JSON Extra Credit

Jean Jimenez

2023-10-22

Introduction

Websites have API that you can use to retrieve data that you find necessary. NobelPrize.org has an API you can interact with to obtain data on Nobel Prizes. For this extra credit assignment, we were asked to come up with four interesting questions that we can answer with the data from the API, using 2 JSON's. After going through the API documentation and familiarizing myself with what type of data was available, I came up with the following four questions to answer:

1. Which country had the most Nobel Prize-winning physicists born there?
2. Which category of Nobel Prize usually has the most people sharing the same award?
3. How many Nobel Laureates died in the same country that they were born?
4. What country has given birth to the highest representation of female Nobel Laureates?

Work and Answer

API Requests and Retrieving JSON DATA

We need to work with two JSON files to answer these four questions so we will be making two API requests to retrieve the JSON data. The website provided in the Extra Credit assignment has an interactive API request builder that you can use to build and test your request for the language of your choice.

First API Request

This first API requests gets all Nobel Prize information from 1901 (901 because in YYY format) to 2023 and returns it in JSON format.

```r
library(httr)

url_prize = "http://api.nobelprize.org/v1/prize.json"

queryString_prize = list(
  year = "2023",
  yearTo = "901"
)

response_prize = VERB("GET", url_prize, query = queryString_prize, content_type("application/octet-strea

prize_txt=content(response_prize, "text")
```

Second API Request

This second API request gets all the Nobel Prize Laureates information. I'll just retrieve it for everyone and filter later. I will also retrieve the country code data.

```
url_laureate = "http://api.nobelprize.org/v1/laureate.json"

response_laureate = VERB("GET", url_laureate, content_type("application/octet-stream"), accept("applica
laureate_txt=content(response_laureate, "text")

url_country = "http://api.nobelprize.org/v1/country.json"

response_country = VERB("GET", url_country, content_type("application/octet-stream"), accept("applicati
country_txt=content(response_country, "text")
```

Converting JSON data to Data Frame

I will use `jsonlite` to parse out the JSON files and place them in data frames.

```
library(jsonlite)
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   3.4.4     v tibble    3.2.1
## v lubridate 1.9.3     v tidyr     1.3.0
## v purrr     1.0.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()  masks stats::filter()
## x purrr::flatten() masks jsonlite::flatten()
## x dplyr::lag()     masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become error
```

```
prize_df=fromJSON(prize_txt, flatten = TRUE)
prize_df=as.data.frame(prize_df)
prize_df=as_tibble(prize_df)

laureate_df=fromJSON(laureate_txt, flatten = TRUE)
laureate_df=as.data.frame(laureate_df)
laureate_df=as_tibble(laureate_df)

country_df=fromJSON(country_txt, flatten = TRUE)
country_df=as.data.frame(country_df)
country_df=as_tibble(country_df)
```

Cleaning Data and Answering Questions

Will clean the data based on each individual question and answer it.

1

Which country had the most Nobel Prize-winning physicists born there?

```
#getting only physics nobel and unnesting/ making long table

physics_nobel=prize_df %>%
  filter(prizes.category=='physics') %>%
  unnest(prizes.laureates)

#getting data from laureate table based on laureate ID and getting count per country

physics_laureates = laureate_df %>%
  filter(laureates.id %in% physics_nobel$id) %>%
  count(laureates.bornCountryCode) %>%
  arrange(desc(n))

#adding proportion
total_physics_laureates = sum(physics_laureates$n)

physics_laureates = physics_laureates %>%
  mutate(proportion = n / total_physics_laureates)

top_5_physics_laureates = physics_laureates %>%
  arrange(desc(n)) %>%
  head(5)

total_top_5_physics_laureates = sum(top_5_physics_laureates$n)
proportion_top_5_over_total = total_top_5_physics_laureates / total_physics_laureates

cat("The top 5 Countries with the most Nobel Laureates born are:", top_5_physics_laureates$laureates.bo
```

```
## The top 5 Countries with the most Nobel Laureates born are: US DE GB FR JP
```

```
cat("The total number of physics laureates from the top 5 countries is:", total_top_5_physics_laureates
```

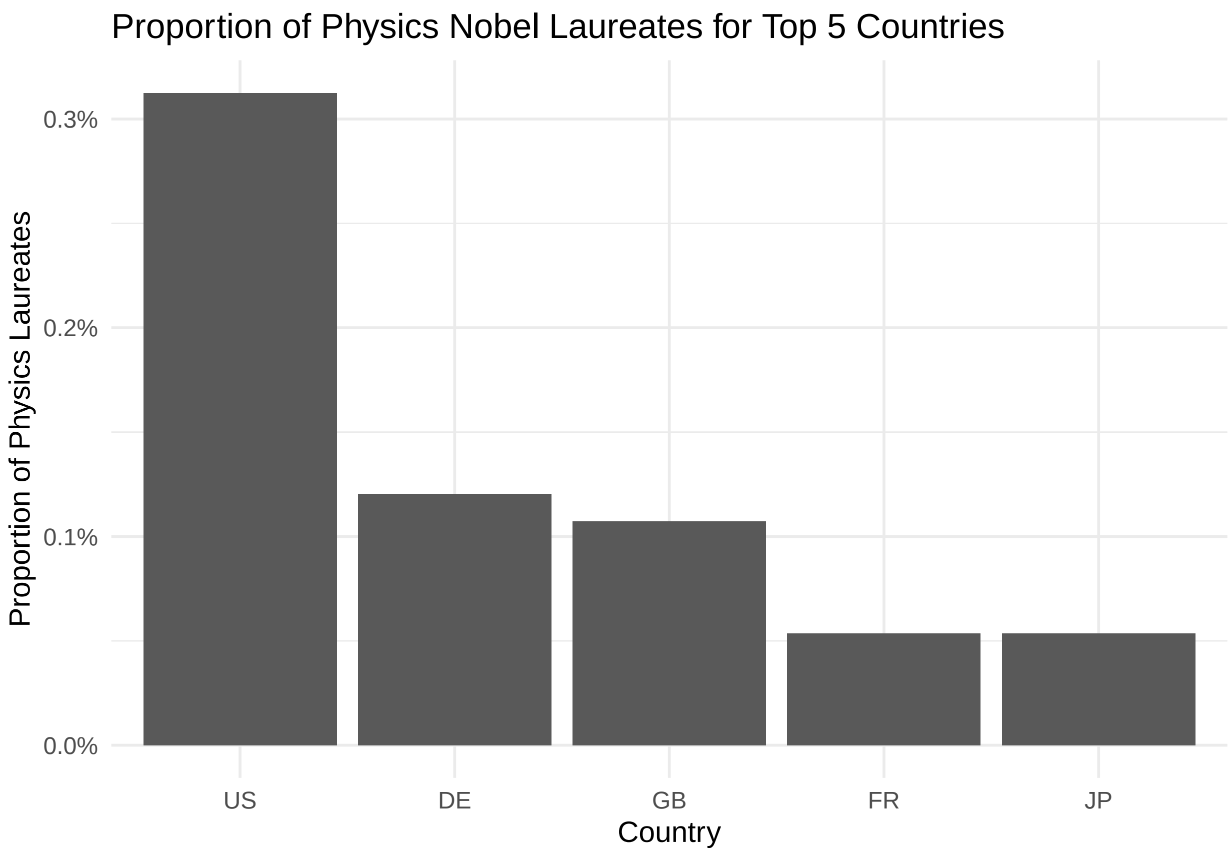
```
## The total number of physics laureates from the top 5 countries is: 145
```

```
cat("The proportion of laureates from the top 5 countries over the total is:", proportion_top_5_over_to
```

```
## The proportion of laureates from the top 5 countries over the total is: 0.6473214 or 64.73214 %
```

```
library(ggplot2)

ggplot(top_5_physics_laureates, aes(x = reorder(laureates.bornCountryCode, -proportion), y = proportion
  geom_bar(stat = "identity") +
  xlab("Country") +
  ylab("Proportion of Physics Laureates") +
  ggtitle("Proportion of Physics Nobel Laureates for Top 5 Countries") +
  scale_y_continuous(labels = scales::percent_format(scale = 1)) +
  theme_minimal()
```



2

Which category of Nobel Prize usually has the most people sharing the same award?

```r
#getting num shares per category per year.
shared_per_yr=prize_df %>%
  unnest(prizes.laureates) %>%
  select(prizes.year, prizes.category, share) %>%
  distinct(prizes.year, prizes.category, .keep_all = TRUE)

average_share_per_category = shared_per_yr %>%
  group_by(prizes.category) %>%
  summarise(average_share = mean(as.numeric(share), na.rm = TRUE)) %>%
  ungroup()%>%
  arrange(desc(average_share))

cat("The category to recive the most average shares is",average_share_per_category$prizes.category[1],"
```

```
## The category to recive the most average shares is medicine with an average of 2.008772 people per aw
```

```r
cat("The category to recive the least average shares is",average_share_per_category$prizes.category[6],
```

```
## The category to recive the least average shares is literature with an average of 1.034483 people per
```

3

How many Nobel Laureates died in the same country that they were born?

```r
#getting birth and death countries and how many  died in same place
total_laureates=nrow(laureate_df)

birth_and_death = laureate_df %>%
  select(laureates.bornCountryCode, laureates.diedCountryCode) %>%
  filter(laureates.bornCountryCode == laureates.diedCountryCode)

count_same_country = nrow(birth_and_death)

cat("The number of laureates who were born and died in the same country is", count_same_country, "laurea
```

```
## The number of laureates who were born and died in the same country is 451 laureates out of 992 laurea
```

4

What country has given birth to the highest representation of female Nobel Laureates?

```r
female_laureates = laureate_df %>%
  select(laureates.bornCountryCode, laureates.gender) %>%
  filter(laureates.gender == 'female') %>%
  count(laureates.bornCountryCode) %>%
  arrange(desc(n))

tot_female=sum(female_laureates$n)
prop_female=((female_laureates$n[1])/(tot_female))*100

cat("The countries that gave birth to the highest number of Female Nobel Laureates is the", female_laur
```

```
## The countries that gave birth to the highest number of Female Nobel Laureates is the US with 17 femal
```