ANLY 500 Laboratory #1 (Part 3) – Descriptive Statistics

Evans Chapter 4 and 5

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

Contents

[Introduction 2](#_Toc488410844)

[Chapter 4 2](#_Toc488410845)

[Part 1 2](#_Toc488410846)

[Step 1 (Part a) 2](#_Toc488410847)

[Step 2 (Part b) 4](#_Toc488410848)

[Step 3 (Part c) 6](#_Toc488410849)

[Step 4 (Part d) 7](#_Toc488410850)

[Step 5 (Part e) 8](#_Toc488410851)

[Chapter 5 12](#_Toc488410852)

[Part 1 12](#_Toc488410853)

[Step 1 12](#_Toc488410854)

[Step 2 13](#_Toc488410855)

[Step 3 13](#_Toc488410856)

[Step 4 14](#_Toc488410857)

[Step 5 14](#_Toc488410858)

[Step 6 14](#_Toc488410859)

[Step 7 15](#_Toc488410860)

[Step 8 15](#_Toc488410861)

[Step 9 16](#_Toc488410862)

[Step 10 16](#_Toc488410863)

# Introduction

This laboratory follows the exercises in the book, specifically the Performance Lawn Equipment Case Study homework assigned exercises Chapters 3 and 5, except this laboratory requires that you use R to complete the exercises. By the end of this lab you are supposed to answer to following questions:

# 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

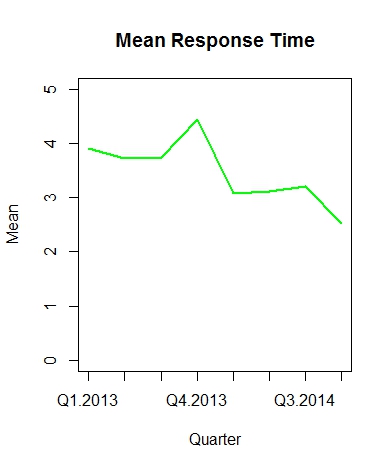
Try to find the value for alabels yourself to get a plot as follows:

> 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)

Try to find the definition of *alabels* yourself.

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 7 variables:

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

$ 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 7 variables:

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

$ 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 12 variables:

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

$ 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("Date", "NorthA", "SA", "Eur", "Pac", "China", "World", "IndustryNorthA", "IndustrySA", "IndustryEur", "IndustryPac", "IndustryWorld")

> head(totalMowerSales)

Date NorthA SA Eur Pac China World IndustryNorthA IndustrySA IndustryEur IndustryPac IndustryWorld

1 Jan-10 6000 200 720 100 0 7020 60000 571 13091 1045 74662

2 Feb-10 7950 220 990 120 0 9280 77184 611 17679 1111 96585

3 Mar-10 8100 250 1320 110 0 9780 77885 658 22759 1068 102369

4 Apr-10 9050 280 1650 120 0 11100 86190 778 27966 1237 116171

5 May-10 9900 310 1590 130 0 11930 96117 886 27895 1313 126210

6 Jun-10 10200 300 1620 120 0 12240 97143 882 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)

Date NorthA SA Eur Pac China World IndustryNorthA IndustrySA IndustryEur

nbr.val NA 6.000000e+01 6.000000e+01 6.000000e+01 6.000000e+01 60.000000 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 51.000000 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 0.000000 0.000000e+00 0.000000e+00 0.000000e+00 0.000000e+00

min NA 4.350000e+03 1.800000e+02 3.000000e+02 1.000000e+02 0.000000 5.350000e+03 4.259600e+04 4.620000e+02 6.977000e+03

max NA 1.037000e+04 3.900000e+02 1.650000e+03 2.400000e+02 26.000000 1.228000e+04 1.006800e+05 8.860000e+02 3.056600e+04

range NA 6.020000e+03 2.100000e+02 1.350000e+03 1.400000e+02 26.000000 6.930000e+03 5.808400e+04 4.240000e+02 2.358900e+04

sum NA 4.525400e+05 1.694000e+04 6.894000e+04 1.035000e+04 113.000000 5.488830e+05 4.354853e+06 4.055200e+04 1.267206e+06

median NA 7.870000e+03 2.800000e+02 1.260000e+03 1.700000e+02 0.000000 9.390000e+03 7.588300e+04 6.540000e+02 2.383150e+04

mean NA 7.542333e+03 2.823333e+02 1.149000e+03 1.725000e+02 1.883333 9.148050e+03 7.258088e+04 6.758667e+02 2.112010e+04

SE.mean NA 2.273237e+02 6.108097e+00 4.870278e+01 4.810681e+00 0.709138 2.672965e+02 2.159852e+03 1.343922e+01 8.605123e+02

CI.mean NA 4.548737e+02 1.222227e+01 9.745403e+01 9.626151e+00 1.418982 5.348591e+02 4.321854e+03 2.689182e+01 1.721881e+03

var NA 3.100564e+06 2.238531e+03 1.423176e+05 1.388559e+03 30.172599 4.286845e+06 2.798977e+08 1.083676e+04 4.442888e+07

std.dev NA 1.760842e+03 4.731312e+01 3.772501e+02 3.726338e+01 5.492959 2.070470e+03 1.673014e+04 1.040998e+02 6.665499e+03

coef.var NA 2.334612e-01 1.675789e-01 3.283291e-01 2.160196e-01 2.916615 2.263291e-01 2.305034e-01 1.540241e-01 3.155998e-01

IndustryPac IndustryWorld

nbr.val 6.000000e+01 6.000000e+01

nbr.null 0.000000e+00 0.000000e+00

nbr.na 0.000000e+00 0.000000e+00

min 1.045000e+03 5.398200e+04

max 2.182000e+03 1.297680e+05

range 1.137000e+03 7.578600e+04

sum 9.769200e+04 5.760249e+06

median 1.552500e+03 9.795500e+04

mean 1.628200e+03 9.600415e+04

SE.mean 4.272222e+01 2.816721e+03

CI.mean 8.548696e+01 5.636245e+03

var 1.095113e+05 4.760350e+08

std.dev 3.309249e+02 2.181823e+04

coef.var 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[2:12]))

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

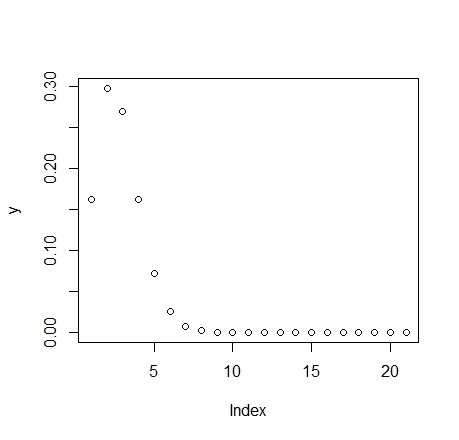
[11] 1.205213e-05 1.807485e-06 2.457222e-07 3.048911e-08 3.472938e-09

[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

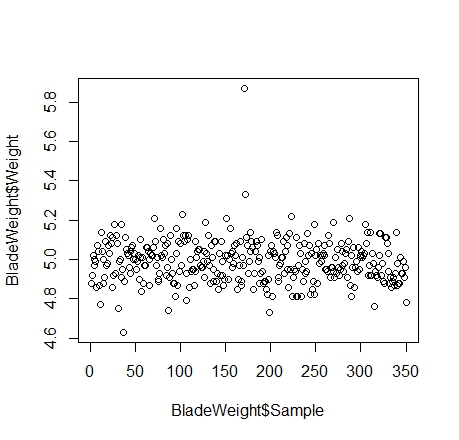
Notice that I could have gotten a different answer if I had set the test to be greater than or equal to using the >= logical operator. Likewise I could have gotten a different answer if I had use < rather than <= as the logical operator in the second computation.

### 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

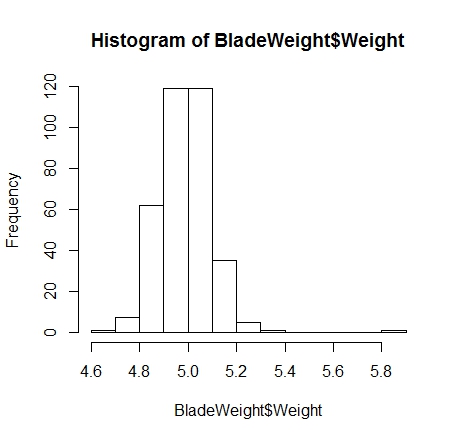
Looking at the scatter plot, only close to the 200th blade was there any trouble. It isn’t too much trouble to find that this is blade #171.

### 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.