DATA201 Group Project Report

Yazeed, Chaarvee and Heran – Taking an inside look at AirBNBs

Our research questions:

- Comparing AirBNB prices around the world and if the country's population and GDP per capita has any relation to the AirBNB prices?
- We also wanted to compare the number of bedrooms and bathrooms for each AirBNB and to see if that also has any relation to the price of AirBNBs.

In this project we used 3 different sources.:

- http://insideairbnb.com/get-the-data/ was used to scrape all the AirBNB Data
- https://www.worldometers.info/gdp/gdp-by-country/ was used to scrape countries names, population and GDP per capita
- https://fixer.io/ fixer.io was an API tool we used that had live exchange rate that could help us exchange the pricing of AirBNBs from other currencies to NZD, as the AirBNBs were all listed in their local currency.

Why did we choose these data sources?

Originally, we wanted to scrape https://www.stats.govt.nz/ and wanted to compare things such as population, education, income etc. between major cities in New Zealand. However, we had some struggles scraping this site. After a group meeting we decided to scrape http://insideairbnb.com/get-the-data/ and https://www.worldometers.info/gdp/gdp-by-country/ because we were all interested in this data as a group and felt like it was something unique.

What was the intended use of the data?

Before starting our project, we came up with two research questions. These 2 questions were, if countries population and GDP per capita had any relationship with AirBNB prices. For example, does a country with a higher GDP per capita and population mean that their AirBNB pricing would be more expensive than countries with a lower GDP per capita.

Our second question was if a AirBNB has more bedrooms and bathrooms does that mean the AirBNB prices would increase with the more bathrooms and bedrooms it has?

What difficulties did we encounter during the project?

During the project we did encounter some difficulties. One of the difficulties we encountered was some of the bathroom and bedroom values were showing as NA, unfortunately we couldn't fix this as it was out of our control. We assume these NA values occurred because the users listing their AirBNBs left these sections blank for some reason. The best we could do to overcome this issue was to plot NA values separately one the graphs we created.

Another difficulty we encountered was the pricing of the AirBNBs being in the local currency of the country we scraped. For example, AirBNBs in Tokyo were listed in Japanese Yen. However, we did have a solution for this we used https://fixer.io/ which gave us a private API key that had live exchange rates and helped us convert all the pricing to NZD which made it much easier for us to interpret the graphs and compare the pricing of AirBNBs.

Another issue we encountered was the CSVs contained lots and lots of raw data for example the Tokyo dataset had 46,000 rows which made it a little difficult to graph. However, we did subset some of the data to smaller numbers to make it easier to work with.

The final issue we encountered was a group member not actively participating in the project. However, we felt we managed this well as a group by evenly assigning tasks and making sure the rest of us were all equally participating in the project.

What techniques did we use?

In this project we used many techniques that we were taught in this course. We tried to incorporate as many techniques as possible we were taught. For example, in this project

- We implemented the skills of reading CSVs when reading in our AirBNB data
- Wrangling our AirBNB data into readable tables
- Using our web scraping techniques to scrape the country's names, population and GDP per capita from the world data site
- Interacting with APIs to convert AirBNB pricing to $NZD.

 ``` # Reading in the CSV that contains all the data about Air BNBs in London london_airbnb <- read_csv("http://data.insideairbnb.com/united-kingdom/england/london/2023-09-06/data/listings.csv.gz") ``` ``` Warning message: "One or more parsing issues, call `problems()` on your data frame for details, e.g.: dat <- vroom(...) problems(dat)" Rows: 87946 Columns: 75 ── Column specification ── Delimiter: "," chr (24): listing_url, source, name, description, neighborhood_overview, pi... dbl (37): id, scrape_id, host_id, host_listings_count, host_total_listings_... lgl (9): host_is_superhost, host_has_profile_pic, host_identity_verified, ... date (5): last_scraped, host_since, calendar_last_scraped, first_review, la... 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. ``` ``` # looking into the CSV and viewing the variable names glimpse(london_airbnb) ``` ``` Rows: 87,946 Columns: 75 $ id <dbl> 92644, 93015, 13913, 1540… $ listing_url <chr> "https://www.airbnb.com/r… $ scrape_id <dbl> 2.023091e+13, 2.023091e+1… $ last_scraped <date> 2023-09-06, 2023-09-06, … $ source <chr> "city scrape", "city scra… $ name <chr> "Rental unit in Earlsfiel… $ description <chr> "The space Hi… $ neighborhood_overview <chr> NA, "A bit of history abo… $ picture_url <chr> "https://a0.muscache.com/… $ host_id <dbl> 498381, 499704, 54730, 60 ```	Here is a screenshot that shows an example of a skill we incorporated which is reading in a CSV
`` # wrangling the data and creating a table with the selected variables london_airbnb <- (london_airbnb %>% select(price,bathrooms_text, bedrooms)) london_airbnb ``` A tibble: 87946 × 3 price | bathrooms_text | bedrooms <dbl> | <chr> | <dbl> 42 | 1.5 shared baths | NA 175 | 1 bath | 2 79 | 1 shared bath | NA 150 | 1 bath | 1 46 | 1 shared bath | NA 476 | 2 baths | 3 371 | 2 baths | 2 250 | 1.5 baths | 1 75 | 1 bath | 1 29 | 1.5 shared baths | NA 120 | 1 bath | 2 50 | 1 shared bath | NA 151 | 1 bath | 1 35 | 1 shared bath | NA 140 | 1 private bath | NA	Here is a screenshot that shows an example of a skill we incorporated which is wrangling our data
`` # API key for fixer.io a live exchange rate site so we can convert the price column from GBP to NZD api_key <- "58c8eb71a98bc0e70380918c4ec1041c" ``` ``` # making a GET request to get the exchange rate data for GBP to NZD response <- GET(paste0("http://data.fixer.io/api/latest?access_key=", api_key, "&symbols=GBP,NZD")) ``` ``` # parsing the JSON data to create a data structure and makes it easier to work with data <- fromJSON(content(response, as = "text")) ``` ``` # calculating the exchange of GBP to NZD exchange_rate_gbp_to_nzd <- data$rates$NZD / data$rates$GBP ``` ``` # swapping the previous price column with the new price in NZD london_airbnb$price_nzd <- london_airbnb$price * exchange_rate_gbp_to_nzd ``` ``` # creating the same table with the new price in NZD london_airbnb_nzd <- london_airbnb %>% mutate(price_nzd = price * exchange_rate_gbp_to_nzd) %>% select(price_nzd, bathrooms_text, bedrooms) london_airbnb_nzd ``` A tibble: 87946 × 3 price_nzd | bathrooms_text | bedrooms <dbl> | <chr> | <dbl> 86.91186 | 1.5 shared baths | NA 362.13274 | 1 bath | 2 163.47707 | 1 shared bath | NA 310.39949 | 1 bath | 1 95.18918 | 1 shared bath | NA 985.00106 | 2 baths | 3 767.72141 | 2 baths | 2	Here is a screenshot that shows an example of a skill we incorporated which is interacting with an API to convert the price column from GBP to NZD
`` scrape_and_filter <- function(url) { # Scrape data webpage <- read_html(url) countries <- webpage %>% html_nodes('#example2 a') %>% html_text() population <- webpage %>% html_nodes('td:nth-child(6)') %>% html_text() %>% gsub("[^0-9.]", "", .) %>% as.numeric() gdp_per_capita <- webpage %>% html_nodes('td:nth-child(7)') %>% html_text() %>% gsub("[^0-9.]", "", .) %>% as.numeric() # Convert to data frame data <- data.frame(Country = countries, Population = population, GDP_per_Capita = gdp_per_capita) # Filter for the required countries filtered_data <- data %>% filter(Country %in% c('United Kingdom', 'Japan', 'New Zealand', 'Greece', 'Belgium', 'Netherlands', 'United States', 'Spain', 'Thailand', 'Australia')) return(filtered_data) } # Call the function url <- "https://www.worldometers.info/gdp/gdp-by-country/" result <- scrape_and_filter(url) print(result) ```	Here is a screenshot that shows an example of a skill we incorporated which is creating a function that allows to scrape an entire webpage for country's names, population and GDP per capita.

Another technique we used that helped us work well as a group was setting up a Trello board that could keep us up to date on what we still needed to get done, what is in progress and what we have completed. This

What did we manage to achieve in this project and what we could not achieve?

In this project we managed to scrape AirBNB data for 10 different countries around the world, we also scraped every country's population and GDP per capita. With this data we created various graphs and tables that helped us try to achieve our research questions.

There are some things we did not achieve and if we had more time during the course, we would 100% try incorporating into our project or even something we can add into our project during our free time after the course has ended.

1) Creating interactive plots that make it much easier to read and understand our data. Interactive plots also look so much better than the current plots we have created.
2) Linking in all our notebooks to make it easier for users to run as currently you must run around 12 different notebooks which is quite unprofessional and messy.
3) Creating our own R library with our data so it can be used by the public.