Diary Entry

Ho Wei Ni

2023-11-03

Week 9

What is the topic that you have finalized?

Answer: My finalized topic is on music. I plan to present a data story on the popularity of songs in different regions of the world based on the song attributes (e.g. danceability / energy / key / mode / loudness / speechiness / acousticness / instrumentalness / liveness / valence / tempo), genres and artists etc.

What are the data sources that you have curated so far?

Answer: I will be using this Spotify Dataset found on Kaggle. Since the data set is rather large, I will be using Radiant to filter data extracted during 2022 only. I will also be removing some variables e.g. collab / source / previous rank / pivot, that I have deemed irrelevant to my data story.

Week 10

What is the question that you are going to answer?

Answer: What factors shape the popularity of songs in today's diverse music landscape?

Why is this an important question?

Answer: According Forbes, every society appears to have some form of music, a type of communication that is often overlooked, as part of their culture. From a business perspective, studying the factors behind song popularity aids artists and marketers in understanding audience preferences, which is crucial for creating content that resonates. From the CNM perspective, popular songs mirror cultural sentiments, offering insights into collective mood, social trends, and cultural shifts.

Which rows and columns of the dataset will be used to answer this question?

Answer: My Week 9 Diary Entry mentioned that I am filtering the data to show 2022 results only. However, I have since discovered that the file size is still too large so I will be further filtering the data to show **January 2022** results only.

(Note: The terms 'song' and 'track' are used interchangeably in the following paragraph.)

To begin, I want readers to understand the general music landscape of January 2022. This can be done by listing the tracks and artists listened to in January 2022 in terms of descending order of popularity.

The overall popularity of a track will be determined by calculating the total streams of each track, using the ***streams*** column, taking care to group by ***track_name***. Similarly, the popularity of an artist will be determined by calculating their total streams, taking care to group by ***artist_individual***. Additionally, I will be showing the general distribution of song attributes: ***danceability***, ***energy***, ***key***, ***mode***, ***loudness***, ***speechiness***, ***acousticness***, ***instrumentalness***, ***liveness***, ***valence***, ***tempo***, ***duration***. This indicates the type of songs currently being produced.

To understand the factors that shape the popularity of a song, I will investigate how each song attribute correlate with total streams and chart rankings. Total streams implies the long-term popularity of a song, identifying which attributes are more "timeless" whereas chart rankings implies the short-term popularity of a song that could be of a trend-specific nature. For the latter, ***peak_ranking***, ***previous_week*** and ***weeks_on_chart*** will be used. This analysis will be further refined using ***country*** and ***language*** to compare global and local trends in hopes of revealing similarities and differences.

Challenges & Errors Faced

Embed Shiny into Quarto

Prior to Professor uploading the instructions on how to embed Shiny into Quarto, I was facing difficulties making my Shiny app interactive in the qmd file. Initially, I simply pasted the code in my qmd file, but the output was an image of my Shiny app that is non-interactive.

Thanks to this discussion, I found out that Quarto is a static website and will only run the html files generated, and hence will not run the R code. After trying various methods, I eventually found this tutorial that allows me to embed multiple Shiny apps into my qmd file directly.

I have yet to try the Shinyapps.io method provided by Professor, but have noticed that I will be limited to 5 apps only. While I understand that I can combine my apps such that it appears as a carousel, I am unsure if doing so will disrupt the flow of my data story. I will decide on the method to use after finalising the structure and flow of my story.

Changing the hovertext in my Plotly graph

When playing around with the data, I was trying to display a horizontal bar chart showing the top 10 tracks in January based on their total streams. However, the default hovertext is based on the x and y variable names, making it look messy as there were some redundant information

I came across the tooltip method suggested by cpsievert in this discussion, where I use the text aesthetic specify what information I want to be displayed, before supplying the tooltip text as a character vector, then the tooltip argument in ggplotly(). This will definitely be helpful in making my interactive visualisations neat and hence effective.

Plot distribution of selected attributes on the same histogram based on checkbox selection

Since there are so many song attributes, I wanted to plot them in the same histogram as I thought it looked nice, added to the interactive portion and makes for easier comparison. However, I had no idea how to go about linking the input selection to the data shown.

After lots of research, I found this discussion where the user was attempting to do something similar. Taking inspiration from this line `filter(value %in% as.vector(input$picker)`, I filtered my data set based on the inputs selected and it seems to be working as of now.

However, I am now facing difficulties making the histogram show proportion instead of frequency of count. Showing proportion would make for a fairer comparison as the frequency range for each attribute vary greatly. I will be exploring the solutions to this challenge next week.

Week 11

List the visualizations that you are going to use in your project. How do you plan to make it interactive?

Answer:

- Bar graph to show the top 10 songs & artists using total_streams, grouped_by track_name and artist_individual respectively. This allows readers to get a sensing of what popular songs and artists are like, as they are likely to recognise these songs/artists from their personal experience.
  - The ggplot will be converted to a plotly object, where the tooltip feature will be used to showcase selected texts when the mouse hovers over a specific data point.
- Histogram displaying the distribution of attributes will be incorporated in a Shiny app. This allows readers to get a general understanding of the music landscape.
  - This is made interactive using the check box feature, where users can select which attributes to be plotted on the histogram. A slider input will also be included so that users can choose the number of bins to be displayed.
- Correlation matrix to show the correlation between the song attributes and total_track_streams for long-term popularity, and between the song attributes and change_rank, peak_rank and weeks_on_chart for short-term popularity. Readers can understand which song attributes are more correlated to a song's popularity.
  - Made interactive as users can hover over the boxes to see the co-efficient and which 2 variables are being examined
  - (Subject to change) Incorporate into Shiny app such that users can select the attributes they want to see. Might display a paragraph of analysis to inform users of the top 3 attributes with the highest co-efficient and bottom 3 attributes. Users will not have to make their own analysis.
- (KIV!!!!) Bar graph showing the importance of different song attributes in relation to total_track_streams for long-term popularity. The importance of different song attributes is determined by a random forest model (elaborated on under challenges). The aim is to confirm the findings from the correlation matrix.
  - Made interactive through conversion to a plotly object, where the tooltip feature will be used to showcase selected texts when the mouse hovers over a specific data point.
- (KIV!!!!) World heat map / bubble map to show the average value of a specific song attribute. Readers can compare a certain song attribute across different countries, e.g. average danceability in Argentina is 0.6 while average danceability in Spain is 0.8
  - Made interactive through tooltip function and zoom feature of map
  - (Subject to change) Incorporate into Shiny app so users can pick which attribute they want to analyse

What concepts incorporated in your project were taught in the course and which ones were self-learnt?

Answer:

Table 1: Concepts incorporated

Topic	Week
Data Cleaning	Week 3 - explicit coercion: when manipulating data Week 4 - mutate to add new columns, select & filter to create data subsets Week 9 - pivot_longer, pivot_wider to reshape data Computing correlation matrix
Data Visualisation	Week 2 & Week 7 - ggplot2: basics of plotting histograms & bar charts Week 8 - Shiny exploration Plotly: converts ggplot objects into interactive plotly objects Heat maps
Data Prediction	Random Forest Model: A machine-learning algorithm that **combines the output of multiple decision trees to reach a single result**, can essentially make predictions • Look at "Test" tab for examples of output

Challenges & Errors Faced

Making a carousel for the interactive plotly plots

In my research, I found the slickR package meant for making carousels. However, when I used the package for my plotly objects, they remained as static images. As such, I stuck to using stacking my plots above each other, using subplot.

Unable to navigate to clickable link in my hover text

Currently, I have input the link to the song in the hover text. However, the hover text disappears when my mouse navigates over to the link, as the mouse no longer hovers over the bar. While it is not necessary to include the song link, I thought it would be interesting to do so and increase interactivity.

One possible alternative I have found is by making my bar chart clickable through the girafe package. While the code is working in R console, the bar chart does not appear when I render the qmd file. I think the issue is that the output is a htmlwidget, and according to this discussion, there seems to be an ongoing error where Quarto fails to render htmlwidgets when running knitr.

Another alternative is by delaying the disappearance of the hover text such that users have the time to navigate over to the clickable link in my tooltip. However, doing so seems to require custom JavaScript. Thus, I've yet to decide the solution to this challenge.

Shiny graph is taking a very long time to load when Quarto is rendered

My Shiny app works completely fine and fast when I run the R script directly in the local host. However, when rendered in my Quarto website, the graph does not appear even after very long. This could be due to the method I've chosen to employ in embedding Shiny onto my webpage. I will be trying the Professor's method to see if the same error occurs.

How to show the relationship between different song attributes and song popularity

This is more of a math problem rather than a coding problem. I'm not sure what calculations I can do to showcase the above relationship in answering my overall question. The only way I can think of is by calculating the correlation between each song attribute and track_total_streams. A higher correlation would indicate that the attribute has more influence in shaping the a song's popularity.

Due to the whole “correlation does not equate causation” thing, I plan to substantiate my findings with the Random Forest Model that is supposed to predict the importance of different song attributes based on my defined variable (track_total_streams for e.g.).

Assuming that the above logic is correct, one challenge I faced was converting the output of the model to a data frame for easier visualisation. Turns out, I can just do explicit coercion using the as.data.frame method. I also had to do some research on converting the row names to a new column.