DATA201/422 Group Project

31/10/2022

Authors:

Anna McCarthy - (amc452) - 13855141

Thomas Arnold - (tar80) - 94702392

Zheng Chao - (zch65) - 21671773

Ezeckiel Mautoatasi - (ema161) - 51262582

Introduction:

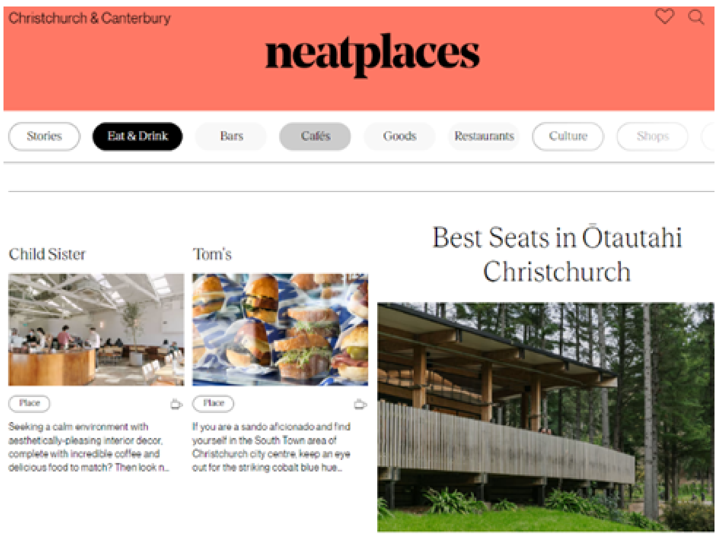
This report aims to demonstrate our group's process from initial design to final proposal. It contains interesting data sources that we identify to process, organise them into a suitable target data model and present this data to a wider audience.

Our team came up with a number of different ideas during the prototyping process, some of the initial topic ideas included looking into the number of reported cases of rule breakers in Christchurch region during the lockdown. Also, we looked at comparing the number of road works in the Christchurch area over time. Due to difficulties with obtaining data of some of these topic areas. After looking for available data sets, we decided to make playlists of NZ music for Christchurch cafes.

By taking the names of cafés in Christchurch and splitting their names into characters, we matched each character with a New Zealand artist whose name starts with the character and selected one of their albums to go into the custom playlist. To obtain the required information we scraped several websites to get data about the cafes around Christchurch as well as information about New Zealand artists and their songs.

The Café Names Data Source

The first data source that we needed was a list of Christchurch café names, we had some difficulties finding a website that stored the information inside a class with no other information inside of it. Neatplaces is a website that stores information about all shops, bars, cafes, and restaurants in New Zealand. We were able to scrape the café names easily using the read_html() function because the names were in a class that had nothing else inside of it.

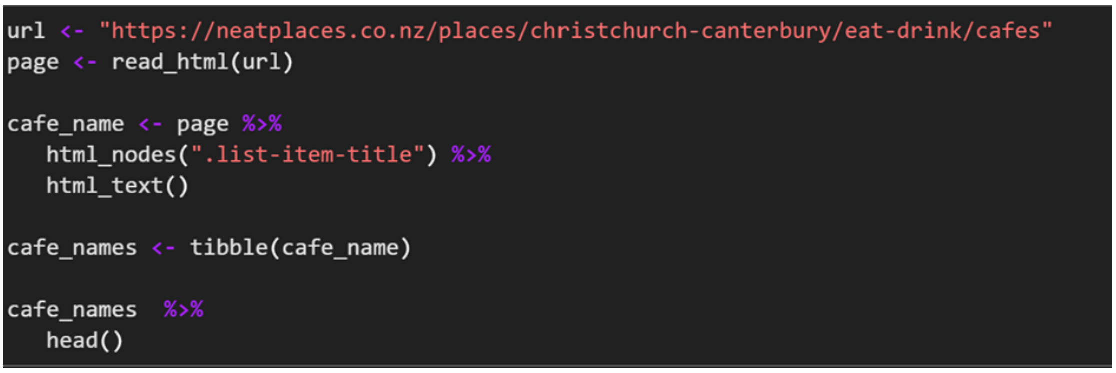


Seeking a calm environment with aesthetically-pleasing interior decor, complete with incredible coffee and delicious food to match? Then look n...

If you are a sando aficionado and find yourself in the South Town area of Christchurch city centre, keep an eye out for the striking cobalt blue hue...

Figure 1: Webpage 'Neatplaces', from which the Christchurch Cafe's Database was acquired. Link to website: https://neatplaces.co.nz/places/christchurch-canterbury/eat-drink/cafes.

Because we spent the time researching different websites to find the easiest one, the code to scrape the café names was simple.



```r
url <- "https://neatplaces.co.nz/places/christchurch-canterbury/eat-drink/cafes"
page <- read_html(url)

cafe_name <- page %>%
   html_nodes(".list-item-title") %>%
   html_text()

cafe_names <- tibble(cafe_name)

cafe_names  %>%
   head()
```

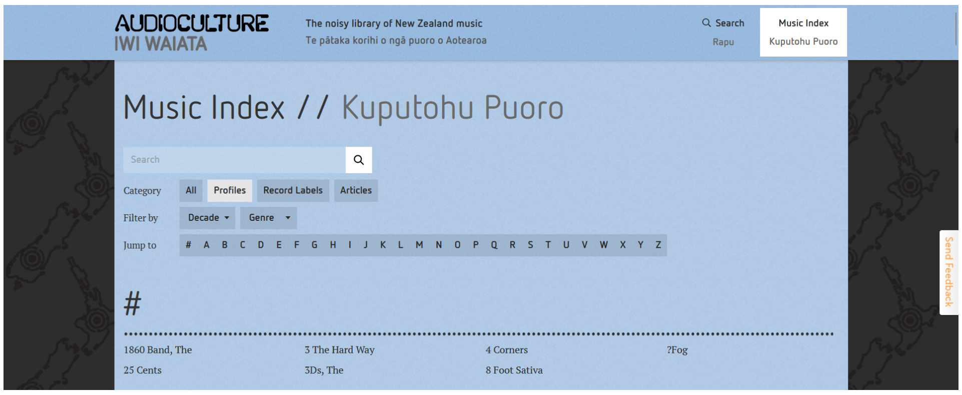
Figure 2: Fragment of R code which webscraped the Cafe names from the Neatplaces website.

As well as making playlists for Christchurch cafés we created a function for other developers to input other New Zealand cities. Developers can input any one of these cites into our function which will give a list of cafes names in that area.

The Music Data Source

The second data source that we needed information about was New Zealand artists. We wanted New Zealand artists in particular as it's our home and we wanted to do something to help us appreciate that. We found our data on a website called AudioCulture which contained many of New Zealand's artists and information about their music.

AudioCulture website: https://www.audioculture.co.nz/music_index?category=Person


AUDIOCULTURE
IWI WAIATA
The noisy library of New Zealand music
Te pātaka korihi o ngā puoro o Aotearoa
Search
Rapu
Music Index
Kuputohu Puoro
Music Index // Kuputohu Puoro
Search
Category
All
Profiles
Record Labels
Articles
Filter by
Decade
Genre
Jump to
# A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
Send Feedback

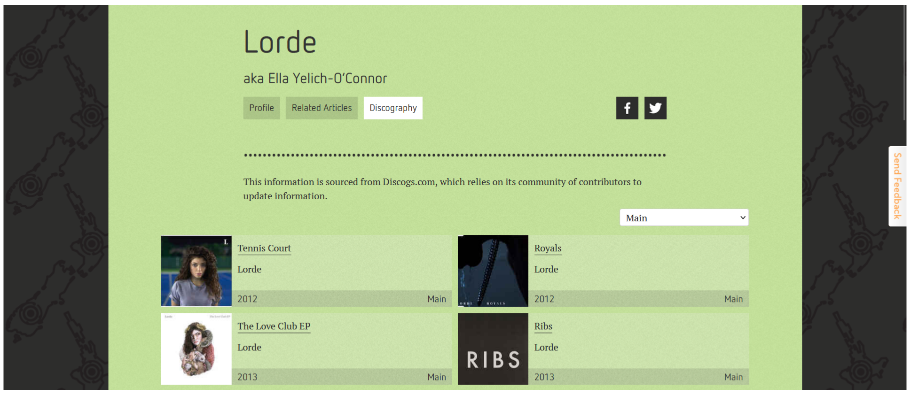
#

1860 Band, The	3 The Hard Way	4 Corners	?Fog
25 Cents	3Ds, The	8 Foot Sativa	

Figure 3: AudioCulture music index search

We chose this website because of its previously stated vast library of New Zealand artists and also because its formatted neatly which was helpful for web scraping. The intended use of the dataset was to find New Zealand music starting with a given character so that in the end, we would have a playlist of music for each character in a Christchurch cafe name.

One of the challenges we faced was that the profile pages didn't include all the songs by a profile, but their albums (image below) or sometimes nothing at all. This made it difficult to find songs to add to the playlist.



This information is sourced from Discogs.com, which relies on its community of contributors to update information.

Tennis Court	Lorde	2012	Main
Royals	Lorde	2012	Main
The Love Club EP	Lorde	2013	Main
Ribs	Lorde	2013	Main

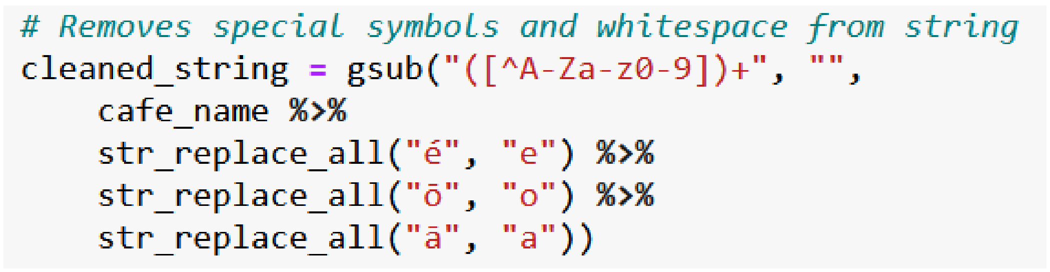
Figure 4: Lorde discography page

The original plan was to take a sample of profiles and take one song from their discography. Now instead we take an album, if one exists, from a profile. This was actually a better idea as before choosing one song per character meant that the music playlists would be much too short.

The main function used to wrangle the dataset is “create_profile_list”. This function takes a string of characters, the Christchurch cafe name, as an input and returns a dataframe of New Zealand profile albums based on each character of the given string. It does this with the assistance of the helper function get_discography. This function takes a string of characters, the name of a profile, as an input and returns the discography of said profile in the form of a dataframe.

When it came to wrangling most of the challenge was around searching for an album beginning with a given character. This was because we had to learn to use unfamiliar techniques such as “gsub”, “grep”, and REGEX. Thankfully, there were many R documentations on the internet that went into detail on how functions worked. There were also StackOverflow forums that had similar questions regarding the use of these functions followed by user submitted explanations. This made the challenge easier to understand and solve.

As previously stated, REGEX was one of the techniques we utilised for our code. REGEX allowed us to compare variables to a certain pattern to see if it was a match. One use was to remove special characters from a profile name for easier searching/filtering. This was done using the “gsub” function which allowed us to replace characters in a given string using a given REGEX pattern. The image below demonstrates how the function worked. The REGEX pattern “[^A-Za-z0-9]” means anything that’s not uppercase or lowercase characters or an integer from 0-9. This tells the gsub function to substitute anything that doesn’t match that with nothing. In the “gsub” args, we also use “str_replace_all”. This just replaces specified characters with other characters(s). This is to keep characters important to the cafe name such as accented letters. In this example, cafe_name = “Vick’s Café” will become cleaned_string = “Vicks Cafe.



```r
# Removes special symbols and whitespace from string
cleaned_string = gsub("([^A-Za-z0-9])+", "",
    cafe_name %>%
    str_replace_all("é", "e") %>%
    str_replace_all("ō", "o") %>%
    str_replace_all("ā", "a"))
```

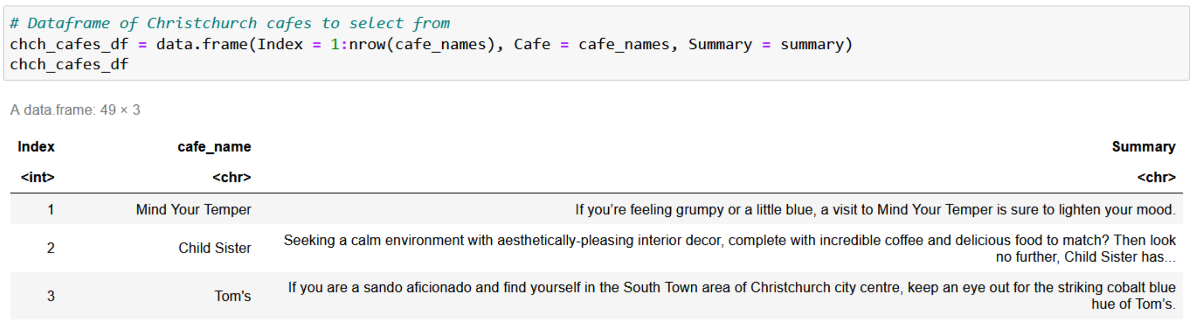
Figure 5: Code snippet of the string cleaning process

Another use of REGEX was alongside the “grep” function. The “grep” function was what searched for the profile names that began with a certain character. The function took in a REGEX pattern formatted using the glue function to find a character at the beginning of a string that contained a given letter (uppercase or lowercase). The glue function is there to concatenate the uppercase and lowercase version of the first character in the given string. For example, if the string was Vick's cafe then the REGEX pattern would look like "^[Vv]". The "^" left of the brackets looks at the first character and "Vv" means look for uppercase or lowercase V. In the image below, the grep function also takes in "Profiles", which is a list of the profile names, and "value = TRUE" meaning return the element and not the element index.

```r
# Creates list of profiles that match REGEX pattern (Starts with given char)
match_list = grep(glue('^[{upper_char}{lower_char}]'), Profiles, value = TRUE)
```

Figure 6: Code snippet of the grep() function in use

In the end, we managed to do what we set out to do with this website. We were able to scrape it and create a dataframe of New Zealand albums for each character in a Christchurch cafe name. The user will be shown a list of cafes in Christchurch, each with a unique index. The user will select an index for their desired cafe and enter it. See images below.

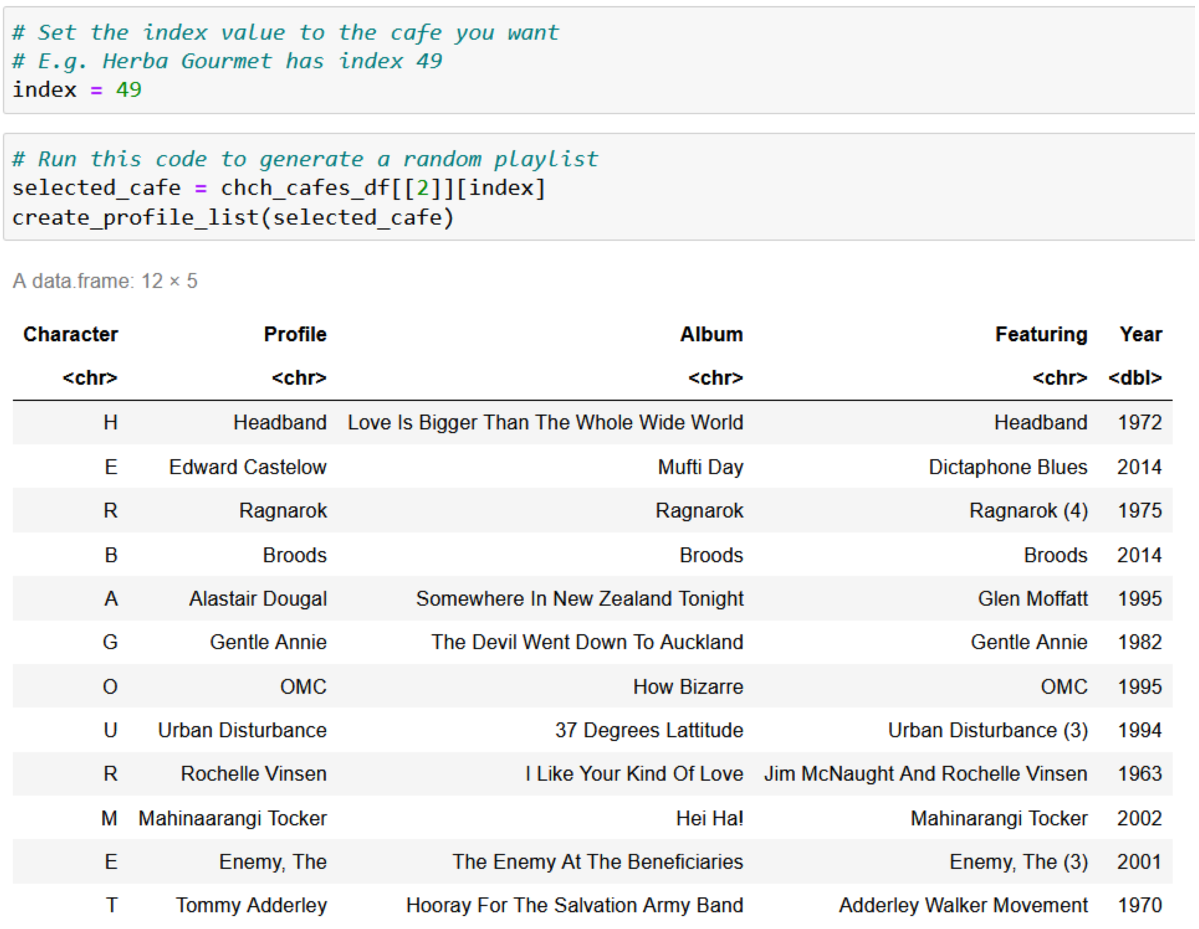


```r
# Dataframe of Christchurch cafes to select from
chch_cafes_df = data.frame(Index = 1:nrow(cafe_names), Cafe = cafe_names, Summary = summary)
chch_cafes_df
```

A data.frame: 49 × 3

Index	cafe_name	Summary
<int>	<chr>	<chr>
1	Mind Your Temper	If you're feeling grumpy or a little blue, a visit to Mind Your Temper is sure to lighten your mood.
2	Child Sister	Seeking a calm environment with aesthetically-pleasing interior decor, complete with incredible coffee and delicious food to match? Then look no further, Child Sister has...
3	Tom's	If you are a sando aficionado and find yourself in the South Town area of Christchurch city centre, keep an eye out for the striking cobalt blue hue of Tom's.

Figure 7: First 3 rows of the Christchurch cafe dataframe



```
# Set the index value to the cafe you want
# E.g. Herba Gourmet has index 49
index = 49
```

```
# Run this code to generate a random playlist
selected_cafe = chch_cafes_df[[2]][index]
create_profile_list(selected_cafe)
```

A data.frame: 12 × 5

Character	Profile	Album	Featuring	Year
<chr>	<chr>	<chr>	<chr>	<dbl>
H	Headband	Love Is Bigger Than The Whole Wide World	Headband	1972
E	Edward Castelow	Mufti Day	Dictaphone Blues	2014
R	Ragnarok	Ragnarok	Ragnarok (4)	1975
B	Broods	Broods	Broods	2014
A	Alastair Dougal	Somewhere In New Zealand Tonight	Glen Moffatt	1995
G	Gentle Annie	The Devil Went Down To Auckland	Gentle Annie	1982
O	OMC	How Bizarre	OMC	1995
U	Urban Disturbance	37 Degrees Lattitude	Urban Disturbance (3)	1994
R	Rochelle Vinsen	I Like Your Kind Of Love	Jim McNaught And Rochelle Vinsen	1963
M	Mahinaarangi Tocker	Hei Ha!	Mahinarangi Tocker	2002
E	Enemy, The	The Enemy At The Beneficiaries	Enemy, The (3)	2001
T	Tommy Adderley	Hooray For The Salvation Army Band	Adderley Walker Movement	1970

Figure 8: The generated New Zealand music playlist of albums

SpotifyR API

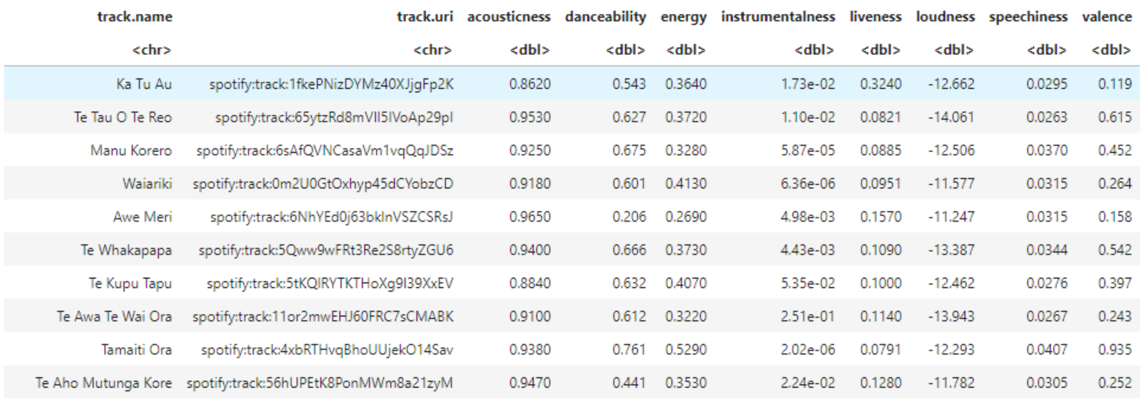
The next stage in this project was the conversion of the playlists generated for the cafes from a dataframe format in the R code, into a usable playlist available on a streaming service such as Youtube or Spotify. After thorough research, no method for this type of conversion was found. This could have been completed had the music database included external links to streaming sources for each track. However, a Spotify API wrapper package, Spotifyr on GitHub, was utilised to acquire information about the tracks within a tester playlist. A playlist was generated for the cafe "Riverside Cafe", which was then manually turned into a playlist on Spotify, under the premium account of one of the group members. Cafe's commonly purchase some form of premium streaming subscription to enable them to play music in their establishment without infringing copyright laws.

Making use of the Spotifyr package, many features of the tracks within the playlist can be analysed. Some of these features are displayed in the Table below, gathered from the Spotify for Developers documentation (Link: https://developer.spotify.com/documentation/web-api/reference/#/operations/get-audio-features).

Feature	Units	Description
Acousticness	<float> 0.0 to 1.0	A confidence measure from 0.0 to 1.0 of whether the track is acoustic. 1.0 represents high confidence the track is acoustic.
Danceability	<float> 0.0 to 1.0	Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. A value of 0.0 is least danceable and 1.0 is most danceable.
Energy	<float> 0.0 to 1.0	Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity. Typically, energetic tracks feel fast, loud, and noisy. For example, death metal has high energy, while a Bach prelude scores low on the scale. Perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy.
Instrumentalness	<float> 0.0 to 1.0	Predicts whether a track contains no vocals. "Ooh" and "aah" sounds are treated as instrumental in this context. Rap or spoken word tracks are clearly "vocal". The closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content. Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0.
Liveness	<float> 0.0 to 1.0	Detects the presence of an audience in the recording. Higher liveness values represent an increased probability that the track was performed live. A value above 0.8 provides strong likelihood that the track is live.

Loudness	<float> decibels (dB)	The overall loudness of a track in decibels (dB). Loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks. Loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude). Values typically range between -60 and 0 db.
Speechiness	<float> 0.0 to 1.0	Speechiness detects the presence of spoken words in a track. The more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value. Values above 0.66 describe tracks that are probably made entirely of spoken words. Values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music. Values below 0.33 most likely represent music and other non-speech-like tracks.
Valence	<float> 0.0 to 1.0	A measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track. Tracks with high valence sound more positive (e.g. happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g. sad, depressed, angry).

The code used to acquire these features for the playlist generated for 'Riverside Cafe' will not be shown as it makes use of private client information regarding the group members' Spotify account used. However, a sample output of the code, displaying the features of each track of the playlist, is displayed below.



track.name	track.uri	acousticness	danceability	energy	instrumentalness	liveness	loudness	speechiness	valence
<chr>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Ka Tu Au	spotify:track:1fkePNizDYMz40XJjgFp2K	0.8620	0.543	0.3640	1.73e-02	0.3240	-12.662	0.0295	0.119
Te Tau O Te Reo	spotify:track:65ytzRd8mVII5IVoAp29pl	0.9530	0.627	0.3720	1.10e-02	0.0821	-14.061	0.0263	0.615
Manu Korero	spotify:track:6sAfQVNCasaVm1vqQqJDSz	0.9250	0.675	0.3280	5.87e-05	0.0885	-12.506	0.0370	0.452
Waiariki	spotify:track:0m2U0GtOxhyp45dCYobzCD	0.9180	0.601	0.4130	6.36e-06	0.0951	-11.577	0.0315	0.264
Awe Meri	spotify:track:6NhYEd0j63bkInVSZCSRsJ	0.9650	0.206	0.2690	4.98e-03	0.1570	-11.247	0.0315	0.158
Te Whakapapa	spotify:track:5Qww9wFRt3Re2S8rtyZGU6	0.9400	0.666	0.3730	4.43e-03	0.1090	-13.387	0.0344	0.542
Te Kupu Tapu	spotify:track:5tKQIRYTKTHoXg9I39XxEV	0.8840	0.632	0.4070	5.35e-02	0.1000	-12.462	0.0276	0.397
Te Awa Te Wai Ora	spotify:track:11or2mwEHJ60FRC7sCMABK	0.9100	0.612	0.3220	2.51e-01	0.1140	-13.943	0.0267	0.243
Tamaiti Ora	spotify:track:4xbRTHvqBhoUUjekO14Sav	0.9380	0.761	0.5290	2.02e-06	0.0791	-12.293	0.0407	0.935
Te Aho Mutunga Kore	spotify:track:56hUPEtK8PonMWm8a21zyM	0.9470	0.441	0.3530	2.24e-02	0.1280	-11.782	0.0305	0.252

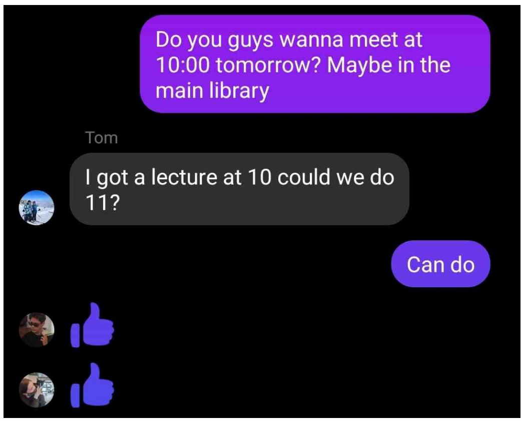
Figure 9: Sample output of the Spotifyr analysis of test Cafe playlist, generated for "Riverside Cafe".

The analysis provided by the Spotifyr package would enable users of our playlists, namely the owners and employees of Christchurch Cafes, to filter out music based on their preferences and aesthetics of their specific Cafe environment. For example, a country style cafe may not want particularly loud and speechy music (such as rap) for their Cafe's vibe.

While the use of the Spotify API was not implemented as we first intended, we were able to find ways to incorporate it into our project and allow it to still be a useful resource to aid our efforts.

Collaboration

To help us collaborate with each other, we created a Facebook group chat (first image) to make communication simple. This allowed us to communicate ideas or plan meetings before coming to the labs. We also created a GitHub group, see second image below.



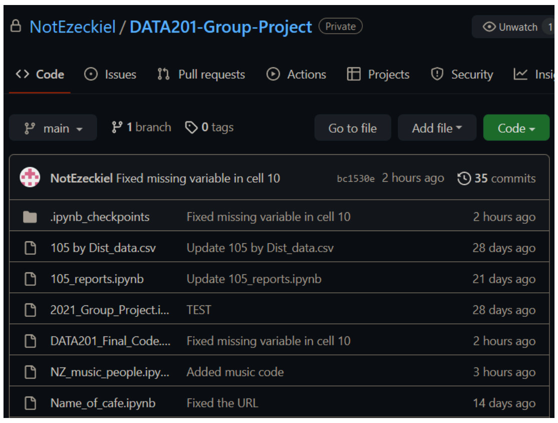
Do you guys wanna meet at 10:00 tomorrow? Maybe in the main library

Tom

I got a lecture at 10 could we do 11?

Can do

Figure 9: Our Facebook group chat



NotEzeckiel / DATA201-Group-Project Private

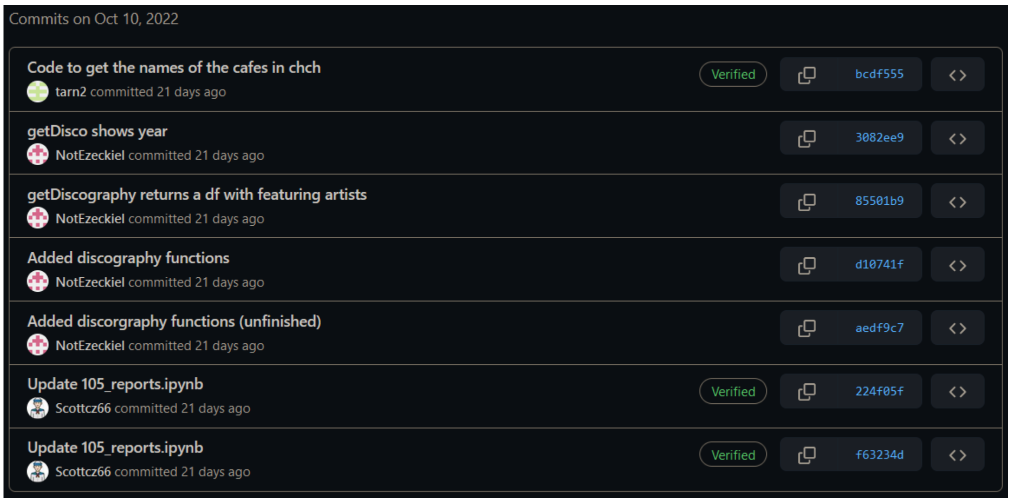
NotEzeckiel Fixed missing variable in cell 10 bc1530e 2 hours ago 35 commits

.ipynb_checkpoints	Fixed missing variable in cell 10	2 hours ago
105 by Dist_data.csv	Update 105 by Dist_data.csv	28 days ago
105_reports.ipynb	Update 105_reports.ipynb	21 days ago
2021_Group_Project.i...	TEST	28 days ago
DATA201_Final_Code....	Fixed missing variable in cell 10	2 hours ago
NZ_music_people.ipy...	Added music code	3 hours ago
Name_of_cafe.ipynb	Fixed the URL	14 days ago

Figure 10: GitHub main page

GitHub allowed us to share and collaborate with each other's code. It also allowed us to see what parts of our code were updated (see image below) or work on our code from different computers.

Our Github link is https://github.com/NotEzeckiel/DATA201-Group-Project.



Commits on Oct 10, 2022

Code to get the names of the cafes in chch
tarn2 committed 21 days ago — Verified — bcdf555

getDisco shows year
NotEzeckiel committed 21 days ago — 3082ee9

getDiscography returns a df with featuring artists
NotEzeckiel committed 21 days ago — 85501b9

Added discography functions
NotEzeckiel committed 21 days ago — d10741f

Added discorgraphy functions (unfinished)
NotEzeckiel committed 21 days ago — aedf9c7

Update 105_reports.ipynb
Scottcz66 committed 21 days ago — Verified — 224f05f

Update 105_reports.ipynb
Scottcz66 committed 21 days ago — Verified — f63234d

Figure 11: GitHub commits (code updates) page

Conclusion

The aim of this project was to generate a music playlist of New Zealand artists\bands based on Christchurch cafe names. This playlist would be used by the Christchurch cafe owners to have music by our people playing as we drink coffee. This was done using the programming skills we learnt in the labs and the new techniques we taught ourselves. We believe we were successful in satisfying our project aim and are happy with the outcome.