DATA 606 Fall 2018 - Final Exam

Jimmy Ng

```
## -- Attaching packages ---------------------------------------------------
--------- tidyverse 1.2.1 --

## v ggplot2 3.0.0     v purrr   0.2.5
## v tibble  1.4.2     v dplyr   0.7.6
## v tidyr   0.8.1     v stringr 1.3.1
## v readr   1.1.1     v forcats 0.3.0

## -- Conflicts ------------------------------------------------------------
--- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()

##
## Attaching package: 'wrapr'

## The following object is masked from 'package:dplyr':
##
##     coalesce

##
## Attaching package: 'rlang'

## The following object is masked from 'package:wrapr':
##
##     :=

## The following objects are masked from 'package:purrr':
##
##     %@%, %||%, as_function, flatten, flatten_chr, flatten_dbl,
##     flatten_int, flatten_lgl, invoke, list_along, modify, prepend,
##     rep_along, splice
```

Part I

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

1. a
2. a
3. a
4. c
5. b
6. d

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

Distribution A is skewed heavily to the right. The median is greater than the mean because of the right skew. Distribution B is a sampling distribution of A. It is composed by 500 random samples of size 30 each from A. Therefore, we can use it to estimate the properties of A, e.g. population mean. Distribution B is close to a normal distribution with peak in center and equal spread on both sides.

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 just a sampling distribution of distribution A. With large enough independent, random samples (in this case, 500) and large sample size of each (in this case, 30), we expect to see a normal distribution of the sampling distribution regardless of the underlying distribution of the observation, i.e. Central Limit Theorem. The means are therefore similar and we can use it (the sampling mean from distribution B) to estimate the mean for the population. On the other hand, the SD are different because the SD of distribution B is referred to the SD of the sampling distribution (not the observation). This is the standard error of the mean, not the standard distribution of the observations seen in distribution A. The standard error is used to estimate for the spread for the population mean. In this case, we can esimate the population mean is approximately 5.04 plus/minus 1.95 * 0.58 with 95% confidence.

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

As described above, this is the Central Limit Theorem. There are two conditions that must be met: (1) samples are independent and radom; and (2) large enough size, e.g. 30, in each sample.

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).

```
# set up list
dataList <- list(d1x = data1$x, d1y = data1$y,
                 d2x = data2$x, d2y = data2$y,
                 d3x = data3$x, d3y = data3$y,
                 d4x = data4$x, d4y = data4$y)

# set up function
calculation <- function(x, y, z) {
        if(!require(purrr)){install.packages("purrr"); require(purrr)}
        if(!require(wrapr)){install.packages("wrapr"); require(wrapr)}

        for(i in 1:length(y)){
                purrr::map(c(x),
                    function(x) {
                            round(x(y[[i]]), 2) %.>%
                                    assign(names(z)[i], ., envir =
.GlobalEnv)
                    }
                )
        }

}

# set up names for global environment
variables.mean <- list(data1.x.mean = NA, data1.y.mean = NA,
                data2.x.mean = NA, data2.y.mean = NA,
```

```
               data3.x.mean = NA, data3.y.mean = NA,
               data4.x.mean = NA, data4.y.mean = NA)

# run the function, assign result to each variable name back to global
environment
calculation(mean,
            dataList,
            variables.mean)

# print results
for(i in 1:length(variables.mean)){
        if(!require(rlang)){install.packages("rlang"); require(rlang)}
        print(paste0(names(variables.mean)[i],
                     " is equal to ",
                     eval( rlang::sym(names(variables.mean[i])) )
                     ))
}
```

```
## [1] "data1.x.mean is equal to 9"
## [1] "data1.y.mean is equal to 7.5"
## [1] "data2.x.mean is equal to 9"
## [1] "data2.y.mean is equal to 7.5"
## [1] "data3.x.mean is equal to 9"
## [1] "data3.y.mean is equal to 7.5"
## [1] "data4.x.mean is equal to 9"
## [1] "data4.y.mean is equal to 7.5"
```

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

```
# set up names for global environment
variables.median <- list(data1.x.median = NA, data1.y.median = NA,
               data2.x.median = NA, data2.y.median = NA,
               data3.x.median = NA, data3.y.median = NA,
               data4.x.median = NA, data4.y.median = NA)

# run the function, assign result to each variable name back to global
environment
calculation(median,
            dataList,
            variables.median)

# print results
for(i in 1:length(variables.median)){
        if(!require(rlang)){install.packages("rlang"); require(rlang)}
        print(paste0(names(variables.median)[i],
                     " is equal to ",
                     eval( rlang::sym(names(variables.median[i])) )
                     ))
}
```

```
## [1] "data1.x.median is equal to 9"
## [1] "data1.y.median is equal to 7.58"
## [1] "data2.x.median is equal to 9"
## [1] "data2.y.median is equal to 8.14"
## [1] "data3.x.median is equal to 9"
## [1] "data3.y.median is equal to 7.11"
## [1] "data4.x.median is equal to 8"
## [1] "data4.y.median is equal to 7.04"
```

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

```
# set up names for global environment
variables.sd <- list(data1.x.sd = NA, data1.y.sd = NA,
                     data2.x.sd = NA, data2.y.sd = NA,
                     data3.x.sd = NA, data3.y.sd = NA,
                     data4.x.sd = NA, data4.y.sd = NA)

# run the function, assign result to each variable name back to global
environment
calculation(sd,
            dataList,
            variables.sd)

# print results
for(i in 1:length(variables.sd)){
        if(!require(rlang)){install.packages("rlang"); require(rlang)}
        print(paste0(names(variables.sd)[i],
                     " is equal to ",
                     eval( rlang::sym(names(variables.sd[i])) )
                     ))
}
```

```
## [1] "data1.x.sd is equal to 3.32"
## [1] "data1.y.sd is equal to 2.03"
## [1] "data2.x.sd is equal to 3.32"
## [1] "data2.y.sd is equal to 2.03"
## [1] "data3.x.sd is equal to 3.32"
## [1] "data3.y.sd is equal to 2.03"
## [1] "data4.x.sd is equal to 3.32"
## [1] "data4.y.sd is equal to 2.03"
```

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

d. The correlation (1 pt).

```
data1.correlation <- round(cor(data1$x, data1$y), 2)
data2.correlation <- round(cor(data2$x, data2$y), 2)
data3.correlation <- round(cor(data3$x, data3$y), 2)
data4.correlation <- round(cor(data4$x, data4$y), 2)

# create a list for the data.frames
dfList <- list(df1 = data1,
```

```
               df2 = data2,
               df3 = data3,
               df4 = data4)

# print detail cor.test result
lapply(dfList, function(x) cor.test(x$x, x$y))
```

```
## $df1
##
##  Pearson's product-moment correlation
##
## data:  x$x and x$y
## t = 4, df = 9, p-value = 0.002
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.42 0.95
## sample estimates:
##  cor
## 0.82
##
##
## $df2
##
##  Pearson's product-moment correlation
##
## data:  x$x and x$y
## t = 4, df = 9, p-value = 0.002
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.42 0.95
## sample estimates:
##  cor
## 0.82
##
##
## $df3
##
##  Pearson's product-moment correlation
##
## data:  x$x and x$y
## t = 4, df = 9, p-value = 0.002
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.42 0.95
## sample estimates:
##  cor
## 0.82
##
##
## $df4
```

```
##
##  Pearson's product-moment correlation
##
## data:  x$x and x$y
## t = 4, df = 9, p-value = 0.002
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.42 0.95
## sample estimates:
##  cor
## 0.82
```

e. Linear regression equation (2 pts).

```
data1.slope <- lm(y ~ x, data1) %.>% coef(.)[1]
data2.slope <- lm(y ~ x, data2) %.>% coef(.)[1]
data3.slope <- lm(y ~ x, data3) %.>% coef(.)[1]
data4.slope <- lm(y ~ x, data4) %.>% coef(.)[1]

data1.intercept <- lm(y ~ x, data1) %.>% coef(.)[2]
data2.intercept <- lm(y ~ x, data2) %.>% coef(.)[2]
data3.intercept <- lm(y ~ x, data3) %.>% coef(.)[2]
data4.intercept <- lm(y ~ x, data4) %.>% coef(.)[2]
```

f. R-Squared (2 pts).

```
data1.rsquared <- lm(y ~ x, data1) %.>% summary(.)$r.squared
data2.rsquared <- lm(y ~ x, data2) %.>% summary(.)$r.squared
data3.rsquared <- lm(y ~ x, data3) %.>% summary(.)$r.squared
data4.rsquared <- lm(y ~ x, data4) %.>% summary(.)$r.squared
```

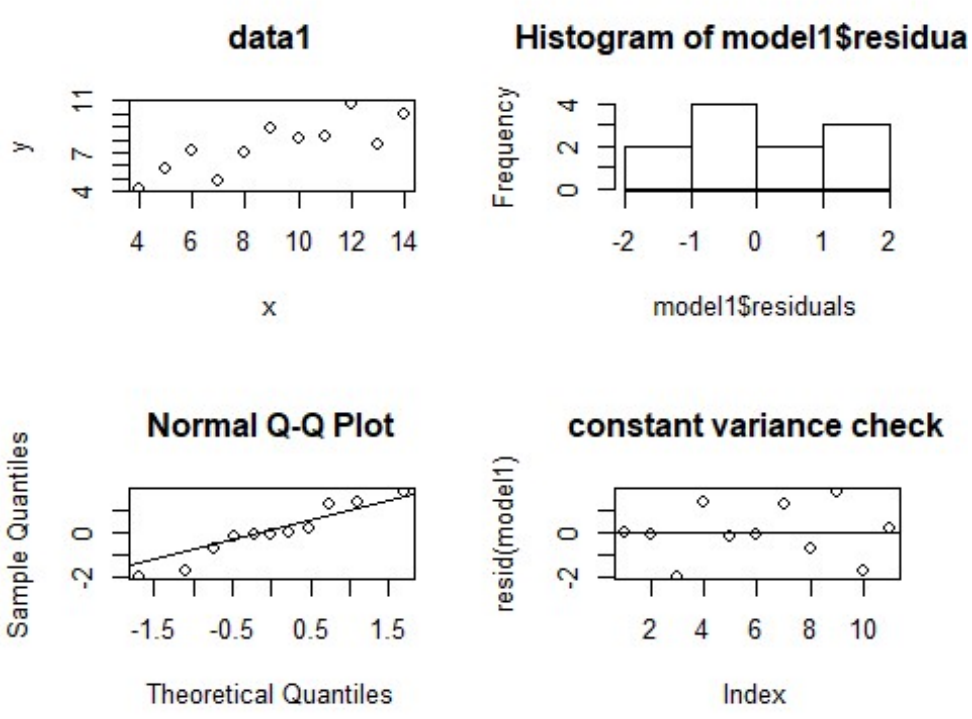
Data 1
Data 2
Data 3
Data 4
x
y
x
y
x
y
x
y
Mean
9.00
7.50
9.00
7.50

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
0.50
0.50
0.50
0.50
Slope
3.00
3.00
3.00
3.00
R-Squared
0.67
0.67

0.67
0.67

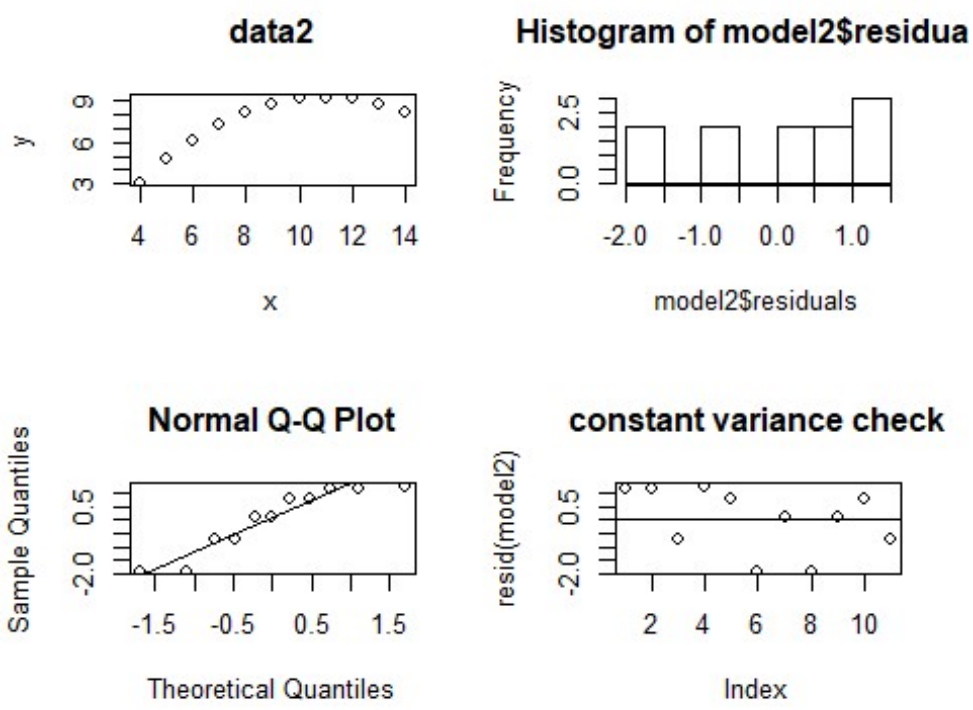
g. For each pair, is it appropriate to estimate a linear regression model? Why or why not? Be specific as to why for each pair and include appropriate plots! (4 pts)

```
# model 1 for data1
model1 <- lm(y ~ x, data1)

# set up
par(mfrow = c(2, 2))

# scatterplot of the raw data
plot(data1, main = "data1")

# histogram of the residuals
hist(model1$residuals)

# qqplot
qqnorm(model1$residuals)
qqline(model1$residuals)

# homoscedasticity check
plot(resid(model1), main = "constant variance check")
abline(h = 0)
```



Model 1 for data1 satisfies all conditions.

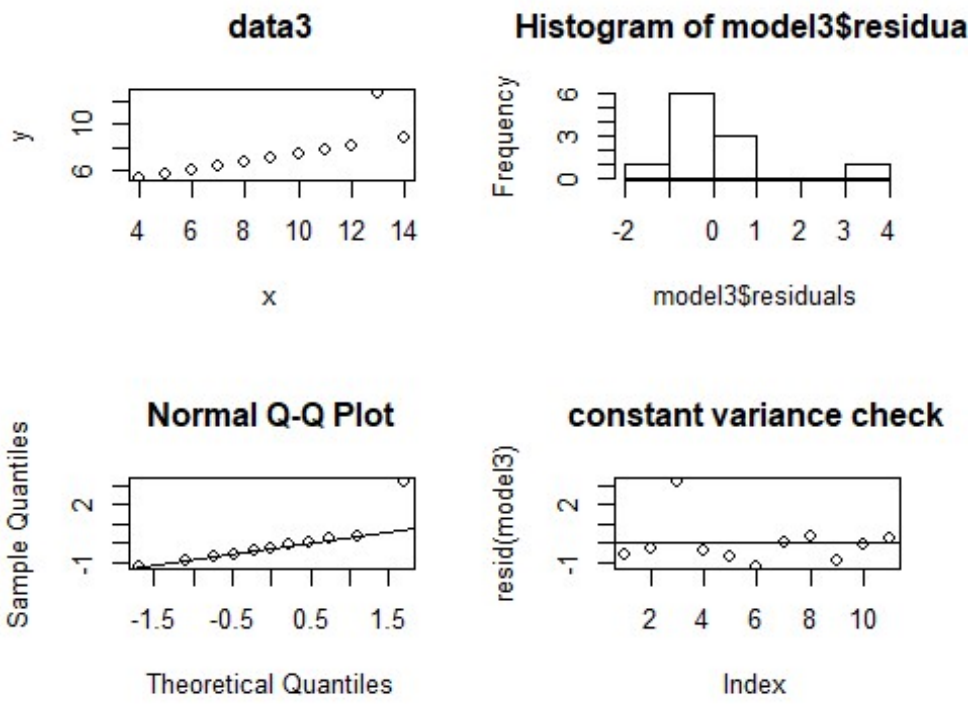
```r
# model 2 for data2
model2 <- lm(y ~ x, data2)

# set up
par(mfrow = c(2, 2))

# scatterplot of the raw data
plot(data2, main = "data2")

# histogram of the residuals
hist(model2$residuals)

# qqplot
qqnorm(model2$residuals)
qqline(model2$residuals)

# homoscedasticity check
plot(resid(model2), main = "constant variance check")
abline(h = 0)
```



Model 2 for data2 does not satisfy the conditions, e.g. there's a curvilinear relationship between the two variables, residuals are not normal and there's an issue of heteroscedasticity.

```r
# model 3 for data3
model3 <- lm(y ~ x, data3)
```

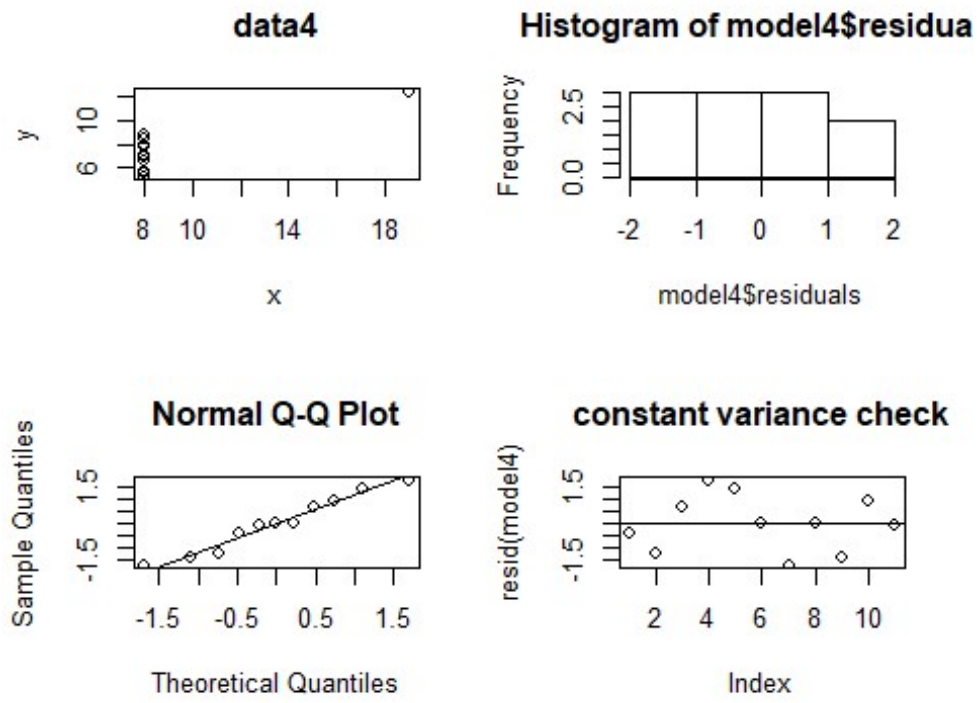
```r
# set up
par(mfrow = c(2, 2))

# scatterplot of the raw data
plot(data3, main = "data3")

# histogram of the residuals
hist(model3$residuals)

# qqplot
qqnorm(model3$residuals)
qqline(model3$residuals)

# homoscedasticity check
plot(resid(model3), main = "constant variance check")
abline(h = 0)
```



Model 3 for data3 also does not satisfy the conditions, e.g. there's an outliner skewing the data, residuals are not normal and there's an issue of heteroscedasticity.

```r
# model 4 for data4
model4 <- lm(y ~ x, data4)

# set up
par(mfrow = c(2, 2))
```

```r
# scatterplot of the raw data
plot(data4, main = "data4")

# histogram of the residuals
hist(model4$residuals)

# qqplot
qqnorm(model4$residuals)
qqline(model4$residuals)

# homoscedasticity check
plot(resid(model4), main = "constant variance check")
abline(h = 0)
```



Model 4 for data4 also does not satisfy the conditions, e.g. there's a binary relationship between the two variables, and residuals are not normally distributed. Therefore it is not suitable for linear regression, perhaps a logistic regression would suit better.

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

As shown from various plots above, it's important to visualize the relationship between variables before choosing an appropriate model to run, e.g. data4 would fit better with a logistic regression model. There are many assumptions that need to be met in regression, such as linearity, homosedasticity. Applying visualization such as qqplot, histogram,

residual plot, we can identify any problem underlying our data and check out the statistical assumptions are safely met.