bmatrix}
\begin{pmatrix}
    b_1 \\
    b_2 \\
    b_3
\end{pmatrix}$$

We find the eigenvalues of matrix $B$ to determine the visual shape of the data. We define the vector $b_1$ as the first row times the vector of unknowns:

$$b_1 =
\begin{bmatrix}
    n - \sum_{i=1}^{n} l_i^2 & -\sum_{i=1}^{n} l_i m_i & -\sum_{i=1}^{n} l_i n_i \\
    -\sum_{i=1}^{n} l_i m_i & n - \sum_{i=1}^{n} m_i^2 & -\sum_{i=1}^{n} m_i n_i \\
    -\sum_{i=1}^{n} l_i n_i & -\sum_{i=1}^{n} m_i n_i & n - \sum_{i=1}^{n} n_i^2
\end{bmatrix}
\begin{pmatrix}
    b_1 \\
    b_2 \\
    b_3
\end{pmatrix} =
\begin{pmatrix}
    b_1(n - \sum_{i=1}^{n} l_i^2) \\
    -b_2 \sum_{i=1}^{n} l_i m_i \\
    -b_3 \sum_{i=1}^{n} l_i n_i
\end{pmatrix}$$

Vector $b_2$ corresponds to row 2 and vector $b_3$ corresponds to row 3.
Finding the eigenvalues of $B$ involves two steps:

1. Find the roots $\lambda$ to the determinant $\text{Det}(B - \lambda I)$ where the matrix $I$ is the $3 \times 3$ identity matrix. There may be at most three unique roots to the determinant.

2. For each eigenvalue $\lambda_j$, solve the system $(B - \lambda_j I)v = 0$.

Equation (8.15) gives the determinant for $B - \lambda_j I$.

$$\text{Det}(B - \lambda_j I) =$$

$$\left(n - \sum_{i=1}^{n} l_i^2 - \lambda_j\right) \left(n - \sum_{i=1}^{n} m_i^2 - \lambda_j\right) \left(n - \sum_{i=1}^{n} n_i^2 - \lambda_j\right) -$$

$$\sum_{i=1}^{n} l_im_i \sum_{i=1}^{n} m_i n_i \sum_{i=1}^{n} l_i n_i - \sum_{i=1}^{n} l_in_i \sum_{i=1}^{n} l_i m_i \sum_{i=1}^{n} m_in_i +$$

$$\sum_{i=1}^{n} l_in_i \sum_{i=1}^{n} l_i n_i \left(n - \sum_{i=1}^{n} m_i^2 - \lambda_j\right) +$$

$$\sum_{i=1}^{n} m_in_i \sum_{i=1}^{n} m_i n_i \left(n - \sum_{i=1}^{n} l_i^2 - \lambda_j\right) +$$

$$\sum_{i=1}^{n} l_im_i \sum_{i=1}^{n} l_i m_i \left(n - \sum_{i=1}^{n} n_i^2 - \lambda_j\right).$$

Arrange the eigenvalues $\lambda_1, \lambda_2, \text{and} \lambda_3$ 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.

$\beta_i, i = 1, 2, 3$ 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.

$$\lambda_i = n - \beta_i, \quad i = 1, 2, 3, \quad \lambda_1 + \lambda_2 + \lambda_3 = n.$$

The terms large and small are subjective. The reader may ask whether there is a statistical test to differentiate between the values for $\lambda_i$? [Mardia (45), pages 276-283] shows such a test statistic. Let the null hypothesis be $H_0: \lambda_1 = \lambda_2 = \lambda_3 = \frac{4}{3}$, which states that the eigenvalues have equal values. The test statistic is $S_U$

$$S_U = \frac{15}{2n} \sum_{i=1}^{3} \left(\lambda_i - \frac{1}{3}n\right)^2 \sim \chi^2(5).$$  \hspace{1cm} \text{(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 $\lambda_i’s$. Other statistical tests test for girdle and rotational symmetry using the $\lambda_i’s$. 
Figure 8.2  This figure shows the diagram of a triangle with the quantities for the longitude $\theta$, $n_0$, and the resultant length $R$.

Table 8.1  Interpretation of the Eigenvalues

<table>
<thead>
<tr>
<th>Eigenvalues</th>
<th>Condition</th>
<th>Distribution</th>
<th>Build</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_1 \approx \lambda_2 \approx \lambda_3$</td>
<td>Uniform</td>
<td>Axes having no orientation</td>
<td></td>
</tr>
<tr>
<td>Small $\lambda_1$, $\lambda_2$, Large $\lambda_3$</td>
<td>$\lambda_1 \neq \lambda_2$ Unimodal for large $R$. Bimodal otherwise.</td>
<td>If unimodal, the concentration at end of the eigenvector $t_3$. Otherwise, concentration at both ends of the eigenvector $t_3$.</td>
<td></td>
</tr>
<tr>
<td>Small $\lambda_1$, $\lambda_2$, Large $\lambda_3$</td>
<td>$\lambda_1 \approx \lambda_2$ Unimodal for large $R$. Bipolar otherwise.</td>
<td>Rotational symmetry about $t_3$.</td>
<td></td>
</tr>
<tr>
<td>Small $\lambda_1$, Large $\lambda_2$, and $\lambda_3$.</td>
<td>$\lambda_2 \neq \lambda_3$ Girdle</td>
<td>Girdle plane spanned by the eigenvectors $t_2$ and $t_1$.</td>
<td></td>
</tr>
<tr>
<td>Small $\lambda_1$, Large $\lambda_2$, and $\lambda_3$.</td>
<td>$\lambda_2 \approx \lambda_3$ Symmetric girdle</td>
<td>Rotational symmetry about the eigenvector $t_3$.</td>
<td></td>
</tr>
</tbody>
</table>
### 8.5.1 Finding the Eigenvalues

This section presents a simple Newton algorithm for finding the eigenvalues \( \lambda_j \), \( j = 1, 2, 3 \). The Gaussian elimination algorithm must be run to find the eigenvectors for each \( \lambda_j \). The subroutine must be run multiple times. The 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 \( \lambda \):

\[
(a - \lambda)(b - \lambda)(c - \lambda) + dgh + efi - \]

\[
he(b - \lambda) - ig(a - \lambda) - fd(c - \lambda). \tag{8.17}
\]

The derivative of Equation (8.17) with respect to \( \lambda \) is given in Equation (8.18).

\[
-(b - \lambda)(c - \lambda) - (a - \lambda)(c - \lambda) - (a - \lambda)(b - \lambda) + eh + ig + fd. \tag{8.18}
\]

[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 \( \lambda_j \), \( j = 1, 2 \) or 3. The recursive definition of Newton’s method is

\[
\lambda_{k+1} = \lambda_k - \frac{f(\lambda_k)}{f'(\lambda_k)}
\]

where \( k = 0, 1, 2, \ldots \) 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 \( f'(\lambda_k) \).

### 8.5.2 Finding the Eigenvectors

This book comes with a Gaussian elimination algorithm that performs row reduction on a \( 3 \times 3 \) matrix. The subroutine only needs to be run once for each \( \lambda_j \), \( j = 1, 2, 3 \). 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 \( 3 \times 3 \) 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.
8.6 Kentucky Example

Figure 8.3 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 8.4 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 \( \lambda_3 \) can be deduced by the equation \( \lambda_3 = n - \lambda_1 - \lambda_2 \) where \( \lambda_1 \) and \( \lambda_2 \) are known eigenvalues that have been found using the `Newton()` subroutine. Using a value close to \( n \) is a good guess for \( \lambda_3 \). Here, there is no implied ordering between \( \lambda_1 \), \( \lambda_2 \), and \( \lambda_3 \). Table 8.2 gives the eigenvalues and eigenvectors for the Kentucky 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

<table>
<thead>
<tr>
<th>Value</th>
<th>Eigenvector ( \mathbf{t}_i )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \lambda_1 )</td>
<td>0.006629744 (0.734788415, 0.573334304, 1.0)</td>
</tr>
<tr>
<td>( \lambda_2 )</td>
<td>0.034874073 (0.944911442, -2.955186475, 1.0)</td>
</tr>
<tr>
<td>( \lambda_3 )</td>
<td>89.95849618 (-1.300507805, -0.077445098, 1.0)</td>
</tr>
</tbody>
</table>

The reader can verify that the property \( T \mathbf{u}^T = \lambda_i \mathbf{u}^T \) where the vector \( \mathbf{u} \) is the eigenvector corresponding to the eigenvalue \( \lambda_i \) for \( i = 1, 2, 3 \). Just looking at the magnitude of the values of \( \lambda_1 \), \( \lambda_2 \), and \( \lambda_3 \) 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
Table 8.3  Results for the Spherical Model for Kentucky

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Center</td>
<td>(45.35963106, 2.701085798, −34.87836501)</td>
</tr>
<tr>
<td>((l_0, m_0, n_0))</td>
<td>(4082.366795, 243.0977218, −3139.052851)</td>
</tr>
<tr>
<td>Resultant Length</td>
<td>(R = 0.999769376)</td>
</tr>
<tr>
<td>Sample Size</td>
<td>(n = 90)</td>
</tr>
<tr>
<td>Data Type</td>
<td>Directional</td>
</tr>
<tr>
<td>Data Shape</td>
<td>Bipolar</td>
</tr>
<tr>
<td>Spherical Variance</td>
<td>(S = 0.000461194)</td>
</tr>
</tbody>
</table>

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 `Spherical_DEV_Ellipse()` 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.
- 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 \(\lambda_3\) is large; \(\lambda_1\) and \(\lambda_2\) are small. The concentration of the spherical data is at both ends of the vector \(t_3\). We knew the data originated from a bipolar coordinate system with a north and south pole. The vector \(t_3\) in radians is \(t_3 = (-1.300507805, -0.077445098, 1.0)\).

8.7 Crime Example

We look at the histogram in Figure 8.5 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 8.6 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 matricies \(B\) and \(T\) are as follow:
Figure 8.5  This figure shows the histogram for the Violent Crime 2008 data. Three possible modes appear at California, Texas, and Florida.

Figure 8.6  This figure shows the 2008 US Crime converted data.

\[
B = \begin{pmatrix}
19.82660121 & 0.299933647 & 22.9389611 \\
0.299933647 & 47.65668036 & -0.234303397 \\
22.9389611 & -0.234303397 & 30.51671843
\end{pmatrix}
\]

\[
T = \begin{pmatrix}
29.17339879 & -0.299933647 & -22.9389611 \\
-0.299933647 & 1.343319644 & 0.234303397 \\
-22.9389611 & 0.234303397 & 18.48328157
\end{pmatrix}
\]
Table 8.4 gives the eigenvalues and eigenvectors.

<table>
<thead>
<tr>
<th>Diagonal Value</th>
<th>Eigenvector $t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_1$ 0.27486963</td>
<td>(0.793812871, 0.003545129, 1.0)</td>
</tr>
<tr>
<td>$\lambda_2$ 1.340181802</td>
<td>(-0.649158203, -136.7206467, 1.0)</td>
</tr>
<tr>
<td>$\lambda_3$ 47.38494857</td>
<td>(-1.259802117, 0.013295803, 1.0)</td>
</tr>
</tbody>
</table>

We test the hypothesis that the eigenvalues are equal versus they are not equal using Equation (8.16) using the test statistic $S_U$.

\[
S_U = \frac{15}{2(49)} \sum_{i=1}^{3} \left( \lambda_i - \frac{1}{3}(49) \right)^2 = \]

\[
\frac{15}{98} \left[ (0.27486963 - 16.33)^2 + (1.340181802 - 16.33)^2 + (47.38494857 - 16.33)^2 \right] = 221.46.
\]

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 $\lambda_i's$. Using the $\lambda_i's$ in Table 8.4 and the interpretation in Table 8.1, we most likely have a unimodal distribution because $\lambda_3$ is large; $\lambda_1$ and $\lambda_2$ 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 $t$ where $t_3 = (-1.259802117, 0.013295803, 1.0)$.

<table>
<thead>
<tr>
<th>Center</th>
<th>(44.11484743, -0.535389408, -35.03421925)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$(l_0, m_0, n_0)$</td>
<td>(2161.627524, -26.23408097, -1716.676743)</td>
</tr>
<tr>
<td>Resultant Length</td>
<td>$R = 0.983257676$</td>
</tr>
<tr>
<td>Sample Size</td>
<td>$n = 48$</td>
</tr>
<tr>
<td>Data Type</td>
<td>Directional</td>
</tr>
<tr>
<td>Data Shape</td>
<td>Unimodal</td>
</tr>
<tr>
<td>Spherical Variance</td>
<td>$S = 0.033204342$</td>
</tr>
</tbody>
</table>
Figure 8.7  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 8.8  This figure shows the 2008 Gross Domestic Product converted data.

8.8 GDP Example

We begin by looking at the histogram of the observations in Figure 8.7. 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 8.8 shows the converted data. The data is spread out into the eight quadrants of the $X$, $Y$, and $Z$ planes.
Table 8.6  Eigenvalues and Eigenvectors for the 2008 Gross Domestic Product

<table>
<thead>
<tr>
<th>Diagonal Value</th>
<th>Eigenvector $\mathbf{t}'$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\lambda_1$</td>
<td>3.728832776, (-0.187814653, 0.00927436, 1.0)</td>
</tr>
<tr>
<td>$\lambda_2$</td>
<td>6.216179462, (5.008334449, -6.400592894, 1.0)</td>
</tr>
<tr>
<td>$\lambda_3$</td>
<td>29.05498776, (5.546422255, 4.496198733, 1.0)</td>
</tr>
</tbody>
</table>

The matrices $B$ and $T$ for the GDP 2008 data is

$$B = \begin{pmatrix} 19.35173834 & -10.96157833 \ & -2.888240385 \ -10.96157833 & 23.90143271 \ & -1.953298019 \ -2.888240385 & -1.953298019 \ & 34.74682895 \ \end{pmatrix}$$

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 $S_U$.

$$S_U = \frac{15}{2(39)} \sum_{i=1}^{3} \left( \lambda_i - \frac{1}{3}(39) \right)^2 = \frac{15}{78} \left[ (3.728832776 - 13)^2 + (6.216179462 - 13)^2 + (29.05498776 - 13)^2 \right] = 74.95.$$

We reject the null hypothesis $H_0: \lambda_i = \frac{4}{3}, i = 1, 2, 3$ 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 $\lambda_3$ is larger relative to $\lambda_2$ and $\lambda_3$, and since $R$ is not close to 1, this may be a bimodal distribution. The concentration is at both ends of the vector $\mathbf{t}_3 = (5.546422255, 4.496198733, 1.0)$.  


Table 8.7  Results for the Spherical Model for GPD

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>2008</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Center</strong></td>
<td>(27.14815799, 24.99879834, 6.46609507)</td>
</tr>
<tr>
<td>((l_0, m_0, n_0))</td>
<td>(1058.778161, 974.9531353, 252.1777077)</td>
</tr>
<tr>
<td><strong>Resultant Length</strong></td>
<td>(R = 0.653921742)</td>
</tr>
<tr>
<td><strong>Sample Size</strong></td>
<td>(n = 39)</td>
</tr>
<tr>
<td><strong>Data Type</strong></td>
<td>Directional</td>
</tr>
<tr>
<td><strong>Data Shape</strong></td>
<td>Bimodal</td>
</tr>
<tr>
<td><strong>Spherical Variance</strong></td>
<td>(S = 0.572386356)</td>
</tr>
</tbody>
</table>
8.9 VBA Code for Spherical Statistics

Dim n As Integer 'number of observations
Dim l0, m0, n0, l0sqrd, m0sqrd, n0sqrd, lmsum, lnsum, mnsum As Double
Dim lsum, msum, nsum As Double
Dim R_length, R_length2 As Double
Dim bar_x, bar_y1, bar_y2 As Double
Dim TempSecSin As Double
Dim SumWeights As Double 'sum of the weight column
Dim R_zPrime As Double 'test statistic for the resultant R
Dim F_theory As Double

Sub SphereSetWeights(A Var)
    n = Selection.Rows.Count
    'title the columns
    With ActiveSheet
        .Cells (1, 4) = "Latitude (Radians)"
        .Cells (1, 5) = "Abs Longitude (Radians)"
        .Cells (1, 6) = "Sin Latitude"
        .Cells (1, 7) = "Cos Latitude"
        .Cells (1, 8) = "Sin Longitude"
        .Cells (1, 9) = "l"
        .Cells (1, 10) = "m"
        .Cells (1, 11) = "n"
    End With

    SumWeights = 0
    'sum the weight column
    For i = 2 To n
        SumWeights = SumWeights + ActiveSheet.Cells(i, 3).Value 'unweighted calculations
    Next

    For i = 2 To n
        With ActiveSheet
            .Cells (i, 4) = Abs(ActiveSheet.Cells(i, 2).Value) * WorksheetFunction.Pi() / 180 'convert longitude to radians
            .Cells (i, 5) = Abs(ActiveSheet.Cells(i, 1).Value) * WorksheetFunction.Pi() / 180 'convert latitude to radians
            .Cells (i, 6) = Sin(ActiveSheet.Cells(i, 4).Value) 'sin of longitude in radians
            .Cells (i, 7) = Cos(ActiveSheet.Cells(i, 4).Value) 'cos of longitude in radians
            .Cells (i, 8) = Sin(ActiveSheet.Cells(i, 5).Value) 'sin of latitude in radians
            .Cells (i, 9) = ActiveSheet.Cells(i, 8).Value * ActiveSheet.Cells(i, 7).Value 'l_i
            .Cells (i, 10) = ActiveSheet.Cells(i, 8).Value * ActiveSheet.Cells(i, 6).Value 'm_i
            .Cells (i, 11) = Cos(ActiveSheet.Cells(i, 5).Value) 'n_i
        End With
    Next
End Sub
Sub SphericalDevEllipse()
Dim trace As Double 'trace of the matrix T; use this instead of n.
Dim AVar As Integer 'hides procedure names in pick list
Call SphereSetWeights(AVar)
Call Calculate_B(l0, m0, n0, l0_sqrd, lm_sum, ln_sum, m0_sum, mn_sum, n0_sqrd)
Call Calculate_T(l0, m0, n0, l0_sqrd, m0_sqrd, n0_sqrd, lm_sum, ln_sum, mn_sum)
R_length = Sqr(l0^2 + m0^2 + n0^2)
S = (n - 1 - R_length) / (n - 1) 'spherical variance
l_sum = l0 / R_length
m_sum = m0 / R_length
n_sum = n0 / R_length
bar_x = WorksheetFunction.Acos(n_sum) 'longitude mean
TempSecSin = Sin(bar_x)
bar_y1 = WorksheetFunction.Acos(l_sum / TempSecSin) 'latitude mean
bar_y2 = WorksheetFunction.Asin(m_sum / TempSecSin) 'latitude mean
' statistical testing
bar_degrees = bar_x
' 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 = "l0 (degrees)"
.Cells(6, 14).Value = "m0 (degrees)"
.Cells(7, 14).Value = "n0 (degrees)"
.Cells(8, 14).Value = "Spherical Variance"
.Cells(9, 14).Value = "Resultant Length (R)"
.Cells(18, 14).Value = "Matrix T"
.Cells(2, 15).Value = bar_x * 180 / WorksheetFunction.Pi()
.Cells(6, 15).Value = m_sum * 180 / WorksheetFunction.Pi()
.Cells(8, 15).Value = S
.Cells(9, 15).Value = R_length
' matrix T
.Cells (19, 15).Value = l0^2
.Cells (19, 16).Value = lm_sum
.Cells (19, 17).Value = ln_sum
.Cells (20, 16).Value = m0^2
.Cells (20, 15).Value = lm_sum
.Cells (20, 17).Value = mn_sum
.Cells (21, 17).Value = n0^2
.Cells (21, 15).Value = ln_sum
.Cells (21, 16).Value = mn_sum
End With
End Sub
8.10 VBA Code for Numerical Techniques

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
        For j = 1 To n
            T_Prime(i, j) = Worksheets("STATS").Cells(i + 18, j + 14).Value
            If i = j Then
                T_Prime(i, j) = Worksheets("STATS").Cells(i + 18, j + 14).Value - Lambda
            End If
        Next
    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
        T_Prime(3, i) = T_Prime(3, i) - T_Prime(2, i)
    Next
reduce row 1
TPrime(1, 1) = TPrime(1, 1) / TPrime(1, 2)
TPrime(1, 3) = TPrime(1, 3) / TPrime(1, 2)
TPrime(1, 2) = 1
'subtract row 2 and row 1 to give row 1
For i = 1 To 3
TPrime(1, i) = TPrime(1, i) - TPrime(2, i)
Next
'end of subtraction and row reductions
Worksheets("STATS").Cells(26, 14).Value = "T minus Lambda (1) row reduced"
For i = 1 To n
For j = 1 To n
Worksheets("STATS").Cells(25 + i, 14 + j).Value = TPrime(i, j)
Next
Next ‘i
Worksheets("STATS").Cells(30, 15).Value = "The reader will have to determine the eigenvector from here."
End Sub

Function fx(Lambda)
'the determinate of the matrix T
Dim a, b, c, d, e, g As Double
'matrix T is symmetrical, not all of it needs to be read in.
a = Worksheets("STATS").Cells(19, 15).Value ’sum l^2
\ i\n d = Worksheets("STATS").Cells(19, 16).Value ’sum l m^\ i
\ i\n e = Worksheets("STATS").Cells(19, 17).Value ’sum l n^\ i
\ i\n b = Worksheets("STATS").Cells(20, 16).Value ’sum m^2
\ i\n g = Worksheets("STATS").Cells(20, 17).Value ’sum m n^2
\ i\n c = Worksheets("STATS").Cells(21, 17).Value ’sum n^2
\ i\n f = d
i = g
h = e
'fxomega = (a - Lambda) * (d - Lambda) * (g - Lambda) + 2 * b * c * d -
c * 2 * (d - Lambda) - e * 2 * (a - Lambda) - b * 2 * (g - Lambda)
f = (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
a = Worksheets("STATS").Cells(19, 15).Value ' sum i * i
b = Worksheets("STATS").Cells(20, 16).Value ' sum m * i
f = d
i = g
g = Worksheets("STATS").Cells(21, 17).Value ' sum n * i
h = e
18 gx = -1 * (b - Lambda) * (c - Lambda) - (a - Lambda) * (c - Lambda) - (a - Lambda) -
(b - Lambda) + c * h + i * g + f * d
' gx = -1 * a * d + a * g + 2 * Lambda * a - d * g + 2 * Lambda * d - 2 * Lambda * g +
3 * Lambda * 2 + c * 2 + e * 2 + b * 2
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
\[ f_x = f(x) \]
\[ \text{While } (\text{diff} > 0.0000000001) \]
\[ \lambda_0 = \lambda \]
\[ \lambda = \lambda - \frac{f_x}{g_x} \]
\[ \text{diff} = |\lambda_0 - \lambda| \]
\[ i = i + 1 \]
\[ \text{Wend} \]
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
The code for calculating the weighted standard deviational ellipse resides in the Developer Environment under the module named \texttt{Ch4StandardDeviationalEllipse}. In the Excel spreadsheet environment, only highlight the observations themselves using the \texttt{<shift>↓} key combination, not the entire column by clicking on the column headings. It is important that the reader only use columns 1 – 3 for his or her data. The VBA code in module \texttt{Ch4StandardDeviationalEllipse} uses the driver \texttt{StandardDev_Ellipse()} to calculate the following statistics.

- The weighted mean center \((\bar{x}, \bar{y})\) from Equation (5.9).
- The angle of rotation on the axes \(\theta_x\) and \(\theta_y\) from Equation (5.10).
- The error terms on the axes \(\delta_x\) and \(\delta_y\) from Equations (5.11) and (5.12).
- The semi-major \(a\) and semi-minor \(b\) axes lengths from Equations (5.15) and (5.16).
The area $F$ from Equation (5.17).

The eccentricity of the ellipse $e$ from Equation (5.18).

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 $<\text{SHIFT}>$ key, arrow over to Column C. The headings of the data should be highlighted and selected.
   
   (c) While still holding down the $<\text{SHIFT}>$ key, hold down the $<\text{CRTL}>$ 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 $Ch4\text{StandardDeviationalEllipse}$. To enter the VBA environment and run the standard deviational ellipse code:

1. Click on the Developer ribbon.

2. Click on the Visual Basic button on the left side.

3. Click on $Ch4\text{StandardDeviationalEllipse}$ on the left-side pane.

This module calls the following subroutines.

- $\text{Set_Weights}(\text{sum_weight})$ — This subroutine calculates the weights and saves them to the spreadsheet in the fourth column.

- $\text{SDE_Stats}(\text{AVar})$ — This subroutine calculates the following statistics:
  
  - The mean latitude, $\bar{x}$, and the mean longitude, $\bar{y}$.
  
  - The major axis rotation and minor axis rotation. The sum of the absolute value of both will always sum to $90^\circ$.
  
  - The $\delta_x$ for the latitude and the $\delta_y$ for the longitude.
  
  - The major axis length, minor axis length, the area, and the eccentricity.

- $\text{SDE_Likelihood}(\text{AVar})$ — This subroutine calculates the log likelihood function for the weighted regression model.

- $\text{Put_SDEOutput_In_Spreadsheet}(\text{AVar})$ — This subroutine saves the results to the STATS spreadsheet in columns A and B.
In Excel, double click on the module named Ch4StandardDeviationalEllipse on the left-hand pane circled in red to display the VBA code for the Standard Deviational Ellipse. The subroutine called Standard Dev Ellipse() runs the programs to calculate the statistics in Chapter 5.

This figure shows the VBA environment for the module named Ch4StandardDeviationalEllipse. It also highlights the driver subroutine Standard Dev Ellipse().
Figure A.3 This figure shows the STATS worksheet and the results from the Standard_Dev_Ellipse() subroutine.

The Standard_Dev_Ellipse() subroutine (1) drives the standard deviational ellipse subroutines. The weighted standard deviational ellipse is one of the simplest concepts to implement, and thus, does not require input from the user. A circular coordinate system is not required, either. The Standard_Dev_Ellipse() routine will return the same results with either data set.

To run the Standard_Dev_Ellipse() 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 Standard_Dev_Ellipse() subroutine run quickly. As stated several times, the results will appear in the STATS spreadsheet.

Upon clicking on the STATS tab, the Standard_Dev_Ellipse() results appear in columns A and B. See Figure A.3. The SDE statistics and their locations in the STATS spreadsheet are as follow:
Dim n, m As Integer  'sample size
Dim sum_weight As Double  'sum of the weights
Dim mean_latitude, mean_longitude As Double  'mean latitude and longitude
Dim atheta, itheta As Double  'axes rotations
Dim x, y, xy As Double  'intermediate calculations used to calculate the area F
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  'exponential distribution ellipse variables
Dim a, b As Double  'semi-major and semi-minor axes
Dim note As String  'note on axis of rotation of ellipse
Dim delta_x, delta_y As Double  'deltas on the x and y axis
Dim f As Double  'area
Dim e As Double  'eccentricity
Dim area_check2 As Double  'area check
Dim L_reg As Double  'regression likelihood function value

Sub Standard_Dev_Ellipse()
    'run the routines that calculate the standard deviational ellipse statistics
    Dim AVar As Integer  'hides procedure names in pick list
    Call Set_Weights(sum_weight)
    Call SDE_Stats(AVar)
    Call SDE_Likelihood(AVar)
    Call Put_SDEOutput_In_Spreadsheet(AVar)
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 \(\theta\) appears in cell B5.
- The minor axis angle of rotation \(\theta\) 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 (5.8) appears in cell B14. This will be discussed in Section 7.10.

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 \( \text{mod}() \) 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 \( \text{mod}() \) 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

EXPONENTIAL VBA DRIVER

The module Ch5ExponentialDeviationalEllipse provides an Excel 2010 VBA program to calculate the statistics for the weighted, exponential deviational ellipse. The VBA code calculates the following statistics.

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 <CRTL> key.

   (d) Touch the down arrow key once. This should take you to the bottom of the data.
Figure B.1 This figure shows the VBA environment for the module Ch5ExponentialDeviationalEllipse. It also highlights the driver subroutine ExponentialDevEllipse().

To enter the VBA environment and run the exponential ellipse code:

1. Click on the Developer ribbon.
2. Click on the Visual Basic button on the left side.
3. Click on the module Ch5ExponentialDeviationalEllipse on the left-side pane.

This module calls the following subroutines.

- **Set_Weights(sum_weight)** — This subroutine calculates the weights and saves them to the active data spreadsheet.
- **EXP_Stats(AVar)** — This subroutine calculates the parameter estimates, the major axis length, minor axis length, the area, and the eccentricity. It also writes the log likelihood value to the STATS spreadsheet.
- **Put_EXPOutput_In_Spreadsheet(AVar)** — This subroutine puts the results of the calculates with labels for the weighted standard deviational ellipse to the STATS spreadsheet. The subroutine writes the results columns D and E.
Figure B.2 This figure shows the STATS worksheet and the results from the Exponential_Dev_Ellipse() subroutine.

The Exponential_Dev_Ellipse() subroutine drives the exponential deviational ellipse subroutines. The weighted exponential deviational ellipse is another one of the simplest concepts to implement, and thus, does not require input from the user. A circular coordinate system is required.

To run the Exponential_Dev_Ellipse() 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 Exponential_Dev_Ellipse() subroutine run quickly. As previous stated, the results will appear in the STATS spreadsheet. Upon clicking on the STATS tab, the Exponential_Dev_Ellipse() results appear in columns D and E. See Figure B.2. The Exponential_Dev_Ellipse() statistics and their locations in the STATS spreadsheet are as follow:

- Using Equations (6.12) and (6.13) for the weighted mean center \((\bar{x}, \bar{y})\), \(\bar{x}\) appears in cell E2, and \(\bar{y}\) appears in cell E3.
- Using Equation (6.17), the axis of rotation appears in cell E4.
- Using Equation (6.17), the major axis angle of rotation \(\theta\) appears in cell E5. The plus and minus operator gives two values.
Using Equation (6.17), the minor axis angle of rotation $\theta$ appears in cell E6. The plus and minus operator gives two values.

Using Equation (6.18), the standard deviation in the latitude direction $x$ appears in cell E7.

Using Equation (6.19), the standard deviation in the longitude direction $y$ appears in cell E8.

Using Equation (6.20), the semi-major axis length $a$ appears in cell E10.

Using Equation (6.21), the semi-minor axis length $b$ appears in cell E11.

Using Equation (6.22), the area $F$ of the weighted exponential deviational ellipse appears in cell E9.

The area check using the formula $F = \pi ab$ appears in cell E12.

Using Equation (6.23), the eccentricity $e$ of the exponential deviational ellipse appears in cell E13.

Using Equation (6.2), the logarithm of the likelihood function of the exponential model for the latitude $x$ appears in cell E14.

Using Equation (6.6), the logarithm of the likelihood function of the exponential model for the longitude $y$ appears in cell E15.

Using Equation (6.9), the logarithm of the likelihood function of the exponential 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.

Should the reader wish to calculate un-weighted exponential deviational ellipses, set the weight column to 1.0.
Sub Exponential

    Dim wx, wy As Double
    Dim lambda_x, lambda_y, lambda_xy As Double 'parameter estimates
    Dim x, y, xy, xsqr, ysqr, xysqr As Double 'intermediate calculations
    Dim sum_weight As Double 'sum of the weights
    Dim lambda, lambda_y, lambda_xy As Double 'angles of rotation
    Dim t_x, t_y, t_xy, t_xsqr, t_ysqr, t_xysqr As Double 'log likelihood functions
    Dim n, m As Integer
    Dim a, b As Double 'semi-major and semi-minor axes
    Dim note As String 'note on axis of rotation of ellipse
    Dim error_x, error_y As Double 'deviations on the x and y axis
    Dim f As Double 'area
    Dim e As Double 'eccentricity
    Dim area_check2 As Double 'area check

    Dim AVar As Integer 'hides procedure names in pick list
    Call Set_Weights(sum_weight)
    Call EXP_Stats(AVar)
    Call Put_ExpOutput_In_Spreadsheet(AVar)

End Sub
APPENDIX C
WEIBULL VBA DRIVER

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 the module Ch6 Weibull Deviational Ellipse. This module has five subroutines.

- **Weibull_DevelEllipse()** — This subroutine is the driver that calls the other subroutines.

- **Secant1()** — This subroutine calculates the point estimates for the latitude \(\hat{\lambda}_x\) and \(\hat{\gamma}_x\) from Equation (7.8). It can also be used to calculate the point estimates for the longitude \(\hat{\lambda}_y\), and \(\hat{\gamma}_y\) from Equation (7.17)
Figure C.1  This figure shows the VBA environment for the module Ch6 Weibull Deviational Ellipse. It also highlights the driver subroutine Weibull Dev Ellipse().

- **Secant2()** — This subroutine calculates the point estimates for the joint distribution $\hat{x}{y}$ and $\hat{y}{x}$ from Equation (7.21).
- **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 $\theta_x$ and $\theta_y$, the standard deviations on the axes $\delta_x$ and $\delta_y$, 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 $\gamma_x$ or $\gamma_y$.
- **Gamma2** — This parameter is the second initial guess at $\gamma_x$ or $\gamma_y$. Note that $\text{Gamma1} < \text{Gamma2}$.

The Secant1() algorithm returns the values $\hat{\gamma}_x$ and $\hat{\lambda}_x$, or $\hat{\gamma}_y$ and $\hat{\lambda}_y$, depending on if $\text{Variable} = 1$ or $\text{Variable} = 2$. All four point estimates have to be saved or recorded where they will not be over-written.
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 $\gamma_x$ or $\gamma_y$.
- **Gamma2** — This parameter is the second initial guess at $\gamma_x$ or $\gamma_y$. 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 pointestimates 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 from Equation (7.9).
- $\hat{\lambda}_y$ — The point estimate for the longitude for the scale parameter from Equation (7.18).
- $\hat{\lambda}_{xy}$ — 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:

- Using Equations (7.9) and (7.18), for the weighted mean center $(\bar{x}, \bar{y})$, $\bar{x}$ appears in cell L2, and $\bar{y}$ appears in cell L3.
- Using Equation (7.26), the axis of rotation appears in cell L4.
- Using Equation (7.26), the major axis angle of rotation $\theta$ appears in cell L5. The plus and minus sign in Equation (7.26) gives two values.
- Using Equation (7.26), the minor axis angle of rotation $\theta$ appears in cell L6. The plus and minus sign in Equation (7.26) gives two values.
- Using Equation (7.27), the standard deviation in the latitude direction $x$ appears in cell L7.
- Using Equation (7.28), the standard deviation in the longitude direction $y$ appears in cell L8.
Using Equation (7.31), the area $F$ of the weighted standard deviational ellipse appears in cell L9.

Using Equation (7.29), the semi-major axis length $a$ appears in cell L10.

Using Equation (7.30), the semi-minor axis length $b$ appears in cell L11.

The area check using the formula $F = \pi ab$ appears in cell L12.

Using Equation (7.32), 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.
Dim iterations As Integer
Dim Gamma1, gamma2, gamma3 As Double
Dim variable As Integer
Dim n, m As Integer
Dim logt_x, logt_y, logt_xy, logt_xsqr, logt_ysqr As Double 'intermediate calculations
Dim L_xy As Double 'likelihood function
Dim a, b As Double 'semi-major and semi-minor axes
Dim atheta, itheta As Double 'axes rotations
Dim note As String 'note on axis of rotation of ellipse

Sub Weibull_Init_Ellipse()
  '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
  'Call secant2(9, 1, 11)
  'new data set
  'Call Weibull(0.768572408, 0.762370962, 1, 2.407513111, 0.329444607, 0.579579441)
  'old data set
  '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
  '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
  '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)
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 $\gamma_x$, $\gamma_y$, and $\gamma_{xy}$. 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 Ch6WeibullDeviationalEllipse calculates the parameters $\lambda_x, \lambda_y, \gamma_x$ and $\gamma_y$. The program Weibull() in Ch6WeibullDeviationalEllipse calculates the parameters $\lambda_{xy}$ and $\gamma_{xy}$.

Double click on the module Ch6 Weibull Deviational Ellipse to display the VBA code for the Secant code and the Weibull Deviational Ellipse in Excel. The subroutines called Secant1(), Secant2(), and Weibull_Dev_Ellipse() 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.
The statistics associated with the spherical distribution in Chapter 8 are complicated. They usually begin with finding the some statistic of a $3 \times 3$ matrix from either $B$ or $T$. This Section describes the VBA code in Ch7NewtonGaussRoutines and Ch7 Spherical Statistics. It is best to start with Ch7 Spherical Statistics. Figure D.1 shows a screen capture of the VBA environment for Ch7 Spherical Statistics. The subroutines in Ch7SphericalStatistics 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. Ch7 Spherical Statistics contains the following six subroutines:

1. NewtonGauss() — This is a driver subroutine.

2. Spherical_Dev_Ellipse() — This is a driver subroutine.
Figure D.1 This figure shows the VBA environment for the module Ch7NewtonGaussRoutines. It also highlights the driver subroutine NewtonGauss().
3. Calculate $B()$ — This subroutine calculates the matrix $B$ from which the diagonal elements $\beta_1, \beta_2$ and $\beta_3$ are determined. We can re-use the subroutine in Ch7SphericalStatistics for calculating this matrix.

4. 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.

5. 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 $3 \times 3$ matrix, the code in the Gauss() subroutine performs row reduction as one would do on paper.

6. fx(Lambda) — This subroutine finds the determinant of matrix $T$ given the eigenvalue Lambda.

7. gx(Lambda) — This subroutine finds the derivative of the determinant of matrix $T$ given the eigenvalue Lambda.

8. 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 Spherical_Dev_Ellipse() subroutine and ensure that it contains the proper calls and parameters to the Newton() and Gauss() subroutines. Ordering is important. The subroutines in Ch7SphericalStatistics, particularly the Newton() subroutine, must be run before the Gauss() can be run. The statements in the Spherical_Dev_Ellipse() subroutine clearly show the ordering of the other subroutine calls.
**NEWTON AND GAUSSIAN ELIMINATION DRIVER**

Dim Lambda As Double '1 to 3 possible eigenvalues
Dim l, m, n, l0, l0_sqrd, m0, m0_sqrd, n0, n0_sqrd, lm_sum, ln_sum, mn_sum As Double

Dim l_sum, m_sum, n_sum As Double

Sub NewtonGauss()
    Dim AVar As Integer 'hides procedure names in pick list
    'syntax
    Sub Newton(Lambda)
    Sub Gauss(Lambda)
    Sub Calculation_T()
    Call SphereSetWeights(AVar)
    Call Calculate_B(l0, m0, n0, l0_sqrd, m0, m0_sqrd, n0, n0_sqrd, lm_sum, ln_sum, mn_sum)
    Call Calculate_T(l0, m0, n0, l0_sqrd, m0, m0_sqrd, n0, n0_sqrd, lm_sum, ln_sum, mn_sum)
    "kentucky 2004
    "lambda_2
    "Call Newton(0)
    "Call Gauss(0.006533004)
    "lambda_1
    "Call Newton(2)
    "Call Gauss(0.035928191)
    "final step — round the chosen shape to an integer for interpretation
    "Call Newton(92)
    "Call Gauss(91.95753881)
    "crime 2008
    "lambda_1
    "Call Newton(1)
    "Call Gauss(0.047899513)
    "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 2010
\lambda 1
\texttt{Call Newton(0)}
\texttt{Call Gauss(3.728832776)}
\lambda 2
\texttt{Call Newton(10)}
\texttt{Call Gauss(6.216179462)}
\lambda 3
\texttt{Call Newton(39)}
\texttt{Call Gauss(29.05498776)}
\texttt{End Sub}
APPENDIX E
REFERENCES


REFERENCES


44. mardia2


REFERENCES


Index

Arctan function, 48
R, 122
Ch5ExponentialDeviationalEllipse, 152
Developer, 1
Dim statement, 3
weighted mean center, 37
Active.Cells, 9
altitude, 37
angle of rotation, 44, 47, 48, 51, 72, 81, 99, 145
Angle_of_Rotation(), 146
anti-mode, 120, 124
approximate formula, 79
Arc GIS, 35
area, xxv, 50, 73, 74, 99, 100, 146, 149, 150, 154, 160
area frame, xxiii
area frame survey, xxiii
area preserving plot, 122
Area(), 146
area, check, 100
area, total, 71
array, 4
arrays, 4
asymptotically, 125
average, 8, 122
averages, 29
axes lengths, 71, 99
axes relationship, 81
axis, 19
axis length, semi-major, 43
axis length, semi-minor, 43
axis of rotation, 20, 149, 153, 159
axis, semi-major, 14
axis, semi-minor, 14
bar graph, 75
bars, 18
beginning row, 9
bimodal, 124
browse, 16, 18
Calculate_B(), 167
Calculate_T(), 167
Cartesian coordinate system, 42, 43
cell, 7
cell contents, 7
cell, properties, 7
central point, xxv

Random Variables, Their Properties, and Deviational Ellipses.
By Roger L. Goodwin Copyright © 2015 Roger L. Goodwin
center of gravity, xxiv, 29, 43, 47, 76, 77, 102, 104
center of system, xxv
Ch4StandardDeviationalEllipse, 146
Ch6WeibullDeviationalEllipse, 157
Ch7NewtonGaussRoutines, 165
Ch7SphericalStatistics, 165, 167
chi-square distribution, 125
circle, xxv, 45, 50
circular coordinate system, 92, 148
circular coordinates, 95
circular data, 43, 122
closed form, 98
centered, 124
centered pattern, 124
column chart, 18
column names, 18
columns, 8
common syntax, 8
concentration, 50
centration of the system, xxv
condition variable, 7
continental U.S., 75
conventions, 8
convergence, 92, 102
convergence analysis, 101
coordinate pairs, 47
coordinates, 13, 15, 36
copying and pasting, 13
correlation, 120
correlation coefficient, 45
county, 36
county level, 38
county-level, 38
credible parameter estimates, 101
crime data, 75, 76
Crimestat, 35
cross hairs, 21
cubic equation, 127
cumulative distribution function, xxvi, 66, 67, 92, 96, 97
cumulative distribution function, joint, 70
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, 119, 133
data, transform, 65, 68
data, transforming, 65
date, 3, 4
decimal degrees, 36
default names, 8
degrees, 122
derivatives, 24, 92
determinant, 125, 127, 129
Developer ribbon, 9, 146, 152
development environment, 21
deviation, xxiv
deviations, 72, 99
diagonal elements, 125, 129
direction, xxv
directional data, 121
directional statistics, 120–122
directional, data, 124, 129
dispersed pattern, 124
dispersion, 45
disproportional weight, 128
distance, 42
distribution, 65
distribution of the data, 73
distribution of the points, xxv
distribution, sample, 105
distribution, underlyin, 105
double, 3
draw ellipses, 14
drawing lines, 13
Earth, 13, 15, 16, 19, 43
eccentricity, xxv, 50, 72, 73, 99, 146, 150, 154, 160
eigenvalue, 125
eigenvalues, 124, 125, 132, 134, 165
eigenvectors, 125, 132, 134, 165
ellipse, xxv, 50, 72, 73
ellipse, center, 43
eellipse, draw, 19, 44
eellipse, multi-modal, 120
eellipse, rotate, 20, 44
elliptical pattern, 124
elliptical formula, 26
ELSEIF, 6
ending row, 9
Equator, 121
error messages, reasons, 102
error term, 145
error terms, 48
estimates, 158
estimator, 29, 48
Excel, 3
Excel 2010, VBA, 36
Excel workbook, 16
exp, 9
exponential deviational ellipse, 73, 74, 76
Exponential distribution, 24
exponential distribution, 68, 71
exponential distribution function, 65, 68
exponential ellipse, 71, 73, 74, 76, 77–82, 90, 93, 105
exponential ellipse code, 151
exponential mean center, 71
exponential model, 48
ExponentialDevEllipse(), 153
exponentially distributed, 67, 69
exponentially distributed, weighted, deviational ellipse, 71
exponential model, 154
EXPstats(), 152
extreme values, 82
first derivative, 90, 93
font, 7
font size, 7
FOR-NEXT, 6, 7
format shape, 20
formulas, 8
FORTRAN, 35
fx(Lambda), 167
Gamma1, 158
Gamma2, 158
Gauss(Lambda), 167
Gaussian elimination, 127
general linear means model, 48, 71
go-co-de, 13, 35
go-geo-coding, 43
go-geo-statistics, 45
go-geo-data, 43
girdle, 124, 125
girdle distribution, 130
global variables, 8
Google Earth, 13, 35, 36
graph, 69
graph paper, 65
graphing, 13
graphs, 105
gx(Lambda), 167
histogram, 120, 128, 133
hoop coordinate system, 42
IF-ELSE-IF, 5
IF-THEN-ELSE, 5
image classification, xxiii
imaginary axis, 44
Import data wizard, 15, 16, 18
initial values, 91
initial guess, 159
initial guesses, 157
initial value, 102, 127
initial values, 92, 94
initialization, 92, 94
input parameters, Weibull, 101
integrals, 25
integer numbers, 3
iteration, 24, 91, 96
Iterations, 158
iterations, 92, 102, 159
iterations, number of, 92
Jacobian matrix, 91, 94
joint density function, 95
joint distribution, 67, 69, 95, 96, 154, 159, 160
joint likelihood function, 69
joint probability density function, 69, 96
June Area Survey, 36, 37
kilometers, 19
kurtosis, 121
label text, 18
land segments, 75
largest weights, 75
latitude, xxiv, 9, 24, 43, 65, 68, 69, 71, 79, 89, 96, 100, 102, 121, 130, 154
latitude observation, 48
least squares, 71, 120
length, 13, 19
likelihood function, 67, 68, 90, 101, 105, 150, 154, 160
likelihood function, Weibull, 90, 93
likelihood functions, 157
linear means model, 98
linear regression, 120
linear regression model, 48, 65, 71
linear relation, 67, 69
linear relationship, 65, 67, 68
local variables, 9
location parameter, 65, 71
location problem, xxiv
log, 9
Log10(X), 158
logical, 4
long integer, 4
longitude, xxiv, 9, 24, 43, 49, 68, 69, 71, 79, 93, 96, 100, 102, 121, 130, 154
longitude data, 92
longitude observations, 92
loop, For-Next, 9
loop, While-Wend, 9
major axis, 72, 99, 149
major axis length, 73
major axis of rotation, 153, 159
map, 43, 44
Map Point, 13, 18, 20
Map Soft, 35
matrix $B$, 124, 129, 134
matrix $T$, 125, 129, 130, 134, 165
maximum likelihood estimates, 96
maximum likelihood estimator, 65, 67–69, 90, 93, 95, 96
maximum likelihood theory, 105
mean center, xxiv, 13, 19, 45, 46, 81, 92, 95, 99, 103, 119, 154
mean center coordinates, 154
mean center, transform, 150
mean direction, 122
mean joint distribution, 71
mean joint distribution estimate, 98
mean latitude, 122, 123
mean longitude, 122
measure lines, 44
menu, Insert, 12
merged cells, 39
miles, 19
minor axis, 72, 73, 99, 149
minor axis length, 73
minor axis of rotation, 154, 159
mod, 8
mode, 120, 124
modes, 133
module, 4
modules, 9
MS Word, 20
multimodal statistics, 120
Newton algorithm, 127
Newton’s method, 127
Newton() subroutine, 129
Newton(Lambda), 167
Newton-Raphson method, 91, 94
NewtonGauss(), 165
no fill, 20
non-uniform distribution, xxv
normal distribution, 71
Northern Hemisphere, 121
null hypothesis, 132, 134
number of iterations, 92
numerical algorithms, 91

observation, 68
observed value, 70, 90
one pole, 42
original data, 98

oval, 20
parameter, 24, 67
parameter estimate, 90, 92, 93, 95
parameter estimates, 101
parameter estimation, 92
parameter estimator, 68
parameters, 90, 105, 158, 159, 167
percentage, 121
pie chart, 18
plane, 124
plot, 14, 67, 69
plotting, 13
plotting the transformed data, 65
point estimates, 159
point pattern, 45
points, 43
points, distribution, 130
polar coordinate, 122
polar coordinate system, 43
positive definite, 79, 80
predicted values, 48
preferred direction, no, 124
Prime Meridian, 121
probability density function, 66, 69, 89, 92, 95
program code, 8
properties, 7
pushpins, 16
Pythagoras’ theorem, 45
quadratic equation, 26
quadratic equation, roots, 26, 47
quantile, 121
quantity, 38
radians, 122
random, 128
random sample, 90, 93
random variable, xxiv, 9, 24, 35, 43, 48, 49, 67, 68, 70, 71, 93, 97, 130
random variable $w_i$, 90
range, 95
real number, 4
real numbers, 3
record, 4
records, 4
regional data, 39
regional level, 38, 39
regression, 71
regression estimator, 45
regression line, xxvi
relationship, 71, 79, 80
relative concentration, xxv
resultant length, 123, 124
roots, 125
rotate, 14
rotational symmetry, 125
row reduction, 127
ruler, 44
ruler, anchor, 19

sample data, 66
sample size, 24, 122
sample space, 68
satellite images, 36
saving work, 13
scale parameter, 24, 66, 71, 159
scatter diagram, 121
SDE code, 146
SDE statistics, 148
SDELikelhood(), 146
SDEdeltas(), 146
Secant algorithm, 96, 157, 159
Secant estimation, 162
Secant iterations, 98
Secant method, 91, 92, 94
Secant programs, 100
Secant1(), 157, 158
Secant2(), 158, 159
segment, xxiv
Selection.Rows.Count, 8
semi-major axes, 79, 81
semi-major axis, 44, 48, 78
semi-major axis length, 49, 51, 80, 81, 99, 100, 145, 149, 154, 160
semi-minor axes, 81
semi-minor axis, 44, 48, 81
semi-minor axis length, 49, 51, 81, 99, 100, 145, 150, 154, 160
series column chart, 18
Setweights(), 146, 152
shaded circles, 16
shape parameter, 65, 159
shapes, 20
single pole, 42
sized circles, 16
sized pie chart, 18
skewness, 121
smaller weights, 75
Southern Hemisphere, 121
spatial analysis, 119
spatial data, 37, 122
spatial statistics, xxiv
specialty software, 44
spherical distribution, 124, 165
spherical statistics, 42, 43, 119, 127
spherical variance, 123
SphericalDevEllipse(), 130, 165
spreadsheet, 39
standard deviation, xxiv, 119, 149, 154, 159
standard deviational ellipse, xxiv, 45, 49, 52, 73, 74, 76–82, 96, 102, 104, 146
standard deviational ellipse formula, 70
standard deviations, 72, 73, 99
standard distance, 45
standard ellipse, 79
standard error terms, 48
state name, 36
STATS, 9, 148, 153, 159
stereographic-net, 122
straight line, 66, 98
string characters, 3
subroutine, 4
sufficient statistics, 90, 93
sum, 8
sums of squares, 71, 79, 81, 99, 105
survey, 37
survey area, 19, 43
survey data, 13

termination condition, 159
test statistic, 129, 132, 134
THEN, 6
time, 3
trace, 125
transformation, weighted latitude, 66
transformed data, 67, 71, 98
transformed variable, 67
trend of the system, xxv
triadic, 129
trial and error, 92, 127
two dimensions, 42
TYPE-END-TYPE, 4
uniform, 124, 125, 134
uniform distribution, xxiv, xxv
units, 19
unweighted exponential deviational ellipse, 154
unweighted, calculations, 119

Variable, 158
VBA code, 92
VBA Development environment, 1
VBA Editor, 12, 148
VBA editor, 4
VBA environment, 8, 146, 165
VBA functions, 9
VBA numeric calculations, 9
VBA program, 151
VBA statements, 8
VBA, StandardDevEllipse(), 145
vector \mathbf{\hat{z}}, 124
vectors, 42
INDEX

violent crime, 53, 130
violent crime, survey, 28
visible, 4
Visual Basic button, 12, 152
Visual Basic Editor, 151
Visual Basic for Applications (VBA), 36
Visual Basic ribbon, 146

Weibull deviational ellipse, 98, 101, 102, 104
Weibull distribution, 24, 92, 98
Weibull distribution, sufficient statistics, 90, 93
Weibull ellipse, 90, 93, 102–105, 157
Weibull models, weighted, 95
WeibullDevEllipse(), 158, 159
weighted, 24, 29, 48, 130, 146
weight column, 154
weight, disproportionately, 130
weighted data, 28, 75
weighted ellipse, 48
weighted exponential deviational ellipse, 75, 78, 79, 81
weighted exponential distribution, 65
weighted exponential model, 69
weighted exponential standard deviational ellipse, 71
weighted joint distribution, 80
weighted latitude, 66, 67, 71
weighted longitude, 69, 71, 80
weighted mean, 78, 79
weighted mean center, xxv, 45, 46, 49, 95, 145, 149, 153, 159
weighted mean estimate, 71
weighted mean response, 48
weighted means estimate, 98
weighted points, 72
weighted standard deviational ellipse, xxv, 37, 46, 47, 79, 81, 145
weighted sum of squares, 48
weighted Weibull models, 95
weighted, Weibull ellipse, 99
weighted, Weibull standard deviational ellipse, 98
weights, 97
WHILE-WEND, 6, 7
whole numbers, 3
WITH-END-WITH, 7
worksheet, 8
worksheetfunction.pi(), 9
Worksheets().Cells().Value, 9
Wulff-net, 122