HW3

Lance

2/2023

Instructions

Before we get started, I wanted to let you know something about the data for this homework: it's from the Transatlantic slave trade. This is a very sensitive subject, but I think it's valuable for us to confront the vast extent of and role slavery played in the history of the U.S. and other parts of the world. Please take a moment to read an introduction to this data here. It does a good job of capturing my intent with this assignment.

However, my intent is irrelevant if you feel completing this assignment would be harmful to you. *If you are uncomfortable working with this data, please do not hesitate to reach out to me for a different dataset. You will not be judged or penalized in any way.*

This HW is worth 10 total points.

1. Make sure to load any packages you may need right at the start. Do *NOT* include the `learnr` package, ever, unless you are writing an interactive Tutorial (which you won't do in this course) - this will cause problems.

2. Ensure that no chunks have the `include = FALSE` or `echo = FALSE` option, as I want to be able to see *all* your code and output.

3. Brief but descriptive headings and document organization (answers under headings, text near relevant code, brief explanatory text as necessary, etc.) (1 pt)

   Look to my HW1 and RMarkdown Organization examples for how to write good headings, organize your assignment, and how much narrative text (outside of code chunks) I want. But a good rule of thumb is: explain what YOU'RE doing, NOT what YOUR CODE is doing. I do NOT need to see a repeat, line-by-line narration of what your code does - you can use code comments for that. (In complex blocks you usually want at least a comment every few lines.) I DO want to see an overall summary of what you did in your analysis.

4. Figure out how to import the `slave_routes` data frame two ways:

   i) Using a single line of code in the linked introduction above to get the data directly from Github (0.3 pts),

   Comment out this code before you knit the assignment so it doesn't run and try to pull the data every time you knit. Don't delete it so I can still grade its correctness, but do comment it out once you successfully import the data this way for the first time and successfully export and re-import it locally in Q4ii.

   ii) By saving the file to your local computer and importing it "manually" (as laid out in Tutorial 2.1). The easiest way to do this is probably to export the version you got from Github. *You must use working directory notation (i.e. a "./" rather than a full file path).* (0.7 pts)

5. Using a single continuous block of code with pipes (no intermediate stopping points) for each of the below sub-parts (3 blocks in total), give me output that shows:

   i) Which port received the largest total number of enslaved people (from ships with a known number of enslaved people), and what was that number? You may show more than one port, as long as the top port is visible. HINT: The top port should have received 683310 slaves. (2 pts)

   ii) Which `ship_name` made the most journeys before the year 1800 (you may show more than one ship, as long as the top ship is visible), and during what years (prior to 1800) did it operate? Exclude the `NA` values of `ship_name` as part of your first `dplyr` verb.

   Note all the data you need to answer these questions should be visible in one piece of output. HINT: you may want the `min()` and `max()` functions. (2 pts)

   iii) Which *two* `ship_name`s made the most journeys without a named captain, in any years? Exclude the `NA` values of `ship_name` as part of your first `dplyr` verb.

   HINT: you could either create a new variable that indicates whether the ship had a missing captain or not, or simply figure out a way to count only ship journeys with missing captains.

   You may show more than two ships, as long as the top two ships are visible. (2 pts)

   Provide about 1 sentence of narrative text that interprets the output and answers each question, i-iii.

6. Using *two* separate continuous blocks of code with pipes - one for any data manipulation, and then one for the visualization - produce a line graph of the number of total slaves that arrived in all ports by *decade* (again, only for ships with a known number of slaves; include all years in this analysis). Each decade must run from 0-9 (e.g. 1770-1779, not 1765-1774). Give the axes human-readable labels.

   HINT: You'll have to create a new decade variable. If you need help, think back to how we extracted the hour and minute values from the military time in the `flights` data. You should *not* use `case_when()` to set a criterion for each individual decade; rather you should figure out a way to create this variable in 1-2 lines.

   Answer the following questions:

   i) In what decade did the Transatlantic slave trade peak?
   ii) You can calculate this manually in a separate block, but in this decade approximately how many enslaved people were being transported across the Atlantic *per day*? (3 pts)

To submit this assignment:

Ideally, knit straight to PDF by changing `html_document` to `pdf_document` in line 5 above. Otherwise:

1. Knit to HTML. An HTML document should open automatically in another RStudio window.
2. Click "Open in Browser" in that HTML document. It should open as a webpage in your default browser (e.g. Chrome).
3. Click Ctrl+P/Command+P, but instead of printing a hard copy on your printer click "Save as PDF."
4. Save and upload that document to Canvas.

A note on PDF formatting: you may notice that long lines of code "fly off the side of the page" when you knit to PDF. To fix this:

If you're on a Windows machine:

- Install the `formatR` package
- Change your `opts_chunk$set` code line to the following: `knitr::opts_chunk$set(echo = TRUE, tidy.opts=list(width.cutoff=80), tidy=TRUE)`

That should force your code to always wrap rather than fly off the edge of the page of a PDF. Note this does not fix issues of, say, plot titles that are too long getting cut off. But it should fix all the errors with your code not wrapping. Happy PDFing!

If you're on a Mac: I don't have an easy solution for you. Try and keep your lines of code under about 80 characters. Feel free to use more vertical lines of code to accomplish this. But don't waste large amounts of time formatting. I'll ask you for clarification if something critical is missing.

——BEGIN ANSWER BELOW———

Before we begin our analysis, we will first load the necessary packages.

```
pacman::p_load(tidyverse)

# 1 liner
#slave_routes <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/
```

```
#write_csv(slave_routes, "./slave_routes.csv")
slave_routes = read_csv("./slave_routes.csv")
```

```
## Rows: 36110 Columns: 8
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): ship_name, port_origin, place_of_purchase, port_arrival, captains_name
## dbl (3): voyage_id, year_arrival, n_slaves_arrived
##
## 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.
```

```
# Check that it is working
knitr::kable(head(slave_routes, n=5))
```

voyage_id	ship_name	port_origin	place_of_purchase	port_arrival	year_arrival	n_slaves_arrived	captains_name
81711	Hannah	Liverpool	Calabar	St. Vincent, port unspecified	1787	390	Smith, Bryan
81712	Hannah	Liverpool	New Calabar	Grenada, port unspecified	1789	351	Wilson, Charles
81713	Hannah	Liverpool	Bight of Biafra and Gulf of Guinea Islands, port unspecified	Kingston	1789	303	Wilson, Charles
81714	Hannah	Liverpool	Bonny	St. Vincent, port unspecified	1791	316	Young, William
81715	Hannah	Liverpool	Congo River	Grenada, port unspecified	1792	331	Young, William

We are getting an output, and I can see in my local file explorer that I have the new file `slave_routes.csv`. We are now ready to begin.

Analyzing `slave_routes` with data transformations

Port receiving the largest total number of enslaved people

We want to find the port that received the largest total number of enslaved people (from ships with a known number of enslaved people). We will group the data by the `port_arrival` variable, create an intermediate variable `n_slaves_arrived`, then use `arrange()` in combination with `desc()` to sort the dataframe in descending order according to the number of slaves arrived at each port.

```
slave_routes %>%
  select(port_arrival, n_slaves_arrived) %>%
  group_by(port_arrival) %>%
  summarize(n_ships_port = n(),
            n_slaves_port = sum(n_slaves_arrived, na.rm=TRUE)
            ) %>%
  arrange(desc(n_slaves_port)) %>%
  head(n=5) %>%
  knitr::kable()
```

port_arrival	n_ships_port	n_slaves_port
Rio de Janeiro	2887	683310
Bahia, port unspecified	4223	549810
Kingston	1622	439916
Barbados, port unspecified	2038	275703
Cap Francais	1127	272983

We see that Rio de Janeiro had the most slaves at 683310, followed by Bahia at 549810 and Kingston at 439916.

Ship with the most journeys made before the year 1800

We want to find the details of the ship that made the most number of trips before the year 1800. We will first remove all the nameless ships and the records from 1800 and onwards using `filter()`. We will then group by `ship_name` and use the `min()` and `max()` functions to find the earliest and latest recorded trips of each eligible ship, then sort the entire thing in descending order with a combination of `arrange()` and `desc()`.

```
slave_routes %>%
  filter(!is.na(ship_name) & year_arrival < 1800) %>%
  select(ship_name, year_arrival) %>%
  group_by(ship_name) %>%
  summarize(n_trips = n(),
            first_trip = min(year_arrival),
            last_trip = max(year_arrival)) %>%
  arrange(desc(n_trips)) %>%
  head(n=5) %>%
  knitr::kable()
```

ship_name	n_trips	first_trip	last_trip
Mary	211	1645	1799
Nancy	183	1711	1799
NS do Rosario S Antônio e Almas	182	1698	1797
NS da Conceição S Antônio e Almas	175	1687	1796
Africa	138	1668	1799

We see that the *Mary* has made 211 trips before the year 1800, and has been in operation from at least 1645 to 1799.

Ships with the most journeys without recorded captains

We want to find the 2 ships that made the most trips without a named captain in any years. We will first exclude ships without a name, as well as trips with captain names using `filter()`. We then group by `ship_name` and create an intermediate variable `n_trips`, which we use to count the number of trips each ship has made without a captain. We then use `arrange()` and `desc()` to sort the dataframe.

```
slave_routes %>%
  filter(!is.na(ship_name) & is.na(captains_name)) %>%
  group_by(ship_name) %>%
  summarize(n_trips = n()) %>%
  arrange(desc(n_trips)) %>%
  head(n=5) %>%
  knitr::kable()
```

ship_name	n_trips
Esperança	18
NS do Rosario S Antônio e Almas	18
NS da Conceição S Antônio e Almas	15
NS da Conceição	13
Carolina	12

We can see that the *Esperança* and the *NS do Rosario S Antônio e Almas* both made 18 trips without a named captain.

Number of Total Slaves that arrived in all ports by decade

We will count the number of slaves transported every decade by summing the known slave arrivals in that decade. We will perform integer division `%/%` to find the decade of the trip, then group all trips with known slave arrivals by decade. We then sort it by descending order with `arrange()` and `desc()` for easier viewing.

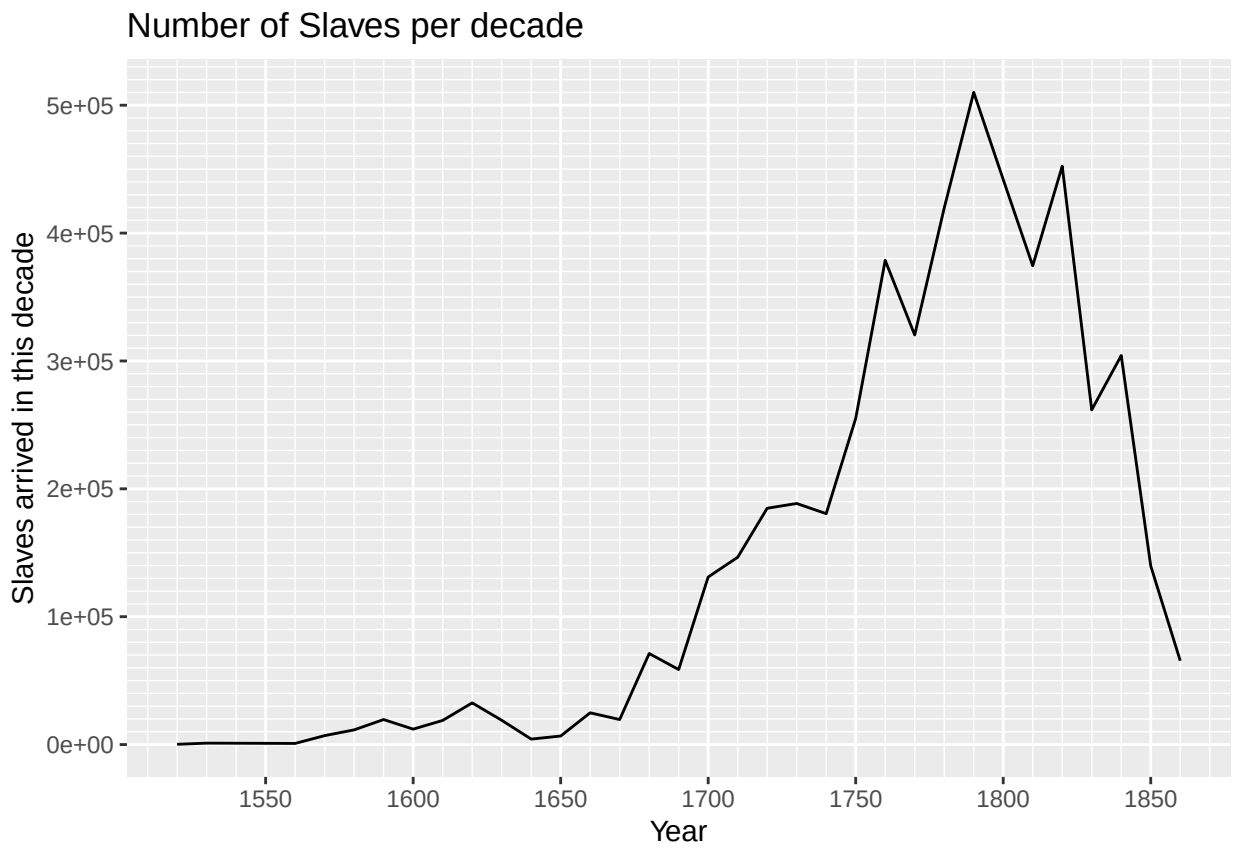
```
slave_decades <- slave_routes %>%
  filter(!is.na(n_slaves_arrived)) %>%
  select(year_arrival, n_slaves_arrived) %>%
  mutate(decade = 10 * (year_arrival %/% 10)) %>%
  group_by(decade) %>%
  summarize(n_slaves_decade = sum(n_slaves_arrived)) %>%
  arrange(desc(n_slaves_decade))

knitr::kable(head(slave_decades, n=10))
```

decade	n_slaves_decade
1790	510082
1820	452344
1800	441944
1780	419573
1760	378691
1810	374430
1770	320324
1840	304277
1830	261760
1750	255050

To plot this data, we will use a ggplot `geom_line()` line plot. We change the number of breaks and minor breaks for easier viewing, and change the labels for human readability. The y-axis is left in exponential form for less clutter.

```
ggplot(slave_decades) +
  geom_line(aes(x=decade, y=n_slaves_decade)) +
  scale_x_continuous(n.breaks=10,
                     minor_breaks = seq(1500,1900,10)) +
  labs(title = "Number of Slaves per decade") +
  scale_y_continuous(minor_breaks = seq(0,550000,10000)) +
  xlab("Year") +
  ylab("Slaves arrived in this decade")
```



```
# I wasn't sure how (or why) to utilize pipes for ggplot since
# everything is a noun and we wouldn't need pipes.
```

The slave trade (from what we can see in this dataset) peaked in the decade of 1790-1799, as indicated by the highest peak at that time. To find how many slaves were being transported per day during that decade:

```
slave_count <- slave_decades[1,2]

print(slave_count/(365.25*10))
```

```
##   n_slaves_decade
## 1        139.6528
```

Approximately 140 slaves were transported every day during the decade of 1700-1799.