Milestone #3

Rachael Baartmans, Lara Petalio, Christine Truong

Sys.Date()

```
library(tidyverse)
```

```
## Warning in system("timedatectl", intern = TRUE): running command ’timedatectl’
## had status 1

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

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

```
race_data <-read_csv("ca_csc_outcome_race_data.csv",
            col_select = c(NERVOUS, WORRYING, PROBINTR,
                           PROBDOWN, ASTHMA, HEARTDIS,
                           DIABETES, OTHMENILL, race01, race02, race03,
                           race04, race05, race06, race07, race08,
                           race09, race10, race11, race12, race13,
                           race14, race15),
            na = c("", "NA", "NA/Not Applicable", "N/A", "n/a",
                   "(DO NOT READ) NA/Not Applicable",
                   "(DO NOT READ) Refused",
                   "(DO NOT READ) Don't know"))
```

```
## Rows: 1000 Columns: 23

## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (23): NERVOUS, WORRYING, PROBINTR, PROBDOWN, ASTHMA, HEARTDIS, DIABETES,...
##
## 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.
```

```
smoker_data <- read_csv("ca_csc_smoker_data.csv",
            col_select = c(smokstat, WHEREBUY, BUYCALIF,
                           HOWMANY, SMOK6NUM, SMOK6UNI),
            na = c("", "NA", "NA/Not Applicable", "N/A", "n/a",
                   "(DO NOT READ) NA/Not Applicable",
                   "(DO NOT READ) Refused",
                   "(DO NOT READ) Don't know"))
```

```
## Rows: 1000 Columns: 6
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): smokstat, HOWMANY, SMOK6UNI, BUYCALIF, WHEREBUY
## dbl (1): SMOK6NUM
##
## 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.
```

Subset rows and columns as needed

We have decided that we do not need to subset any rows and columns since we already did this during the importing process of our data (specified in the col_select argument of the read_csv function).

Clean variables for analysis

Minimum of 2

**Examples: Recode invalid values/handle missing fields/recode categories **

```
#Changed casing for variables from capitals to lowercase in both dataframes
#of race_data and smoker_data
names(race_data) <- tolower(names(race_data))
names(smoker_data) <- tolower(names(smoker_data))

#Changed the data type of "howmany" from character to numeric in order to
#perform calculations for pack-years in the future
smoker_data$howmany <- as.numeric(smoker_data$howmany)
```

```
## Warning: NAs introduced by coercion
```

```
#Removed NA's and string values from smoker_data in order to calculate
#pack-years later for variables "howmany", "smok6uni", and "smok6num"
smoker_data_2 <- smoker_data %>% filter(!is.na(howmany),
  howmany != "100 or more cigarettes", smok6uni == "Years", !is.na(smok6num))

##NEED TO ASK GSI ABOUT "100 OR MORE CIGARETTES" BEING FILTERED OUT - CAN
##IMPACT PACK-YEARS CALCULATION

#Viewed unique values assigned to the variables "howmany", "smok6uni", and
#smok6num to see if all NA's and strings were removed
unique(smoker_data_2$howmany)
```

```
##  [1] 30 20 15  7 10  5  6 60  8 25 40  4 18 24  2  9 12 35 11 48 50  3 13  1 21
## [26] 17 14 29 16
```

```
unique(smoker_data_2$smok6uni)
```

```
## [1] "Years"
```

```
unique(smoker_data_2$smok6num)
```

```
##  [1] 36 25 20  7 45 19  2 15 40 27  4 23 38 34 13 44 17 30 35  8 33 22 12 10  6
## [26] 28 11  3 42 14 39 16 46 37 29  5 41 18 47 31 21  1 53 43  9 26 49 24 32 48
```

Create New Variables needed for analysis

Minimum of 2 created from existing columns

Examples: calculating the rate or combining character strings

```
#Created new variable "race" to combine variables race01:race15
race_data_2 <- race_data %>%
  mutate(race = case_when(race01 == "Yes" ~ "race01",
        race02 == "Yes" ~ "race02",
        race03 == "Yes" ~ "race03",
        race04 == "Yes" ~ "race04",
        race05 == "Yes" ~ "race05",
        race06 == "Yes" ~ "race06",
        race07 == "Yes" ~ "race07",
        race08 == "Yes" ~ "race08",
        race09 == "Yes" ~ "race09",
        race10 == "Yes" ~ "race10",
        race11 == "Yes" ~ "race11",
        race12 == "Yes" ~ "race12",
        race13 == "Yes" ~ "race13",
        race14 == "Yes" ~ "race14",
        race15 == "Yes" ~ "race15")) %>%
  select(-(race01:race15))
#Used select() function to remove original race01:race15 variables

#Viewed the updated data set, race_data_2
race_data_2
```

```
## # A tibble: 1,000 x 9
##    nervous   worrying probintr probdown asthma heartdis diabetes othmenill race
##    <chr>     <chr>    <chr>    <chr>    <chr>  <chr>    <chr>    <chr>     <chr>
##  1 Nearly e~ Not at ~ Nearly ~ Not at ~ No     Yes      No       No        race~
##  2 Several ~ Several~ Several~ Not at ~ No     No       No       No        race~
##  3 Not at a~ Not at ~ Not at ~ Not at ~ No     No       No       No        race~
##  4 Several ~ Not at ~ Not at ~ Not at ~ Yes    No       No       No        race~
##  5 Not at a~ Several~ Not at ~ Not at ~ No     No       No       No        race~
##  6 Not at a~ Not at ~ Not at ~ Not at ~ No     Yes      No       No        race~
##  7 Not at a~ Not at ~ Not at ~ Several~ Yes    Yes      No       No        race~
##  8 Several ~ Nearly ~ Several~ Several~ No     No       No       No        race~
##  9 Several ~ Several~ Several~ <NA>     No     No       No       Yes       race~
## 10 More tha~ Several~ Not at ~ Not at ~ No     No       No       No        race~
## # ... with 990 more rows
```

```
#Created new variable "packs_per_day" for future calculations for pack-years
smoker_data_3 <- smoker_data_2 %>% mutate(packs_per_day = howmany/20)

#Viewed the final cleaned data set, race_data_3
smoker_data_3
```

```
## # A tibble: 816 x 7
##    smokstat             wherebuy buycalif howmany smok6num smok6uni packs_per_day
##    <chr>                <chr>    <chr>      <dbl>    <dbl> <chr>            <dbl>
##  1 Current daily smok~ At othe~ In Cali~      30       36 Years              1.5
##  2 Current daily smok~ At toba~ In Cali~      20       25 Years             1
##  3 Current daily smok~ At conv~ In Cali~      15       20 Years             0.75
##  4 Current daily smok~ At conv~ In Cali~      15        7 Years             0.75
##  5 Current daily smok~ At liqu~ In Cali~      20       45 Years             1
##  6 Current daily smok~ At othe~ In Cali~      15       19 Years             0.75
##  7 Current daily smok~ At conv~ In Cali~       7        2 Years             0.35
##  8 Current daily smok~ At toba~ In Cali~      20       15 Years             1
##  9 Current daily smok~ In mili~ In Cali~      10       40 Years             0.5
## 10 Current daily smok~ <NA>     <NA>          20       27 Years             1
## # ... with 806 more rows
```

Data dictionary based on clean dataset

must include: variable name, data type, and description

```
#For each of the 4 data elements we pick, we must use typeof() function and
#describe what it stands for using the research documents published for
#each variable

#Instructions: "Data dictionary based on clean dataset
#(minimum 4 data elements), including: Variable name, Data type, Description.
#Data dictionary can be included as text or table, but should be easy for
#teaching team to interpret/read."

typeof(smoker_data_3$wherebuy)
```

```
## [1] "character"
```

```
typeof(smoker_data_3$howmany)
```

```
## [1] "double"
```

```
typeof(race_data_2$nervous)
```

```
## [1] "character"
```

```
typeof(race_data_2$asthma)
```

```
## [1] "character"
```

Variable 1:wherebuy

- Data Type: character
- Description: The "wherebuy" variable contains the responses to the survey question of 'where do/did you usually buy your cigarettes?' this variable gave options of general locations, somewhere else, and don't know/refused.

Variable 2:howmany

- Data Type: double
- Description: The variable "howmany" contains the numeric data related to how many cigarettes were smoked in the last 30 days. The values given for this question were 1 to 100.

Variable 3:nervous

- Data Type: character
- Description: The variable "nervous" is a character variable that looks at whether individuals felt nervous, anxious, or on edge in the last two weeks. The responses for this question were don't know, refused, not at all, several days, more than half the days, and nearly everyday. These responses correlated to a numeric value but were put into the dataset in character form.

Variable 4:asthma

- Data Type: character
- Description: The variable “asthma” is related to medical history as given by a doctor in the past. This question asks if a physician has ever told you that you have asthma. This had three response options fo Yes, No, Refused.

Tables with descriptive statistics for 4 data elements

```
#Use Kable to make tables like in problem set 5
```