Problem Set 4

Patrick Traynor, Ph.D., R.D.

September 22, 2021

Due date: Monday, September 27

For this problem set you will tidy up a dataset of 500 individuals. We also want to calculate each individual's BMI and appropriately categorize them.

Load your data (500_Person_Gender_Height_Weight.csv):

Question 1

Clean the column headers to be all lower case, have no spaces, and rename "Location information" to location.

```
library(tidyverse)
```

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

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

```
## v ggplot2 3.3.3     v dplyr   1.0.6
## v tibble  3.1.2     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1
## v purrr   0.3.4
```

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

```
cleaned_X500_Person_Gender_Height_Weight <- rename_with(
  X500_Person_Gender_Height_Weight, ~ tolower(gsub(" ","_",.x,fixed=TRUE)))

cleaned_X500_Person_Gender_Height_Weight <-rename(
  cleaned_X500_Person_Gender_Height_Weight, location = location_information)
cleaned_X500_Person_Gender_Height_Weight
```

```
## # A tibble: 500 x 4
##    location       gender height weight
##    <chr>          <chr>   <dbl>  <dbl>
##  1 New York       Male     5.71   212.
##  2 United Kingdom Male   189       87
##  3 New York       Female   6.07   243.
##  4 Taiwan         Female 195      104
##  5 Taiwan         Male   149       61
##  6 Taiwan         Male   189      104
##  7 Colorado       Male     4.82   203.
##  8 New York       Male     5.05   245.
##  9 United Kingdom Male   174       90
```

```
## 10 Hawaii         Female   5.54   227.
## # ... with 490 more rows
```

Question 2

Create a new variable that calculates BMI for each individual.

You will need to navigate the different system of measurements (metric vs imperial). Only the United States is using imperial.

- BMI calculation and conversions:
  - metric: $BMI = weight(kg)/[height(m)]^2$
  - imperial: $BMI = 703 * weight(lbs)/[height(in)]^2$
  - 1 foot = 12 inches
  - 1 cm = 0.01 meter

Although there's many ways you can accomplish this task, we want you to use an if_else() to calculate BMI with the appropriate formula based on each person's location.

```
vector_for_unique_values <- unique(cleaned_X500_Person_Gender_Height_Weight$location)
vector_for_unique_values
```

```
## [1] "New York"       "United Kingdom" "Taiwan"         "Colorado"
## [5] "Hawaii"
```

```
vector_using_c_function_for_unique_values <- c(unique(cleaned_X500_Person_Gender_Height_Weight$location)
vector_using_c_function_for_unique_values
```

```
## [1] "New York"       "United Kingdom" "Taiwan"         "Colorado"
## [5] "Hawaii"
```

```
vector_for_us_locations <- c("Hawaii", "Colorado", "New York")
#page 8 of week 5 reader for "or"
cleaned_X500_Person_Gender_Height_Weight <-
  cleaned_X500_Person_Gender_Height_Weight %>%
  mutate(
    bmi = if_else(
      location == "United Kingdom" | location == "Taiwan",
      weight/(height/100)^2, 703*weight/(height*12)^2)
    )

# cleaned_X500_Person_Gender_Height_Weight <-
#   cleaned_X500_Person_Gender_Height_Weight %>%
#   mutate(
#     bmi = if_else(
#       location %in% vector_for_unique_values,
#       weight/(height/100)^2,703*weight/(height*12)^2)
#     )
```

Question 3

Create a new variable that categorizes BMI with case_when():

- Underweight: BMI below 18.5
- Normal: 18.5-24.9
- Overweight: 25.0-29.9
- Obese: 30.0 and Above

```
cleaned_X500_Person_Gender_Height_Weight <-
  cleaned_X500_Person_Gender_Height_Weight %>%
  mutate(
    bmi_cat = case_when(
    bmi<18.5 ~ "Underweight",
    bmi<24.9 ~ "Normal",
    bmi<29.9 ~"Overweight",
    TRUE ~ "Obese"
    ))
```

```
cleaned_X500_Person_Gender_Height_Weight <-
  cleaned_X500_Person_Gender_Height_Weight %>%
  select(
    location, gender, height, weight, bmi, bmi_cat)
```

Could we have used if_else()?

YOUR ANSWER HERE Yes, we could have, if we were to embed an if_else() into another if_else() E.g.,

```
cleaned_X500_Person_Gender_Height_Weight <-
  cleaned_X500_Person_Gender_Height_Weight %>%
  mutate(
    bmi_cat = if_else(bmi<18.5,"Underweight",if_else(
      bmi<24.9,"Normal",if_else(bmi<29.9,"Overweight","Obese"))))
```

Question 4

Arrange your data first by location and then by descending order of BMI.

```
cleaned_X500_Person_Gender_Height_Weight <-
  cleaned_X500_Person_Gender_Height_Weight %>%
  arrange(location,desc(bmi))
```

Question 5

Use a dplyr method to remove the height, weight, and BMI columns from your data.

```
cleaned_X500_Person_Gender_Height_Weight_stuff_removed <- cleaned_X500_Person_Gender_Height_Weight %>%
  select(c(location, gender, bmi_cat))
# cleaned_X500_Person_Gender_Height_Weight_stuff_removed <- cleaned_X500_Person_Gender_Height_Weight %>
#   select(-c(height, weight, bmi))
```

Optional Challenge

Perform all the actions in this problem set with one dpylr call.

```
cleaned_X500_Person_Gender_Height_Weight <-
  X500_Person_Gender_Height_Weight %>%
  rename_with(~ tolower(gsub(" ","_",.x,fixed=TRUE))) %>%
  rename(location = location_information) %>%
  mutate(bmi = if_else(
      location == "United Kingdom" | location == "Taiwan",
      weight/(height/100)^2, 703*weight/(height*12)^2),
    bmi_cat = case_when(
    bmi<18.5 ~ "Underweight",
    bmi<24.9 ~ "Normal",
    bmi<29.9 ~"Overweight",
    TRUE ~ "Obese"
    ))%>%
  arrange(location,desc(bmi)) %>%
  select(c(location, gender, bmi_cat))
# cleaned_X500_Person_Gender_Height_Weight <-
#   X500_Person_Gender_Height_Weight %>%
#   rename_with(
#     X500_Person_Gender_Height_Weight, ~ tolower(gsub(" ","_",.x,fixed=TRUE)))
#
# cleaned_X500_Person_Gender_Height_Weight <- rename_with(
#   X500_Person_Gender_Height_Weight, ~ tolower(gsub(" ","_",.x,fixed=TRUE)))
#
# cleaned_X500_Person_Gender_Height_Weight <-rename(cleaned_X500_Person_Gender_Height_Weight, location
```