SPEC Lab REU R Resources: Data Management III: The Search for the Tidyverse

Alix Ziff, Miriam Barnum, Gaea Morales, and Zachary Johnson

Summer 2020, Version 3: July 23

Data Management for Visualization

In this module, we will continue working with the data to hone our data management skillset. We will use the packages `tidyverse` and `countrycode` to work with country-year data.

```
library(tidyverse)
library(countrycode)
library(readr)

setwd("~/Google Drive File Stream/My Drive/SPEC Summer 2020/Training Data Science/0 Training Data/")

socioeconomic_status <- read_csv("GLOB.SES.csv")
```

##Collapsing Country-Year Data ###univariate First, we want to collapse the country-year data down to global averages. In other words, we need to `groupby()`. We'll start by focusing on a single variable: SES (Socioeconomic status score (percentile) based on GDP per capita and educational attainment (n=174)). *Helpful Hint: make sure you save your steps as an object*

```
SES_global <- socioeconomic_status %>%
#Using the object 'socioeconomic_status' (created in the earlier Data Management exercises)
#we will create a new object rents_global
  group_by(year) %>% #we're using group_by to split our data into groups by decade and
#then use the dplyr command summarise() to aggregate observations by decade
  dplyr::summarise(SES_mean = mean(SES, na.rm = T))
#looking at Socioeconomic Status from the data and removing all NAs from the dataset
View(SES_global)
```

Helpful Hint: We use the package:: to specify a function within a package that may have a similar name to a command in another package.

###multivariate Let's get global averages for multiple variables at once. We may want to compare resource rents with trade. To take a look, we'll do the same thing as above, but will use the `summarise_at()` function to bring in our second variable. We'll call it trade_rents.

Calculate global averages and then group_by year.

```
SES_gdppc_global <- socioeconomic_status %>%
  group_by(year) %>%
#integrating two variables into a new object and grouping it by year,
```

```
# checking the mean and removing NAs
  summarise_at(vars(SES, gdppc), mean, na.rm = T)
#specifying that we will apply the summary to rents and trade
View(trade_rents_global)
#same thing but with 2 variables
```

Helpful Hint vars() serves the same purpose as select() but you can use it within another function.

There are a lot of countries, different definitions of what a country is, and different ways to classify or label these countries. The `countrycode` package allows us to easily access different aspects of countries based on the code in the given set of data. In this instance we specify the wbid (world bank character id) variable as the sourcevar which just specifies what variable we are pulling values from. We specify "wb" as the origin, or the indexing format we are pulling from, and the countrycode source code knows to parse this as the appropriate world bank index format. Lastly we specify that we wish to pull the regionality that the countrycode source code also knows to do for us.

```
SES_regional <- socioeconomic_status %>%
  mutate(region = countrycode(sourcevar =  wbid, origin = "wb", destination = "region")) %>%
#using the countrycode package to add our region variable with the world band id regional classificatio
  group_by(region, year) %>% #grouping our data by both region and year
  dplyr::summarise(SES_mean_regional = mean(SES, na.rm = T))
#aggregating so that the output is the global averages of resource rents by year and region
View(SES_regional)
```

Summarizing & Simplifying

###univariate Let's try to simplify our data a bit so we can identify broad patterns. We'll take a look at our averages, minimum, maximum, and median.

```
region_SES_stat <- socioeconomic_status %>%
  mutate(region = countrycode(sourcevar =  wbid, origin = "wb", destination = "region")) %>%
  group_by(region, year) %>%
  dplyr::summarise(SES_mean = mean(SES, na.rm = T),
                   SES_med = median(SES, na.rm = T),
                   SES_max = max(SES, na.rm = T),
                   SES_min = min(SES, na.rm = T))
#taking the summary statistics for our data

View(summary_stats_regionality)
```

###multivariate Now with multiple variables.

```
region_ALL_stat <- socioeconomic_status %>%
  mutate(region = countrycode(sourcevar =  wbid,
                              origin = "wb",
                              destination = "region")) %>%
  group_by(region, year) %>%
  summarise_at(vars(SES, gdppc, yrseduc, popshare),
               tibble::lst(mean, median, max, min),
               na.rm = T)
```