SPEC Lab R Resources: Data Management 3- Group Work

Alix Ziff and Miriam Barnum

Summer 2020

Data Management for Visualization

We're will continuing to work with our IDC powersharing data to hone our data management skillset. We will use the packages `dplyr` and `countrycode` to work with country-year data. For these exercises, you will follow similar protocol to the Walk-Through-Work but apply it to new variables.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(countrycode)
```

```
## Warning: package 'countrycode' was built under R version 3.6.2
```

```
load("/Volumes/GoogleDrive/Shared drives/SPEC Powersharing S20/data/MERGED_IDC_Controls_MB_05222020.RDa
```

Exercise 1 Say we want to look at global averages of subnational data. Collapse our country-year data to get global averages by year for the variable subpolice.

```
subpolice_global <- idc_controls %>%
  group_by(year) %>%
  dplyr::summarise(subpolice_mean = mean(subpolice_IDC, na.rm = T))
```

Exercise 1.2 Take a look at your results, what is the average for 2007?

```
View(subpolice_global)
```

Exercise 2 Now, incorporate the other subnational variables in the IDC dataset on subnational education and taxes (subed, subtax).

```
subnational_global <- idc_controls %>%
  group_by(year) %>%
  summarise_at(vars(subpolice_IDC, subed_IDC, subtax_IDC), mean, na.rm = T)
```

Exercise 3 For our next exercise, let's look at regional averages for another variable. You choose. *Helpful Hint*: You'll need to add a new variable, either by region or continent. *Helpful Hint*: Avoid categorical variables or those with several NA values... review Data Management I for how to explore your data.

```
GDP_capita_regional <- idc_controls %>%

# add region variable ("region" uses the WDI regions, "continent" is another option, run ?codelist to s
```

```
  mutate(region = countrycode(gwno, "gwn", "region", custom_match = c("55" = "GRN", "56"= "SLU",
                                                                       "591" = "SEY", "816" = "DRV", "935
                                                                       "972" = "TON", "990" = "WSM"))) %
  group_by(region, year) %>%
  dplyr::summarise(GDP_capita_mean = mean(gdppc_WDI_PW, na.rm = T)) %>%
  filter(region %in% c("East Asia & Pacific", "Europe & Central Asia", "Latin America & Caribbean",
                       "Middle East & North Africa", "North America", "South Asia", "Sub-Saharan Africa
```

Exercise 4 Let's practice simplifying and summarizing data. Choose a single variable and find either the decade averages, minima, maxima, or median values.

```
freedom_expression_decade <- idc_controls %>%

  # add decade variable (year %% 10 gives us the last digit, then we subtract from the year to get the
  mutate(decade = year - year %% 10) %>%

  group_by(gwno, decade) %>%
  dplyr::summarise(freedom_expression_mean = mean(v2x_freexp_altinf_VDEM, na.rm = T),
                   freedom_expression_med = median(v2x_freexp_altinf_VDEM, na.rm = T),
                   freedom_expression_max = max(v2x_freexp_altinf_VDEM, na.rm = T),
                   freedom_expression_min = min(v2x_freexp_altinf_VDEM, na.rm = T))
```

```
## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf
```

```
## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf
```

```
## Warning in max(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## max; returning -Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf
```

```
## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf

## Warning in min(v2x_freexp_altinf_VDEM, na.rm = T): no non-missing arguments to
## min; returning Inf
```

Exercise 5 Do the same thing as above, but incorporate multiple variables.

```
subnational_policy_decade <- idc_controls %>%
  mutate(decade = year - year %% 10) %>%
  group_by(gwno, decade) %>%
  summarise_at(vars(subed_IDC, subpolice_IDC, subtax_IDC), funs(mean, median, max, min), na.rm = T)
```

```
## Warning: funs() is soft deprecated as of dplyr 0.8.0
## Please use a list of either functions or lambdas:
##
##   # Simple named list:
##   list(mean = mean, median = median)
##
##   # Auto named with `tibble::lst()`:
##   tibble::lst(mean, median)
```

```
##
##   # Using lambdas
##   list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))
## This warning is displayed once per session.
```

Exercise 6 Pick a year so that we have a single value per country. *Helpful Hint*: Remember you can either do this in base R or dplyr.

```
idc_2002 <- idc_controls[idc_controls$year == 2002,]
```

Exercises 7-10 DIY: apply all the above steps to make a data set which includes annual values for some individual countries within one region, and the global average.

```
# lets do this for the subnational policy variables again

# add a region name to global average values we have
subnational_global$region <- "Global"

# regional averages (by continent)
subnational_americas <- idc_controls %>%
  mutate(region = countrycode(gwno, "gwn", "continent")) %>%
  group_by(region, year) %>%
  summarise_at(vars(subpolice_IDC, subed_IDC, subtax_IDC), mean, na.rm = T)
```

```
## Warning in countrycode(gwno, "gwn", "continent"): Some values were not matched unambiguously: 55, 56
```

```
subnational_americas <- filter(subnational_americas, region %in% c("Americas")) #just keep the Americas

# annual data for US, Canada, and Mexico
idc_sub <- idc_controls %>%
  filter(country %in% c("United States of America", "Canada", "Mexico")) %>%

  # rbind for standardized names across the dfs
  select(region = country, year, subpolice_IDC, subed_IDC, subtax_IDC)

# bind into one df
df <- idc_sub %>%
  bind_rows(subnational_americas) %>%
  bind_rows(subnational_global)
```