Lecture Assignment 12

Viraj Vijaywargiya

2022-05-11

```
library(tidyverse)
```

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

## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.8
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1

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

Part 12.2.1

Question 1

```
table1
```

```
## # A tibble: 6 x 4
##   country      year  cases population
##   <chr>       <int>  <int>      <int>
## 1 Afghanistan  1999    745   19987071
## 2 Afghanistan  2000   2666   20595360
## 3 Brazil       1999  37737  172006362
## 4 Brazil       2000  80488  174504898
## 5 China        1999 212258 1272915272
## 6 China        2000 213766 1280428583
```

In table 1, each variable (country, year, cases, and population) has its own column, and the columns contain the values for these variables, associated with each other. Each observation has its own row, that represents the **country** and the **cases** and **population** associated with it for the **year**s given.

```
table2
```

```
## # A tibble: 12 x 4
##    country      year type            count
##    <chr>       <int> <chr>           <int>
##  1 Afghanistan  1999 cases             745
##  2 Afghanistan  1999 population   19987071
##  3 Afghanistan  2000 cases            2666
##  4 Afghanistan  2000 population   20595360
##  5 Brazil       1999 cases           37737
##  6 Brazil       1999 population  172006362
##  7 Brazil       2000 cases           80488
##  8 Brazil       2000 population  174504898
##  9 China        1999 cases          212258
## 10 China        1999 population 1272915272
## 11 China        2000 cases          213766
## 12 China        2000 population 1280428583
```

In table 2, each observation is a row representing a combination of country, year, type of variable (cases and population) and count associated with the type. For variables, **country** and **year** is represented as 2 separate columns, whereas **cases** and **population** is represented using the columns type and count, where type is either **cases** or **population** and count contains their respective type values.

```
table3
```

```
## # A tibble: 6 x 3
##   country      year rate
## * <chr>       <int> <chr>
## 1 Afghanistan  1999 745/19987071
## 2 Afghanistan  2000 2666/20595360
## 3 Brazil       1999 37737/172006362
## 4 Brazil       2000 80488/174504898
## 5 China        1999 212258/1272915272
## 6 China        2000 213766/1280428583
```

In table 3, each observation is a row representing a combination of country, year, and rate (dividing the variables cases/population). For variables, **country** and **year** is represented as 2 separate columns, whereas the column, rate, uses the values of **cases** and **population** as characters in the format "**cases**/**population**".

```
table4a
```

```
## # A tibble: 3 x 3
##   country     `1999` `2000`
## * <chr>        <int>  <int>
## 1 Afghanistan    745   2666
## 2 Brazil       37737  80488
## 3 China       212258 213766
```

```
table4b
```

```
## # A tibble: 3 x 3
##   country         `1999`     `2000`
## * <chr>            <int>      <int>
## 1 Afghanistan   19987071   20595360
## 2 Brazil       172006362  174504898
## 3 China       1272915272 1280428583
```

The table4 is split into two tables, table4a and table4b.

In table4a, each observation is a row representing a combination of country and the years, 1999 and 2000, showing their respective cases. For variables, **country** is represented as a separate column, whereas, the columns, 1999 and 2000, are **year**s that contain their respective **cases** as values.

In table4b, each observation is a row representing a combination of country and the years, 1999 and 2000, showing their respective population. For variables, **country** is represented as a separate column, whereas, the columns, 1999 and 2000, are **year**s that contain their respective **population** values.

Question 2

Computing the rate for table2,

```
cases2 <- filter(table2, type == "cases") %>%
  rename(cases = count) %>%
  arrange(country, year)

population2 <- filter(table2, type == "population") %>%
  rename(population = count) %>%
  arrange(country, year)

all_cases <- tibble(year = cases2$year, country = cases2$country,
  cases = cases2$cases, population = population2$population) %>%
  mutate(final_cases = (cases / population) * 10000) %>%
  select(country, year, final_cases)

all_cases <- all_cases %>%
  mutate(type = "final_cases") %>%
  rename(count = final_cases)

bind_rows(table2, all_cases) %>%
  arrange(country, year, type, count)
```

```
## # A tibble: 18 x 4
##    country      year type           count
##    <chr>       <int> <chr>          <dbl>
##  1 Afghanistan  1999 cases       7.45e+2
##  2 Afghanistan  1999 final_cases 3.73e-1
##  3 Afghanistan  1999 population  2.00e+7
##  4 Afghanistan  2000 cases       2.67e+3
##  5 Afghanistan  2000 final_cases 1.29e+0
##  6 Afghanistan  2000 population  2.06e+7
##  7 Brazil       1999 cases       3.77e+4
##  8 Brazil       1999 final_cases 2.19e+0
##  9 Brazil       1999 population  1.72e+8
## 10 Brazil       2000 cases       8.05e+4
## 11 Brazil       2000 final_cases 4.61e+0
## 12 Brazil       2000 population  1.75e+8
## 13 China        1999 cases       2.12e+5
## 14 China        1999 final_cases 1.67e+0
## 15 China        1999 population  1.27e+9
## 16 China        2000 cases       2.14e+5
## 17 China        2000 final_cases 1.67e+0
## 18 China        2000 population  1.28e+9
```

Computing the rate for table4a + table4b,

```
table4_cases <- tibble( country = table4a$country,
    `1999` = table4a[["1999"]] / table4b[["1999"]] * 10000,
    `2000` = table4a[["2000"]] / table4b[["2000"]] * 10000)
table4_cases
```

```
## # A tibble: 3 x 3
##   country     ‘1999‘ ‘2000‘
##   <chr>        <dbl>  <dbl>
## 1 Afghanistan  0.373   1.29
## 2 Brazil       2.19    4.61
## 3 China        1.67    1.67
```

table4a + table4b was easier to work with, than that with table2, as it had split the variables, cases and population, into 2 different tables already so that made it easier to divide cases by population. For table2, we had to create a table with columns for cases and population because it had separate rows for cases and population. So, table2 was harder to work with. However, an ideal table that would be the easiest to work with would be where there are separate columns for country, year, cases, and population. With this, computing cases could be done by just using the mutate() function.

Part 12.3.3

Question 1

```
stocks <- tibble(
  year   = c(2015, 2015, 2016, 2016),
  half   = c(   1,    2,     1,    2),
  return = c(1.88, 0.59, 0.92, 0.17)
)
stocks %>%
  pivot_wider(names_from = year, values_from = return) %>%
  pivot_longer(`2015`:`2016`, names_to = "year", values_to = "return")
```

```
## # A tibble: 4 x 3
##    half year  return
##   <dbl> <chr>  <dbl>
## 1     1 2015    1.88
## 2     1 2016    0.92
## 3     2 2015    0.59
## 4     2 2016    0.17
```

```
glimpse(stocks)
```

```
## Rows: 4
## Columns: 3
## $ year   <dbl> 2015, 2015, 2016, 2016
## $ half   <dbl> 1, 2, 1, 2
## $ return <dbl> 1.88, 0.59, 0.92, 0.17
```

When a data frame is converted from wide to long, column type information is lost, therefore, the functions pivot_longer() and pivot_wider() not perfectly symmetrical.

The names_ptypes argument fails to convert the year column to a numeric vector which throws an error.

Question 2

The columns, 1999 and 2000, are not non-syntactic variable names. Therefore, the column names must be used with backticks (‘) or as strings to select the columns, 1999 and 2000. For example,

```
table4a %>%
  pivot_longer(c(`1999`, `2000`), names_to = "year", values_to = "cases")
```

```
## # A tibble: 6 x 3
##   country     year   cases
##   <chr>       <chr>  <int>
## 1 Afghanistan 1999     745
## 2 Afghanistan 2000    2666
## 3 Brazil      1999   37737
## 4 Brazil      2000   80488
## 5 China       1999  212258
## 6 China       2000  213766
```

Question 3

```
people <- tribble(
  ~name,             ~names,  ~values,
  #-----------------/--------/------
  "Phillip Woods",   "age",       45,
  "Phillip Woods",   "height",   186,
  "Phillip Woods",   "age",       50,
  "Jessica Cordero", "age",       37,
  "Jessica Cordero", "height",   156
)
glimpse(people)
```

```
## Rows: 5
## Columns: 3
## $ name   <chr> "Phillip Woods", "Phillip Woods", "Phillip Woods", "Jessica Cor~
## $ names  <chr> "age", "height", "age", "age", "height"
## $ values <dbl> 45, 186, 50, 37, 156
```

If you widen this table using pivot_wider, it will create columns that are lists of numeric vectors as the columns, name and names, do not identify the rows uniquely.

To uniquely identify each value, we can create a new variable that has the count for different observations for every combination of name and names.

```
people2 <- people %>%
  group_by(name, names) %>%
  mutate(diff_obs = row_number())
people2
```

```
## # A tibble: 5 x 4
## # Groups:   name, names [4]
##   name            names  values diff_obs
##   <chr>           <chr>   <dbl>    <int>
## 1 Phillip Woods   age        45        1
## 2 Phillip Woods   height    186        1
## 3 Phillip Woods   age        50        2
## 4 Jessica Cordero age        37        1
## 5 Jessica Cordero height    156        1
```