R Notebook

This is an R Markdown (http://rmarkdown.rstudio.com) Notebook. When you execute code within the notebook, the results appear beneath the code.

Try executing this chunk by clicking the *Run* button within the chunk or by placing your cursor inside it and pressing *Ctrl+Shift+Enter*.

```
library(MASS)
library(mosaic)
```

```
## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2
```

```
##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by thi
s.
```

```
##
## Attaching package: 'mosaic'
```

```
## The following objects are masked from 'package:dplyr':
##
##     count, do, tally
```

```
## The following object is masked from 'package:Matrix':
##
##     mean
```

```
## The following object is masked from 'package:ggplot2':
##
##     stat
```

```
## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var
```

```
## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
library(ggplot2)
library(dplyr)
library(resampledata)
```

```
##
## Attaching package: 'resampledata'
```

```
## The following object is masked from 'package:datasets':
##
##     Titanic
```

```
library(nlme)
```

```
##
## Attaching package: 'nlme'
```

```
## The following object is masked from 'package:resampledata':
##
##     Spruce
```

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

```
#Part I
dataset <- read.csv("C:\\Users\\safah\\Documents\\Winter24\\bondsdata.csv")
head(dataset)
```

	season <int>	hrat <dbl>
1	1987	0.045372
2	1988	0.044610
3	1989	0.032759
4	1990	0.063584
5	1991	0.049020
6	1992	0.071882

6 rows

```
tail(dataset)
```

	season <int>	hrat <dbl>
10	1996	0.081238
11	1997	0.075188
12	1998	0.067029
13	1999	0.095775
14	2000	0.102083
15	2001	0.153400

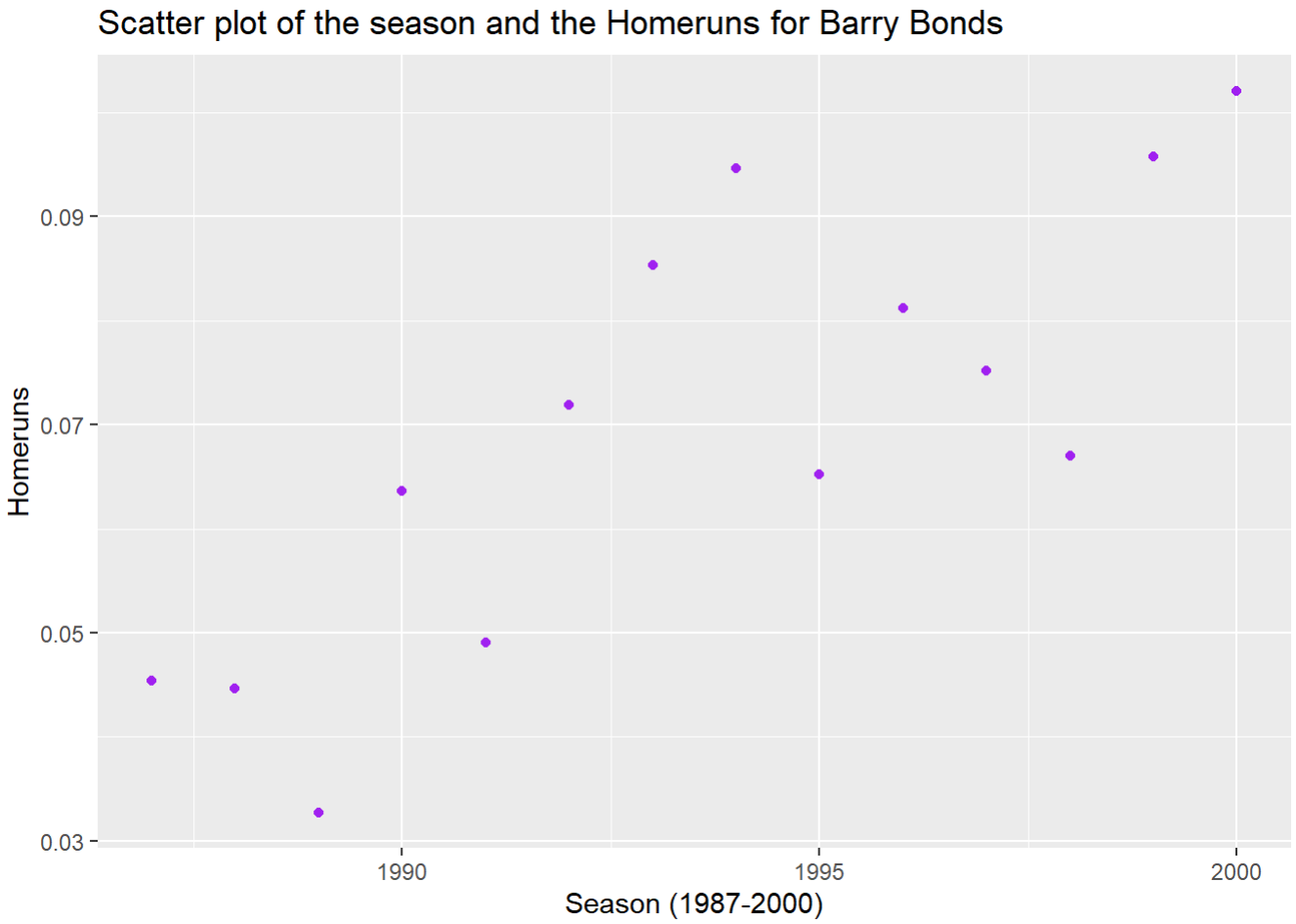
6 rows

```
#Step 1: Data Filtering

data_set = filter(dataset, season != 2001)
tail(data_set)
```

	season <int>	hrat <dbl>
9	1995	0.065217
10	1996	0.081238
11	1997	0.075188
12	1998	0.067029
13	1999	0.095775
14	2000	0.102083

6 rows

```
#Scatterplot for the relationship between the season and the homeruns

ggplot(data_set, aes(x=season, y = hrat)) +
  geom_point(color = "purple") +
  labs(title = "Scatter plot of the season and the Homeruns for Barry Bonds", x = "Season (1987-2000)", y ="Homeruns")
```




```
#There seems to be a positive linear relationship between the seasons and the number of home-run
s.

#Step 2: Modeling the linear regression

Y = favstats(data_set$hrat)
X = favstats(data_set$season)
Beta1 = sum((data_set$season-X$mean)*(data_set$hrat - Y$mean))/sum((data_set$season-X$mean)^2)

Beta1
```

```
## [1] 0.004044169
```

```
Beta2 = Y$mean - Beta1*X$mean
Beta2
```

```
## [1] -7.992499
```

```
Bonds_reg <- lm(hrat ~ season, data = data_set)
summary(Bonds_reg)
```

```
##
## Call:
## lm(formula = hrat ~ season, data = data_set)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.020722 -0.009931  0.001841  0.007701  0.023055
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -7.9924993  1.7566775  -4.550 0.000666 ***
## season       0.0040442  0.0008812   4.589 0.000622 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01329 on 12 degrees of freedom
## Multiple R-squared:  0.6371, Adjusted R-squared:  0.6068
## F-statistic: 21.06 on 1 and 12 DF,  p-value: 0.0006222
```

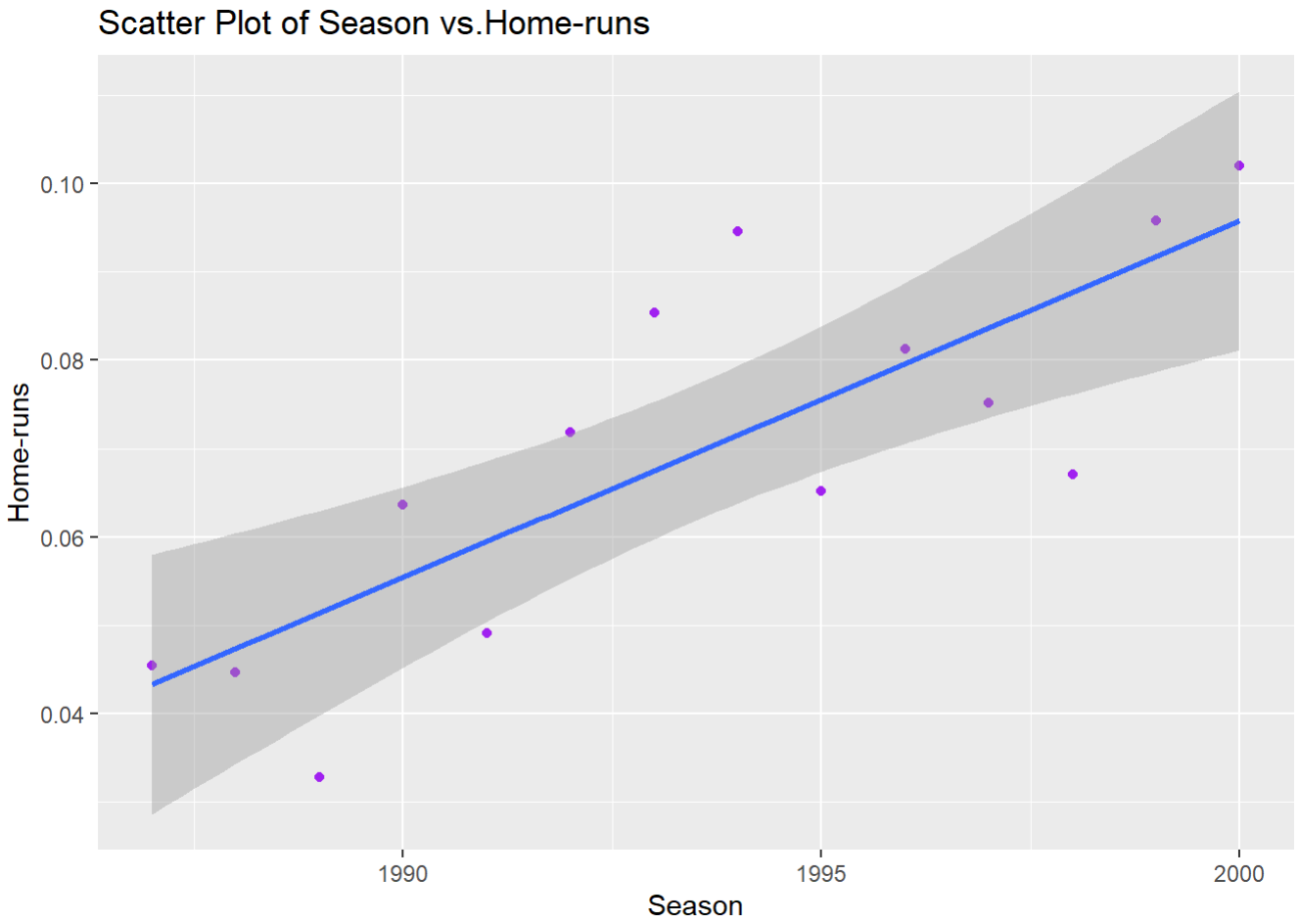
```
#y= -7.9924993+(0.0040442*x)

#Step 3: R-squared

summary(Bonds_reg)$r.squared
```

```
## [1] 0.6370504
```

```
rsquared(Bonds_reg)
```

```
## [1] 0.6370504
```

```
#r_squared is equal to 0.6370504

#Given that the r-squared value is equal to 63.71%, indicating a high r-squared. This means that
the regression model is a good fit for the estimates. Therefore, the variability of the dependen
t variable, being the home-runs in this case, can be explained by the season.

ggplot(data_set, aes(x=season, y = hrat)) +
  geom_point(color = "purple") + stat_smooth(method = "lm",
                                             formula = y ~ x, geom = "smooth") + labs(title = "S
catter Plot of Season vs.Home-runs", x = "Season", y ="Home-runs")
```




```
#predictions

er <- Bonds_reg$residuals #residuals of the regression model
er
```

```
##           1           2           3           4           5           6
##  0.002107029 -0.002699141 -0.018594310  0.008186521 -0.010421648  0.008396182
##           7           8           9          10          11          12
##  0.017813013  0.023054844 -0.010401325  0.001575505 -0.008518664 -0.020721833
##          13          14
##  0.003979998  0.006243829
```

```
stand_err <- rstandard(Bonds_reg) #standardized residual of the regression model
stand_err
```

```
##          1          2          3          4          5          6          7
##  0.1839298 -0.2276729 -1.5268222  0.6585639 -0.8260086  0.6590734  1.3916219
##          8          9         10         11         12         13         14
##  1.8011341 -0.8164707  0.1248729 -0.6852831 -1.7015181  0.3357134  0.5450454
```

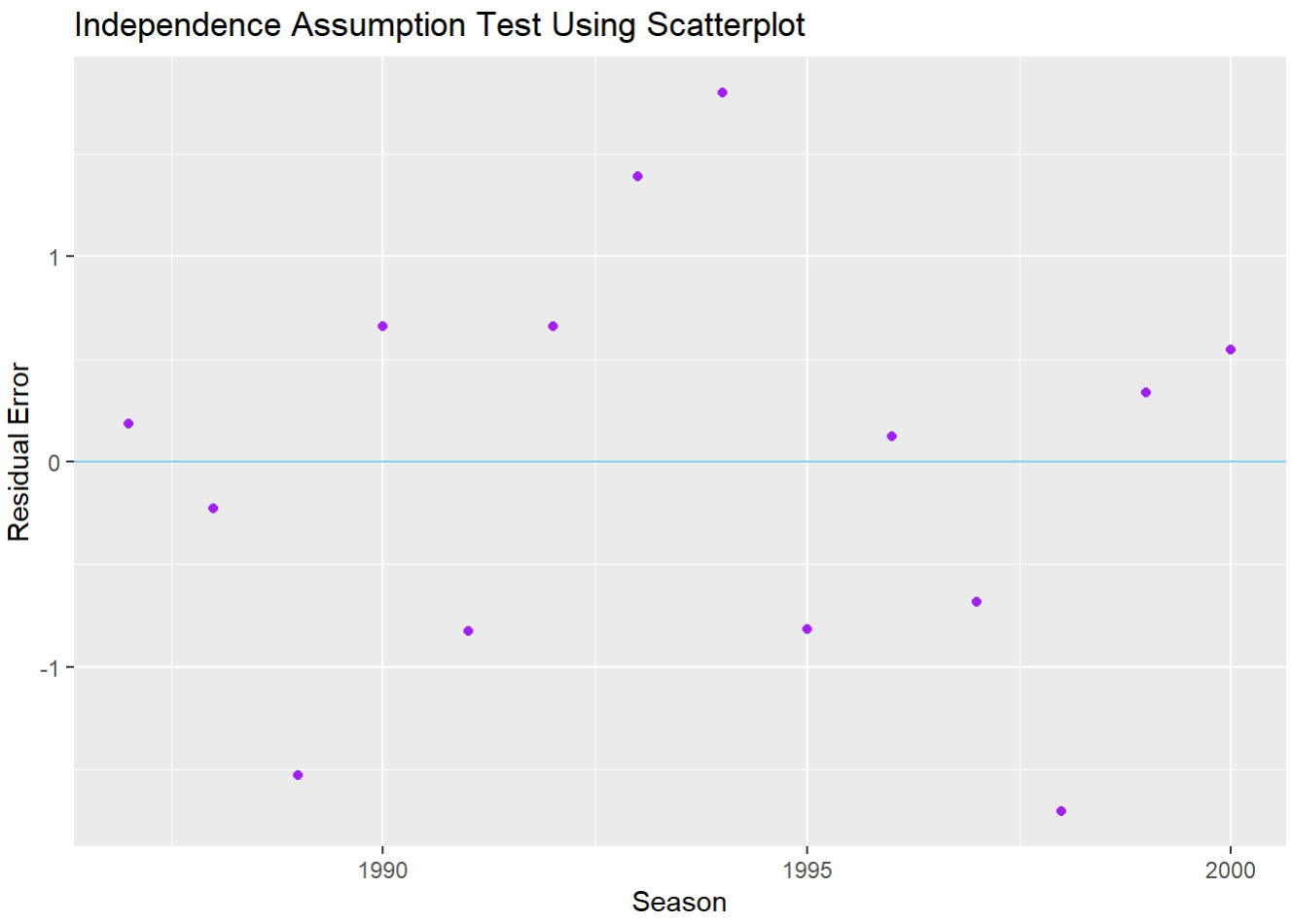
```
homeruns = data_set$hrat
seas = data_set$season
sum <- summary(Bonds_reg)$sigma
sum
```

```
## [1] 0.01329124
```

```
estimates <- Bonds_reg$fitted.values
estimates
```

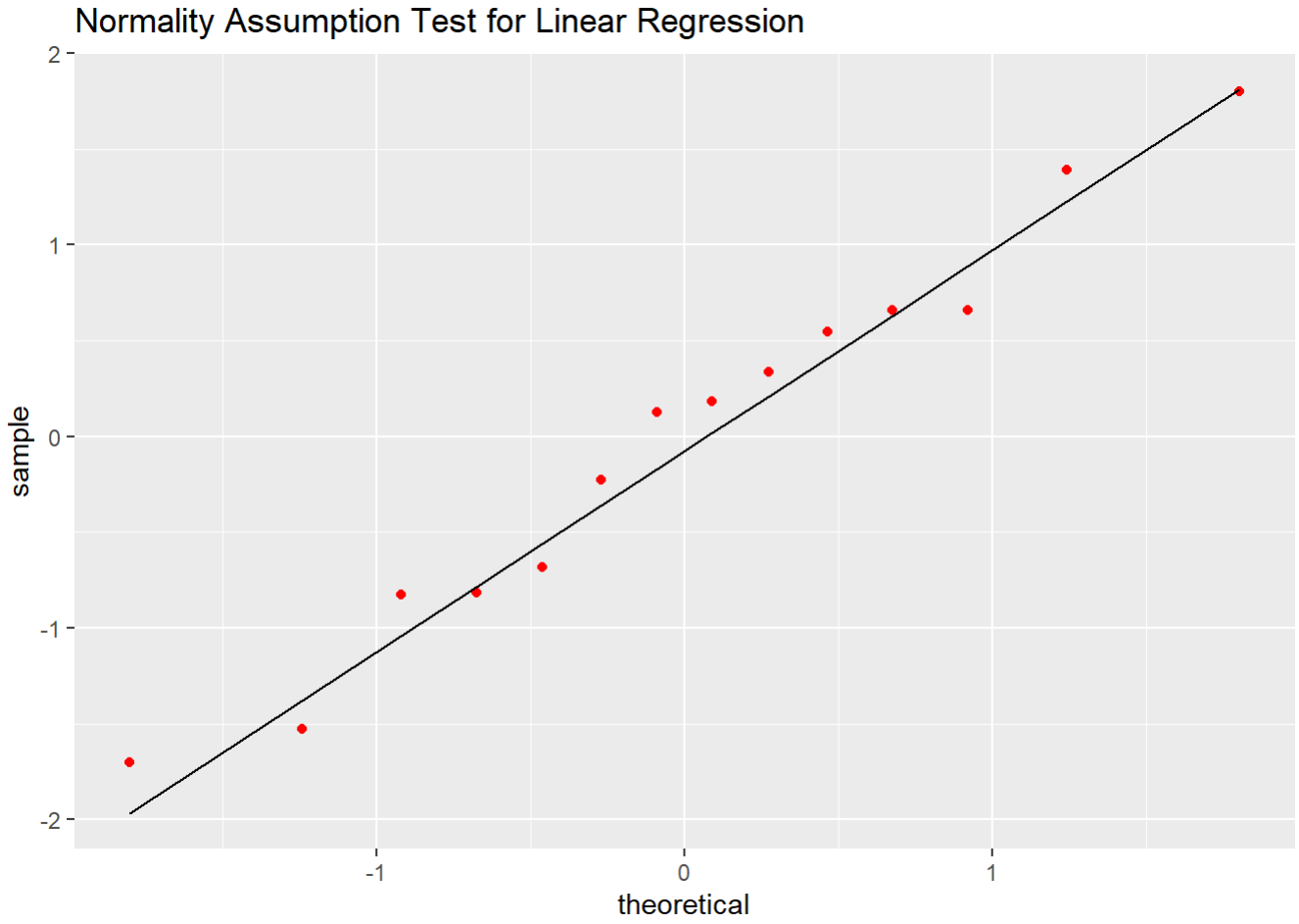
```
##          1          2          3          4          5          6          7
## 0.04326497 0.04730914 0.05135331 0.05539748 0.05944165 0.06348582 0.06752999
##          8          9         10         11         12         13         14
## 0.07157416 0.07561833 0.07966249 0.08370666 0.08775083 0.09179500 0.09583917
```

```
Bonds <- data.frame(seas, homeruns, estimates, er, stand_err)

#Conditions:
ggplot(Bonds, aes(x=seas, y= stand_err))+geom_point(color ="purple")+geom_hline(yintercept = 0,
color="skyblue")+ggtitle("Independence Assumption Test Using Scatterplot")+ylab("Residual Erro
r")+xlab("Season")
```

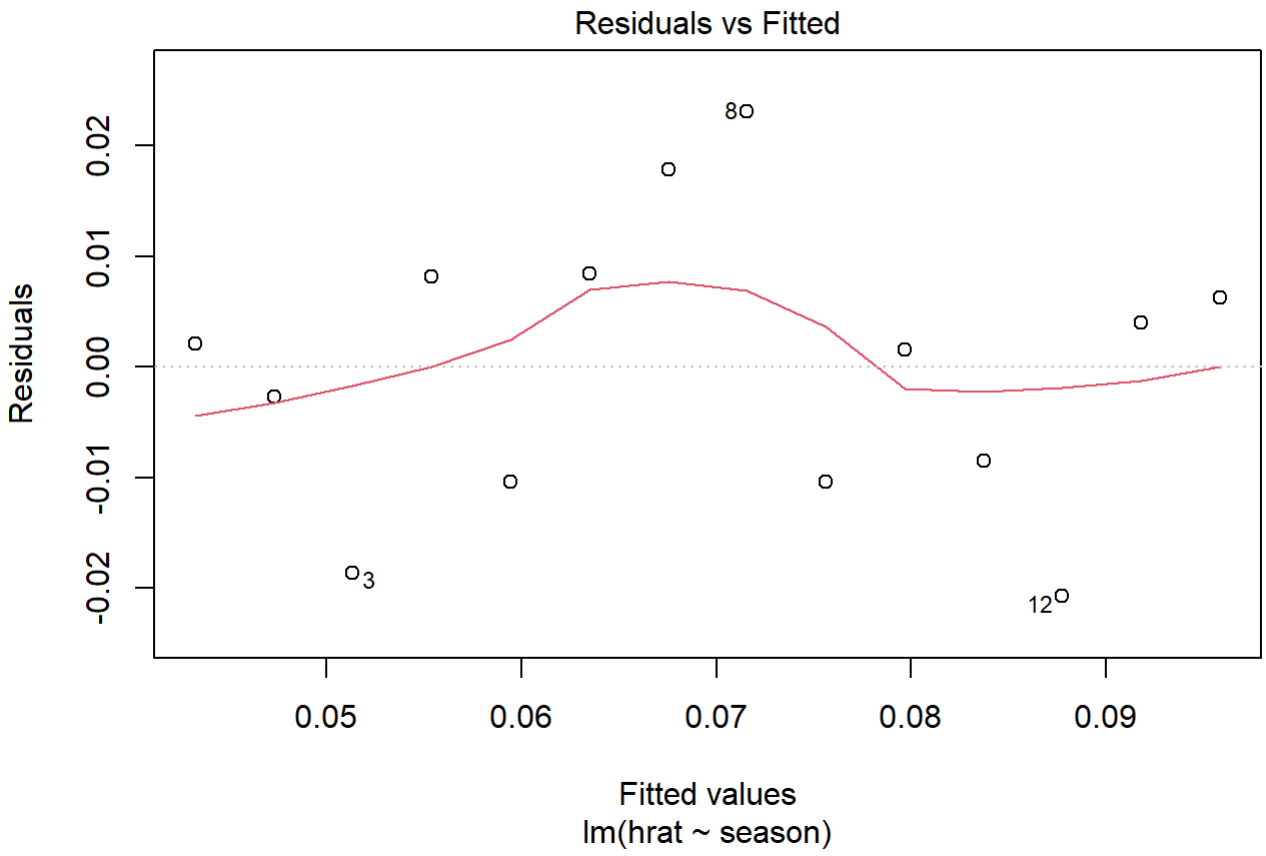


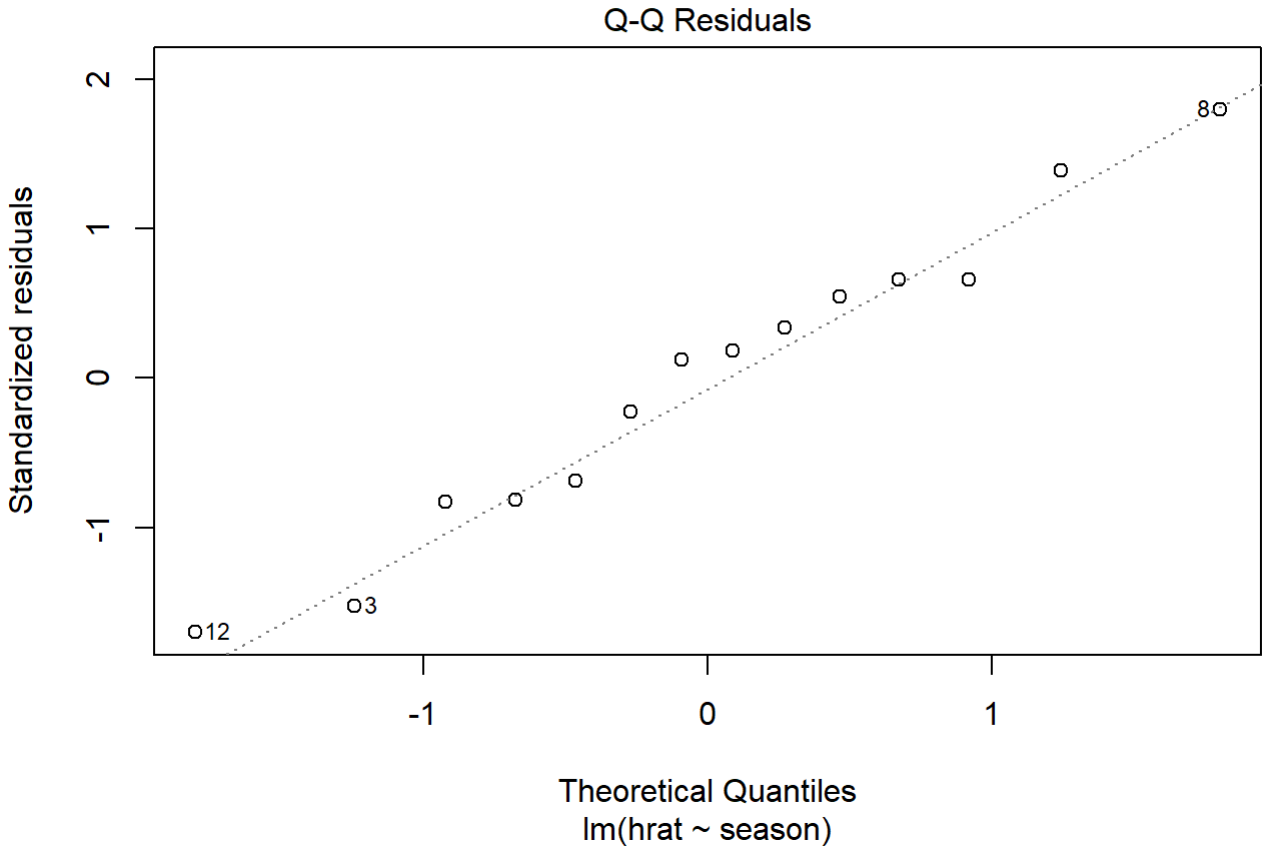
```
#homoscedastic, equal variances

#Normality Assumption
ggplot(Bonds, aes(sample=stand_err))+stat_qq(col="red")+stat_qqline(col="black")+ggtitle("Normal
ity Assumption Test for Linear Regression")
```

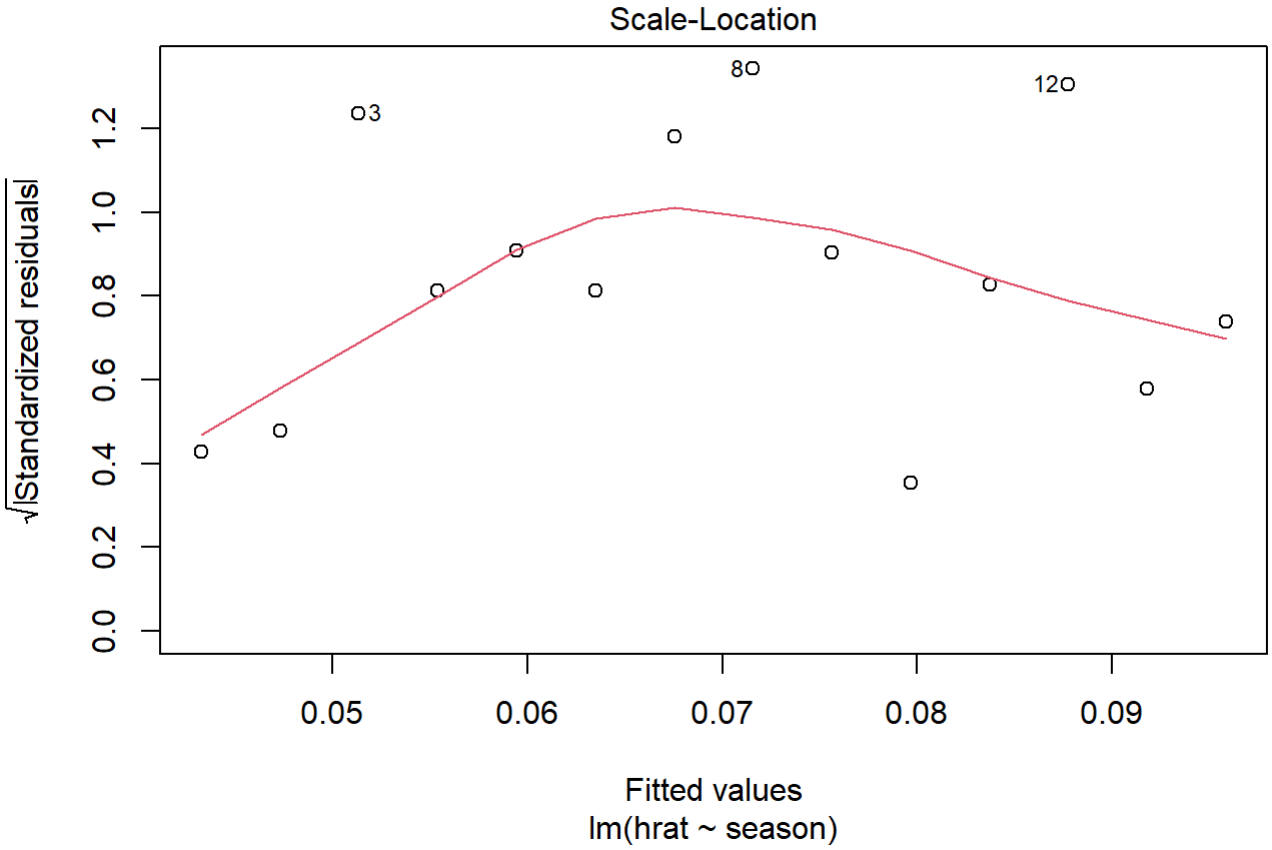
```
## Warning: The following aesthetics were dropped during statistical transformation: sample
## ℹ This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## ℹ Did you forget to specify a `group` aesthetic or to convert a numerical
##   variable into a factor?
```

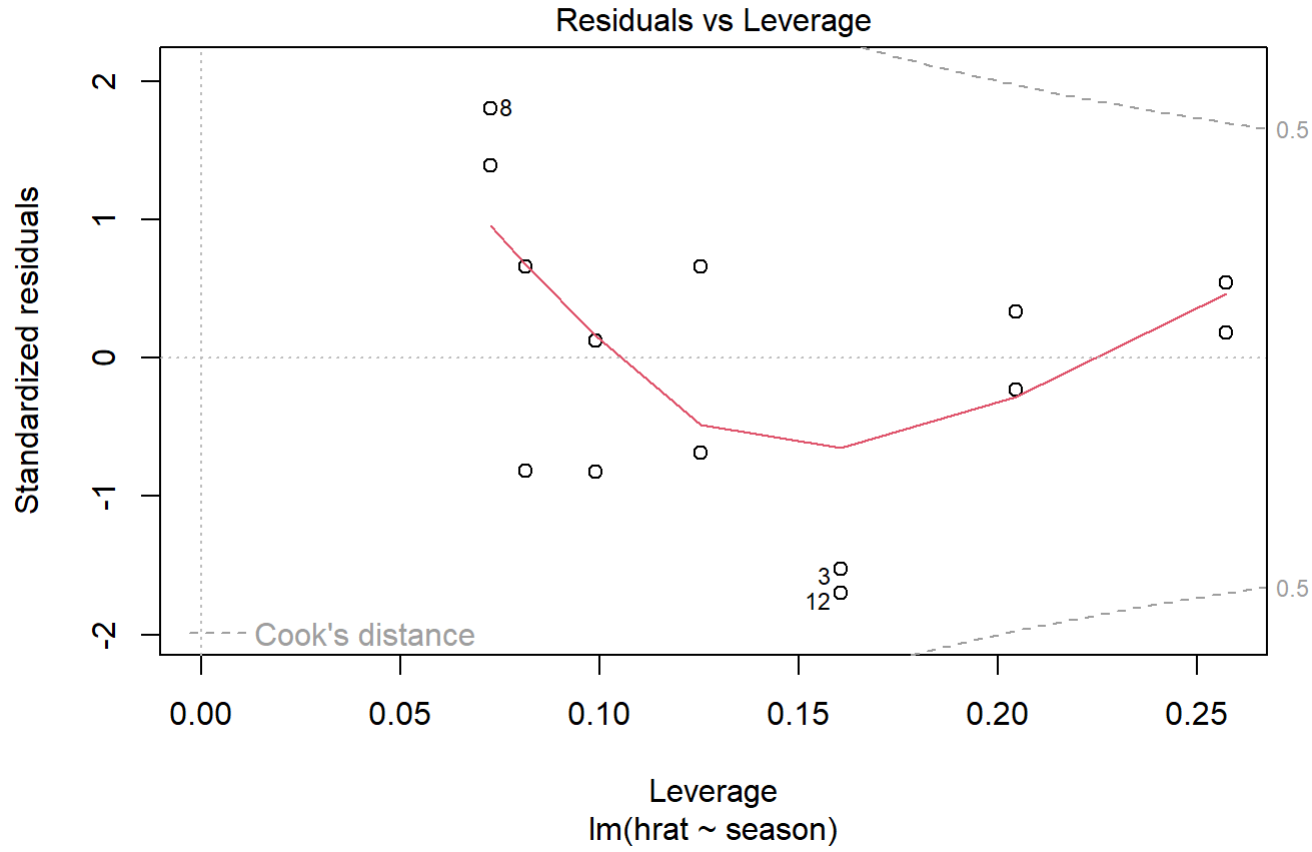



```
#As seen on the Q-Q plot, the points of the linear regression are aligned with the line that ass
umes normality. Therefore, we can assume normality for this data-set.


#Independence Assumption

plot(lm(hrat ~ season, data = data_set))
```


Residuals vs Fitted
Residuals
0.02
0.01
0.00
-0.01
-0.02
8
3
12
0.05
0.06
0.07
0.08
0.09
Fitted values
lm(hrat ~ season)


Q-Q Residuals
Standardized residuals
2
1
0
-1
8
3
12
-1
0
1
Theoretical Quantiles
lm(hrat ~ season)


Scale-Location
√|Standardized residuals|
0.0
0.2
0.4
0.6
0.8
1.0
1.2
3
8
12
0.05
0.06
0.07
0.08
0.09
Fitted values
lm(hrat ~ season)


Residuals vs Leverage
Standardized residuals
2
1
0
-1
-2
8
3
12
0.5
0.5
Cook's distance
0.00
0.05
0.10
0.15
0.20
0.25
Leverage
lm(hrat ~ season)

```
#the independence assumption test fails, as the data seems to be following a pattern, therefore, the variables depend on one another.


#t-test
#Ho: Beta1 = 0
#The null hypothesis states that the slope or B1 is equal to zero. Meaning there is no relationship between the season and the number of home-runs that Barry Bonds had.

#Ha: Beta1 != 0
#The alternative hypothesis states that beta1 is not equal to zero, indicating that there is a relationship between the season and the number of home-runs.

#t-statistic = (B1/SE)


t_stat = 0.0040442/0.0008812
t_stat
```

```
## [1] 4.589424
```

```
Pvalue = pt(4.589424, Bonds_reg$df, lower.tail = FALSE)*2
Pvalue
```

```
## [1] 0.0006222045
```

```
#Given that the p-value is 0.0006222045, which is less than alpha of 0.05, we have strong evidence against the null hypothesis that states that there is no relationship between the season and the home-runs for Barry Bonds. Meaning, we would be accepting the alternative hypothesis. Therefore, we can be 95% that there is a relationship between the season and the number of home-runs that Barry Bonds scores.


#Prediction for the season 2001
seas_2001 = data.frame(season =2001)
predict(Bonds_reg, newdata=seas_2001)
```

```
##          1
## 0.09988334
```

```
#predicted value is equal to 0.09988334

#Prediction Interval for season 2001
predict(Bonds_reg, newdata=seas_2001, interval="confidence", level=0.95)
```

```
##          fit        lwr       upr
## 1 0.09988334 0.08353537 0.1162313
```

```
cat("The 95% Confidence Interval of the Home-runs for season 2001 for Barry Bonds is", "(",0.083
53537, ",",0.1162313,")")
```

```
## The 95% Confidence Interval of the Home-runs for season 2001 for Barry Bonds is ( 0.08353537
, 0.1162313 )
```

```
#Part II

Sleep_data <- read.csv("C:\\Users\\safah\\Documents\\Winter24\\Sleep_Efficiency.csv")
names(Sleep_data) <- gsub(" ", ".", names(Sleep_data))
# Display the first few rows of the Sleep_data data frame
head(Sleep_data)
```

	ID	A..	Gen…	Bedtime	Wakeup.time	Sleep.duration	Sleep.efficie
	<int>	<int>	<chr>	<chr>	<chr>	<dbl>	<(
1	1	65	Female	2021-03-06 01:00:00	2021-03-06 07:00:00	6.0	(
2	2	69	Male	2021-12-05 02:00:00	2021-12-05 09:00:00	7.0	(
3	3	40	Female	2021-05-25 21:30:00	2021-05-25 05:30:00	8.0	(
4	4	40	Female	2021-11-03 02:30:00	2021-11-03 08:30:00	6.0	(
5	5	57	Male	2021-03-13 01:00:00	2021-03-13 09:00:00	8.0	(
6	6	36	Female	2021-07-01 21:00:00	2021-07-01 04:30:00	7.5	(

6 rows | 1-8 of 16 columns

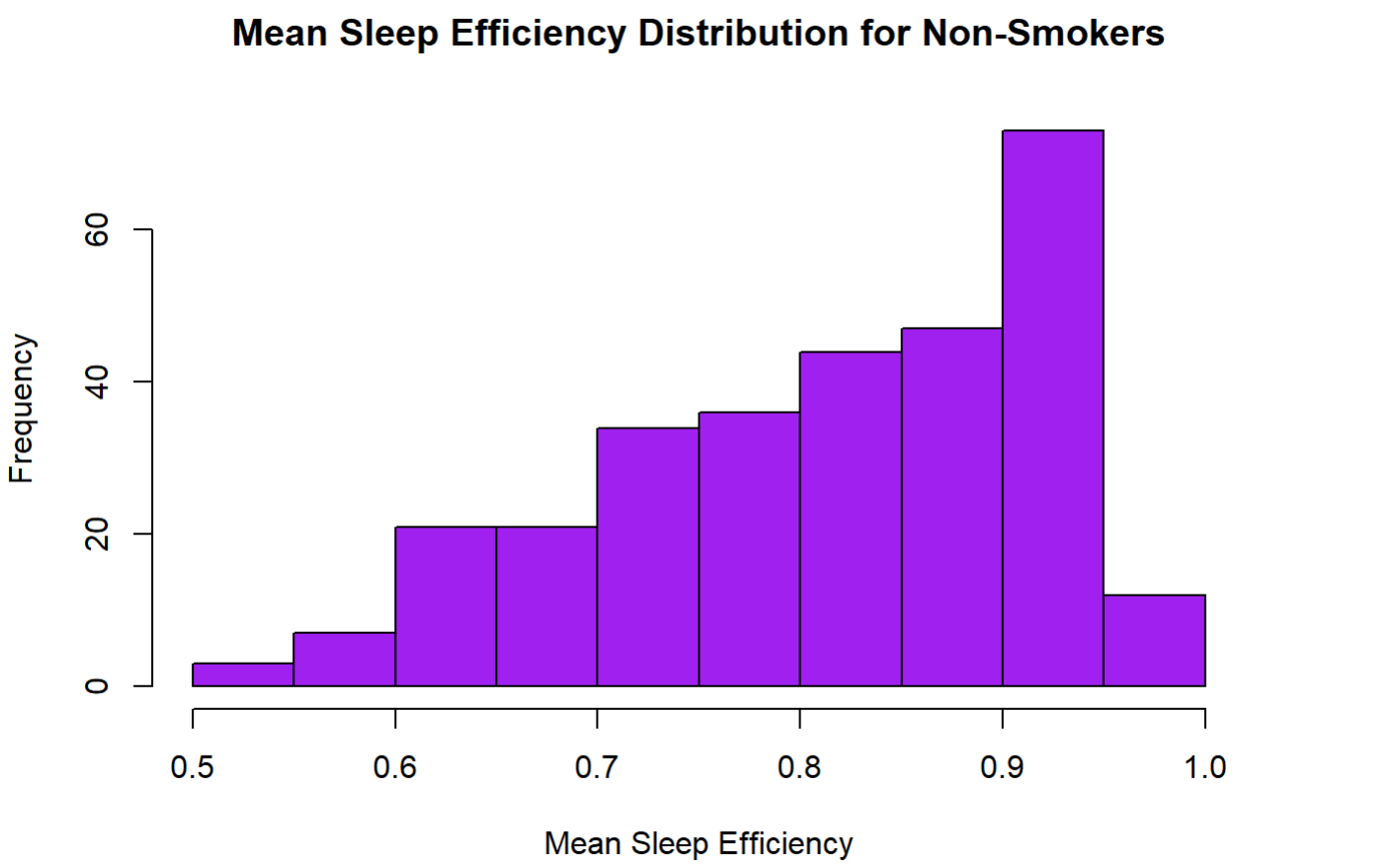
```
#Part Two:
#We aim to investigate the relationship between sleep efficiency and smoking status, with a spec
ific focus on
#comparing the sleep efficiency of non-smokers to smokers. We hypothesize that non-smokers exhib
it higher sleep efficiency compared to smokers. To test this hypothesis, we will use a two-taile
d test to determine if there is a statistically significant difference in the mean sleep efficie
ncy between smokers and non-smokers

#Hypothesis Test:

#Null Hypothesis (Ho): The mean sleep efficiency for smokers is equal to the mean sleep efficien
cy for non-smokers.
#Alternative Hypothesis (Ha): The mean sleep efficiency for smokers is not equal to the mean sle
ep efficiency for non-smokers.


# Subset data for smokers and non-smokers
smokers <- Sleep_data$Sleep.efficiency[Sleep_data$Smoking.status == "Yes"]
non.smokers <- Sleep_data$Sleep.efficiency[Sleep_data$Smoking.status == "No"]

# Calculate mean for smokers and non-smokers
mean_smokers <- mean(smokers, na.rm = TRUE)
mean_non_smokers <- mean(non.smokers, na.rm = TRUE)
diff_mean <-mean_smokers-mean_non_smokers
print(diff_mean)
```

```
## [1] -0.08266495
```

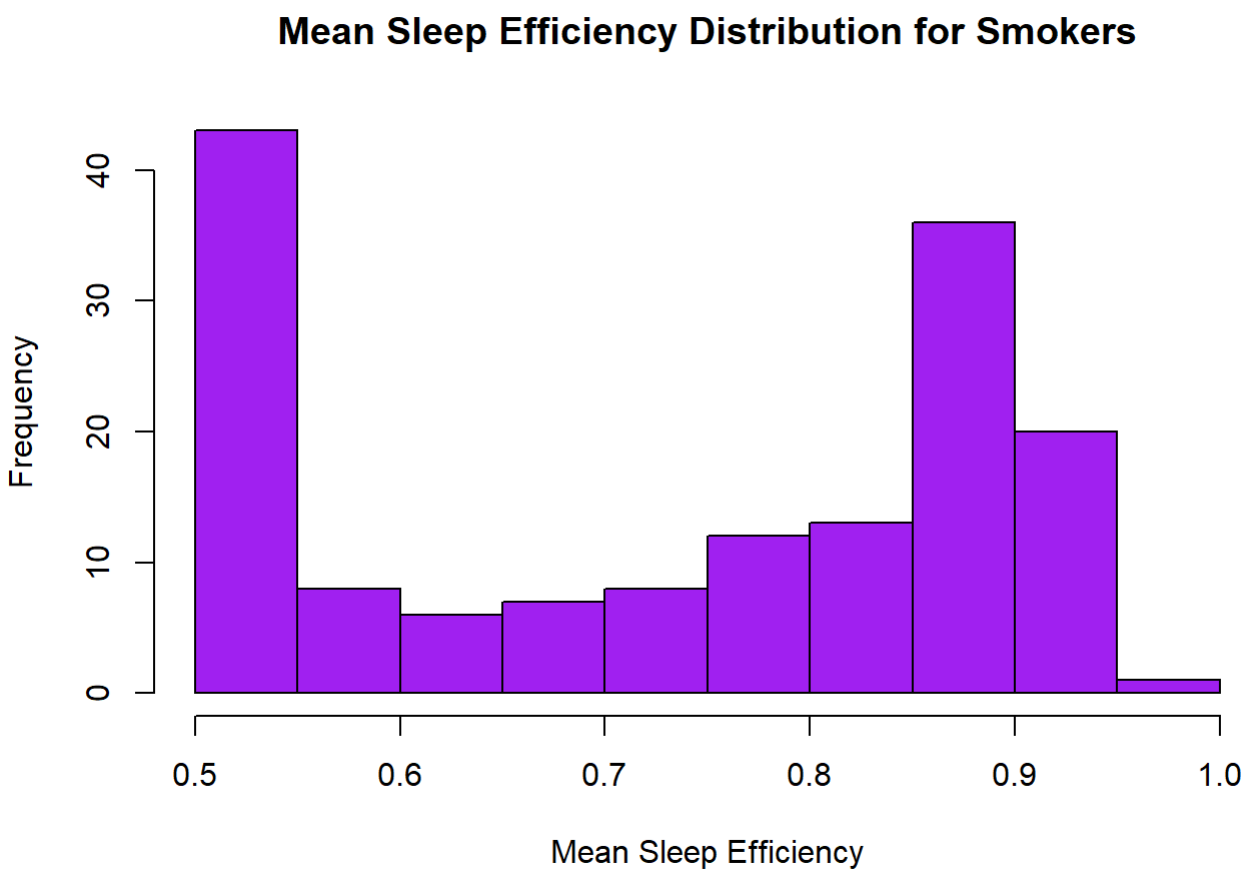
```
#T-Test
t_result <- t.test(smokers, non.smokers)
print(t_result)
```

```
##
##  Welch Two Sample t-test
##
## data:  smokers and non.smokers
## t = -5.705, df = 227.36, p-value = 3.603e-08
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.11121665 -0.05411325
## sample estimates:
## mean of x mean of y
## 0.7344156 0.8170805
```

```
#Results:

#The level of significance that we will be using is 0.05 for a one-tailed
#test and 0.025 for a two-tailed test, which is relevant to this test.
#The obtained p-value was p-value = 1.979e-07, which is much smaller than
#both the levels of significance, 0.05 and 0.025. Therefore, we have strong
#evidence to reject the null hypothesis in favor of the alternative.
#This indicates that the mean sleep efficiency differs between smokers
#and non-smokers. The 95% confidence interval is -0.11423816 to -0.05303355, which means that we
are 95% confident
#that the true difference in means falls within that range. Since the 95% confidence interval
#for the difference in means does not include 0, it further supports the conclusion that
#there is a significant difference in sleep efficiency between the two groups.
#Lastly, the negative t-value suggests that, on average, non-smokers have higher sleep efficienc
y compared to smokers.

#Checking for the validity of our test
#1. Are the means normally distributed?

hist(non.smokers, col = "purple", main = "Mean Sleep Efficiency Distribution for Non-Smokers", x
lab = "Mean Sleep Efficiency", breaks = 15)
```



```
hist(smokers, col = "purple", main = "Mean Sleep Efficiency Distribution for Smokers", xlab = "M
ean Sleep Efficiency", breaks = 15)
```




```
# Perform Shapiro-Wilk test for normality
shapiro_test_smokers <- shapiro.test(smokers)
print(shapiro_test_smokers)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  smokers
## W = 0.86735, p-value = 1.879e-10
```

```
shapiro_test_non_smokers <- shapiro.test(non.smokers)
print(shapiro_test_non_smokers)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  non.smokers
## W = 0.94235, p-value = 2.155e-09
```

```
#The p-value represents the probability of observing the data if the
#null hypothesis (that the data comes from a normally distributed population)
#were true. In both cases, p-values are extremely small, therefore, we reject
#the null hypothesis in both cases. Based on the Shapiro-Wilk tests, we
#have evidence to suggest that both the smokers' group and the non-smokers'
#group do not follow a normal distribution. Despite this violation of normality, we will still u
se the
#t-test as our sample size is quite large. The t-test is known to be robust to violations of nor
mality, particularly with larger sample sizes.
```

```
#Part Three

#H0 (Null Hypothesis): There is no significant impact of caffeine consumption on the percentage
of deep sleep among individuals.
#Ha (Alternative Hypothesis): There is a significant impact of caffeine consumption on the perce
ntage of deep sleep among individuals.


model <- lm(Sleep.efficiency ~ Caffeine.consumption, data = Sleep_data)
print(model)
```

```
##
## Call:
## lm(formula = Sleep.efficiency ~ Caffeine.consumption, data = Sleep_data)
##
## Coefficients:
##          (Intercept)  Caffeine.consumption
##            0.7826999             0.0002908
```

```
summary(model)
```

```
##
## Call:
## lm(formula = Sleep.efficiency ~ Caffeine.consumption, data = Sleep_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.30451 -0.08724  0.03730  0.11413  0.20730
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.7826999  0.0082913  94.401   <2e-16 ***
## Caffeine.consumption 0.0002908  0.0002163   1.345    0.179
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1348 on 425 degrees of freedom
##   (25 observations deleted due to missingness)
## Multiple R-squared:  0.004236,   Adjusted R-squared:  0.001893
## F-statistic: 1.808 on 1 and 425 DF,  p-value: 0.1795
```

```
t = 0.0002908/0.0002163
t
```

```
## [1] 1.344429
```

```
pvalue = pt(summary(model)[["coefficients"]][2, "t value"], model$df, lower.tail = FALSE)
pvalue
```

```
## [1] 0.08974389
```

```
confint(model, level=0.95)[2,]
```

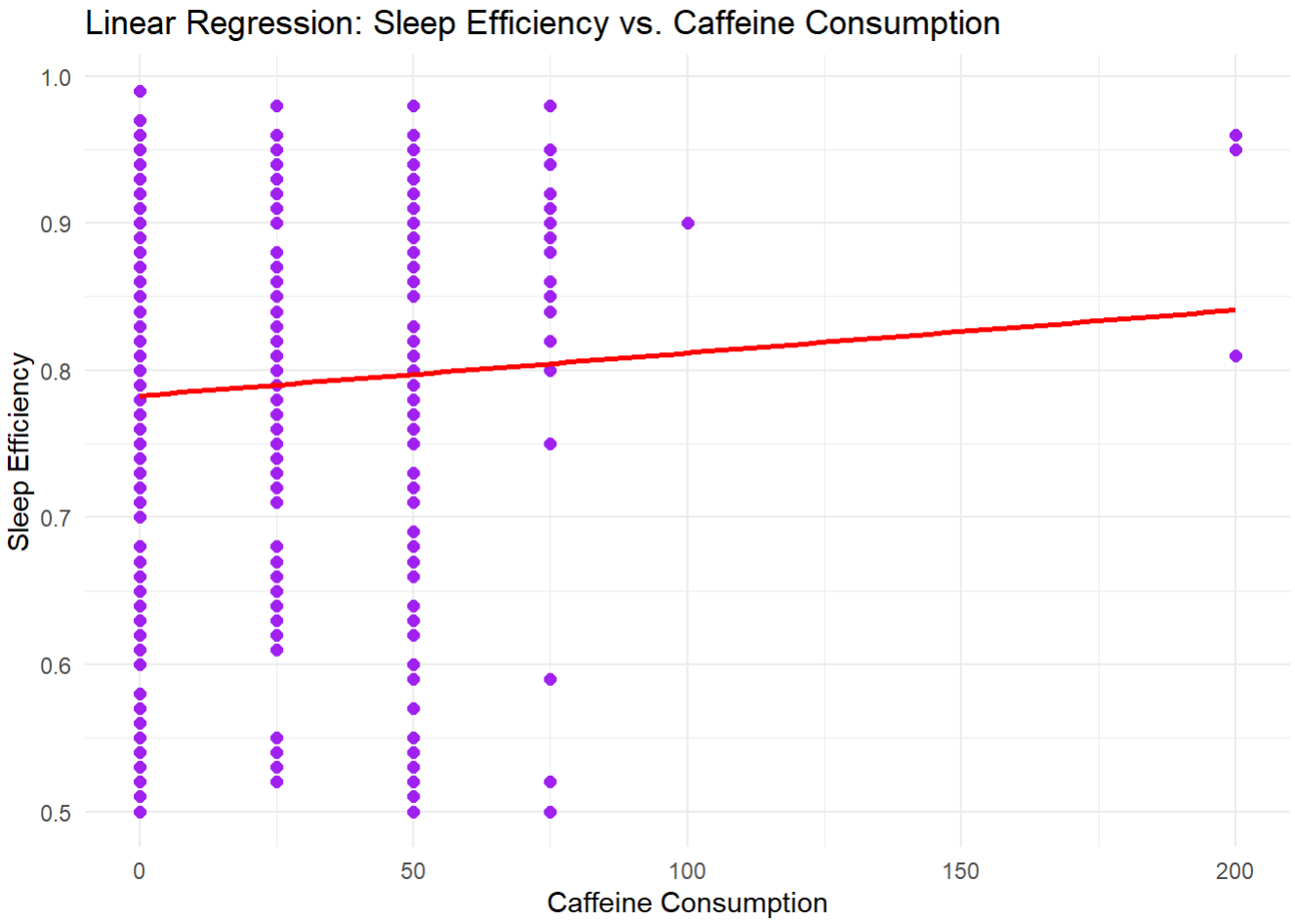
```
##         2.5 %        97.5 %
## -0.0001343157  0.0007159287
```

```
ggplot(Sleep_data, aes(x = Caffeine.consumption, y = Sleep.efficiency)) +
  geom_point(col = "purple", size = 2) +  # Scatter plot of data points
  geom_smooth(method = "lm", se = FALSE, col = "red") +  # Add a regression line
  xlab("Caffeine Consumption") +  # X-axis label
  ylab("Sleep Efficiency") +       # Y-axis label
  ggtitle("Linear Regression: Sleep Efficiency vs. Caffeine Consumption") +  # Plot title
  theme_minimal()
```

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

```
## Warning: Removed 25 rows containing non-finite values (`stat_smooth()`).
```

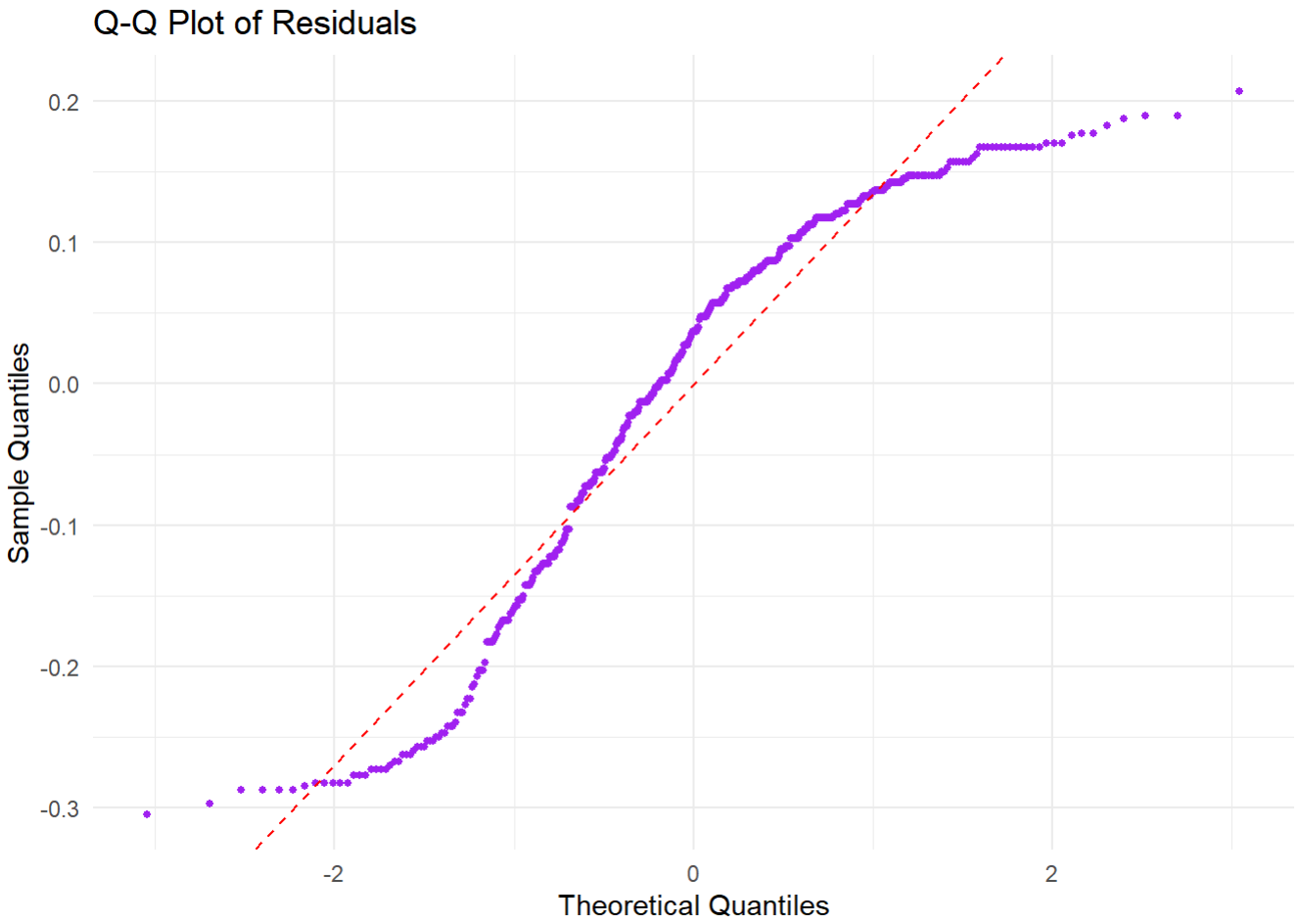
```
## Warning: Removed 25 rows containing missing values (`geom_point()`).
```



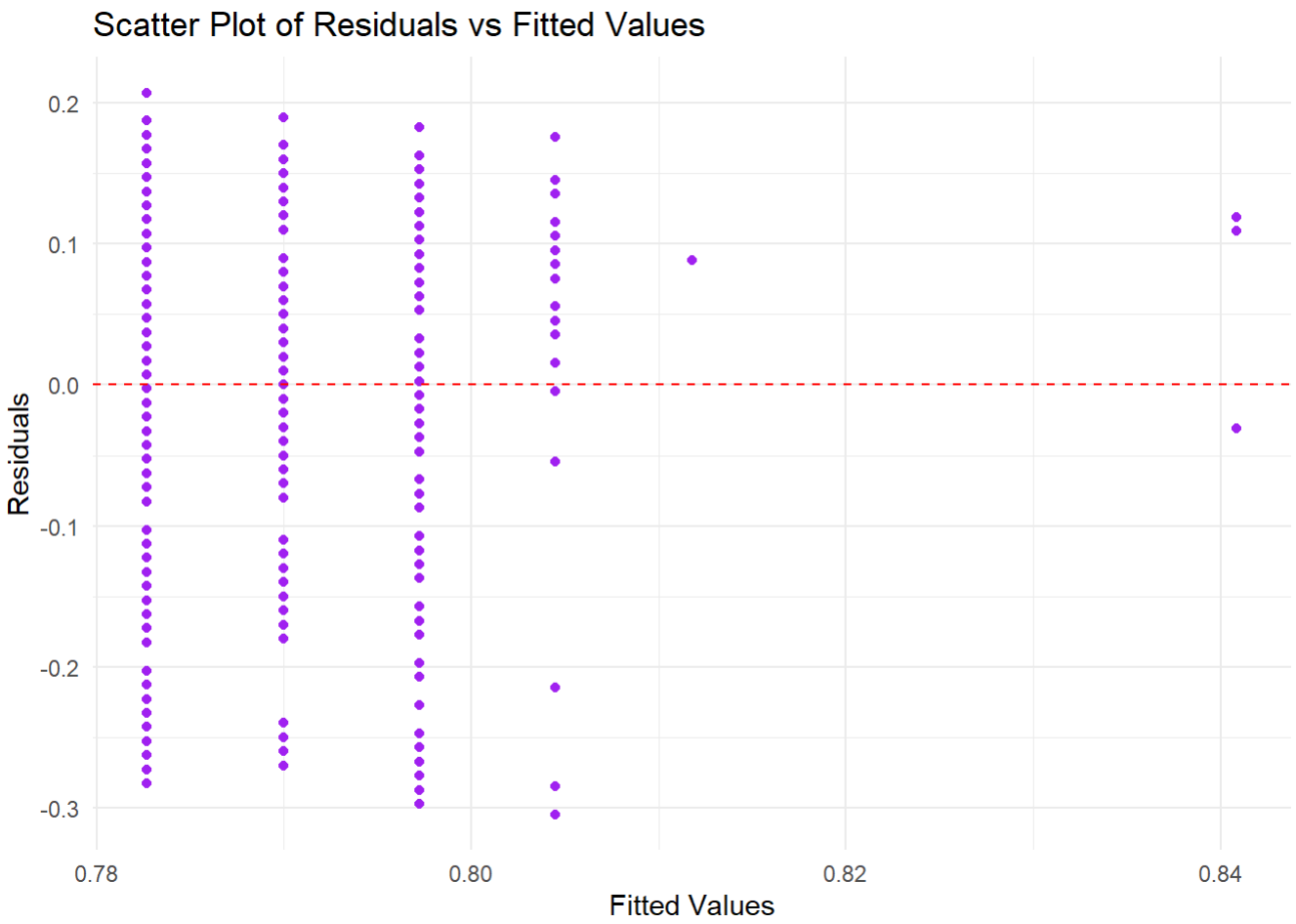
```
#By plotting the linear regression of sleep efficiency versus caffeine consumption, we observe t
hat there is little evidence of a meaningful relationship between the two variables.
#Upon running the linear regression model, we find that the intercept coefficient is approximate
ly 0.7817, indicating the estimated sleep efficiency when all independent variables, including c
affeine consumption, are set to zero. Additionally, the coefficient for caffeine consumption is
approximately 0.0003314. This suggests that for each additional unit increase in caffeine consum
ption, the model predicts a negligible increase in sleep efficiency, approximately 0.0003314 uni
ts, holding all other factors constant.
#In summary, while there is a slight positive association between caffeine consumption and sleep
efficiency according to the model, the effect size is minimal and may not be practically signifi
cant. Other factors not included in the model may play a more substantial role in determining sl
eep efficiency.

#Checking if a linear regression is the appropriate test.
#We want to assess if the linear regression model used in our analysis is appropriate. To do thi
s, we need to check for the normality of residuals and test for homoscedasticity, both of which
are assumptions in using a linear regression model.
#From the plots, it's evident that the residuals are not normally distributed. Additionally, whe
n running a linear regression model, homoscedasticity is assumed, meaning that the spread or dis
persion of the data points around the regression line remains constant throughout the range of v
alues of the independent variables. However, as observed in the scatter plot, this assumption is
clearly violated.
#To further support why a regression model may not be appropriate for our analysis, we conducted
a Shapiro-Wilk normality test on the residuals. The resulting p-value, which is 2.231e-13, indic
ates strong evidence against the null hypothesis that the residuals are normally distributed, as
it is smaller than a significance level of 0.05.
#In conclusion, the residuals deviate significantly from a normal distribution, suggesting that
the linear regression model may not be suitable for our analysis.


residuals <- residuals(model)
shapiro.test(residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals
## W = 0.92016, p-value = 2.852e-14
```

```
#Need to check for Normality of Residuals
qq_plot <- ggplot(data = data.frame(residuals = residuals), aes(sample = residuals)) +
  geom_qq(color = "purple", size = 1) +
  geom_abline(intercept = mean(residuals), slope = sd(residuals), color = "red", linetype = "das
hed") +
  labs(title = "Q-Q Plot of Residuals", x = "Theoretical Quantiles", y = "Sample Quantiles") +
  theme_minimal()
print(qq_plot)
```




```
#Now test for homoscedasticity with a scatter plot:
model <- lm(Sleep.efficiency ~ Caffeine.consumption, data = Sleep_data)
residuals <- residuals(model)
fitted_values <- fitted(model)
residuals_df <- data.frame(Fitted_Values = fitted_values, Residuals = residuals)

# Plot residuals against fitted values
ggplot(residuals_df, aes(x = Fitted_Values, y = Residuals)) +
  geom_point(color = "purple") +
  geom_hline(yintercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Scatter Plot of Residuals vs Fitted Values",
       x = "Fitted Values",
       y = "Residuals") +
  theme_minimal()
```



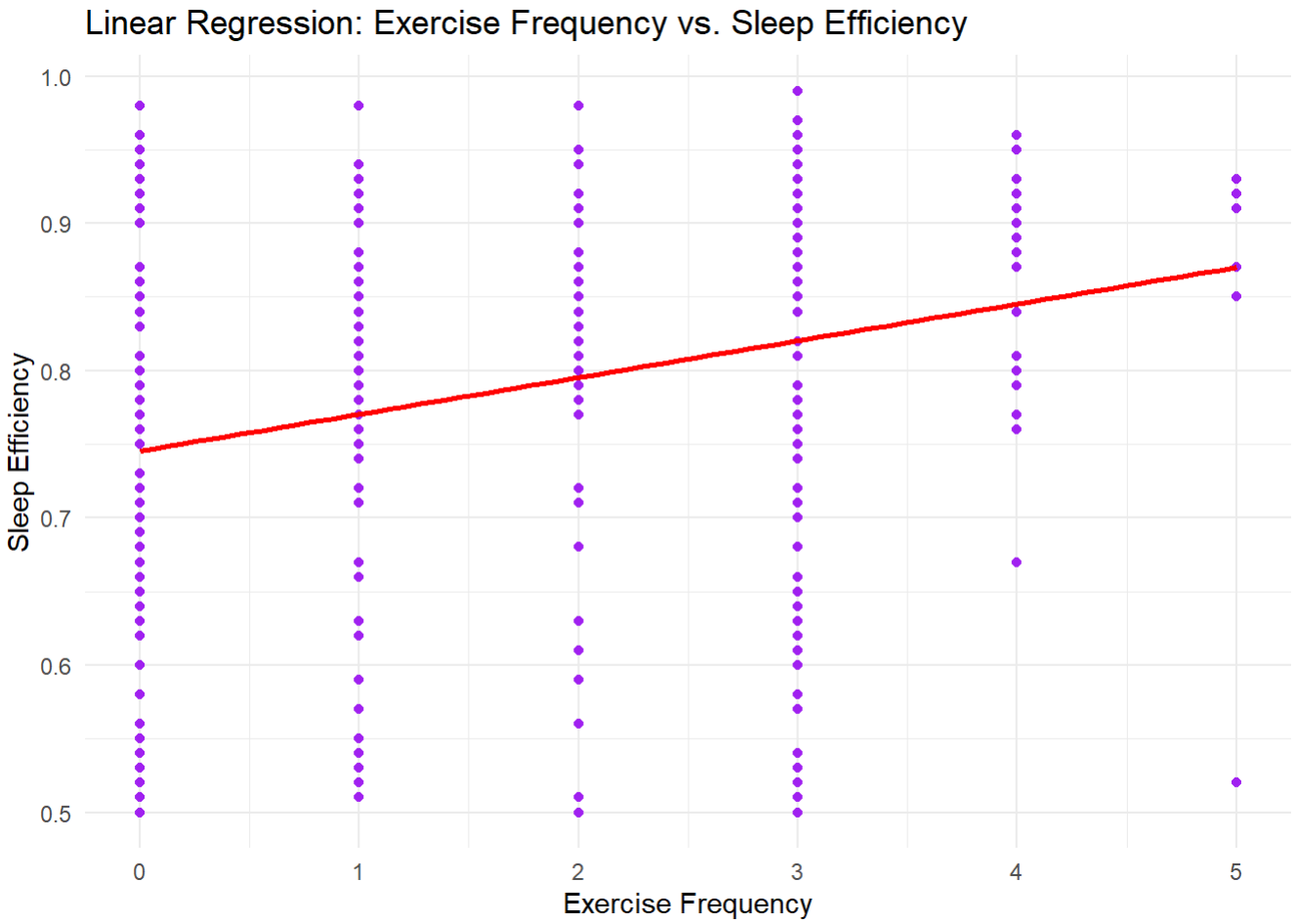

```
#Conclusively, we decided to run a GLS model, which offers a more flexible modeling framework to address violations of homoscedasticity and normality of residuals often encountered in linear regression. GLS adjusts for varying levels of variance across observations by modeling the variance-covariance structure of the errors, thus yielding more accurate parameter estimates. Unlike linear regression, GLS does not assume that the residuals follow a normal distribution.
#In this GLS model, the estimated intercept is approximately 0.7818, and the estimated coefficient for Caffeine.consumption is approximately 0.0003274. However, similar to the linear regression model, it appears that in the context of this data and model, caffeine does not have a significant impact on sleep efficiency.



Sleep_data <- na.omit(Sleep_data)
modelgsl <- gls(Sleep.efficiency ~ Caffeine.consumption, data = Sleep_data, correlation = corAR1
())
summary(modelgsl)
```

```
## Generalized least squares fit by REML
##   Model: Sleep.efficiency ~ Caffeine.consumption
##   Data: Sleep_data
##        AIC       BIC   logLik
##   -424.1852 -408.3619 216.0926
##
## Correlation Structure: AR(1)
##  Formula: ~1
##  Parameter estimate(s):
##         Phi
## -0.09421422
##
## Coefficients:
##                           Value   Std.Error  t-value p-value
## (Intercept)           0.7818429 0.008229946 94.99975  0.0000
## Caffeine.consumption 0.0003274 0.000235550  1.38988  0.1654
##
##  Correlation:
##                       (Intr)
## Caffeine.consumption -0.649
##
## Standardized residuals:
##        Min         Q1        Med         Q3        Max
## -2.2610954 -0.6509733  0.2647822  0.8719559  1.5361227
##
## Residual standard error: 0.1355081
## Degrees of freedom: 388 total; 386 residual
```

#The p-value associated with the coefficient for caffeine consumption is 0.1654, which exceeds t he conventional significance level of 0.05. Thus, we fail to reject the null hypothesis and conc lude that there is no statistically significant relationship between sleep efficiency and caffei ne consumption at the 0.05 level of significance. Regarding the intercept: The p-value associate d with the intercept is essentially zero (0.0000). Therefore, we reject the null hypothesis and conclude that there is a significant relationship between sleep efficiency and caffeine consumpt ion when caffeine consumption is zero

```
#Part Four
#hypothesis, is there a significant relationship between exercise frequency and sleep efficiency
ggplot(data=Sleep_data, aes(x=Exercise.frequency, y =Sleep.efficiency))+geom_point(color="purpl
e")+geom_smooth(method = "lm", se = FALSE, col = "red")+xlab("Exercise Frequency") +ylab("Sleep
Efficiency") +ggtitle("Linear Regression: Exercise Frequency vs. Sleep Efficiency") + theme_mini
mal()
```

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



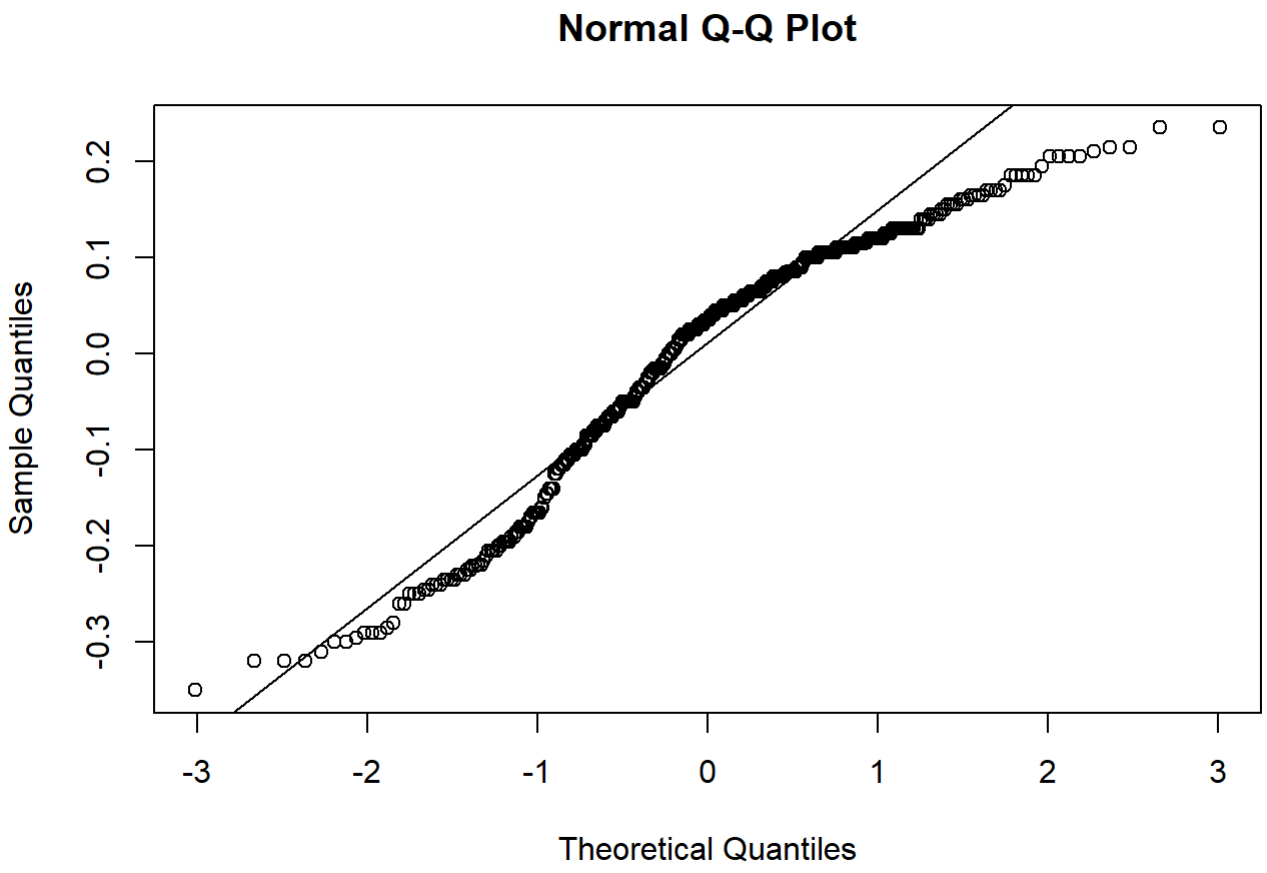

```
reg <- lm(Sleep.efficiency ~ Exercise.frequency, data=Sleep_data)

summary(reg)
```

```
##
## Call:
## lm(formula = Sleep.efficiency ~ Exercise.frequency, data = Sleep_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.35011 -0.08147  0.03464  0.10458  0.23458
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)         0.745420   0.010467  71.215  < 2e-16 ***
## Exercise.frequency 0.024937   0.004599   5.422 1.04e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.131 on 386 degrees of freedom
## Multiple R-squared:  0.07078,    Adjusted R-squared:  0.06838
## F-statistic:  29.4 on 1 and 386 DF,  p-value: 1.039e-07
```

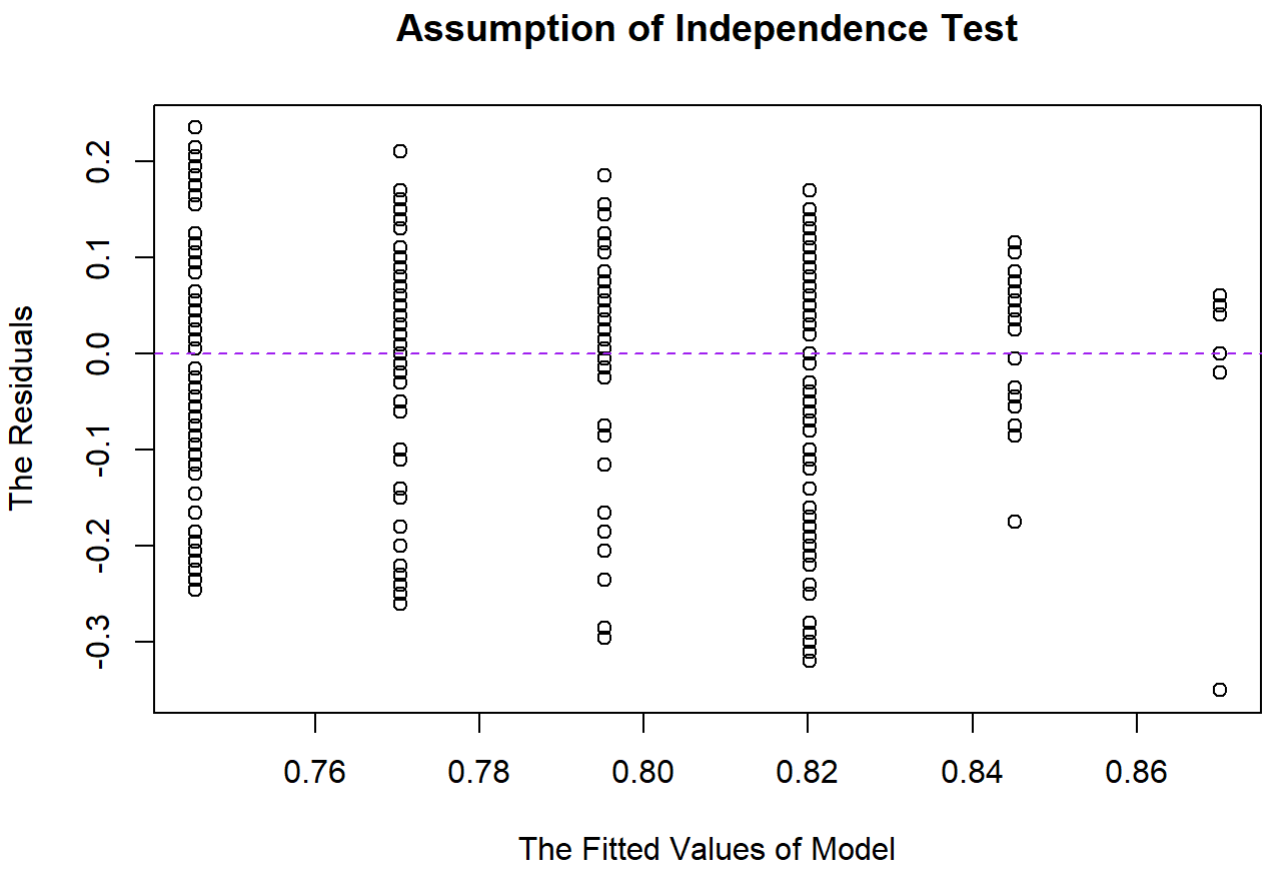
```
Beta_1 <- summary(reg)$coefficients['Exercise.frequency', 'Estimate']

Beta_1
```

```
## [1] 0.02493722
```

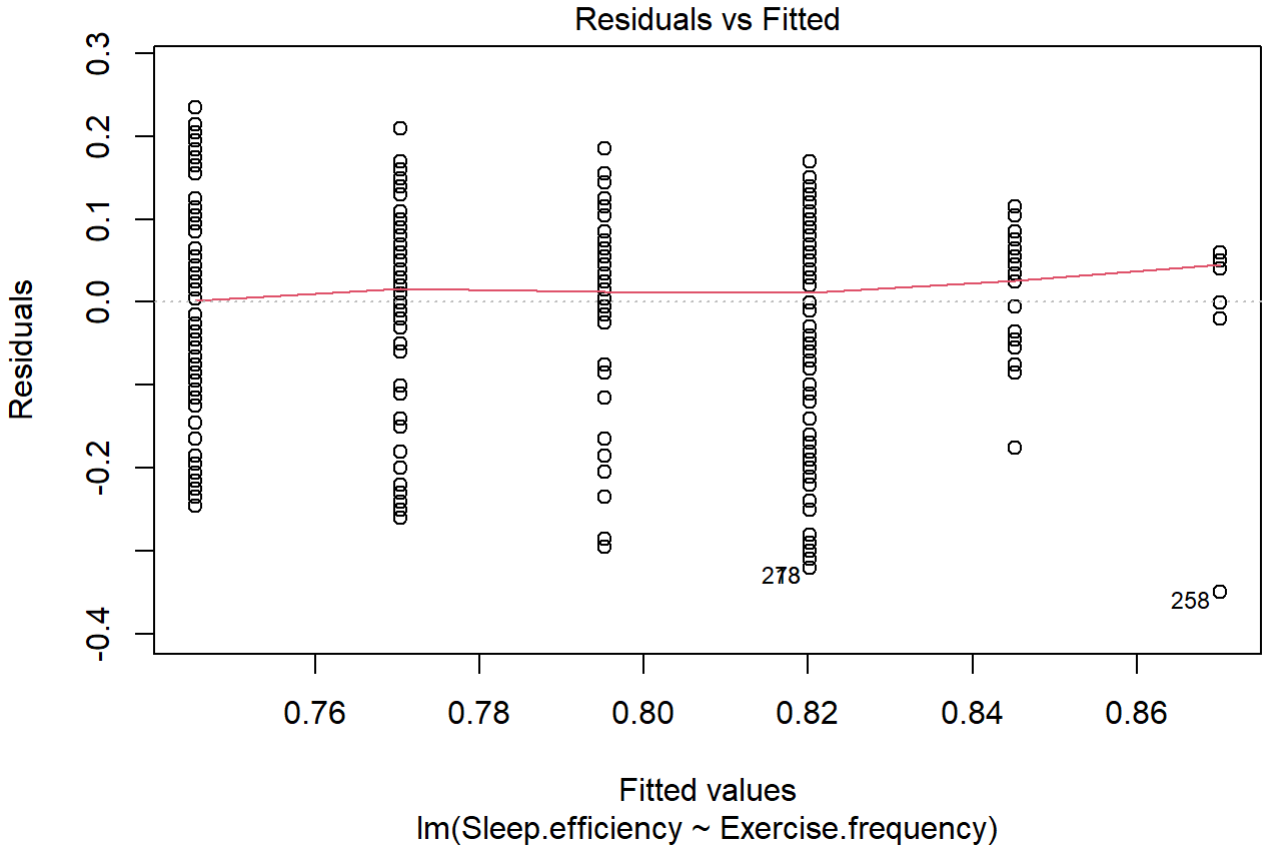
```
#y = 0.745420+0.024937x
#Assumption of normality plot:
resid <- residuals(reg)
qqnorm(resid)
qqline(resid)
```

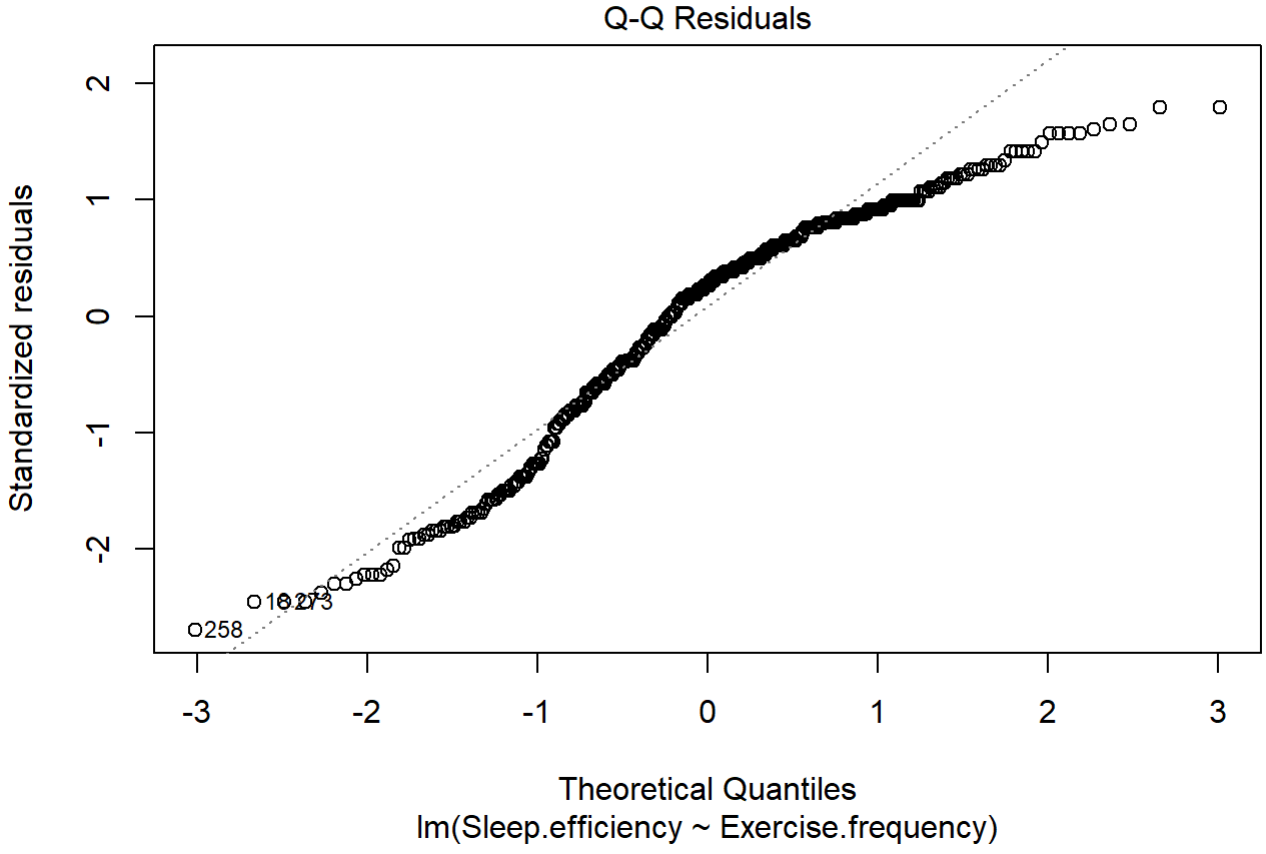


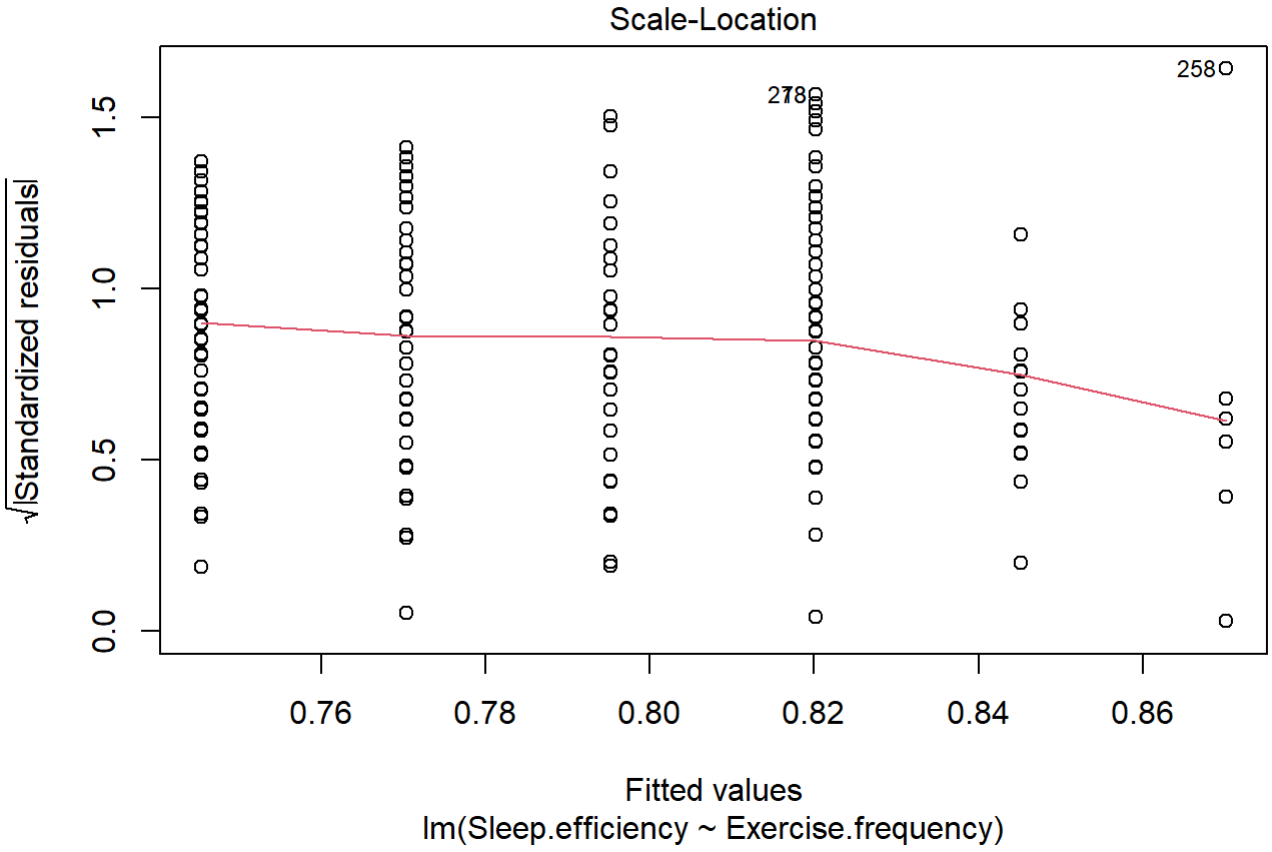

```
#Assumption of independence plot:
plot(fitted(reg), resid, xlab="The Fitted Values of Model", ylab="The Residuals", main="Assumpti
on of Independence Test")
abline(h=0, col="purple", lty=2)
```

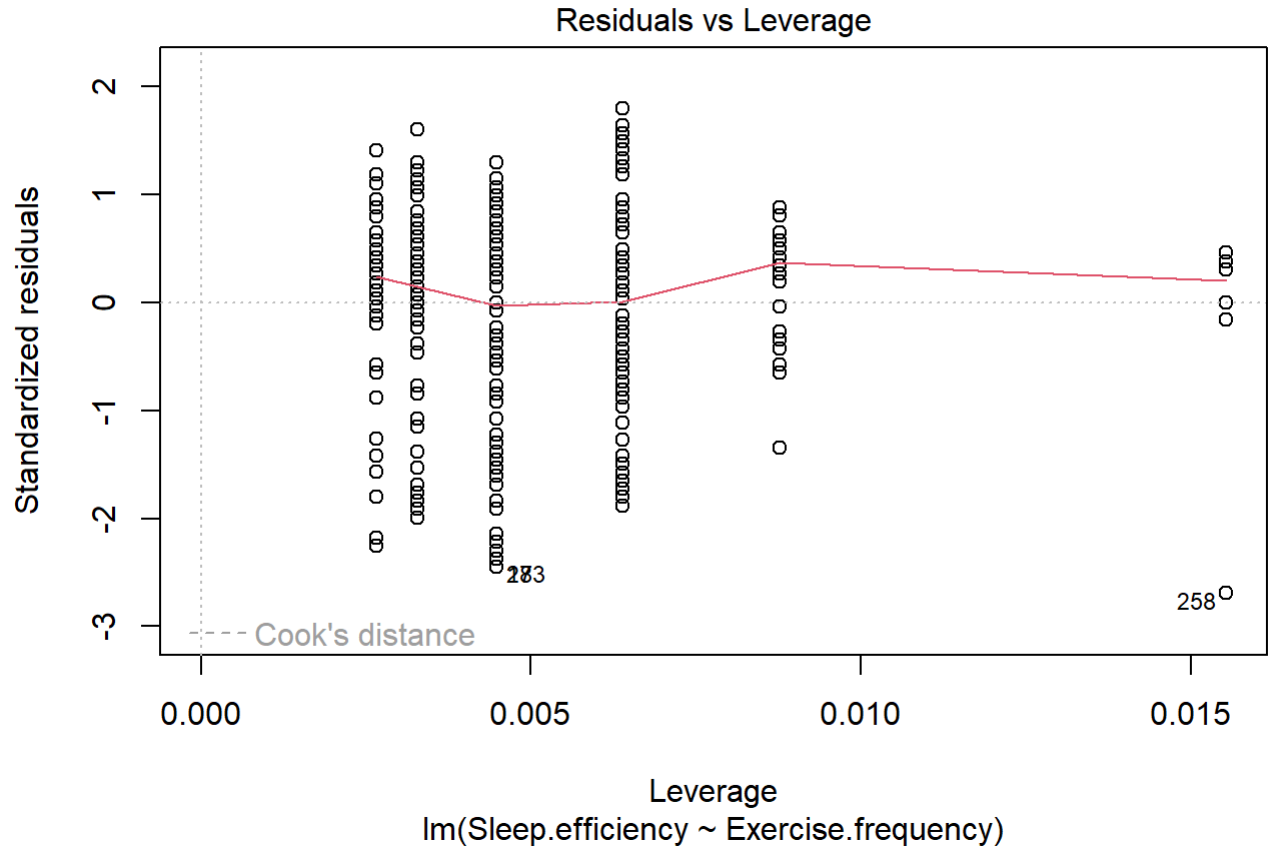



```
plot(lm(Sleep.efficiency ~ Exercise.frequency, data=Sleep_data))
```


Residuals vs Fitted
Residuals
0.3
0.2
0.1
0.0
-0.2
-0.4
278
258
0.76
0.78
0.80
0.82
0.84
0.86
Fitted values
lm(Sleep.efficiency ~ Exercise.frequency)


Q-Q Residuals
Standardized residuals
2
1
0
-1
-2
258
-3
-2
-1
0
1
2
3
Theoretical Quantiles
lm(Sleep.efficiency ~ Exercise.frequency)


Scale-Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
0.76
0.78
0.80
0.82
0.84
0.86
Fitted values
lm(Sleep.efficiency ~ Exercise.frequency)


Residuals vs Leverage
Standardized residuals
2
1
0
-1
-2
-3
Cook's distance
0.000
0.005
0.010
0.015
Leverage
lm(Sleep.efficiency ~ Exercise.frequency)

```
set.seed(42)
iteration <- 1000
vector <- numeric(iteration)


for (i in 1:iteration) {
  sample <- sample(reg$residuals, replace=TRUE)
  effien <- reg$fitted.values + sample
  new_mod <- lm(effien ~ Exercise.frequency, data= Sleep_data)
  vector[i]<- coef(new_mod)[2]
}

quantile(vector, c(0.025,0.975))
```

```
##       2.5%      97.5%
## 0.01567148 0.03380947
```

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.