Lab 3 - Linear regression

Load libraries

```
## Warning: package 'haven' was built under R version 3.5.2
```

```
## Warning: package 'synthpop' was built under R version 3.5.2
```

```
## Loading required package: lattice
```

```
## Loading required package: MASS
```

```
## Loading required package: nnet
```

```
## Loading required package: ggplot2
```

```
## ── Attaching packages ───────────────────────────────────────────────────────────────────────────
───────────────────── tidyverse 1.2.1 ──
```

```
## ✓ tibble  2.1.1       ✓ purrr   0.3.2
## ✓ tidyr   0.8.3       ✓ dplyr   0.8.0.1
## ✓ readr   1.3.1       ✓ stringr 1.4.0
## ✓ tibble  2.1.1       ✓ forcats 0.4.0
```

```
## Warning: package 'tibble' was built under R version 3.5.2
```

```
## Warning: package 'tidyr' was built under R version 3.5.2
```

```
## Warning: package 'purrr' was built under R version 3.5.2
```

```
## Warning: package 'dplyr' was built under R version 3.5.2
```

```
## Warning: package 'stringr' was built under R version 3.5.2
```

```
## Warning: package 'forcats' was built under R version 3.5.2
```

```
## ── Conflicts ──────────────────────────────────────────────────────────────────────────────────
─────────────── tidyverse_conflicts() ──
## ✕ dplyr::filter() masks stats::filter()
## ✕ dplyr::lag()    masks stats::lag()
## ✕ dplyr::select() masks MASS::select()
```

```
## Warning: package 'jtools' was built under R version 3.5.2
## Warning: package 'summarytools' was built under R version 3.5.2
```

```
##
## Attaching package: 'summarytools'
```

```
## The following object is masked from 'package:tibble':
##
##     view
```

```
## Warning: package 'kableExtra' was built under R version 3.5.2
```

```
##
## Attaching package: 'kableExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```
## Warning: package 'ggfortify' was built under R version 3.5.2
```

Load NALS data

```
nals <- read.csv("NALS.csv")
head(nals)
```

	id	annearn	occprest	sei	age	yearsed	gender	ln_earn
	<dbl>	<int>	<int>	<int>	<int>	<int>	<int>	<dbl>
1	1.02e+10	*NA*	49	44	44	16	1	5.446737
2	1.15e+10	18200	49	44	26	13	1	5.860786
3	3.10e+10	18000	49	44	49	13	1	5.993961
4	4.09e+10	13800	49	44	41	8	0	5.707110
5	5.42e+10	*NA*	49	44	32	13	1	*NA*
6	6.08e+10	11700	49	44	40	12	1	5.420535

6 rows | 1-9 of 15 columns

Today we are going to analyze linear regression in R, using the `NALS` data.

The variables we are going to be using in the analysis
- `X` = parent years of education
- `Z` = respondent years of education
- `Y` = adult literacy

Writing a data set

Using `mutate` function from `dplyr` we are going to create new variables as columns in the `nals` dataset. The variables we are going to need in this analysis - `id` , `parented` (X), `yearsed` (Z) and `literacy` (Y)

```
#  we can  assign this subsetted data into a new dataframe
# Renaming Variables in terms of X,Y and Z for the sake of our analysis.
nals <- nals %>% mutate(id=id, x=parented, z=yearsed, y=literacy)
#mutate creates new variables, transmute replaces variables
summarize(nals)
```

```
0 rows
```

```
# check the current column names
colnames(nals)
```

```
##  [1] "id"        "annearn"   "occprest"  "sei"       "age"
##  [6] "yearsed"   "gender"    "ln_earn"   "literacy"  "unemp"
## [11] "parented"  "Ethnicity" "Language"  "Education" "x"
## [16] "z"         "y"
```

summary statistics

```
summary(nals)
```

```
##        id               annearn          occprest         sei       
##  Min.   :1.010e+10   Min.   :     2   Min.   :17.0   Min.   :17.0  
##  1st Qu.:3.110e+10   1st Qu.: 12972   1st Qu.:33.0   1st Qu.:32.0  
##  Median :5.070e+10   Median : 20800   Median :43.0   Median :42.0  
##  Mean   :4.767e+10   Mean   : 26028   Mean   :43.8   Mean   :48.5  
##  3rd Qu.:6.650e+10   3rd Qu.: 33020   3rd Qu.:51.0   3rd Qu.:64.0  
##  Max.   :8.570e+10   Max.   :700024   Max.   :86.0   Max.   :97.0  
##                      NA's   :1313                                  
##       age           yearsed          gender          ln_earn     
##  Min.   :25.00   Min.   : 0.00   Min.   :0.0000   Min.   :1.099  
##  1st Qu.:31.00   1st Qu.:12.00   1st Qu.:0.0000   1st Qu.:5.595  
##  Median :38.00   Median :13.00   Median :0.0000   Median :6.054  
##  Mean   :38.97   Mean   :13.28   Mean   :0.4825   Mean   :6.025  
##  3rd Qu.:46.00   3rd Qu.:16.00   3rd Qu.:1.0000   3rd Qu.:6.479  
##  Max.   :59.00   Max.   :18.00   Max.   :1.0000   Max.   :9.508  
##                                                   NA's   :318    
##     literacy          unemp            parented       Ethnicity    
##  Min.   : 37.28   Min.   :0.00000   Min.   : 0.00   Min.   :1.000  
##  1st Qu.:254.64   1st Qu.:0.00000   1st Qu.: 9.00   1st Qu.:2.000  
##  Median :293.58   Median :0.00000   Median :12.00   Median :5.000  
##  Mean   :286.11   Mean   :0.08445   Mean   :10.93   Mean   :3.968  
##  3rd Qu.:327.93   3rd Qu.:0.00000   3rd Qu.:13.00   3rd Qu.:5.000  
##  Max.   :441.40   Max.   :1.00000   Max.   :18.00   Max.   :5.000  
##                                                                    
##     Language       Education           x               z        
##  Min.   :1.000   Min.   :1.000   Min.   : 0.00   Min.   : 0.00  
##  1st Qu.:5.000   1st Qu.:3.000   1st Qu.: 9.00   1st Qu.:12.00  
##  Median :5.000   Median :3.000   Median :12.00   Median :13.00  
##  Mean   :4.445   Mean   :3.469   Mean   :10.93   Mean   :13.28  
##  3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:13.00   3rd Qu.:16.00  
##  Max.   :5.000   Max.   :6.000   Max.   :18.00   Max.   :18.00  
##                                                                 
##        y         
##  Min.   : 37.28  
##  1st Qu.:254.64  
##  Median :293.58  
##  Mean   :286.11  
##  3rd Qu.:327.93  
##  Max.   :441.40  
## 
```

Graphical analysis

Before jumping in to the syntax of the linear model, lets try to understand these variables graphically. Typically, for each of the independent variables (predictors), the following plots are drawn to visualize the following behavior:

1. **Scatter plot**: Visualize the linear relationship between the predictor and response

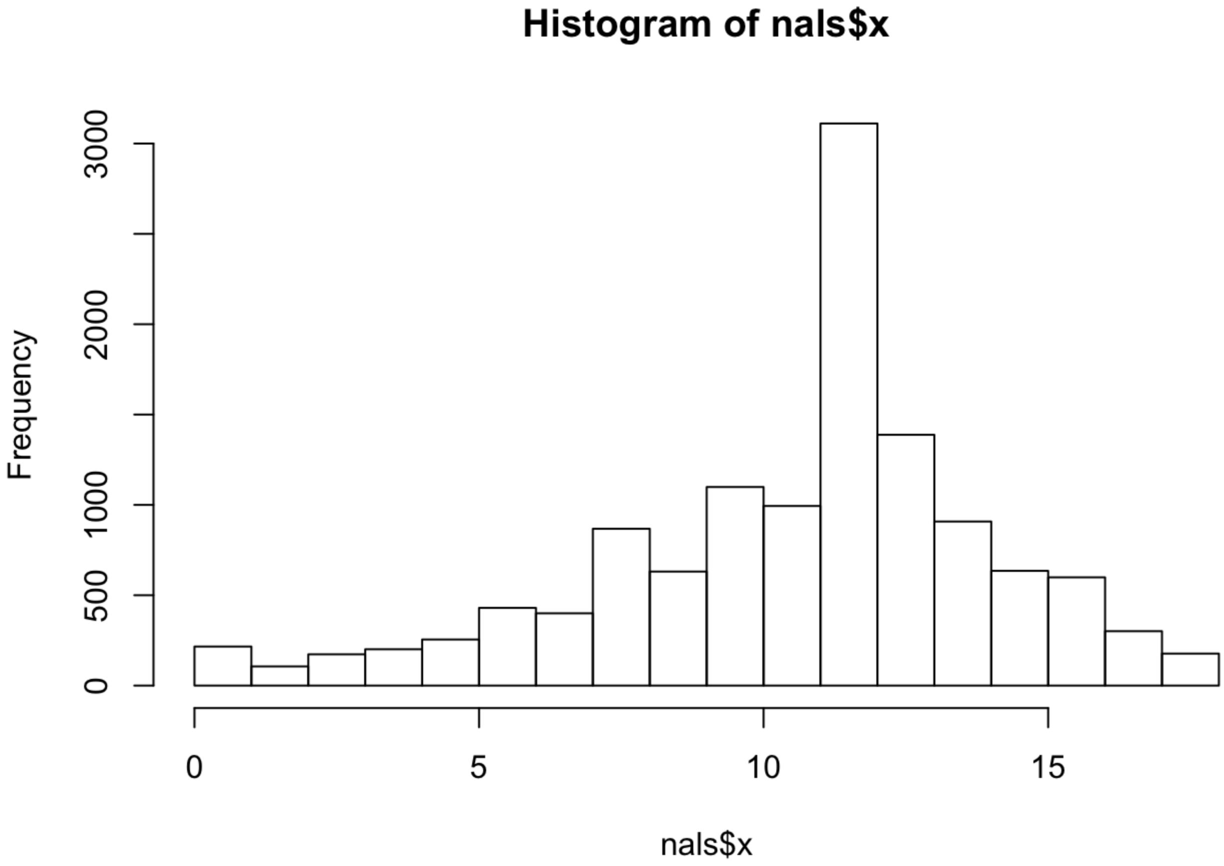
2. **Box plot**: To spot any outlier observations in the variable. Having outliers in your predictor can drastically affect the predictions as they can easily affect the direction/slope of the line of best fit.

Scatter Plot

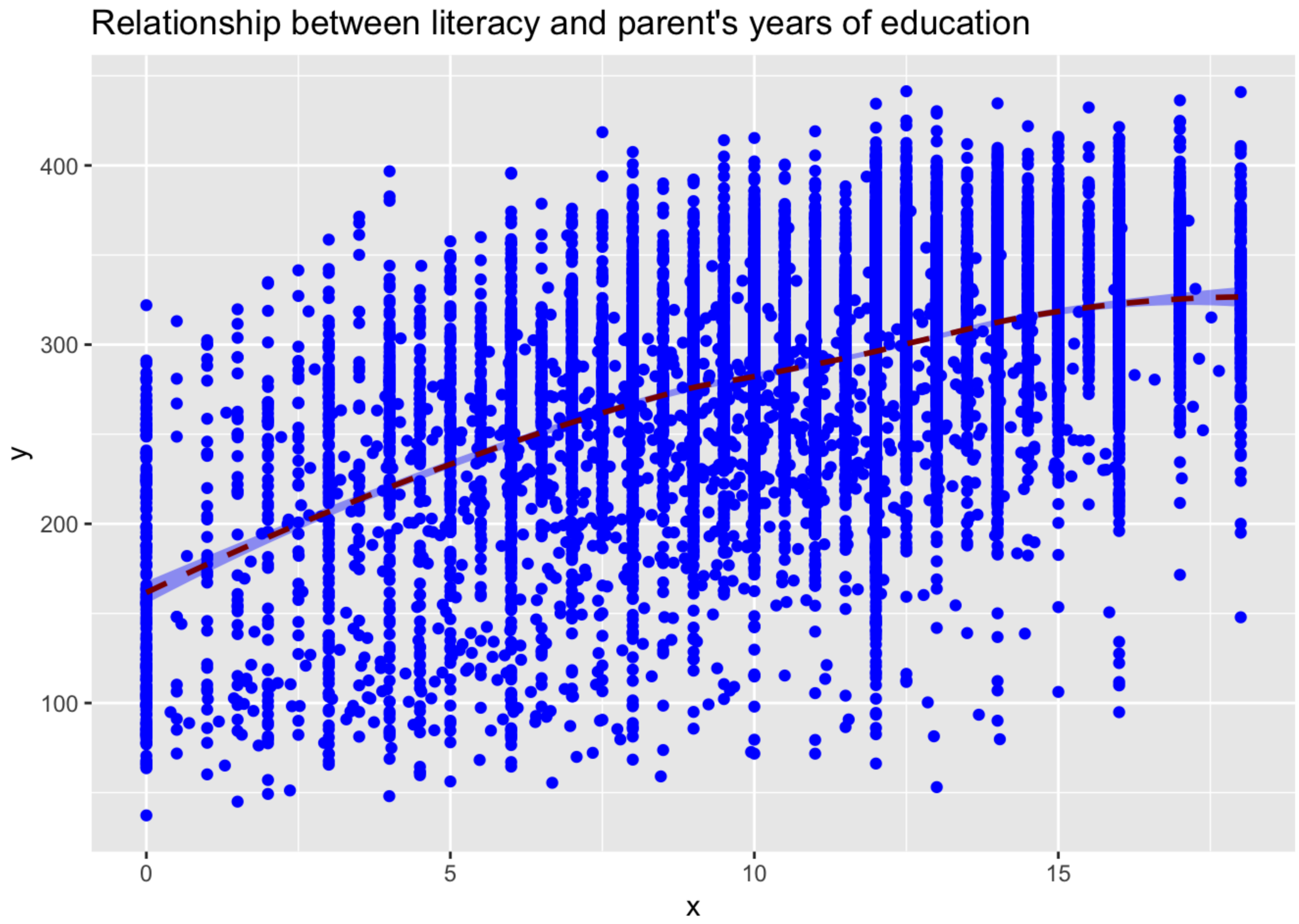
Let us look at scatter plots for each of the variables in our data

Relationship between literacy and parent’s years of education

```
hist(nals$x)
```

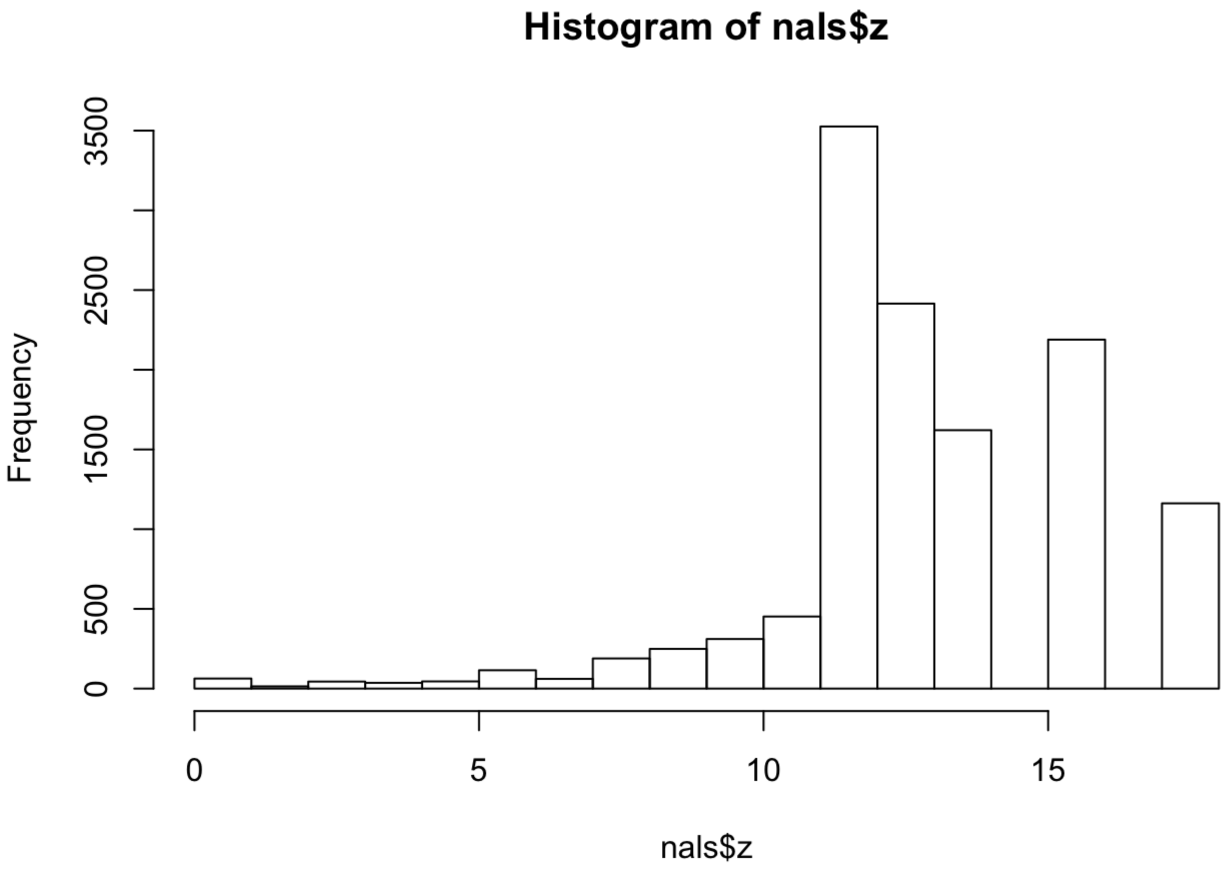


```
ggplot(nals, aes(x,y)) + geom_point(color="blue") +
  labs(title="Relationship between literacy and parent's years of education") +
  geom_smooth(method="loess", linetype="dashed",color="darkred", fill="blue")
```



Relationship between literacy and years of education

```
hist(nals$z)
```



```
nals.cutoff.z <- nals %>% filter(z>=10)
ggplot(nals.cutoff.z, aes(z,y)) + geom_point(color="blue") +
  labs(title="Relationship between literacy and parent's years of education") +
  geom_smooth(method="loess", linetype="dashed",color="darkred", fill="blue")
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : pseudoinverse used at 14
```

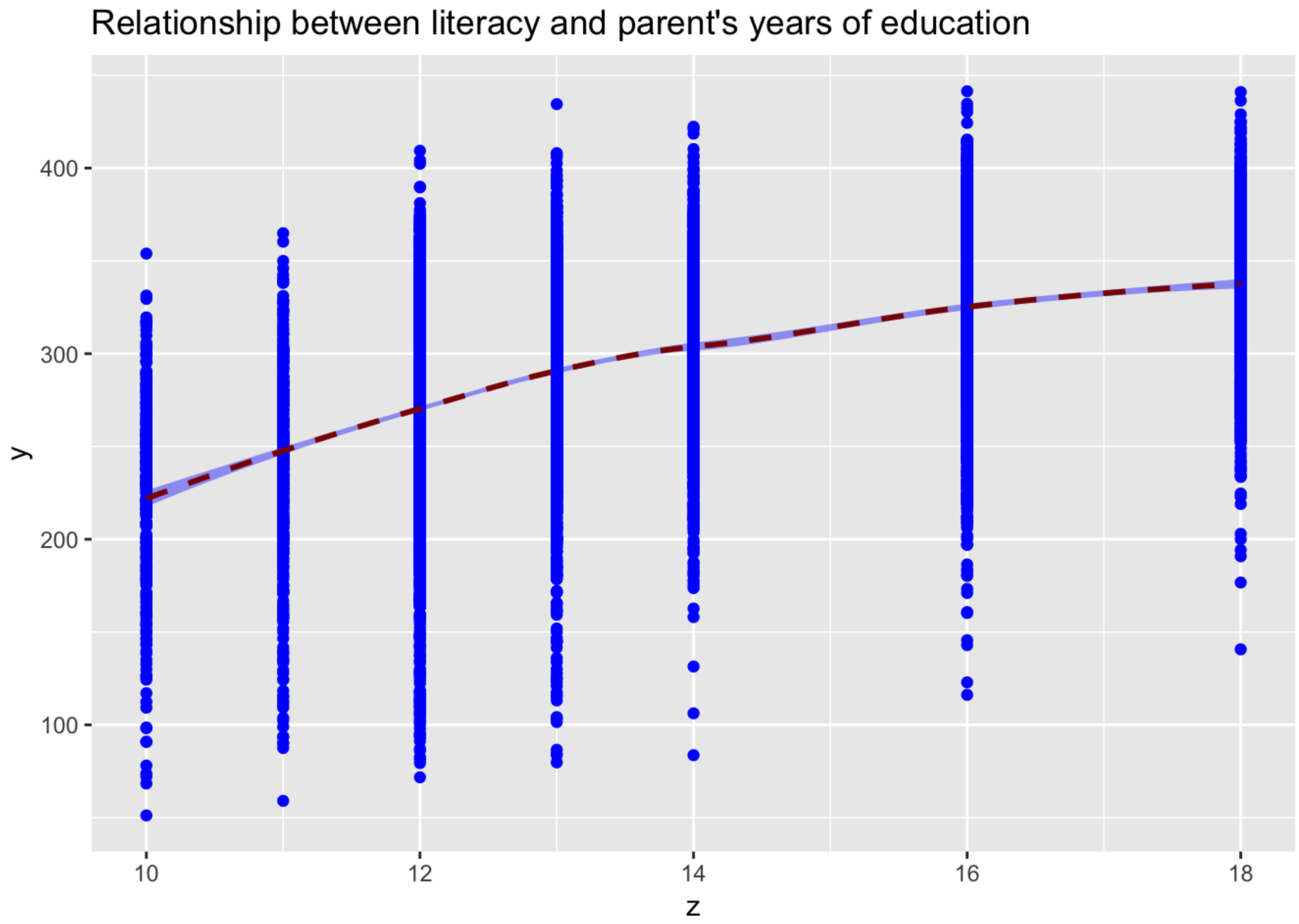
```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : neighborhood radius 2
```

```
## Warning in simpleLoess(y, x, w, span, degree = degree, parametric =
## parametric, : reciprocal condition number 1.74e-14
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : pseudoinverse used
## at 14
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : neighborhood radius
## 2
```

```
## Warning in predLoess(object$y, object$x, newx = if
## (is.null(newdata)) object$x else if (is.data.frame(newdata))
## as.matrix(model.frame(delete.response(terms(object)), : reciprocal
## condition number 1.74e-14
```



The scatter plot along with the smoothing line above suggests a linearly increasing relationship between the `y - x` and `y - z`. This is a good thing, because, one of the underlying assumptions in linear regression is that the relationship between the response and predictor variables is linear and additive.

Correlation

Correlation is a statistical measure that suggests the level of linear dependence between two variables, that occur in pair – just like what we have here in speed and dist. Correlation can take values between -1 to +1. If we observe for every instance where **years of parental education** increases increases, the **adult literacy** also increases along with it, then there is a high positive correlation between them and therefore the correlation between them will be closer to 1. The opposite is true for an inverse relationship, in which case, the correlation between the variables will be close to -1.

A value closer to 0 suggests a weak relationship between the variables. A low correlation (-0.2 < x < 0.2) probably suggests that much of variation of the response variable (Y) is unexplained by the predictor (X), in which case, we should probably look for better explanatory variables.

Correlation between adult literacy and parent’s years of education*

```
cor(nals$y, nals$x)
```

```
## [1] 0.5063804
```

Correlation between adult literacy and years of education*

```
cor(nals$y, nals$z)
```

```
## [1] 0.6570505
```

Linear Modeling Theoretical Background

Now that we have looked at the graphical plots, let us look how to model this linear relationship and what assumptions need to be satisfied for the same.

We will begin with the simple univariate predictor scenario where the response variable is `adult literacy (y)` and the predictor is `parental education (x)`. The model would look like

$$adult\_literacy_i = \beta_0 + \beta_1 parental\_education_i + \epsilon$$

For the rest of the document we will refer to the above equation in its alternate form

$$y_i = \beta_0 + \beta_1 x_i + \epsilon$$

Given that,

$$y = \beta_0 + \beta_1 x + \epsilon$$

On taking expectation on both sides, we get

$$\mathbb{E}\,Y = \mu(x)$$

Note that we are talking about $\mu(x)$ (expected value of `Y` is a function of `x`) and not μ_x (expectation of `x`). For instance,

$$\mu(x) = \beta_0 + \beta_1$$

Now we try to estimate β_0 and β_1. Now let us look at the assumptions required to consistently estimate the β coefficients.

Now that we have seen the linear relationship pictorially in the scatter plot and by computing the correlation, lets see the syntax for building the linear model. The function used for building linear models is lm(). The lm() function takes in two main arguments, namely:

1. Formula
2. Data

The data is typically a data.frame and the formula is a object of class formula. But the most common convention is to write out the formula directly in place of the argument as written below.

```
linearMod <- lm(y ~ x, data=nals)  # build linear regression model on full data
print(linearMod)
```

```
##
## Call:
## lm(formula = y ~ x, data = nals)
##
## Coefficients:
## (Intercept)            x
##      190.57         8.74
```

Now that we have built the linear model, we also have established the relationship between the predictor and response in the form of a mathematical formula for `Adult literacy(y)` as a function for `parental education(x)`.

For the above output, you can notice the 'Coefficients' part having two components:

```
Intercept: 190.57
x: 8.74
```

These are also called the *beta* coefficients. In other words,

$$y = \beta_0 + \beta_1 x + \epsilon$$

is the same as

$$y = 190.57 + 8.74x + \epsilon$$

Let us construct the linear model and then we will see if the model is a good choice by checking if it satisfies the assumptions we need to be satisfied to get an unbiased estimate.

In the univariate scenario we know that

$$\beta = \frac{cov(x, y)}{var(x)}$$

But when we have multivariate data we use the $\mathbf{X}$ to represent the matrix of data, including a column of `1` s (for intercept term) and the same equation in the matrix form becomes

$$beta = \frac{\mathbf{X^T Y}}{\mathbf{X^T X}}$$

We will be using this in the code section below to show that the results you get from manual calculations are still the same as the ones you get from the regression table.

```
# beta calulation when you have more than 1 covariate done by hand
#make a new column called intercept and assign value of to all of the rows
nals$intercept <- 1

#create a new data frame of covariates (intercept and x)
X <- cbind(nals$intercept, nals$x)
# convert it into a matrix
X <- as.matrix(X)
colnames(X) <- c("intercept","x")

# convert Y into matrix
Y <- as.matrix(nals$y)

# showing that beta = cov(X,Y)/Var(X) or as cross products in vector form
solve(crossprod(X), crossprod(X,Y))
```

```
##                 [,1]
## intercept 190.568310
## x           8.739734
```

```
summ(linearMod)
```

Observations	12492
Dependent variable	y
Type	OLS linear regression

F(1,12490)	4307.14
R²	0.26
Adj. R²	0.26

	Est.	S.E.	t val.	p
(Intercept)	190.57	1.53	124.57	0.00
x	8.74	0.13	65.63	0.00

Standard errors: OLS

We see that the estimates from manual calculation and that from regression are the same.

```
**What assumptions do we need to satisfy to be able to report these results?**
```

Assumptions

1. **Linearity of data** - The relationship between the predictor (x) and the outcome (y) is assumed to be linear.

2. **Normality of residuals** - The residual errors are assumed to be normally distributed.

3. **Homogeneity of residuals variance** - The residuals are assumed to have a constant variance (homoscedasticity).

4. Independence of residuals error terms.

You should check whether or not these assumptions hold true. Potential problems include:

1. **Non-linearity of the outcome** - predictor relationships
2. **Heteroscedasticity**: Non-constant variance of error terms.
3. Presence of influential values in the data that can be:
   3.1. **Outliers**: extreme values in the outcome (y) variable
   3.2. **High-leverage points**: extreme values in the predictors (x) variable

All these assumptions and potential problems can be checked by producing some diagnostic plots visualizing the residual errors.

Graphical verification of the assumptions

Residual Plots

to check the linearity assumptions, we are going to plot residuals vs. fitted values. if model is linear then the residual plot should show no fitted pattern (since model explains the structural variance)

getting residuals from the model

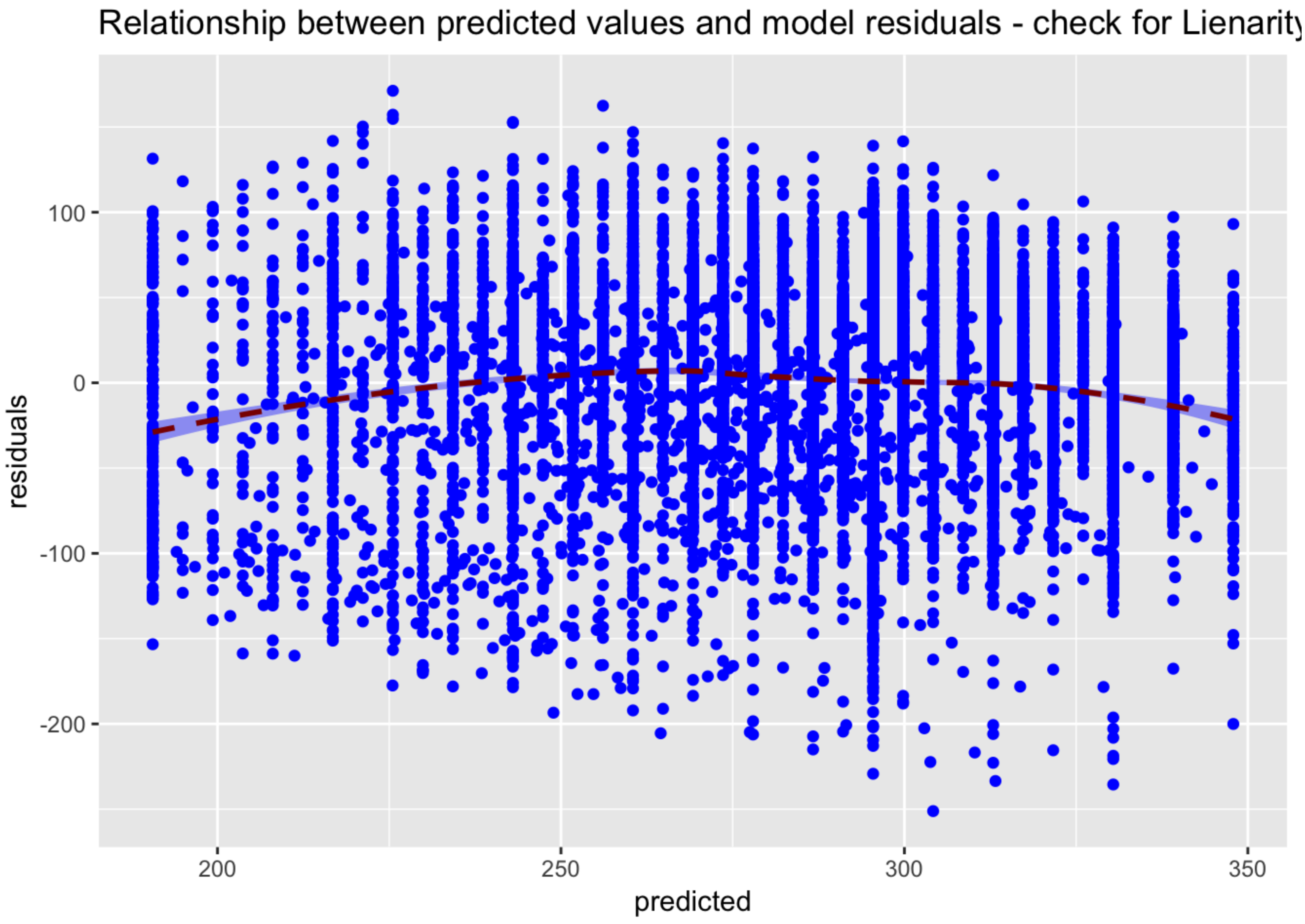
```
nals$residuals <- residuals(linearMod)
```

getting predicted values from the model

```
nals$predicted <- predict(linearMod)
```

Let us check the plot now

```
ggplot(nals, aes(predicted,residuals)) + geom_point(color="blue") +
  labs(title="Relationship between predicted values and model residuals - check for Lienarity") +
  geom_smooth(method="loess", linetype="dashed",color="darkred", fill="blue")
```



linearity assumption not really satisfied

to check for heteroskedasticity we can use Brusch - Pagan test

```
lmtest::bptest(linearMod)
```

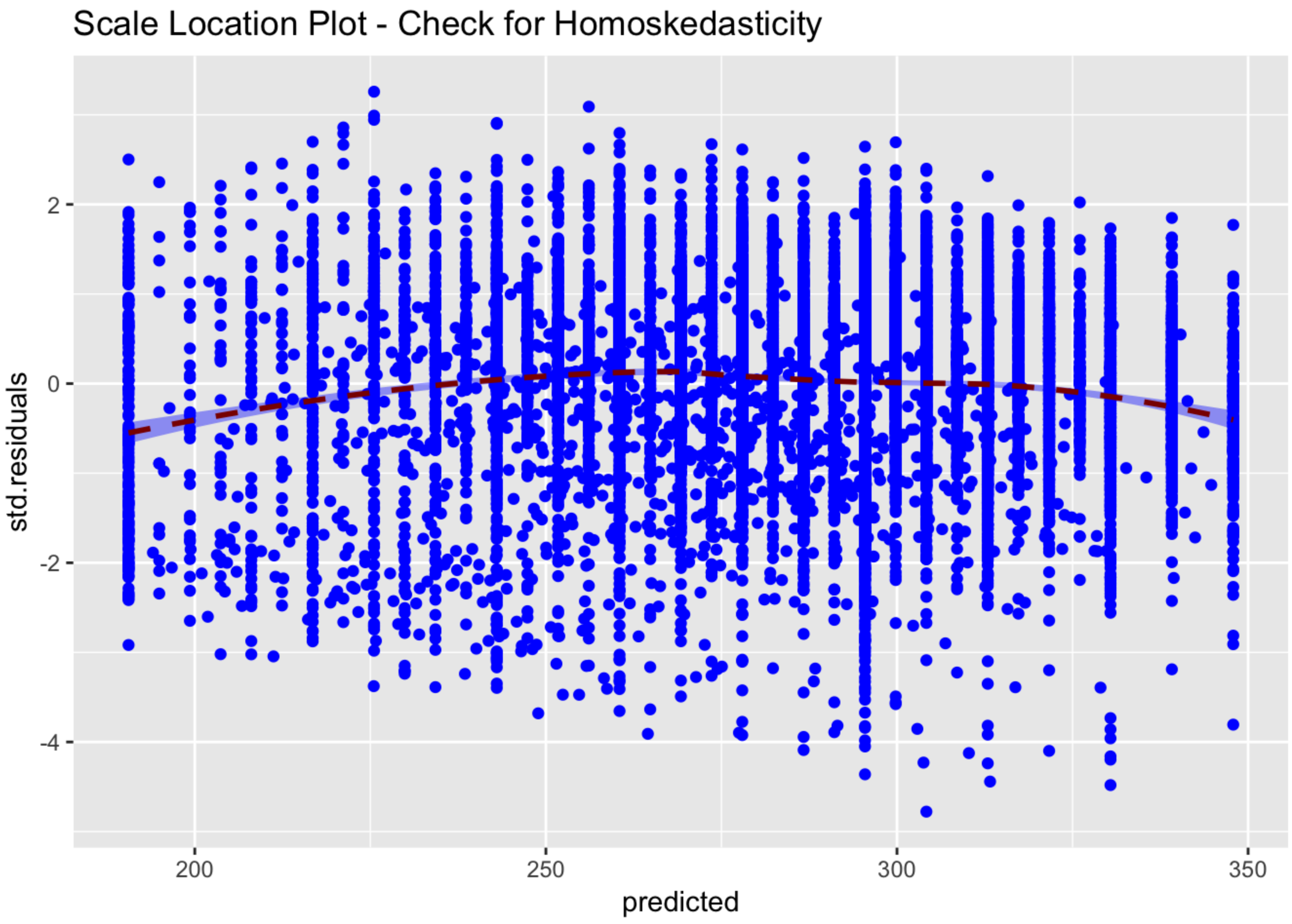
```
##
##  studentized Breusch-Pagan test
##
## data:  linearMod
## BP = 375.01, df = 1, p-value < 2.2e-16
```

Unable to reject the null hypothesis that variance is constant. Let us see grahically in the Scale-location plot

```
nals$std.residuals<- rstandard(linearMod)
```

Let us check the plot now

```
ggplot(nals, aes(predicted,std.residuals)) + geom_point(color="blue") +
  labs(title="Scale Location Plot - Check for Homoskedasticity") +
  geom_smooth(method="loess", linetype="dashed",color="darkred", fill="blue")
```



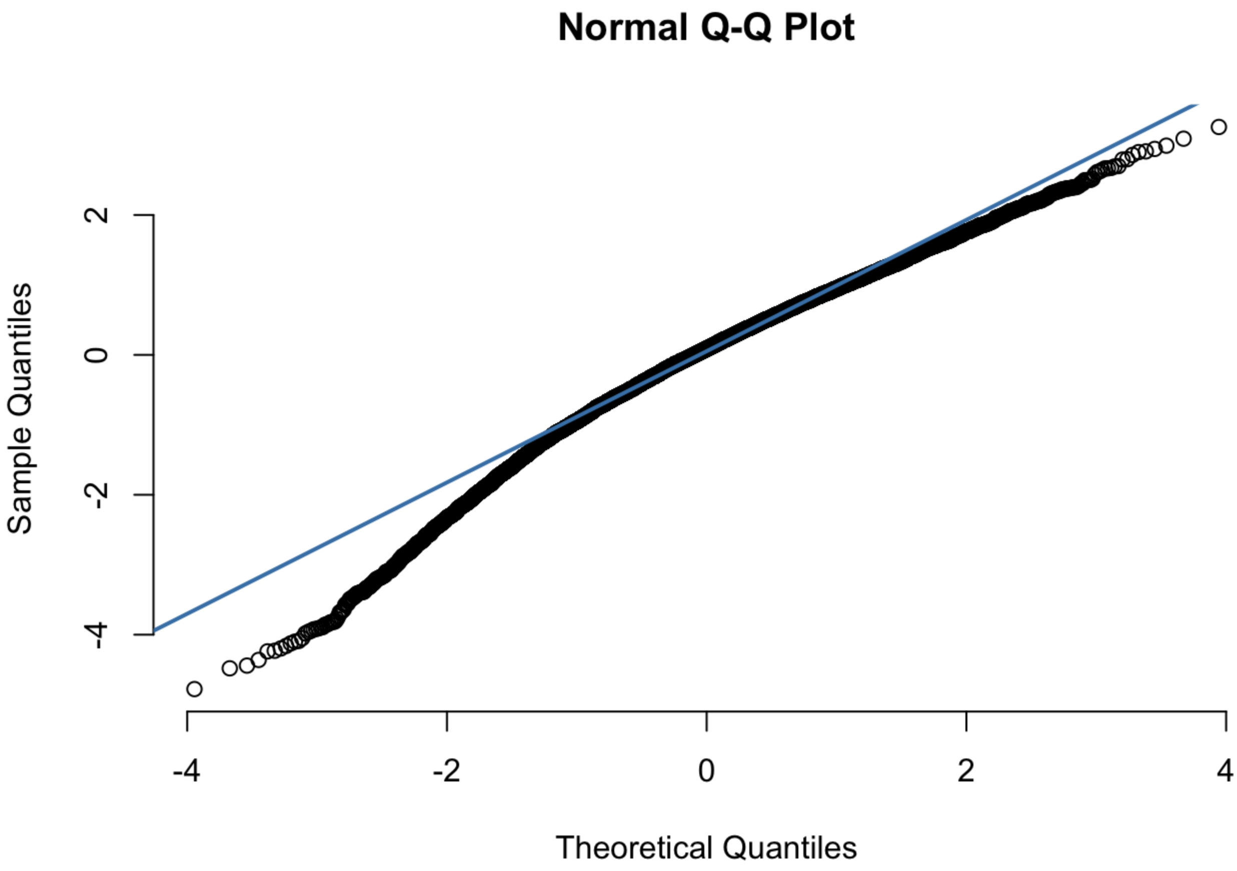
homoskedasticity is also not satisfied

let us check for the normality assumption for residuals now

The QQ plot of residuals can be used to visually check the normality assumption.

The normal probability plot of residuals should approximately follow a straight line.

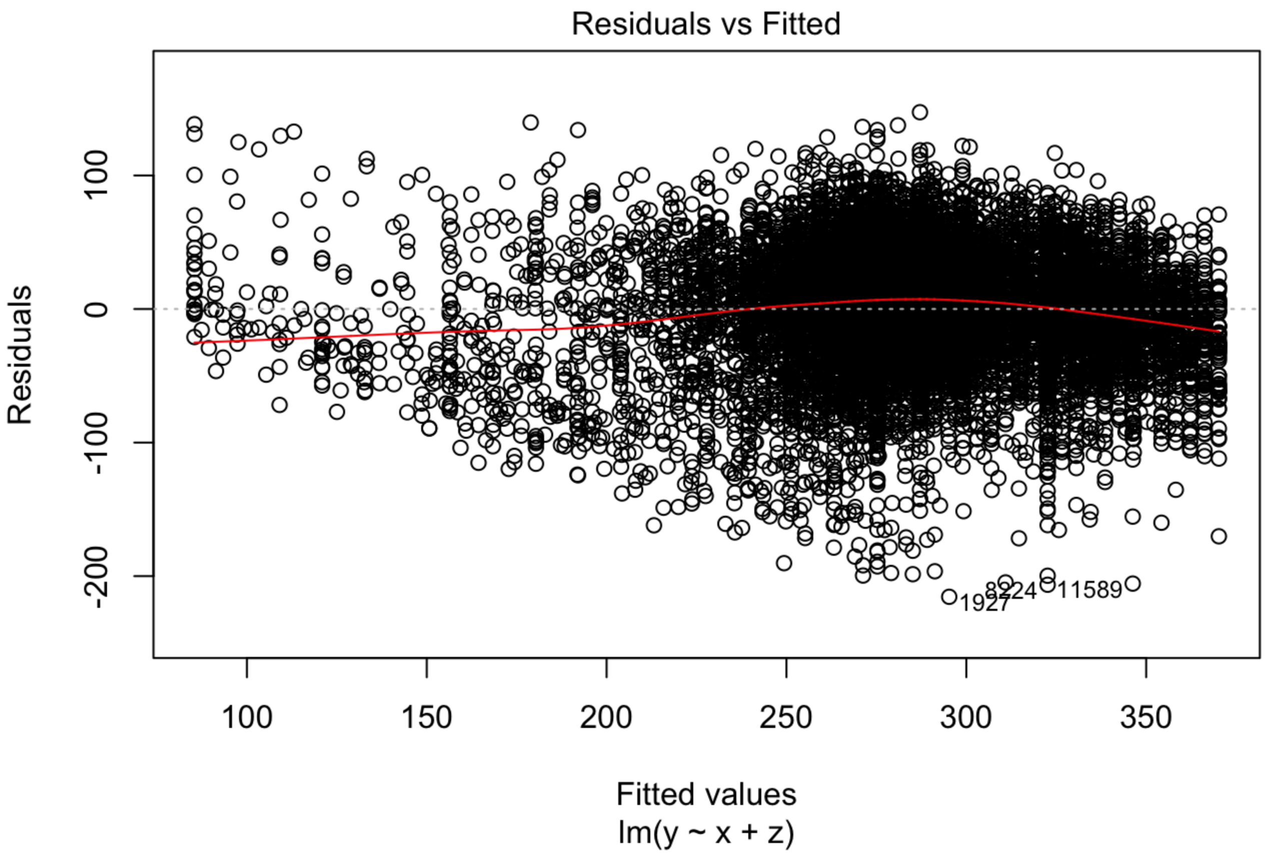
```
qqnorm(nals$std.residuals, pch = 1, frame = FALSE)
qqline(nals$std.residuals, col = "steelblue", lwd = 2)
```

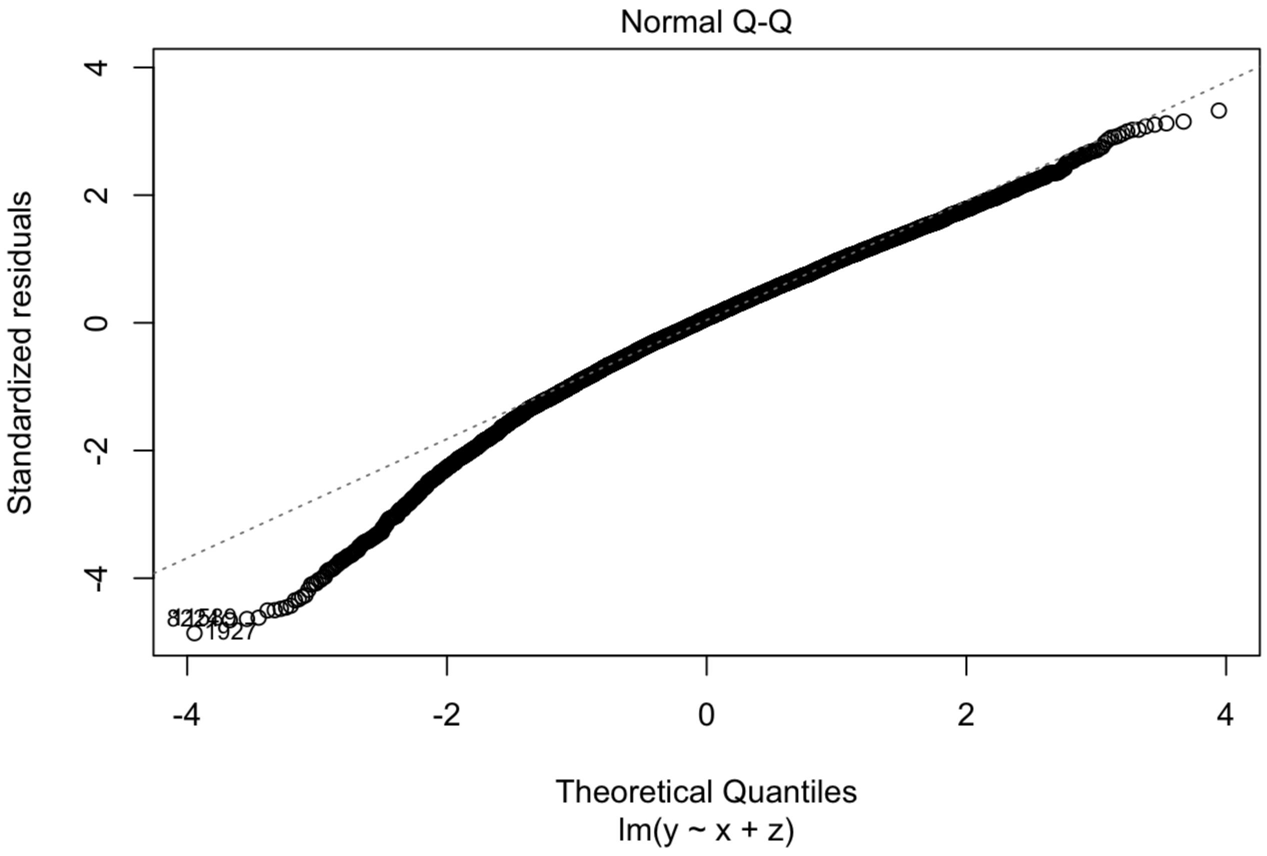


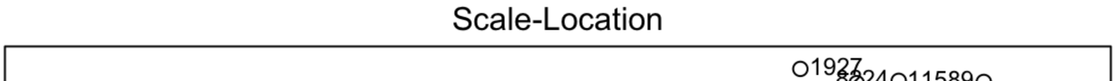
```
linearmodel <- lm(y~x+z, data = nals)
linearmodel
```

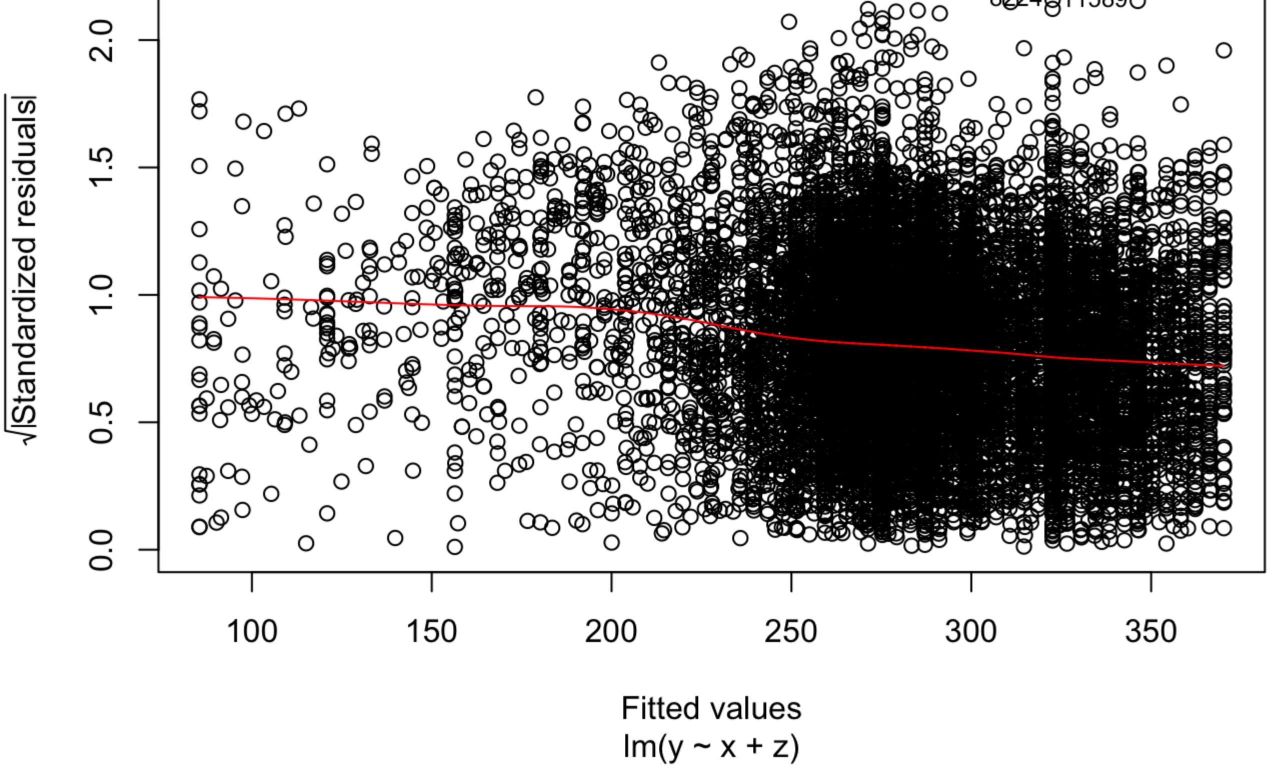
```
##
## Call:
## lm(formula = y ~ x + z, data = nals)
##
## Coefficients:
## (Intercept)            x            z
##       85.42         3.99        11.83
```

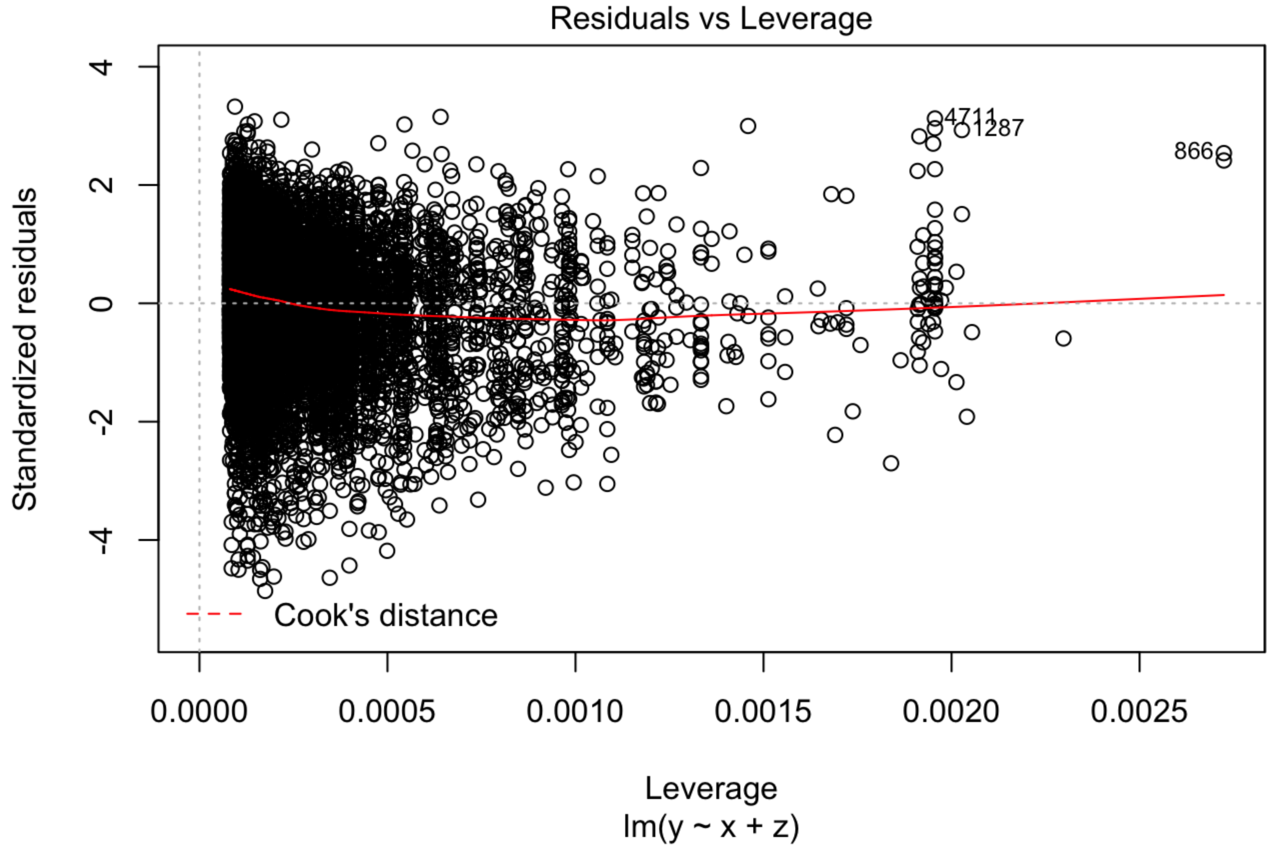
```
## plot (linearmodel) tells you about the residuals and assumptions
plot(linearmodel)
```


Residuals vs Fitted
Residuals
100
0
-100
-200
100
150
200
250
300
350
Fitted values
lm(y ~ x + z)
1927
8224
11589


Normal Q-Q
Standardized residuals
4
2
0
-2
-4
-4
-2
0
2
4
Theoretical Quantiles
lm(y ~ x + z)
8224
11589
1927


Scale-Location
1927
8224
11589





Explanation

Residual review

The residual plots can be used in four different ways

1. **Residuals vs. Fitted**. Used to check the linear relationship assumptions. A horizontal line, without distinct patterns is an indication for a linear relationship, what is good.

2. **Normal Q-Q**. Used to examine whether the residuals are normally distributed. It’s good if residuals points follow the straight dashed line.

3. **Scale-Location (or Spread-Location)**. Used to check the homogeneity of variance of the residuals (homoscedasticity). Horizontal line with equally spread points is a good indication of homoscedasticity. This is not the case in our example, where we have a heteroscedasticity problem.

4. **Residuals vs Leverage**. Used to identify influential cases, that is extreme values that might influence the regression results when included or excluded from the analysis. This plot will be described further in the next sections.

Linear Regression Diagonistics

Now the linear model is built and we have a formula that we can use to predict the `adult literacy` if the corresponding `parental years of education` is known. Is this enough to actually use this model? NO! Before using a regression model, you have to ensure that it is statistically significant. How do you ensure this? Lets begin by printing the summary statistics for linearMod.

```
summary(linearMod)  # model summary
```

```
##
## Call:
## lm(formula = y ~ x, data = nals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -251.130  -30.467    4.758   36.072  171.227
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 190.5683     1.5299  124.57   <2e-16 ***
## x             8.7397     0.1332   65.63   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 52.57 on 12490 degrees of freedom
## Multiple R-squared:  0.2564, Adjusted R-squared:  0.2564
## F-statistic:  4307 on 1 and 12490 DF,  p-value: < 2.2e-16
```

p-value : Checking for statistical significance

The summary statistics above tells us a number of things. One of them is the model p-Value (bottom last line) and the p-Value of individual predictor variables (extreme right column under ‘Coefficients’). The p-Values are very important because, We can consider a linear model to be statistically significant only when both these p-Values are less that the pre-determined statistical significance level, which is ideally 0.05. This is visually interpreted by the significance stars at the end of the row. The more the stars beside the variable’s p-Value, the more significant the variable.

Now lets add an interaction to the model because we see that the model assumptions are not satisfied and it is not linear

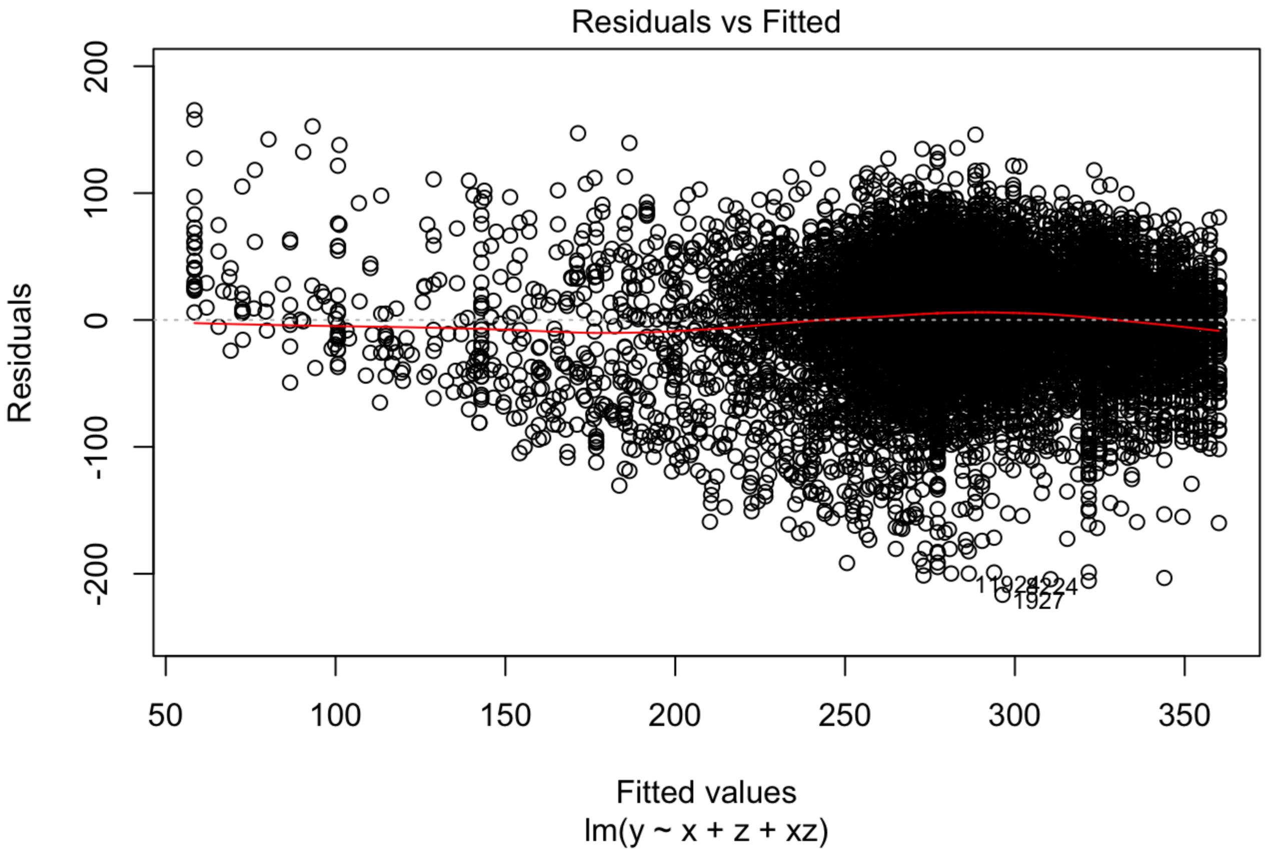
Interaction between x and z

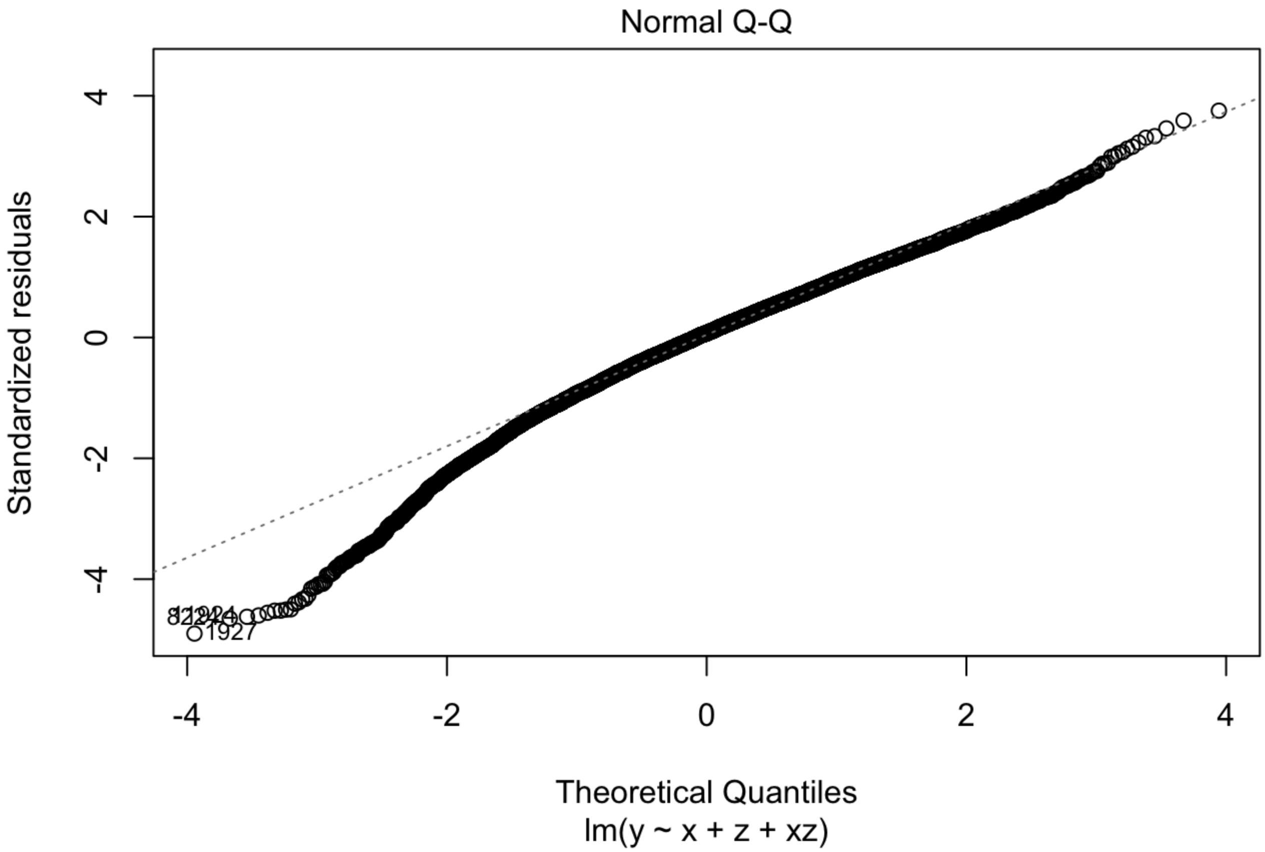
```
nals$xz <- (nals$x)*(nals$z)
model.interaction <- lm (y ~ x+z+xz, data=nals)
summary(model.interaction)
```

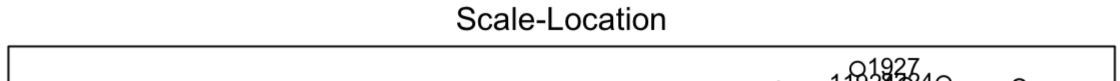
```
##
## Call:
## lm(formula = y ~ x + z + xz, data = nals)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -216.572  -25.529    3.241   29.449  165.335
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 58.45371    3.68257  15.873   <2e-16 ***
## x            7.11243    0.38406  18.519   <2e-16 ***
## z           14.07891    0.30833  45.662   <2e-16 ***
## xz          -0.24644    0.02852  -8.642   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 44.2 on 12488 degrees of freedom
## Multiple R-squared:  0.4744, Adjusted R-squared:  0.4742
## F-statistic:  3757 on 3 and 12488 DF,  p-value: < 2.2e-16
```

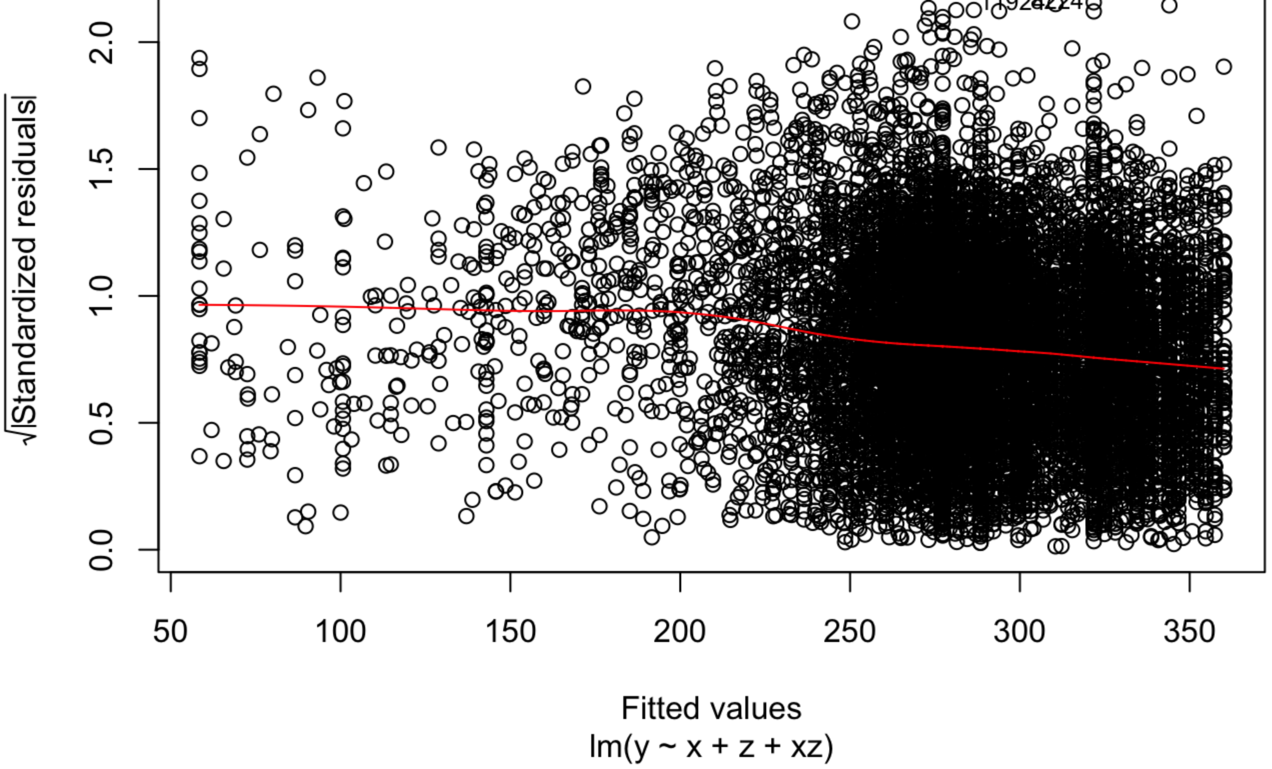
now we have a new model, lets check assumptions again

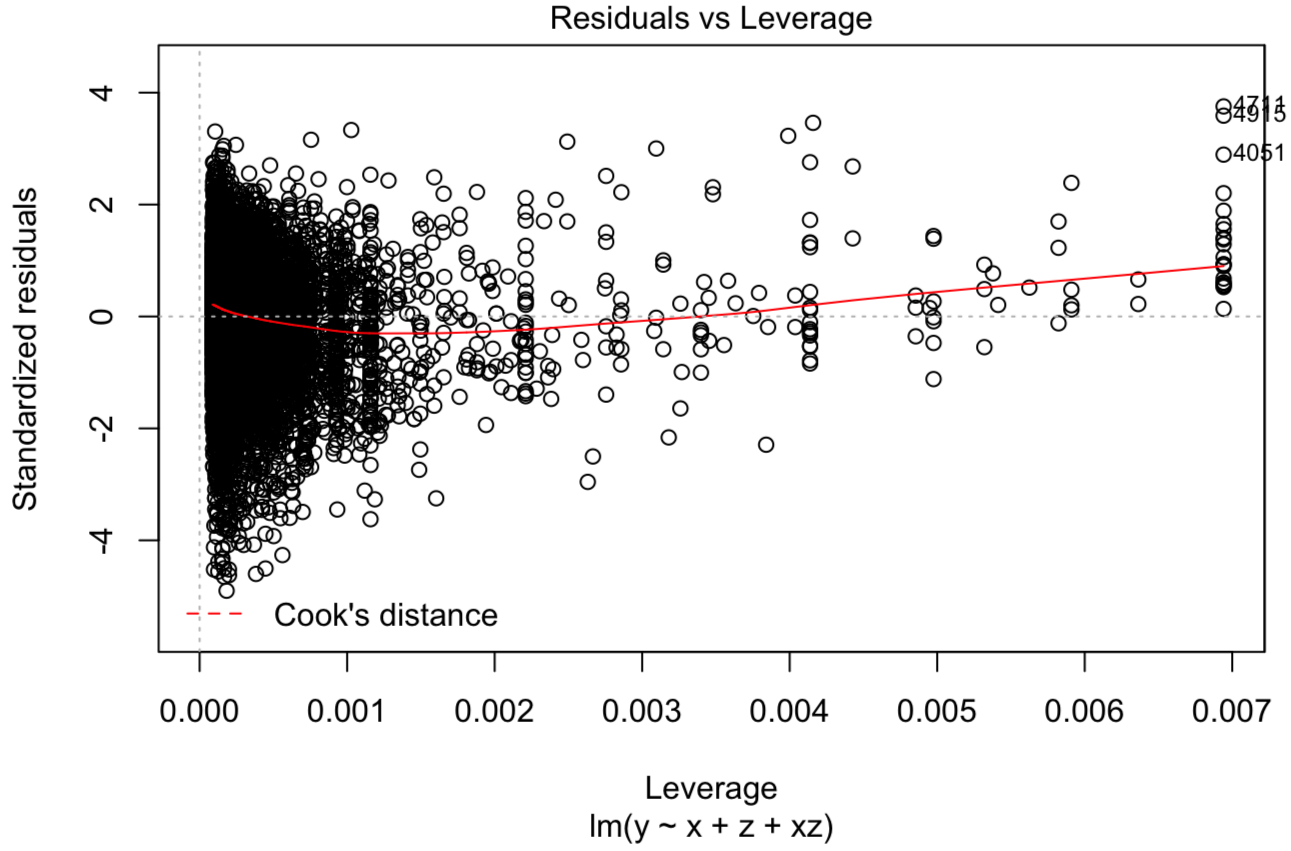
```
plot(model.interaction)
```


Residuals vs Fitted
Residuals
200
100
0
-100
-200
1192
4224
1927
50
100
150
200
250
300
350
Fitted values
lm(y ~ x + z + xz)


Normal Q-Q
Standardized residuals
4
2
0
-2
-4
1192
4224
1927
-4
-2
0
2
4
Theoretical Quantiles
lm(y ~ x + z + xz)


Scale-Location
1927
1192
4224





getting residuals from the model - math equation? extracting (y - predicted y)

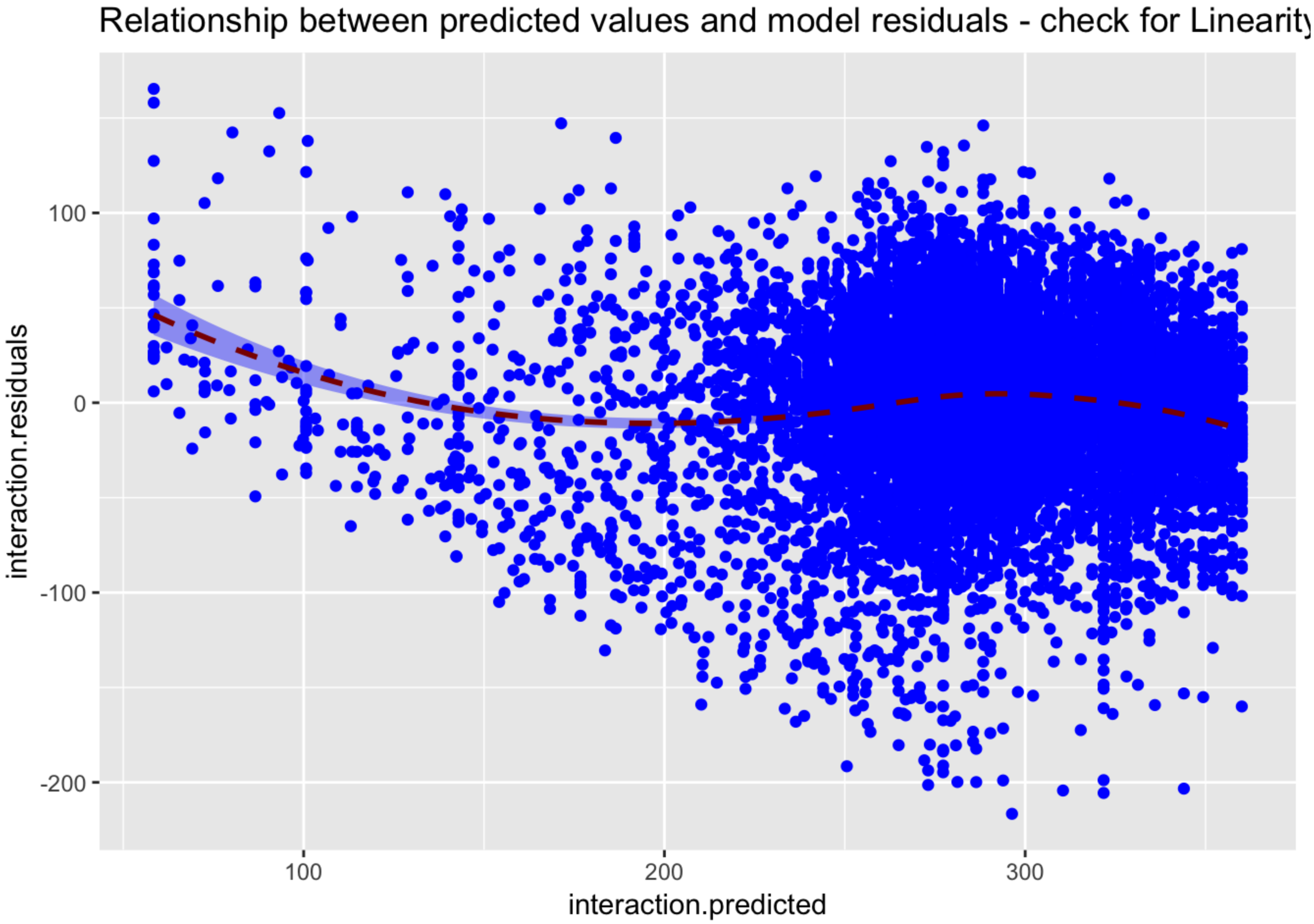
```
nals$interaction.residuals <- residuals(model.interaction)
```

getting predicted values from the model

```
nals$interaction.predicted <- predict(model.interaction)
```

Let us check the plot now

```
ggplot(nals, aes(interaction.predicted,interaction.residuals)) + geom_point(color="blue") +
  labs(title="Relationship between predicted values and model residuals - check for Linearity") +
  geom_smooth(method="loess", linetype="dashed",color="darkred", fill="blue")
```



Linearity assumption not really satisfied

to check for heteroskedasticity we can use Brusch - Pagan test

```
lmtest::bptest(model.interaction)
```

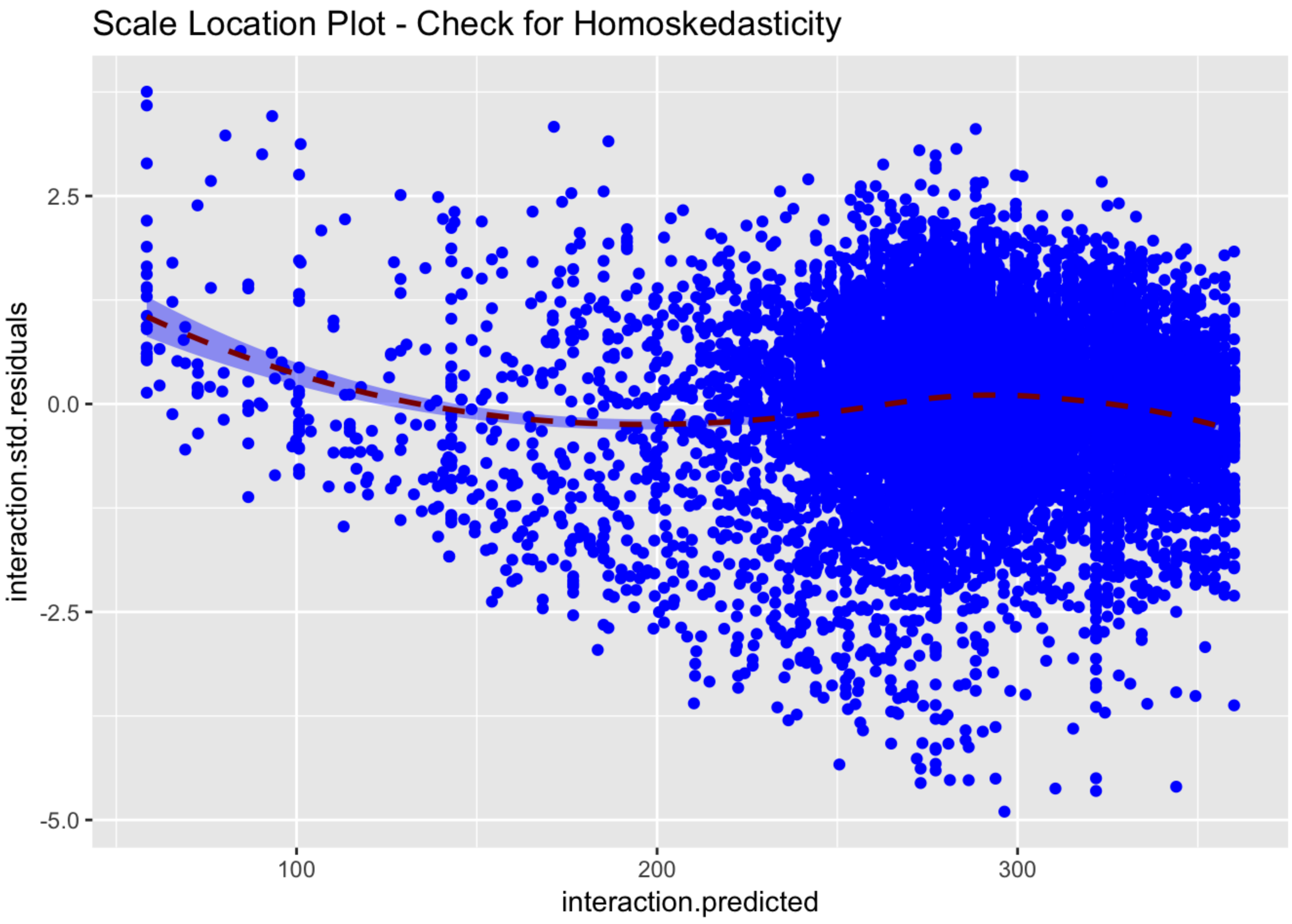
```
##
##  studentized Breusch-Pagan test
##
## data:  model.interaction
## BP = 207.23, df = 3, p-value < 2.2e-16
```

Unable to reject the null hypothesis that variance is constant. Let us see grahically in the Scale-location plot

```
nals$interaction.std.residuals<- rstandard(model.interaction)
```

Let us check the plot now

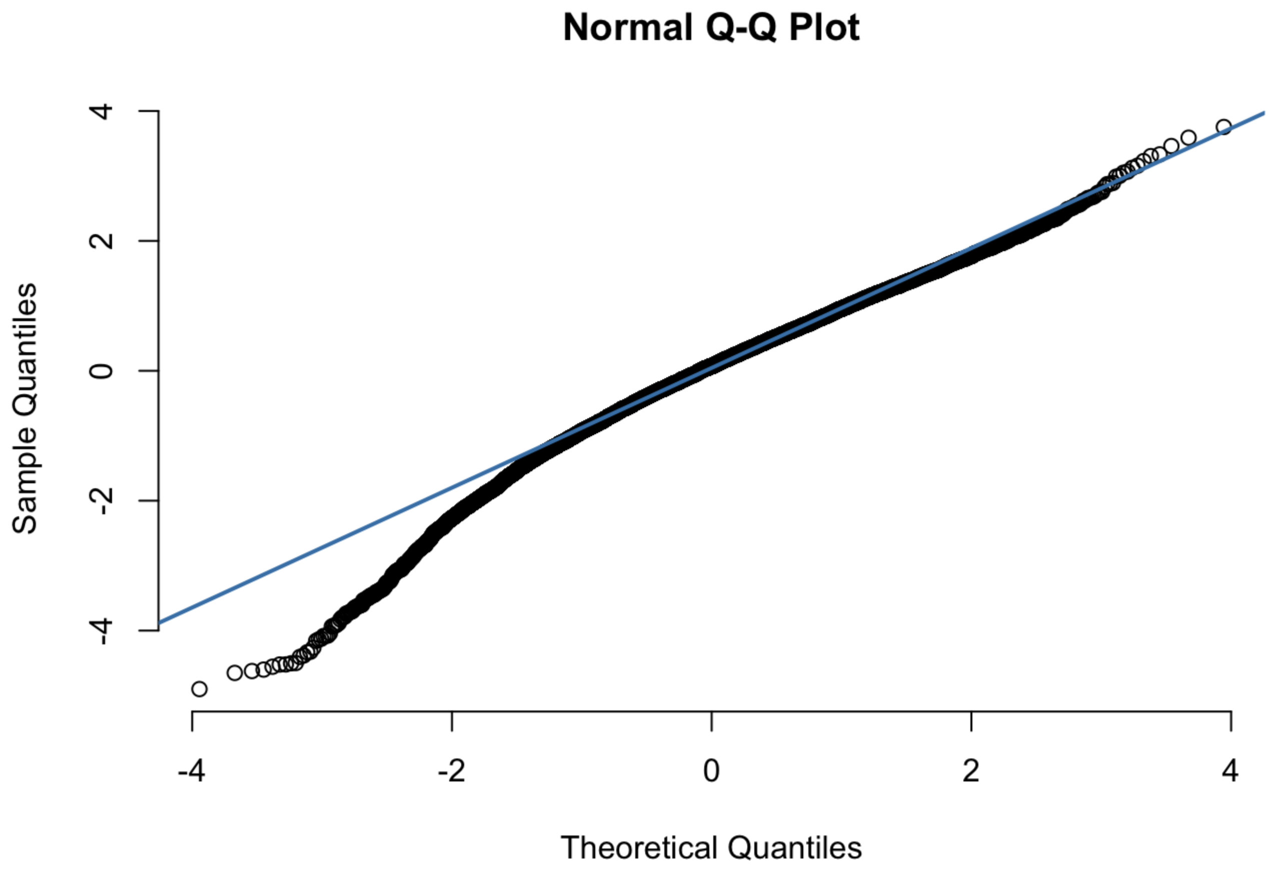
```
ggplot(nals, aes(interaction.predicted,interaction.std.residuals)) + geom_point(color="blue") +
  labs(title="Scale Location Plot - Check for Homoskedasticity") +
  geom_smooth(method="loess", linetype="dashed",color="darkred", fill="blue")
```



Homoskedasticity is also not satisfied.

let us check for the normality assumption for residuals now The QQ plot of residuals can be used to visually check the normality assumption. The normal probability plot of residuals should approximately follow a straight line.

```
qqnorm(nals$interaction.std.residuals, pch = 1, frame = FALSE)
qqline(nals$interaction.std.residuals, col = "steelblue", lwd = 2)
```



Centering Variables

You can redo all the analysis done earlier with mean centered variables so that we can get a meaningful interpretation for the `intercept` term. Centering also helps to reduce the correlation between variables and their corresponding **interaction terms** in the model.

```
nals <- nals %>%  mutate(x.center=x-mean(x))
```