R Bootcamp: Loading Data

August 23-24, 2021

UBC MFRE | Master of Food and Resource Economics

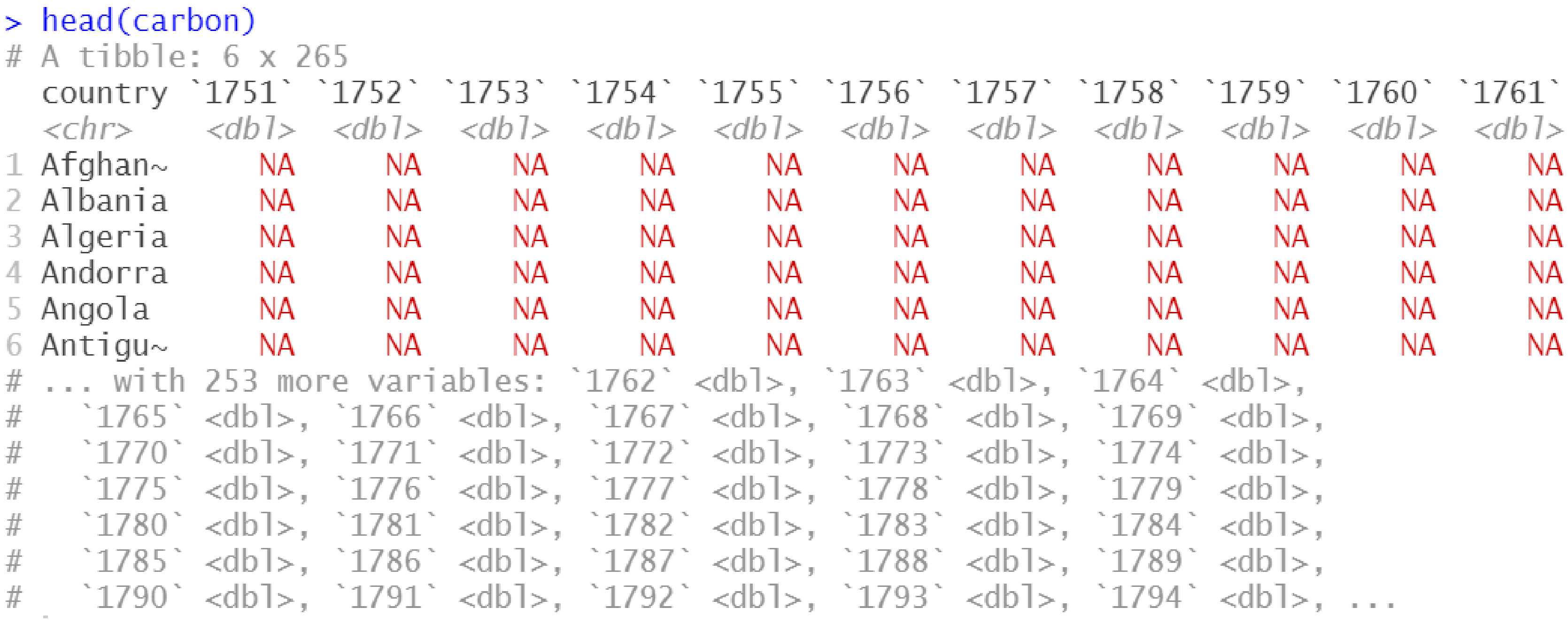
Learning Objectives

- Load different files into R using {base} and {readr} packages
  - Flat files (csv and txt)
  - Excel file (xlsx)
  - Stata data format (dta)
  - Google Sheet
  - Statistics Canada data through an API
- Specify which sheet of an Excel spreadsheet to load
- Tell R to skip empty rows when loading data

Flat files

- csv files – read_csv() from {readr}

```
carbon <- read_csv(here("data", "yearly_co2_emissions.csv"))
```



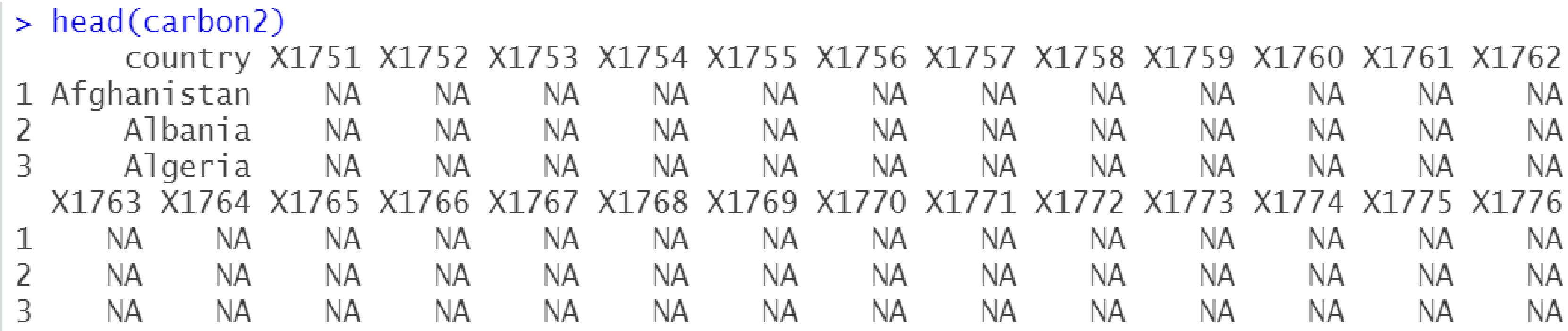
```
> head(carbon)
# A tibble: 6 x 265
  country `1751` `1752` `1753` `1754` `1755` `1756` `1757` `1758` `1759` `1760` `1761`
  <chr>    <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
1 Afghan~     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
2 Albania     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
3 Algeria     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
4 Andorra     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
5 Angola      NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
6 Antigu~     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA     NA
# ... with 253 more variables: `1762` <dbl>, `1763` <dbl>, `1764` <dbl>,
#   `1765` <dbl>, `1766` <dbl>, `1767` <dbl>, `1768` <dbl>, `1769` <dbl>,
#   `1770` <dbl>, `1771` <dbl>, `1772` <dbl>, `1773` <dbl>, `1774` <dbl>,
#   `1775` <dbl>, `1776` <dbl>, `1777` <dbl>, `1778` <dbl>, `1779` <dbl>,
#   `1780` <dbl>, `1781` <dbl>, `1782` <dbl>, `1783` <dbl>, `1784` <dbl>,
#   `1785` <dbl>, `1786` <dbl>, `1787` <dbl>, `1788` <dbl>, `1789` <dbl>,
#   `1790` <dbl>, `1791` <dbl>, `1792` <dbl>, `1793` <dbl>, `1794` <dbl>, ...
```

Additional info: https://medium.com/r-tutorials/r-functions-daily-read-csv-3c418c25cba4#:~:text=The%20read_csv%20function%20imports%20data,load%20faster

Flat files

- csv files – read.csv() from {base}

carbon <- read.csv(here("data", "yearly_co2_emissions.csv"))



```
> head(carbon2)
      country X1751 X1752 X1753 X1754 X1755 X1756 X1757 X1758 X1759 X1760 X1761 X1762
1 Afghanistan    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
2     Albania    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
3     Algeria    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
  X1763 X1764 X1765 X1766 X1767 X1768 X1769 X1770 X1771 X1772 X1773 X1774 X1775 X1776
1    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
2    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
3    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA    NA
```

Additional info: https://medium.com/r-tutorials/r-functions-daily-read-csv-3c418c25cba4#:~:text=The%20read_csv%20function%20imports%20data,load%20faster

Flat files

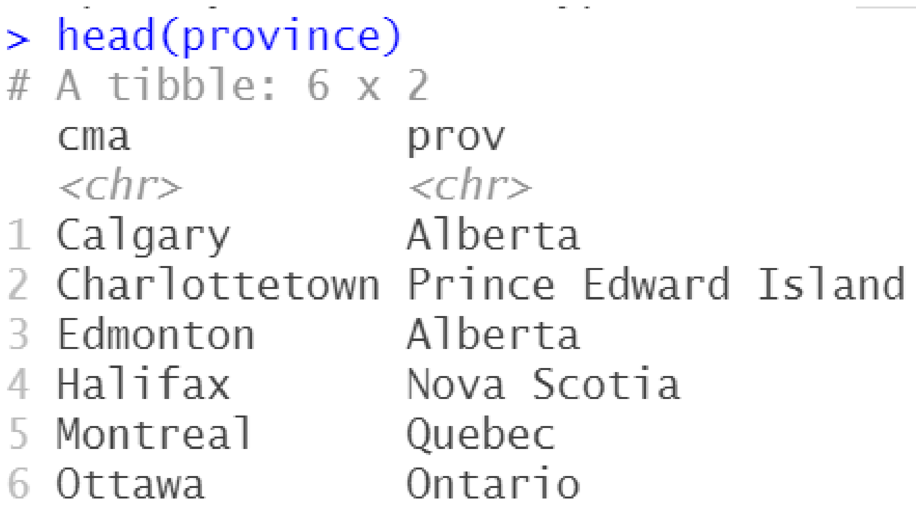
- read_csv() or read.csv()?
  - Both will load your csv files. The main difference is that read_csv() will create a tibble on the backend while the read.csv() will create a data.frame.
  - read_csv() will load bigger files faster.

- In both cases, you noticed that the data is a bit messy because of the variable names. R does not like variable names that start with a number.
  - Tibbles allow for that but will enclose the variables with a backtick. (`1751`, `1752`, …)
  - Dataframes will put an X in front (X1751, X1752, …)

Flat files

- txt files - read_tsv() from {readr}

province <- read_tsv(here("data", "province.txt"))

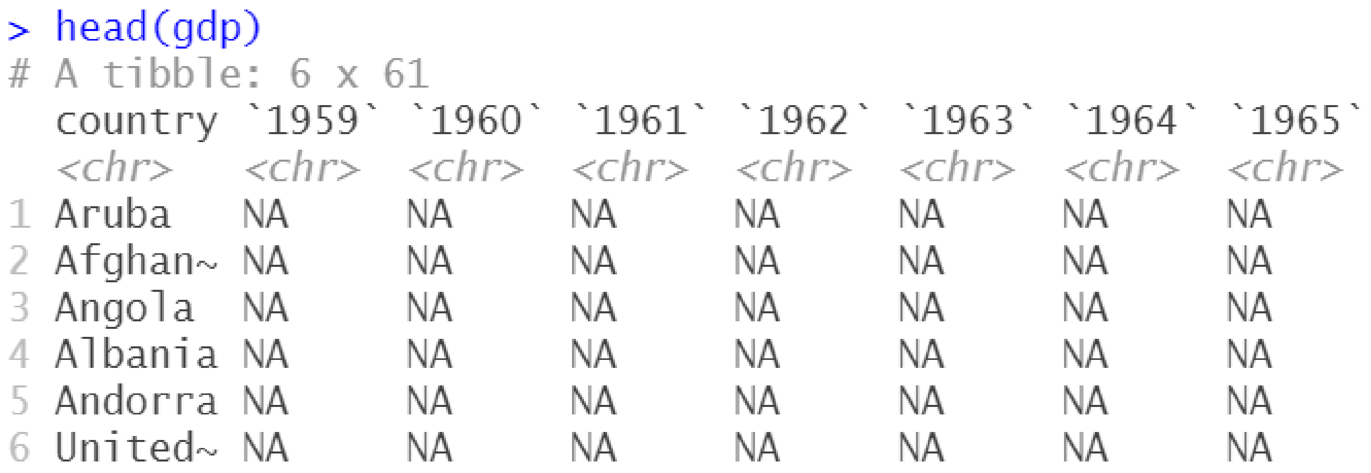


```
> head(province)
# A tibble: 6 x 2
  cma           prov
  <chr>         <chr>
1 Calgary       Alberta
2 Charlottetown Prince Edward Island
3 Edmonton      Alberta
4 Halifax       Nova Scotia
5 Montreal      Quebec
6 Ottawa        Ontario
```

Excel files

- xlsx files – read_xlsx() from {readxl}

```
gdp <- read_excel(here("data", "gdp_pc.xlsx"))
```



```
> head(gdp)
# A tibble: 6 x 61
  country `1959` `1960` `1961` `1962` `1963` `1964` `1965`
  <chr>   <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr> 
1 Aruba   NA     NA     NA     NA     NA     NA     NA    
2 Afghan~ NA     NA     NA     NA     NA     NA     NA    
3 Angola  NA     NA     NA     NA     NA     NA     NA    
4 Albania NA     NA     NA     NA     NA     NA     NA    
5 Andorra NA     NA     NA     NA     NA     NA     NA    
6 United~ NA     NA     NA     NA     NA     NA     NA    
```

Excel files

- If data is stored in separate sheets, use the `sheet` argument. You can specify the sheet by
  - Sheet number

```
energy_hist <- read_xlsx(here("data", "energy_use_per_person.xlsx"), sheet = 1)
energy_new <- read_xlsx(here("data", "energy_use_per_person.xlsx"), sheet = 2)
```

  - Sheet name

```
energy_hist2 <- read_excel(here("data", "energy_use_per_person.xlsx"), sheet = "hist")
energy_new <- read_excel(here("data", "energy_use_per_person.xlsx"), sheet = "recent")
```

UBC
MFRE

Merge

- Let’s merge or join the two energy files we just loaded into R.
- Can use full_join() function of {dplyr}

```
energy <- full_join(energy_hist, energy_new, by = c("country"))
```

- Can also use merge() function of {base} but need to specify that we are doing a full join using the all.x = T and all.y = T arguments

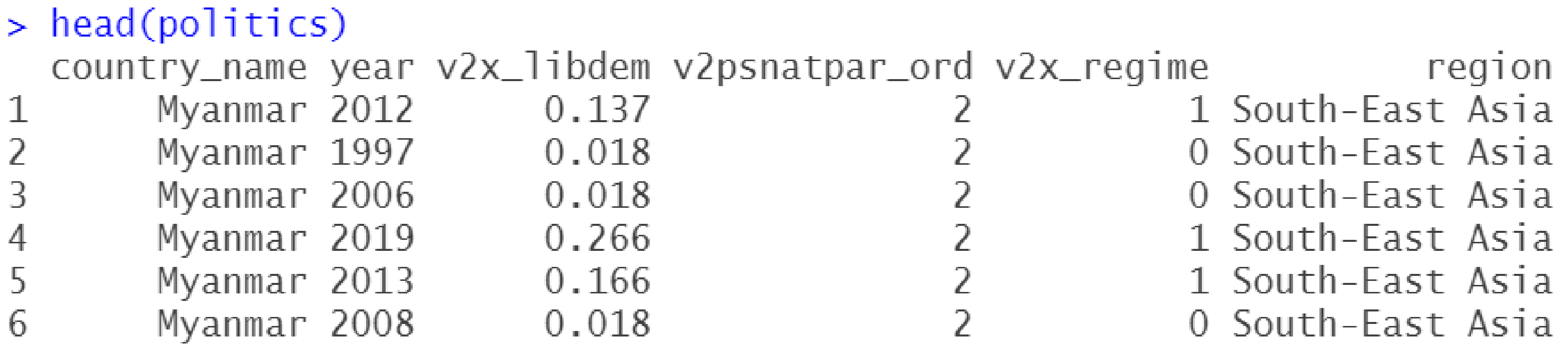
```
energy_basemerge <- merge(energy_hist, energy_new, by =
c("country"), all.x = TRUE, all.y = TRUE)
```

UBC
MFRE

Stata .dta files

- Stata files – read.dta13() of {readstata13}

```
politics <- read.dta13(here("data", "politics.dta"))
```



```
> head(politics)
  country_name year v2x_libdem v2psnatpar_ord v2x_regime          region
1      Myanmar 2012      0.137              2          1 South-East Asia
2      Myanmar 1997      0.018              2          0 South-East Asia
3      Myanmar 2006      0.018              2          0 South-East Asia
4      Myanmar 2019      0.266              2          1 South-East Asia
5      Myanmar 2013      0.166              2          1 South-East Asia
6      Myanmar 2008      0.018              2          0 South-East Asia
```

Stata .dta files

- You will notice a warning that factor codes were identified. This warning means that in Stata, some variables were coded as factors (usually dummy or categorical variables)

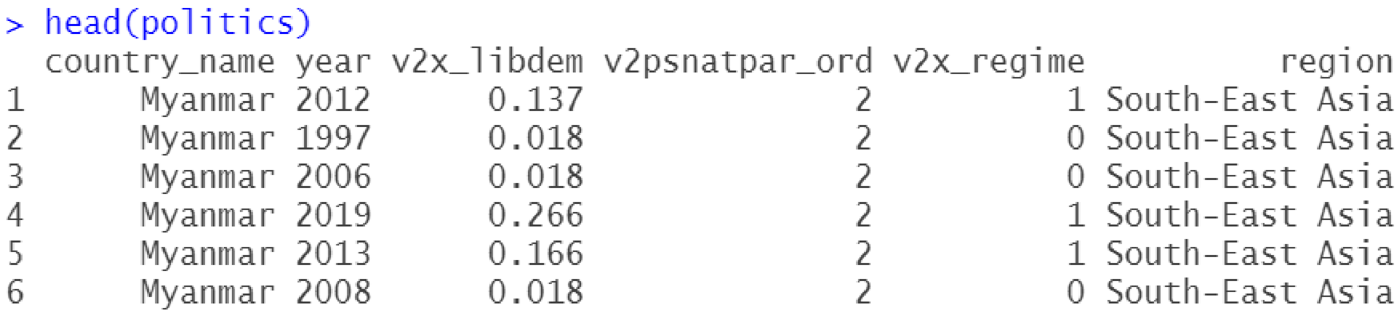
- We can add the argument `nonint.factors = TRUE` to keep factor labels instead of the value itself.

```
politics <- read.dta13(here("data", "politics.dta"), nonint.factors = TRUE)
```

UBC
MFRE

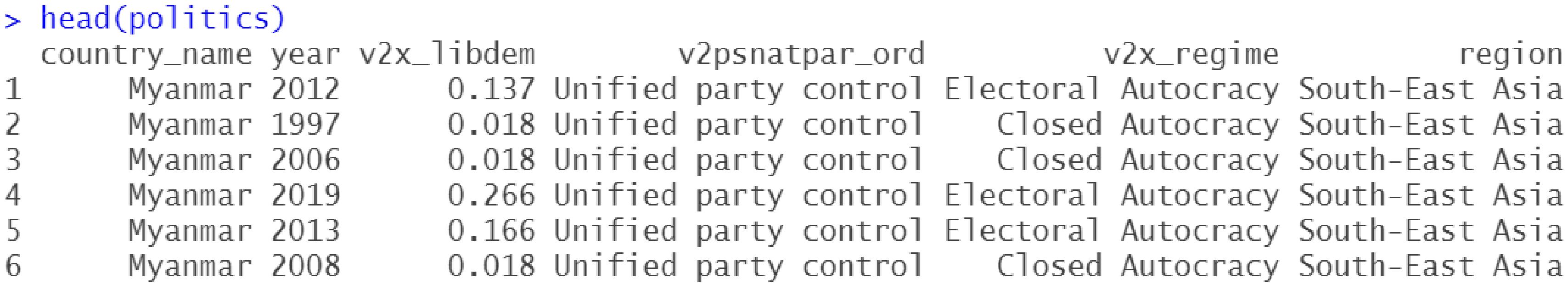
Stata .dta files

- politics <- read.dta13(here("data", "politics.dta"))



```
> head(politics)
  country_name year v2x_libdem v2psnatpar_ord v2x_regime          region
1      Myanmar 2012      0.137              2          1 South-East Asia
2      Myanmar 1997      0.018              2          0 South-East Asia
3      Myanmar 2006      0.018              2          0 South-East Asia
4      Myanmar 2019      0.266              2          1 South-East Asia
5      Myanmar 2013      0.166              2          1 South-East Asia
6      Myanmar 2008      0.018              2          0 South-East Asia
```

- politics <- read.dta13(here("data", "politics.dta"), nonint.factors = TRUE)



```
> head(politics)
  country_name year v2x_libdem       v2psnatpar_ord          v2x_regime          region
1      Myanmar 2012      0.137 Unified party control Electoral Autocracy South-East Asia
2      Myanmar 1997      0.018 Unified party control    Closed Autocracy South-East Asia
3      Myanmar 2006      0.018 Unified party control    Closed Autocracy South-East Asia
4      Myanmar 2019      0.266 Unified party control Electoral Autocracy South-East Asia
5      Myanmar 2013      0.166 Unified party control Electoral Autocracy South-East Asia
6      Myanmar 2008      0.018 Unified party control    Closed Autocracy South-East Asia
```

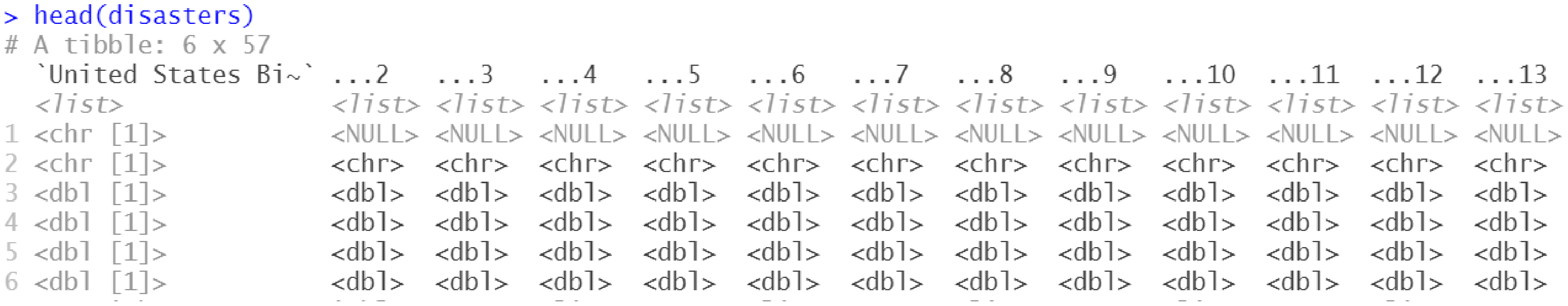
UBC
MFRE

Google Sheets

- Google Sheet files – read_sheet() of {googlesheets4}

gs4_deauth() # so no need to sign in to Google

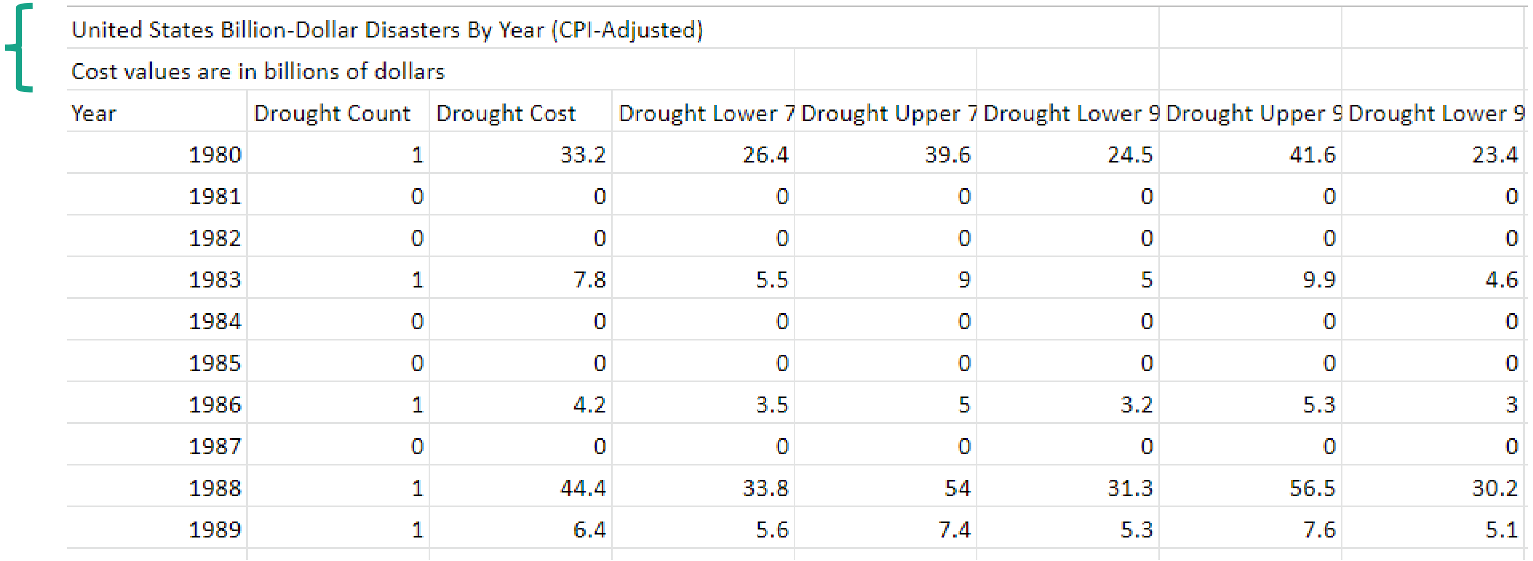
disasters <- read_sheet("https://docs.google.com/spreadsheets/d/17s15o7jdDpGSKgsIboZdnYU2UxHtU9DHKNRmYVVgwJo/edit#gid=0")



```
> head(disasters)
# A tibble: 6 x 57
  `United States Bi~` ...2   ...3   ...4   ...5   ...6   ...7   ...8   ...9   ...10  ...11  ...12  ...13
  <list>              <list> <list> <list> <list> <list> <list> <list> <list> <list> <list> <list> <list>
1 <chr [1]>           <NULL> <NULL> <NULL> <NULL> <NULL> <NULL> <NULL> <NULL> <NULL> <NULL> <NULL> <NULL>
2 <chr [1]>           <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>  <chr>
3 <dbl [1]>           <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
4 <dbl [1]>           <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
5 <dbl [1]>           <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
6 <dbl [1]>           <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
```

Google Sheets

- The first 5 rows of the data look really odd. If we go to the Google sheet link, you will notice that the first two rows are table headers and not data



United States Billion-Dollar Disasters By Year (CPI-Adjusted)

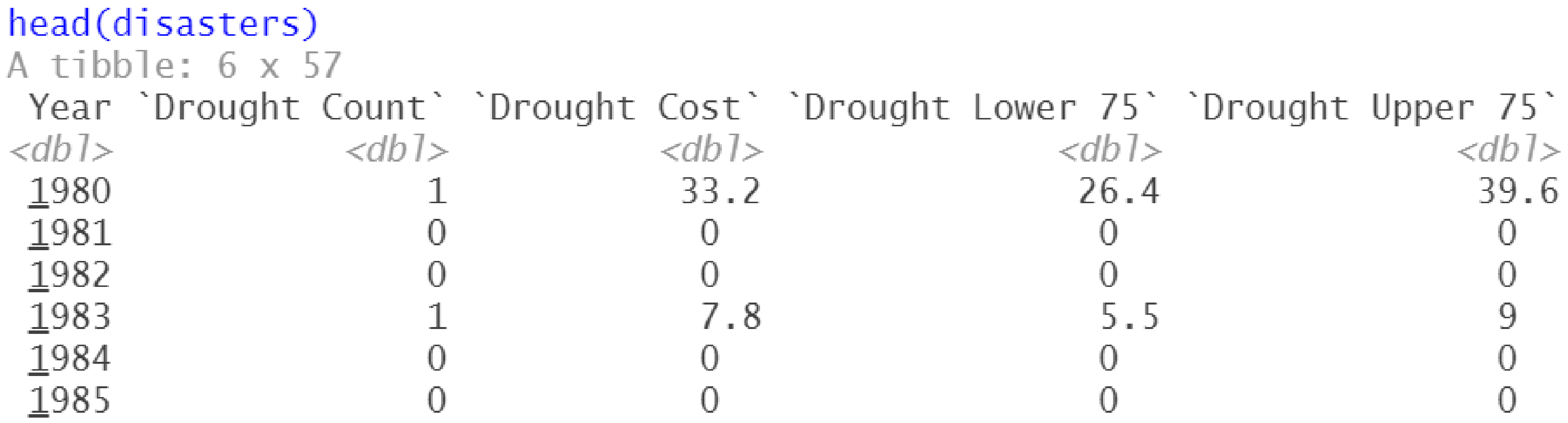
Cost values are in billions of dollars

Year	Drought Count	Drought Cost	Drought Lower 7	Drought Upper 7	Drought Lower 9	Drought Upper 9	Drought Lower 9
1980	1	33.2	26.4	39.6	24.5	41.6	23.4
1981	0	0	0	0	0	0	0
1982	0	0	0	0	0	0	0
1983	1	7.8	5.5	9	5	9.9	4.6
1984	0	0	0	0	0	0	0
1985	0	0	0	0	0	0	0
1986	1	4.2	3.5	5	3.2	5.3	3
1987	0	0	0	0	0	0	0
1988	1	44.4	33.8	54	31.3	56.5	30.2
1989	1	6.4	5.6	7.4	5.3	7.6	5.1

Google Sheets

- To tell R to skip the first two rows, we use the argument **skip = 2**. You can also use this argument in the read_csv() and read_excel() functions.

- disasters <- read_sheet("https://docs.google.com/spreadsheets/d/17s15o7jdDpGSKgsIboZdnYU2UxHtU9DHKNRmYVVgwJo/edit#gid=0", skip = 2)

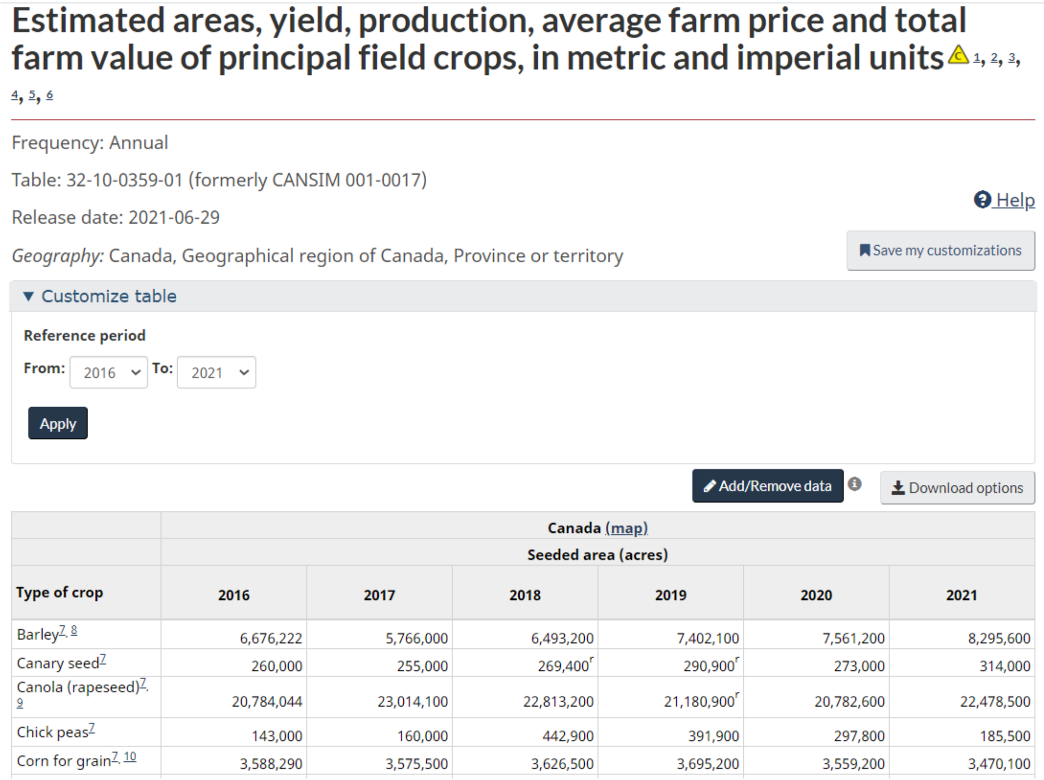


```
head(disasters)
A tibble: 6 x 57
 Year `Drought Count` `Drought Cost` `Drought Lower 75` `Drought Upper 75`
<dbl>           <dbl>          <dbl>              <dbl>              <dbl>
 1980               1           33.2               26.4               39.6
 1981               0            0                  0                  0
 1982               0            0                  0                  0
 1983               1            7.8                5.5                9
 1984               0            0                  0                  0
 1985               0            0                  0                  0
```

UBC
MFRE

Statistics Canada data

- Let’s say you want to work with this table from Statistics Canada.



Estimated areas, yield, production, average farm price and total farm value of principal field crops, in metric and imperial units ⚠ [1, 2, 3, 4, 5, 6]

Frequency: Annual

Table: 32-10-0359-01 (formerly CANSIM 001-0017)

Release date: 2021-06-29

Geography: Canada, Geographical region of Canada, Province or territory

Reference period

From: 2016 **To:** 2021

	Canada (map)					
	Seeded area (acres)					
Type of crop	**2016**	**2017**	**2018**	**2019**	**2020**	**2021**
Barley[7, 8]	6,676,222	5,766,000	6,493,200	7,402,100	7,561,200	8,295,600
Canary seed[7]	260,000	255,000	269,400[r]	290,900[r]	273,000	314,000
Canola (rapeseed)[7, 9]	20,784,044	23,014,100	22,813,200	21,180,900[r]	20,782,600	22,478,500
Chick peas[7]	143,000	160,000	442,900	391,900	297,800	185,500
Corn for grain[7, 10]	3,588,290	3,575,500	3,626,500	3,695,200	3,559,200	3,470,100

UBC MFRE

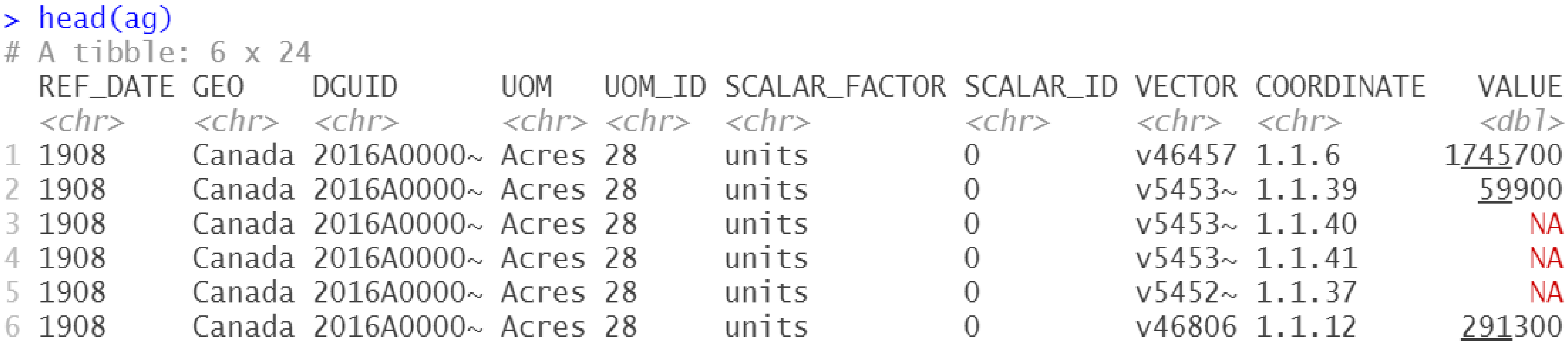
Statistics Canada data

- You can download the Excel file, save it in your computer, and load it using one of the functions discussed earlier.

- Or you can use the {cansim} package which connects straight to the Statistics Canada database using an API (you'll learn more about an API in FRE521D).

Statistics Canada data

- ag <- get_cansim('32-10-0359-01’)



```
> head(ag)
# A tibble: 6 x 24
  REF_DATE GEO    DGUID      UOM   UOM_ID SCALAR_FACTOR SCALAR_ID VECTOR COORDINATE   VALUE
  <chr>    <chr>  <chr>      <chr> <chr>  <chr>         <chr>     <chr>  <chr>        <dbl>
1 1908     Canada 2016A0000~ Acres 28     units         0         v46457 1.1.6      1745700
2 1908     Canada 2016A0000~ Acres 28     units         0         v5453~ 1.1.39       59900
3 1908     Canada 2016A0000~ Acres 28     units         0         v5453~ 1.1.40          NA
4 1908     Canada 2016A0000~ Acres 28     units         0         v5453~ 1.1.41          NA
5 1908     Canada 2016A0000~ Acres 28     units         0         v5452~ 1.1.37          NA
6 1908     Canada 2016A0000~ Acres 28     units         0         v46806 1.1.12      291300
```

UBC
MFRE

What we just did

- Load data in different ways
  - Flat files (csv and txt) – **read_csv()** or **read.csv()**; **read_tsv()**
    - Can use **skip = x** argument
  - Excel file (xlsx) – **read_excel()**
    - Can use **skip = x** argument
    - Add **sheet = number** or **sheet = "sheet_name"** argument to specify which sheet to import
  - Stata data format (dta) – **read.dta13()**
  - Google Sheet – **read_sheet()**
    - gs4_deauth() to not require signing in to Google
  - Statistics Canada data – **get_cansim()**

UBC
MFRE

mfre.landfood.ubc.ca