STA130 Week 3 Class: Data Wrangling

Nathalie Moon

January 25, 2021

Synchronous class meeting

Taking up tricky questions from quiz 3

Loading the coffee ratings data

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
```

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

```
coffee_ratings <- read_csv("coffee_ratings.csv")
```

```
##
## -- Column specification --------------------------------------------------------
## cols(
##   .default = col_character(),
##   total_cup_points = col_double(),
##   aroma = col_double(),
##   flavor = col_double(),
##   aftertaste = col_double(),
##   acidity = col_double(),
##   body = col_double(),
##   balance = col_double(),
##   uniformity = col_double(),
##   clean_cup = col_double(),
##   sweetness = col_double(),
##   cupper_points = col_double(),
##   moisture = col_double(),
##   category_one_defects = col_double(),
##   quakers = col_double(),
##   category_two_defects = col_double(),
##   altitude_low_meters = col_double(),
##   altitude_high_meters = col_double(),
##   altitude_mean_meters = col_double()
```

```
## )
## i Use `spec()` for the full column specifications.
```

Let's look at the distribution of the "color" variable

```
# What type of variable is color?
# What type of plot do we use to visualize this type of distribution? Barplot
# Let's plot it!

# Lets generate a summary table to look at the exact number of observations for each color of coffee be

# How many different categories are there really?

# Let's combine similar categories:


# With case_when, any cases we DON'T list automatically lead to values of NA
# This can be a useful feature, but be careful not to forget to list all
# the cases that you intend to list


# Whenever we use case_when to create/modify a variable, it's a good idea to create a summary table to


# How can we edit our summary table to look at the mean and median overall coffee ratings (`total_cup_p


# What are two ways to make the table above more useful?

# What are pros/cons of the two approaches?
```

```
# For the top 5 rated coffees produced in Mexico, produce a tibble containing the overall coffee rating


# What are the 5 countries with the highest average coffee ratings, based on the observations in the co


# Among all countries with at least 10 coffee samples in these data, what are the 5 countries with the
```

R Code for slides/videos

```
library(tidyverse) # Load the tidyverse package to gain access to functions we'll use

# Load data from a csv file using read_csv
olympics <- read_csv("oly12countries.csv")
glimpse(olympics)
```

```
## Rows: 204
## Columns: 10
## $ Country        <chr> "Afghanistan", "Albania", "Algeria", "American Samoa...
## $ ISO            <chr> "AFG", "ALB", "DZA", "ASM", "AND", "AGO", "ATG", "AR...
## $ GDP.2011       <dbl> 2.034346e+10, 1.295956e+10, 1.886810e+11, 5.370000e+...
## $ pop.2010       <dbl> 34385000, 3205000, 35468000, 68420, 84864, 19082000,...
## $ athletes_f     <dbl> 1, 4, 18, 1, 2, 30, 2, 43, 4, 1, 188, 31, 14, 11, 8,...
## $ athletes_m     <dbl> 5, 7, 21, 4, 4, 5, 3, 99, 21, 3, 225, 39, 39, 15, 4,...
## $ athletes_total <dbl> 6, 11, 39, 5, 6, 35, 5, 142, 25, 4, 413, 70, 53, 26,...
## $ gold           <dbl> 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 7, 0, 2, 1, 0, 0, 0, 2...
## $ silver         <dbl> 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 16, 0, 2, 0, 0, 0, 0, ...
## $ bronze         <dbl> 1, 0, 0, 0, 0, 0, 0, 2, 2, 0, 12, 0, 6, 0, 1, 0, 0, ...
```

```
head(olympics)
```

```
## # A tibble: 6 x 10
##   Country ISO   GDP.2011 pop.2010 athletes_f athletes_m athletes_total  gold
##   <chr>   <chr>    <dbl>    <dbl>      <dbl>      <dbl>          <dbl> <dbl>
## 1 Afghan~ AFG    2.03e10 34385000          1          5              6     0
## 2 Albania ALB    1.30e10  3205000          4          7             11     0
## 3 Algeria DZA    1.89e11 35468000         18         21             39     1
## 4 Americ~ ASM    5.37e 8    68420          1          4              5     0
## 5 Andorra AND    3.49e 9    84864          2          4              6     0
## 6 Angola  AGO    1.01e11 19082000         30          5             35     0
## # ... with 2 more variables: silver <dbl>, bronze <dbl>
```

```
olympics %>%
  filter(athletes_total >= 300)
```

```
## # A tibble: 8 x 10
##   Country ISO   GDP.2011 pop.2010 athletes_f athletes_m athletes_total  gold
##   <chr>   <chr>    <dbl>    <dbl>      <dbl>      <dbl>          <dbl> <dbl>
## 1 Austra~ AUS    1.37e12   2.23e7        188        225            413     7
## 2 China   CHN    7.30e12   1.34e9        208        163            371    38
## 3 France  FRA    2.77e12   6.49e7        148        187            335    11
## 4 Germany DEU    3.57e12   8.18e7        176        219            395    11
## 5 Japan   JPN    5.87e12   1.27e8        162        141            303     7
## 6 Russia  RUS    1.86e12   1.42e8        227        208            435    24
## 7 UK      GBR    2.43e12   6.22e7        269        287            556    29
## 8 US      USA    1.51e13   3.09e8        271        260            531    46
## # ... with 2 more variables: silver <dbl>, bronze <dbl>
```

```
olympics %>%
  select(Country, athletes_total, gold, silver, bronze)
```

```
## # A tibble: 204 x 5
##    Country             athletes_total  gold silver bronze
##    <chr>                        <dbl> <dbl>  <dbl>  <dbl>
```

```
##  1 Afghanistan                     6     0     0     1
##  2 Albania                        11     0     0     0
##  3 Algeria                        39     1     0     0
##  4 American Samoa                  5     0     0     0
##  5 Andorra                         6     0     0     0
##  6 Angola                         35     0     0     0
##  7 Antigua and Barbuda             5     0     0     0
##  8 Argentina                     142     1     1     2
##  9 Armenia                        25     0     1     2
## 10 Aruba                           4     0     0     0
## # ... with 194 more rows
```

```
bigteams <- olympics %>%
  filter(athletes_total >= 300) %>%
  select(Country, athletes_total, gold, silver, bronze)
bigteams ## type the name of the R object to print it
```

```
## # A tibble: 8 x 5
##   Country   athletes_total  gold silver bronze
##   <chr>              <dbl> <dbl>  <dbl>  <dbl>
## 1 Australia            413     7     16     12
## 2 China                371    38     27     23
## 3 France               335    11     11     12
## 4 Germany              395    11     19     14
## 5 Japan                303     7     14     17
## 6 Russia               435    24     26     32
## 7 UK                   556    29     17     19
## 8 US                   531    46     29     29
```

```
olympics %>%
  filter(athletes_total < 100 & gold > 1) %>%
  select(Country, athletes_total, gold, silver, bronze)
```

```
head(olympics) %>% select(Country, athletes_total, gold, silver, bronze)
```

```
## # A tibble: 6 x 5
##   Country        athletes_total  gold silver bronze
##   <chr>                   <dbl> <dbl>  <dbl>  <dbl>
## 1 Afghanistan                 6     0      0      1
## 2 Albania                    11     0      0      0
## 3 Algeria                    39     1      0      0
## 4 American Samoa              5     0      0      0
## 5 Andorra                     6     0      0      0
## 6 Angola                     35     0      0      0
```

```
olynew <- olympics %>%
  mutate(total_medals = gold + silver + bronze,
         avg_medals = total_medals / athletes_total) %>%
  select(Country, athletes_total, gold, silver, bronze, total_medals, avg_medals)
head(olynew)
```

```
## # A tibble: 6 x 7
##   Country        athletes_total  gold silver bronze total_medals avg_medals
##   <chr>                   <dbl> <dbl>  <dbl>  <dbl>        <dbl>      <dbl>
## 1 Afghanistan                 6     0      0      1            1      0.167
## 2 Albania                    11     0      0      0            0      0
```

```
## 3 Algeria                    39     1      0      0           1     0.0256
## 4 American Samoa              5     0      0      0           0     0
## 5 Andorra                     6     0      0      0           0     0
## 6 Angola                     35     0      0      0           0     0
```

```
olympics %>%
  select(Country, athletes_total, athletes_f, athletes_m) %>%
  head(n=10)
```

```
## # A tibble: 10 x 4
##    Country             athletes_total athletes_f athletes_m
##    <chr>                        <dbl>      <dbl>      <dbl>
##  1 Afghanistan                      6          1          5
##  2 Albania                         11          4          7
##  3 Algeria                         39         18         21
##  4 American Samoa                   5          1          4
##  5 Andorra                          6          2          4
##  6 Angola                          35         30          5
##  7 Antigua and Barbuda              5          2          3
##  8 Argentina                      142         43         99
##  9 Armenia                         25          4         21
## 10 Aruba                            4          1          3
```

```
oly_newvar <- olympics %>%
  mutate(majority = case_when(athletes_f > athletes_m ~ "Female",
                              athletes_f == athletes_m ~ "Balanced",
                              athletes_f < athletes_m ~ "Male"),
         total_medals = gold + silver + bronze)

#oly_newvar <- olympics %>%
#  mutate(majority_female = ifelse(athletes_f > athletes_m, yes="Yes", no="No"),
#         total_medals = gold + silver + bronze)

oly_newvar %>% select(Country, athletes_total,
                      athletes_f, athletes_m, majority, total_medals)
```

```
## # A tibble: 204 x 6
##    Country             athletes_total athletes_f athletes_m majority total_medals
##    <chr>                        <dbl>      <dbl>      <dbl> <chr>           <dbl>
##  1 Afghanistan                      6          1          5 Male                1
##  2 Albania                         11          4          7 Male                0
##  3 Algeria                         39         18         21 Male                1
##  4 American Samoa                   5          1          4 Male                0
##  5 Andorra                          6          2          4 Male                0
##  6 Angola                          35         30          5 Female              0
##  7 Antigua and Barbu~               5          2          3 Male                0
##  8 Argentina                      142         43         99 Male                4
##  9 Armenia                         25          4         21 Male                3
## 10 Aruba                            4          1          3 Male                0
## # ... with 194 more rows
```

```
olynew %>%
  arrange(desc(total_medals)) %>%
  select(Country, total_medals, avg_medals) %>%
  head()
```

```
## # A tibble: 6 x 3
##   Country total_medals avg_medals
##   <chr>          <dbl>      <dbl>
## 1 US               104      0.196
## 2 China             88      0.237
## 3 Russia            82      0.189
## 4 UK                65      0.117
## 5 Germany           44      0.111
## 6 Japan             38      0.125
```

```
olynew %>%
  arrange(desc(avg_medals)) %>%
  select(Country, total_medals, avg_medals) %>%
  head()
```

```
## # A tibble: 6 x 3
##   Country  total_medals avg_medals
##   <chr>           <dbl>      <dbl>
## 1 Botswana            1      0.25
## 2 Jamaica            12      0.24
## 3 China              88      0.237
## 4 Iran               12      0.226
## 5 Kenya              11      0.22
## 6 Ethiopia            7      0.2
```

```
# Summary tables
olympics %>% summarise(n=n(),
                       mean_gold=mean(gold),
                       min_gold=min(gold),
                       max_gold=max(gold))
```

```
## # A tibble: 1 x 4
##       n mean_gold min_gold max_gold
##   <int>     <dbl>    <dbl>    <dbl>
## 1   204      1.48        0       46
```

```
olympics %>%
  mutate(teamsize = case_when(athletes_total >= 100 ~ "big",
                              athletes_total < 100 & athletes_total >= 20 ~ "medium",
                              athletes_total < 20 ~ "small")) %>%
  group_by(teamsize) %>%
  summarise(n=n(),
            mean_gold=mean(gold),
            min_gold=min(gold),
            max_gold=max(gold))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 5
##   teamsize     n mean_gold min_gold max_gold
##   <chr>    <int>     <dbl>    <dbl>    <dbl>
## 1 big         36    7.39          0       46
## 2 medium      51    0.667         0        4
## 3 small      117    0.0171        0        1
```

```
oly_newvar %>%
  group_by(majority) %>%
```

```
  summarise(n=n(),
            mean_gold=mean(gold),
            min_gold=min(gold),
            max_gold=max(gold))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 4 x 5
##   majority     n mean_gold min_gold max_gold
##   <chr>    <int>     <dbl>    <dbl>    <dbl>
## 1 Balanced    25      0.16        0        4
## 2 Female      34      3.71        0       46
## 3 Male       144      1.19        0       29
## 4 <NA>         1      0           0        0
```

```
oly_newvar %>%
  group_by(majority) %>%
  summarise(n=n(),
            mean_medals=mean(gold + silver + bronze),
            min_medals=min(gold + silver + bronze),
            max_medals=max(gold + silver + bronze))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 4 x 5
##   majority     n mean_medals min_medals max_medals
##   <chr>    <int>       <dbl>      <dbl>      <dbl>
## 1 Balanced    25        0.6           0         12
## 2 Female      34       10.8           0        104
## 3 Male       144        4.02          0         65
## 4 <NA>         1        0             0          0
```

```
oly_newvar %>%
  filter(is.na(majority)) %>%
  select(Country, ISO, athletes_f, athletes_m, gold, silver, bronze, majority)
```

```
## # A tibble: 1 x 8
##   Country  ISO   athletes_f athletes_m  gold silver bronze majority
##   <chr>    <chr>      <dbl>      <dbl> <dbl>  <dbl>  <dbl> <chr>
## 1 Barbados BRB           NA          6     0      0      0 <NA>
```

```
oly_newvar %>%
  filter(!is.na(majority)) %>%
  group_by(majority) %>%
  summarise(n=n(),
            mean_medals=mean(gold + silver + bronze),
            min_medals=min(gold + silver + bronze),
            max_medals=max(gold + silver + bronze))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 3 x 5
##   majority     n mean_medals min_medals max_medals
##   <chr>    <int>       <dbl>      <dbl>      <dbl>
## 1 Balanced    25        0.6           0         12
## 2 Female      34       10.8           0        104
## 3 Male       144        4.02          0         65
```