Analysis of the Prediction of Melting Points

Samuel Dummer

9/14/2021

Abstract
This document summarizes the analyses performed on data comparing many different compound's melting points in relation to many other factors. These values include, the number of aromatic rings, atoms heavier than helium, single bond, triple bonds, whether or not the compound is reactive, the molar mass of each compound, the refractivity, the formal charge, the topological polar surface area, the dipole moment, the energy, the density, the volume, and the principal component analysis 1 and 2. We started off by loading in the data and checking to see how clean it is. Then, we created a summary of the data and cleaned up anything else that needed cleaning. Additionally, as part of our Exploratory Data Analysis(EDA), we created some histograms, linear regression, BIC scores, and multiple regression.

Model Setup

Before any EDA or cleaning of the data we need to clear the environment and set the working directory.

```
rm(list=ls())
setwd("C:/Users/isabe/Desktop/RFLoder")
```

Loading in Library and User Functions

Once the environment is cleared and the directory is set, we had to load in any libraries we were going to use, in this case tidyverse, and a use function in case we needed to use it anywhere.

```
source("myfunctions.R")
library(tidyverse)
```

Loading in the Dataset

Finally, once everything is set up, we are ready to load in the data for the working directory that we provided.

```
meltpoint <- read.csv("dirtyMPdata.csv", header = T)
```

Checking to See How Clean the Data Given is

After the data is loaded, we would first like to check how dirty/how many missing values there are in the table. We do so by looking through all the values and marking it as a 1 if it has a values in it and 0 if it is blank. Then we output the number of blanks to tell us how many rows have missing data.

```
clean <- ifelse(complete.cases(meltpoint)==TRUE,1,0)
table(clean)
```

```
## clean
##    0    1
##   13 4437
```

```
paste("There are ",dim(meltpoint)[1]-sum(clean), " rows with missing data.")
```

```
## [1] "There are  13  rows with missing data."
```

Removing Rows with Missing Data

Since there were only 13 rows out of 4450 that had missing data we decided that it would be best if we removed the rows with missing values since it wouldn't affect the dataset too much. This was done using the na.omit command which looks through the data for missing values and removes any rows with that have missing values.

```
meltpoint <- na.omit(meltpoint)
```

Looking Over the Data Strucure and Summary

After the data is loaded in, we would like to see the overall summary and structure of the data. This can be done by running many different commands to tell us different information about this data set. This includes the head, tail, names, dim, str, sapply, and summary commands.

```
names(meltpoint)
```

```
##  [1] "ï..structure"  "mp"            "rings"         "heavy.atoms"
##  [5] "single.bonds"  "triple.bonds"  "reactive"      "molar.mass"
##  [9] "refractivity"  "formal.charge" "logP"          "tpsa"
## [13] "dipole.moment" "energy"        "density"       "volume"
## [17] "PCA1"          "PCA2"
```

```
dim(meltpoint)
```

```
## [1] 4437   18
```

```
str(meltpoint)
```

```
## 'data.frame':    4437 obs. of  18 variables:
##  $ ï..structure : chr  "O=C1Cc2ccccc21" "Clc1ccc(cc1)C1c2c(OC(N)=C1C#N)[nH][nHO]c2C(F)(F)F" "O=C(OC)(
##  $ mp           : num  14 20.5 27.5 30.5 31 31.5 32 32.5 33 34 ...
##  $ rings        : int  6 11 12 10 6 6 6 5 12 12 ...
##  $ heavy.atoms  : int  9 23 19 14 12 14 12 10 16 14 ...
##  $ single.bonds : int  9 20 22 10 16 19 14 12 21 16 ...
##  $ triple.bonds : int  0 1 0 0 0 0 0 0 0 0 ...
##  $ reactive     : int  0 0 0 0 1 0 0 0 0 0 ...
```

```
##  $ molar.mass   : num  118 341 252 197 162 ...
##  $ refractivity : num  3.56 7.73 7.8 4.73 4.65 ...
##  $ formal.charge: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ logP         : num  1.43 3.64 3.49 3.57 1.94 ...
##  $ tpsa         : num  17.1 87.7 26.3 12.9 26.3 ...
##  $ dipole.moment: num  2.72 6.32 1.91 5.2 1.76 ...
##  $ energy       : num  -57.8 -189.1 -123.5 -119.7 -83.5 ...
##  $ density      : num  0.959 1.296 0.946 1.195 0.978 ...
##  $ volume       : num  123 263 267 165 166 ...
##  $ PCA1         : num  15.55 1.92 2.98 11.03 10.97 ...
##  $ PCA2         : num  1.622 0.502 2.723 1.826 -0.905 ...
##  - attr(*, "na.action")= 'omit' Named int [1:13] 980 1473 1718 2388 2402 2434 2551 2774 2794 2927 ..
##   ..- attr(*, "names")= chr [1:13] "980" "1473" "1718" "2388" ...
```

```
sapply(meltpoint, class)
```

```
##  ï..structure            mp         rings   heavy.atoms  single.bonds
##   "character"     "numeric"     "integer"     "integer"     "integer"
##  triple.bonds      reactive    molar.mass  refractivity formal.charge
##     "integer"     "integer"     "numeric"     "numeric"     "integer"
##          logP          tpsa dipole.moment        energy       density
##     "numeric"     "numeric"     "numeric"     "numeric"     "numeric"
##        volume          PCA1          PCA2
##     "numeric"     "numeric"     "numeric"
```

```
head(meltpoint)
```

```
##                                       ï..structure   mp rings heavy.atoms
## 1                                     O=C1Cc2ccccc21 14.0     6           9
## 2 Clc1ccc(cc1)C1c2c(OC(N)=C1C#N)[nH][nH0]c2C(F)(F)F 20.5    11          23
## 3                      O=C(OC)C(=Cc1ccccc1)Cc1ccccc1 27.5    12          19
## 4                         FC(F)(F)c1[nH0]cc2ccccc2c1 30.5    10          14
## 5                               O=C(OC1Cc2ccccc21)C 31.0     6          12
## 6                            O=C(OC)C1=Cc2ccccc2C1C 31.5     6          14
##   single.bonds triple.bonds reactive molar.mass refractivity formal.charge
## 1            9            0        0    118.135     3.557232             0
## 2           20            1        0    340.692     7.729887             0
## 3           22            0        0    252.313     7.799344             0
## 4           10            0        0    197.159     4.729193             0
## 5           16            0        1    162.188     4.651153             0
## 6           19            0        0    188.226     5.533352             0
##       logP  tpsa dipole.moment     energy   density   volume    PCA1    PCA2
## 1 1.425370 17.07      2.722717  -57.76601 0.9589853 123.1875 15.5507  1.6219
## 2 3.637884 87.72      6.320718 -189.06262 1.2962539 262.8281  1.9216  0.5024
## 3 3.485670 26.30      1.911006 -123.45357 0.9463216 266.6250  2.9803  2.7226
## 4 3.565100 12.89      5.203625 -119.71445 1.1952425 164.9531 11.0316  1.8263
## 5 1.942370 26.30      1.759848  -83.47468 0.9782332 165.7969 10.9710 -0.9051
## 6 2.360100 26.30      2.046720  -95.11963 0.9741601 193.2188  8.9700  0.3246
```

```
tail(meltpoint)
```

```
##                                           ï..structure  mp rings
```

```
## 4445                        N=1CCNC=1Cc1c(C)cc(cc1C)C(C)(C)C 131     6
## 4446            O=C(OC)C1C(O)CCC2CN3CCc4c5ccccc5[nH]c4C3CC21 234     9
## 4447              O=C1NC(=O)C(C)=CN1C1OC(CO)C(N=[N+]=[N-])C1 106     0
## 4448                                s1c2ccccc2cc1C(N(O)C(=O)N)C 157     9
## 4449 [S+2]([O-])([O-])(C)c1ccc(cc1)c1[nH0]c2[nH0](C=CC=C2)c1 242    11
## 4450                      Clc1ccc2Sc3ccccc3C=C(OCCN(C)C)c2c1  90    12
##      heavy.atoms single.bonds triple.bonds reactive molar.mass refractivity
## 4445          18           37            0        0    245.390     7.652000
## 4446          26           46            0        1    355.458     9.993037
## 4447          19           28            0        1    267.245     6.299167
## 4448          16           18            0        1    236.295     6.457717
## 4449          19           20            0        0    272.328     7.474073
## 4450          22           30            0        0    332.875     9.547561
##      formal.charge    logP   tpsa dipole.moment    energy   density   volume
## 4445             1 1.22571  26.00     13.028417 -112.7691 0.8720617 281.3906
## 4446             1 1.32547  66.76      8.226915 -177.9712 1.0188236 348.8906
## 4447             0 0.22820 103.59      4.728840 -146.1079 1.1644663 229.5000
## 4448             0 2.82770  66.56      1.821058 -112.0673 1.0833784 218.1094
## 4449             0 2.45110  51.96      8.410813 -127.4689 1.0959562 248.4844
## 4450             1 3.26210  13.67     23.597216 -146.3722 1.0327710 322.3125
##         PCA1    PCA2
## 4445  2.2338  1.6217
## 4446 -3.7905 -1.1574
## 4447  2.7954 -7.6893
## 4448  5.4399 -3.3301
## 4449  3.0028 -0.5994
## 4450 -0.2923  4.3546
```

```
summary(meltpoint)
```

```
##  ï..structure             mp            rings         heavy.atoms
##  Length:4437        Min.   : 14.0   Min.   : 0.000   Min.   : 6.00
##  Class :character   1st Qu.:117.0   1st Qu.: 6.000   1st Qu.:17.00
##  Mode  :character   Median :161.0   Median :11.000   Median :22.00
##                     Mean   :165.2   Mean   : 9.943   Mean   :22.19
##                     3rd Qu.:209.5   3rd Qu.:12.000   3rd Qu.:26.00
##                     Max.   :392.5   Max.   :36.000   Max.   :59.00
##   single.bonds     triple.bonds       reactive        molar.mass
##  Min.   :  4.00   Min.   :0.0000   Min.   :0.0000   Min.   : 84.08
##  1st Qu.: 19.00   1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:243.22
##  Median : 24.00   Median :0.0000   Median :0.0000   Median :307.35
##  Mean   : 28.16   Mean   :0.1062   Mean   :0.2206   Mean   :316.80
##  3rd Qu.: 33.00   3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:374.82
##  Max.   :102.00   Max.   :4.0000   Max.   :1.0000   Max.   :815.62
##   refractivity    formal.charge          logP             tpsa
##  Min.   : 1.992   Min.   :-3.00000   Min.   :-6.023   Min.   :  0.00
##  1st Qu.: 6.602   1st Qu.: 0.00000   1st Qu.: 2.103   1st Qu.: 38.33
##  Median : 8.300   Median : 0.00000   Median : 3.214   Median : 61.03
##  Mean   : 8.496   Mean   : 0.01217   Mean   : 3.328   Mean   : 67.37
##  3rd Qu.:10.156   3rd Qu.: 0.00000   3rd Qu.: 4.524   3rd Qu.: 87.72
##  Max.   :19.354   Max.   : 2.00000   Max.   :12.780   Max.   :413.24
##  dipole.moment          energy           density          volume
##  Min.   :  0.0051   Min.   :-489.00   Min.   :0.7945   Min.   : 78.89
##  1st Qu.:  2.3868   1st Qu.:-186.18   1st Qu.:0.9814   1st Qu.:230.34
```

```
##  Median :  3.8198   Median :-150.42   Median :1.0541   Median :284.77
##  Mean   :  5.2256   Mean   :-157.51   Mean   :1.0806   Mean   :293.83
##  3rd Qu.:  5.6404   3rd Qu.:-118.36   3rd Qu.:1.1433   3rd Qu.:348.47
##  Max.   :248.3031   Max.   : -44.35   Max.   :1.8956   Max.   :681.75
##       PCA1               PCA2
##  Min.   :-33.38560   Min.   :-19.975400
##  1st Qu.: -4.03710   1st Qu.: -2.856000
##  Median :  0.67600   Median :  0.298300
##  Mean   :  0.04787   Mean   : -0.006885
##  3rd Qu.:  4.69770   3rd Qu.:  3.254400
##  Max.   : 17.34860   Max.   : 12.489900
```

Changing the Labels for the "Reactive" columns

Once we have looked over the summary and strucure of the data, we noticed that the reactive column was stored as 1s and 0s and decided to use the factor command to label the 1s as yes, for yes it is reactive, and 0s for no, for no its isn't reactive.

```
meltpoint$reactive  <- factor(meltpoint$reactive, levels=c(0,1), labels = c("no", "yes"))
head(meltpoint)
```

```
##                                         ï..structure   mp rings heavy.atoms
## 1                                     O=C1Cc2ccccc21 14.0     6           9
## 2 Clc1ccc(cc1)C1c2c(OC(N)=C1C#N)[nH][nH0]c2C(F)(F)F 20.5    11          23
## 3                      O=C(OC)C(=Cc1ccccc1)Cc1ccccc1 27.5    12          19
## 4                         FC(F)(F)c1[nH0]cc2ccccc2c1 30.5    10          14
## 5                               O=C(OC1Cc2ccccc21)C 31.0     6          12
## 6                             O=C(OC)C1=Cc2ccccc2C1C 31.5     6          14
##   single.bonds triple.bonds reactive molar.mass refractivity formal.charge
## 1            9            0       no    118.135     3.557232             0
## 2           20            1       no    340.692     7.729887             0
## 3           22            0       no    252.313     7.799344             0
## 4           10            0       no    197.159     4.729193             0
## 5           16            0      yes    162.188     4.651153             0
## 6           19            0       no    188.226     5.533352             0
##       logP  tpsa dipole.moment     energy   density   volume    PCA1    PCA2
## 1 1.425370 17.07      2.722717  -57.76601 0.9589853 123.1875 15.5507  1.6219
## 2 3.637884 87.72      6.320718 -189.06262 1.2962539 262.8281  1.9216  0.5024
## 3 3.485670 26.30      1.911006 -123.45357 0.9463216 266.6250  2.9803  2.7226
## 4 3.565100 12.89      5.203625 -119.71445 1.1952425 164.9531 11.0316  1.8263
## 5 1.942370 26.30      1.759848  -83.47468 0.9782332 165.7969 10.9710 -0.9051
## 6 2.360100 26.30      2.046720  -95.11963 0.9741601 193.2188  8.9700  0.3246
```

Renaming Columns

There were also some problems with the columns names. We decided that it would be a good idea to change "ï..structure" to "structure" and "mp" to "melting.point" to make the data more understandable.

```
meltpoint <- rename(meltpoint, "structure" = "ï..structure", "melting.point" = "mp")
names(meltpoint)
```

```
##  [1] "structure"     "melting.point" "rings"         "heavy.atoms"
##  [5] "single.bonds"  "triple.bonds"  "reactive"      "molar.mass"
##  [9] "refractivity"  "formal.charge" "logP"          "tpsa"
## [13] "dipole.moment" "energy"        "density"       "volume"
## [17] "PCA1"          "PCA2"
```

Creating a Quick Summary Table

Next, we decided it would be in our best interest to create a summary table that goes over the mean, first quartile, median, third quartile, minimum, and maximum values. We tried to come as close as we could to the table in the reading. We mostly got is, but the mean and median were slightly off since we removed some of the rows. Luckily, this difference was very minute.

```
sumtab <- meltpoint[c(2, 8, 4, 11, 9, 13)]
summary(sumtab)
```

```
##  melting.point     molar.mass      heavy.atoms         logP
##  Min.   : 14.0   Min.   : 84.08   Min.   : 6.00   Min.   :-6.023
##  1st Qu.:117.0   1st Qu.:243.22   1st Qu.:17.00   1st Qu.: 2.103
##  Median :161.0   Median :307.35   Median :22.00   Median : 3.214
##  Mean   :165.2   Mean   :316.80   Mean   :22.19   Mean   : 3.328
##  3rd Qu.:209.5   3rd Qu.:374.82   3rd Qu.:26.00   3rd Qu.: 4.524
##  Max.   :392.5   Max.   :815.62   Max.   :59.00   Max.   :12.780
##   refractivity    dipole.moment
##  Min.   : 1.992   Min.   :  0.0051
##  1st Qu.: 6.602   1st Qu.:  2.3868
##  Median : 8.300   Median :  3.8198
##  Mean   : 8.496   Mean   :  5.2256
##  3rd Qu.:10.156   3rd Qu.:  5.6404
##  Max.   :19.354   Max.   :248.3031
```

Creating Linear Regressions and Saving Fitted Values

Once we have finished all our summarising and cleaning of the data, it is on to EDA. To start, we found the slope and intercept of the linear regression and used that to find the fitted values for formal charge, volume, and refractivity all compared to melting point. We did this by using the lm function which gives us the slope and intercept of each linear regression. Then we ran through each linear regression and found the fitted values for each. In the end, this gave us the predicted melting point values for each comparisson.

```
meltpointFit <- lm(meltpoint$melting.point~meltpoint$formal.charge)
meltchargeFit <-fitted.values(meltpointFit)
meltpointFit #Formal Charge Linear Regression
```

```
##
## Call:
## lm(formula = meltpoint$melting.point ~ meltpoint$formal.charge)
##
## Coefficients:
##              (Intercept)  meltpoint$formal.charge
##                    165.5                   -22.4
```

```
meltpointFit1 <- lm(meltpoint$melting.point~meltpoint$volume)
meltvolumeFit <-meltpointFit1$fitted.values
meltpointFit1 #Volume Linear Regression
```

```
##
## Call:
## lm(formula = meltpoint$melting.point ~ meltpoint$volume)
##
## Coefficients:
##      (Intercept)  meltpoint$volume
##         122.4875            0.1455
```

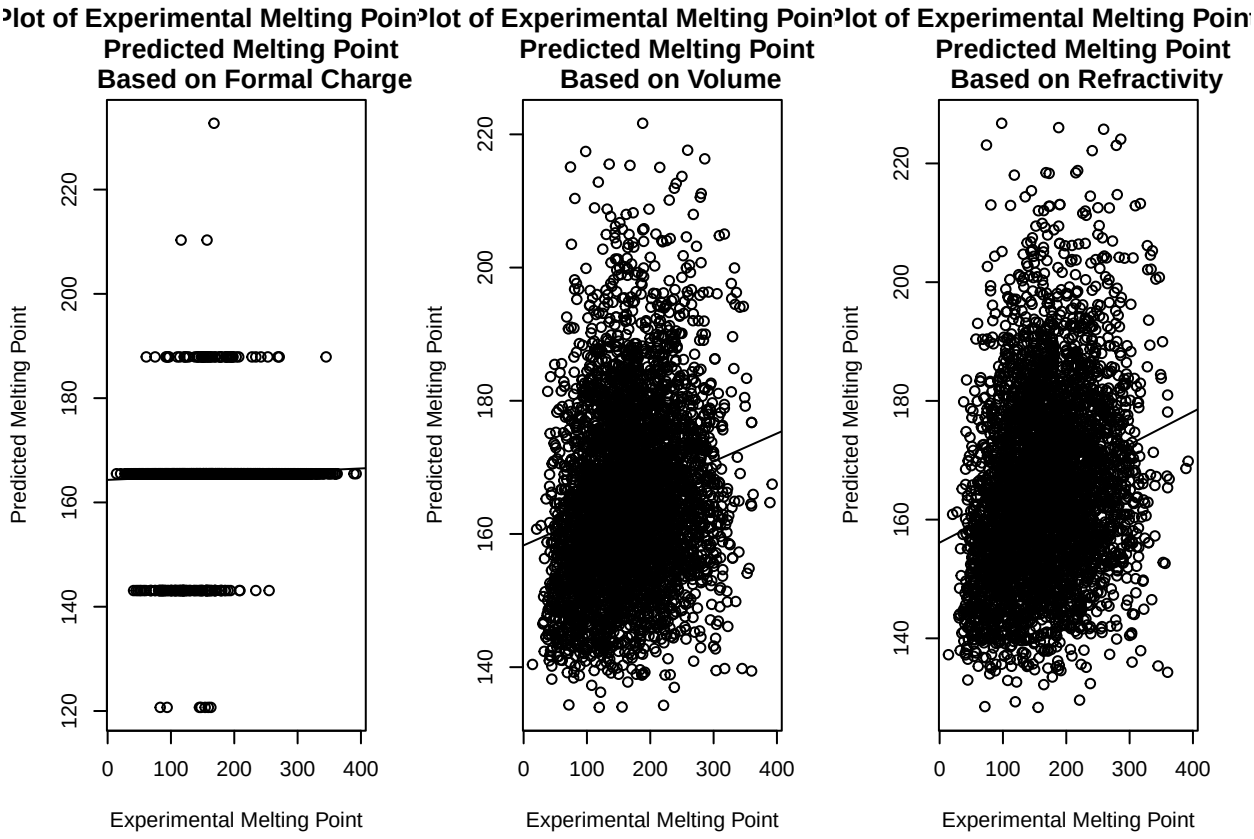
```
meltpointFit2 <- lm(meltpoint$melting.point~meltpoint$refractivity)
meltrefracFit <-meltpointFit2$fitted.values
meltpointFit2 #Refractivity Linear Regression
```

```
##
## Call:
## lm(formula = meltpoint$melting.point ~ meltpoint$refractivity)
##
## Coefficients:
##            (Intercept)  meltpoint$refractivity
##                117.080                   5.667
```

Plotting the Predicted Values vs. Experimental Values

Once we have obtained the predicted values above, we are able to plot it against the experimental values to create a scatter plot of the Predicted Values vs. Experimental Values. We also plotted all the scatter plots together using the par function. Additionally, we plotted linear regression for each scatter plot.

```
par(mfrow = c(1,3))
plot(meltchargeFit~meltpoint$melting.point,
     main = "Plot of Experimental Melting Point vs.
     Predicted Melting Point
     Based on Formal Charge", xlab = "Experimental Melting Point",
     ylab = "Predicted Melting Point")
fitline <- lm(meltchargeFit~meltpoint$melting.point)
abline(fitline)
plot(meltvolumeFit~meltpoint$melting.point,
     main = "Plot of Experimental Melting Point vs.
     Predicted Melting Point
     Based on Volume", xlab = "Experimental Melting Point",
     ylab = "Predicted Melting Point")
fitline <- lm(meltvolumeFit~meltpoint$melting.point)
abline(fitline)
plot(meltrefracFit~meltpoint$melting.point,
     main = "Plot of Experimental Melting Point vs.
     Predicted Melting Point
     Based on Refractivity", xlab = "Experimental Melting Point",
     ylab = "Predicted Melting Point")
fitline <- lm(meltrefracFit~meltpoint$melting.point)
abline(fitline)
```

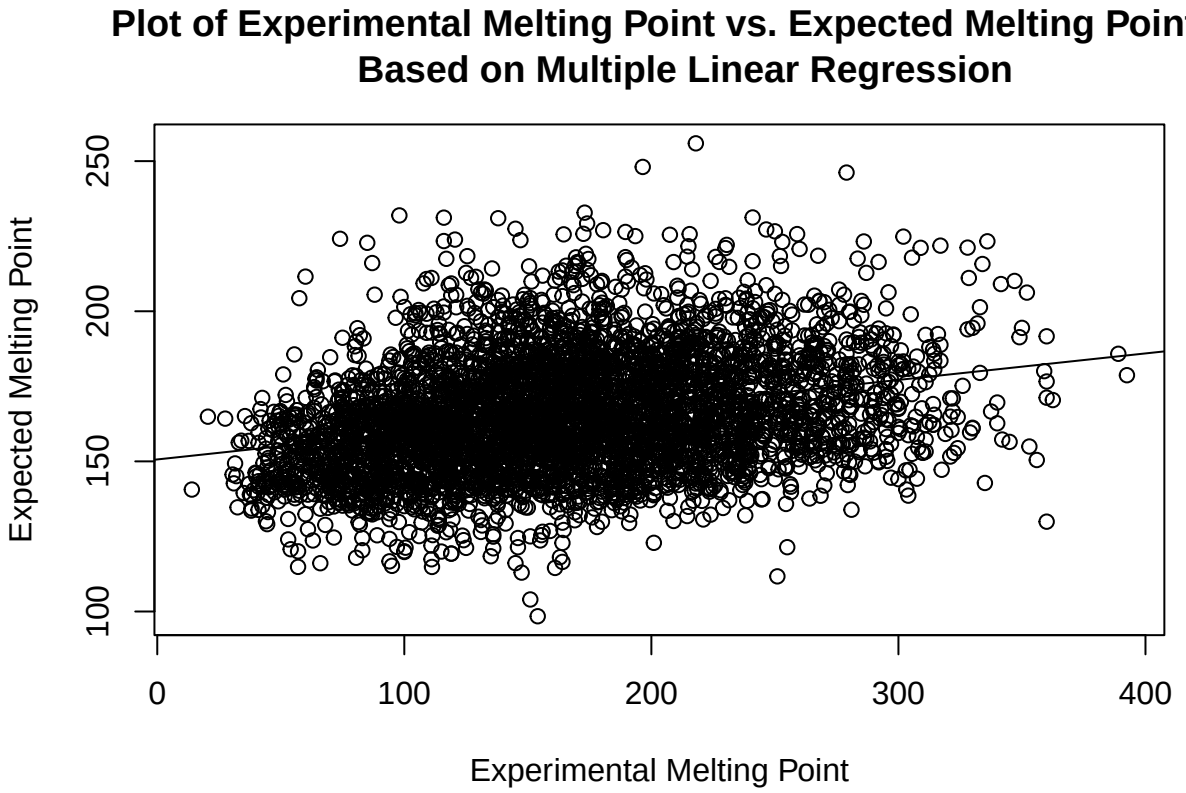



```
par(mfrow = c(1, 1))
```

Plotting Predicted Melting Point vs. Experimental Melting Point Based on Multiple Regression

After we created scatter plots of Predicted Melting Point vs. Experimental Melting Point based on each seperate predictor, we created a multiple regression using all three predictors together this time. After finding the multiple regression, we used the fitted.values command again to find the predicted values for the melting point, then graphed it against the experimental melting point.

```
multiPlot <- lm(meltpoint$melting.point~meltpoint$formal.charge
                +meltpoint$volume+meltpoint$refractivity)
meltmultiFit <- multiPlot$fitted.values
plot(meltmultiFit~meltpoint$melting.point,
main = "Plot of Experimental Melting Point vs. Expected Melting Point
     Based on Multiple Linear Regression", xlab = "Experimental Melting Point",
ylab = "Expected Melting Point")
fitline <- lm(meltmultiFit~meltpoint$melting.point)
abline(fitline)
```



BIC Model Analyses Finding Causation between Molar Mass and Melting Point

Lastly, we decided to find how much causation there was between molar mass and melting point. This means we were trying to find if having a higher molar mass causes you to have a specific melting point. To find this, we ran the BIC value for molar mass and 1 to find our first BIC value. Then ran it finding the causation for molar mass and melting point. To find if there is causatoin between the two, the second BIC value obtained should be 10 points lower than the first. In this case it was over 100 points meaning there is a lot of evidence that there is causation between the two.

```
BIC(lm(meltpoint$molar.mass~1))
```

```
## [1] 53775.5
```

```
BIC(lm(meltpoint$molar.mass~meltpoint$melting.point))
```

```
## [1] 53448.88
```

Conclusion

Overall, there were a lot of simiarlities obtained between the article and our EDA. In the article, there is a plot of the predicted melting point vs experimental melting point and a few of our graphs actaully match up very close to what they got. We were also able to observe that there is a very high chance that there is causation between molar mass and melting point.