D80A2

Shansong Huang

05/02/2021

```r
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.3
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## Warning: package 'ggplot2' was built under R version 3.6.2

## Warning: package 'tibble' was built under R version 3.6.2

## Warning: package 'tidyr' was built under R version 3.6.2

## Warning: package 'readr' was built under R version 3.6.2

## Warning: package 'purrr' was built under R version 3.6.2

## Warning: package 'dplyr' was built under R version 3.6.2

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Question 1 : Housepricing

```r
# 1a
# houseprice <- file.choose()
housedata <- read_csv("/Users/samhuang/course_2020/stad80/Assignments/a2/_data_hw2/housingprice.csv")
```

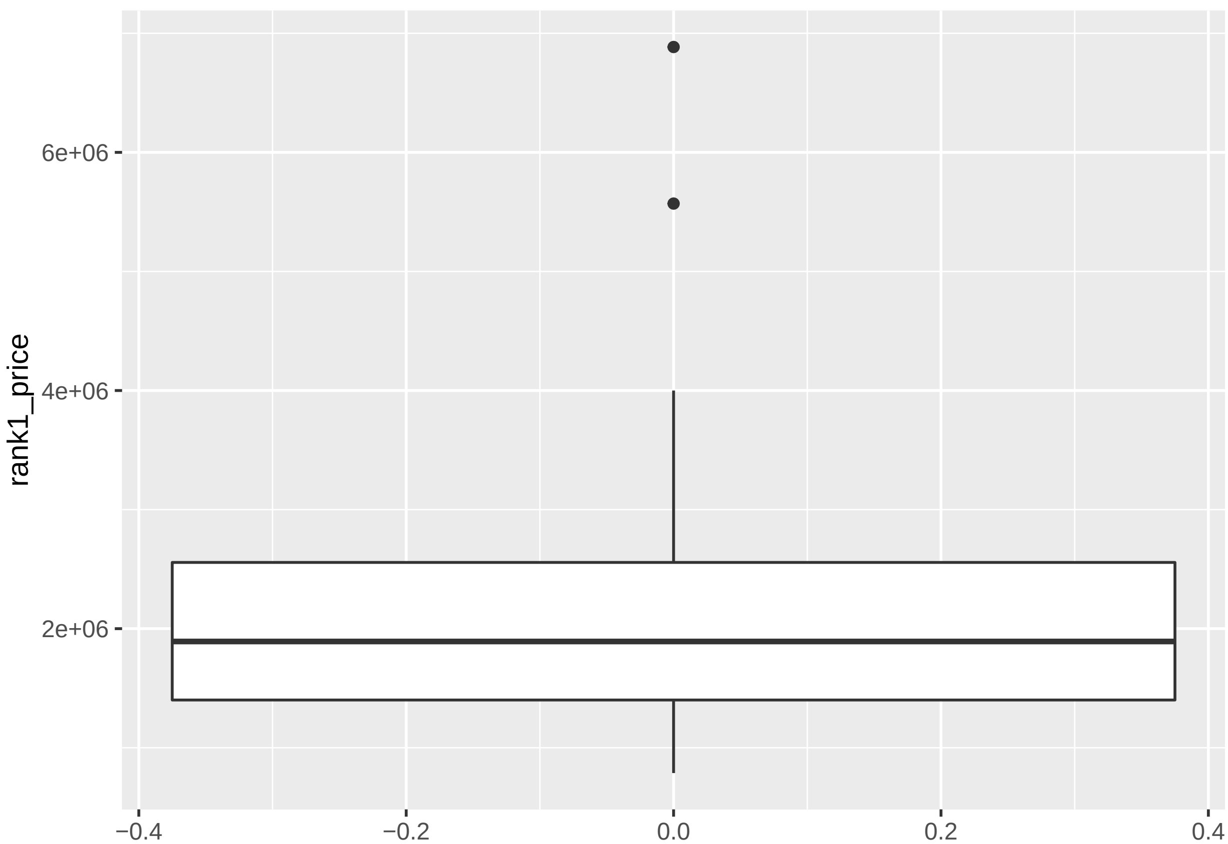
```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   id = col_character(),
##   date = col_datetime(format = "")
## )
## i Use `spec()` for the full column specifications.
```

```r
zipcode_mean <- tapply(housedata$price,housedata[,c("zipcode")],mean)
price_sort <- sort(zipcode_mean, decreasing = TRUE)
price_sort[1:3]
```

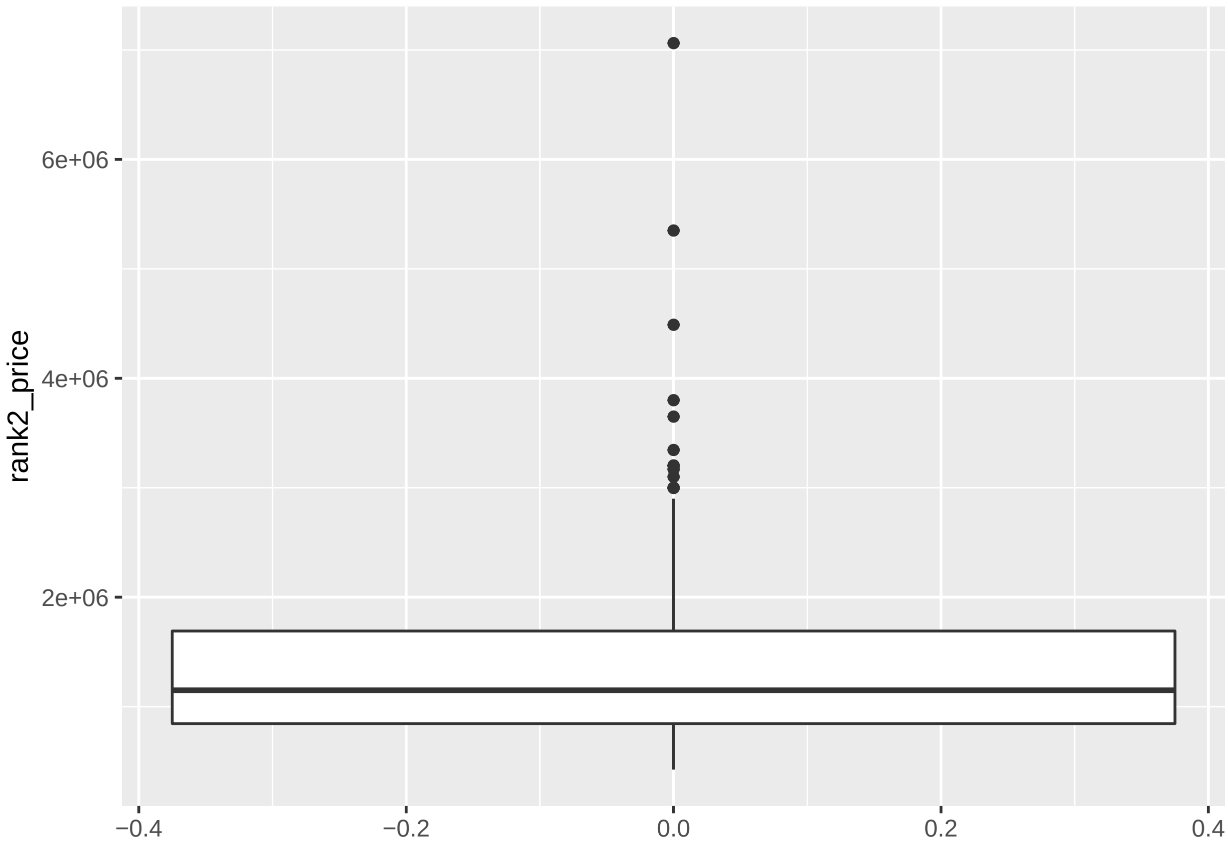
```
## zipcode
##   98039   98004   98040
## 2160607 1355927 1194230
```

Top 3 zipcodes whose average housing prices are most expensive: 98039 98004 98040, (Rank 1, Rank 2, Rank 3).

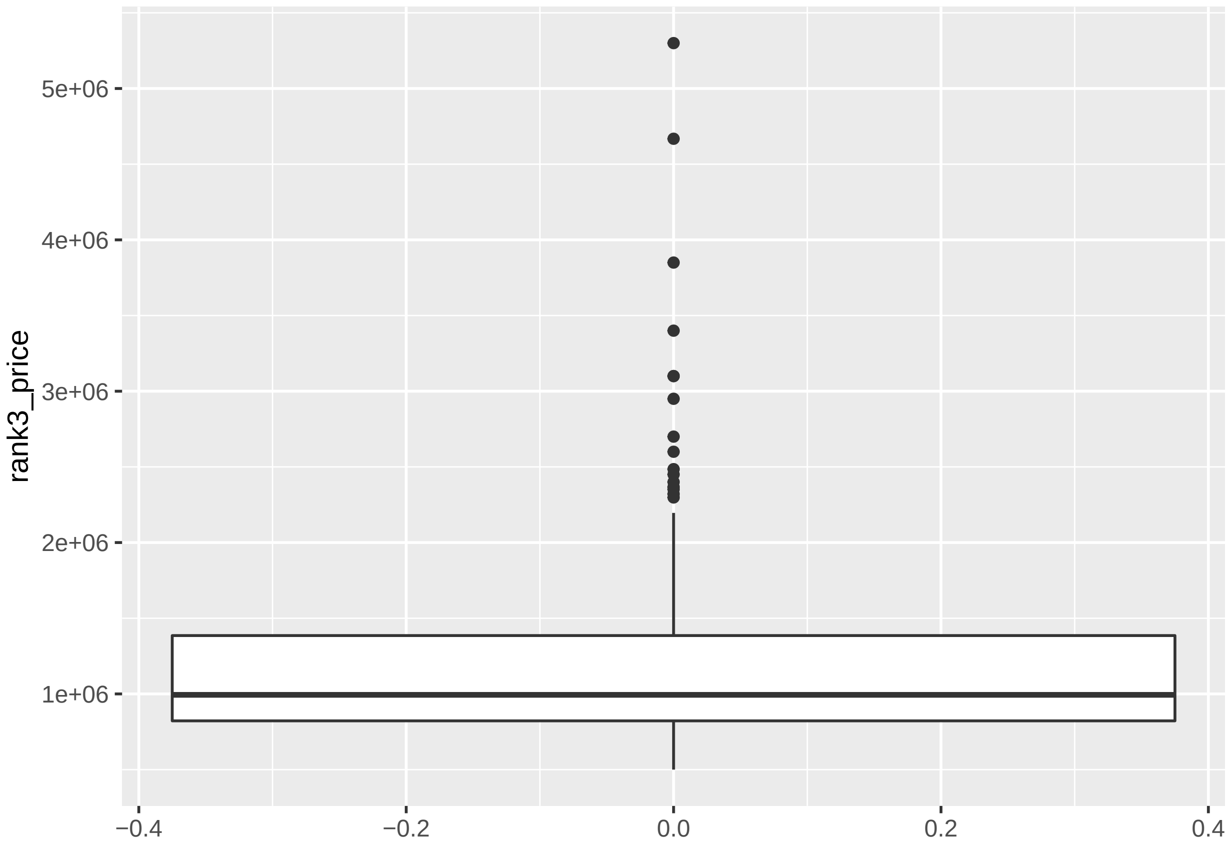
```r
# 1a boxplot
# rank = 1, 98039 boxplot
rank1 <- housedata %>% filter(zipcode == 98039)
rank1_price <- rank1$price
ggplot(data = NULL, aes(y = rank1_price)) + geom_boxplot()
```




```r
# 1a boxplot
# rank=2 , 98004 boxplot
rank2 <- housedata %>% filter(zipcode == 98004)
rank2_price <- rank2$price
ggplot(data = NULL, aes(y = rank2_price)) + geom_boxplot()
```



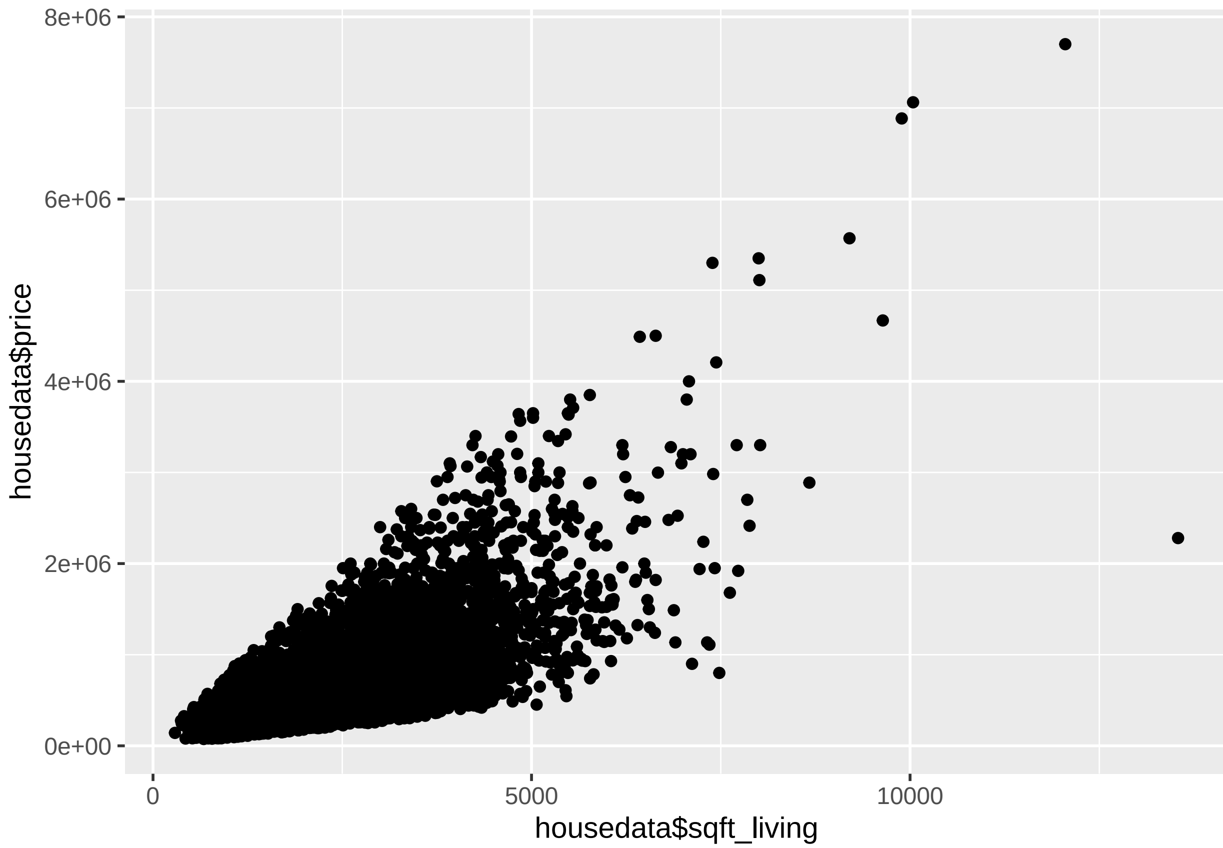

```r
# 1a boxplot
# rank = 3 , 98040 boxplot
rank3 <- housedata %>% filter(zipcode == 98040)
rank3_price <- rank3$price
ggplot(data = NULL, aes(y = rank3_price)) + geom_boxplot()
```



1(a) End

1 (b) scatter plot sqft_living and housing price

```
ggplot(housedata, aes(x=housedata$sqft_living, y = housedata$price)) +geom_point()
```




```r
# td <- file.choose()
train_data <- read_csv("/Users/samhuang/course_2020/stad80/Assignments/a2/_data_hw2/train.data.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   date = col_datetime(format = "")
## )
## i Use `spec()` for the full column specifications.
```

```r
# testd <-file.choose()
test_data <- read_csv("/Users/samhuang/course_2020/stad80/Assignments/a2/_data_hw2/test.data.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_double(),
##   date = col_datetime(format = "")
## )
## i Use `spec()` for the full column specifications.
```

1 (c) build a linear model on the taining data, regressing housing price on : : bedrooms, bathrooms, sqft_living, sqft_lot

```
train_model <- lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot, data = train_data)
summary(train_model)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot,
##     data = train_data)
##
## 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
```

The Rsquared of the model on training data is 0.5101.

```
test.pred <- predict(train_model, newdata = test_data)
test.y <- test_data$price
SS.total <- sum((test.y - mean(test.y))^2)
SS.residual <- sum((test.y - test.pred)^2)
test.rsq <- 1 - SS.residual/SS.total
test.rsq
```

```
## [1] 0.5049945
```

The Rsquared testing data is 0.5049945

1(c) end

1 (d) add zipcode in learni model, what's the Rsquared of the new model on the training data and testing data

```
train_model_zip <-  lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot + zipcode, data = train_da
summary(train_model_zip)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     zipcode, data = train_data)
##
## 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
```

```r
test.pred <- predict(train_model_zip, newdata = test_data)
test.y <- test_data$price
SS.total <- sum((test.y - mean(test.y))^2)
SS.residual <- sum((test.y - test.pred)^2)
test.rsq <- 1 - SS.residual/SS.total
test.rsq
```

```
## [1] 0.5120097
```

If I add zipcode in my linear model, the Rsq of the new model with zipcode on traning data is 0.5161, the Rsq of the new model on testing data is 0.5120097

1(d) end

1 (e) Guess the price of Bill Gate's house

```r
# bg <- file.choose()
bill_house <- read_csv("/Users/samhuang/course_2020/stad80/Assignments/a2/_data_hw2/fancyhouse.csv")
```

```
## Warning: Missing column names filled in: 'X1' [1]
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   X1 = col_double(),
##   bedrooms = col_double(),
```

```
##   bathrooms = col_double(),
##   sqft_living = col_double(),
##   sqft_lot = col_double(),
##   floors = col_double(),
##   zipcode = col_double(),
##   condition = col_double(),
##   grade = col_double(),
##   waterfront = col_double(),
##   view = col_double(),
##   sqft_above = col_double(),
##   sqft_basement = col_double(),
##   yr_built = col_double(),
##   yr_renovated = col_double(),
##   lat = col_double(),
##   long = col_double(),
##   sqft_living15 = col_double(),
##   sqft_lot15 = col_double()
## )
```

```r
bill_house
```

```
## # A tibble: 1 x 19
##      X1 bedrooms bathrooms sqft_living sqft_lot floors zipcode condition grade
##   <dbl>    <dbl>     <dbl>       <dbl>    <dbl>  <dbl>   <dbl>     <dbl> <dbl>
## 1     1        8        25       50000   225000      4   98039        10    10
## # ... with 10 more variables: waterfront <dbl>, view <dbl>, sqft_above <dbl>,
## #   sqft_basement <dbl>, yr_built <dbl>, yr_renovated <dbl>, lat <dbl>,
## #   long <dbl>, sqft_living15 <dbl>, sqft_lot15 <dbl>
```

```r
predict(train_model_zip, bill_house)
```

```
##        1
## 15642273
```

Guess by using the model, the price of Bill Gates' house is 15642273. My guess by the linear model is not reasonable, as we can see house price of 15642273 is far away from the boxplot for zipcode(Bill Gates' house zipcode) = 98039, where for zipcode = 98039, the most house price range from about 0.8 million to 1.8 million, but my guess is about 15 million, which is not reasonable.

1(e) end

1 (f) If n > d+1, show that adding another covariate in the model near hurts Rsq over the training data. # 1(f) end

Question 2 2 (a)

```r
new_model <- lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot + zipcode + (bedrooms*bathrooms),
                data = train_data)
summary(new_model)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     zipcode + (bedrooms * bathrooms), data = train_data)
##
## 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
```

```r
test.pred <- predict(new_model, newdata = test_data)
test.y <- test_data$price
SS.total <- sum((test.y - mean(test.y))^2)
SS.residual <- sum((test.y - test.pred)^2)
test.rsq <- 1 - SS.residual/SS.total
test.rsq
```

```
## [1] 0.5165114
```

The Rsq of the new model on training data is 0.5224 The Rsq of the new model on testing data is 0.5254333 # 2(a) end

```r
names(train_data)
```

```
##  [1] "X1"            "id"            "date"          "price"
##  [5] "bedrooms"      "bathrooms"     "sqft_living"   "sqft_lot"
##  [9] "floors"        "waterfront"    "view"          "condition"
## [13] "grade"         "sqft_above"    "sqft_basement" "yr_built"
## [17] "yr_renovated"  "zipcode"       "lat"           "long"
## [21] "sqft_living15" "sqft_lot15"
```

2(b) Consider that in general if the house is renovated ,then the house price will go up, we have to consider which year is house renovated.

```r
my_new_model <- lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot + zipcode + (bedrooms*bathrooms
                data = train_data)
summary(my_new_model)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     zipcode + (bedrooms * bathrooms) + yr_renovated, data = train_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2166888  -138341   -22296   101050  3623178
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)        -4.567e+07  3.919e+06 -11.654  < 2e-16 ***
## bedrooms           -1.201e+05  5.335e+03 -22.521  < 2e-16 ***
## bathrooms          -9.620e+04  8.651e+03 -11.120  < 2e-16 ***
## sqft_living         3.094e+02  3.729e+00  82.972  < 2e-16 ***
## sqft_lot           -3.506e-01  5.440e-02  -6.444  1.2e-10 ***
## zipcode             4.685e+02  3.993e+01  11.733  < 2e-16 ***
## yr_renovated        6.302e+01  5.128e+00  12.291  < 2e-16 ***
## bedrooms:bathrooms  3.056e+04  2.229e+03  13.708  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 252700 on 15121 degrees of freedom
## Multiple R-squared:  0.5271, Adjusted R-squared:  0.5269
## F-statistic:  2408 on 7 and 15121 DF,  p-value: < 2.2e-16
```

```r
test.pred <- predict(my_new_model, newdata = test_data)
test.y <- test_data$price
SS.total <- sum((test.y - mean(test.y))^2)
SS.residual <- sum((test.y - test.pred)^2)
test.rsq <- 1 - SS.residual/SS.total
test.rsq
```

```
## [1] 0.5254333
```

The Rsq of new model with yr_renovated on training data is 0.5271 The Rsq of new model with yr_renovated on testing data is 0.5254333 We can see that Rsq increased when we added yr_renovated in our model, so the model fitted better, also our Rsq for testing data is very closed to our Rsq in training data. So adding yr_renovated made our model better. # 2(b) end

2(c) Polynomial regression, add polynomial terms of the bedrooms and bathrooms variables of degrees 2 and 3 in the model.

```r
poly_model <-  lm(price ~ bedrooms + bathrooms + sqft_living + sqft_lot + zipcode +poly(bedrooms,2) + po
                    data = train_data)
summary(poly_model)
```

```
##
## Call:
## lm(formula = price ~ bedrooms + bathrooms + sqft_living + sqft_lot +
##     zipcode + poly(bedrooms, 2) + poly(bathrooms, 3), data = train_data)
##
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -3312253  -136245   -26067    98812  2733696
##
## Coefficients: (2 not defined because of singularities)
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -3.952e+07  3.865e+06 -10.224  < 2e-16 ***
## bedrooms            -5.316e+04  2.702e+03 -19.672  < 2e-16 ***
## bathrooms            2.250e+04  4.082e+03   5.512 3.61e-08 ***
## 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 ***
## poly(bedrooms, 2)1          NA         NA      NA       NA
## poly(bedrooms, 2)2   1.803e+06  2.556e+05   7.054 1.82e-12 ***
## poly(bathrooms, 3)1         NA         NA      NA       NA
## 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
## ---
## 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
```

```
test.pred <- predict(poly_model, newdata = test_data)
```

```
## Warning in predict.lm(poly_model, newdata = test_data): prediction from a rank-
## deficient fit may be misleading
```

```
test.y <- test_data$price
SS.total <- sum((test.y - mean(test.y))^2)
SS.residual <- sum((test.y - test.pred)^2)
test.rsq <- 1 - SS.residual/SS.total
test.rsq
```

```
## [1] 0.5285121
```

The Rsq of new model with polynomial terms on training data is 0.5423 The Rsq of new model with polynomial terms on testing data is 0.5285121

Question 3 Wine Pricing

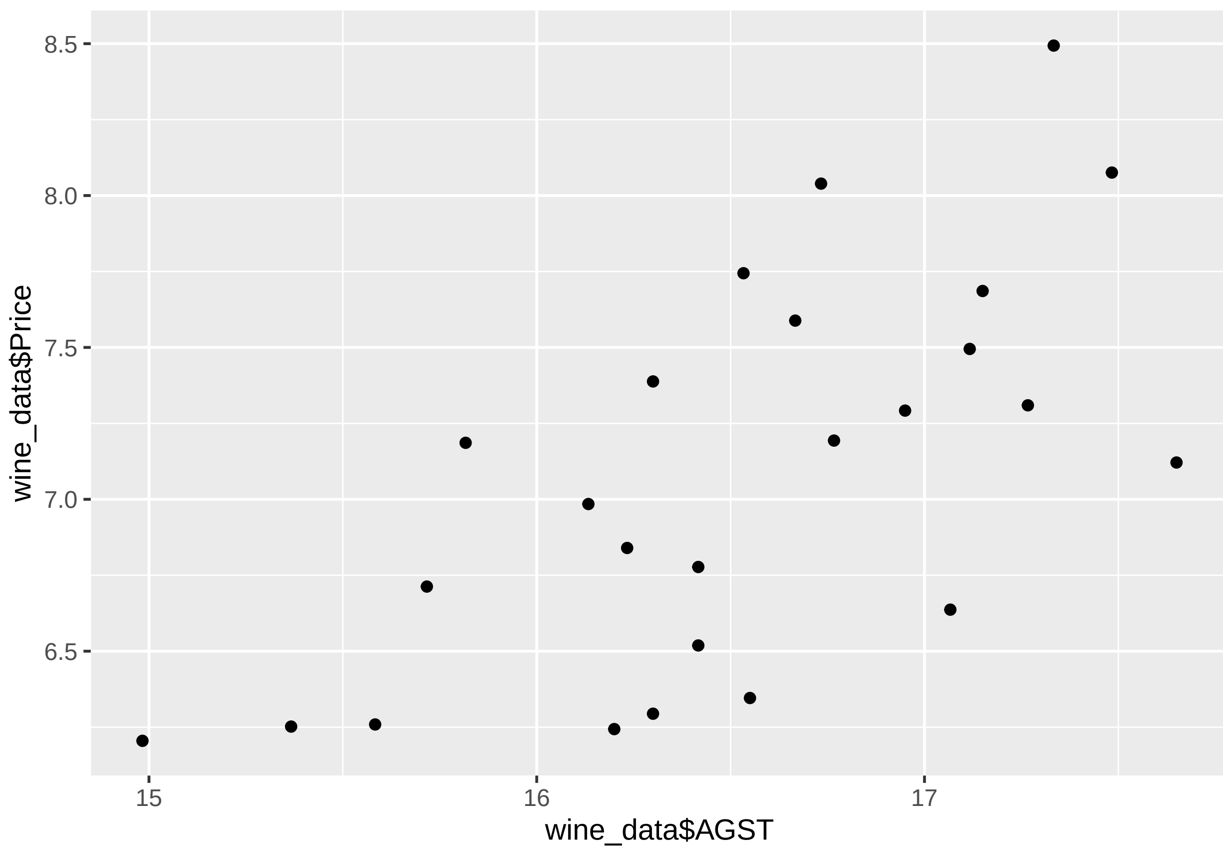
```
#wi <- file.choose()
wine_data <- read_csv("/Users/samhuang/course_2020/stad80/Assignments/a2/_data_hw2/wine.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Year = col_double(),
##   Price = col_double(),
##   WinterRain = col_double(),
##   AGST = col_double(),
##   HarvestRain = col_double(),
```

```
##   Age = col_double(),
##   FrancePop = col_double()
## )
```

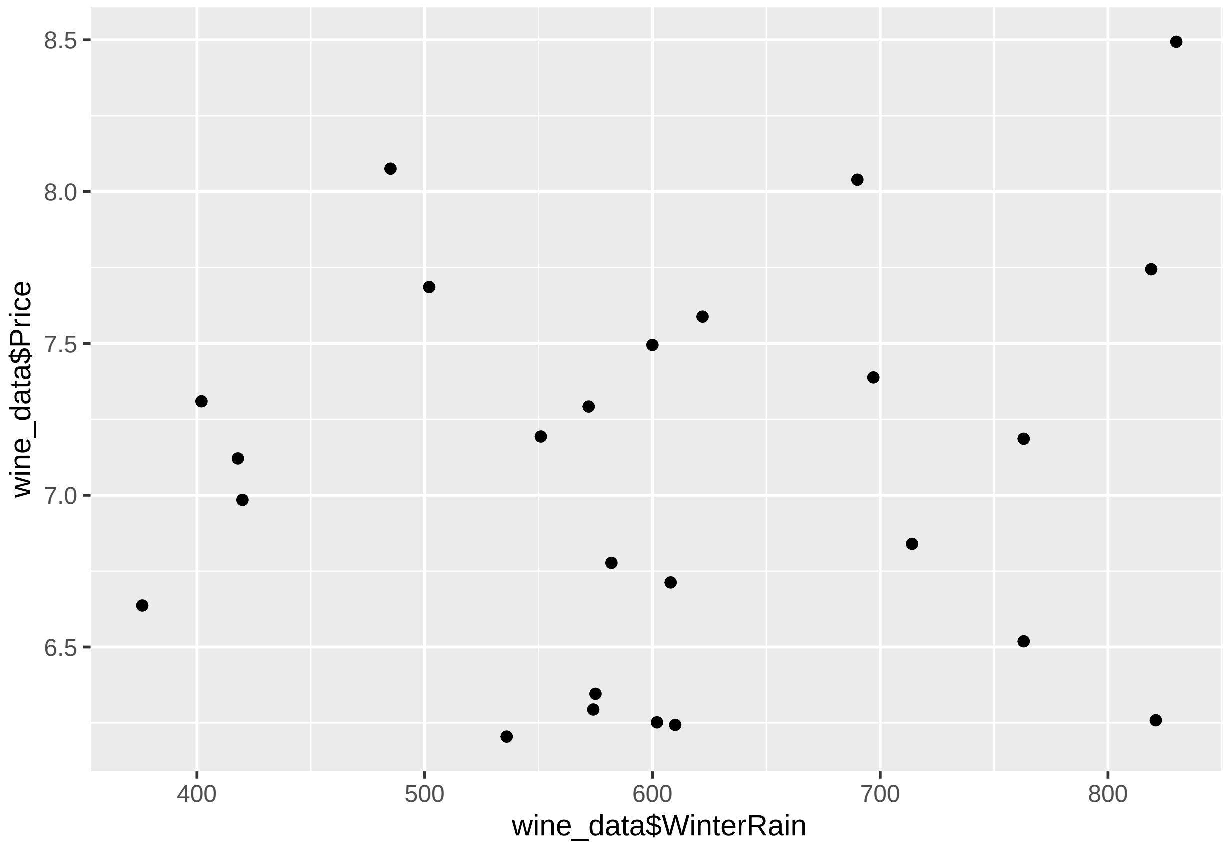
Q2 Part I scatter plot : Price v.s. AGST

```r
ggplot(wine_data,aes(x = wine_data$AGST, y = wine_data$Price)) + geom_point()
```



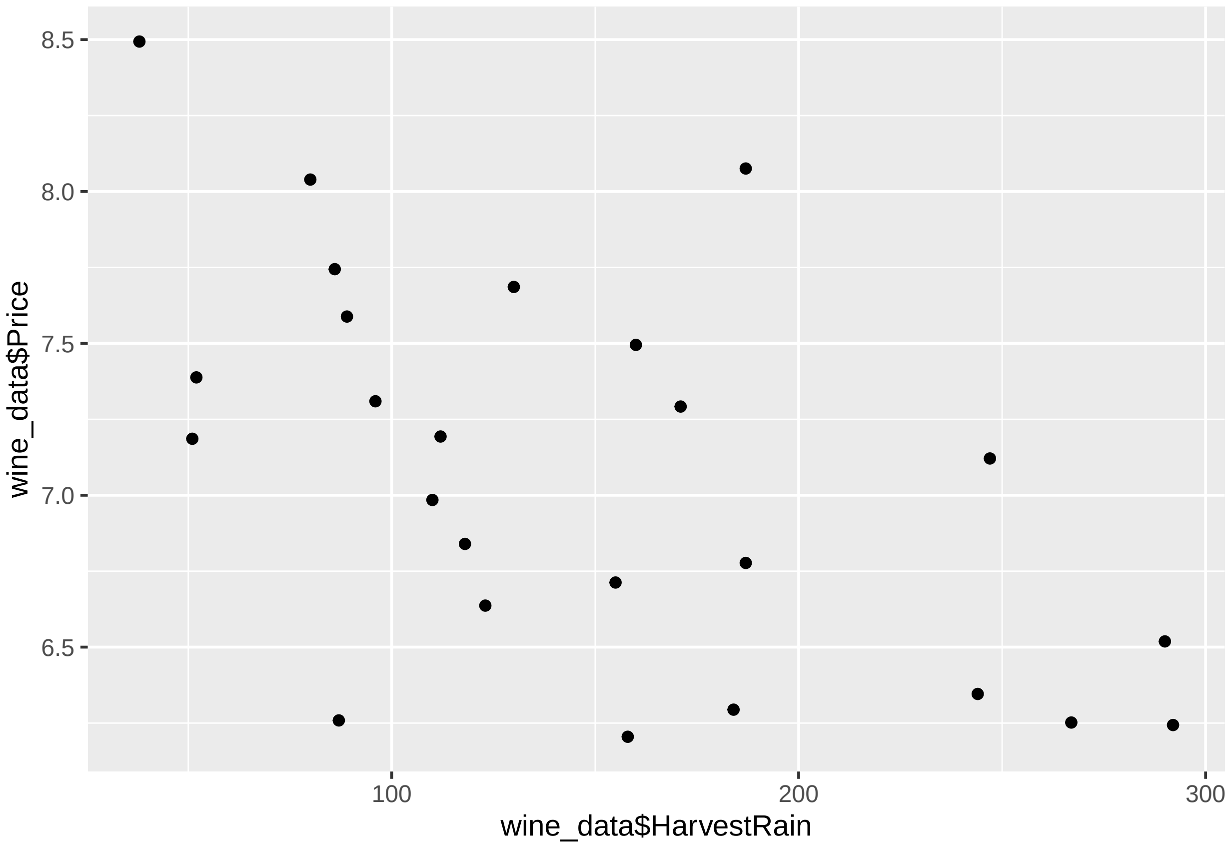
scatter plot : Price v.s. WinterRain

```r
ggplot(wine_data,aes(x = wine_data$WinterRain, y = wine_data$Price)) + geom_point()
```



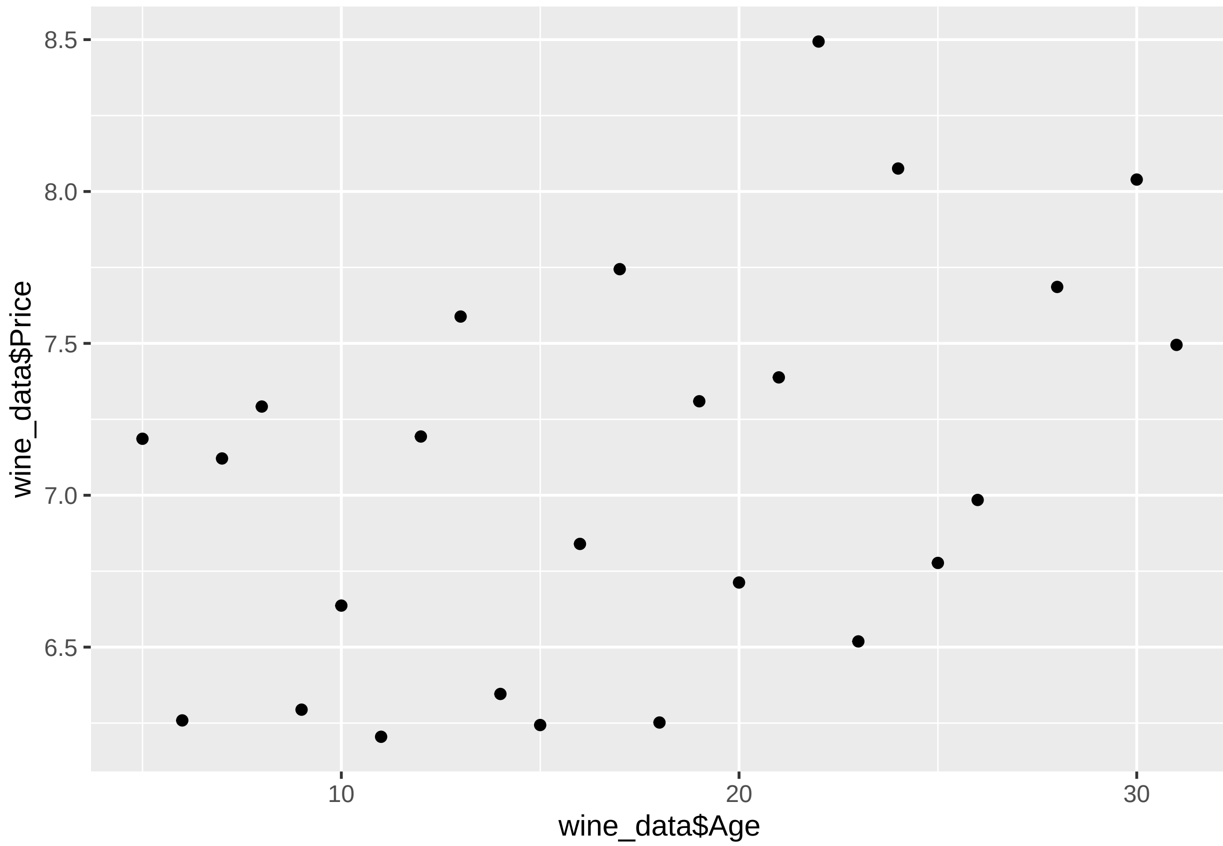
scatter plot : Price v.s. HarvestRain

```
ggplot(wine_data,aes(x = wine_data$HarvestRain, y = wine_data$Price)) + geom_point()
```



scatter plot : Price v.s. Age

```
ggplot(wine_data,aes(x = wine_data$Age, y = wine_data$Price)) + geom_point()
```



From the four plots, we can see that both AGST and Harvestrain is correlated with Price. There is a postive trend between Price and AGST, and there is a negative trend between Price and HarvestRain.

Jusify by calculating the Pearson's correlation.

```
AGST_Price <- cor.test(wine_data$Price, wine_data$AGST,
                    method = "pearson")
AGST_Price
```

```
##
##  Pearson's product-moment correlation
##
## data:  wine_data$Price and wine_data$AGST
## t = 4.2083, df = 23, p-value = 0.000335
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.3576371 0.8366511
## sample estimates:
##       cor
## 0.6595629
```

For correlation of Price and AGST, the Pearson' correlation is 0.6595629, which is a positive trend.

```
Harvestrain_Price <- cor.test(wine_data$Price, wine_data$HarvestRain,
                    method = "pearson")
Harvestrain_Price
```

```
##
##  Pearson's product-moment correlation
##
## data:  wine_data$Price and wine_data$HarvestRain
## t = -3.2698, df = 23, p-value = 0.003366
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.7839554 -0.2163467
## sample estimates:
##        cor
## -0.5633219
```

For correlation of Price and AGST, the Pearson' correlation is -0.5633219, which is a negative trend.

Q2 Part I End

Q2 Part II Marginal Regression Analysis

```r
library("margins")
```

```
## Warning: package 'margins' was built under R version 3.6.2
```

```r
marginal_model <- lm(formula =  Price ~ AGST, data = wine_data)
summary(marginal_model)$r.squared
```

```
## [1] 0.4350232
```

```r
lm(formula =  Price ~ AGST, data = wine_data)
```

```
##
## Call:
## lm(formula = Price ~ AGST, data = wine_data)
##
## Coefficients:
## (Intercept)         AGST
##     -3.4178       0.6351
```

The fitted coefficient for AGST is 0.6351. THe Rsq is 0.4350232.

Q2 Part II End

Q2 Part III Mutiple regression Analysis

```r
# wtest <-file.choose()
wine_test <- read_csv("/Users/samhuang/course_2020/stad80/Assignments/a2/_data_hw2/winetest.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   Year = col_double(),
##   Price = col_double(),
##   WinterRain = col_double(),
##   AGST = col_double(),
##   HarvestRain = col_double(),
##   Age = col_double(),
##   FrancePop = col_double()
## )
```

Add HarvestRain, Age, WinterRain , FrancePop to model one by one. Add HarvestRain

```r
# linear model
Muti_model_1 <- lm(formula =  Price ~ AGST + HarvestRain, data = wine_data)
summary(Muti_model_1)$r.squared
```

```
## [1] 0.7073708
```

```r
# predict and calculate R squared on test data
test.pred <- predict(Muti_model_1, newdata = wine_test)
test.y <- wine_test$Price
SS.total <- sum((test.y - mean(test.y))^2)
SS.residual <- sum((test.y - test.pred)^2)
test.rsq <- 1 - (SS.residual/SS.total)
test.rsq
```

```
## [1] -2.503339
```

Add HarvestRain, Age

```r
# linear model
Muti_model_2 <- lm(formula =  Price ~ AGST + HarvestRain + Age, data = wine_data)
summary(Muti_model_2)$r.squared
```

```
## [1] 0.7900362
```

```r
# predict and calculate R squared on test data
test.pred <- predict(Muti_model_2, newdata = wine_test)
test.y <- wine_test$Price
SS.total <- sum((test.y - mean(test.y))^2)
SS.residual <- sum((test.y - test.pred)^2)
test.rsq <- 1 - (SS.residual/SS.total)
test.rsq
```

```
## [1] -0.5080824
```

Add HarvestRain, Age, WinterRain ,

```r
# linear model
Muti_model_3 <- lm(formula =  Price ~ AGST + HarvestRain + Age + WinterRain, data = wine_data)
summary(Muti_model_3)$r.squared
```

```
## [1] 0.8285662
```

```r
# predict and calculate R squared on test data
test.pred <- predict(Muti_model_3, newdata = wine_test)
test.y <- wine_test$Price
SS.total <- sum((test.y - mean(test.y))^2)
SS.residual <- sum((test.y - test.pred)^2)
test.rsq <- 1 - (SS.residual/SS.total)
test.rsq
```

```
## [1] 0.3343905
```

Add HarvestRain, Age, WinterRain , FrancePop

```r
# linear model
Muti_model_4 <- lm(formula =  Price ~ AGST + HarvestRain + Age + WinterRain + FrancePop, data = wine_da
summary(Muti_model_4)$r.squared
```

```
## [1] 0.8293592
```

```r
# predict and calculate R squared on test data
test.pred <- predict(Muti_model_4, newdata = wine_test)
test.y <- wine_test$Price
SS.total <- sum((test.y - mean(test.y))^2)
SS.residual <- sum((test.y - test.pred)^2)
test.rsq <- 1 - (SS.residual/SS.total)
test.rsq
```

```
## [1] 0.2120672
```

As we can see, when we added WinterRain to our model, it increased Rsq on training data, and even increased Rsq on testing data enormously. So based on Rsq, we should choose WinterRain to our model. Our model is consistent with Prof.Ashenfelter's finding , since we found WinterRain is a very important feature for the Wine price predicting, and also we found a negative trend from our scatter plot HarvestRain v.s. Price.

Q2 Part III End

Question 4 Moneyball

```r
# basedata <- file.choose()
bdata <- read_csv( "/Users/samhuang/course_2020/stad80/Assignments/a2/_data_hw2/baseball.csv")
```

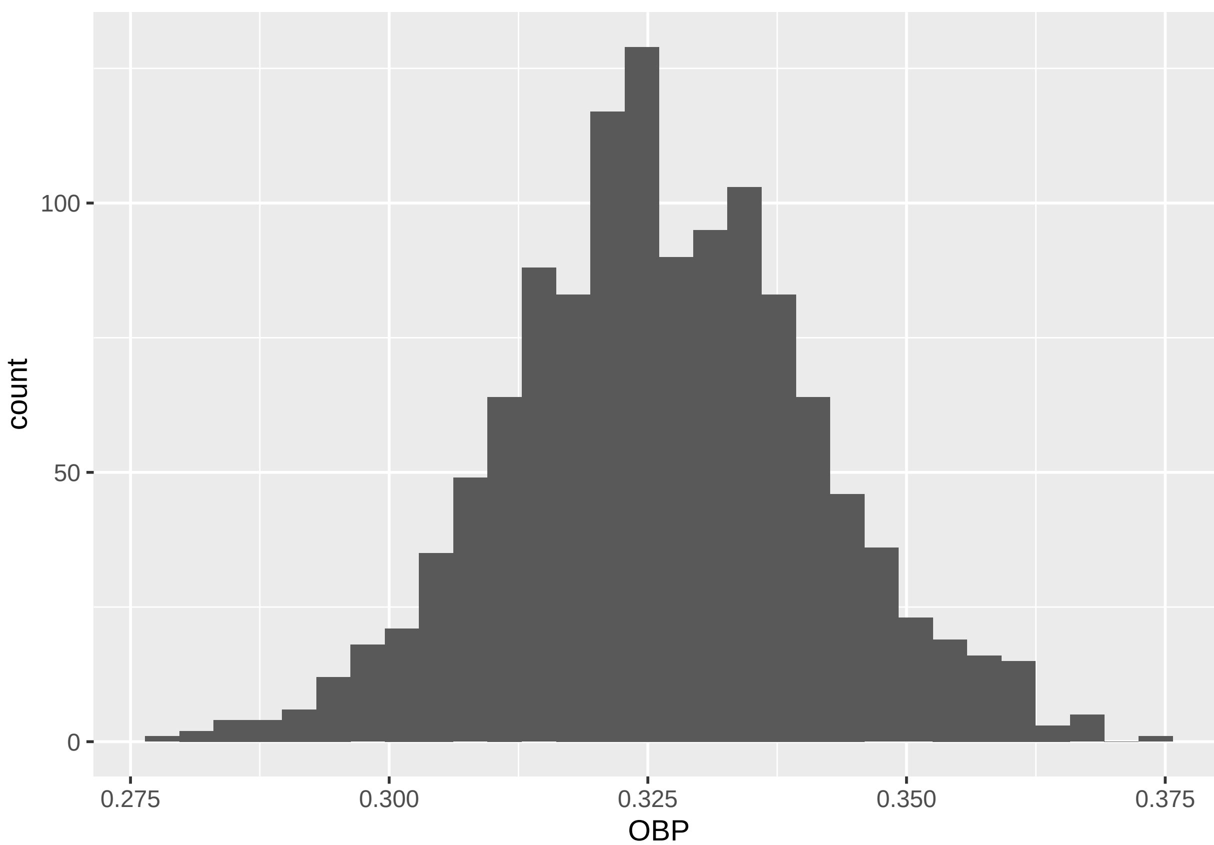
```
##
## -- Column specification --------------------------------------------------------
```

```
## cols(
##   Team = col_character(),
##   League = col_character(),
##   Year = col_double(),
##   RS = col_double(),
##   RA = col_double(),
##   W = col_double(),
##   OBP = col_double(),
##   SLG = col_double(),
##   BA = col_double(),
##   Playoffs = col_double(),
##   RankSeason = col_double(),
##   RankPlayoffs = col_double(),
##   G = col_double(),
##   OOBP = col_double(),
##   OSLG = col_double()
## )
```

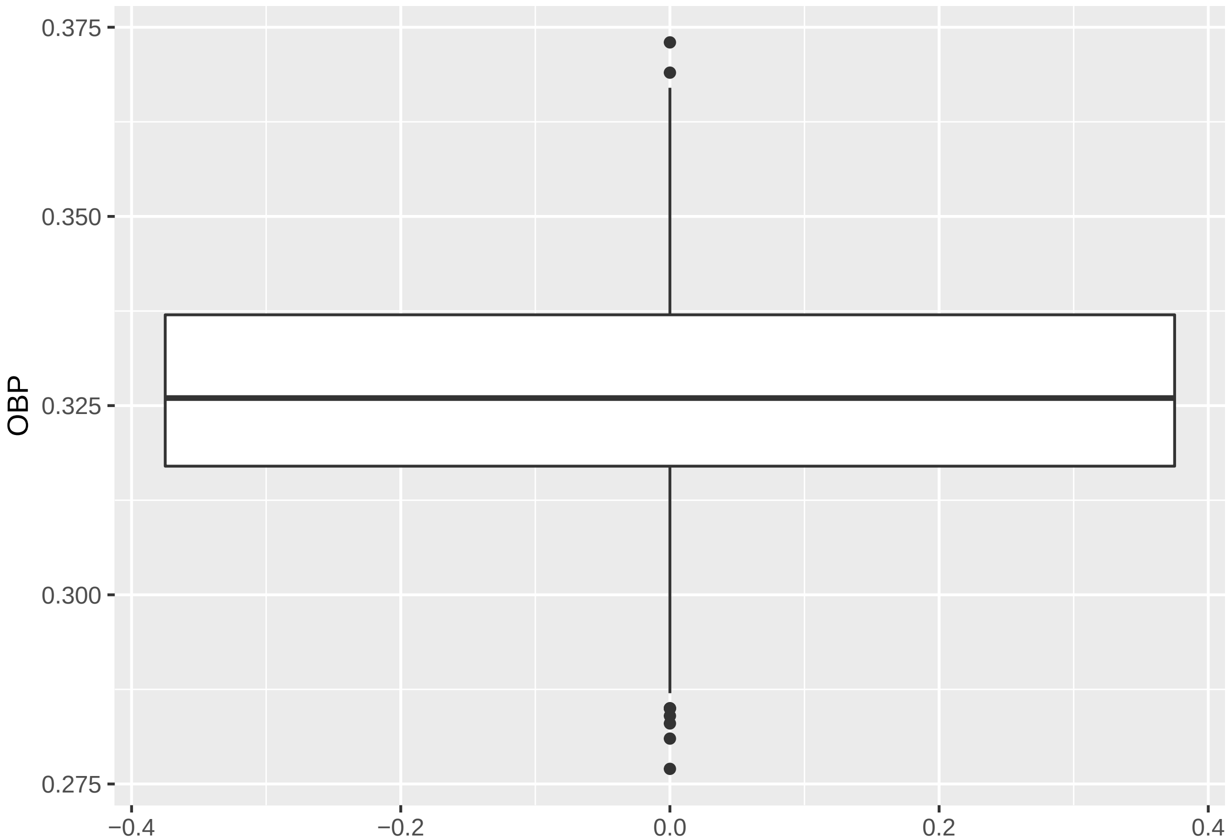
Q4 Part I Plot histogram and boxplots for OBP,SLG,BA OBP :histogram , boxplots, mean ,median

```r
ggplot(bdata, aes(x = OBP)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```r
ggplot(bdata, aes(y = OBP)) + geom_boxplot()
```




```r
mean(bdata$OBP)
```

```
## [1] 0.3263312
```

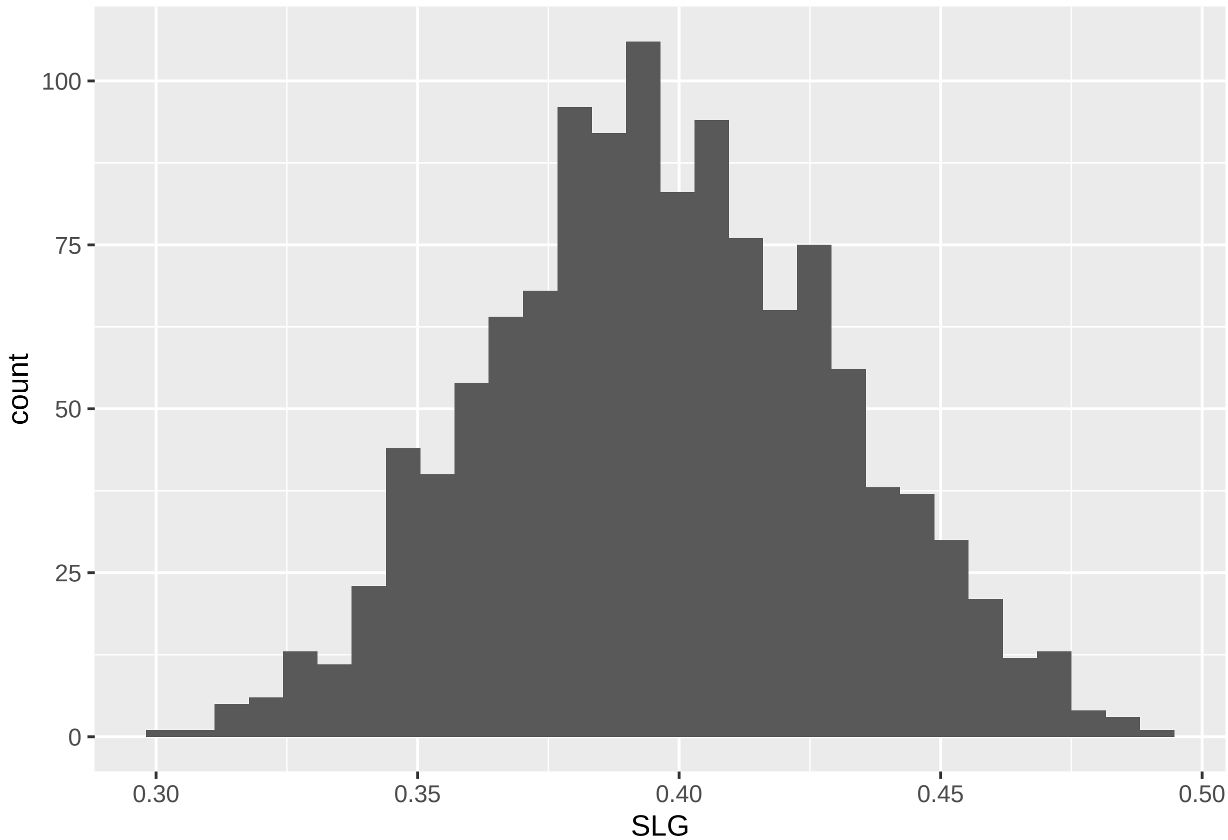
```r
median(bdata$OBP)
```

```
## [1] 0.326
```

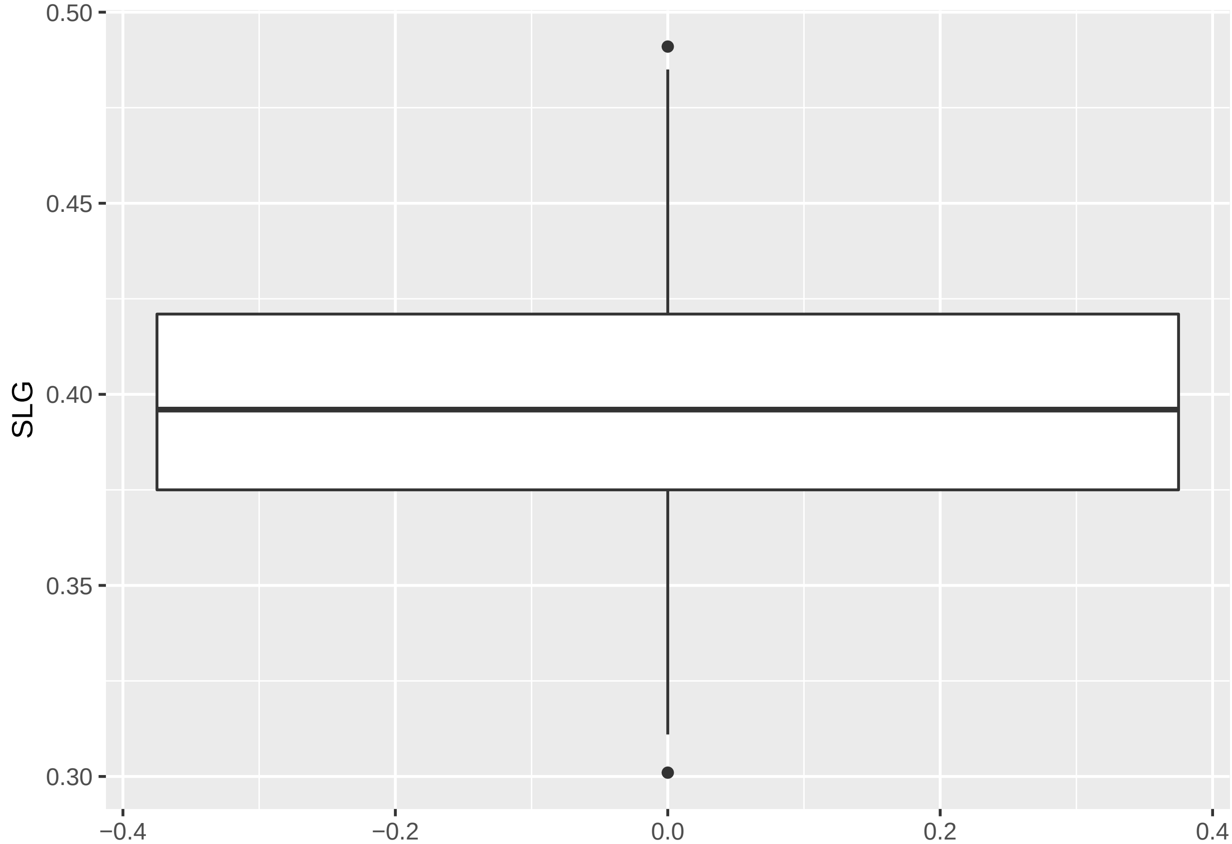
SLG :histogram , boxplots, mean ,median

```r
ggplot(bdata, aes(x = SLG)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




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




```
mean(bdata$SLG)
```

```
## [1] 0.3973417
```

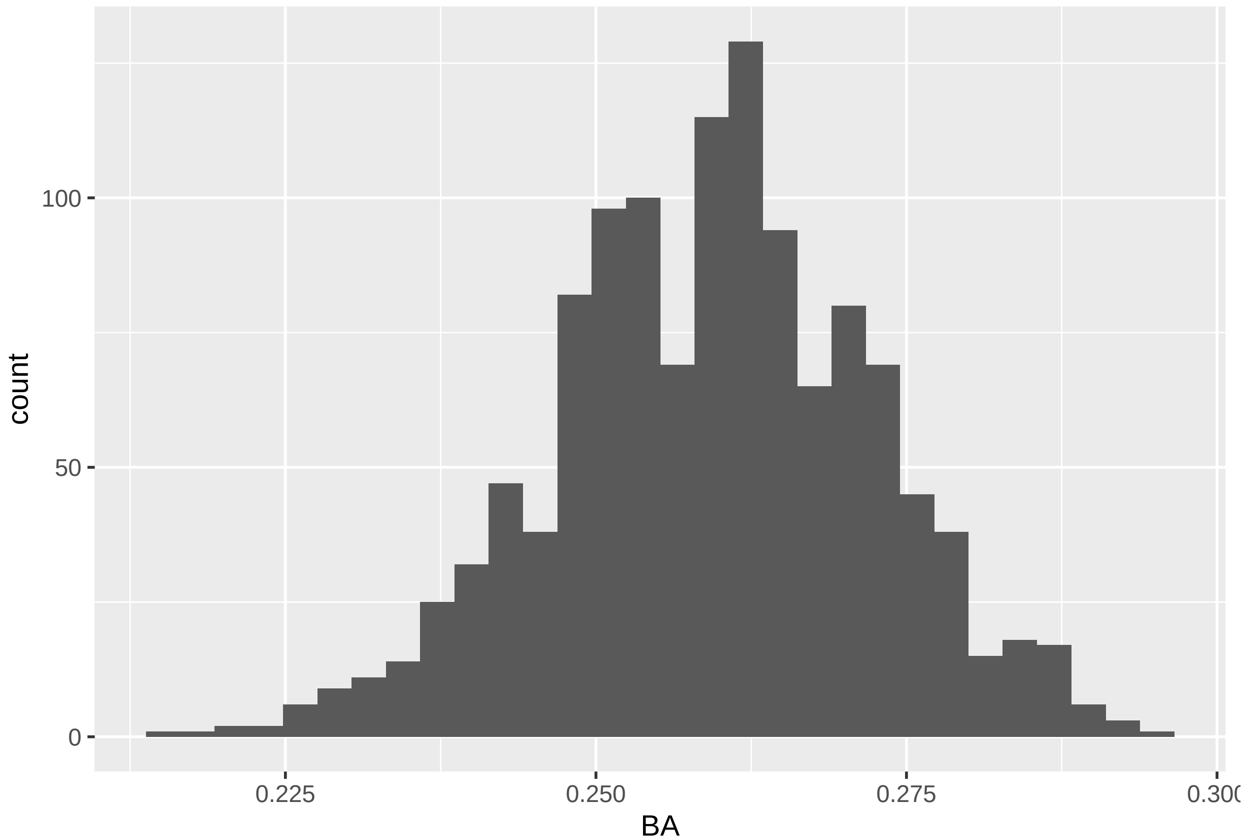
```
median(bdata$SLG)
```

```
## [1] 0.396
```

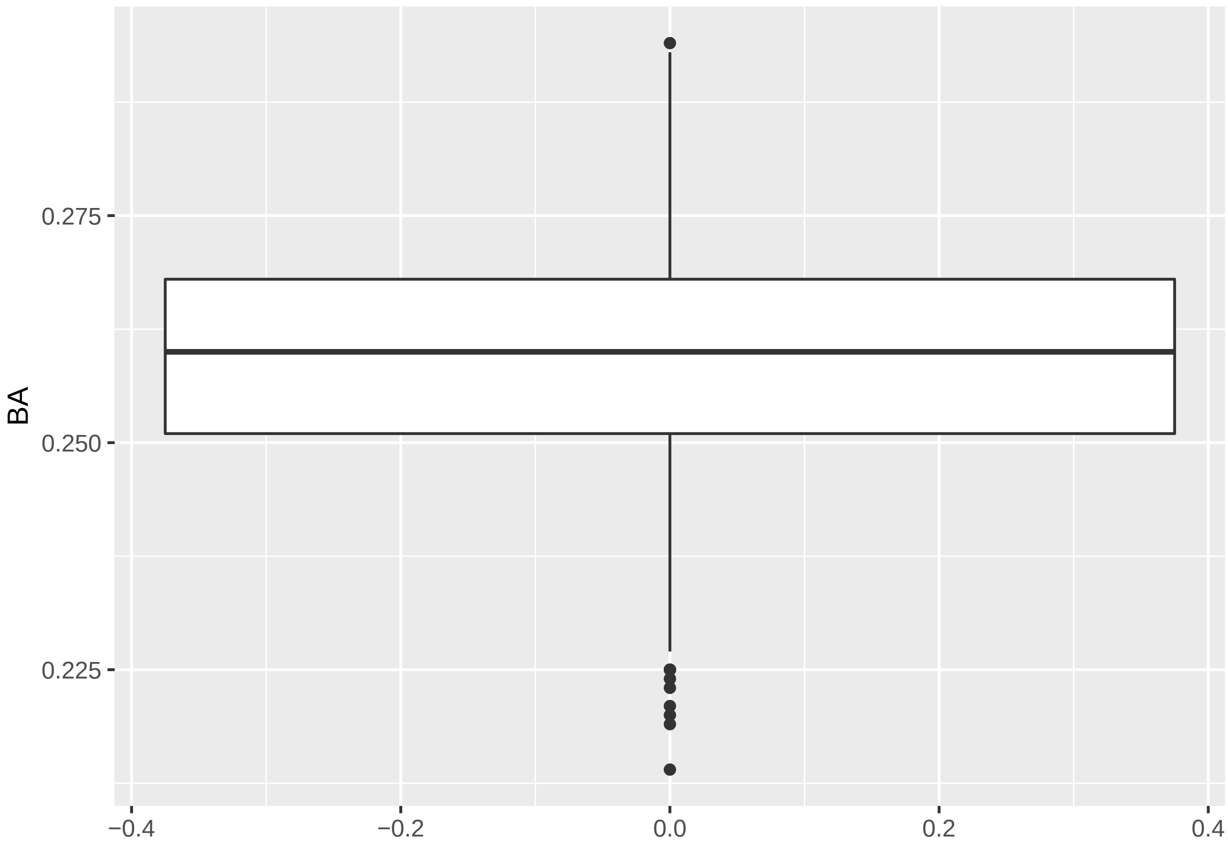
BA :histogram , boxplots, mean ,median

```
ggplot(bdata, aes(x = BA)) + geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```r
ggplot(bdata, aes(y = BA)) + geom_boxplot()
```




```r
mean(bdata$BA)
```

```
## [1] 0.2592727
```

```r
median(bdata$BA)
```

```
## [1] 0.26
```

Q4 Part I End

Q4 Part II Marginal Regression Analysis Marginally regress RS on BA, OBP ,SLG Give the scatter plot and the fitted line. Report the coefficient value and Rsq Give the QQ-plot of the fitted residual

Analysis for RS on BA:

```r
# library(MASS)
# Regression Model
RS_BA <-  lm(RS~BA, data=bdata)
# Rsq and coefficient value
summary(RS_BA)$r.squared
```

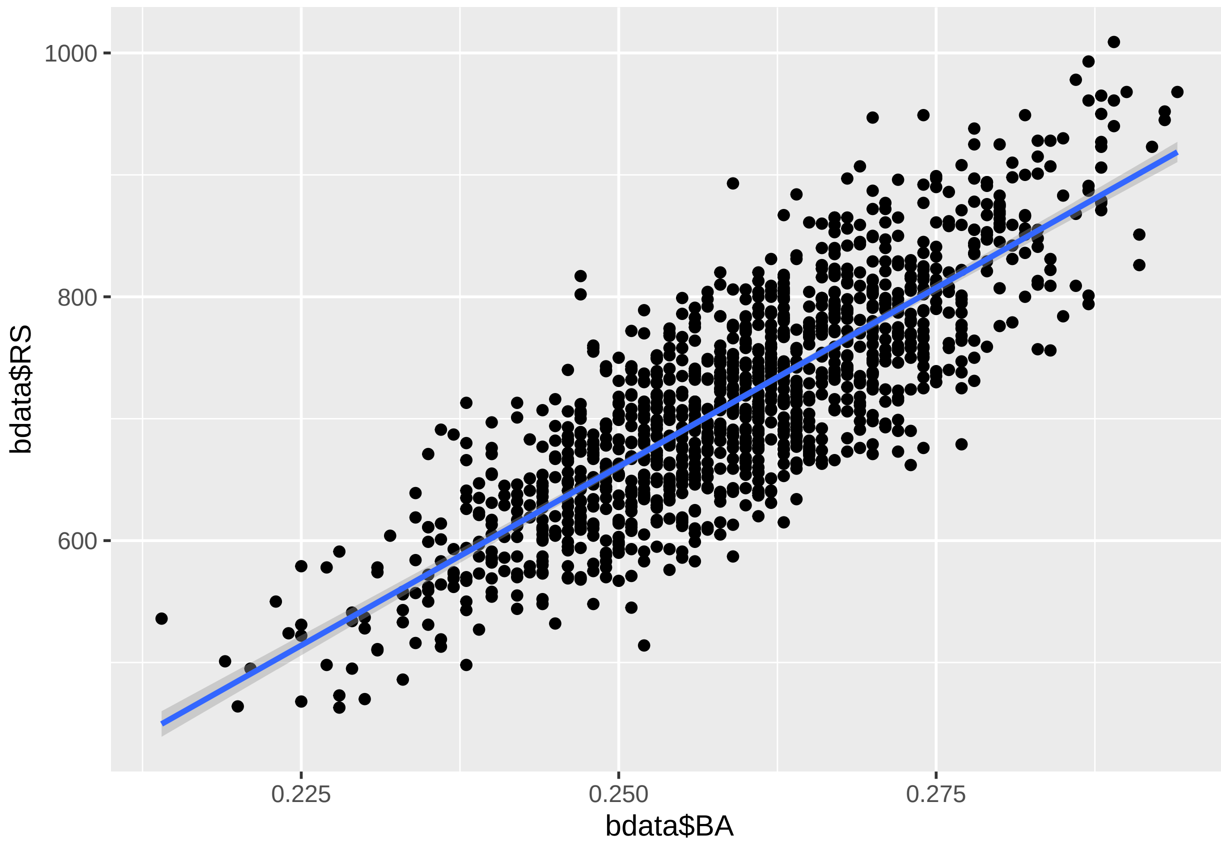
```
## [1] 0.6839284
```

```r
lm(RS~BA, data=bdata)
```

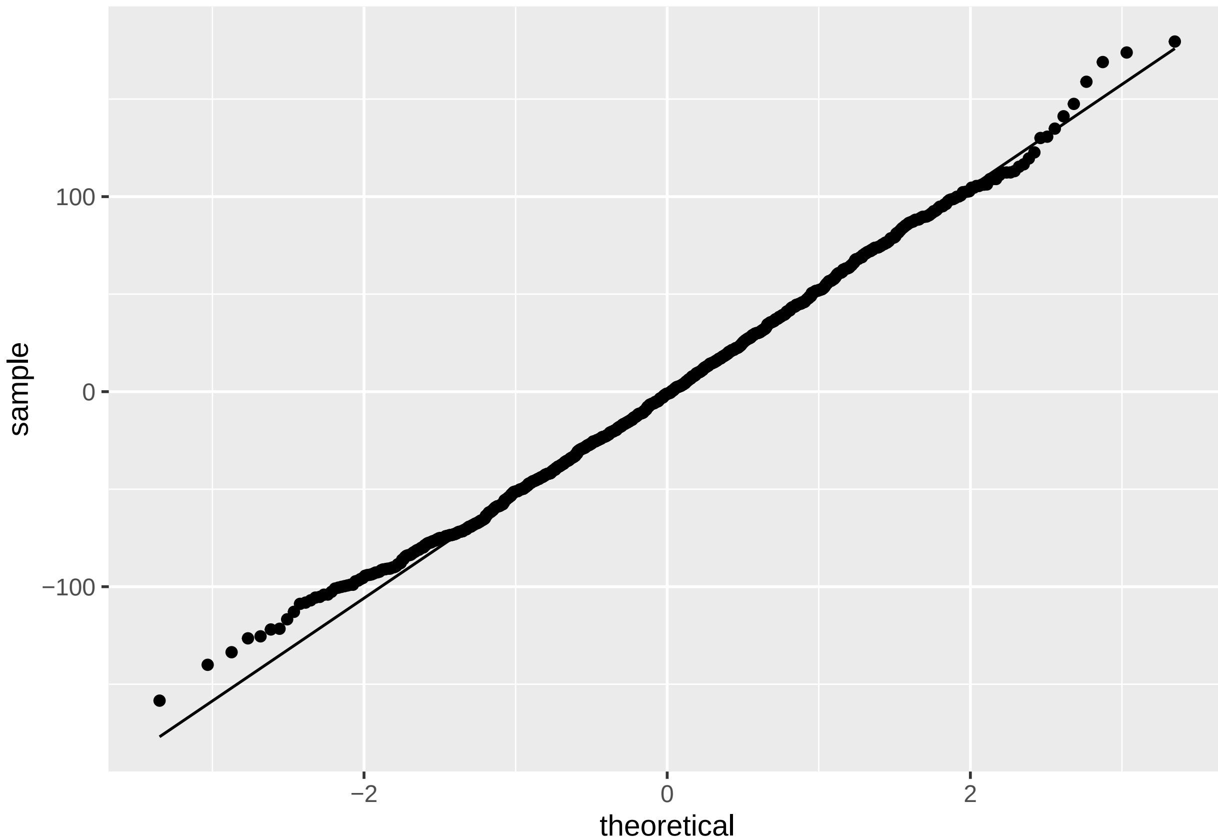
```
##
## Call:
## lm(formula = RS ~ BA, data = bdata)
##
## Coefficients:
## (Intercept)           BA
##      -805.5       5864.8
```

```r
# Scatter plot and fitted line
ggplot(bdata,aes(x=bdata$BA, y = bdata$RS)) + geom_point() + geom_smooth(method = lm)
```

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




```r
# QQ-plot of the fitted residual
ggplot(bdata, aes(sample = RS_BA$residuals)) + stat_qq() + stat_qq_line()
```



The intercept and slope are -805.5 and 5864.8 respectively, and R_squared = 0.6839284.

Analysis for RS on OBP:

```r
# Regression Model
RS_OBP <-  lm(RS~OBP, data=bdata)
# Rsq and coefficient value
summary(RS_OBP)$r.squared
```

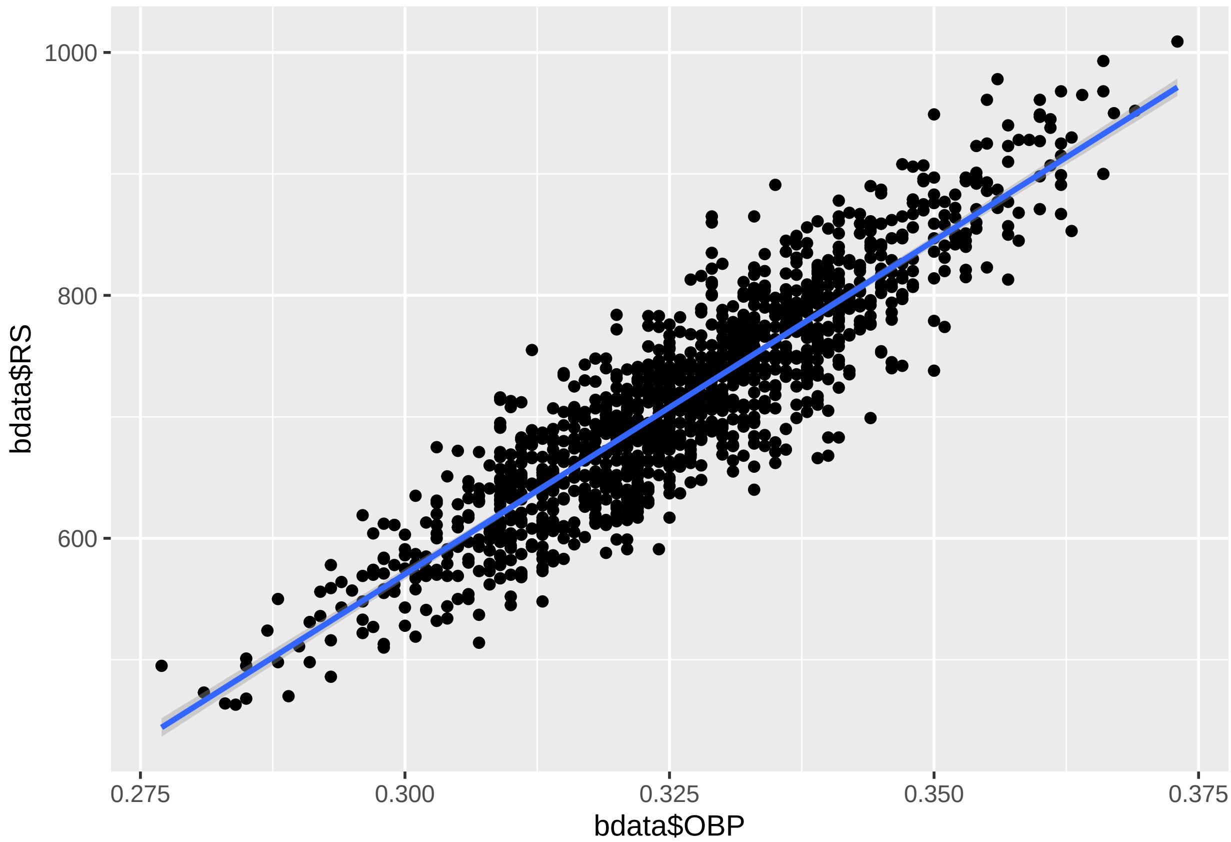
```
## [1] 0.8108862
```

```r
lm(RS~OBP, data=bdata)
```

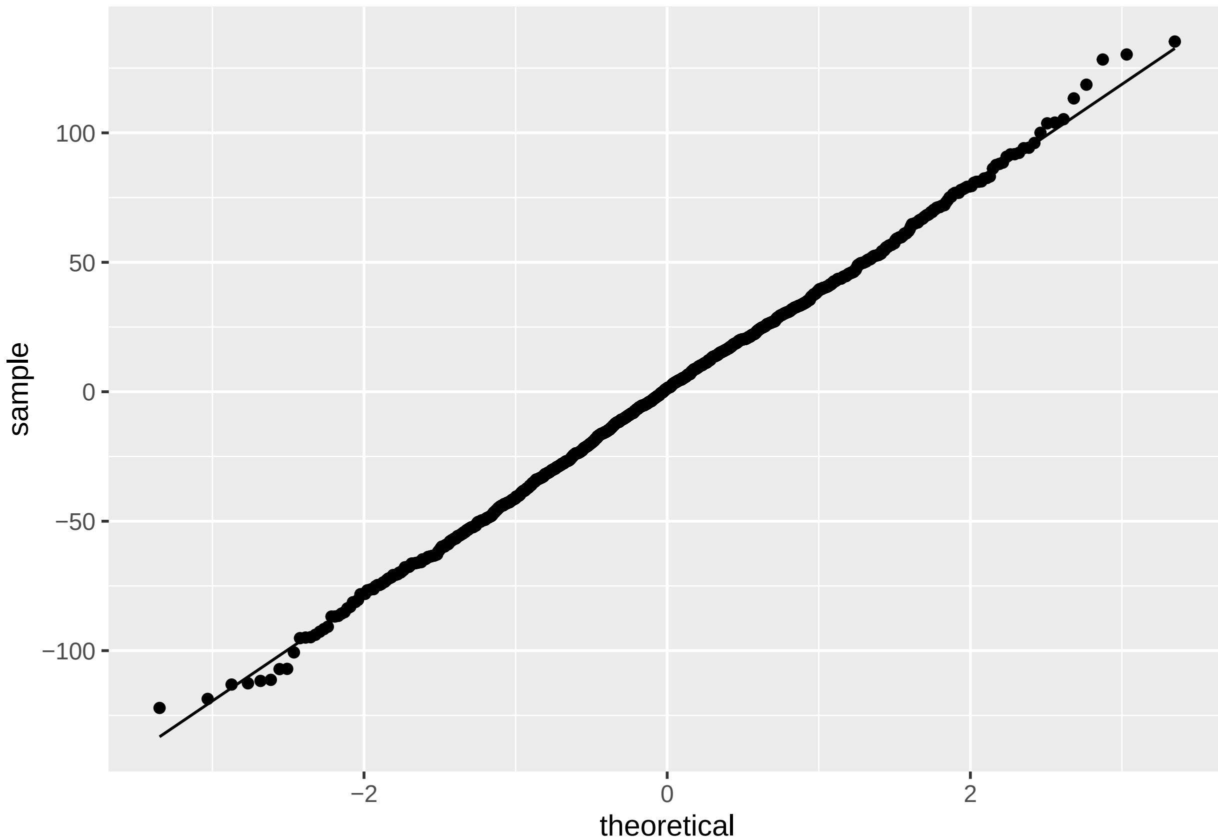
```
##
## Call:
## lm(formula = RS ~ OBP, data = bdata)
##
## Coefficients:
## (Intercept)          OBP
##       -1077         5490
```

```r
# Scatter plot and fitted line
ggplot(bdata,aes(x=bdata$OBP, y = bdata$RS)) + geom_point() + geom_smooth(method = lm)
```

```
## ‘geom_smooth()‘ using formula ’y ~ x’
```




```
# QQ-plot of the fitted residual
ggplot(bdata, aes(sample = RS_OBP$residuals)) + stat_qq() + stat_qq_line()
```



The intercept and slope are -1077 and 5490 respectively, and R_squared = 0.8108862

Analysis for RS on SLG:

```
# Regression Model
RS_SLG <-  lm(RS~SLG, data=bdata)
# Rsq and coefficient value
summary(RS_SLG)$r.squared
```

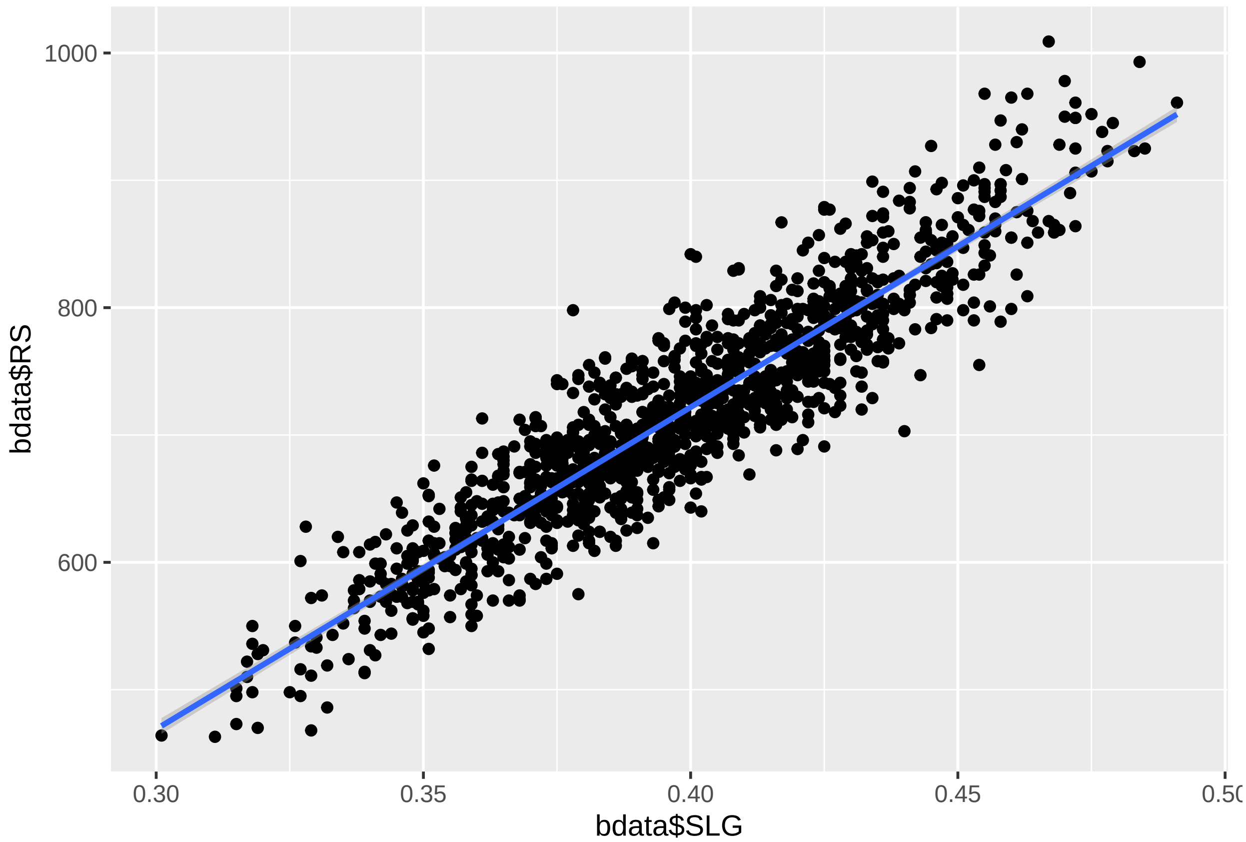
```
## [1] 0.8440831
```

```
lm(RS~SLG, data=bdata)
```

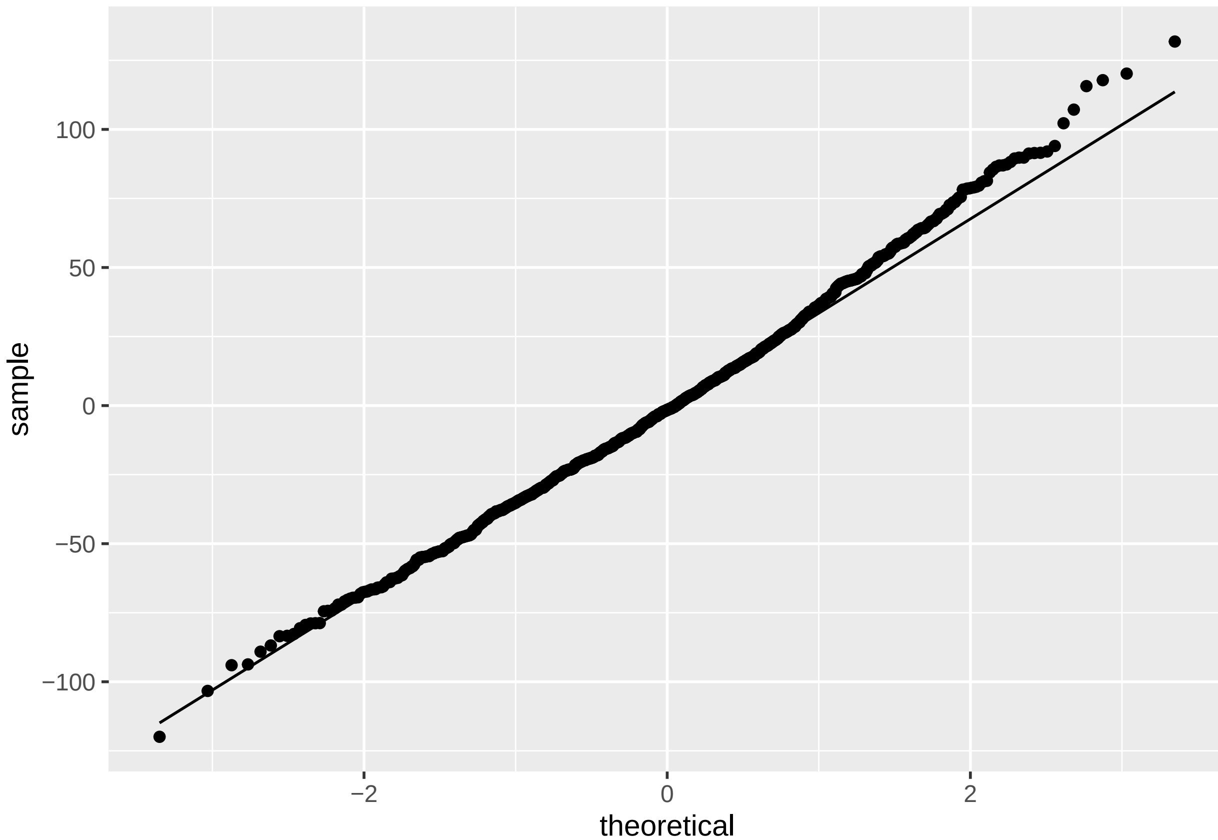
```
##
## Call:
## lm(formula = RS ~ SLG, data = bdata)
##
## Coefficients:
## (Intercept)          SLG
##      -289.4       2527.9
```

```
# Scatter plot and fitted line
ggplot(bdata,aes(x=bdata$SLG, y = bdata$RS)) + geom_point() + geom_smooth(method = lm)
```

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




```r
# QQ-plot of the fitted residual
ggplot(bdata, aes(sample = RS_SLG$residuals)) + stat_qq() + stat_qq_line()
```



The intercept and slope are -289.4 and 2527.9 respectively, and R_squared = 0.8440831 # Q4 Part II End

Q4 Part III Mutiple Regression Analysis. Fit the model RS~ BA+SLG+OBP Report the estimated coefficients for these covariates Check the model by giving QQ plots of the residuals

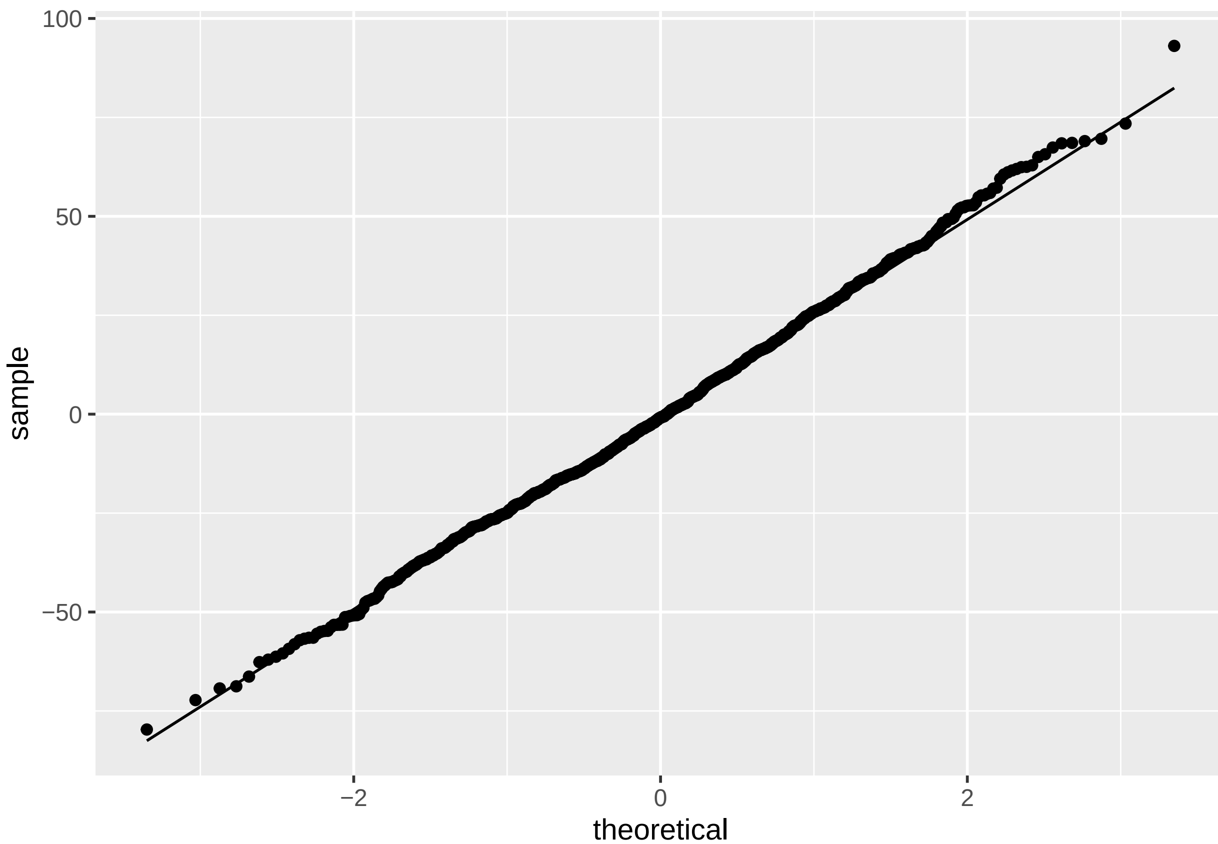
```
RS_BA_SLG_OBP <- lm(RS~BA + SLG + OBP, data=bdata)
summary(RS_BA_SLG_OBP)
```

```
##
## Call:
## lm(formula = RS ~ BA + SLG + OBP, data = bdata)
##
## 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
```

```r
# estimated coefficients
lm(RS~BA + SLG + OBP, data=bdata)
```

```
##
## Call:
## lm(formula = RS ~ BA + SLG + OBP, data = bdata)
##
## Coefficients:
## (Intercept)           BA          SLG          OBP
##      -806.1       -134.9       1533.9       2900.9
```

```r
# QQ plot
ggplot(bdata, aes(sample = RS_BA_SLG_OBP$residuals)) + stat_qq() + stat_qq_line()
```



The intercept is -806.1 , and slope for BA, SLG and OBP are -134.9, 1533.9 and 2900.9 respectively. The fitting result is considerbly consistent with fitted coefficient of BA in part II, the bottom and top part of the plot is slightly differnet.

Fit model RS~BA + SLG Compare R-squared of two models

```r
RS_BA_SLG <- lm(RS~BA + SLG, data=bdata)
summary(RS_BA_SLG)
```

```
##
## Call:
## lm(formula = RS ~ BA + SLG, data = bdata)
```

```
##
## 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
```

The R-squared is 0.8711, and R-squared for previous model is 0.9249, I would prefer the previous model, since its R-squared is higher. # Q4 Part III End

Q4 Part IV

```
oakland_2002 <- bdata %>% filter(Year == 2002, Team == "OAK")
oakland_2002
```

```
## # A tibble: 1 x 15
##   Team  League  Year    RS    RA     W   OBP   SLG    BA Playoffs RankSeason
##   <chr> <chr>  <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>    <dbl>      <dbl>
## 1 OAK   AL      2002   800   654   103 0.339 0.432 0.261        1          1
## # ... with 4 more variables: RankPlayoffs <dbl>, G <dbl>, OOBP <dbl>,
## #   OSLG <dbl>
```

```
RD <- bdata$RS - bdata$RA
bdata$RD <- RD
history_2001 <- bdata %>% filter(Year < 2002)

# W ~ RD
W_RD <- lm(W~RD , data=history_2001)
W_RD
```

```
##
## Call:
## lm(formula = W ~ RD, data = history_2001)
##
## Coefficients:
## (Intercept)           RD
##     80.8814       0.1058
```

```
# RS~OBP +SLG
lm_RS_OBP_SLG <- lm(RS~OBP +SLG , data=history_2001)
lm_RS_OBP_SLG
```

```
##
## Call:
## lm(formula = RS ~ OBP + SLG, data = history_2001)
##
## Coefficients:
## (Intercept)          OBP          SLG
##      -804.6       2737.8       1584.9
```

```r
# RA ~ OOBP + OSLG
lm_RA_OOBP_OSLG <- lm(RA ~ OOBP + OSLG , data=history_2001)
lm_RA_OOBP_OSLG
```

```
##
## Call:
## lm(formula = RA ~ OOBP + OSLG, data = history_2001)
##
## Coefficients:
## (Intercept)         OOBP         OSLG
##      -837.4       2913.6       1514.3
```

```r
# predict
predict_RS <- -804.6 + 2737.8 * 0.349 + 1584.9 * 0.430
predict_RA <- -837.4 + 2913.6 * 0.307 + 1514.3 * 0.373
# RD = RS-RA
predict_RD <- predict_RS - predict_RA
predict_W <- 80.8814 + 0.1058*predict_RD
predict_W
```

```
## [1] 103.1513
```

Our prediction is accurate, Oakland won 103 games in 2002.

Q4 Part IV End