DATA 106 - Assignment 1 Solutions

Jillian Morrison

September 2, 2019

1. a. Create 2 data frames, buildings (first data frame) and data (second data frame)

```
buildings <- data.frame(location=c(1, 2, 3), name=c("building1", "building2", "building3"))

data <- data.frame(survey=c(1,1,1,2,2,2), location=c(1,2,3,2,3,1),
efficiency=c(51,64,70,71,80,58))

buildings
```

```
##   location      name
## 1        1 building1
## 2        2 building2
## 3        3 building3
```

```
data
```

```
##   survey location efficiency
## 1      1        1         51
## 2      1        2         64
## 3      1        3         70
## 4      2        2         71
## 5      2        3         80
## 6      2        1         58
```

Notice that the 2 dataframes have the variable location in common. Merge the two dataframes by this variable. Name the resulting dataframe COW_Buildings

```
COW_Buildings<- merge(buildings, data, by="location")
```

b. Rename the location variable in the 'building' dataset as "Location.ID". Call this new dataset 'buildings_2'

```
buildings_2 <- buildings
colnames(buildings_2)[1]<-"Location.ID"
buildings_2
```

```
##   Location.ID      name
## 1           1 building1
## 2           2 building2
## 3           3 building3
```

c. Merge the datasets buildings_2 and data. Call this new dataframe NewCOWbuildings

```
NewCOWbuildings <- merge(buildings_2, data, by.x="Location.ID", by.y="location")

NewCOWbuildings
```

```
##   Location.ID      name survey efficiency
## 1           1 building1      1         51
## 2           1 building1      2         58
## 3           2 building2      1         64
## 4           2 building2      2         71
## 5           3 building3      1         70
## 6           3 building3      2         80
```

d. explain the difference between inner join, outer join, right join, left join and cross join.

2. Refer to the table below:

```
Gender <- c("Female","Female","Male","Male")
Restaurant <- c("Yes","No","Yes","No")
Count <- c(220, 780, 400, 600)
DiningSurvey <- data.frame(Gender, Restaurant, Count)
DiningSurvey
```

```
##   Gender Restaurant Count
## 1 Female        Yes   220
## 2 Female         No   780
## 3   Male        Yes   400
## 4   Male         No   600
```

a. Check if any row has count more than 400

```
which(DiningSurvey$Count > 400)
```

```
## [1] 2 4
```

```
table(DiningSurvey$Count > 400)
```

```
## 
## FALSE  TRUE 
##     2     2
```

b. Append the new variable Flavour to the DiningSurvey dataset.

```
DiningSurvey$Flavour <- c("Yes", "No", "Yes", NA)
DiningSurvey
```

```
##   Gender Restaurant Count Flavour
## 1 Female        Yes   220     Yes
## 2 Female         No   780      No
## 3   Male        Yes   400     Yes
## 4   Male         No   600    <NA>
```

c. Use the “is.na()” argument to find missing Restaurant data by Gender. Hint(Use the table function to tabulate the variables is.na(Flavour) and Gender)

```
table(DiningSurvey$Gender,is.na(DiningSurvey$Flavour))
```

```
##
##          FALSE TRUE
##   Female     2    0
##   Male       1    1
```

4. Consider the RentalUnits Dataset

```
RentalUnits <- matrix(c(45,37,34,10,15,12,24,18,19),ncol=3,byrow=TRUE)
colnames(RentalUnits) <- c("Section1","Section2","Section3")
rownames(RentalUnits) <- c("Rented","Vacant","Reserved")
RentalUnits <- as.table(RentalUnits)
RentalUnits
```

```
##          Section1 Section2 Section3
## Rented         45       37       34
## Vacant         10       15       12
## Reserved       24       18       19
```

a. Use the margin.table() or rowSums() function to find the amount of Occupancy summed over Sections.

```
margin.table(RentalUnits,1)   #Over Columns
```

```
##   Rented   Vacant Reserved
##      116       37       61
```

```
rowSums((RentalUnits))
```

```
##   Rented   Vacant Reserved
##      116       37       61
```

b. Find the amount of Units summed by Section.

```
margin.table(RentalUnits, 2)  #Over rows
```

```
## Section1 Section2 Section3
##       79       70       65
```

```
colSums(RentalUnits)
```

```
## Section1 Section2 Section3
##       79       70       65
```

c. Use the “prop.table()” function to create a basic table of proportions.

```
prop.table(RentalUnits)
```

```
##             Section1   Section2   Section3
## Rented   0.21028037 0.17289720 0.15887850
## Vacant   0.04672897 0.07009346 0.05607477
## Reserved 0.11214953 0.08411215 0.08878505
```

d. Find row percentages, and column percentages.

```
prop.table(RentalUnits, 1)*100  #ROW
```

```
##          Section1 Section2 Section3
## Rented   38.79310 31.89655 29.31034
## Vacant   27.02703 40.54054 32.43243
## Reserved 39.34426 29.50820 31.14754
```

```
prop.table(RentalUnits, 2)*100  #Columns
```

```
##          Section1 Section2 Section3
## Rented   56.96203 52.85714 52.30769
## Vacant   12.65823 21.42857 18.46154
## Reserved 30.37975 25.71429 29.23077
```

e. Use “summary()” to perform a Chi-Square Test of Independence, of the “RentalUnits” variables. Describe what the Chi- Square test of indendence does (You do not need to go into details).

```
summary(RentalUnits)
```

```
## Number of cases in table: 214
## Number of factors: 2
## Test for independence of all factors:
##  Chisq = 2.2034, df = 4, p-value = 0.6984
```

4. Consider the url ‘https://statbel.fgov.be/en/themes/population/structure-population’ I have extracted all the information in table ‘Structure of Population’ of Belgium. You will need to install the package called rvest.

```
#install.packages('rvest')
library('rvest')
```

```
## Warning: package 'rvest' was built under R version 3.5.3
```

```
## Loading required package: xml2
```

```
## Warning: package 'xml2' was built under R version 3.5.2
```

```r
url='https://statbel.fgov.be/en/themes/population/structure-population'
TAB=read_html(url)%>%html_nodes('td')%>%html_text()
NAMES=read_html(url)%>%html_nodes('th')%>%html_text()


M_ <- as.numeric(gsub(",","",unlist(TAB)))
```

```
## Warning: NAs introduced by coercion
```

```r
M=data.frame(matrix(M_,ncol=7,byrow=T))


#df <- as.data.frame(matrix(as.numeric(as.character(M_)), nrow=length(M), byrow=F))

M=cbind(NAMES[9:23],M)
names(M)=NAMES[1:8]
M
```

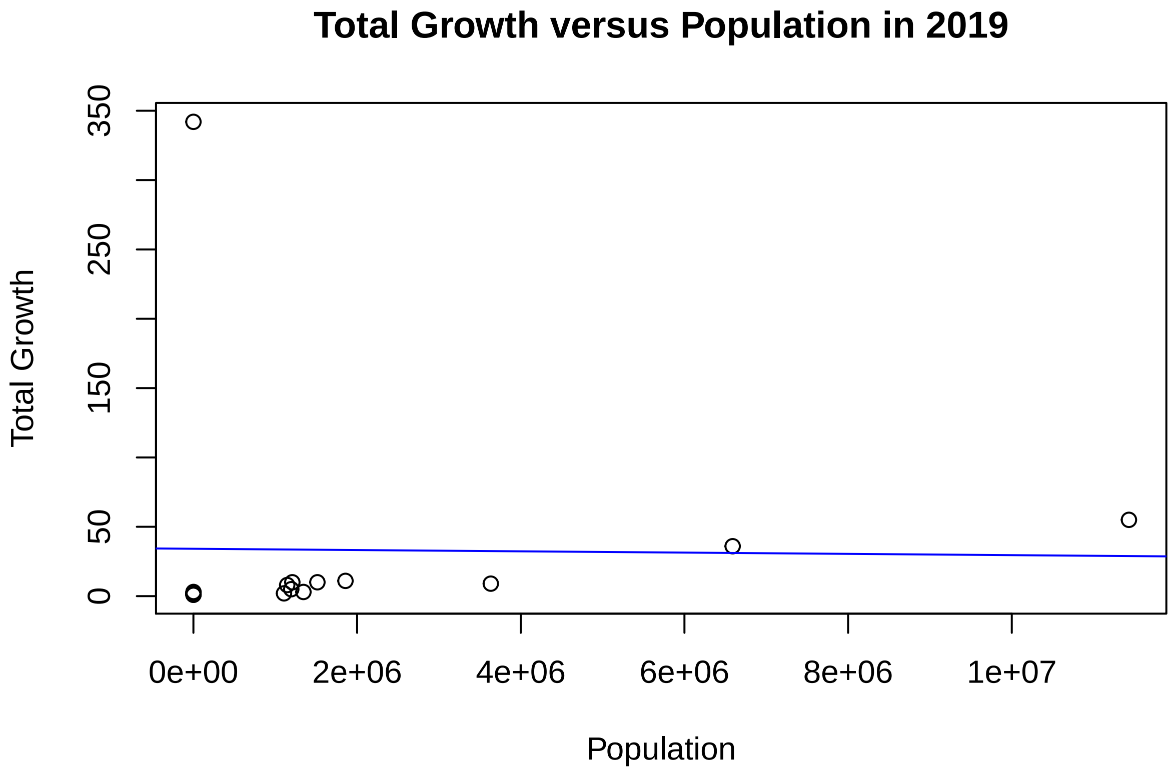
```
##             Place of residence Population on 1st January 2018
## 1                      Belgium                       11376070
## 2      Brussels-Capital Region                        1198726
## 3               Flemish Region                        6552967
## 4               Walloon Region                        3624377
## 5    German-speaking Community                             77
## 6          Province of Antwerp                        1847486
## 7          Province of Limburg                            871
## 8    Province of East Flanders                        1505053
## 9  Province of Flemish Brabant                        1138489
## 10   Province of West Flanders                        1191059
## 11 Province of Walloon Brabant                            401
## 12         Province of Hainaut                        1341645
## 13           Province of Liège                        1105326
## 14      Province of Luxembourg                            283
## 15           Province of Namur                            493
##    Natural balance Internal migration balance
## 1                7                          0
## 2                8                        -15
## 3              939                         12
## 4               -2                          3
## 5               60                         79
## 6                2                       -448
## 7              -49                        180
## 8              225                          4
## 9              373                          5
## 10              -2                          3
## 11             100                          2
## 12              -2                          2
## 13            -476                       -522
## 14             124                        311
## 15            -278                        221
##    International migration balance Statistical adjustment Total growth
## 1                               50                     -2           55
## 2                               17                   -730           10
## 3                               25                     -1           36
## 4                                8                    -24            9
## 5                              208                     -5          342
## 6                               NA                   -478           11
## 7                                3                   -131            3
## 8                                6                   -279           10
## 9                                3                   -254            8
## 10                               4                   -103            5
## 11                             652                    -57            2
## 12                               2                    268            3
## 13                               3                   -260            2
## 14                             965                     11            1
## 15                               1                     14            1
##    Population on 1st January 2019
## 1                        11431406
## 2                         1208542
## 3                         6589069
## 4                         3633795
## 5                              78
## 6                         1857986
## 7                             874
## 8                         1515064
## 9                         1146175
## 10                        1195796
## 11                            404
## 12                        1344241
## 13                        1106992
## 14                            285
## 15                            494
```

```
#######NOTE#########
##Header cells - contains header information (created with the <th> element)
##Standard cells - contains data (created with the <td> element)

##These can be found in the page source see: https://smallbusiness.chron.com/see-html-code-46954.html
###################
```

a. Create a scatterplot of Total Growth on the y axis and Population on 1st January 2019 on the x axis. Be sure to dd axis and column names. Add a linear regression line to the plot (see http://www.sthda.com/english/wiki/scatter-plots-r-base-graphs)

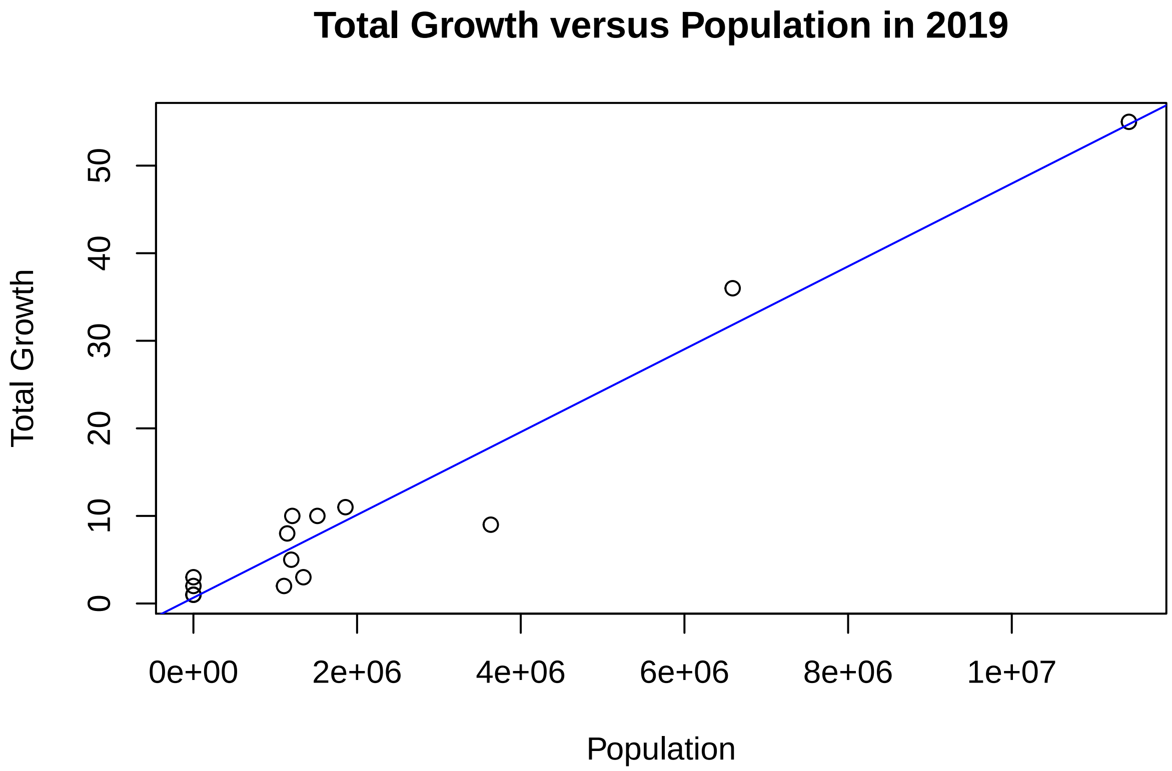
```
plot(M[,8],M[,7], main="Total Growth versus Population in 2019", xlab="Population", ylab="Total Growth")
abline(lm(M[,7] ~ M[,8]), col = "blue")
```



b. Remove the outlier from part a and remake the plot. Also add a linear regression line to the plot.

```
M_nooutlier= subset(M, M$`Total growth`<300)

plot(M_nooutlier[,8],M_nooutlier[,7], main="Total Growth versus Population in 2019", xlab="Population",
abline(lm(M_nooutlier[,7]~M_nooutlier[,8]), col = "blue")
```



c. Describe what you see with and without the outlier.

d. Which element of the table was “coerced” into being missing (i.e. NA). How would you replace the NA with the correct value?

```
M$`International migration balance`[6]=8992
```