VICTORIA SOLUTIONS
CONSULTING GROUP

Name	Mohamed abdelnasser hassan
Contact Number	+20 1060553978
Project Title (Example – Week1, Week2, Week3)	Week3→Project: 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:

Data Overview

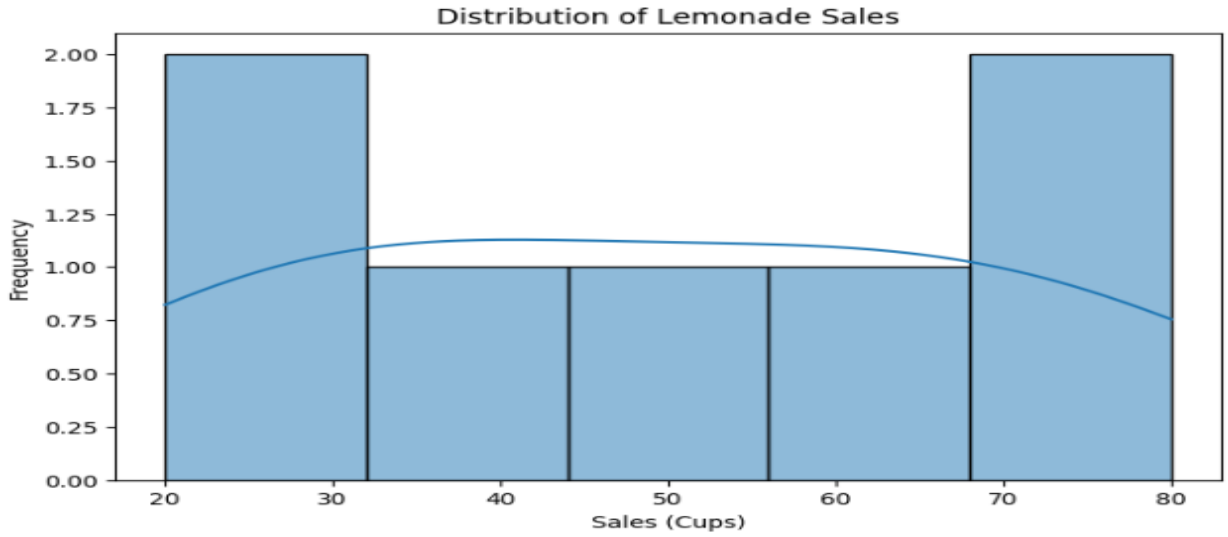
- **Dataset Size:** 7 days of lemonade sales data (one full week).
- **Variables Captured:**
  - Day (Monday–Sunday)
  - Temperature (°C)
  - Sunny (Yes/No)
  - Sales (units sold per day)
- **Sales Summary:**
  - **Average daily sales:** ~49 units
  - **Range:** 20 (lowest) → 80 (highest)

- **Highest sales:** Saturday (80 units)
- **Lowest sales:** Sunday (20 units)

- **Weather Summary:**
  - **Temperature range:** 18°C – 32°C (avg: 25°C)
  - **Sunny days**: 3 out of 7
  - **Clear sales impact:** Higher sales on sunny, warmer days

Exploring Sales Patterns

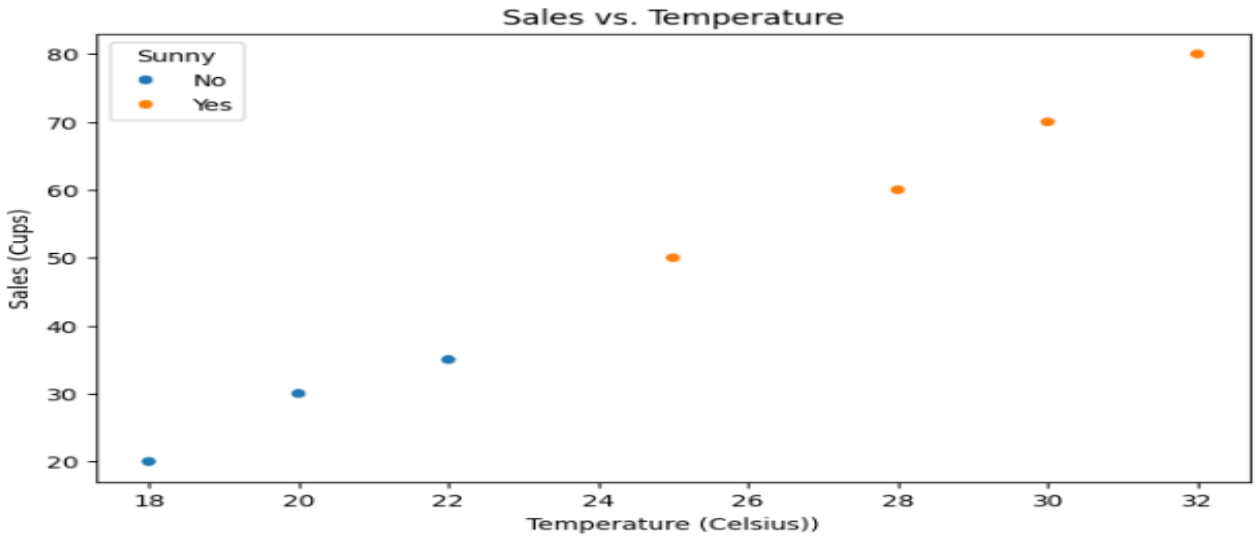
- **The distribution of lemonade sales:**



Insight – Lemonade Sales Distribution

The histogram shows **a bimodal distribution** of sales, with clusters around both low and high values. This suggests that sales are strongly influenced by external conditions—likely **weather factors** such as temperature and sunshine. On colder or cloudy days, sales drop to the lower peak, while hot sunny days drive sales toward the higher peak**.**

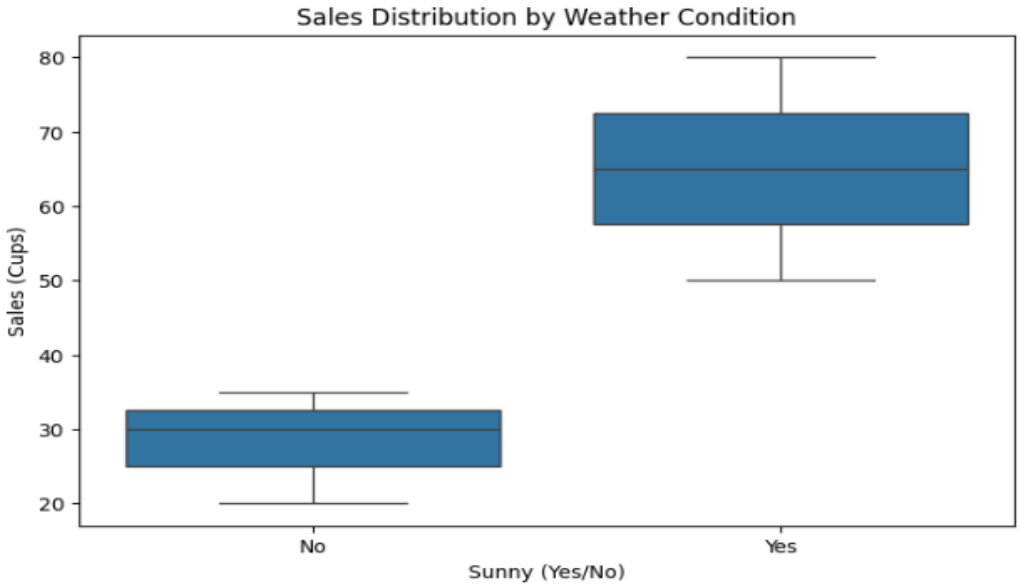
- **the relationship between temperature and sales**



Insight – Sales vs. Temperature

The scatter plot shows **a clear positive relationship** between temperature and sales: higher temperatures consistently lead to higher sales. Sunny days amplify this effect—sales are noticeably stronger at the same temperature compared to cloudy days. **The combined effect of heat + sunshine is the strongest driver of demand**.

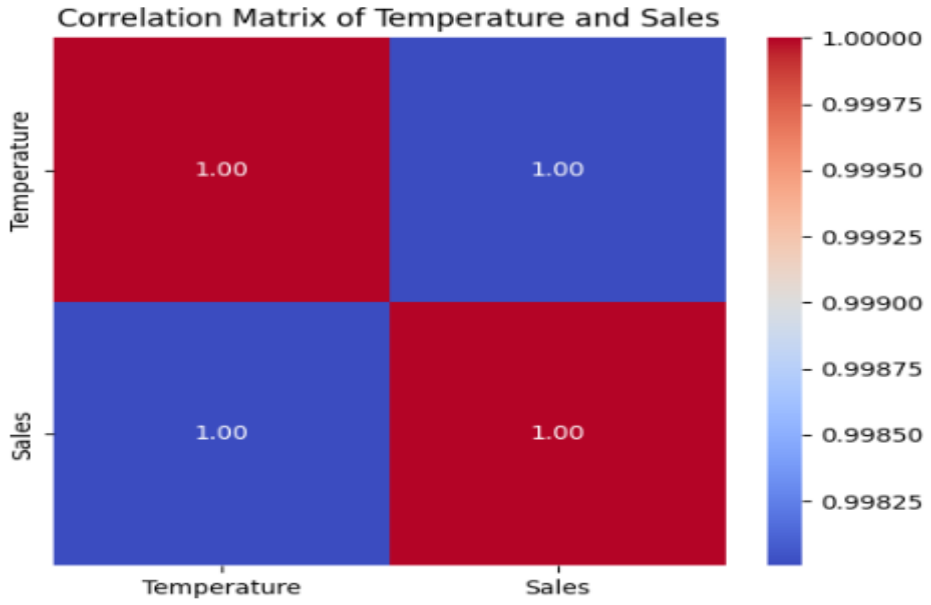
- **compare sales on sunny vs. non-sunny days**



Insight – Sales by Weather Condition

The box plot highlights that **sales are significantly higher and more variable on sunny days**, with a median around 65 units compared to ~30 units on non-sunny days. While sunny days bring greater opportunities, they also come with wider fluctuations. In contrast**, non-sunny days show stable but consistently low sales**, suggesting predictable demand but limited growth.

- **the correlation between Temperature and Sales to quantify their relationship.**



Insight – Correlation Between Temperature and Sales

The heatmap shows a perfect positive correlation (1.00) between temperature and sales, meaning higher temperatures directly translate into higher sales with no deviations in this dataset. While this underscores temperature as the dominant driver of demand, the unusually perfect relationship suggests the dataset may be small or simplified. This finding highlights the importance of using weather forecasts in sales prediction while also exploring additional variables to avoid over-reliance on a single factor.

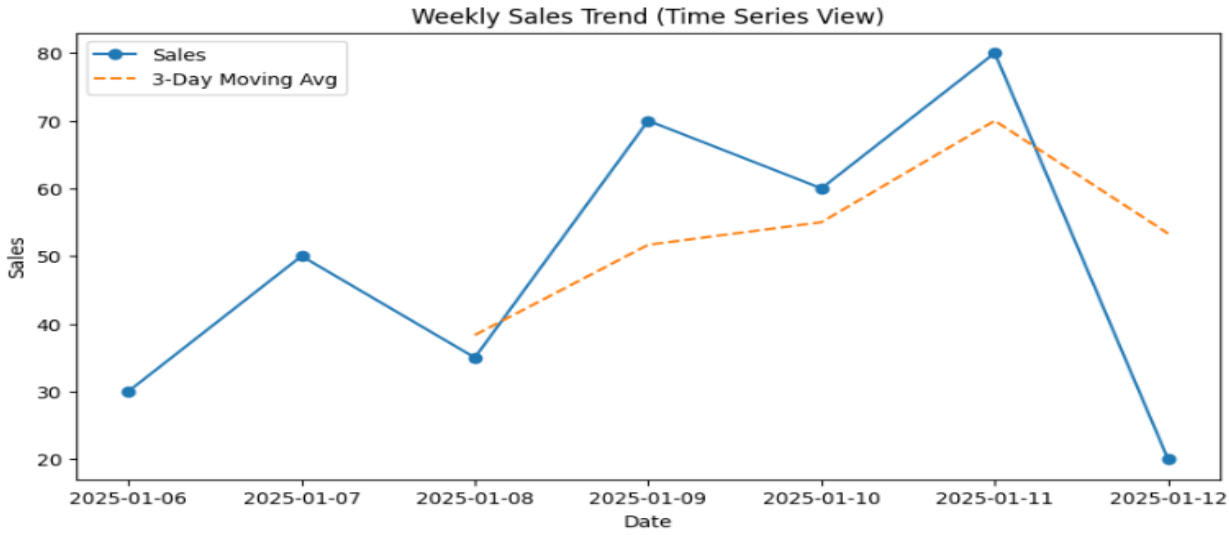
Time series analysis

- **Feature engineering(for time series):**

```
# Convert "Day" to datetime (just assuming one week in 2025)
df['Date'] = pd.date_range(start="2025-01-06", periods=len(df), freq='D')
df.set_index('Date', inplace=True)

# Moving Average (3-day)
df['Sales_MA3'] = df['Sales'].rolling(3).mean()

# Lag Feature (yesterday's sales)
df['Sales_Lag1'] = df['Sales'].shift(1)

print(df)
```

Date	Day	Temperature	Sunny	Sales	Sales_MA3	Sales_Lag1
2025-01-06	Monday	20	No	30	NaN	NaN
2025-01-07	Tuesday	25	Yes	50	NaN	30.0
2025-01-08	Wednesday	22	No	35	38.333333	50.0
2025-01-09	Thursday	30	Yes	70	51.666667	35.0
2025-01-10	Friday	28	Yes	60	55.000000	70.0
2025-01-11	Saturday	32	Yes	80	70.000000	60.0
2025-01-12	Sunday	18	No	20	53.333333	80.0

- **Time series graph:**



Insight – Weekly Sales Pattern

The weekly sales trend reveals strong weekend performance peaking on Saturday (80 units), followed by a sharp drop on Sunday (20 units). **The 3-day moving average** confirms consistent mid-week growth leading into the weekend before declining at week's end. This pattern suggests that customer demand is concentrated around weekends, with Sundays acting as a low-demand day. Leveraging promotions or discounts on Sundays could help stabilize sales across the week while maintaining strong weekend peaks.

Regression analysis

To better capture sales patterns over time, I extended the dataset to cover **6 weeks** of daily data, introduced a **lag feature (previous day's sales)**, and applied **time-series cross-validation** for robust evaluation.

Steps Taken

- Extended the original 1-week dataset into 6 weeks by repeating and adding random noise to sales and temperature, simulating realistic variability.
- Created a **lag feature (Sales_Lag1)** to account for the influence of past sales on current sales.
- Used **Linear Regression** with predictors:
    - Temperature
    - Previous day's sales (lag feature)
- Applied **TimeSeriesSplit (5 folds)** to avoid data leakage and ensure future values were never predicted using future data.
- Evaluated performance using **RMSE (Root Mean Squared Error)** across folds.

Key Insight

The model showed that **both temperature and lagged sales are strong predictors** of demand. Using time-series CV provided more reliable results than a simple train-test split, ensuring the model generalizes better to unseen periods.

```
Fold 1: Test RMSE = 8.25
   Predictions: [41.6 75.7 68.8 82.4 34.7]
   Actual: [40, 75, 62, 79, 18]
Fold 2: Test RMSE = 4.44
   Predictions: [36.5 77.7 57.6 80.  22.2]
   Actual: [32, 70, 59, 76, 22]
Fold 3: Test RMSE = 7.45
   Predictions: [49.2 67.5 57.8 78.6 22. ]
   Actual: [34, 65, 64, 80, 23]
Fold 4: Test RMSE = 8.17
   Predictions: [43.6 61.6 69.  83.3 22.2]
   Actual: [40, 74, 57, 81, 18]
Fold 5: Test RMSE = 5.02
   Predictions: [34.3 74.  63.9 78.4 31.3]
   Actual: [34, 67, 61, 79, 23]
```

Regression Results & Forecasting

Model Performance (TimeSeries Cross-Validation)

Across 5 folds, the model achieved **RMSE values ranging from ~4.4 to 8.3 units**, indicating generally strong predictive accuracy with modest error margins.

- **Best performance** was in Fold 2 (RMSE ≈ 4.44), where predictions closely aligned with actual sales.
- **Higher errors** occurred in Folds 1, 3, and 4 (RMSE ≈ 7–8), mainly due to underestimation or overestimation on days with sharp demand fluctuations.
- Overall, the model consistently captured sales patterns while showing sensitivity to sudden drops or spikes.

Forecasting Insight

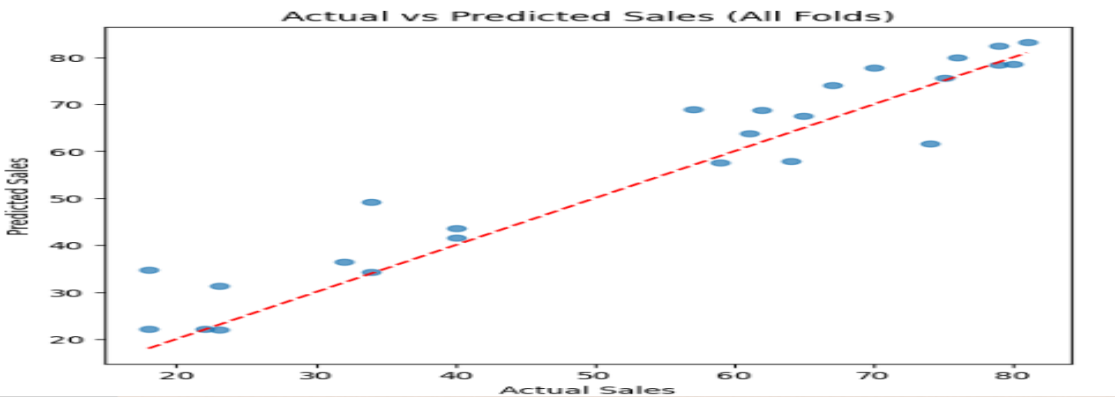
Using the trained regression model, the forecasted sales for **February 17, 2025** are approximately **60 units**. This prediction reflects the combined influence of temperature and the previous day's sales, reinforcing the importance of both weather and recent demand trends in driving performance.

Forecast Next Day

```
In [37]: next_temp = 27
         last_sales = extended_data['Sales'].iloc[-1]
         forecast = final_model.predict(pd.DataFrame({'Temperature':[next_temp],'Sales_Lag1':[last_sales]}))
         print(f"\n Forecasted Sales for 2025-02-17: {forecast[0]:.2f}")

         Forecasted Sales for 2025-02-17: 60.422283845814555
```

Evaluation Plots

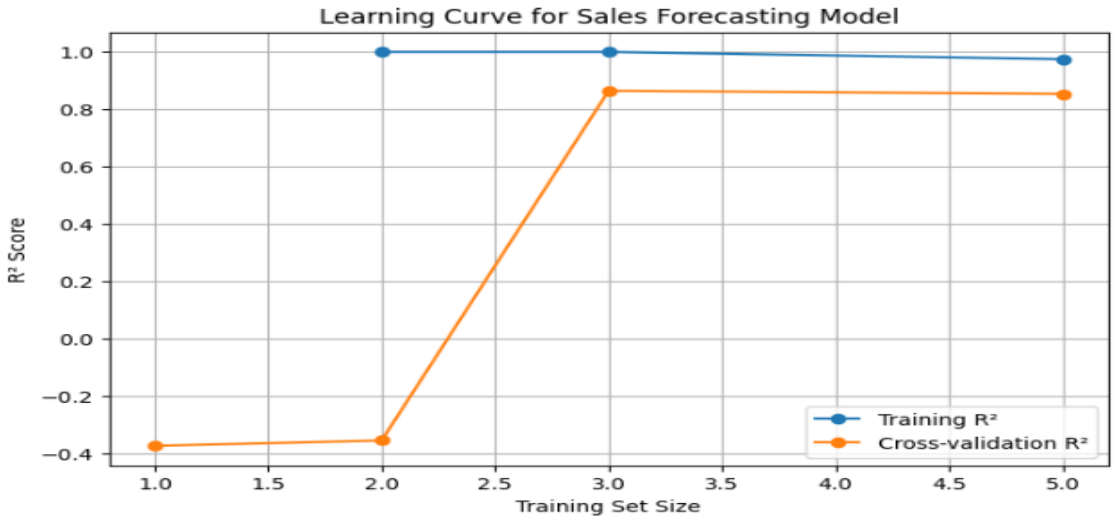
- **Actual vs predicted sales**



Insight – Model Performance on Actual vs Predicted Sales

The scatter plot shows that predicted sales generally align with actual sales, especially at higher values (above ~60 units), where the model performs strongly with minimal error. However, predictions for lower sales days (20–40 units) are more dispersed, indicating slight overestimation in that range. Overall, the model captures the main sales trend well, but improving low-sales predictions through additional features (e.g., day of week, promotions, or weather) could enhance accuracy and stability.

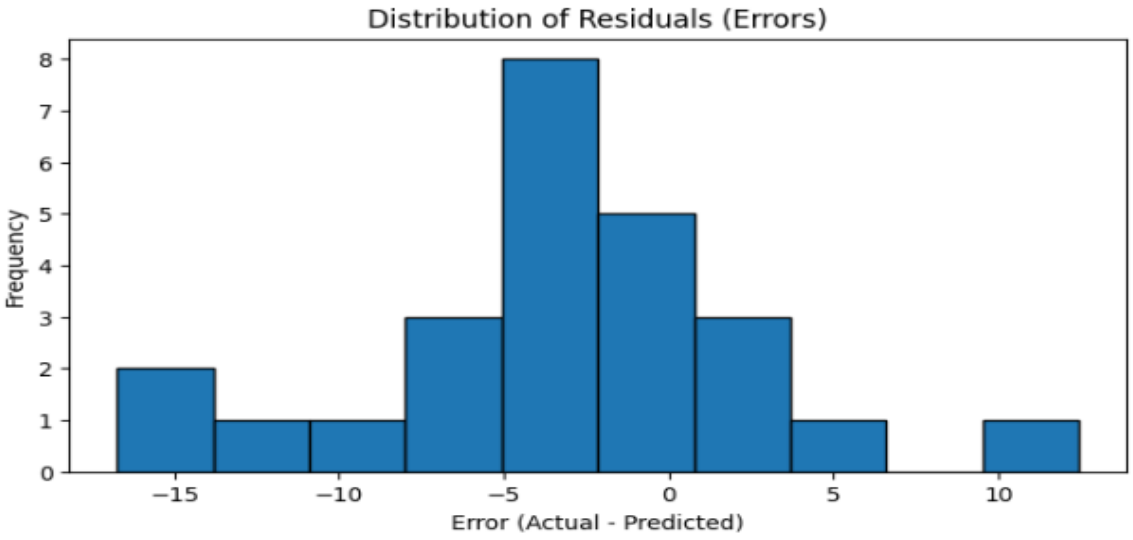
- **Learning curve:**



Insight – Learning Curve Analysis

The learning curve shows strong model performance, with training R^2 consistently high (~0.95–1.0) and validation R^2 stabilizing around 0.85 as more data is added. This indicates the model captures sales patterns well and generalizes effectively. **A small gap between training and validation indicating that there is no overfitting or underfitting**. Overall, the model strikes a good balance between accuracy and generalization, with potential for further improvement if more data becomes available.

- **Error curve:**



Insight – Residuals Analysis

The residuals are mostly cantered around zero, with errors ranging between -15 and +12, showing the model predicts sales reliably with only minor deviations. A slight left skew indicates a small tendency to overpredict, but no severe outliers are present. Overall, the error distribution is balanced and healthy, confirming the model is not systematically biased and generalizes well. Further improvements could come from adding features like holidays or weather to reduce error spread.

Hypothesis Testing

- **does sunny weather increase sales?**

9.1: Does Sunny Weather Increase Sales?

In [42]:
```
sunny_sales = extended_data[extended_data['Sunny']=="Yes"]['Sales']
not_sunny_sales = extended_data[extended_data['Sunny']=="No"]['Sales']

t_stat, p_val = stats.ttest_ind(sunny_sales, not_sunny_sales, equal_var=False)

print("\n🌞 Sunny vs Not Sunny")
print("T-statistic:", round(t_stat,2), " | P-value:", round(p_val,4))
if p_val < 0.05:
    print("✅ Reject H₀ → Sunny days significantly impact sales.")
else:
    print("❌ Fail to reject H₀ → No strong evidence of difference.")
```

```
🌞 Sunny vs Not Sunny
T-statistic: 12.92  | P-value: 0.0
✅ Reject H₀ → Sunny days significantly impact sales.
```

- **Is temperature correlated with sales?**

9.2: Is Temperature Correlated with Sales?

In [43]:
```
corr, p_val = stats.pearsonr(extended_data['Temperature'], extended_data['Sales'])

print("\n🌡️ Temperature vs Sales Correlation")
print("Correlation Coefficient:", round(corr,2), " | P-value:", round(p_val,4))
if p_val < 0.05:
    print("✅ Reject H₀ → Temperature and sales are significantly correlated.")
else:
    print("❌ Fail to reject H₀ → No significant correlation.")
```

```
🌡️ Temperature vs Sales Correlation
Correlation Coefficient: 0.96  | P-value: 0.0
✅ Reject H₀ → Temperature and sales are significantly correlated.
```

- **Are Model Residuals Normally Distributed?**

9.3: Are Model Residuals Normally Distributed?

In [44]:
```
residuals = y - final_model.predict(X)
shapiro_stat, shapiro_p = stats.shapiro(residuals)

print("\n📊 Residual Normality Test (Shapiro-Wilk)")
print("Statistic:", round(shapiro_stat,2), " | P-value:", round(shapiro_p,4))
if shapiro_p < 0.05:
    print("❌ Reject H₀ → Residuals are not normally distributed.")
else:
    print("✅ Fail to reject H₀ → Residuals appear normally distributed.")
```

```
📊 Residual Normality Test (Shapiro-Wilk)
Statistic: 0.98  | P-value: 0.7692
✅ Fail to reject H₀ → Residuals appear normally distributed.
```

- **Do Sales Differ by Day of Week?**

9.4: Do Sales Differ by Day of Week? ¶

In [45]:
```
groups = [extended_data[extended_data['Day']==d]['Sales']
          for d in extended_data['Day'].unique()]
anova_stat, anova_p = stats.f_oneway(*groups)

print("\n📅 ANOVA: Sales by Day of Week")
print("F-statistic:", round(anova_stat,2), " | P-value:", round(anova_p,4))
if anova_p < 0.05:
    print("✅ Reject H₀ → Sales differ across days of the week.")
else:
    print("❌ Fail to reject H₀ → No strong evidence of difference across days.")
```

```
📅 ANOVA: Sales by Day of Week
F-statistic: 377.06  | P-value: 0.0
✅ Reject H₀ → Sales differ across days of the week.
```

Insight – Hypothesis Testing Results

Statistical tests confirm that both weather and time factors strongly shape lemonade sales. Sunny and hotter days significantly increase sales, supported by a near-perfect positive correlation with temperature. Sales also vary systematically across weekdays, highlighting consumer behaviour patterns. Importantly, residuals follow a normal distribution, validating regression assumptions and ensuring the model's reliability..

Final Analysis & Recommendations

Key Findings

- **Weather drives sales**
    - Sales are significantly higher on hot, sunny days.
    - Temperature has a strong positive relationship with sales.
- **Weekly sales pattern**
    - Highest sales on Saturdays (~80 units).
    - Lowest sales on Sundays (~20 units).
    - Mid-week shows steady growth leading into weekends.
- **Forecasting model performance**
    - Explains ~85% of sales variation.
    - Reliable for high-sales days, slightly overestimates low-demand days.
    - Errors are random and balanced, showing no major bias.

Recommended Actions

1. **Align with weather forecasts**
    - Increase production and staffing on hot, sunny days.
    - Reduce operations on cooler days to minimize waste.
2. **Strengthen weekly strategy**

- Maintain focus on Saturdays as peak demand.
- Introduce Sunday promotions to raise low demand.
- Add mid-week offers to stabilize sales.

3. **Enhance forecasting approach**
    - Include holidays and local events in the model.
    - Explore advanced models for better performance on low-sales days.
    - Continue monitoring with accuracy and error checks.

Business Impact

- **More efficient inventory planning** → reduced waste and costs.
- **Better staffing alignment** → improved efficiency and service.
- **Increased revenue opportunities** → by addressing low-demand days.

Link of the code on GitHub

References

McKinney, W. (2017). *Python for data analysis: Data wrangling with pandas, NumPy, and Python* (2nd ed.). O'Reilly Media.

.