DS hw2

Yunxi Zhang

10/9/2021

```
library(tidyverse)
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1

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

```
library(readxl)
library(lubridate)
```

```
##
## Attaching package: 'lubridate'

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

Problem 1

This problem uses the Mr. Trash Wheel dataset, available as an Excel file on the course website.

Read and clean the Mr. Trash Wheel sheet:

Specify the sheet in the Excel file and to omit non-data entries (rows with notes / figures; columns containing notes) using arguments in read_excel use reasonable variable names

omit rows that do not include dumpster-specific data

round the number of sports balls to the nearest integer

```
Trash_df =
  read_excel("./data/Trash-Wheel-Collection-Totals-7-2020-2.xlsx", sheet =   "Mr. Trash Wheel") %>%
  janitor::clean_names() %>%
  select(-x15, -x16, -x17) %>%
  drop_na() %>%  ## omit rows with NA
  mutate(sports_balls = round(sports_balls)) ## round sports balls to integer
```

```
## New names:
## * `` -> ...15
## * `` -> ...16
## * `` -> ...17
```

```
head(Trash_df)
```

```
## # A tibble: 6 x 14
##   dumpster month  year date                weight_tons volume_cubic_yards
##   <chr>    <chr> <dbl> <dttm>                    <dbl>              <dbl>
## 1 1        May    2014 2014-05-16 00:00:00        4.31                 18
## 2 2        May    2014 2014-05-16 00:00:00        2.74                 13
## 3 3        May    2014 2014-05-16 00:00:00        3.45                 15
## 4 4        May    2014 2014-05-17 00:00:00        3.1                  15
## 5 5        May    2014 2014-05-17 00:00:00        4.06                 18
## 6 6        May    2014 2014-05-20 00:00:00        2.71                 13
## # ... with 8 more variables: plastic_bottles <dbl>, polystyrene <dbl>,
## #   cigarette_butts <dbl>, glass_bottles <dbl>, grocery_bags <dbl>,
## #   chip_bags <dbl>, sports_balls <dbl>, homes_powered <dbl>
```

Read and clean precipitation data for 2018 and 2019.

For each, omit rows without precipitation data and add a variable for year.

```
precip2018_df =
  read_excel("./data/Trash-Wheel-Collection-Totals-7-2020-2.xlsx",
             sheet = "2018 Precipitation", skip = 1) %>%
  janitor::clean_names() %>%
  drop_na() %>%
  mutate(year = 2018)

precip2018_df
```

```
## # A tibble: 12 x 3
##    month total  year
##    <dbl> <dbl> <dbl>
##  1     1  0.94  2018
##  2     2  4.8   2018
##  3     3  2.69  2018
##  4     4  4.69  2018
##  5     5  9.27  2018
##  6     6  4.77  2018
##  7     7 10.2   2018
```

```
##  8     8  6.45  2018
##  9     9 10.5   2018
## 10    10  2.12  2018
## 11    11  7.82  2018
## 12    12  6.11  2018
```

```
precip2019_df =
  read_excel("./data/Trash-Wheel-Collection-Totals-7-2020-2.xlsx", sheet = "2019 Precipitation", skip =
  janitor::clean_names() %>%
  drop_na() %>%
  mutate(year = 2019)

precip2019_df
```

```
## # A tibble: 12 x 3
##    month total  year
##    <dbl> <dbl> <dbl>
##  1     1  3.1   2019
##  2     2  3.64  2019
##  3     3  4.47  2019
##  4     4  1.46  2019
##  5     5  3.58  2019
##  6     6  0.42  2019
##  7     7  3.85  2019
##  8     8  2.39  2019
##  9     9  0.16  2019
## 10    10  5.45  2019
## 11    11  1.86  2019
## 12    12  3.57  2019
```

Next, combine precipitation datasets and convert month to a character variable (the variable month.name is built into R and should be useful).

```
precip_df =
  full_join(precip2018_df, precip2019_df) %>%
  arrange(year, month) %>%
  mutate(month = month.name[month])
```

```
## Joining, by = c("month", "total", "year")
```

```
precip_df
```

```
## # A tibble: 24 x 3
##    month     total  year
##    <chr>     <dbl> <dbl>
##  1 January    0.94  2018
##  2 February   4.8   2018
##  3 March      2.69  2018
##  4 April      4.69  2018
##  5 May        9.27  2018
```

```
##  6 June       4.77  2018
##  7 July      10.2   2018
##  8 August     6.45  2018
##  9 September 10.5   2018
## 10 October    2.12  2018
## # ... with 14 more rows
```

Write a paragraph about these data; you are encouraged to use inline R. Be sure to note the number of observations in both resulting datasets, and give examples of key variables. For available data, what was the total precipitation in 2018? What was the median number of sports balls in a dumpster in 2019?

Solution:

The Mr.Trash Wheel dataset has `rncol(Trash_data)` variables and `rnrows(Trash_data)` observations. The data collected different type of trash in several dumpsters through 2014 to 2016. Trash types include plastic_bottles, polystyrene, cigarette_butts, glass_bottles, grocery_bags, chip_bags, sports_balls.

```
## Calculate the mean of each type of trash
Trash_df %>%
  mutate(mean = Trash_df)
```

```
## # A tibble: 453 x 15
##    dumpster month  year date                weight_tons volume_cubic_yards
##    <chr>    <chr> <dbl> <dttm>                    <dbl>              <dbl>
##  1 1        May    2014 2014-05-16 00:00:00        4.31                 18
##  2 2        May    2014 2014-05-16 00:00:00        2.74                 13
##  3 3        May    2014 2014-05-16 00:00:00        3.45                 15
##  4 4        May    2014 2014-05-17 00:00:00        3.1                  15
##  5 5        May    2014 2014-05-17 00:00:00        4.06                 18
##  6 6        May    2014 2014-05-20 00:00:00        2.71                 13
##  7 7        May    2014 2014-05-21 00:00:00        1.91                  8
##  8 8        May    2014 2014-05-28 00:00:00        3.7                  16
##  9 9        June   2014 2014-06-05 00:00:00        2.52                 14
## 10 10       June   2014 2014-06-11 00:00:00        3.76                 18
## # ... with 443 more rows, and 9 more variables: plastic_bottles <dbl>,
## #   polystyrene <dbl>, cigarette_butts <dbl>, glass_bottles <dbl>,
## #   grocery_bags <dbl>, chip_bags <dbl>, sports_balls <dbl>,
## #   homes_powered <dbl>, mean <tibble[,14]>
```

```
numcol_df = select(Trash_df, weight_tons:homes_powered)
colMeans(numcol_df)
```

```
##        weight_tons volume_cubic_yards    plastic_bottles        polystyrene
##           3.200221          15.412804        1898.929360        1920.920530
##    cigarette_butts      glass_bottles       grocery_bags          chip_bags
##       24521.677704          22.452539        1103.823400        1558.397351
##       sports_balls      homes_powered
##          11.746137          45.320088
```

"Cigarette_butts", "polystyrene" and "plastic_bottles" are the top three types of trash. The meadian number of sports balls in a dumpster in 2019 is 9.

The 2018_precipitation dataset has 3 variables and 24 observations. The dataset collected the amount of precipitation for each month in 2018 and 2019. The sum of precipitation of 2018 is `rsum(precip2018$total)`.

Problem 2

This problem uses the FiveThirtyEight data; these data were gathered to create the interactive graphic on this page. In particular, we'll use the data in pols-month.csv, unemployment.csv, and snp.csv. Our goal is to merge these into a single data frame using year and month as keys across datasets.

First, clean the data in pols-month.csv. Use separate() to break up the variable mon into integer variables year, month, and day; replace month number with month name; create a president variable taking values gop and dem, and remove prez_dem and prez_gop; and remove the day variable.

```
pols_df =
  read_csv("./data/fivethirtyeight_datasets/pols-month.csv") %>%
  janitor::clean_names() %>%

  ## break up into year, month, and day
  separate(mon, c("year", "month", "day"), sep = "-") %>%
  # replace month number with month name
  mutate(month = month.name[as.integer(month)],
         year = as.integer(year),
         day = as.integer(day)) %>%

  select(-day, -prez_dem, -prez_gop)
```

```
## Rows: 822 Columns: 9
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl  (8): prez_gop, gov_gop, sen_gop, rep_gop, prez_dem, gov_dem, sen_dem, r...
## date (1): mon
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
pols_df
```

```
## # A tibble: 822 x 8
##     year month     gov_gop sen_gop rep_gop gov_dem sen_dem rep_dem
##    <int> <chr>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl>
##  1  1947 January        23      51     253      23      45     198
##  2  1947 February       23      51     253      23      45     198
##  3  1947 March          23      51     253      23      45     198
##  4  1947 April          23      51     253      23      45     198
##  5  1947 May            23      51     253      23      45     198
##  6  1947 June           23      51     253      23      45     198
##  7  1947 July           23      51     253      23      45     198
##  8  1947 August         23      51     253      23      45     198
##  9  1947 September      23      51     253      23      45     198
## 10  1947 October        23      51     253      23      45     198
## # ... with 812 more rows
```

Second, clean the data in snp.csv using a similar process to the above. For consistency across datasets, arrange according to year and month, and organize so that year and month are the leading columns.

```
snp_df =
  read_csv("./data/fivethirtyeight_datasets/snp.csv") %>%
  janitor::clean_names() %>%
  mutate(date = mdy(date)) %>%
  ## break up into year, month, and day
  separate(date, c("year", "month", "day"), sep = "-") %>%
  # replace month number with month name
  mutate(
    month = month.name[as.integer(month)],
    day = as.integer(day),
    year = as.integer(year)) %>%
  ## arrange according to year and month
  arrange(year, month) %>%
  ## set year and month as leading variable
  relocate(year, month)
```

```
## Rows: 787 Columns: 2
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): date
## dbl (1): close
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
snp_df
```

```
## # A tibble: 787 x 4
##     year month      day close
##    <int> <chr>    <int> <dbl>
##  1  1969 April        1 104.
##  2  1969 August       1  95.5
##  3  1969 December     1  92.1
##  4  1969 February     3  98.1
##  5  1969 January      2 103.
##  6  1969 July         1  91.8
##  7  1969 June         2  97.7
##  8  1969 March        3 102.
##  9  1969 May          1 103.
## 10  1969 November     3  93.8
## # ... with 777 more rows
```

Third, tidy the unemployment data so that it can be merged with the previous datasets. This process will involve switching from "wide" to "long" format; ensuring that key variables have the same name; and ensuring that key variables take the same values.

```
unemp_df <-
  read_csv("./data/fivethirtyeight_datasets/unemployment.csv") %>%
  pivot_longer(cols = 2:13,
               names_to = "month",
               values_to = "unemployment rate") %>%
  janitor::clean_names()
```

```
## Rows: 68 Columns: 13

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (13): Year, Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec

##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
unemp_df
```

```
## # A tibble: 816 x 3
##     year month unemployment_rate
##    <dbl> <chr>              <dbl>
##  1  1948 Jan                  3.4
##  2  1948 Feb                  3.8
##  3  1948 Mar                  4
##  4  1948 Apr                  3.9
##  5  1948 May                  3.5
##  6  1948 Jun                  3.6
##  7  1948 Jul                  3.6
##  8  1948 Aug                  3.9
##  9  1948 Sep                  3.8
## 10  1948 Oct                  3.7
## # ... with 806 more rows
```

Join the datasets by merging snp into pols, and merging unemployment into the result.

```
prob2_df <-
  left_join(pols_df, snp_df, by = c("year", "month")) %>%
  left_join(unemp_df, by = c("year", "month"))

prob2_df
```

```
## # A tibble: 822 x 11
##     year month     gov_gop sen_gop rep_gop gov_dem sen_dem rep_dem   day close
##    <dbl> <chr>       <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <int> <dbl>
##  1  1947 January        23      51     253      23      45     198    NA    NA
##  2  1947 February       23      51     253      23      45     198    NA    NA
##  3  1947 March          23      51     253      23      45     198    NA    NA
##  4  1947 April          23      51     253      23      45     198    NA    NA
##  5  1947 May            23      51     253      23      45     198    NA    NA
##  6  1947 June           23      51     253      23      45     198    NA    NA
##  7  1947 July           23      51     253      23      45     198    NA    NA
##  8  1947 August         23      51     253      23      45     198    NA    NA
##  9  1947 September      23      51     253      23      45     198    NA    NA
## 10  1947 October        23      51     253      23      45     198    NA    NA
## # ... with 812 more rows, and 1 more variable: unemployment_rate <dbl>
```

Write a short paragraph about these datasets. Explain briefly what each dataset contained, and describe the resulting dataset (e.g. give the dimension, range of years, and names of key variables).

The pols dataset has 8 variables and 822 observations. It contains the number of national politicians who are democratic or republican during the time ranging from 1947 to 2015with variables including year, month, gov_gop, sen_gop, rep_gop, gov_dem, sen_dem, rep_dem.

The snp dataset has 4 variables and 787 observations. It contains Standard & Poor's stock market index (S&P) during the time ranging from 1969 to 2068 with variables including: year, month, day, close.

The unemployment dataset has 3 variables and 816 observations. It contains unemployment rate at time ranging from 1948 to 2015 with variables including year, month, unemployment_rate.

The final dataset is merged by the 3 datasets above, which has 11 variables and 822 observations, containing the number of national politicians who are democratic or republican, S&P, and unemployment rate at time ranging from 1947 to 2015.The variables in the dataset are: year, month, gov_gop, sen_gop, rep_gop, gov_dem, sen_dem, rep_dem, day, close, unemployment_rate.

Problem 3

This problem uses data from NYC Open data on the popularity of baby names, and can be downloaded here.

Load and tidy the data. Note that, although these data may seem fairly well formatted initially, the names of a categorical predictor and the case structure of string variables changed over time; you'll need to address this in your data cleaning. Also, some rows seem duplicated, and these will need to be removed (hint: google something like "dplyr remove duplicate rows" to get started).

```
names_df <- read_csv("./data/Popular_Baby_Names.csv") %>%
  janitor::clean_names() %>%
  mutate(
    ethnicity = recode(
      ethnicity,
      "BLACK NON HISP" = "BLACK NON HISPANIC",
      "WHITE NON HISP" = "WHITE NON HISPANIC",
```

```
      "ASIAN AND PACI" = "ASIAN AND PACIFIC ISLANDER"
    )
  ) %>%

  distinct() ## remove duplicate rows
```

Produce a well-structured, reader-friendly table showing the rank in popularity of the name "Olivia" as a female baby name over time; this should have rows for ethnicities and columns for year. Produce a similar table showing the most popular name among male children over time.

```
## Showing the rank in popularity of the name "Olivia" as a female baby name over time

female_names_df =
  names_df %>%
  filter(childs_first_name == "Olivia") %>%
  filter(gender =="FEMALE") %>%
  relocate(year_of_birth, rank) %>%
  arrange(year_of_birth, rank) %>%
  group_by(year_of_birth) %>%
  select(-childs_first_name)

female_names_df %>%
  pivot_wider(
    names_from = "ethnicity",
    values_from = "year_of_birth"
  )
```

```
## # A tibble: 16 x 7
##     rank gender count `WHITE NON HISPANIC` `ASIAN AND PACIFIC~ `BLACK NON HISPA~
##    <dbl> <chr>  <dbl>                <dbl>               <dbl>             <dbl>
##  1     1 FEMALE   233                 2013                  NA                NA
##  2     3 FEMALE   109                   NA                2013                NA
##  3     6 FEMALE    64                   NA                  NA              2013
##  4    22 FEMALE    87                   NA                  NA                NA
##  5     1 FEMALE   141                   NA                2014                NA
##  6     1 FEMALE   248                 2014                  NA                NA
##  7     8 FEMALE    52                   NA                  NA              2014
##  8    16 FEMALE    96                   NA                  NA                NA
##  9     1 FEMALE   188                   NA                2015                NA
## 10     1 FEMALE   225                 2015                  NA                NA
## 11     4 FEMALE    82                   NA                  NA              2015
## 12    16 FEMALE    94                   NA                  NA                NA
## 13     1 FEMALE   172                   NA                2016                NA
## 14     1 FEMALE   230                 2016                  NA                NA
## 15     8 FEMALE    49                   NA                  NA              2016
## 16    13 FEMALE   108                   NA                  NA                NA
## # ... with 1 more variable: HISPANIC <dbl>
```

```
female_names_df
```

```
## # A tibble: 16 x 5
## # Groups:   year_of_birth [4]
##    year_of_birth  rank gender ethnicity                  count
##            <dbl> <dbl> <chr>  <chr>                      <dbl>
##  1          2013     1 FEMALE WHITE NON HISPANIC           233
##  2          2013     3 FEMALE ASIAN AND PACIFIC ISLANDER   109
##  3          2013     6 FEMALE BLACK NON HISPANIC            64
##  4          2013    22 FEMALE HISPANIC                      87
##  5          2014     1 FEMALE ASIAN AND PACIFIC ISLANDER   141
##  6          2014     1 FEMALE WHITE NON HISPANIC           248
##  7          2014     8 FEMALE BLACK NON HISPANIC            52
##  8          2014    16 FEMALE HISPANIC                      96
##  9          2015     1 FEMALE ASIAN AND PACIFIC ISLANDER   188
## 10          2015     1 FEMALE WHITE NON HISPANIC           225
## 11          2015     4 FEMALE BLACK NON HISPANIC            82
## 12          2015    16 FEMALE HISPANIC                      94
## 13          2016     1 FEMALE ASIAN AND PACIFIC ISLANDER   172
## 14          2016     1 FEMALE WHITE NON HISPANIC           230
## 15          2016     8 FEMALE BLACK NON HISPANIC            49
## 16          2016    13 FEMALE HISPANIC                     108
```

```
## Showing the rank in popularity of the name as a male baby name over time

  male_names_df =
    names_df %>%
    filter(gender =="MALE") %>%
    relocate(year_of_birth, rank, childs_first_name) %>% ## show the first name rank
    arrange(rank, year_of_birth) %>%
    group_by(year_of_birth)

male_names_df
```

```
## # A tibble: 5,963 x 6
## # Groups:   year_of_birth [6]
##    year_of_birth  rank childs_first_name gender ethnicity                  count
##            <dbl> <dbl> <chr>             <chr>  <chr>                      <dbl>
##  1          2011     1 ETHAN             MALE   ASIAN AND PACIFIC ISLANDER   177
##  2          2011     1 JAYDEN            MALE   BLACK NON HISPANIC           184
##  3          2011     1 JAYDEN            MALE   HISPANIC                     426
##  4          2011     1 MICHAEL           MALE   WHITE NON HISPANIC           292
##  5          2012     1 RYAN              MALE   ASIAN AND PACIFIC ISLANDER   197
##  6          2012     1 JAYDEN            MALE   BLACK NON HISPANIC           171
##  7          2012     1 JAYDEN            MALE   HISPANIC                     364
##  8          2012     1 JOSEPH            MALE   WHITE NON HISPANIC           300
##  9          2013     1 Jayden            MALE   ASIAN AND PACIFIC ISLANDER   220
## 10          2013     1 Ethan             MALE   BLACK NON HISPANIC           146
## # ... with 5,953 more rows
```

Finally, for male, white non-hispanic children born in 2016, produce a scatter plot showing the number of children with a name (y axis) against the rank in popularity of that name (x axis).

```
plt_df <- names_df %>%
  filter(gender == "MALE" &
           ethnicity == "WHITE NON HISPANIC" &
           year_of_birth == 2016)

## scatter plot
ggplot(plt_df, aes(x = rank, y = count)) +
  geom_point(alpha = .5)
```