R coding for public policy

Assignment 3

[Name removed]

Assignment Instruction:

Once you download and open Assignment3.rmd in R studio,

Please complete all the problems in the empty line between "'{r} and"' You can add more empty lines by press Enter in the empty line You can also click the green arrow next to each code chunk to check your code Please include only the relevant codes in the chunks Once you complete all the problems, click "Knit"

Submit the knitted word document and your markdown as attachments to NYU Classes - Assignment - Assignment 3

In assignment 3, we will use the following datasets:

1. The SAT dataset from assignment 2 and lecture 3 link: https://data.cityofnewyork.us/Education/2012-SAT-Results/f9bf-2cp4 *make sure you find the correct url for the csv file

The Baby names dataset from lecture 3 link: https://health.data.ny.gov/Health/Baby-Names-Beginning-2007/jxy9-yhdk *make sure you find the correct url for the csv file

#import and clean the SAT and baby names datasets

```
SAT<-read.csv("https://data.cityofnewyork.us/resource/f9bf-2cp4.csv")
Baby<-read.csv("https://health.data.ny.gov/api/views/jxy9-
yhdk/rows.csv?accessType=DOWNLOAD")

#Prepping SAT - converting to character and some to numeric when appropriate.
Convert 's' to NAs
SAT[,2]<-as.character(SAT[,2])
SAT[,3]<-as.numeric(levels(SAT[,3])[SAT[,3]])

## Warning: NAs introduced by coercion

SAT[,4]<-as.numeric(levels(SAT[,4])[SAT[,4]])

## Warning: NAs introduced by coercion

SAT[,5]<-as.numeric(levels(SAT[,5])[SAT[,5]])

## Warning: NAs introduced by coercion
```

```r
SAT[,6]<-as.numeric(levels(SAT[,6])[SAT[,6]])

## Warning: NAs introduced by coercion

#Prepping Baby
Baby[,1]<-as.numeric(Baby[,1])
Baby[,2]<-as.character(Baby[,2])
Baby[,5]<-as.numeric(Baby[,5])
```

#Problem 1 [Baby] try using both baby[,] and subset() to create a baby sample for "KINGs" county in 2015. Name the two datasets, then check for general differences and similaries between the two. You only need to report the class(es) for the two datasets, by printing them to the console.

```r
Baby_ind_Kings2015<-Baby[Baby$County=="Kings" & Baby$Year==2015,]

Baby_ss_Kings2015<-subset(Baby, subset = (County == "Kings" & Year == 2015))

class(Baby_ind_Kings2015)

## [1] "data.frame"

str(Baby_ind_Kings2015)

## 'data.frame':    11455 obs. of  5 variables:
##  $ Year      : num  2015 2015 2015 2015 2015 ...
##  $ First.Name: chr  "DAVID" "JACOB" "MOSHE" "LEAH" ...
##  $ County    : Factor w/ 125 levels "Albany","ALBANY",..: 46 46 46 46 46 4
6 46 46 46 46 ...
##  $ Sex       : Factor w/ 2 levels "F","M": 2 2 2 1 2 2 2 2 1 1 ...
##  $ Count     : num  257 218 201 197 196 195 193 184 179 176 ...

head(Baby_ind_Kings2015)

##       Year First.Name County Sex Count
## 89942 2015      DAVID  Kings   M   257
## 89943 2015      JACOB  Kings   M   218
## 89944 2015      MOSHE  Kings   M   201
## 89945 2015       LEAH  Kings   F   197
## 89946 2015     JOSEPH  Kings   M   196
## 89947 2015     JAYDEN  Kings   M   195

class(Baby_ss_Kings2015)

## [1] "data.frame"

str(Baby_ss_Kings2015)

## 'data.frame':    11455 obs. of  5 variables:
##  $ Year      : num  2015 2015 2015 2015 2015 ...
##  $ First.Name: chr  "DAVID" "JACOB" "MOSHE" "LEAH" ...
##  $ County    : Factor w/ 125 levels "Albany","ALBANY",..: 46 46 46 46 46 4
```

```
6 46 46 46 46 ...
##  $ Sex       : Factor w/ 2 levels "F","M": 2 2 2 1 2 2 2 2 1 1 ...
##  $ Count     : num  257 218 201 197 196 195 193 184 179 176 ...
```

```
head(Baby_ss_Kings2015)
```

```
##       Year First.Name County Sex Count
## 89942 2015      DAVID  Kings   M   257
## 89943 2015      JACOB  Kings   M   218
## 89944 2015      MOSHE  Kings   M   201
## 89945 2015       LEAH  Kings   F   197
## 89946 2015     JOSEPH  Kings   M   196
## 89947 2015     JAYDEN  Kings   M   195
```

#Problem 2 [Baby] use mean() and sd() to calculate the mean and standard deviation for "Count" of "kings" county in 2016, print the mean and the standard deviation

```
Baby_Kings_2016<-Baby[Baby$County=="Kings" & Baby$Year==2016, ]

mean(Baby_Kings_2016$Count, na.rm = TRUE)
```

```
## [1] 3.659716
```

```
sd(Baby_Kings_2016$Count, na.rm = TRUE)
```

```
## [1] 11.60765
```

#Problem 3 [Baby] use max() to find the most popular baby boy name and the most popular baby girl name for "KINGs" county in 2016

```
#Male Max
Baby_Kings_2016_male <- subset(Baby_Kings_2016, Sex == "M", select = c("First
.Name", "Count") )
print(Baby_Kings_2016_male[which.max(Baby_Kings_2016_male$Count),])
```

```
##   First.Name Count
## 1      DAVID   231
```

```
#Female Max
Baby_Kings_2016_female <- subset(Baby_Kings_2016, Sex == "F", select = c("Fir
st.Name", "Count") )
print(Baby_Kings_2016_female[which.max(Baby_Kings_2016_female$Count),])
```

```
##   First.Name Count
## 5     OLIVIA   210
```

#Problem 4 [Baby] use table() to create a two-way frequency table of "Year.f" and "Sex" for only "new york" and "kings"

```
Year.f <-as.factor(Baby$Year)
Baby$Year.f=Year.f
Baby_NY <- subset(Baby, County == "New York", select = c("Year.f", "Sex", "Co
unty") )
```

```
Baby_Kings <- subset(Baby, County == "Kings", select = c("Year.f", "Sex", "Co
unty") )
Baby_NY_Kings<-rbind(Baby_NY,Baby_Kings)

Baby_Table_NYKings<-table(Baby_NY_Kings$Year.f, Baby_NY_Kings$Sex)
```

#Problem 5 [Baby] use prop.table() to create the row percent table for the two-way frequency table in Problem 4

```
prop.table(Baby_Table_NYKings,1)

##
##                F         M
##   2007
##   2008
##   2009
##   2010
##   2011
##   2012
##   2013
##   2014 0.5254079 0.4745921
##   2015 0.5440303 0.4559697
##   2016 0.5405554 0.4594446

prop.table(table(Baby_NY_Kings$Year.f, Baby_NY_Kings$Sex))

##
##                  F          M
##   2007 0.00000000 0.00000000
##   2008 0.00000000 0.00000000
##   2009 0.00000000 0.00000000
##   2010 0.00000000 0.00000000
##   2011 0.00000000 0.00000000
##   2012 0.00000000 0.00000000
##   2013 0.00000000 0.00000000
##   2014 0.03153594 0.02848588
##   2015 0.26138176 0.21907267
##   2016 0.24839802 0.21112572
```

#Problem 6 [SAT] use the mean() and sd() and one of members from the apply() family to calculate means and standard deviations of all three subject tests, print the answers.

```
test_scores<-list("SAT_Crit_Read"=SAT$sat_critical_reading_avg_score,"SAT_mat
h"=SAT$sat_math_avg_score,"SAT_Writing"=SAT$sat_writing_avg_score)

print("Mean test scores")

## [1] "Mean test scores"

lapply(test_scores, mean, na.rm=TRUE)
```

```
## $SAT_Crit_Read
## [1] 400.8504
##
## $SAT_math
## [1] 413.3682
##
## $SAT_Writing
## [1] 393.9857
```

```
print("Std Dev test scores")
```

```
## [1] "Std Dev test scores"
```

```
lapply(test_scores, sd, na.rm=TRUE)
```

```
## $SAT_Crit_Read
## [1] 56.80278
##
## $SAT_math
## [1] 64.68466
##
## $SAT_Writing
## [1] 58.63511
```

#Problem 7 *bonus* [Baby] What are the means of "Count" by year?

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
group_by(Baby, Year) %>%
  summarize(m=mean(Count))
```

```
## # A tibble: 10 x 2
##     Year     m
##    <dbl> <dbl>
##  1  2007 18.9
##  2  2008 18.6
##  3  2009 18.8
##  4  2010 18.8
##  5  2011 18.5
##  6  2012 18.5
##  7  2013 18.2
```

```
##  8  2014 15.1
##  9  2015  2.54
## 10  2016  2.53
```