use-R analysis

```
library(readr)
StudentsPerformance <- read_csv("StudentsPerformance.csv")
```

```
## Rows: 1000 Columns: 8

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): gender, race/ethnicity, parental level of education, lunch, test pr...
## dbl (3): math score, reading score, writing score

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(StudentsPerformance)
```

```
## # A tibble: 6 x 8
##   gender `race/ethnicity` `parental level ~ lunch  `test preparati~ `math score`
##   <chr>  <chr>            <chr>             <chr>  <chr>                   <dbl>
## 1 female group B          bachelor's degree stand~ none                       72
## 2 female group C          some college      stand~ completed                  69
## 3 female group B          master's degree   stand~ none                       90
## 4 male   group A          associate's degr~ free/~ none                       47
## 5 male   group C          some college      stand~ none                       76
## 6 female group B          associate's degr~ stand~ none                       71
## # ... with 2 more variables: reading score <dbl>, writing score <dbl>
```

###Explore the data

See the summary statistics of the data

```
summary(StudentsPerformance)
```

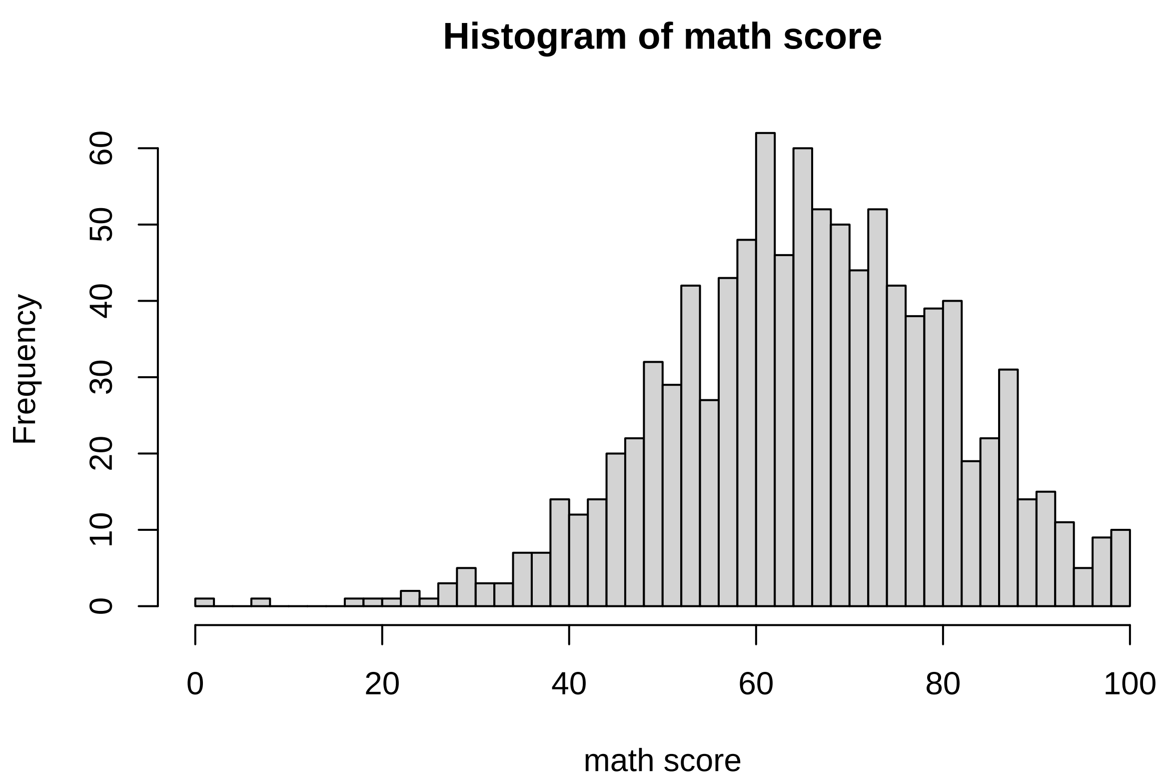
```
##     gender          race/ethnicity     parental level of education
##  Length:1000        Length:1000        Length:1000
##  Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character
##
##
##
##     lunch           test preparation course   math score     reading score
##  Length:1000        Length:1000              Min.   :  0.00   Min.   : 17.00
##  Class :character   Class :character         1st Qu.: 57.00   1st Qu.: 59.00
##  Mode  :character   Mode  :character         Median : 66.00   Median : 70.00
```

```
##                                                Mean   : 66.09   Mean   : 69.17
##                                                3rd Qu.: 77.00   3rd Qu.: 79.00
##                                                Max.   :100.00   Max.   :100.00
##  writing score
##  Min.   : 10.00
##  1st Qu.: 57.75
##  Median : 69.00
##  Mean   : 68.05
##  3rd Qu.: 79.00
##  Max.   :100.00
```

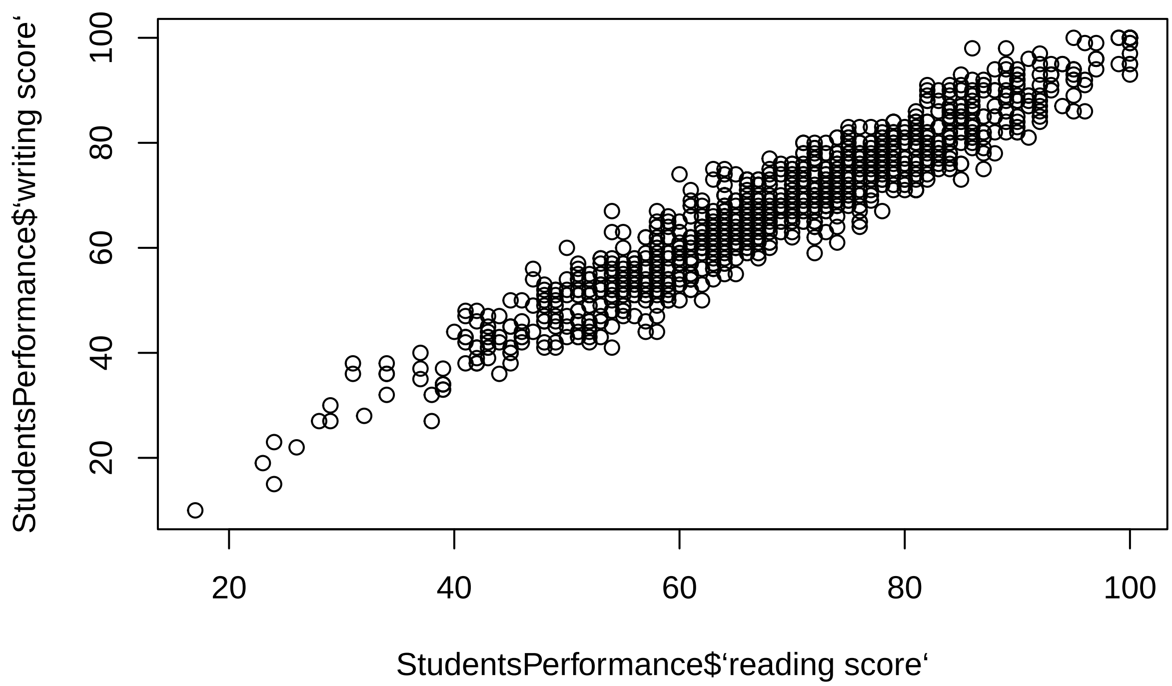
We can see that every columns have their own data, like the quantitative data,the 4 different score of their min,1st quantile median ,mean and max.

Visualization

```r
#Plot a histogram of math score
hist(StudentsPerformance$`math score`, breaks = 50, xlab = "math score", main = "Histogram of math scor
```




```r
#Plot a scatter plot
plot(x = StudentsPerformance$`reading score`, y = StudentsPerformance$`writing score`)
```

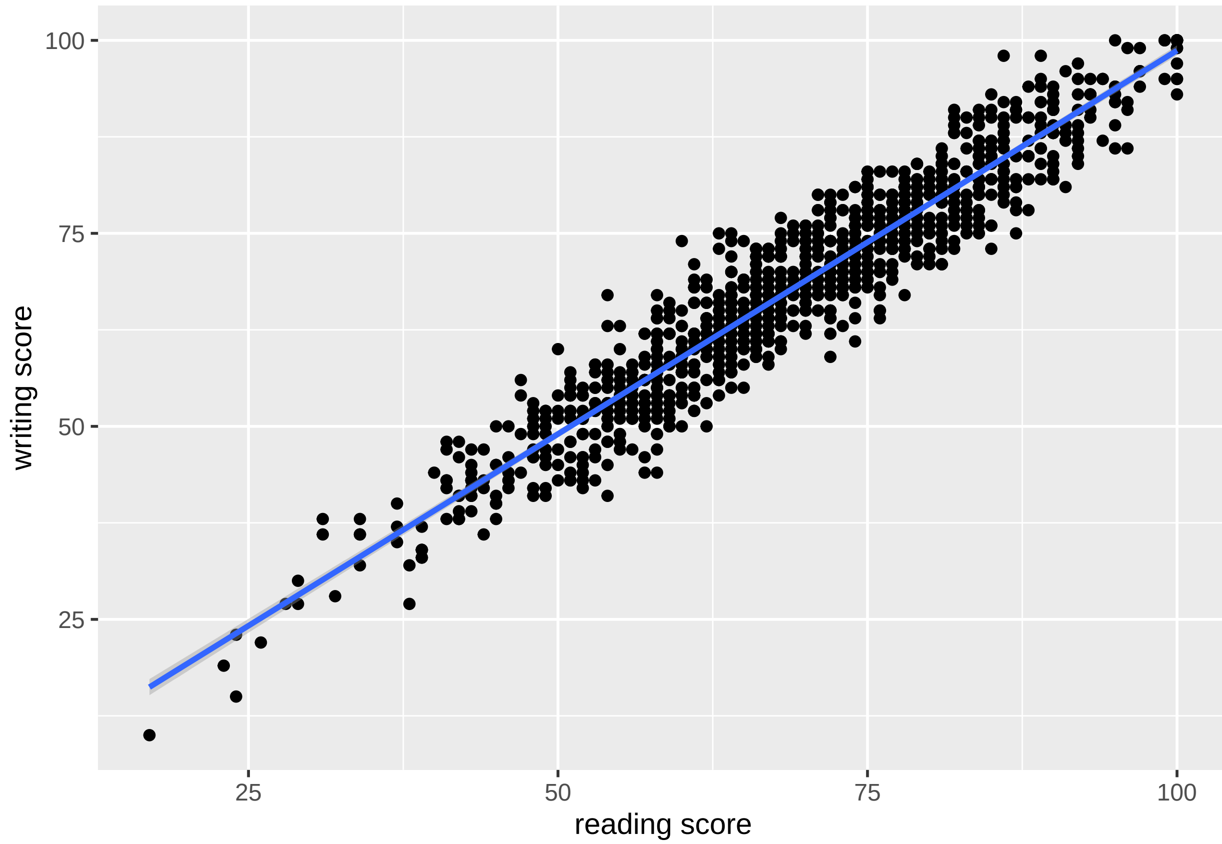


These plots are technically sound, but are not aesthetically pleasing. So we make more appealing visualizations using GGPLOT2 package.

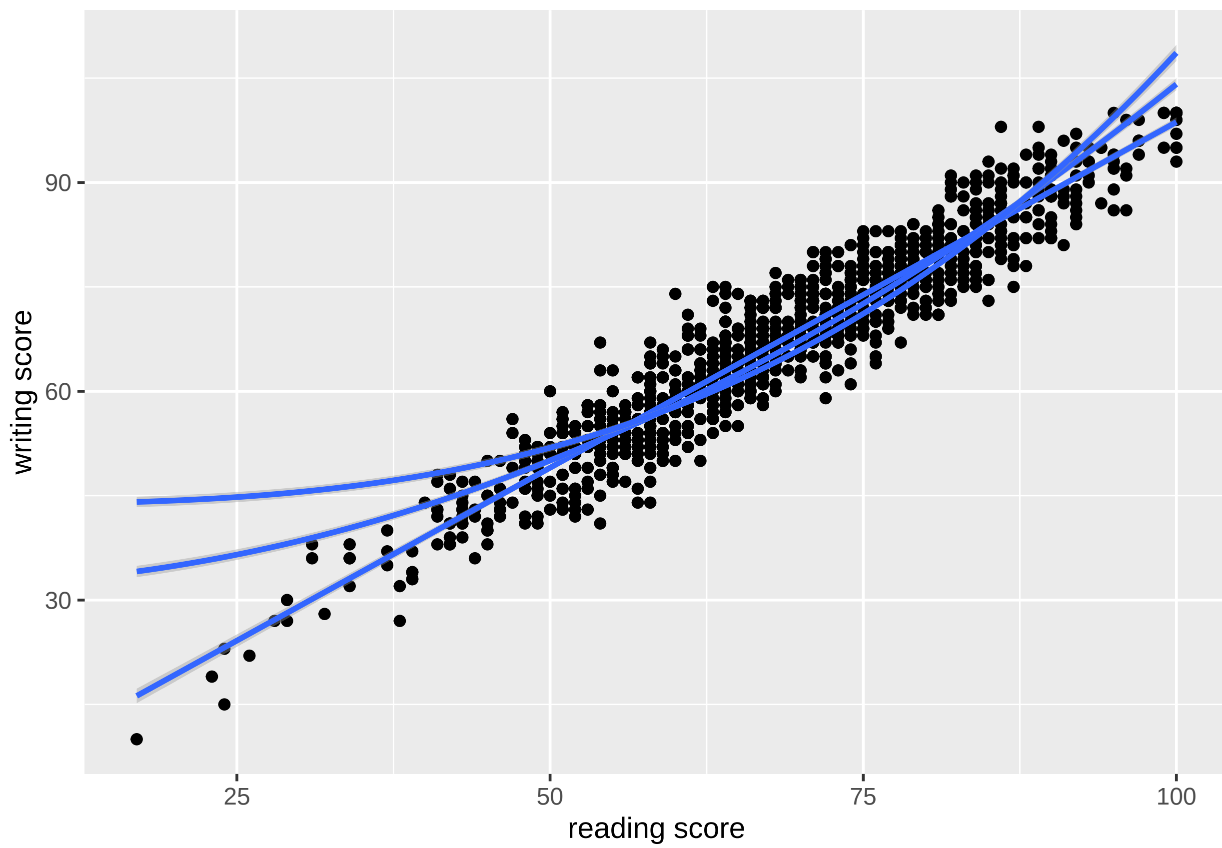
```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.5     v dplyr   1.0.7
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.1.4     v forcats 0.5.1
## v purrr   0.3.4

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

```
# Scatter plot with the best linear fit
plot <- ggplot(data = StudentsPerformance, aes(x = `reading score`,y = `writing score`)) + geom_point()
  geom_smooth(method='lm',formula=y~x) # define the plot
plot # see the plot
```




```
# The line doesn't seem to fit the dots very well
# Try quadratic and cubic transformation of the explanatory variable

plot + geom_smooth(method='lm',formula=y~I(x^2)) +
  geom_smooth(method='lm',formula=y~I(x^3))
```



t test - testing the differences between genders

```
# significant difference (p<0.05)
t.test(StudentsPerformance$`writing score`~StudentsPerformance$gender)
```

```
##
##  Welch Two Sample t-test
##
## data:  StudentsPerformance$‘writing score‘ by StudentsPerformance$gender
## t = 9.9977, df = 997.53, p-value < 2.2e-16
## alternative hypothesis: true difference in means between group female and group male is not equal to
## 95 percent confidence interval:
##   7.358849 10.953107
## sample estimates:
## mean in group female   mean in group male
##             72.46718             63.31120
```

```
t.test(StudentsPerformance$`reading score`~StudentsPerformance$gender)
```

```
##
##  Welch Two Sample t-test
##
## data:  StudentsPerformance$‘reading score‘ by StudentsPerformance$gender
```

```
## t = 7.9684, df = 996.36, p-value = 4.376e-15
## alternative hypothesis: true difference in means between group female and group male is not equal to
## 95 percent confidence interval:
##  5.377941 8.892218
## sample estimates:
## mean in group female   mean in group male
##             72.60811             65.47303
```

Define the model

First is the null model Null models is a model that contains only the dependent variable and an intercept (mean)

```r
n0 <- lm(formula =`math score`~1,data = StudentsPerformance) # fit the model
summary(n0)
```

```
##
## Call:
## lm(formula = `math score` ~ 1, data = StudentsPerformance)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -66.089  -9.089  -0.089  10.911  33.911
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  66.0890     0.4795   137.8   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.16 on 999 degrees of freedom
```

```r
# Note that the expected value is actually the mean and that the residual error is actually the
# standard deviation
mean(StudentsPerformance$`math score`);sd(StudentsPerformance$`math score`)
```

```
## [1] 66.089
```

```
## [1] 15.16308
```

```r
# We can use other variables to improve the model and reduce errors
```

Define a simple linear model with one explanatory variable

```r
#Basic model
m1 <- lm(formula = `math score` ~ `writing score`,data = StudentsPerformance)
summary(m1)
```

```
##
## Call:
## lm(formula = ‘math score‘ ~ ‘writing score‘, data = StudentsPerformance)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -24.8467  -6.4600   0.1464   6.4356  25.5515
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)     11.58310    1.31369   8.817   <2e-16 ***
## ‘writing score‘  0.80092    0.01884  42.511   <2e-16 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Residual standard error: 9.049 on 998 degrees of freedom
## Multiple R-squared:  0.6442, Adjusted R-squared:  0.6439
## F-statistic:  1807 on 1 and 998 DF,  p-value: < 2.2e-16
```

Note that the intercept predicts math score when writing score is 0 The slope measures expected change in weight for 1 unit change in height We need to center explanatory variables to achieve a more meaningful coefficient

Note that slope is actually correlation coefficient adjusted for relative dispersion of the two variables

```
cor(StudentsPerformance$`writing score`,StudentsPerformance$`math score`)*sd(StudentsPerformance$`math s
```

```
## [1] 0.8009213
```

Model with writing squared and cubed

```
math<- StudentsPerformance$`math score`
writing<-StudentsPerformance$`writing score`


m2 <- lm(formula = math~I((writing/100)^2),data = StudentsPerformance)
summary(m2)
```

```
##
## Call:
## lm(formula = math ~ I((writing/100)^2), data = StudentsPerformance)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -37.982  -6.486  -0.285   6.432  26.855
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         37.3918     0.7669   48.76   <2e-16 ***
## I((writing/100)^2)  59.0232     1.4560   40.54   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.325 on 998 degrees of freedom
## Multiple R-squared:  0.6222, Adjusted R-squared:  0.6218
## F-statistic:  1643 on 1 and 998 DF,  p-value: < 2.2e-16
```

```r
m3 <- lm(formula = math~I((writing/100)^3),data = StudentsPerformance)
summary(m3)
```

```
##
## Call:
## lm(formula = math ~ I((writing/100)^3), data = StudentsPerformance)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.773  -6.278  -0.184   6.679  28.091
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)         46.7192     0.6008   77.76   <2e-16 ***
## I((writing/100)^3)  53.6167     1.4267   37.58   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9.762 on 998 degrees of freedom
## Multiple R-squared:  0.5859, Adjusted R-squared:  0.5855
## F-statistic:  1412 on 1 and 998 DF,  p-value: < 2.2e-16
```

```r
# Use information criteria to choose the best model
# (smaller values of AIC indicate better model)

AIC(m1,m2,m3)
```

```
##    df      AIC
## m1  3 7247.121
## m2  3 7307.288
## m3  3 7398.860
```

From above,we choose the model m1 with the smallest AIC.

Multiple regression model

```r
read <- StudentsPerformance$`reading score`
m4 <- lm(formula = math~writing+read,data = StudentsPerformance)
summary(m4)
```

```
##
## Call:
## lm(formula = math ~ writing + read, data = StudentsPerformance)
##
```

```
## Residuals:
##      Min       1Q   Median       3Q      Max 
## -23.8779  -6.1750   0.2693   6.0184  24.8727 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  7.52409    1.32823   5.665 1.93e-08 ***
## writing      0.24942    0.06057   4.118 4.14e-05 ***
## read         0.60129    0.06304   9.538  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 8.667 on 997 degrees of freedom
## Multiple R-squared:  0.674,  Adjusted R-squared:  0.6733 
## F-statistic:  1031 on 2 and 997 DF,  p-value: < 2.2e-16
```

```r
m5 <- lm(formula = math~writing*read,data = StudentsPerformance)
summary(m5)
```

```
## 
## Call:
## lm(formula = math ~ writing * read, data = StudentsPerformance)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -24.052  -6.254   0.230   6.009  24.705 
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   3.7614864  4.1071551   0.916  0.35997    
## writing       0.3076971  0.0853945   3.603  0.00033 ***
## read          0.6617578  0.0887452   7.457 1.92e-13 ***
## writing:read -0.0008916  0.0009210  -0.968  0.33321    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 8.667 on 996 degrees of freedom
## Multiple R-squared:  0.6743, Adjusted R-squared:  0.6733 
## F-statistic: 687.3 on 3 and 996 DF,  p-value: < 2.2e-16
```

```r
AIC(m4,m5)
```

```
##    df      AIC
## m4  4 7161.804
## m5  5 7162.863
```

We see from the AIC , the interaction of terms do not improve the model And we compare the models above

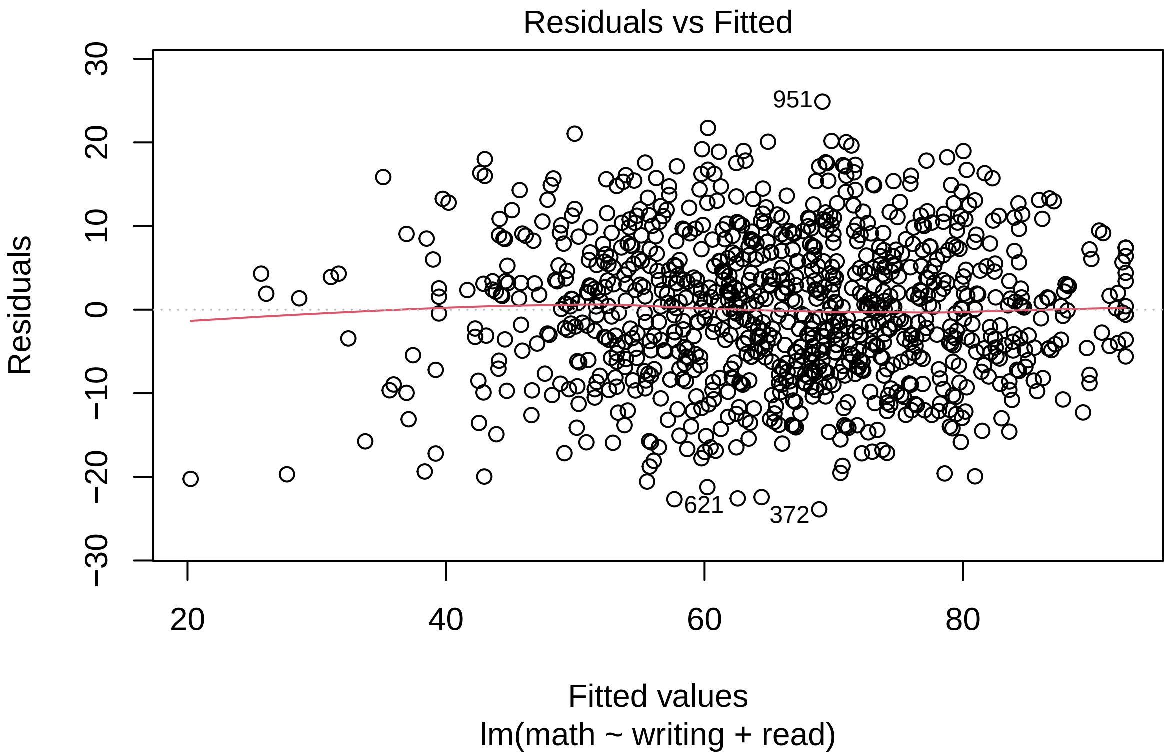
```r
AIC(m1,m2,m3,m4,m5)
```

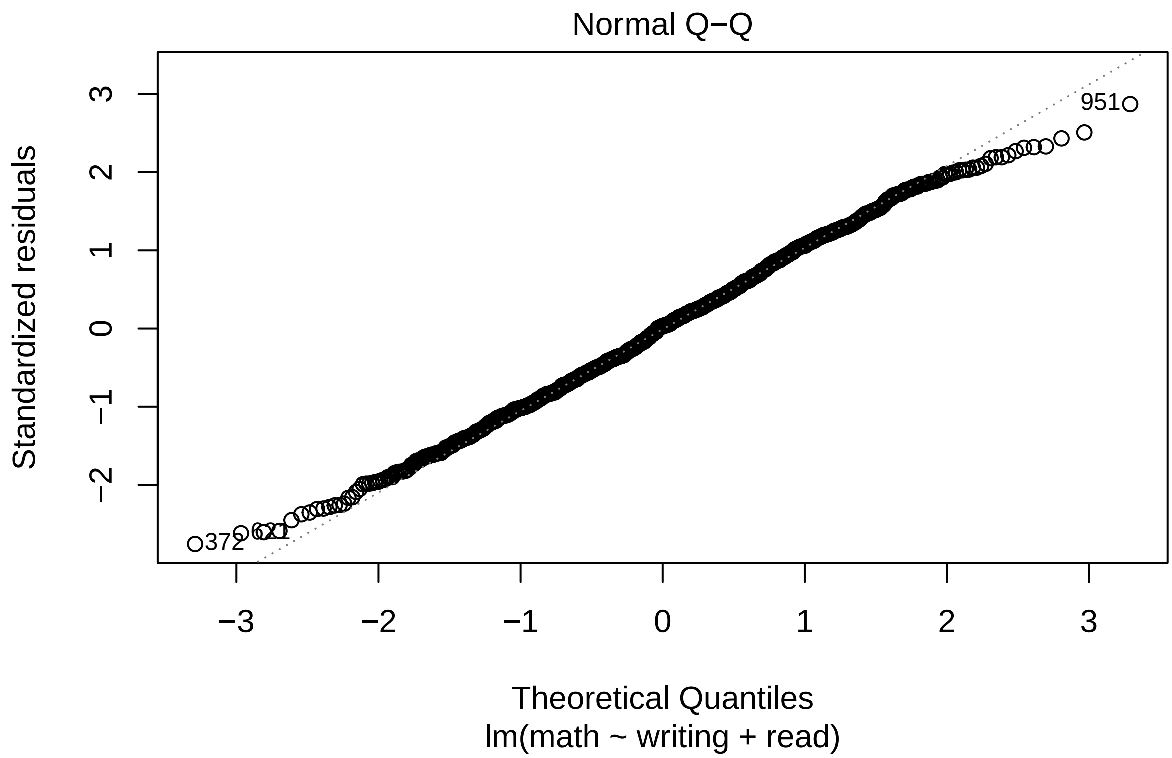
```
##    df      AIC
## m1  3 7247.121
```

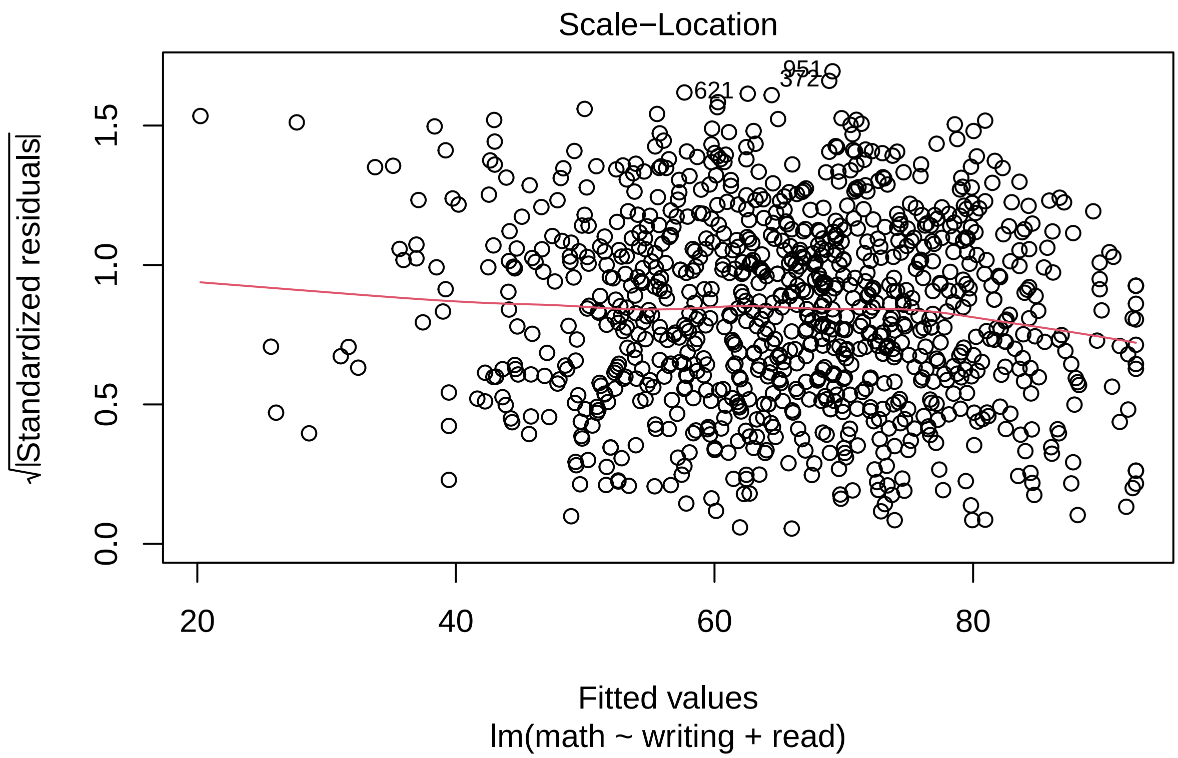
```
## m2  3 7307.288
## m3  3 7398.860
## m4  4 7161.804
## m5  5 7162.863
```

We can see that m4 has the smallest AIC value so the final model is m4 with #{math~writing+read}

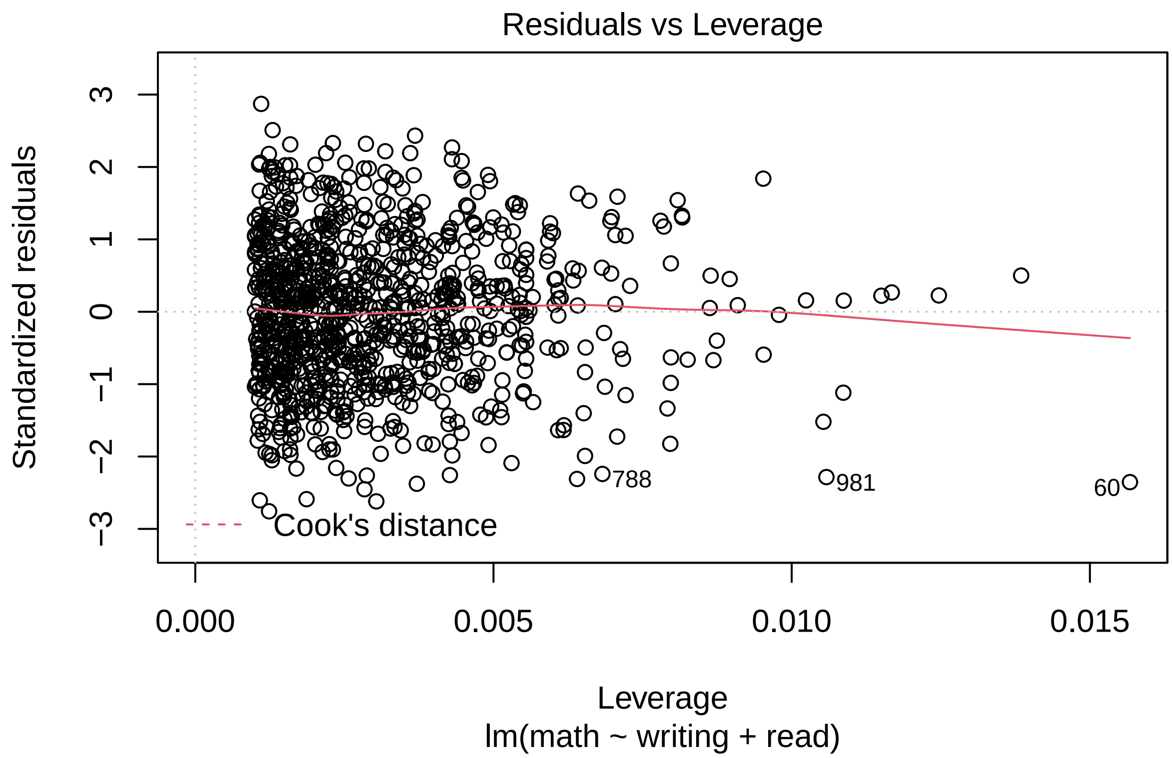
```
plot(m4)
```




Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(math ~ writing + read)
951
372
−3
−2
−1
0
1
2
3



Scale−Location
951
372
621
√|Standardized residuals|
0.0
0.5
1.0
1.5
20
40
60
80
Fitted values
lm(math ~ writing + read)




```
#plot 1: residual vs fitted :satisfy linear
#plot 2: qq plot : normal distribution
#plot 3: variance : averagely distributed between the red line
#plot 4: sr vs leverage
```

Making predictions with regressions

```
parent<- StudentsPerformance$`parental level of education`
predict(m4,newdata=data.frame(writing=79,read=60),
        type='response')
```

```
##        1
## 63.30597
```

```
predict(m4,newdata=data.frame(writing=79,read=60),
        interval=c('prediction'),level=0.95)
```

```
##        fit      lwr      upr
## 1 63.30597 46.12128 80.49066
```

```
predict(m4,newdata=data.frame(writing=79,read=60),
        interval=c('confidence'),level=0.95)
```

```
##        fit     lwr      upr
## 1 63.30597 60.8389 65.77304
```

The prediction interval is the range in which future `random` observation can be thought most likely to occur, whereas the confidence interval is where the `mean` of future observation is most likely to reside. The confidence interval is generally much more narrow than the prediction interval and its "narrowness" will increase with increasing numbers of observations, whereas the prediction interval will not decrease in width. So we can use the model 4 to predict the math score using the two variables which are reading and writing scores with the 95% confidence interval.

LOGISTIC REGRESSION

```
# Null model
maths <- math/100
logit0 <- glm(formula = maths~1,
              data = StudentsPerformance, family = binomial(link = "logit"))
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```
summary(logit0)
```

```
##
## Call:
## glm(formula = maths ~ 1, family = binomial(link = "logit"), data = StudentsPerformance)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -1.47067  -0.18855  -0.00188   0.23817   0.91013
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.6673     0.0668   9.989   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 110.97  on 999  degrees of freedom
## Residual deviance: 110.97  on 999  degrees of freedom
## AIC: 1030.5
##
## Number of Fisher Scoring iterations: 3
```

Model with explanatory variables

```
logit1 <- glm(formula = maths~read+writing,
              data = StudentsPerformance, family = binomial(link = "logit"))
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```r
summary(logit1)
```

```
##
## Call:
## glm(formula = maths ~ read + writing, family = binomial(link = "logit"),
##     data = StudentsPerformance)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.65461  -0.13904   0.00121   0.14195   0.60621
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.03051    0.34116  -5.952 2.65e-09 ***
## read         0.02858    0.01587   1.801   0.0717 .
## writing      0.01131    0.01519   0.744   0.4567
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 110.974  on 999  degrees of freedom
## Residual deviance:  40.497  on 997  degrees of freedom
## AIC: 860.42
##
## Number of Fisher Scoring iterations: 4
```

model with interactions

```r
logit2 <- glm(formula = maths~read*writing, data = StudentsPerformance, family = binomial(link = "logit
```

```
## Warning in eval(family$initialize): non-integer #successes in a binomial glm!
```

```r
summary(logit2)
```

```
##
## Call:
## glm(formula = maths ~ read * writing, family = binomial(link = "logit"),
##     data = StudentsPerformance)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.77944  -0.13677   0.00206   0.13762   0.60565
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -1.3761291  1.0666416  -1.290    0.197
```

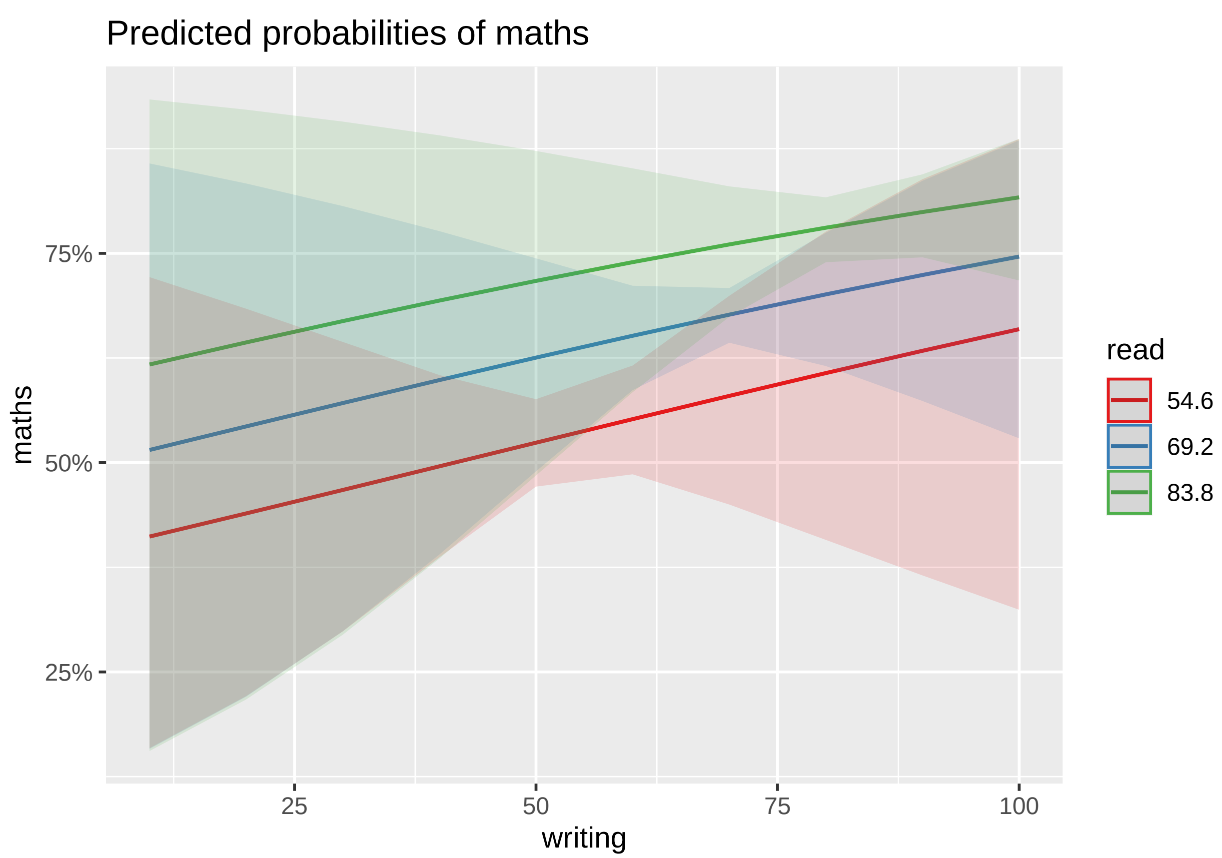
```
## read          0.0178686  0.0229666   0.778    0.437
## writing       0.0008882  0.0221516   0.040    0.968
## read:writing  0.0001629  0.0002533   0.643    0.520
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 110.974  on 999  degrees of freedom
## Residual deviance:  40.085  on 996  degrees of freedom
## AIC: 867.94
##
## Number of Fisher Scoring iterations: 4
```

```
predict(object = logit2,newdata = data.frame(writing=79,read=60), type = "response")
```

```
##         1
## 0.6313702
```

```
library(sjPlot)
plot_model(logit1, terms = c("writing","read"), type = "eff")
```

```
## Package 'effects' is not available, but needed for 'ggeffect()'. Either install package 'effects', o

## Data were 'prettified'. Consider using 'terms="writing [all]"' to get smooth plots.
```



```r
# Compare the models
```

```r
AIC(logit0,logit1,logit2)
```

```
##        df       AIC
## logit0  1 1030.5146
## logit1  3  860.4174
## logit2  4  867.9357
```

Obviously, logit1 model is meaningful.So we don't use the interact term.