R Project

Ishita Dutta, Justin Lu

3/11/2021

Problem 1 - Fitbit.csv

a) List the names of the columns

```
## [1] "Steps"  "Miles"  "Floors" "Sleep"  "Day"    "Month"
```

b) Find the number of rows in the data set.

```
## [1] 88
```

c) Use the function «summary» on the data set and display the results. Describe how this function treats categorical columns, and how it treats numeric columns.

```
##      Steps           Miles           Floors           Sleep          Day
##  Min.   :  114   Min.   :0.050   Min.   :  1.00   Min.   :0.000   F  :13
##  1st Qu.: 7722   1st Qu.:3.390   1st Qu.: 11.00   1st Qu.:7.383   M  :13
##  Median :10920   Median :4.930   Median : 16.00   Median :7.617   R  :12
##  Mean   :10749   Mean   :4.759   Mean   : 20.78   Mean   :7.407   Sat:13
##  3rd Qu.:13780   3rd Qu.:6.093   3rd Qu.: 27.00   3rd Qu.:8.104   Sun:13
##  Max.   :20122   Max.   :8.790   Max.   :140.00   Max.   :9.333   T  :12
##                                                                   W  :12
##    Month
##  Feb  :28
##  Jan  :31
##  March:29
##
##
##
##
```

The summary function will treat the numerical columns in a 5-number-summary manner. It returns the minimum, maximum, first quartile, median, and third quartile values, as well as the mean for the data set. In terms of the categorical numbers, the summary function will count them in terms of occurence and provide this data (i.e. F occured 13 times in the day column).

d) Find the mean of the column Steps.

```
## [1] 10749.34
```

e) Find and display the average and standard deviation for steps taken for every day of the week.

Values for each day are printed average, then standard deviation.

Monday

```
## [1] 14500.85
```

```
## [1] 5362.416
```

Tuesday

```
## [1] 10501.58
```

```
## [1] 2631.131
```

Wednesday

```
## [1] 11863.5
```

```
## [1] 3441.038
```

Thursday

```
## [1] 10843.67
```

```
## [1] 2105.69
```

Friday

```
## [1] 13068.62
```

```
## [1] 3365.953
```

Saturday

```
## [1] 8222.538
```

```
## [1] 3270.769
```

Sunday

```
## [1] 6318.538
```

```
## [1] 3424.365
```

f) Find and display the average hours and standard deviation of sleep for every day of the week.

Values for each day are printed average, then standard deviation.

Monday

```
## [1] 6.341026
```

```
## [1] 2.161389
```

Tuesday

```
## [1] 8.019444
```

```
## [1] 0.754409
```

Wednesday

```
## [1] 7.444444
```

```
## [1] 0.434749
```

Thursday

```
## [1] 7.4125
```

```
## [1] 1.441172
```

Friday

```
## [1] 7.591026
```

```
## [1] 0.9029431
```

Saturday

```
## [1] 7.85
```

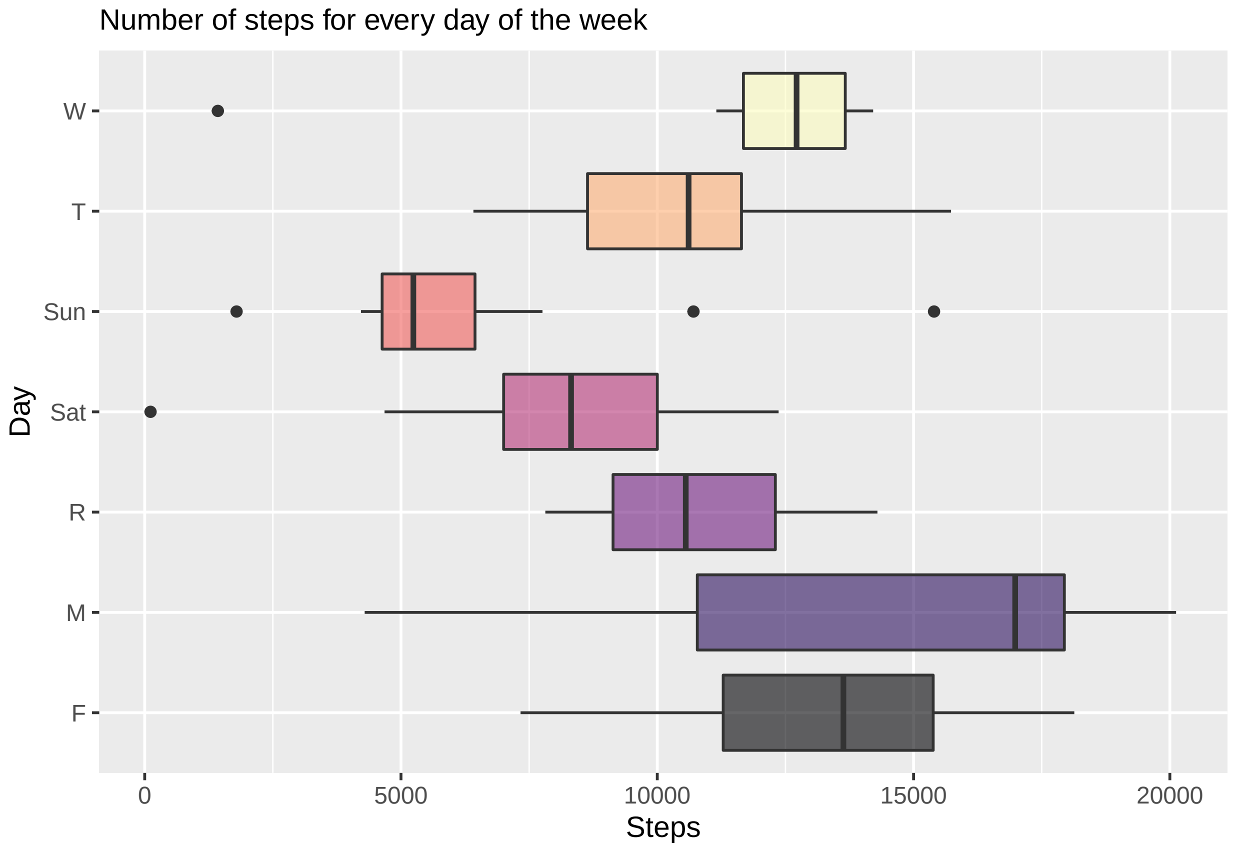
```
## [1] 0.7228096
```

Sunday

```
## [1] 7.238462
```

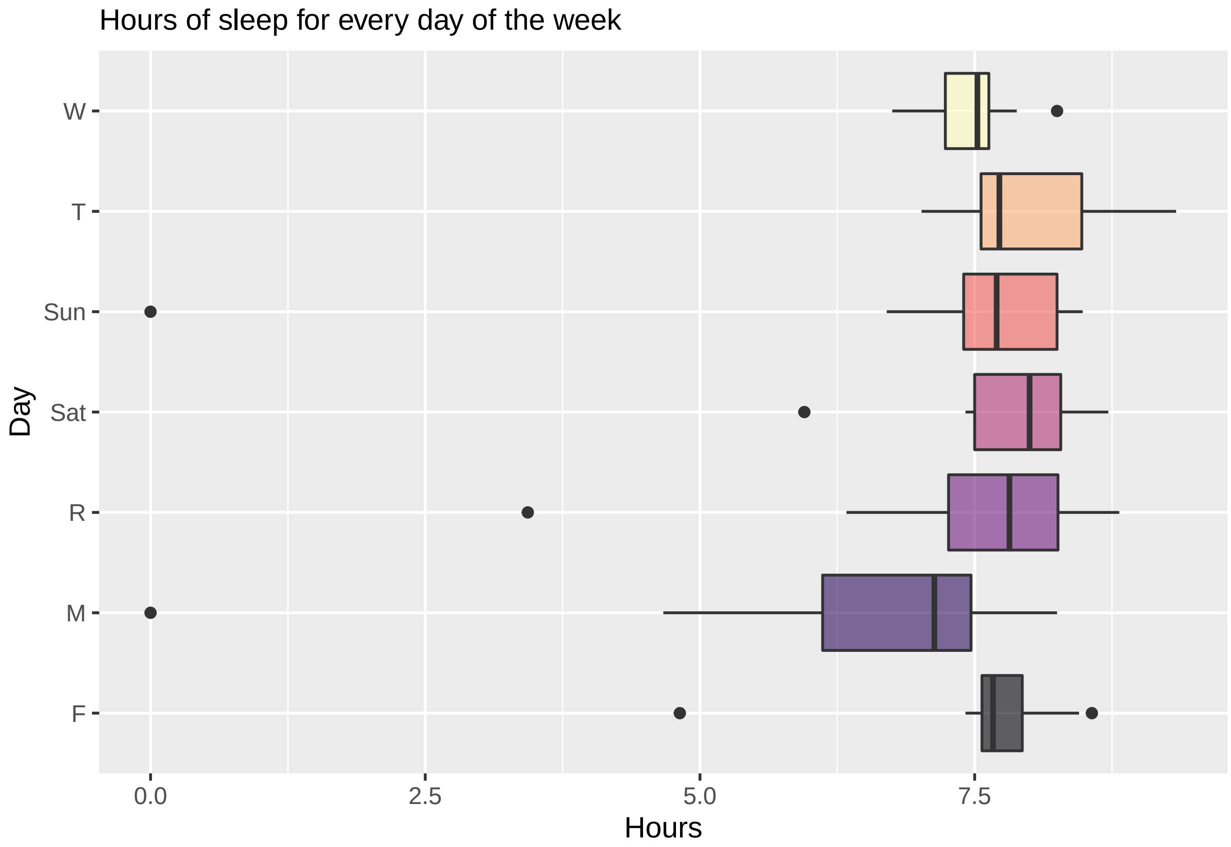
```
## [1] 2.239072
```

g) Create a boxplot of the total number of steps for every day of the week (there should be 7 sub-plots). Does it appear one day is less active than the rest? Explain.



It seems that Sunday is less active than the other days of the week, as the entire subplot for Sunday is much closer to 0 than the other subplots.

h) Create a boxplot of the total hours of sleep for every day of the week (there should be 7 sub-plots). Does it appear one day is less restful than the rest? Explain.



It seems that Monday is less restful than the other days of the week, as 50% of the data for Monday is under the 25% data line for almost all of the other days of the week (the exception being Wednesday)

i) Calculate the number of days where the total steps were above 10000.

```
## [1] 51
```

j) Calculate the average number of steps taken when total sleep was below 7 hours.

```
## [1] 12052.69
```

Problem 2 - Creating a Function

a) Takes in a vector and subtracts the mean and divides by the standard deviation. Then returns the standard deviation of the result. Test on the vector X = 1:100.

```
#2a)
vector_function <- function(vector) {
  mean_value = mean(vector)
  standard_dev = sd(vector)

  # initialize newVector
  n = length(vector)
  newVector = seq(1,n)
  # then loop vector and move values to new vector
  for(x in 1:n){
    value = (vector[x] - mean_value) / standard_dev
    newVector[x] = value
  }

  new_sd = sd(newVector)
  return(new_sd)
}
sampleVector = seq(1,100)
vector_function(sampleVector)
```

```
## [1] 1
```

b) Takes in a vector and finds values 2s below and above the mean, where s is standard deviation and returns both labeled respectively. Test on vector X = 1:100.

```
#2b)
vector_function2 <- function(vector) {
  mean_value = mean(vector)
  standard_dev = sd(vector)
  Lower = mean_value - (2*standard_dev)
  Upper = mean_value + (2*standard_dev)
  answer = list("upper" = Upper, "Lower" = Lower)
  return (answer)
}
sampleVector = seq(1,100)
vector_function2(sampleVector)
```

```
## $upper
## [1] 108.523
##
## $Lower
## [1] -7.522984
```

c) Create your own function. Provide a description and test.

My custom function takes in a vector and returns the greatest value in it.

```
#2c)
custom_function <- function(vector) {
  max = 0
  n = length(vector)
  for(x in 1:n){
    if (vector[x] > max){
      max = vector[x]
    }
  }
  return (max)
}
sampleVector = seq(1,100)
sampleVector2 = c(3,54,21,-22,321,0,100,911192,32,321,32,1,8)
custom_function(sampleVector)
```

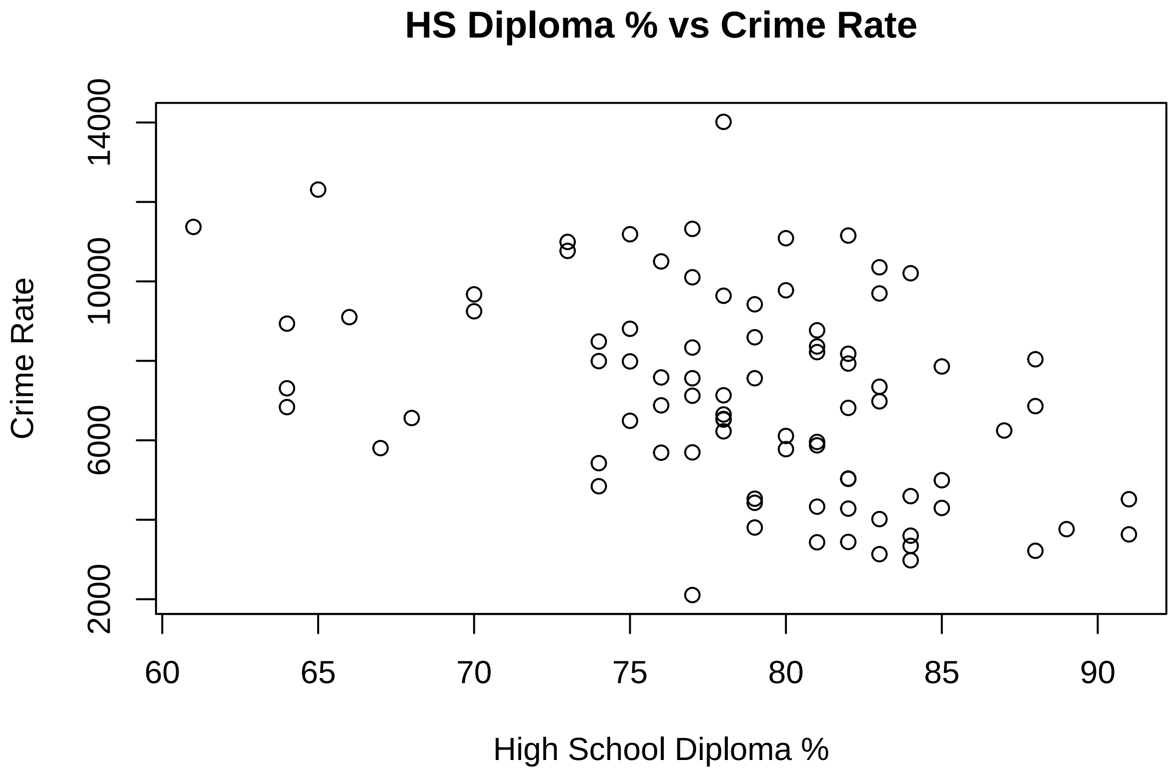
```
## [1] 100
```

```
custom_function(sampleVector2)
```

```
## [1] 911192
```

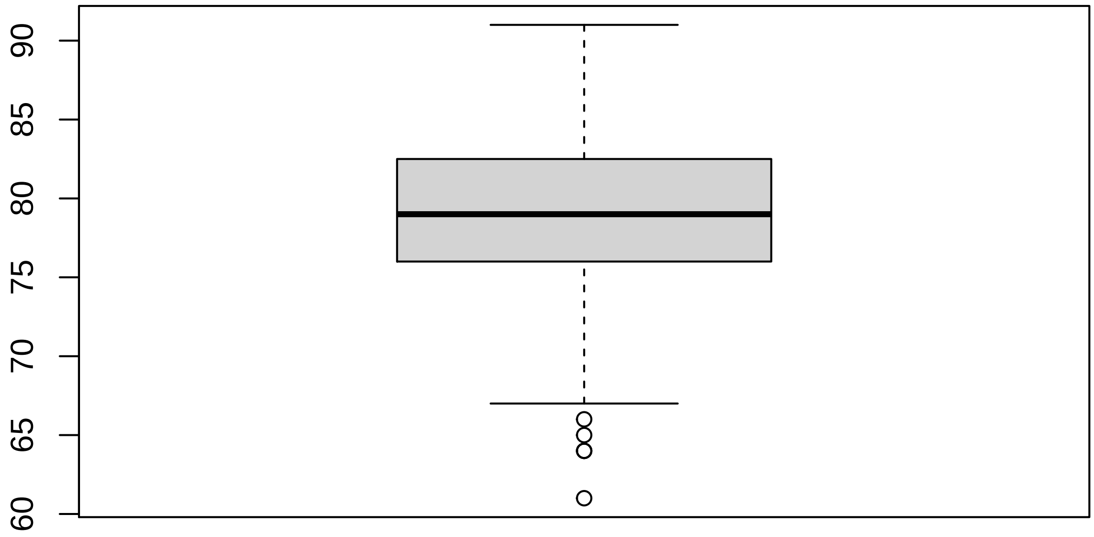
Problem 3 - Crime.csv

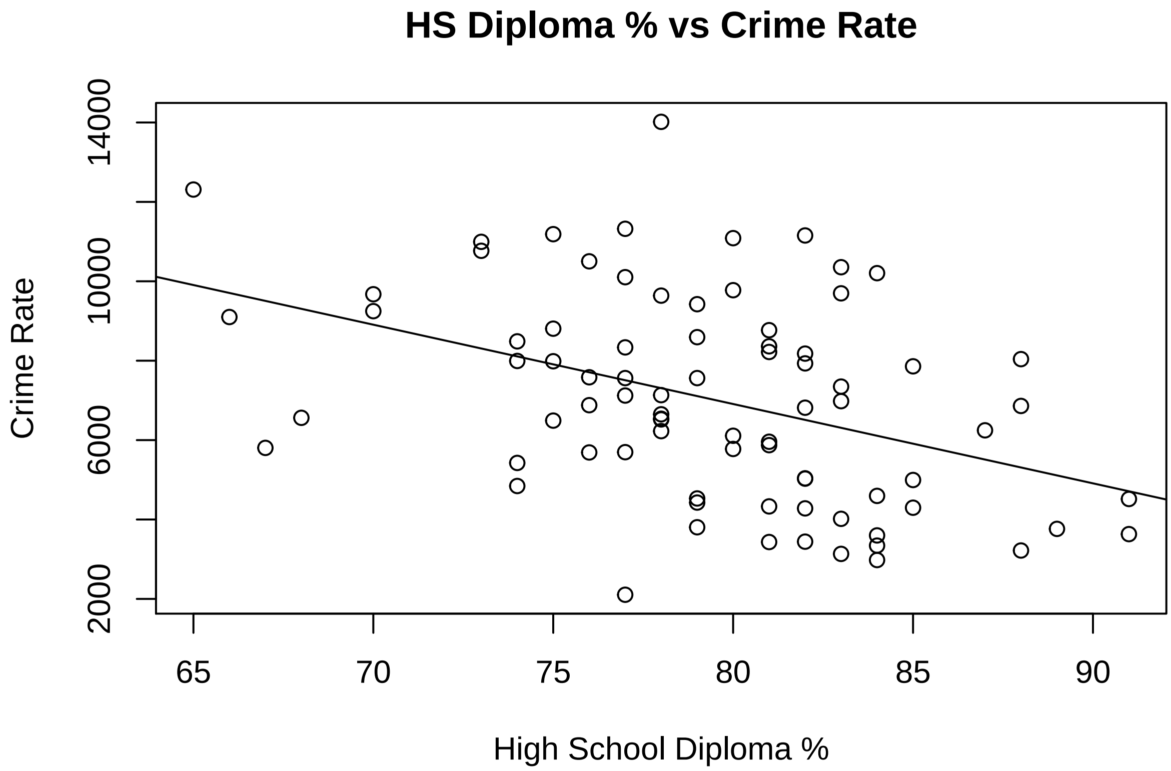
a) Plot a scatter plot of Y and X, being sure to label the axes and give a main title.



b) Does there appear to be outliers in the plot from (a)? If so, identify them in R (for example, list the pair (X,Y)that are outliers, or equivalently the row). Remove any outliers.

Yes, there does appear to be a outliers in our plot. Based off of the box plot we have 4 outliers, we can remove those from our data set and redo our scatterplot


90
85
80
75
70
65
60



c) Calculate the estimated regression line.

```
##
## Call:
## lm(formula = crimeRate ~ dip)
##
## Coefficients:
## (Intercept)          dip
##     22892.5       -199.8
```

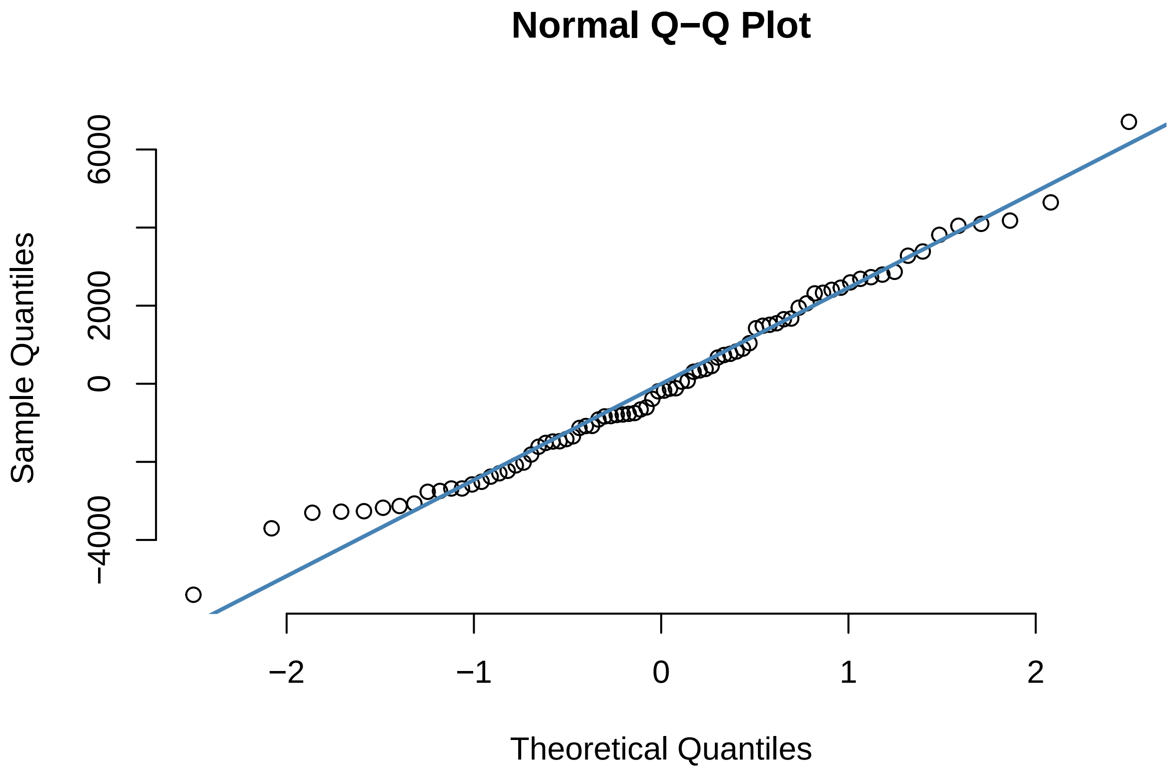
y = -171.9x + 20605.1

d) Interpret the slope and intercept (if appropriate) in terms of the problem.

Solution: The slope is = -171.9, while the intercept is = 20605.1.For our problem, let us interpret this: Starting with slope we can say that, when our X variable (Diploma %) increases by 1 percent, based on our regression line, the average crime rate decreases by 171.9.

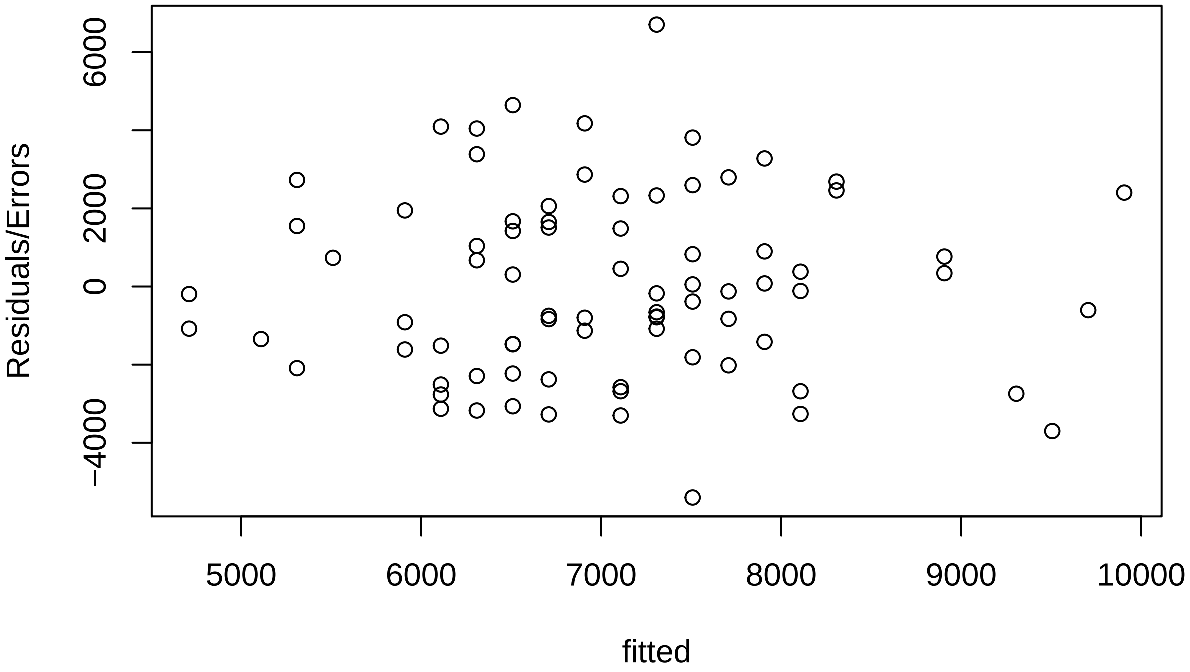
To interpret our intercept, 20605.1 is essentially that number to fit our line. Since our regression line is given as y = mx + b, our y intercept (b) is where the line would be when x is = to 0. In this case it logically does not have any relevance, and is just pertaining to our specific data set.

e) Create a QQ plot (normal probability plot) of the residuals. Does it appear that they are normally distributed?Explain.



Yes the graph does appear to be normally distributed. For the most part our graph follows our line and the deviation is minimal, meaning we have a normal distribution. We can safely deduce that our data set originates from a normal distribution.

f) Create a plot of the errors vs. the fitted values (^Yi's). Does it appear the variance of the errors is constant? Explain.



Yes, the variance of the errors does appear to be constant. Since our graph is scattered and we do not observe a relationship of any kind, we have a mean of 0 and the variance is constant.

g) Find the 95 percent confidence interval for the slope and interpret it in terms of the problem. Does the interval suggest there is a significant linear relationship? Explain.

```
##          2.5 %    97.5 %
## dip -300.0276 -99.56674
```

Yes we have a linear relationship as our model fits a normal distribution and our interval does not contain 0 we are working with a real linear slope value. We can interpret our confidence interval as we have a 95% confidence the true relationship between crime rate and one unit change of high school diploma % is between -300.0276 and -99.55674.

Problem 4 - Completing a Hypothesis test from Fitbit.csv

a) State your problem.

Problem: Is the number of steps on Sundays significantly different than the rest of the week?

b) State null and alternate hypotheses

mu-x = average number of steps on every other day except Sunday

mu-y = average number of steps on Sunday

null hypothesis mu-x - mu-y = 0

alternate hypothesis mu-x - mu-y != 0

c) Calculate the test statistic.

```
## [1] 4.927426
```

d) Calculate and interpret the p-value.

```
## [1] 8.332005e-07
```

The p-value of 8.33(eˆ(-7)) means that the probability of obtaining 0 as the difference between mu-x and mu-y is the 8.33(eˆ(-7)).

e) Make your decision. Use alpha = 0.05.

Using alpha = 0.05, we can reject the null hypothesis, as 8.33(eˆ(-7)) < 0.05.

f) State your conclusion in terms of the problem.

This concludes that the number of steps on Sunday is significantly different than the other days of the week. By looking at either the values or any graph, we know the number of steps is significantly less than the other days of the week on Sundays.

g) Construct and interpret a 90 percent confidence interval. Does this support your decision?

```
## [1] 5198.309
```

```
## [1] 5199.308
```

The confidence interval is nowhere near the 0 that we based our null hypothesis on, let alone be within the interval. The interval supports the decision.

h) What kind of error could you have made in your hypothesis test? Interpret this error.

We could have made a Type I error, in which we rejected an actually true null hypothesis. This would mean that we decided the number of steps on Sunday is significantly different than the rest of the week when in actuality there was no significant difference.

Appendix

```r
knitr::opts_chunk$set(
 echo = FALSE,
 error = FALSE,
 message = FALSE,
 warning = FALSE
)
library(ggplot2)
library(viridis)

#1) reading the file
fitbit <- read.csv("Fitbit.csv", stringsAsFactors = TRUE)

#1a)
```

```
colnames(fitbit)

#1b)
nrow(fitbit)

#1c)
summary(fitbit)

#1d)
mean(fitbit[,1])

#1e)
monday_values<- subset(fitbit, Day == "M")
mean(monday_values[,1])
sd(monday_values[,1])

tuesday_values<- subset(fitbit, Day == "T")
mean(tuesday_values[,1])
sd(tuesday_values[,1])

wednesday_values<- subset(fitbit, Day == "W")
mean(wednesday_values[,1])
sd(wednesday_values[,1])

thursday_values<- subset(fitbit, Day == "R")
mean(thursday_values[,1])
sd(thursday_values[,1])

friday_values<- subset(fitbit, Day == "F")
mean(friday_values[,1])
sd(friday_values[,1])

saturday_values<- subset(fitbit, Day == "Sat")
mean(saturday_values[,1])
sd(saturday_values[,1])

sunday_values<- subset(fitbit, Day == "Sun")
mean(sunday_values[,1])
sd(sunday_values[,1])

#1f)
mean(monday_values[,4])
sd(monday_values[,4])

mean(tuesday_values[,4])
sd(tuesday_values[,4])

mean(wednesday_values[,4])
sd(wednesday_values[,4])

mean(thursday_values[,4])
sd(thursday_values[,4])
```

```
mean(friday_values[,4])
sd(friday_values[,4])

mean(saturday_values[,4])
sd(saturday_values[,4])

mean(sunday_values[,4])
sd(sunday_values[,4])

#1g)
ggplot(fitbit,aes(x =Steps, y = Day, fill=Day))+
  geom_boxplot()+
  scale_fill_viridis(discrete = TRUE, alpha=0.6, option="A")+
  theme(legend.position="none",plot.title =element_text(size=11))+
  ggtitle("Number of steps for every day of the week")+
  xlab("Steps")

#1h)
ggplot(fitbit,aes(x =Sleep, y = Day, fill=Day))+
  geom_boxplot()+
  scale_fill_viridis(discrete = TRUE, alpha=0.6, option="A")+
  theme(legend.position="none",plot.title =element_text(size=11))+
  ggtitle("Hours of sleep for every day of the week")+
  xlab("Hours")

#1i)
steps_above <- subset(fitbit, Steps > 10000)
nrow(steps_above)

#1j)
sleep_under<- subset(fitbit, Sleep < 7)
mean(sleep_under[,1])


#2a)
vector_function <- function(vector) {
  mean_value = mean(vector)
  standard_dev = sd(vector)

  # initialize newVector
  n = length(vector)
  newVector = seq(1,n)
  # then loop vector and move values to new vector
  for(x in 1:n){
    value = (vector[x] - mean_value) / standard_dev
    newVector[x] = value
  }

  new_sd = sd(newVector)
  return(new_sd)
}
sampleVector = seq(1,100)
vector_function(sampleVector)
```

```r
#2b)
vector_function2 <- function(vector) {
  mean_value = mean(vector)
  standard_dev = sd(vector)
  Lower = mean_value - (2*standard_dev)
  Upper = mean_value + (2*standard_dev)
  answer = list("upper" = Upper, "Lower" = Lower)
  return (answer)
}
sampleVector = seq(1,100)
vector_function2(sampleVector)


#2c)
custom_function <- function(vector) {
  max = 0
  n = length(vector)
  for(x in 1:n){
    if (vector[x] > max){
      max = vector[x]
    }
  }
  return (max)
}
sampleVector = seq(1,100)
sampleVector2 = c(3,54,21,-22,321,0,100,911192,32,321,32,1,8)
custom_function(sampleVector)
custom_function(sampleVector2)

#3) reading the file
crimes = read.csv("crime.csv")

#3a)
dip = crimes$dip
crimeRate = crimes$rate
plot(x = dip, y = crimeRate,
     xlab="High School Diploma %",
     ylab="Crime Rate" ,
     main = "HS Diploma % vs Crime Rate")

#3b)
boxplot(dip)
crimes <- crimes[-c(71, 76, 51, 39),]
dip = crimes$dip
crimeRate = crimes$rate


plot(x = dip, y = crimeRate, xlab="High School Diploma %", ylab="Crime Rate" , main = "HS Diploma % vs
abline(lm(crimeRate ~ dip))
#3c)
lm(crimeRate ~ dip)
```

```
#3e)
residuals = resid(lm(crimeRate ~ dip))

qqnorm(residuals, pch = 1, frame = FALSE)
qqline(residuals, col = "steelblue", lwd = 2)

#3f)
fittedValues = fitted(lm(crimeRate ~ dip))
plot(fittedValues, residuals, ylab="Residuals/Errors", xlab="fitted" )

#3g)
reg_line = lm(crimeRate ~ dip)
confint(reg_line,’dip’,level=0.95)

#4c)
every_except_sunday <- subset(fitbit, Day != "Sun")
mux = mean(every_except_sunday[,1])
muy = mean(sunday_values[,1])
sx = sd(every_except_sunday[,1])
sy = sd(sunday_values[,1])
numerator = mux - muy
rowx = nrow(every_except_sunday)
rowy = nrow(sunday_values)
varx = sx^2
vary = sy^2
inside_square = (varx/rowx) + (vary/rowy)
ts =numerator/sqrt(inside_square)
print(ts)

#4d)
prob = 1 - pnorm(ts)
pval = 2* prob
pval

#4g)
degfreedom = rowx + rowy - 2
tvalu = abs(qt(0.95, 88, lower.tail = TRUE))
tvall = abs(qt(0.05, 88, lower.tail = TRUE))
lower = numerator - (tvall * (sqrt((1/rowx)+(1/rowy))))
lower
upper = numerator + (tvalu * (sqrt((1/rowx)+(1/rowy))))
upper
```