PS1

Homework 1

The total points on this homework is 125. Out of these 6 points are reserved for clarity of presentation, punctuation and commenting with respect to the code.

1. This is an exercise with the partial objective of trying to get ideas from the demo(graphics) package in R to solve our problem.

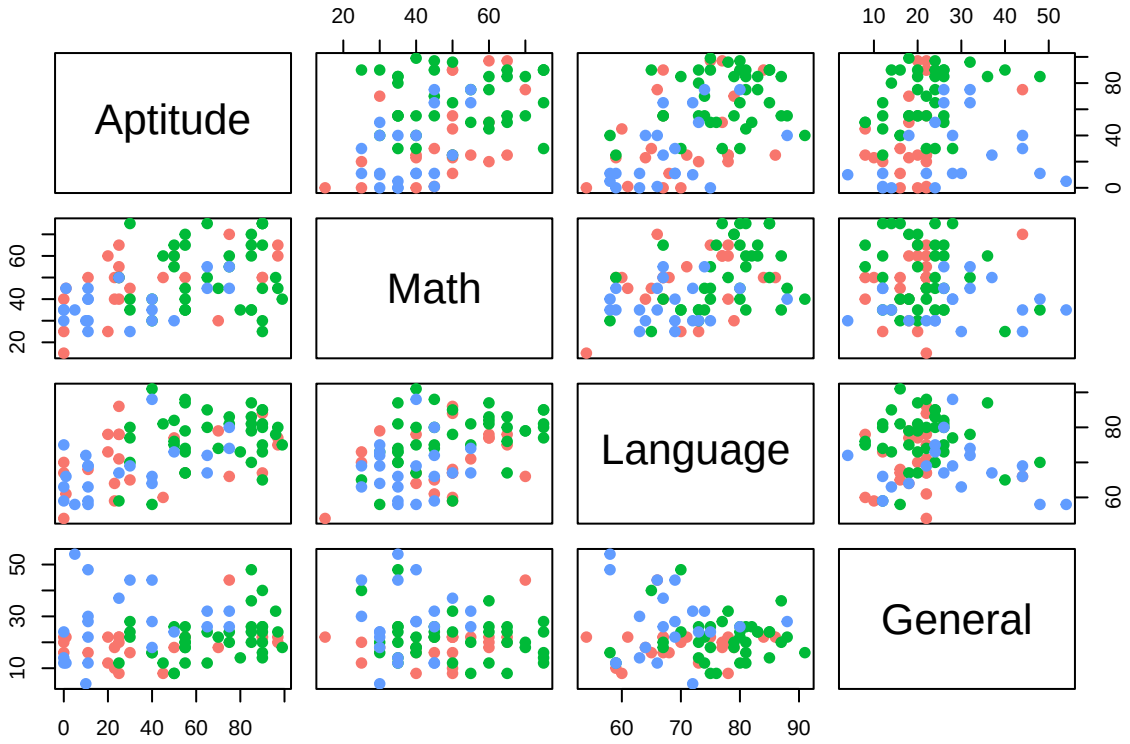
The dataset student-apt.dat has data on student scores in aptitude, mathematics, language and general knowledge for students in technical disciplines (group 1), architecture (group 2) and medical technology students (group 3) as indicated in column 1.

(a) Read in the dataset. Note that column names are not provided in the file. So, please supply them either through the appropriate argument in the read.table() function, or by invoking the names() function after reading and storing in as a dataframe. Please read the help file on read.table() by using the function call ?read.table for more information. [5 points]

```
studentApt <- read.table(file = "C:/Users/samue/OneDrive/Desktop/Iowa_State_PS/STAT 5790/PS/PS1/student
                         header = F,
                         col.names = c("studentGroup", "Aptitude", "Math", "Language", "General")
                         )
```

(b) Display the observations in a set of pairwise scatter plots of the scores in aptitude, mathematics, language and general knowledge, with color to indicate the groups of the observations. [6 points]

```
colorsPair <- c("#F8766D", "#00BA38", "#619CFF")
pairs(x = studentApt[2:5], col = colorsPair[studentApt$studentGroup], pch = 19)
```



(c) Comment on characteristics of the students in the three groups. [2 points]

Generally, the above pairwise scatter plots don't provide anything immediately apparent, at least with a greater degree of rigor, e.g. we don't have correlation coefficients to directly compare across groups. That is to say: The commentary given is predominantly descriptive.

Student group "3" tends to exhibit the highest scoring individuals in Aptitude, Math, and Language. Interestingly this doesn't appear to carry over to the General Score, i.e. despite students in student group "3" scoring higher in Aptitude, Math, or Language individually, this doesn't appear to translate directly to higher General scores. Intuitively however, this makes some sense though, as we'd expect General score to be a composite of all scores, and we only have information about pairs of variables, not the combination of 3 or more.

2. The National Institute of Standards and Technology has a web page that lists the first 5,000 digits of the irrational number π. You can read these digits into R from the website http://www.itl.nist.gov/div898/strd/univ/data/PiDigits.dat.

(a) Read in the dataset. Note that the file on the website has the first 60 lines which are really different statistics on the data. These 60 lines should be skipped. Look at the help function on the read.table to see how to skip these lines. [7 points]

```
piData <- read.table(file ="http://www.itl.nist.gov/div898/strd/univ/data/PiDigits.dat",
                     skip = 60)
```

(b) Construct a frequency table of the digits 1 through 9. (Hint: search on terms to get an appropriate function.) [5 points]

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

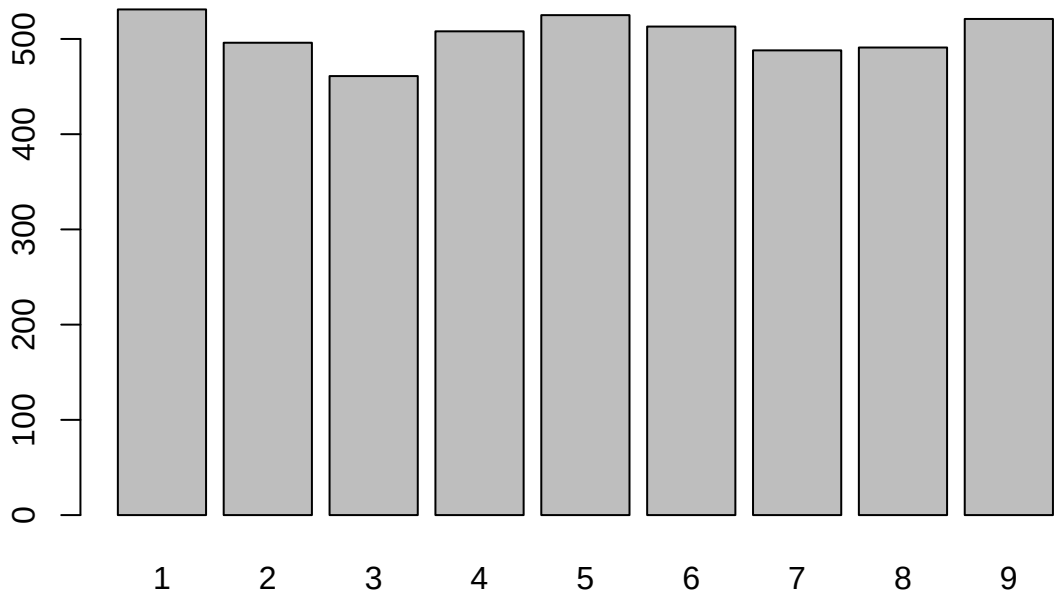
```
analyze <- piData %>%
  filter(V1 > 0)

piTable <- table(analyze)
piTable
```

```
## V1
##   1   2   3   4   5   6   7   8   9
## 531 496 461 508 525 513 488 491 521
```

(c) Construct a bar plot of the frequencies found in part (b). [5 points]

```
barplot(piTable)
```



(d) Use the chi-square test to test the hypothesis that the digits 1 through 9 are equally probable in the digits of π. What conclusions can you draw? (Hint: use the function chisq.test().) [8 points]

```
chiData <- chisq.test(piTable)
expectedDigits <- chiData$expected

differenceTable <- rep(NA, 9)
for (i in 1:9) {
  differenceTable[i] <- piTable[[i]] - chiData$expected[[i]]
}

differenceData <- data.frame("digit" = 1:9, "difference" = differenceTable)

piTable
```

```
## V1
##   1   2   3   4   5   6   7   8   9
## 531 496 461 508 525 513 488 491 521
```

```
expectedDigits
```

```
##        1        2        3        4        5        6        7        8
## 503.7778 503.7778 503.7778 503.7778 503.7778 503.7778 503.7778 503.7778
##        9
## 503.7778
```

```
differenceData
```

```
##   digit difference
## 1     1  27.222222
## 2     2  -7.777778
## 3     3 -42.777778
## 4     4   4.222222
## 5     5  21.222222
## 6     6   9.222222
## 7     7 -15.777778
## 8     8 -12.777778
## 9     9  17.222222
```

```
library(dplyr)
meanCount <- mean(piTable)
meanSd <- sd(piTable)
coefVariation <- meanCount / meanSd

meanCount
```

```
## [1] 503.7778
```

```
meanSd
```

```
## [1] 22.06115
```

```
coefVariation
```

```
## [1] 22.83552
```

```
differenceData %>%
  mutate("deviationAmt" = difference / coefVariation)
```

```
##   digit difference deviationAmt
## 1     1  27.222222    1.1921002
## 2     2  -7.777778   -0.3406001
## 3     3 -42.777778   -1.8733003
## 4     4   4.222222    0.1848972
## 5     5  21.222222    0.9293516
## 6     6   9.222222    0.4038544
## 7     7 -15.777778   -0.6909315
## 8     8 -12.777778   -0.5595572
## 9     9  17.222222    0.7541858
```

```
differenceData$deviationAmt
```

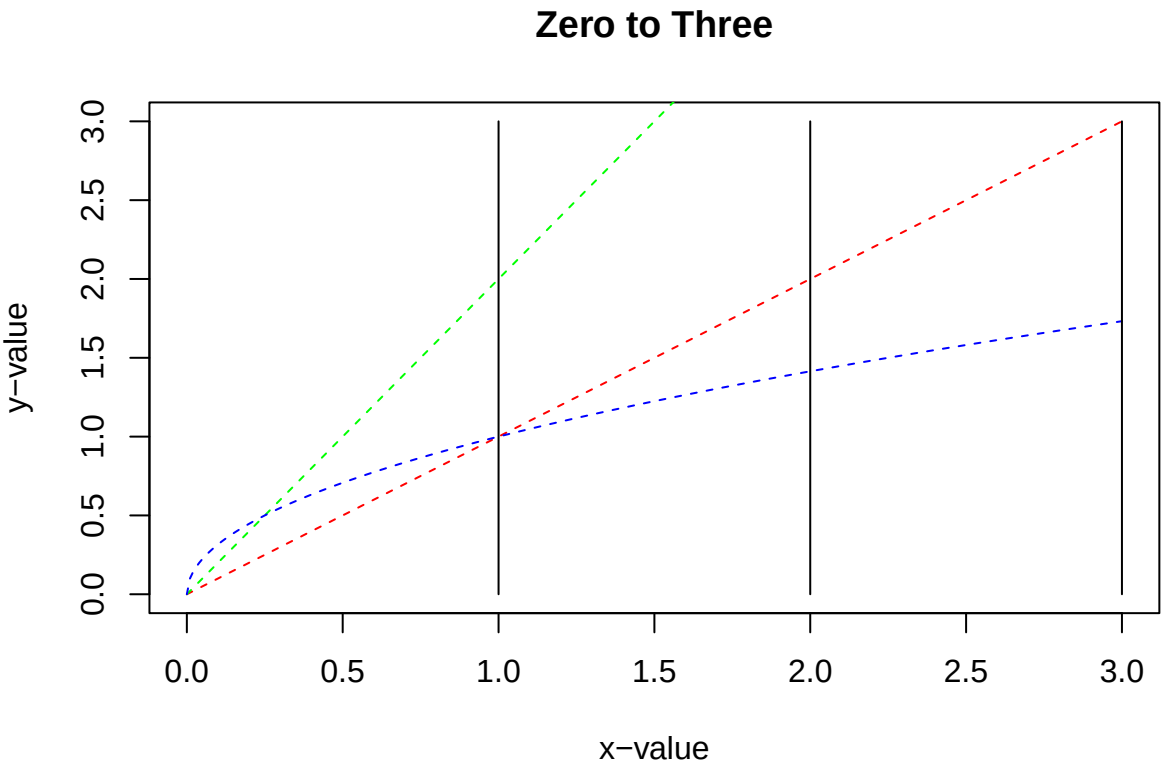
```
## NULL
```

We tend to see a similar distribution of counts for a given digit compared to its expected count, the latter of which is defined using a chi-squared test. Calculating the coefficient of variation for this data gives us a sense of how many standard deviations "off" an observed count is from its expected count. We observe that 7 of the 9 digits are within 1 coefficient of variation (looking at how many deviations away the difference is between the observed and expected counts of a given digit), with the remaining two digits within 1.96 standard deviations of the expected counts. Overall, I would say that all of our data falls within the bounds of expectation.

3. Plot a graph that shows three curves y = x, y = x2, and y = $\sqrt{x}$, for x from 0 to 3. Plot a vertical line at 1, 2 and 3, (curves and lines on the same plot). Hint: Decompose this problem into multiple parts: In the first instance, create a vector x consisting of 0 to 3, in increments of 0.01 (say). Create another vector y1 = x2 and a third vector y2 = $\sqrt{x}$ and then combine them all to form a dataframe. Use the plot function as well as the lines function to add lines to an existing plot. Turn in the final plot and also the R code you used for the problem. [15 points]

```
x <- seq(from = 0, to = 3, by = 0.01)
y <- x
y1 <- x*2
y2 <- sqrt(x)
dataX <- data.frame(x, y, y1, y2)

plot(x = c(0,3),
     y = c(0,3),
     pch = 1,
     xlim = c(0,3),
     ylim = c(0,3),
     xlab = "x-value",
     ylab = "y-value",
     main = "Zero to Three",
     type = "n"
     )
lines(x = dataX$x,
      y = dataX$y,
      col = "red",
      lty = "dashed")
lines(x = dataX$x,
      y = dataX$y1,
      col = "green",
      lty = "dashed")
lines(x = dataX$x,
      y = dataX$y2,
      col = "blue",
      lty = "dashed")
lines(x = rep(x = 1, times = length(x)),
      y = x,
      col = "black",
      lty = "solid")
lines(x = rep(x = 2, times = length(x)),
      y = x,
      col = "black",
      lty = "solid")
lines(x = rep(x = 3, times = length(x)),
      y = x,
      col = "black",
      lty = "solid")
```


Zero to Three
y−value
0.0
0.5
1.0
1.5
2.0
2.5
3.0
x−value

4. Consider the dataset available in R called cars with the help file that also has more information.

(a - c): (a) Read in the dataset from the file. Call it cars (say). [2 points] (b) Attach the dataframe so that the variables in the dataframe are now globally available. [1 point] (c) The speeds are provided in miles per hour. Convert the speeds into feet per second and store the result in an appropriate vector. [5 points]

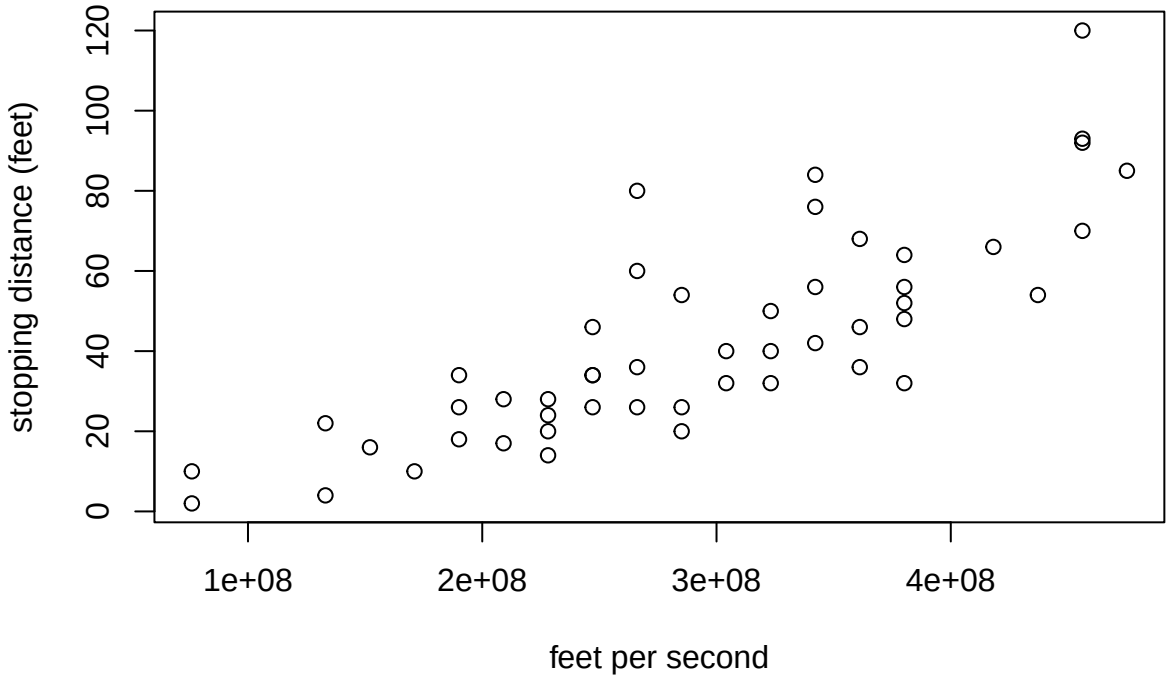
speed: Represents the speed of the car in miles per hour dist: Represents the stopping distance of the car in feet

miles per hour * (feet/miles) * (seconds/hour) = feet per second feet per second = speed * (5280/1) * (3600/1)

```
data(cars)
attach(cars)
feetSpeed <- cars$speed * 5280 * 3600
# feetDistance <- cars$dist * 5280
```

(d) Plot the speed (in feet per second) against the distance (in feet). [4 points]

```
plot(x = feetSpeed, y = cars$dist, xlab = "feet per second", ylab = "stopping distance (feet)")
```



(e) Convert the measurements into the metric system. Note that one mile is equal to 1.6093 kilome-tres. Store the results in appropriate vectors. [5 points]

speed: Represents the speed of the car in miles per hour. dist: Represents the stopping distance of the car in feet

m per hour = miles per hour * (km/mile) * (m / km) m stopping distance = feet stopping distance (1 km / ft) * (m / km)

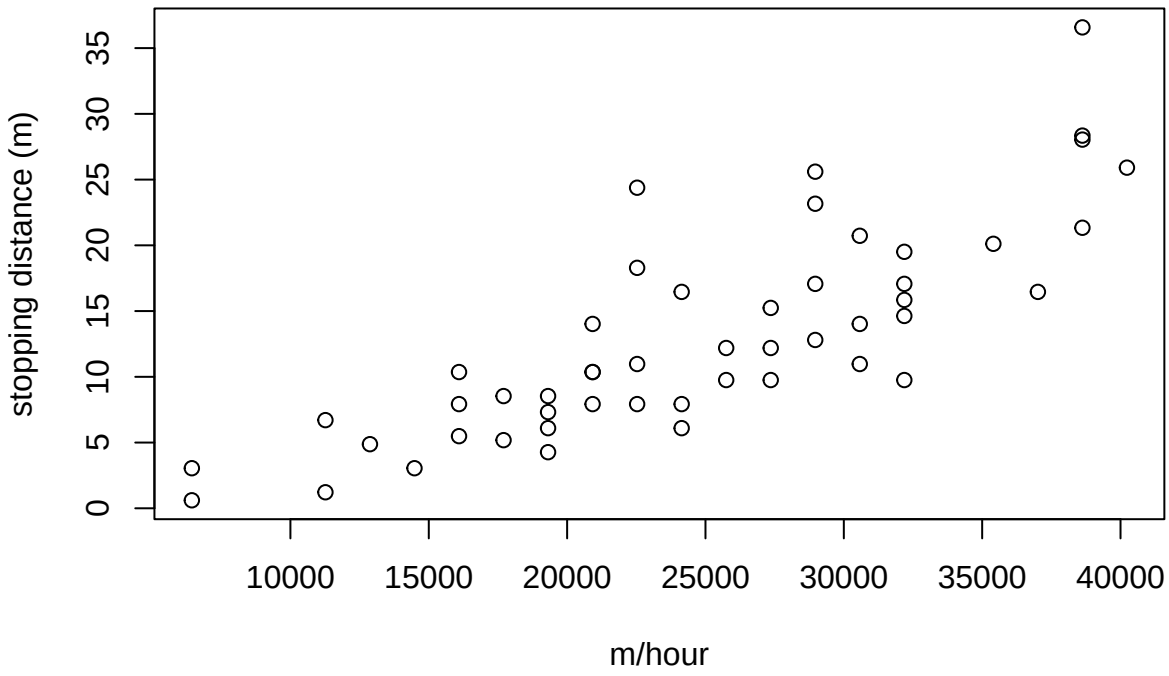
```
metricSpeed <- cars$speed*1.6093*1000
metricDistance <- cars$dist/3280.84*1000
```

(f) Detach the dataframe. [1 point]

```
detach(cars)
```

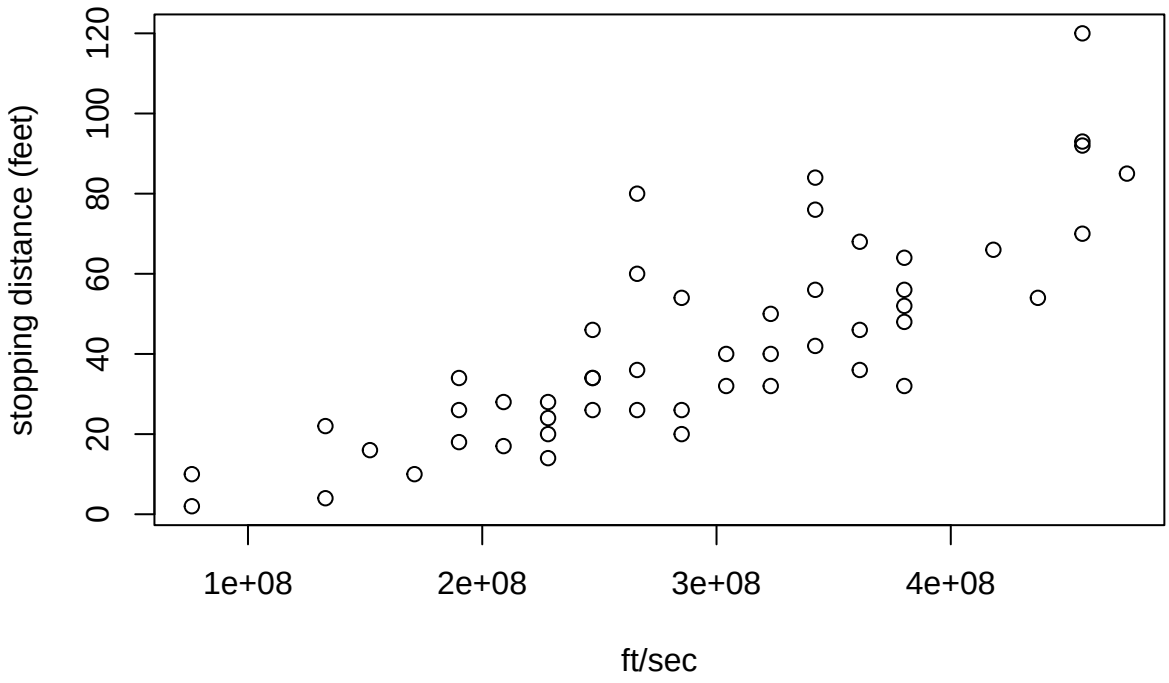
(g) Plot the speed (in metres per second) against the distance (in metres). [4 points]

```
plot(x = metricSpeed, y = metricDistance, xlab = "m/hour", ylab = "stopping distance (m)")
```

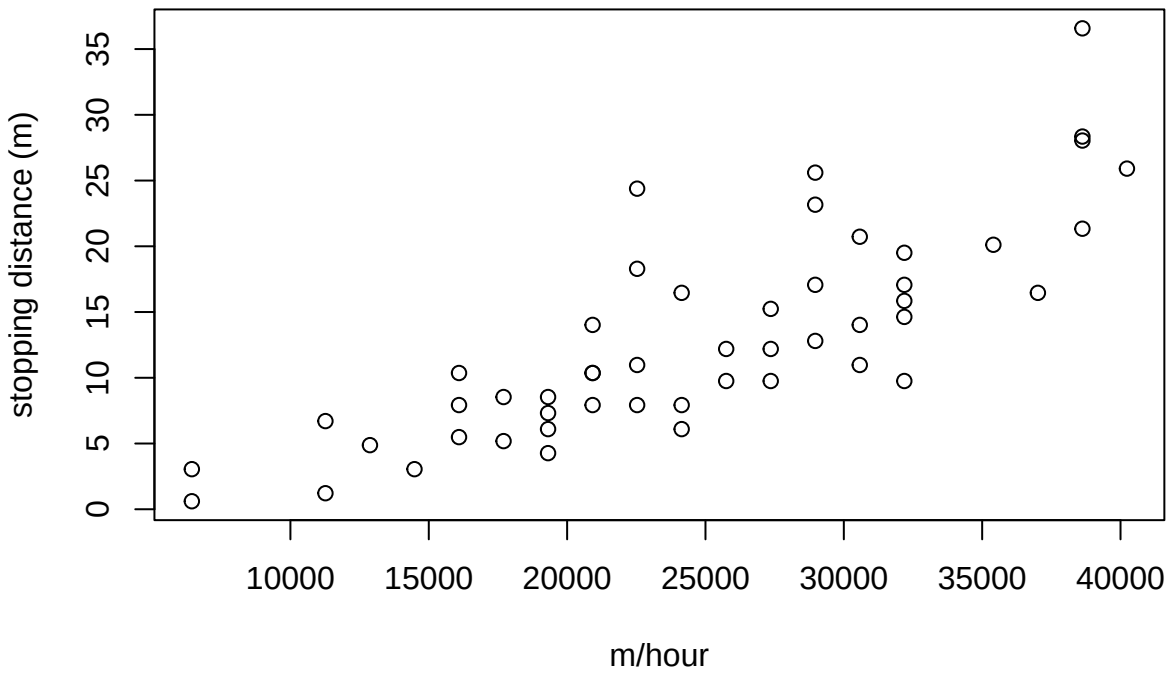


(h) Make sure that the plots above are labeled and titled appropriately. Print out the plots using dev.print() or otherwise. What can you tell, if anything, looking at the two plots? [2 points]

```
plot(x = feetSpeed, y = cars$dist, xlab = "ft/sec", ylab = "stopping distance (feet)")
```




```
plot(x = metricSpeed, y = metricDistance, xlab = "m/hour", ylab = "stopping distance (m)")
```



We see that converting units concurrently, i.e. converting the x and y axes at the same time and by the same unit conversion, does not change the interpretation of the plot and overall trend(s) we observe.

5. Consider the dataset pressure which is in the R software base installation. You may type help(pressure) to get more information on this dataset.

(a) The temperatures are provided on the Celsius scale. Convert them to the Fahrenheit scale and store them in an appropriate vector. [3 points]

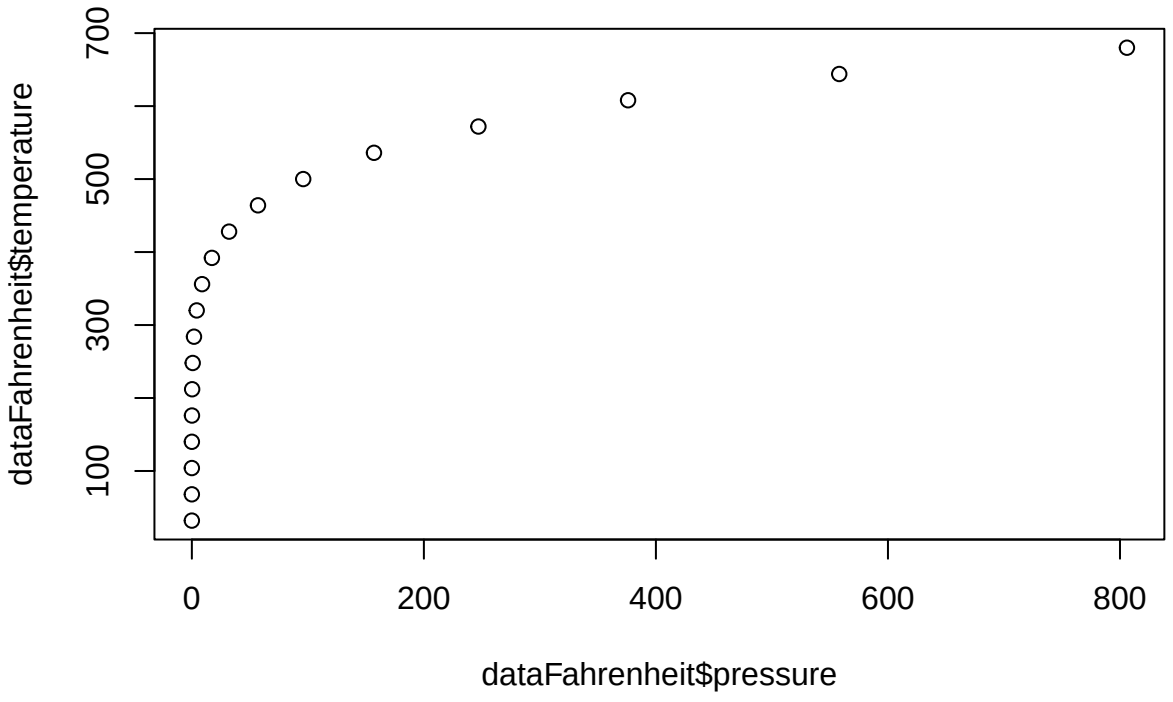
```
Fahrenheit <- pressure$temperature*9/5 + 32
```

(b) Create a dataframe consisting of the temperature in the Fahrenheit scale and Pressure. [4 points]

```
dataFahrenheit <- data.frame("pressure" = pressure$pressure, "temperature" = Fahrenheit)
```

(c) Plot temperature against pressure in the Fahrenheit scale. [3 points]

```
plot(y = dataFahrenheit$temperature, x = dataFahrenheit$pressure)
```

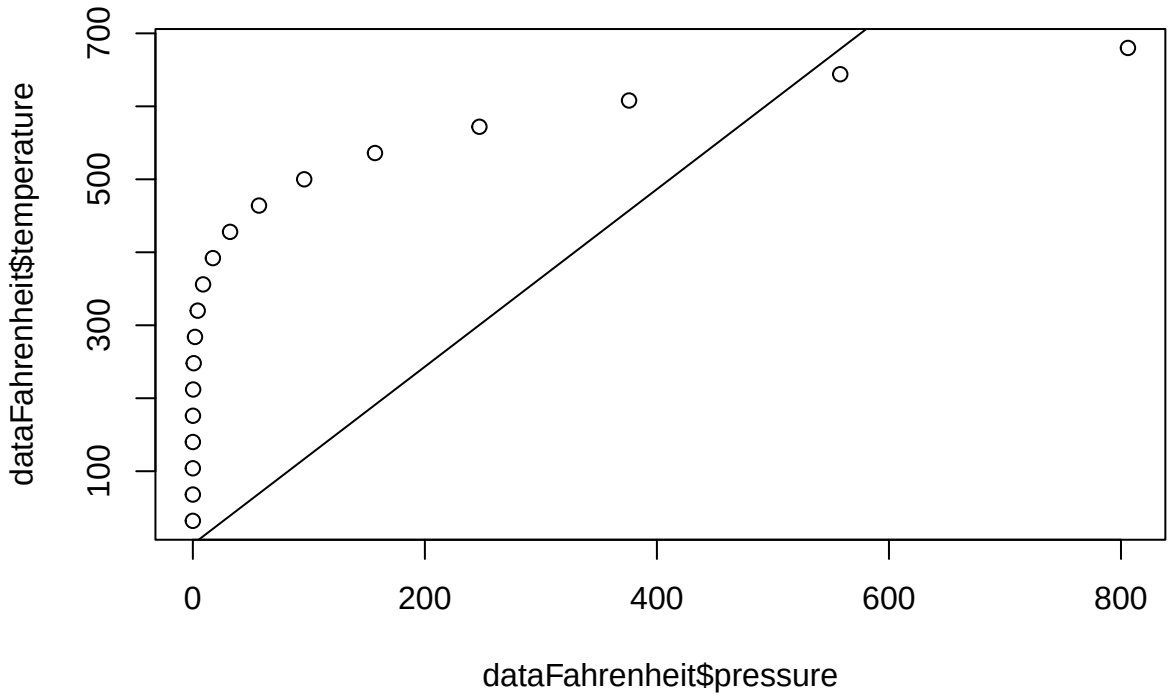


(d) Perform a simple linear regression with temperature (in Fahrenheit) against pressure, but with no intercept in the model. Report a summary of the results and plot the fitted line on the plot in (c). Comment. [4 + 2 points]

```r
regression <- lm(formula = Fahrenheit ~ 0 + pressure, data = dataFahrenheit)
summary(regression)
```

```
##
## Call:
## lm(formula = Fahrenheit ~ 0 + pressure, data = dataFahrenheit)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -300.2  122.0  247.1  345.2  394.7
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## pressure   1.2161     0.2516   4.834 0.000133 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 275.8 on 18 degrees of freedom
## Multiple R-squared:  0.5649, Adjusted R-squared:  0.5408
## F-statistic: 23.37 on 1 and 18 DF,  p-value: 0.000133
```

```r
plot(y = dataFahrenheit$temperature, x = dataFahrenheit$pressure)
abline(a = 0, b = 1.2161)
```

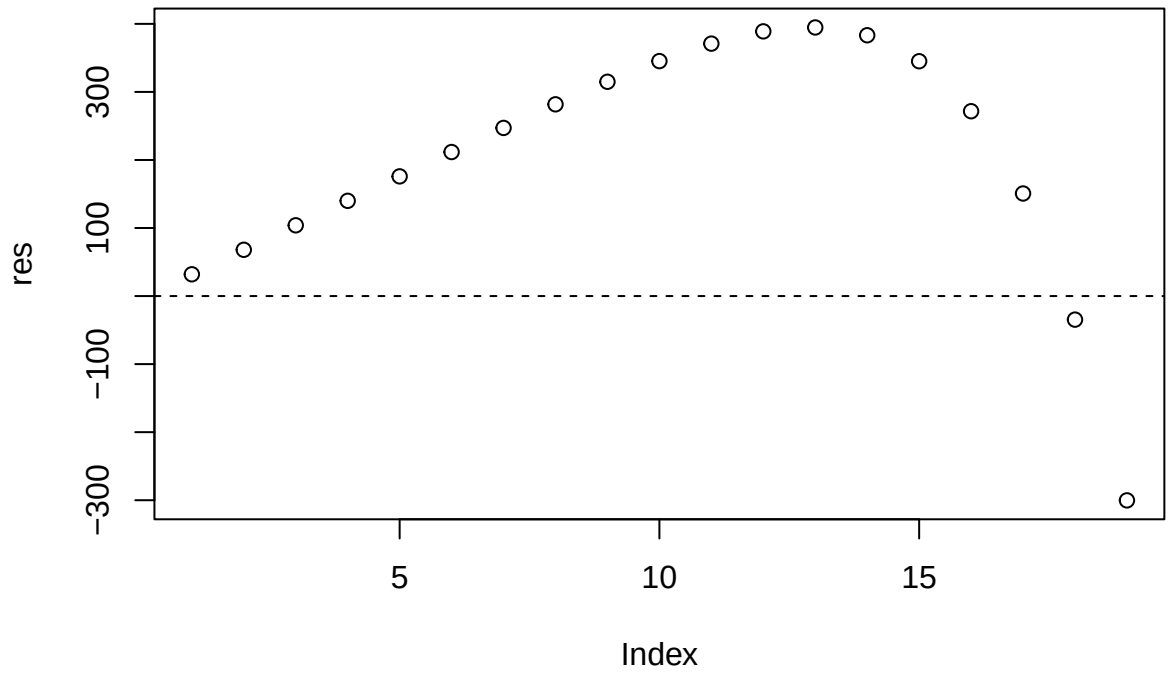


This is a very poor fit! We should suspect that a simple linear fit for the model will not suffice to accurately

approximate the relationship observed. The actual trend appears non-linear, specifically a non-linear relationship between pressure and temperature with diminishing returns, i.e. the slope of the fit line decreases with greater pressure.

(e) Plot the residuals against the fitted values. Comment. [3 + 2 points]

```
res <- residuals(regression)
plot(res)
abline(h = 0, lty = "dashed")
```



We see a clear pattern in the residual plot, at first increasing positively then sharply falling off and veering into negative values when approaching high pressure values.

(f) Clearly pressure is not adequate to explain the relationship with temperature. Create another dataframe with four columns, given by temperature in Fahrenheit, pressure, the square of pressure and cubed pressure. [5 points]

```
dataNL <- data.frame("temperature" = dataFahrenheit$temperature, "pressure" = dataFahrenheit$pressure,
```

(g) Use the above to perform multiple linear regression (with intercept) of temperature on the rest. What coefficients are significant? [6 points]

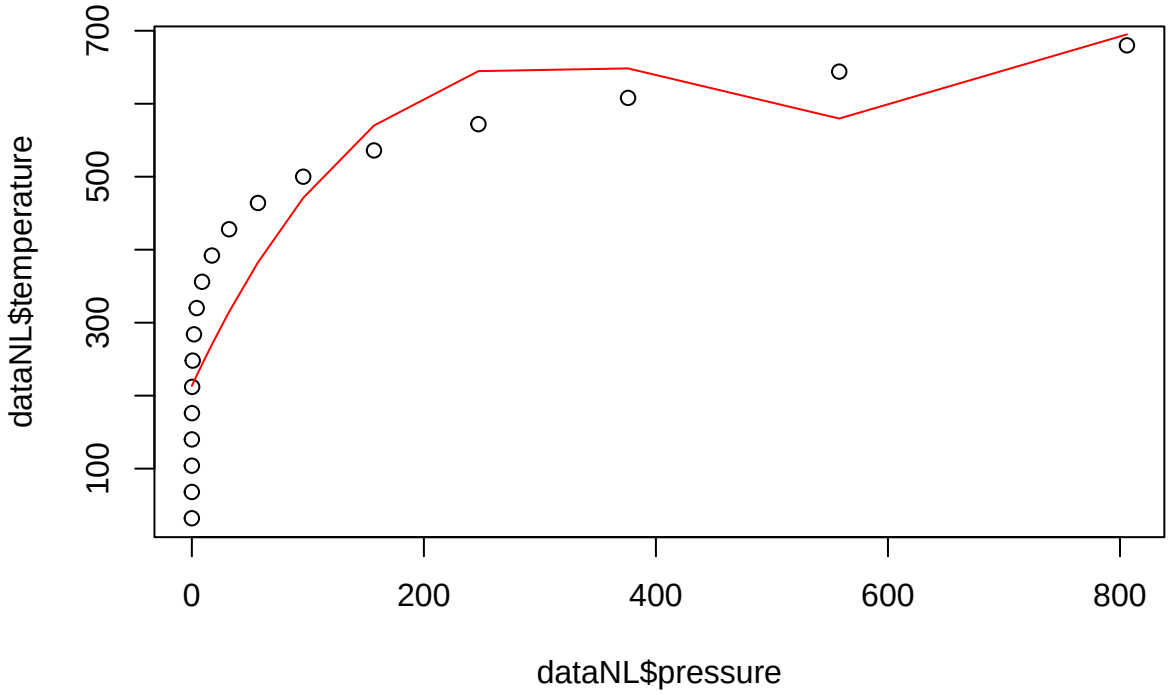
```r
regression2 <- lm(formula = temperature ~ pressure + square + cube, data = dataNL)
summary(regression2)
```

```
##
## Call:
## lm(formula = temperature ~ pressure + square + cube, data = dataNL)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -181.39  -56.54   -2.31   72.88  121.93
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.134e+02  2.966e+01   7.195  3.1e-06 ***
## pressure     3.417e+00  7.671e-01   4.454 0.000464 ***
## square      -8.207e-03  2.826e-03  -2.904 0.010898 *
## cube         5.842e-06  2.473e-06   2.362 0.032094 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 98.98 on 15 degrees of freedom
## Multiple R-squared:  0.8011, Adjusted R-squared:  0.7613
## F-statistic: 20.14 on 3 and 15 DF,  p-value: 1.618e-05
```

All coefficients are significant to some degree, though the initial linear coefficient appears most significant (excluding interpretation of the intercept term). Each successive term becomes less significant than its prior iteration, i.e. pressure is more significant than its square, and its squared term is more significant than the cubed term.

(h) On the plot of (c) above, put in the fitted line. Comment on the previous fit and this one. [3 + 2 points]

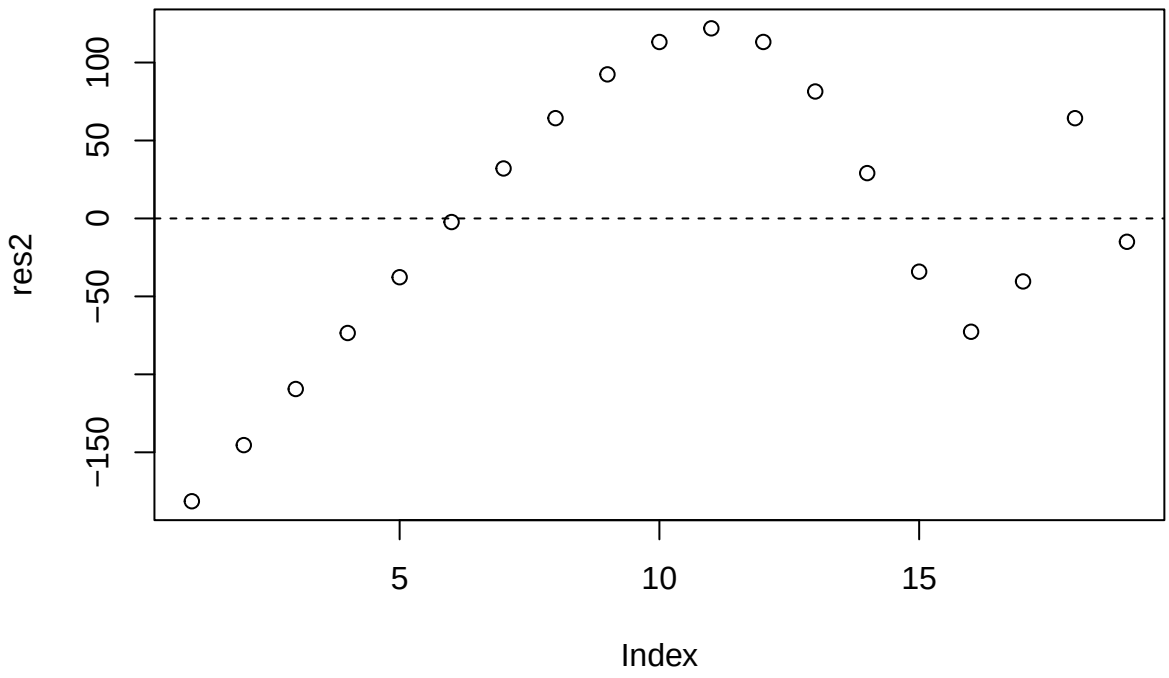
```r
plot(x = dataNL$pressure, y = dataNL$temperature)
lines(x = dataNL$pressure, y = predict(regression2), col = "red")
```



This fit is much better than the prior linear fit! The line of best fit tends to be much closer to the observed values (pairs of pressure-temperature) than before.

(i) Plot the residuals against the fitted values. Comment. [3 + 2 points]

```
res2 <- residuals(regression2)
plot(res2)
abline(h = 0, lty = "dashed")
```



The residual plot shows that our predictions tend to be more inaccurate for low pressure predictions, and specifically that these predictions tend to be greater than the observed values. Interestingly we see a change between overestimating (negative residual) and underestimating (positive residual) across the values. However, we also see the range of residual values is much smaller than the range of residual values from the first linear fit.