Homework #2

Gbemisola Talabi

11 October 2022

Instructions

1. Please submit your knitted `.pdf` file to the assignment drop box on eLC. If you are still having trouble knitting your file you can submit your `.Rmd` file.
2. The assignment is due October 11, 2022 by 7:00pm EST. This assignment will be graded for accuracy. Please reach out to us if you need help before this time!
3. Please add your name as "author" to the YAML header above.
4. Below each question is a `r` code chunk that can be used to explore the question. Use the space below the code chunk to directly answer the question.

rm(list=ls(all=T))

```
## you can add more, or change...
library(tidyverse)
library(readr)
library(dplyr)
```

Problem Set

1. Import the dataset `tb.rds` into R. Once you get the file, read the dataset in using `read_rds()` and assign it the name "tb". This data set is a time series of tuberculosis incidence (i.e., number of TB cases per 100,000 population per year).

```
tb <- read_rds("../data/tb.rds")
```

2. Run the `colnames()` function to take a look at the dataset column names. You should see that the first column name does not represent the variable well. Rename the first column of "tb" to "country" using the `rename()` function in `dplyr`. Remember back ticks can be used for non-character column names

```
colnames(tb)
```

```
##  [1] "TB incidence, all forms (per 100 000 population per year)"
##  [2] "1990"
##  [3] "1991"
##  [4] "1992"
##  [5] "1993"
##  [6] "1994"
```

```
##  [7] "1995"
##  [8] "1996"
##  [9] "1997"
## [10] "1998"
## [11] "1999"
## [12] "2000"
## [13] "2001"
## [14] "2002"
## [15] "2003"
## [16] "2004"
## [17] "2005"
## [18] "2006"
## [19] "2007"
```

```
tb <- rename(tb, country = "TB incidence, all forms (per 100 000 population per year)")
```

3. Use the `pct_complete()` function in the `naniar` package to determine the percent missing data in "tb". You might need to load and install `naniar`!

```
library(naniar)

naniar::pct_complete(tb)
```

```
## [1] 99.59514
```

4. How many countries that have a complete record in "tb" across all years? Just look at the output here, don't reassign it. **Hint:** look for complete records by dropping all NAs from the dataset using `drop_na()`.

```
drop_na(tb)
```

```
## # A tibble: 207 x 19
##    country `1990` `1991` `1992` `1993` `1994` `1995` `1996` `1997` `1998` `1999`
##    <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
##  1 Afghan~    168    168    168    168    168    168    168    168    168    168
##  2 Albania     25     24     25     26     26     27     27     28     28     27
##  3 Algeria     38     38     39     40     41     42     43     44     46     47
##  4 Americ~     21      7      2      9      9     11      0     12      6      8
##  5 Andorra     36     34     32     30     29     27     26     26     25     23
##  6 Angola     205    209    214    218    222    226    231    236    240    245
##  7 Anguil~     24     24     24     24     23     23     23     23     23     23
##  8 Antigu~     10     10      9      9      8      8      8      7      7      7
##  9 Argent~     60     57     55     53     51     49     47     45     44     42
## 10 Armenia     33     32     33     37     41     47     53     58     63     67
## # ... with 197 more rows, and 8 more variables: `2000` <dbl>, `2001` <dbl>,
## #   `2002` <dbl>, `2003` <dbl>, `2004` <dbl>, `2005` <dbl>, `2006` <dbl>,
## #   `2007` <dbl>
## # i Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

```
nrow(tb)
```

```
## [1] 208
```

```
#208 countries
```

5. How many country names begin with the letter "D"?

```
country_d <- tb %>% filter(stringr::str_starts(country, "D"))
count(country_d)
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1     4
```

```
# Four countries begins with the letter "D"
```

6. Subset the "tb" dataset to only keep the first 10 rows. Call the new data frame "tb_small".

```
tb_small <- tb[ c(1:10), ]
```

7. Bring an additional dataset into R. The dataset is csv file named "WorldBank_population.csv". This is the total estimated population size by country and year. Assign this dataset to "pop".

```
pop <- read_csv("../data/WorldBank_population.csv")
```

```
## Rows: 261 Columns: 46
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (4): Country Name, Country Code, Indicator Name, Indicator Code
## dbl (42): 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, ...
##
## 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.
```

8. Rename the first column in "pop" to "country". Use the `rename()` function. Don't forget to reassign the renamed data to "pop".

```
pop <- rename(pop, country = "Country Name")
```

9. Subset the "pop" data to only keep the first 10 rows. Call the new data frame "pop_small".

```
pop_small <- pop[ c(1:10), ]
```

10. Our goal is to estimate the number of TB cases each year based on "pop_small" and "tb_small" datasets. Therefore, we need to evaluate which countries are overlapping and modify some strings if needed. First, determine which countries do not match or might need cleaning using `setdiff()`. How many countries do not match? Which countries need to be recoded in order to match across the two datasets?

```
tb_country <- tb_small %>% pull(country)
pop_country <- pop_small %>% pull(country)
setdiff(tb_country, pop_country)
```

```
## [1] "Algeria"             "Andorra"             "Anguilla"
## [4] "Antigua and Barbuda"
```

```
setdiff(pop_country, tb_country)
```

```
## [1] "andorra"           "Antigua & Barbuda" "Australia"
## [4] "Austria"
```

```
# Four countries in both datasets do not match each other.
# andorra to Andorra, Antigua & Barbuda to Antigua and Barbuda needs to be recoded.
```

11. Use the `recode()` function to match "pop_small" country names to "tb_small" country names. Then use `setdiff()` again to check it worked.

```
pop_small <- pop_small %>%
  mutate(country = recode(country, "andorra" = "Andorra"))
pop_small <- pop_small %>%
  mutate(country = recode(country, "Antigua & Barbuda" = "Antigua and Barbuda"))
tb_country <- tb_small %>% pull(country)
pop_country <- pop_small %>% pull(country)
setdiff(tb_country, pop_country)
```

```
## [1] "Algeria"  "Anguilla"
```

```
setdiff(pop_country, tb_country)
```

```
## [1] "Australia" "Austria"
```

12. Reshape the "tb_small" data to long form.

- There should be a column for country ("country"), a column for year ("year"), and a column for the TB incidence value ("TB_incidence").
- Use `pivot_longer()`.
- You should pivot all columns except "country".
- **Hint:** listing `!COLUMN_NAME` or `-COLUMN_NAME` means everything except the column you have named.
- Assign the reshaped data to "long_tb".

```
long_tb <- tb_small %>% pivot_longer(!country,
                                     names_to = "year",
                                     values_to = "TB_incidence")
head(long_tb)
```

```
## # A tibble: 6 x 3
##   country     year  TB_incidence
##   <chr>       <chr>        <dbl>
## 1 Afghanistan 1990           168
## 2 Afghanistan 1991           168
## 3 Afghanistan 1992           168
## 4 Afghanistan 1993           168
## 5 Afghanistan 1994           168
## 6 Afghanistan 1995           168
```

13. What is the `typeof()` for the variable `year` in the "long_tb" dataset? If it's not an integer, turn it into integer form with `as.integer()`. Check to make sure it worked.

```
typeof(long_tb$year)
```

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

```
long_tb <- long_tb %>%
  mutate(year = as.integer(year))
typeof(long_tb$year)
```

```
## [1] "integer"
```

14. Subset "long_tb" based on years 1995-2005, including 1995 and 2005 and call this "long_tb_sub" using `&` or the `between()` function. Confirm your filtering worked by looking at the range of "year". .

```
long_tb_sub <- filter(long_tb, between(year, 1995, 2005))
range(long_tb_sub$year)
```

```
## [1] 1995 2005
```

15. Reshape the "pop_small" data to long form.

- There should be a column for country ("country"), a column for year ("year"), and a column for the population value ("population").
- Use `select()` to remove columns that are no longer needed including "Country Code", "Indicator Name", and "Indicator Code"
- Use `pivot_longer()` to pivot year columns.
- Assign the reshaped data to "long_pop".

```
pop_small <- select(pop_small, -c("Country Code", "Indicator Name", "Indicator Code"))
long_pop <- pop_small %>% pivot_longer(!country,
                                names_to = "year",
                                values_to = "population")
head(long_pop)
```

```
## # A tibble: 6 x 3
##   country     year  population
##   <chr>       <chr>      <dbl>
## 1 Afghanistan 1980    13356500
## 2 Afghanistan 1981    13171679
## 3 Afghanistan 1982    12882518
## 4 Afghanistan 1983    12537732
## 5 Afghanistan 1984    12204306
## 6 Afghanistan 1985    11938204
```

16. What is the `typeof()` for the variable `year`in the "long_pop" dataset? If it's not an integer, turn it into integer form with `as.integer()`. Check to make sure it worked.

```
typeof(long_pop$year)
```

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

```
long_pop <- long_pop %>%
  mutate(year = as.integer(year))
typeof(long_pop$year)
```

```
## [1] "integer"
```

17. Subset "long_pop" based on years 1995-2005, including 1995 and 2005 and call this "long_pop_sub". Confirm your filtering worked by looking at the range of "year".

```
long_pop_sub <- filter(long_pop, between(year, 1995, 2005))
range(long_pop_sub$year)
```

```
## [1] 1995 2005
```

18. Using "long_tb_sub" and "long_pop_sub", `inner_join()` the two datasets by "country" and "year". Call this new dataset "joined".

```
joined <- inner_join(long_tb_sub, long_pop_sub, by = c("country", "year"))
head(joined)
```

```
## # A tibble: 6 x 4
##   country      year TB_incidence population
##   <chr>       <int>        <dbl>      <dbl>
## 1 Afghanistan  1995          168   18110662
## 2 Afghanistan  1996          168   18853444
## 3 Afghanistan  1997          168   19357126
## 4 Afghanistan  1998          168   19737770
## 5 Afghanistan  1999          168   20170847
## 6 Afghanistan  2000          168   20779957
```

19. How many unique countries are in the "joined" dataset? Is this as expected given the number of countries that didn't match above? How many unique countries would we have had if we did `right_join(long_tb_sub, long_pop_sub, by=c("country", "year"))`, why?

```
rightjoin <- right_join(long_tb_sub, long_pop_sub, by=c("country", "year"))
head(rightjoin)
```

```
## # A tibble: 6 x 4
##   country      year TB_incidence population
##   <chr>       <int>        <dbl>      <dbl>
## 1 Afghanistan  1995          168   18110662
## 2 Afghanistan  1996          168   18853444
## 3 Afghanistan  1997          168   19357126
## 4 Afghanistan  1998          168   19737770
## 5 Afghanistan  1999          168   20170847
## 6 Afghanistan  2000          168   20779957
```

```
# eight unique countries are in the joined dataset.
#yes, it is as expected because only countries that match in both datasets are kept.
#we have 10 unique countries after running 'right_join'
#because the 'right_join' function ensures all rows(countries) in the long_pop_sub dataset are kept.
```

20. Calculate the number of TB cases for each country and year in the dataset "joined" by multiplying the size of the population by TB incidence divided by 100,000. Name this new variable "TB_cases"

```
joined <- joined %>% mutate(TB_cases = population * TB_incidence / 100000)
head(joined)
```

```
## # A tibble: 6 x 5
##   country      year TB_incidence population TB_cases
##   <chr>       <int>        <dbl>      <dbl>    <dbl>
## 1 Afghanistan  1995          168   18110662   30426.
## 2 Afghanistan  1996          168   18853444   31674.
## 3 Afghanistan  1997          168   19357126   32520.
## 4 Afghanistan  1998          168   19737770   33159.
## 5 Afghanistan  1999          168   20170847   33887.
## 6 Afghanistan  2000          168   20779957   34910.
```

21. Make a scatter plot the number of TB cases over time in Afghanistan

```
Afghan_country <- filter(joined, country == "Afghanistan")
library(ggplot2)
plot(x = Afghan_country$year, y = Afghan_country$TB_cases)
```