Data Science - Homework 2

Jake Rozran

2022-02-01

```
library(tidyverse)
```

Task 1) Add your name and the appropriate date in the header above.

Task 2) Enter the PollingReport.com data

PollingReport.com conducted a poll in 1999 in which they asked both men and women the following question: "All things considered, in our society today, do you think there are more advantages in being a man, more advantages, in being a woman, or are there no more advantages in being one than the other?" These results are labeled as man, woman, or none, respectively, in the data below. Those who did not know the answer to the question were labeled as "notknow".

Of women, 57% said man, 6% said woman, 33% said none, and 4% said notknow. Of men, 41% said man, 14% said woman, 40% said none, and 5% said notknow.

Create three variables, `men` which contains the four percentages listed above for men, `women` containing the percentages for women, and `response` which is a vector of character strings that state what response was given ("man", "woman", "none", and "notknow"). For the percentages, you are welcome to use either proportions or percentages, but do not include the "%" sign if you do the latter.

For this task and all others, make sure to verify that data are read in properly before moving forward

```
women <- c(0.57, 0.06, 0.33, 0.04)
women
```

```
## [1] 0.57 0.06 0.33 0.04
```

```
men <- c(0.41, 0.14, 0.4, 0.05)
men
```

```
## [1] 0.41 0.14 0.40 0.05
```

```
response <- c("man", "woman", "none", "notknown")
response
```

```
## [1] "man"      "woman"    "none"     "notknown"
```

Task 3) Explore the data and create new variables

a) Verify that the percentages in both `men` and `women` sum to 1

```
sum(women) == 1
```

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

```
sum(men) == 1
```

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

Does each one Sum to 1? Remove this line and answer the question.

b) Create a logical vector called `men_more` of length 4, which is a function of both `men` and `women`, which equals `TRUE` if percentage of men is higher than the percentage of women and `FALSE` otherwise.

```
men_more <- if_else(men > women, TRUE, FALSE)
men_more
```

```
## [1] FALSE  TRUE  TRUE  TRUE
```

c) Combine all four of the variables you created into a data frame called `advantage`. *(Hint: You could use either `cbind()` or `data.frame()`)*

```
advantage <- cbind(response, women, men)
advantage
```

```
##      response   women  men
## [1,] "man"      "0.57" "0.41"
## [2,] "woman"    "0.06" "0.14"
## [3,] "none"     "0.33" "0.4"
## [4,] "notknown" "0.04" "0.05"
```

```
advantage <- data.frame(response, women, men)
advantage
```

```
##   response women  men
## 1      man  0.57 0.41
## 2    woman  0.06 0.14
## 3     none  0.33 0.40
## 4 notknown  0.04 0.05
```

d) Use `ifelse` (or `if_else`) to create a new variable called `who_more` that equals "men" if men_more is `TRUE` and "women" if men_more if `FALSE`. **This variable should be created directly within the `advantage` data frame.**

```
advantage$who_more <- if_else(men_more == TRUE, "men", "women")
advantage
```

```
##   response women  men who_more
## 1      man  0.57 0.41    women
## 2    woman  0.06 0.14      men
## 3     none  0.33 0.40      men
## 4 notknown  0.04 0.05      men
```

Task 4) Add a new chunk below this question

Explore the `gapminder` data to discover...

Reminder, to reference a variable within the `gapminder` dataset, use `gapminder$varname` where varname is the name of the variable you want to explore.

a) the earliest year (the variable is called `year`) in the dataset
b) the latest year in the dataset

c) the number of years between the latest and earliest (it's better to use the functions here rather than just subtract the previous values)
d) the average population size (`pop`)
e) the average population size (`pop`) in 1000s (divide by 1000)
f) the median GDP per capita (`gdpPercap`)
g) whether there are any missing values in the dataset (any variable) *[hint: use the any() command]*
h) the `midhinge` [the average of the first and third quartile] of GDP per capita *[hint: use the quantile() command]*

```
library(gapminder)
data(gapminder) # YOU CAN TECHNICALLY SKIP THIS STEP

# A
min(gapminder$year)
```

```
## [1] 1952
```

```
# B
max(gapminder$year)
```

```
## [1] 2007
```

```
# C
max(gapminder$year) - min(gapminder$year)
```

```
## [1] 55
```

```
# D
mean(gapminder$pop)
```

```
## [1] 29601212
```

```
# E
mean(gapminder$pop) / 1000
```

```
## [1] 29601.21
```

```
mean(gapminder$pop / 1000)
```

```
## [1] 29601.21
```

```
# F
median(gapminder$gdpPercap)
```

```
## [1] 3531.847
```

```
# G
any(is.na(gapminder))
```

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

```
# H
midhinge <- mean(c(quantile(gapminder$gdpPercap, 0.25),
                   quantile(gapminder$gdpPercap, 0.75)))
midhinge
```

```
## [1] 5263.761
```

Task 5) Read data from external file

Many cities are publicizing their data as part of an "Open Data" initiative. Philadelphia's is located at Open Data Philly. Let's take a look at the cleanliness of neighborhoods around Philadelphia. I downloaded a csv file on Child Blood Lead Levels in Philadelphia from here. It can be found in the data section of the website. The "metadata" (information about the variables) can be found here.

Read the data file into R. Run a `str()` command to make sure it was read in properly. Verify that there are 46 observations and 5 variables.

```
data_url <- paste0("https://phl.carto.com/api/v2/sql?q=SELECT+*+FROM+child_bl",
                   "ood_lead_levels_by_zip&filename=child_blood_lead_levels_b",
                   "y_zip&format=csv&skipfields=cartodb_id,the_geom,the_geom_",
                   "webmercator")

odp <- read_csv(data_url)
```

```
## Rows: 46 Columns: 5
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## dbl (4): zip_code, num_bll_5plus, num_screen, perc_5plus
## lgl (1): data_redacted
##
## 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.
```

```
str(odp)
```

```
## spec_tbl_df [46 x 5] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ zip_code     : num [1:46] 19102 19103 19107 19104 19106 ...
##  $ data_redacted: logi [1:46] TRUE TRUE TRUE FALSE TRUE FALSE ...
##  $ num_bll_5plus: num [1:46] NA NA NA 28 NA 33 NA 8 NA 20 ...
##  $ num_screen   : num [1:46] 51 224 139 805 118 ...
##  $ perc_5plus   : num [1:46] NA NA NA 3.5 NA 3.1 NA 2.1 NA 3.5 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   zip_code = col_double(),
##   ..   data_redacted = col_logical(),
##   ..   num_bll_5plus = col_double(),
##   ..   num_screen = col_double(),
##   ..   perc_5plus = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

Task 6) Explore the Lead Level data

a) Verify the following. Unless otherwise stated, feel free to use whatever functions you wish.

b) There are 10 values missing for `num_bll_5plus` and for `perc_5plus`.

ii) These 10 missing values (see above) are the ones that have `data_redacted` equal to `TRUE`.

```
sum(is.na(odp$num_bll_5plus))
```

```
## [1] 10
```

```
which(is.na(odp$num_bll_5plus))
```

```
##  [1]  1  2  3  5  7  9 11 13 29 45
```

```
sum(is.na(odp$perc_5plus))
```

```
## [1] 10
```

```
which(is.na(odp$perc_5plus))
```

```
##  [1]  1  2  3  5  7  9 11 13 29 45
```

```
which(is.na(odp$num_bll_5plus)) == which(is.na(odp$num_bll_5plus))
```

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

```
odp[odp$data_redacted == TRUE, ]
```

```
## # A tibble: 10 x 5
##    zip_code data_redacted num_bll_5plus num_screen perc_5plus
##       <dbl> <lgl>                <dbl>      <dbl>      <dbl>
##  1    19102 TRUE                    NA         51         NA
##  2    19103 TRUE                    NA        224         NA
##  3    19107 TRUE                    NA        139         NA
##  4    19106 TRUE                    NA        118         NA
##  5    19114 TRUE                    NA        294         NA
##  6    19115 TRUE                    NA        397         NA
##  7    19116 TRUE                    NA        330         NA
##  8    19118 TRUE                    NA        121         NA
##  9    19137 TRUE                    NA        120         NA
## 10    19153 TRUE                    NA        276         NA
```

```
odp[odp$data_redacted == "true", ]
```

```
## # A tibble: 0 x 5
## # ... with 5 variables: zip_code <dbl>, data_redacted <lgl>,
## #   num_bll_5plus <dbl>, num_screen <dbl>, perc_5plus <dbl>
```

```
odp %>%
    filter(data_redacted == TRUE)
```

```
## # A tibble: 10 x 5
##    zip_code data_redacted num_bll_5plus num_screen perc_5plus
##       <dbl> <lgl>                <dbl>      <dbl>      <dbl>
##  1    19102 TRUE                    NA         51         NA
##  2    19103 TRUE                    NA        224         NA
##  3    19107 TRUE                    NA        139         NA
##  4    19106 TRUE                    NA        118         NA
##  5    19114 TRUE                    NA        294         NA
##  6    19115 TRUE                    NA        397         NA
##  7    19116 TRUE                    NA        330         NA
##  8    19118 TRUE                    NA        121         NA
##  9    19137 TRUE                    NA        120         NA
## 10    19153 TRUE                    NA        276         NA
```

```
odp %>%
    filter(data_redacted == "true")
```

```
## # A tibble: 0 x 5
## # ... with 5 variables: zip_code <dbl>, data_redacted <lgl>,
## #   num_bll_5plus <dbl>, num_screen <dbl>, perc_5plus <dbl>
```

b) Which zip code has the highest percent of kids with a high lead level? Which zip code has the lowest? Use the `perc_5plus` variable to determine these.

```
# Highest
max(odp$perc_5plus, na.rm = TRUE)
```

```
## [1] 9.2
```

```
odp$zip_code[!is.na(odp$perc_5plus) &
                 odp$perc_5plus == max(odp$perc_5plus, na.rm = TRUE)]
```

```
## [1] 19144
```

```
odp %>%
    filter(!is.na(perc_5plus) &
               perc_5plus == max(perc_5plus, na.rm = TRUE)) %>%
    select(zip_code)
```

```
## # A tibble: 1 x 1
##   zip_code
##      <dbl>
## 1    19144
```

```
# LOWEST
min(odp$perc_5plus, na.rm = TRUE)
```

```
## [1] 0
```

```
odp$zip_code[!is.na(odp$perc_5plus) &
                 odp$perc_5plus == min(odp$perc_5plus, na.rm = TRUE)]
```

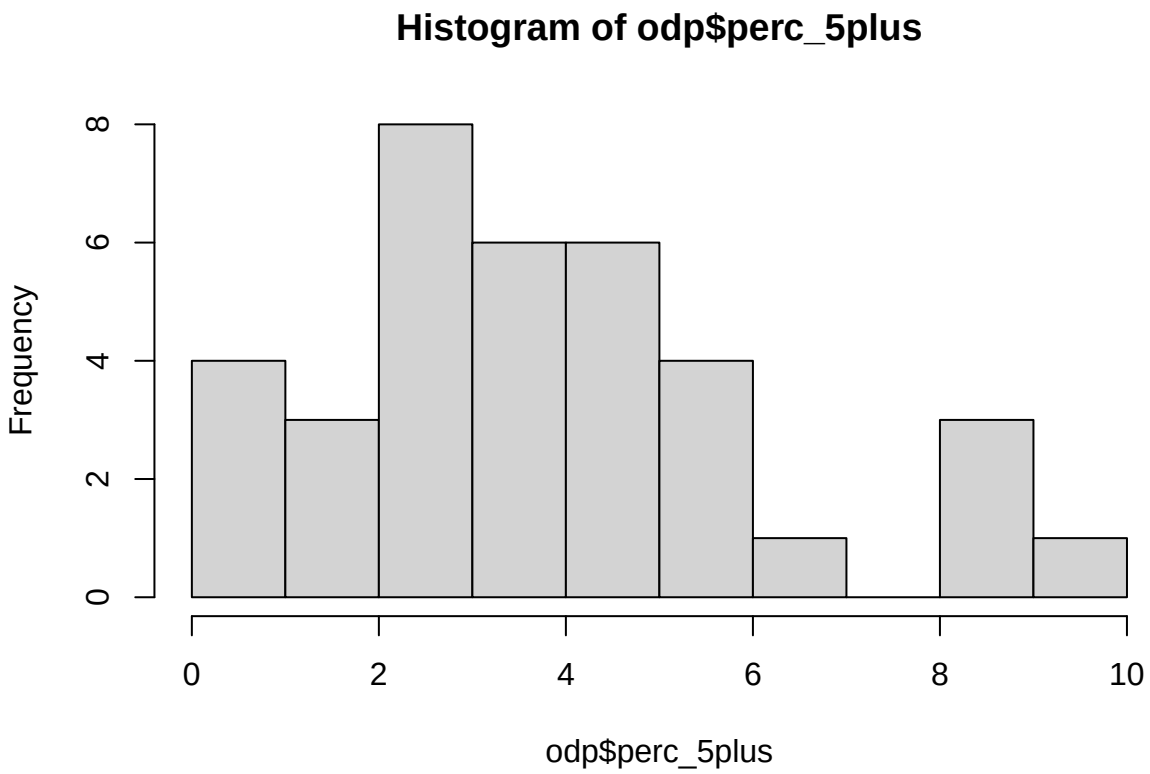
```
## [1] 19127 19154
```

```
odp %>%
    filter(!is.na(perc_5plus) &
               perc_5plus == min(perc_5plus, na.rm = TRUE)) %>%
    select(zip_code)
```

```
## # A tibble: 2 x 1
##   zip_code
##      <dbl>
## 1    19127
## 2    19154
```

c) Use the `hist()` function to show the distribution of `perc_5plus`. Comment on what you see.

```
hist(odp$perc_5plus)
```



Unimodal, skewed right, possible outliers > 8.