VICTORIA SOLUTIONS
CONSULTING GROUP

Name	Eda Nur Arslan
Contact Number	
Project Title (Example – Week1, Week2, Week3)	Exploratory Data Analysis with Python

Project Guidelines and Rules

1. **Formatting and Submission**
   - **Format:** Use a readable font (e.g., Arial/Times New Roman), size 12, 1.5 line spacing.
   - **Title:** Include Week and Title (Example - Week 1: TravelEase Case Study.)
   - **File Format:** Submit as PDF or Word file to contact@victoriasolutions.co.uk
   - **Page Limit:** 4–5 pages, including the title and references.

2. **Answer Requirements**
   - **Word Count:** Each answer should be 100–150 words; total 800–1,200 words.
   - **Clarity:** Write concise, structured answers with key points.
   - **Tone:** Use formal, professional language.

3. **Content Rules**
   - Answer all questions thoroughly, referencing case study concepts.
   - Use examples where possible (e.g., risk assessment techniques).
   - Break complex answers into bullet points or lists.

4. **Plagiarism Policy**
   - Submit original work; no copy-pasting.
   - Cite external material in a consistent format (e.g., APA, MLA).

5. **Evaluation Criteria**
   - **Understanding:** Clear grasp of business analysis principles.
   - **Application:** Effective use of concepts like cost-benefit analysis and Agile/Waterfall.
   - **Clarity:** Logical, well-structured responses.
   - **Creativity:** Innovative problem-solving and examples.
   - **Completeness:** Answer all questions within the word limit.

6. **Deadlines and Late Submissions**

- **Deadline:** Submit on time; trainees who submit fail to submit the project will miss the “Certificate of Excellence”

7. **Additional Resources**
   - Refer to lecture notes and recommended readings.
   - Contact the instructor or peers for clarifications before the deadline.

START YOUR PROJECT FROM HERE:

Section 1 – Dataset Creation

A small dataset was created using Python and the Pandas library to support exploratory data analysis. The dataset contains seven days of lemonade stand activity, with four key variables: **Day**, **Temperature**, **Sunny**, and **Sales**. The dataset includes both numerical and categorical data, allowing for descriptive statistics and basic visual analyses. Temperature values range from cool to warm days, while the *Sunny* column indicates whether conditions were favorable. Sales values reflect the number of lemonade cups sold each day.

```
data = {
    'Day': [
        'Monday', 'Tuesday', 'Wednesday',
        'Thursday', 'Friday', 'Saturday', 'Sunday'
    ],
    'Temperature': [20, 25, 22, 30, 28, 32, 18],
    'Sunny': ['No', 'Yes', 'No', 'Yes', 'Yes', 'Yes', 'No'],
    'Sales': [30, 50, 35, 70, 60, 80, 20]
}

df = pd.DataFrame(data)
df
```

	Day	Temperature	Sunny	Sales
0	Monday	20	No	30
1	Tuesday	25	Yes	50
2	Wednesday	22	No	35
3	Thursday	30	Yes	70
4	Friday	28	Yes	60
5	Saturday	32	Yes	80
6	Sunday	18	No	20

Fig 1.1 – Dataset Creation

Section 2 – Dataset Inspection

The dataset was inspected to understand its structure and identify key characteristics before performing any analysis.

Using the **df.info()** function, the dataset was confirmed to contain four columns: **Day**, **Temperature**, **Sunny**, and **Sales**, with no missing values and appropriate data types for each variable.

The **df.head()** output allowed the first five rows to be reviewed, confirming that the entries were correctly formatted and consistent with the intended design of the sample data.

Descriptive statistics generated through **df.describe()** provided insight into numerical attributes, such as average temperature and typical sales values.

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7 entries, 0 to 6
Data columns (total 4 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Day          7 non-null      object
 1   Temperature  7 non-null      int64
 2   Sunny        7 non-null      object
 3   Sales        7 non-null      int64
dtypes: int64(2), object(2)
memory usage: 356.0+ bytes
None
```

```
print(df.head())
```

```
         Day  Temperature Sunny  Sales
0     Monday           20    No     30
1    Tuesday           25   Yes     50
2  Wednesday           22    No     35
3   Thursday           30   Yes     70
4     Friday           28   Yes     60
```

```
print(df.describe())
```

```
       Temperature      Sales
count     7.000000   7.000000
mean     25.000000  49.285714
std       5.259911  22.065919
min      18.000000  20.000000
25%      21.000000  32.500000
50%      25.000000  50.000000
75%      29.000000  65.000000
max      32.000000  80.000000
```

Fig 2.1 – Dataset Inspection

Section 3 – Data Visualisation

Three visualisations were created to explore key relationships within the dataset.

First, a **histogram** was used to display the distribution of daily lemonade sales. This chart showed how frequently different sales levels occurred across the week, allowing an initial view of overall sales patterns.

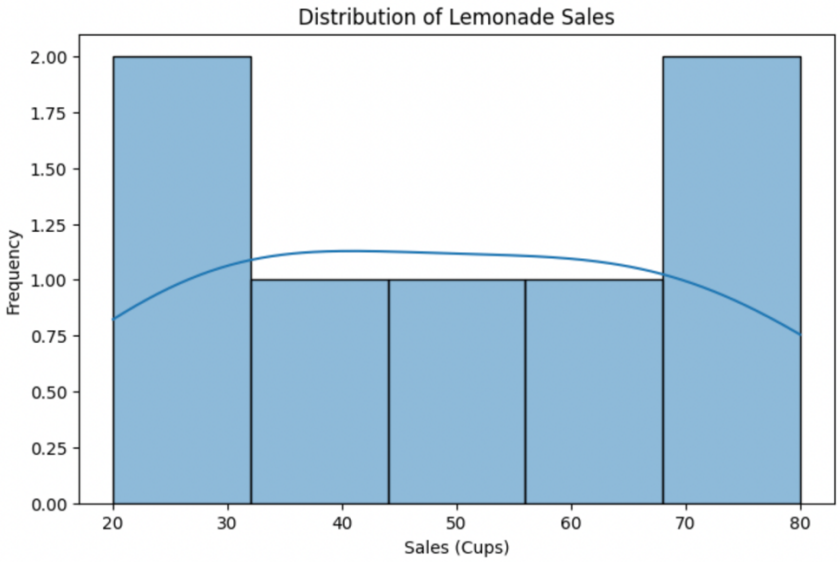

Fig 3.1 - Histogram

Next, a **scatter plot** was generated to examine the relationship between **Temperature** and **Sales**, with points coloured by the **Sunny** variable. This visualisation made it possible to observe how higher temperatures were associated with increased sales, and how weather conditions influenced purchasing behaviour.

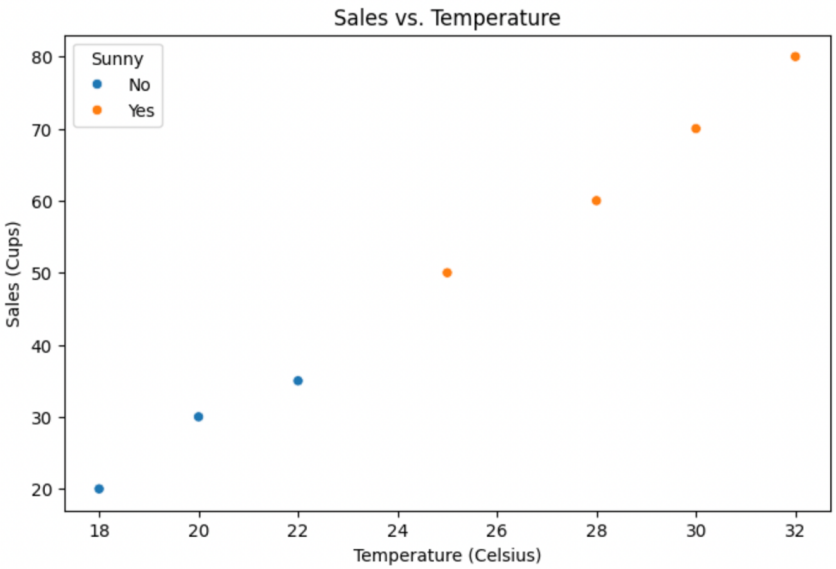

Fig 3.2 – Scatter Plot

Finally, a **bar chart** compared the average sales on sunny versus non-sunny days. This chart clearly highlighted higher average sales during sunny weather, confirming the influence of environmental factors. Together, these visualisations provided an intuitive and informative overview of the dataset's main trends.

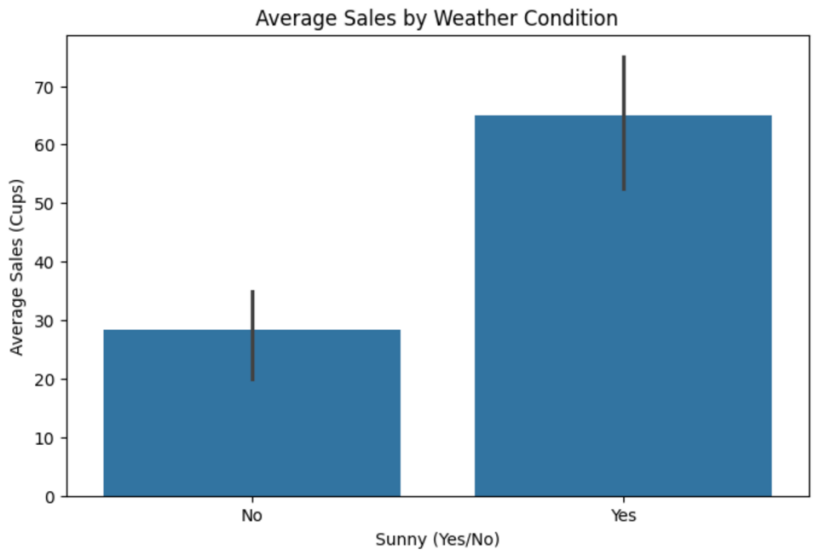

Fig 3.3 – Bar Chart

Section 4 – Summary of Findings

The analysis of the dataset revealed several clear patterns based on both statistical outputs and visualisations.

A positive correlation was found between **Temperature** and **Sales**, indicating that higher temperatures were generally associated with increased lemonade sales.

This relationship was supported by the scatter plot, where higher sales values appeared more often on warmer days.

```
correlation = df['Temperature'].corr(df['Sales'])
print(f"Correlation between Temperature and Sales: {correlation:.2f}")
```

```
Correlation between Temperature and Sales: 1.00
```

Fig 4.1 - Correlation

The comparison of sunny and non-sunny conditions showed that **sunny days consistently produced higher average sales**, suggesting that weather plays a meaningful role in customer behaviour.
The histogram further demonstrated that sales were not evenly distributed, with certain values occurring more frequently, especially on days with favourable weather.

Overall, the findings suggest that both temperature and sunshine have a noticeable impact on lemonade sales, with warmer and sunnier conditions contributing to stronger daily performance.