Data Importing and "Tidy" Data

El Mex

Contents

1 Basic functions for data tidying **1**

1.1 Using `readr` 1

1.2 "Tidy" data 2

1.3 Converting to "tidy" data 3

1.4 Case study: Democracy in Guatemala 6

1 Basic functions for data tidying

```
# import the libraries to be used
library(dplyr)
library(ggplot2)
library(readr)
library(tidyr)
library(nycflights13)
library(fivethirtyeight)
```

1.1 Using `readr`

```
# import the ".csv" file by using "read_csv()" function
library(readr)
dem_score <- read_csv("dem_score.csv")

head(dem_score)
```

```
## # A tibble: 6 x 10
##   country    `1952` `1957` `1962` `1967` `1972` `1977` `1982` `1987` `1992`
##   <chr>       <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 Albania        -9     -9     -9     -9     -9     -9     -9     -9      5
## 2 Argentina      -9     -1     -1     -9     -9     -9     -8      8      7
## 3 Armenia        -9     -7     -7     -7     -7     -7     -7     -7      7
## 4 Australia      10     10     10     10     10     10     10     10     10
## 5 Austria        10     10     10     10     10     10     10     10     10
## 6 Azerbaijan     -9     -7     -7     -7     -7     -7     -7     -7      1
```

1.2 "Tidy" data

```
# explore the data frame "drinks" from the "fivethirtyeight" package
glimpse(drinks)
```

```
## Rows: 193
## Columns: 5
## $ country                       <chr> "Afghanistan", "Albania", "Algeria", "...
## $ beer_servings                 <int> 0, 89, 25, 245, 217, 102, 193, 21, 261...
## $ spirit_servings               <int> 0, 132, 0, 138, 57, 128, 25, 179, 72, ...
## $ wine_servings                 <int> 0, 54, 14, 312, 45, 45, 221, 11, 212, ...
## $ total_litres_of_pure_alcohol <dbl> 0.0, 4.9, 0.7, 12.4, 5.9, 4.9, 8.3, 3....
```

```
head(drinks)
```

```
## # A tibble: 6 x 5
##   country      beer_servings spirit_servings wine_servings total_litres_of_pure~
##   <chr>                <int>           <int>         <int>                  <dbl>
## 1 Afghanistan              0               0             0                    0
## 2 Albania                 89             132            54                    4.9
## 3 Algeria                 25               0            14                    0.7
## 4 Andorra                245             138           312                   12.4
## 5 Angola                 217              57            45                    5.9
## 6 Antigua & B~           102             128            45                    4.9
```

Apply these actions:

1. `filter()` the drinks data frame to only consider 4 countries: USA, China, Italy, and Saudi Arabia.
2. `select()` all columns except `total_litres_of_pure_alcohol` by using the - sign
3. `rename()` the variables `beer_servings`, `spirit_servings`, and `wine_servings` to `beer`, `spirit`, and `wine`, respectively.
4. Save all in a variable called `drinks_smaller`

```
drinks_smaller <- drinks %>%
  filter(country %in% c("USA", "China", "Italy", "Saudi Arabia")) %>%
  select(-total_litres_of_pure_alcohol) %>%
  rename(beer = beer_servings, spirit = spirit_servings, wine = wine_servings)

drinks_smaller
```

```
## # A tibble: 4 x 4
##   country       beer spirit  wine
##   <chr>        <int>  <int> <int>
## 1 China           79    192     8
## 2 Italy           85     42   237
## 3 Saudi Arabia     0      5     0
## 4 USA            249    158    84
```

1.3 Converting to "tidy" data

Most original data frames are in wide (non-"tidy") format as it is above. Use `pivot_longer()` function in the `tidyr` package to convert it to "tidy" format

Reminder:

1. `names_to` is the name of the variable in the new "tidy"/long data frame that will contain the *column names* of the original data
2. `values_to` is the name of the variable in the new "tidy" data frame that will contain the *values* of the original data
3. `cols` is the columns in the data frame you either want to or don't want to "tidy"

```
# We don't want to tidy it to the "country" column so we put "-" sign to country (-country)
drinks_smaller_tidy <- drinks_smaller %>%
  pivot_longer(names_to = "type",
               values_to = "servings",
               cols = -country)

drinks_smaller_tidy
```

```
## # A tibble: 12 x 3
##    country      type   servings
##    <chr>        <chr>     <int>
##  1 China        beer         79
##  2 China        spirit      192
##  3 China        wine          8
##  4 Italy        beer         85
##  5 Italy        spirit       42
##  6 Italy        wine        237
##  7 Saudi Arabia beer          0
##  8 Saudi Arabia spirit        5
##  9 Saudi Arabia wine          0
## 10 USA          beer        249
## 11 USA          spirit      158
## 12 USA          wine         84
```

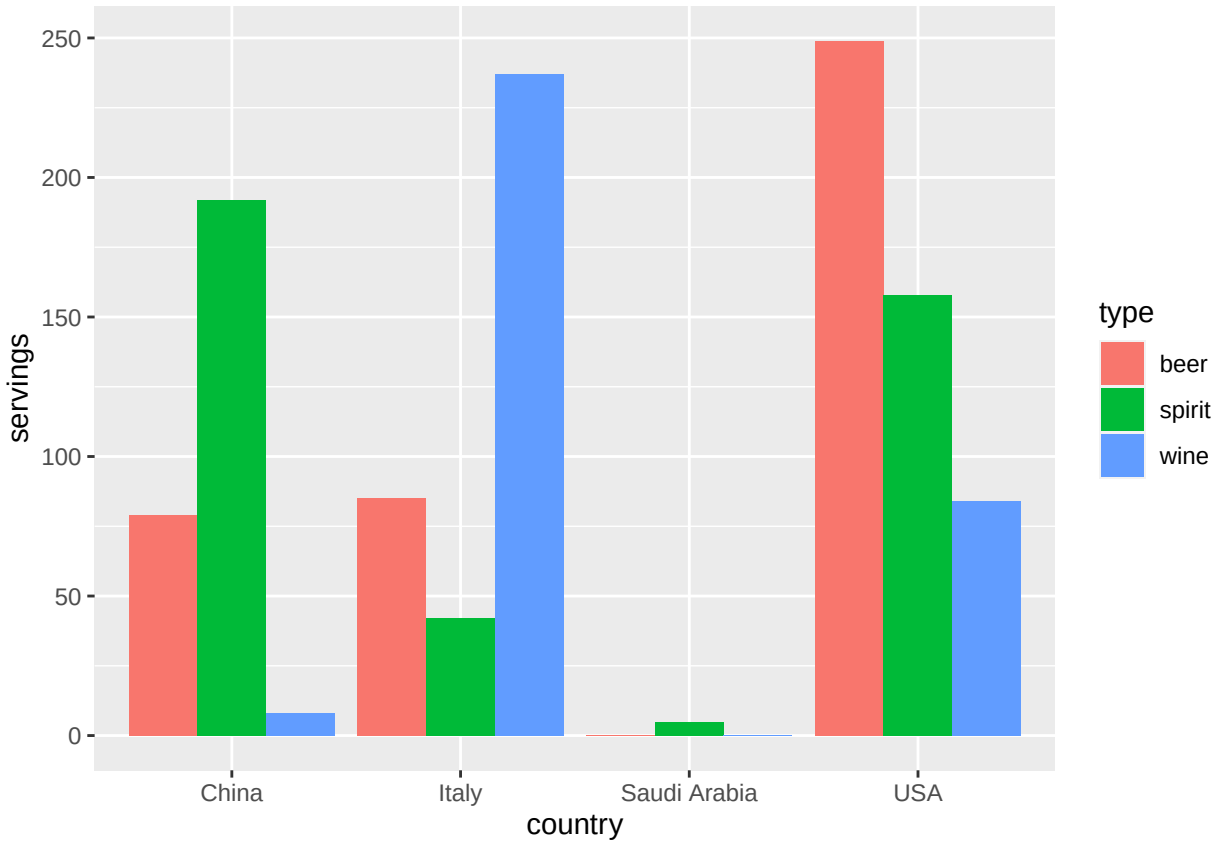
As the variable names of `type` & `servings` don't appear in `drinks_smaller`, we use quotation marks around it, on the other hand, `country` appear in `drinks_smaller` so no need for quotation marks

```
# another way to get the same results
drinks_smaller %>%
  pivot_longer(names_to = "type",
               values_to = "servings",
               cols = c(beer, spirit, wine))

# since these three columns appear one after another in "drinks_smaller", we could also do this:
drinks_smaller %>%
  pivot_longer(names_to = "type",
               values_to = "servings",
               cols = beer:wine)
```

Create the side-by-side barplot:

1. The categorical variable `country` with four levels (China, Italy, Saudi Arabia, USA) would have to be mapped to the `x`-position of the bars
2. The numerical variable `servings` would have to be mapped to the `y`-position of the bars (the height of the bars)
3. The categorical variable `type` with three levels (beer, spirit, wine) would have to be mapped to the `fill` color of the bars

```
ggplot(drinks_smaller_tidy, aes(x = country, y = servings, fill = type)) +
  geom_col(position = "dodge")
```



Let's do another exercise using the `airline_safety` data frame in the `fivethirtyeight` package

```
# explore it
glimpse(airline_safety)
```

```
## Rows: 56
## Columns: 9
## $ airline                <chr> "Aer Lingus", "Aeroflot", "Aerolineas Argent...
## $ incl_reg_subsidiaries  <lgl> FALSE, TRUE, FALSE, TRUE, FALSE, FALSE, TRUE...
## $ avail_seat_km_per_week <dbl> 320906734, 1197672318, 385803648, 596871813,...
## $ incidents_85_99        <int> 2, 76, 6, 3, 2, 14, 2, 3, 5, 7, 3, 21, 1, 5,...
## $ fatal_accidents_85_99  <int> 0, 14, 0, 1, 0, 4, 1, 0, 0, 2, 1, 5, 0, 3, 0...
```

```
## $ fatalities_85_99       <int> 0, 128, 0, 64, 0, 79, 329, 0, 0, 50, 1, 101,...
## $ incidents_00_14        <int> 0, 6, 1, 5, 2, 6, 4, 5, 5, 4, 7, 17, 1, 0, 6...
## $ fatal_accidents_00_14  <int> 0, 1, 0, 0, 0, 2, 1, 1, 1, 0, 0, 3, 0, 0, 0,...
## $ fatalities_00_14       <int> 0, 88, 0, 0, 0, 337, 158, 7, 88, 0, 0, 416, ...
```

```
airline_safety_smaller <- airline_safety %>%
  select(airline, starts_with("fatalities"))

airline_safety_smaller
```

```
## # A tibble: 56 x 3
##    airline               fatalities_85_99 fatalities_00_14
##    <chr>                            <int>            <int>
##  1 Aer Lingus                           0                0
##  2 Aeroflot                           128               88
##  3 Aerolineas Argentinas                0                0
##  4 Aeromexico                          64                0
##  5 Air Canada                           0                0
##  6 Air France                          79              337
##  7 Air India                          329              158
##  8 Air New Zealand                      0                7
##  9 Alaska Airlines                      0               88
## 10 Alitalia                            50                0
## # ... with 46 more rows
```

How would you convert this data frame to be in "tidy" format, in particular so that it has a variable `fatalities_years` indicating the incident year and a variable `count` of the fatality counts?

```
airline_tidy <- airline_safety_smaller %>%
  pivot_longer(names_to = "fatalities_years",
               values_to = "count",
               cols = -airline)

airline_tidy
```

```
## # A tibble: 112 x 3
##    airline               fatalities_years count
##    <chr>                 <chr>            <int>
##  1 Aer Lingus            fatalities_85_99     0
##  2 Aer Lingus            fatalities_00_14     0
##  3 Aeroflot              fatalities_85_99   128
##  4 Aeroflot              fatalities_00_14    88
##  5 Aerolineas Argentinas fatalities_85_99     0
##  6 Aerolineas Argentinas fatalities_00_14     0
##  7 Aeromexico            fatalities_85_99    64
##  8 Aeromexico            fatalities_00_14     0
##  9 Air Canada            fatalities_85_99     0
## 10 Air Canada            fatalities_00_14     0
## # ... with 102 more rows
```

1.4 Case study: Democracy in Guatemala

Let's use the `dem_score` data frame we imported, but focus on only data corresponding to Guatemala

```
guat_dem <- dem_score %>%
  filter(country == "Guatemala")

guat_dem
```

```
## # A tibble: 1 x 10
##   country   `1952` `1957` `1962` `1967` `1972` `1977` `1982` `1987` `1992`
##   <chr>      <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1 Guatemala      2     -6     -5      3      1     -3     -7      3      3
```

Our data isn't tidy. Create a data frame with three columns: `country`, `year`, and `democracy_score`

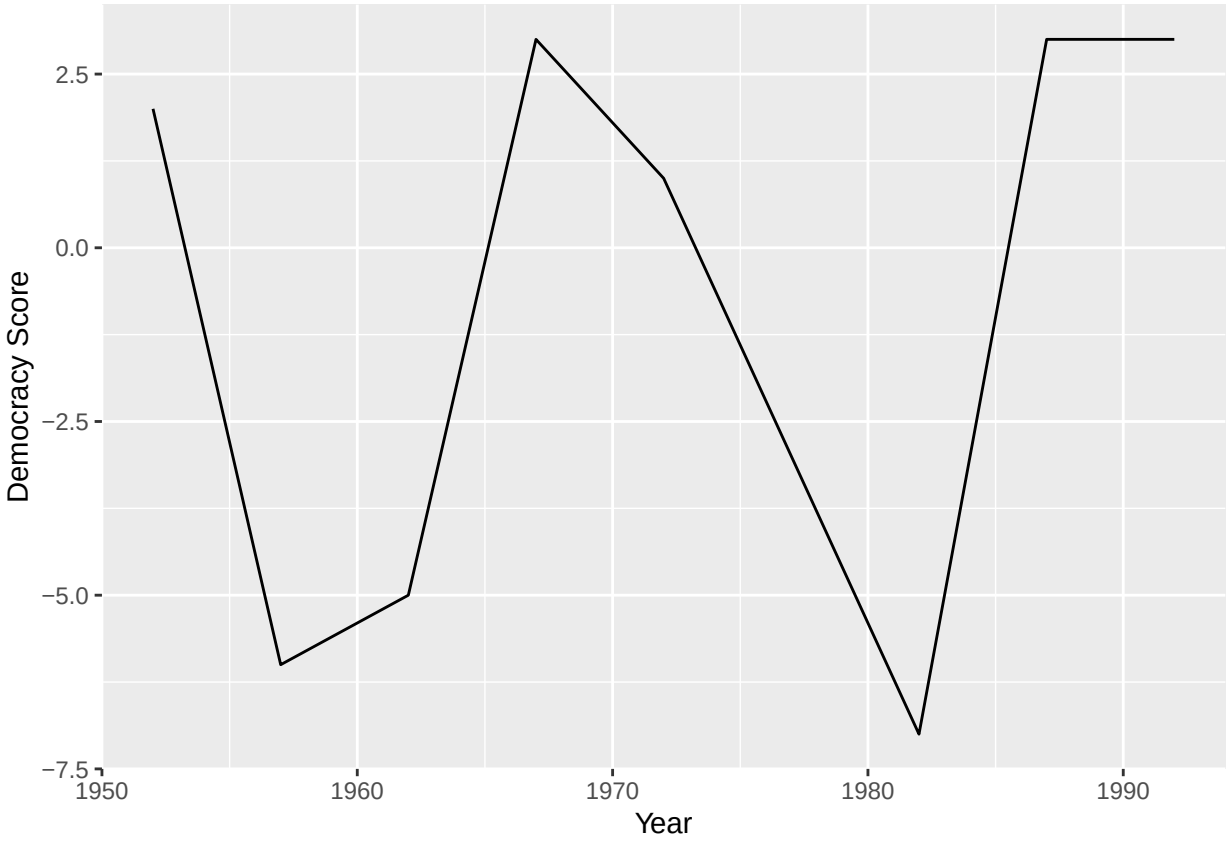
```
 guat_dem_tidy <- guat_dem %>%
  pivot_longer(names_to = "year",
               values_to = "democracy_score",
               cols = -country,
               names_transform = list(year = as.integer))

guat_dem_tidy
```

```
## # A tibble: 9 x 3
##   country    year democracy_score
##   <chr>     <int>           <dbl>
## 1 Guatemala  1952               2
## 2 Guatemala  1957              -6
## 3 Guatemala  1962              -5
## 4 Guatemala  1967               3
## 5 Guatemala  1972               1
## 6 Guatemala  1977              -3
## 7 Guatemala  1982              -7
## 8 Guatemala  1987               3
## 9 Guatemala  1992               3
```

The original column names ('1952', '1957'...) were made of a list of characters, so we had to convert the new variable `year` as integers

Create a time-series plot

```
ggplot(guat_dem_tidy, aes(x = year, y = democracy_score)) +
  geom_line() +
  labs(x = "Year", y = "Democracy Score")
```


2.5
0.0
−2.5
−5.0
−7.5
Democracy Score
1950
1960
1970
1980
1990
Year