Partial answers for the Section 2 Homework

3.7

Get the data into R!

# Change to the folder location on your computer
setwd(dir = "C:\\chris")

set1 <- read.table(file = "ex3-7.txt", header = TRUE, sep = "")
head(set1)

# One could partition out the variables into objects called x and y
 using
x <- set1[,1]
y <- set1[,2]

# Otherwise, just use set1$StandardTherapy and set1$NewTherapy

# Also, one could form the data by typing it directly into a program
temp<-c(4,5,
15,20,
24,29,
10,15,
1,7,
27,32,
31,36,
14,17,
2,15,
16,19,
32,35,
7,10,
13,16,
36,39,
29,27,
6,14,
12,10,
18,16,
14,12,
15,13,
18,16,
6,9,
13,18,
21,33,
20,30,
8,29,
3,31,
24,27)

# Create a row and column format for the data via matrix()
temp2 <- matrix(data = temp, nrow = 28, ncol = 2, byrow = TRUE)
x <- temp2[,1]
y <- temp2[,2]

It probably is easier to read in the data and then extract out the needed variables from the data frame using “$” rather than using x and y objects. However, it is instructive to know the various ways to get data into R!

a.

For the 1st histogram, read data into object x first, then use

hist(x = x, main = "Standard Therapy", xlab = "Survival times (in months)")

Similarly for the 2nd histogram, read data into object y, then use

hist(x = y, main = "New Therapy", xlab = "Survival times (in months)")

Put these histograms in the same plotting window so that there are two rows of plots. Adjust the x-axis on the second plot to match the x-axis of the first plot.

Also, we can make side-by-side box and dot plots:

Therapy <- rep(c("Standard", "New"), each = 28)

p3.6 <- data.frame(time = c(x,y), therapy = Therapy)

boxplot(formula = time ~ therapy, data = p3.6, col = "lightblue",

 main = "Box plot", ylab = "Survival times (in months)", xlab = "Therapy")

stripchart(x = time ~ therapy, data = p3.6, method = "jitter", vertical = TRUE, pch = 1,

 main = "Dot plot", ylab = "Survival times (in months)", xlab = "Therapy")

Another way to make the side-by-side box and dot plots:

boxplot(x = x, col = "lightblue",

 main = "Box plot", ylab = "Survival times (in months)", xlab = "Therapy")

stripchart(x = x, method = "jitter", vertical = TRUE, pch = 1,

 main = "Dot plot", ylab = "Survival times (in months)", xlab = "Therapy")

b. The histogram for the New Therapy appears to have more density in longer survival times (≥ 25) than the histogram for the Standard Therapy, but there is no convincing evidence.

3.21

a. Mean = 8.04, Median = 1.54. Put the Terrestrial feeders data into x, and put the Aquatic feeders data into y, then use

mean(c(x,y))

median(c(x,y))

b. Terrestrial: Mean = 15.01, Median = 6.03

Aquatic: Mean = 0.38, Median = .375

mean(x)

median(x)

mean(y)

median(y)

c. In the terrestrial feeders, there are two large values (76.50 and 41.70). By excluding these two values, the mean decreases radically from 15.01 to 5.21, while the median only changes from 6.03 to 4.24. Thus, the mean is more sensitive to extreme values than the median in a data set.

mean(x[-c(1,8)])

median(x[-c(1,8)])

d. Terrestrial: Median, because the two large values (76.50 and 41.70) results in a mean which is larger than most of the values in the data set.

Aquatic: Mean or median because the data set is relatively symmetric.

We can also make histograms to see how the data is distributed.

hist(x = x, main = "Terrestrial feeders", xlab = "DDE to PCB Ratio")

hist(x = y, main = "Aquatic feeders", xlab = "DDE to PCB Ratio")

3.31

a. Luxury: Mean = 145.0; Standard deviation = 27.6

Budget: Mean = 46.1; Standard deviation = 5.13

Put the data for luxury hotel into x and put the data for budget hotel into y, then use

mean(x)

sd(x)

mean(y)

sd(y)

The rule of thumb  gives an interval of (89.8, 200.2) for luxury hotels and an interval of (35.8, 56.4) for budget hotels.

data.frame(lower = mean(x) - 2\*sd(x), upper = mean(x) + 2\*sd(x))

c. Luxury hotels vary in quality and location; therefore, their prices can be quite different from each other. Budget hotels are more similar to each other.

d. The range can also be used to measure the variability for the two types of hotels.

3.35

a. CAN: *Q*1 ≈ 1*.*45*, Q*2 ≈ 1*.*65*, Q*3 ≈ 2*.*4

DRY: *Q*1 *≈* 0*.*55*, Q*2 *≈* 0*.*60*, Q*3 *≈* 0*.*7

b. Canned dogfood is more expensive (values much greater than those of dry dogfood), highly skewed to the right with a few large outliers. Dry dogfood is slightly left skewed with considerably less variability than canned dogfood.

3.40

The means and standard deviations are:

|  |  |  |
| --- | --- | --- |
| Supplier |  | s |
| 1 | 189.23 | 2.96 |
| 2 | 156.28 | 3.30 |
| 3 | 203.94 | 8.96 |

Put data into a data frame lenses with two columns: dev and supplier. Then you can use the following code to make the dot plot:

stripchart(x = lenses$dev ~ lenses$supplier, method =

 "jitter", vertical = TRUE, pch = 1, main = "Dot plot",

ylab = "Deviations from Target Power Value", xlab = "Supplier")

The 20% and 80% quantile of the deviations of all suppliers combined can be calculated to be 157.6 and 202.2 respectively.

quantile(x = lense$dev, probs = c(0.2,0.8), type = 5)

b. The three distributions are relatively symmetric but supplier 3 is considerably more variable and is overall higher than supplier 1 and 2’s values. Supplier 1 has on average larger values than supplier 2.

c. Supplier 3 has the largest mean but also the largest standard deviation. Suppliers 1 and 2 have similar variability but supplier 1 has a greater mean than supplier 2.

d. Supplier 2 because it has the smallest mean and has about the same degree of variability as supplier 1.

3.43

a. Mean = 57.5, Median = 34.0

b. Median since the data has a few very large values which results in the mean being larger than all but a few of the data values.

c. Range = 273, s = 70.2

d. One way to approximate the standard deviation is to use the range divided by 4. A question about this will not be on any projects or tests. Using the approximation, s ≈ range/4 = 273 *=* 4 = 68.3. The approximation is fairly accurate.

e.

(-12.7, 127.7); yields 82%

(-82.9, 197.0); yields 94%

(-153.1, 268.1); yields 97%

Read the data into object y, then use the following code to calculate 

save.interval <- data.frame(lower = mean(y) - 2\*sd(y), upper = mean(y) + 2\*sd(y))

and similarly for  and 

The percentage of demands between failures falling in each of the three intervals can be calculated by

sum(y < save.interval$upper & y > save.interval$lower)/length(y)

The first percentage does not match the Empirical Rule well (68%) and the other two percentages approximate fairly well (95% and 99.7%).

Using the rule of thumb, the percentages for the last two intervals are 75% and 89%. They are much lower than the actual percentages because the rule of thumb gives the lower bound of the percentages and is often too conservative.

f. The Empirical Rule applies to data sets with roughly a “mound-shaped” histogram. The distribution of this data set is highly skewed right.

Additional problem:

Read data into a data frame food that looks like:

> food

 Glucose Fructose Maltose Saccharose

1 2.1 4.5 0.0 1.3

2 4.4 2.7 0.0 6.4

3 0.6 0.2 0.3 2.3

4 0.7 0.7 0.0 0.0

5 1.3 0.9 0.0 0.0

6 1.3 2.0 0.0 0.0

Then use the following codes to make the parallel coordinates plot:

library(package = MASS)

parcoord(x = food, col = 1:6, main = "Parallel coordinate plot for food data")

legend(locator(1), legend = c("Apples", "Bananas", "Corn", "Cucumber", "Lettuce", "Tomatoes"), lty = "solid", col = 1:6, bty = "n")