DATA EXPLORATION IN PYTHON

SDV602 Assessment 2, pt. 1

Kira Byrne **October 2, 2023** **Student ID# 13509995**

Table of Contents

Introduction

Scenario

In July 2024, the New Zealand Board of Transport has been granted government funding toward upgrading public transport systems throughout the country. The following areas have been selected for upgrades:

- Auckland
- Waikato
- Bay of Plenty
- Wellington
- Nelson
- Marlborough
- West Coast
- Canterbury
- Dunedin
- Invercargill

The Board of Transport has developed an execution plan to streamline the upgrade funding from the areas in most need to least. The proposed criteria to discern this has been through data collection and analysis. Since 2017, bi-yearly surveys have been conducted in each area that have collected quantitative data on modes of daily commute. As of 2023, this raw data has been compiled into a single source and is to be uploaded to a data exploration program.

The New Zealand government has commissioned a data analysis student from Nelson to develop a data exploration program that can be accessed national and local government organisations.

What is this app?

This application is a data exploration manipulation program that can produce a plethora of graphics that breakdown raw data into visual representations for business and scientific analytics. This application will be able to explore a 5-year collection of daily commute survey data, provided by the Board of Transport, into varying applicable charts that can be viewed, manipulated, and shared between colleagues.

What is its purpose?

The purpose of this app is to explore quantitative data through a selection of graphic interfaces that can be manipulated, formatted, and shared. The current substrate for this program is daily commute data that will allow the national board of transport to explore and format raw data collected from transport surveys and share their findings with colleagues all within the program. Users should be able to open up to three data exploration screens while using the program, each screen having full manipulation, formatting, and sharing tools. Navigation between windows will be simple clicking, dragging, minimising, and maximising as most other computer application work.

What does it aim to achieve?

The primary goal of this application is to provide insight to the Board of Transport so that they may discern which areas in the country are in most to least need of funding to upgrade their local public transport services. The ability to explore the 5-years of daily commute data through various graphics including pie charts, bar charts, and trend charts aim to show a difference in how much of a local population is using the currently existing public transport. Further analysis will help to determine why there is greater or lesser adoption of public transport and can help to further develop survey material for better quality data collection and true attainment of the area in most need.

What are its features?

The application will feature:

- Organisation-base user accounts with login for privacy and auditing
- A home screen / main navigation hub where raw data can be retrieved, and graphs may be selected
- Three graph options that open into separate data exploration screens (DES) for multiple graph viewing:
    - Bar chart
    - Pie graph
    - Trend chart
- Data formatting and manipulation features including:
    - Zoom
    - Pan
    - Filter
    - Sort
- Instant messaging within graph windows

How will it be developed?

This application will be developed using Python and the following Python libraries:

PieSimpleGUI – A general user interface (GUI) library that provides a customisable window with basic interface elements (e.g. buttons, input, images, canvasing, etc.). This will be the main interface of the application (Driscoll, 2020).

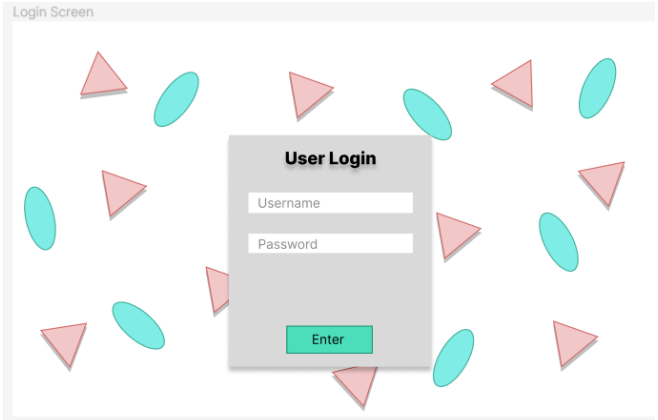
MatPlotLib – A data exploration library that is used to convert sets of provided data into various plotted graph types (e.g. bar, pie, trend) (ActiveState, 2021).

NumPy – Extended from MatPlotLib, NumPy offers scientific computing of many array object in Python to provide logical, mathematical, and manipulative operations and routines (NumPy, 2022).

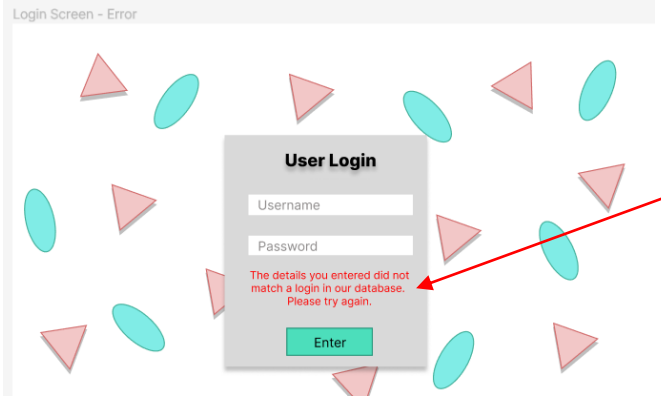
Panadas – Extended from NumPy, Pandas is used for working with numerical and time-series data providing tools for manipulation and analysis (GeeksForGeeks, 2023). Pandas can be used for cleaning, merging, and joining of data sets, as well as data visualisation.

The data will be connected to an online source where live updates can be retrieved in real time.

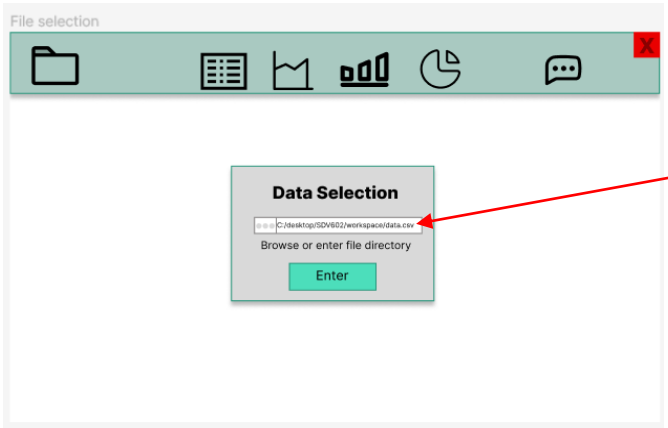
Conceptual Designs & Story Boards



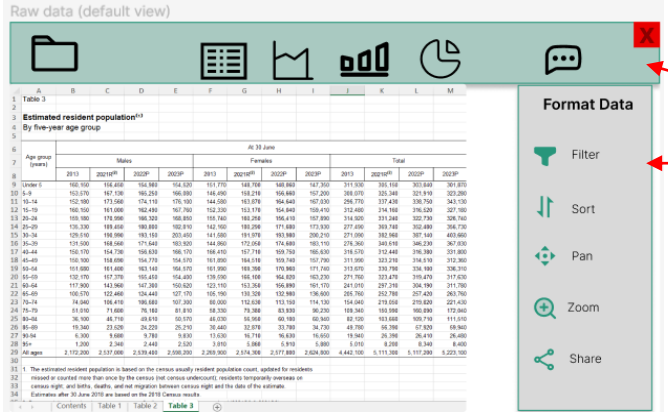
A concept of an organisation login screen taking a valid username and password.



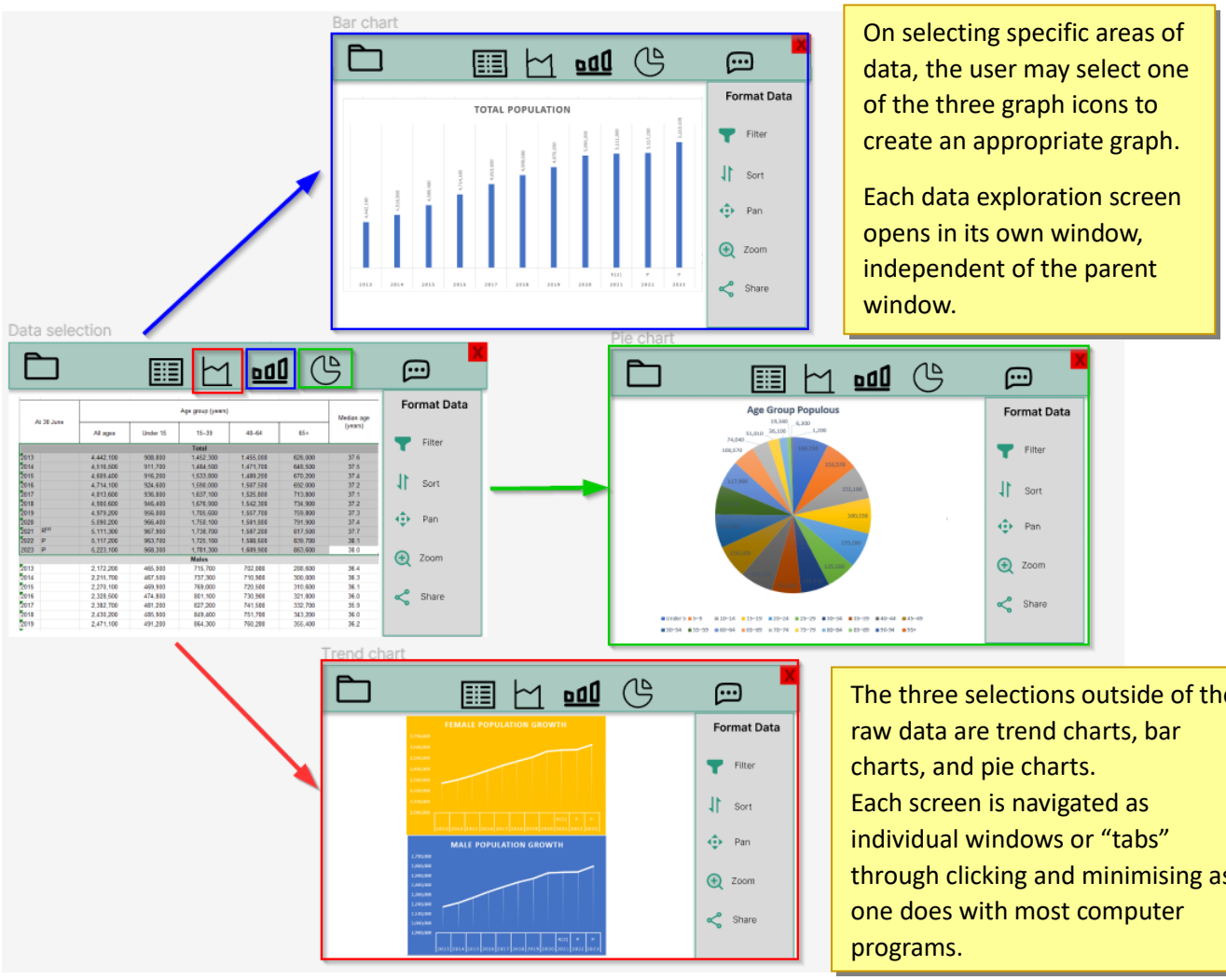
Login screen handling the error of an invalid login attempt with an error message.



Upon login, the user is brought the main interface where they must first select the data they wish to work with through browsing recent files or selecting the three dots to browse the files on their system.



After selecting the desired file, the user is presented with a raw spreadsheet to which they can manipulate with the formatting features on the right-hand ribbon and create graphs with the over-hanging icons.



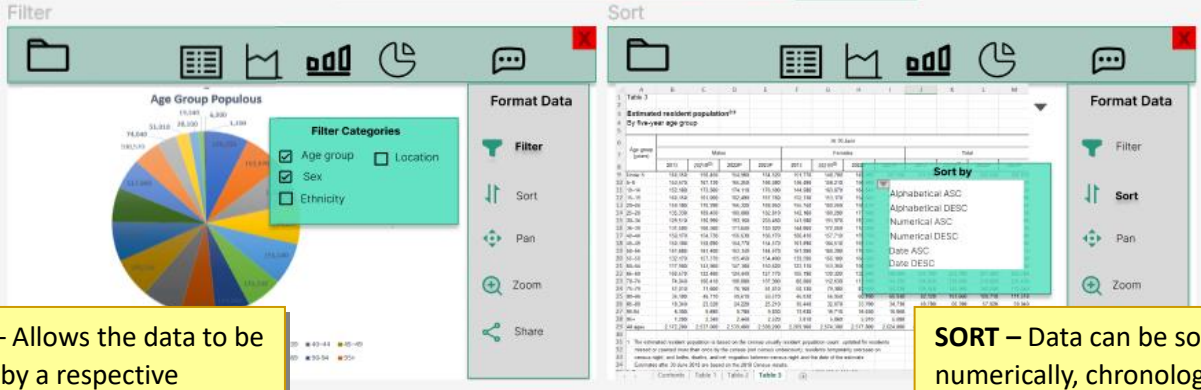
On selecting specific areas of data, the user may select one of the three graph icons to create an appropriate graph.

Each data exploration screen opens in its own window, independent of the parent window.

The three selections outside of the raw data are trend charts, bar charts, and pie charts.
Each screen is navigated as individual windows or “tabs” through clicking and minimising as one does with most computer programs.

The data contained within each screen, be it in raw form or in a graph, can be formatted through the tools on the right-hand ribbon.

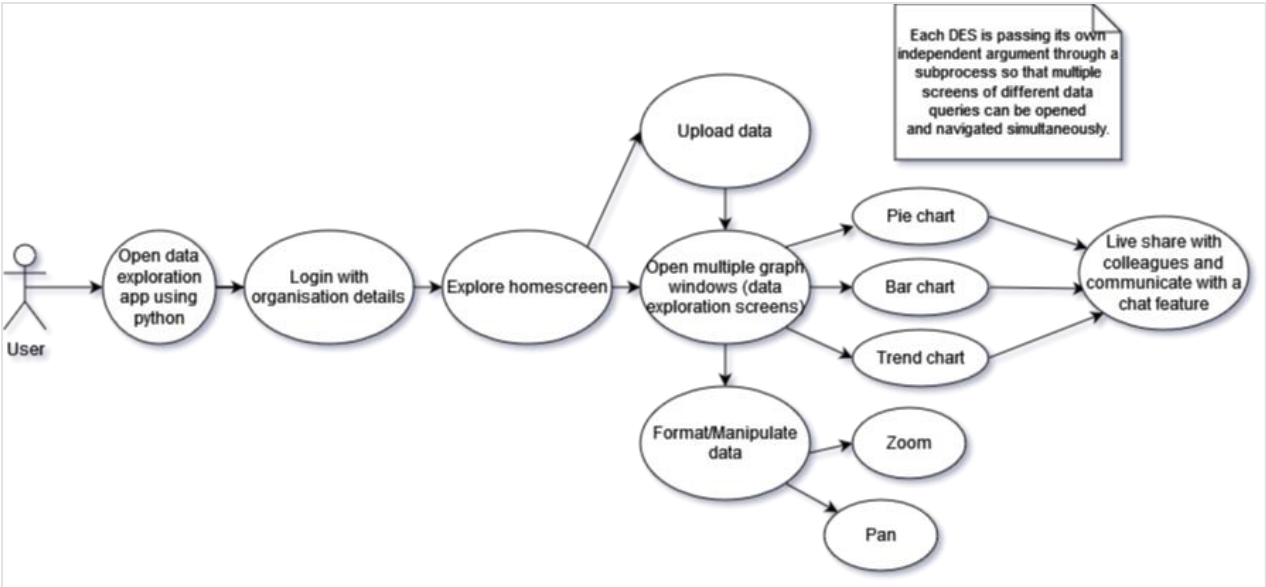
SHARE – A user may a copy of the formatted file via email within their registered organisation. This feature keeps a contacts list of all users within the registered organisation for east of viewing.



FILTER – Allows the data to be filtered by a respective category (e.g. age, sex, location).

SORT – Data can be sorted numerically, chronologically, or alphabetically by the respective categories.

Program Use Case



Reflective Journal

August 20th, 2023

This week in class, we all searched through the New Zealand data and statistics databases to find some substrate data to use. I have not yet decided what I will use to present my graphs. Perhaps data on population growth, or if I can find it, public transportation. I am interested in seeing how many migrants make up the population and if there was a spike with the post-pandemic census. If I cannot find this information, I would like to look into public transport trends and see if the uptake has spiked due to rising petrol prices.

September 4th, 2023

After much searching, I could not find any transportation data or population data that made sense to me. I felt that the csv files downloaded from StatsNZ were all formatted very oddly and used lots of acronyms that were not clear. So I spoke with Todd about creating mock data with my own format and through Mockaroo. I showed him the idea and after a dubious approval, he kindly helped me with uploading the csv file I made to a url storage space. The process was a bit confusing which I should have taken notes on, but it has been uploaded and can be retrieved when it is needed to fill the graph prototypes.

September 6th, 2023

After reading the assignment requirements again, I am unsure how exactly to visualise this graphic program. The requirements describe it as needing to display three separate windows with a graph canvas showing a different type of graph but there needs to be a navigation option to switch between them. What I struggle to understand is why these graphs can't be on a single window with a menu or buttons to select the graph type and canvas to present them on. Why do they all have to be separate? I suppose it would make sense if a user needed to be looking at multiple tabs like in web browsing and have separate chat windows for each graph. Hopefully Todd can clarify this for us on Friday.

What concerns me most at this stage is weather or not we will be provided some examples or boiler plates. I really relied on that for the last assessment, but I am feeling a lot more capable now in making a GUI. Presenting dynamic graphics sounds very challenging though.

September 8th, 2023

In class, we went over the muse of the MatPlotLib library which will allow us to create graphs. We were also given some example files of what a graph/chart program would look like. I'm glad to see how the expected code is written out so that I can follow the structure of it in creating mine.

I began storyboarding this week using Figma. I had the idea of creating a very simplified version of Microsoft Excel. The user would first be prompted to login using valid details (which I will most likely have to create using an online database like firebase or mongo). The home screen would then prompt them to open a data file to view from a URL, then it will present the raw csv data. From here, the user can select the different graph types, pie, bar, and trend from a top ribbon. The top ribbon also includes a chat bar option. There will be a graph canvas taking up 90% of the screen and a formatting ribbon on the right side which will allow the user to filter, sort, zoom, pan, and share the graph.

September 14th, 2023

The repository has finally been used today. I have created a main GUI file that draws the graph canvas and displays buttons for each graph type. I am yet to format the buttons, but I plan to code each in a way that opens the respective graph in a new window as the assessment sheet requires. I'm not sure what the home page will look like but perhaps for now it can just be the three buttons.

I need to investigate the login page this weekend and get this underway. Time is ticking!

September 19th, 2023

I have caught a very bad cold that is going to put me back a bit. Todd has kindly given me an extension to the 2nd of October which should hopefully be enough time.

September 23rd, 2023

I am finally getting over my cold, it really took a lot out of me. After going over my test scripts, I was able to create simple input boxes for a username and password to create a psudo-login screen. I have also made buttons which call each graph type from my graphs.py file. The main issue now is that I can only open one graph at a time and when I close the graph window, the whole program closes. The main error is that the app is trying to pass two arguments into the backend service before crashing the app. I have been looking into subprocesses and kwargs to see about handling this but I am still not sure what code to write and where.

So far the app works to have a home page and open a separate window for each graph. I'm quickly running out of time to get these bugs fixed but hopefully it will click soon.

Test Scripts

Test scripts can be found at the following github repository link:

https://github.com/KByz/sdv602-assessment-2-part-1-kira-byrne

References

ActiveState. (2021, January 14). *What is Matplotlib in Python? How to use it for plotting?* https://www.activestate.com/resources/quick-reads/what-is-matplotlib-in-python-how-to-use-it-for-plotting/

Driscoll, M. (2020, June 17). *PySimpleGUI: The simple way to create a GUI with Python – Real Python*. Python Tutorials. https://realpython.com/pysimplegui-python/

GeeksForGeeks. (2023, July 25). *Introduction to pandas*. GeeksforGeeks. https://www.geeksforgeeks.org/introduction-to-pandas-in-python/

NumPy. (2023). *What is NumPy? — NumPy v1.26 manual*. https://numpy.org/doc/stable/user/whatisnumpy.html