Digital Analytics – Individual Project + Report (51%)

Carefully read this entire document
Updated 27 April 2021

Objective

The goal of the individual project is to combine all the research and analytical skills you acquired throughout the five training modules. You are tasked with completing an entire project from research question/hypothesis to reporting. This means you will independently collect, process, analyse, and visualise data to answer a research question. You will write Python code to accomplish this and report on the entire process in a written document of 1,500 to 2,000 words.

Independence is key. Apart from the set research question and a handful of requirements, there is no ready-made template to work from. That is the whole point of this project. When you work as a professional analyst, you will only get an overview of a problem and maybe some requirements. It is your responsibility to deliver, to make decisions and to come up with solutions.

You will inevitably get stuck a few times. Do not let it discourage you. That is just the way it goes with any project. Plan ahead, sensibly combine the skills that you acquired throughout the course, and critically weigh the benefits and limitations. There is not one single approach.

The problem

It is often said that popular music increasingly homogenises. It all ‘sounds the same’ in favour of synthetic dance music. In your project, you will put this hypothesis to the test by comparing the top hits in of the past 50 years for their audio features (1970-2019).

You need to mould this research problem into a testable hypothesis.

Format

The final product of your research project is a Blackboard assignment submission (each file separately, **not** in a consolidated archive such as a .zip or .rar) that contains:

All Python files

These are the Python files you used to collect, process, analyse, and visualise data. The filenames should contain numbers that make clear in what sequence they should be run to replicate your project (e.g., datacollection_1.py, datacollection_2.py, dataprocessing_3.py, dataanalysis_4.py, datavisualisation_5.py). There is no expectation of a set number of files, but they should cover your entire project and be labelled chronologically.

There is no need to submit any image files and/or data files. Running your Python code should (re)produce these.

A written report

1. **Introduction.** This report starts with a concise introduction of the research problem (about 300 words), which leads to the formulation of a testable hypothesis. You are encouraged to support this introduction with academic references (APA-style).
2. **Method section.** In this section you explain the steps you took, and you explain the decisions you made in collection and processing the data prior to analysis. You explain where you got what data, why you need the data that you are collecting, how they are combined and cleaned, and what they look like (i.e., sample size, variables in the dataset, and what these variables represent). Do not assume that your reader has the courage to sift through any of your Python code.

   Important:
   - The method section does not include any Python code. None at all. You write about what you did in an overviewing *narrative*. You should enable your reader to understand the steps you took, but you do not want to overwhelm with minute detail (the code is in appendix anyway).
   - Do not use abstract variable names in your method section. Name or explain your variables in plain English so everyone can understand what they mean. Again, your text should be accessible, even for a slightly less motivated reader. If it becomes hard work to read your report, your reader is likely to tune out.

3. **Results section.** Explain the steps in taken in your data analysis and sketch out the results. What did you do with the data, and what are the results? You need to explain what technique you use to establish what insight. Sensibly combine textual description, tables, and/or data visualisations. Explain what the data tell you, but do not yet start discussing what the results exactly mean.

   Important:
   - Carefully format your tables and visualisations. Do not screenshot outputs from the console. That just looks horrible and reflects poorly on your work, not doing it any justice. Format your own ‘clean and lean’ tables whenever needed.
   - Avoid (excessive) redundancy. There is no need to include the same results in textual AND tabular AND visual form. That is overkill and just confuses your reader. It makes you look indecisive. Make decisions on what presentation form communicates your results the clearest. A good approach is to visualise and tabulate results, and describe the key findings in the results text.

4. **Discussion section.** You interpret the results in light of the hypothesis and emphasize what this means in light of the research problem. What did you learn from the data? What does it potentially mean for the literature in the introduction? Make sure you also

critically assess the strengths and weaknesses of the method you used. Nothing is perfect and choices have consequences. What are they in your case?

5. **Appendix section** with a chronological overview of your code files. You do not only have to submit your .py files, you also need to copy/paste your code into the appendix section of your paper in chronological order (first script first, last script last).

Please consider these formatting requirements:

- Use Times New Roman 11pt and 1.15 spacing.
- Title page contains the title of your project, your name, your student number. Do not include any visuals and/or logo's.
- Write at least 1,500 and maximum 2,000 words (references, tables, figures are excluded from the word count). This might not seem a lot, but it is plenty for what you need to write, and it will keep you focused. Make every word count.
- Tables and figures are included in the text and should be captioned.

Most important is that your lay-out is tidy and looks like a professional research report. Avoid frivolities, keep everything nice and clean.

Assessment criteria

This project makes up 51% of your final grade. Since we have 7 graded exercises of equal weight (7%), that should roughly make up half of the grade, the remaining 1% needed to go somewhere. That explains the somewhat 'strange' number.

The marking rubric:

Criterion category	Description	Weight	Label			
			Fail	Pass	Credit	Distinction
			Work is of poor quality and has multiple fundamental shortcomings	Work is of sufficient quality, it has no fundamental shortcomings but there are multiple minor shortcomings	Work is of more than sufficient quality, with only rare minor shortcomings	Work is of outstanding quality, nearing perfection: there are no shortcomings whatsoever
Introduction: contextualising research questions	A clear identifiable, sound research question or a testable hypothesis that is contextualised in literature: why is it relevant to research, how does it fit in with literature?	10%	0	1	2	3
Method: transparency in procedure	A clear narative that aplty explains the choices that have been been and the steps that have been taken in collecting and processing data. It describes the proces and the data (i.e., its dimensions and distributions)	20%	0	1	2	3
Method: efficacy in processing and analyzing data	Soundness of the Python scripts used for collecting, processing, analysing and visualising data - Are there no (structural) shortcomings/mistakes in the procedure/coding?	20%	0	1	2	3
Results: transparency in reporting results (textually and visually)	Are the results clearly communicated in an appropriate form?	20%	0	1	2	3
Discussion: quality of discussion	Are the results correctly interpreted and actively linked with the introduction?	15%	0	1	2	3
Discussion: quality of applied method	Is there a thorough discussion of strengths and weaknesses of the method?	10%	0	1	2	3
Overall formatting	Is the text carefully formatted (including figures and tables)?	5%	0	1	2	3

Timing

The deadline of this assessment is set on 28 May 2021 at 4pm Brisbane time.

You are encouraged to start early on the project. You can take a head start by writing the introduction. Upon completion of each module, you can advance in your project as well (see tips and tricks for concrete pointers on what module you need for what step). Ideally, you start with the scraping job once you feel confident with the Module 3 materials. The same goes for the further data enrichment through the Spotify API. You can do that as soon as Module 4 is finished. There will be ample time after we have concluded Module 5 for you to do the analysis and the write up of your process/results/discussion.

The last three weeks of the course are fully dedicated to the project. The regular tutorial slots in the last three weeks are fully reserved for Q&As on the project. Questions on any part of the project are welcomed during these tutorial slots.

Tips and tricks

To get you started, this document gives some valuable pointers for the different steps of the research process.

Data collection

- You need data on what music was the most popular in the past 50 years (1970 until - so not including - 2020). What better data source than the official charts? We can use https://www.billboard.com/ to get data. This will be a scraping job, so you can use the resources in Module 3. Billboard has an end-of-year top 100 for every year (although some years have missing values at the source) e.g., https://www.billboard.com/charts/year-end/1970/hot-100-songs. If you change the year in that URL, it will lead you to the info of that year, e.g., https://www.billboard.com/charts/year-end/1971/hot-100-songs. There is a system in the URLs, which makes it easy to automatically request the html of page after page. Per year, you can scrape the artist and track titles. Store this in a csv file with three variables: year, artist, and track. Billboard.com is sensitive to rapid consecutive requests, so make sure you pause your script for about 2-3 seconds at the end of each iteration using the Python pause module. Also make sure to 'sanity check' your scaping script: are you getting all the information that is available?

IMPORTANT ADDENDUM 27 April 2021: Issue with billboard.com

The Billboard website causes a critical issue: for the years 1991-2005 it returns the 2006 data. We do not want this to invalidate the outcomes of the project. What you are supposed to do:

- Scrape the data from Billboard as if there is no issue. This means you write a scraper script and you will collect data as if nothing is wrong. I will assess the logic of your script as part of the project. This implies no change to your project at all.
- Instead of using the data you scraped from Billboard, use the dataset provided: http://www.digitalanalytics.id.au/static/files/chartdata.csv. The missing years were scraped from Wikipedia that quotes Billboard.com as its source. You should use this dataset to further enrich it through the Spotify API. Do not use the data you scraped from Billboard yourself.

- Once you have a dataset with the most popular hits of the past five decades, you need to get their audio features. The Spotify API provides a brilliant resource with its Audio Features Endpoint: https://developer.spotify.com/console/get-audio-features-track/. Since you are using an API, you will find the appropriate resources in Module 4. You will notice in the documentation that the Audio Features Endpoint requires a Track ID as an argument for each call. This implies that you will first have to use the Search Endpoint (that is one we know, and we trained this procedure in Module 4) to get that piece of information before proceeding with the actual API call you are after. Make sure you query the combination of artist and track in the Search Endpoint, just to be sure you are getting the right ID. After all, titles could refer to songs by multiple artists. It is possible that on rare occasions you do not get a Track ID. Make sure your script is able to handle such exceptions and keep track of how often it happens to include in your report, as it is relevant information in discussing the procedure and the method. It is highly advised to do the Search API queries first and immediately write the results into a new version of your csv datafile (adding a variable trackid).
- You can read this saved file (enriched with track IDs) when you are ready to actually query the Audio Features Endpoint. This fits the best practice of (a) breaking down the project in multiple steps and (b) keeping a record of these steps in multiple code and data files. Using the Audio Features Endpoint, it is up to you to decide what audio features you consider and to defend that choice in your paper. There is no single 'right' choice, you'll have to be convincing (and preferably substantiate your choices for appropriate indicators with literature and/or a sound argument).

- Note that Spotify is also sensitive for excessive requests rates, so I would advise to pause your script in between requests as well. After all, you are about to perform over 9,000 requests. Time-outs of 0.05 seconds (one twentieth of a second) do the trick – I tested this, varying the intervals, and Spotify seemed OK with it. Just check that you are getting [200] response codes, and not [229] responses.

Data processing

- You will need to aggregate data per year. Data aggregation is explained and trained at length in Module 5. Eventually you need a dataset with one aggregated observation per relevant audio feature per year. You likely want to describe central tendency for a measure such as e.g., bpm (mean and/or median). That would allow you to sketch how music has generally changed through the years. It does not tell us much (or anything) about the diversity, the homogeneity or heterogeneity. That why, in this case, you definitely also need the *dispersion* (e.g., standard deviation) as we are assuming that the variability over time decreases (i.e., an indication of homogenisation, which it is all about). In fact, that dispersion is the key focus. The previous sentence is set in red, so it must be REALLY important!
- It would make sense here to do two rounds of aggregations of variables: one for central tendency and one for dispersion and describe them in tandem. Again, the most valuable way of communicating your results in by sensibly combining statistics and data visualisations: the mind is good as visually interpreting data but looks can be deceiving. That is where statistics are helpful.

Do you need a refresher on dispersion? See https://www.statisticshowto.com/dispersion/

Data visualisation and analysis

- You are working with longitudinal data. A time series is likely the most insightful way to visually describe the trends over time in your data. Again, we trained this in Module 5.
- Looks can deceive, so definitely include formal statistical tests to consider the relations between time and aggregated audio features per year. Module 5 contains the necessary resources.

Good luck!