Final

Stephanie

7/30/2021

#Problem 1

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.6
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

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

```
library(readr)
alab_pop <- read_csv("~/Documents/Data/PSCI Data/sub-est2016_1.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   SUMLEV = col_character(),
##   STATE = col_character(),
##   COUNTY = col_character(),
##   PLACE = col_character(),
##   COUSUB = col_character(),
##   CONCIT = col_character(),
##   PRIMGEO_FLAG = col_double(),
##   FUNCSTAT = col_character(),
##   NAME = col_character(),
##   STNAME = col_character(),
##   CENSUS2010POP = col_character(),
##   ESTIMATESBASE2010 = col_double(),
##   POPESTIMATE2010 = col_double(),
##   POPESTIMATE2011 = col_double(),
##   POPESTIMATE2012 = col_double(),
##   POPESTIMATE2013 = col_double(),
##   POPESTIMATE2014 = col_double(),
##   POPESTIMATE2015 = col_double(),
##   POPESTIMATE2016 = col_double()
## )
```

```
alask_pop <- read_csv("~/Documents/Data/PSCI Data/sub-est2016_2.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   SUMLEV = col_character(),
##   STATE = col_character(),
##   COUNTY = col_character(),
##   PLACE = col_character(),
##   COUSUB = col_character(),
##   CONCIT = col_character(),
##   PRIMGEO_FLAG = col_double(),
##   FUNCSTAT = col_character(),
##   NAME = col_character(),
##   STNAME = col_character(),
##   CENSUS2010POP = col_character(),
##   ESTIMATESBASE2010 = col_double(),
##   POPESTIMATE2010 = col_double(),
##   POPESTIMATE2011 = col_double(),
##   POPESTIMATE2012 = col_double(),
##   POPESTIMATE2013 = col_double(),
##   POPESTIMATE2014 = col_double(),
##   POPESTIMATE2015 = col_double(),
##   POPESTIMATE2016 = col_double()
## )
```

#Appending the two datasets

```
comb_pop<- rbind(alab_pop,alask_pop)
```

```
state_pop <- read_csv("~/Documents/Data/PSCI Data/sub-est2016_all.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   SUMLEV = col_character(),
##   STATE = col_character(),
##   COUNTY = col_character(),
##   PLACE = col_character(),
##   COUSUB = col_character(),
##   CONCIT = col_character(),
##   PRIMGEO_FLAG = col_double(),
##   FUNCSTAT = col_character(),
##   NAME = col_character(),
##   STNAME = col_character(),
##   CENSUS2010POP = col_character(),
##   ESTIMATESBASE2010 = col_double(),
##   POPESTIMATE2010 = col_double(),
##   POPESTIMATE2011 = col_double(),
##   POPESTIMATE2012 = col_double(),
##   POPESTIMATE2013 = col_double(),
##   POPESTIMATE2014 = col_double(),
```

```
##   POPESTIMATE2015 = col_double(),
##   POPESTIMATE2016 = col_double()
## )
```

#Inspecting the unique states in the csv

```
states <- unique(state_pop$STNAME)
states
```

```
##  [1] "Alabama"              "Alaska"               "Arizona"
##  [4] "Arkansas"             "California"           "Colorado"
##  [7] "Connecticut"          "Delaware"             "District of Columbia"
## [10] "Florida"              "Georgia"              "Hawaii"
## [13] "Idaho"                "Illinois"             "Indiana"
## [16] "Iowa"                 "Kansas"               "Kentucky"
## [19] "Louisiana"            "Maine"                "Maryland"
## [22] "Massachusetts"        "Michigan"             "Minnesota"
## [25] "Mississippi"          "Missouri"             "Montana"
## [28] "Nebraska"             "Nevada"               "New Hampshire"
## [31] "New Jersey"           "New Mexico"           "New York"
## [34] "North Carolina"       "North Dakota"         "Ohio"
## [37] "Oklahoma"             "Oregon"               "Pennsylvania"
## [40] "Rhode Island"         "South Carolina"       "South Dakota"
## [43] "Tennessee"            "Texas"                "Utah"
## [46] "Vermont"              "Virginia"             "Washington"
## [49] "West Virginia"        "Wisconsin"            "Wyoming"
```

There are 51 unique states included in the dataset.

```
sub_pop <- subset(state_pop, select = c("NAME", "STNAME", "POPESTIMATE2012"))
```

#To find the rows that give the total population for the entire state, will subset by the condition where NAME is equal to STNAME. Also used distinct function so that there are not redundant observations.

```
total_pop<- subset(sub_pop, NAME==STNAME)%>%
  distinct(NAME, .keep_all=TRUE)%>%
  select(-STNAME)
dim(total_pop)
```

```
## [1] 51  2
```

```
sq_mileage <- read_csv("https://raw.githubusercontent.com/jakevdp/PythonDataScienceHandbook/master/notel
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   state = col_character(),
##   `area (sq. mi)` = col_double()
## )
```

```
merged_pop = inner_join(sq_mileage,total_pop, by=c("state"="NAME"))
```

Observations will be matched by the state name as this data is present in both datasets. The questions says not to merge observations that have no match, so an inner join will be used. #Creating new variable that tells each state's pop density

```
merged_pop$pop_density<- merged_pop$POPESTIMATE2012/merged_pop$`area (sq. mi)`
```

```
econ_data <- read_csv("~/Documents/Data/PSCI Data/ECN_2012_US_52A1.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   GEO.id = col_character(),
##   GEO.id2 = col_character(),
##   `GEO.display-label` = col_character(),
##   GEO.annotation.id = col_character(),
##   NAICS.id = col_character(),
##   `NAICS.display-label` = col_character(),
##   NAICS.annotation.id = col_character(),
##   YEAR.id = col_character(),
##   ESTAB = col_character(),
##   RCPTOT = col_character(),
##   PAYANN = col_character(),
##   PAYQTR1 = col_character(),
##   EMP = col_character(),
##   ADMIN_PCT = col_character(),
##   IMPUT_PCT = col_character()
## )
```

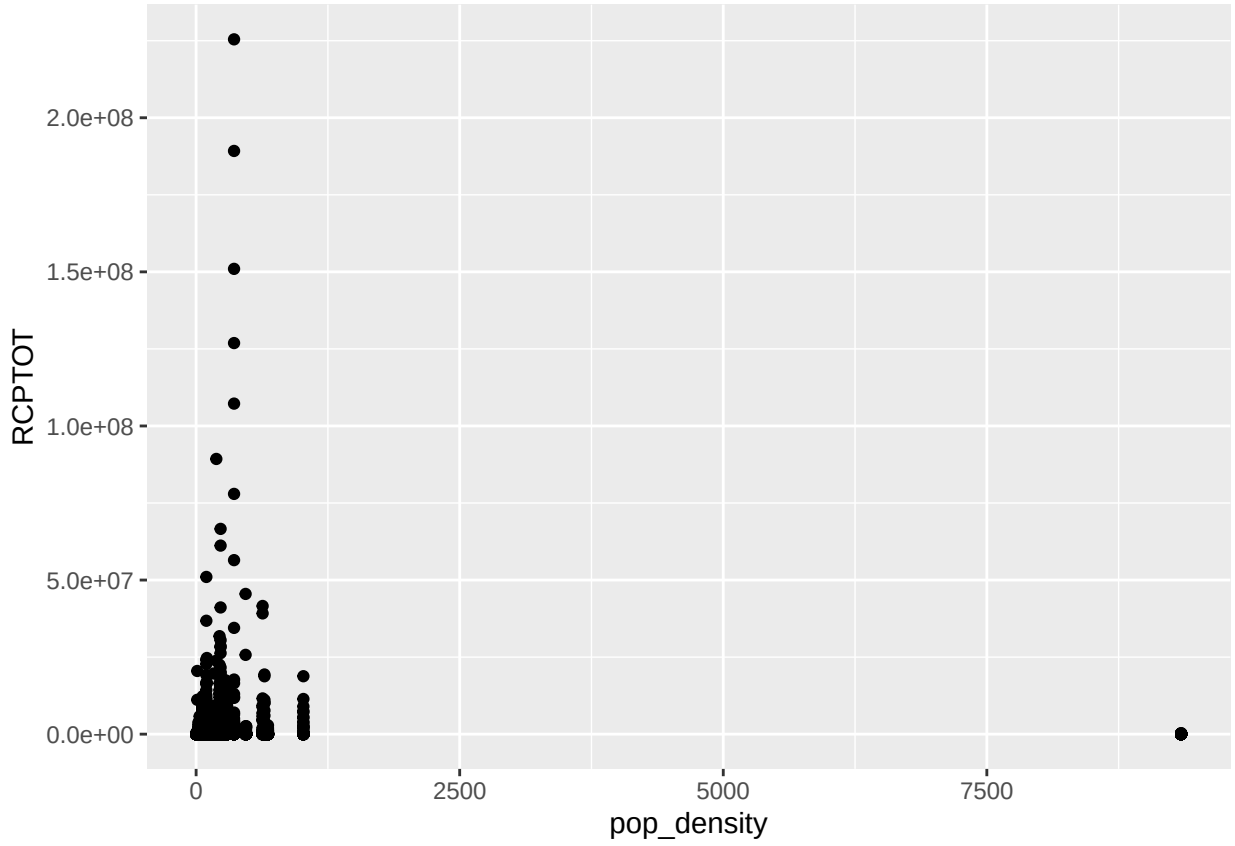
```
#Filter out the first row
econ_data<-econ_data%>%
  filter(!row_number()==1)
```

```
econ_data$RCPTOT<-as.numeric(econ_data$RCPTOT)
#Grouped data first by State, then by sector
#Found the total revenue per sector and renamed GEO label column to "state" for simplicity
total_rev <- econ_data%>%
  group_by(`GEO.display-label`,`NAICS.display-label`)%>%
  summarise(RCPTOT=sum(RCPTOT, na.rm=TRUE))%>%
  rename(state = `GEO.display-label`)
```

```
## `summarise()` has grouped output by 'GEO.display-label'. You can override using the `.groups` argume
```

` #Merge revenue and pop density dataset

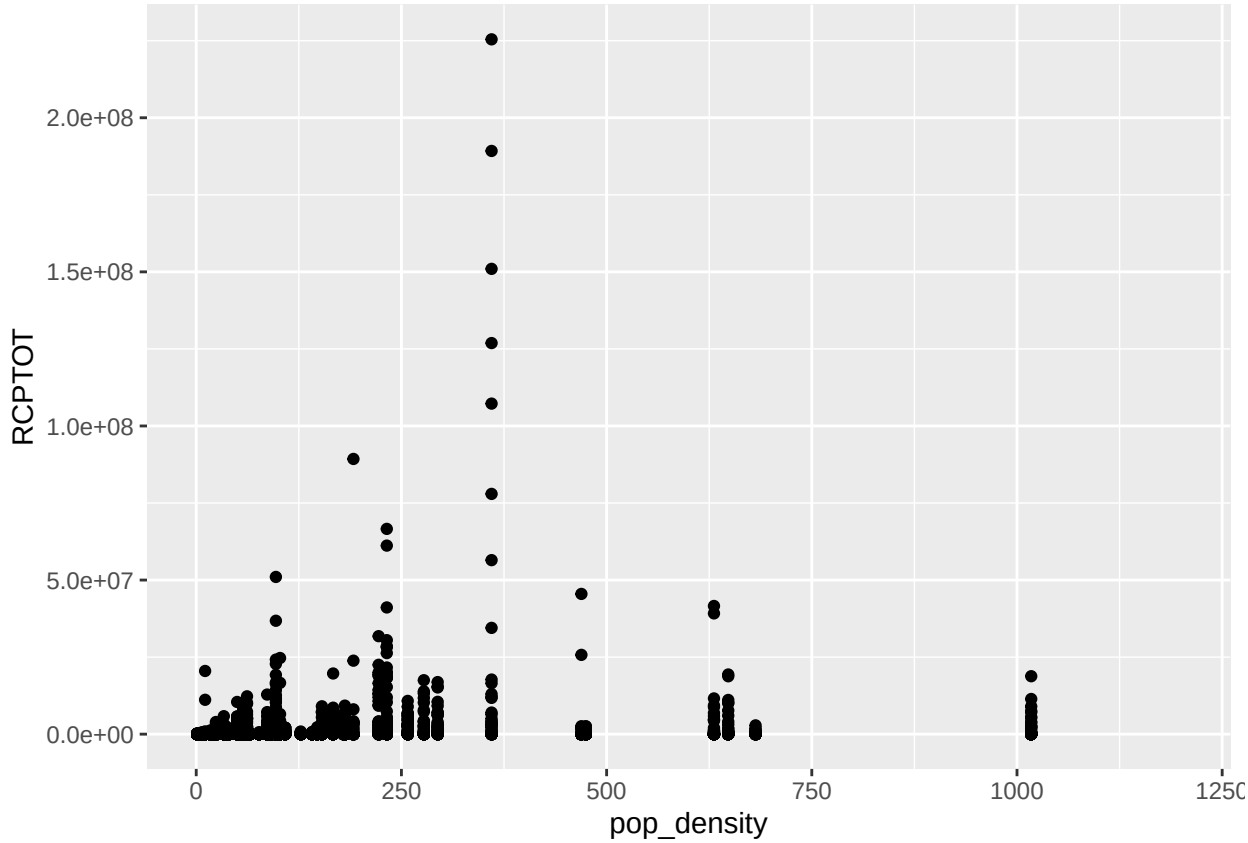
```
rev_pop_data <- left_join(merged_pop, total_rev, by ="state")

ggplot(rev_pop_data, aes(pop_density, RCPTOT))+
  geom_point()
```



The pop density outlier >7500 is making the relationship difficult to analyze. The x limits will therefore be changed for better visualization.

```
#Changing x limits to 0~1200
ggplot(rev_pop_data, aes(pop_density, RCPTOT))+
  geom_point()+
  xlim(0, 1200)
```

```
## Warning: Removed 56 rows containing missing values (geom_point).
```



Fromt this graph there doesn't seem to be a strong linear relationship between population density and total revenue. The overall total revenue does increase as population density goes from 0 to 375, but there are many points in between that don't follow the trend. After 375, the total revenue appears to decrease as population density increases to ~1000. It also seems that at a population denisty around 375, the total revenue reaches its highest value.

#Problem 2

```
load("~/Documents/Data/PSCI Data/nes.rda")
```

#2.1

```
#Convert the rda to a dataframe to extract the columns
nes <- as.data.frame(nes)

#Code for those who voted is 1
#Extract the number of people who voted in the 2008 election
voted <- sum(nes$interest_voted2008==1)
```

```
#Code for those who voted for Obama is 1
voted_obama<- sum(nes$interest_whovote2008==1)
voted_obama/voted
```

```
## [1] 0.6732312
```

67.11% of respondents voted for Barack Obama,

#2.2 Filtered the scores to be greater than/equal to 0 and less than/equal to 100.

```r
nes<- nes%>%
  filter(ftgr_fedgov>=0 & ftgr_fedgov<=100)
unique(nes$ftgr_fedgov)
```

```
##  [1]  85 100  60  70  25  40  30  50  15   0  10  80  65  55  45   4  90   5  35
## [20]  20  75  58  61  59   6  95   7  47  51
```

```r
mean(nes$ftgr_fedgov)
```

```
## [1] 52.48652
```

The average feeling thermometer score is 52.49.

```r
#In the 'prevote_regpty' variable, Democrats coded as 1 and Republicans coded as 2
nes$Demo_or_Repub <- "NA"
nes$Demo_or_Repub[nes$prevote_regpty==1]<-"Democratic"
nes$Demo_or_Repub[nes$prevote_regpty==2]<- "Republican"
```

```r
demo<-mean(nes$ftgr_fedgov[nes$Demo_or_Repub=="Democratic"])
repub<-mean(nes$ftgr_fedgov[nes$Demo_or_Repub=="Republican"])
demo
```

```
## [1] 59.57834
```

```r
repub
```

```
## [1] 40.29787
```

```r
demo-repub
```

```
## [1] 19.28046
```

The difference in mean feeling thermometer score of Democrats compared to Repubicans is 19.28. This means that on average Democratic respondents feel more warmly towards the federal goverment in Washington by 19.28 points than Republican respondents feel towards the federal government.

#Problem 3

```r
#Randomly draw a birthday between 1 and 365 five hundred times
birthdays<-sample(365,500,replace=TRUE)

#check if no one has a Jan 1 birthday
#gives 0 if Jan 1 birthday, 1 if no Jan 1 birthday
as.numeric(!1%in%birthdays)
```

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

```r
#vector that stores whether there is no Jan 1 birthday
notfirst_vec<-c()
for(i in 1:1000){
  birthdays<-sample(365,500,replace=TRUE)
  notfirst<- as.numeric(!1%in%birthdays)
  notfirst_vec<-append(notfirst_vec, notfirst)
}
mean(notfirst_vec)
```

```
## [1] 0.247
```

There is a 24.9% probability that in a group of 500, no one has a birthday on January 1.

```r
prob_notfirst <- function(group_size){
  notfirst_vec<-c()
  for(i in 1:1000){
  birthdays<-sample(365,group_size,replace=TRUE)
  notfirst<- as.numeric(!1%in%birthdays)
  notfirst_vec<-append(notfirst_vec, notfirst)
  }
  return(mean(notfirst_vec))
}
prob_notfirst(750)
```

```
## [1] 0.117
```

There is a 12.4% probability that in a group of 750, no one has a birthday on January 1.

#Problem 4

The P-value should not be used as the sole determinator of a good model/hypothesis. Additionally, it is NOT the probability that the hypothesis being tested is true. It is simply the probability of obtaining an effect as big or bigger than the currently observed effect, given that the null hypothesis is true.

It is also wrong to assume that the P-value indicates the percent chance that the null hypothesis is true. Statements like a P-value of 0.06 indicates the hypothesis has a 6% chance of being true or a P-value of 0.0001 indicates the null hypothesis being true is one in 10,000 are both false.

Also the critical level of 0.05 is arbitrary. For example, one can choose to assign the threshold to 0.01 or 0.10 based on how strict they want their test to be.

A smaller P-value does not indicate a greater confidence in the conclusion. This is because having a larger sample size naturally causes the P-value to become smaller, and therefore a small P-value may be possible for something that is actually not meaningful. It is therefore important to consider effect sizes when evaluating P-values.

#Problem 5

Overfitting is when a model becomes fitted almost exactly to its training data, resulting in it also fitting the noise within the data. This occurs especially for more flexible models because they will model the nuances in the data. The problem with this is that the model can no longer be generalized to new sets of data.

Overfitting is not as big a problem for normal linear models as they are less flexible and won't be able to model the nuanced noise as well. They are not good at handling nonlinearities and so will mainly model the dominant relationship in the data.

To minimize overfitting, a solution is to expand the training data with more clean, relevant data so that the noise is less prominent. Having a validation dataset to crosscheck the model after it's been fitted to the training dataset is another way of evaluating the model's performance on new, unseen data. To do this, one can split the data in half –> train on the first half to learn the parameters –> test the data on second half. One more method is early stopping, where the training process is stopped at a certain iteration before the model can start fitting the noise. When using this method though, one must also be careful not to halt the process too early as that may result in underfitting.

#Problem 6

I expect the normal linear model and logit model to produce slightly different results. A normal linear model will lead to predictions outside the 0-1 range. The regression line will be straight so it will not capture the difference in marignal effects as the number of games played increases. A logit model will stay within the 0-1 range and have a S-shaped regression curve that captures marginal effects. It is suitable for when the dependent variable is dichotomous.

I expect the logit and normal linear model to produce more similar results for the Sixers as the binary values for that team is more balanced, since they have won ~60% of their games. When the probabilities are in the middle range, the linear and logit model will look very similar. However when an event is very low/high probability, the logit model will produce different results as the curve is really different from the linear model. The Phillies have only won 10% of games so that means only 10% of the data will be 1s (imbalanced binary data). Therefore, the logit model for this team will look different from the linear model.

#Problem 7

In this problem, level of coffee consumption is the independent variable and level of happiness is the dependent variable.

Time and weather of that day should be variables that are conditioned on. This is because they are confounding variables that have an influence on both the DV and IV. It's reasonable to assume that the time of day and weather (whether its sunny or cloudy out) can influence the happiness level of a person. And like the question says, the individual likes to drink coffee early in the morning when it is cool outside, meaning that these variables also drive the IV. They are pretreatment covariates that can be measured before treatment is assigned, and therefore should be conditioned on.

Level of caffeine should not be conditioned on because it is something happening after/driven by the IV. It depends on how much coffee the person is consuming, and therefore is NOT a pretreatment covariate.

Whether other people are nice on the SEPTA should not be conditioned on because it is not a confounding variable. It does not influence both the IV and DV.