Lecture_4_Notes_Data_Wrangling

Athena Zhang

2023-09-28

Data Wrangling

0. Load the tidyverse package

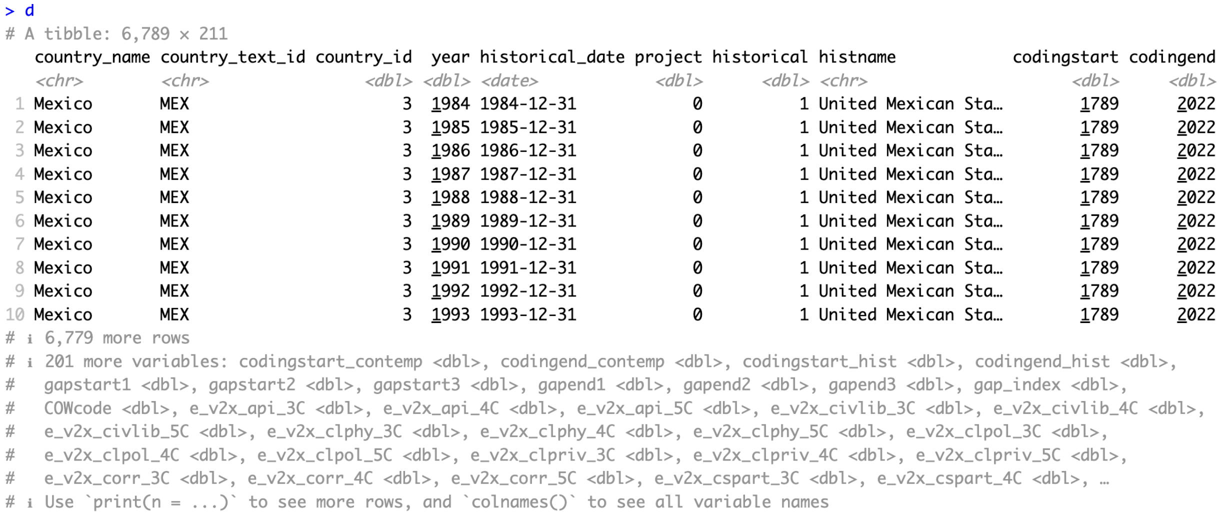
```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.3     v readr     2.1.4
## v forcats   1.0.0     v stringr   1.5.0
## v ggplot2   3.4.3     v tibble    3.2.1
## v lubridate 1.9.2     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

1. Import the V-Dem Data

```
d <- read_csv("_DataPublic_/vdem/1984_2022/vdem_1984_2022_external.csv")
```

```
## Rows: 6789 Columns: 211
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (3): country_name, country_text_id, histname
## dbl  (207): country_id, year, project, historical, codingstart, codingend, c...
## date   (1): historical_date
##
## 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.
```



```
> d
# A tibble: 6,789 × 211
   country_name country_text_id country_id  year historical_date project historical histname            codingstart codingend
   <chr>        <chr>                <dbl> <dbl> <date>              <dbl>      <dbl> <chr>                     <dbl>     <dbl>
 1 Mexico       MEX                      3  1984 1984-12-31              0          1 United Mexican Sta…        1789      2022
 2 Mexico       MEX                      3  1985 1985-12-31              0          1 United Mexican Sta…        1789      2022
 3 Mexico       MEX                      3  1986 1986-12-31              0          1 United Mexican Sta…        1789      2022
 4 Mexico       MEX                      3  1987 1987-12-31              0          1 United Mexican Sta…        1789      2022
 5 Mexico       MEX                      3  1988 1988-12-31              0          1 United Mexican Sta…        1789      2022
 6 Mexico       MEX                      3  1989 1989-12-31              0          1 United Mexican Sta…        1789      2022
 7 Mexico       MEX                      3  1990 1990-12-31              0          1 United Mexican Sta…        1789      2022
 8 Mexico       MEX                      3  1991 1991-12-31              0          1 United Mexican Sta…        1789      2022
 9 Mexico       MEX                      3  1992 1992-12-31              0          1 United Mexican Sta…        1789      2022
10 Mexico       MEX                      3  1993 1993-12-31              0          1 United Mexican Sta…        1789      2022
# ℹ 6,779 more rows
# ℹ 201 more variables: codingstart_contemp <dbl>, codingend_contemp <dbl>, codingstart_hist <dbl>, codingend_hist <dbl>,
#   gapstart1 <dbl>, gapstart2 <dbl>, gapstart3 <dbl>, gapend1 <dbl>, gapend2 <dbl>, gapend3 <dbl>, gap_index <dbl>,
#   COWcode <dbl>, e_v2x_api_3C <dbl>, e_v2x_api_4C <dbl>, e_v2x_api_5C <dbl>, e_v2x_civlib_3C <dbl>, e_v2x_civlib_4C <dbl>,
#   e_v2x_civlib_5C <dbl>, e_v2x_clphy_3C <dbl>, e_v2x_clphy_4C <dbl>, e_v2x_clphy_5C <dbl>, e_v2x_clpol_3C <dbl>,
#   e_v2x_clpol_4C <dbl>, e_v2x_clpol_5C <dbl>, e_v2x_clpriv_3C <dbl>, e_v2x_clpriv_4C <dbl>, e_v2x_clpriv_5C <dbl>,
#   e_v2x_corr_3C <dbl>, e_v2x_corr_4C <dbl>, e_v2x_corr_5C <dbl>, e_v2x_cspart_3C <dbl>, e_v2x_cspart_4C <dbl>, …
# ℹ Use `print(n = ...)` to see more rows, and `colnames()` to see all variable names
```

2. Select interested columns (to operate with)

1. Look at the identifiers of the data (names)

```
names(d)
```

```
##   [1] "country_name"                "country_text_id"
##   [3] "country_id"                  "year"
##   [5] "historical_date"             "project"
##   [7] "historical"                  "histname"
##   [9] "codingstart"                 "codingend"
##  [11] "codingstart_contemp"         "codingend_contemp"
##  [13] "codingstart_hist"            "codingend_hist"
##  [15] "gapstart1"                   "gapstart2"
##  [17] "gapstart3"                   "gapend1"
##  [19] "gapend2"                     "gapend3"
##  [21] "gap_index"                   "COWcode"
##  [23] "e_v2x_api_3C"                "e_v2x_api_4C"
##  [25] "e_v2x_api_5C"                "e_v2x_civlib_3C"
##  [27] "e_v2x_civlib_4C"             "e_v2x_civlib_5C"
##  [29] "e_v2x_clphy_3C"              "e_v2x_clphy_4C"
##  [31] "e_v2x_clphy_5C"              "e_v2x_clpol_3C"
##  [33] "e_v2x_clpol_4C"              "e_v2x_clpol_5C"
##  [35] "e_v2x_clpriv_3C"             "e_v2x_clpriv_4C"
##  [37] "e_v2x_clpriv_5C"             "e_v2x_corr_3C"
##  [39] "e_v2x_corr_4C"               "e_v2x_corr_5C"
##  [41] "e_v2x_cspart_3C"             "e_v2x_cspart_4C"
##  [43] "e_v2x_cspart_5C"             "e_v2x_delibdem_3C"
##  [45] "e_v2x_delibdem_4C"           "e_v2x_delibdem_5C"
##  [47] "e_v2x_EDcomp_thick_3C"       "e_v2x_EDcomp_thick_4C"
##  [49] "e_v2x_EDcomp_thick_5C"       "e_v2x_egal_3C"
##  [51] "e_v2x_egal_4C"               "e_v2x_egal_5C"
##  [53] "e_v2x_egaldem_3C"            "e_v2x_egaldem_4C"
##  [55] "e_v2x_egaldem_5C"            "e_v2x_elecoff_3C"
##  [57] "e_v2x_elecoff_4C"            "e_v2x_elecoff_5C"
##  [59] "e_v2x_execorr_3C"            "e_v2x_execorr_4C"
##  [61] "e_v2x_execorr_5C"            "e_v2x_feduni_3C"
##  [63] "e_v2x_feduni_4C"             "e_v2x_feduni_5C"
##  [65] "e_v2x_frassoc_thick_3C"      "e_v2x_frassoc_thick_4C"
##  [67] "e_v2x_frassoc_thick_5C"      "e_v2x_freexp_3C"
##  [69] "e_v2x_freexp_4C"             "e_v2x_freexp_5C"
##  [71] "e_v2x_freexp_altinf_3C"      "e_v2x_freexp_altinf_4C"
##  [73] "e_v2x_freexp_altinf_5C"      "e_v2x_gencl_3C"
##  [75] "e_v2x_gencl_4C"              "e_v2x_gencl_5C"
##  [77] "e_v2x_gencs_3C"              "e_v2x_gencs_4C"
##  [79] "e_v2x_gencs_5C"              "e_v2x_gender_3C"
##  [81] "e_v2x_gender_4C"             "e_v2x_gender_5C"
##  [83] "e_v2x_genpp_3C"              "e_v2x_genpp_4C"
##  [85] "e_v2x_genpp_5C"              "e_v2x_jucon_3C"
##  [87] "e_v2x_jucon_4C"              "e_v2x_jucon_5C"
##  [89] "e_v2x_libdem_3C"             "e_v2x_libdem_4C"
##  [91] "e_v2x_libdem_5C"             "e_v2x_liberal_3C"
##  [93] "e_v2x_liberal_4C"            "e_v2x_liberal_5C"
##  [95] "e_v2x_mpi_3C"                "e_v2x_mpi_4C"
##  [97] "e_v2x_mpi_5C"                "e_v2x_partip_3C"
##  [99] "e_v2x_partip_4C"             "e_v2x_partip_5C"
## [101] "e_v2x_partipdem_3C"          "e_v2x_partipdem_4C"
## [103] "e_v2x_partipdem_5C"          "e_v2x_polyarchy_3C"
## [105] "e_v2x_polyarchy_4C"          "e_v2x_polyarchy_5C"
## [107] "e_v2x_pubcorr_3C"            "e_v2x_pubcorr_4C"
## [109] "e_v2x_pubcorr_5C"            "e_v2x_suffr_3C"
## [111] "e_v2x_suffr_4C"              "e_v2x_suffr_5C"
## [113] "e_v2xcl_rol_3C"              "e_v2xcl_rol_4C"
## [115] "e_v2xcl_rol_5C"              "e_v2xcs_ccsi_3C"
## [117] "e_v2xcs_ccsi_4C"             "e_v2xcs_ccsi_5C"
## [119] "e_v2xdd_dd_3C"               "e_v2xdd_dd_4C"
## [121] "e_v2xdd_dd_5C"               "e_v2xdl_delib_3C"
## [123] "e_v2xdl_delib_4C"            "e_v2xdl_delib_5C"
## [125] "e_v2xeg_eqdr_3C"             "e_v2xeg_eqdr_4C"
## [127] "e_v2xeg_eqdr_5C"             "e_v2xeg_eqprotec_3C"
## [129] "e_v2xeg_eqprotec_4C"         "e_v2xeg_eqprotec_5C"
## [131] "e_v2xel_frefair_3C"          "e_v2xel_frefair_4C"
## [133] "e_v2xel_frefair_5C"          "e_v2xel_locelec_3C"
## [135] "e_v2xel_locelec_4C"          "e_v2xel_locelec_5C"
## [137] "e_v2xel_regelec_3C"          "e_v2xel_regelec_4C"
## [139] "e_v2xel_regelec_5C"          "e_v2xlg_legcon_3C"
## [141] "e_v2xlg_legcon_4C"           "e_v2xlg_legcon_5C"
## [143] "e_v2xme_altinf_3C"           "e_v2xme_altinf_4C"
## [145] "e_v2xme_altinf_5C"           "e_v2xps_party_3C"
## [147] "e_v2xps_party_4C"            "e_v2xps_party_5C"
## [149] "e_boix_regime"               "e_democracy_breakdowns"
## [151] "e_democracy_omitteddata"     "e_democracy_trans"
## [153] "e_fh_cl"                     "e_fh_pr"
## [155] "e_fh_rol"                    "e_fh_status"
## [157] "e_wbgi_cce"                  "e_wbgi_gee"
## [159] "e_wbgi_pve"                  "e_wbgi_rle"
## [161] "e_wbgi_rqe"                  "e_wbgi_vae"
## [163] "e_lexical_index"             "e_uds_median"
## [165] "e_uds_mean"                  "e_uds_pct025"
## [167] "e_uds_pct975"                "e_coups"
## [169] "e_legparty"                    "e_autoc"
## [171] "e_democ"                       "e_p_polity"
## [173] "e_polcomp"                     "e_polity2"
## [175] "e_bnr_dem"                     "e_chga_demo"
## [177] "e_ti_cpi"                      "e_vanhanen"
## [179] "e_peaveduc"                    "e_peedgini"
## [181] "e_area"                        "e_regiongeo"
## [183] "e_regionpol"                   "e_regionpol_6C"
## [185] "e_cow_exports"                 "e_cow_imports"
## [187] "e_gdp"                         "e_gdp_sd"
## [189] "e_gdppc"                       "e_gdppc_sd"
## [191] "e_miinflat"                    "e_pop"
## [193] "e_pop_sd"                      "e_total_fuel_income_pc"
## [195] "e_total_oil_income_pc"         "e_total_resources_income_pc"
## [197] "e_radio_n"                     "e_miferrat"
## [199] "e_mipopula"                    "e_miurbani"
## [201] "e_miurbpop"                    "e_pefeliex"
## [203] "e_peinfmor"                    "e_pelifeex"
## [205] "e_pematmor"                    "e_wb_pop"
## [207] "e_civil_war"                   "e_miinteco"
## [209] "e_miinterc"                    "e_pt_coup"
## [211] "e_pt_coup_attempts"
```

2. `select` the interested identifiers

   e.g. "country_name", "country_id", "year"

```
d |> select(country_name, country_id, year)
```

```
## # A tibble: 6,789 x 3
##    country_name country_id  year
##    <chr>             <dbl> <dbl>
##  1 Mexico                3  1984
##  2 Mexico                3  1985
##  3 Mexico                3  1986
##  4 Mexico                3  1987
##  5 Mexico                3  1988
##  6 Mexico                3  1989
##  7 Mexico                3  1990
##  8 Mexico                3  1991
##  9 Mexico                3  1992
## 10 Mexico                3  1993
## # i 6,779 more rows
```

3. Find out unique data categories (use `select` and `distinct`)

   e.g. What countries are distinct?

```
d |> select(country_name) |> distinct()
```

```
## # A tibble: 181 x 1
##    country_name
##    <chr>
##  1 Mexico
##   2 Suriname
##   3 Sweden
##   4 Switzerland
##   5 Ghana
##   6 South Africa
##   7 Japan
##   8 Burma/Myanmar
##   9 Russia
##  10 Albania
## # i 171 more rows
```

4. Select the interested columns

   e.g. country identifiers, GDP, and GDP per capita

   ```
   d |> select(country_name, country_id, year, e_gdp, e_gdppc)
   ```

   ```
   ## # A tibble: 6,789 x 5
   ##    country_name country_id  year   e_gdp e_gdppc
   ##    <chr>             <dbl> <dbl>   <dbl>   <dbl>
   ##  1 Mexico                3  1984  93563.    11.7
   ##  2 Mexico                3  1985  94259.    11.5
   ##  3 Mexico                3  1986  92750.    11.1
   ##  4 Mexico                3  1987  93220.    10.9
   ##  5 Mexico                3  1988  94687.    10.8
   ##  6 Mexico                3  1989  98145.    11.0
   ##  7 Mexico                3  1990 103254.    11.4
   ##  8 Mexico                3  1991 107374.    11.6
   ##  9 Mexico                3  1992 111533.    11.9
   ## 10 Mexico                3  1993 114611.    12.0
   ## # i 6,779 more rows
   ```

5. Assign interest columns into a new object/data set (<-)

   e.g. d_gdp

   ```
   d_gdp <- d |> select(country_name, country_id, year, e_gdp, e_gdppc)
   ```

3. rename columns to make names more informative

```
d_gdp |>
  rename("GDP"= "e_gdp", "GDP_per_capita" = "e_gdppc",
         "Country" = "country_name", "ID" = "country_id",
         "Year" = "year")
```

```
## # A tibble: 6,789 x 5
##    Country     ID  Year     GDP GDP_per_capita
##    <chr>    <dbl> <dbl>   <dbl>          <dbl>
##  1 Mexico       3  1984  93563.           11.7
##  2 Mexico       3  1985  94259.           11.5
##  3 Mexico       3  1986  92750.           11.1
##  4 Mexico       3  1987  93220.           10.9
```

```
##  5 Mexico      3  1988  94687.           10.8
##  6 Mexico      3  1989  98145.           11.0
##  7 Mexico      3  1990 103254.           11.4
##  8 Mexico      3  1991 107374.           11.6
##  9 Mexico      3  1992 111533.           11.9
## 10 Mexico      3  1993 114611.           12.0
## # i 6,779 more rows
```

note: make sure names don't have spaces in them, use _ or .

```
d_gdp <- d_gdp |>
  rename("GDP"= "e_gdp", "GDP_per_capita" = "e_gdppc",
         "Country" = "country_name", "ID" = "country_id",
         "Year" = "year")
```

4. slice rows (extract)

1. extract maximum values `slice_max`

e.g. want countries-year with the highest GDP

```
d_gdp |>
  slice_max(order_by = GDP, n = 10)
```

```
## # A tibble: 10 x 5
##    Country                     ID  Year      GDP GDP_per_capita
##    <chr>                    <dbl> <dbl>    <dbl>          <dbl>
##  1 China                      110  2019 2279809.           15.4
##  2 China                      110  2018 2205730.           14.9
##  3 China                      110  2017 2136176.           14.5
##  4 United States of America    20  2019 2118706.           60.6
##  5 United States of America    20  2018 2077898.           59.6
##  6 China                      110  2016 2039529.           13.9
##  7 United States of America    20  2017 2023242.           58.5
##  8 United States of America    20  2016 1980809.           57.6
##  9 China                      110  2015 1953127.           13.3
## 10 United States of America    20  2015 1942092.           56.7
```

2. extract minimum values `slice_min`

e.g. want countries-year with the lowest GDP

```
d_gdp |> slice_min(order_by = GDP, n =10)
```

```
## # A tibble: 10 x 5
##    Country                  ID  Year   GDP GDP_per_capita
##    <chr>                 <dbl> <dbl> <dbl>          <dbl>
##  1 Sao Tome and Principe   196  1988  24.0           2.04
##  2 Sao Tome and Principe   196  1987  24.0           2.08
##  3 Sao Tome and Principe   196  1986  24.4           2.17
##  4 Sao Tome and Principe   196  1984  24.7           2.29
##  5 Sao Tome and Principe   196  1985  24.9           2.26
##  6 Sao Tome and Principe   196  1989  25.0           2.06
##  7 Sao Tome and Principe   196  1990  25.2           2.03
##  8 Sao Tome and Principe   196  1992  25.2           1.95
##  9 Sao Tome and Principe   196  1991  25.3           1.99
## 10 Sao Tome and Principe   196  1993  25.5           1.93
```

3. Sample 10 observations

```
d_gdp |>
  slice_sample(n=10)
```

```
## # A tibble: 10 x 5
##    Country                ID  Year     GDP GDP_per_capita
##    <chr>               <dbl> <dbl>   <dbl>          <dbl>
##  1 Sweden                  5  1998  32149.          34.2
##  2 Tanzania               47  2014  11648.           2.21
##  3 Cameroon              108  2000   3795.           2.27
##  4 Eritrea               115  1990   4067.          17.2
##  5 Georgia               118  2012   4104.           9.17
##  6 Palestine/West Bank   128  2010      NA          NA
##  7 Malawi                 87  2021      NA          NA
##  8 United Kingdom        101  2016 279498.          39.6
##  9 Lebanon                44  1984   1849.           6.25
## 10 Papua New Guinea       93  2001   1684.           2.67
```

4. Sample 10% of the observations

```
d_gdp |>
  slice_sample(prop=0.1)
```

```
## # A tibble: 678 x 5
##    Country              ID  Year     GDP GDP_per_capita
##    <chr>             <dbl> <dbl>   <dbl>          <dbl>
##  1 Equatorial Guinea   160  1986    91.8           2.46
##  2 Lebanon              44  1991  1859.           5.76
##  3 Eswatini            132  2010   907.           8.00
##  4 Libya               124  2015  6076.           8.89
##  5 Cape Verde           70  2017   355.           6.17
##  6 Costa Rica           73  2020     NA          NA
##  7 Botswana             68  2016  3367.          14.5
##  8 Guatemala            78  1993  4787.           4.62
##  9 Nepal                58  2003  4009.           1.53
## 10 Peru                 30  1993 10643.           4.36
## # i 668 more rows
```

5. specify a random seed with which the system usese to generate the "random sample" (defined random seed is able to reproduce same random sample) `set.seed`

```r
set.seed(52)
d_gdp |> slice_sample(prop = 0.1)
```

```
## # A tibble: 678 x 5
##    Country         ID  Year      GDP GDP_per_capita
##    <chr>        <dbl> <dbl>    <dbl>          <dbl>
##  1 Cape Verde      70  1988     76.5           2.18
##  2 Oman           187  1991   2955.           14.7
##  3 Romania        190  2010  30202.           14.0
##  4 South Korea     42  2001 124701.           24.6
##  5 Mozambique      57  2012   3589.            1.41
##  6 Bulgaria       152  1992   8739.            9.53
##  7 Morocco         90  2001  15549.            5.03
##  8 Vietnam         34  1990  10537.            1.47
##  9 Canada          66  1985  83713.           30.4
## 10 Serbia         198  1987  17430.            7.64
## # i 668 more rows
```

5. Subset data by row (`fliter`)

e.g. want: data from 2000 to 2005 (inclusive)

```r
d_gdp |>
  filter(Year >= 2000 & Year <= 2005)
```

```
## # A tibble: 1,062 x 5
##    Country      ID  Year     GDP GDP_per_capita
##    <chr>     <dbl> <dbl>   <dbl>          <dbl>
##  1 Mexico        3  2000 145206.          13.7
##  2 Mexico        3  2001 146993.          13.6
##  3 Mexico        3  2002 148549.          13.6
##  4 Mexico        3  2003 151035.          13.7
##  5 Mexico        3  2004 156578.          14.1
##  6 Mexico        3  2005 162094.          14.3
##  7 Suriname      4  2000    383.           7.67
##  8 Suriname      4  2001    402.           7.93
##  9 Suriname      4  2002    423.           8.25
## 10 Suriname      4  2003    451.           8.67
## # i 1,052 more rows
```

e.g. want: China data

```r
d_gdp |>
  filter(Country == "China")
```

```
## # A tibble: 39 x 5
##    Country    ID  Year     GDP GDP_per_capita
##    <chr>   <dbl> <dbl>   <dbl>          <dbl>
##  1 China     110  1984 243976.           2.21
##  2 China     110  1985 265805.           2.36
##  3 China     110  1986 285707.           2.50
##  4 China     110  1987 308227.           2.65
##  5 China     110  1988 322596.           2.73
##  6 China     110  1989 327739.           2.74
##  7 China     110  1990 315683.           2.63
##  8 China     110  1991 329836.           2.71
##  9 China     110  1992 359817.           2.90
## 10 China     110  1993 393449.           3.15
## # i 29 more rows
```

e.g. want: 2000~2005 from China

```
d_gdp |>
  filter(Year >= 2000 & Year <= 2005|Country == "China")
```

```
## # A tibble: 1,095 x 5
##    Country     ID  Year     GDP GDP_per_capita
##    <chr>    <dbl> <dbl>   <dbl>          <dbl>
##  1 Mexico       3  2000 145206.          13.7
##  2 Mexico       3  2001 146993.          13.6
##  3 Mexico       3  2002 148549.          13.6
##  4 Mexico       3  2003 151035.          13.7
##  5 Mexico       3  2004 156578.          14.1
##  6 Mexico       3  2005 162094.          14.3
##  7 Suriname     4  2000    383.           7.67
##  8 Suriname     4  2001    402.           7.93
##  9 Suriname     4  2002    423.           8.25
## 10 Suriname     4  2003    451.           8.67
## # i 1,085 more rows
```

6. arrange

e.g. want: sort the rows by GDP per capita (lowest to highest)

```
d_gdp |> arrange(GDP_per_capita)
```

```
## # A tibble: 6,789 x 5
##    Country                             ID  Year    GDP GDP_per_capita
##    <chr>                            <dbl> <dbl>  <dbl>          <dbl>
##  1 Liberia                             86  1995   62.3          0.286
##  2 Liberia                             86  1994   65.5          0.307
##  3 Liberia                             86  1996   70.6          0.309
##  4 Liberia                             86  1993   81.5          0.383
##  5 Liberia                             86  1997  107.           0.429
##  6 Liberia                             86  1992  113.           0.53
##  7 Democratic Republic of the Congo   111  2002 2966.           0.538
##  8 Democratic Republic of the Congo   111  2001 2890.           0.54
##  9 Liberia                             86  1998  147.           0.543
## 10 Democratic Republic of the Congo   111  2003 3141.           0.552
## # i 6,779 more rows
```

e.g. want: sort the rows by GDP per capita (highest to lowest)

```r
d_gdp |> arrange(desc(GDP_per_capita))
```

```
## # A tibble: 6,789 x 5
##    Country                 ID  Year    GDP GDP_per_capita
##    <chr>                <dbl> <dbl>  <dbl>          <dbl>
##  1 United Arab Emirates   207  1984 16817.          115.
##  2 United Arab Emirates   207  1985 15946.          103.
##  3 Qatar                   94  2012 23055.          101.
##  4 Qatar                   94  2011 21273.          100.
##  5 Qatar                   94  2013 24074.           98.9
##  6 United Arab Emirates   207  1991 20567.           96.5
##  7 United Arab Emirates   207  1992 21506.           95.7
##  8 Qatar                   94  2014 24194.           95.3
##  9 Qatar                   94  2010 18107.           94.4
## 10 United Arab Emirates   207  2000 31871.           93.3
## # i 6,779 more rows
```

7. Perform (4) (5) (6) group by groups: `group_by`, `ungroup`

e.g. want year of the highest development level for each country/region respectively

```r
d_gdp |>
    group_by(Country) |>
    slice_max(GDP, n = 1)
```

```
## # A tibble: 341 x 5
## # Groups:   Country [181]
##    Country        ID  Year     GDP GDP_per_capita
##    <chr>       <dbl> <dbl>   <dbl>          <dbl>
##  1 Afghanistan    36  2019   6775.           1.74
##  2 Albania        12  2019   3490.          11.3
##  3 Algeria       103  2019  52143.          11.6
##  4 Angola        104  2015  17449.           6.56
##  5 Argentina      37  2017  80302.          17.2
##  6 Armenia       105  2019   3903.          12.3
##  7 Australia      67  2019 127644.          48.1
##  8 Austria       144  2019  44063.          46.2
##  9 Azerbaijan    106  2014  15216.          15.1
## 10 Bahrain       146  2018   5149.          30.9
## # i 331 more rows
```

e.g. want number of entries there are for each country `count`

```r
d_gdp |>
  group_by(Country) |>
  count()
```

```
## # A tibble: 181 x 2
## # Groups:   Country [181]
##    Country         n
##    <chr>       <int>
##  1 Afghanistan    39
##  2 Albania        39
##  3 Algeria        39
##  4 Angola         39
##  5 Argentina      39
##  6 Armenia        33
##  7 Australia      39
##  8 Austria        39
##  9 Azerbaijan     33
## 10 Bahrain        39
## # i 171 more rows
```

e.g. want: for each country, get the year when it has worst GDP

```
d_gdp |>
  group_by(Country) |>
  slice_min(order_by = GDP, n=1)
```

```
## # A tibble: 341 x 5
## # Groups:   Country [181]
##    Country        ID  Year    GDP GDP_per_capita
##    <chr>       <dbl> <dbl>  <dbl>          <dbl>
##  1 Afghanistan    36  1994  1573.           0.85
##  2 Albania        12  1992   995.           2.98
##  3 Algeria       103  1988 22997.           8.83
##  4 Angola        104  1984  3001.           3.06
##  5 Argentina      37  1985 25577.           8.43
##  6 Armenia       105  1994  1037.           3.12
##  7 Australia      67  1984 42768.          25.6
##  8 Austria       144  1984 18343.          22.9
##  9 Azerbaijan    106  1996  2362.           2.91
## 10 Bahrain       146  1986   726.          15.4
## # i 331 more rows
```

8. Create new column in the data: `group_by`, `mutate`, `ungroup`

e.g. column name is 'New'

```
d_gdp |> mutate(New = 1)
```

```
## # A tibble: 6,789 x 6
##    Country     ID  Year     GDP GDP_per_capita   New
##    <chr>    <dbl> <dbl>   <dbl>          <dbl> <dbl>
##  1 Mexico      3  1984  93563.           11.7     1
##  2 Mexico      3  1985  94259.           11.5     1
##  3 Mexico      3  1986  92750.           11.1     1
##  4 Mexico      3  1987  93220.           10.9     1
##  5 Mexico      3  1988  94687.           10.8     1
##  6 Mexico      3  1989  98145.           11.0     1
##  7 Mexico      3  1990 103254.           11.4     1
##  8 Mexico      3  1991 107374.           11.6     1
##  9 Mexico      3  1992 111533.           11.9     1
## 10 Mexico      3  1993 114611.           12.0     1
## # i 6,779 more rows
```

```
d_gdp |> mutate(New = GDP)
```

```
## # A tibble: 6,789 x 6
##    Country    ID  Year     GDP GDP_per_capita     New
##    <chr>   <dbl> <dbl>   <dbl>          <dbl>   <dbl>
##  1 Mexico      3  1984  93563.           11.7  93563.
##  2 Mexico      3  1985  94259.           11.5  94259.
##  3 Mexico      3  1986  92750.           11.1  92750.
##  4 Mexico      3  1987  93220.           10.9  93220.
##  5 Mexico      3  1988  94687.           10.8  94687.
##  6 Mexico      3  1989  98145.           11.0  98145.
##  7 Mexico      3  1990 103254.           11.4 103254.
##  8 Mexico      3  1991 107374.           11.6 107374.
##  9 Mexico      3  1992 111533.           11.9 111533.
## 10 Mexico      3  1993 114611.           12.0 114611.
## # i 6,779 more rows
```

```
d_gdp |> mutate(New = log(GDP))
```

```
## # A tibble: 6,789 x 6
##    Country    ID  Year     GDP GDP_per_capita   New
##    <chr>   <dbl> <dbl>   <dbl>          <dbl> <dbl>
##  1 Mexico      3  1984  93563.           11.7  11.4
##  2 Mexico      3  1985  94259.           11.5  11.5
##  3 Mexico      3  1986  92750.           11.1  11.4
##  4 Mexico      3  1987  93220.           10.9  11.4
##  5 Mexico      3  1988  94687.           10.8  11.5
##  6 Mexico      3  1989  98145.           11.0  11.5
##  7 Mexico      3  1990 103254.           11.4  11.5
##  8 Mexico      3  1991 107374.           11.6  11.6
##  9 Mexico      3  1992 111533.           11.9  11.6
## 10 Mexico      3  1993 114611.           12.0  11.6
## # i 6,779 more rows
```

```
d_gdp |> mutate(New = log(GDP) + 1)
```

```
## # A tibble: 6,789 x 6
##    Country    ID  Year     GDP GDP_per_capita   New
##    <chr>   <dbl> <dbl>   <dbl>          <dbl> <dbl>
##  1 Mexico      3  1984  93563.           11.7  12.4
##  2 Mexico      3  1985  94259.           11.5  12.5
##  3 Mexico      3  1986  92750.           11.1  12.4
##  4 Mexico      3  1987  93220.           10.9  12.4
##  5 Mexico      3  1988  94687.           10.8  12.5
##  6 Mexico      3  1989  98145.           11.0  12.5
##  7 Mexico      3  1990 103254.           11.4  12.5
##  8 Mexico      3  1991 107374.           11.6  12.6
##  9 Mexico      3  1992 111533.           11.9  12.6
## 10 Mexico      3  1993 114611.           12.0  12.6
## # i 6,779 more rows
```

e.g. want new column to be the GDP relative to average GDP

```
d_gdp |>
  mutate(GDP_over_avg = GDP / mean(GDP, na.rm = TRUE))
```

```
## # A tibble: 6,789 x 6
##    Country    ID  Year     GDP GDP_per_capita GDP_over_avg
##    <chr>   <dbl> <dbl>   <dbl>          <dbl>        <dbl>
##  1 Mexico      3  1984  93563.           11.7         2.11
##  2 Mexico      3  1985  94259.           11.5         2.13
##  3 Mexico      3  1986  92750.           11.1         2.09
##  4 Mexico      3  1987  93220.           10.9         2.10
##  5 Mexico      3  1988  94687.           10.8         2.14
##  6 Mexico      3  1989  98145.           11.0         2.21
##  7 Mexico      3  1990 103254.           11.4         2.33
##  8 Mexico      3  1991 107374.           11.6         2.42
##  9 Mexico      3  1992 111533.           11.9         2.52
## 10 Mexico      3  1993 114611.           12.0         2.59
## # i 6,779 more rows
```

e.g. want new column to be GDP relative to average GDP in the world 1984-2022

```
d_gdp |>
  group_by(Country) |>
  mutate(GDP_over_avg = GDP / mean(GDP, na.rm = TRUE))
```

```
## # A tibble: 6,789 x 6
## # Groups:   Country [181]
##    Country    ID  Year     GDP GDP_per_capita GDP_over_avg
##    <chr>   <dbl> <dbl>   <dbl>          <dbl>        <dbl>
##  1 Mexico      3  1984  93563.           11.7        0.624
##  2 Mexico      3  1985  94259.           11.5        0.628
##  3 Mexico      3  1986  92750.           11.1        0.618
##  4 Mexico      3  1987  93220.           10.9        0.622
##  5 Mexico      3  1988  94687.           10.8        0.631
##  6 Mexico      3  1989  98145.           11.0        0.654
##  7 Mexico      3  1990 103254.           11.4        0.688
##  8 Mexico      3  1991 107374.           11.6        0.716
##  9 Mexico      3  1992 111533.           11.9        0.744
## 10 Mexico      3  1993 114611.           12.0        0.764
## # i 6,779 more rows
```

e.g. country-year development level with reference to that of 1984 `first`

```
d_gdp |>
  group_by(Country) |>
  arrange(Year) |>
  mutate(GDP_over_1984 = GDP / first(GDP)) |>
  ungroup() |>
  arrange(Country, Year)
```

```
## # A tibble: 6,789 x 6
##    Country        ID  Year   GDP GDP_per_capita GDP_over_1984
##    <chr>       <dbl> <dbl> <dbl>          <dbl>         <dbl>
##  1 Afghanistan    36  1984 2723.          2.03          1
##  2 Afghanistan    36  1985 2690.          2.01          0.988
##  3 Afghanistan    36  1986 2617.          1.97          0.961
##  4 Afghanistan    36  1987 2471.          1.86          0.907
##  5 Afghanistan    36  1988 2317.          1.73          0.851
##  6 Afghanistan    36  1989 2173.          1.59          0.798
##  7 Afghanistan    36  1990 2066.          1.46          0.759
##  8 Afghanistan    36  1991 1953.          1.32          0.717
##  9 Afghanistan    36  1992 1842.          1.16          0.676
## 10 Afghanistan    36  1993 1676.          0.973         0.616
## # i 6,779 more rows
```

e.g. want year-on-year economic growth

```
d_gdp |>
  group_by(Country) |>
  arrange(Year) |>
  mutate(GDP_yoy_change = GDP-lag(GDP, n=1))|>
  ungroup()|>
  arrange(Country, Year)
```

```
## # A tibble: 6,789 x 6
##    Country        ID  Year   GDP GDP_per_capita GDP_yoy_change
##    <chr>       <dbl> <dbl> <dbl>          <dbl>          <dbl>
##  1 Afghanistan    36  1984 2723.          2.03            NA
##  2 Afghanistan    36  1985 2690.          2.01           -33.1
##  3 Afghanistan    36  1986 2617.          1.97           -72.8
##  4 Afghanistan    36  1987 2471.          1.86          -146.
##  5 Afghanistan    36  1988 2317.          1.73          -154.
##  6 Afghanistan    36  1989 2173.          1.59          -144.
##  7 Afghanistan    36  1990 2066.          1.46          -107.
##  8 Afghanistan    36  1991 1953.          1.32          -113.
##  9 Afghanistan    36  1992 1842.          1.16          -111.
## 10 Afghanistan    36  1993 1676.          0.973         -166.
## # i 6,779 more rows
```

9. Summarise the data: group_by, summarise, ungroup

e.g. want: average GDP level of the world

```
d_gdp |> summarise (gdp_average = mean(GDP, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   gdp_average
##         <dbl>
## 1      44324.
```

e.g. want: average developmental level from 1984 to 2022

```
d_gdp |>
  group_by(Country) |>
  summarise(GDP_average = mean(GDP, na.rm = TRUE),
            GDPpc_average = mean(GDP_per_capita, na.rm = TRUE))
```

```
## # A tibble: 181 x 3
##    Country     GDP_average GDPpc_average
##    <chr>             <dbl>         <dbl>
##  1 Afghanistan       3374.          1.35
##  2 Albania           2029.          6.33
##  3 Algeria          35153.         10.1
##  4 Angola            8133.          4.07
##  5 Argentina        53263.         13.2
##  6 Armenia           2163.          6.83
##  7 Australia        83495.         38.3
##  8 Austria          31285.         35.6
##  9 Azerbaijan        8230.          8.72
## 10 Bahrain           2493.         24.4
## # i 171 more rows
```

e.g. want: GDP growth and GDP per capita growth: comparing 2019 with 1984

```
d_gdp|>
  filter(Year >= 1984, Year <= 2019) |>
  group_by(Country)|>
  arrange(Year)|>
  summarise(GDP_growth_2019_1984 = (last(GDP)-first(GDP)) / first(GDP),
            GDPpc_growth_2019_1984 = (last(GDP_per_capita)-first(GDP_per_capita)) / first(GDP_per_capita
  ungroup() |>
  arrange(Country)
```

```
## # A tibble: 181 x 3
##    Country     GDP_growth_2019_1984 GDPpc_growth_2019_1984
##    <chr>                      <dbl>                  <dbl>
##  1 Afghanistan                 1.49                 -0.142
##  2 Albania                     1.84                  1.82
##  3 Algeria                     1.14                  0.118
##  4 Angola                      4.64                  0.763
##  5 Argentina                   2.03                  0.922
##  6 Armenia                    NA                    NA
##  7 Australia                   1.98                  0.879
##  8 Austria                     1.40                  1.02
##  9 Azerbaijan                  1.47                  0.766
## 10 Bahrain                     5.50                  0.711
## # i 171 more rows
```

10. Data availability/integrity check

e.g. want: find which GDP values are missing

```
d_gdp|>
  mutate(GDP_missing = is.na(GDP), .after = GDP)
```

```
## # A tibble: 6,789 x 6
##    Country     ID  Year     GDP GDP_missing GDP_per_capita
##    <chr>    <dbl> <dbl>   <dbl> <lgl>                <dbl>
##  1 Mexico       3  1984  93563. FALSE                 11.7
##  2 Mexico       3  1985  94259. FALSE                 11.5
##  3 Mexico       3  1986  92750. FALSE                 11.1
##  4 Mexico       3  1987  93220. FALSE                 10.9
##  5 Mexico       3  1988  94687. FALSE                 10.8
##  6 Mexico       3  1989  98145. FALSE                 11.0
##  7 Mexico       3  1990 103254. FALSE                 11.4
##  8 Mexico       3  1991 107374. FALSE                 11.6
##  9 Mexico       3  1992 111533. FALSE                 11.9
## 10 Mexico       3  1993 114611. FALSE                 12.0
## # i 6,779 more rows
```

e.g. want: find how many GDP values are missing for each country

```
d_gdp|>
  mutate(GDP_missing = as.numeric(is.na(GDP)), .after = GDP)|>
  group_by(Country)|>
  summarise(N_GDP_missing = sum(GDP_missing))
```

```
## # A tibble: 181 x 2
##    Country     N_GDP_missing
##    <chr>               <dbl>
##  1 Afghanistan             3
##  2 Albania                 3
##  3 Algeria                 3
##  4 Angola                  3
##  5 Argentina               3
##  6 Armenia                 4
##  7 Australia               3
##  8 Austria                 3
##  9 Azerbaijan              3
## 10 Bahrain                 3
## # i 171 more rows
```