Kyle Ford

CPSC 5530

Due Date: 2/12/2023

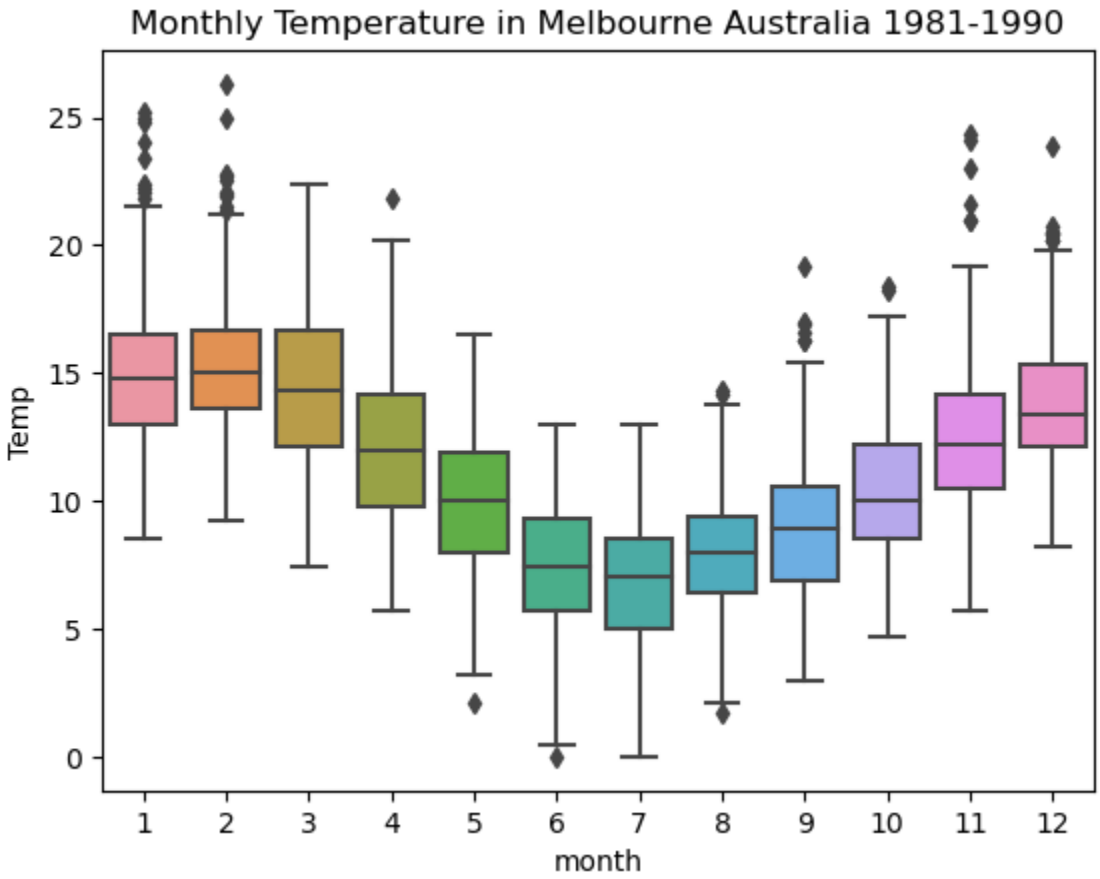
Assignment 1 Report

Dataset 1: Daily Minimum Temperature in Melbourne, Australia

1. The data used for this visualization comes from the website https://machinelearningmastery.com/time-series-data-visualization-with-python/ . The purpose of this website is to provide an overview of data visualization techniques for time-series data. The URL the data was downloaded from is https://raw.githubusercontent.com/jbrownlee/Datasets/master/daily-min-temperatures.csv
2. This data is in the form of a table. Specifically, this table contains two columns corresponding to two total attributes. The two attributes in this dataset are date and temperature. Each row, or entry, in the dataset corresponds to the temperature on a given day.
3. This data set makes use of items and attributes to describe the data. There are two attributes given in this data set. They are date and temperature. Each row in the table represents its own item, which corresponds to a given day.
4. The date attribute in this data set can be described as ordinal data. There is an implicit ordering to the data as some dates fall after others, but the numerical values stored in the date attribute do not mean anything quantitatively. For example, it would be inappropriate to calculate the average date stored in the table. This attribute can also be described as sequential, as there is a minimum and maximum range. For this dataset, we begin with date 1/6/1981 as our minimum value and end with date 12/31/1990 as our maximum value. Certain aspects of this attribute can be considered cyclic data as they occur multiple times. For example, there are multiple instances of the same month occurring in different years, and different days of the month occurring within each month. The second attribute, temperature, is continuous, quantitative data. The values fall on a continuous range that can take on any measured quantitative value. Additionally, the values support arithmetic computation. For example, it would be appropriate to evaluate the average temperature observed in the month of March, or to compute the standard deviation of temperatures collected throughout all years measured.
5. Preprocessing for this data set involved the following steps. First, the data is imported as a pandas data frame inside of Python. We check the data type each attribute is encoded as, and note that the date column is stored as an object. To perform our analysis, we want the date column to be stored as a date time object. This is accomplished using the pd.to_datetime() function, which converts the string value into a date time value. Once this is complete, we create three new columns in the data frame corresponding to the year, month, and day for each observed temperature. Doing so allows for more in depth

analysis and easier comparison. This is accomplished using the pd.DatetimeIndex() function.

6. Visualization: Box Plot of Temperature statistics by month:



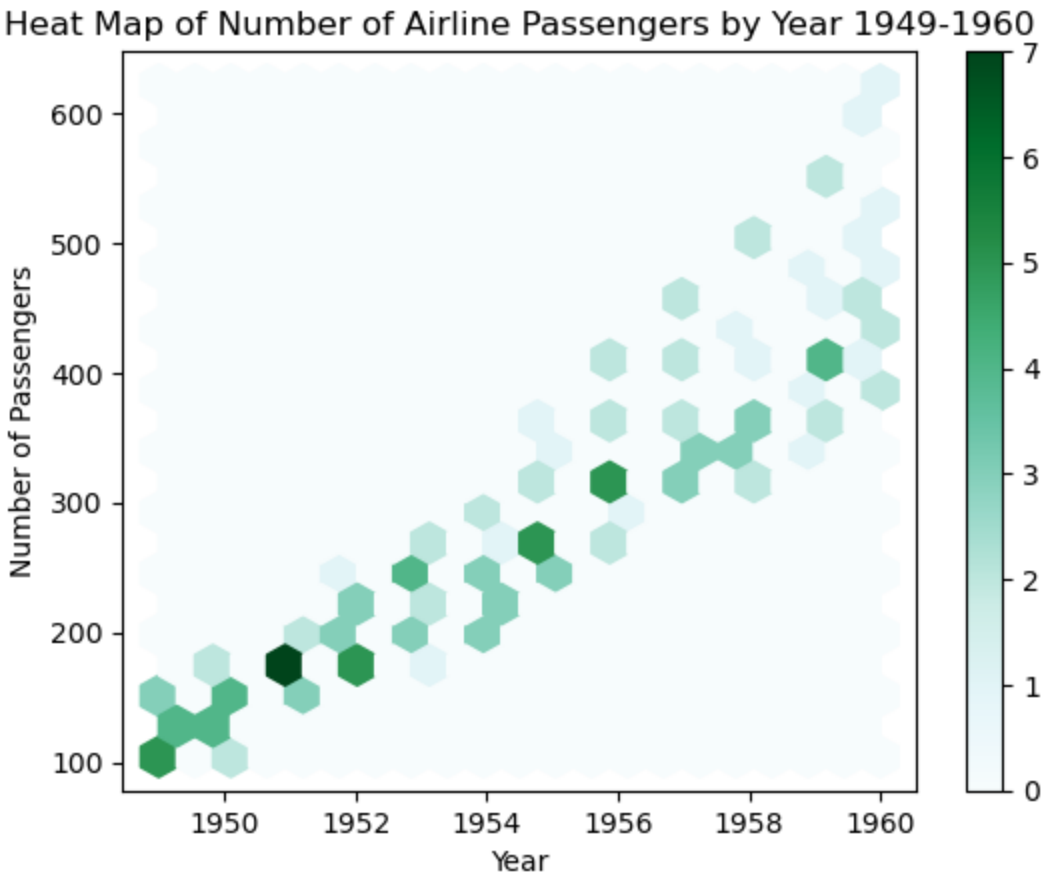
There are a few things to note that the above visualization illustrates. First, note how the average temperature appears to be cyclical, which higher values around 15 degrees C occurring in months 1, 2, 3, 11, and 12, while lower values fall in the middle months, namely months 6 and 7. Since this data is accumulated over a ten year period, this is a fair assumption to make. Second, note that outlier values tend to occur during the winter months where the average temperature is higher. Specifically, outlier temperature values tend to skew towards the high end of the scale, being higher than the 3rd quartile temperature. There are only 3 instances of outlier values that occur on the low end, below the 1st quartile temperature. All of these values all occur during the months when the average temperature is low. In sum, what these values point to is seasonality in Melbourne, Australia being characterized by warmer winter months and cooler summer months.

Dataset 2: Monthly Air Passengers from 1949-1960

1. The data used for this visualization comes from the website https://kanoki.org/2020/04/27/time-series-analysis-data-visualization/, whose goal is to

give an overview of data visualization techniques for time-series data. The specific dataset was downloaded from the Kaggle page https://www.kaggle.com/datasets/rakannimer/air-passengers.

2. This data set is tabular in nature. Specifically, this data set can be classified as a simple flat table with two columns. The first column is named month and contains the year-month pair of the date. The second column corresponds to the number of passengers that traveled by air in that particular year-month pair.
3. Since this data is arranged in a simple table with two columns, this data has two attributes. The first can be called month and corresponds to the specific year-month pair date given. The second attribute is number of passengers. Each row in the table represents its own entry, or item, corresponding to the number of passengers that traveled by air in a given year-month pair.
4. The month attribute is ordinal data. The data is ordered as certain months of certain years come before or after others, which implies a ranking. We can describe the dates in terms of other dates, but the quantitative value of the dates is of little consequence. For example, we cannot calculate the standard deviation of the dates. We can calculate, for example, the difference between two dates, say the number of months in between two months in question, but this refers back to the ordering of the data and not to the quantitative values of the data. Date data is thus also sequential. Finally, the month column is cyclical, as the same months occur repeatedly in different years. The number of passengers attribute is discrete quantitative data. The data can take on a fixed number of values who numerical values are of consequence. For example, we cannot achieve a fractional part of an air passenger, making this data discrete. The data is quantitative because it supports arithmetic comparison. For example, we can calculate the average number of air passengers that traveled in the month of June for all years data is available. Semantically, this data gives the number of passengers that traveled by air in a given year-month pair for the years 1949-1960.
5. First, the csv file is imported into a pandas data frame using the read_csv() function in Python. We check the data types of each attribute to ensure each attribute is encoded as the proper data type. This is done with the dtypes() function in pandas. We see that the date object is encoded as a string object, and use the pd.to_datetime() function to convert the string into a date time object. Once this has been completed, we rename the month column to be Date using the .rename() function. Doing so allows us to create individual columns for year and month without confusion. It also better encapsulates the data held in the newly named Date column. We create individual columns for year and month using the pd.DatetimeIndex() function.
6. Visualization: heatmap of number of air passengers traveling in a given year.

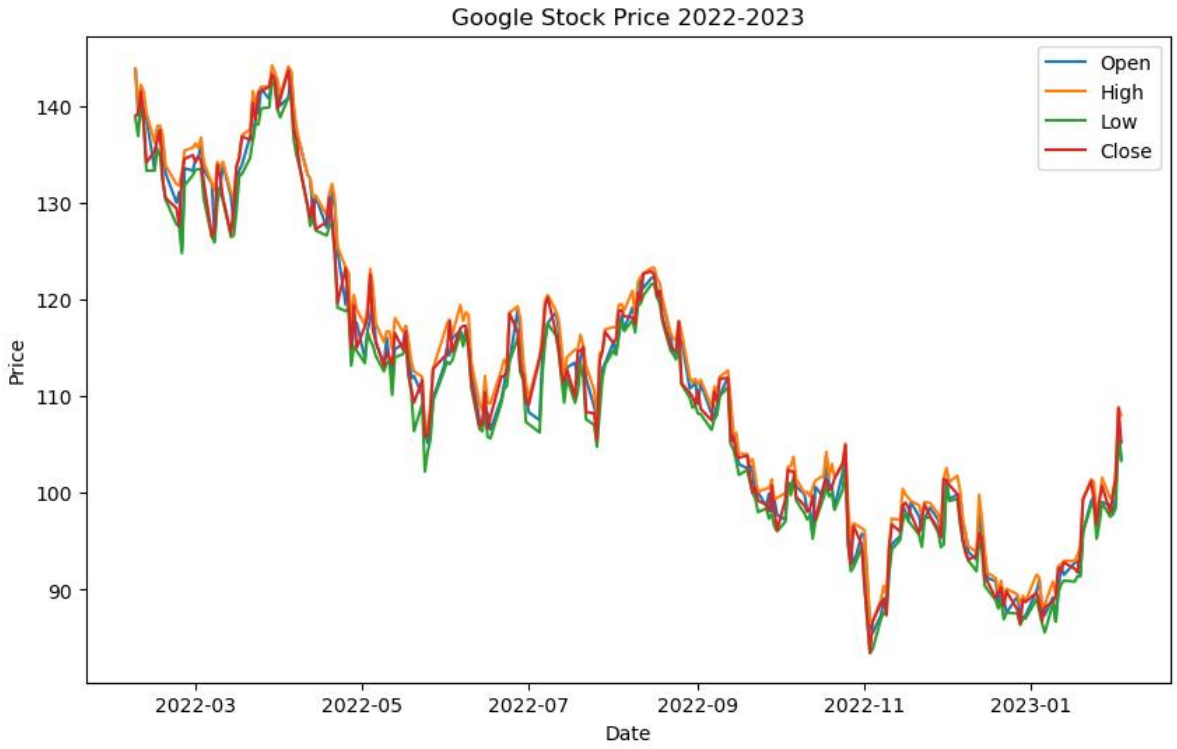


This visualization shows the number of air passengers that traveled per year. Specifically, it illustrates a distribution of values by highlighting where the number of passengers that traveled in a given year was highest. For example, we can see that in the year 1951 there was a large value of around 200 air passengers. Thus, from the visualization we can identify a couple of trends. First, the darkest hexagons, corresponding to the highest occurrences of that number of air travelers, occur for the lowest values of passengers in the beginning years of our range. Thus, we can state that air travel was relatively consistent in the years 1949-1952. By contrast, we see the lightest hexagons occur for the later years with a larger number of passengers. There is a greater spread in the number of air passengers as the date goes on, and as the number of passengers increases.

Dataset 3: Google Stock Price from 2022-2023:

1. This data was collected from the website https://www.vshsolutions.com/blogs/time-series-data-visualization-in-python/, a blog whose goal is to detail data visualization techniques for time-series analysis. The data set from downloaded from the website https://finance.yahoo.com/quote/GOOG/history?p=GOOG
2. Like the other two data sets, this data set is a simple flat table composed of rows and columns. As such, it is composed of attributes, or column values, and rows, where each row represents a data entry in the table.

3. Each column in the flat table represents an attribute. Thus, this table is composed of the attributes Date, Open, High, Low, Close, Adj Close, and Volume. Each attribute corresponds to a column name. Each row in the table represents an item, which is an individual entry in the table. Specifically, each item in the table represents the values of Google stock on a particular day.
4. The first attribute in this data set is date. This is an ordinal, sequential data type. The date data type is ordinal because it represents ordered data, where some dates occur before or after others. It is also sequential in that there is a maximum and a minimum range for our data values. With the date data type, this corresponds to the first day for which data is available and the last day for which data is available. The Open, High, Low, Close, and Adj Close data types are all continuous quantitative data types. They are continuous in that their values can take on any value and quantitative in that their values support arithmetic comparison. Volume, the final attribute in this data set, is a discrete quantitative data type. It is discrete in that it's value is countable, and it is quantitative in that the numerical values of this attribute support arithmetic operation. Semantically, the date attribute represents the date being observed, Open corresponds to Google's stock price when the market opened, High represents the peak stock value for Google stock in a given day, Low represents the lowest value for Google stock in a given day, Close represents the price of Google stock when the market closed, Adj Close represents a normalized closing price, and Volume represents the number of transactions made involving Google stock in a given day.
5. First, the data is imported as a pandas data frame using the pd.read_csv() function in Python. Next, we check the data types encoded in the data frame using the dtypes() function. Note that the date object is stored as a string object. We convert this to a date time type using the pd.to_datetime() function. Now that we are working with date objects, we sort the values in descending order by date using the sort_values() function. Now, we can begin our visualization. Using Matplotlib's pyplot library, we plot four separate lines corresponding to the Open, High, Low, and Closing Google stock prices.
6. Visualization:



First, note the relative similarity in all stock prices for a given day. This visualization represents high similarity between all observed values of Google stock price on a given day. That is, there is little separation between the high and low values, and the opening an closing values. We also note that over the observed range of dates, Google stock price fell from a value of about $140 to $110. However, the most recent trend shows Google stock price steadily increasing since the onset of the year 2023.The lowest value of Google stock was observed in November of 2022 and the highest value of Google stock was observed in April of 2022.