Homework 1

Steve Harms

September 8, 2017

Exercise 1

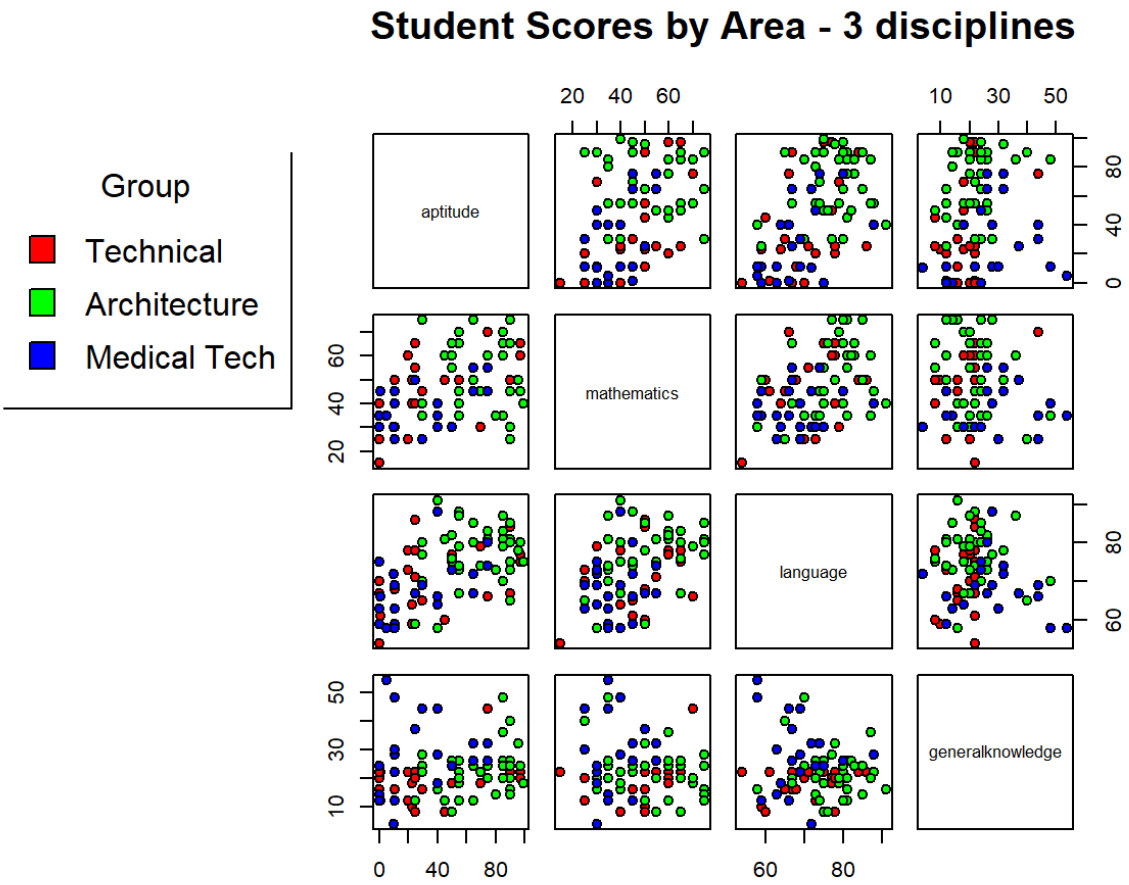
a)

```
#Read the data from the webpage. No headers, so I name the variables with col.names
()
scores <- read.table(file = "http://maitra.public.iastate.edu/stat579/datasets/stud
ent-apt.dat", header = FALSE, col.names = c("group", "aptitude","mathematics","lang
uage", "generalknowledge"))

#Then I attach the table so I can reference the variables directly
attach(scores)
```

b)

Note, the legend may not appear as below, depending on how your plotting area is formatted. It may move if you zoom in or zoom out.

```
#Create a scatterplot matrix to compare scores across all variables. I color them b
y group, and expand the margin area so I can include a legend.
pairs(~ aptitude + mathematics + language + generalknowledge, main = "Student Score
s by Area - 3 disciplines", pch = 21, bg = c("red", "green", "blue")[scores$group]
, oma = c(5,20,5,5))

#Now add a legend for the colors of each group. Need to specify location with x and
y coordinates
legend(x = .05, y = .903, legend = c("Technical", "Architecture", "Medical Tech"),
fill=c("red", "green", "blue"), title = "Group")
```



c)

In general, it appears that the architecture students scored consistently higher than other groups in every area except for general knowledge, in which medical technology students scored higher. The students in technical disiplines generally scored between the other 2 groups, and were very scattered in every area. Across all 3 groups there is a positive correlation between aptitude, mathematics, and language scores, but no visible correlation between general knowledge and any of the other scores areas.

Exercise 2

a)

```
#Read the Pi Digits file from the web page. I skip the first 60 irrelevant rows, an
d name the variable.
PiDigits <- read.table(file = "http://www.itl.nist.gov/div898/strd/univ/data/PiDigi
ts.dat", header = FALSE, skip = 60, col.names = "Digit")

#attach the table so I can reference the variable directly
attach(PiDigits)
```

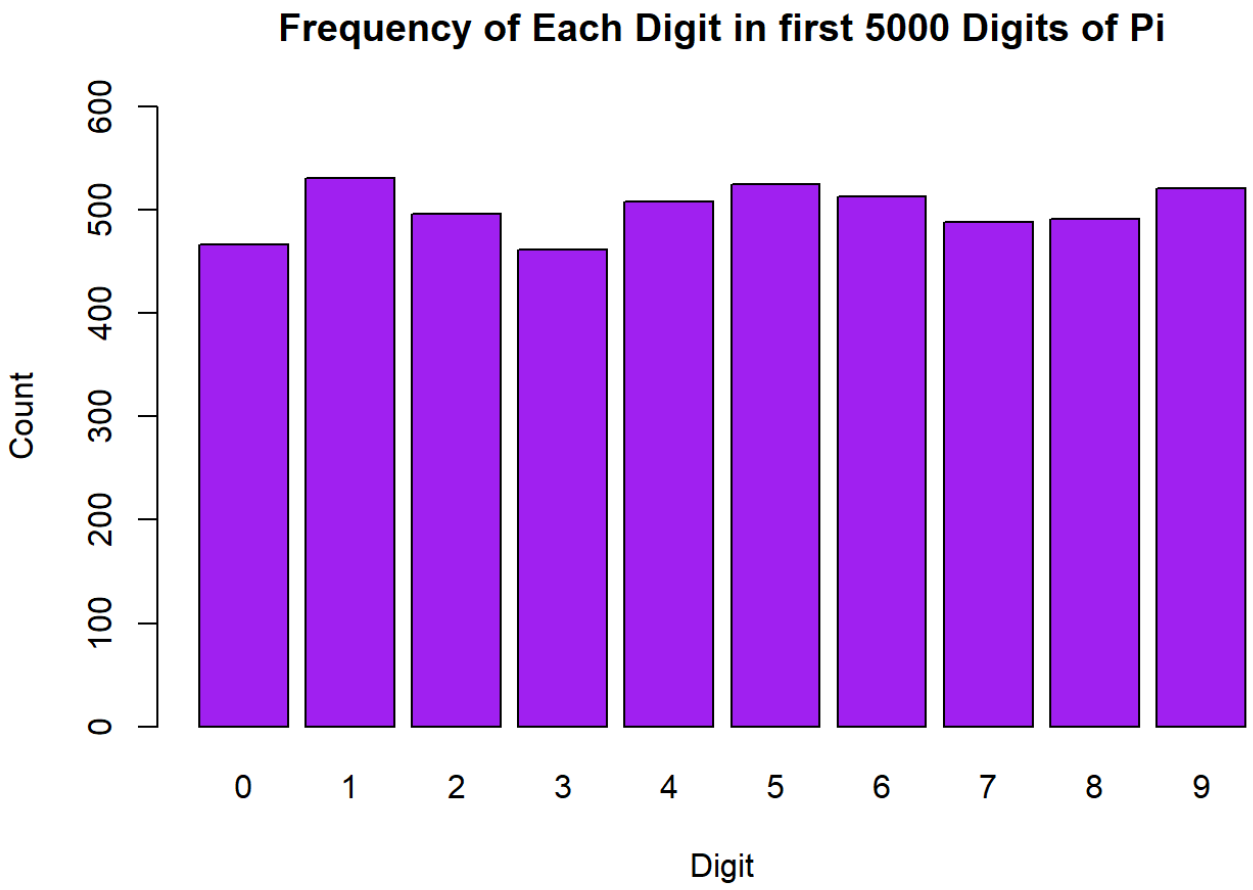
b)

```r
#Use the table() funciton to create a frequency table, then view the table in the c
onsole
freqtable <- table(PiDigits)

freqtable
```

```
## PiDigits
##   0   1   2   3   4   5   6   7   8   9
## 466 531 496 461 508 525 513 488 491 521
```

c)

```r
#Create a bar plot with using the frequency table. Add titles, expand the y-axis an
d change the color to make the chart more appealing
barplot(freqtable, main = "Frequency of Each Digit in first 5000 Digits of Pi", xla
b = "Digit", ylab = "Count", ylim = c(0, 600), col = "purple")
```



d)

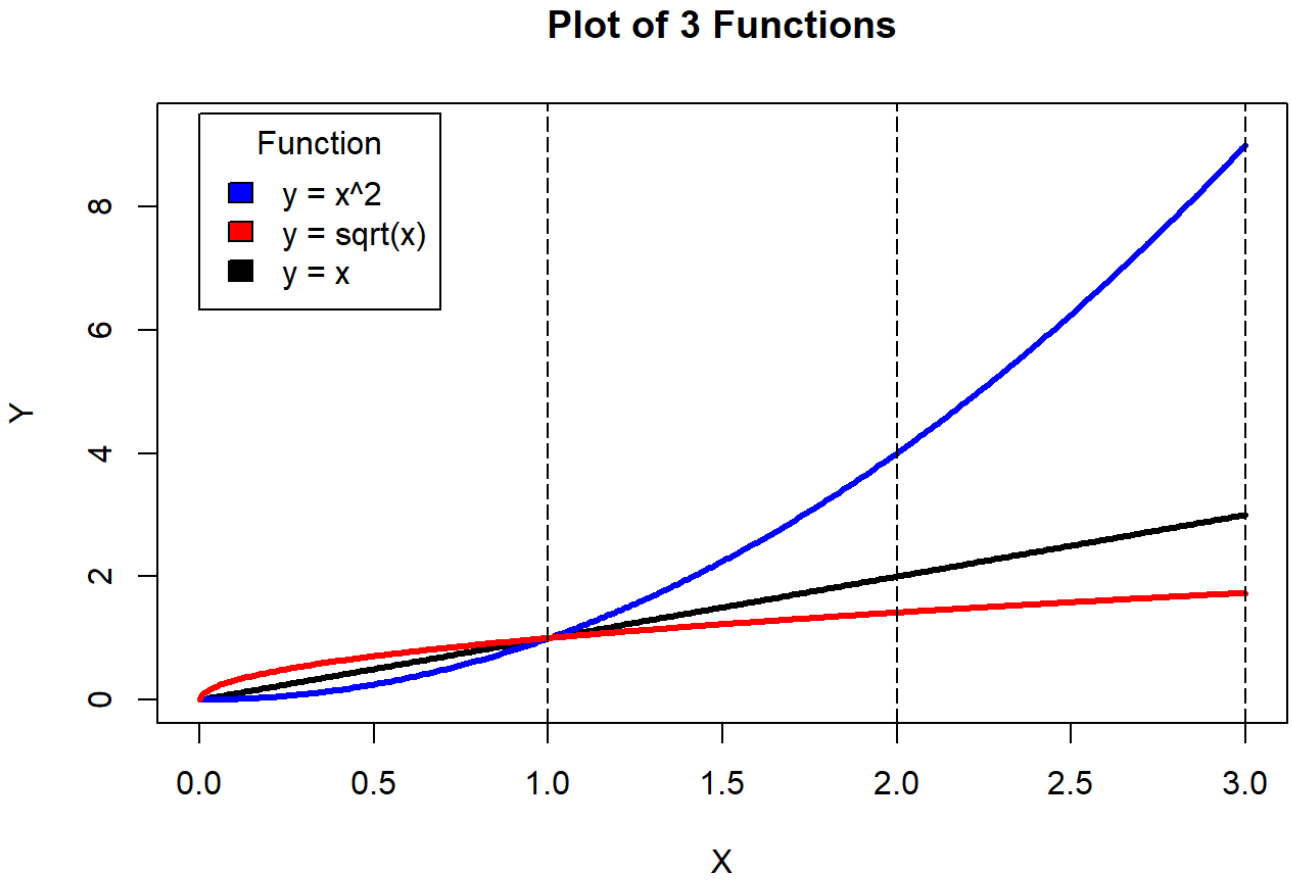
```r
#Use the chisq.test() function to conduct a chi-square test for the count of the di
gits
chisq.test(freqtable)
```

```
##
##  Chi-squared test for given probabilities
##
## data:  freqtable
## X-squared = 10.356, df = 9, p-value = 0.3224
```

The p-value for the Chi-square test is 0.3224, which means that we cannot reject the null hypothesis, which is that all digits appear at an equal rate. The bar plot illustrates this well. There is no evidence from the first 5000 digits that one digit appears more than any of the others.

Exercise 3

```
#First I generate my functions to plot.

xvals <- seq(from = 0, to = 3, by = .01) #generate a sequence of x values

y1 <- xvals #generate function y = x

y2 <- xvals ^ 2 #generate function y = x^2

y3 <- sqrt(xvals) #generate function y = sqrt(x)

df <- data.frame(xvals, y1, y2, y3) #combine them all into a data frame

#Plot all 3 lines in the same frame. I expand the y-axis to include the maximum val
ues, and also color the lines.

plot(x = xvals, y = y1, type = "l", lwd = 3, xlim = c(0 , max(xvals)), xlab = "X",
ylim = c(0, max(y1, y2, y3) + .3), ylab = "Y", main = "Plot of 3 Functions", oma =
c(5,20,5,5))

lines(x = xvals, y = y2, col = "blue", lwd = 3) #a line for the y = x^2 function

lines(x = xvals, y = y3, col = "red", lwd = 3) #another line for the y = sqrt(x) f
unction

abline(v= c(1,2,3), lty= 5) #add some vertical lines for x, I make them dashed

legend(x = 0, y = 9.5, legend = c("y = x^2", "y = sqrt(x)", "y = x"),  fill=c("blu
e", "red", "black"), title = "Function") #Add a legend for the curves
```



Exercise 4

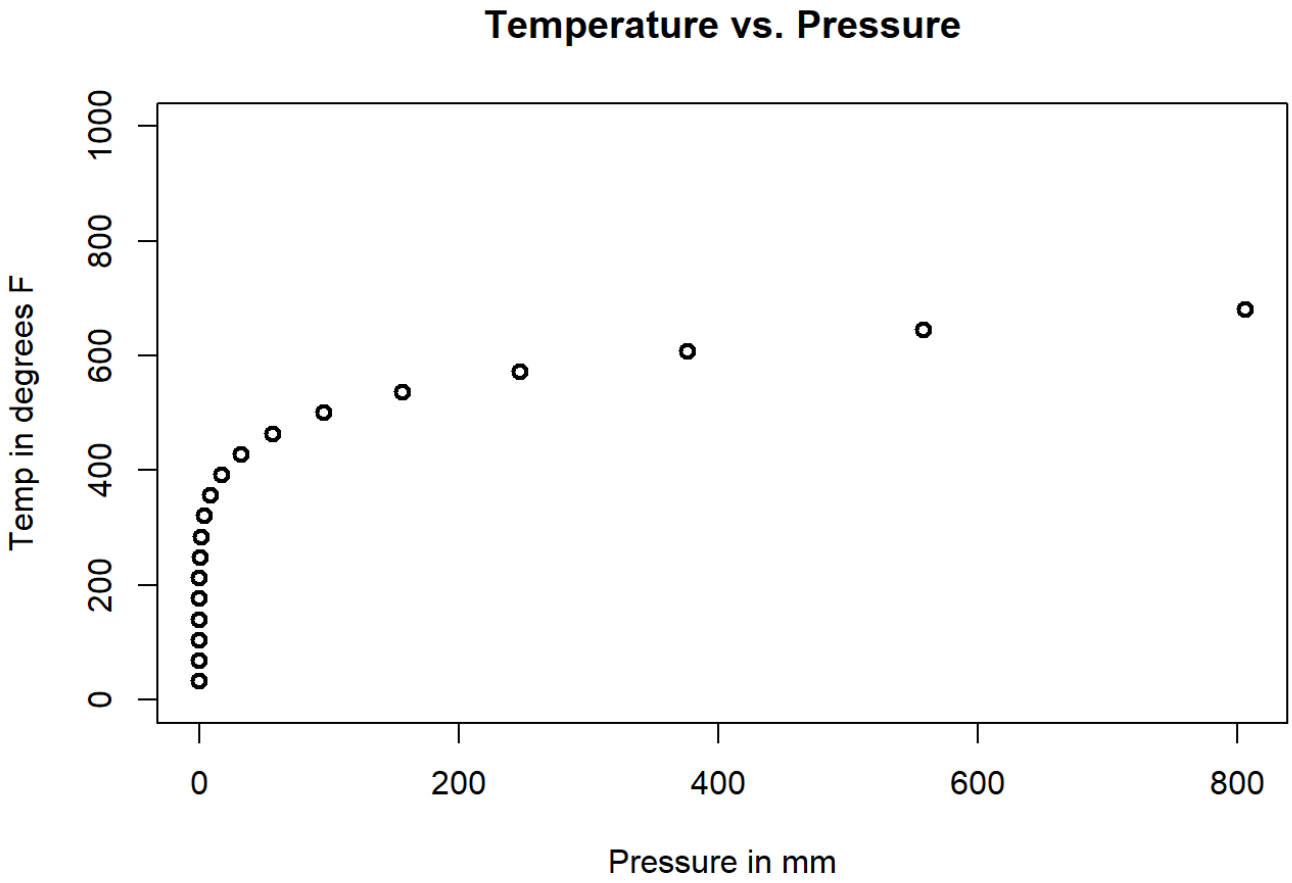
I did a) and b) together here. I put the pressure data into a data frame first, and then added my fahrenheit vector directly into the frame. The result should be the same.

a) and b)

```
frame <- data.frame(pressure) #create data frame

names(frame)= c("Celsius", "Pressure") #add my names for the variables into the fra
me

frame$Fahrenheit = 9*frame$Celsius/5 + 32 #Create a new variable that converts Cel
sius to Fahrenheit

attach(frame) #attach the frame for easy variable reference
```

c)

```
#Plot Fahrenheit vs. Pressure, and store it as a variable for easy reference later
plot(y = Fahrenheit, x = Pressure, type = "p", lwd = 2, col = "black", main = "Tem
perature vs. Pressure", ylab = "Temp in degrees F", xlab="Pressure in mm", ylim = c
(0, 1000))
```



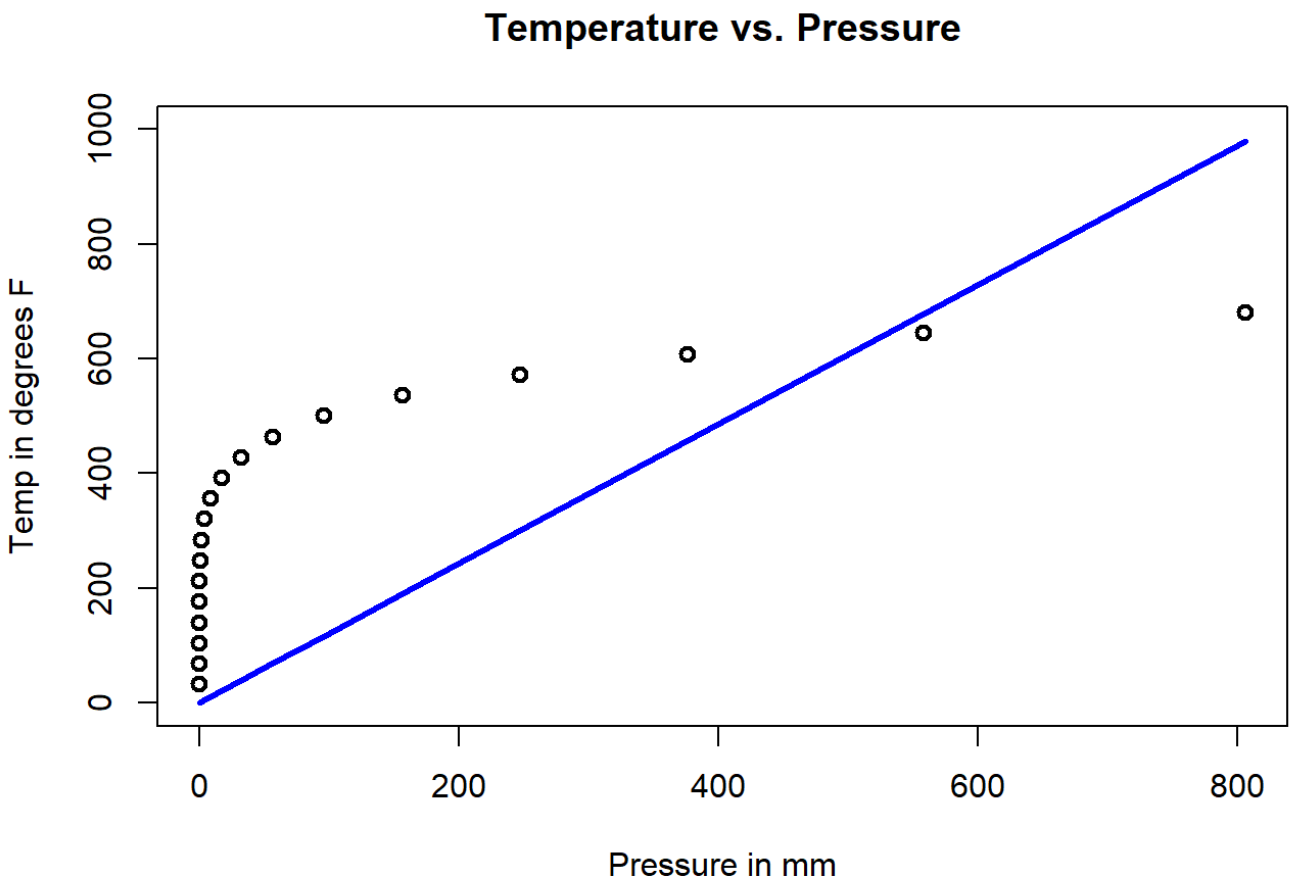

```
plotC <- recordPlot() #save the plot for later
```

d)

```
#Create a simple linear regression model for Temperature vs. Pressure
noint <- lm(formula = Fahrenheit ~ Pressure -1, data=frame) #The -1 will remove th
e intercept term from the model

fitted <- fitted(noint) #Create fitted values from the model

summary(noint) #View the summary of the model results
```

```
##
## Call:
## lm(formula = Fahrenheit ~ Pressure - 1, data = frame)
##
## 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
```

```
plot.new() #Clear the previous plot

plotC #recall the old plot

lines(x = Pressure, y = fitted, lwd = 3, col = "blue") #Plot the fitted line onto
the plot from (c)
```

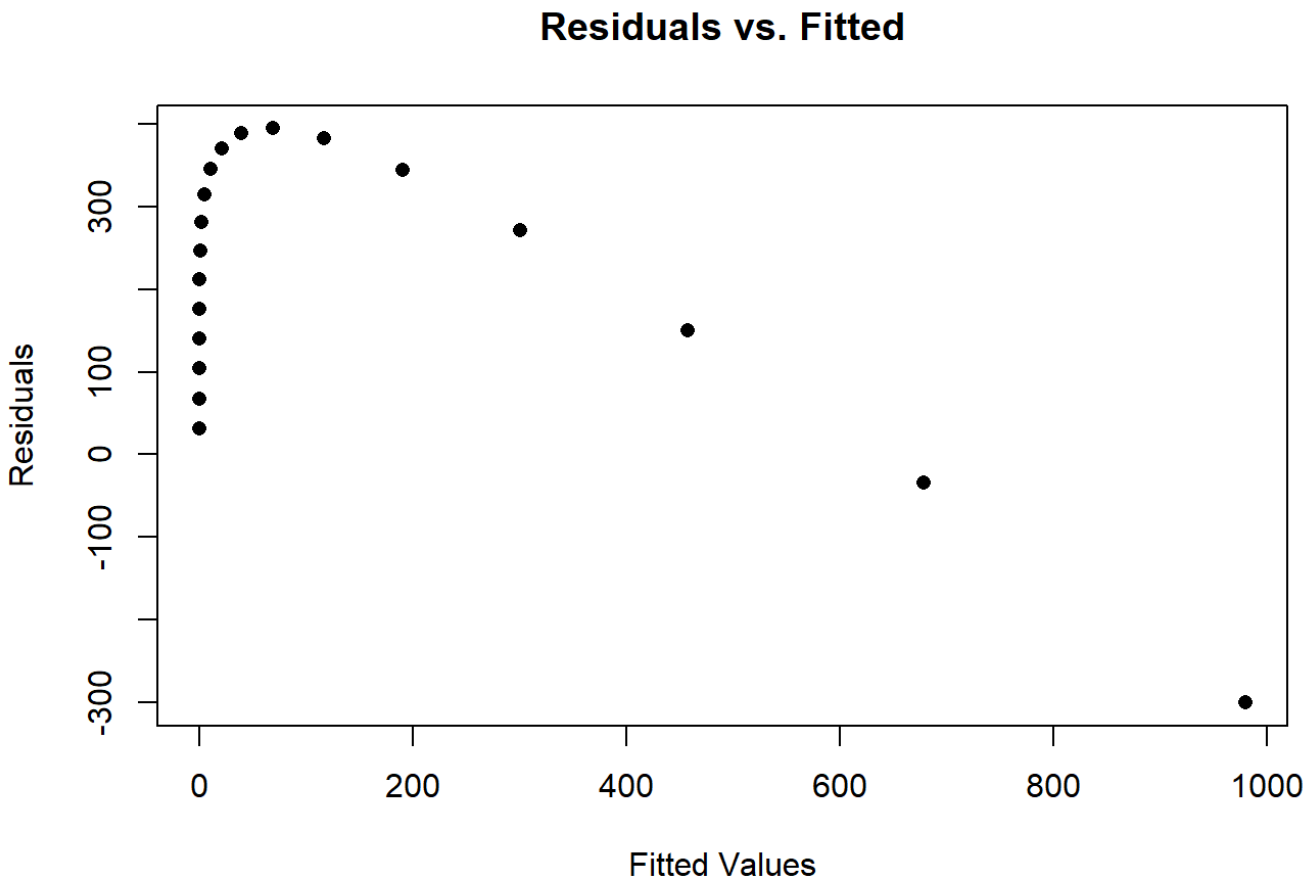



```
plotD <- recordPlot() #Save the plot for later
```

The coefficient estimated by the model is significant at a 5% level, but it is clear from the plot that the simple linear model does not fit the relationship well. The R-squared is only .56, and using a non-linear or higher order model will probably fit better.

e)

```
#Create a vector of residuals from the no intercept simple linear regression
resids <- resid(noint)

#Plot the Residuals vs. the Fitted values
plot.new()

residualplotE <- plot(x = fitted, y = resids, xlab = "Fitted Values", ylab = "Resi
duals", main = "Residuals vs. Fitted", pch = 16)
```



Clearly the residuals are highly correlated to another function, as they fit a very smooth curve. Also note that all except for the last 2 are positive. If we removed one or both of those values, our line would change drastically. Both of the above indicate that the simple linear regression is not a good fit. The residuals do not have constant variance, and clearly are not normally distributed

f)

```r
#Create a new data frame, directly applying the functions to pressure data
newframe <- data.frame(Fahrenheit, Pressure, Pressure^2, Pressure^3)

#Add names for our new vectors in our new data frame
names(newframe)<- c("Fahrenheit", "Pressure", "SquaredPressure", "CubedPressure")

detach(frame) #detach the old data frame

attach(newframe)  #attach the new one
```

g)

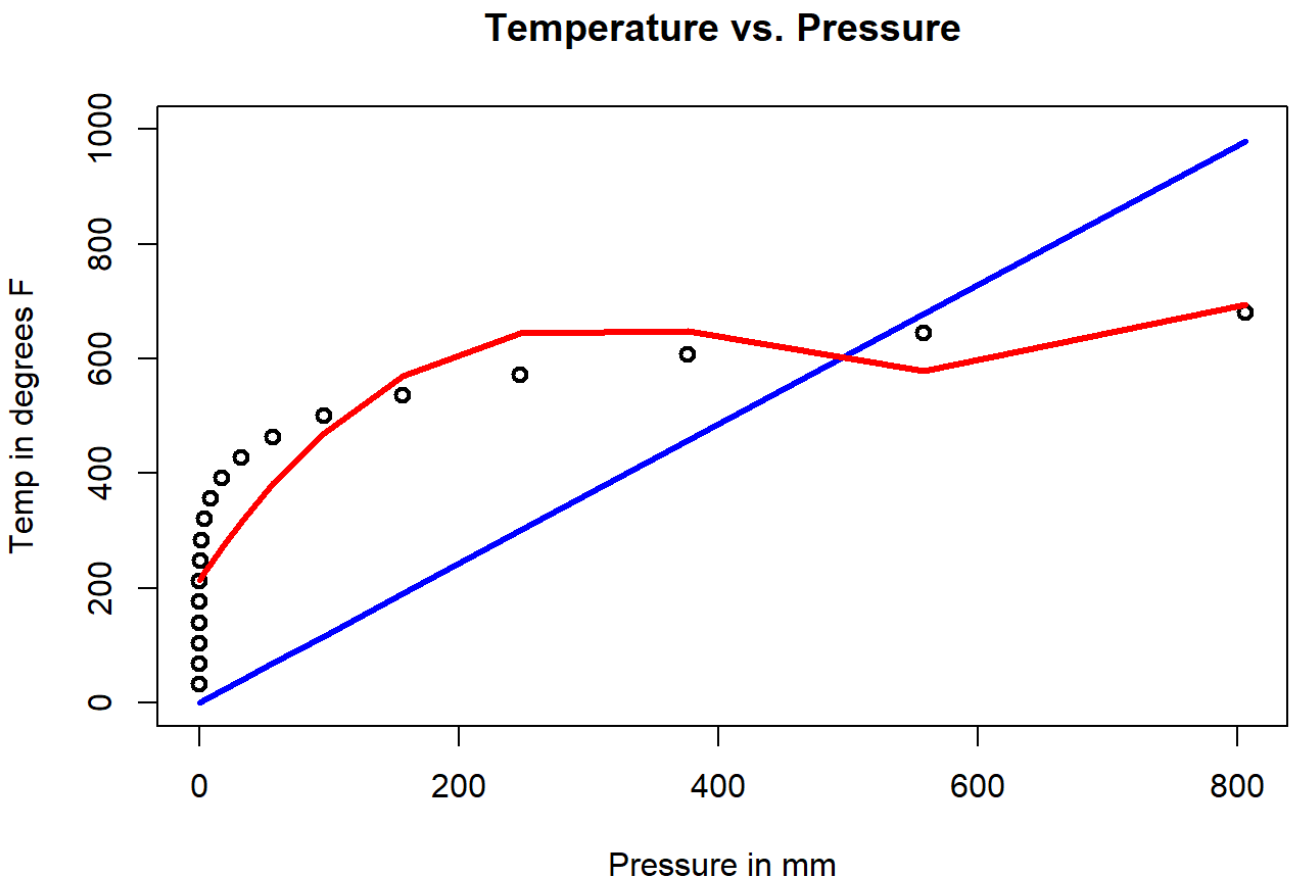
```r
#Create a multiple regression model using our new function
newmodel <- lm(formula = Fahrenheit ~ Pressure + SquaredPressure + CubedPressure,
data=newframe)

#view a summary for the model output
summary(newmodel)
```

```
##
## Call:
## lm(formula = Fahrenheit ~ Pressure + SquaredPressure + CubedPressure,
##     data = newframe)
##
## 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 ***
## SquaredPressure -8.207e-03  2.826e-03  -2.904 0.010898 *
## CubedPressure    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 3 coefficients and the intercept are significant at the 5% level, although the higher order coefficients may not be significant if we used alpha = 1%. This, along with higher Adjusted R-Squared indicate we may have a better fitting model here.

h)

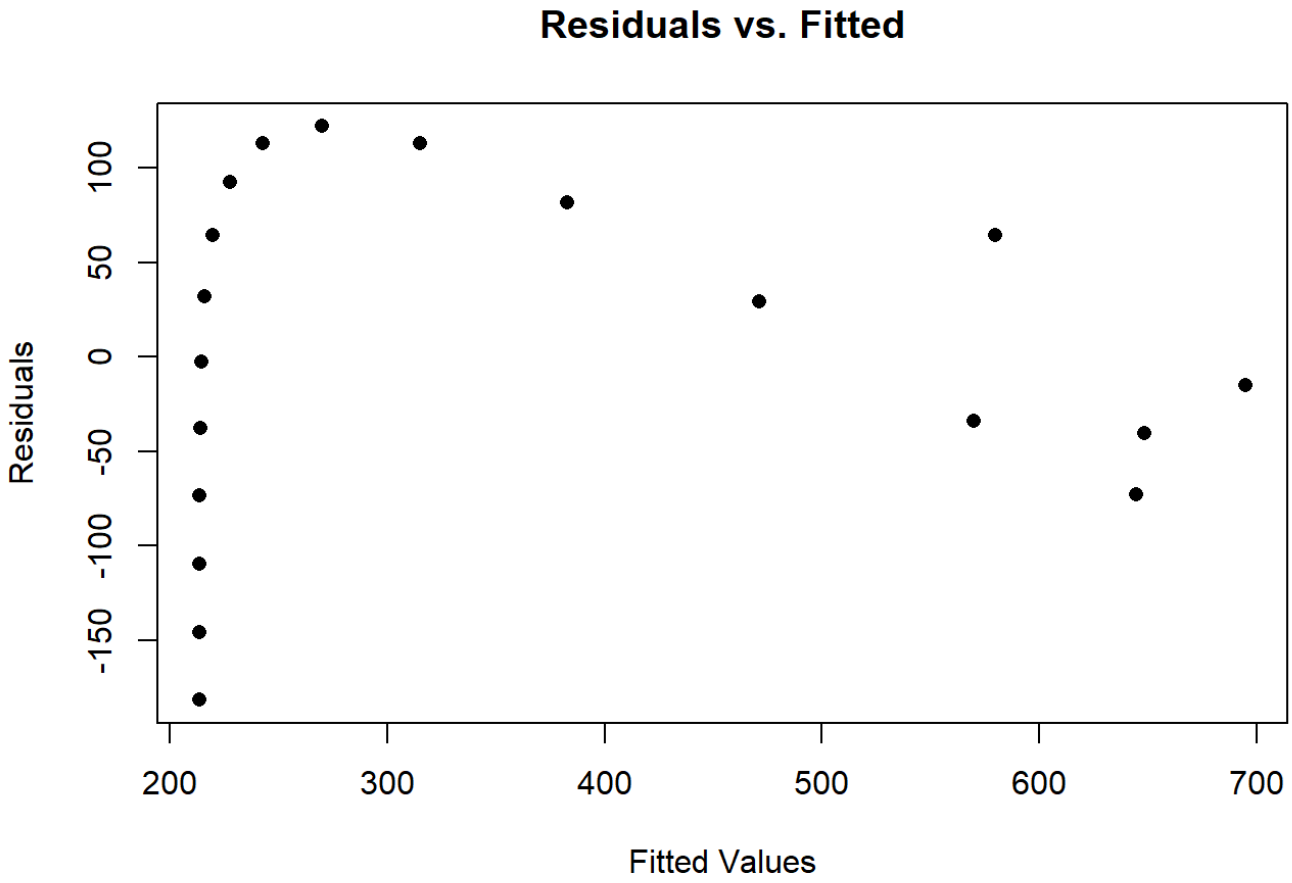
```
#First we create fitted values for the new multiple regression model
newfit <- fitted(newmodel)

#Then add it into our plot with previous regression line and actual data
plot.new()

plotD #Recall the plot from earlier

lines(x = Pressure, y = newfit, lwd = 3, col = "red") #add the new fitted line to
the old plot
```



This new curve (in red) is shows a much better fit than the simple linear regression, but is still not ideal.

i)

```
#Create Residuals from the new multiple regression model
newresids <- resid(newmodel)

#Create a new residual vs. fitted value plot
plot.new()

residualplotI <- plot(x = newfit, y = newresids, xlab = "Fitted Values", ylab= "Re
siduals", main = "Residuals vs. Fitted", pch = 16)
```



The residuals still show a somewhat clear correlation to some function, but for this model they are smaller in absolute value and there are fewer residuals that have high leverage and would drastically change the model if we removed the data point. So, while the fit is better, the residuals still do not appear normally distributed and are definitely not of constant variance.

```
rm(list = ls()) # Remove all of the variables from the workspace
```