US coal exports

Joey Schechter and Devin Bunch

20 April 2021

Contents

Preliminaries:

Load libraries

It's a good idea to load your libraries at the top of the Rmd document so that everyone can see what you're using. Similarly, it's good practice to set `cache=FALSE` to ensure that the libraries are dynamically loaded each time you knit the document.

Hint: I've only added the libraries needed to download and read the data. You'll need to load additional libraries to complete this assignment. Add them here once you discover that you need them.

```
## Install the pacman package if necessary
if (!require("pacman")) install.packages("pacman")
## Install other packages using pacman::p_load()
pacman::p_load(httr, readxl, here,tinytex,tidyverse,data.table,rlang,janitor)
```

Read in the data

Use `httr::GET()` to fetch the EIA excel file for us from web. (We'll learn more about `httr`, GET and other HTTP methods when we get to webscraping next week.)

```
# library(here) ## Already loaded
# library(httr) ## Already loaded
url = "https://www.eia.gov/coal/archive/coal_historical_exports.xlsx"
## Only download the file if we need to
if(!file.exists(here::here("data/coal.xlsx"))) {
  GET(url, write_disk(here::here("data/coal.xlsx")))
}
```

```
## Response [https://www.eia.gov/coal/archive/coal_historical_exports.xlsx]
##   Date: 2021-04-20 22:22
##   Status: 200
##   Content-Type: application/vnd.openxmlformats-officedocument.spreadsheetml.sheet
##   Size: 785 kB
## <ON DISK>  /Users/devin3/Desktop/510/assignment-02-wrangling-team-03/data/coal.xlsx
```

Next, we read in the file.

```
coal = read_excel(here::here("data/coal.xlsx"), skip = 3, na = ".")
```

We are now ready to go.

1) Clean the column names

The column (i.e. variable) names aren't great: Spacing, uppercase letters, etc.

```
names(coal)
```

```
##  [1] "Year"                      "Quarter"
##  [3] "Type"                      "Customs District"
##  [5] "Coal Origin Country"       "Coal Destination Country"
##  [7] "Steam Coal"                "Steam Revenue"
##  [9] "Metallurgical"             "Metallurgical Revenue"
## [11] "Total"                     "Total Revenue"
## [13] "Coke"                      "Coke Revenue"
```

```
coal <- coal %>% clean_names()

names(coal)
```

```
##  [1] "year"                    "quarter"
##  [3] "type"                    "customs_district"
##  [5] "coal_origin_country"     "coal_destination_country"
##  [7] "steam_coal"              "steam_revenue"
##  [9] "metallurgical"           "metallurgical_revenue"
## [11] "total"                   "total_revenue"
## [13] "coke"                    "coke_revenue"
```

Clean them.

Hint: Use either `gsub()` and regular expressions or, more simply, the `janitor()` package. You will need to install the latter first.

2) Total US coal exports over time (year only)

Plot the US's total coal exports over time by year ONLY. What secular trends do you notice in the data?

Hints: If you want nicely formatted y-axis label, add `+ scale_y_continuous(labels = scales::comma)` to your `ggplot2` code.

Please put your (verbal) answers in bold.

3) Total US coal exports over time (year AND quarter)

Now do the same as the above, expect aggregated quarter of year (2001Q1, 2002Q2, etc.). Do you notice any seasonality that was masked from the yearly averages?

Hint: ggplot2 is going to want you to convert your quarterly data into actual date format before it plots nicely. (i.e. Don't leave it as a string.)

4) Exports by destination country

4.1) Create a new data frame

Create a new data frame called `coal_country` that aggregates total exports by destination country (and quarter of year). Make sure you print the resulting data frame so that it appears in the knitted R markdown document.

4.2) Inspect the data frame

It looks like some countries are missing data for a number of years and periods (e.g. Albania). Confirm that this is the case. What do you think is happening here?

4.3) Complete the data frame

Fill in the implicit missing values, so that each country has a representative row for every year-quarter time period. In other words, you should modify the data frame so that every destination country has row entries for all possible year-quarter combinations (from 2002Q1 through the most recent quarter). Order your updated data frame by country, year and, quarter.

Hints: See `?tidyr::complete()` for some convenience options. Again, don't forget to print `coal_country` after you've updated the data frame so that I can see the results.

4.4 Some more tidying up

In answering the previous question, you *may* encounter a situation where the data frame contains a quarter — probably 2021q1 — that is missing total export numbers for *all* countries. Did this happen to you? Filter out the completely missing quarter if so. Also: Why do you think this might have happened? (Please answer the latter question even if it didn't happen to you.)

4.5) Culmulative top 10 US coal export destinations

Produce a vector — call it `coal10_culm` — of the top 10 top coal destinations over the full 2002–`r`max(coal[, which(grepl('Year|year', names(coal)))], na.rm=T)' study period. What are they?

4.6) Recent top 10 US coal export destinations

Now do the same, except for most recent period on record (i.e. final quarter in the dataset). Call this vector `coal10_recent` and make sure to print it so that I can see it too. Are there any interesting differences between the two vectors? Apart from any secular trends, what else might explain these differences?

4.7) US coal exports over time by country

Plot the quarterly coal exports over time, but now disaggregated by country. In particular, highlight the top 10 (cumulative) export destinations and then sum the remaining countries into a combined "Other" category. (In other words, your figure should contain the time series of eleven different countries/categories.)

4.8) Make it pretty

Take your previous plot and add some swag to it. That is, try to make it as visually appealing as possible without overloading it with chart junk.

Hint: You've got loads of options here. If you haven't already done so, consider a more bespoke theme with the `ggthemes`*,* `hrbrthemes`*, or* `cowplot` *packages. Try out* `scale_fill_brewer()` *and* `scale_colour_brewer()` *for a range of interesting colour palettes. Try some transparency effects with* `alpha`*. Give your axis labels more refined names with the* `labs()` *layer in ggplot2. While you're at it, you might want to scale (i.e. normalise) your y-variable to get rid of all those zeros. You can shorten any country names to their ISO abbreviation; see* `?countrycode::countrycode`*. More substantively — but more complicated — you might want to re-order your legend (and the plot itself) according to the relative importance of the destination countries. See* `?forcats::fct_reorder` *or forcats::fct_relevel'.*

4.9) Make it interactive

Create an interactive version of your previous figure.

Hint: Take a look at plotly::ggplotly(), or the gganimate package.

5) Show me something interesting

There's a lot still to explore with this data set. Your final task is to show me something interesting. Drill down into the data and explain what's driving the secular trends that we have observed above. Or highlight interesting seasonality within a particular country. Or go back to the original `coal` data frame and look at exports by customs district, or by coal type. Do we changes or trends there? Etcetera. Etcetera. My only requirement is that you show your work and tell me what you have found.