Problem Set 4

Due April 2, 10:00 AM (Before Class)

Instructions

1. The following questions should each be answered within an R script. Be sure to provide many comments in the script to facilitate grading. Undocumented code will not be graded.
2. Work on git. Fork the repository found at https://github.com/domlockett/PDS-PS3 and add your code, committing and pushing frequently. Use meaningful commit messages – these may affect your grade.
3. You may work in teams, but each student should develop their own R script. To be clear, there should be no copy and paste. Each keystroke in the assignment should be your own.
4. If you have any questions regarding the Problem Set, contact the TAs or use their office hours.
5. For students new to programming, this may take a while. Get started.
6. You will need to install `ggplot2` and `dplyr` to complete this dataset.

Statistics

1. Load the following data: http://politicaldatascience.com/PDS/Datasets/GSS-data.csv. \

   The variable `poleff11` asks participants to rate their level of agreement with the statement "People like me don't have any say about what the government does" (see the codebook for more information on all variables in this dataset at: http://politicaldatascience.com/PDS/Datasets/gss_codebook.csv).

2. Convert this variable into a numeric where higher values indicate higher levels of political efficacy (1- strongly agrees with the statement; 5- strongly disagrees with the statment)

   ```r
   library(tidyverse)
   ```

   ```
   ## -- Attaching packages ---- tidyverse 1.3.0 --
   ```

   ```
   ## v ggplot2 3.2.1     v purrr   0.3.3
   ## v tibble  2.1.3     v dplyr   0.8.3
   ## v tidyr   1.0.0     v stringr 1.4.0
   ## v readr   1.3.1     v forcats 0.4.0
   ```

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

   ```r
   data <- read_csv('http://politicaldatascience.com/PDS/Datasets/GSS-data.csv')
   ```

   ```
   ## Parsed with column specification:
   ## cols(
   ##   .default = col_character(),
   ##   id_ = col_double()
   ## )
   ```

   ```
   ## See spec(...) for full column specifications.
   ```

   ```r
   data$poleff <- recode(data$poleff11, "Strongly agree"= 1, "Agree" = 2, 'Neither agree nor disagree'
   ```

3. What is the proportion of individuals from the entire sample who do not feel as though they have a say in the government? The proportion of those who do feel as though they have a say?

```r
sum(data$poleff < 3, na.rm=T)/sum(is.na(data$poleff)==F)
```

```
[1] 0.4118158
```

```r
sum(data$poleff > 3, na.rm=T)/sum(is.na(data$poleff)==F)
```

```
[1] 0.3953084
```

4. Using a sample of **25** what is the average level of efficacy individuals feel on the one to five scale? At 100? 500?

```r
mean(sample(data$poleff,size=25, replace =F), na.rm =T)
```

```
[1] 2.923077
```

```r
mean(sample(data$poleff,size=100, replace =F), na.rm =T)
```

```
[1] 2.836364
```

```r
mean(sample(data$poleff,size=500, replace =F), na.rm =T)
```

```
[1] 2.850806
```

5. Create a variable called `trials_25` where we pull a random sample of 25 from the `poleff11` data and calculate the mean 500 times .

```r
library(mosaic)
```

```
## Loading required package: lattice

## Loading required package: ggformula

## Loading required package: ggstance

##
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':
##
##     geom_errorbarh, GeomErrorbarh

##
## New to ggformula?  Try the tutorials:
##  learnr::run_tutorial("introduction", package = "ggformula")
##  learnr::run_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack

## Registered S3 method overwritten by 'mosaic':
##   method                           from
##   fortify.SpatialPolygonsDataFrame ggplot2

##
## The 'mosaic' package masks several functions from core packages in order to add
## additional features.  The original behavior of these functions should not be affected by this.
```

```
##
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.
##
## Have you tried the ggformula package for your plots?

##
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':
##
##     mean

## The following objects are masked from 'package:dplyr':
##
##     count, do, tally

## The following object is masked from 'package:purrr':
##
##     cross

## The following object is masked from 'package:ggplot2':
##
##     stat

## The following objects are masked from 'package:stats':
##
##     binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
##     quantile, sd, t.test, var

## The following objects are masked from 'package:base':
##
##     max, mean, min, prod, range, sample, sum
```

```
trials <- do(500)*mean(sample(data$poleff,size=25, replace =F), na.rm =T)
```

6. Now create a variable called `trials_100` where we do 500 trials with a sample of 100 instead of 25. Draw a histogram of the sampling distribution fo the trials for the two trials you just conducted. Save these plots in your repository.

   ```
   trials <- do(1000)*mean(sample(data$poleff,size=100, replace =F), na.rm =T)
   ```

7. What notable difference occur when we use a larger sample size in our trials?

Bootstrapping

(a) **Run the command `set.seed(25)`.** Create a new variable \texttt{eff_subset} which draws a sample of 300 observations from the original `poleff11` data.

```
set.seed(25)
eff_subset <- sample(data$poleff, size = 300, replace =F, na.rm=T)
```

- Draw a sample of 25 from \texttt{eff_subset} and calculate the mean. Repeat this task 500 times and store the results in a variable named `trials`.

  ```
  trials <- do(500)*mean(sample(eff_subset,size=25, replace =F), na.rm =T)
  ```

- What is the mean of the 500 trials? 1000 trials? How do these values compare to the mean of the actual `poleff11` variable?

```r
trials2 <- do(1000)*mean(sample(eff_subset,size=25, replace =F), na.rm =T)
mean(trials)
```

```
[1] NA
```

```r
mean(trials2)
```

```
[1] NA
```

```r
mean(data$poleff, na.rm=T)
```

```
[1] 2.908775
```

- Notice that with pulling random samples of only 25 observations from **a quarter of the data** we are able to accurately approximate the mean of our entire data. That is the logic of bootstrapping, with only a small sample (here our population is the entire dataset), one can make reasonable approximations about the population!

Supervised Learning

2. -
   -
   -
   -
   -
   -
   -
   -
   -
   -
3. -
   -
   -