Data Science for Beginners, University of Essex

Day 4: Working with Data Frame

Dr. Howard Liu

13-01-2022

Learning Objectives Today

1. Import and export data files
2. Explore the data
3. Pipe operator
4. Working with data frames: dplyr functions for data wrangling

1. Import data

To read in (=import) a data set from a file, we use read.XXX functions, such as

- read.csv Comma-separated text file
- read.table Tab-separated text file
- read.dta Stata file
- read.spss SPSS file
- read.xlsx Excel file

To load data, your computer needs to know where the data file is saved. The key is to set the path for R to locate

```
myPath <- "/Users/howardliu/Dropbox/Essex/data-programming-beginners/Lecture/" # THIS IS IMPORTANT!
setwd(myPath)

library(tidyverse) # I need this package for the as_tibble function. I will introduce tidyverse this inclusive package more later on
world.data = read.csv("world.csv") %>% as_tibble() # here I make it a tibble data frame just for illustration purpose
```

1-2 Export data

To export data, use the write.csv() function

```
myPath <- "/Users/howardliu/Dropbox/Essex/data-programming-beginners/Lecture/"
setwd(myPath) # set working directory will let R to use this as your default path

# write.csv(world.data, file = "world_tmp.csv") # You can export your data as a csv fil
e.

# save(world.data, world.data2, file ="world_tmp.rda") # This allows you to save multipl
e data frames together into one file
```

2. Explore your data

Whenever you read in a data set from a file, it's always a good idea to take a look at it first to have an idea about what it looks like. There are a few functions that will let us take a look at the data set.

The dim function tells us the dimension (the number of rows and the number of columns).

```
dim(world.data)
```

```
## [1] 191  62
```

So this contains 191 rows (observations) and 62 columns (variables)

The `head()` and `tail()` functions let us see the first or last 5 rows of the data set.

```
head(world.data)
```

```
## # A tibble: 6 × 62
##   country   colony confidence decentralization dem_other dem_other5 democ_regime
##   <fct>     <fct>       <dbl>            <dbl>     <dbl> <fct>      <fct>
## 1 Afghanis… UK           NA              NA         10.5 10%        No
## 2 Albania   Sovie…       49.3             0.74      63   Approx 60% Yes
## 3 Algeria   France       52.1            NA         40.8 Approx 40% No
## 4 Andorra   Spain        NA              NA        100   100%       Yes
## 5 Angola    Portu…       NA              NA         40.8 Approx 40% No
## 6 Antigua … UK           NA              NA         87.5 Approx 90% Yes
## # … with 55 more variables: district_size3 <fct>, durable <int>,
## #   effectiveness <dbl>, enpp_3 <fct>, eu <fct>, fhrate04_rev <fct>,
## #   fhrate08_rev <int>, frac_eth <dbl>, frac_eth3 <fct>, free_business <dbl>,
## #   free_corrupt <int>, free_finance <int>, free_fiscal <dbl>,
## #   free_govspend <dbl>, free_invest <int>, free_labor <dbl>,
## #   free_monetary <dbl>, free_overall <dbl>, free_property <int>,
## #   free_trade <dbl>, gdp08 <dbl>, gdp_10_thou <dbl>, gdp_cap2 <fct>, …
```

```
tail(world.data)
```

```
## # A tibble: 6 × 62
##   country   colony confidence decentralization dem_other dem_other5 democ_regime
##   <fct>     <fct>       <dbl>            <dbl>     <dbl> <fct>      <fct>
## 1 Vietnam   France       99.9           NA          58.3 Approx 60% No
## 2 Western … Other        NA             NA          58.3 Approx 60% No
## 3 Yemen     UK           NA             NA          10.5 10%        No
## 4 Serbia &… Sovie…       31.6           NA          63   Approx 60% No
## 5 Zambia    UK           NA             NA          40.8 Approx 40% Yes
## 6 Zimbabwe  UK           60.0            0.87       40.8 Approx 40% No
## # … with 55 more variables: district_size3 <fct>, durable <int>,
## #   effectiveness <dbl>, enpp_3 <fct>, eu <fct>, fhrate04_rev <fct>,
## #   fhrate08_rev <int>, frac_eth <dbl>, frac_eth3 <fct>, free_business <dbl>,
## #   free_corrupt <int>, free_finance <int>, free_fiscal <dbl>,
## #   free_govspend <dbl>, free_invest <int>, free_labor <dbl>,
## #   free_monetary <dbl>, free_overall <dbl>, free_property <int>,
## #   free_trade <dbl>, gdp08 <dbl>, gdp_10_thou <dbl>, gdp_cap2 <fct>, …
```

Or we can do this using the square brackets:

```
world.data[1:10, 1:5]
```

```
## # A tibble: 10 × 5
##    country           colony       confidence decentralization dem_other
##    <fct>             <fct>             <dbl>            <dbl>     <dbl>
##  1 Afghanistan       UK                NA               NA         10.5
##  2 Albania           Soviet Union      49.3              0.74      63
##  3 Algeria           France            52.1             NA         40.8
##  4 Andorra           Spain             NA               NA        100
##  5 Angola            Portugal          NA               NA         40.8
##  6 Antigua & Barbuda UK                NA               NA         87.5
##  7 Argentina         Spain              7.30             2.4       87.5
##  8 Armenia           Soviet Union      27.1             NA         63
##  9 Australia         UK                46.8              1.74      58.3
## 10 Austria           Other             49.7              1.81     100
```

This tells R to show the first 10 rows and first 5 columns. We can easily see that country is the unit of observation. We can also see that there are variables such as colony, confidence, decentralization, and dem_other.

When working with a large data set like this one, you may want to use the spreadsheet view (just like Excel). To do so, we use the `View()` function, which opens a spreadsheet tab in the script pane.

```
# View(world.data)
```

If you want to take a further look at a specific variable, you can use the `$` to call it and the `summary()` function.

```
world.data$colony
```

```
##   [1] UK           Soviet Union France       Spain        Portugal
##   [6] UK           Spain        Soviet Union UK           Other
##  [11] Soviet Union UK           UK           Other        UK
##  [16] Soviet Union Netherlands  UK           France       Other
##  [21] Spain        Soviet Union UK           Portugal     UK
##  [26] Soviet Union France       Belgium      France       France
##  [31] UK           Portugal     France       France       Spain
##  [36] none         France       France       France       Belgium
##  [41] Spain        France       Soviet Union Spain        UK
##  [46] Soviet Union none         France       UK           Other
##  [51] Spain        UK           Spain        Spain        Other
##  [56] Soviet Union none         UK           none         France
##  [61] France       UK           Soviet Union none         UK
##  [66] Ottoman      UK           Spain        Spain        Portugal
##  [71] UK           France       Spain        Soviet Union none
##  [76] UK           Netherlands  none         UK           UK
##  [81] UK           none         UK           none         UK
##  [86] Soviet Union UK           UK           Other        Other
##  [91] UK           Soviet Union France       Soviet Union France
##  [96] UK           UK           Other        none         Soviet Union
## [101] Netherlands  Soviet Union France       UK           UK
## [106] UK           France       UK           none         France
## [111] UK           Spain        none         Soviet Union none
## [116] Other        France       Portugal     UK           Other
## [121] UK           none         Spain        UK           Spain
## [126] France       UK           none         Portugal     UK
## [131] none         Spain        Other        Spain        Spain
## [136] Spain        Soviet Union Portugal     UK           Soviet Union
## [141] Soviet Union Belgium      none         Portugal     UK
## [146] France       UK           UK           Other        Soviet Union
## [151] Soviet Union UK           UK           UK           Spain
## [156] UK           UK           UK           UK           UK
## [161] Netherlands  UK           none         none         France
## [166] Other        Soviet Union UK           none         France
## [171] UK           UK           France       Ottoman      Soviet Union
## [176] UK           UK           UK           Soviet Union UK
## [181] UK           Other        Soviet Union France       Spain
## [186] France       Other        UK           Soviet Union UK
## [191] UK
## 10 Levels: Belgium France Netherlands none Other Ottoman ... UK
```

```
summary(world.data$colony)
```

```
##      Belgium       France  Netherlands         none        Other      Ottoman
##            3           28            4           20           15            2
##     Portugal Soviet Union        Spain           UK
##            8           27           21           63
```

Or you can simply summarize the entire data frame, which in some cases might be useful.

```
summary(world.data)
```

```
##              country             colony     confidence      decentralization
##  Afghanistan      :  1   UK          :63   Min.   : 0.5167   Min.   :0.380
##  Albania          :  1   France      :28   1st Qu.:38.3669   1st Qu.:1.225
##  Algeria          :  1   Soviet Union:27   Median :49.1978   Median :1.510
##  Andorra          :  1   Spain       :21   Mean   :47.9704   Mean   :1.516
##  Angola           :  1   none        :20   3rd Qu.:59.2929   3rd Qu.:1.800
##  Antigua & Barbuda:  1   Other       :15   Max.   :99.8624   Max.   :2.450
##  (Other)          :185   (Other)     :17   NA's   :120       NA's   :124
##    dem_other           dem_other5 democ_regime           district_size3
##  Min.   : 10.50   10%       :19   No  : 75                     :44
##  1st Qu.: 40.80   100%      :27   Yes :114     >1 to 5 members  :28
##  Median : 58.30   Approx 40%:49   NA's:  2     6 or more members:37
##  Mean   : 60.51   Approx 60%:64                single member    :82
##  3rd Qu.: 87.50   Approx 90%:32
##  Max.   :100.00
##
##     durable       effectiveness             enpp_3                 eu
##  Min.   :  0.00   Min.   :  0.00              :93   EU Member state: 25
##  1st Qu.:  4.00   1st Qu.: 28.19   1-3 parties :43   Not member     :166
##  Median :  9.00   Median : 40.31   4-5 parties :24
##  Mean   : 22.49   Mean   : 45.77   6-11 parties:31
##  3rd Qu.: 31.25   3rd Qu.: 62.77
##  Max.   :191.00   Max.   :100.00
##  NA's   :31       NA's   :5
##     fhrate04_rev  fhrate08_rev       frac_eth       frac_eth3  free_business
##  Most free:46    Min.   : 0.000   Min.   :0.0000         : 3   Min.   :10.00
##  2.5      :24    1st Qu.: 4.000   1st Qu.:0.1997   High  :62   1st Qu.:55.70
##  5        :16    Median : 8.000   Median :0.4343   Low   :62   Median :65.80
##  6        :16    Mean   : 7.553   Mean   :0.4394   Medium:64   Mean   :64.92
##  6.5      :15    3rd Qu.:11.250   3rd Qu.:0.6611               3rd Qu.:76.60
##  5.5      :12    Max.   :12.000   Max.   :0.9302               Max.   :99.90
##  (Other)  :62    NA's   :3        NA's   :3                    NA's   :18
##   free_corrupt    free_finance    free_fiscal    free_govspend
##  Min.   : 5.00   Min.   :10.00   Min.   :35.90   Min.   : 6.90
##  1st Qu.:26.00   1st Qu.:30.00   1st Qu.:68.20   1st Qu.:54.95
##  Median :34.00   Median :50.00   Median :77.50   Median :73.40
##  Mean   :40.42   Mean   :48.61   Mean   :75.62   Mean   :67.59
##  3rd Qu.:51.75   3rd Qu.:60.00   3rd Qu.:84.00   3rd Qu.:83.25
##  Max.   :93.00   Max.   :90.00   Max.   :99.90   Max.   :98.40
##  NA's   :17      NA's   :18      NA's   :18      NA's   :24
##   free_invest      free_labor    free_monetary    free_overall   free_property
##  Min.   : 5.00   Min.   :20.00   Min.   :46.50   Min.   : 1.00   Min.   : 5.0
##  1st Qu.:35.00   1st Qu.:50.10   1st Qu.:66.85   1st Qu.:51.35   1st Qu.:30.0
##  Median :50.00   Median :60.80   Median :71.90   Median :59.30   Median :40.0
##  Mean   :50.75   Mean   :62.08   Mean   :71.30   Mean   :59.18   Mean   :43.9
##  3rd Qu.:70.00   3rd Qu.:75.90   3rd Qu.:76.55   3rd Qu.:67.30   3rd Qu.:60.0
##  Max.   :95.00   Max.   :98.90   Max.   :88.80   Max.   :86.10   Max.   :95.0
##  NA's   :24      NA's   :18      NA's   :19      NA's   :17      NA's   :18
##    free_trade        gdp08          gdp_10_thou     gdp_cap2    gdp_cap3
##  Min.   :31.90   Min.   :    0.2   Min.   :0.0090       :14         :14
##  1st Qu.:67.20   1st Qu.:   11.9   1st Qu.:0.0503   High:88   High  :59
##  Median :75.90   Median :   41.7   Median :0.1897   Low :89   Low   :59
```

```
##  Mean   :74.37   Mean   :  390.4   Mean   :0.6018            Middle:59
##  3rd Qu.:85.00   3rd Qu.:  242.4   3rd Qu.:0.6320
##  Max.   :90.00   Max.   :14200.0   Max.   :4.7354
##  NA's   :18      NA's   :14        NA's   :14
##    gdppcap08      gender_equal3     gini04          gini08           hi_gdp
##  Min.   :   188         :113    Min.   :24.40   Min.   :24.70          :14
##  1st Qu.:  2308   High  : 26    1st Qu.:32.42   1st Qu.:33.55   High GDP:88
##  Median :  7703   Low   : 26    Median :37.95   Median :39.20   Low GDP :89
##  Mean   : 13828   Medium: 26    Mean   :40.14   Mean   :40.74
##  3rd Qu.: 19996                 3rd Qu.:46.88   3rd Qu.:47.10
##  Max.   :118040                 Max.   :70.70   Max.   :74.30
##  NA's   :16                     NA's   :65      NA's   :64
##       indy                     oecd        old2006          old2003
##  Min.   : 301   Not member       :161   Min.   : 1.076   Min.   : 1.846
##  1st Qu.:1915   OECD Member state: 30   1st Qu.: 3.375   1st Qu.: 3.173
##  Median :1960                           Median : 4.924   Median : 4.865
##  Mean   :1891                           Mean   : 7.300   Mean   : 6.979
##  3rd Qu.:1977                           3rd Qu.:11.210   3rd Qu.:10.656
##  Max.   :1994                           Max.   :20.232   Max.   :18.997
##  NA's   :3                              NA's   :17       NA's   :10
##               pmat12_3       pop03               pop08
##                   :127   Min.   :2.000e+04   Min.   :   0.00
##  High post-mat    : 21   1st Qu.:1.758e+06   1st Qu.:   2.70
##  Low post-mat     : 21   Median :6.720e+06   Median :   8.30
##  Moderate post-mat: 22   Mean   :3.318e+07   Mean   :  36.95
##                          3rd Qu.:2.121e+07   3rd Qu.:  24.60
##                          Max.   :1.288e+09   Max.   :1300.00
##                          NA's   :4           NA's   :14
##          pop08_3               popcat3    pr_sys        protact3
##              :14   0               :  1   No :124           :126
##  <=4.3 mil   :59   Large (30m+)    : 33   Yes: 67   High    : 21
##  >=16.8 mil  :59   Moderate (1-29m):116             Low     : 21
##  4.4-16.4 mil:59   Small (under 1m): 41             Moderate: 23
##
##
##
##               regime_type3          region   sources                  typerel
##                     :23    Africa      :49   Mode:logical   Roman Catholic:63
##  Dictatorship       :75    Asia-Pacific:37   NA's:191       Muslim        :50
##  Parliamentary democ:56    S. America  :32                  Protestant    :35
##  Presidential democ :37    C&E Europe  :27                  eastern       :15
##                            Middle East :19                  Orthodox      :13
##                            W. Europe   :19                  other         :12
##                            (Other)     : 8                  (Other)       : 3
##      unions         urban03           urban06         vi_rel3
##  Min.   : 2.00   Min.   :  6.556   Min.   : 10.32         :115
##  1st Qu.:11.45   1st Qu.: 36.413   1st Qu.: 35.49   <20%  : 25
##  Median :19.10   Median : 57.491   Median : 56.76   >50%  : 25
##  Mean   :24.74   Mean   : 55.620   Mean   : 54.55   20-50%: 26
##  3rd Qu.:30.80   3rd Qu.: 73.830   3rd Qu.: 72.75
##  Max.   :96.10   Max.   :100.000   Max.   :100.00
##  NA's   :100     NA's   :5         NA's   :4
```

```
##    votevap00s       women05         women09         womyear   
##  Min.   :18.29   Min.   : 0.00   Min.   : 0.00   Min.   :1893  
##  1st Qu.:54.58   1st Qu.: 8.25   1st Qu.: 9.70   1st Qu.:1931  
##  Median :65.12   Median :13.00   Median :15.55   Median :1949  
##  Mean   :65.08   Mean   :15.38   Mean   :17.18   Mean   :1947  
##  3rd Qu.:77.66   3rd Qu.:20.45   3rd Qu.:22.95   3rd Qu.:1960  
##  Max.   :98.39   Max.   :45.30   Max.   :56.30   Max.   :1990  
##  NA's   :92      NA's   :80      NA's   :11      NA's   :16    
##            womyear2      yng2003         young06        
##                : 16   Min.   :14.02   Min.   :13.50  
##  1944 or before: 60   1st Qu.:21.31   1st Qu.:19.53  
##  After 1944    :115   Median :31.95   Median :30.65  
##                       Mean   :31.41   Mean   :30.45  
##                       3rd Qu.:41.30   3rd Qu.:39.72  
##                       Max.   :49.77   Max.   :50.50  
##                       NA's   :10      NA's   :17
```

3. Understanding pipe operator %>% in R

Before introducing the dplyr package (a fast, powerful tool for working with data frame), we need to know the pipe operator first.

```
# The pipe operator makes it possible to easily chain a sequence of calculations.

# First, install it from this package (You don't have to if you have installed the tidyv
erse package)
library(magrittr)

# Suppose we create an object and assign values. If we want to do multiple calculations,
we need to write a lot of parentheses, which can be very annoying.

# For example:
x <- c(0.109, 0.359, 0.63, 0.996, 0.515, 0.142, 0.017, 0.829, 0.907)

# Like compute the logarithm of `x`, return suitably lagged and iterated differences,
# compute the exponential function and round the result
round(exp(diff(log(x))), 1)
```

```
## [1]  3.3  1.8  1.6  0.5  0.3  0.1 48.8  1.1
```

```
# Same task can be completed by using pipes. Everything looks much clearer and easy.
x %>% log %>% diff %>% exp %>% round(., 1)
```

```
## [1]  3.3  1.8  1.6  0.5  0.3  0.1 48.8  1.1
```

The hot key for %>% is `Command + shift + m` (on MAC)

Now, let's proceed on to discuss dplyr functionality then.

4. dplyr in R

It is a fast, powerful tool for working with data frame. You don't need to worry about parentheses when writing commands anymore. It is magical!

dplyr is based on the concepts of functions as "verbs" that manipulate data frames. It's also powerful when you manage large datasets,

Single data frame functions / verbs:

- `filter()` : pick rows matching criteria
- `slice()` : pick rows using index(es)
- `select()` : pick columns by name
- `pull()` : grab a column as a vector
- `rename()` : rename specific columns
- `arrange()` : reorder rows
- `mutate()` : add new variables
- `transmute()` : create new data frame with variables
- `distinct()` : filter for unique rows
- `sample_n() / sample_frac()` : randomly sample rows
- `summarise()` : reduce variables to values
- … (many more)

dplyr rules for functions

1. First argument is *always* a data frame
2. Subsequent arguments say what to do with that data frame
3. *Always* return a data frame
4. Don't modify in place
5. Lazy evaluation magic

Let's load the dplyr package first.

You can either use the dplyr package directly. Or you can use the tidyverse package that includes other important packages that you're likely to use in everyday data analyses (i.e., ggplot2, dplyr, readr, tibble, stringr etc.)

```
library(dplyr)
library(tidyverse) # or this
```

Example Data

We will demonstrate dplyr's functionality using the `world.data` again.

`filter()` - Show only countries in the EU

```
world.data %>% filter(eu =="EU Member state")
```

```
## # A tibble: 25 × 62
##    country  colony confidence decentralization dem_other dem_other5 democ_regime
##    <fct>    <fct>       <dbl>            <dbl>     <dbl> <fct>      <fct>
##  1 Austria  Other        49.7             1.81       100 100%       Yes
##  2 Belgium  Nethe…       43.3             1.38       100 100%       Yes
##  3 Cyprus   UK           NA              NA          100 100%       Yes
##  4 Czech R… Sovie…       35.2             1.44        63 Approx 60% Yes
##  5 Denmark  none         63.1             2.11       100 100%       Yes
##  6 Estonia  Sovie…       39.2             1.47        63 Approx 60% Yes
##  7 Finland  none         46.9             1.52       100 100%       Yes
##  8 France   France       55.1             1.72       100 100%       Yes
##  9 Germany  none         52.7             2.18       100 100%       Yes
## 10 Greece   Ottom…       NA              NA          100 100%       Yes
## # … with 15 more rows, and 55 more variables: district_size3 <fct>,
## #   durable <int>, effectiveness <dbl>, enpp_3 <fct>, eu <fct>,
## #   fhrate04_rev <fct>, fhrate08_rev <int>, frac_eth <dbl>, frac_eth3 <fct>,
## #   free_business <dbl>, free_corrupt <int>, free_finance <int>,
## #   free_fiscal <dbl>, free_govspend <dbl>, free_invest <int>,
## #   free_labor <dbl>, free_monetary <dbl>, free_overall <dbl>,
## #   free_property <int>, free_trade <dbl>, gdp08 <dbl>, gdp_10_thou <dbl>, …
```

`filter()` - Show countries in the EU and GDP per capita is high. Plus I only want to show the country name and not anything else.

```
world.data %>% filter(eu =="EU Member state" & gdp_cap2 == "High") %>% select(country)
```

```
## # A tibble: 25 × 1
##    country
##    <fct>
##  1 Austria
##  2 Belgium
##  3 Cyprus
##  4 Czech Republic
##  5 Denmark
##  6 Estonia
##  7 Finland
##  8 France
##  9 Germany
## 10 Greece
## # … with 15 more rows
```

`slice()` - Pick the first 10 countries

```
world.data %>% slice(1:10)
```

```
## # A tibble: 10 × 62
##    country  colony confidence decentralization dem_other dem_other5 democ_regime
##    <fct>    <fct>       <dbl>            <dbl>     <dbl> <fct>      <fct>
##  1 Afghani… UK          NA               NA         10.5 10%        No
##  2 Albania  Sovie…      49.3              0.74      63   Approx 60% Yes
##  3 Algeria  France      52.1             NA         40.8 Approx 40% No
##  4 Andorra  Spain       NA               NA        100   100%       Yes
##  5 Angola   Portu…      NA               NA         40.8 Approx 40% No
##  6 Antigua… UK          NA               NA         87.5 Approx 90% Yes
##  7 Argenti… Spain        7.30             2.4       87.5 Approx 90% Yes
##  8 Armenia  Sovie…      27.1             NA         63   Approx 60% Yes
##  9 Austral… UK          46.8              1.74      58.3 Approx 60% Yes
## 10 Austria  Other       49.7              1.81     100   100%       Yes
## # … with 55 more variables: district_size3 <fct>, durable <int>,
## #   effectiveness <dbl>, enpp_3 <fct>, eu <fct>, fhrate04_rev <fct>,
## #   fhrate08_rev <int>, frac_eth <dbl>, frac_eth3 <fct>, free_business <dbl>,
## #   free_corrupt <int>, free_finance <int>, free_fiscal <dbl>,
## #   free_govspend <dbl>, free_invest <int>, free_labor <dbl>,
## #   free_monetary <dbl>, free_overall <dbl>, free_property <int>,
## #   free_trade <dbl>, gdp08 <dbl>, gdp_10_thou <dbl>, gdp_cap2 <fct>, …
```

`slice()` - Last 5 countries and show country names only

```
world.data %>% slice( (n()-4) : n() ) %>% select(country)
```

```
## # A tibble: 5 × 1
##   country
##   <fct>
## 1 Western Samoa
## 2 Yemen
## 3 Serbia & Montenegro
## 4 Zambia
## 5 Zimbabwe
```

`select()` - Pick Individual Columns

```
world.data %>% select(country, eu, gdp_cap2)
```

```
## # A tibble: 191 × 3
##    country           eu              gdp_cap2
##    <fct>             <fct>           <fct>
##  1 Afghanistan       Not member      ""
##  2 Albania           Not member      "Low"
##  3 Algeria           Not member      "Low"
##  4 Andorra           Not member      ""
##  5 Angola            Not member      "Low"
##  6 Antigua & Barbuda Not member      "High"
##  7 Argentina         Not member      "High"
##  8 Armenia           Not member      "Low"
##  9 Australia         Not member      "High"
## 10 Austria           EU Member state "High"
## # … with 181 more rows
```

`select()` - Exclude Columns

```
world.data %>% select(-country, -eu, -gdp_cap2)
```

```
## # A tibble: 191 × 59
##    colony       confidence decentralization dem_other dem_other5 democ_regime
##    <fct>             <dbl>            <dbl>     <dbl> <fct>      <fct>
##  1 UK                NA               NA         10.5 10%        No
##  2 Soviet Union      49.3              0.74      63   Approx 60% Yes
##  3 France            52.1             NA         40.8 Approx 40% No
##  4 Spain             NA               NA        100   100%       Yes
##  5 Portugal          NA               NA         40.8 Approx 40% No
##  6 UK                NA               NA         87.5 Approx 90% Yes
##  7 Spain              7.30             2.4       87.5 Approx 90% Yes
##  8 Soviet Union      27.1             NA         63   Approx 60% Yes
##  9 UK                46.8              1.74      58.3 Approx 60% Yes
## 10 Other             49.7              1.81     100   100%       Yes
## # … with 181 more rows, and 53 more variables: district_size3 <fct>,
## #   durable <int>, effectiveness <dbl>, enpp_3 <fct>, fhrate04_rev <fct>,
## #   fhrate08_rev <int>, frac_eth <dbl>, frac_eth3 <fct>, free_business <dbl>,
## #   free_corrupt <int>, free_finance <int>, free_fiscal <dbl>,
## #   free_govspend <dbl>, free_invest <int>, free_labor <dbl>,
## #   free_monetary <dbl>, free_overall <dbl>, free_property <int>,
## #   free_trade <dbl>, gdp08 <dbl>, gdp_10_thou <dbl>, gdp_cap3 <fct>, …
```

`select()` - Matching: Pick variables that contain these characters

```
world.data %>% select(contains("gdp"),
                      contains("dem"))
```

```
## # A tibble: 191 × 9
##    gdp08 gdp_10_thou gdp_cap2 gdp_cap3 gdppcap08 hi_gdp     dem_other dem_other5
##    <dbl>       <dbl> <fct>    <fct>        <int> <fct>          <dbl> <fct>
##  1  30.6     NA      ""       ""              NA ""              10.5 10%
##  2  24.3      0.154  "Low"    "Middle"      7715 "Low GDP"       63   Approx 60%
##  3 276        0.178  "Low"    "Middle"      8033 "Low GDP"       40.8 Approx 40%
##  4  NA       NA      ""       ""              NA ""             100   100%
##  5 106.       0.0857 "Low"    "Middle"      5899 "Low GDP"       40.8 Approx 40%
##  6  NA        1.04   "High"   "High"          NA "High GDP"      87.5 Approx 90%
##  7 572.       0.280  "High"   "Middle"     14333 "High GDP"      87.5 Approx 90%
##  8  18.7      0.0771 "Low"    "Low"         6070 "Low GDP"       63   Approx 60%
##  9 763.       2.08   "High"   "High"       35677 "High GDP"      58.3 Approx 60%
## 10 318.       2.54   "High"   "High"       38152 "High GDP"     100   100%
## # … with 181 more rows, and 1 more variable: democ_regime <fct>
```

`mutate()` - Modify columns and create a new variable

```
world.data %>% select(1:2) %>% mutate(newVar = paste(country, colony, sep="/") )
```

```
## # A tibble: 191 × 3
##    country           colony       newVar
##    <fct>             <fct>        <chr>
##  1 Afghanistan       UK           Afghanistan/UK
##  2 Albania           Soviet Union Albania/Soviet Union
##  3 Algeria           France       Algeria/France
##  4 Andorra           Spain        Andorra/Spain
##  5 Angola            Portugal     Angola/Portugal
##  6 Antigua & Barbuda UK           Antigua & Barbuda/UK
##  7 Argentina         Spain        Argentina/Spain
##  8 Armenia           Soviet Union Armenia/Soviet Union
##  9 Australia         UK           Australia/UK
## 10 Austria           Other        Austria/Other
## # … with 181 more rows
```

`rename()` - Change column names

```
world.data %>% rename(gdp_08 = gdp08) %>% select(gdp_08)
```

```
## # A tibble: 191 × 1
##    gdp_08
##     <dbl>
##  1   30.6
##  2   24.3
##  3  276
##  4   NA
##  5  106.
##  6   NA
##  7  572.
##  8   18.7
##  9  763.
## 10  318.
## # … with 181 more rows
```

summarise() with group_by()

It will do calculation for the group that you want to create.

```
world.data %>% group_by(eu) %>%
  summarise(meanGDP = mean(gdp08, na.rm = TRUE)) # Here I calculate the mean. You could replace by using `sum()` for example.
```

```
## # A tibble: 2 × 2
##   eu              meanGDP
##   <fct>             <dbl>
## 1 EU Member state    595.
## 2 Not member         357.
```

left_join() - Merge two datasets

```
# first I create two datasets
dat1 = world.data %>% select(1:3)
dat2 = world.data %>% select(1:2, 4)

dat_merged = left_join(dat1, dat2, by = c("country" = "country", "colony" = "colony")) # merge them by picking two variables that are used as mergers
dat_merged
```

```
## # A tibble: 191 × 4
##    country           colony       confidence decentralization
##    <fct>             <fct>             <dbl>            <dbl>
##  1 Afghanistan       UK                NA               NA
##  2 Albania           Soviet Union      49.3              0.74
##  3 Algeria           France            52.1             NA
##  4 Andorra           Spain             NA               NA
##  5 Angola            Portugal          NA               NA
##  6 Antigua & Barbuda UK                NA               NA
##  7 Argentina         Spain              7.30             2.4
##  8 Armenia           Soviet Union      27.1             NA
##  9 Australia         UK                46.8              1.74
## 10 Austria           Other             49.7              1.81
## # … with 181 more rows
```

Great! We've finished the lecture and you can go to day4 exercise to do some additional practices for today's content.