

# Anscombe Datasets - Linear Regression Analysis

Eric B.

6/12/2022

Question 1 (10 points) As explained in Lesson 5, data exploration through visualization is important because statistics alone might not tell the whole story. This is best shown by the French statistician Francis Anscombe in 1973 when he presented four sets of data.

```
##reading in dataset from excel worksheet
library(readxl)
#each workbook has multiple sheets in it that contain each of the four datasets used for this assignment
#each worksheet has a different name
sheet_names <- excel_sheets("C:/Users/ericb/Desktop/TE 575/Week 5/Anscombe.xlsx")
#iterate function to get each dataset from each sheet in the workbook excel file
anscombe <- lapply(sheet_names, function(x) {
  #read all sheets to list
  ans_data <- read_excel("C:/Users/ericb/Desktop/TE 575/Week 5/Anscombe.xlsx", sheet = x) })
anscombeData
```

```
## [1]
##      x      y
## 1  10  8.04
## 2   8  6.95
## 3  13  7.58
## 4   9  6.81
## 5  11  8.33
## 6  14  9.96
## 7   7  7.24
## 8   4  4.26
## 9  12 10.84
## 10  7  5.26
## 11  5  5.68
## [2]
##      x      y
## 1  10  9.14
## 2   8  8.14
## 3  13 12.74
## 4   9  8.77
## 5  11  9.26
## 6  14  8.10
## 7   7  6.13
## 8   4  3.10
## 9  12  9.13
## 10  7  7.26
## 11  5  4.74
## [3]
##      x      y
## 1  10  7.46
## 2   8  6.77
## 3  13 12.74
## 4   9  7.11
## 5  11  8.81
## 6  14  8.84
## 7   7  8.47
## 8   4  5.39
## 9  12  8.15
## 10  7  6.42
## 11  5  5.73
## [4]
##      x      y
## 1   8  6.58
## 2   8  5.76
## 3   8  7.71
## 4   8  8.84
## 5   8  8.47
## 6   8  7.84
## 7  10 10.15
## 8  10 12.50
## 9   8  5.96
## 10  8  7.91
## 11  8  6.89
```

```
names(anscombeData) <- sheet_names #changing the default names for each dataset back to their original sheet name
```

Calculate the mean, variance, correlation, and a linear regression for each data set (No data partition). Using base R or ggplot2, create a visual representation of this data. What does this visualization show?

```
#Dataset 1 - calculations
anscombeData1$Data1

##      x      y
## 1  10  8.04
## 2   8  6.95
## 3  13  7.58
## 4   9  6.81
## 5  11  8.33
## 6  14  9.96
## 7   7  7.24
## 8   4  4.26
## 9  12 10.84
## 10  7  5.26
## 11  5  5.68

#mean calculations
Data1_mn <- mean(anscombeData1$x)
Data1_mn

## [1] 9

Data1_mn <- mean(anscombeData1$y)
Data1_mn

## [1] 7.589899

#variance calculations
Data1_var <- var(anscombeData1$x)
Data1_var

## [1] 11

Data1_var <- var(anscombeData1$y)
Data1_var

## [1] 4.127289

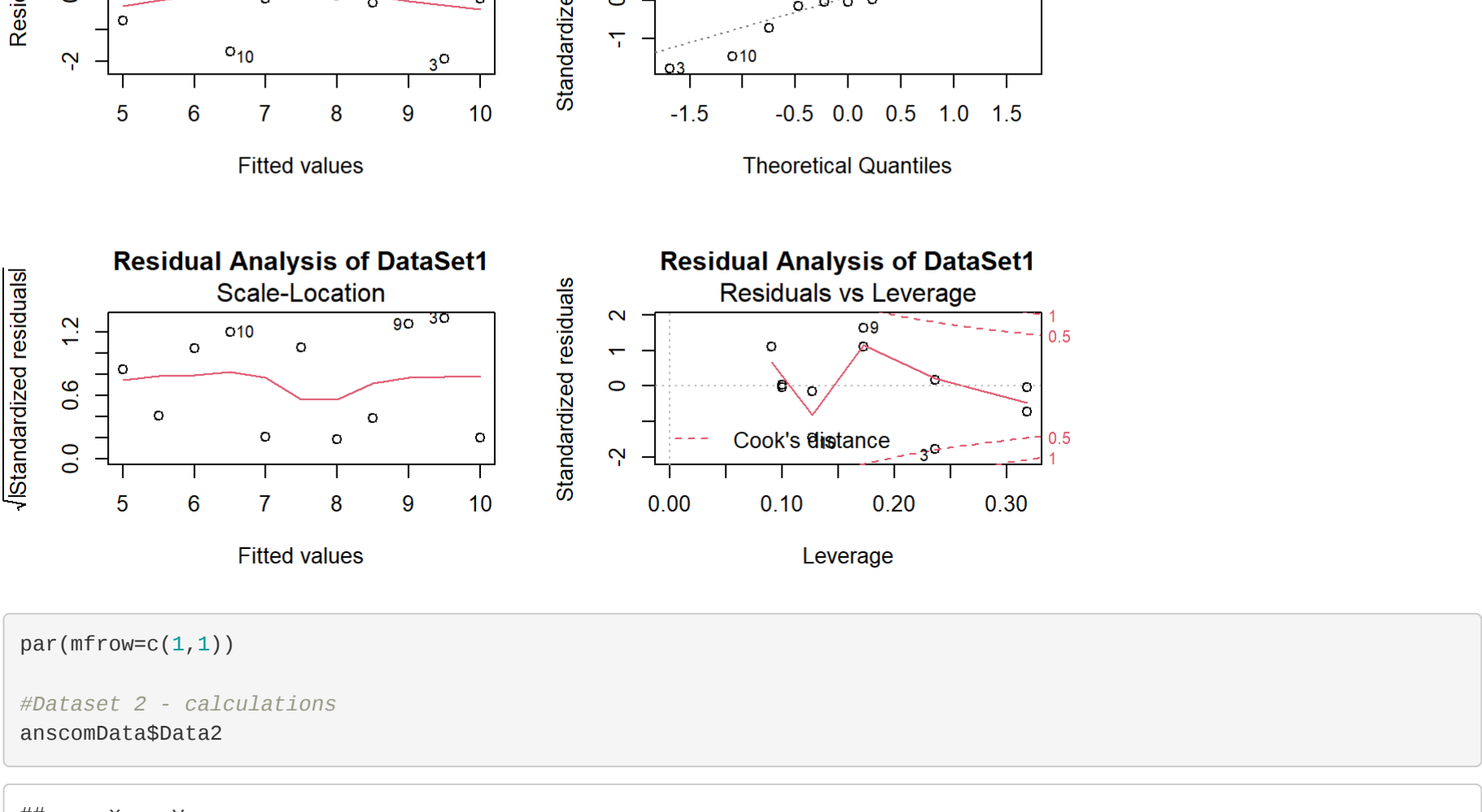
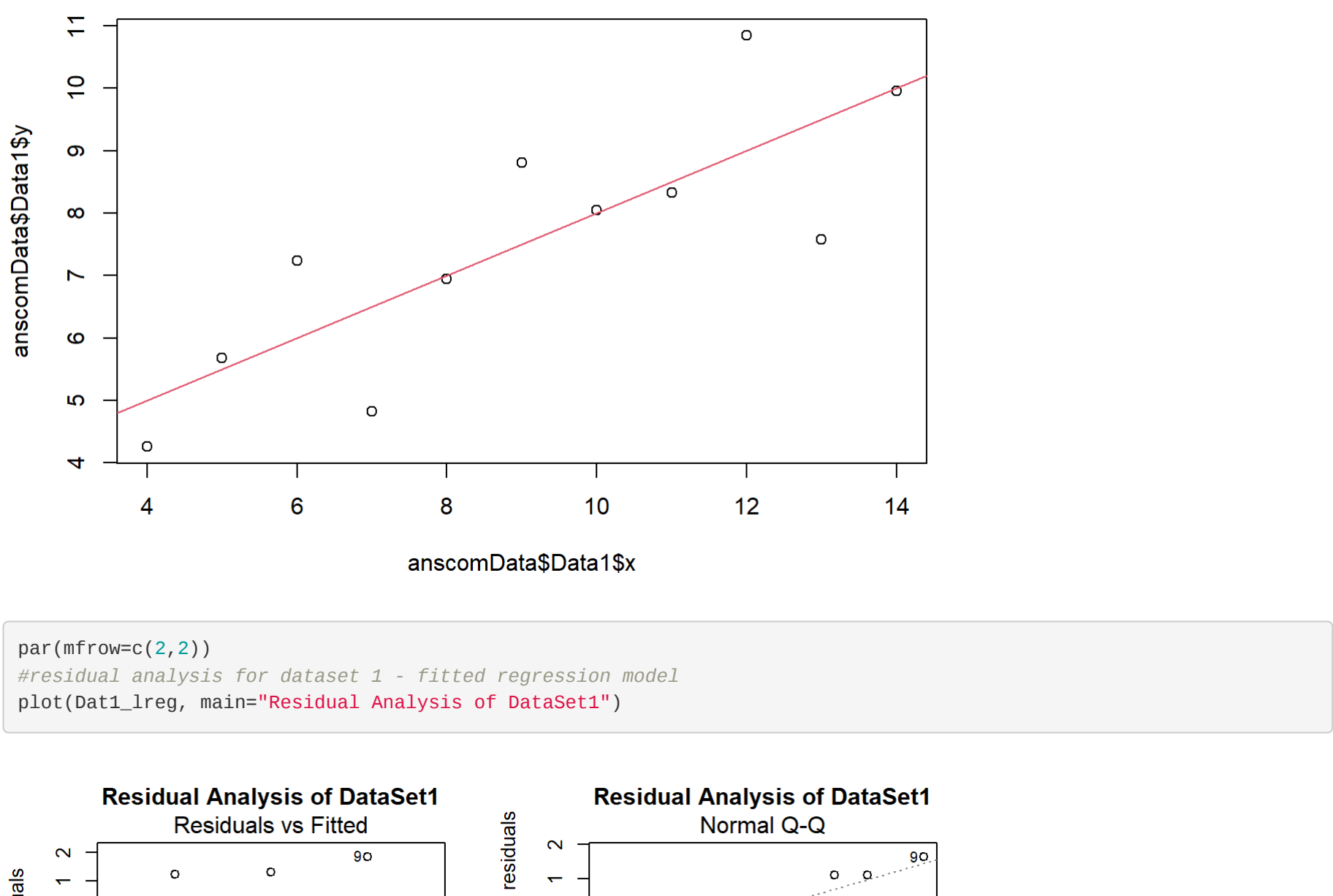
#correlation calculations
Data1_corr <- cor(anscombeData1$x, anscombeData1$y)
Data1_corr

## [1] 0.8162905

#creating linear regression model for dataset 1
Data1_lmreg <- lm(Data1$y~Data1$x, anscombeData1)
summary(Data1_lmreg)

##
## Call:
## lm(formula = Data1$y ~ Data1$x, data = anscombeData1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.02127 -0.45577 -0.04136  0.70941  1.33882
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.00001    1.12477   2.687  0.02073 *
## Data1$x      0.90505    0.11719   7.743  0.00017 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6868, Adjusted R-squared:  0.6205
## F-statistic: 17.99 on 1 and 9 DF, p-value: 0.00217

#visualizations for dataset 1
plot(anscombeData1$x, anscombeData1$y, main="Data1 - Fitted Linear Regression")
abline(Data1_lmreg, col="red")
```



```
par(mfrow=c(1,1))
#Dataset 2 - calculations
anscombeData2$Data2

##      x      y
## 1  10  9.14
## 2   8  8.14
## 3  13  8.74
## 4   9  8.77
## 5  11  9.26
## 6  14  8.10
## 7   7  6.13
## 8   4  3.10
## 9  12  9.13
## 10  7  7.26
## 11  5  4.74

#mean calculations
Data2_mn <- mean(anscombeData2$x)
Data2_mn

## [1] 9

Data2_mn <- mean(anscombeData2$y)
Data2_mn

## [1] 7.589899

#variance calculations
Data2_var <- var(anscombeData2$x)
Data2_var

## [1] 11

Data2_var <- var(anscombeData2$y)
Data2_var

## [1] 4.127289

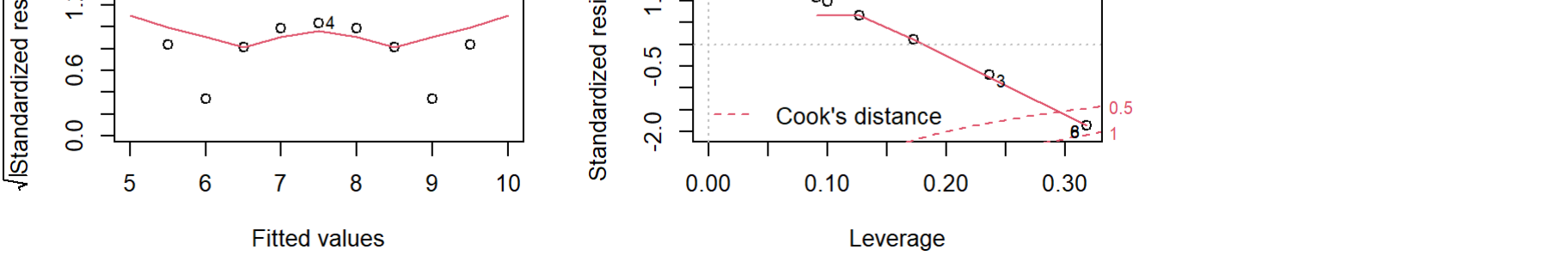
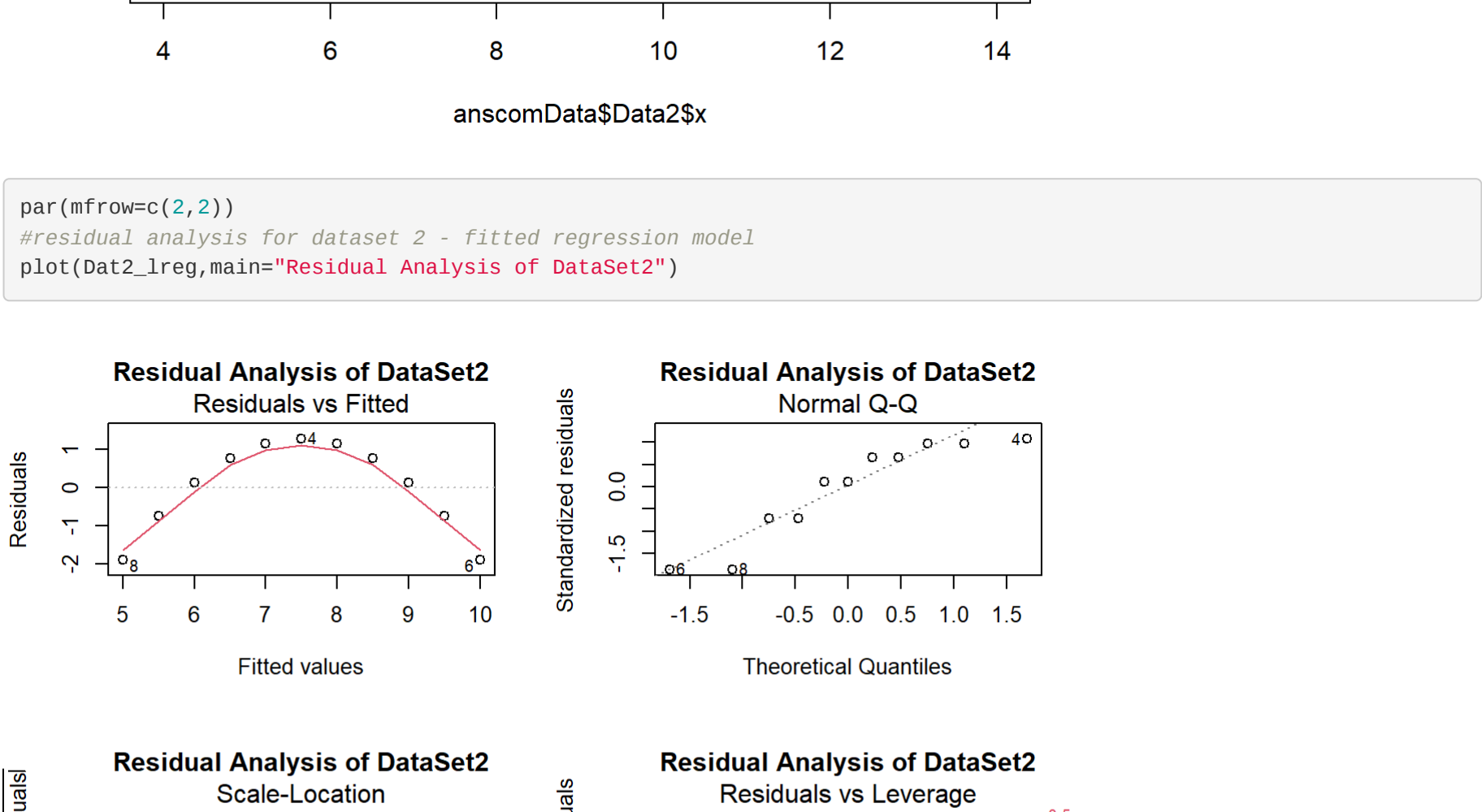
#correlation calculations
Data2_corr <- cor(anscombeData2$x, anscombeData2$y)
Data2_corr

## [1] 0.8162905

#creating linear regression model for dataset 2
Data2_lmreg <- lm(Data2$y~Data2$x, anscombeData2)
summary(Data2_lmreg)

##
## Call:
## lm(formula = Data2$y ~ Data2$x, data = anscombeData2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.00009 -0.76089  0.12081  0.94891  1.29931
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0001    1.125    2.687  0.02076 *
## Data2$x      0.9050    0.118    7.629  0.00018 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.237 on 9 degrees of freedom
## Multiple R-squared:  0.6868, Adjusted R-squared:  0.6202
## F-statistic: 17.97 on 1 and 9 DF, p-value: 0.002179

#visualizations for dataset 2
plot(anscombeData2$x, anscombeData2$y, main="Data2 - Fitted Linear Regression")
abline(Data2_lmreg, col="red")
```



```
par(mfrow=c(1,1))
#Dataset 3 - calculations
anscombeData3$Data3

##      x      y
## 1  10  7.46
## 2   8  6.77
## 3  13 12.74
## 4   9  7.11
## 5  11  7.81
## 6  14  8.84
## 7   7  8.47
## 8   4  5.39
## 9  12  8.15
## 10  7  6.42
## 11  5  5.73

#mean calculations
Data3_mn <- mean(anscombeData3$x)
Data3_mn

## [1] 9

Data3_mn <- mean(anscombeData3$y)
Data3_mn

## [1] 7.5

#variance calculations
Data3_var <- var(anscombeData3$x)
Data3_var

## [1] 11

Data3_var <- var(anscombeData3$y)
Data3_var

## [1] 4.127289

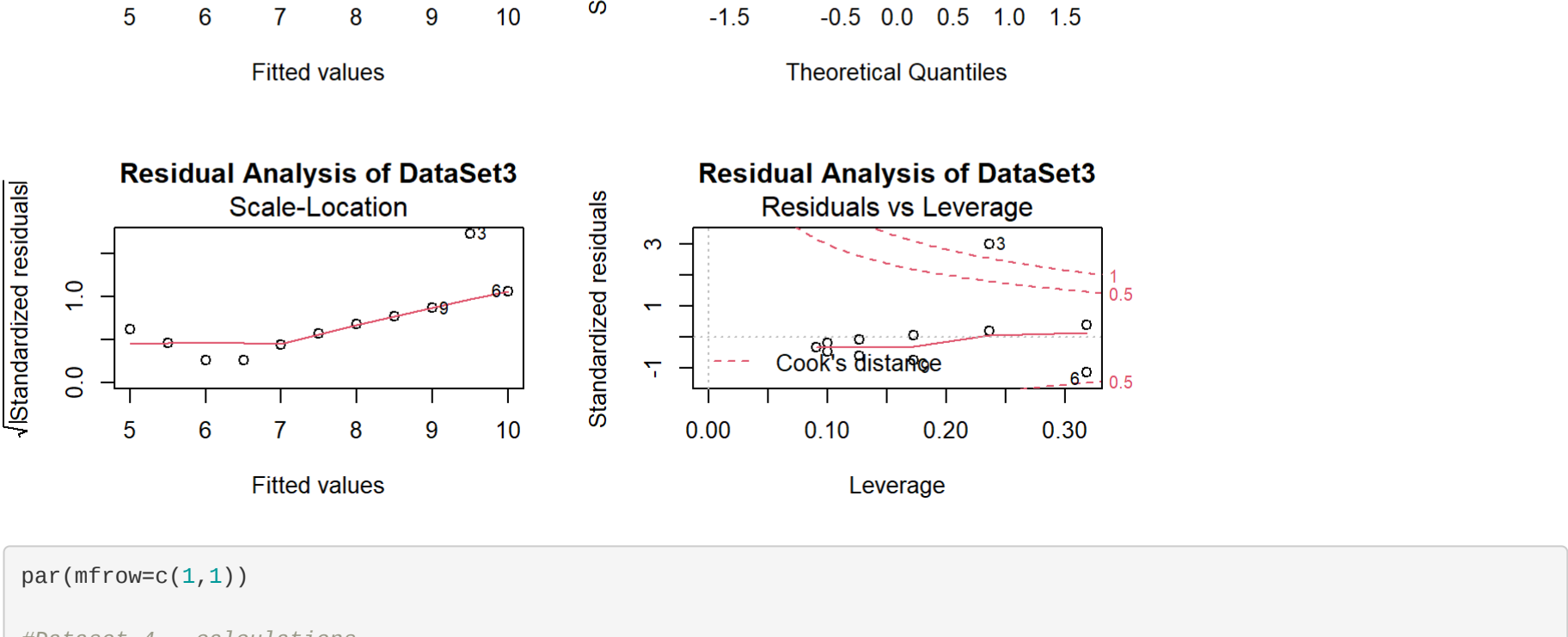
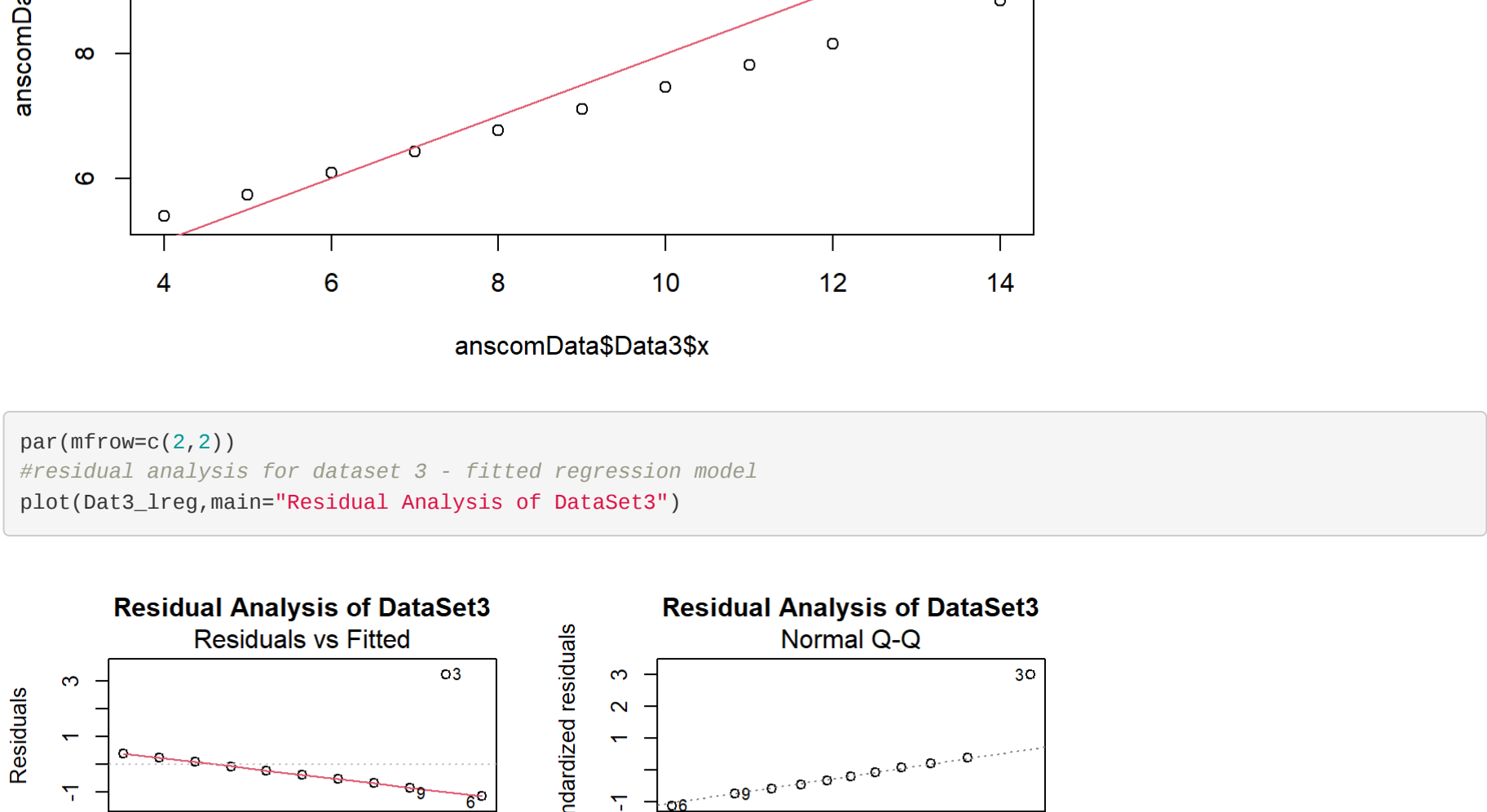
#correlation calculations
Data3_corr <- cor(anscombeData3$x, anscombeData3$y)
Data3_corr

## [1] 0.8162887

#creating linear regression model for dataset 3
Data3_lmreg <- lm(Data3$y~Data3$x, anscombeData3)
summary(Data3_lmreg)

##
## Call:
## lm(formula = Data3$y ~ Data3$x, data = anscombeData3)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.1586 -0.6146 -0.2383  0.1540  1.2411
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.0029    1.125    2.676  0.02082 *
## Data3$x      0.8997    0.1179    7.629  0.00018 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.236 on 9 degrees of freedom
## Multiple R-squared:  0.6868, Adjusted R-squared:  0.6202
## F-statistic: 17.97 on 1 and 9 DF, p-value: 0.002165

#visualizations for dataset 3
plot(anscombeData3$x, anscombeData3$y, main="Data3 - Fitted Linear Regression")
abline(Data3_lmreg, col="red")
```



```
par(mfrow=c(1,1))
#Dataset 4 - calculations
anscombeData4$Data4

##      x      y
## 1   8  6.58
## 2   8  5.76
## 3   8  7.71
## 4   8  8.84
## 5   8  8.47
## 6   8  7.84
## 7  10  8.25
## 8  10 12.50
## 9   8  5.98
## 10  8  7.91
## 11  8  6.89

#mean calculations
Data4_mn <- mean(anscombeData4$x)
Data4_mn

## [1] 8

Data4_mn <- mean(anscombeData4$y)
Data4_mn

## [1] 7.589899

#variance calculations
Data4_var <- var(anscombeData4$x)
Data4_var

## [1] 11

Data4_var <- var(anscombeData4$y)
Data4_var

## [1] 4.122249

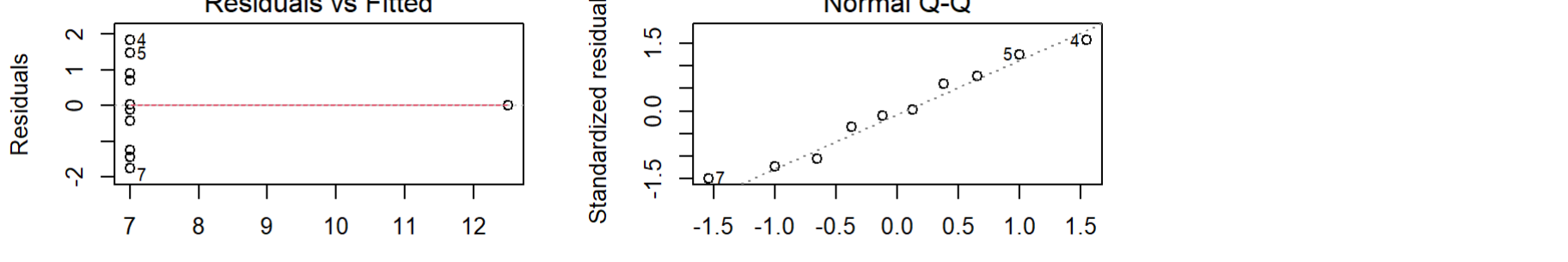
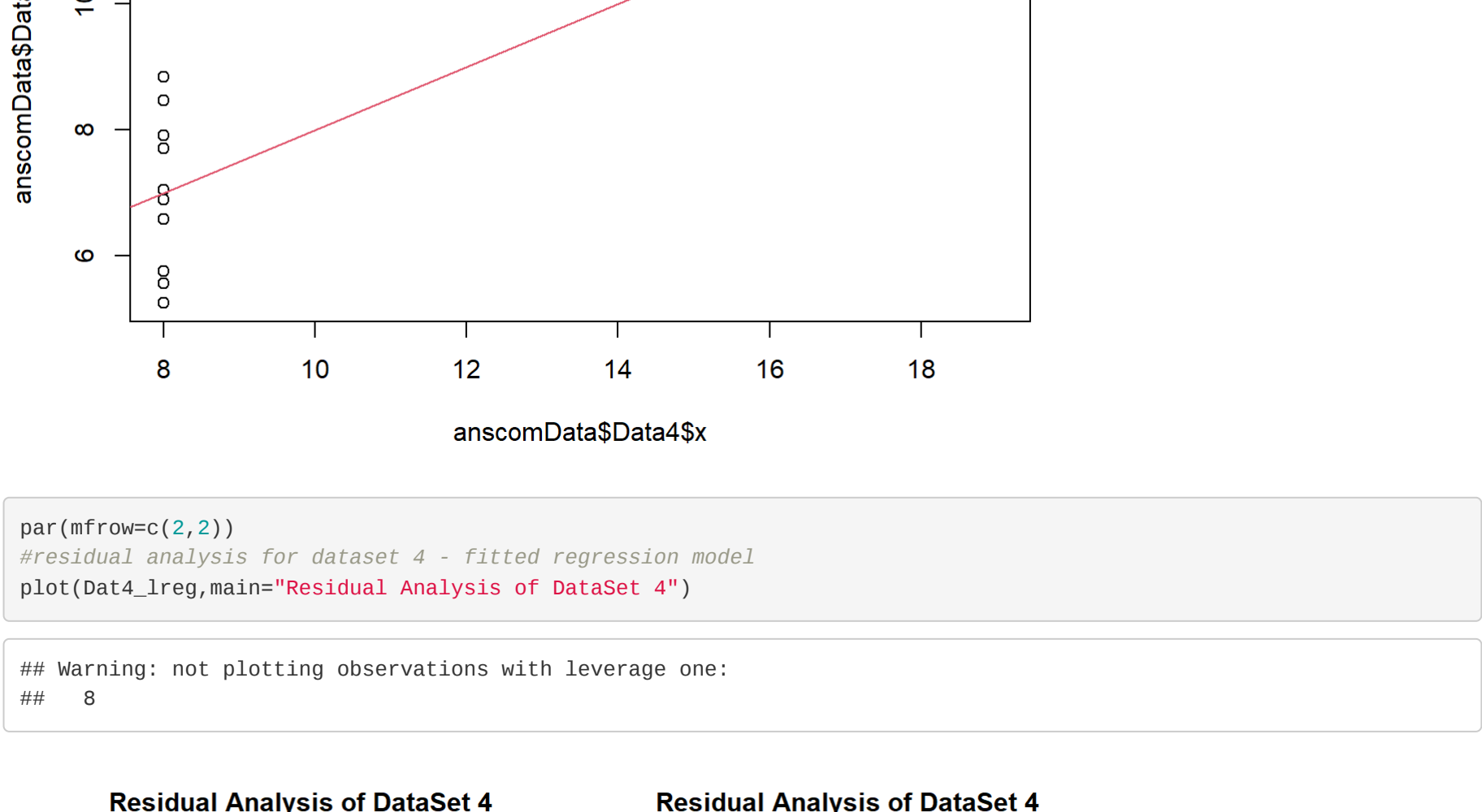
#correlation calculations
Data4_corr <- cor(anscombeData4$x, anscombeData4$y)
Data4_corr

## [1] 0.8162234

#creating linear regression model for dataset 4
Data4_lmreg <- lm(Data4$y~Data4$x, anscombeData4)
summary(Data4_lmreg)

##
## Call:
## lm(formula = Data4$y ~ Data4$x, data = anscombeData4)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.751 -1.031  0.008  0.809  1.839
##
## Coefficients:
##      Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.8671    1.129    3.421  0.00559 *
## Data4$x      0.4999    0.1178    4.243  0.00216 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.238 on 9 degrees of freedom
## Multiple R-squared:  0.6683, Adjusted R-squared:  0.6207
## F-statistic: 18 on 1 and 9 DF, p-value: 0.002165

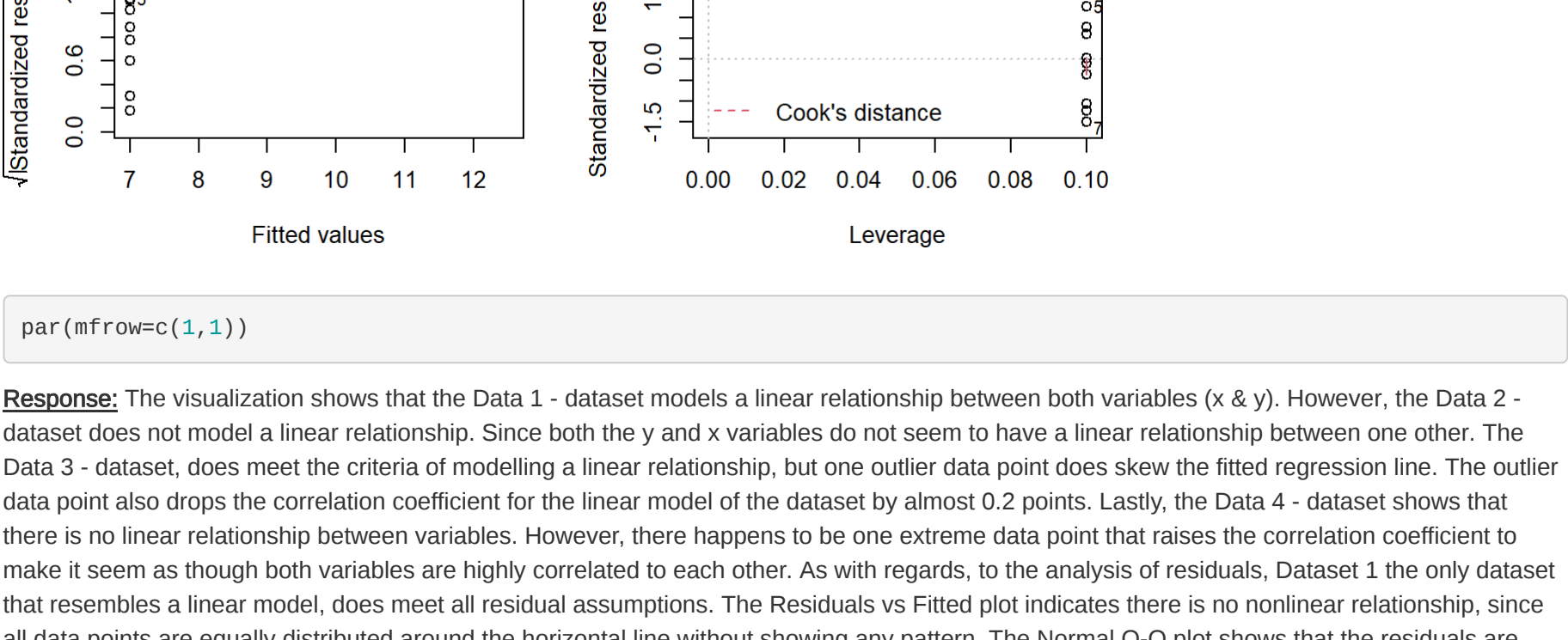
#visualizations for dataset 4
plot(anscombeData4$x, anscombeData4$y, main="Data4 - Fitted Linear Regression")
abline(Data4_lmreg, col="red")
```



```
par(mfrow=c(1,1))
#Dataset 5 - calculations
anscombeData5$Data5

##      x      y
## 1   8  6.58
## 2   8  5.76
## 3   8  7.71
## 4   8  8.84
## 5   8  8.47
## 6   8  7.84
## 7  10  8.25
## 8  10 12.50
## 9   8  5.98
## 10  8  7.91
## 11  8  6.89
```

Warning: not plotting observations with leverage one:



Question 2 (20 points) (25 points) How do you evaluate and compare the developed linear regression models in Question 1? There are any issues with the models you built?

One could evaluate the developed linear regression models created in question 1 through different types of statistical calculations expressed through the summary function output shown for each model. The main statistical calculation that most statisticians use to evaluate linear regression models is the R<sup>2</sup>-statistic and the adjusted R<sup>2</sup>-statistic. Since the R<sup>2</sup>-statistic shows "the amount of variance in output that the developed linear regression model explains." In our example, Dataset 4's, linear model R<sup>2</sup>-statistic value happens to be the highest out of all four models with a value of 0.6687. Since, each model that we created does not have more than one feature/variable in its respective model, we do not have to worry about the adjusted R<sup>2</sup>-statistic. Because the adjusted R<sup>2</sup>-statistic is used mainly to compare models who have more than one feature in them. Though, there is a problem with the interpretation of the R-squared statistic value for dataset 4. Normally, we would say that 66.87% of the variability observed in y is explained by the regression model. However, as previously noted dataset 4 has an issue with an extreme outlier data point. This outlier data point happens to be raising the correlation coefficient to make it seem as though both variables are highly correlated to each other. Which as a result, happens to be raising the R-squared statistic of the model. To make the model, fairly more significant than the other three models. Consequently, out of the only two remaining datasets that model a linear relationship, dataset 1 & dataset 3, dataset 1 should be considered the model with the highest variability explained in its regression model with a value of 65.69%.

(25 points) Is there any way to improve your linear regression models for Data1, 4, and 6?

Yes, there are ways to improve our linear regression models for the three datasets(1,4,6). One way we could help improve our linear regression model R-squared scores would be to remove outlier data points from the dataset. However, this method is not suggested when your dataset is based on a small sample size. Another way we could improve our linear regression models without decreasing the size of our current dataset, would be to floor capping outlier data values. This concept involves determining a percentile to cap data values at and changing all values that are above this percentile value to this percentile value. For example, we could establish the 75th percentile to be our highest percentile for our dataset. Thus, changing any data points above this percentile value to the same value as the 75th percentile value. Another way we could handle outlier data points to improve our linear regression models would be to change all outlier data points to our established median value of the dataset.