A2

Yue Han

```
library(tidyverse)
library(ggplot2)
```

Question 1

```
housing <- read.csv("housingprice.csv")
head(housing)
```

```
##           id            date   price bedrooms bathrooms sqft_living
sqft_lot
## 1 7129300520 20141013T000000  221900        3      1.00        1180
5650
## 2 6414100192 20141209T000000  538000        3      2.25        2570
7242
## 3 5631500400 20150225T000000  180000        2      1.00         770
10000
## 4 2487200875 20141209T000000  604000        4      3.00        1960
5000
## 5 1954400510 20150218T000000  510000        3      2.00        1680
8080
## 6 7237550310 20140512T000000 1225000        4      4.50        5420
101930
##   floors waterfront view condition grade sqft_above sqft_basement yr_built
## 1      1          0    0         3     7       1180             0     1955
## 2      2          0    0         3     7       2170           400     1951
## 3      1          0    0         3     6        770             0     1933
## 4      1          0    0         5     7       1050           910     1965
## 5      1          0    0         3     8       1680             0     1987
## 6      1          0    0         3    11       3890          1530     2001
##   yr_renovated zipcode     lat     long sqft_living15 sqft_lot15
## 1            0   98178 47.5112 -122.257          1340       5650
## 2         1991   98125 47.7210 -122.319          1690       7639
## 3            0   98028 47.7379 -122.233          2720       8062
## 4            0   98136 47.5208 -122.393          1360       5000
## 5            0   98074 47.6168 -122.045          1800       7503
## 6            0   98053 47.6561 -122.005          4760     101930
```

1 a.)

```
housing$zipcode <- as.factor(housing$zipcode)
head(housing)
```

```
##           id            date   price bedrooms bathrooms sqft_living
sqft_lot
## 1 7129300520 20141013T000000  221900        3      1.00        1180
5650
```

```
## 2 6414100192 20141209T000000  538000        3      2.25        2570
7242
## 3 5631500400 20150225T000000  180000        2      1.00         770
10000
## 4 2487200875 20141209T000000  604000        4      3.00        1960
5000
## 5 1954400510 20150218T000000  510000        3      2.00        1680
8080
## 6 7237550310 20140512T000000 1225000        4      4.50        5420
101930
##   floors waterfront view condition grade sqft_above sqft_basement yr_built
## 1      1          0    0         3     7       1180             0     1955
## 2      2          0    0         3     7       2170           400     1951
## 3      1          0    0         3     6        770             0     1933
## 4      1          0    0         5     7       1050           910     1965
## 5      1          0    0         3     8       1680             0     1987
## 6      1          0    0         3    11       3890          1530     2001
##   yr_renovated zipcode     lat     long sqft_living15 sqft_lot15
## 1            0   98178 47.5112 -122.257          1340       5650
## 2         1991   98125 47.7210 -122.319          1690       7639
## 3            0   98028 47.7379 -122.233          2720       8062
## 4            0   98136 47.5208 -122.393          1360       5000
## 5            0   98074 47.6168 -122.045          1800       7503
## 6            0   98053 47.6561 -122.005          4760     101930
```

```
p_mean <- aggregate(housing$price, list(housing$zipcode), mean)
p_mean = p_mean[order(-p_mean$x),]
head(p_mean)
```

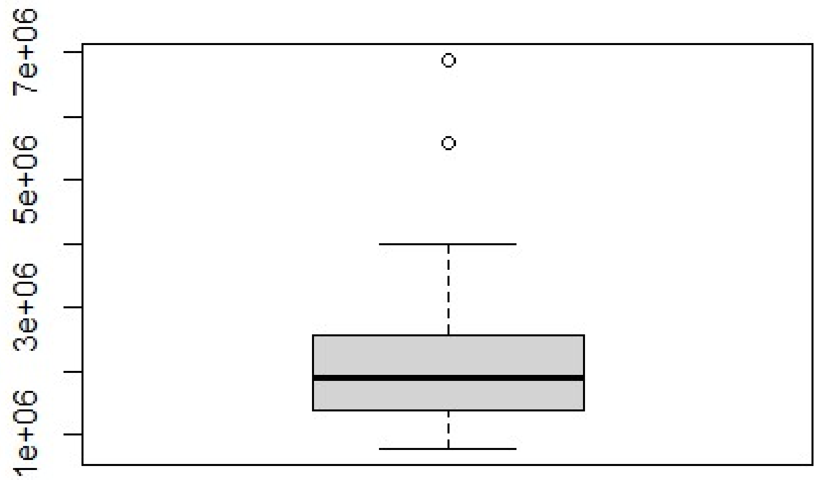
```
##    Group.1         x
## 25   98039 2160606.6
## 4    98004 1355927.1
## 26   98040 1194230.0
## 49   98112 1095499.4
## 42   98102  901258.2
## 48   98109  879623.6
```

The most expensive average price zipcodes are 98039, 98004 and 98040

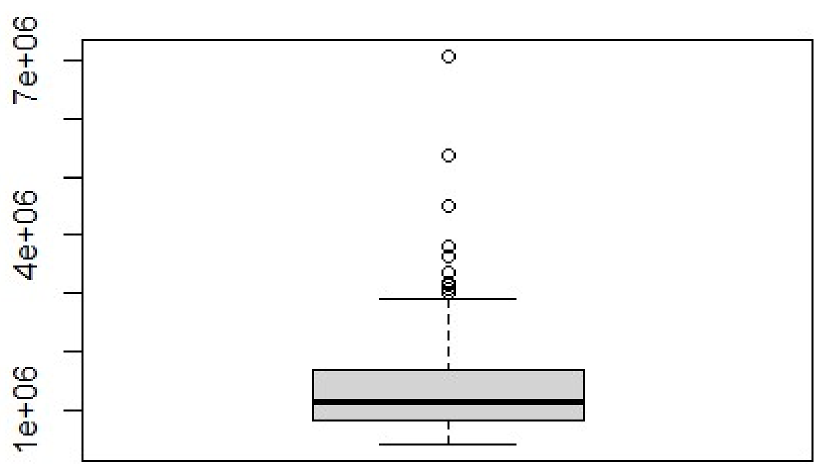
```
top1 = housing[housing$zipcode == p_mean$Group.1[1],]
top2 = housing[housing$zipcode == p_mean$Group.1[2],]
top3 = housing[housing$zipcode == p_mean$Group.1[3],]
p_mean[1:3,] #three highest average price zipcodes in order from most to
third most and their corresponding average prices
```

```
##    Group.1       x
## 25   98039 2160607
## 4    98004 1355927
## 26   98040 1194230
```

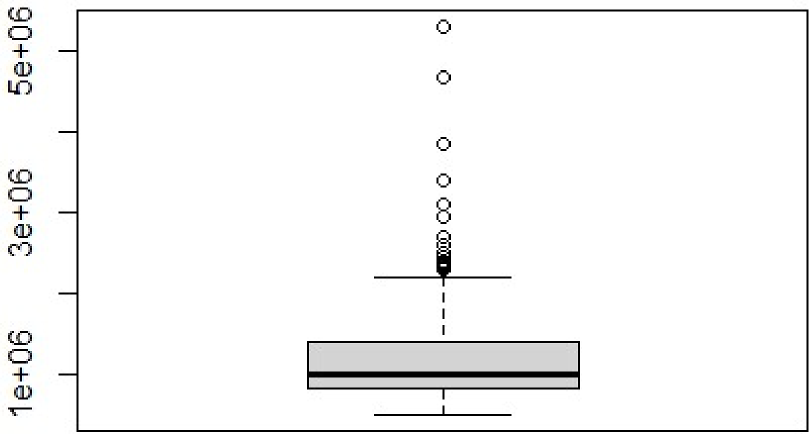
```
boxplot(top1$price) #most expensive average price for zipcode
```




```
boxplot(top2$price) #2nd most expensive average price for zipcode
```



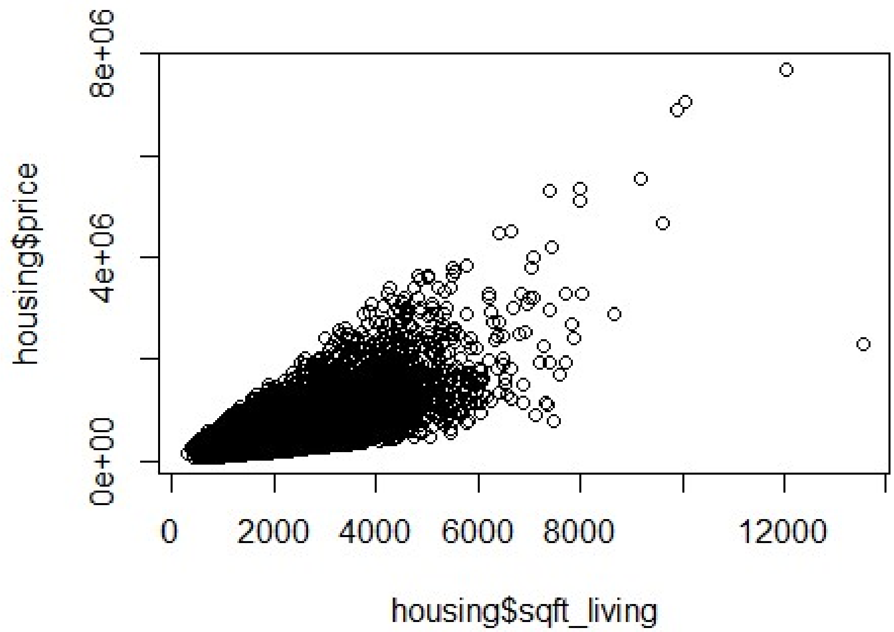

```
boxplot(top3$price) #3rd most expensive average price for zipcode
```



1 b.)

From the plot we can see that as sqft_living increases so does the price

```
plot(housing$sqft_living, housing$price)
```



1 c.)

```
train <- read.csv("train.data.csv")
head(train)
```

```
##   X         id            date   price bedrooms bathrooms sqft_living
sqft_lot
## 1 2 6414100192 20141209T000000  538000        3      2.25        2570
7242
## 2 4 2487200875 20141209T000000  604000        4      3.00        1960
5000
## 3 5 1954400510 20150218T000000  510000        3      2.00        1680
8080
## 4 6 7237550310 20140512T000000 1225000        4      4.50        5420
101930
## 5 7 1321400060 20140627T000000  257500        3      2.25        1715
6819
## 6 8 2008000270 20150115T000000  291850        3      1.50        1060
9711
##   floors waterfront view condition grade sqft_above sqft_basement yr_built
## 1      2          0    0         3     7       2170           400     1951
## 2      1          0    0         5     7       1050           910     1965
## 3      1          0    0         3     8       1680             0     1987
## 4      1          0    0         3    11       3890          1530     2001
## 5      2          0    0         3     7       1715             0     1995
## 6      1          0    0         3     7       1060             0     1963
##   yr_renovated zipcode     lat     long sqft_living15 sqft_lot15
## 1         1991   98125 47.7210 -122.319          1690       7639
```

```
## 2             0   98136 47.5208 -122.393          1360       5000
## 3             0   98074 47.6168 -122.045          1800       7503
## 4             0   98053 47.6561 -122.005          4760     101930
## 5             0   98003 47.3097 -122.327          2238       6819
## 6             0   98198 47.4095 -122.315          1650       9711
```

```
test <- read.csv("test.data.csv")
head(test)
```

```
##    X         id            date  price bedrooms bathrooms sqft_living
sqft_lot
## 1  1 7129300520 20141013T000000 221900        3       1.0        1180
5650
## 2  3 5631500400 20150225T000000 180000        2       1.0         770
10000
## 3 11 1736800520 20150403T000000 662500        3       2.5        3560
9796
## 4 18 6865200140 20140529T000000 485000        4       1.0        1600
4300
## 5 20 7983200060 20150424T000000 230000        3       1.0        1250
9774
## 6 24 8091400200 20140516T000000 252700        2       1.5        1070
9643
##   floors waterfront view condition grade sqft_above sqft_basement yr_built
## 1    1.0          0    0         3     7       1180             0     1955
## 2    1.0          0    0         3     6        770             0     1933
## 3    1.0          0    0         3     8       1860          1700     1965
## 4    1.5          0    0         4     7       1600             0     1916
## 5    1.0          0    0         4     7       1250             0     1969
## 6    1.0          0    0         3     7       1070             0     1985
##   yr_renovated zipcode     lat     long sqft_living15 sqft_lot15
## 1            0   98178 47.5112 -122.257          1340       5650
## 2            0   98028 47.7379 -122.233          2720       8062
## 3            0   98007 47.6007 -122.145          2210       8925
## 4            0   98103 47.6648 -122.343          1610       4300
## 5            0   98003 47.3343 -122.306          1280       8850
## 6            0   98030 47.3533 -122.166          1220       8386
```

For training data The $R^2 = 0.5101$

```
train_price = lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot, data
= train)
summary(train_price)
```

```
## 
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot,
##     data = train)
## 
## Residuals:
##       Min       1Q   Median       3Q      Max
```

```
## -1571803  -143678   -22595   103133  4141210
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  8.083e+04  8.208e+03   9.848  < 2e-16 ***
## bedrooms    -5.930e+04  2.753e+03 -21.537  < 2e-16 ***
## bathrooms    3.682e+03  4.178e+03   0.881    0.378
## sqft_living  3.167e+02  3.750e+00  84.442  < 2e-16 ***
## sqft_lot    -4.267e-01  5.504e-02  -7.753 9.52e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 257200 on 15124 degrees of freedom
## Multiple R-squared:  0.5101, Adjusted R-squared:   0.51
## F-statistic:  3937 on 4 and 15124 DF,  p-value: < 2.2e-16
```

For testing data $R^2 = 0.5054$

```
test_price = lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot, data =
test)
summary(test_price)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot,
##     data = test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1139078  -144975   -22073   101977  4137692
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.539e+04  1.287e+04   5.859 4.88e-09 ***
## bedrooms    -5.997e+04  4.424e+03 -13.555  < 2e-16 ***
## bathrooms    1.228e+04  6.473e+03   1.897   0.0579 .
## sqft_living  3.093e+02  5.703e+00  54.231  < 2e-16 ***
## sqft_lot    -2.990e-01  6.947e-02  -4.304 1.70e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 257700 on 6479 degrees of freedom
## Multiple R-squared:  0.5054, Adjusted R-squared:  0.5051
## F-statistic:  1655 on 4 and 6479 DF,  p-value: < 2.2e-16
```

d.) Training data we have $R^2 = 0.5163$

```
train_price_z = lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
zipcode, data = train)
summary(train_price_z)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     zipcode, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1638518  -141274   -22673   101293  4074728
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.460e+07  3.933e+06 -13.883  < 2e-16 ***
## bedrooms    -5.760e+04  2.739e+03 -21.034  < 2e-16 ***
## bathrooms    8.631e+03  4.167e+03   2.071   0.0383 *
## sqft_living  3.185e+02  3.729e+00  85.420  < 2e-16 ***
## sqft_lot    -3.443e-01  5.501e-02  -6.259 3.98e-10 ***
## zipcode      5.573e+02  4.008e+01  13.904  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 255600 on 15123 degrees of freedom
## Multiple R-squared:  0.5163, Adjusted R-squared:  0.5161
## F-statistic:  3228 on 5 and 15123 DF,  p-value: < 2.2e-16
```

For testing data we have $R^2 = 0.5124$

```
test_price_z = lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
zipcode, data = test)
summary(test_price_z)

##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     zipcode, data = test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1129578  -141251   -21264    99654  4150457
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -5.760e+07  5.989e+06  -9.617  < 2e-16 ***
## bedrooms    -5.823e+04  4.397e+03 -13.243  < 2e-16 ***
## bathrooms    1.709e+04  6.447e+03   2.651  0.00804 **
## sqft_living  3.111e+02  5.666e+00  54.908  < 2e-16 ***
## sqft_lot    -2.263e-01  6.939e-02  -3.262  0.00111 **
## zipcode      5.878e+02  6.104e+01   9.630  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 255900 on 6478 degrees of freedom
## Multiple R-squared:  0.5124, Adjusted R-squared:  0.5121
## F-statistic:  1362 on 5 and 6478 DF,  p-value: < 2.2e-16
```

e.)

The predicted price is $15642273

```
fancy <- read.csv("fancyhouse.csv")
head(fancy)

##   X bedrooms bathrooms sqft_living sqft_lot floors zipcode condition grade
## 1 1        8        25       50000   225000      4   98039        10    10
##   waterfront view sqft_above sqft_basement yr_built yr_renovated      lat
## 1          1    4      37500         12500     1994         2010 47.62761
##        long sqft_living15 sqft_lot15
## 1 -122.2421          5000      40000

predict(train_price_z, fancy)

##        1
## 15642273
```

Below we see the most expensive houses in the data we have

```
head(housing[order(-housing$price),])

##              id            date   price bedrooms bathrooms sqft_living
sqft_lot
## 7253 6762700020 20141013T000000 7700000        6      8.00       12050
27600
## 3915 9808700762 20140611T000000 7062500        5      4.50       10040
37325
## 9255 9208900037 20140919T000000 6885000        6      7.75        9890
31374
## 4412 2470100110 20140804T000000 5570000        5      5.75        9200
35069
## 1449 8907500070 20150413T000000 5350000        5      5.00        8000
23985
## 1316 7558700030 20150413T000000 5300000        6      6.00        7390
24829
##      floors waterfront view condition grade sqft_above sqft_basement
yr_built
## 7253    2.5          0    3         4    13       8570          3480
1910
## 3915    2.0          1    2         3    11       7680          2360
1940
## 9255    2.0          0    4         3    13       8860          1030
2001
## 4412    2.0          0    0         3    13       6200          3000
2001
## 1449    2.0          0    4         3    12       6720          1280
```

```
2009
## 1316    2.0          1    4         4    12       5000          2390
1991
##      yr_renovated zipcode      lat     long sqft_living15 sqft_lot15
## 7253          1987   98102 47.6298 -122.323          3940       8800
## 3915          2001   98004 47.6500 -122.214          3930      25449
## 9255             0   98039 47.6305 -122.240          4540      42730
## 4412             0   98039 47.6289 -122.233          3560      24345
## 1449             0   98004 47.6232 -122.220          4600      21750
## 1316             0   98040 47.5631 -122.210          4320      24619
```

The most expensive house is just 7700000. The most expensive house with the same zipcode is $6885000. Bill gate's house has more than 5 times the sqft_living, more than 7 times the sqft_lot and more than 3 times the amount of bathrooms. For it being just 2.271935 times more expensive than the 6885000 dollar home seems unreasonable. Especially since we know from the scatter plot that more sqft_living means more pricey and that the zipcode has the highest price.

f.) $\hat{\beta}_1 = argmin_{\beta \in R^{d+1}} | \quad |Y - X_1\beta| \quad |_2^2 = argmin_{\beta \in R^{d+1}} \left(\sqrt{\sum_{i=1}^{n} (y_i - \beta W_i - \beta x_{id})^2}\right)^2 = argmin_{\beta \in R^{d+1}} \sum_{i=1}^{n} (y_i - \beta W_i - \beta x_{id})^2$ Where W_i is the ith row of X

$$\hat{\beta} = argmin_{\beta \in R^{d+1}} | \quad |Y - X\beta| \quad |_2^2 = argmin_{\beta \in R^{d+1}} \left(\sqrt{\sum_{i=1}^{n} (y_i - \beta W_i)^2}\right)^2$$

$$= argmin_{\beta \in R^{d+1}} \sum_{i=1}^{n} (y_i - \beta W_i)^2$$

So we have $| \quad |Y - X\hat{\beta}| \quad |_2^2 = | \quad |Y - X\left(argmin_{\beta \in R^{d+1}} \sum_{i=1}^{n} (y_i - \beta W_i)^2\right)| \quad |_2^2 = \sum_i^n y_i - W_i\left(argmin_{\beta \in R^{d+1}} \sum_{i=1}^{n} (y_i - \beta W_i)^2\right)$

and

$$| \quad |Y - X_1\hat{\beta}_1| \quad |_2^2 = | \quad |Y - X_1\left(argmin_{\beta \in R^{d+1}} \left(\sum_{i=1}^{n} y_i - \beta W_i - \beta x_{id}\right)^2\right)| \quad |_2^2$$

$$= \sum_i^n y_i - W_i\left(argmin_{\beta \in R^{d+1}} \left(\sum_{i=1}^{n} y_i - \beta W_i - \beta x_{id}\right)^2\right) - x_{id}\left(argmin_{\beta \in R^{d+1}} \left(\sum_{i=1}^{n} y_i - \beta W_i - \beta x_{id}\right)^2\right)$$

We want the smaller of the two and we can see that by adding a covariate $||Y-X_1_1||^2_2 ||Y-X||^2_2 $ because if it is larger it will just assign a 0 to the estimated coefficient. We want the smaller one because it is the SS_{res} and minimizing SS_{res} maximizes R^2 since $SS_{tot} = SS_{res} + SS_{reg}$ and $R^2 = \frac{SS_{reg}}{SS_{tot}}$

Question 2

Below is the results for training data $R^2 = 0.5224$

```
summary(lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot + zipcode +
bedrooms * bathrooms, data = train))

##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     zipcode + bedrooms * bathrooms, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2202454  -139444   -23520   100249  3685052
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -4.920e+07  3.928e+06 -12.526  < 2e-16 ***
## bedrooms           -1.216e+05  5.359e+03 -22.697  < 2e-16 ***
## bathrooms          -9.739e+04  8.694e+03 -11.203  < 2e-16 ***
## sqft_living         3.110e+02  3.745e+00  83.054  < 2e-16 ***
## sqft_lot           -3.502e-01  5.467e-02  -6.405 1.55e-10 ***
## zipcode             5.045e+02  4.001e+01  12.608  < 2e-16 ***
## bedrooms:bathrooms  3.107e+04  2.240e+03  13.871  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 254000 on 15122 degrees of freedom
## Multiple R-squared:  0.5224, Adjusted R-squared:  0.5222
## F-statistic:  2756 on 6 and 15122 DF,  p-value: < 2.2e-16
```

Below is for the testing data $R^2 = 0.517$

```
summary(lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot + zipcode +
bedrooms * bathrooms, data = test))

##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     zipcode + bedrooms * bathrooms, data = test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1272995  -138589   -21660    97298  4090954
##
## Coefficients:
```

```
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -5.317e+07  5.988e+06  -8.880  < 2e-16 ***
## bedrooms            -1.191e+05  8.916e+03 -13.356  < 2e-16 ***
## bathrooms           -8.121e+04  1.410e+04  -5.762  8.7e-09 ***
## sqft_living          3.058e+02  5.681e+00  53.837  < 2e-16 ***
## sqft_lot            -2.201e-01  6.908e-02  -3.187  0.00145 **
## zipcode              5.448e+02  6.101e+01   8.930  < 2e-16 ***
## bedrooms:bathrooms   2.880e+04  3.676e+03   7.834  5.5e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 254700 on 6477 degrees of freedom
## Multiple R-squared:  0.517,  Adjusted R-squared:  0.5166
## F-statistic:  1156 on 6 and 6477 DF,  p-value: < 2.2e-16
```

2b.) We can add condition as another feature. As we can see below it increases the R^2 to 0.5266

```
summary(lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot + zipcode +
bedrooms * bathrooms + condition, data = test))
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     zipcode + bedrooms * bathrooms + condition, data = test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1278989  -138037   -22286    99911  4107203
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -5.323e+07  5.929e+06  -8.979  < 2e-16 ***
## bedrooms            -1.253e+05  8.844e+03 -14.169  < 2e-16 ***
## bathrooms           -7.453e+04  1.397e+04  -5.336  9.8e-08 ***
## sqft_living          3.041e+02  5.626e+00  54.056  < 2e-16 ***
## sqft_lot            -2.202e-01  6.839e-02  -3.220  0.00129 **
## zipcode              5.435e+02  6.040e+01   8.998  < 2e-16 ***
## condition            5.617e+04  4.895e+03  11.476  < 2e-16 ***
## bedrooms:bathrooms   2.976e+04  3.640e+03   8.176  3.5e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 252200 on 6476 degrees of freedom
## Multiple R-squared:  0.5266, Adjusted R-squared:  0.5261
## F-statistic:  1029 on 7 and 6476 DF,  p-value: < 2.2e-16
```

2c.) Below are the results for training data $R^2 = 0.5423$

```
summary(lm(price ~ poly(bedrooms,2) + poly(bathrooms,3) + sqft_living +
sqft_lot + zipcode, data = train))
```

```
##
## Call:
## lm(formula = price ~ poly(bedrooms, 2) + poly(bathrooms, 3) +
##     sqft_living + sqft_lot + zipcode, data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -3312253  -136245   -26067    98812  2733696
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -3.965e+07  3.865e+06 -10.260  < 2e-16 ***
## poly(bedrooms, 2)1  -6.137e+06  3.119e+05 -19.672  < 2e-16 ***
## poly(bedrooms, 2)2   1.803e+06  2.556e+05   7.054 1.82e-12 ***
## poly(bathrooms, 3)1  2.137e+06  3.877e+05   5.512 3.61e-08 ***
## poly(bathrooms, 3)2  7.116e+06  2.576e+05  27.621  < 2e-16 ***
## poly(bathrooms, 3)3  2.093e+05  2.492e+05   0.840    0.401
## sqft_living          3.011e+02  3.736e+00  80.610  < 2e-16 ***
## sqft_lot            -4.209e-01  5.359e-02  -7.855 4.27e-15 ***
## zipcode              4.035e+02  3.940e+01  10.241  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 248600 on 15120 degrees of freedom
## Multiple R-squared:  0.5423, Adjusted R-squared:  0.5421
## F-statistic:  2240 on 8 and 15120 DF,  p-value: < 2.2e-16
```

Below are the results for testing data $R^2 = 0.5296$

```
summary(lm(price ~ poly(bedrooms,2) + poly(bathrooms,3) + sqft_living +
sqft_lot + zipcode, data = test))
```

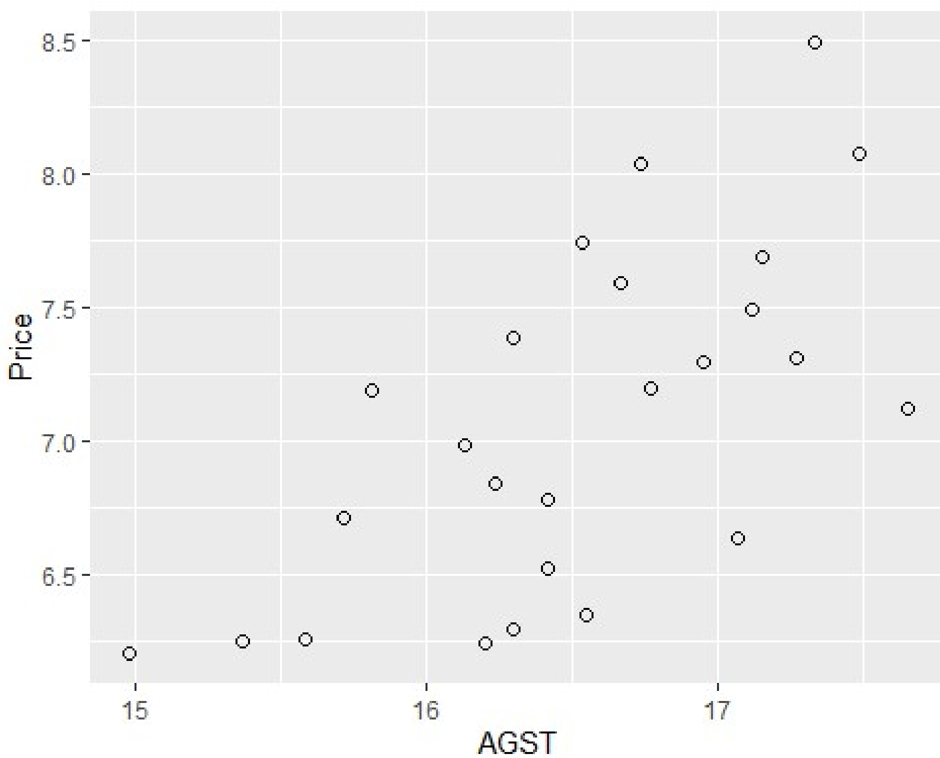
```
##
## Call:
## lm(formula = price ~ poly(bedrooms, 2) + poly(bathrooms, 3) +
##     sqft_living + sqft_lot + zipcode, data = test)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1920754  -134506   -25457    96332  4012647
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -4.761e+07  5.953e+06  -7.998 1.49e-15 ***
## poly(bedrooms, 2)1  -3.842e+06  3.192e+05 -12.038  < 2e-16 ***
## poly(bedrooms, 2)2   1.105e+05  2.672e+05   0.414 0.679154
## poly(bathrooms, 3)1  1.665e+06  3.924e+05   4.244 2.23e-05 ***
## poly(bathrooms, 3)2  3.894e+06  2.704e+05  14.401  < 2e-16 ***
## poly(bathrooms, 3)3  4.810e+05  2.530e+05   1.901 0.057285 .
## sqft_living          2.947e+02  5.692e+00  51.771  < 2e-16 ***
```

```
## sqft_lot            -2.273e-01  6.820e-02  -3.333 0.000863 ***
## zipcode              4.847e+02  6.069e+01   7.987 1.63e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 251400 on 6475 degrees of freedom
## Multiple R-squared:  0.5296, Adjusted R-squared:  0.529
## F-statistic: 911.1 on 8 and 6475 DF,  p-value: < 2.2e-16
```

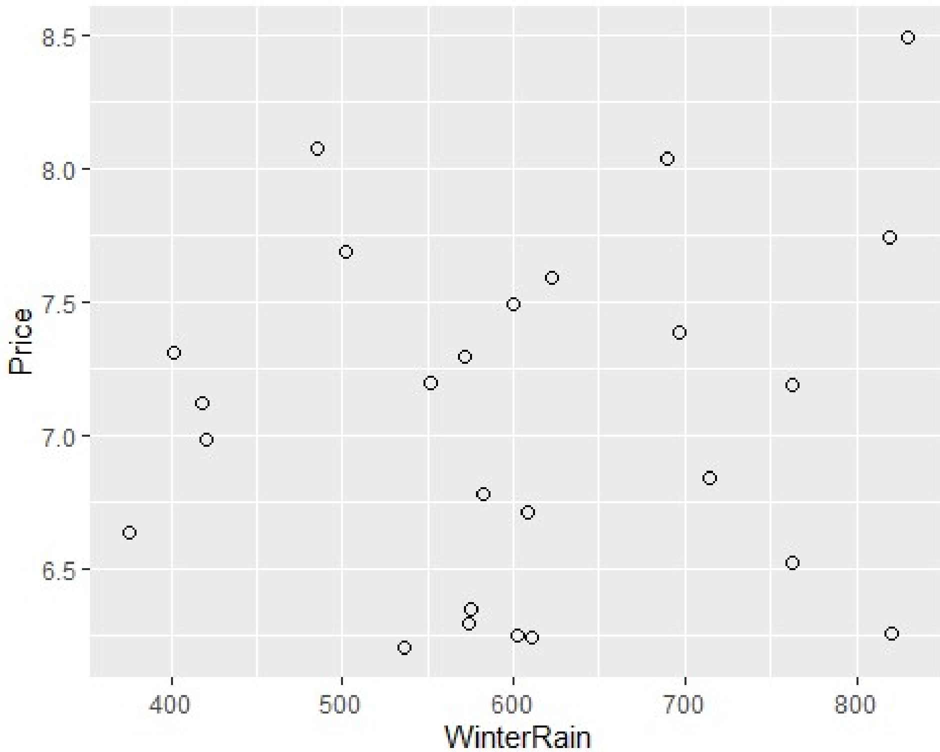
Question 3

3 Part 1.)

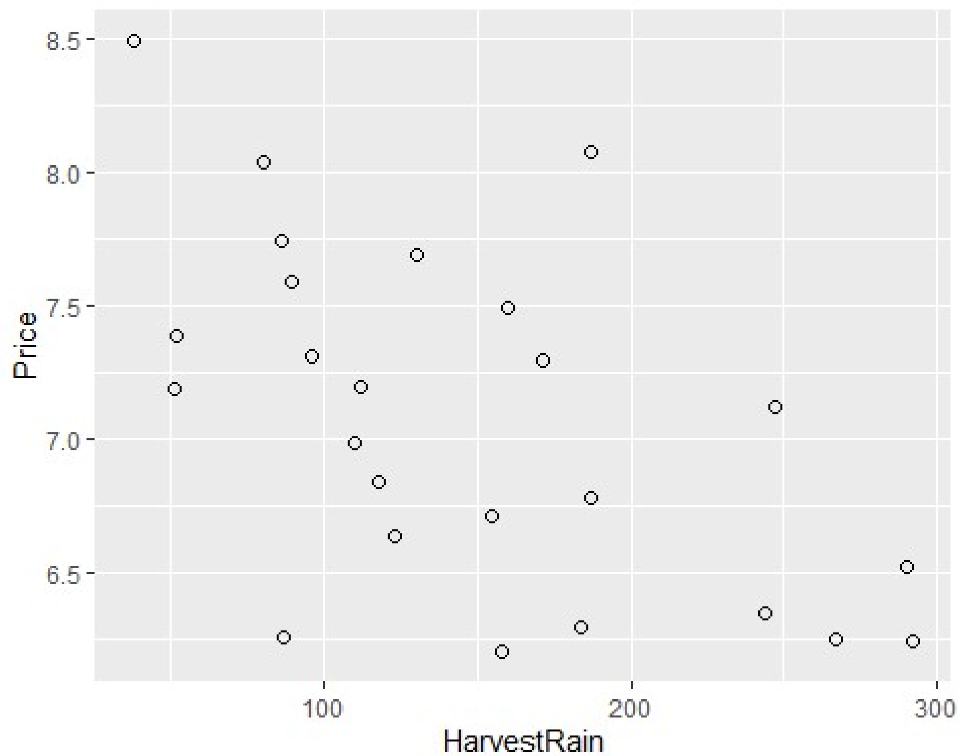
```
wine <- read.csv("wine.csv")
head(wine)

##   Year  Price WinterRain    AGST HarvestRain Age FrancePop
## 1 1952 7.4950        600 17.1167         160  31  43183.57
## 2 1953 8.0393        690 16.7333          80  30  43495.03
## 3 1955 7.6858        502 17.1500         130  28  44217.86
## 4 1957 6.9845        420 16.1333         110  26  45152.25
## 5 1958 6.7772        582 16.4167         187  25  45653.81
## 6 1959 8.0757        485 17.4833         187  24  46128.64

ggplot(wine, aes(x =AGST, y=Price)) +
  geom_point(size=2, shape=1)
```



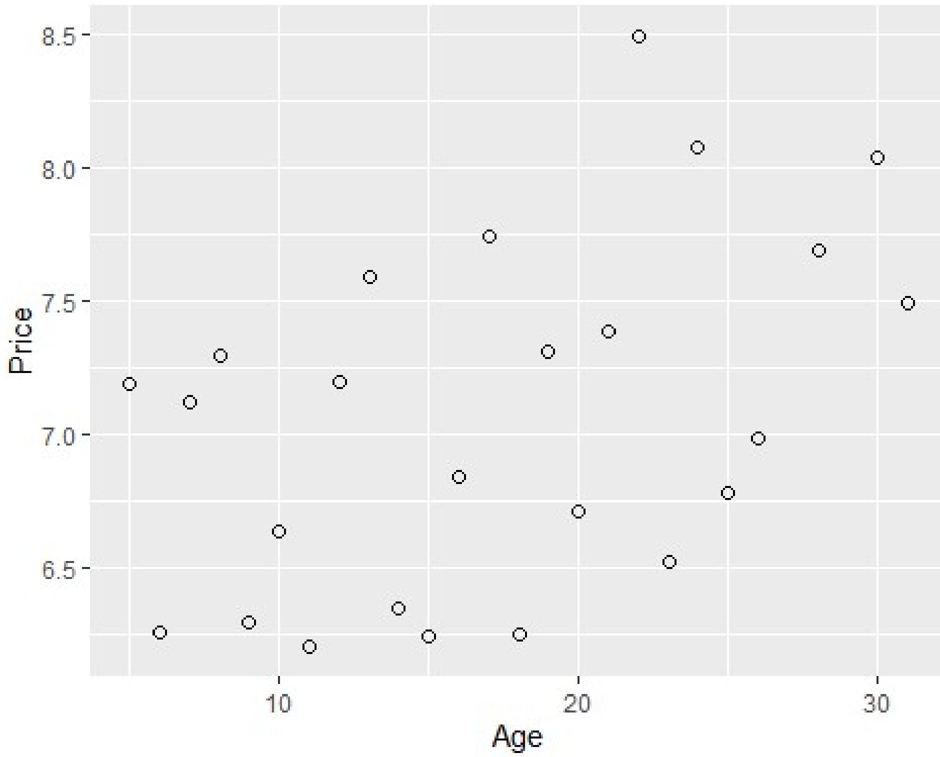
```
ggplot(wine, aes(x =WinterRain, y=Price)) +
  geom_point(size=2, shape=1)
```




```
ggplot(wine, aes(x =HarvestRain, y=Price)) +
  geom_point(size=2, shape=1)
```




```
ggplot(wine, aes(x =Age, y=Price)) +
  geom_point(size=2, shape=1)
```



The variable most correlated with price based on the scatterplots is AGST since compared to the other graphs, there is a clear increase of price as AGST increases.

Pearson's correlation for AGST is the highest. It is 0.6595629 so about 0.66. Therefore my observation is justified.

```
cor(wine$Price, wine$AGST)

## [1] 0.6595629

cor(wine$Price, wine$WinterRain)

## [1] 0.1366505

cor(wine$Price, wine$HarvestRain)

## [1] -0.5633219

cor(wine$Price, wine$Age)

## [1] 0.4477679
```

Part 2 $R^2 = 0.435$ Coefficient for intercept is -3.4178 and coefficient for AGST is 0.6351

```
summary(lm(Price~AGST, data = wine))

##
## Call:
## lm(formula = Price ~ AGST, data = wine)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.78450 -0.23882 -0.03727  0.38992  0.90318
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -3.4178     2.4935  -1.371 0.183710
## AGST          0.6351     0.1509   4.208 0.000335 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4993 on 23 degrees of freedom
## Multiple R-squared:  0.435,  Adjusted R-squared:  0.4105
## F-statistic: 17.71 on 1 and 23 DF,  p-value: 0.000335
```

3 Part 3

```
wine_test <- read.csv("winetest.csv")
head(wine_test)
```

```
##   Year  Price WinterRain    AGST HarvestRain Age FrancePop
## 1 1979 6.9541        717 16.1667         122   4  54835.83
## 2 1980 6.4979        578 16.0000          74   3  55110.24
```

For training data with AGST, HarvestRain as covariates we have $R^2 = 0.7074$

```
summary(lm(Price~AGST + HarvestRain , data = wine))
```

```
##
## Call:
## lm(formula = Price ~ AGST + HarvestRain, data = wine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.88321 -0.19600  0.06178  0.15379  0.59722
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.20265    1.85443  -1.188 0.247585
## AGST         0.60262    0.11128   5.415 1.94e-05 ***
## HarvestRain -0.00457    0.00101  -4.525 0.000167 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3674 on 22 degrees of freedom
## Multiple R-squared:  0.7074, Adjusted R-squared:  0.6808
## F-statistic: 26.59 on 2 and 22 DF,  p-value: 1.347e-06
```

For testing data with AGST and HarvestRain as covariates we have $R^2 = -2.503339$

```
fit = lm(Price~ AGST  + HarvestRain, data = wine)
pred = predict(fit, wine_test)
1 - sum(( pred-wine_test$Price )^2) / sum(( mean( wine_test$Price )-
wine_test$Price )^2)

## [1] -2.503339
```

For training data with AGST, Age and HarvestRain as covariates we have $R^2 = 0.79$

```
summary(lm(Price~AGST + Age + HarvestRain, data = wine))

##
## Call:
## lm(formula = Price ~ AGST + Age + HarvestRain, data = wine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.66258 -0.22953 -0.00268  0.27236  0.49391
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -1.4778196  1.6274142  -0.908  0.37414
## AGST         0.5322922  0.0995343   5.348 2.65e-05 ***
## Age          0.0250875  0.0087249   2.875  0.00905 **
## HarvestRain -0.0045386  0.0008757  -5.183 3.90e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3186 on 21 degrees of freedom
## Multiple R-squared:   0.79,  Adjusted R-squared:   0.76
## F-statistic: 26.34 on 3 and 21 DF,  p-value: 2.596e-07
```

For testing data with AGST Age and HarvestRain as covariates we have $R^2 = -0.5080824$

```
fit = lm(Price~ AGST + Age + HarvestRain, data = wine)
pred = predict(fit, wine_test)
1 - sum(( pred-wine_test$Price)^2)/sum(( mean(wine_test$Price)-
wine_test$Price )^2)

## [1] -0.5080824
```

For training data with AGST HarvestRain Age and WinterRain as covariates we have $R^2 = 0.75374$

```
fit = lm(Price~AGST + HarvestRain + Age + WinterRain, data = wine)
summary(fit)

##
## Call:
## lm(formula = Price ~ AGST + HarvestRain + Age + WinterRain, data = wine)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.45470 -0.24273  0.00752  0.19773  0.53637
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.4299802  1.7658975  -1.942 0.066311 .
## AGST         0.6072093  0.0987022   6.152  5.2e-06 ***
## HarvestRain -0.0039715  0.0008538  -4.652 0.000154 ***
## Age          0.0239308  0.0080969   2.956 0.007819 **
## WinterRain   0.0010755  0.0005073   2.120 0.046694 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.295 on 20 degrees of freedom
## Multiple R-squared:  0.8286, Adjusted R-squared:  0.7943
## F-statistic: 24.17 on 4 and 20 DF,  p-value: 2.036e-07
```

For Testing data we have $R^2 = 0.3343905$

```
pred = predict(fit, wine_test)
1 - sum(( pred-wine_test$Price )^2)/ sum(( mean( wine_test$Price )-
wine_test$Price )^2)

## [1] 0.3343905
```

For training data with AGST Age HarvestRain WinterRain FrancePOp as covariates we have $R^2 = 0.8294$

```
fit = lm(Price~AGST + Age +HarvestRain + WinterRain + FrancePop, data = wine)
summary(fit)

##
## Call:
## lm(formula = Price ~ AGST + Age + HarvestRain + WinterRain +
##     FrancePop, data = wine)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.48179 -0.24662 -0.00726  0.22012  0.51987
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -4.504e-01  1.019e+01  -0.044 0.965202
## AGST         6.012e-01  1.030e-01   5.836 1.27e-05 ***
## Age          5.847e-04  7.900e-02   0.007 0.994172
## HarvestRain -3.958e-03  8.751e-04  -4.523 0.000233 ***
## WinterRain   1.043e-03  5.310e-04   1.963 0.064416 .
## FrancePop   -4.953e-05  1.667e-04  -0.297 0.769578
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.3019 on 19 degrees of freedom
## Multiple R-squared:  0.8294, Adjusted R-squared:  0.7845
## F-statistic: 18.47 on 5 and 19 DF,  p-value: 1.044e-06
```

For Testing data we have $R^2 = 0.2120672$

```
pred = predict(fit, wine_test)
1 - sum((pred - wine_test$Price)^2)/ sum(( mean( wine_test$Price) -
wine_test$Price )^2)

## [1] 0.2120672
```

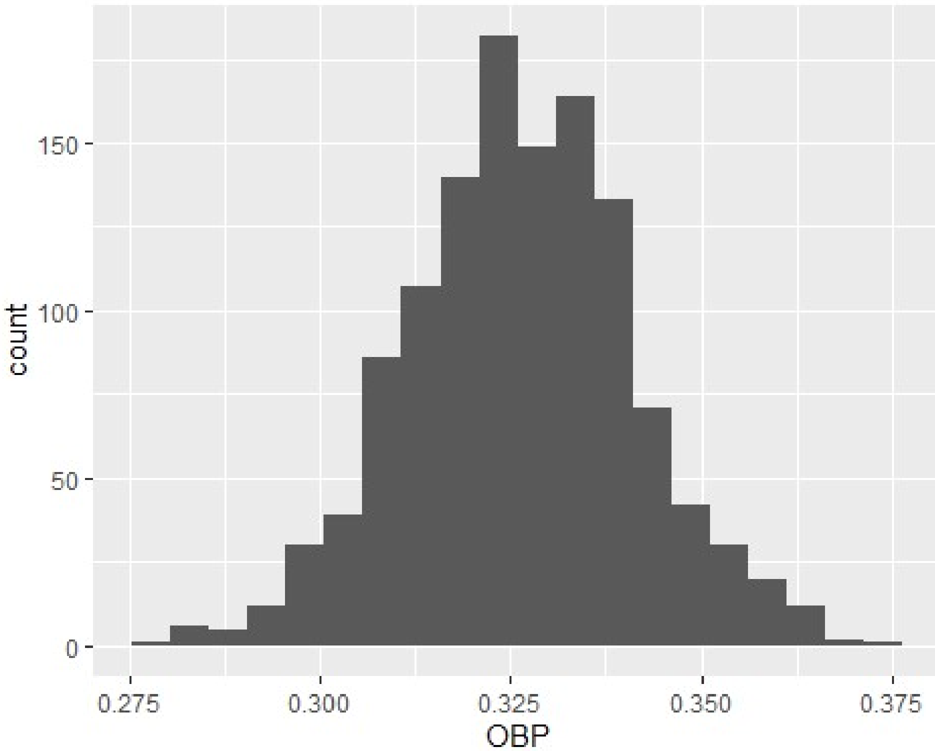
Based on what we did above, the best R^2 for testing data is 0.3343905 so we use that model. That model is the one with AGST, HarvestRain, Age, WinterRain ### Question 4

4 Part 1

```
baseball <- read.csv("baseball.csv")
head(baseball)

##   Team League Year  RS  RA  W   OBP   SLG    BA Playoffs RankSeason
## 1  ARI     NL 2012 734 688 81 0.328 0.418 0.259        0         NA
## 2  ATL     NL 2012 700 600 94 0.320 0.389 0.247        1          4
## 3  BAL     AL 2012 712 705 93 0.311 0.417 0.247        1          5
## 4  BOS     AL 2012 734 806 69 0.315 0.415 0.260        0         NA
## 5  CHC     NL 2012 613 759 61 0.302 0.378 0.240        0         NA
## 6  CHW     AL 2012 748 676 85 0.318 0.422 0.255        0         NA
##   RankPlayoffs   G  OOBP  OSLG
## 1           NA 162 0.317 0.415
## 2            5 162 0.306 0.378
## 3            4 162 0.315 0.403
## 4           NA 162 0.331 0.428
## 5           NA 162 0.335 0.424
## 6           NA 162 0.319 0.405
```

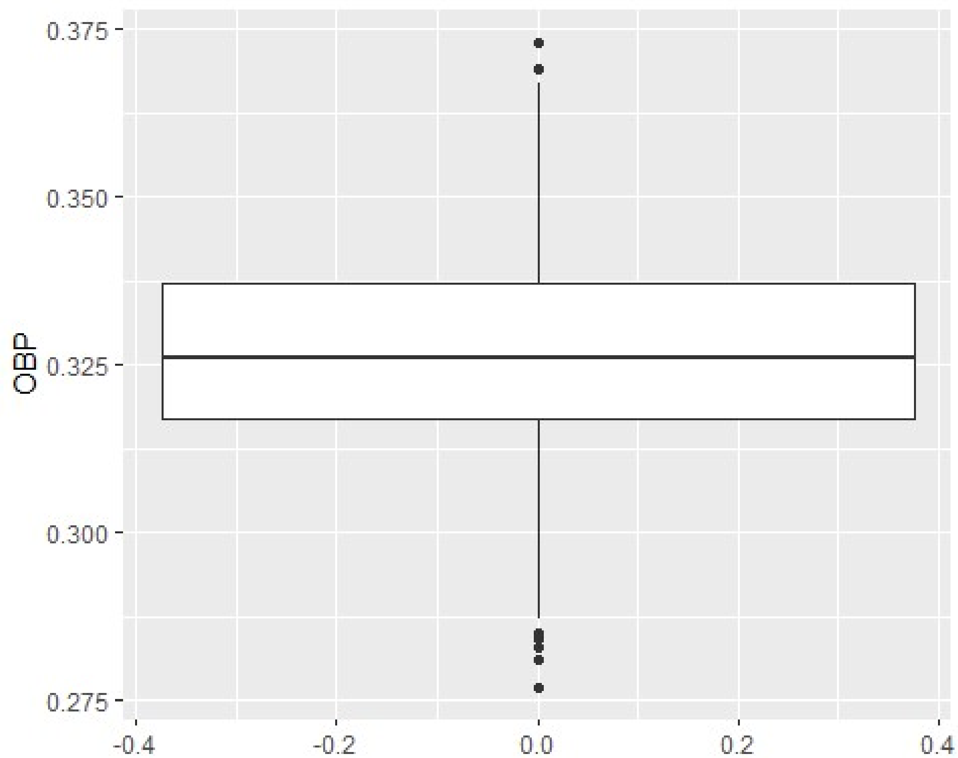
Histogram for OBP (on-base percentage) has a non-skewed distribuiton as the distribution is centralized and no side overpowers the other

```
ggplot(baseball, aes(x=OBP),) + geom_histogram(bins =20)
```



Boxplot for OBP (on-base percentage) median is centralized and the whiskers are equidistant from the median

```
ggplot(baseball, aes(y=OBP),) + geom_boxplot()
```

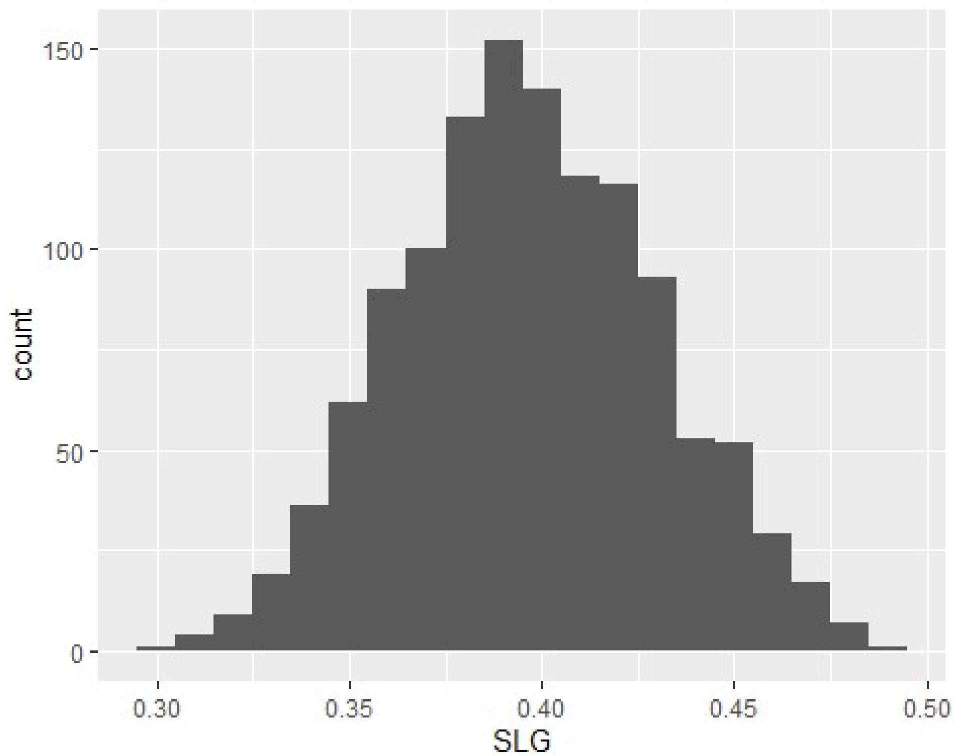


Mean for OBP (on-base percentage) is 0.3263312 and median is 0.326 which is around the same so the distribution is not skewed

```
mean(baseball$OBP)

## [1] 0.3263312

median(baseball$OBP)

## [1] 0.326
```

Histogram for SLG (slugging percentage) has non-skewed distribution as the distribution is centralized and no side overpowers the other.

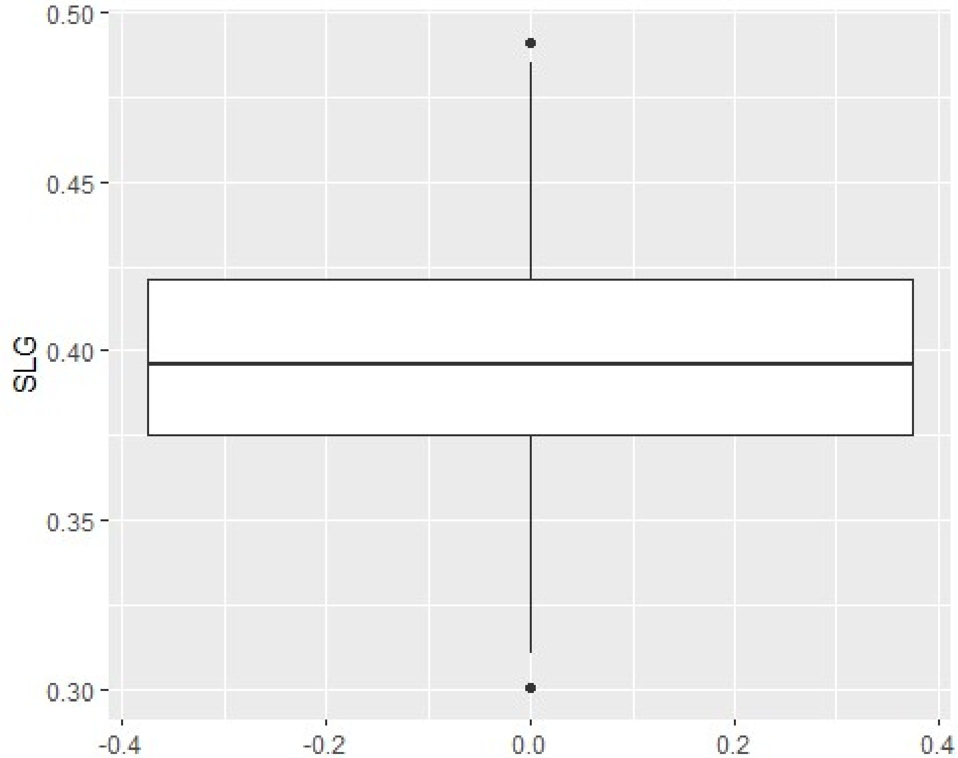
```
ggplot(baseball, aes(x=SLG),) + geom_histogram(bins =20)
```



boxplot for SLG (slugging percentage)

We see from the boxplot that the distribution is not skewed as the median is central.

```
ggplot(baseball, aes(y=SLG),) + geom_boxplot()
```

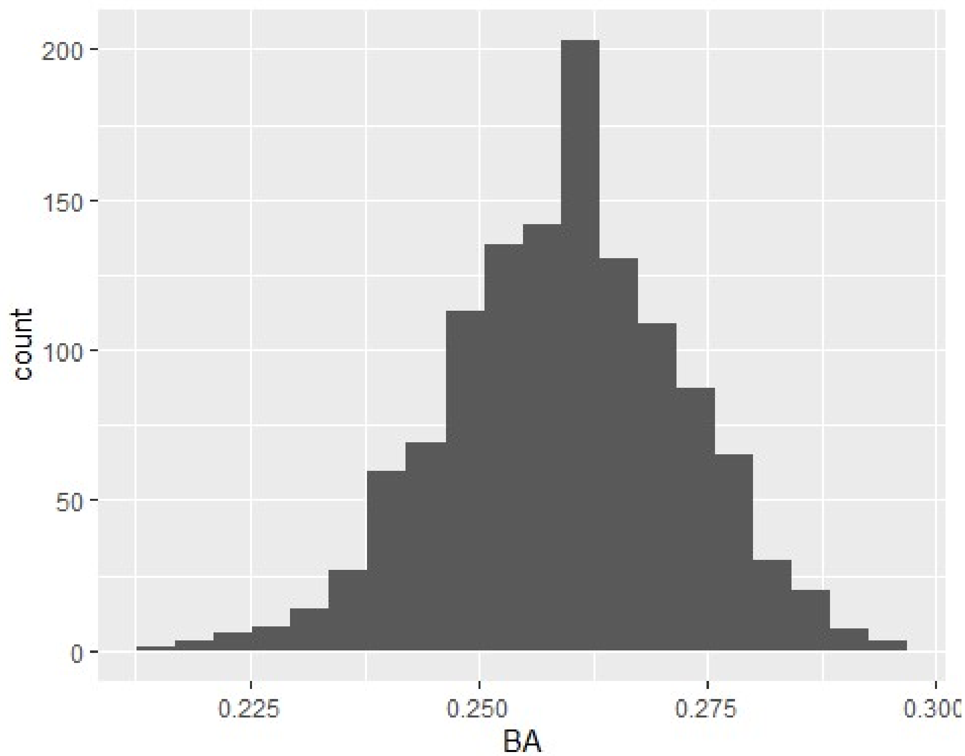


The mean SLG is 0.3973417 and median is 0.396 which is around the same so we see that the distribution is not skewed

```
mean(baseball$SLG)

## [1] 0.3973417

median(baseball$SLG)

## [1] 0.396
```

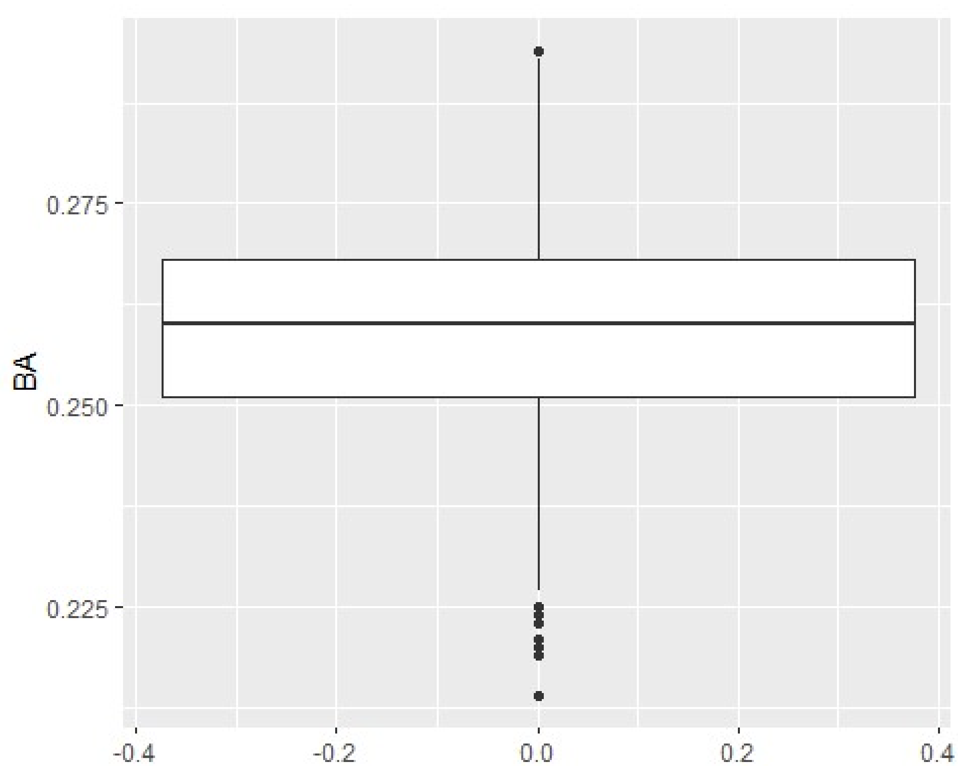
Histogram for BA (batting average) we see that the distribution is not skewed as the histogram is centralized and no one side overpowers the other.

```
ggplot(baseball, aes(x=BA),) + geom_histogram(bins =20)
```



Boxplot for BA (batting average) distribution is not skewed as the median is centralized and the whiskers are equidistant from the median

```
ggplot(baseball, aes(y=BA),) + geom_boxplot()
```



The mean for BA is 0.2592727 and median is 0.26 which is very close thus showing distribution not skewed

```
mean(baseball$BA)

## [1] 0.2592727

median(baseball$BA)

## [1] 0.26
```

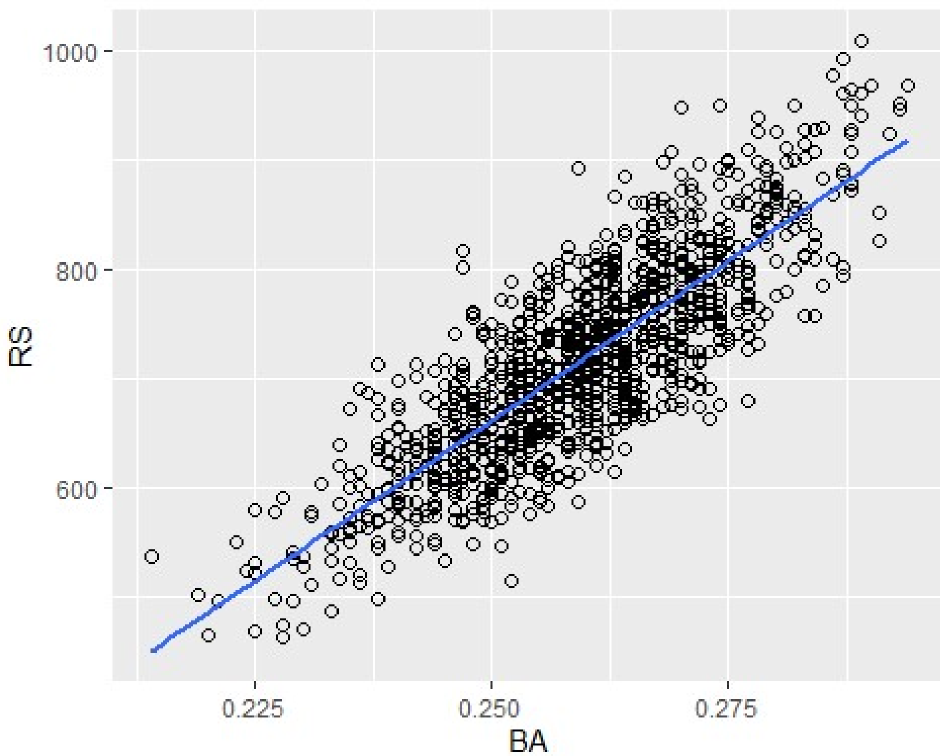
4 Part 2

Marginally regressing RS on BA we have intercept -805.51 and coefficient for BA to be 5864.84. The R^2 is 0.6839

```
RS_BA =lm(RS~BA, data = baseball )
summary(RS_BA)

##
## Call:
## lm(formula = RS ~ BA, data = baseball)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -158.429  -36.057   -1.064   35.018  179.518
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -805.51      29.51  -27.30   <2e-16 ***
## BA           5864.84     113.68   51.59   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 51.48 on 1230 degrees of freedom
## Multiple R-squared:  0.6839, Adjusted R-squared:  0.6837
## F-statistic:  2662 on 1 and 1230 DF,  p-value: < 2.2e-16
```

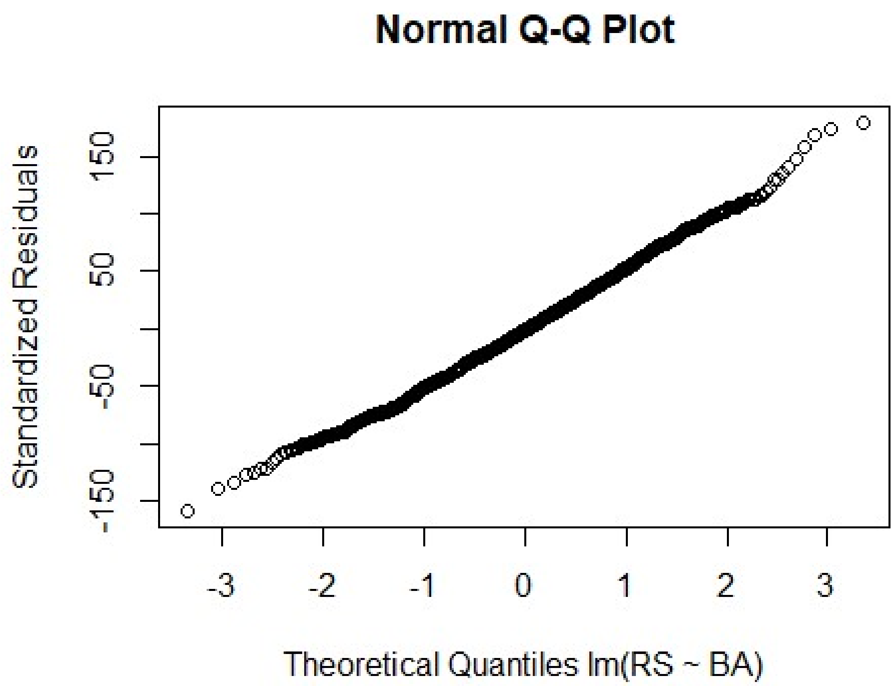
Scatterplot of RS and BA. We see that it follows the fitted line decently close thus showing it is not skewed and therefore the model is reasonable

```
ggplot(baseball, aes(x =BA, y=RS)) +
  geom_point(size=2, shape=1) + geom_smooth(method = "lm", se = FALSE)

## `geom_smooth()` using formula = 'y ~ x'
```



qqplot of fitted residuals RS and BA follows a straight line and only some points at the tails deviate from it thus showing it is not skewed thus reasonable model.

```
qqnorm(RS_BA$residuals,ylab="Standardized Residuals" ,xlab="Theoretical Quantiles lm(RS ~ BA)")
```

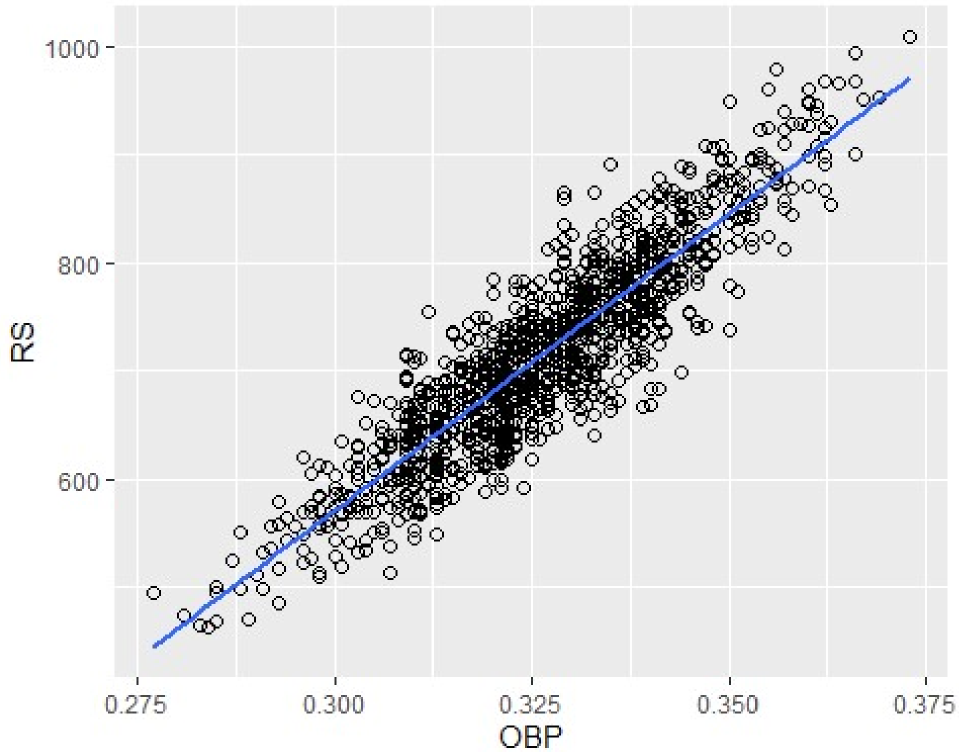


Marginally regressing RS on OBP we get -1076.6 for intercept and 5490.4 for OBP coefficient. While R^2 is 0.8109 which is higher than the BA marginal regression model

```
RS_OBP =lm(RS~OBP, data = baseball )
summary(RS_OBP)

##
## Call:
## lm(formula = RS ~ OBP, data = baseball)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -122.129  -27.110    1.284   26.441  135.265
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1076.6       24.7  -43.59   <2e-16 ***
## OBP           5490.4       75.6   72.62   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 39.82 on 1230 degrees of freedom
## Multiple R-squared:  0.8109, Adjusted R-squared:  0.8107
## F-statistic:  5274 on 1 and 1230 DF,  p-value: < 2.2e-16
```

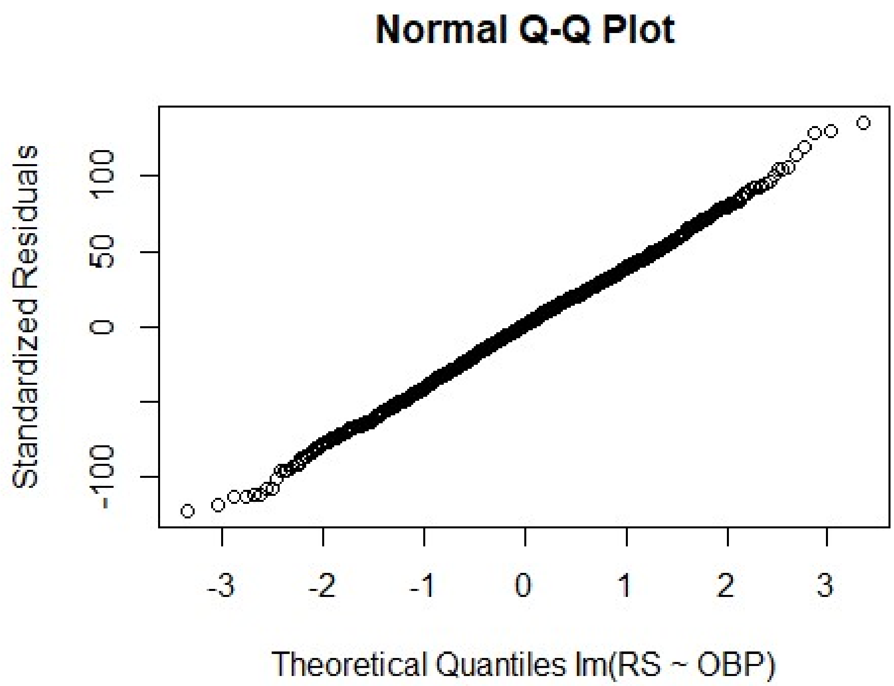
Scatterplot of RS and OBP. We see that the points follow the fitted line quite close thus showing it is not skewed thus a good model. We also see the points hug the fitted line more than the one from the marginal regression of BA

```
ggplot(baseball, aes(x =OBP, y=RS)) +
  geom_point(size=2, shape=1) + geom_smooth(method = "lm", se = FALSE)

## `geom_smooth()` using formula = 'y ~ x'
```



qqplot of fitted residuals RS and OBP. We see that the qqplot follows a straight line that has on a few deviations from the line at that tail thus showing it is not skewed. Therefore the model is reasonable. We also see that the qqplot is more straight than the qqplot of BA.

```
qqnorm(RS_OBP$residuals,ylab="Standardized Residuals" ,xlab="Theoretical
Quantiles lm(RS ~ OBP)")
```

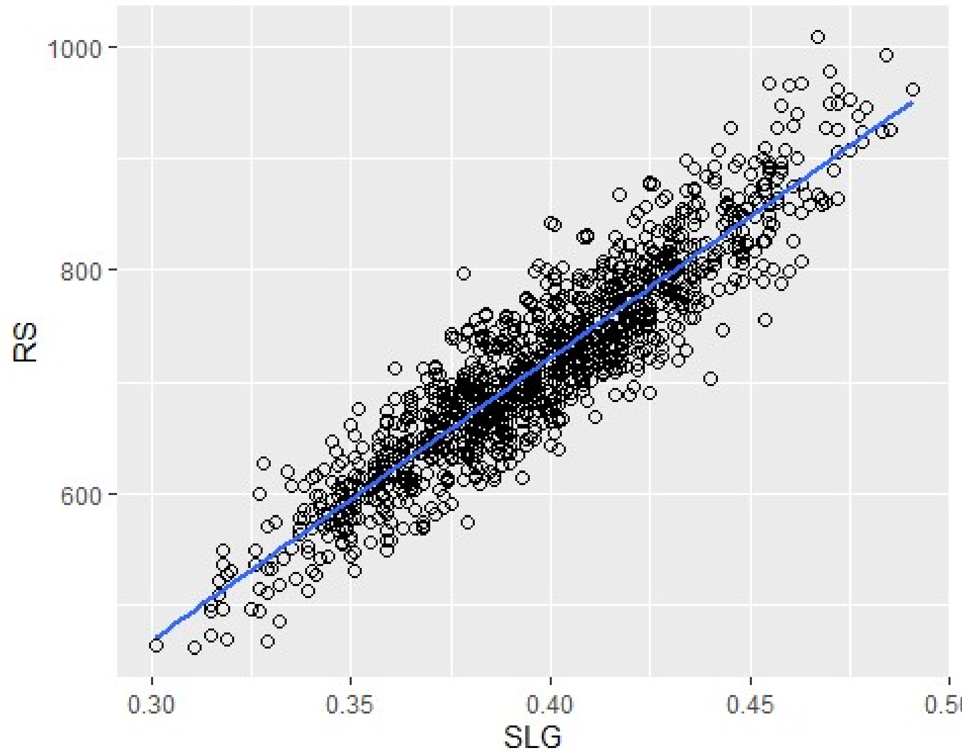


Marginally regressing RS on SLG intercept is -289.37 and slope is 2527.92 $R^2 = 0.8441$ which is higher than the R^2 for BA

```
RS_SLG =lm(RS~SLG, data = baseball )
summary(RS_SLG)

##
## Call:
## lm(formula = RS ~ SLG, data = baseball)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -119.919  -23.666   -1.541   22.353  131.812
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -289.37      12.35  -23.43   <2e-16 ***
## SLG          2527.92      30.98   81.60   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 36.16 on 1230 degrees of freedom
## Multiple R-squared:  0.8441, Adjusted R-squared:  0.844
## F-statistic:  6659 on 1 and 1230 DF,  p-value: < 2.2e-16
```

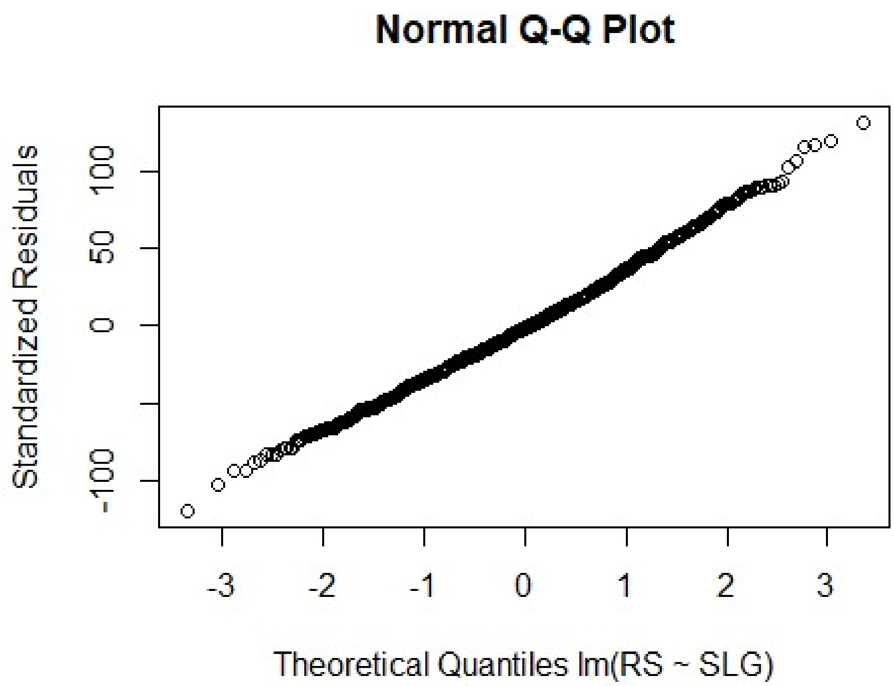
Scatterplot of RS and SLG. The scatterplot follows the fitted line quite well thus showing it is not skewed and the model is reasonable. Also the points hug the fitted line better thant the marginal regression with BA

```
ggplot(baseball, aes(x =SLG, y=RS)) +
  geom_point(size=2, shape=1) + geom_smooth(method = "lm", se = FALSE)

## `geom_smooth()` using formula = 'y ~ x'
```



qqplot of fitted residuals RS and SLG. We see that the qqplot follows a straight 45 degree line well which shows it is not skewed and the model is reasonable. We see only a few tail points that do not follow the straight line but it is still better than the marginal regression on BA

```
qqnorm(RS_SLG$residuals,ylab="Standardized Residuals" ,xlab="Theoretical
Quantiles lm(RS ~ SLG)")
```



Thus we see from the scatterplots qqplots and R^2 that BA is not the best choice. Rather, SLG and OBP seem to perform better

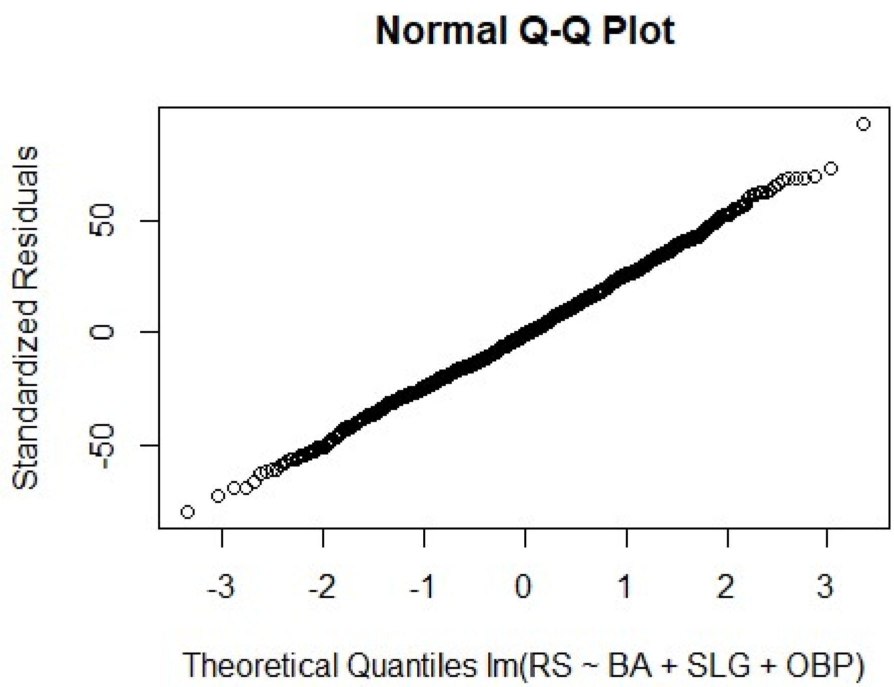
4 Part 3 For model lm(RS~BA +SLG + OBP) we have $R^2 = 0.9249$ which is even better than all the marginal regression models We have -806.08 for intercept -134.90 for BA coefficient, 1533.88 for SLG coefficient and 2900.94 for OBP coefficient. We see that SLG and OBP have a significant positive relationship in our model while BA does not. From part two the coefficient for BA was 5864.84 which is much higher. So it is not consistent with part two however our observation that OBP and SLG were more correlated with RS from part 2 seems to match what we have in this model.

```
RS_all = lm(RS~BA +SLG +OBP, data = baseball)
summary(RS_all)

##
## Call:
## lm(formula = RS ~ BA + SLG + OBP, data = baseball)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -79.693 -16.667  -0.892  16.556  93.068
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -806.08      17.39 -46.348   <2e-16 ***
## BA           -134.90     113.73  -1.186    0.236
## SLG          1533.88      37.76  40.623   <2e-16 ***
```

```
## OBP          2900.94      97.87  29.640   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 25.12 on 1228 degrees of freedom
## Multiple R-squared:  0.9249, Adjusted R-squared:  0.9247
## F-statistic:  5040 on 3 and 1228 DF,  p-value: < 2.2e-16
```

We see from qqplot that it is not skewed as the standardized residuals follow a line with only a few points at the tail that don't.

```
qqnorm(RS_all$residuals,ylab="Standardized Residuals" ,xlab="Theoretical
Quantiles lm(RS ~ BA + SLG + OBP)")
```



For model lm(RS~BA +SLG) we have $R^2 = 0.8711$. Intercept is -551.08, BA coefficient is 1904.66 and SLG coefficient is 1943.77. The R^2 is smaller than 0.9249 which was the R^2 from the previous model. Thus based on the R^2 I would prefer the previous model more.

```
RS_BA_SLG = lm(RS~BA +SLG , data = baseball)
summary(RS_BA_SLG)

##
## Call:
## lm(formula = RS ~ BA + SLG, data = baseball)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -115.432  -23.284   -2.048   21.068  113.415
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -551.08      19.79  -27.85   <2e-16 ***
## BA           1904.66     118.56   16.07   <2e-16 ***
## SLG          1943.77      46.00   42.26   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 32.88 on 1229 degrees of freedom
## Multiple R-squared:  0.8711, Adjusted R-squared:  0.8709
## F-statistic:  4154 on 2 and 1229 DF,  p-value: < 2.2e-16
```

4 Part 4

```
predict_data = (baseball[baseball$Year<2002 & baseball$Team == "OAK",])
RD = predict_data$RS-predict_data$RA
predict_data = cbind(predict_data, RD)
head(predict_data)
```

```
##     Team League Year  RS  RA   W   OBP   SLG    BA Playoffs RankSeason
## 351  OAK     AL 2001 884 645 102 0.345 0.439 0.264        1          2
## 381  OAK     AL 2000 947 813  91 0.360 0.458 0.270        1          4
## 411  OAK     AL 1999 893 846  87 0.355 0.446 0.259        0         NA
## 441  OAK     AL 1998 804 866  74 0.338 0.397 0.257        0         NA
## 470  OAK     AL 1997 764 946  65 0.339 0.423 0.260        0         NA
## 498  OAK     AL 1996 861 900  78 0.344 0.452 0.265        0         NA
##     RankPlayoffs   G  OOBP  OSLG   RD
## 351            4 162 0.308 0.380  239
## 381            4 161 0.348 0.423  134
## 411           NA 162 0.344 0.428   47
## 441           NA 162    NA    NA  -62
## 470           NA 162    NA    NA -182
## 498           NA 162    NA    NA  -39
```

```
W_RD = lm(W~RD, data= predict_data)
summary(W_RD)
```

```
##
## Call:
## lm(formula = W ~ RD, data = predict_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -5.5960 -2.3350 -0.3218  1.6018  6.9646
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 81.667331   0.614140   133.0   <2e-16 ***
## RD           0.100932   0.004899    20.6   <2e-16 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.332 on 28 degrees of freedom
## Multiple R-squared:  0.9381, Adjusted R-squared:  0.9359
## F-statistic: 424.4 on 1 and 28 DF,  p-value: < 2.2e-16
```

```
RS_OBP_SLG = lm(RS~OBP+SLG, data=predict_data)
summary(RS_OBP_SLG)
```

```
##
## Call:
## lm(formula = RS ~ OBP + SLG, data = predict_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.686 -13.542  -0.076  20.950  60.333
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -949.2      138.1  -6.874 2.19e-07 ***
## OBP           3332.3      728.3   4.575 9.53e-05 ***
## SLG           1499.2      347.0   4.320 0.000189 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 27.51 on 27 degrees of freedom
## Multiple R-squared:  0.9191, Adjusted R-squared:  0.9131
## F-statistic: 153.4 on 2 and 27 DF,  p-value: 1.798e-15
```

```
RA_OOBP_OSLG = lm(RA~OOBP + OSLG, data=predict_data)
summary(RA_OOBP_OSLG)
```

```
##
## Call:
## lm(formula = RA ~ OOBP + OSLG, data = predict_data)
##
## Residuals:
## ALL 3 residuals are 0: no residual degrees of freedom!
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -910.5        NaN     NaN      NaN
## OOBP         -1556.5        NaN     NaN      NaN
## OSLG          5354.8        NaN     NaN      NaN
##
## Residual standard error: NaN on 0 degrees of freedom
##   (27 observations deleted due to missingness)
## Multiple R-squared:      1, Adjusted R-squared:    NaN
## F-statistic:   NaN on 2 and 0 DF,  p-value: NA
```

```
W_RD_int = 81.667331
RD = 0.100932

RS_OBP_SLG_int = -949.2
OBP = 3332.3
SLG = 1499.2


RA_OOBP_OSLG_int = -910.5
OOBP = -1556.5
OSLG = 5354.8

#values we predicted for 2002
OBP_2002 =  0.349
SLG_2002 = 0.430
OOBP_2002 = 0.307
OSLG_2002 = 0.373

form = W_RD_int + RD*((RS_OBP_SLG_int + OBP*OBP_2002 + SLG* SLG_2002)-
(RA_OOBP_OSLG_int+OOBP *OOBP_2002 + OSLG*OSLG_2002))

form

## [1] 106.8432
```

We get 106.8432 wins which is close to the actual result of 103

```
(baseball[baseball$Year==2002 & baseball$Team == "OAK",])

##     Team League Year  RS  RA   W   OBP   SLG    BA Playoffs RankSeason
## 321  OAK     AL 2002 800 654 103 0.339 0.432 0.261        1          1
##     RankPlayoffs   G  OOBP  OSLG
## 321            4 162 0.315 0.384
```