Assignment 3

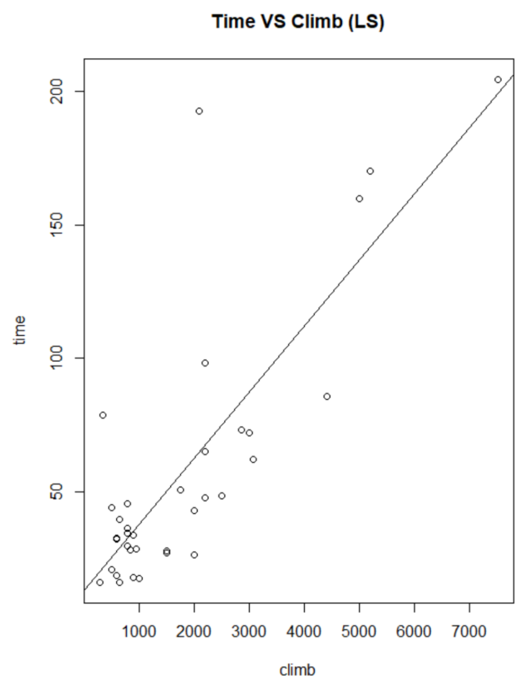
1) Problem 4.8 on Page 273 of the book by Carmona

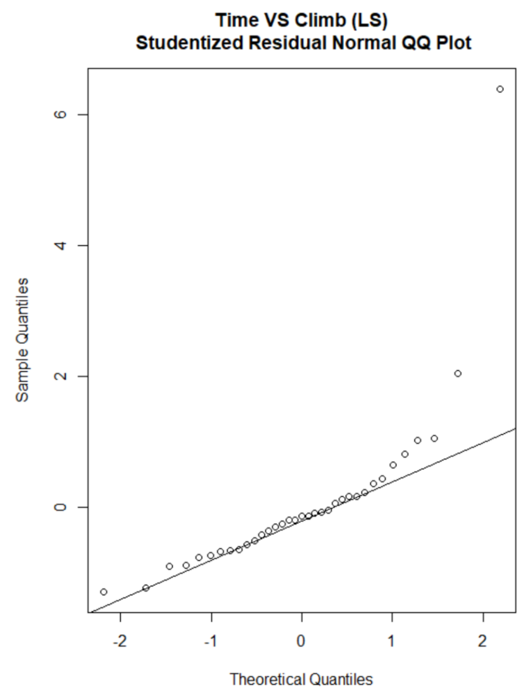
1.1)

```
HillsData <- read.table("hills.csv",header = T, sep=",")
attach(HillsData)
head(HillsData)
```

Time against Climb

```
  # LS regression 2 [LM] (time x climb)
HILLS.LSM2 <- lm(time~climb)
plot(climb,time, main="Time VS Climb (LS)")
abline(HILLS.LSM2)

  # Residual QQ Plot
qqnorm(studres(HILLS.LSM2), main="Time VS Climb (LS)\n Studentized Residual Normal QQ Plot")
qqline(studres(HILLS.LSM2))
#plot(density(studres(HILLS.LSM2))

  # Summary
summary(HILLS.LSM2)
```





The studentized residual QQ plot shows that the residuals for time against climb are not normally distributed. There seems to be some drifting upwards behavior for the points, which is more prominent in the upper tail. This shows that the upper tail distribution of sample residuals is heavier than normal distribution. The lower tails seems to indicate a semi-straight line, which shows that the lower tail distribution of sample residuals is somewhat normally distributed. There is a possibility the lower distribution of sample residuals is lighter than normal distribution.

```
Call:
lm(formula = time ~ climb)

Residuals:
    Min      1Q  Median      3Q     Max 
-36.616 -18.293  -4.215   5.103 127.706 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 12.69917    7.71050   1.647    0.109    
climb        0.02489    0.00319   7.801 5.45e-09 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 30.12 on 33 degrees of freedom
Multiple R-squared:  0.6484,    Adjusted R-squared:  0.6378 
F-statistic: 60.86 on 1 and 33 DF,  p-value: 5.452e-09
```

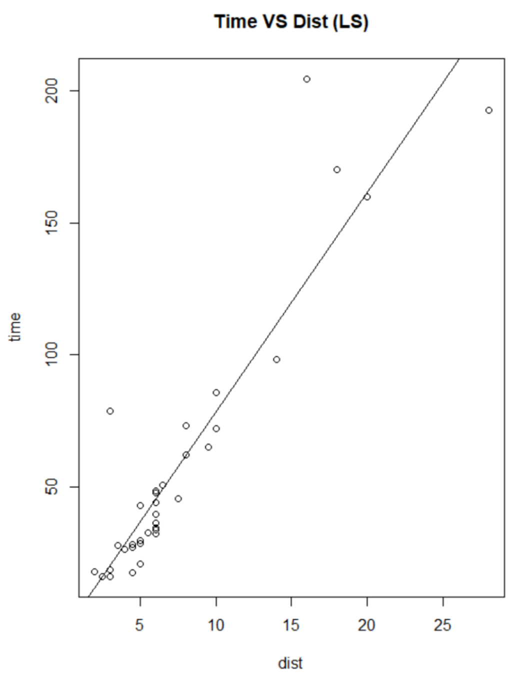
At 5% significance, critical value = **1.960**

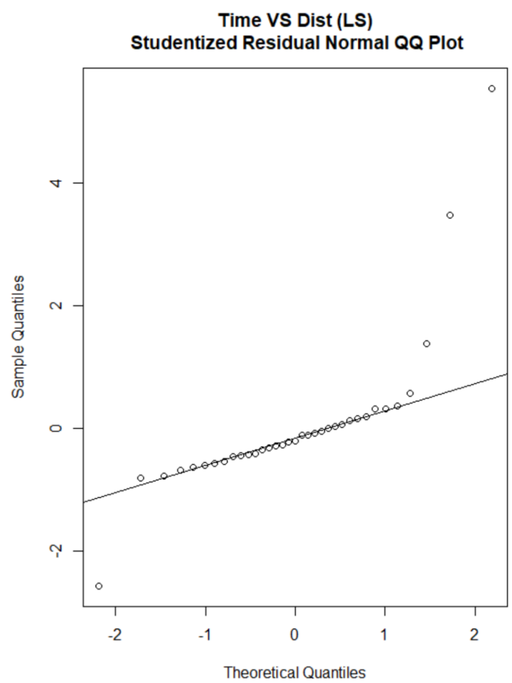
R^2	0.6484

(Intercept)	12.69917 with t-value 1.647 > 1.960 = not significant at 5%
Climb	0.02489 with t-value 7.801 > 1.960 = significant at 5%

Time again Distance

```
  # LS regression 1 [LM] (time x dist)
HILLS.LSM1 <- lm(time~dist)
plot(dist,time, main="Time VS Dist (LS)")
abline(HILLS.LSM1)

  # Residual QQ Plot
qqnorm(studres(HILLS.LSM1), main="Time VS Dist (LS)\n Studentized Residual Normal QQ Plot")
qqline(studres(HILLS.LSM1))
#plot(density(studres(HILLS.LSM1))

  # Summary
summary(HILLS.LSM1)
```





The studentized residual QQ plot shows that the residuals for time against climb are not normally distributed. There seems to be some drifting downwards and upwards behavior for the points in the lower and upper tail, respectively. This shows that both the lower and upper tail distribution of sample residuals is heavier than normal distribution.

```
Call:
lm(formula = time ~ dist)

Residuals:
    Min      1Q  Median      3Q     Max 
-35.745  -9.037  -4.201   2.849  76.170 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -4.8407     5.7562  -0.841    0.406    
dist          8.3305     0.6196  13.446 6.08e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 19.96 on 33 degrees of freedom
Multiple R-squared:  0.8456,	Adjusted R-squared:  0.841 
F-statistic: 180.8 on 1 and 33 DF,  p-value: 6.084e-15
```

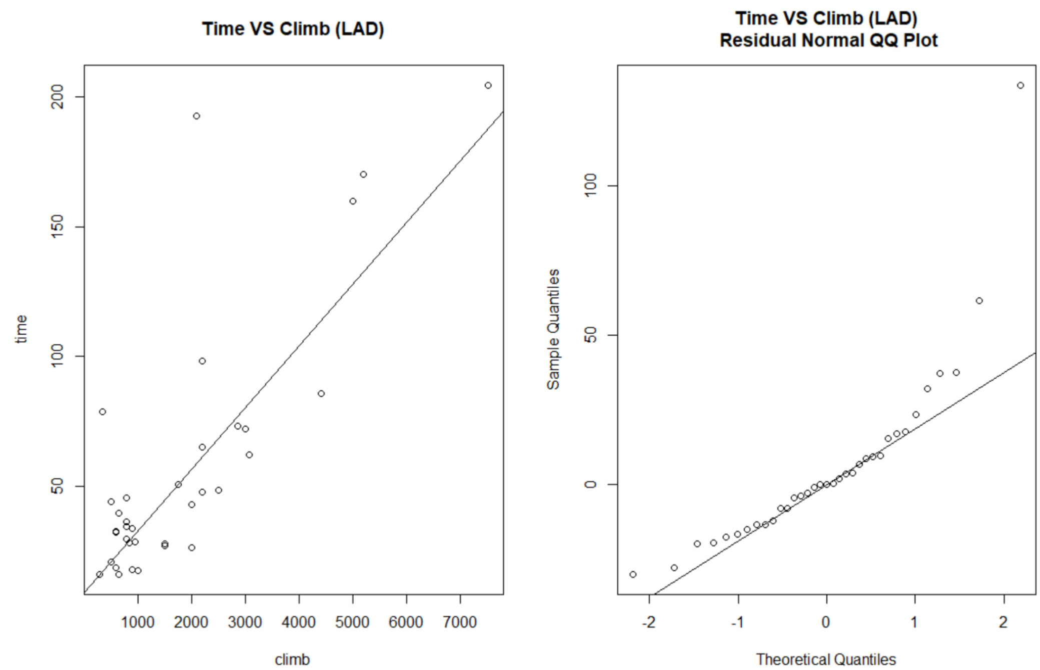
At 5% significance, critical value = **1.960**

R^2	0.8456
(Intercept)	-4.8407 with t-value -0.841 < 1.960 = not significant at 5%
Dist	8.3305 with t-value 13.446 > 1.960 = significant at 5%

1.2)

<u>Time against Climb</u>

```
# LAD regression 2 (time x climb)
HILLS.LAD2 <- rq(time~climb,0.5)
plot(climb,time, main="Time VS Climb (LAD)")
abline(HILLS.LAD2)

  # Residual QQ Plot
qqnorm(HILLS.LAD2$residuals, main="Time VS Climb (LAD)\n Residual Normal QQ Plot")
qqline(HILLS.LAD2$residuals)
#plot(density(HILLS.LAD2$residuals))

# Analog R-sqr (manual)
HILLS.LAD2.rhat <- HILLS.LAD2$coef[1]+HILLS.LAD2$coef[2]*climb
time.rbar = mean(time)

HILLS.LAD2.sse <- sum((time-HILLS.LAD2.rhat)^2)
HILLS.LAD2.tot_var <- sum((time-time.rbar)^2)

1 - (HILLS.LAD2.sse/HILLS.LAD2.tot_var)

  # Summary
summary(HILLS.LAD2)
```



The studentized residual QQ plot shows that the residuals for time against climb are not normally distributed. There seems to be some drifting upwards behavior for the points in both lower and upper tail. This shows that the upper distribution of sample residuals is heavier than normal distribution. While the lower distribution of sample residuals is lighter than normal distribution.

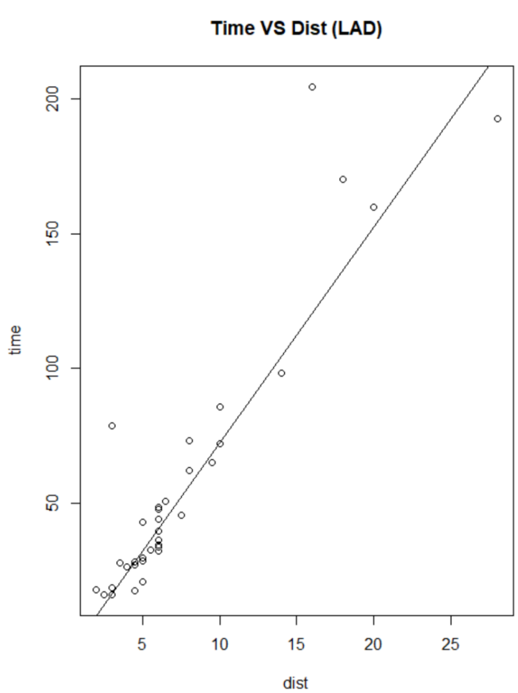
```
> 1 - (HILLS.LAD2.sse/HILLS.LAD2.tot_var)
[1] 0.6333144
>   # Summary
> summary(HILLS.LAD2)

Call: rq(formula = time ~ climb, tau = 0.5)

tau:  [1] 0.5

Coefficients:
            coefficients lower bd upper bd
(Intercept)  8.80172      2.87371 20.47260
climb        0.02383      0.01461  0.02788
```

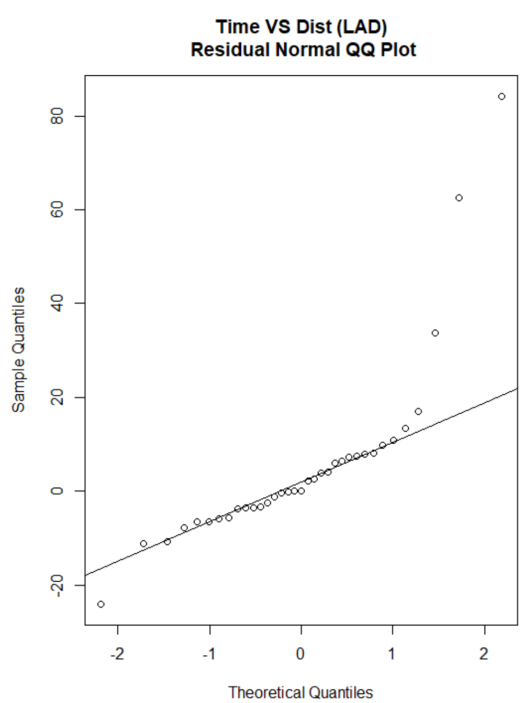
At 10% significance confidence interval bound.

Analog R^2	0.6333144
(Intercept)	8.80172 with bounds [2.87371, 20.47260] = significant at 10%
Dist	0.02383 with bounds [0.01461, 0.02788] = significant at 10%

Time against Distance

```
  # LAD regression 1 (time x dist)
HILLS.LAD1 <- rq(time~dist,0.5)
plot(dist,time, main="Time VS Dist (LAD)")
abline(HILLS.LAD1)

  # Residual QQ Plot
qqnorm(HILLS.LAD1$residuals, main="Time VS Dist (LAD)\n Residual Normal QQ Plot")
qqline(HILLS.LAD1$residuals)
#plot(density(HILLS.LAD1$residuals))

  # Analog R-sqr (manual)
HILLS.LAD1.rhat <- HILLS.LAD1$coef[1]+HILLS.LAD1$coef[2]*dist
time.rbar = mean(time)

HILLS.LAD1.sse <- sum((time-HILLS.LAD1.rhat)^2)
HILLS.LAD1.tot_var <- sum((time-time.rbar)^2)

1 - (HILLS.LAD1.sse/HILLS.LAD1.tot_var)
```





The studentized residual QQ plot shows that the residuals for time against distance are not normally distributed. There seems to be some drifting downwards and upwards behavior for the points in the lower and upper tail, respectively. This shows that both the lower and upper tail distribution of sample residuals is heavier than normal distribution.

```
> 1 - (HILLS.LAD1.sse/HILLS.LAD1.tot_var)
[1] 0.8322485
>   # Summary
> summary(HILLS.LAD1)

Call: rq(formula = time ~ dist, tau = 0.5)

tau: [1] 0.5

Coefficients:
            coefficients lower bd  upper bd
(Intercept)  -8.02273    -12.02026   3.69881
dist          8.02727      6.95710  10.46222
```

At 10% significance confidence interval bound.

Analog R^2	0.8322485
(Intercept)	-8.02273 with bounds [-12.02026, 3.69881] = not significant at 10%
Dist	8.02727 with bounds [6.95710, 10.46222] = significant at 10%

<u>Comparison</u>

Model	R^2 / Analog R^2	Intercept	Regressor
LS (Time x Climb)	0.6482	not significant at 5%	significant at 5%
LS (Time x Distance)	0.8456	not significant at 5%	significant at 5%
LAD (Time x Climb)	0.6333	significant at 10%	significant at 10%
LAD (Time x Distance)	0.8322	not significant at 10%	significant at 10%

It is clear that distance is a better regressor at predicting time, which is evident in both models. It seems that the LS model performs better in terms of R^2, which indicates that it may be a better fit. The intercept tends to not be significant at 5% / 10%, while the regressor is significant at 5% / 10%.

1.3)

```
c(min(dist),max(dist))

  # Interpolation
for (X_dist in c(5,10,15,20,25)) {
  predi2<-HILLS.LSM1$coef[1]+HILLS.LSM1$coef[2]*X_dist
  predi1<-HILLS.LAD1$coef[1]+HILLS.LAD1$coef[2]*X_dist
  predis<-round(c(predi2,predi1,predi2 - predi1),0)
  names(predis)<-c("LS", "LAD", "Diff")
  print(c(X_dist, predis))
}

  # Extrapolation
for (X_dist in c(30,35,40,45,50)) {
  predi2<-HILLS.LSM1$coef[1]+HILLS.LSM1$coef[2]*X_dist
  predi1<-HILLS.LAD1$coef[1]+HILLS.LAD1$coef[2]*X_dist
  predis<-round(c(predi2,predi1,predi2 - predi1),0)
  names(predis)<-c("LS", "LAD", "Diff")
  print(c(X_dist, predis))
}
```

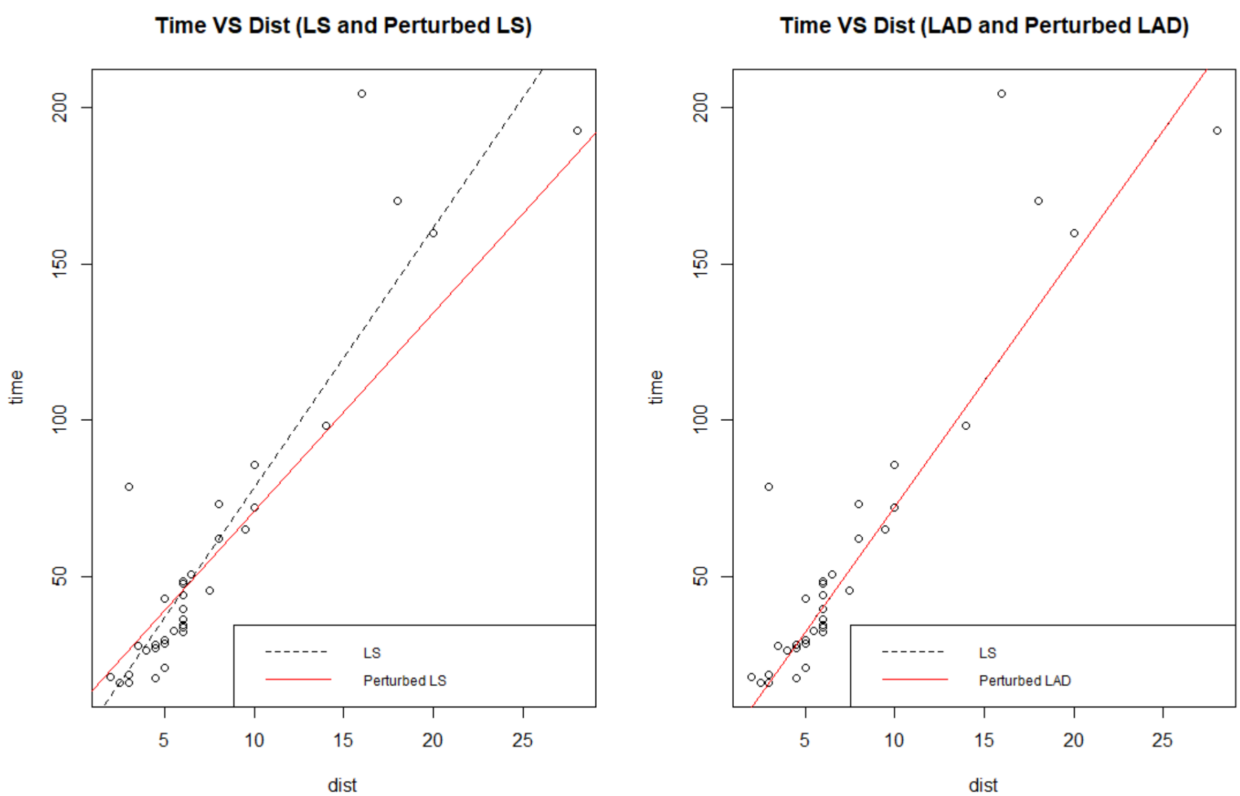
The range of the dataset for distance is between (2,28). Any predictions outside that range is considered extrapolation, while any predictions inside that range is considered interpolation.

<table>
<tr><th>Interpolation</th><th>Extrapolation</th></tr>
<tr><td><pre>
       LS LAD Diff
 5   37   32   5
       LS LAD Diff
10   78   72   6
       LS LAD Diff
15   120 112   8
       LS LAD Diff
20   162 153   9
       LS LAD Diff
25   203 193   11
</pre></td><td><pre>
      LS LAD Diff
30   245 233   12
      LS LAD Diff
35   287 273   14
      LS LAD Diff
40   328 313   15
      LS LAD Diff
45   370 353   17
      LS LAD Diff
50   412 393   18
</pre></td></tr>
</table>

1.4)

```
  # Perturbed data

Thills <- HILLS.data
colnames(Thills) <- c("TX", "Tdist", "Tclimb", "Ttime")
Thills["Ttime"][Thills["TX"] == "Lairig Ghru"] <- 92.667
Thills[Thills["TX"] == "Lairig Ghru"]
attach(Thills)
```

```
  # Scatterplot of time x dist (superimposed LS and perturbed LS)

par(mfrow=c(1,2))

plot(dist,time, main="Time VS Dist (LS and Perturbed LS)")
abline(HILLS.LSM1, lty=2) # LS

HILLS.LSM1_new <- lm(Ttime~Tdist)
abline(HILLS.LSM1_new, col="red") # perturbed LS

  # Summary
summary(HILLS.LSM1_new)

legend("bottomright", cex = 0.75, c("LS","Perturbed LS"), lty=c(2,1), col=c("black","red"))

  # Scatterplot of time x dist (superimposed LAD and perturbed LAD)

plot(dist,time, main="Time VS Dist (LAD and Perturbed LAD)")
abline(HILLS.LAD1, lty=2) # LAD

HILLS.LAD1_new <- rq(Ttime~Tdist,0.5) # Perturbed LAD
abline(HILLS.LAD1_new, col="red")

  # Summary
summary(HILLS.LAD1_new)

legend("bottomright", cex = 0.75, c("LS","Perturbed LAD"), lty=c(2,1), col=c("black","red"))
```



Parameters	LS	Perturbed LS	LAD	Perturbed LAD
Intercept	-4.8407	7.1575	-8.02273	-8.02273
Distance	8.3305	6.3573	8.02727	8.02727

The graph shows that even a small change in the data will cause the LS regression to change drastically. Both the intercept and the regressor (distance) is changed when the LS regression is perturbed. While changes in the data does not cause the LAD regression to change. Both the intercept and the regressor (distance) remain the same when the LAD regression is perturbed. This is because the LS regression and the LAD regression each use mean and median, respectively, to calculate the regression line. Perturbance in the data will less likely alter the median, while it has adverse effects on the mean. THe LAD regression is more robust.

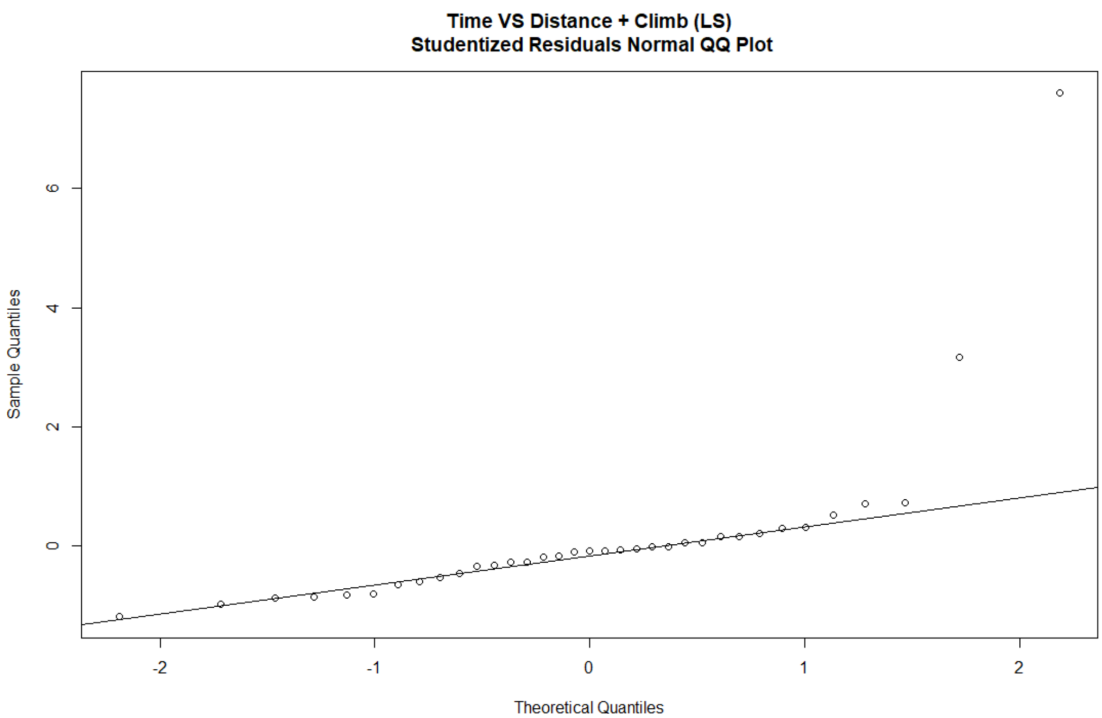
1.5)

```
  # Multiple Regression
HILLS.multi <- lm(time~dist+climb)
summary(HILLS.multi)

par(mfrow=c(1,1))

  # Residual QQ Plot
qqnorm(studres(HILLS.multi), main="Time VS Distance + Climb (LS)\n Studentized Residuals Normal QQ Plot")
qqline(studres(HILLS.multi))
#plot(density(HILLS.multi$residuals))
```

```
Call:
lm(formula = time ~ dist + climb)

Residuals:
    Min      1Q  Median      3Q     Max 
-16.215  -7.129  -1.186   2.371  65.121 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept) -8.992039   4.302734  -2.090   0.0447 *  
dist         6.217956   0.601148  10.343 9.86e-12 ***
climb        0.011048   0.002051   5.387 6.45e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 14.68 on 32 degrees of freedom
Multiple R-squared:  0.9191,	Adjusted R-squared:  0.914 
F-statistic: 181.7 on 2 and 32 DF,  p-value: < 2.2e-16
```

Multiple Regression R^2	Time against Distance R^2	Time against Climb R^2
0.9191	0.8456	0.6482

The multiple regression has a higher R^2 than both the simple regression models. The reason for that is it combines both the regressors of distance and climb to predict time. It's clear that both distance and climb are correlated to time.



The residuals QQ plot shows that the residuals for the multiple regression are not completely normally distributed. There seems to be some drifting upwards behavior for the points in the upper tail. THis shows that the upper distribution of sample residuals is heavier than normal distribution. However, the lower distribution indicates that the residuals are normally distributed. When the regression is combined it takes the effect of the two simple regressions. Because time against climb residuals tends to be normally distributed in the lower tail, the multiple regression has taken this effect. While for the upper tail, both the simple regressions tend to show heaviness compared to normal distribution, which is shown at a more exaggerated rate in the multiple regression.

2) Analysis for Google

```
GOOG <- read.table("Google.csv",header = T, sep=",")
attach(GOOG)
head(GOOG)
```

2.1)

```
# 2.1 Simple LS Regression

  # Fit model
excess_return <- rGoog-rf
GOOG.fit <- lm(excess_return~rM_ex)

  # Summary
summary(GOOG.fit)

Call:
lm(formula = excess_return ~ rM_ex)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.157875 -0.031535 -0.004447  0.030739  0.209860 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 0.005715   0.004926   1.160    0.248    
rM_ex       0.996074   0.113755   8.756 6.43e-15 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.05584 on 138 degrees of freedom
Multiple R-squared:  0.3572,	Adjusted R-squared:  0.3525 
F-statistic: 76.67 on 1 and 138 DF,  p-value: 6.431e-15
```

The R^2 of the regression model is 0.3572, which means that only 35.72% of the excess returns of GOOG can be explained by market excess return.

2.2 i)

Yes, the market (excess) return is significant in explaining the variation in the return of GOOG. Based on the F-statistics, the respective p-value is 6.431×10^{-15}. This is an extremely low p-value, which indicates that the regression is significant. Since market (excess) return is the only regressor, it implies that it is significant.

2.2 ii)

Parameter	Estimate	T value	Significance (10%, 5%, 1%)
α	0.005715	1.160 < 1.645 < 1.960 < 2.567	Not Significant at any levels

Under model diagnostics:

α is not significantly different from 0. Therefore, the α is as efficient as the optimal market under CAPM. Because we are determining whether α is significant from 0, we can use the t value obtained from the model diagnostics.

2.2 iii)

Parameter	Estimate	T value	Significance (10%, 5%, 1%)
β	0.996074	1.160 < 1.645 < 1.960 < 2.567	Significant at all levels

Under model diagnostics:

β is < 1. Therefore, the β is a little less volatile than the market under CAPM. Because we are determining whether β is significant from 1, we cannot use the t value obtained from the mode diagnostics.

Quantitative approach:

$H_0: \hat{\beta}_1 = \beta_1^0 \quad \therefore \hat{\beta}_1 = 1$

$H_1: \beta_1 \neq \beta_1^0 \quad \therefore \hat{\beta}_1 \neq 1$

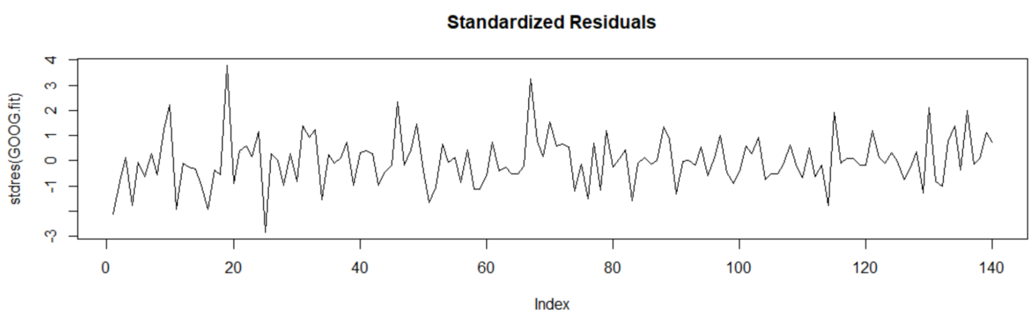
$$\left| \frac{\hat{\beta}_1 - \beta_1^0}{SE(\hat{\beta}_1)} \right| = \left| \frac{0.996074 - 1}{0.113735} \right| = 0.0345127$$

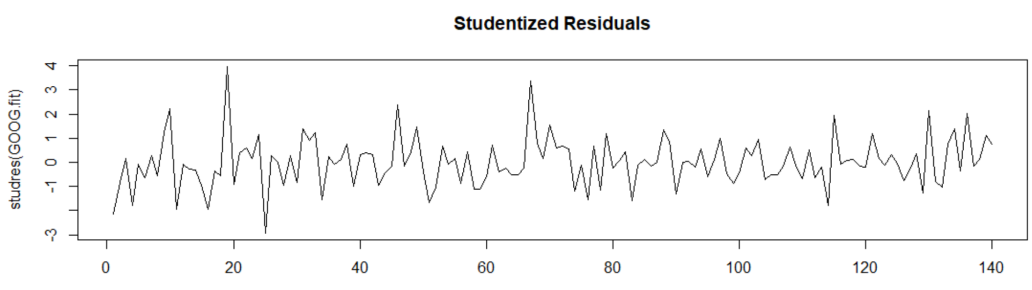
$$0.0345127 < 1.645 < 1.960 < 2.567$$

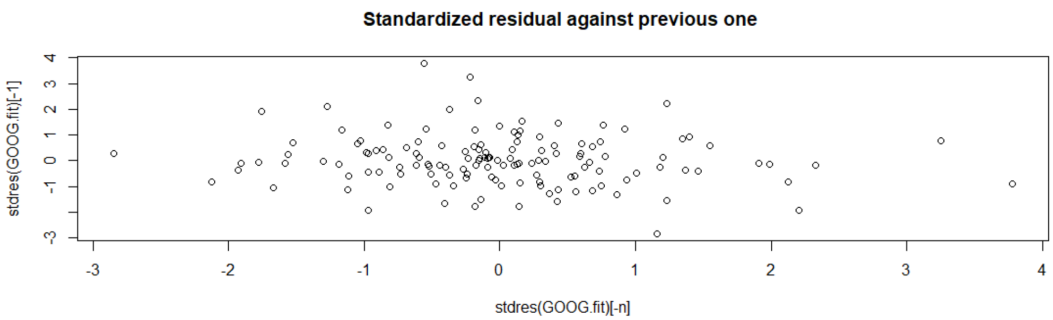
$\hat{\beta}_1$ is not significantly different from 1 β is not significantly different from 1. Therefore, the β is as volatile as the optimal market under CAPM.

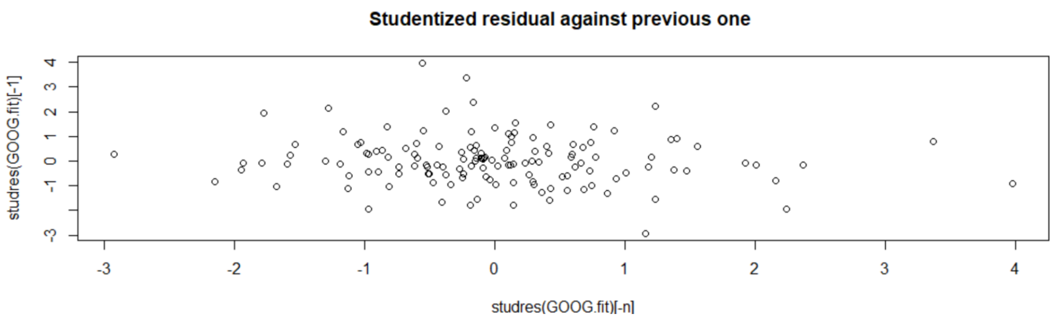
2.2 iv)

```
# 2.2iv Standardized/Studentized Residuals - model diagnostics

par(mfrow=c(2,1))

plot(stdres(GOOG.fit),type="l",main="Standardized Residuals")
plot(studres(GOOG.fit),type="l",main="Studentized Residuals")

n <- length(excess_return) - 1
plot(stdres(GOOG.fit)[-n],stdres(GOOG.fit)[-1],main="Standardized residual against previous one ")
cor(stdres(GOOG.fit)[-n],stdres(GOOG.fit)[-1])

plot(studres(GOOG.fit)[-n],studres(GOOG.fit)[-1],main="Studentized residual against previous one ")
cor(studres(GOOG.fit)[-n],studres(GOOG.fit)[-1])

acf(stdres(GOOG.fit), type = "correlation", plot = TRUE)
acf(studres(GOOG.fit), type = "correlation", plot = TRUE)

par(mfrow=c(1,2))
qqnorm(stdres(GOOG.fit),main="Q-Q Plot of Standardized Residuals")
abline(0,1,col="red")
qqnorm(studres(GOOG.fit),main="Q-Q Plot of Studentized Residuals")
abline(0,1,col="red")
```

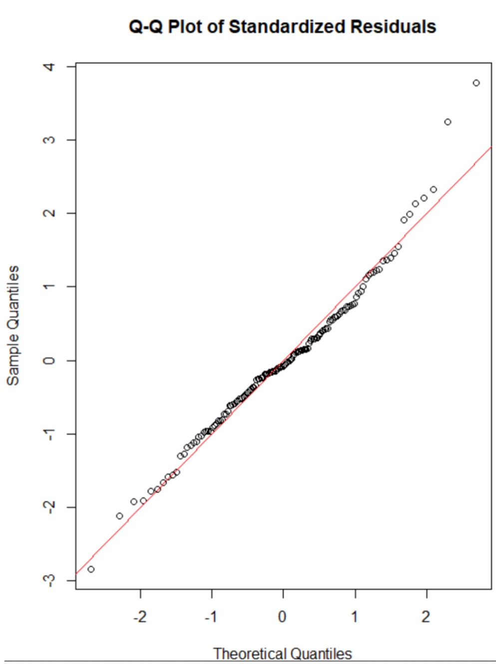


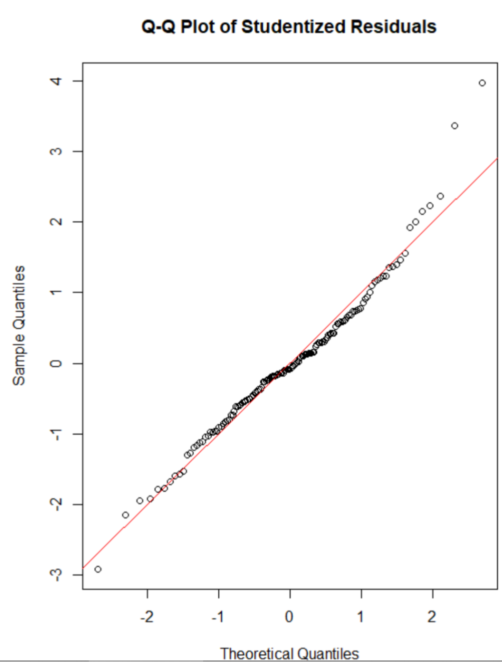




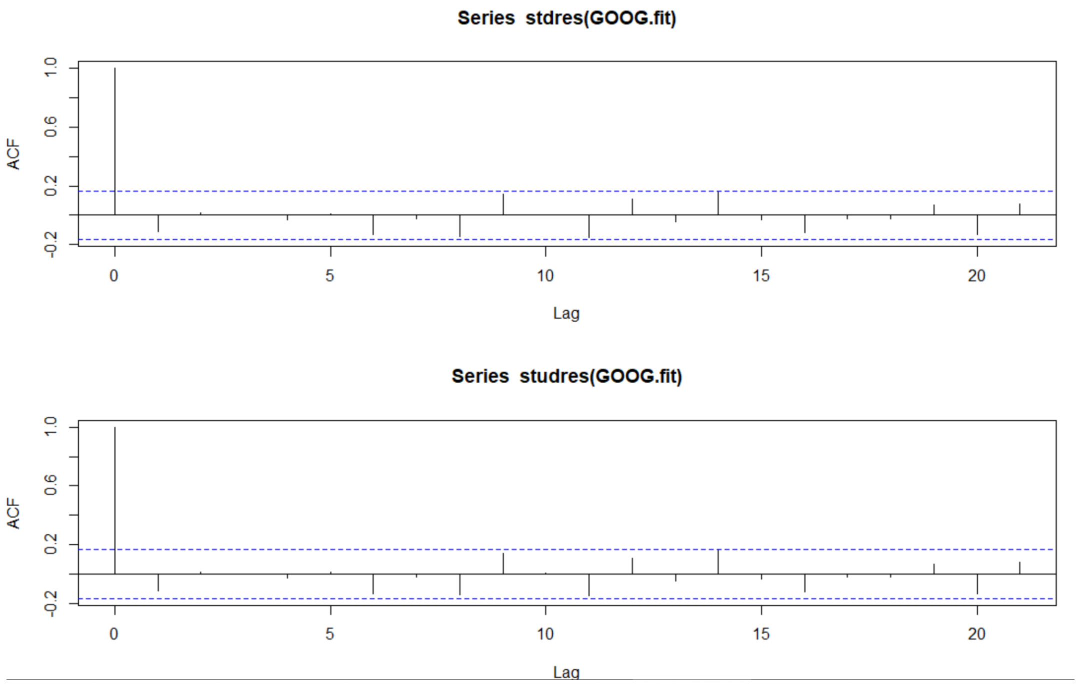


Both the residual plots show that each residual is independent and uncorrelated to each other. There is no clear trend among the residuals, which means the regression is not biased and can be used.



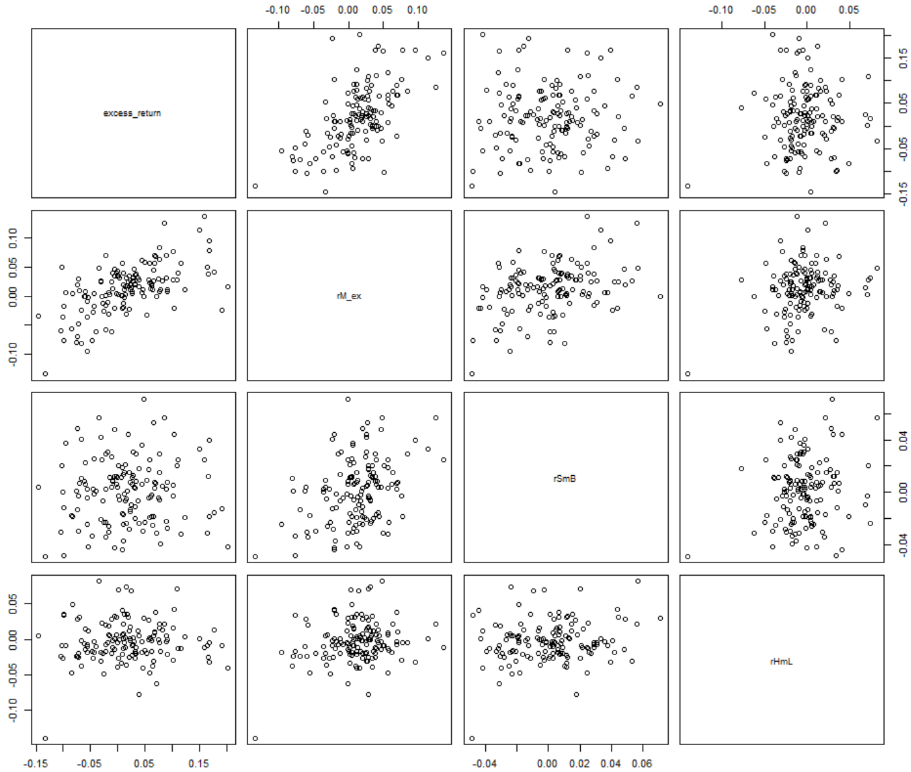


Residuals are normally distributed according to the QQ Plots.



2.3)

```
# 3. Fama-French Three-factor Model (LS Regression)

pairs(cbind(excess_return,rM_ex,rSmB,rHmL))

FF3factor <- lm(excess_return ~ rM_ex + rSmB + rHmL)
summary(FF3factor)
```




```
Call:
lm(formula = excess_return ~ rM_ex + rSmB + rHmL)

Residuals:
      Min        1Q    Median        3Q       Max
-0.152989 -0.034296 -0.000344  0.024856  0.206244

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.003929   0.004872   0.806  0.42140
rM_ex        1.142631   0.120601   9.474  < 2e-16 ***
rSmB        -0.578906   0.204550  -2.830  0.00536 **
rHmL        -0.151607   0.163848  -0.925  0.35646
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.05437 on 136 degrees of freedom
Multiple R-squared:  0.3993,    Adjusted R-squared:  0.3861
F-statistic: 30.14 on 3 and 136 DF,  p-value: 5.294e-15
```

2.4)

Parameter	P-value	Significance (0.1, 0.05, 0.01)	Significance

$\beta_1 + \beta_2 + \beta_3$ (Whole)	5.294×10^{-15}	**5.294×10^{-15}** < 0.1 < 0.05 < 0.01	Yes
α (Intercept)	0.42140	0.1 < 0.05 < 0.01 < **0.42140**	No
β_1 (rM_ex)	2×10^{-16}	**2×10^{-16}** < 0.1 < 0.05 < 0.01	Yes
β_2 (rSmB)	0.00536	**0.00536** < 0.1 < 0.05 < 0.01	Yes
β_3 (rHmL)	0.35646	0.1 < 0.05 < 0.01 < **0.35646**	No

2.5)

```
# 5. Single Factor vs Multi Factor (variation explanation)
round(c(summary(GOOG.fit)$r.squared, summary(FF3factor)$r.squared),3)
```

Single Factor R^2	Multi-Factor R^2
0.357	0.399

According to R^2, the single factor model can explain 35.7% of the variation in GOOG return, while the multi-factor model can explain 39.9% of the variation.

2.6)

```
# 6. Single Factor vs Multi Factor (statistically significance)
anova(GOOG.fit,FF3factor)
```

```
Analysis of Variance Table

Model 1: excess_return ~ rM_ex
Model 2: excess_return ~ rM_ex + rSmB + rHmL
  Res.Df     RSS Df Sum of Sq      F   Pr(>F)
1    138 0.43029
2    136 0.40207  2  0.028223 4.7732 0.009921 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The 3-factor model explains statistically significantly more variation than the single factor model. According to the p-value (0.00921 < < 0.1 < 0.05 < 0.01), it is significant at 99%.

Single factor and the multi-factor model has a R^2 of 35.7% and 39.9%, respectively, which is a 4.2% difference. This means that the multi-factor model is ~11.76% better than the single factor model.