Class 3 - Linear Regression (Supply and Demand)

Load Libraries

```
knitr::opts_chunk$set(echo = TRUE)
library(dplyr)
```

```
##
## Attaching package: ’dplyr’

## The following objects are masked from ’package:stats’:
##
##     filter, lag

## The following objects are masked from ’package:base’:
##
##     intersect, setdiff, setequal, union
```

```
library(MASS)
```

```
##
## Attaching package: ’MASS’

## The following object is masked from ’package:dplyr’:
##
##     select
```

Load Dataset

```
adverts = read.csv("E:/MSc DSc/Sem 1/Business Analytics/Ch4_marketing.csv")
head(adverts, 10)
```

```
##    google_adwords facebook twitter marketing_total revenues
## 1           65.66    47.86   52.46          165.98    39.26
## 2           39.10    55.20   77.40          171.70    38.90
## 3          174.81    52.01   68.01          294.83    49.51
## 4           34.36    61.96   86.86          183.18    40.56
## 5           78.21    40.91   30.41          149.53    40.21
## 6           34.19    15.09   12.79           62.07    38.09
## 7          225.71    15.91   33.31          274.93    44.21
## 8           90.03    17.13   34.33          141.49    40.23
## 9          238.40    35.10   13.90          287.40    48.80
## 10          43.53    42.23   71.83          157.59    36.63
```

```
str(adverts)
```

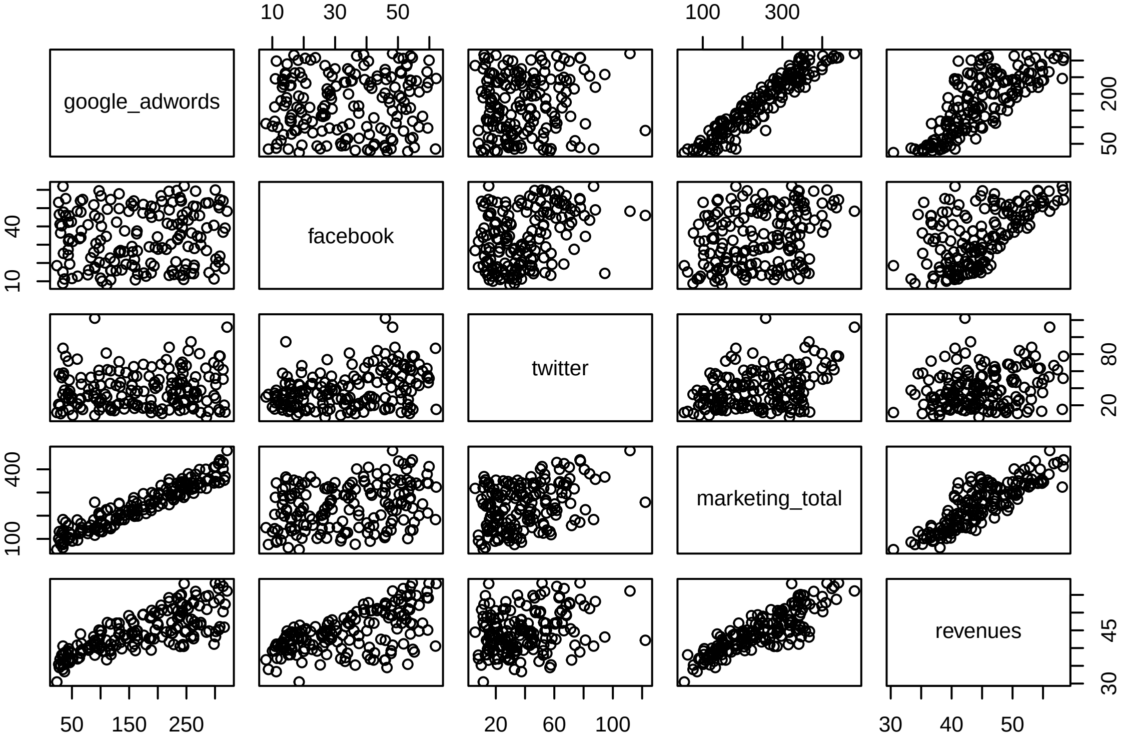
```
## 'data.frame':    172 obs. of  5 variables:
##  $ google_adwords : num  65.7 39.1 174.8 34.4 78.2 ...
##  $ facebook       : num  47.9 55.2 52 62 40.9 ...
##  $ twitter        : num  52.5 77.4 68 86.9 30.4 ...
##  $ marketing_total: num  166 172 295 183 150 ...
##  $ revenues       : num  39.3 38.9 49.5 40.6 40.2 ...
```

```
summary(adverts)
```

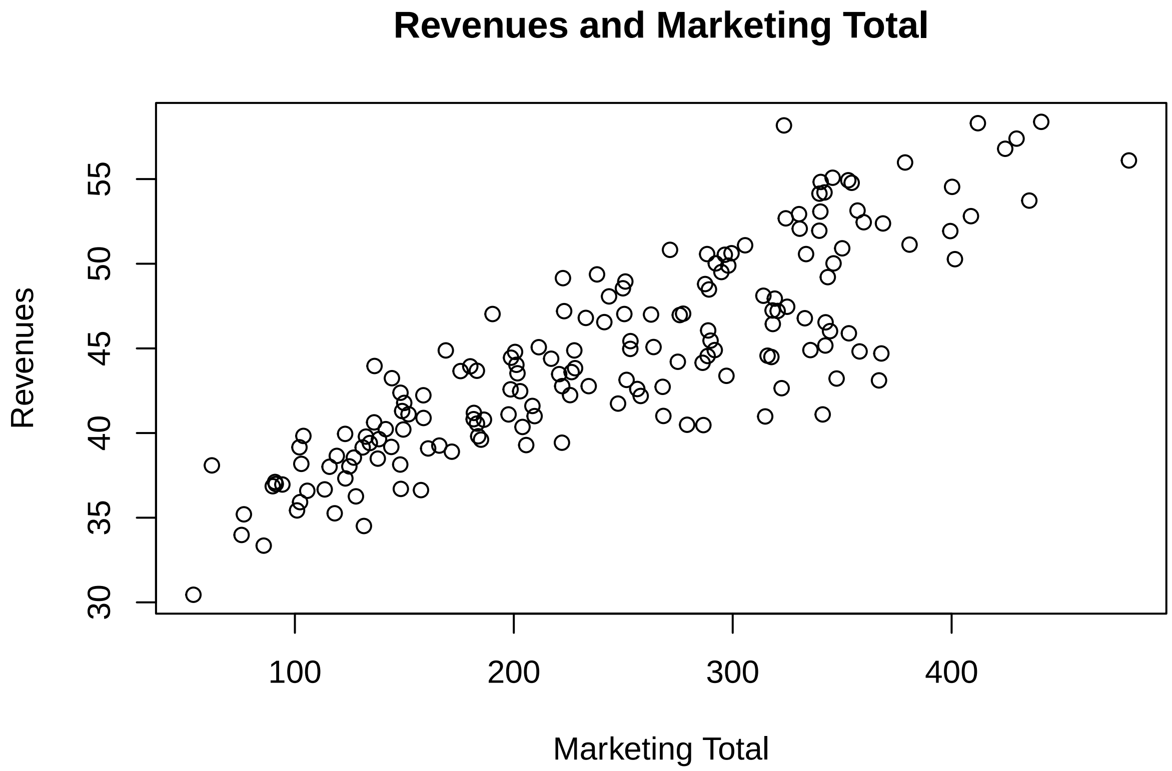
```
##  google_adwords      facebook        twitter       marketing_total
##  Min.   : 23.65   Min.   : 8.00   Min.   :  5.89   Min.   : 53.65
##  1st Qu.: 97.25   1st Qu.:19.37   1st Qu.: 20.94   1st Qu.:158.41
##  Median :169.47   Median :33.66   Median : 34.59   Median :245.56
##  Mean   :169.87   Mean   :33.87   Mean   : 38.98   Mean   :242.72
##  3rd Qu.:243.10   3rd Qu.:47.80   3rd Qu.: 52.94   3rd Qu.:322.62
##  Max.   :321.00   Max.   :62.17   Max.   :122.19   Max.   :481.00
##     revenues
##  Min.   :30.45
##  1st Qu.:40.33
##  Median :43.99
##  Mean   :44.61
##  3rd Qu.:48.61
##  Max.   :58.38
```

Scatterplot

```
pairs(adverts)
```



```
plot(adverts$marketing_total, adverts$revenues, ylab="Revenues",xlab="Marketing Total", main="Revenues
```



Fit the model

```
m1 = lm(revenues ~ marketing_total, data = adverts)
m1
```

```
##
## Call:
## lm(formula = revenues ~ marketing_total, data = adverts)
##
## Coefficients:
##     (Intercept)  marketing_total
##        32.00670          0.05193
```

Look at the structure of model

Find a way to call yhat,beta0,beta1, and e

```
yhat_model = m1$fitted.values
beta0_model = m1$coefficients[1]
beta1_model = m1$coefficients[2]
Res_model = m1$residuals
```

Compute yhat and e manually

```
yhat_manual = beta0_model + (beta1_model*adverts$marketing_total)
Res_manual = adverts$revenues - yhat_manual
```

Construct DF (yhat_model, yhat_manual, Res_model, Res_manual)

values from manual calculation and from the formula should be the same

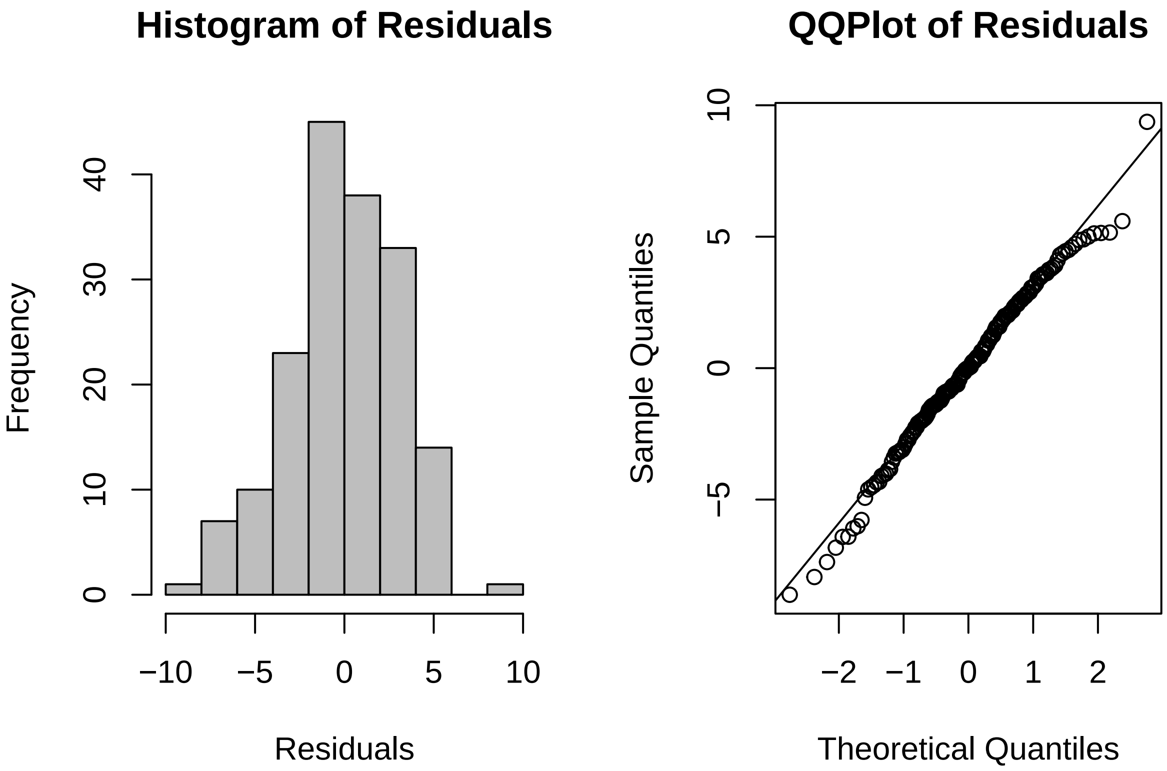
```
compiled = data.frame(
  "yhat_model" = yhat_model,
  "yhat_manual" = yhat_manual,
  "Diff_yhat" = sum(round(yhat_manual,8)) - sum(round(yhat_model,8)),
  "Res_model" = Res_model,
  "Res_manual" = Res_manual
)
head(compiled, 5)
```

```
##   yhat_model yhat_manual Diff_yhat  Res_model Res_manual
## 1   40.62587    40.62587         0 -1.3658695 -1.3658695
## 2   40.92290    40.92290         0 -2.0229033 -2.0229033
## 3   47.31692    47.31692         0  2.1930804  2.1930804
## 4   41.51905    41.51905         0 -0.9590481 -0.9590481
## 5   39.77164    39.77164         0  0.4383624  0.4383624
```

Assumptions

Normality

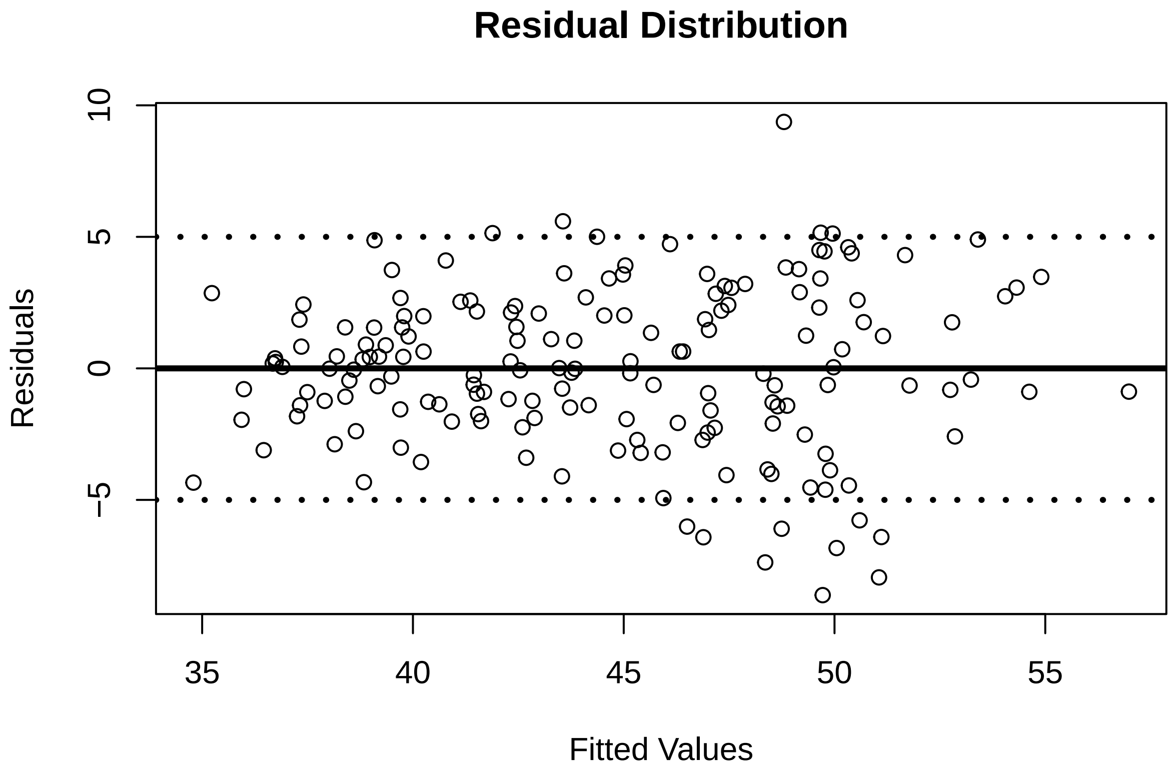
```
par(mfrow = c(1,2))
hist(m1$residuals, xlab = "Residuals", col = 'grey', main = "Histogram of Residuals")
qqnorm(m1$residuals, main = "QQPlot of Residuals")
qqline(m1$residuals)
```



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

Equal Variance

```r
plot(m1$fitted.values, m1$residuals, ylab = "Residuals", xlab = 'Fitted Values', main = "Residual Distr
abline(h = 0,lwd=3); abline(h = c(-5,5), lwd=3,lty=3)
```



```
summary(m1)
```

```
##
## Call:
## lm(formula = revenues ~ marketing_total, data = adverts)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6197 -1.8963 -0.0006  2.1705  9.3689
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)     32.006696   0.635590   50.36   <2e-16 ***
## marketing_total  0.051929   0.002437   21.31   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.054 on 170 degrees of freedom
## Multiple R-squared:  0.7277, Adjusted R-squared:  0.7261
## F-statistic: 454.2 on 1 and 170 DF,  p-value: < 2.2e-16
```

Prediction

```r
newdata = data.frame(marketing_total = 460)
predict.lm(m1, newdata, interval = 'predict')
```

```
##        fit      lwr      upr
## 1 55.89403 49.75781 62.03025
```

```r
predict.lm(m1,newdata, level = 0.99, interval = 'predict')
```

```
##        fit      lwr      upr
## 1 55.89403 47.79622 63.99184
```

```r
newdata = data.frame(marketing_total = c(450,460,470))
predict.lm(m1, newdata, interval = 'predict')
```

```
##        fit      lwr      upr
## 1 55.37474 49.24653 61.50295
## 2 55.89403 49.75781 62.03025
## 3 56.41332 50.26873 62.55791
```

```r
predict.lm(m1, newdata, interval = 'confidence')
```

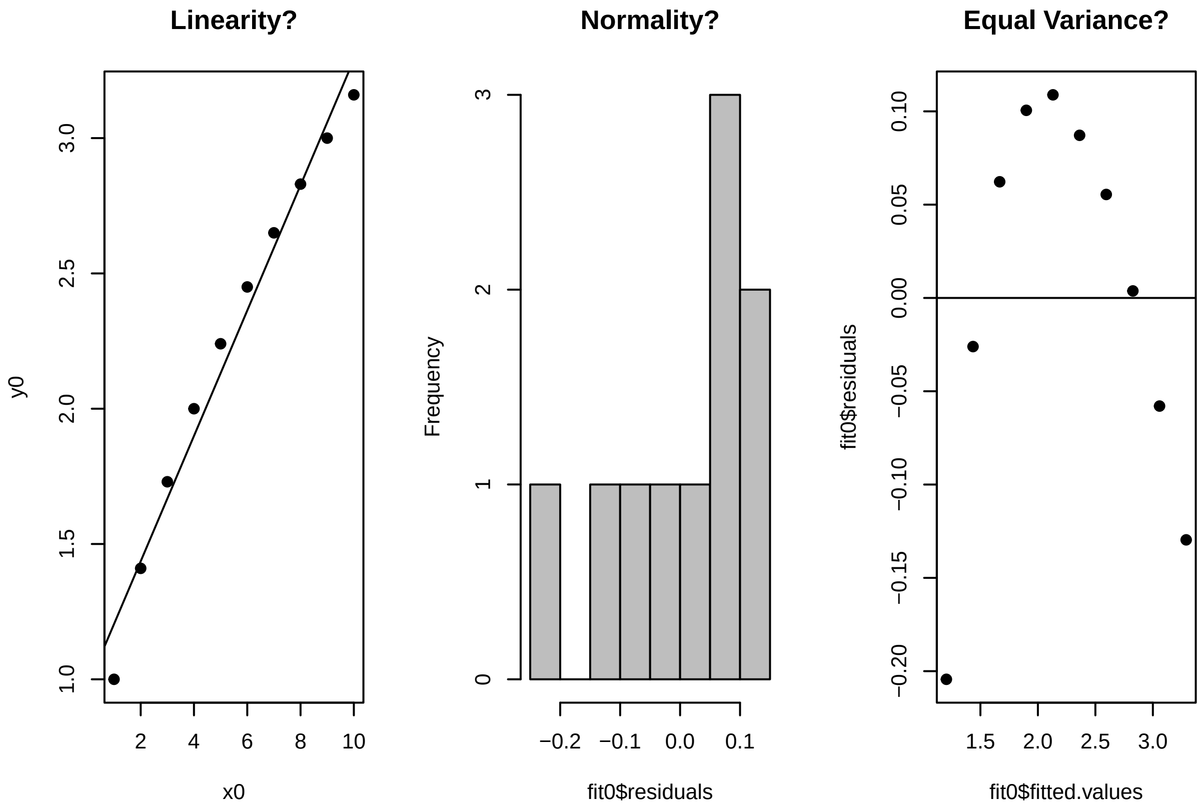
```
##        fit      lwr      upr
## 1 55.37474 54.27690 56.47258
## 2 55.89403 54.75234 57.03572
## 3 56.41332 55.22744 57.59920
```

```r
market_sample = sample_frac(adverts, 0.3, replace = FALSE)
```

Transforming Data

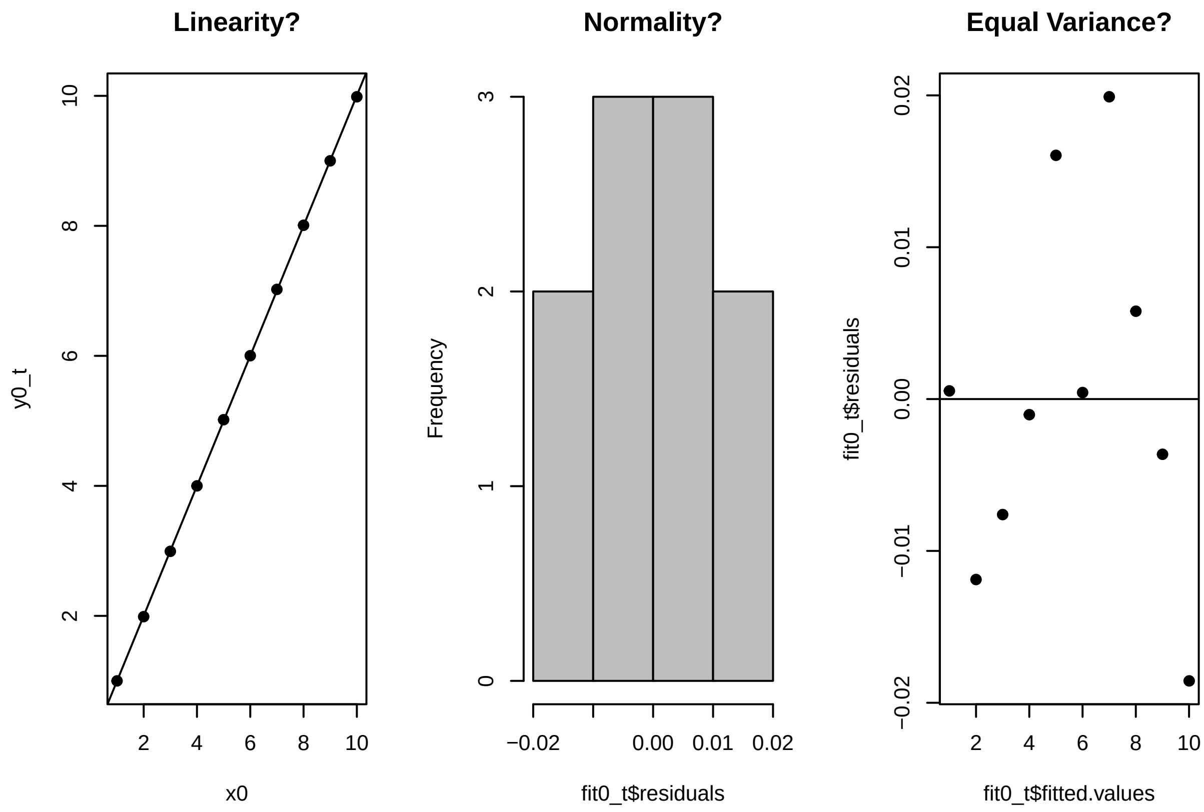
```r
x0 = c(1,2,3,4,5,6,7,8,9,10)
y0 = c(1.00,1.41,1.73,2.00,2.24,2.45,2.65,2.83,3.00,3.16)
fit0=lm(y0~x0)
```

```r
par(mfrow = c(1,3))
plot(x0,y0, pch = 19, main="Linearity?"); abline(fit0)
hist(fit0$residuals,main="Normality?", col="gray")
plot(fit0$fitted.values, fit0$residuals,
     main="Equal Variance?", pch=19); abline(h=0)
```



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

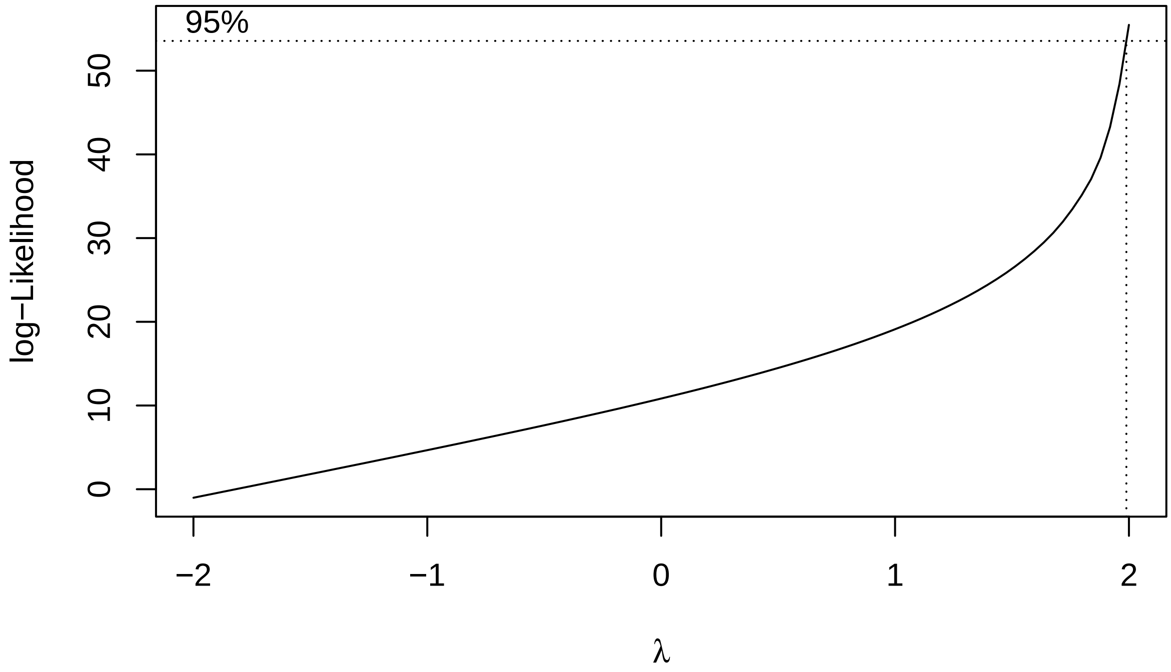
```
y0_t = y0^2
fit0_t = lm(y0_t ~ x0)
par(mfrow = c(1,3))
plot(x0,y0_t, pch = 19, main="Linearity?"); abline(fit0_t)
hist(fit0_t$residuals,main="Normality?", col="gray")
plot(fit0_t$fitted.values, fit0_t$residuals,
     main="Equal Variance?", pch=19); abline(h=0)
```



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

Boxcox graph shows to what power should the variable be increased to

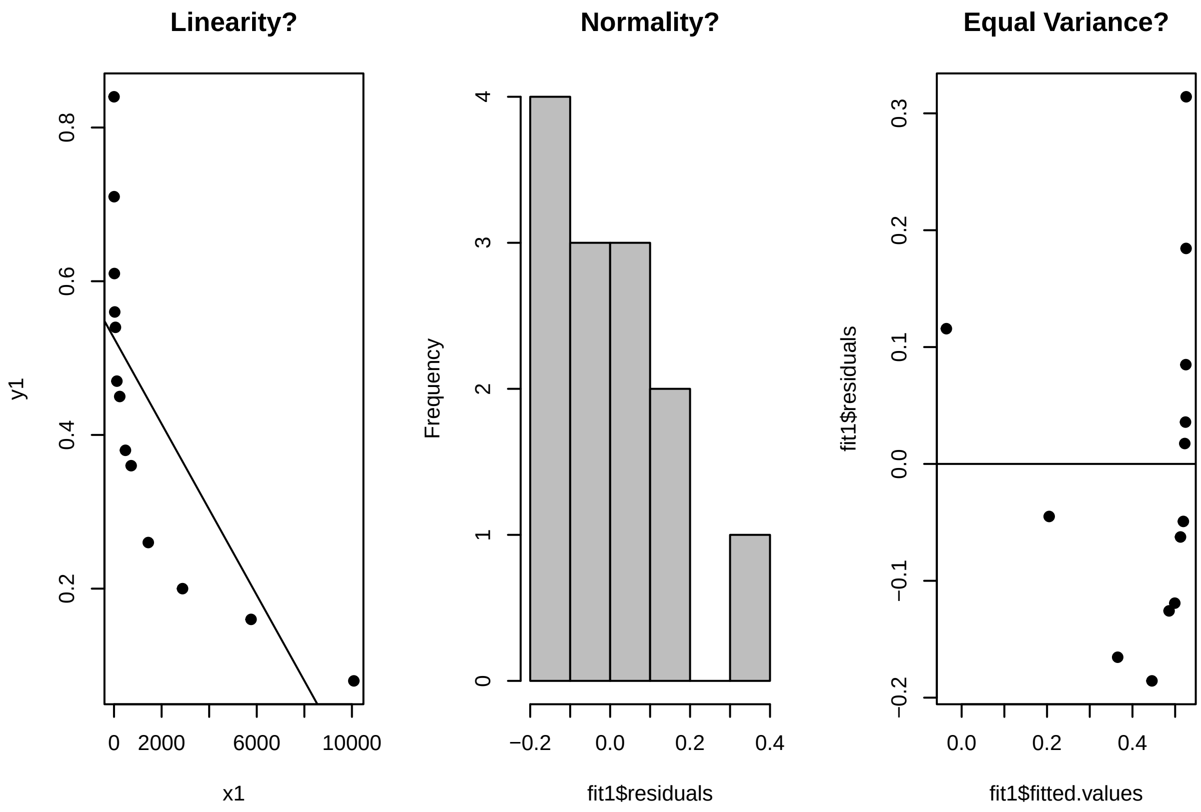
```
boxcox(fit0)
```



Exercise

```
x1 = c(1,5,15,30,60,120,240,480,720,1440,2880,5760,10080)
y1 = c(0.84,0.71,0.61,0.56,0.54,0.47,0.45,0.38,0.36,0.26,0.2,0.16,0.08)
fit1 = lm(y1~x1)
```

```
par(mfrow = c(1,3))
plot(x1,y1, pch=19, main="Linearity?"); abline(fit1)
hist(fit1$residuals,main="Normality?", col="gray")
plot(fit1$fitted.values,fit1$residuals,
     main="Equal Variance?", pch=19);abline(h=0)
```

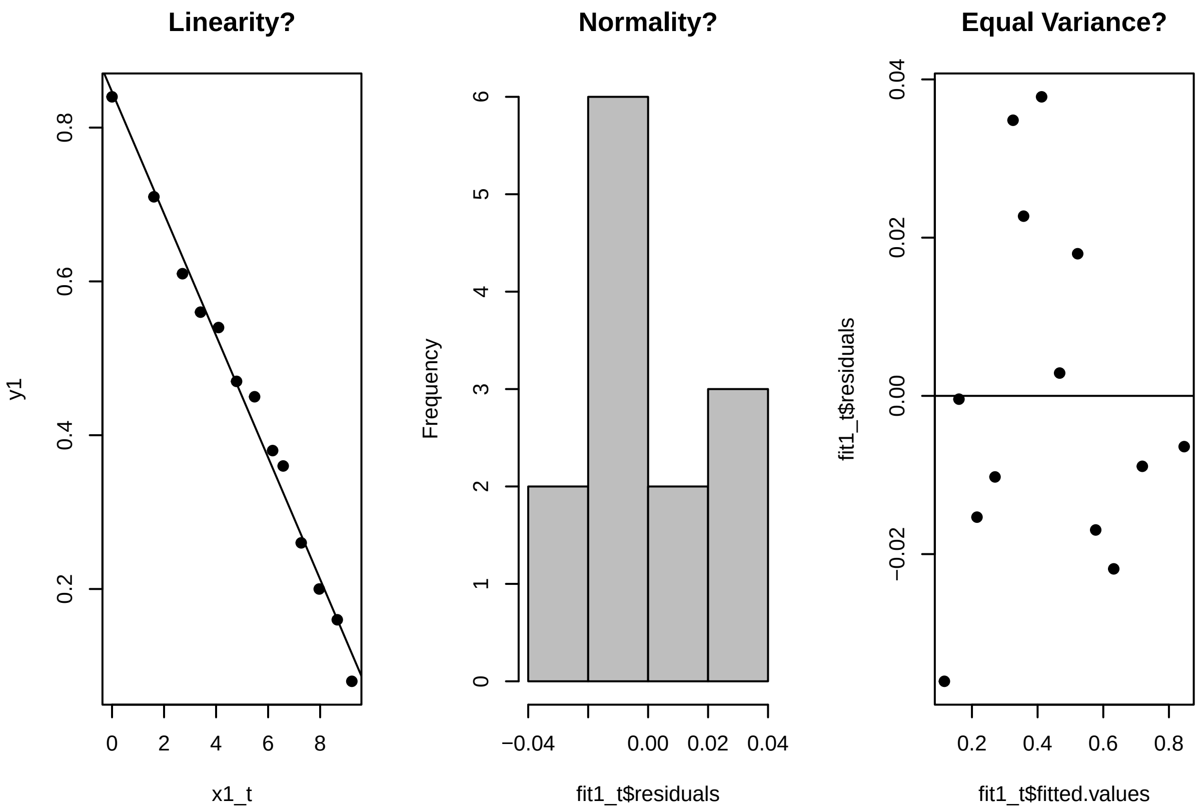


```
par(mfrow = c(1,1))
```

Transform the data x1_t=log(x)

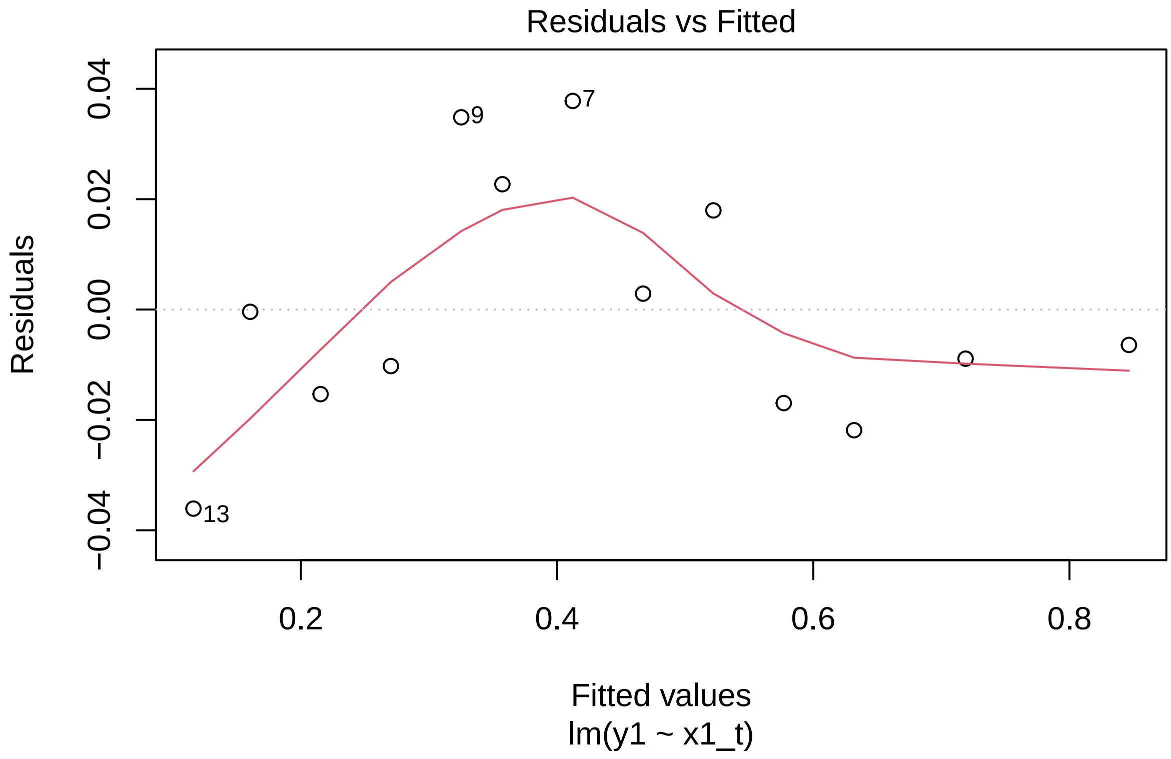
```
x1_t = log(x1)
fit1_t = lm(y1~x1_t)
```

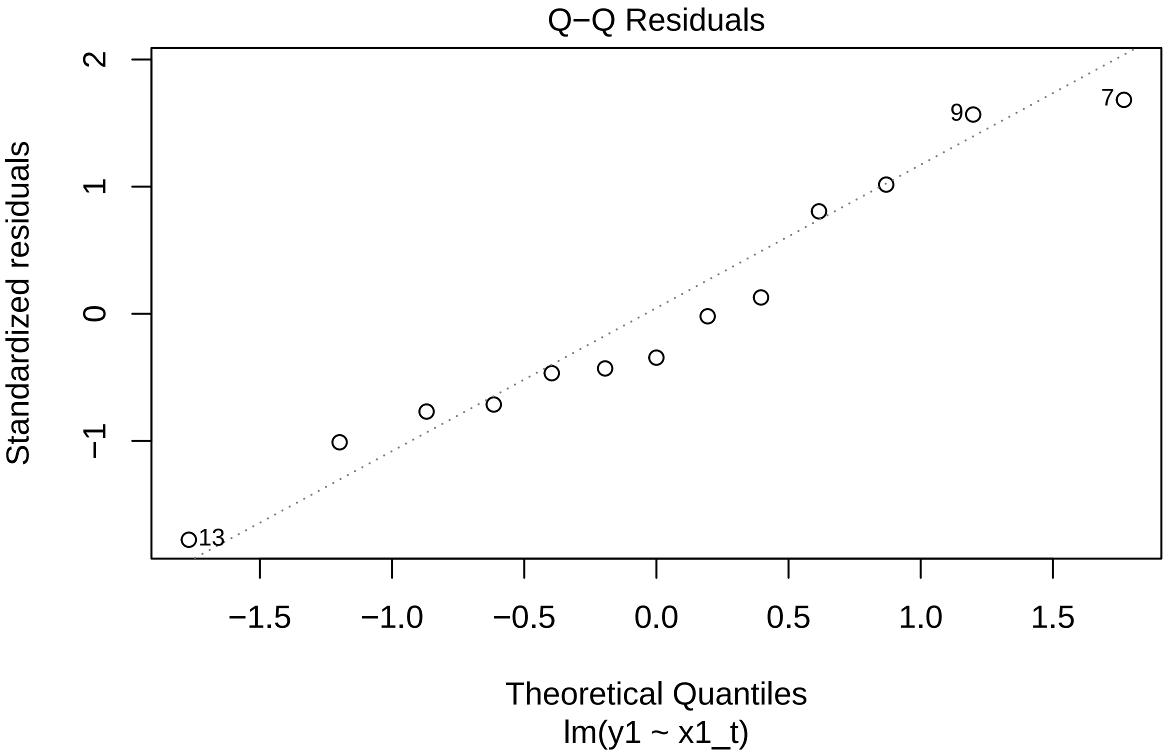
```
par(mfrow = c(1,3))
plot(x1_t,y1, pch=19, main="Linearity?"); abline(fit1_t)
hist(fit1_t$residuals,main="Normality?", col="gray")
plot(fit1_t$fitted.values,fit1_t$residuals,
     main="Equal Variance?", pch=19);abline(h=0)
```

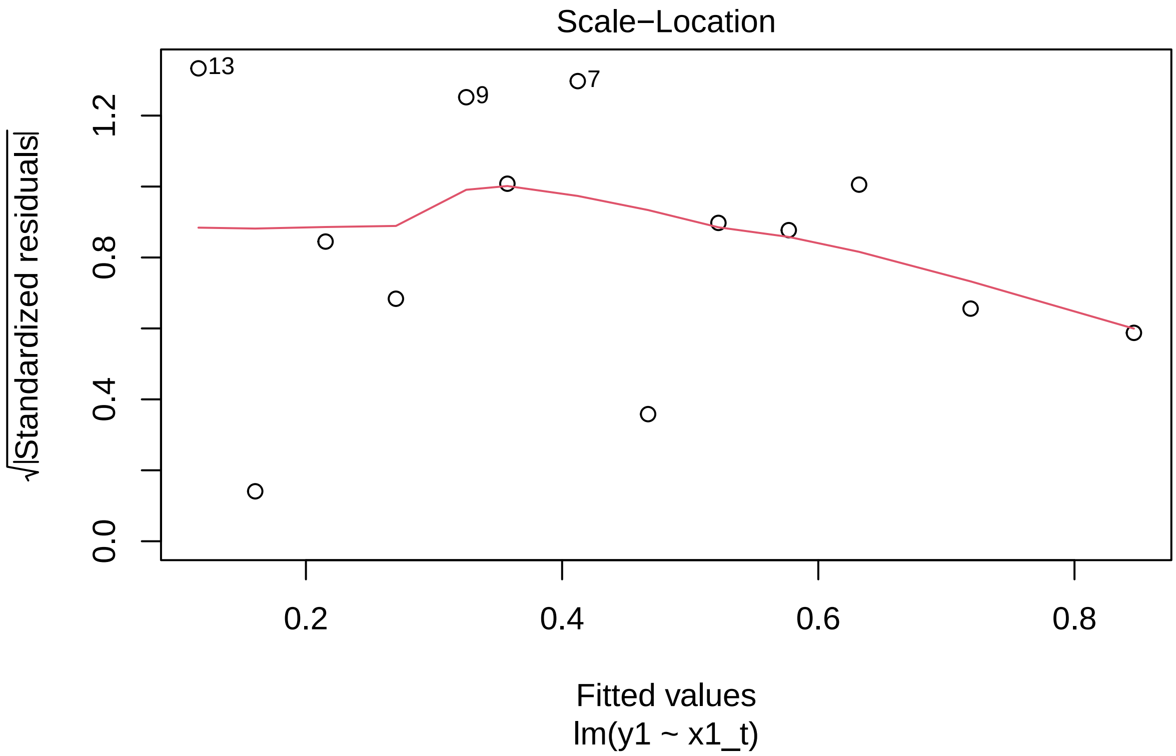


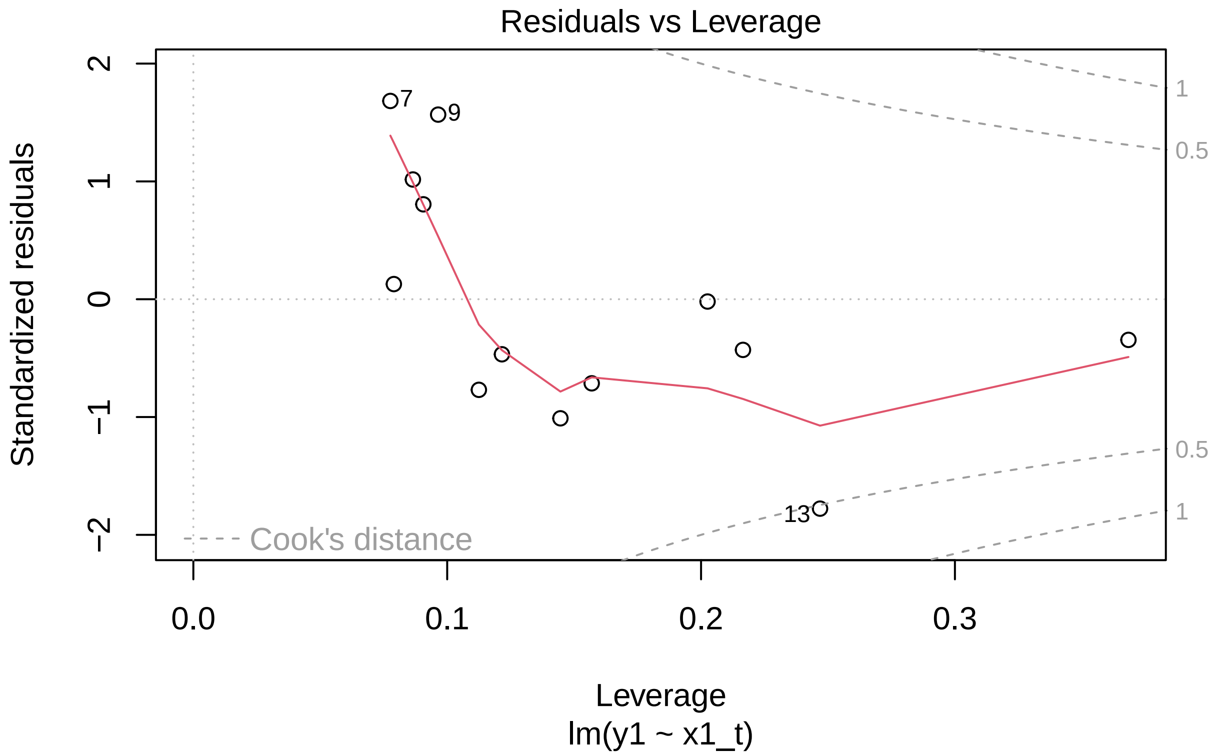
```
par(mfrow = c(1,1))
```

```
plot(fit1_t)
```


Residuals vs Fitted
Residuals
0.04
0.02
0.00
−0.02
−0.04
9
7
13
0.2
0.4
0.6
0.8
Fitted values
lm(y1 ~ x1_t)


Q−Q Residuals
Standardized residuals
2
1
0
−1
9
7
13
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Theoretical Quantiles
lm(y1 ~ x1_t)


Scale−Location
√|Standardized residuals|
0.0
0.4
0.8
1.2
0.2
0.4
0.6
0.8
13
9
7
Fitted values
lm(y1 ~ x1_t)


Residuals vs Leverage
Standardized residuals
2
1
0
−1
−2
7
9
13
1
0.5
0.5
1
Cook's distance
0.0
0.1
0.2
0.3
Leverage
lm(y1 ~ x1_t)