# Problem Set 3

Ariel Huckabay and Kendall Weistroffer

February 2, 2018

## Introduction

Please complete the following tasks regarding the data in R. Please generate a solution document in R markdown and upload the .Rmd document and a rendered .doc, .docx, or .pdf document. Your work should be based on the data's being in the same folder as the .Rmd file. Please turn in your work on Canvas. Your solution document should have your answers to the questions and should display the requested plots.

These questions were rendered in R markdown through RStudio (https://www.rstudio.com/wp-content/uploads/2015/02/rmarkdown-cheatsheet.pdf, http://rmarkdown.rstudio.com ).

## Part 1

### Question 1

In this question, we return to ps1_data.csv. Please read in ps1_data.csv and find the values of m and b that minimize the sum of the square differences (FTOTINC-(m*FAMSIZE+b))^2 over all the families in the sample. In this context, the hhwt values are typically not used because each sampled family gives information only about its relation between FAMSIZE and FTOTINC. Please use the analytic formula derived in class and plot the least squares best fit line on a scatterplot of the data points. You may find geom_jitter helpful. That geometry also accepts an alpha parameter.

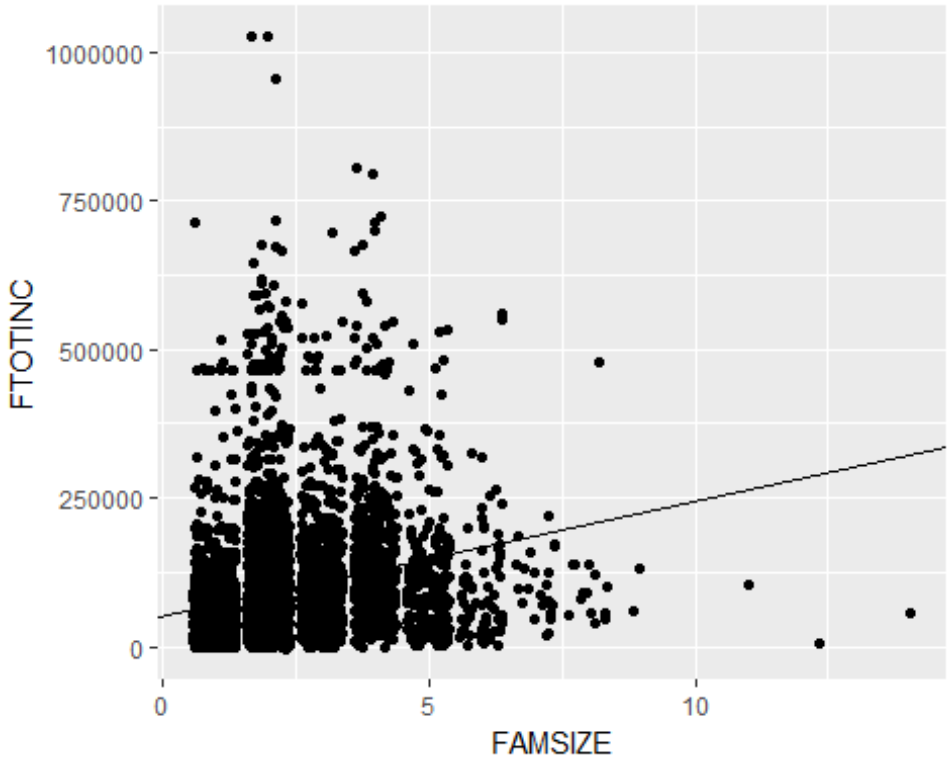
```
#read in problem set 1 dat
dat<-read.csv("ps1_data.csv")

#find minima
n <- nrow(dat)
yBar <- mean(dat$FTOTINC)
xBar <- mean(dat$FAMSIZE)

mNum <- (1/n) * sum((dat$FTOTINC * dat$FAMSIZE) - (yBar * xBar))
mDenom <- (1/n) * sum((dat$FAMSIZE)^2 - (xBar^2))

#store the values of m and b
mFinal <- mNum/mDenom
bFinal <- yBar - (mFinal*xBar)
```

```
#hatch the plot
g<-ggplot(dat, aes(x=FAMSIZE, y=FTOTINC)) + geom_point(position = "jitter")
g<-g+geom_abline(slope= mFinal, intercept = bFinal)
g
```




## Question 2

2a. What change in income is associated with each additional person according to the linear model?

## Response

According to the linear model, the change in income for each additional family member would be 1.934128610^{4}.

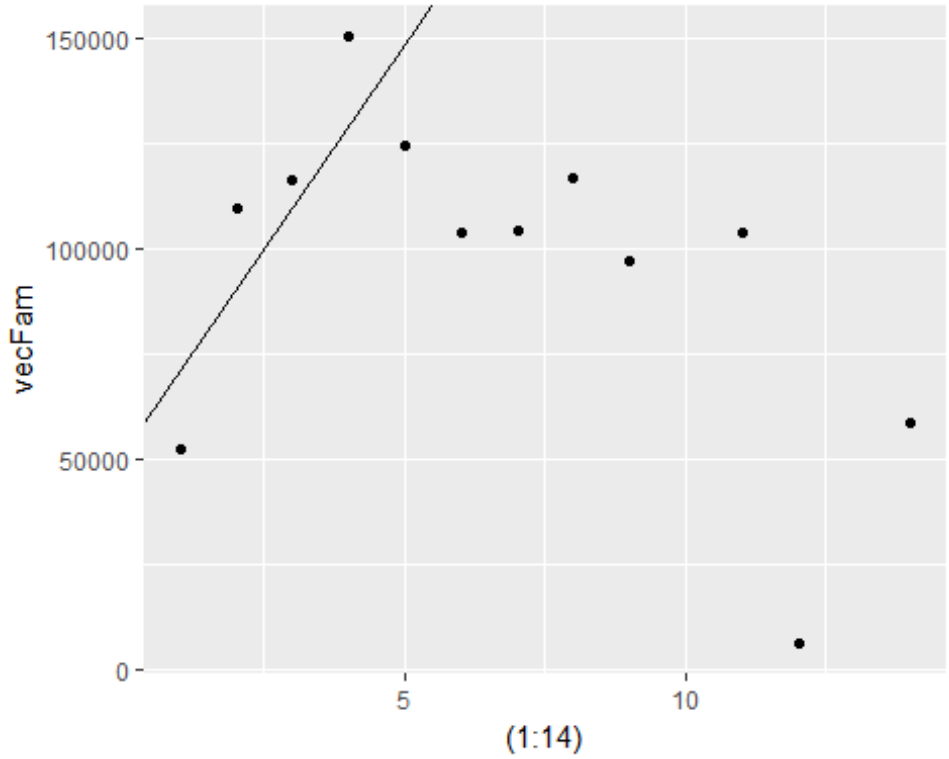
2.b. Plot unweighted mean income in each FAMSIZE. How do they compare to the fitted line?

```
#find mean income in each family size and place it into vecFam vector
vecFam<-c()
for(j in 1:max(dat$FAMSIZE)){
  count = 0
  sum = 0
  for(i in 1:nrow(dat)){
    if(dat$FAMSIZE[i] == j){
      count = count + 1
      sum = sum + dat$FTOTINC[i]
```

```
    }
  }
  vecFam[j] = sum/count
}

#sort for pleasant graph reasons
s<-sort(vecFam, decreasing = FALSE, na.last = FALSE)
vecPoints<-data.frame(s)
g1<-ggplot(vecPoints, aes(x= (1:14), y=vecFam))+geom_point()
g1<-g1+geom_abline(slope = mFinal, intercept = bFinal)
g1

## Warning: Removed 2 rows containing missing values (geom_point).
```




## Response

As can be seen from placing the fitted line from question 1 on this plot, it does not appear to approximate the distribution very well at all. It appears that the data is not linear.

### Question 3

If the incomes for 2 person families follow a Normal distribution, what are the maximum likelihood values of $\mu$ and $\sigma^2$? Is the histogram of the incomes consistent with the Normal distribution with this value for the mean and $\sqrt{\sigma^2}$ for sd?

```r
#Grab incomes of 2-Person Families
Total2Person <- (dat$FTOTINC[dat$FAMSIZE == 2])

#Compute values for mu and sigma^2
n <- length(Total2Person)
xBar2Person <- mean(Total2Person)
sigmaDen <- sum((Total2Person - xBar2Person)^2)
sigmaSquared <- sigmaDen/n
mu <- xBar2Person

#Print output for mu and sigma^2
cat("Mu:", mu)
```

```
## Mu: 109493.1
```

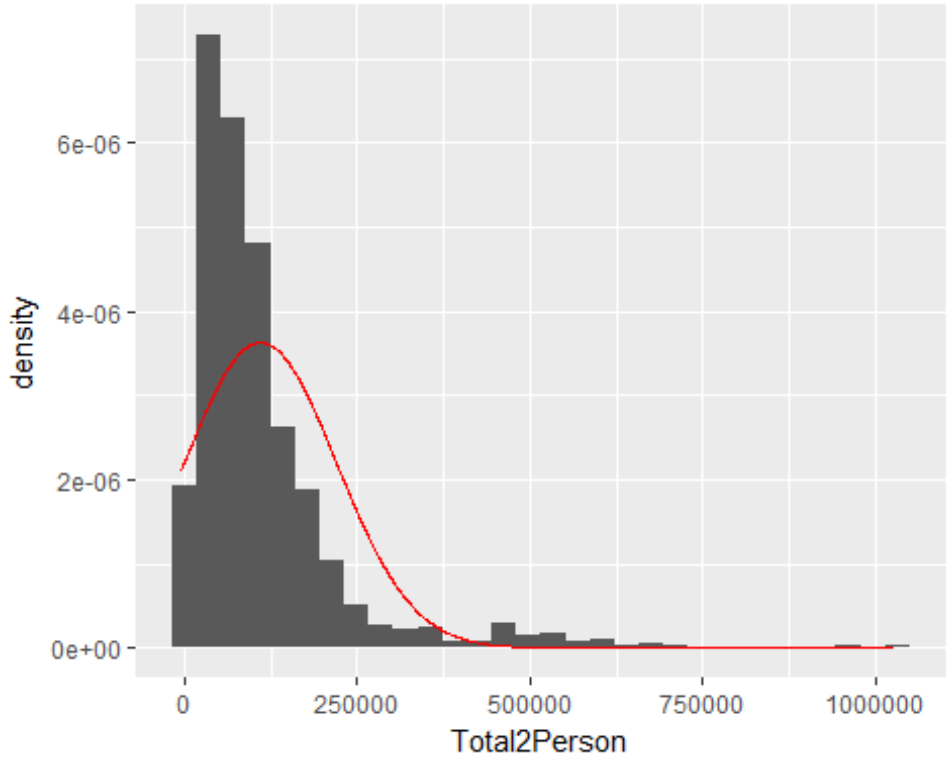
```r
cat(" Sigma Squared:", sigmaSquared)
```

```
##  Sigma Squared: 12092689704
```

```r
#Compute sd for graph comparisons
sdTest <- sqrt(sigmaSquared)

#Plot
TwoPersonIncome <- data.frame(Total2Person)
g3 <- ggplot(TwoPersonIncome)+geom_histogram(aes(x=Total2Person, y = ..densit
y..)) +
  stat_function(fun = dnorm, n = n, args = list(mean = xBar2Person, sd = sdTe
st), color = "red")
g3
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




##Response The historgram and normal curve bear a similar shape. However, historgram's peak is higher than the normal distributions.They are shape-wise consistent, but not a complete match.

## Question 4

Create 10,000 samples of size 10 from the standard Normal distribution and calculate the maximum likelihood values of $\mu$ and $\sigma$ for each. Compute the mean of the $\mu$s and the mean of the $\sigma$s. What proportion of the $\mu$s are less than or equal to 0? What proportion of the $\sigma$s are less than or equal to 1?

```
#Create 10,000 samples
nSamples <- 10000

#Create normal distributions of size 10 for each sample:
listSamples <- lapply(1:nSamples, function(x) rnorm(n=10))

#Function to compute sigma:
maxLSigmas<-function(x){
  sampleN <- length(x)
  sampleMu <- mean(x)
  sampleSigmaDen <- sum((x - sampleMu)^2)
  sampleSigmaSquared <- sampleSigmaDen/sampleN
  return (sampleSigmaSquared)
}

#Function to compute mu:
maxLMus<- function(x){
```

```
  sampleMu <- mean(x)
  return (sampleMu)
}

#Apply functions for Mu and Sigma:
sigmaList <- sapply(listSamples, maxLSigmas)
muList <- sapply(listSamples, maxLMus)

#Print out the mean of Mu and the mean of Sigma:
meanMu <-mean(muList)
cat("Mean Mu:", meanMu)
```

```
## Mean Mu: 0.009243256
```

```
meanSigma <- mean (sigmaList)
cat("\nMean Sigma:", meanSigma)
```

```
##
## Mean Sigma: 0.9042969
```

```
#Compute number of mus and sigmas less than or equal to 0:
negMus <- 0
lessSigmas <- 0
for(i in 1:length(muList)){
  if(muList[i] <= 0){
    negMus = negMus + 1
  }
  if(sigmaList[i] <= 1){
    lessSigmas = lessSigmas + 1
  }

}

#Calculate the proportions:
proportionMu <- negMus/length(muList)
proportionSigma <- lessSigmas/length(sigmaList)

cat("\nNegative Mus:", negMus, " Proportion:", proportionMu)
```

```
##
## Negative Mus: 4912  Proportion: 0.4912
```

```
cat("\nSigmas Less Than 1:", lessSigmas, " Proportion:", proportionSigma)
```

```
##
## Sigmas Less Than 1: 6475  Proportion: 0.6475
```

## Response

The mean Mu is 0.0092433 while the mean Sigma is 0.9042969. About 0.4912 of them were negative while about 0.6475 of the sigmas were less than 1.

## Question 5

The exponential distributions are a one parameter family of continuous distributions, $Exp(\lambda)$. Given $\lambda$, the sample space is $[0, \infty)$ and the probability density function is $f(x) = \lambda\exp(-\lambda x)$. Thus if $x_1 \ldots x_n$ are $n$ independent draws from an exponential distribution with parameter $\lambda$, the likelihood function of this sample is $\prod_{i=1}^{n} \lambda \exp(-\lambda x_i)$. What is the maximum likelihood value of $\lambda$ as a function of $x_1 \ldots x_n$? That is, given $x_1 \ldots x_n$, what value of $\lambda$ maximizes $\prod_{i=1}^{n} \lambda \exp(-\lambda x_i)$?

## Response

The maximum value for this function occurs when $\lambda$ is equal to the reciprocal of the mean of the inputs.

$$f(x_1, \ldots, x_n) = \prod \lambda e^{-\lambda x_i}$$

Take the natural log of both sides. Since natural log is an increasing function we know that the maximum value of $f(x_1, \ldots, x_n)$ occurs at the same place as that of $\ln(f(x_1, \ldots, x_n))$.

$$\ln(f(x_1, \ldots, x_n)) = \ln(\prod \lambda e^{-\lambda x_i})$$

$$\ln\big(f(x_1, \ldots, x_n)\big) = \sum \ln(\lambda e^{-\lambda x_i})$$

Differentiate with respect to $\lambda$:

$$\frac{f'(x_1, \ldots, x_n)}{f(x_1, \ldots, x_n)} = \sum \frac{e^{-\lambda x_i} - \lambda x_i e^{-\lambda x_i}}{\lambda e^{-\lambda x_i}}$$

Solve for 0 to find the critical point:

$$\sum \frac{e^{-\lambda x_i}(1 - x_i\lambda)}{\lambda e^{\lambda x_i}} = 0$$

$$\frac{1}{\lambda}\sum(1 - x_i\lambda) = 0$$

$$n - \lambda\sum x_i = 0$$

$$\lambda = \frac{n}{\sum x_i}$$

Which is the reciprocal of the mean of the inputs. This essentially minimizes the size of the denominator of the expression in the product, and since exponentials grow faster than polynomials of any power, we know that this must be a maximum.

## Question 6 (4441 only)

Projection view of regression: Consider the vectors $\overrightarrow{FTOTINC}$ and $\overrightarrow{FAMSIZE}$ from Question 1 and the vector $\vec{1}$ of the same length in which each entry is 1. Construct the matrix $\mathbf{X}$ that has $\vec{1}$ as its first column and $\overrightarrow{FAMSIZE}$ as its second column.

Note that $\overrightarrow{FTOTINC}$ is not in the span of $\vec{1}$ and $\overrightarrow{FAMSIZE}$. That is, there is no vector $\vec{b}$ of length 2 such that $\overrightarrow{FTOTINC} = \mathbf{X}\vec{b}$. However, $\overrightarrow{FTOTINC}$ can be written as a sum of a vector $\mathbf{X}\vec{b}$ in the span of $\vec{1}$ and $\overrightarrow{FAMSIZE}$ and a vector $\vec{e}$ orthogonal to both $\vec{1}$ and $\overrightarrow{FAMSIZE}$. Thus $\mathbf{X}^T\overrightarrow{FTOTINC} = \mathbf{X}^T(\mathbf{X}\vec{b} + \vec{e})$

6.a. Use the fact above, with the specification the vector $\vec{e}$ is orthogonal to both $\vec{1}$ and $\overrightarrow{FAMSIZE}$, to solve for $\vec{b}$ in terms of $\mathbf{X}$ and $\overrightarrow{FTOTINC}$ algebraically.

$$\mathbf{X}^T\overrightarrow{FTOTINC} = \mathbf{X}^T\left(\mathbf{X}\vec{b} + \vec{e}\right)$$

Use the distributive property of matrix multiplication over matrix addition:

$$\mathbf{X}^T\overrightarrow{FTOTINC} = \mathbf{X}^T(\mathbf{X}\vec{b}) + \mathbf{X}^T\vec{e}$$

Use the fact that $\vec{e}$ is orthogonal to both rows of $\mathbf{X}^T$, which yields a 0 vector, and that matrix multiplication is associative:

$$\mathbf{X}^T\overrightarrow{FTOTINC} = (\mathbf{X}^T\mathbf{X})\vec{b}$$

Use the fact that, upon inspection, the columns of $\mathbf{X}$ are linearly independent (clear upon inspection) and thus the inverse of $(\mathbf{X}^T\mathbf{X})$ must exist:

$$(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\overrightarrow{FTOTINC} = (\mathbf{X}^T\mathbf{X})^{-1}(\mathbf{X}^T\mathbf{X})\vec{b} = \vec{b}$$

This completes the algebraic computation of the vector $\vec{b}$.

6.b. Carry out the calcuation for $\vec{b}$ in 6.a. Compare $\vec{b}$ to your answers for Question 1. Note that %*% is the symbol for matrix multiplication in R, and the inverse of a matrix M can be calculated using the function call solve(M).

```
#make matrix
vex<-c()

for(i in 1:(2*nrow(dat))){
  if (i<=nrow(dat)){vex[i]=1}
  else{vex[i] = dat$FAMSIZE[i-nrow(dat)]}
}

#set up some objects that will be needed for the computation
```

```
X = matrix(vex, nrow = nrow(dat), ncol=2)
FTOTINC<-matrix(X[,2], nrow = nrow(dat), ncol=1)
xT<-t(X)
right<-xT%*%X
rightInv<-solve(right)

#computerize
mult = rightInv%*%xT
b = mult%*%FTOTINC
b

##                [,1]
## [1,] -3.034682e-15
## [2,]  1.000000e+00
```

## Response

These are the values for b. It differs significantly from the values obtained in question 1, with smaller values for slope and intercept.