master_script_md

Savannah McNair

2025-04-12

#1. Set up a local Spark server and add the two datasets you used in the first assignment from their GitHub repository. (20%)

#prerequisites had to download java

```
#system("java -version")
#install.packages("sparklyr")
#packageVersion("sparklyr")
library(sparklyr)
```

```
## Warning: package 'sparklyr' was built under R version 4.3.3
```

```
##
## Attaching package: 'sparklyr'
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
#spark_install(version = "3.0")
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.3.3
```

```
##
## 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(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.3.2
```

#connecting - had to change the version to be the most recent compatible version

```r
sc <- spark_connect(master = "local", version = "3.0")
```

#add the two data sets in the first assignment #csv urls

```r
url_lookup <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/UID_
url_confirmed <- "https://raw.githubusercontent.com/CSSEGISandData/COVID-19/master/csse_covid_19_data/cs
```

#filenames to save locally

```r
file_lookup <- "lookup.csv"
file_confirmed <- "confirmed_global.csv"
```

#download the data

```r
download.file(url_lookup, destfile = file_lookup, mode = "wb")
download.file(url_confirmed, destfile = file_confirmed, mode = "wb")
```

#load into spark

```r
lookup_tbl <- spark_read_csv(sc, name = "lookup_tbl", path = file_lookup, header = TRUE, infer_schema =
confirmed_tbl <- spark_read_csv(sc, name = "confirmed_tbl", path = file_confirmed, header = TRUE, infer_
```

#2. In Spark, merge the two datasets, make a smaller version that includes only: Germany, China, Japan, United Kingdom, US, Brazil, Mexico and calculate the number of cases and rate of cases (cases/population) by country and day. Do two graphs and interpret them: change in the number of cases and change in rate by country. (25%)

#reshape the cases dataset to long

```r
#pull into r
confirmed_rdf <- confirmed_tbl %>% collect()
lookup_rdf <- lookup_tbl %>% collect()

#identify date cols
date_cols <- names(confirmed_rdf)[!names(confirmed_rdf) %in% c("ProvinceState", "CountryRegion", "Lat",

#pivot longer, reformat date, new days var
confirmed_long <- confirmed_rdf %>%
  pivot_longer(
    cols = all_of(date_cols),
    names_to = "date",
    values_to = "cases"
  ) %>%
  mutate(
    date = as.Date(date, format = "%m%d%y"),
    days_since_start = as.numeric(date - min(date, na.rm = TRUE))
  )
```

#clean lookup for merge

```
lookup_clean <- lookup_rdf %>%
  rename(
    CountryRegion = Country_Region,
    ProvinceState = Province_State,
    Long = Long_,
    Population = Population
  )
```

#merge datasets by country/region province, lat and long

```
merged_rdf <- confirmed_long %>%
  left_join(lookup_clean, by = c("CountryRegion", "ProvinceState","Lat","Long"))

merged_rdf <- merged_rdf %>%
  mutate(
    case_rate = cases / Population
  )
```

#filter for countries of interest

```
countries <- c("Germany", "China", "Japan", "United Kingdom", "US", "Brazil", "Mexico")

filtered_rdf <- merged_rdf %>%
  filter(`CountryRegion` %in% countries)
```

#push back to spark

```
merged_tbl <- copy_to(sc, filtered_rdf, name = "merged_tbl", overwrite = TRUE)
```

#in spark, create summary table for plotting in r

```
summary_tbl <- merged_tbl %>%
  group_by(CountryRegion, days_since_start) %>%
  summarise(
    total_cases = sum(cases, na.rm = TRUE),
    avg_case_rate = mean(case_rate, na.rm = TRUE)
  )

summary_df <- summary_tbl %>% collect()
```

```
## `summarise()` has grouped output by "CountryRegion". You can override using the
## `.groups` argument.
```

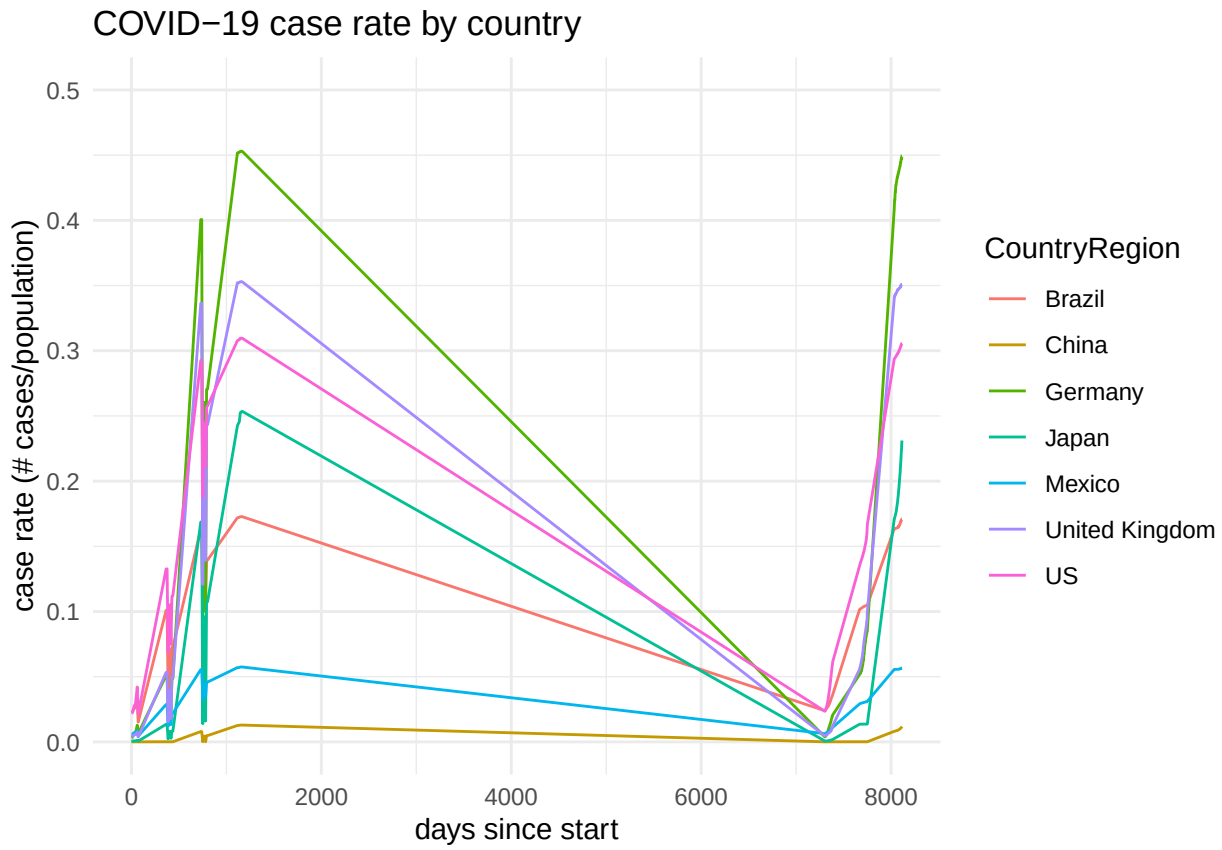
#plotting case rate

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.2
```

```
ggplot(summary_df, aes(x = days_since_start, y = avg_case_rate, color = CountryRegion)) +
  geom_line() +
  scale_y_continuous(
    limits = c(0, 0.5),
    labels = scales::comma
  ) +
  labs(
    y = "case rate (# cases/population)",
    x = "days since start",
    title = "COVID-19 case rate by country"
  ) +
  theme_minimal()
```

```
## Warning: Removed 7 rows containing missing values (`geom_line()`).
```



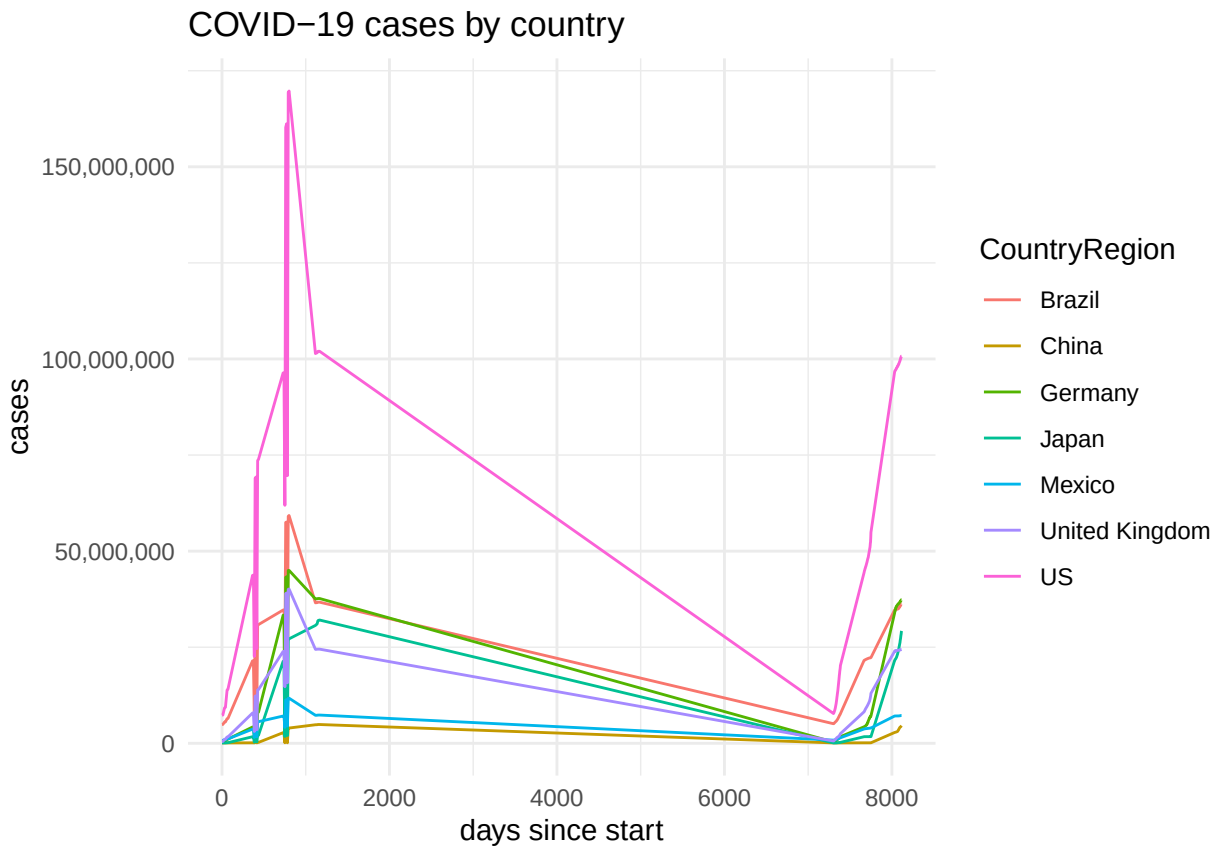
We see from this graph that there is a series of spikes over the course of the first two years, after which the case rates diminish internationally to nearly zero before a second spike. China has consistently low reported case rates per capita, while Germany has consistently high case rates per capita.

#plotting cases

```
summary_df_clean <- summary_df %>%
  filter(!is.na(total_cases) & !is.na(days_since_start)) %>%
  filter(!is.nan(days_since_start)) %>%
  mutate(
    days_since_start = ifelse(days_since_start == 0, 0.1, days_since_start)
```

```
  )

ggplot(summary_df_clean, aes(x = days_since_start, y = total_cases, color = CountryRegion)) +
  geom_line() +
  scale_y_continuous(
    labels = scales::comma
  ) +
  labs(
    y = "cases",
    x = "days since start",
    title = "COVID-19 cases by country"
  ) +
  theme_minimal()
```



This graph tells a different story. We see from this graph that from the start of the pandemic (day 0), there is an initial global spike leading up to around 600 days. From there, there is a gradual decline until about 7000 days and then an additional, smaller spike in cases which has not yet ended. The united states has had the highest number of cases in general throughout the entire pandemic (Pink).

#3. Run a ml_linear_regression explaining the log of number of cases using: country, population size and day since the start of the pandemic. Interpret the results. (25%)

#log transform

```
merged_tbl <- merged_tbl %>%
  mutate(log_cases = log(cases + 1))
```

```r
ML_rdf <- merged_tbl %>% collect()
```

#I really struggled here to be able to run the ML linear regression in Spark because of the categorical nature of country. I tried changing it to a numeric several different ways to be able to use the

cars %>% ml_linear_regression(mpg ~ .) %>% summary(),

#method, but this also didnt work in sparklyr so I did this portion in R

```r
set.seed(42)
train_indices <- sample(1:nrow(ML_rdf), size = 0.7 * nrow(ML_rdf))
training <- ML_rdf[train_indices, ]
test <- ML_rdf[-train_indices, ]

lm_model <- lm(log_cases ~ CountryRegion + Population + days_since_start, data = training)

summary(lm_model)
```

```
##
## Call:
## lm(formula = log_cases ~ CountryRegion + Population + days_since_start,
##     data = training)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.2637  -1.3031  -0.0397   1.4275   9.4119
##
## Coefficients:
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   8.274e+00  3.331e-01  24.842  < 2e-16 ***
## CountryRegionChina           -3.170e+00  2.921e-01 -10.849  < 2e-16 ***
## CountryRegionGermany          3.723e+00  3.277e-01  11.360  < 2e-16 ***
## CountryRegionJapan            1.053e+00  3.097e-01   3.401 0.000675 ***
## CountryRegionMexico           1.505e+00  3.105e-01   4.848 1.27e-06 ***
## CountryRegionUnited Kingdom -2.172e+00  3.253e-01  -6.679 2.57e-11 ***
## CountryRegionUS             -3.543e+00  3.264e-01 -10.856  < 2e-16 ***
## Population                    3.710e-08  1.179e-09  31.467  < 2e-16 ***
## days_since_start              6.108e-05  1.120e-05   5.456 5.02e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.662 on 8197 degrees of freedom
##   (34999 observations deleted due to missingness)
## Multiple R-squared:  0.5326, Adjusted R-squared:  0.5321
## F-statistic:  1168 on 8 and 8197 DF,  p-value: < 2.2e-16
```

This model tries to predict the log # of cases based on country, population, and days since start of the pandemic. With a median of -0.04, this model is reasonably well fitted. There are some prediction errors. 53% of the variance can be explained by the models predictors. The model as a whole is significant! With reference to countries in the model, on average the United States had 3.5 log fewer cases as compared to other countries, while Germany has on average 3.7 log more cases than other countries on a given day, on average. These are the max and min. With reference to population, this coefficient suggests that for every additional person, the log number of cases increases by 0.0000000371. This is statistically significant! Meaning that

larger populations have higher log case numbers. In terms of days, we see that for every day after the start of the pandemic the log # of cases increases by 0.00006108. This would map onto how we understand the progression of pandemics.

#4. Write up everything in an analytic notebook (pdf Rmarkdown) that shows all the syntax you used, the results and walk the reader through the steps of your analysis. (20%)

I will be adding notes to this rmarkdown and publishing to a pdf, which will output to the repository.

#5. Presentation, presentation, presentation: README gives overview of the project and has session info, report text is easy to follow, graphs are easy to understand and properly formatted. (10%)

README has additional notes, session info, and an overview of this project. Report texts walks the user through each step of my code and analyses, with interpretations of the graphs and my ML regression. I updated the scales of all graphs and labelled to ensure these graphs are intuitive and readable.