COMP1800 Data Visualisation

Module Leader: Prof. Chris Walshaw

Coursework

001255924

Hadis Babakhani Roudbardeh

A brief Introduction to Data visualization

The graphical representation of data and information is known as data visualisation. It is a strong tool that allows us to graphically explain complex information, making it easier to understand, analyse, and make informed decisions.

Data visualisation is developing visually appealing representations of data such as charts, graphs, and maps that are simple to read and analyse. These graphics assist us in identifying patterns, trends, and outliers in the data and provide insights that raw data alone may not deliver.

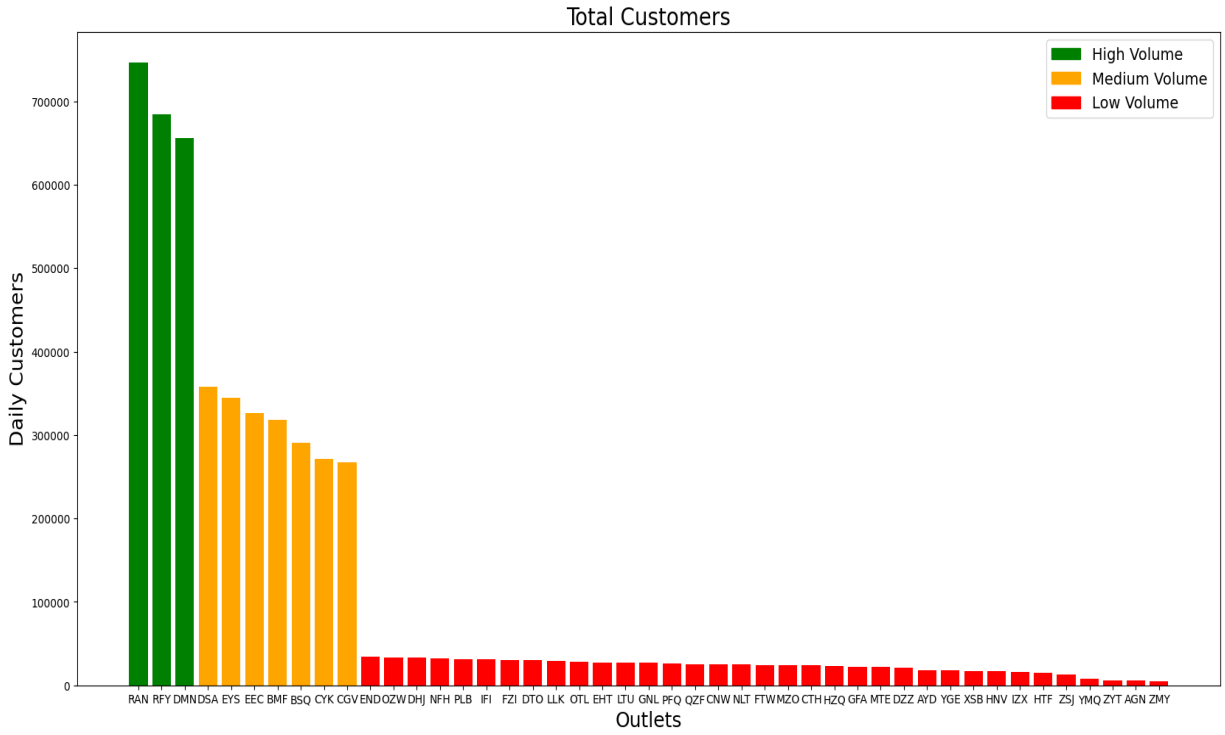
In today's data-driven world, effective data visualisation is a key ability since it lets us to deliver our results to a wide range of people, from executives and stakeholders to the general public. It has many implications, including business, science, education, and media, to name a few.To develop effective representations, it is essential to understand data visualisation techniques such as selecting the appropriate type of chart or graph, choosing the suitable colour scheme, labelling and titling, and designing for clarity and simplicity. Everyone can produce successful visualisations that improve their data analysis and communication skills with the correct tools and strategies.

First of all I created summary dataframe for each datasets and a dataframe for daily customers.

Visualisation 1:Segmented Barchart

```
import matplotlib.patches as mpatche
colours = []
for name in data.columns:
    total_sales = data[name].sum()
    if total_sales > 500000
        colour = 'green'
    elif total_sales > 100000
        colour = 'orange'
    elif total_sales > 0
        colour = 'red'
    else:
        colour = 'black'
    colours.append(colour)
high_colour = mpatches.Patch(color='green', label='High Volume')
medium_colour = mpatches.Patch(color='orange', label='Medium Volume')
low_colour = mpatches.Patch(color='red', label='Low Volume')


plt.figure(figsize=(20, 10))
x_pos = np.arange(len(data.columns))
plt.bar(x_pos, data.sum(), align='center', color=colours)
plt.xticks(x_pos, data.columns)
plt.xlabel('Outlets', fontsize=18)
```

```
plt.ylabel('Daily Customers', fontsize=18)
plt.title('Total Customers', fontsize=20)
plt.legend(handles=[high_colour, medium_colour, low_colour], fontsize=14)
plt.show()
```

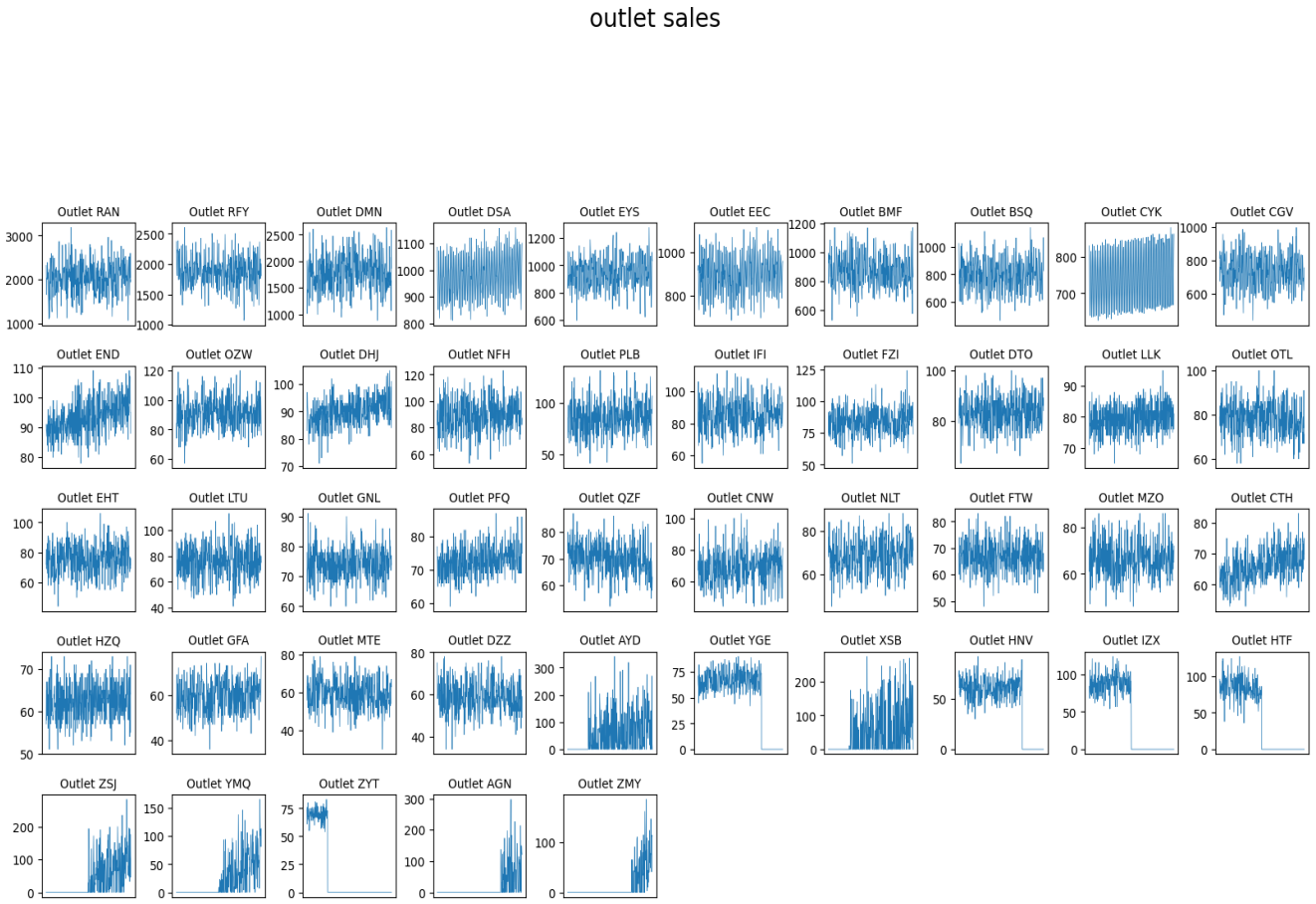


Justification: The bar chart's color-coding divides the outlets into three groups based on the volume of their sales: high volume, medium volume, and low volume. High-volume outlets are those with total sales of more than 500,000, which are indicated by green-colored outlets. These stores are the best-performing among all the stores and have the highest overall sales. Medium-volume stores are those that are orange and have annual sales between 100,000 and 500,000. These businesses are doing okay and have average sales. Low-volume outlets are those with total sales between 0 and 100,000 that are indicated by the colour red. In comparison to the other outlets, these ones perform poorly and have the lowest overall sales.

Explanation: The visualisation gives a clearer picture of how the outlets perform by segmenting them depending on sales volume. It draws attention to the sources that perform the best, the average performers, and the outlets that perform poorly. It is simpler to determine which outlets need to improve and which outlets are performing well thanks to this segmentation. It can assist managers and company owners in concentrating on the areas that require the most attention and in allocating resources appropriately. I can split the outlets depending on sales volume and acquire insights into their sales performance by using the color-coding I used in the bar chart.

Visualisation 2: Line plot All subplots

```
counter = 1
fig = plt.figure(figsize=(20, 20))
fig.suptitle('outlet sales', fontsize=20, position=(0.5, 1.0))
for name in data.columns:
    sub = fig.add_subplot(10, 10, counter)
    sub.set_title('Outlet ' + name, fontsize=10)
    sub.plot(data.index, data[name], linewidth=0.5)
    sub.axes.get_xaxis().set_ticks([])
    counter += 1
plt.subplots_adjust(wspace=0.4, hspace=0.4)
plt.show()
```



Justification: This visualisation is included because it offers a quick and simple method to see the daily sales of each outlet over time. The plot is comprised of a grid of smaller plots, each of which represents the sales information for a particular outlet. This makes it simple to find any patterns or trends in the data and compare the sales trends across various outlets.It shows outlets YGE,HNV,IZX,HTF and ZYT have the least sales during the year and they have been closed by the company.In contrast, Outlets AYD,XSB,ZSJ,YMQ,AGN and ZMY are the new that have been opened during the year.

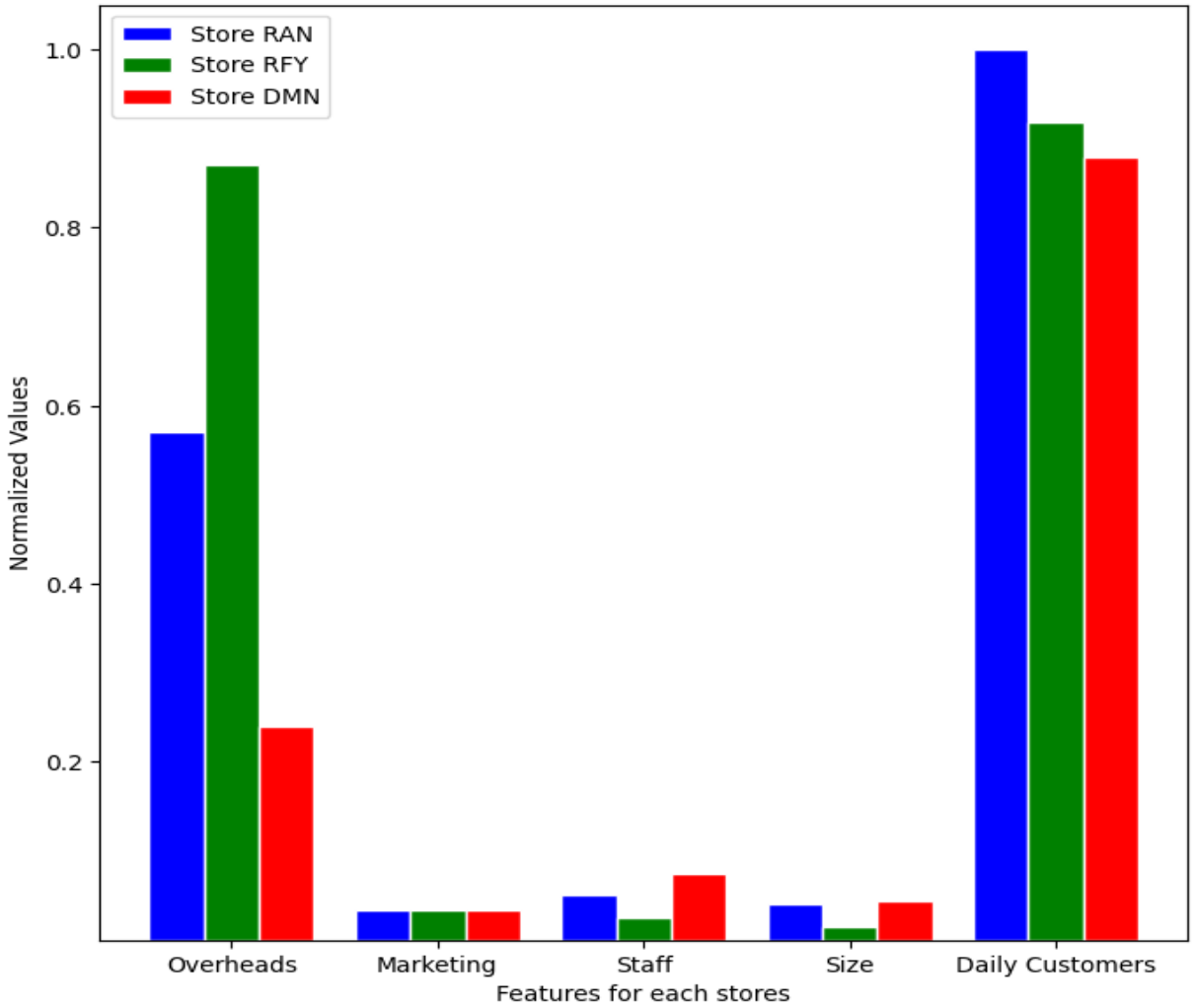
Explanation: There is a lot of information about the data that is shown by the visualisation. The viewer may easily determine which stores have the highest and lowest sales by graphing the daily sales of each location over time. Also, the plot makes it simple to spot any variations in sales trends over time. For instance, the spectator can spot any abrupt variations in sales volume or seasonal tendencies in sales.

The plot also enables a more thorough examination of each outlet's sales information. The visitor can determine which stores consistently have the highest and lowest sales by examining each subplot separately. This level of specificity might be helpful in pinpointing problem areas or creating focused marketing campaigns.

Visualisation 3: Comparative Barchart High Volume

```
selected = ['RAN', 'RFY', 'DMN']  # high volume
colours = ['b', 'g', 'r', 'c', 'm', 'y', 'k']
plt.figure(figsize=(8, 8))
c = 0
n_bars = len(selected)
x_pos_base = np.arange(len(summary_data.columns))
bar_width = 0.8 / n_bars
for name in selected:
    values = normalised_data.loc[[name]].values.flatten().tolist()
    x_pos = [x + (bar_width * c) for x in x_pos_base]
    plt.bar(x_pos, values, color=colours[c % len(colours)],
            width=bar_width, edgecolor='white', label='Store ' + name)
    c += 1
plt.yticks([0.2, 0.4, 0.6, 0.8, 1.0])
x_pos = [x + (bar_width * (c - 1) / 2) for x in x_pos_base]
plt.xticks(x_pos, summary_data.columns)
plt.legend()
```

```
plt.xlabel('Features for each stores')
plt.ylabel('Normalized Values')
plt.show()
```



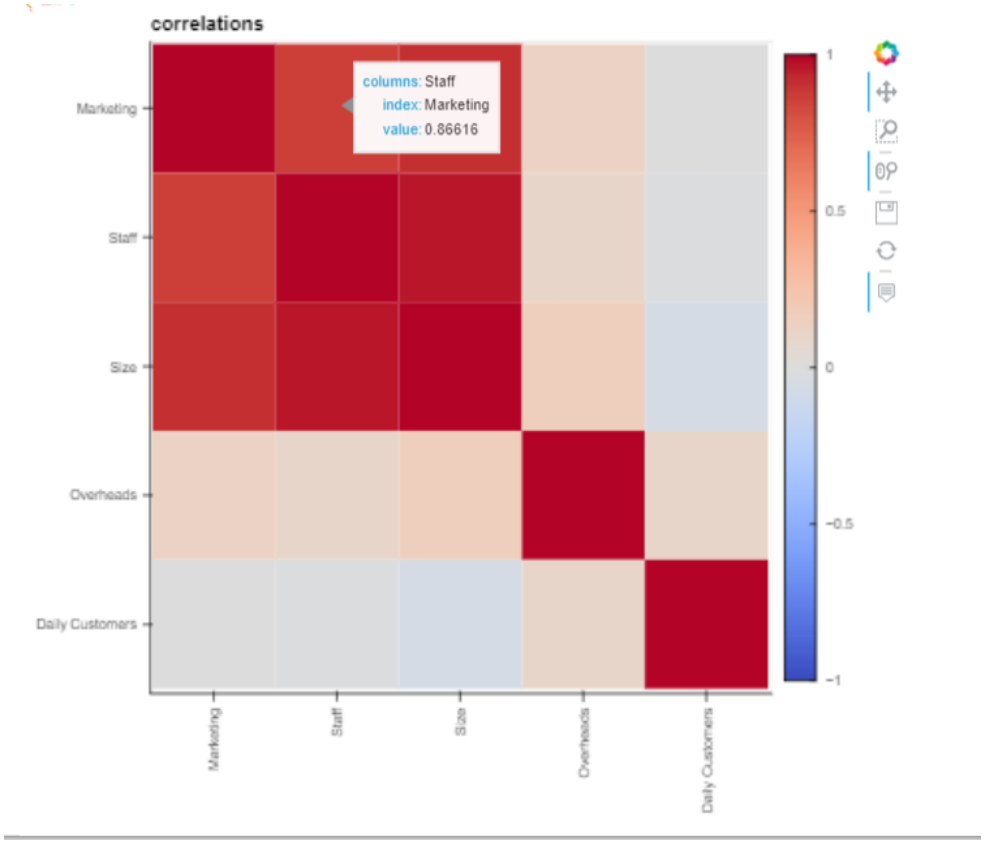
Justification: The comparison of the sales data for a selected group of high-volume stores is the foundation on which to create the visualisation above. The viewer can determine any similarities or differences in the sales patterns of these stores by comparing their sales data, and they can then use this information to create specific sales improvement strategies.As we can see Daily customers for store RAN is the highest and for store DMN is the lowest.or we can see Marketing for three of them are in equal range.

Explanation: The comparison of the sales data for the identified high-volume stores is shown through the graphic. A bar chart, with each bar representing the sales information for a single product category, forms the plot. It is simple to compare the sales information for the various retailers because the bars for each shop are displayed side by side. The visual

comparison of the data is further improved by the use of different colours for each shop.The plot additionally enables a comparison of the sales information for various product categories. The observer may easily determine which product categories have the biggest sales and which have the lowest sales by examining the height of each bar. Determining out which product categories to concentrate on for marketing or increasing sales can be done with the help of this information.The plot offers an efficient way to compare the sales data for a certain selection of high-volume retailers, making it possible to spot similarities and variances in sales patterns quickly. It can be a helpful tool for managers and business owners when creating focused marketing campaigns and enhancing sales results.

Visualisation 4:Interactive Heatmap correlation

```
import hvplot.pandas
plot = summary_data.corr().hvplot.heatmap(
    frame_height=500, frame_width=500,
    title=' correlations',
    rot=90, cmap='coolwarm'  # see http://holoviews.org/user_guide/Colormaps.html
).opts(invert_yaxis=True, clim=(-1, 1))
hv.extension('bokeh')
plot
```



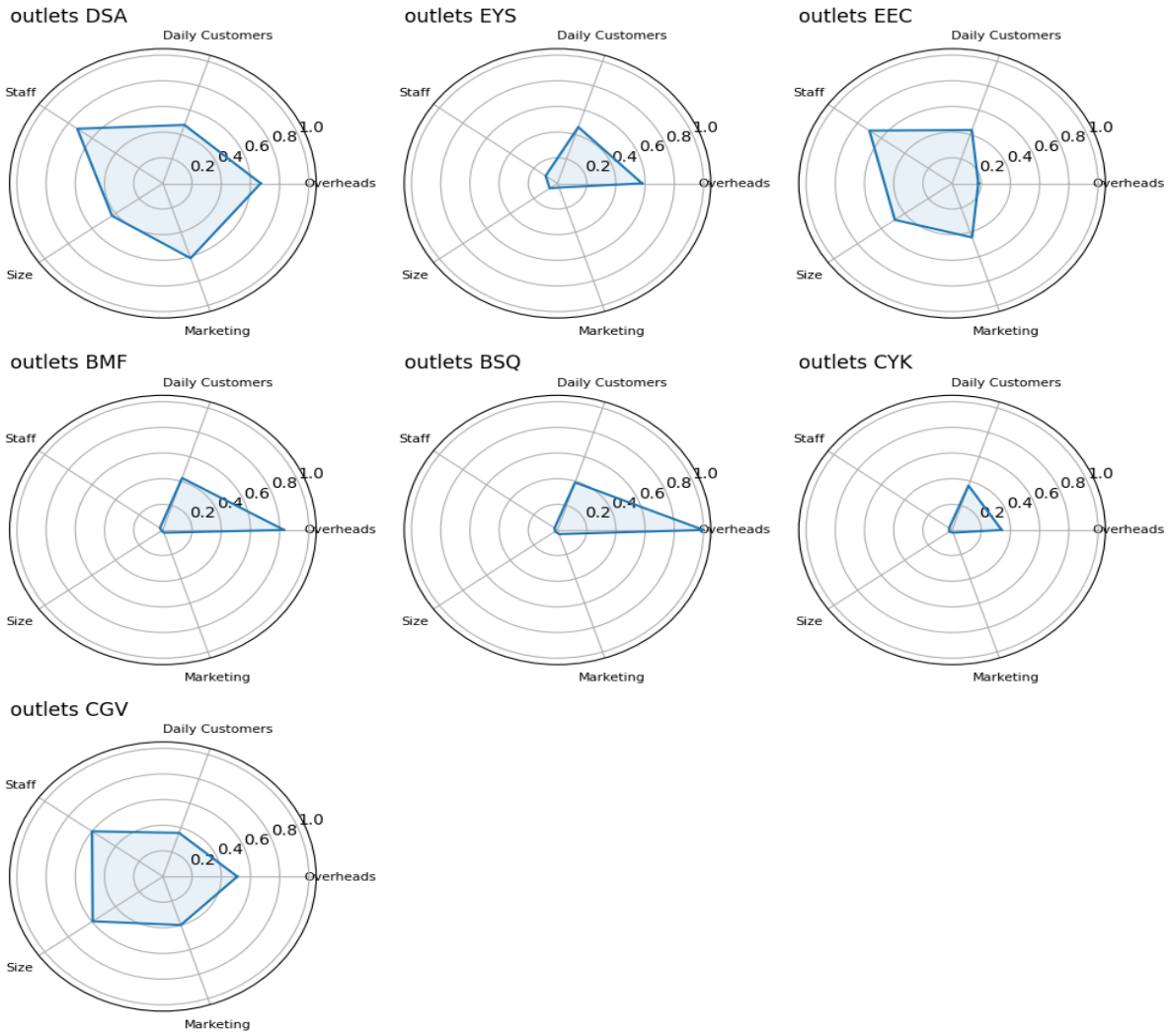
Justification: Data visualisation can assist reveal patterns and trends in data by identifying links between variables. The heatmap displays a range of colours that correlate to the degree and direction of the association when the coolwarm colour map is used and the colour range is set to (-1, 1). The usage of HoloViews enables interactivity and customization of the display, making it easier to explore and analyse the data.We can see the correlation between each feature by using hover.Here I use hover for Marketing and staff.we can see that the correlation is high and it's 0.886.

Explanation: A colour map ranging from blue to red is used in the visualisation, with blue signifying negative correlation, red representing positive correlation, and white representing no correlation. The frame height and frame width options, as well as the title argument, can be used to change the size of the heatmap and the title of the visualisation. The graphic that results allows the user to examine the data and uncover patterns and relationships between variables. The visualization's interactivity allows the user to hover over cells to reveal correlation coefficient values and zoom in or out to focus on specific regions of interest. Overall, this visualisation is a valuable tool for data exploration and analysis.

Visualisation 5:Radar plots for Medium volumes

```
selected=['DSA', 'EYS', 'EEC','BMF','BSQ','CYK','CGV']

n_attributes = len(normalised_data.columns)

angles = [n / float(n_attributes) * 2 * np.pi for n in range(n_attributes + 1)]

plt.figure(figsize=(8, 8))

counter = 1
```

```
for name in selected:
    values = normalised_data.loc[[name]].values.flatten().tolist()
    values += values[:1]
    sub = plt.subplot(3,3, counter, polar=True)
    sub.plot(angles, values)
    sub.fill(angles, values, alpha=0.1)
    sub.set_ylim(ymax=1.05)
    sub.set_yticks([0.2, 0.4, 0.6, 0.8, 1.0])
    sub.set_xticks(angles[0:-1])
    sub.set_xticklabels(normalised_data.columns, fontsize=8)
    sub.set_title('Outlets ' + name, fontsize=12, loc='left')
    counter += 1
plt.tight_layout()
plt.show()
```



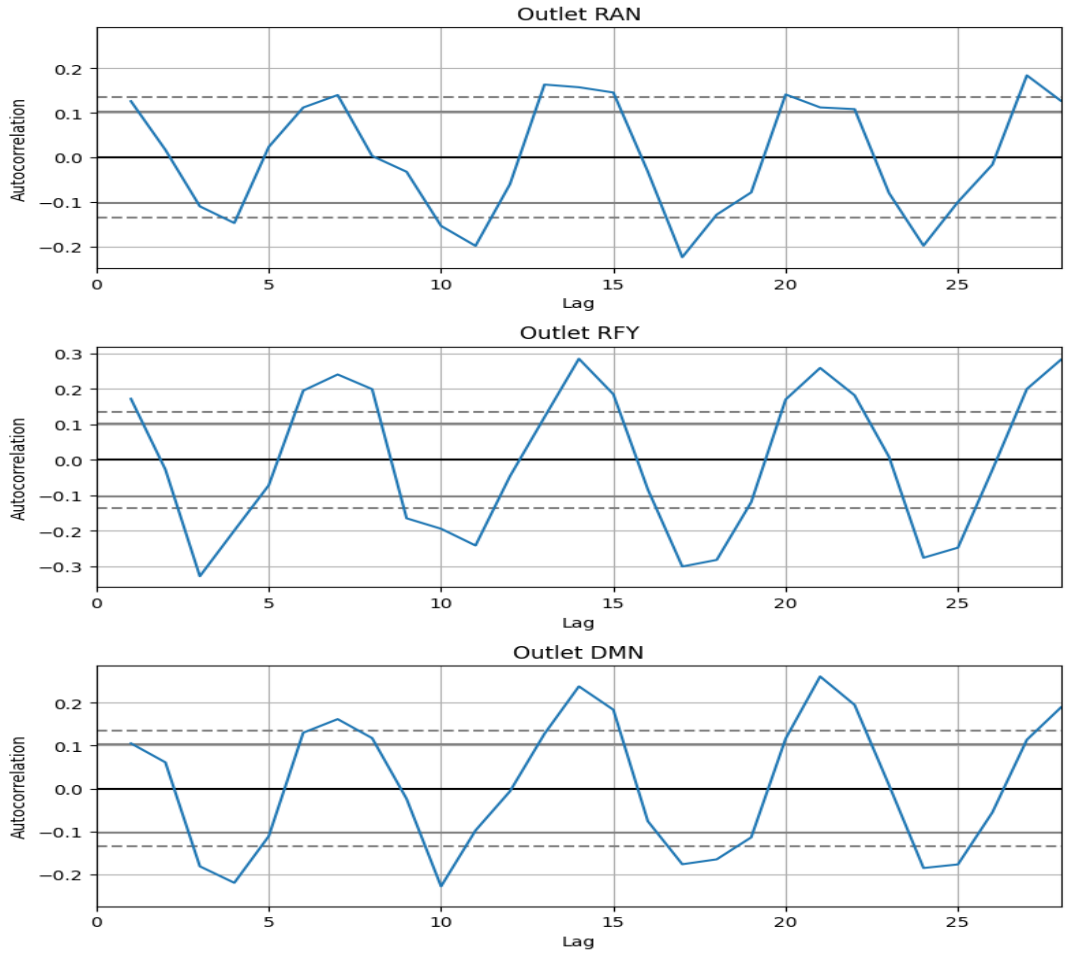
Justification: there is a collection of radar chart representations for the normalised data dataset's Medium volume outlets. The radar chart is an useful tool for comparing the characteristics of different outlets in a compact and understandable manner, making it especially useful for medium volume outlets that sell a big number of products. It might be difficult to assess and compare the properties of different products in such outlets, particularly if the products are comparable or have overlapping features. By presenting a selection of retailers on a series of radar maps, it becomes easier to identify and compare the most important qualities of the stores. Inventory management, product selection, and marketing techniques can all benefit from this.

Explanation: Each selected outlet is displayed on a separate radar chart in this view. The outlets' attributes are plotted on the axes, and the values for each attribute are joined by lines to form a polygon. The polygon's colour is set to a light shade to make the displayed data easier to view. The attributes are shown on the x-axis, and the y-axis indicates the value range for each property, ranging from 0 to 1. To aid comprehension, the values are shown at regular intervals along the y-axis. Each radar chart's title corresponds to the name of the outlet represented. By comparing the polygons throughout the multiple radar charts, it becomes easier to identify and compare the most important characteristics of each outlet. This can help with inventory management, outlet selection, and marketing techniques, especially in

medium volume outlets with a high number of customers. The radar chart visualisation is an effective approach to summarise and represent multivariate data in a compact and intuitive manner.

Visualisation 6:seasonality for 28 days

```
selected = ['RAN', 'RFY', 'DMN']


# Create a figure with subplots
fig, axs = plt.subplots(nrows=len(selected), ncols=1, figsize=(8, 10))


# Loop through each store and plot its autocorrelation plot in a different subplot
for i, name in enumerate(selected):
    pd.plotting.autocorrelation_plot(data[name], ax=axs[i])
    axs[i].set_xlim([0,28])
    axs[i].set_title('Outlet ' + name)


# Adjust the layout of subplots
fig.tight_layout()


# Save the figure as a PNG image
fig.savefig('autocorrelation_plots.png', dpi=300)
```

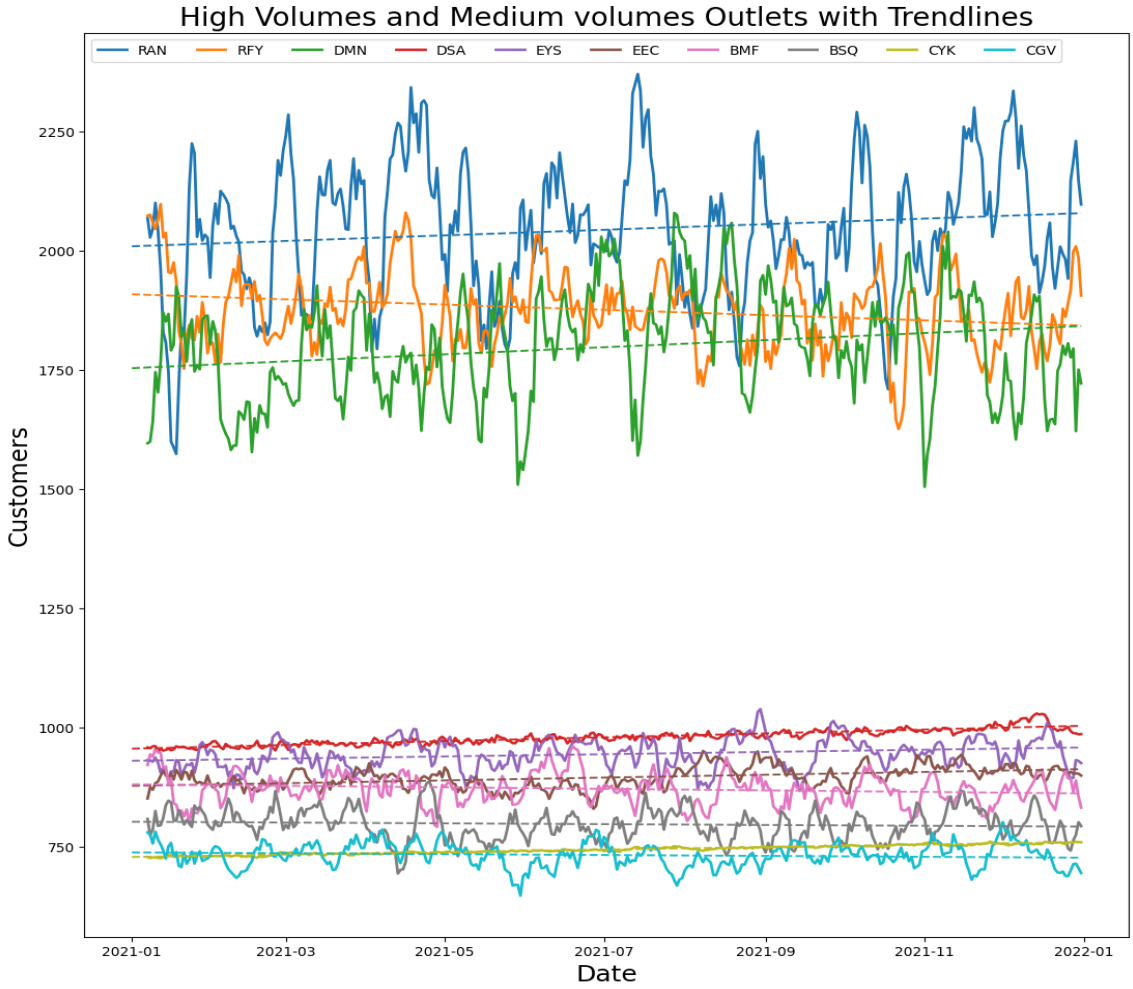


Justification: Those are my high-volume outlets, and I have been visualising them for 28 days. As we can see, there are four distinct patterns for each of the 28 days. Throughout one month, there are four cycles (28 days). It displays daily visitors throughout the period of one month, with a spike at the end of the weekend. The autocorrelation plot is a valuable tool for examining time series data trends, such as the daily number of visitors to rising outlets. I choose high volume outlets, RAN, RFY, and DMN, and showed their autocorrelation functions across 28 days in this graphic. We can observe data patterns and the existence of cycles or trends by visualising the autocorrelation function. The visualisation enables us to identify the high and low points in the time series as well as the frequency of the data cycles. This data can be used to make business decisions, such as modifying staff schedules or promotions, in order to increase sales and improve customer satisfaction.

Explanation: The visualisation demonstrates that the daily number of visitors to high-traffic sources follows a cyclical pattern with a specific weekly cycle. The number of visitors rises during the week and peaks at the end of the week, then falls on weekends. This trend repeats every week, indicating that customers visit the outlets on a regular basis. We can see a correlation between the number of visitors on one day and the number of visits on subsequent days. This link steadily declines as the time lag rises, demonstrating that short-term factors such as weather or marketing influence the daily number of visitors. The graphic also reveals a seasonal influence in the data, with the number of visits growing in the middle of the month and dropping at the end. This data can be used to estimate future trends and make informed business decisions.

Visualisation 7: High and medium Volume Outlets with Trendlines

```
period = 7
rolling_average = data.rolling(window=period).mean()
selected=['RAN', 'RFY', 'DMN','DSA', 'EYS', 'EEC','BMF','BSQ','CYK','CGV']
print(data[selected].head())
plt.figure(figsize=(8, 8))
plt.gca().set_prop_cycle(None)
plt.plot(rolling_average[selected], linewidth=2)
plt.gca().set_prop_cycle(None)
for name in selected:
    x = np.arange(len(data[name]))
    z = np.polyfit(x, data[name], 1)
    trend = np.poly1d(z)
    plt.plot(data.index, trend(x), linestyle='--')
plt.xlabel('Date', fontsize=18)
plt.ylabel('Customers', fontsize=18)
plt.title('High and medium Volume Outlets with Trendlines ', fontsize=20)
plt.legend(selected, loc=2)
plt.show()
```



Justification: A valuable tool for evaluating time series data and identifying patterns and trends is the visualisation of rolling averages and trend lines. In this visualisation, I chose ten sources with high and medium volume and showed their rolling averages over a 7-day period. In order to visualise the overall trend of the data, I attached trend lines to each outlet. This visualisation can provide insights into individual outlet performance and customer behaviour trends. It can assist organisations in making data-driven decisions to optimise sales and improve customer experience, such as altering employee levels or promotional activities.

Explanation: The visualisation displays rolling averages of the daily number of customers for 10 stores(medium and high) over a seven-day period. The rolling average helps to balance out the data and assists in the recognition of trends over time. The daily customer count at each outlet fluctuates throughout time, yet there are significant trends to be seen. The image also contains trend lines for each outlet, which reflect the data's general trend. We can see that the number of consumers at some outlets is increasing while it is falling in others. Businesses can make informed decisions regarding workforce levels, promotional activities, and other things that influence customer behaviour by evaluating trend lines. The visualisation also aids in identifying outliers or unusual behaviour, which may then be studied further to determine the main reason. The visualisation provides a full perspective of individual outlet performance and can assist organisations in making data-driven decisions to optimise their operations.

Visualisation 8: Interactive Histogram for High volume.

```
!pip install hvplot
import pandas as pd
import hvplot.pandas
import holoviews as hv
selected = ['RAN', 'RFY', 'DMN']
plot = data[selected].hvplot.hist(
    frame_height=500, frame_width=500,
    xlabel='outlet sold per day', ylabel='Frequency',
    title='High Volume Outlets',
    alpha=0.5, muted_alpha=0, muted_fill_alpha=0, muted_line_alpha=0,
    tools=['pan', 'box_zoom', 'wheel_zoom', 'undo', 'redo', 'hover', 'save', 'reset']
)
hv.extension('bokeh')
Plot
```

Justification: The interactive histogram is a valuable visualisation tool for investigating the distribution of daily sales at high volume outlets. We can investigate patterns and trends by picking specific stores and observing how sales data is spread. This graphic can assist firms in identifying high-performing shops and optimising inventory management and personnel methods.

Explanation: The interactive histogram shows the daily sales distribution for three high-volume outlets: RAN, RFY, and DMN. The x-axis indicates the number of outlets sold per day, while the y-axis indicates the frequency of those sales. The histogram categorises the data and displays the frequency of sales in each category. We can see the actual sales volume in each bin by hovering over the bars. The interactive histogram is an useful tool for investigating the distribution of sales for each store. We can choose other outlets to compare their distribution, and we can also zoom in and out to analyse the data in greater depth. The visualisation also includes interactive capabilities such as pan, box zoom, and reset, which allow us to adjust the view and explore the data in many ways. the interactive histogram is a fantastic tool for examining the distribution of sales data and exploring patterns and trends. It can assist firms in identifying high-performing locations and optimising their inventory management and personnel practises to maximise revenue.

Critical Review:

My data visualisation coursework included eight visualisations, two of which were interactive, utilizing histogram and heatmap correlation with hvplot and bokeh. I also created segmented barcharts for high, medium, and low volume stores, a line plot to show stores that were closed and newly opened during the time period, a comparative bar chart for high volume stores, radar plots for medium volume stores, a 28-day seasonality plot for high volume stores, and trendlines for both high and medium volume stores. Basically, This report displayed an awareness of the course topic as well as business processes in data visualisation. The use of interactive visualisations, segmentation, and trendlines enhanced the data's depth and clarity. The use of colour coding and labelling assisted in distinguishing between high, medium, and low volume outlets, making the data easier to interpret for the viewer. But, by simplifying the design, some of the visualisations could be improved.

I've learned how to determine which visualisation is better for each component I wish to display. Because we had so many different types of visualisation, it was important to choose the top eight visualisations among all of them.

Conclusions:

- RAN, RFY and DMN are three stores with the most daily customers among all.(we call them high volume in the report)
- We have so many stores as low volume but it doesn't mean that they always have the lowest customers, It means some of them are new outlets which have been opened during the year.(I showed it as visualisation 2)
- We have some stores as low volume which have been closed completely during the year(visualisation 2)
- In our high volume stores, RAN has the most daily customers.
- Overheads(Total cost) for RFY is the highest among three other high volume stores.
- Marketing feature for each Outlet has a strong correlation with Staff and Size of the store(from visualisation 4-heatmap)
- Number of staff has a strong correlation with size of the store and Marketing.(visualisation 4-heatmap)
- There is no correlation with other features for Overheads(total costs) and Daily Customers(visualisation 4-heatmap)
- There Are seven Outlets as Medium volume and the Daily customers for DSA is the highest among them and as we see in visualisation 5 in radar plots there is a good balance among All features
- In BSQ from medium volumes, there is a high difference between overheads and Daily customers, as we can see overheads is much higher and there isn't any profit.(visualisation 5 –radar plot)
- EEC is the store that we can have profit among medium volume stores as the Daily customers are higher than overheads.(visualisation 5-radar plot)
- There are 4 cycles in daily customers for high volume outlets in a 28 days period.