Data Analysis and Statistical Methods
Frans Simanjuntak - S3038971
Problem set 5

1. Question 1

a) Fit a linear model for the perimeter as a function of the area.

$$perimeter = \beta_0 + \beta_1 * area$$

Calculate the coefficients (i.e slope and intercept), using the equations we learned in the lecture.

R-code

```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#load csv data
df <- read.csv(file = "assignment5/rock.txt",header = TRUE)

#slope
b1=cov(df$area,df$peri)/var(df$area)
#interceps
b0=mean(df$peri)-b1*mean(df$area)
```

```
> b1
[1] 0.4387544
> b0
[1] -471.4358
```

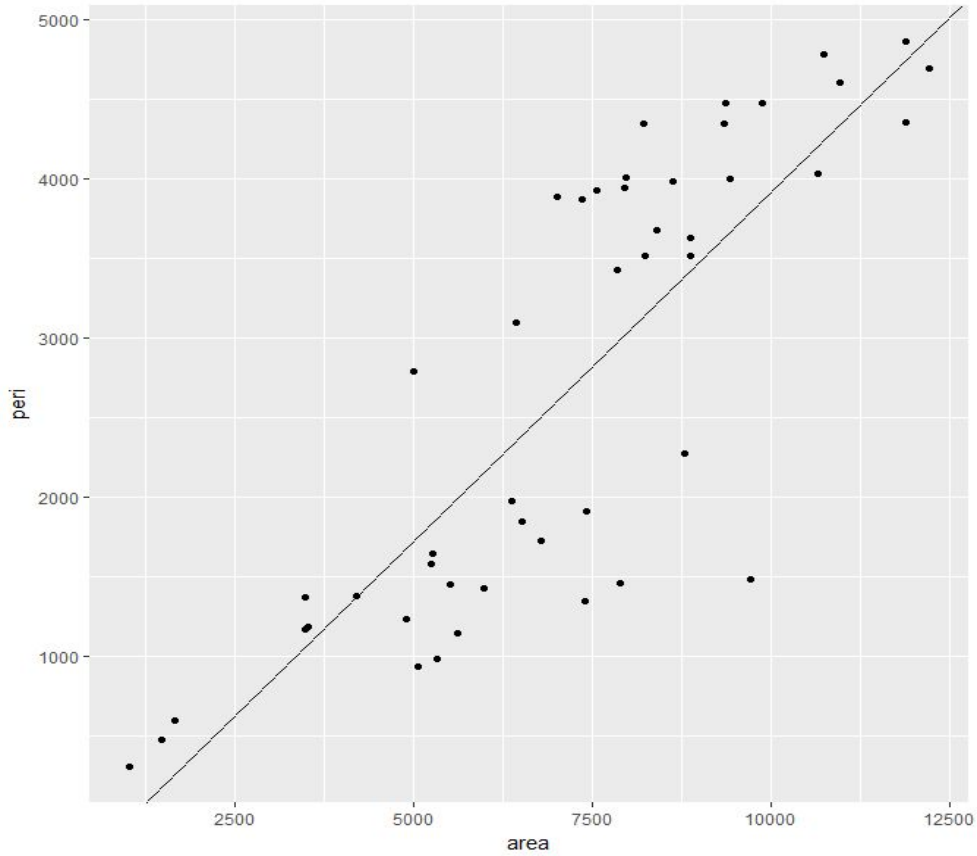
The intercept is -471.4358
The slope is 0.4387544

b) Plot the data as a scatter plot and add the regression line.

R-Code

```
#plot the data as a scatter plot and add the regression line
ggplot(df, aes(x=area, y=peri)) +
  geom_point() +
  geom_abline(intercept = b0, slope = b1)
```

The output



c) ***Do the linear fit using lm() in R and give the summary of the linear model.***

The R-Code

```
#perform a linear regression and show the summary
z = lm(peri~area, data=df)
summary(z)
```

The Summary

```
Call:
lm(formula = peri ~ area, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-2306.8  -502.3   122.5   564.5  1291.9

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -471.43579  342.77487  -1.375    0.176
area           0.43875    0.04473   9.808 7.51e-13 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 823.1 on 46 degrees of freedom
Multiple R-squared:  0.6765,    Adjusted R-squared:  0.6695
F-statistic:  96.2 on 1 and 46 DF,  p-value: 7.506e-13
```

Explain every element of the output that was covered in the lecture

- **Formula Call**
  It is the formula R used to fit the data.
  In this case the formula is *lm(formula = peri ~ area, data = df)*.
- **Residuals**
  Residuals are essentially the difference between the actual observed response values and the response values that the model predicted. The Residuals section of the model output breaks it down into 5 summary points
- **Coefficients**
  The coefficients are two constants that represent the *intercept* and *slope* terms in the linear model. It consist of coefficient estimate, coefficient standard error, coefficient t-value, and coefficient *Coefficient - Pr(>t)*.
  - *Coefficient estimate* contains two rows, the intercept and the slope. The coefficient of intercept is -471.43579 and the slope is 0.43875.
  - *Coefficient standard error* measures the average amount that the coefficient estimates vary from the actual average value of our response variable .The intercept standard error of intercept is 342.77487 and the standard error of the slope is 0.043873.
  - *Coefficient t-value* is a measure of how many standard deviations our coefficient estimate is far away from 0. The t-value of intercept is -1.375 and the t-value of the slope is 9.808.
  - *Coefficient - Pr(>t)* relates to the probability of observing any value equal or larger than t. The p-value of intercept is 0.176 and the slope is 7.51e-13.
- **Significant Codes**
  Give the information of how significant the intercept and slopes in statistic.
- **Residual Standard Error**
  The residual standard error is measure of the *quality* of a linear regression fit. In this case, the value is 823.1
- **Multiple R-squared, Adjusted R-squared**
  The multiple R-squared statistic provides a measure of how well the model is fitting the actual data. In our case the value is 0.6765 (67%)
- **F-Statistic**
  The F-Statistic is an indicator of whether there is a relationship between our predictor and the response variables. The value includes the F-value (96.2), degree of freedom (46) and p-value (7.506e-13).

State the hypotheses

H_0 : There is no relationship between variable peri and area

H_a : There may be a relationship between variable peri and area

Draw a conclusion

Since the p-value (7.506e-13) is lower than 0.05 therefore we must reject the null hypothesis. We can draw a conclusion, there may be a relationship between variable peri and area.

d) If you do it right, the intercept is negative, whereas a negative perimeter is not physically possible. Is this a problem? Why/why not?

It's not a problem since the mean of our dependent variable (area) is positive therefore the dependent values would be positive as well.

But again, this is not always acceptable. Depending on our dependent variable, a negative value for our intercept should not be a cause for concern. This simply means that the expected value on our dependent variable will be less than 0 when all independent variables are set to 0. For some dependent variables, this would be expected. For example, if the mean value of our dependent variable is negative, it would be no surprise whatsoever that the constant is negative; in fact, if we got a positive value for the constant in this situation, it might be cause for concern.

e) Use the standard errors of the coefficients in the R output to estimate 95% confidence intervals of the parameters a and b. Do this by hand using the tables in the book .

Given:

n = 48

$\alpha = 0.05$

Hypothesis:

$H_0: \beta_0 = 0$ $\beta_1 = 0$ (coefficient is not significant)

$H_a: \beta_0 \neq 0$ $\beta_1 \neq 0$ (coefficient is significant)

$\widehat{\beta}_0 = -471.43579$, $\widehat{\beta}_1 = 0.43875$, $SE(\widehat{\beta}_0) = 342.77487$, $SE(\widehat{\beta}_1) = 0.04473$

$T_{\alpha/2,\, n-2} = T_{0.025,\, 46} = -2.009$ *df=46 is close to df=50

$T_{1-\alpha/2,\, n-2} = T_{0.925,\, 46} = 2.009$

Rules:

Do not Reject T_0 if: $\widehat{\beta 0} + T_{\alpha/2,\, n-2} * SE(\widehat{\beta}_0) < T_0 < \widehat{\beta 0} + T_{1-\alpha/2,\, n-2} * SE(\widehat{\beta}_0)$

$(-1161.405955 < T_0 < 218.5343747)$

Do not Reject T_1 if: $\widehat{\beta 1} + T_{\alpha/2,\, n-2} * SE(\widehat{\beta}_1) < T_1 < \widehat{\beta 1} + T_{1-\alpha/2,\, n-2} * SE(\widehat{\beta}_1)$

$(0.3487131619 < T_1 < 0.5287868381)$

$T_0 = \frac{\widehat{\beta 0}}{SE(\widehat{\beta 0})} = \frac{-471.43579}{342.77487} = -1.375351087$

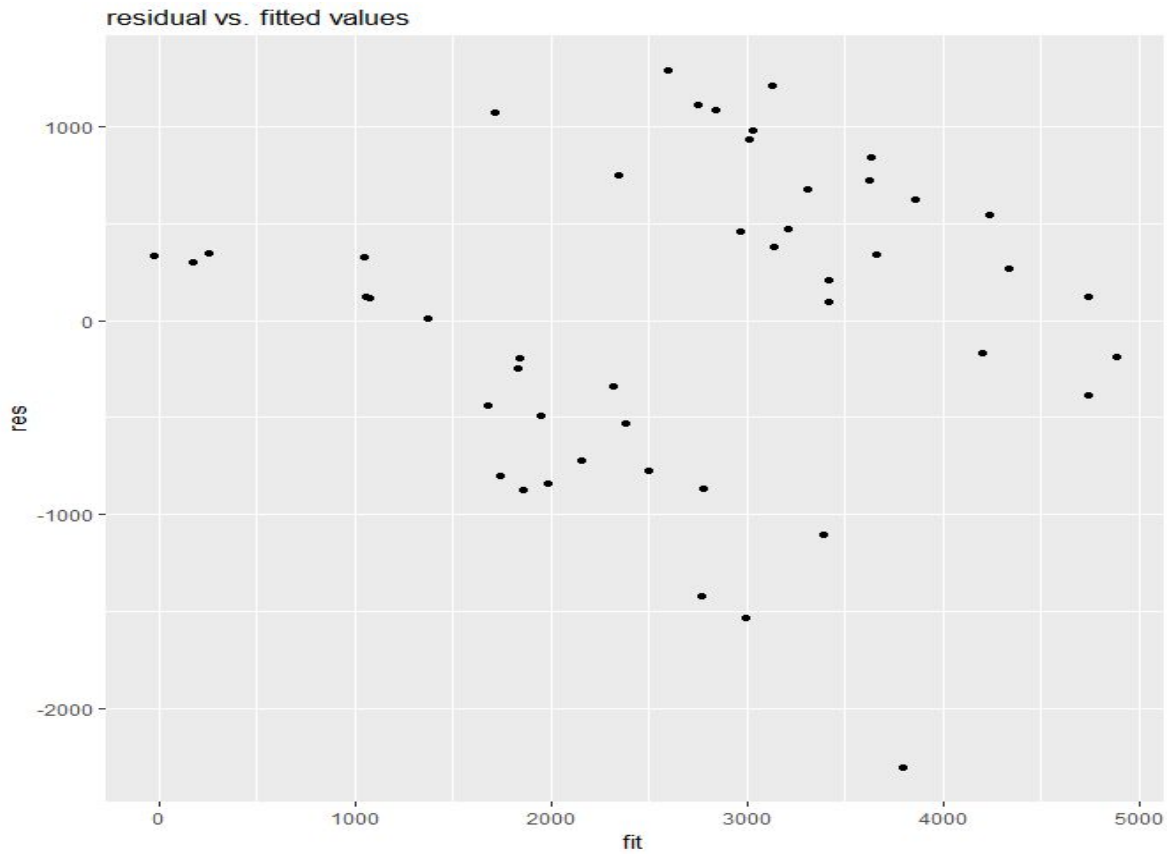
$T_1 = \frac{\widehat{\beta 1}}{SE(\widehat{\beta 1})} = \frac{0.43875}{0.04473} = 9.808853119$

Since $-1161.405955 < T_0 < 218.5343747$ therefore we fail to reject the null hypothesis however we must reject the null hypothesis for T_1 since the value is greater than 0.5287868381 .
Thus, $\widehat{\beta}_0$ is not significant but β_1 is significant.

f) Use the command residuals() to calculate the residuals and fitted() to calculate the fitted values. Plot the residuals vs. the fitted values. Use this plot to qualitatively evaluate some basic assumptions of the classic linear regression, namely linearity of the data set and equal variance.

R-code

```
#get the fitted values corresponding to each value of area then get the residuals
#Add in columns to the df data frame for the fit and res values
df$fit = fitted(z)
df$res = residuals(z)

#plot them
ggplot(df, aes(x=fit, y=res)) + geom_point() + ggtitle("residual vs. fitted values")
```



The linearity of the data says that if our original data was from a straight line, then we would just see residual values of just zero. Also the error terms in the regression model are normally distributed with a mean of zero and constant variance. If this assumption does not hold, the various inferences that were made with the hypothesis test, confidence intervals, and prediction intervals are suspect.

Looking at the linear plot shown in question 1b, we can see that the data tends to form a straight line which means there is an increasing linear relationship between the area and perri. When conducting a residual analysis, a "residuals versus fits plot" is the most frequently created plot. It is a scatter plot of residuals on the *y* axis and fitted values (estimated responses) on the *x* axis. The plot is used to detect non-linearity, unequal error variances, and outliers.

The above plot depicts that:

- ➢ The residuals "bounce randomly" around the 0 line. This suggests that the assumption that the relationship is linear is reasonable.
- ➢ The residuals roughly form a "horizontal band" around the 0 line. This suggests that the variances of the error terms are equal.

Do these seem valid for our data?
I think those seem valid to the given data.

g) *Use the residuals to estimate* σ^2.
The formula to estimate σ^2 is

$$\hat{\sigma}^2 = \frac{1}{n-2}\sum_{i=1}^{n}(y_i - \hat{\beta}_0 - \hat{\beta}_1 x_i)^2$$

In R, the above formula can be interpreted as:

```
#Use the residuals to estimate variance
variance = sum(df$res^2)/(count(df)-2)
sd = sqrt(variance)
```

The output:
σ^2 = 677442.2
σ = 823.068

What is the interpretation of σ^2*?*
The Residual Standard Error is the average amount that the response will deviate from the true regression line. In our example, the actual perimeter required to stop can deviate from the true regression line by approximately **823.068**, on average. In other words, given that the mean perimeter for all data points to stop is **-471.43579** and that the Residual Standard Error is **823.068**. It's also worth noting that the Residual Standard Error was calculated with 46 degrees of freedom. Simplistically, degrees of freedom are the number of data points that went into the estimation of the parameters used after taking into account these parameters (restriction). In our case, we had 48 data points and two parameters (intercept and slope).

2. Question 2

a) In R: To test whether the water is cooling, plot a graph of the temperatures versus the time and make a least square fit of a straight line to the data.

R-Code

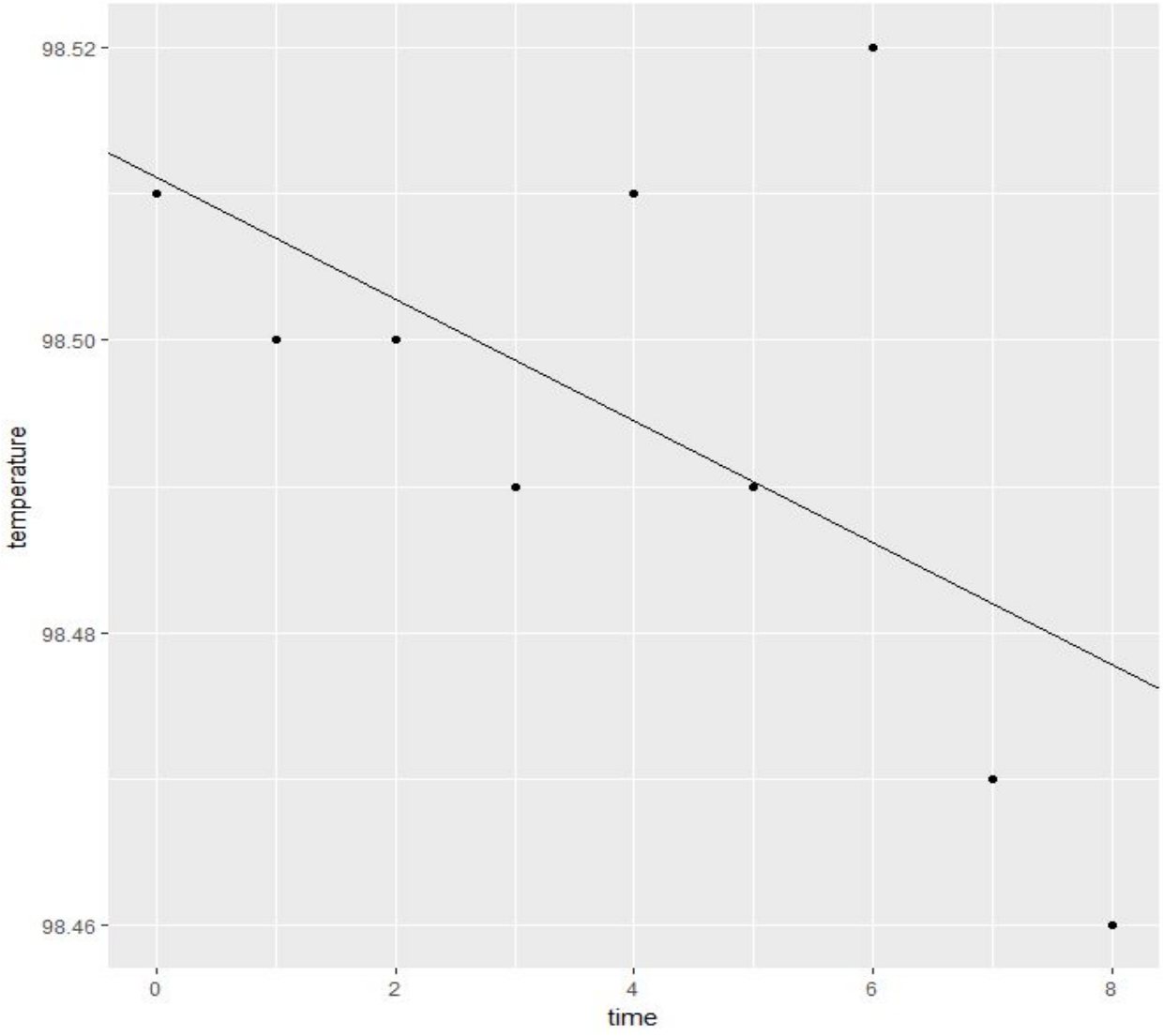
```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#load csv data
df = data.frame(time = c(seq(0,8,1)),
          temperature = c(98.51,98.50,98.50,98.49,98.51,98.49,98.52,98.47,98.46))

#perform a linear regression and show the summary
z = lm(temperature~time, data=df)

#plot the data as a scatter plot and add the regression line
ggplot(df, aes(x=time, y=temperature)) +
 geom_point() +
 geom_abline(intercept = z$coefficients[1],slope = z$coefficients[2])

summary(z)
```

Summary of z

```
Call:
lm(formula = temperature ~ time, data = df)

Residuals:
      Min        1Q    Median        3Q       Max
-0.017778 -0.008611 -0.002778 -0.000278  0.033889

Coefficients:
             Estimate Std. Error  t value Pr(>|t|)
(Intercept) 98.511111   0.010339 9528.456   <2e-16 ***
time        -0.004167   0.002172   -1.919   0.0965 .
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.01682 on 7 degrees of freedom
Multiple R-squared:  0.3447,    Adjusted R-squared:  0.2511
F-statistic: 3.682 on 1 and 7 DF,  p-value: 0.09651
```

Is there a statistically significant slope to the graph at the 95% confidence level?
No, since the the significant code of the slope was indicated by (.) which means the slope is only statistically significant at the 90% not 95 %.

The plot

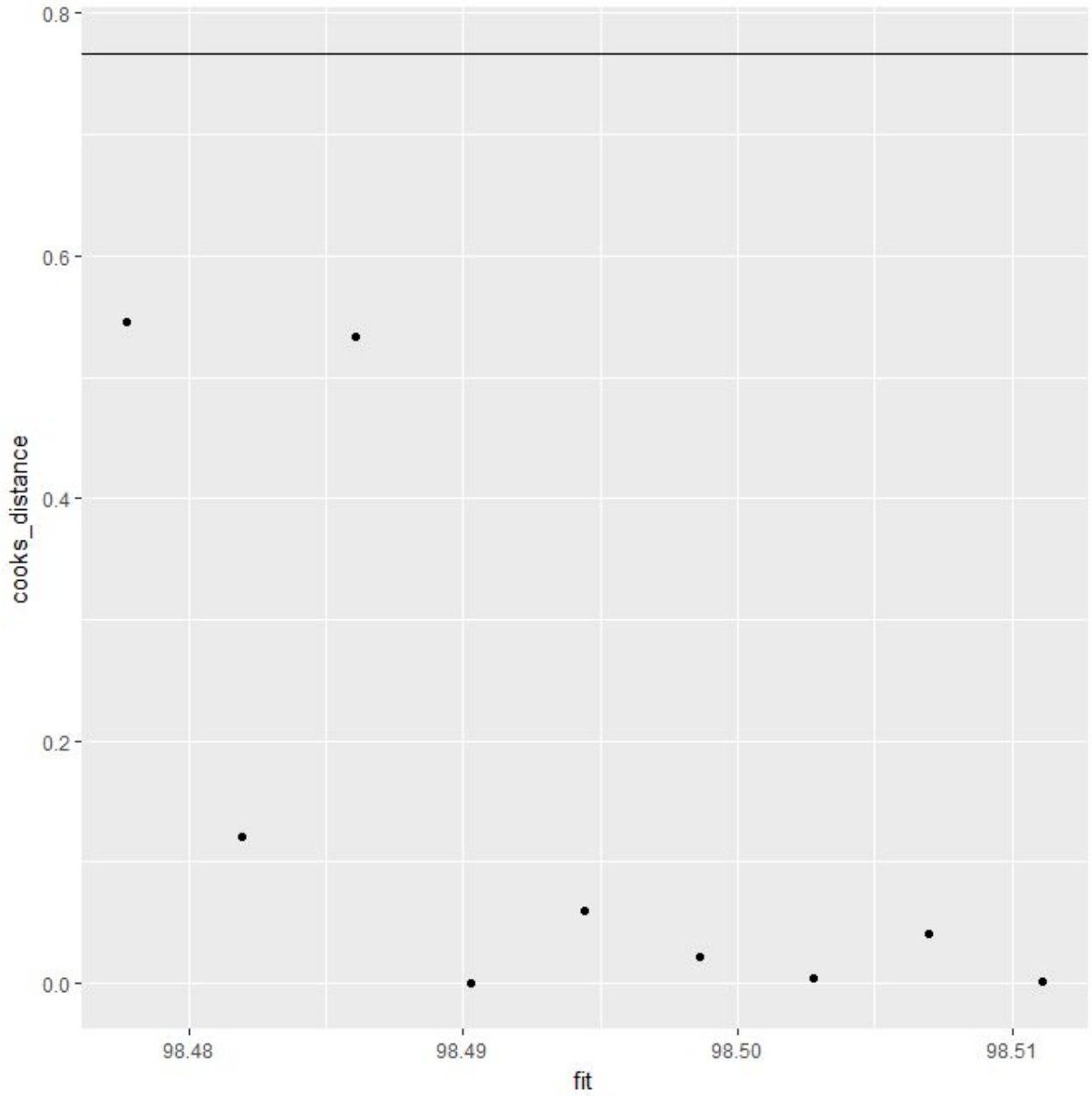


b) In R: Plotting Cook's distance.

The R-code

```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#create data frame
df = data.frame(time = c(seq(0,8,1)),
          temperature = c(98.51,98.50,98.50,98.49,98.51,98.49,98.52,98.47,98.46))

#perform a linear regression and show the summary
z = lm(temperature~time, data=df)

#get the summary of the linier regression
summary(z)

#get cook distance
cook <- cooks.distance(z)

#put the fitted values and Cook's distance into a data frame
data = data.frame(fit = fitted(z),  cooks_distance = cook)
#cut-off value
cut <- qf(0.5, 2, length(data$fit)-2, lower.tail = FALSE)

ggplot(data, aes(x=fit, y=cooks_distance)) +
  geom_point() +
  geom_hline(yintercept=cut)
```

The plot



Is there a data point that seems an outlier with strong influence on the regression?

It seems there isn't since all data points lie below the cut off cook's distance. Therefore we can assume that there is not any potential outliers in our data points.

c) *In R: The student is suddenly unsure about the data point at time = 6 min. Was there a mistake? The only way to know is to repeat the experiment. However, the student only wants to repeat the experiment, if omitting this value would change the conclusions. Should the student repeat the experiment?*

I think the student should not repeat the experiment since the all data points are still below the cut off cook distance. However, if the student insists to do that, of course he should repeat the experiment.

In order to proof this statement, let's remove the data point at time = 6 min and repeat the experiment.

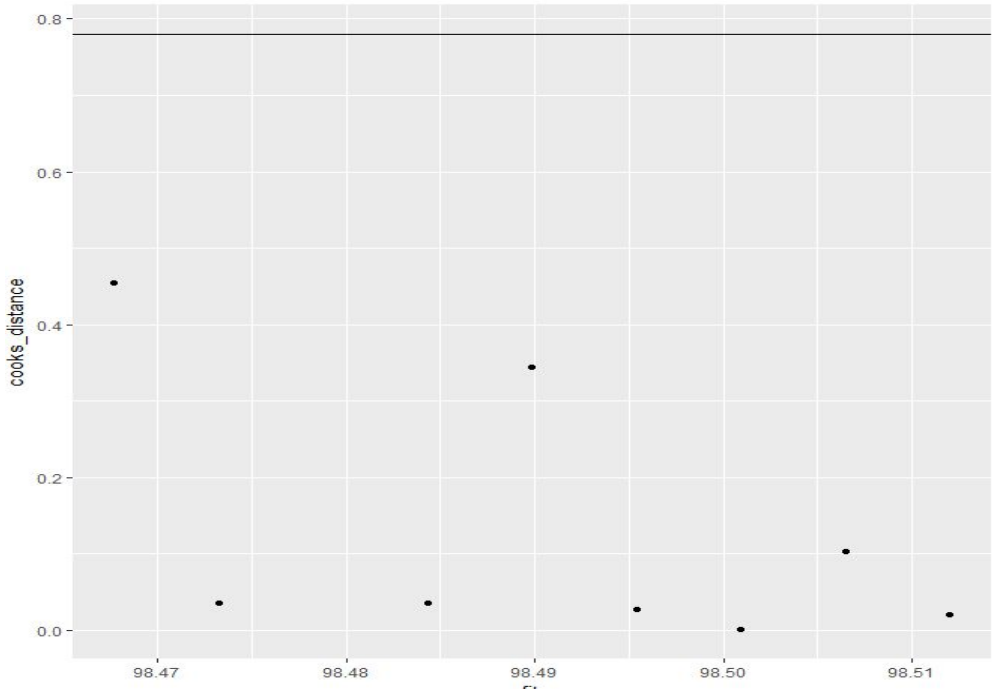
The R-Code

```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#create data frame
df = data.frame(time = c(seq(0,8,1)),
          temperature = c(98.51,98.50,98.50,98.49,98.51,98.49,98.52,98.47,98.46))

#omitting the value at minute 6
df <- df[-7,]

#perform a linear regression and show the summary
z = lm(temperature~time, data=df)

#get the summary of the linier regression
summary(z)

#get cook distance
cook <- cooks.distance(z)

#put the fitted values and Cook's distance into a data frame
data = data.frame(fit = fitted(z), cooks_distance = cook)
#cut-off value
cut <- qf(0.5, 2, length(data$fit)-2, lower.tail = FALSE)

ggplot(data, aes(x=fit, y=cooks_distance)) +
  geom_point() +
  geom_hline(yintercept=cut)
```

The summary of the linear regression

```
Call:
lm(formula = temperature ~ time, data = df)

Residuals:
       Min         1Q     Median         3Q        Max
-0.0077027 -0.0056757 -0.0026351  0.0007095  0.0201351

Coefficients:
             Estimate Std. Error   t value Pr(>|t|)
(Intercept) 98.512027   0.006068 16235.875  < 2e-16 ***
time        -0.005541   0.001324    -4.185  0.00578 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.009864 on 6 degrees of freedom
Multiple R-squared:  0.7448,    Adjusted R-squared:  0.7023
F-statistic: 17.51 on 1 and 6 DF,  p-value: 0.005784
```

The plot of the cook’s distance:



From the summary, we can see that there is not any significant changes of the interceptor and the slope. Also, If we observe the cook distance, all data points still lie below the cut off cook distance. So, the student should not worry about if there is outlier in dataset.

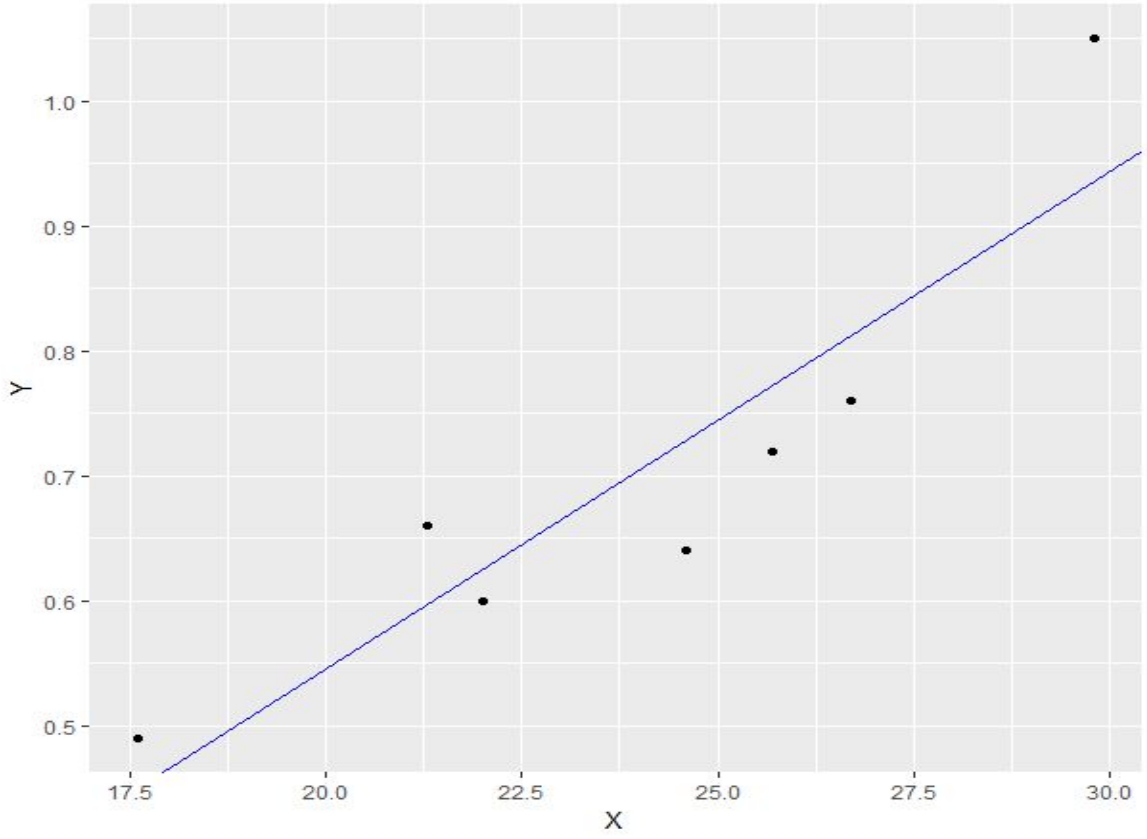
3. Question 3

a) *Do first a classical LS fit neglecting the uncertainties in Y.*

The R-Code

```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#load csv data
df = data.frame(X = c(26.7,22.0,17.6,25.7,21.3,29.8,24.6),
                Y = c(0.76,0.60,0.49,0.72,0.66,1.05,0.64))

#perform a linear regression and show the summary
z = lm(Y~X, data=df)

#plot the both unweight and weight linear model
ggplot(df, aes(x=X, y=Y)) +
  geom_point() +
  geom_abline(intercept = z$coefficients[1],slope = z$coefficients[2], color="blue")
```

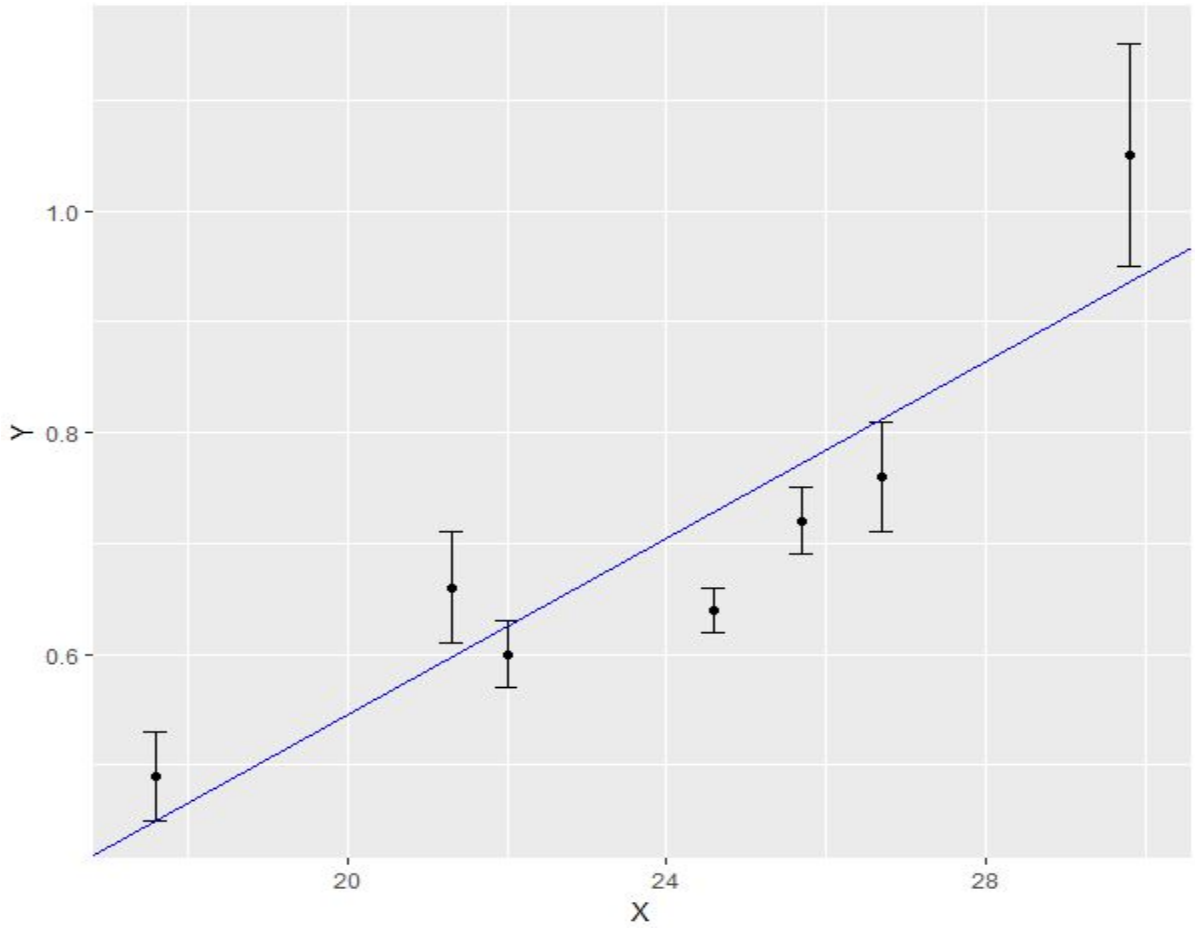
The output



Make an errorbar plot of the data and add the regression line. Give a summary of the fit results and interpret the results.

R-Code

```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#load csv data
df = data.frame(X = c(26.7,22.0,17.6,25.7,21.3,29.8,24.6),
            Y = c(0.76,0.60,0.49,0.72,0.66,1.05,0.64),
            Y_uncertainty=c(0.05,0.03,0.04,0.03,0.05,0.10,0.02))

#perform a linear regression and show the summary
z = lm(Y~X, data=df)

#plot the both the regression plus the erro bar
ggplot(df, aes(x=X, y=Y)) +
  geom_point() +
  geom_errorbar(aes(ymin = Y - Y_uncertainty,
                ymax = Y + Y_uncertainty,
                width=0.3)) +
  geom_abline(intercept = z$coefficients[1],slope = z$coefficients[2], color="blue")

#show the summary of the linear regression
summary(z)
```



The summary of the linear regression

```
Call:
lm(formula = Y ~ X, data = df)

Residuals:
       1        2        3        4        5        6        7
-0.05218 -0.02485  0.04052 -0.05232  0.06305  0.11427 -0.08848

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.251993   0.199295  -1.264  0.26181
X            0.039857   0.008221   4.848  0.00468 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.08072 on 5 degrees of freedom
Multiple R-squared:  0.8246,    Adjusted R-squared:  0.7895
F-statistic: 23.51 on 1 and 5 DF,  p-value: 0.004681
```

The interpretation of the results:

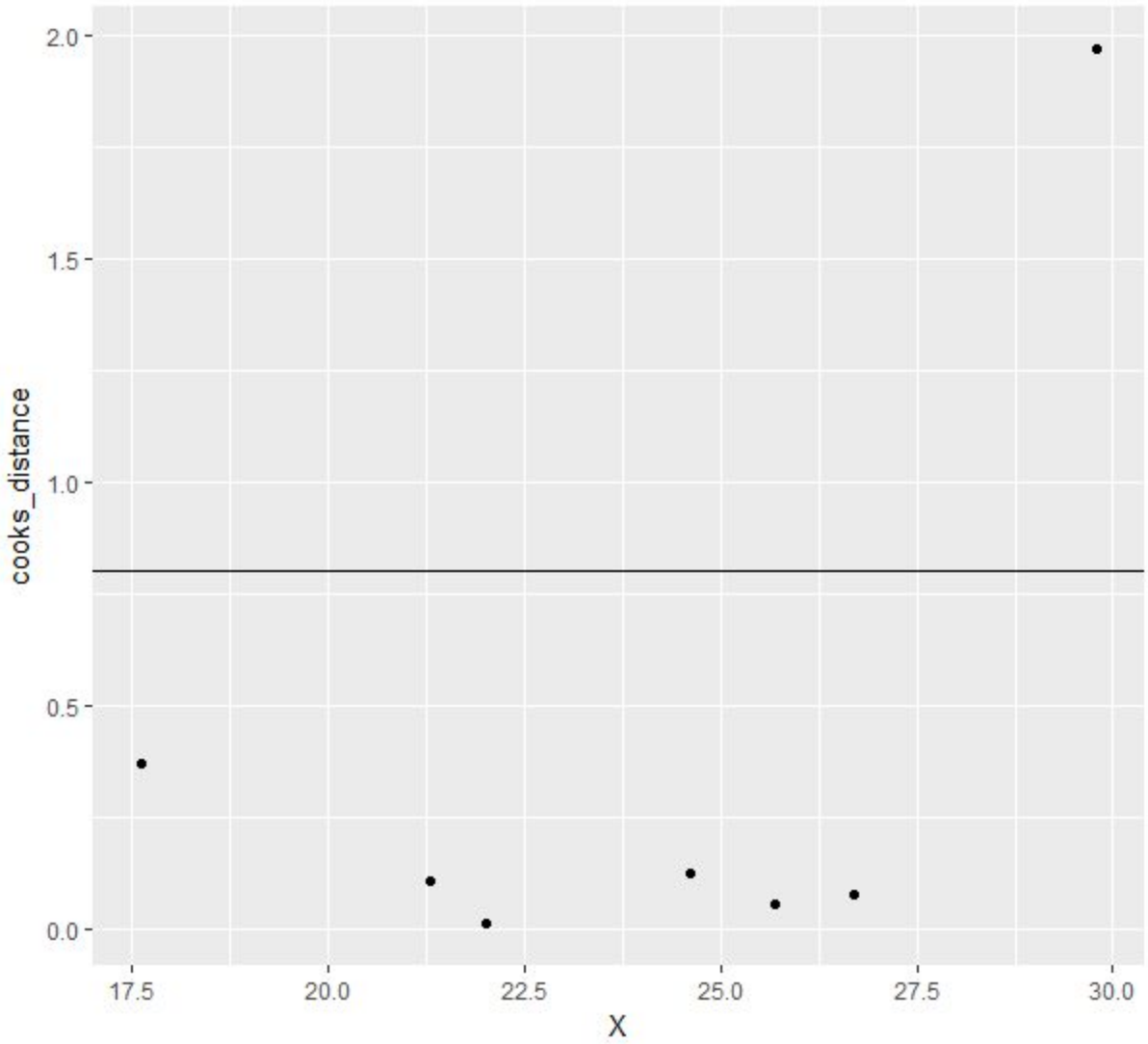
- The formula call is lm(formula = Y ~ X, data = df)
- The residuals are the difference between the difference between the actual observed response values and the response values that the model predicted.
- The coefficients are:
    - The estimate value of interceptor is -0.251993, the standard error 0.199295, the t-value is -1.264, the p-value is 0.26181 and the significant code is blank
    - The estimate value of slope is 0.039857, the standard error is 0.008221, the t value is ** (0.01)
    - The significant code tells us at which confidence level the intercept or the slope fit
- The residual standard error measures of the *quality* of a linear regression fit. In our case the value is 0.09072
- The multiple R-squared measures of how well the model is fitting the actual data. The value is 0.8246
- The F-statistic tells us the whether there is a relationship between our estimator and the response variables. In this case the value is 23.51, with degree of freedom 5 and p value 0.004681

b) Plot the cook's distance of the classical LS vs. X.

The R-Code

```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#load csv data
df = data.frame(X = c(26.7,22.0,17.6,25.7,21.3,29.8,24.6),
          Y = c(0.76,0.60,0.49,0.72,0.66,1.05,0.64),
          Y_uncertainty=c(0.05,0.03,0.04,0.03,0.05,0.10,0.02))

#perform a linear regression and show the summary
z = lm(Y~X, data=df)


#Plot the cook's distance of the classical LS vs. X
cook <- cooks.distance(z)

#put the fitted values and Cook's distance into a data frame
```

```
data = data.frame(X= df$X, cooks_distance = cook)

#cut-off value
cut <- qf(0.5, 2, length(data$X)-2, lower.tail = FALSE)

#plot the cook distance
ggplot(data, aes(x=X, y=cooks_distance)) +
  geom_point() +
  geom_hline(yintercept=cut)
```



Calculate the cut-off value for problematic data points.

```
#cut-off value
cut <- qf(0.5, 2, length(data$X)-2, lower.tail = FALSE)
```

The output

```
[1] 0.7987698
```

Is there a data point that is problematic?

Based on cook's distance theory, there is one data point that seems problematic. This data point lies above the cook distance (X=29.8). The position of this data point is 1.16856 above the cook's distance cut off.

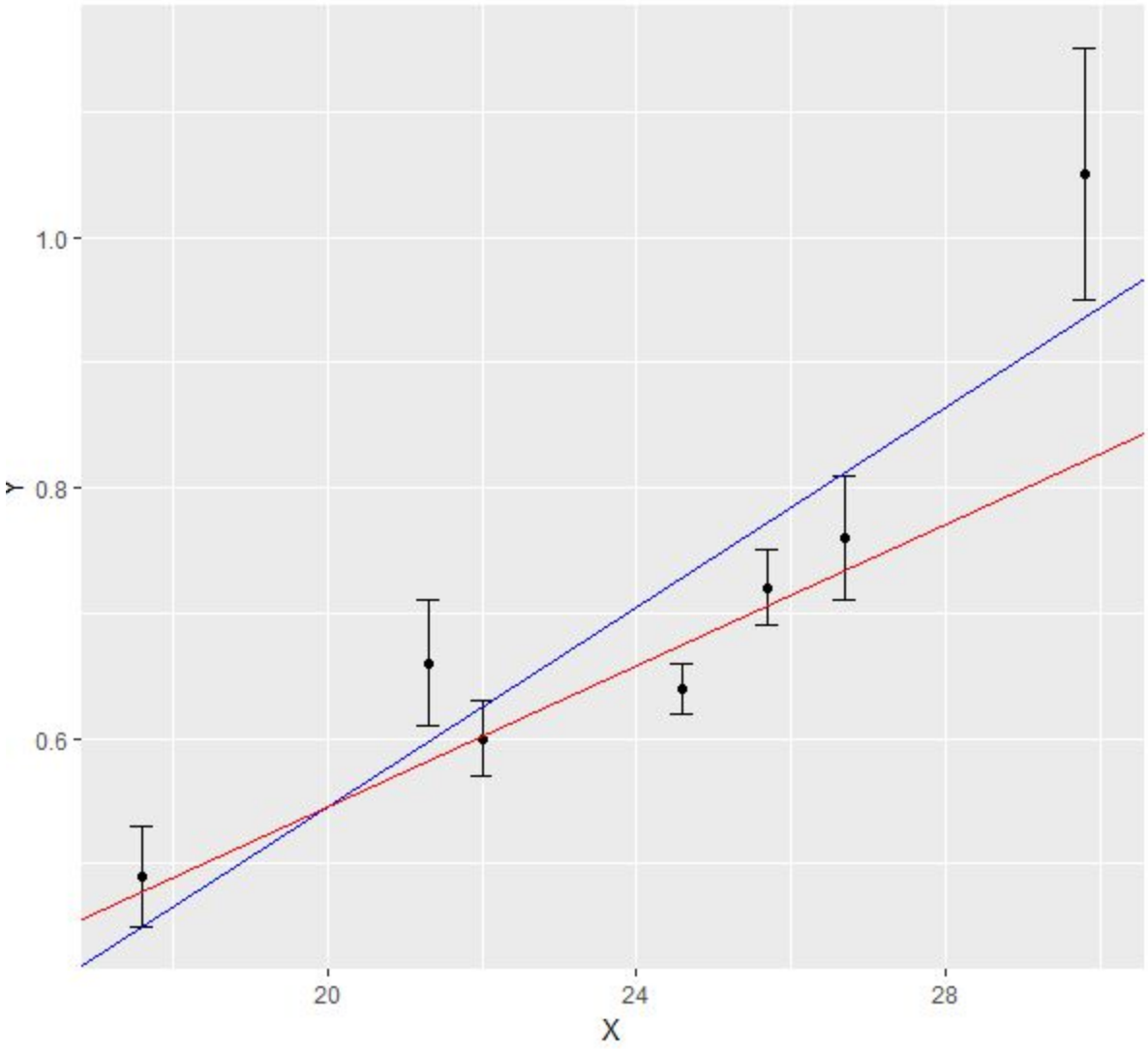
c) *From the error bar plot it is obvious that one of the assumptions of the classical LS fit is violated. Which one?*
   - The least squares assumption of constant variance in the residuals is violated.

Use the error bars to make a weighted LS fit plot the data and regression line (you can also add the new regression line to the previous plot)

The R-Code

```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#load csv data
df = data.frame(X = c(26.7,22.0,17.6,25.7,21.3,29.8,24.6),
          Y = c(0.76,0.60,0.49,0.72,0.66,1.05,0.64),
          Y_uncertainty=c(0.05,0.03,0.04,0.03,0.05,0.10,0.02))

#perform a linear regression and show the summary
z = lm(Y~X, data=df)
#perform a linear regression with weight
z_weight = lm(Y~X, data=df, weights = 1/(df$Y_uncertainty^2))

#plot the both unweight and weight linear model
ggplot(df, aes(x=X, y=Y)) +
  geom_point() +
  geom_errorbar(aes(ymin = Y - Y_uncertainty, ymax = Y + Y_uncertainty,width=0.3)) +
  geom_abline(intercept = z$coefficients[1],slope = z$coefficients[2], color="blue") +
  geom_abline(intercept = z_weight$coefficients[1],slope = z_weight$coefficients[2],
color="red")

#summary z
summary(z)
```

```
#summary z_weight
summary(z_weight)
```



In the above plot, the weighted regression marked in **red** and the classical regression marked in **blue**.

Here is the summary of both regressions:

- The summary of classical regression

```
Call:
lm(formula = Y ~ X, data = df)

Residuals:
       1        2        3        4        5        6        7
-0.05218 -0.02485  0.04052 -0.05232  0.06305  0.11427 -0.08848

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.251993   0.199295  -1.264  0.26181
X            0.039857   0.008221   4.848  0.00468 **
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.08072 on 5 degrees of freedom
Multiple R-squared:  0.8246,    Adjusted R-squared:  0.7895
F-statistic: 23.51 on 1 and 5 DF,  p-value: 0.004681
```

- The summary of weighted regression

```
Call:
lm(formula = Y ~ X, data = df, weights = 1/(df$Y_uncertainty^2))

Weighted Residuals:
       1        2        3        4        5        6        7
 0.51617 -0.04249  0.32895  0.46963  1.57043  2.28140 -1.74015

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -0.020888   0.173322  -0.121   0.9088
X            0.028280   0.007288   3.881   0.0116 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.503 on 5 degrees of freedom
Multiple R-squared:  0.7507,    Adjusted R-squared:  0.7009
F-statistic: 15.06 on 1 and 5 DF,  p-value: 0.01164
```

From the summary results both classical and weighted regressions, we can see that the slope of the classical regression is higher than the weighted one. The difference is 0.01157643. However, looking at the interceptor, the value of weighted regression is greater than the classical regression.

The reason why the value of the slope changes because specifying a column of weights affects the sums of squares and parameter estimates in the following ways:

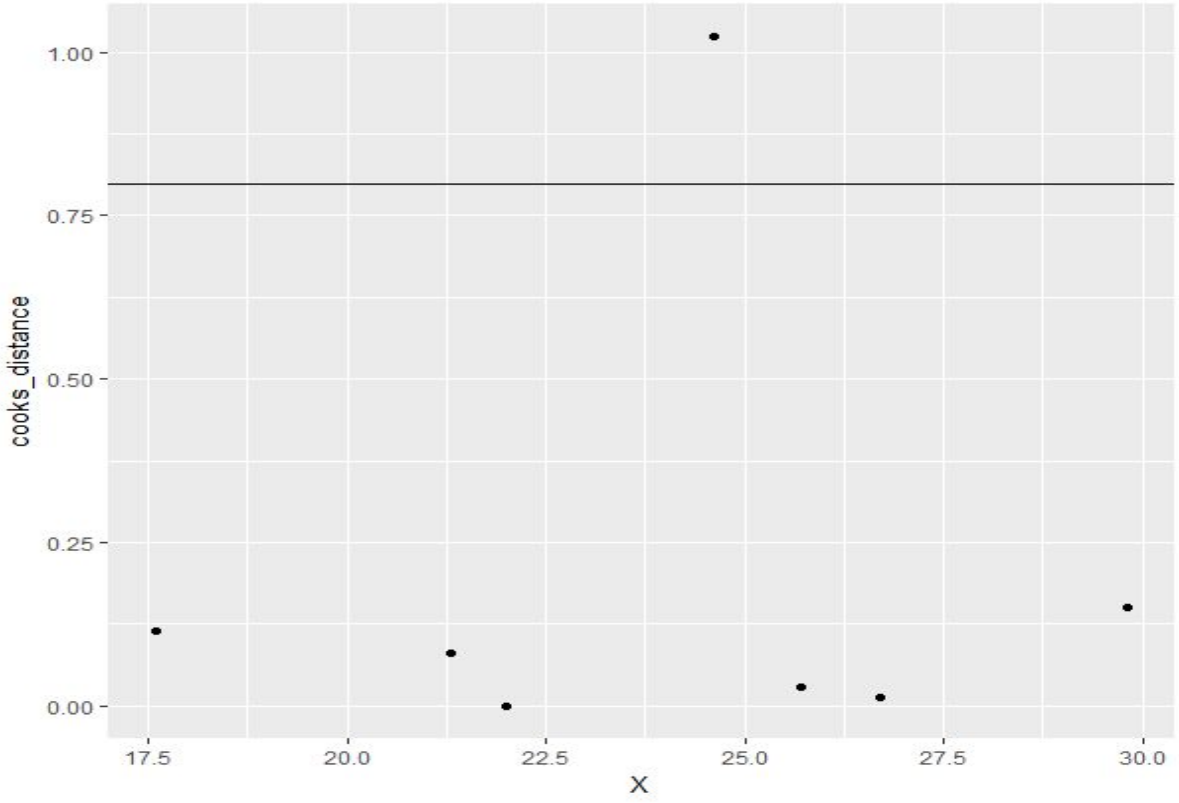
- The sums of squares become weighted sums of squares.
- A weighted mean is used in the total sum of squares.
- A weighted least squares criterion is used to estimate the parameters.

d) Plot the Cook's distance of the weighted fit.

The R-Code

```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#load csv data
df = data.frame(X = c(26.7,22.0,17.6,25.7,21.3,29.8,24.6),
                Y = c(0.76,0.60,0.49,0.72,0.66,1.05,0.64),
                Y_uncertainty=c(0.05,0.03,0.04,0.03,0.05,0.10,0.02))

#perform a linear regression with weight
z_weight = lm(Y~X, data=df, weights = 1/(df$Y_uncertainty^2))

#Plot the cook's distance of the classical LS vs. X
cook <- cooks.distance(z_weight)

#put the fitted values and Cook's distance into a data frame
data = data.frame(X= df$X, cooks_distance = cook)

#cut-off value
cut <- qf(0.5, 2, length(data$X)-2, lower.tail = FALSE)

#plot the cook distance
ggplot(data, aes(x=X, y=cooks_distance)) +
  geom_point() +
  geom_hline(yintercept=cut)
```

The output



Are there still problematic data points?

Looking at the above plot, it seems there is still a data point which seems to be problematic. One data point (X=24.6) lies above the cut off cook's distance, but now the gap is not really high. It's only 0.2243355 above the cook's distance cut off.

4. Question 4

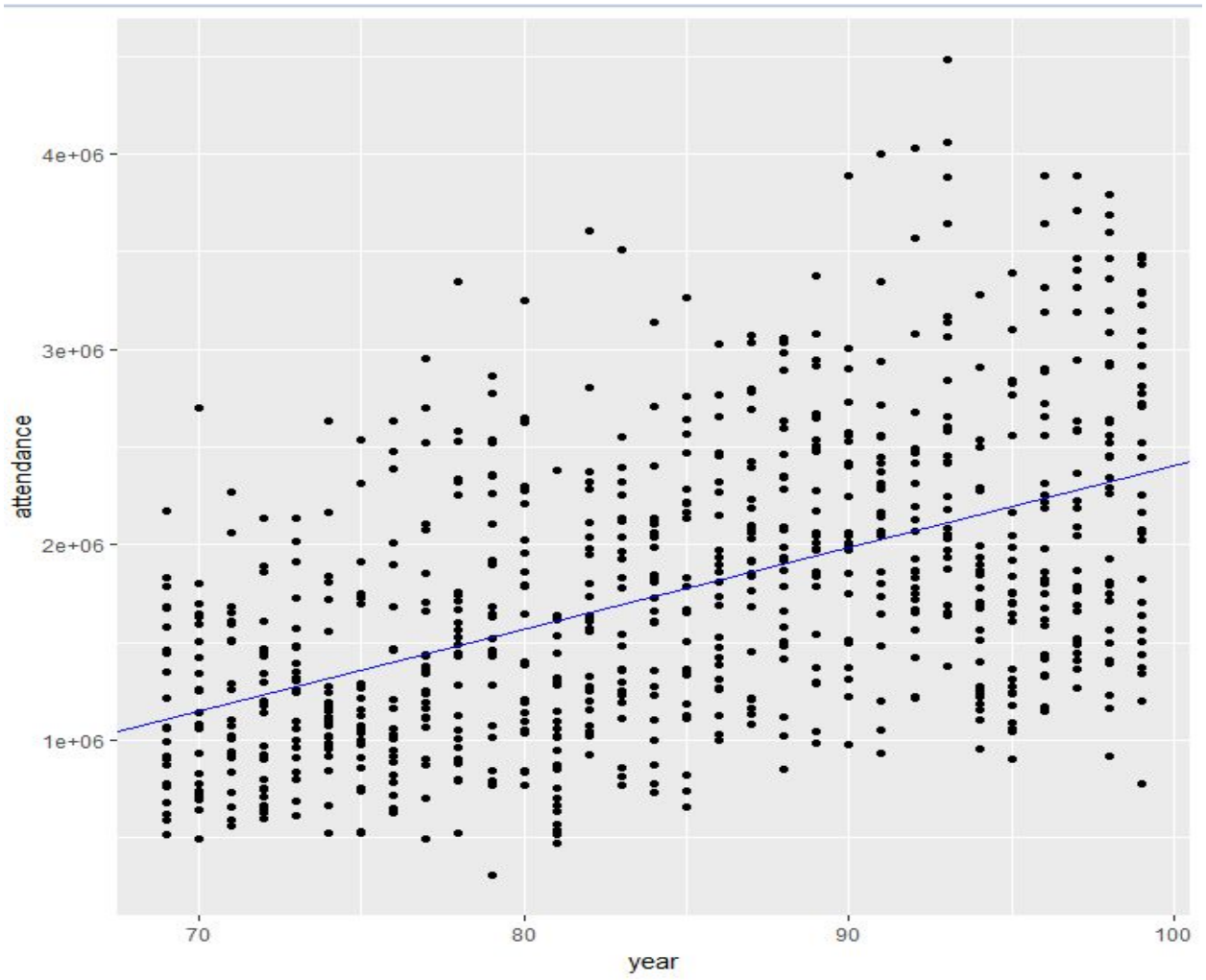
a) Make individual linear regressions of attendance as a function of (i) year, (ii) wins, (iii) runs.scored. Plot each data pair using a scatter plot, add the regression line and report the R^2.

The R-Code

```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#set working directory
setwd("X:/My Desktop/Statistic/assignment5/")
#read csv
df = read.csv(file = "games(2).csv" , header = TRUE)

#question a
#perform a linear regression and show the summary
z_1 = lm(attendance~year, data=df)
z_2 = lm(attendance~wins, data=df)
z_3 = lm(attendance~runs.scored, data=df)

#plot lm(attendance~year, data=df)
ggplot(df, aes(x=year, y=attendance)) +
  geom_point() +
  geom_abline(intercept = z_1$coefficients[1],slope = z_1$coefficients[2], color="blue")


#plot lm(attendance~wins, data=df)
ggplot(df, aes(x=wins, y=attendance)) +
  geom_point() +
  geom_abline(intercept = z_2$coefficients[1],slope = z_2$coefficients[2], color="blue")

#plot lm(attendance~runs.scored, data=df)
ggplot(df, aes(x=runs.scored, y=attendance)) +
  geom_point() +
  geom_abline(intercept = z_3$coefficients[1],slope = z_3$coefficients[2], color="blue")

#summary of each individual regression
summary(z_1)
summary(z_2)
summary(z_3)
```

The plot of each regression and the R^2

- **attendance~year**

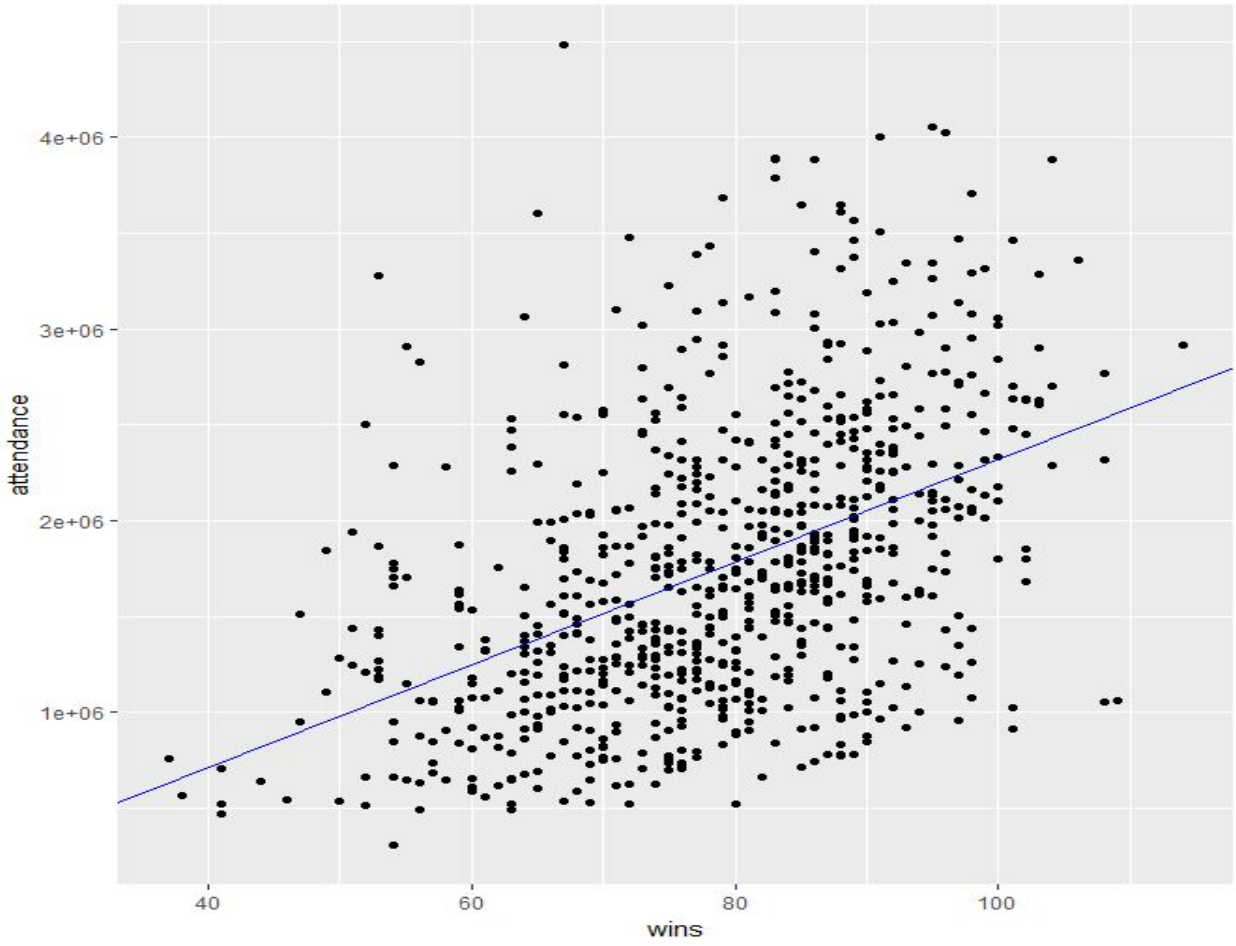


The summary

```
Call:
lm(formula = attendance ~ year, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-1588150  -469444   -66446   403517  2373412

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1788142     214464  -8.338 3.26e-16 ***
year           41915       2524  16.609  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 643300 on 806 degrees of freedom
Multiple R-squared:  0.255,     Adjusted R-squared:  0.2541
F-statistic: 275.9 on 1 and 806 DF,  p-value: < 2.2e-16
```

The value of the R^2 is 0.255

- **attendance~wins**

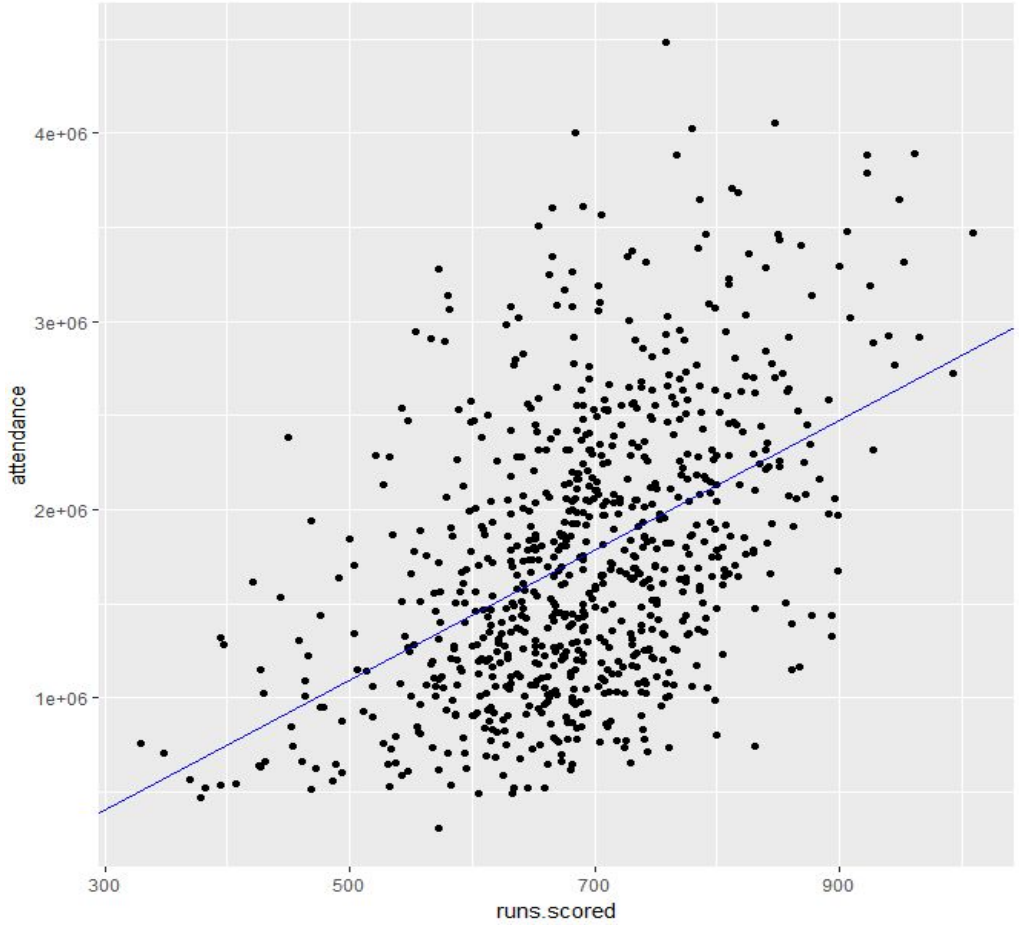


The summary

```
Call:
lm(formula = attendance ~ wins, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-1501175  -441906   -99770   371038  3044550

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -354957     145799  -2.435   0.0151 *
wins           26773       1827  14.653   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 662300 on 806 degrees of freedom
Multiple R-squared:  0.2104,    Adjusted R-squared:  0.2094
F-statistic: 214.7 on 1 and 806 DF,  p-value: < 2.2e-16
```

The value of R^2 is 0.2104

- **attendance~runs.scored**



The summary

```
Call:
lm(formula = attendance ~ runs.scored, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-1500108  -475739   -84457   418910  2494292

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -625214.2   156994.1  -3.982 7.44e-05 ***
runs.scored    3448.9      225.1  15.321  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 655900 on 806 degrees of freedom
Multiple R-squared:  0.2255,    Adjusted R-squared:  0.2246
F-statistic: 234.7 on 1 and 806 DF,  p-value: < 2.2e-16
```

The value of R^2 is 0.2255.

b) Now fit the attendance as a function of two variables, “year” and “wins”. Report the R^2. How does it compare to the R^2 for the individual fits.

The R-Code

```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#set working directory
setwd("X:/My Desktop/Statistic/assignment5/")
#read csv
df = read.csv(file = "games(2).csv" , header = TRUE)

#question a
#perform a linear regression and show the summary
z = lm(attendance~year+wins, data=df)

#summary of each individual regression
summary(z)
```

The output

```
Call:
lm(formula = attendance ~ year + wins, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-1313538  -369281   -52534   306981  2689761

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) -4217515     217019  -19.43   <2e-16 ***
year           44126       2088   21.13   <2e-16 ***
wins           28468       1468   19.39   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 531500 on 805 degrees of freedom
Multiple R-squared:  0.4921,    Adjusted R-squared:  0.4908
F-statistic:   390 on 2 and 805 DF,  p-value: < 2.2e-16
```

After applying the multiple regression using year and wins as independent variables, the value of R^2 is now changed to 0.4921. It’s twice of value R^2 of each individual fits.

c) Now add the variable: runs.scored and make a multiple linear regression using the three variables “year”, “wins”, and “runs.scored”

The R-Code

```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#set working directory
setwd("X:/My Desktop/Statistic/assignment5/")
#read csv
df = read.csv(file = "games(2).csv" , header = TRUE)

#question a
#perform a linear regression and show the summary
z = lm(attendance~year+wins+runs.scored, data=df)

#summary of each individual regression
summary(z)
```

The output

```
Call:
lm(formula = attendance ~ year + wins + runs.scored, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-1307134  -369007   -46193   297975  2713071

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -4237564.0   218233.3 -19.418   <2e-16 ***
year           45119.8     2372.1  19.021   <2e-16 ***
wins           29768.4     2079.9  14.312   <2e-16 ***
runs.scored     -241.1      273.1  -0.883    0.378
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 531600 on 804 degrees of freedom
Multiple R-squared:  0.4926,    Adjusted R-squared:  0.4907
F-statistic: 260.2 on 3 and 804 DF,  p-value: < 2.2e-16
```

➢ *Discuss the output of the fit. If you did it right then there is not much increase in the R^2 by adding the variable "runs.scored".*

After applying multiple regression with three independent variables: (year, wins, and runs.scored), the value of R^2 is 0.4926. However, the R^2 does not increase significantly compared to the multiple regression with only two dependent variables (years and wins). The delta is 0.0005 only.

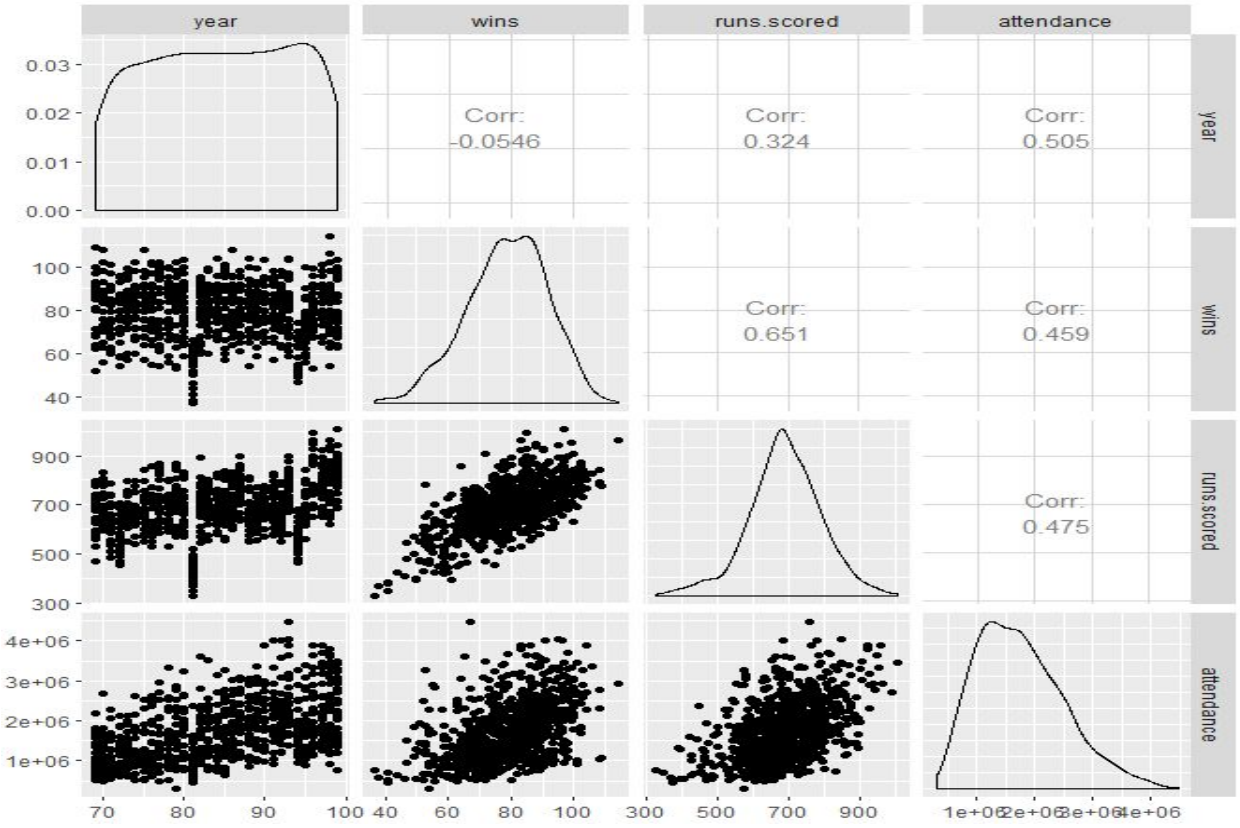
➢ *How is this possible?when it was reasonably well correlated with attendance individually, see (5a)? Investigate using a pairs() plot*

In order to investigate this phenomenon, let's call a method pairs() plot to get an overview of those variables (year, wins, runs.scored, attendance).

The R-Code

```
ggpairs(df[,c("year", "wins", "runs.scored", "attendance")])
```

The output



If we compare the scatter plot between attendance vs wins and attendance vs runs.scored, they are almost similar (see above figure). The similarity can also be seen from the correlation value. The correlation value of attendance vs wins is 0.459 and attendance vs runs.score is 0.475. Thus, adding runs.score to regression does not make any significant changes to the regression.

➢ *Do an anova test if adding the variable runs.scored does significantly change the model.*

The R-Code

```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#set working directory
setwd("X:/My Desktop/Statistic/assignment5/")
#read csv
df = read.csv(file = "games(2).csv" , header = TRUE)

#question a
#perform a linear regression and show the summary
z1 = lm(attendance~year+wins+runs.scored, data=df)
z2 = lm(attendance~year+wins, data=df)
results = anova(z1, z2)
print(results)
```

The output

```
Analysis of Variance Table

Model 1: attendance ~ year + wins + runs.scored
Model 2: attendance ~ year + wins
  Res.Df        RSS Df   Sum of Sq      F Pr(>F)
1    804 2.2719e+14
2    805 2.2741e+14 -1 -2.2032e+11 0.7797 0.3775
```

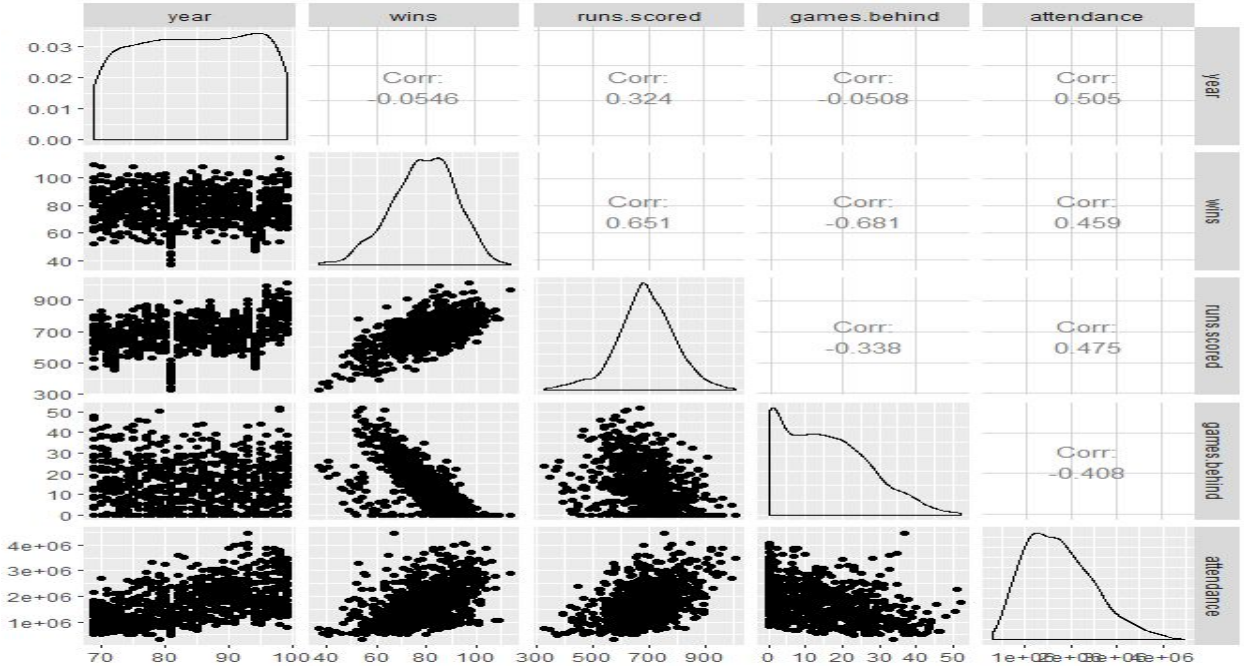
From the above value we can see that the p-value of ANOVA test is >= 0.05 therefore the variable runs.scored does not make any significant impact after we take it out from our regression.

d) Now do a full fit using all the variables. Give an interpretation of the coefficient for the variable "games.behind".

The R-Code

```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#set working directory
setwd("X:/My Desktop/Statistic/assignment5/")
#read csv
df = read.csv(file = "games(2).csv" , header = TRUE)

#question a
#perform a linear regression and show the summary
z = lm(attendance~year+wins+runs.scored+games.behind, data=df)

#summary of each individual regression
summary(z)

#plot individual pairs
ggpairs(df[,c("year", "wins", "runs.scored","games.behind" , "attendance")])
```

Individual plot of all variables



The summary

```
Call:
lm(formula = attendance ~ year + wins + runs.scored + games.behind,
    data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-1224464  -371263   -44829   303166  2785332

Coefficients:
                Estimate Std. Error t value Pr(>|t|)
(Intercept)  -3.796e+06  2.792e+05 -13.596   <2e-16 ***
year          4.362e+04  2.438e+03  17.892   <2e-16 ***
wins          2.504e+04  2.794e+03   8.963   <2e-16 ***
runs.scored  -3.804e+01  2.838e+02  -0.134   0.8934
games.behind -5.721e+03  2.268e+03  -2.523   0.0118 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 529800 on 803 degrees of freedom
Multiple R-squared:  0.4966,    Adjusted R-squared:  0.4941
F-statistic:   198 on 4 and 803 DF,  p-value: < 2.2e-16
```

The interpretation of the coefficients:

- The value of intercept is -3.796e+06, the standard error is 2.792e+05, the t-value is -13.596, the p-value <2e-16. Since the significant code is (***) therefore the intercept statistically significant at 99.9 % confidence interval (highly significant).
- The value of slope with ***year*** as independent variables is 4.362e+04, the standard error is 2.438e+03, the t-value is 17.892, and the p-value is <2e-16. Since the significant code is (***) therefore this slope statistically significant at 99.9 % confidence interval (highly significant).
- The value of slope with ***wins*** as independent variables is 2.504e+04, the standard error is 2.794e+03, the t-value is 8.963, and the p-value is <2e-16. Since the significant code is (***) therefore this slope statistically significant at 99.9 % confidence interval (highly significant).
- The value of slope with ***runs.scored*** as independent variables is -3.804e+01, the standard error is 2.838e+02, the t-value is -0.134, and the p-value is 0.8934. Since the p-value is greater than 0.05 therefore this slope doesn't tell much about the relationship between runs.scored and attendance.
- The value of slope with ***games.behind*** as independent variables is -5.721e+03, the standard error is 2.268e+03, the t-value is -2.523, and the p-value is 0.0118. Since the p-value is greater than 0.05 therefore this slope doesn't tell much about the relationship between games.behind and attendance.

Should you include the variable (i.e., does it significantly change the fit?)

Since the variable games.behind is not really statistically significant to the regression compared to two other variables (year and games), therefore we can take it out from the regression. But, if we compare it with variable runs.scored, we should include this variable in our regression since it gives statistically significant impact over runs.scored.

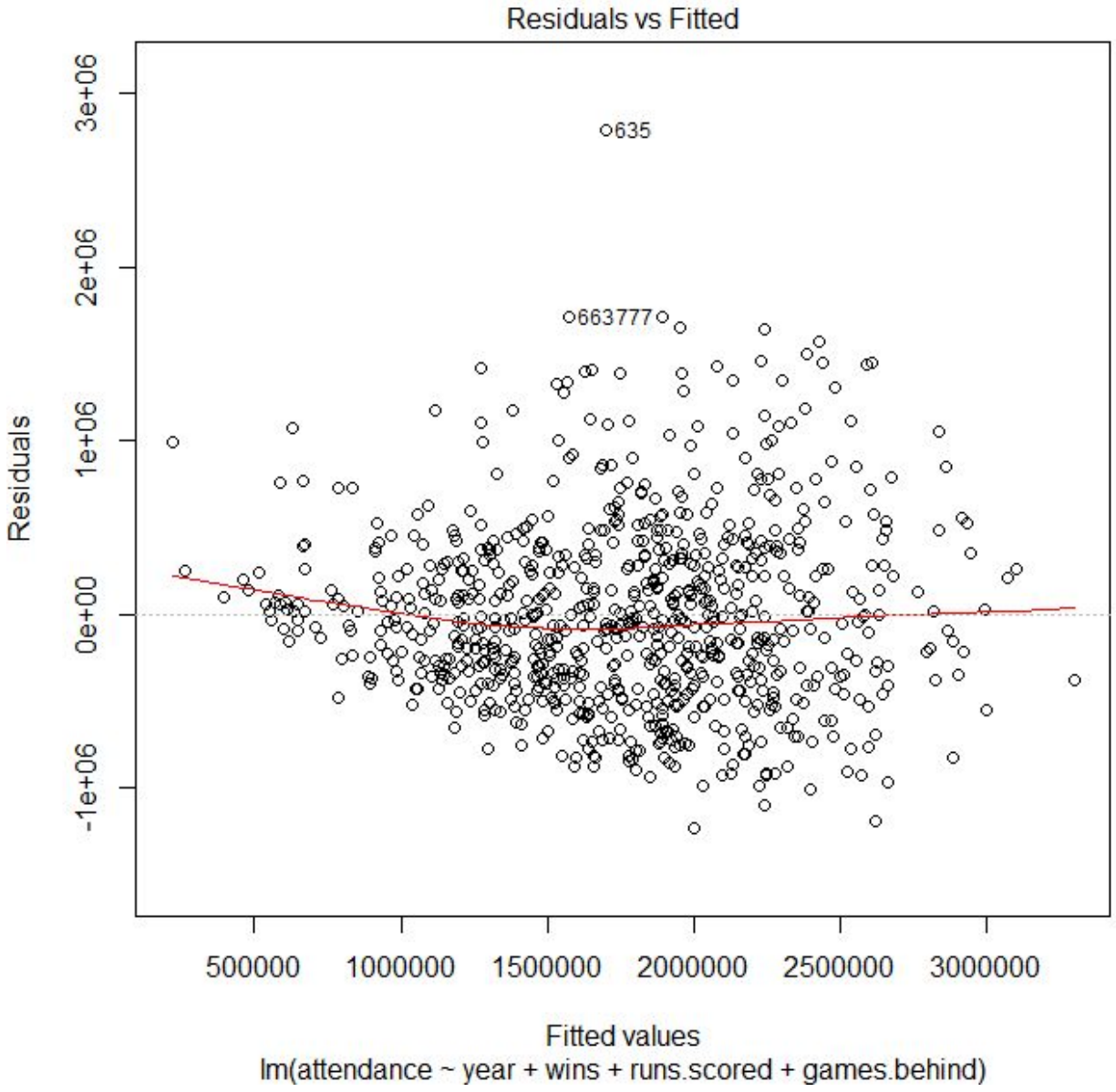
e) Investigate with the help of some regression diagnostics, if the main assumptions for linear regression are satisfied.

The R-Code

```
library(dplyr)
library(GGally)
library(lmodel2)
library(tidyverse)

#set working directory
setwd("X:/My Desktop/Statistic/assignment5/")
#read csv
df = read.csv(file = "games(2).csv" , header = TRUE)

#question a
#perform a linear regression and show the summary
z = lm(attendance~year+wins+runs.scored+games.behind, data=df)

#Create a series of diagnostic plots
plot(z)
```

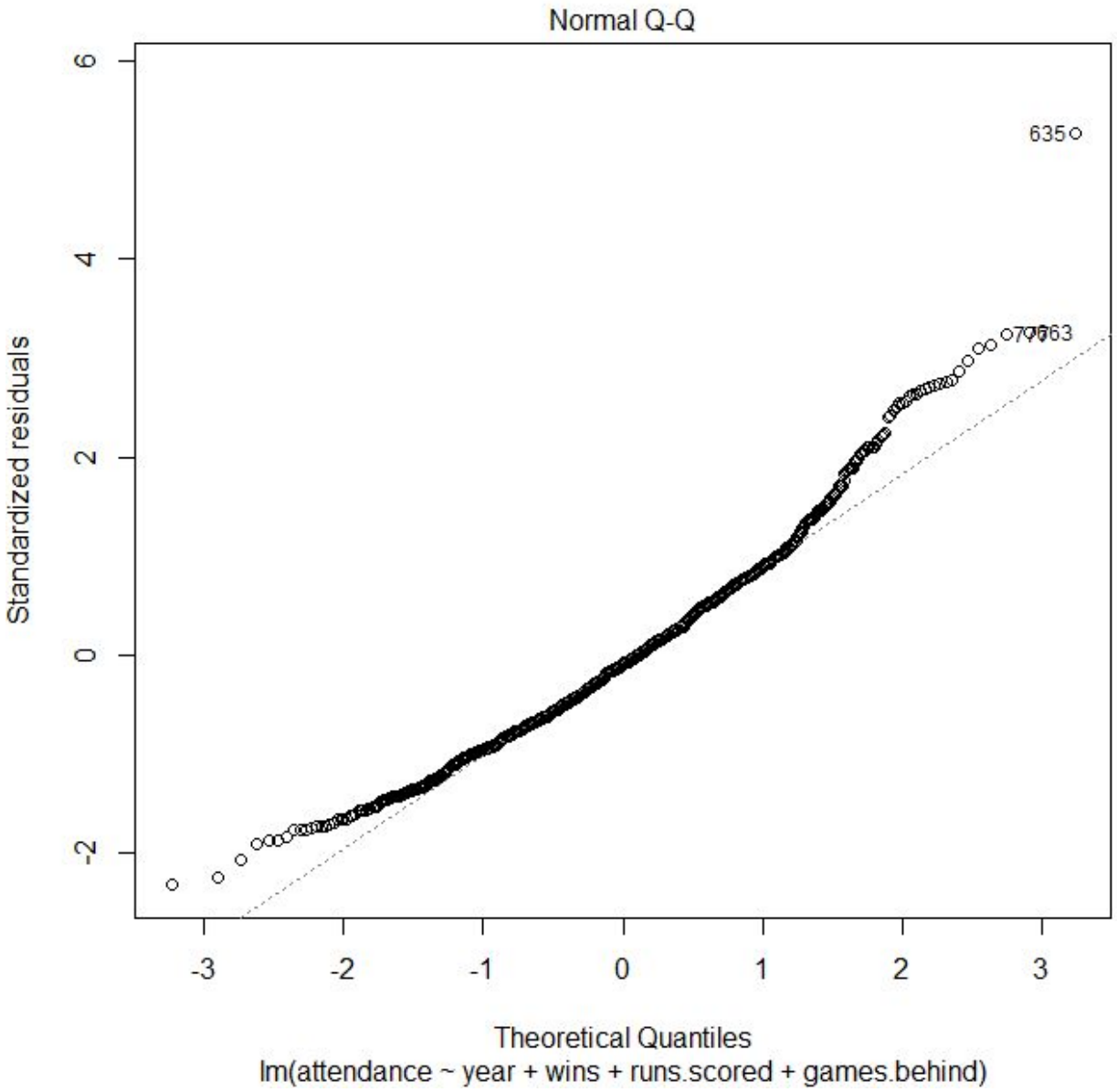
- **Checking Model Assumption (Plot of Residual)**

  The residuals "bounce randomly" around the 0 line. This suggests that the assumption that the relationship is linear is reasonable. Also, the residuals roughly form a "horizontal band" around the 0 line. This suggests that the variances of the error terms are equal.



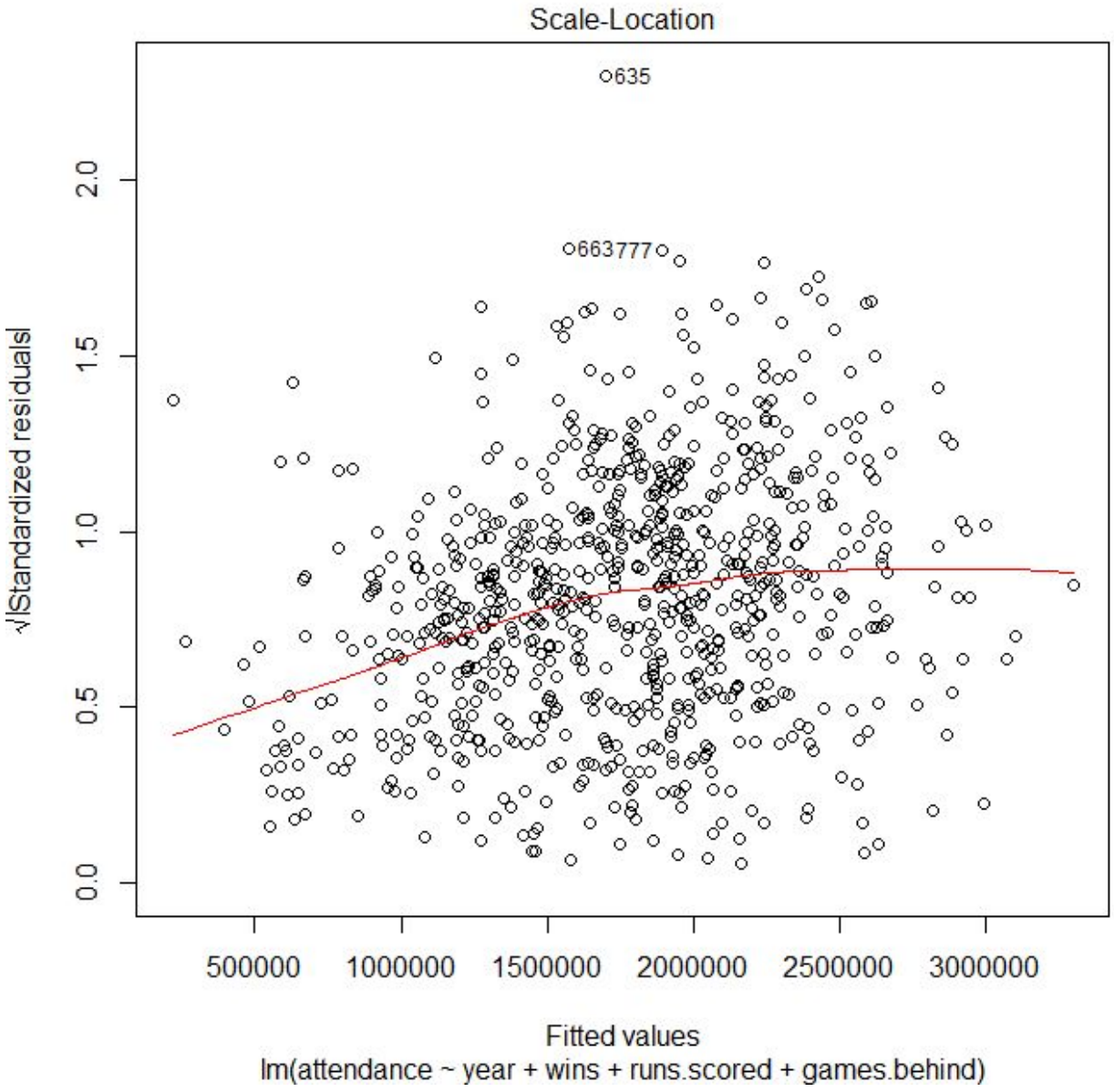
- **Check if residuals follow normal distributions**

  The ε_i should form a normal distribution with mean value of 0. In this case, the ε_i forms a normal distribution since they form a straight line as shown below.



- **ε_i Variance**

  The fitted values should have the same variance for all data points with variance Var(ε_i) = σ^2. The residuals are roughly around the 0 line. This suggests that the variances of the error terms are equal.



- **Outliers and Influential data points**

  We can detect the outliers by using cook's distance. If the values lies above the cook's distance therefore they are considered as outliers. From the below plot, we can see that half of data points lie above the cook's distance which would be considered as outliers. However, the distance is not really high from the cook's ct off.