Introduction to Data Science HW 4


```
# Enter your name here:  Ryan Tervo
# Course Number:         IST 687
# Assignement Name:      Homework #4
# Due Date:              07 Nov 2022
# Submitted Date:        07 Nov 2022
```

Attribution statement: (choose only one and delete the rest)

```
# 1. I did this homework by myself, with help from the book and the professor.
```

Reminders of things to practice from previous weeks:
Descriptive statistics: mean( ) max( ) min( )
Coerce to numeric: as.numeric( )

Part 1: Use the Starter Code

Below, I have provided a starter file to help you.

Each of these lines of code **must be commented** (the comment must that explains what is going on, so that I know you understand the code and results).

```
library(jsonlite)
#  This loads the library jsonlite.  It is necessary in order to utilize jsonlite's functions.
#  This assumes that the jsonlite has been installed prior.  Installation is only required onc
e and then can be accessed using the 'library' command.

dataset <- url("https://intro-datascience.s3.us-east-2.amazonaws.com/role.json")
#  This downloads dataset from the url provided.  This data is set to the variable 'dataset'.
#  Based on later use and the fact that the url ends in 'json' the dataset being read is in th
e json format.

readlines <- jsonlite::fromJSON(dataset)
#  This call the function 'fromJSON' from the library 'jsonlite' to read in the json data stor
ed in the variable dataset.
#  call a function from jsonlite to read the json data that's stored in dataset.  Sets it to v
ariable  'readlines'

df <- readlines$objects$person
#   This stores the 'person' portion of the json data and sets it to the variable 'df'
```

A. Explore the **df** dataframe (e.g., using head() or whatever you think is best).

```
#   EXPLORE THE df DATAFRAME:
str(df)
```

```
## 'data.frame':    100 obs. of  17 variables:
##  $ bioguideid  : chr  "C000880" "G000386" "L000174" "M001153" ...
##  $ birthday    : chr  "1951-05-20" "1933-09-17" "1940-03-31" "1957-05-22" ...
##  $ cspanid     : int  26440 1167 1552 1004138 25277 5929 1859 1962 45465 92069 ...
##  $ firstname   : chr  "Michael" "Charles" "Patrick" "Lisa" ...
##  $ gender      : chr  "male" "male" "male" "female" ...
##  $ gender_label: chr  "Male" "Male" "Male" "Female" ...
##  $ lastname    : chr  "Crapo" "Grassley" "Leahy" "Murkowski" ...
##  $ link        : chr  "https://www.govtrack.us/congress/members/michael_crapo/300030" "http
s://www.govtrack.us/congress/members/charles_grassley/300048" "https://www.govtrack.us/congres
s/members/patrick_leahy/300065" "https://www.govtrack.us/congress/members/lisa_murkowski/30007
5" ...
##  $ middlename  : chr  "D." "E." "J." "A." ...
##  $ name        : chr  "Sen. Michael "Mike" Crapo [R-ID]" "Sen. Charles "Chuck" Grassley [R-
IA]" "Sen. Patrick Leahy [D-VT]" "Sen. Lisa Murkowski [R-AK]" ...
##  $ namemod     : chr  "" "" "" "" ...
##  $ nickname    : chr  "Mike" "Chuck" "" "" ...
##  $ osid        : chr  "N00006267" "N00001758" "N00009918" "N00026050" ...
##  $ pvsid       : chr  "26830" "53293" "53353" "15841" ...
##  $ sortname    : chr  "Crapo, Michael "Mike" (Sen.) [R-ID]" "Grassley, Charles "Chuck" (Sen
.) [R-IA]" "Leahy, Patrick (Sen.) [D-VT]" "Murkowski, Lisa (Sen.) [R-AK]" ...
##  $ twitterid   : chr  "MikeCrapo" "ChuckGrassley" "SenatorLeahy" "LisaMurkowski" ...
##  $ youtubeid   : chr  "senatorcrapo" "senchuckgrassley" "SenatorPatrickLeahy" "senatormurko
wski" ...
```

```
summary(df)
```

```
##   bioguideid          birthday            cspanid         firstname
##  Length:100         Length:100         Min.   :    260   Length:100
##  Class :character   Class :character   1st Qu.:  25277   Class :character
##  Mode  :character   Mode  :character   Median :  68489   Mode  :character
##                                        Mean   : 584001
##                                        3rd Qu.:1004138
##                                        Max.   :9269028
##                                        NA's   :11
##     gender          gender_label         lastname             link
##  Length:100         Length:100         Length:100         Length:100
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   middlename            name             namemod            nickname
##  Length:100         Length:100         Length:100         Length:100
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##      osid              pvsid             sortname          twitterid
```

```
##  Length:100         Length:100         Length:100         Length:100
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##
##   youtubeid
##  Length:100
##  Class :character
##  Mode  :character
##
##
##
##
```

```
head(df)
```

```
##   bioguideid   birthday cspanid firstname gender gender_label  lastname
## 1    C000880 1951-05-20   26440   Michael   male         Male     Crapo
## 2    G000386 1933-09-17    1167   Charles   male         Male  Grassley
## 3    L000174 1940-03-31    1552   Patrick   male         Male     Leahy
## 4    M001153 1957-05-22 1004138      Lisa female       Female Murkowski
## 5    M001111 1950-10-11   25277     Patty female       Female    Murray
## 6    S000148 1950-11-23    5929   Charles   male         Male   Schumer
##                                                                link middlename
## 1    https://www.govtrack.us/congress/members/michael_crapo/300030         D.
## 2 https://www.govtrack.us/congress/members/charles_grassley/300048         E.
## 3    https://www.govtrack.us/congress/members/patrick_leahy/300065         J.
## 4   https://www.govtrack.us/congress/members/lisa_murkowski/300075         A.
## 5     https://www.govtrack.us/congress/members/patty_murray/300076
## 6  https://www.govtrack.us/congress/members/charles_schumer/300087         E.
##                                  name namemod nickname      osid pvsid
## 1     Sen. Michael “Mike” Crapo [R-ID]             Mike N00006267 26830
## 2 Sen. Charles “Chuck” Grassley [R-IA]            Chuck N00001758 53293
## 3            Sen. Patrick Leahy [D-VT]                  N00009918 53353
## 4           Sen. Lisa Murkowski [R-AK]                  N00026050 15841
## 5             Sen. Patty Murray [D-WA]                  N00007876 53358
## 6  Sen. Charles “Chuck” Schumer [D-NY]            Chuck N00001093 26976
##                                  sortname     twitterid           youtubeid
## 1     Crapo, Michael “Mike” (Sen.) [R-ID]     MikeCrapo        senatorcrapo
## 2 Grassley, Charles “Chuck” (Sen.) [R-IA] ChuckGrassley    senchuckgrassley
## 3            Leahy, Patrick (Sen.) [D-VT]  SenatorLeahy SenatorPatrickLeahy
## 4           Murkowski, Lisa (Sen.) [R-AK] LisaMurkowski    senatormurkowski
## 5             Murray, Patty (Sen.) [D-WA]   PattyMurray  SenatorPattyMurray
## 6  Schumer, Charles “Chuck” (Sen.) [D-NY]    SenSchumer      SenatorSchumer
```

```
numRow = nrow(df)
numCol = ncol(df)

print(paste('There are ', numRow, ' rows and ', numCol, ' columns.', sep = ''), quote = F)
```

```
## [1] There are 100 rows and 17 columns.
```

```
#   I like looking a the structure and the summary of the data.
#   The head function provide a quick visual of the top 5 rows of information.
```

B. Explain the dataset
- o What is the dataset about?
- o How many rows are there and what does a row represent?
- o How many columns and what does each column represent?

```
## Part B Answer:
#-------------------------------------------------------------
#The Dataset
#- The dataset contains information of 100 members of the U.S. Senate.
#- There are 100 rows.  Each row represents one Senator.
#- There are 17 columns.  Each column contains a feature or attribute concerning the respectiv
e Sentator.
```

C. What does running this line of code do? Explain in a comment:

```
vals <- substr(df$birthday, 1, 4)

#   PART C ANSWER:
#   ------------------------------------------------------------------------------------------
#   This created a vector of strings of the first four characters in the df%birthday column.
#   Due to the way the birthday column is formated the first four characters are the birthday
year.
```

D. Create a new attribute ‘age’ - how old the person is **Hint:** You may need to convert it to numeric first.

```
#   CREATE THE ATTRIBUTE 'age'
df$age <- c(1:nrow(df))*0

#   DISPLAY THE RESULTS:
head(df)
```

```
##   bioguideid   birthday cspanid firstname gender gender_label  lastname
## 1    C000880 1951-05-20   26440   Michael   male         Male     Crapo
## 2    G000386 1933-09-17    1167   Charles   male         Male  Grassley
## 3    L000174 1940-03-31    1552   Patrick   male         Male     Leahy
## 4    M001153 1957-05-22 1004138      Lisa female       Female Murkowski
## 5    M001111 1950-10-11   25277     Patty female       Female    Murray
## 6    S000148 1950-11-23    5929   Charles   male         Male   Schumer
##                                                                link middlename
## 1    https://www.govtrack.us/congress/members/michael_crapo/300030         D.
## 2 https://www.govtrack.us/congress/members/charles_grassley/300048         E.
## 3    https://www.govtrack.us/congress/members/patrick_leahy/300065         J.
## 4   https://www.govtrack.us/congress/members/lisa_murkowski/300075         A.
## 5     https://www.govtrack.us/congress/members/patty_murray/300076
## 6  https://www.govtrack.us/congress/members/charles_schumer/300087         E.
##                                 name namemod nickname      osid pvsid
```

```
## 1     Sen. Michael "Mike" Crapo [R-ID]             Mike N00006267 26830
## 2 Sen. Charles "Chuck" Grassley [R-IA]            Chuck N00001758 53293
## 3            Sen. Patrick Leahy [D-VT]                  N00009918 53353
## 4          Sen. Lisa Murkowski [R-AK]                   N00026050 15841
## 5            Sen. Patty Murray [D-WA]                   N00007876 53358
## 6  Sen. Charles "Chuck" Schumer [D-NY]            Chuck N00001093 26976
##                                 sortname     twitterid           youtubeid age
## 1     Crapo, Michael "Mike" (Sen.) [R-ID]     MikeCrapo        senatorcrapo   0
## 2 Grassley, Charles "Chuck" (Sen.) [R-IA] ChuckGrassley    senchuckgrassley   0
## 3            Leahy, Patrick (Sen.) [D-VT]  SenatorLeahy SenatorPatrickLeahy   0
## 4          Murkowski, Lisa (Sen.) [R-AK] LisaMurkowski    senatormurkowski   0
## 5            Murray, Patty (Sen.) [D-WA]   PattyMurray  SenatorPattyMurray   0
## 6  Schumer, Charles "Chuck" (Sen.) [D-NY]    SenSchumer      SenatorSchumer   0
```

```
#   Note:  I initially created the attribute age and populated it with the age field using the
 line below.
#       df$age <- as.integer(floor((Sys.Date() - as.Date(df$birthday))/365.25))
#   After seeing Part E I decided to remove the actual values and replace them with zeros.
```

E. Create a function that reads in the role json dataset, and adds the age attribute to the dataframe, and returns that dataframe

```
#   DEFINE FUNCTION:
addAge <- function(df){
    df$age <- as.integer(floor((Sys.Date() - as.Date(df$birthday))/365.25)) #  the .25 account
s for leap years.
    return(df)
}
```

F. Use (call, invoke) the function, and store the results in df

```
#    USE FUNCTION WITH DF:
df <- addAge(df)

#   DISPLAY RESULTS:
head(df)
```

```
##   bioguideid   birthday cspanid firstname gender gender_label  lastname
## 1    C000880 1951-05-20   26440   Michael   male         Male     Crapo
## 2    G000386 1933-09-17    1167   Charles   male         Male  Grassley
## 3    L000174 1940-03-31    1552   Patrick   male         Male     Leahy
## 4    M001153 1957-05-22 1004138      Lisa female       Female Murkowski
## 5    M001111 1950-10-11   25277     Patty female       Female    Murray
## 6    S000148 1950-11-23    5929   Charles   male         Male   Schumer
##                                                                 link middlename
## 1    https://www.govtrack.us/congress/members/michael_crapo/300030         D.
## 2 https://www.govtrack.us/congress/members/charles_grassley/300048         E.
## 3    https://www.govtrack.us/congress/members/patrick_leahy/300065         J.
## 4   https://www.govtrack.us/congress/members/lisa_murkowski/300075         A.
## 5     https://www.govtrack.us/congress/members/patty_murray/300076
## 6  https://www.govtrack.us/congress/members/charles_schumer/300087         E.
```

```
##                                    name namemod nickname      osid pvsid
## 1     Sen. Michael "Mike" Crapo [R-ID]              Mike N00006267 26830
## 2 Sen. Charles "Chuck" Grassley [R-IA]             Chuck N00001758 53293
## 3            Sen. Patrick Leahy [D-VT]                   N00009918 53353
## 4          Sen. Lisa Murkowski [R-AK]                   N00026050 15841
## 5            Sen. Patty Murray [D-WA]                   N00007876 53358
## 6  Sen. Charles "Chuck" Schumer [D-NY]             Chuck N00001093 26976
##                                  sortname     twitterid           youtubeid age
## 1     Crapo, Michael "Mike" (Sen.) [R-ID]     MikeCrapo        senatorcrapo  71
## 2 Grassley, Charles "Chuck" (Sen.) [R-IA] ChuckGrassley    senchuckgrassley  89
## 3            Leahy, Patrick (Sen.) [D-VT]  SenatorLeahy SenatorPatrickLeahy  82
## 4          Murkowski, Lisa (Sen.) [R-AK] LisaMurkowski    senatormurkowski  65
## 5            Murray, Patty (Sen.) [D-WA]   PattyMurray  SenatorPattyMurray  72
## 6  Schumer, Charles "Chuck" (Sen.) [D-NY]    SenSchumer      SenatorSchumer  72
```

Part 2: Investigate the resulting dataframe 'df'

A. How many senators are women?

```
#   GET NECESSARY DATA:
tempGender <- df$gender

#   GET VALUE:  Create vector of TRUE/FALSE and sum.  TRUE will equal 1 and FALSE will equal 0
.
countWoman <- sum(tempGender == 'female')

#   DISPLAY RESULTS:
printString <- paste('According to the JSON list, there are ', countWoman, ' women Senators.',
 sep = "")
print(printString, quote = F)
```

```
## [1] According to the JSON list, there are 24 women Senators.
```

B. How many senators have a YouTube account?

```
#   GET NECESSARY DATA:
tempYoutube <- df$youtubeid

#   GET VALUE:    Create vector of TRUE/FALSE and sum.  TRUE will equal 1 and FALSE will equal
 0.
countYoutube <- sum(! is.na(tempYoutube))

#   DISPLAY RESULTS:
printString <- paste('According to the JSON list, there are ', countYoutube, ' Senators with a
 YouTube account.', sep = "")
print(printString, quote = F)
```

```
## [1] According to the JSON list, there are 73 Senators with a YouTube account.
```

C. How many women senators have a YouTube account?

```
#   GET NECESSARY DATA:
tempDF <- df[df$gender == 'female' & !is.na(df$youtubeid), ]

#   GET VALUE:
numWoman_with_youtube <- nrow(tempDF)

#   DISPLAY RESULTS:
printString <- paste('According to the JSON list, there are ', numWoman_with_youtube, ' female
 Senators with a YouTube account.', sep = "")
print(printString, quote = F)
```

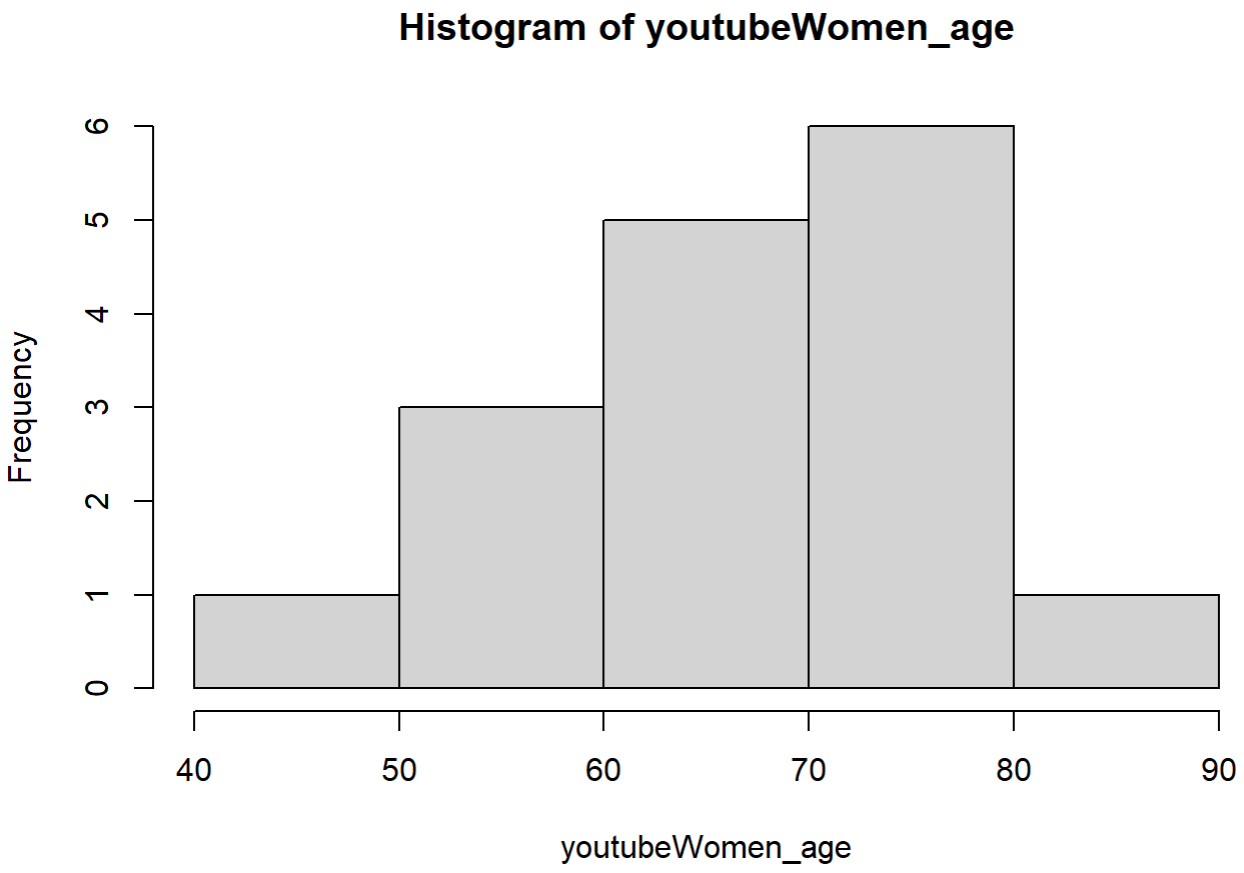
```
## [1] According to the JSON list, there are 16 female Senators with a YouTube account.
```

D. Create a new dataframe called **youtubeWomen** that only includes women senators who have a YouTube account.

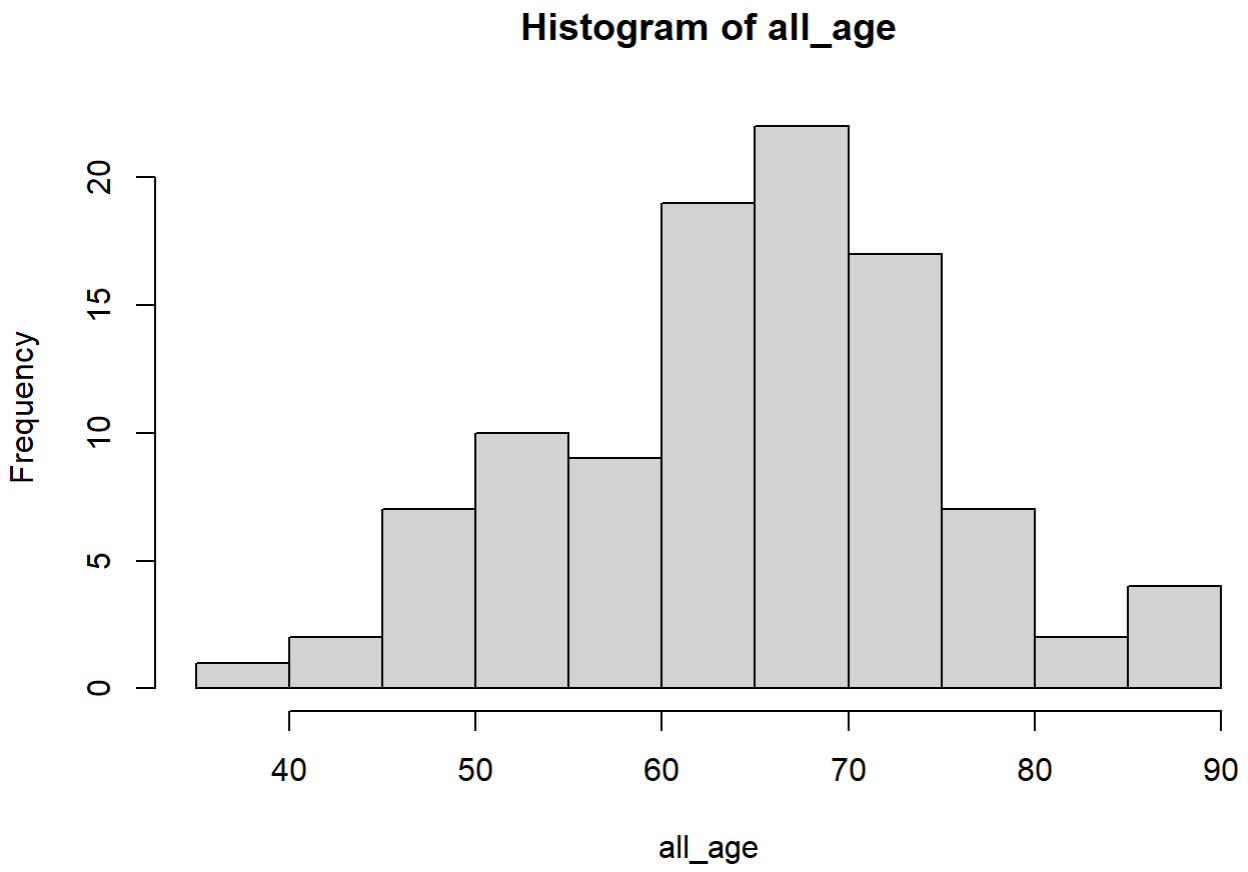
```
#   CREATE:  df, youtubeWomen
youtubeWomen <- df[df$gender == 'female' & !is.na(df$youtubeid), ]
```

E. Make a histogram of the **age** of senators in **youtubeWomen**, and then another for the senetors in **df**. Add a comment describing the shape of the distributions.

```
#   CREATE A HISTOGRAM OF YOUTUBEWOMEN_AGE:
youtubeWomen_age <- youtubeWomen$age
hist(youtubeWomen_age)
```



```
#   CREATE A HISTOGRAM OF ALL_AGE:
all_age <- df$age
hist(all_age)
```




```
#   The histogram for youtubewomen appears to be nearly linearly increasing until age bracket
70-80 and then drops off.
#   The histogram for all looks nearly normal with a peak at the age bracket of 65-70.
```