# Week 4 Exercises

Patrick Sheehan

April 7, 2024

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 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).

For this question I took the wide data and pivoted it to long data using the given regular expression and column names.

*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

```r
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
```

```r
library(tidyr)
library(ggplot2)

#your code here
data(who)
data(population)

who_long <- who %>%
  pivot_longer( cols = 5:60
               , names_pattern = "new_?(.*)_(.)(.*)"
               , names_to = c("diagnosis","gender","age")
               , values_to = "count")
print(who_long)
```

```
## # A tibble: 405,440 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
##  7 Afghanistan AF    AFG    1980 sp        m      65       NA
##  8 Afghanistan AF    AFG    1980 sp        f      014      NA
##  9 Afghanistan AF    AFG    1980 sp        f      1524     NA
## 10 Afghanistan AF    AFG    1980 sp        f      2534     NA
## # i 405,430 more rows
```

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.

For this question I left joined population on country and year.

```
# your code here

population_join <- who_long %>%
left_join(population
  ,by = c("country", "year"))
```

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.

For this question I used the separate function to split the age column. I used sep = -2 so that it would separate by looking at variables starting to the right and fill = left to fill in missing variables when the length of the split columns was too short.

```
# your code here

  population_join<- population_join %>%
  separate(age
          , sep = -2
          , into = c("min_age","max_age")
          , fill = "left"
           )
```

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.

Here I used an ifelse statement to change min_age to 65 if the max age equaled 65 and another ifelse statement to display inf if the max_age column was equal to 65.

```
# your code here

population_join <- population_join %>%
mutate(min_age =

  ifelse((max_age == 65), max_age, min_age)) %>%
mutate(max_age =

      ifelse((max_age == 65), Inf, max_age))
```

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

Here I used summarize to show me the sum of the count of diagnoses grouped by gender. I also used na.re = TRUE to remove NA values in the count column.

*See ?sum for a hint on resolving NA values.*

```
# your code here
population_join %>%
  group_by(gender) %>%
  summarize(diagnosis_count = sum(count, na.rm = TRUE ))
```

```
## # A tibble: 2 x 2
##   gender diagnosis_count
##   <chr>            <dbl>
## 1 f             15907024
## 2 m             27490494
```

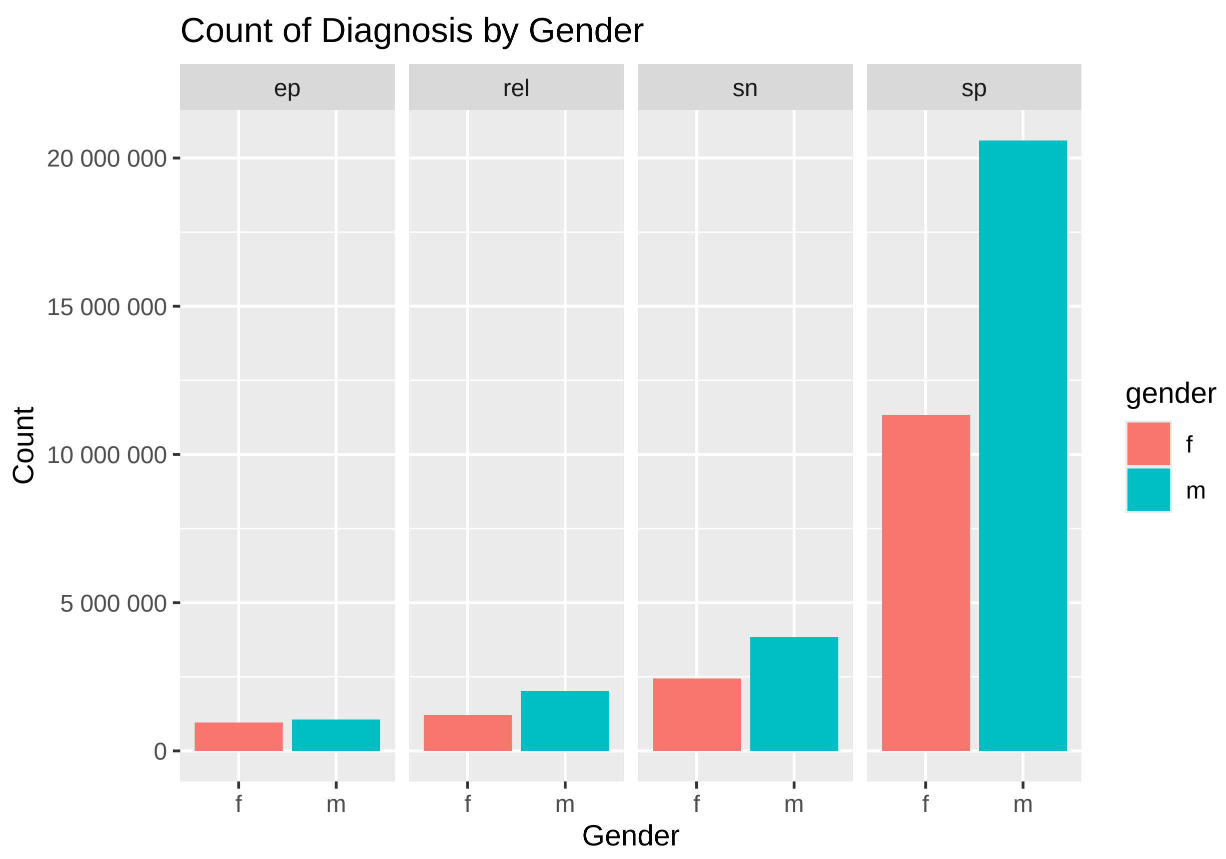
6) Now create a plot using ggplot and geom_col where your x axis is gender, your y axis represents the counts, and facet by diagnosis. Be sure to give your plot a title and resolve the axis labels.

Here I used ggplot to make a bar chart that represented the count of diagnosis by gender and what type of diagnosis that was I scaled the y-axis to show me number formats and used the fill to show the differences in gender.

```
#your code here

ggplot(population_join, aes(x=gender, y=count, fill = gender)) +
geom_col() +
  labs( x= "Gender",
        y= "Count",
        title = "Count of Diagnosis by Gender") +
  facet_grid(.~diagnosis) +
  scale_y_continuous(labels = scales::number_format())
```

```
## Warning: Removed 329394 rows containing missing values or values outside the scale range
## (`geom_col()`).
```



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

Here I used summarize to show me the percentage of population grouped by year, gender, and diagnosis. I used drop_na to remove NA values from both count and population.

```
# your code here

  population_join %>%
  drop_na(population,count) %>%
  group_by(year, gender, diagnosis) %>%
  summarize(percentage_of_population= 100*sum(count)/sum(population))
```

```
## `summarise()` has grouped output by 'year', 'gender'. You can override using
## the `.groups` argument.
```

```
## # A tibble: 94 x 4
## # Groups:   year, gender [38]
##     year gender diagnosis percentage_of_population
##    <dbl> <chr>  <chr>                        <dbl>
##  1  1995 f      sp                        0.000575
##  2  1995 m      sp                        0.000983
##  3  1996 f      sp                        0.000666
##  4  1996 m      sp                        0.00116
##  5  1997 f      sp                        0.000740
##  6  1997 m      sp                        0.00132
##  7  1998 f      sp                        0.000811
##  8  1998 m      sp                        0.00140
##  9  1999 f      ep                              0.000138
## 10  1999 f      sn                              0.000188
## # i 84 more rows
```

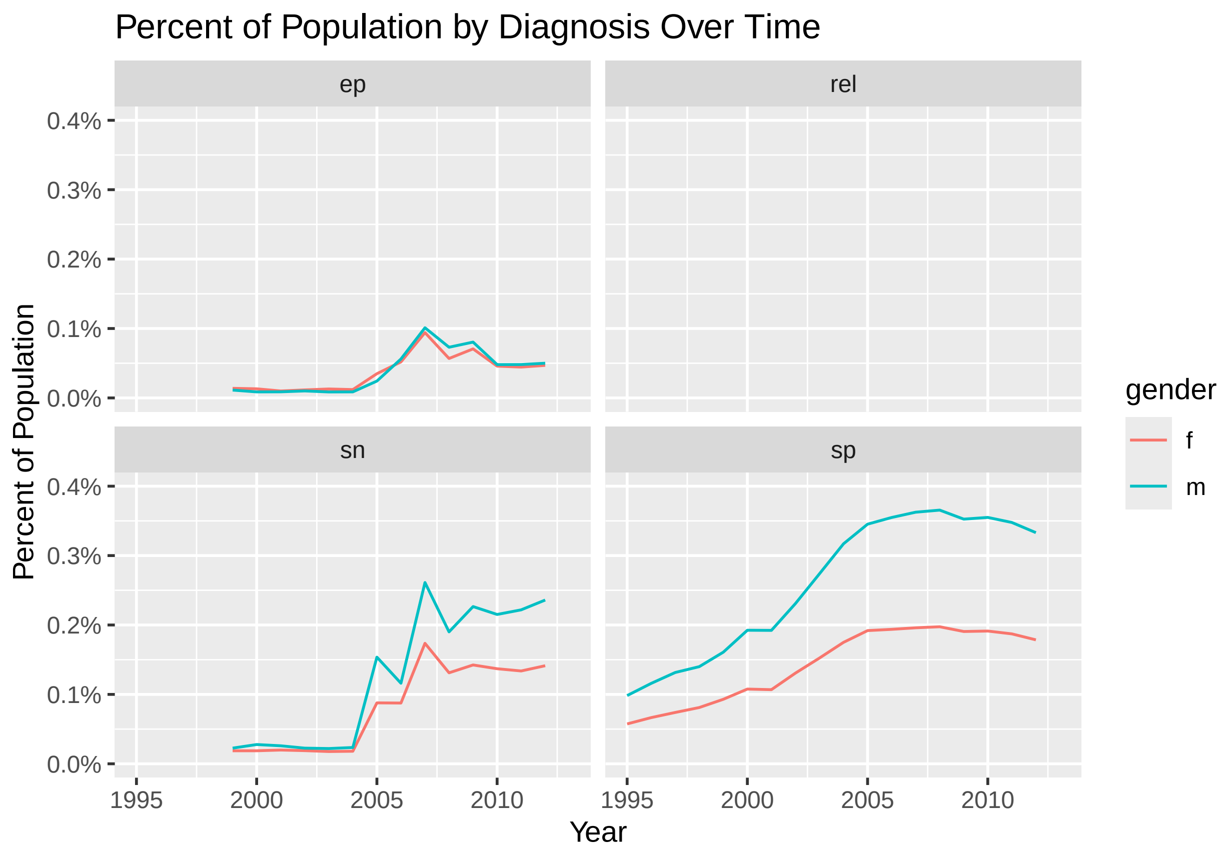
8) Create a line plot in ggplot where your x axis contains the year and y axis contains the percent of world population. Facet this plot by diagnosis with each plot stacked vertically. You should have a line for each gender within each facet. Be sure to format your y axis and give your plot a title.

Here I first dropped na values from count and population. I then added a column to a new data frame using mutate to show me the percentage of population grouped by year, gender and diagnosis. I then used ggplot to make a line graph that shows the percent of population bi diagnosis over time. I used facet_wrap to stack the charts and set limits on the y-axis so that we could see the small percentages.

```r
# your code here

population_join_plot <- population_join %>%
  drop_na(count,population) %>%
  group_by(year, gender, diagnosis) %>%
  mutate(percentage_of_population = 100*sum(count)/sum(population))

ggplot(population_join_plot) +
       geom_line(aes(x=year,y= percentage_of_population, group = gender, color = gender)) +
        labs(x="Year"
              ,y="Percent of Population"
              ,title="Percent of Population by Diagnosis Over Time") +
     facet_wrap(.~diagnosis) +
  scale_y_continuous(labels=scales::percent_format(),
                   limits = c(0,.004))
```



```
##use scale library
```

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

Here I used unite to combine the min and max age range columns using the _ as a separator.

```
# your code here

population_join <-population_join %>%
  unite("age_range",7:8, sep = "_")
```

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.

Here I found the count of all diagnoses using the sum of the count and removing na values. I then found the count of all diagnoses by age group. After that I used a left join too join the two and added a new column to show me the percent of each age group. Lastly, I used ggplot to make a bar chart faceted by age group that displays the percent of diagnosis per age group.

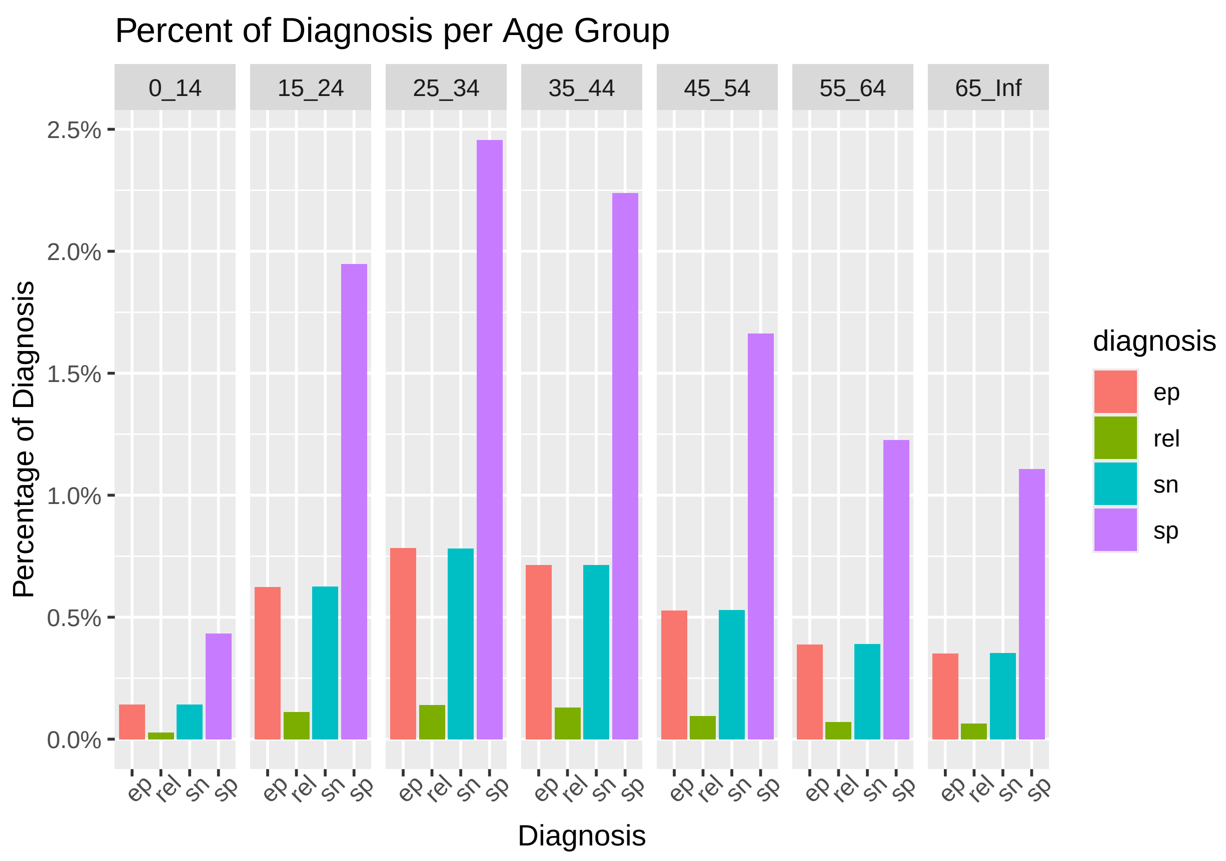
```
# your code here
all_diagnosis <- population_join %>%
  mutate(all_diagnoses = sum(count,na.rm = TRUE))

diagnosis_by_age <-population_join %>%
  group_by(age_range) %>%
  summarize(all_diagnoses_by_age_group = sum(count,na.rm = TRUE))

combined_df <-all_diagnosis %>%
  left_join(diagnosis_by_age,by = "age_range")

combined_df <-combined_df %>%
  group_by(age_range) %>%
  mutate(percent_of_age_group = all_diagnoses_by_age_group/sum(all_diagnoses)) %>%
  drop_na(count)

ggplot(combined_df) +
  geom_col(aes(x= diagnosis, y= percent_of_age_group, fill = diagnosis)) +
      facet_grid(.~age_range) +
        scale_y_continuous(labels=scales::percent_format())+
          labs(title = "Percent of Diagnosis per Age Group"
                 ,x = "Diagnosis"
                 ,y = "Percentage of Diagnosis") +
              theme(axis.text.x = element_text(angle = 45))
```



Percent of Diagnosis per Age Group
0_14
15_24
25_34
35_44
45_54
55_64
65_Inf
2.5%
2.0%
1.5%
1.0%
0.5%
0.0%
Percentage of Diagnosis
ep
rel
sn
sp
Diagnosis
diagnosis
ep
rel
sn
sp