Week 4 Exercises

Jessica Riedy

2024-04-03

Please complete all exercises below. You may use any library that we have covered in class. The data we will be using comes from the tidyr package, so you must use that.

1) Examine the who and population data sets that come with the tidyr library. The who data is not tidy, you will need to reshape the new_sp_m014 to newrel_f65 columns to long format retaining country, iso2, iso3, and year. The data in the columns you are reshaping contains patterns described in the details section below. You will need to assign three columns: diagnosis, gender, and age to the patterns described in the details.

Your tidy data should look like the following:

```
#   country     iso2  iso3   year diagnosis gender age   count
#   <chr>       <chr> <chr> <int> <chr>     <chr>  <chr> <int>
# 1 Afghanistan AF    AFG    1980 sp        m      014      NA
# 2 Afghanistan AF    AFG    1980 sp        m      1524     NA
# 3 Afghanistan AF    AFG    1980 sp        m      2534     NA
# 4 Afghanistan AF    AFG    1980 sp        m      3544     NA
# 5 Afghanistan AF    AFG    1980 sp        m      4554     NA
# 6 Afghanistan AF    AFG    1980 sp        m      5564     NA
```

Details The data uses the original codes given by the World Health Organization. The column names for columns five through 60 are made by combining new_ to a code for method of diagnosis (rel = relapse, sn = negative pulmonary smear, sp = positive pulmonary smear, ep = extrapulmonary) to a code for gender (f = female, m = male) to a code for age group (014 = 0-14 yrs of age, 1524 = 15-24 years of age, 2534 = 25 to 34 years of age, 3544 = 35 to 44 years of age, 4554 = 45 to 54 years of age, 5564 = 55 to 64 years of age, 65 = 65 years of age or older).

Note: use data(who) and data(population) to load the data into your environment. Use the arguments cols, names_to, names_pattern, and values_to. Your regex should be = ("new_?(.)_(.)(.)")

https://tidyr.tidyverse.org/reference/who.html

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

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

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.3.3
```

```r
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```r
library(scales)

# data from `tidyr` package
who <- who
population <- population

# Reshape `who` data
who_longer <- who %>%
  pivot_longer(cols = c("new_sp_m014":"newrel_f65")) %>% # reshape to long format
  mutate(diagnosis = gsub("_", "", substr(name, 4, 6)) # extract method of diagnosis from
                                                       # `name`, remove "_" if applicable
         ,gender = substr(name, 8, 8) # extract gender code from `name`
         ,age= substr(name, 9, 12) # extract age group code from `name`
         ,name = NULL # drop `name` column
         ,count = value # create `value` from `count`
         ,value = NULL # drop `value`
         )
head(who_longer)
```

```
## # A tibble: 6 x 8
##   country     iso2  iso3   year diagnosis gender age   count
##   <chr>       <chr> <chr> <dbl> <chr>     <chr>  <chr> <dbl>
## 1 Afghanistan AF    AFG    1980 sp        m      014      NA
## 2 Afghanistan AF    AFG    1980 sp        m      1524     NA
## 3 Afghanistan AF    AFG    1980 sp        m      2534     NA
## 4 Afghanistan AF    AFG    1980 sp        m      3544     NA
## 5 Afghanistan AF    AFG    1980 sp        m      4554     NA
## 6 Afghanistan AF    AFG    1980 sp        m      5564     NA
```

2) There are two common keys between the data sets, with who as the left table, join the population data by country and year so that the population is available within the who dataset.

```r
who_pop <- who_longer %>%
  left_join(population, by = c("country", "year"))
head(who_pop)
```

```
## # A tibble: 6 x 9
##   country     iso2  iso3   year diagnosis gender age   count population
##   <chr>       <chr> <chr> <dbl> <chr>     <chr>  <chr> <dbl>      <dbl>
## 1 Afghanistan AF    AFG    1980 sp        m      014      NA         NA
## 2 Afghanistan AF    AFG    1980 sp        m      1524     NA         NA
## 3 Afghanistan AF    AFG    1980 sp        m      2534     NA         NA
## 4 Afghanistan AF    AFG    1980 sp        m      3544     NA         NA
## 5 Afghanistan AF    AFG    1980 sp        m      4554     NA         NA
## 6 Afghanistan AF    AFG    1980 sp        m      5564     NA         NA
```

3) Split the age column into two columns, min age and max age. Notice that there is no character separator. Check the documentation with ?separate to understand other ways to separate the age column. Keep in mind that 0 to 14 is coded as 014 (3 characters) and the other age groups are coded with 4 characters. 65 only has two characters, but we will ignore that until the next problem.

```
who_age_sep <- who_pop %>%
  separate(col = age, into = c("min_age", "max_age"), sep = -2)
head(who_age_sep, n = 8)
```

```
## # A tibble: 8 x 10
##   country    iso2  iso3   year diagnosis gender min_age max_age count population
##   <chr>      <chr> <chr> <dbl> <chr>     <chr>  <chr>   <chr>   <dbl>      <dbl>
## 1 Afghanist~ AF    AFG    1980 sp        m      "0"     14         NA         NA
## 2 Afghanist~ AF    AFG    1980 sp        m      "15"    24         NA         NA
## 3 Afghanist~ AF    AFG    1980 sp        m      "25"    34         NA         NA
## 4 Afghanist~ AF    AFG    1980 sp        m      "35"    44         NA         NA
## 5 Afghanist~ AF    AFG    1980 sp        m      "45"    54         NA         NA
## 6 Afghanist~ AF    AFG    1980 sp        m      "55"    64         NA         NA
## 7 Afghanist~ AF    AFG    1980 sp        m      ""      65         NA         NA
## 8 Afghanist~ AF    AFG    1980 sp        f      "0"     14         NA         NA
```

4) Since we ignored the 65+ group in the previous problem we will fix it here. If you examine the data you will notice that 65 was placed into the max_age column and there is no value for min_age for those records. To fix this use mutate() in order to replace the blank value in the min_age column with the value from the max_age column and another mutate to replace the 65 in the max column with an Inf. Be sure to keep the variables as character vectors.

```
who_age_fix <- who_age_sep %>%
  mutate(min_age = case_when(max_age == "65" ~ "65"
                             ,TRUE ~ min_age)
        ,max_age = case_when(max_age == "65" ~ "Inf"
                             ,TRUE ~ max_age)
    )
head(who_age_fix, n = 8)
```

```
## # A tibble: 8 x 10
##   country    iso2  iso3   year diagnosis gender min_age max_age count population
##   <chr>      <chr> <chr> <dbl> <chr>     <chr>  <chr>   <chr>   <dbl>      <dbl>
## 1 Afghanist~ AF    AFG    1980 sp        m      0       14         NA         NA
## 2 Afghanist~ AF    AFG    1980 sp        m      15      24         NA         NA
## 3 Afghanist~ AF    AFG    1980 sp        m      25      34         NA         NA
## 4 Afghanist~ AF    AFG    1980 sp        m      35      44         NA         NA
## 5 Afghanist~ AF    AFG    1980 sp        m      45      54         NA         NA
## 6 Afghanist~ AF    AFG    1980 sp        m      55      64         NA         NA
## 7 Afghanist~ AF    AFG    1980 sp        m      65      Inf        NA         NA
## 8 Afghanist~ AF    AFG    1980 sp        f      0       14         NA         NA
```

5) Find the count per diagnosis for males and females.

See ?sum for a hint on resolving NA values.

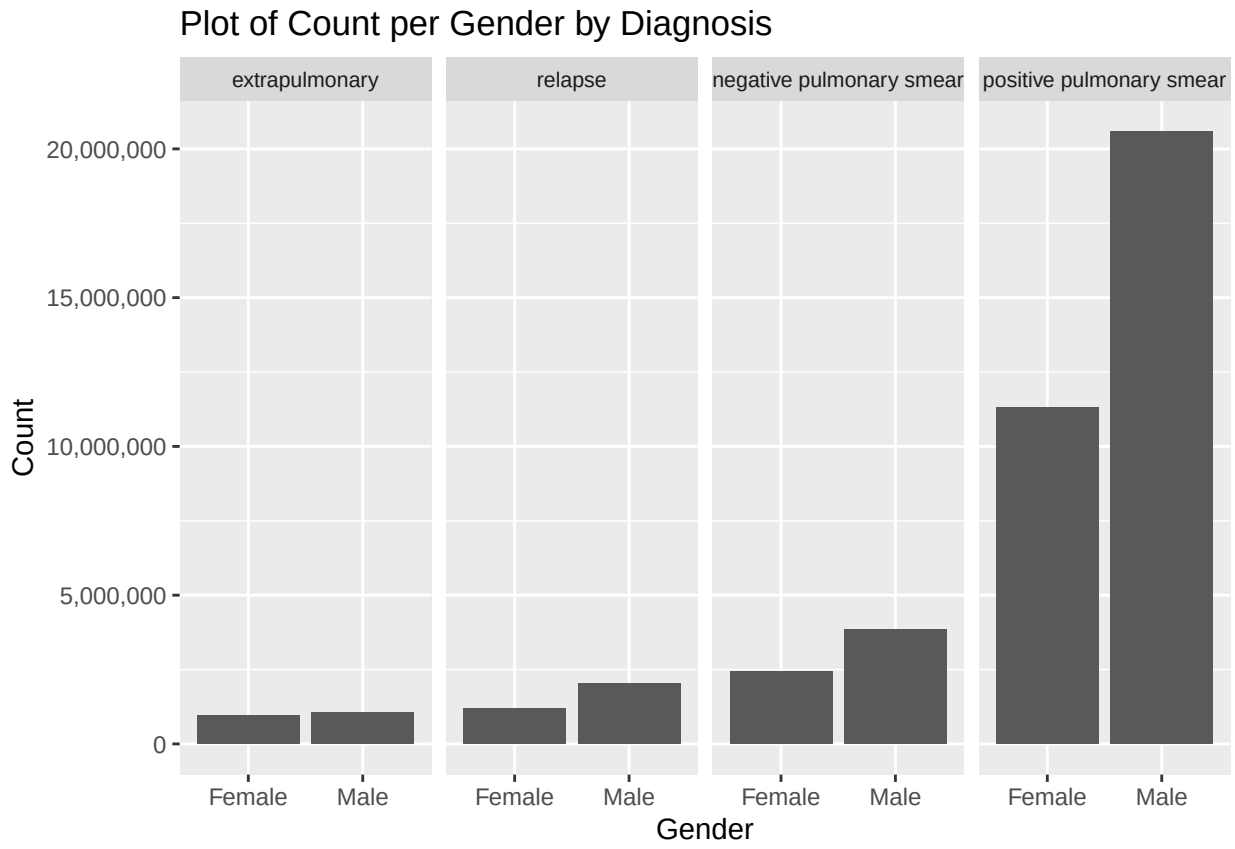
```
who_count <- who_age_fix %>%
  summarise(count = sum(count, na.rm = TRUE), .by = c("gender", "diagnosis"))
head(who_count)
```

```
## # A tibble: 6 x 3
##   gender diagnosis    count
##   <chr>  <chr>        <dbl>
## 1 m      sp        20586831
## 2 f      sp        11324409
## 3 m      sn         3840388
## 4 f      sn         2439139
```

```
## 5 m       ep          1044299
## 6 f       ep           941880
```

6) Now create a plot using ggplot and geom_col where your x axis is gender, your y axis represents the counts, and facet by diagnosis. Be sure to give your plot a title and resolve the axis labels.

```
diagnosis.labs <- c("relapse", "negative pulmonary smear", "positive pulmonary smear",
                    "extrapulmonary")
names(diagnosis.labs) <- c("rel", "sn", "sp", "ep")
ggplot(who_count,
       mapping = aes(x = gender, y = count)) +
       geom_col() +
       ggtitle("Plot of Count per Gender by Diagnosis") +
       xlab("Gender") + ylab("Count") +
       scale_x_discrete(labels=c("Female", "Male")) +
       scale_y_continuous(labels = label_comma()) +
       facet_grid(. ~ diagnosis, labeller = labeller(diagnosis = diagnosis.labs)) +
       theme(strip.text.x = element_text(size = 8))
```



7) Find the percentage of population by year, gender, and diagnosis. Be sure to remove rows containing NA values.

```
population_global <- population %>%
  summarise(population = sum(population, na.rm = TRUE), .by = c("year"))
who_percentage <- who_age_fix %>%
  summarise(count = sum(count, na.rm = TRUE), .by = c("year", "gender", "diagnosis")) %>%
  left_join(population_global) %>%
  mutate(percent = count / population) %>%
```

```
  filter(!is.na(percent))
```

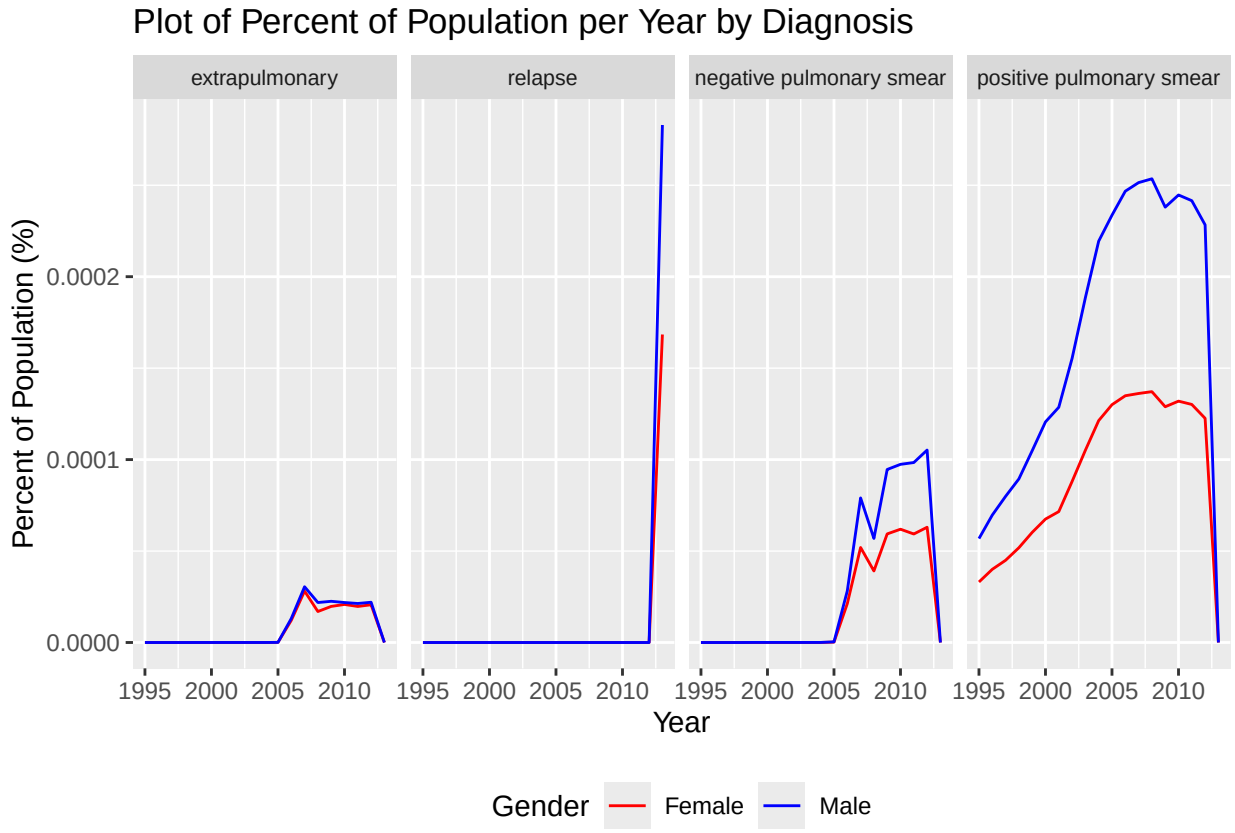
```
## Joining with `by = join_by(year)`
```

```
head(who_percentage)
```

```
## # A tibble: 6 x 6
##    year gender diagnosis  count population   percent
##   <dbl> <chr>  <chr>      <dbl>      <dbl>     <dbl>
## 1  1995 m      sp        324830 5717507165 0.0000568
## 2  1995 f      sp        189141 5717507165 0.0000331
## 3  1995 m      sn             0 5717507165 0
## 4  1995 f      sn             0 5717507165 0
## 5  1995 m      ep             0 5717507165 0
## 6  1995 f      ep             0 5717507165 0
```

8) Create a line plot in ggplot where your x axis contains the year and y axis contains the percent of world population. Facet this plot by diagnosis with each plot stacked vertically. You should have a line for each gender within each facet. Be sure to format your y axis and give your plot a title.

```
ggplot(who_percentage,
       mapping = aes(x = year, y = percent, color = gender)) +
       geom_line() +
       ggtitle("Plot of Percent of Population per Year by Diagnosis") +
       xlab("Year") + ylab("Percent of Population (%)") +
       labs(color = "Gender") +
       scale_color_manual(labels = c("Female", "Male"), values = c("red", "blue")) +
       theme(legend.position = "bottom") +
       scale_y_continuous(labels = label_comma()) +
       facet_grid(. ~ diagnosis, labeller = labeller(diagnosis = diagnosis.labs)) +
       theme(strip.text.x = element_text(size = 8))
```



9) Now unite the min and max age variables into a new variable named age_range. Use a '-' as the separator.

```
who_age_unite <- who_age_fix %>%
  unite("age_range", min_age, max_age, sep = "-")
head(who_age_unite)
```

```
## # A tibble: 6 x 9
##   country     iso2  iso3   year diagnosis gender age_range count population
##   <chr>       <chr> <chr> <dbl> <chr>     <chr>  <chr>     <dbl>      <dbl>
## 1 Afghanistan AF    AFG    1980 sp        m      0-14         NA         NA
## 2 Afghanistan AF    AFG    1980 sp        m      15-24        NA         NA
## 3 Afghanistan AF    AFG    1980 sp        m      25-34        NA         NA
## 4 Afghanistan AF    AFG    1980 sp        m      35-44        NA         NA
## 5 Afghanistan AF    AFG    1980 sp        m      45-54        NA         NA
## 6 Afghanistan AF    AFG    1980 sp        m      55-64        NA         NA
```

10) Find the percentage contribution of each age group by diagnosis. You will first need to find the count of all diagnoses then find the count of all diagnoses by age group. Join the former to the later and calculate the percent of each age group. Plot these as a geom_col where the x axis is the diagnosis, y axis is the percent of total, and faceted by age group.

```
who_diagnosis <- who_age_unite %>%
  summarise(count = sum(count, na.rm = TRUE), .by = c("diagnosis"))
who_percentage_age <- who_age_unite %>%
  summarise(count_by_age = sum(count, na.rm = TRUE), .by = c("age_range", "diagnosis")) %>%
  left_join(who_diagnosis) %>%
  mutate(percent = count_by_age / count)
```

```
## Joining with `by = join_by(diagnosis)`
```

```
head(who_percentage_age)
```

```
## # A tibble: 6 x 5
##   age_range diagnosis count_by_age    count percent
##   <chr>     <chr>            <dbl>    <dbl>   <dbl>
## 1 0-14      sp              628491 31911240  0.0197
## 2 15-24     sp             5897854 31911240  0.185
## 3 25-34     sp             7435960 31911240  0.233
## 4 35-44     sp             6284568 31911240  0.197
## 5 45-54     sp             4986011 31911240  0.156
## 6 55-64     sp             3579457 31911240  0.112
```

```
ggplot(who_percentage_age,
       mapping = aes(x = diagnosis, y = percent)) +
       geom_col() +
       ggtitle("Plot of Age Group per Diagnosis vs Diagnosis by Age Group") +
       xlab("Diagnosis") + ylab("Percent of Age Group per Diagnosis (%)") +
       facet_grid(. ~ age_range)
```