ANLY 500 Laboratory #1 – Descriptive Statistics

Evans Chapter 1 through and including Chapter 7

“Performance Lawn Equipment Case Study” from Evans, **Business Analytics**

Contents

[Introduction 3](#_Toc484158802)

[Chapter 1 5](#_Toc484158803)

[Step 1 5](#_Toc484158804)

[Step 2 5](#_Toc484158805)

[Chapter 2 - Optional 6](#_Toc484158806)

[Step 1 6](#_Toc484158807)

[Step 2 7](#_Toc484158808)

[Step 3 9](#_Toc484158809)

[Step 4 11](#_Toc484158810)

[Chapter 3 11](#_Toc484158811)

[Part 1 11](#_Toc484158812)

[Step 1 12](#_Toc484158813)

[Step 2 17](#_Toc484158814)

[Part 2 17](#_Toc484158815)

[Step 1 17](#_Toc484158816)

[Part 3 18](#_Toc484158817)

[Step 1 18](#_Toc484158818)

[Step 2 21](#_Toc484158819)

[Part 4 21](#_Toc484158820)

[Chapter 4 21](#_Toc484158821)

[Part 1 21](#_Toc484158822)

[Step 1 (Part a) 21](#_Toc484158823)

[Step 2 (Part b) 24](#_Toc484158824)

[Step 3 (Part c) 25](#_Toc484158825)

[Step 4 (Part d) 26](#_Toc484158826)

[Step 5 (Part e) 27](#_Toc484158827)

[Chapter 5 31](#_Toc484158828)

[Part 1 31](#_Toc484158829)

[Step 1 31](#_Toc484158830)

[Step 2 32](#_Toc484158831)

[Step 3 32](#_Toc484158832)

[Step 4 33](#_Toc484158833)

[Step 5 33](#_Toc484158834)

[Step 6 33](#_Toc484158835)

[Step 7 34](#_Toc484158836)

[Step 8 34](#_Toc484158837)

[Step 9 35](#_Toc484158838)

[Step 10 35](#_Toc484158839)

[Chapter 6 36](#_Toc484158840)

[Part 1 36](#_Toc484158841)

[Step 1 36](#_Toc484158842)

[Step 2 37](#_Toc484158843)

[Step 3 39](#_Toc484158844)

[Step 4 39](#_Toc484158845)

[Step 5 42](#_Toc484158846)

[Step 6 43](#_Toc484158847)

[**Chapter 7** 44](#_Toc484158848)

[**Part 1** 45](#_Toc484158849)

[**Part 2** 46](#_Toc484158850)

[**Part 3** 47](#_Toc484158851)

[**Part 4** 48](#_Toc484158852)

[**Part 5** 49](#_Toc484158853)

[Summary – Your Laboratory Report 51](#_Toc484158854)

[Chapter 1: 54](#_Toc484158855)

[Appendix 1: 54](#_Toc484158856)

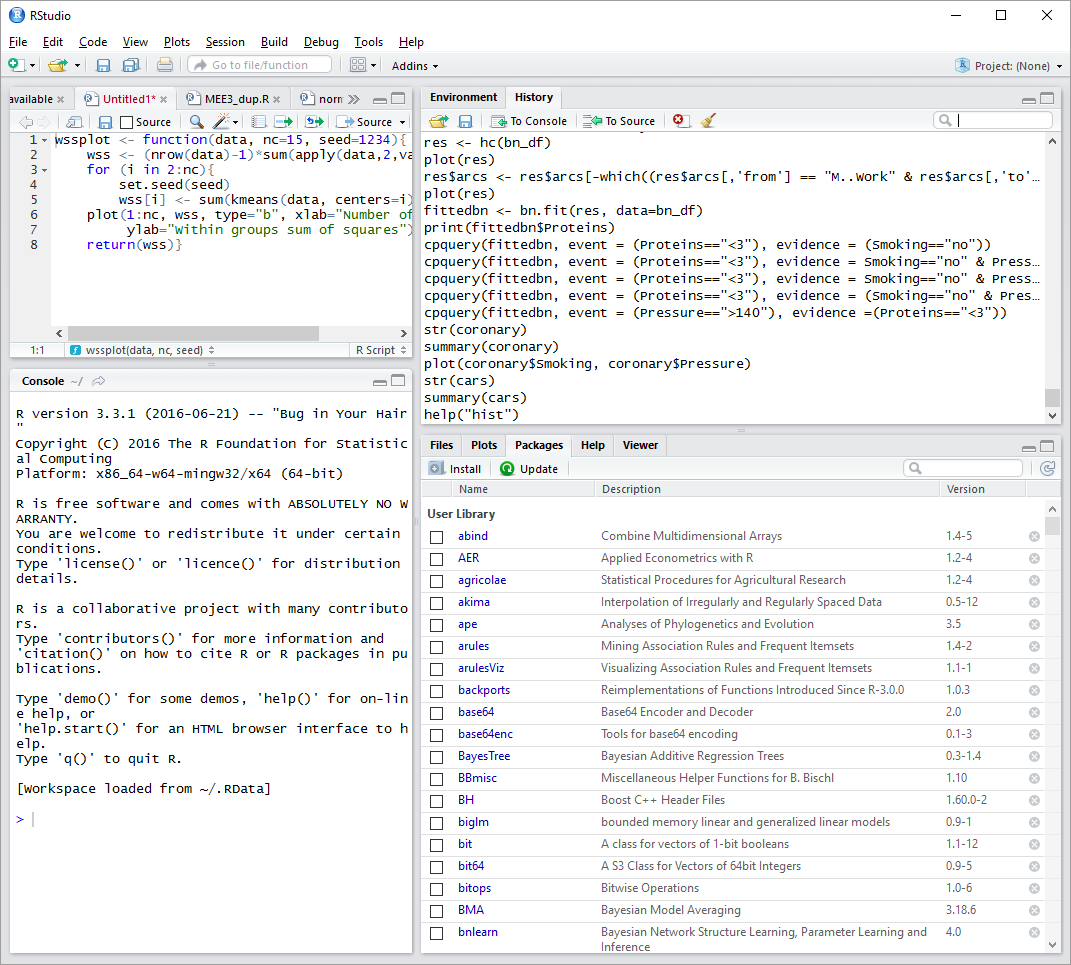
# Introduction

This laboratory follows the exercises in the book, specifically the Performance Lawn Equipment Case Study homework assigned exercises Chapters 1 through and including 7, except this laboratory requires that you use R to complete the exercises. That is, you should answer all questions in the textbook exercises and when necessary to complete computations use R. Each laboratory in ANLY 500 will build on the laboratories you have completed before. So, you will want to set-up a folder or file to keep your work in so that you can refer back to previous laboratories if necessary. If you have not used R before you should install R and RStudio on your computer or laptop. RStudio is a user interface for R that will make your life and work much easier. To get credit for completing this laboratory you must submit a report with your results on Moodle.

Once you have installed R and RStudio you may want to browse through some of the packages available for you. You can do that from the “Quick list of useful R packages” at <https://support.rstudio.com/hc/en-us/articles/201057987-Quick-list-of-useful-R-packages> or <https://cran.r-project.org/web/packages/>. Essentially what R does is use functions already coded in these packages to do the computations you want to perform. Each package will have an associated CRAN package website that provides all the information you need about any package. You can also do a Boolean search on Google or other browser to find additional information about packages or functions. If you need a specific package to complete an exercise you will be told which package that is as part of the exercise.

Unless told you will need to find a data set to use you will be provided data sets through Moodle. This is true for this laboratory, ANLY 500 Laboratory #1. For this laboratory there are a total of 23 data files in csv format. There is one data file for each spreadsheet in the Performance Lawn Equipment Excel Workbook that is also on Moodle. Your first task will be to load these data files into RStudio. However, before you can begin to read data into RStudio you will need to be able to move around the folders on your computer.

When you start RStudio you will see a number of frames in the RStudio window, going clockwise from the upper left: a frame showing contents of your R scripts or data objects; an Environment and History frame; an information frame including your files, plots, packages, help and viewer; and, your Console frame. This will look something like the figure below:



To find out what folder your default folder is set-up to be you can use the pull down menu “Tools” then look at your “Global Options”. If you click on “Browse” by the box for your “Default working directory” it will take you to the directory that RStudio goes to automatically when you start RStudio. If you want to change this directory just browse to the one you want to use and accept that change. I strongly suggest you set-up a separate directory just for your R/RStudio work. I have one I names “MyRWork” and within that folder I have a “data” folder and other folders for specific projects, etc.

The function to find out what directory you are actually in is getwd() which simply stands for get working directory. That is the syntax you need to use. The parentheses, which are empty, return the current directory. To change directory the function is setwd(). So, for example if I am in my default working directory “MyRWork” and want to go to my data directory I enter setwd(“data”) in the console frame. The quotation marks are necessary. R distinguishes between names with no quotation marks, single or double quotation marks and treats each entry differently. So, for the directory name use the double quotation marks around it. If things are not working quite right, e.g. RStudio isn’t reading files, chances are you are not in the correct directory.

Once you are in the directory where you’ve downloaded the data files you can load the data into RStudio. You can do this automatically using the pull down menu “Tools” then “Import Dataset”. Doing this you choose a “From Local File” or “From Web URL”. Since these will be on your computer or laptop choose “From Local File” then just go to the appropriate directory and select the data file you want to import. Or you can immediate begin using R’s functions for reading in data using the following command in the Console frame:

> BladeWeight <- read.csv("~/MyRWork/data/Evans/BladeWeight.csv")

You have extensive help files available through RStudio. To get help, in the Console use the help() function and the function name you need help with in double quotations, e.g.

> help("read.csv")

One thing to watch out for, whether you use the pull down menu to import or the command line, is that the column headings are recognized. For some reason when I read or import the Evans’ data files some files recognize headings and other do not. So, be careful about this.

# Chapter 1

## Step 1

Read the data files for Performance Lawn Equipment (PLE) into R/RStudio.

## 

## sStep 2

Determine the data type for each variable in the PLE data files.

It is easy to determine the data types for variables in R. Simply use the str() function with the data filename or data object in the parentheses. For example,

> str(BladeWeight)

'data.frame': 350 obs. of 2 variables:

$ Sample: int 1 2 3 4 5 6 7 8 9 10 ...

$ Weight: num 4.88 4.92 5.02 4.97 5 4.99 4.86 5.07 5.04 4.87 ...

For the BladeWeight data file the str() function returns the information that there are 350 observations of 2 variables, sample and weight. The sample variable is an integer variable. The weight variable is a numeric variable. You can use the str() function for each data file to determine the data type of all variables. Let’s consider one more of the data files in detail, i.e. the EmployeeRetention data file. Using the str() function we get:

> str(EmployeeRetention)

'data.frame': 40 obs. of 7 variables:

$ YearsPLE : num 10 10 10 10 9.6 8.5 8.4 8.4 8.2 7.9 ...

$ YrsEducation: int 18 16 18 18 16 16 17 16 18 15 ...

$ College.GPA : num 3.01 2.78 3.15 3.86 2.58 2.96 3.56 2.64 3.43 2.75 ...

$ Age : int 33 25 26 24 25 23 35 23 32 34 ...

$ Gender : Factor w/ 2 levels "F","M": 1 2 2 1 1 2 2 2 1 2 ...

$ College.Grad: Factor w/ 2 levels "N","Y": 2 2 2 2 2 2 2 2 2 1 ...

$ Local : Factor w/ 2 levels "N","Y": 2 2 1 2 2 2 2 2 2 2 ...

where we have 40 observations for the 7 listed variables. Gender, College Grad, and Local are all listed as “Factor” variables. This is the same as a categorical variable. If you are not familiar with different data types you should take some time to look into this. One source you can use is: <https://www.tutorialspoint.com/r/r_data_types.htm>.

# Chapter 2 - Optional

## Step 1

Find the total number of responses to each level of the surveys, Dealer and End-User Satisfaction, across all regions for each year.

To do this we’ll need to subset the data by level and by year. Subsetting data is a standard part of data analysis. As you will find with most things in R there are many ways of doing this. You can find lots of information about this online. For example, because the data is essentially in the format of a matrix you could use row and column numbers – if you know those. You can also use variable names or values. For example,

> y2010 <- DealerSatisfaction[ which(DealerSatisfaction$Year == 2010), ]

establishes a new data object “y2010” and redirects observations from the DealerSatisfaction data file into it in which the Year equals 2010. We can see the contents of “y2010” be just typing it on the command line in the Console.

> y2010

Region Year L0 L1 L2 L3 L4 L5 Count

1 <NA> 2010 1 0 2 14 22 11 50

6 SA 2010 0 0 0 2 6 2 10

11 EU 2010 0 0 1 3 7 4 15

16 PA 2010 0 0 1 2 2 0 5

Things to note include the syntax for the data filename and variable separated by a $, e.g. DealerSatisfaction and Year as DealerSatisfaction$Year. To designate the test for “equals” use a double ==. You can use this syntax to find all the data required for this step. One of the really nice things about RStudio is that you can move between commands you have used by the “up” and “down” arrows on your keyboard. So, to go to the next year you only need to use the up arrow twice to go to the command to subset the data and then change the year to 2011 to get the answers for the next year, and so on. When you get to the part that asks for this data for End-User Satisfaction you just need to use the up arrow appropriately and change the file name.

DO NOT forget to change the data object name for each command you use to store your results in. If you do not change the data object name you will be continuously writing over your previous results.

To find the sum by year and return the value use the information you got before and sum, e.g.:

> y2010L0 <- sum(y2010[,3])

> y2010L0

[1] 1

That is, the sum for the year 2010 for all regions is 1. You can do this for all the instances you need to. Choose data object names that make sense so that as you need them you can easily find them again and again.

Keep in mind that R uses typical matrix notation, i.e. [rows, columns]. So you can always find the value in an element in a matrix by its [row, column] designation. ###S

## Step 2

Find the number of failures in the Mower Test.

When the author completed this in Excel he just used the “COUNTIF” function for each column of his spreadsheet Mower Test. This will take a bit more syntax in R but is relatively easy using the sapply() function. There is actually a whole family of “apply” functions; e.g. apply, sapply, lapply, etc. All are helpful in their way so check them out, e.g. in the answer at <http://stackoverflow.com/questions/3505701/r-grouping-functions-sapply-vs-lapply-vs-apply-vs-tapply-vs-by-vs-aggrega>. For our question, we’ll use sapply() and the data file MowerTest but exclude the first column which is the variable “Observations”. We’ll return the values of “Pass” and “Fail” as a table for each of the Samples 1 through 30. The syntax we’ll use is:

> y <- t(sapply(MowerTest[-1], function(x) table(factor(x, levels = c("Pass", "Fail")))))

for which y returns:

> y

Pass Fail

Sample.1 97 3

Sample.2 96 4

Sample.3 99 1

Sample.4 100 0

Sample.5 99 1

Sample.6 95 5

Sample.7 98 2

Sample.8 99 1

Sample.9 100 0

Sample.10 98 2

Sample.11 98 2

Sample.12 97 3

Sample.13 97 3

Sample.14 99 1

Sample.15 99 1

Sample.16 98 2

Sample.17 98 2

Sample.18 97 3

Sample.19 98 2

Sample.20 96 4

Sample.21 98 2

Sample.22 99 1

Sample.23 99 1

Sample.24 98 2

Sample.25 99 1

Sample.26 100 0

Sample.27 98 2

Sample.28 99 1

Sample.29 100 0

Sample.30 98 2

Unfortunately, sapply() has not returned this as a data.frame from which we could simply find the total number of “Fail”. So, we’ll convert the output of sapply() into a data.frame and show the output as:

> y2 <- as.data.frame(y)

> str(y2)

'data.frame': 30 obs. of 2 variables:

$ Pass: int 97 96 99 100 99 95 98 99 100 98 ...

$ Fail: int 3 4 1 0 1 5 2 1 0 2 ...

Since y2 is a data.frame we can just use the sum() function to get the total number of “Fail” as follows:

> sum(y2$Fail)

[1] 54

## Step 3

Compute the gross revenue by months and region as well as worldwide for each product using the data in Mower Unit Sales and Tractor Unit Sales.

We have the numbers of units sold and the price per unit so we just need to compute the gross revenue. I haven’t really stressed it yet but, as always, the first thing to do is look at the data, use the str() and summary() functions as follows:

> str(MowerUnitSales)

'data.frame': 60 obs. of 8 variables:

$ Month : Factor w/ 12 levels "April","August",..: 5 4 8 1 9 7 6 2 12 11 ...

$ Year : int 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...

$ NA. : int 6000 7950 8100 9050 9900 10200 8730 8140 6480 5990 ...

$ SA : int 200 220 250 280 310 300 280 250 230 220 ...

$ Europe : int 720 990 1320 1650 1590 1620 1590 1560 1590 1320 ...

$ Pacific: int 100 120 110 120 130 120 140 130 130 120 ...

$ China : int 0 0 0 0 0 0 0 0 0 0 ...

$ World : int 7020 9280 9780 11100 11930 12240 10740 10080 8430 7650 ...

> summary(MowerUnitSales)

Month Year NA.

April : 5 Min. :2010 Min. : 4350

August : 5 1st Qu.:2011 1st Qu.: 5998

December: 5 Median :2012 Median : 7870

February: 5 Mean :2012 Mean : 7542

January : 5 3rd Qu.:2013 3rd Qu.: 9050

July : 5 Max. :2014 Max. :10370

(Other) :30

SA Europe Pacific

Min. :180.0 Min. : 300 Min. :100.0

1st Qu.:250.0 1st Qu.: 840 1st Qu.:140.0

Median :280.0 Median :1260 Median :170.0

Mean :282.3 Mean :1149 Mean :172.5

3rd Qu.:310.0 3rd Qu.:1440 3rd Qu.:202.5

Max. :390.0 Max. :1650 Max. :240.0

China World

Min. : 0.000 Min. : 5350

1st Qu.: 0.000 1st Qu.: 7335

Median : 0.000 Median : 9390

Mean : 1.883 Mean : 9148

3rd Qu.: 0.000 3rd Qu.:10999

Max. :26.000 Max. :12280

> str(TractorUnitSales)

'data.frame': 60 obs. of 8 variables:

$ Month: Factor w/ 12 levels "April","August",..: 5 4 8 1 9 7 6 2 12 11 ...

$ Year : int 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...

$ NA. : int 570 611 630 684 650 600 512 500 478 455 ...

$ SA : int 250 270 260 270 280 270 264 280 290 280 ...

$ Eur : int 560 600 680 650 580 590 760 645 650 670 ...

$ Pac : int 212 230 240 263 269 280 290 270 263 258 ...

$ China: int 0 0 0 0 0 0 0 0 0 0 ...

$ World: int 1592 1711 1810 1867 1779 1740 1826 1695 1681 1663 ...

> summary(TractorUnitSales)

Month Year NA.

April : 5 Min. :2010 Min. : 360.0

August : 5 1st Qu.:2011 1st Qu.: 637.5

December: 5 Median :2012 Median : 835.0

February: 5 Mean :2012 Mean :1075.0

January : 5 3rd Qu.:2013 3rd Qu.:1407.5

July : 5 Max. :2014 Max. :2490.0

(Other) :30

SA Eur Pac

Min. : 250.0 Min. :480.0 Min. :190.0

1st Qu.: 412.5 1st Qu.:577.5 1st Qu.:250.0

Median : 605.0 Median :647.5 Median :270.0

Mean : 598.4 Mean :648.0 Mean :272.2

3rd Qu.: 806.2 3rd Qu.:720.0 3rd Qu.:300.0

Max. :1002.0 Max. :888.0 Max. :350.0

China World

Min. : 0.00 Min. :1592

1st Qu.: 0.00 1st Qu.:1962

Median : 23.00 Median :2408

Mean : 46.65 Mean :2640

3rd Qu.:100.50 3rd Qu.:3222

Max. :139.00 Max. :4476

> str(Prices)

'data.frame': 5 obs. of 3 variables:

$ Year : int 2010 2011 2012 2013 2014

$ Mower.Price : int 150 175 180 185 190

$ Tractor.Price: int 3250 3400 3600 3700 3800

> summary(Prices)

Year Mower.Price Tractor.Price

Min. :2010 Min. :150 Min. :3250

1st Qu.:2011 1st Qu.:175 1st Qu.:3400

Median :2012 Median :180 Median :3600

Mean :2012 Mean :176 Mean :3550

3rd Qu.:2013 3rd Qu.:185 3rd Qu.:3700

Max. :2014 Max. :190 Max. :3800

We have prices by year for mowers and tractors. So, we’ll need to subset our mower and tractor data and apply the correct price by year. There are some things we know that we can use to do this, e.g. each year has 12 months. So we can subset our data in sets of 12 observations (12 months) and multiply the number of units by the appropriate price. We know that the first two columns are the month and year and will have to be omitted from the computations. One way to do this is to subset the data by rows and columns to get:

> mowerRev2010 <- MowerUnitSales[1:12, 3:8] \* Prices[1,2]

> mowerRev2010

NA. SA Europe Pacific China World

1 900000 30000 108000 15000 0 1053000

2 1192500 33000 148500 18000 0 1392000

3 1215000 37500 198000 16500 0 1467000

4 1357500 42000 247500 18000 0 1665000

5 1485000 46500 238500 19500 0 1789500

6 1530000 45000 243000 18000 0 1836000

7 1309500 42000 238500 21000 0 1611000

8 1221000 37500 234000 19500 0 1512000

9 972000 34500 238500 19500 0 1264500

10 898500 33000 198000 18000 0 1147500

11 798000 31500 148500 19500 0 997500

12 696000 27000 99000 21000 0 843000

You can do this for all the years for both mowers and tractors. Or, you can use the more complicated syntax of one of the “apply()” functions.

## Step 4

Now that you have the revenue for mower and tractor sales you can easily find the market share. In fact, You simply find the total gross mower revenue by region, the sum() function properly applied will do that, and then divide that by the Industry data. You follow the same procedure for the market share of tractor sales. Be sure to provide your results and summary of results in your laboratory report.

# Chapter 3

## Part 1

You have been tasked with putting together an overview of PLE’s business performance and market position. You have specifically been asked to construct appropriate charts and summarize your conclusions for:

1. Dealer Satisfaction
2. End-User Satisfaction
3. Complaints
4. Mower Unit Sales
5. Tractor Unit Sales
6. On-Time Delivery
7. Defects after Delivery
8. Response Time

### Step 1

To begin this we’ll again need to put together subsets of the data, e.g. Dealer Satisfaction for North America, and so on. Once we are through subsetting the data we can create the plots that describe what is going on performance wise. But first, let’s look at the data:

> str(DealerSatisfaction)

'data.frame': 23 obs. of 9 variables:

$ Region: Factor w/ 4 levels "CH","EU","PA",..: NA NA NA NA NA 4 4 4 4 4 ...

$ Year : int 2010 2011 2012 2013 2014 2010 2011 2012 2013 2014 ...

$ L0 : int 1 0 1 1 2 0 0 0 0 1 ...

$ L1 : int 0 0 1 2 3 0 0 0 1 1 ...

$ L2 : int 2 2 1 6 5 0 0 1 1 2 ...

$ L3 : int 14 14 8 12 15 2 2 4 3 4 ...

$ L4 : int 22 20 34 34 44 6 6 11 12 22 ...

$ L5 : int 11 14 15 45 56 2 2 14 33 60 ...

$ Count : int 50 50 60 100 125 10 10 30 50 90 ...

> str(EndUserSatisfaction)

'data.frame': 23 obs. of 9 variables:

$ Region: Factor w/ 4 levels "CH","EU","PA",..: NA NA NA NA NA 4 4 4 4 4 ...

$ Year : int 2010 2011 2012 2013 2014 2010 2011 2012 2013 2014 ...

$ L0 : int 1 1 1 0 0 1 1 0 0 0 ...

$ L1 : int 3 2 2 2 2 2 3 2 2 2 ...

$ L2 : int 6 4 5 4 3 5 6 6 5 5 ...

$ L3 : int 15 18 17 15 15 18 17 19 20 19 ...

$ L4 : int 37 35 34 33 31 36 36 37 37 37 ...

$ L5 : int 38 40 41 46 49 38 37 36 36 37 ...

$ Count : int 100 100 100 100 100 100 100 100 100 100 ...

There are 23 observations of 9 variables in each data file. There are six levels of satisfaction that have been recorded, which is a bit odd. Usually an odd number of levels of satisfaction are used for a Likert Scale. We could also do some descriptive statistics using the summary() function but since the years and regions would be aggregated I’m not sure that would reveal much.

First, let’s subset the data by region to create data tables. For example, we can start with the first region, North America or NA, and print out the return as:

> dealerSat\_NA <- DealerSatisfaction[1:5, ]

> dealerSat\_NA

Region Year L0 L1 L2 L3 L4 L5 Count

1 <NA> 2010 1 0 2 14 22 11 50

2 <NA> 2011 0 0 2 14 20 14 50

3 <NA> 2012 1 1 1 8 34 15 60

4 <NA> 2013 1 2 6 12 34 45 100

5 <NA> 2014 2 3 5 15 44 56 125

In case you had been wondering the expression “NA” is a standard expression in R and most programming environments or languages. NA typically denotes a missing value. So, R has automatically put brackets around NA in our data files even though NA for us means North America. You can set up objects this same way for South America (SA), Europe (EU), Pacific Rim (PA), and China (CH). Now, for the plotting.

Again, there will be many ways to create the required plots, e.g. the lattice package or ggplot2. We’ll use ggplot2 for this example. You’ll need to install the ggplot2 and labeling packages and attach them using library(ggplot2) and library(labeling) commands. If you have any trouble installing packages or attaching them using the library() function get in touch with me as soon as possible. This should not prevent you from completing your assignments.

There is plenty of information online about the ggplot2 package and the ggplot() function. Actually, there is too much information to go into any real detail in this document. The entire series of commands and the respective explanations I’ll use is:

First take the transpose of the desired columns of the original data table to get the data in the proper sequence for the melt command. The melt command is required to create the plot correctly.

> tdealerSat\_NA <- t(dealerSat\_NA[,3:8])

> tdealerSat\_NA

1 2 3 4 5

L0 1 0 1 1 2

L1 0 0 1 2 3

L2 2 2 1 6 5

L3 14 14 8 12 15

L4 22 20 34 34 44

L5 11 14 15 45 56

Next, add the years as column names to get these in the proper sequence in the melded data.

> colnames(tdealerSat\_NA) <- c("2010", "2011", "2012", "2013", "2014")

> tdealerSat\_NA

2010 2011 2012 2013 2014

L0 1 0 1 1 2

L1 0 0 1 2 3

L2 2 2 1 6 5

L3 14 14 8 12 15

L4 22 20 34 34 44

L5 11 14 15 45 56

Use the melt() function to “melt” the data, i.e. put it in the proper sequence for plotting using ggplot(). The melt() function is part of the reshape2 package which you may need to install. Notice that all the levels and counts are sorted by each year. You can check this against the original data to make sure you’ve got the proper sequencing.

> data.m2 <- melt(tdealerSat\_NA, id.vars=var1)

> data.m2

Var1 Var2 value

1 L0 2010 1

2 L1 2010 0

3 L2 2010 2

4 L3 2010 14

5 L4 2010 22

6 L5 2010 11

7 L0 2011 0

8 L1 2011 0

9 L2 2011 2

10 L3 2011 14

11 L4 2011 20

12 L5 2011 14

13 L0 2012 1

14 L1 2012 1

15 L2 2012 1

16 L3 2012 8

17 L4 2012 34

18 L5 2012 15

19 L0 2013 1

20 L1 2013 2

21 L2 2013 6

22 L3 2013 12

23 L4 2013 34

24 L5 2013 45

25 L0 2014 2

26 L1 2014 3

27 L2 2014 5

28 L3 2014 15

29 L4 2014 44

30 L5 2014 56

Add column names to the melded data in order to get the proper axis and legend labels in the plot.

> colnames(data.m2) <- c("Level", "Year", "Counts")

> data.m2

Level Year Counts

1 L0 2010 1

2 L1 2010 0

3 L2 2010 2

4 L3 2010 14

5 L4 2010 22

6 L5 2010 11

7 L0 2011 0

8 L1 2011 0

9 L2 2011 2

10 L3 2011 14

11 L4 2011 20

12 L5 2011 14

13 L0 2012 1

14 L1 2012 1

15 L2 2012 1

16 L3 2012 8

17 L4 2012 34

18 L5 2012 15

19 L0 2013 1

20 L1 2013 2

21 L2 2013 6

22 L3 2013 12

23 L4 2013 34

24 L5 2013 45

25 L0 2014 2

26 L1 2014 3

27 L2 2014 5

28 L3 2014 15

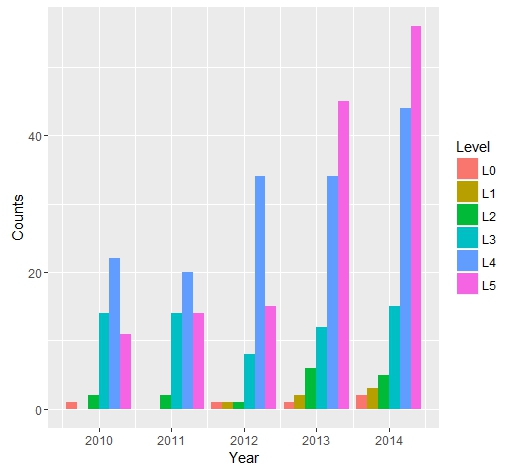
29 L4 2014 44

30 L5 2014 56

Create the plot with the ggplot() function. You can look this up in help() in RStudio or online.

> ggplot(data.m2, aes(x=Year, y=Counts)) + geom\_bar(aes(fill=Level), position="dodge", stat="identity")

The plot this produces is:

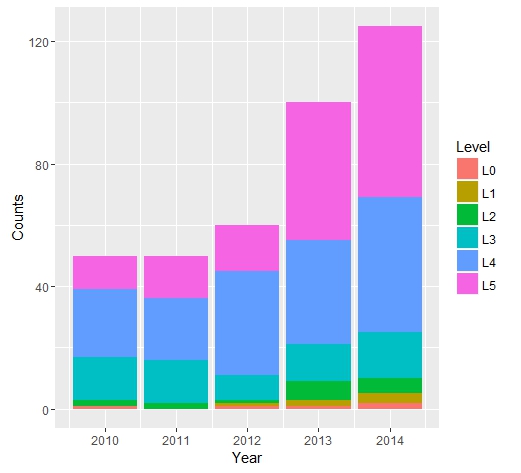


where the number of counts for a particular level of satisfaction for the given year is shown.

Once you have your data in the proper sequence for plotting everything else is easy. To create the stacked bar charts use:

> ggplot(data.m2, aes(x=Year, y=Counts, fill=Level)) + geom\_bar(stat="identity")

The resulting plot is:



You can follow this same procedure for each Region for Dealer Satisfaction and End-User Satisfaction.

### Step 2

There are also many ways in R to do the line plots. I’ll show you an example of a “brute force” method using the simple plot() function as follows:

> plot(Complaints$World, ylim=range(c(0,400)), type="l", xlab="Month", ylab="Number of Complaints")

> par(new=TRUE)

> plot(Complaints$NA., ylim=range(c(0,400)), type="l", col="red", axes = FALSE, xlab = "", ylab = "")

> par(new=TRUE)

> plot(Complaints$SA, ylim=range(c(0,400)), type="l", col="green", axes = FALSE, xlab = "", ylab = "")

> par(new=TRUE)

> plot(Complaints$Eur, ylim=range(c(0,400)), type="l", col="blue", axes = FALSE, xlab = "", ylab = "")

> par(new=TRUE)

> plot(Complaints$Pac, ylim=range(c(0,400)), type="l", col="magenta", axes = FALSE, xlab = "", ylab = "")

> par(new=TRUE)

> plot(Complaints$China, ylim=range(c(0,400)), type="l", col="deeppink4", axes = FALSE, xlab = "", ylab = "")

In this case the axes labels are entered in the first command. The command “par(new=TRUE)” follows each line added to the plot so that you can continue to add lines to the same plot. After the first line the parameters “axes=FALSE, xlab=””, ylab=””” are added so that the axes labels are not continuously overwritten. You can use this or any other plotting methods, e.g. ggplot(), to create the line plots to finish Part 1 of Chapter 3.

## Part 2

For this part of the exercise you are tasked with comparing the costs of shipping between existing locations and proposed locations using quartiles. If you have questions about quartiles the textbook can help or there is a lot of information online.

### Step 1

We can lump all the costs for existing plants into one group and all the costs for proposed plants into a second group. Then we can compute the quartiles for shipping costs based on those groups. I would think that an analysis should contain more information about which locations were shipping to which customers in order to minimize the costs. But, we’ll do this the author’s way.

This is actually quite simple in R. Just use the summary() function that we’ve been using to look at our data as follows:

> summary(ShippingCost\_Existing$Mowers)

Min. 1st Qu. Median Mean 3rd Qu. Max.

1.000 1.312 1.480 1.420 1.528 1.720 S

> summary(ShippingCost\_Existing$Tractors)

Min. 1st Qu. Median Mean 3rd Qu. Max.

1.260 1.768 1.840 1.879 2.105 2.340

> summary(ShippingCost\_Proposed$Mowers)

Min. 1st Qu. Median Mean 3rd Qu. Max.

0.910 1.400 1.520 1.514 1.660 1.980

> summary(ShippingCost\_Proposed$Tractors)

Min. 1st Qu. Median Mean 3rd Qu. Max.

1.170 1.775 2.010 1.958 2.170 2.680

## Part 3

In the third part of this exercise you’ve been tasked with developing a summary about customer attributes. This summary is to be built on the average responses from customers in the 2014 Customer Survey. It should be done by region and include frequency distributions, histograms and quartiles as appropriate. The attributes in the survey include: Quality, Ease of Use, Price and Service.

### Step 1

The only new function we’ll use to complete this part of the exercise is the hist() function to create the required histograms. But first we’ll need to subset the data in order to get it in the proper sequence to calculate the averages and frequency distributions.

First, look at the data as usual:

> str(CustomerSurvey2014)

'data.frame': 200 obs. of 5 variables:

$ Region : Factor w/ 4 levels "China","Eur",..: NA NA NA NA NA NA NA NA NA NA ...

$ Quality : int 4 4 4 5 5 5 5 5 4 4 ...

$ Ease.of.Use: int 1 4 5 4 4 5 4 5 4 5 ...

$ Price : int 3 4 4 4 5 3 4 4 4 4 ...

$ Service : int 4 5 3 4 4 5 2 5 5 5 ...

> summary(CustomerSurvey2014)

Region Quality Ease.of.Use

China: 10 Min. :1.000 Min. :1.000

Eur : 30 1st Qu.:4.000 1st Qu.:4.000

Pac : 10 Median :5.000 Median :4.000

SA : 50 Mean :4.395 Mean :4.165

NA's :100 3rd Qu.:5.000 3rd Qu.:5.000

Max. :5.000 Max. :5.000

Price Service

Min. :1.00 Min. :1.00

1st Qu.:3.00 1st Qu.:4.00

Median :4.00 Median :4.00

Mean :3.67 Mean :4.14

3rd Qu.:4.00 3rd Qu.:5.00

Max. :5.00 Max. :5.00

This has already given us the frequency distribution by region. That is, 100 responses or 50% come from North America and so on. But, we have a problem because the data file codes “North America” as “NA” which is a standard phrase in R for a missing value. This makes the straightforward application of functions a mess. There is a compounding problem in that the variable concerned, “Region”, is a factor variable. So…

If you have already imported the data file CustomerSurvey2014.csv into RStudio you will want to remove it using:

> rm(CustomerSurvey2014)

In order to get the “NA” in this file to just be NA you need to have the parameter “stringsAsFactors” equal FALSE. In order to do that, if you are using the pull down menu to import data you need to uncheck the box “Strings as Factors” before you import the data. If you are using the command line use:

> CustomerSurvey2014 <- read.csv("~/MyRWork/data/Evans/CustomerSurvey2014.csv", stringsAsFactors=FALSE)

Unfortunately, this is not all you need to do. R/RStudio will still recognize the text NA as representing missing values but now it is a text string and not the <NA> symbol R/RStudio uses. This means that you can replace “NA” with another abbreviation for North America that will not cause problems with the functions we are trying to execute. For example, if you want to replace “NA” with “NorthA” you can use this and look at what it returns as:

> CustomerSurvey2014[is.na(CustomerSurvey2014)] <- "NorthA"

> str(CustomerSurvey2014)

'data.frame': 200 obs. of 5 variables:

$ Region : chr "NorthA" "NorthA" "NorthA" "NorthA" ...

$ Quality : int 4 4 4 5 5 5 5 5 4 4 ...

$ Ease.of.Use: int 1 4 5 4 4 5 4 5 4 5 ...

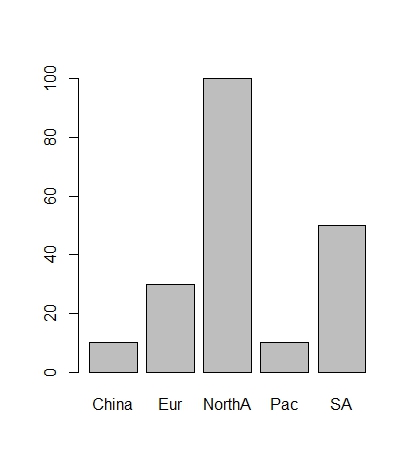
$ Price : int 3 4 4 4 5 3 4 4 4 4 ...

$ Service : int 4 5 3 4 4 5 2 5 5 5 ...

Now, we can easily create a histogram, or because the variable is a factor variable the bar chart, as follows:

> barplot(table(CustomerSurvey2014$Region))

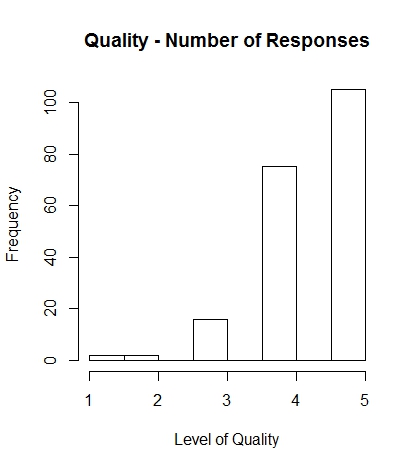
Which produces the plot below.



Because the remaining variables in the CustomerSurvey2014 data file are numeric variables we can use the hist() function as follows:

> hist(CustomerSurvey2014$Quality, main="Quality - Number of Responses", xlab="Level of Quality")

Which produces the following plot.



There are many additional things you can do to make plots in R/RStudio look very professional. I encourage you to explore all the options using the R/RStudio documentation and available information online, e.g. at <https://www.datacamp.com/community/tutorials/15-questions-about-r-plots#gs.kBDIbjY>

### Step 2

The remaining portion of this part of the exercise asks you to compute the Quartiles. You’ve done that before so it isn’t necessary to repeat that here.

## Part 4

You are tasks with proposing a dashboard of the most important business information needed on a routine basis. You are free to complete this part of the exercise as you think best.

# Chapter 4

## Part 1

For the Performance Lawn Equipment case study at the end of Chapter 4 you are tasked with developing the following:

1. The mean satisfaction ratings and standard deviations by year and region in the data files *Dealer Satisfaction* and *End-User Satisfaction*
2. A descriptive statistical summary for the 2014 customer survey data
3. How the response times differ in each quarter of the data file *Response Time*
4. How defects after delivery (data file *Defects after Delivery*) have changed over the years
5. How sales of mowers and tractors compare with industry totals and how strongly monthly product sales are correlated with industry sales

### Step 1 (Part a)

In order to create the clustered and stacked bar charts in Chapter 3 we created a data object “tdealerSat\_NA”. This was the transpose of the Dealer Satisfaction data for the North America region. We can use that data object to compute the mean satisfaction ratings and standard deviations per year. First, let’s recall what the tdealerSat\_NA data object returns, with column headings:

> tdealerSat\_NA

2010 2011 2012 2013 2014

L0 1 0 1 1 2

L1 0 0 1 2 3

L2 2 2 1 6 5

L3 14 14 8 12 15

L4 22 20 34 34 44

L5 11 14 15 45 56

These are frequencies at the different levels. Here is a brute force approach to computing the mean for North America for the year 2010:

> m\_NA2010 <- ((tdealerSat\_NA[2,1]\*1) + (tdealerSat\_NA[3,1]\*2) + (tdealerSat\_NA[4,1]\*3) + (tdealerSat\_NA[5,1]\*4) + (tdealerSat\_NA[6,1]\*5))/sum(tdealerSat\_NA[,1])

> m\_NA2010

[1] 3.78

Because we were only given frequencies for the various levels we really can’t use other somewhat more sophisticated approaches. If you want to try other approaches you will have the same problem with the designation of North American as NA as before. So, you can go through a similar process for the data files *Dealer Satisfaction* and *End-User Satisfaction* to get the designation for North American set to NorthA. The difference is that this time the variable Region is a character variable. So, after we read in the data file with “stringsAsFactors” set to FALSE, we use:

> DealerSatisfaction[is.na(DealerSatisfaction)] <- "NorthA"

To use the brute force approach more easily than the up/down arrows and changing the column number you could write a short R script to loop over the data.

m=0

for (j in 1:25){

# print(j)

m[j] <- ((DealerSatisfaction[j,4]\*1) + (DealerSatisfaction[j,5]\*2) +

(DealerSatisfaction[j,6]\*3) + (DealerSatisfaction[j,7]\*4)

+(DealerSatisfaction[j,8]\*5))/sum(DealerSatisfaction[j,3:8])

# print(m[j])

}

print(m)

which gives you the mean values for all the regions for all the years in a vector m. To get this in a matrix use:

> n <- matrix(m, 5, byrow=FALSE)

> n

[,1] [,2] [,3] [,4] [,5]

[1,] 3.780000 4.000000 3.933333 3.200000 3.000000

[2,] 3.920000 4.000000 4.000000 3.400000 3.142857

[3,] 3.966667 4.266667 4.120000 3.666667 3.687500

[4,] 4.110000 4.500000 4.066667 4.100000 NA

[5,] 4.112000 4.500000 4.066667 3.833333 NA

where the first column is North America, the second is South America and so on, the first row is 2010, the second 2011, and so on. Note that an artifact of looping through the data is the years 2010 and 2011 for China actually appear at the bottom of the column for China.

We can follow the same process to get the standard deviations and in fact combine the two processes. We’ll use the calculation for standard deviation using the mean. If you do not know what the calculation for this is you should look it up and understand how this works. The R script is:

m=0

n=0

for (j in 1:25){

# print(j)

m[j] <- ((DealerSatisfaction[j,4]\*1) + (DealerSatisfaction[j,5]\*2) +

(DealerSatisfaction[j,6]\*3) + (DealerSatisfaction[j,7]\*4)

+(DealerSatisfaction[j,8]\*5))/sum(DealerSatisfaction[j,3:8])

n[j] <- sqrt(((DealerSatisfaction[j,3] \* (0 - m[j])^2) +

(DealerSatisfaction[j,4] \* (1 - m[j])^2) +

(DealerSatisfaction[j,5] \* (2 - m[j])^2) +

(DealerSatisfaction[j,6] \* (3 - m[j])^2) +

(DealerSatisfaction[j,7] \* (4 - m[j])^2) +

(DealerSatisfaction[j,8] \* (5 - m[j])^2))/

(sum(DealerSatisfaction[j,3:8])-1))

# print(m[j])

# print(n[j])

}

print(m)

print(n)

So, we have the standard deviations in a vector n[j] which we can convert to a matrix as before (note that since our std’s are now in the matrix n we’ll increment our naming to p):

> p <- matrix(n, 5, byrow = FALSE)

> p

[,1] [,2] [,3] [,4] [,5]

[1,] 0.9749935 0.6666667 0.8837151 0.8366600 NaN

[2,] 0.8533248 0.6666667 0.8451543 0.8944272 0.6900656

[3,] 0.9382036 0.8276820 0.7257180 1.0327956 0.7932003

[4,] 1.0720979 0.8630747 0.6396838 0.7378648 NA

[5,] 1.0940897 0.9149200 0.7396800 0.8348471 NA

Now that you have a process you can calculate the values for End-User Satisfaction and report your findings.

### Step 2 (Part b)

Getting descriptive statistics in R is very easy. There are a number of packages that have been built to handle this. Let’s use the psych package. You’ll need to install and attach this package. It will provide the following:

* item name
* item number (vars)
* number of valid cases (n)
* mean
* standard deviation (sd)
* trimmed mean (with trim defaulting to .1) (trimmed)
* median (standard or interpolated (mad)
* mad: median absolute deviation (from the median)
* minimum (min)
* maximum (max)
  + and range
* skew
* kurtosis
* standard error (se)

For example, the “item name” will be Quality or Ease of Use, etc. The “item number” is really irrelevant right now. The “number of cases” is the number of observations, e.g. for North America will be 100. You may need to do a few more calculations, e.g. subtracting the minimum from the maximum to get the range. However, it is really easy to get descriptive statistics in R.

The hard part, as usual, is getting the data in the correct format or sequence to use. In this case we’ll need to subset the customer survey data by region in order to complete this part of the exercise. We can use the following to do that:

> custSurveyNA <- CustomerSurvey2014[1:100,-1]

> custSurveySA <-CustomerSurvey2014[102:151,-1]

and so on. The mode is not something that is typically a part of an R package but there is a lot of information about finding the mode in R online. I’ll leave that for you to find the way you prefer to do it. We’ve already used different ways of finding the sum and count. I’ll also leave that for you to review and determine which way you prefer.

Once you have the data subset into data objects then just use the describe() function in the psych package as follows:

> describe(custSurveySA)

vars n mean sd median trimmed mad min max range skew kurtosis

Quality 1 50 4.26 0.78 4 4.38 0.74 1 5 4 -1.49 4.07

Ease.of.Use 2 50 3.94 0.74 4 4.00 0.00 1 5 4 -1.39 3.92

Price 3 50 3.54 1.07 4 3.60 1.48 1 5 4 -0.59 -0.49

Service 4 50 4.20 0.83 4 4.30 1.48 1 5 4 -1.20 2.30

se

Quality 0.11

Ease.of.Use 0.10

Price 0.15

Service 0.12

Another package and function to use are the pastecs package and stat.desc() function. For the same data the stat.desc() function returns:

> stat.desc(custSurveySA)

Quality Ease.of.Use Price Service

nbr.val 50.0000000 50.0000000 50.0000000 50.0000000

nbr.null 0.0000000 0.0000000 0.0000000 0.0000000

nbr.na 0.0000000 0.0000000 0.0000000 0.0000000

min 1.0000000 1.0000000 1.0000000 1.0000000

max 5.0000000 5.0000000 5.0000000 5.0000000

range 4.0000000 4.0000000 4.0000000 4.0000000

sum 213.0000000 197.0000000 177.0000000 210.0000000

median 4.0000000 4.0000000 4.0000000 4.0000000

mean 4.2600000 3.9400000 3.5400000 4.2000000

SE.mean 0.1099536 0.1046276 0.1517517 0.1178030

CI.mean.0.95 0.2209600 0.2102571 0.3049564 0.2367340

var 0.6044898 0.5473469 1.1514286 0.6938776

std.dev 0.7774894 0.7398290 1.0730464 0.8329931

coef.var 0.1825093 0.1877739 0.3031205 0.1983317

So, in these descriptive statistics you get the mean of the Confidence Interval, the sum, the variance as well as the standard deviation and so on.

### Step 3 (Part c)

The data file Response Time is already set-up by quarters as follows:

> str(ResponseTime)

'data.frame': 50 obs. of 8 variables:

$ Q1.2013: num 4.36 5.42 5.5 2.79 5.55 3.65 8.02 4 3.34 4.92 ...

$ Q2.2013: num 4.33 4.73 1.63 4.21 6.89 0.92 5.27 0.9 3.85 5 ...

$ Q3.2013: num 3.71 2.52 2.69 3.47 5.12 1 3.44 6.04 2.53 2.39 ...

$ Q4.2013: num 4.44 4.07 5.11 3.49 4.69 6.36 8.26 1.91 8.93 6.85 ...

$ Q1.2014: num 2.75 3.24 4.35 5.58 2.89 5.09 2.33 1.69 3.88 3.39 ...

$ Q2.2014: num 3.45 1.95 2.77 1.83 3.72 4.59 1.17 1.46 1.9 2.95 ...

$ Q3.2014: num 1.67 2.58 3.47 3.12 1 5.4 3.9 4.49 2.06 4.49 ...

$ Q4.2014: num 2.55 2.3 1.04 1.59 3.11 4.05 3.38 1.26 0.9 2.31 ...

So, to find how the response differs by quarter we can look at our descriptive statistics and create a plot of the mean. To do this we’ll calculate and store the means of the quarters in a data object in one command. I’ll just name this data object “mResponseTime”.

> mResponseTime <- c(mean(ResponseTime$Q1.2013), mean(ResponseTime$Q2.2013), mean(ResponseTime$Q3.2013), mean(ResponseTime$Q4.2013), mean(ResponseTime$Q1.2014), mean(ResponseTime$Q2.2014), mean(ResponseTime$Q3.2014), mean(ResponseTime$Q4.2014))

> mResponseTime

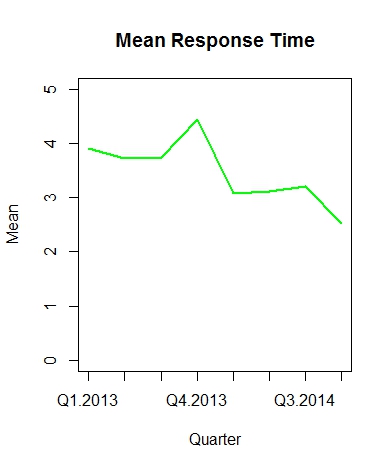
[1] 3.9152 3.7260 3.7472 4.4530 3.0880 3.1136 3.2034 2.5278

Then, we can plot the means by quarter as follows:

> plot(mResponseTime, type = "l", lwd=2, col="green", ylab = "Mean", xlab = "Quarter", main="Mean Response Time", ylim = c(0,5), xaxt="n")

> axis(side = 1, at = c(1:8), labels = alabels, pch=0.5)

To get a plot as follows:



### Step 4 (Part d)

This is very similar to Part c above. We just need to find the means of the defects after delivery over time and plot. Looking at the data we find:

> str(DefectsAfterDelivery)

'data.frame': 12 obs. of 6 variables:

$ Month: Factor w/ 12 levels "April","August",..: 5 4 8 1 9 7 6 2 12 11 ...

$ X2010: int 812 810 813 823 832 848 837 831 827 838 ...

$ X2011: int 828 832 847 839 832 840 849 857 839 842 ...

$ X2012: int 824 836 818 825 804 812 806 798 804 713 ...

$ X2013: int 682 695 692 686 673 681 696 688 671 645 ...

$ X2014: int 571 575 547 542 532 496 472 460 441 445 ...

Again, already set-up as needed. I’ll leave it to you to determine how you want to complete this part of the exercise.

### Step 5 (Part e)

The last part of the exercise for Chapter 4 is a bit different. Now we’re tasked with determining the correlation between PLE’s sales and overall Industry sales by mowers and tractors. As usual, the hardest part will be getting the data in the proper format/sequence to apply a function for computing the correlation. First, looking at the data we find:

> str(MowerUnitSales)

'data.frame': 60 obs. of 8 variables:

$ Month : Factor w/ 12 levels "April","August",..: 5 4 8 1 9 7 6 2 12 11 ...

$ Year : int 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...

$ NA. : int 6000 7950 8100 9050 9900 10200 8730 8140 6480 5990 ...

$ SA : int 200 220 250 280 310 300 280 250 230 220 ...

$ Europe : int 720 990 1320 1650 1590 1620 1590 1560 1590 1320 ...

$ Pacific: int 100 120 110 120 130 120 140 130 130 120 ...

$ China : int 0 0 0 0 0 0 0 0 0 0 ...

$ World : int 7020 9280 9780 11100 11930 12240 10740 10080 8430 7650 ...

> str(IndustryMowerTotalSales)

'data.frame': 60 obs. of 6 variables:

$ Month: Factor w/ 60 levels "Apr-10","Apr-11",..: 21 16 36 1 41 31 26 6 56 51 ...

$ NA. : int 60000 77184 77885 86190 96117 97143 84757 79804 64800 59307 ...

$ SA : int 571 611 658 778 886 882 848 735 657 595 ...

$ Eur : int 13091 17679 22759 27966 27895 30566 29444 28364 28393 24444 ...

$ Pac : int 1045 1111 1068 1237 1313 1176 1359 1238 1215 1154 ...

$ World: int 74662 96585 102369 116171 126210 129768 116409 110141 95065 85500 ..

which is interesting, but involves some work. We really want to combine these data files keeping the month/year variables from the *Mower Unit Sales* data file. Let’s proceed as follows:

> totalMowerSales <- MowerUnitSales[,]

> str(totalMowerSales)

'data.frame': 60 obs. of 8 variables:

$ Month : Factor w/ 12 levels "April","August",..: 5 4 8 1 9 7 6 2 12 11 ...

$ Year : int 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...

$ NA. : int 6000 7950 8100 9050 9900 10200 8730 8140 6480 5990 ...

$ SA : int 200 220 250 280 310 300 280 250 230 220 ...

$ Europe : int 720 990 1320 1650 1590 1620 1590 1560 1590 1320 ...

$ Pacific: int 100 120 110 120 130 120 140 130 130 120 ...

$ China : int 0 0 0 0 0 0 0 0 0 0 ...

$ World : int 7020 9280 9780 11100 11930 12240 10740 10080 8430 7650 ...

> totalMowerSales[,8:12]<- IndustryMowerTotalSales[,-1]

> str(totalMowerSales)

'data.frame': 60 obs. of 13 variables:

$ Month : Factor w/ 12 levels "April","August",..: 5 4 8 1 9 7 6 2 12 11 ...

$ Year : int 2010 2010 2010 2010 2010 2010 2010 2010 2010 2010 ...

$ NA. : int 6000 7950 8100 9050 9900 10200 8730 8140 6480 5990 ...

$ SA : int 200 220 250 280 310 300 280 250 230 220 ...

$ Europe : int 720 990 1320 1650 1590 1620 1590 1560 1590 1320 ...

$ Pacific: int 100 120 110 120 130 120 140 130 130 120 ...

$ China : int 0 0 0 0 0 0 0 0 0 0 ...

$ World : int 7020 9280 9780 11100 11930 12240 10740 10080 8430 7650 ...

$ NA..1 : int 60000 77184 77885 86190 96117 97143 84757 79804 64800 59307 ...

$ SA.1 : int 571 611 658 778 886 882 848 735 657 595 ...

$ Eur : int 13091 17679 22759 27966 27895 30566 29444 28364 28393 24444 ...

$ Pac : int 1045 1111 1068 1237 1313 1176 1359 1238 1215 1154 ...

$ World.1: int 74662 96585 102369 116171 126210 129768 116409 110141 95065 85500 ...

We just need to get our variable names or column headings straightened out as follows:

> colnames(totalMowerSales) <- c("Month", "Year", "NorthA", "SA", "Eur", "Pac", "China", "World", "IndustryNorthA", "IndustrySA", "IndustryEur", "IndustryPac", "IndustryWorld")

> head(totalMowerSales)

Month Year NorthA SA Eur Pac China World IndustryNorthA IndustrySA

1 January 2010 6000 200 720 100 0 7020 60000 571

2 February 2010 7950 220 990 120 0 9280 77184 611

3 March 2010 8100 250 1320 110 0 9780 77885 658

4 April 2010 9050 280 1650 120 0 11100 86190 778

5 May 2010 9900 310 1590 130 0 11930 96117 886

6 June 2010 10200 300 1620 120 0 12240 97143 882

IndustryEur IndustryPac IndustryWorld

1 13091 1045 74662

2 17679 1111 96585

3 22759 1068 102369

4 27966 1237 116171

5 27895 1313 126210

6 30566 1176 129768

and we are set. We can get the coefficient of variance for mower sales using the stat.desc() function as before:

> stat.desc(totalMowerSales)

Month Year NorthA SA Eur

nbr.val NA 6.000000e+01 6.000000e+01 6.000000e+01 6.000000e+01

nbr.null NA 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00

nbr.na NA 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00

min NA 2.010000e+03 4.350000e+03 1.800000e+02 3.000000e+02

max NA 2.014000e+03 1.037000e+04 3.900000e+02 1.650000e+03

range NA 4.000000e+00 6.020000e+03 2.100000e+02 1.350000e+03

sum NA 1.207200e+05 4.525400e+05 1.694000e+04 6.894000e+04

median NA 2.012000e+03 7.870000e+03 2.800000e+02 1.260000e+03

mean NA 2.012000e+03 7.542333e+03 2.823333e+02 1.149000e+03

SE.mean NA 1.841149e-01 2.273237e+02 6.108097e+00 4.870278e+01

CI.mean NA 3.684131e-01 4.548737e+02 1.222227e+01 9.745403e+01

var NA 2.033898e+00 3.100564e+06 2.238531e+03 1.423176e+05

std.dev NA 1.426148e+00 1.760842e+03 4.731312e+01 3.772501e+02

coef.var NA 7.088211e-04 2.334612e-01 1.675789e-01 3.283291e-01

Pac China World IndustryNorthA IndustrySA

nbr.val 6.000000e+01 60.000000 6.000000e+01 6.000000e+01 6.000000e+01

nbr.null 0.000000e+00 51.000000 0.000000e+00 0.000000e+00 0.000000e+00

nbr.na 0.000000e+00 0.000000 0.000000e+00 0.000000e+00 0.000000e+00

min 1.000000e+02 0.000000 5.350000e+03 4.259600e+04 4.620000e+02

max 2.400000e+02 26.000000 1.228000e+04 1.006800e+05 8.860000e+02

range 1.400000e+02 26.000000 6.930000e+03 5.808400e+04 4.240000e+02

sum 1.035000e+04 113.000000 5.488830e+05 4.354853e+06 4.055200e+04

median 1.700000e+02 0.000000 9.390000e+03 7.588300e+04 6.540000e+02

mean 1.725000e+02 1.883333 9.148050e+03 7.258088e+04 6.758667e+02

SE.mean 4.810681e+00 0.709138 2.672965e+02 2.159852e+03 1.343922e+01

CI.mean 9.626151e+00 1.418982 5.348591e+02 4.321854e+03 2.689182e+01

var 1.388559e+03 30.172599 4.286845e+06 2.798977e+08 1.083676e+04

std.dev 3.726338e+01 5.492959 2.070470e+03 1.673014e+04 1.040998e+02

coef.var 2.160196e-01 2.916615 2.263291e-01 2.305034e-01 1.540241e-01

IndustryEur IndustryPac IndustryWorld

nbr.val 6.000000e+01 6.000000e+01 6.000000e+01

nbr.null 0.000000e+00 0.000000e+00 0.000000e+00

nbr.na 0.000000e+00 0.000000e+00 0.000000e+00

min 6.977000e+03 1.045000e+03 5.398200e+04

max 3.056600e+04 2.182000e+03 1.297680e+05

range 2.358900e+04 1.137000e+03 7.578600e+04

sum 1.267206e+06 9.769200e+04 5.760249e+06

median 2.383150e+04 1.552500e+03 9.795500e+04

mean 2.112010e+04 1.628200e+03 9.600415e+04

SE.mean 8.605123e+02 4.272222e+01 2.816721e+03

CI.mean 1.721881e+03 8.548696e+01 5.636245e+03

var 4.442888e+07 1.095113e+05 4.760350e+08

std.dev 6.665499e+03 3.309249e+02 2.181823e+04

coef.var 3.155998e-01 2.032458e-01 2.272634e-01

To find the correlation table and simultaneously find the significance of the correlations we’ll use the Hmisc package. To install this package you may have to install other dependent packages, e.g. acepack and data.table. If you get error messages just look for missing packages and install what you need. Once you have Hmisc installed and attached using the library() function then you can use the rcorr() function to get the correlation table and significance as follows (I’ve highlighted the correlations between PLE’s mower sales and Industry mower sales for SA, Eur and Pac and corresponding P values):

> rcorr(as.matrix(totalMowerSales[3:13]))

NorthA SA Eur Pac China World IndustryNorthA IndustrySA IndustryEur IndustryPac IndustryWorld

NorthA 1.00 0.70 0.70 -0.10 0.21 0.99 1.00 0.85 0.67 -0.08 0.97

SA 0.70 1.00 0.44 0.52 0.50 0.71 0.67 0.76 0.49 0.50 0.68

Eur 0.70 0.44 1.00 -0.33 0.05 0.78 0.69 0.76 0.98 -0.33 0.83

Pac -0.10 0.52 -0.33 1.00 0.44 -0.11 -0.12 -0.09 -0.24 0.99 -0.15

China 0.21 0.50 0.05 0.44 1.00 0.21 0.22 0.23 0.21 0.43 0.24

World 0.99 0.71 0.78 -0.11 0.21 1.00 0.99 0.87 0.76 -0.10 0.99

IndustryNorthA 1.00 0.67 0.69 -0.12 0.22 0.99 1.00 0.83 0.67 -0.10 0.97

IndustrySA 0.85 0.76 0.76 -0.09 0.23 0.87 0.83 1.00 0.76 -0.09 0.87

IndustryEur 0.67 0.49 0.98 -0.24 0.21 0.76 0.67 0.76 1.00 -0.25 0.82

IndustryPac -0.08 0.50 -0.33 0.99 0.43 -0.10 -0.10 -0.09 -0.25 1.00 -0.14

IndustryWorld 0.97 0.68 0.83 -0.15 0.24 0.99 0.97 0.87 0.82 -0.14 1.00

n= 60

P

NorthA SA Eur Pac China World IndustryNorthA IndustrySA IndustryEur IndustryPac IndustryWorl

NorthA 0.0000 0.0000 0.4676 0.1161 0.0000 0.0000 0.0000 0.0000 0.5345 0.0000

SA 0.0000 0.0004 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000 0.0000

Eur 0.0000 0.0004 0.0104 0.6856 0.0000 0.0000 0.0000 0.0000 0.0106 0.0000

Pac 0.4676 0.0000 0.0104 0.0005 0.4020 0.3708 0.5106 0.0594 0.0000 0.2517

China 0.1161 0.0000 0.6856 0.0005 0.1144 0.0946 0.0795 0.1069 0.0007 0.0661

World 0.0000 0.0000 0.0000 0.4020 0.1144 0.0000 0.0000 0.0000 0.4527 0.0000

IndustryNorthA 0.0000 0.0000 0.0000 0.3708 0.0946 0.0000 0.0000 0.0000 0.4287 0.0000

IndustrySA 0.0000 0.0000 0.0000 0.5106 0.0795 0.0000 0.0000 0.0000 0.4921 0.0000

IndustryEur 0.0000 0.0000 0.0000 0.0594 0.1069 0.0000 0.0000 0.0000 0.0517 0.0000

IndustryPac 0.5345 0.0000 0.0106 0.0000 0.0007 0.4527 0.4287 0.4921 0.0517 0.2787

IndustryWorld 0.0000 0.0000 0.0000 0.2517 0.0661 0.0000 0.0000 0.0000 0.0000 0.2787

Because the P values are very low, essentially 0, the correlations are statistically significant. You can follow this same procedure for tractor sales. So, this you can write up in your laboratory report and concludes Chapter 4.

# Chapter 5

## Part 1

For Chapter 5’s Performance Lawn Equipment (PLE) you are tasked with conducting analyses to answer the following questions:

1. For the mower test data, what distribution might be appropriate to model the failure of an individual mower?
2. What fraction of mowers fails the functional performance test using all the mower test data?
3. What is the probability of having x failures in the next 100 mowers tested, for x from 0 to 20?
4. What is the average blade weight and how much variability is occurring in the measurements of blade weights?
5. Assuming that the data are normal, what is the probability that blade weights from this process will exceed 5.20?
6. What is the probability that weights will be less than 4.80?
7. What is the actual percent of weights that exceed 5.20 or are less than 4.80 from the data in the worksheet?
8. Is the process that makes the blades stable over time? That is, are there any apparent changes in the pattern of the blade weights?
9. Could any of the blade weights be considered outliers, which might indicate a problem with the manufacturing process or materials?
10. Was the assumption that blade weights are normally distributed justified? What is the best-fitting probability distribution for the data?

### Step 1

To determine which distribution is appropriate to model the failure of an individual mower consider the section on the Bernoulli Distribution that starts on page 146. Remember that the Bernoulli Distribution has two outcomes; success or failure. So, the answer to Question 1 is the Bernoulli Distribution.

### Step 2

The mower test data has 100 observations and 30 samples per observation. So, you’ll need to read in the csv file for MowerTest. 100 times 30 is 3,000. So, to get the overall failure rate we need to determine how many tests were “Fail”. This is easy is R/RStudio. Use the length() function as follows:

> countFail <- length(which(MowerTest == "Fail"))

> countFail

[1] 54

So there are 54 “Fail” in the MowerTest data set, or the fraction of “Fail”, i.e. the probability of “Fail” is 54/3000 = 0.018.

### Step 3

The next question asks us to find the probability of having from 0 to 20 failures in the next 100 mowers tested. Again, we’ll use the binomial distribution. The R/RStudio function is dbinom(). The entire command is:

> y <- dbinom(0:20, 100, .018)

> y

[1] 1.626106e-01 2.980642e-01 2.704432e-01 1.619354e-01 7.198046e-02

[6] 2.533243e-02 7.352080e-03 1.809677e-03 3.856160e-04 7.225391e-05

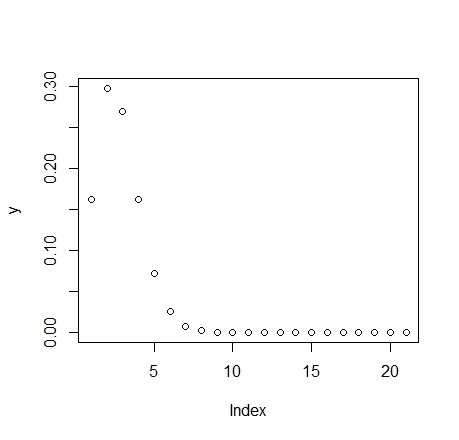
[11] 1.205213e-05 1.807485e-06 2.457222e-07 3.048911e-08 3.472938e-09

[16] 3.649767e-10 3.554063e-11 3.218967e-12 2.720716e-13 2.152308e-14

[21] 1.597793e-15

> plot(y)

And the plot looks like:



### Step 4

Question 4 asks us to find the average blade weight and how much variability there is in blade weights. To answer this question we need the BladeWeight csv file. Read in the data set then use sum() in the command to calculate the average blade weight as follows:

> BladeWeight <- read.csv("~/MyRWork/data/Evans/BladeWeight.csv")

> View(BladeWeight)

> str(BladeWeight)

'data.frame': 350 obs. of 2 variables:

$ Sample: int 1 2 3 4 5 6 7 8 9 10 ...

$ Weight: num 4.88 4.92 5.02 4.97 5 4.99 4.86 5.07 5.04 4.87 ...

> (sum(BladeWeight$Weight)/350)

[1] 4.9908

The variation is calculated as the standard deviation. R/RStudio uses the sd() function for the standard deviation as follows:

> sd(BladeWeight$Weight)

[1] 0.1092876

So, we could expect the blade weight to be 4.99 +/- 2\*0.11. Note that we’ve rounded the standard deviation and assumed a 2-tail solution via the empirical rules.

### Step 5

Question 5 asks us to determine the probability that the blade weight can exceed 5.20. To do this we use the pnorm() function for the Normal Distribution in R/RStudio. The commands are as follows:

> y = pnorm(5.20, mean=4.99, sd=0.11)

> y

[1] 0.9718748

> 1 - y

[1] 0.02812518

### Step 6

Question 6 asks us to determine the probability that the blade weight will be less than 4.80. Again, we’ll use the pnorm() function. The command is:

> pnorm(4.80, mean=4.99, sd=0.11)

[1] 0.04205935

### Step 7

Question 7 asks us to find the number of blades that exceeded 5.20 or were less than 4.80 from the data. We’ll use the length() function again, as follows:

> countblade <- length(which(BladeWeight$Weight > 5.20))

> countblade

[1] 7

> countblade2 <- length(which(BladeWeight$Weight <= 4.80))

> countblade2

[1] 8

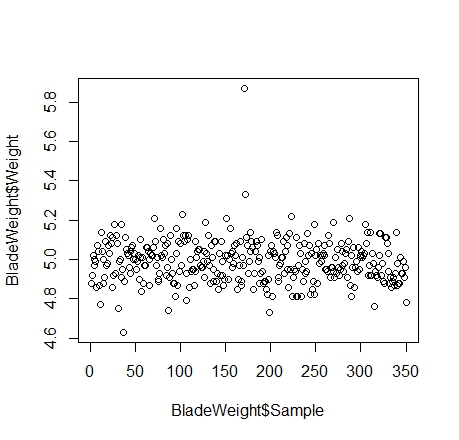
Notice that I could have gotten a different answer if I had set the test to be greater than or equal to using the >= logical operator. Likewise I could have gotten a different answer if I had use < rather than <= as the logical operator in the second computation.

### Step 8

Question 8 asks us to examine, over time, the process that makes the blades by considering changes in blade weights over time. We can just plot the blades manufactured to see if there is any variation over time. We can use the plot() function to generate a scatterplot to look at this as follows:

> plot(BladeWeight$Sample, BladeWeight$Weight)

And, the plot looks like:



From the scatterplot it doesn’t look like there is too much variation about the average blade weight of 4.99.

### Step 9

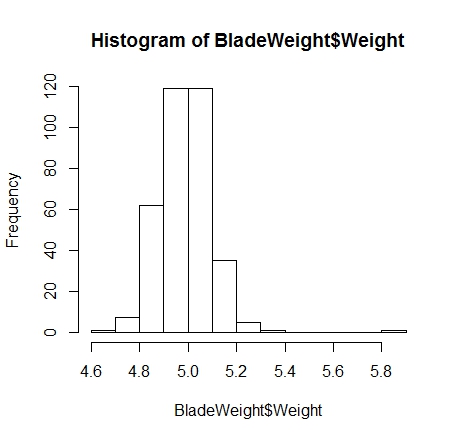
Looking at the scatter plot, only close to the 200th blade was there any trouble. It isn’t too much trouble to find that this is blade #171.

### Step 10

Last, we are asked if the normal distribution is a good assumption for the blade weight data. To do this we’ll want to plot a histogram of the data. Histograms are easy to generate in R/RStudio. Just use the hist() function as follows:

> hist(BladeWeight$Weight)

As is usually true, the hist() function has many additional parameters. You might want to try a few, e.g. setting up the bins the way you want them rather than allowing the function to automatically create bins. The plot looks like:



Which looks pretty normal. If desired you can add a line for the probability density function. I’ll leave it up to you to look up how to do this in R/RStudio.

# Chapter 6

## Part 1

For Chapter 6’s Performance Lawn Equipment (PLE) you are tasked with conducting analyses to answer the following questions:

1. What proportion of customers rate the company with “top box” survey responses (which is defined as scale levels 4 and 5) on quality, east of use, price, and service in the 2014 Customer Survey worksheet? How do these proportions differ by geographic region?
2. What estimates, with reasonable assurance, can PLE give customers for response times to customer service calls?
3. Engineering has collected data on alternative process costs for building transmissions in the worksheet Transmission Costs. Can you determine whether one of the proposed processes is better than the current process?
4. What would be a confidence interval for an additional sample of mower test performance as in the worksheet Mower Test?
5. For the data in the worksheet Blade Weight, what is the sampling distribution of the mean, the overall mean, and the standard error of the mean? Is a normal distribution an appropriate assumption for the sampling distribution of the mean?
6. How many blade weights must be measured to find a 95% confidence interval for the mean blade weight with a sampling error of at most 0.2? What if the sampling error is specified as 0.1?

### Step 1

First, as usual, you’ll need to import the data into R/RStudio. Don’t forget to pay attention to whether or not there are header rows in the file; and, whether or not there may be abbreviations that conflict with R/RStudio’s predefined symbols such as “NA”. If there are such conflicts be sure to either uncheck the “Strings as Factors” box in the Import window or use the parameter “stringsAsFactors=FALSE” in your R command to read.csv().

Count data, such as associated with Binomial and Poisson distributions, are used in frequencies and proportions. Be sure you understand these distributions and what constitute frequencies and proportions. In R/RStudio we can simply create tables that give us the counts, for example:

> countByRegion <- table(CustomerSurvey2014$Region, CustomerSurvey2014$Quality)

> countByRegion

1 2 3 4 5

China 0 1 2 5 2

Eur 0 1 6 12 11

NorthA 1 0 3 30 66

Pac 0 0 1 4 5

SA 1 0 4 24 21

Which gives us, for example, that the North American region has a total of 96 counts of Level 4 or 5 (“Top box” survey responses) for Quality using:

> sum(countByRegion[3,4], countByRegion[3,5])

[1] 96

We can also sum by region (row) to see that there were 100 responses from the North American region. Then it is a simple calculation to find the proportion of “Top box” responses by region. We can also compute the proportion of all customers with “Top box” ratings, which is a simple calculation. You can use these simple sums to compute all the requested information. Of course there are many more sophisticated commands in R that will return or compute and save in a data object the values you want. If you are not used to working with loops, particularly in conjunction with data tables, matrices and/or arrays this simple example will help you sort out it all out by computing the various values required.

### Step 2

To determine how to respond to customers with regard to service response time we need to move more deeply into statistics. The first thing you’ll want to do before you begin any analysis is to thoroughly understand the question, what it is you are going to do. To completely answer this question we’ll need to estimate, with reasonable assurance, the response time for customer service calls. So, the first thing is to establish what we mean by reasonable assurance. For this particular question we can just use Evans’ confidence level, i.e. 95% for this. As we’ve seen from the textbook, that means that our level of significance, or α, is 0.05. For other analyses, you will want to make sure that you have really determined what is meant by “reasonable assurance”. As we have seen in other situations what is reasonable often depends on details inherent to the specific situation.

Because we want to give customers a time range rather than an exact time for response, we’ll need to calculate the average time as well as the margin of error for the computed average time. You can either calculate the Response Time for each quarter or, you can determine the answers for the overall data set.

First, we’ll have to import our data, ResponseTime.csv. Looking at the data using the str() command we see that we have 50 observations of eight variables, one variable for each quarter over a two year period. We can use the entire 50 observations per quarter to calculate the mean and standard deviation per quarter as follows:

> Q1.13mean <- mean(ResponseTime$Q1.2013)

> Q1.13std <- sd(ResponseTime$Q1.2013)

> Q1.13mean

[1] 3.9152

> Q1.13std

[1] 1.482202

where mean() and sd() are the functions required to find the mean and standard deviation of the specified variable.

In R/RStudio, we can use the Basic Statistics and Data Analysis (BSDA) package to find the confidence interval given the standard deviation and knowing that 95% is the default confidence interval. Don’t forget to use library() to attach BSDA after you have installed the package. I’ll include sample commands and output for both the 1st and 2nd Quarters of 2013 below. First, using the standard deviation computed for the 1st Quarter:

> z.test(ResponseTime$Q1.2013, sigma.x = Q1.13std)

One-sample z-Test

data: ResponseTime$Q1.2013

z = 18.678, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

3.504362 4.326038

sample estimates:

mean of x

3.9152

Then, the commands and output for the 2nd Quarter:

> Q2.13mean <- mean(ResponseTime$Q2.2013)

> Q2.13std <- sd(ResponseTime$Q2.2013)

> z.test(ResponseTime$Q2.2013, sigma.x = Q2.13std)

One-sample z-Test

data: ResponseTime$Q2.2013

z = 13.751, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

3.194914 4.257086

sample estimates:

mean of x

3.726

So, actually we didn’t need to compute the mean separately first. The z.test returns the mean as part of the output with the 95% confidence interval. Also, it is very important to note that the output returns the P value indicating that the analysis is statistically significant in both quarters.

### Step 3

Now you are asked to evaluate the proposed processes for building transmissions and compare those to the current process. The question you are asked to answer is whether or not you can determine if one of the proposed processes is better than the current process.

The data you have to do this with is the TransmissionCosts.csv data. You’ll need to import it and look at it. The data look straightforward with 30 observations of the three variables about the processes; current, Process.A, and Process.B. Again we use a 95% confidence interval for each process then all can be compared. So, we can use the same process as we did for the last question as follows:

> current.sd <- sd(TransmissionCosts$Current)

> z.test(TransmissionCosts$Current, sigma.x = current.sd)

One-sample z-Test

data: TransmissionCosts$Current

z = 34.939, p-value < 2.2e-16

alternative hypothesis: true mean is not equal to 0

95 percent confidence interval:

273.3542 305.8458

sample estimates:

mean of x

289.6

You’ll need to complete the computations for all three processes and include your comparison in your lab report.

### Step 4

This question asks what the confidence interval would be for an additional sample of mower test performance. This involves a bit more than the last two questions. Now we’re asked for the confidence interval for an additional sample of mower test performance. Because we do not have data for the entire population, i.e. all tests on all mowers, we cannot calculate the standard deviation for the entire population of mowers. So, we need to use the t-distribution. (Note, I’ll agree with you before the question is raised that the way some of these questions are worded can be very confusing. In fact, in order to determine exactly why we need to use the t-distribution for this question I had to re-review the PLE Case Study notes for all chapters. For quiz questions I will endeavor to be clearer.)

We started looking at the Mower Test data in Chapter 5. There we found that the mean fraction of failures were 54/3000 or 0.018. If we wanted to we could use the mean to find the standard deviation using the formula:

This gives us the value 0.0024273..., which we can round to 0.0024. But, since we cannot compute the mean of the entire population this doesn’t help us a lot here.

The t.test() function parameters we will need include the mower test data and the mean, which we already have. But before we can execute the t.test() function we will need to get the mower test data in the proper format. There are many, many ways you can do this. What you want to have in the end is essentially a single vector with all the data in it. You could play around with unlist() and other functions. I did the following:

A <- data.frame(lapply(MowerTest, function(x) as.numeric(x)))

str(A)

A[,5] <- 2

A[,10] <- 2

A[,27] <- 2

A[,30] <- 2

str(A)

B <- data.matrix(A[,-1])

str(B)

B <- B - 1

str(B)

length(which(B==0))

mean(B)

1-mean(B)

t.test(B, mu=0.982)

1-.9772398

1-.9867602

1. I converted the Pass/Fail levels to numeric values

> A <- data.frame(lapply(MowerTest, function(x) as.numeric(x)))

> str(A)

'data.frame': 100 obs. of 31 variables:

$ Observation: num 1 2 3 4 5 6 7 8 9 10 ...

$ Sample.1 : num 2 2 2 2 2 2 2 2 2 2 ...

$ Sample.2 : num 1 1 2 2 2 2 2 2 2 2 ...

$ Sample.3 : num 2 2 2 2 2 2 2 2 2 2 ...

$ Sample.4 : num 1 1 1 1 1 1 1 1 1 1 ...

$ Sample.5 : num 2 2 2 2 2 2 2 2 2 2 ...

1. I found that for some reason not all cells have converted correctly. For example, Sample.4 looks like it only contains “Fail” tests. So, I went through and converted variables, e.g. Sample.4, that looked suspicious to the correct numeric value, i.e. 2, using:

> A[,5] <- 2

keepin in mind that column 1 is the Observation number so Sample.4 is in column 5.

1. Then I corrected the data so that Pass = 1 and Fail = 0 (rather than Pass = 2 and Fail = 1) by:

> B <- A-1

1. Next, I checked a couple things to make sure my data was ok. I checked that I had the correct total number of failures. And, I checked the mean, and the mean fraction of failures, as follows:

> length(which(B == 0))

[1] 55

Remember that the variable Observation contains the value 1 also so there are a total of 55 1’s in the data set but only 54 1’s that indicate “Fail”.

> mean(B)

[1] 0.982

> 1-mean(B)

[1] 0.018

Believing that the data is ready to process it only takes the simple command below to complete the t.test() function:

> t.test(B, mu=0.982)

One Sample t-test

data: B

t = 0, df = 2999, p-value = 1

alternative hypothesis: true mean is not equal to 0.982

95 percent confidence interval:

0.9772398 0.9867602

sample estimates:

mean of x

0.982

Having completed the analysis this way there is one last thing to do to get the final answer. The output from the t.test() function is the answer for “Pass”. Keeping in mind that we want the interval representing the failures we need to subtract the values for the 95% confidence interval from 1.0 as follows:

> 1-.9772398

[1] 0.0227602

> 1-.9867602

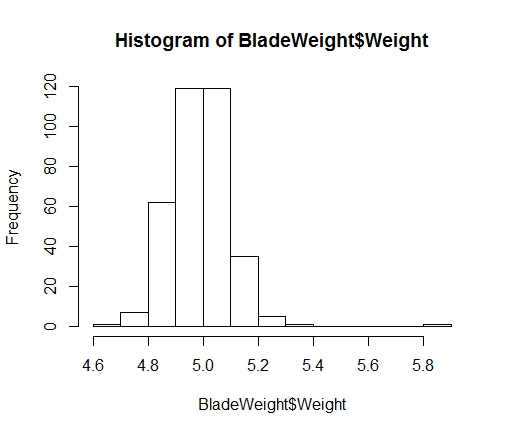
[1] 0.0132398

## Step 5

If you have exited R/RStudio and need to you should import the data set BladeWeight.csv again. There are several questions that we need to answer to complete this. First, we need to determine what the sampling distribution of the mean is. This is pretty easy to do using the hist() function. Simply enter:

> hist(BladeWeight$Weight)

and the plot reveals:



which appears to be a normal distribution. The mean is easy to compute as follows:

> mean(BladeWeight$Weight)

[1] 4.9908

The standard error of the mean for our blade weight data can be found using:

> sem <- sd(BladeWeight$Weight)/sqrt(length(BladeWeight$Weight))

> sem

[1] 0.005841666

If we want to determine if the normal distribution is an appropriate assumption one thing we can do is look at a normal probability plot also called a Q-Q plot. Remember that a normal probability plot is a routine way of testing normality in statistics. To make this easy I’m going to apply a linear model to the blade weight data and use R/RStudio’s built in plotting functions to create the normal probability plot. So:

> A <- lm(BladeWeight$Weight ~ ., BladeWeight)

where I established a data object “A” for the output from the lm() function, i.e. the linear model function in R/RStudio. Assuming the data are normal and linear or close to linear that should give us a good fit on a normal probability plot. So I:

> plot(A)

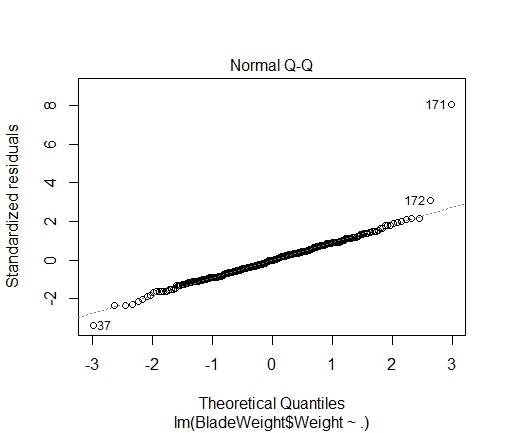
Hit <Return> to see next plot:

Hit <Return> to see next plot:

Hit <Return> to see next plot:

Hit <Return> to see next plot:

and since I’m most interest in the normal probability plot here it is:



which really looks good (all data fall on the line) except for the couple outliers we’ve talked about before. So, I would argue that yes, a normal distribution is appropriate.

## Step 6

You are asked to find the answers to two questions in this last part of Chapter 6, i.e. “How many blade weights must be measured to find a 95% confidence interval for the mean blade weight with a sampling error of at most 0.2? What if the sampling error is specified as 0.1?” The equation to determine n for a given sampling error is in the textbook. We had already found that the average blade weight was 4.99 and the standard deviation was 0.11. To apply this equation we need to determine the value of z for the specified level of significance, . But, in actuality the calculation for this value is well understood and the value is already known. It is 1.96. If this doesn’t sound familiar be sure to review because this is one of a set of standard values that will very likely appear on exams.

So, to get n, the number of samples required we just apply what we know to the equation:

but since we can’t have a partial sample we’ll round up to 2. You can use this same approach to find the reduced sample error = 0.1. You should take a minute to think about this problem in general. For example, if you are reducing the sampling error would you expect a larger number of samples to be required or a smaller number?

**Chapter 7**

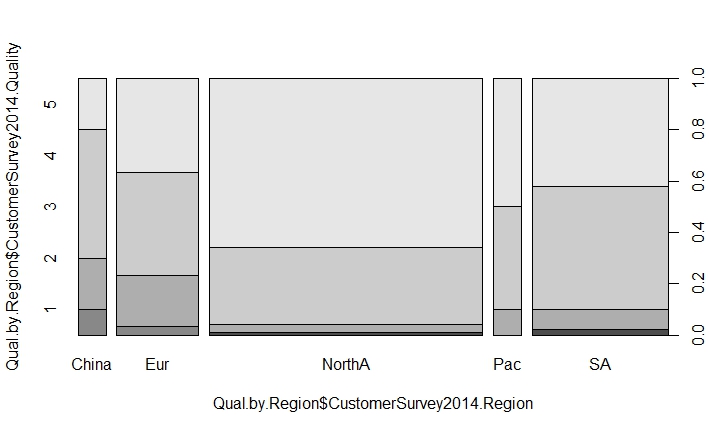
For Chapter 7’s Performance Lawn Equipment (PLE) you are tasked with finding answers for the following questions:

1. Are there significant differences in ratings of specific product/service attributes in the *2014CustomerSurvey.csv* ~~worksheet~~ data file?
2. In the ~~worksheet~~ data file OnTimeDelivery.csv, has the proportion of on-time deliveries in 2014 significantly improved since 2010?
3. Have the data in the ~~worksheet~~ data file *DefectsAfterDelivery.csv* changed significantly over the past 5 years?
4. Although engineering has collected data on alternative process costs for building transmissions in the ~~worksheet~~ data file *TransmissionCosts.csv,* why didn’t they reach a conclusion as to whether one of the proposed processes is better than the current process?
5. Are there differences in employee retention due to gender, college graduation status, or whether the employee is from the local area in the data in the ~~worksheet~~ data file *EmployeeRetention.csv?*

**Part 1**

We have already looked at most of the data required for Chapter 7 analyses before. However, now we’ll start making more sophisticated analyses using that data. First, we’re asked to make an evaluation across variables in the 2014 Customer Survey data, i.e. across Quality, Ease of Use, Price and Service. One of the questions you need to think about before you start is whether you want to do a multivariable linear regression or use analysis of variance (ANOVA). If you don’t recall what the differences between multivariable analysis and ANOVA are you will want to review that. In short, because you are asked to make comparisons across variables you’ll want to use ANOVA.

You won’t need to worry about the “NA” being a special symbol in R/RStudio because we are not going to include regions in this analysis. This is really kind of a “cheat”. It appears to me that this was done to avoid the complication of having an “unbalanced” design, i.e. having unequal sample sizes which is true if we do the analysis by region where North America has 100 samples, South America has 50 samples, and so on. This would be true if we compared customer responses about ease of use, for example, across regions. Consider the plot below in which there are obvious differences between the means of the groups.



To perform a one-way ANOVA in R/RStudio that includes all 200 samples requires a couple steps. The first step is similar to the kind of pre-processing we did in Chapter 3 in order to get the data in the proper format for our final analysis. That means that we need to put the data in two columns, the first column will be the actual numeric values, the second will be the “group,” which are really the variables; Quality, Ease of Use, etc. First, I set-up the variables. Then, I’m going to use the column heading “Y” for the actual values of the variables and “Para” (which I have shortened from parameter) for the column heading of the groups. Then I create a data.frame and look at it as follows:

> Quality <- CustomerSurvey2014$Quality

> Ease.of.Use <- CustomerSurvey2014$Ease.of.Use

> Price <- CustomerSurvey2014$Price

> Service <- CustomerSurvey2014$Service

> Data <- data.frame(Y=c(Quality, Ease.of.Use, Price, Service), Para=factor(rep(c("Quality", "Ease.of.Use", "Price", "Service"), times=c(length(Quality), length(Ease.of.Use), length(Price), length(Service)))))

> str(Data)

'data.frame': 800 obs. of 2 variables:

$ Y : int 4 4 4 5 5 5 5 5 4 4 ...

$ Para: Factor w/ 4 levels "Ease.of.Use",..: 3 3 3 3 3 3 3 3 3 3 ...

Then I run the aov() function to perform the analysis of variance and list the output as follows:

> fm1 <- aov(Y~Para, data = Data)

> anova(fm1)

Analysis of Variance Table

Response: Y

Df Sum Sq Mean Sq F value Pr(>F)

Para 3 55.51 18.502 23.691 1.079e-14 \*\*\*

Residuals 796 621.65 0.781

---

Signif. codes: 0 ‘\*\*\*’ 0.001 ‘\*\*’ 0.01 ‘\*’ 0.05 ‘.’ 0.1 ‘ ’ 1

There are several websites I found that state you can do the same analysis with different sample sizes. I did not try any of the proposed methods to prove it one way or the other.

**Part 2**

If you need to you should import the OnTimeDelivery.csv data again. Before you start think about what you need to know to do the analysis. In the data we have the number of deliveries and the number of on-time deliveries for each of the years 2010 through 2014. In addition, you should always check that all the assumptions for hypothesis testing are true. We’ll set this up as follows. First, we can calculate the proportion of on-time deliveries in 2010 as:

> P0 <- sum(OnTimeDelivery$Number.On.Time[1:12])/sum(OnTimeDelivery$Number.of.deliveries[1:12])

[1] 0.9850219

We can also calculate the proportion of on-time deliveries for 2014 as:

> pbar <- sum(OnTimeDelivery$Number.On.Time[49:60])/sum(OnTimeDelivery$Number.of.deliveries[49:60])

[1] 0.9906526

which would seem to be an answer. However, we can conduct a bit more sophisticated analysis that will give us more confidence in the answer. We can use hypothesis testing to determine if this change is statistically significant or not. If you don’t remember whether or not you will set-up a lower- or upper- tail test review your statistics notes or use the “Cliff Notes” document uploaded to Moodle. We can do the analysis either or both of two ways:

> n <- 12

> z <- (pbar-p0)/sqrt(p0\*(1-p0)/n)

> z

[1] 0.1605825

> alpha <- .05

> z.alpha <- qnorm(1-alpha)

> z.alpha

[1] 1.644854

In this case the test statistic, z, is not greater than the critical value, z.alpha. Therefore, we do not reject the null hypothesis and cannot determine if there is a statistically significant improvement in on-time deliveries. If you don’t remember why we would use the z test statistic here you should review that. The second method is:

> pval <- pnorm(z, lower.tail = FALSE)

> pval

[1] 0.4362111

Here again, at a pval = 0.44, we do not reject the null hypothesis because the p-value is greater than 0.05. Essentially we cannot answer the question.

**Part 3**

Once you have imported the DefectsAfterDelivery.csv data you could create a plot of that data by year as follows:

(Calculate the defects by year either before or inside the command)

> sum(DefectsAfterDelivery$X2014)/12

[1] 496.25

> defects.by.year <- c(826.33, 837.42, 785.92, 669.08, 496.25)

> plot(defects.by.year, type="b", lwd=2)

In this case we can only assume that the numbers we are given are the total number of monthly shipments. We do not know this is true. As always, we should check that all the assumptions are true, e.g. equal variances. If we set-up the year 2010 as “a” and 2014 as “b”, then we can run a two-sample t-test assuming unequal variances as follows:

> t.test(a,b, var.equal = FALSE, paired = FALSE)

Welch Two Sample t-test

data: a and b

t = 20.619, df = 12.011, p-value = 9.631e-11

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

295.2067 364.9600

sample estimates:

mean of x mean of y

826.3333 496.2500

Which yields a very small p-value. So, we reject the null hypothesis that there is no difference and conclude that there has been a statistically significant change in the number of defects.

**Part 4**

We are asked to evaluate the new processes for building transmissions (Process A and Process B) relative to the current process and determine which is best. We can complete this evaluation in two steps; first we compare Process A to the current process; then, second we compare Process B to the current process.

> t.test(TransmissionCosts$Current, TransmissionCosts$Process.A, var.equal = FALSE, paired = FALSE)

Welch Two Sample t-test

data: TransmissionCosts$Current and TransmissionCosts$Process.A

t = 0.2834, df = 51.88, p-value = 0.778

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-24.93166 33.13166

sample estimates:

mean of x mean of y

289.6 285.5

> t.test(TransmissionCosts$Current, TransmissionCosts$Process.B, var.equal = FALSE, paired = FALSE)

Welch Two Sample t-test

data: TransmissionCosts$Current and TransmissionCosts$Process.B

t = -0.96832, df = 40.728, p-value = 0.3386

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-27.259977 9.593311

sample estimates:

mean of x mean of y

289.6000 298.4333

In both cases the p-value is greater than .05. Therefore we cannot conclude that there is a statistically significant improvement due to either of the new processes relative to the current process.

**Part 5**

We’ll follow the same logic to answer the questions in this part as we have been. We’ll need the EmployeeRetention.csv data. And, as we have had to do before we’ll need to pre-process the data a bit to get it in the proper format for the analyses. First we get the variables we want to test in a data frame:

> gender <- EmployeeRetention$Gender

> yearsPLE <- EmployeeRetention$YearsPLE

> male.female <- data.frame(gender=gender, yearsPLE=yearsPLE)

Then we reorder, or sort the data to get all the female employees and all the male employees together.

> male.female[order(male.female$gender),]

gender yearsPLE

1 F 10.0

4 F 10.0

5 F 9.6

9 F 8.2

14 F 7.2

15 F 6.8

19 F 5.9

24 F 4.7

28 F 3.9

31 F 3.7

37 F 0.9

39 F 0.7

40 F 0.3

2 M 10.0

3 M 10.0

6 M 8.5

7 M 8.4

8 M 8.4

10 M 7.9

11 M 7.6

12 M 7.5

13 M 7.5

16 M 6.5

17 M 6.3

18 M 6.2

20 M 5.8

21 M 5.4

22 M 5.1

23 M 4.8

25 M 4.5

26 M 4.3

27 M 4.0

29 M 3.7

30 M 3.7

32 M 3.5

33 M 3.4

34 M 2.5

35 M 1.8

36 M 1.5

38 M 0.8

Last we perform the actual test:

> t.test(male.female$yearsPLE ~ male.female$gender, var.equal = FALSE, paired = FALSE)

Welch Two Sample t-test

data: male.female$yearsPLE by male.female$gender

t = -0.0091748, df = 18.311, p-value = 0.9928

alternative hypothesis: true difference in means is not equal to 0

95 percent confidence interval:

-2.290560 2.270617

sample estimates:

mean in group F mean in group M

5.530769 5.540741

Once again, the p-value is greater than .05 who we conclude that there is no statistically significant difference between men and women in terms of the number of years of service at PLE.

# Summary – Your Laboratory Report

To get credit for this laboratory you must submit a laboratory report on Moodle. You should use typical graduate style writing, e.g. APA style or similar. You should only include enough pages to report the results you get using R to answer the questions in the assigned Performance Lawn Equipment Case Study for each chapter, chapters 1 through and including 7.

In addition, you should include as an appendix a copy of the R coding you used. You can simply copy and paste your commands and results into the appendices of a Word document to satisfy this requirement.

Here are the questions which are supposed to be answered in your report:

Chapter 1 Questions:

AS you may have seen in text book, Mrs. Bruke, your client has provided the dataset for you (excel file). To prepare for this task, you have decided to review each worksheet and determine whether the data were gathered from internal sources, external sources, or have been generated from special studies. Also, you need to know whether the measures are categorical, ordinal, interval, or ratio. Prepare a report summarizing the characteristics of the metrics used in each worksheet.

Chapter 3 Questions:

[**Part 1**](http://e.pub/abir4zjwxxqarzd6pt6y.vbk/OPS/xhtml/fileP7000496307000000000000000004815.xhtml#P7000496307000000000000000004815): PLE originally produced lawn mowers, but a significant portion of sales volume over recent years has come from the growing small-tractor market. As we noted in the case in Chapter 1, PLE sells their products worldwide, with sales regions including North America, South America, Europe, and the Pacific Rim. Three years ago a new region was opened to serve China, where a booming market for small tractors has been established. PLE has always emphasized quality and considers the quality it builds into its products as its primary selling point. In the past 2 years, PLE has also emphasized the ease of use of their products.

Before digging into the details of operations, Elizabeth Burke wants to gain an overview of PLE’s overall business performance and market position by examining the information provided in the database. Specifically, she is asking you to construct appropriate charts for the data in the following worksheets and summarize your conclusions from analysis of these charts.

1. *Dealer Satisfaction*
2. *End-User Satisfaction*
3. *Complaints*
4. *Mower Unit Sales*
5. *Tractor Unit Sales*
6. *On-Time Delivery*
7. *Defects after Delivery*
8. *Response Time*

[**Part 2**](http://e.pub/abir4zjwxxqarzd6pt6y.vbk/OPS/xhtml/fileP700049630700000000000000000481F.xhtml#P700049630700000000000000000481F): As noted in the case in Chapter 1, the supply chain worksheets provide cost data associated with logistics between existing plants and customers as well as proposed new plants. Ms. Burke wants you to extract the records associated with the unit shipping costs of proposed plant locations and compare the costs of existing locations against those of the proposed locations using quartiles.

[**Part 3**](http://e.pub/abir4zjwxxqarzd6pt6y.vbk/OPS/xhtml/fileP7000496307000000000000000004829.xhtml#P7000496307000000000000000004829): Ms. Burke would also like a quantitative summary of the average responses for each of the customer attributes in the worksheet *2014 Customer Survey* for each market region as a cross-tabulation (use PivotTables as appropriate), along with frequency distributions, histograms, and quartiles of these data.

[**Part 4**](http://e.pub/abir4zjwxxqarzd6pt6y.vbk/OPS/xhtml/fileP7000496307000000000000000004833.xhtml#P7000496307000000000000000004833)**:** Propose a monthly dashboard of the most important business information that Ms. Burke can use on a routine basis as data are updated. Create one using the most recent data. Your dashboard should not consist of more than 6–8 charts, which should fit comfortably on one screen.

Chapter 4 Questions:

Your client wants some detailed statistical information about much of the data in the PLE database. In particular, she wants to know the following:

1. the mean satisfaction ratings and standard deviations by year and region in the worksheets *Dealer Satisfaction* and *End-User Satisfaction*
2. a descriptive statistical summary for the *2012 customer survey data*
3. how the response times differ in each quarter of the worksheet *Response Time*
4. how defects after delivery (worksheet *Defects after Delivery*) have changed over these 5 years

How sales of mowers and tractors compare with industry totals and how strongly monthly product sales are correlated with industry sales

Chapter 5 Questions:

PLE collects a variety of data from special studies, many of which are related to the quality of its products. The company collects data about functional test performance of its mowers after assembly; results from the past 30 days are given in the worksheet *Mower Test*. In addition, many in-process measurements are taken to ensure that manufacturing processes remain in control and can produce according to design specifications. The worksheet *Blade Weight* shows 350 measurements of blade weights taken from the manufacturing process that produces mower blades during the most recent shift. Elizabeth Burke has asked you to study these data from an analytics perspective. Drawing upon your experience, you have developed a number of questions.

1. For the mower test data, what distribution might be appropriate to model the failure of an individual mower?
2. What fraction of mowers fails the functional performance test using all the mower test data?
3. What is the probability of having x failures in the next 100 mowers tested, for x from 0 to 20?
4. What is the average blade weight and how much variability is occurring in the measurements of blade weights?
5. Assuming that the data are normal, what is the probability that blade weights from this process will exceed 5.20?
6. What is the probability that weights will be less than 4.80?
7. What is the actual percent of weights that exceed 5.20 or are less than 4.80 from the data in the worksheet?
8. Is the process that makes the blades stable over time? That is, are there any apparent changes in the pattern of the blade weights?
9. Could any of the blade weights be considered outliers, which might indicate a problem with the manufacturing process or materials?
10. Was the assumption that blade weights are normally distributed justified? What is the best-fitting probability distribution for the data?

Chapter 6 Questions:

In reviewing your previous reports, several questions came to Elizabeth Burke’s mind. Use point and interval estimates to help answer these questions.

1. What proportion of customer’s rate the company with “top box” survey responses (which is defined as scale levels 4 and 5) on quality, ease of use, price, and service in the *2014 Customer Survey* worksheet? How do these proportions differ by geographic region?
2. What estimates, with reasonable assurance, can PLE give customers for response times to customer service calls?
3. Engineering has collected data on alternative process costs for building transmissions in the worksheet *Transmission Costs*. Can you determine whether one of the proposed processes is better than the current process?
4. What would be a confidence interval for an additional sample of mower test performance as in the worksheet *Mower Test*?
5. For the data in the worksheet *Blade Weight*, what is the sampling distribution of the mean, the overall mean, and the standard error of the mean? Is a normal distribution an appropriate assumption for the sampling distribution of the mean?
6. How many blade weights must be measured to find a 95% confidence interval for the mean blade weight with a sampling error of at most 0.2? What if the sampling error is specified as 0.1?

Chapter 7 Questions:

1) Are there significant differences in ratings of specific product/service attributes in the *2014 Customer Survey* worksheet?

2) In the worksheet On-Time Delivery, has the proportion of on-time deliveries in 2014 significantly improved since 2010?

3) Have the data in the worksheet *Defects After Delivery* changed significantly over the past 5 years?

4) Although engineering has collected data on alternative process costs for building transmissions in the worksheet *Transmission Costs,* why didn’t they reach a conclusion as to whether one of the proposed processes is better than the current process?

5) Are there differences in employee retention due to gender, college graduation status, or whether the employee is from the local area in the data in the worksheet *Employee Retention?*

You can format your Lab Report as follows;

ANLY 500-53 Laboratory #1 Report

Date

Name

# Chapter 1:

Copy the text of the questions before your answers to each question. If you want to try to break up the questions you can use Part 1, Part 2, …, etc. as I tried to in your laboratory #1 documentation. If you need to break things down more, again you can follow what I’ve tried to do and set-up steps for each part. After the text of the question(s) you’ve copied insert your answer(s).

# Appendix 1:

There should be an appendix that corresponds to each chapter. Each appendix should contain a copy of the R/RStudio commands you used to find solutions to the questions. I will cut and paste the commands you’ve included in your appendices. If they do not work then I’ll be in touch. If we can’t resolve any non-working R/RStudio commands I will not be able to give you credit for the related questions/answers.