DATA 606 Fall 2019 - Final Exam

Raghed Mirza

Part I

Please put the answers for Part I next to the question number (2pts each):

1. b (gasMonth is also quantitative but would be continuous)
2. a (left skewed with mean should be smaller than median and a reasonable median should be around 3.5 from the amount of observations)
3. d (prospective and retrospective)
4. a (large x2 will mean that the null hypothesis is rejected and there is a difference between eye color and hair color)
5. b (please refer to the calculation below)

```
q1 <- 37; q3 <- 49.8; iqr_monkeys <- q3 - q1

up_limit <- q3 + 1.5 * iqr_monkeys
print(up_limit, digits = 4)
```

```
## [1] 69
```

```
low_limit <- q1 - 1.5 * iqr_monkeys
print(low_limit, digits = 4)
```

```
## [1] 17.8
```

6. d

7a. Describe the two distributions (2pts).

Both of the distributions are unimodal. Distribution A is right-skewed and distribution B seems to be normally distributed with almost no skewness.

7b. Explain why the means of these two distributions are similar but the standard deviations are not (2 pts).

The means are similar because distribution B is a sample of A. The difference in standard deviations is due to the sample sizes. The standard deviation will decrease for distribution A as its sample size will grow.

7c. What is the statistical principal that describes this phenomenon (2 pts)?

The statistical principal that explains this phenomenon is called central limit theorem (CLT). The conditions are that the samples in the distributions must be independent and random, not strongly skewed, and the distributions should be normal.

Part II

Consider the four datasets, each with two columns (x and y), provided below. Be sure to replace the `NA` with your answer for each part (e.g. assign the mean of `x` for `data1` to the `data1.x.mean` variable). When you Knit your answer document, a table will be generated with all the answers.

```
options(digits=2)
data1 <- data.frame(x=c(10,8,13,9,11,14,6,4,12,7,5),
                    y=c(8.04,6.95,7.58,8.81,8.33,9.96,7.24,4.26,10.84,4.82,5.68))
data2 <- data.frame(x=c(10,8,13,9,11,14,6,4,12,7,5),
                    y=c(9.14,8.14,8.74,8.77,9.26,8.1,6.13,3.1,9.13,7.26,4.74))
data3 <- data.frame(x=c(10,8,13,9,11,14,6,4,12,7,5),
                    y=c(7.46,6.77,12.74,7.11,7.81,8.84,6.08,5.39,8.15,6.42,5.73))
data4 <- data.frame(x=c(8,8,8,8,8,8,8,19,8,8,8),
                    y=c(6.58,5.76,7.71,8.84,8.47,7.04,5.25,12.5,5.56,7.91,6.89))
```

For each column, calculate (to two decimal places):

a. The mean (for x and y separately; 1 pt).

```
data1.x.mean <- mean(data1$x)
data1.y.mean <- mean(data1$y)
data2.x.mean <- mean(data2$x)
data2.y.mean <- mean(data2$y)
data3.x.mean <- mean(data3$x)
data3.y.mean <- mean(data3$y)
data4.x.mean <- mean(data4$x)
data4.y.mean <- mean(data4$y)
data1.x.mean
```

```
## [1] 9
```

```
data1.y.mean
```

```
## [1] 7.5
```

```
data2.x.mean
```

```
## [1] 9
```

```
data2.y.mean
```

```
## [1] 7.5
```

```
data3.x.mean
```

```
## [1] 9
```

```r
data3.y.mean
```

```
## [1] 7.5
```

```r
data4.x.mean
```

```
## [1] 9
```

```r
data4.y.mean
```

```
## [1] 7.5
```

b. The median (for x and y separately; 1 pt).

```r
data1.x.median <- median(data1$x)
data1.y.median <- median(data1$y)
data2.x.median <- median(data2$x)
data2.y.median <- median(data2$y)
data3.x.median <- median(data3$x)
data3.y.median <- median(data3$y)
data4.x.median <- median(data4$x)
data4.y.median <- median(data4$y)

data1.x.median
```

```
## [1] 9
```

```r
data1.y.median
```

```
## [1] 7.6
```

```r
data2.x.median
```

```
## [1] 9
```

```r
data2.y.median
```

```
## [1] 8.1
```

```r
data3.x.median
```

```
## [1] 9
```

```r
data3.y.median
```

```
## [1] 7.1
```

```
data4.x.median
```

```
## [1] 8
```

```
data4.y.median
```

```
## [1] 7
```

c. The standard deviation (for x and y separately; 1 pt).

```
data1.x.sd <- sd(data1$x)
data1.y.sd <- sd(data1$y)
data2.x.sd <- sd(data2$x)
data2.y.sd <- sd(data2$y)
data3.x.sd <- sd(data3$x)
data3.y.sd <- sd(data3$y)
data4.x.sd <- sd(data4$x)
data4.y.sd <- sd(data4$y)

data1.x.sd
```

```
## [1] 3.3
```

```
data1.y.sd
```

```
## [1] 2
```

```
data2.x.sd
```

```
## [1] 3.3
```

```
data2.y.sd
```

```
## [1] 2
```

```
data3.x.sd
```

```
## [1] 3.3
```

```
data3.y.sd
```

```
## [1] 2
```

```
data4.x.sd
```

```
## [1] 3.3
```

```
data4.y.sd
```

```
## [1] 2
```

For each x and y pair, calculate (also to two decimal places; 1 pt):

d. The correlation (1 pt).

```
data1.correlation <- cor(data1)[1,2]
data2.correlation <- cor(data2)[1,2]
data3.correlation <- cor(data3)[1,2]
data4.correlation <- cor(data4)[1,2]

print(data1.correlation, digits = 2)
```

```
## [1] 0.82
```

```
print(data2.correlation, digits = 2)
```

```
## [1] 0.82
```

```
print(data3.correlation, digits = 2)
```

```
## [1] 0.82
```

```
print(data4.correlation, digits = 2)
```

```
## [1] 0.82
```

e. Linear regression equation (2 pts).

```
lm1 <- lm(y ~ x, data = data1)
lm2 <- lm(y ~ x, data = data2)
lm3 <- lm(y ~ x, data = data3)
lm4 <- lm(y ~ x, data = data4)
summary(lm1)
```

```
##
## Call:
## lm(formula = y ~ x, data = data1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9213 -0.4558 -0.0414  0.7094  1.8388
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.000      1.125    2.67   0.0257 *
## x              0.500      0.118    4.24   0.0022 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.2 on 9 degrees of freedom
## Multiple R-squared:  0.667,  Adjusted R-squared:  0.629
## F-statistic:   18 on 1 and 9 DF,  p-value: 0.00217
```

```
summary(lm2)
```

```
##
## Call:
## lm(formula = y ~ x, data = data2)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.901 -0.761  0.129  0.949  1.269
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.001      1.125    2.67   0.0258 *
## x              0.500      0.118    4.24   0.0022 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.2 on 9 degrees of freedom
## Multiple R-squared:  0.666,  Adjusted R-squared:  0.629
## F-statistic:   18 on 1 and 9 DF,  p-value: 0.00218
```

```
summary(lm3)
```

```
##
## Call:
## lm(formula = y ~ x, data = data3)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.159 -0.615 -0.230  0.154  3.241
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.002      1.124    2.67   0.0256 *
## x              0.500      0.118    4.24   0.0022 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.2 on 9 degrees of freedom
## Multiple R-squared:  0.666,  Adjusted R-squared:  0.629
## F-statistic:   18 on 1 and 9 DF,  p-value: 0.00218
```

```r
summary(lm4)
```

```
##
## Call:
## lm(formula = y ~ x, data = data4)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -1.751 -0.831  0.000  0.809  1.839
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)    3.002      1.124    2.67   0.0256 *
## x              0.500      0.118    4.24   0.0022 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.2 on 9 degrees of freedom
## Multiple R-squared:  0.667,  Adjusted R-squared:  0.63
## F-statistic:   18 on 1 and 9 DF,  p-value: 0.00216
```

```r
data1.slope <- coef(lm1)["x"]
data2.slope <- coef(lm2)["x"]
data3.slope <- coef(lm3)["x"]
data4.slope <- coef(lm4)["x"]

data1.intercept <- coef(lm1)["(Intercept)"]
data2.intercept <- coef(lm2)["(Intercept)"]
data3.intercept <- coef(lm3)["(Intercept)"]
data4.intercept <- coef(lm4)["(Intercept)"]

print(data1.slope, digits = 5)
```

```
##       x
## 0.50009
```

```r
print(data2.slope, digits = 5)
```

```
##   x
## 0.5
```

```r
print(data3.slope, digits = 5)
```

```
##       x
## 0.49973
```

```r
print(data4.slope, digits = 5)
```

```
##       x
## 0.49991
```

```
print(data1.intercept, digits = 5)
```

```
## (Intercept)
##      3.0001
```

```
print(data2.intercept, digits = 5)
```

```
## (Intercept)
##      3.0009
```

```
print(data3.intercept, digits = 5)
```

```
## (Intercept)
##      3.0025
```

```
print(data4.intercept, digits = 5)
```

```
## (Intercept)
##      3.0017
```

f. R-Squared (2 pts).

```
data1.rsquared <- summary(lm1)$r.squared
data2.rsquared <- summary(lm2)$r.squared
data3.rsquared <- summary(lm3)$r.squared
data4.rsquared <- summary(lm4)$r.squared

print(data1.rsquared, digits = 5)
```

```
## [1] 0.66654
```

```
print(data2.rsquared, digits = 5)
```

```
## [1] 0.66624
```

```
print(data3.rsquared, digits = 5)
```

```
## [1] 0.66632
```

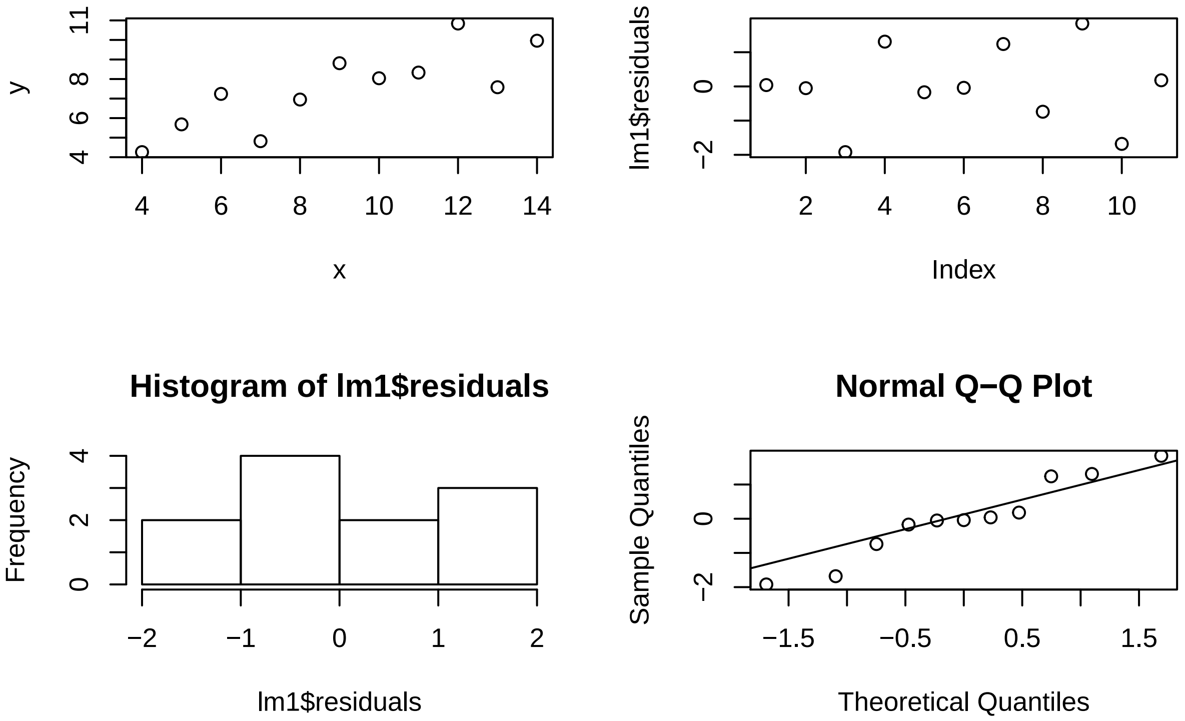
```
print(data4.rsquared, digits = 5)
```

```
## [1] 0.66671
```

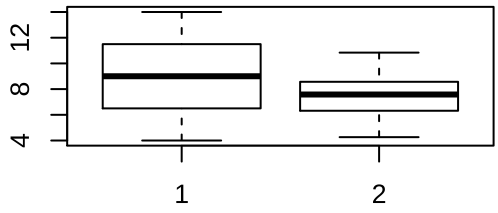
	Data 1		Data 2		Data 3		Data 4	
	x	y	x	y	x	y	x	y
Mean	9.00	7.50	9.00	7.50	9.00	7.50	9.00	7.50
Median	9.00	7.58	9.00	8.14	9.00	7.11	8.00	7.04
SD	3.32	2.03	3.32	2.03	3.32	2.03	3.32	2.03
r	0.82		0.82		0.82		0.82	
Intercept	3.00		3.00		3.00		3.00	
Slope	0.50		0.50		0.50		0.50	
R-Squared	0.67		0.67		0.67		0.67	

g. For each pair, is it appropriate to estimate a linear regression model? Why or why not? Be specific as to why for each pair and include appropriate plots! (4 pts)

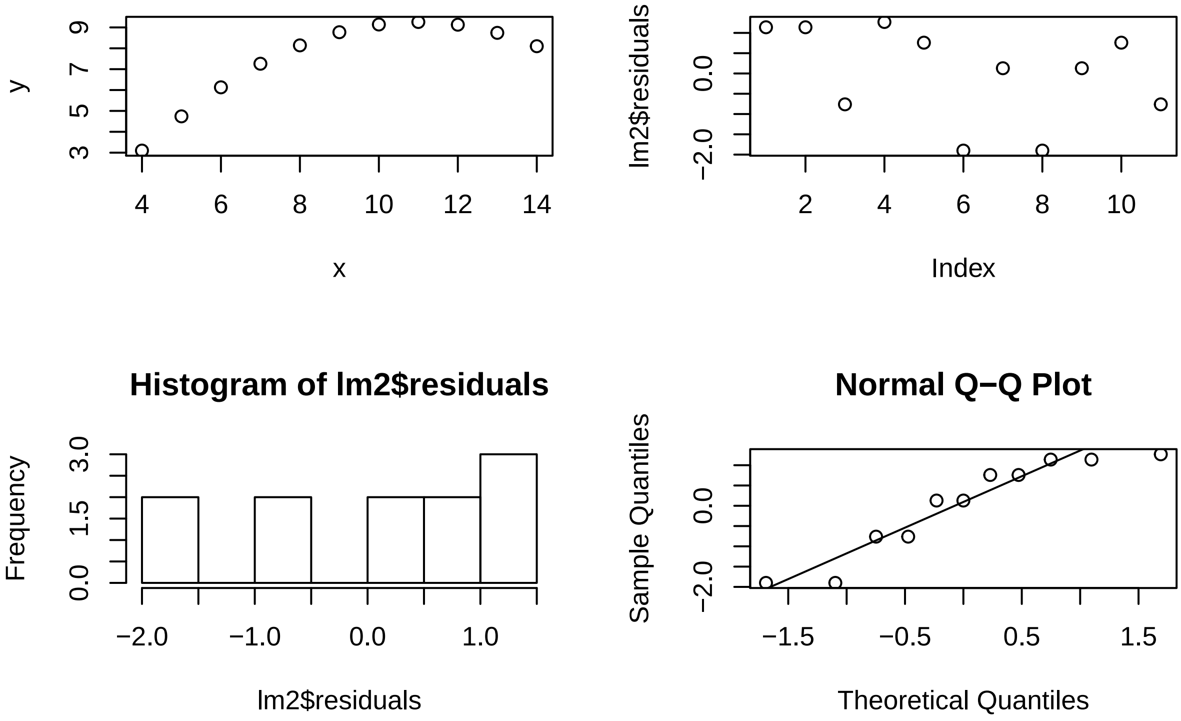
```r
par(mfrow=c(2,2))
plot(data1)
plot(lm1$residuals)
hist(lm1$residuals)
qqnorm(lm1$residuals)
qqline(lm1$residuals)
```



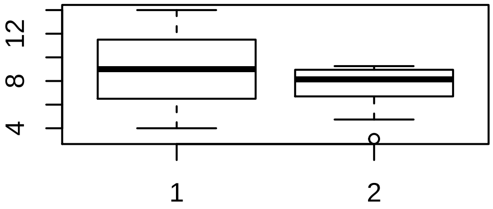
```r
boxplot(data1$x,data1$y)
# For data1, the plots seem to depict linearity. Looking at the histogram, the
# residuals seem to be randomly distributed but the boxplot is not showing any
# outliers. A linear regression model seems plausible.
```



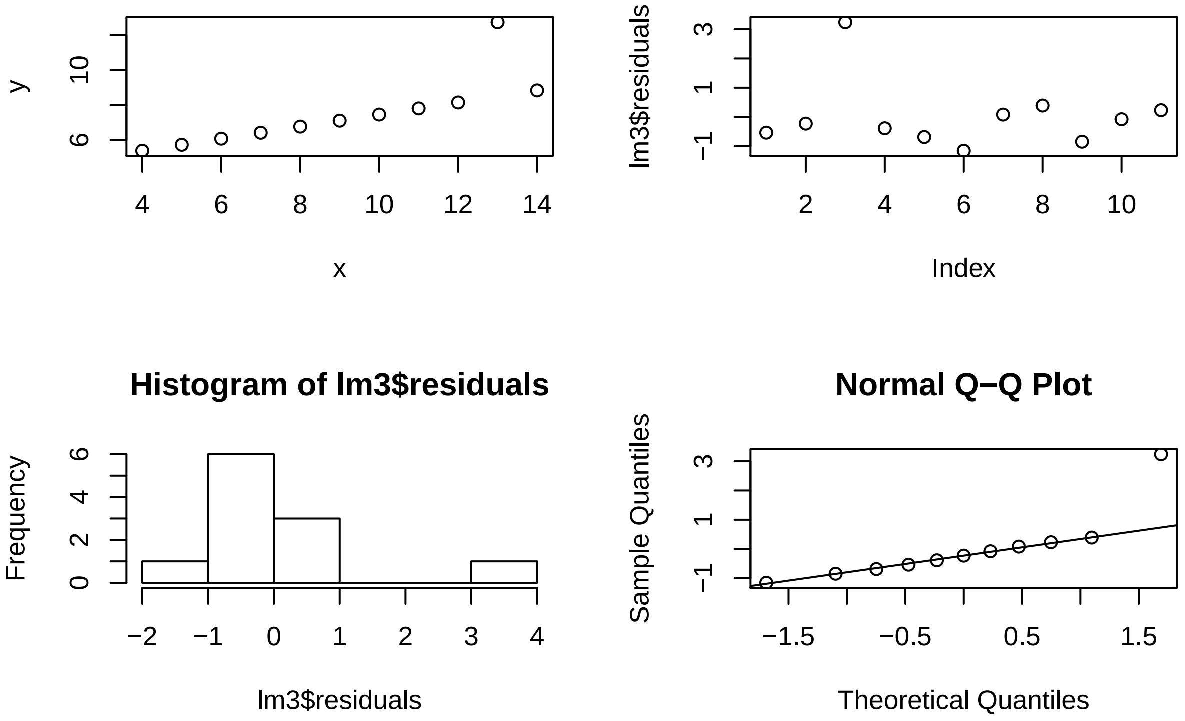
```r
par(mfrow=c(2,2))
plot(data2)
plot(lm2$residuals)
hist(lm2$residuals)
qqnorm(lm2$residuals)
qqline(lm2$residuals)
```



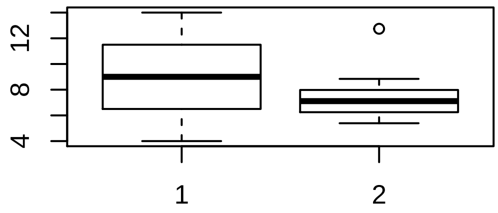
```
boxplot(data2$x,data2$y)
# For data2, it seems that a linear regression model is not possible. The boxblot is
# predicting an outlier and the historgram shows a random distribution for the
# residuals.
```



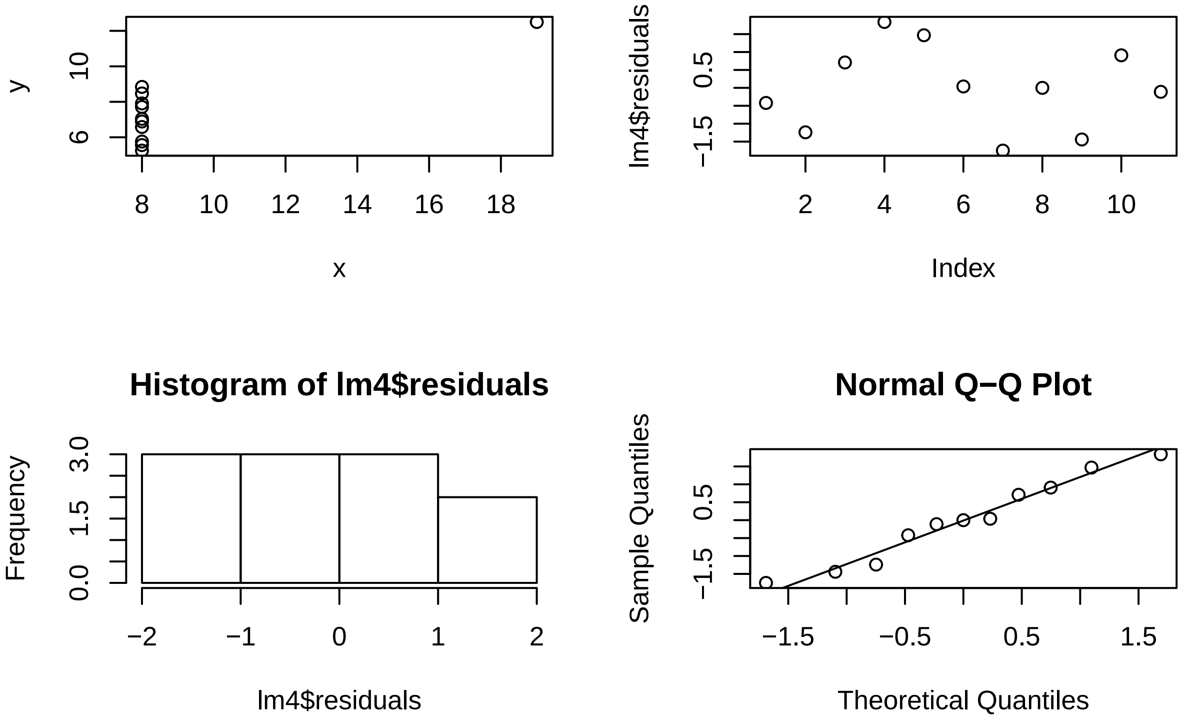
```
par(mfrow=c(2,2))
plot(data3)
plot(lm3$residuals)
hist(lm3$residuals)
qqnorm(lm3$residuals)
qqline(lm3$residuals)
```



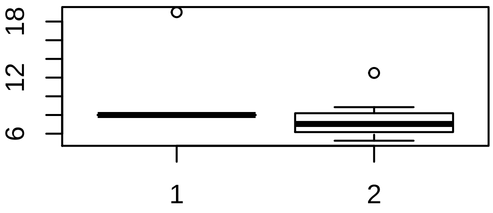
```
boxplot(data3$x,data3$y)
# For data3, a linear model might be possible once the impact of removing the outlier
# is carefully determined.
```



```
par(mfrow=c(2,2))
plot(data4)
plot(lm4$residuals)
hist(lm4$residuals)
qqnorm(lm4$residuals)
qqline(lm4$residuals)
```



```r
boxplot(data4$x,data4$y)
# For data4, there are two extreme outliers skewing the results. A linear regression model
# does not seem to be the right choice here.
```



h. Explain why it is important to include appropriate visualizations when analyzing data. Include any visualization(s) you create. (2 pts)

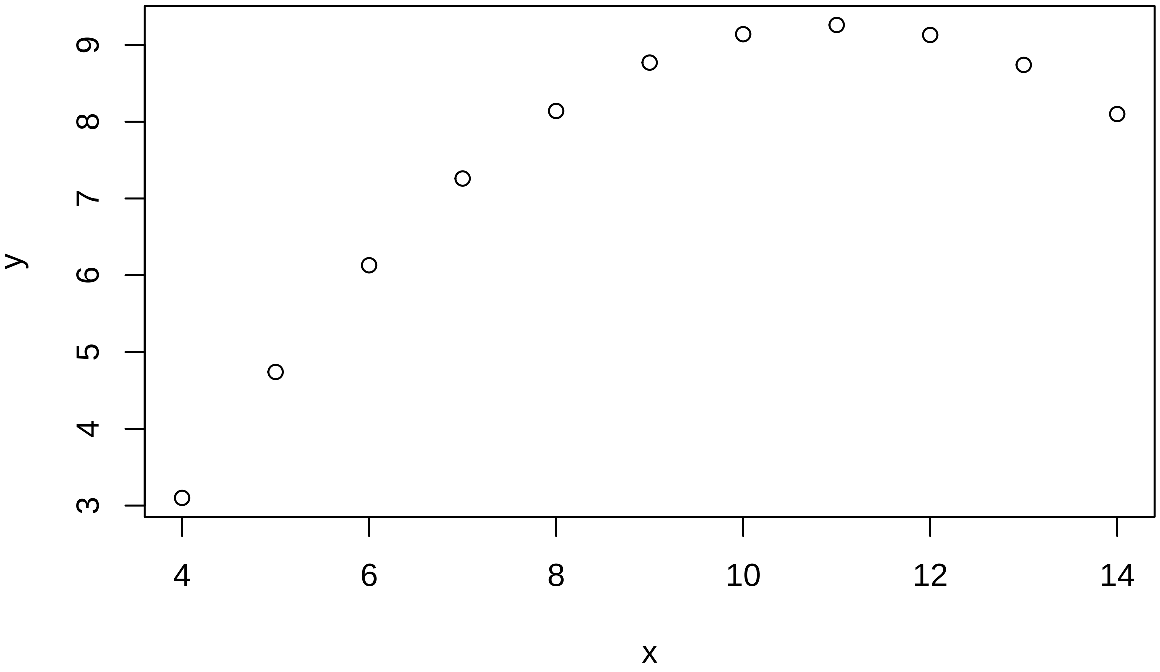
It is much easier to depict a pattern or trend by looking at the data through visualizations. It is often said that a picture is worth a thousand words, and this phrase seems to fit best to the comparison between a tabular form of data versus a visualization (like a graph plot). As an example, the data from data2 is produced as a table below:

```r
head(data2, n = 11)
```

```
##     x   y
## 1  10 9.1
## 2   8 8.1
## 3  13 8.7
## 4   9 8.8
## 5  11 9.3
## 6  14 8.1
## 7   6 6.1
## 8   4 3.1
## 9  12 9.1
## 10  7 7.3
## 11  5 4.7
```

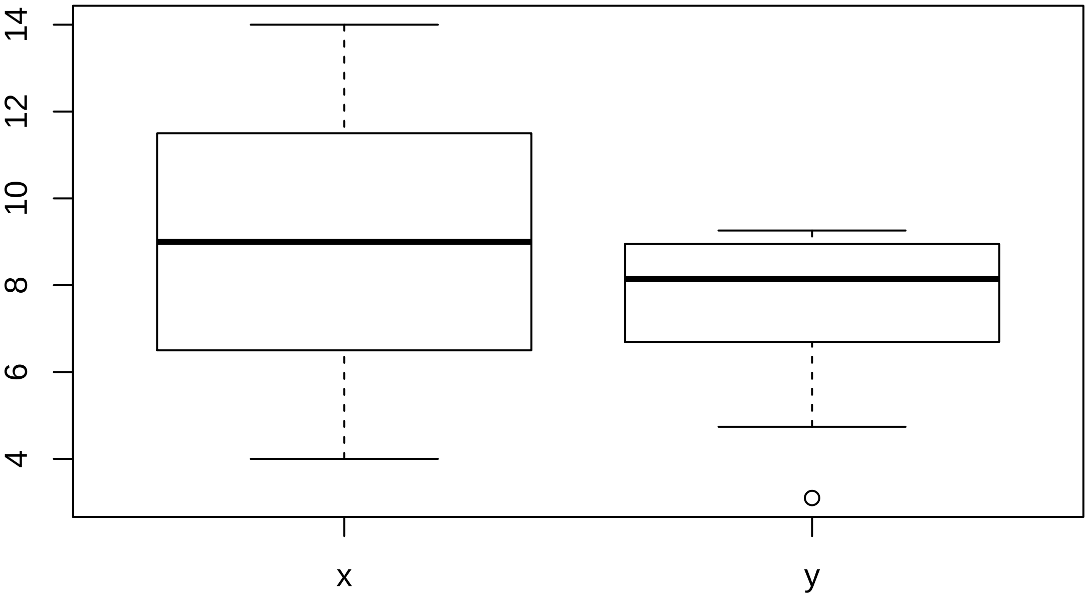
However, when the same data is put in a graph, it is much easier to see the pattern/trend and any possible outliers:

```
plot(data2)
```



To further illustrate the power of visualization, let us look at the boxplot below:

```
boxplot(data2)
```



It is evident from the above boxplot, that there is a potential outlier (which could be authenticated by further testing). This was not that clear from the regular plot, and certainly, not at all obvious from the table!