Assignment4

March 22, 2025

[71]:
```python
import pandas as pd
import numpy as np
```

[72]:
```python
df = pd.read_csv('sales.csv' , encoding = 'latin-1')
```

[73]:
```python
df.head()
```

[73]:
```
   ORDERNUMBER  QUANTITYORDERED  PRICEEACH  ORDERLINENUMBER    SALES  \
0        10107               30      95.70                2  2871.00
1        10121               34      81.35                5  2765.90
2        10134               41      94.74                2  3884.34
3        10145               45      83.26                6  3746.70
4        10159               49     100.00               14  5205.27

         ORDERDATE   STATUS  QTR_ID  MONTH_ID  YEAR_ID  …  \
0   2/24/2003 0:00  Shipped       1         2     2003  …
1    5/7/2003 0:00  Shipped       2         5     2003  …
2    7/1/2003 0:00  Shipped       3         7     2003  …
3   8/25/2003 0:00  Shipped       3         8     2003  …
4  10/10/2003 0:00  Shipped       4        10     2003  …

                  ADDRESSLINE1  ADDRESSLINE2           CITY STATE  \
0        897 Long Airport Avenue           NaN            NYC    NY
1             59 rue de l'Abbaye           NaN          Reims   NaN
2  27 rue du Colonel Pierre Avia           NaN          Paris   NaN
3             78934 Hillside Dr.           NaN       Pasadena    CA
4                7734 Strong St.           NaN  San Francisco    CA

  POSTALCODE COUNTRY TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME DEALSIZE
0      10022     USA       NaN              Yu             Kwai    Small
1      51100  France      EMEA         Henriot             Paul    Small
2      75508  France      EMEA        Da Cunha           Daniel   Medium
3      90003     USA       NaN           Young            Julie   Medium
4        NaN     USA       NaN           Brown            Julie   Medium

[5 rows x 25 columns]
```

[74]:
```python
df.describe(include='all')
```

[74]:
```
        ORDERNUMBER  QUANTITYORDERED    PRICEEACH  ORDERLINENUMBER  \
count   2823.000000      2823.000000  2823.000000      2823.000000
unique          NaN              NaN          NaN              NaN
top             NaN              NaN          NaN              NaN
freq            NaN              NaN          NaN              NaN
mean   10258.725115        35.092809    83.658544         6.466171
std       92.085478         9.741443    20.174277         4.225841
min    10100.000000         6.000000    26.880000         1.000000
25%    10180.000000        27.000000    68.860000         3.000000
50%    10262.000000        35.000000    95.700000         6.000000
75%    10333.500000        43.000000   100.000000         9.000000
max    10425.000000        97.000000   100.000000        18.000000

              SALES        ORDERDATE   STATUS       QTR_ID     MONTH_ID  \
count   2823.000000             2823     2823  2823.000000  2823.000000
unique          NaN              252        6          NaN          NaN
top             NaN  11/14/2003 0:00  Shipped          NaN          NaN
freq            NaN               38     2617          NaN          NaN
mean    3553.889072              NaN      NaN     2.717676     7.092455
std     1841.865106              NaN      NaN     1.203878     3.656633
min      482.130000              NaN      NaN     1.000000     1.000000
25%     2203.430000              NaN      NaN     2.000000     4.000000
50%     3184.800000              NaN      NaN     3.000000     8.000000
75%     4508.000000              NaN      NaN     4.000000    11.000000
max    14082.800000              NaN      NaN     4.000000    12.000000

           YEAR_ID  …          ADDRESSLINE1  ADDRESSLINE2    CITY STATE  \
count   2823.00000  …                  2823           302    2823  1337
unique         NaN  …                    92             9      73    16
top            NaN  …  C/ Moralzarzal, 86       Level 3  Madrid    CA
freq           NaN  …                   259            55     304   416
mean    2003.81509  …                   NaN           NaN     NaN   NaN
std        0.69967  …                   NaN           NaN     NaN   NaN
min     2003.00000  …                   NaN           NaN     NaN   NaN
25%     2003.00000  …                   NaN           NaN     NaN   NaN
50%     2004.00000  …                   NaN           NaN     NaN   NaN
75%     2004.00000  …                   NaN           NaN     NaN   NaN
max     2005.00000  …                   NaN           NaN     NaN   NaN

       POSTALCODE COUNTRY TERRITORY CONTACTLASTNAME CONTACTFIRSTNAME DEALSIZE
count        2747    2823      1749            2823             2823     2823
unique         73      19         3              77               72        3
top         28034     USA      EMEA          Freyre            Diego   Medium
freq          259    1004      1407             259              259     1384
mean          NaN     NaN       NaN             NaN              NaN      NaN
```

```
std            NaN     NaN       NaN             NaN              NaN      NaN
min            NaN     NaN       NaN             NaN              NaN      NaN
25%            NaN     NaN       NaN             NaN              NaN      NaN
50%            NaN     NaN       NaN             NaN              NaN      NaN
75%            NaN     NaN       NaN             NaN              NaN      NaN
max            NaN     NaN       NaN             NaN              NaN      NaN

[11 rows x 25 columns]
```

[75]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2823 entries, 0 to 2822
Data columns (total 25 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   ORDERNUMBER       2823 non-null   int64
 1   QUANTITYORDERED   2823 non-null   int64
 2   PRICEEACH         2823 non-null   float64
 3   ORDERLINENUMBER   2823 non-null   int64
 4   SALES             2823 non-null   float64
 5   ORDERDATE         2823 non-null   object
 6   STATUS            2823 non-null   object
 7   QTR_ID            2823 non-null   int64
 8   MONTH_ID          2823 non-null   int64
 9   YEAR_ID           2823 non-null   int64
 10  PRODUCTLINE       2823 non-null   object
 11  MSRP              2823 non-null   int64
 12  PRODUCTCODE       2823 non-null   object
 13  CUSTOMERNAME      2823 non-null   object
 14  PHONE             2823 non-null   object
 15  ADDRESSLINE1      2823 non-null   object
 16  ADDRESSLINE2      302 non-null    object
 17  CITY              2823 non-null   object
 18  STATE             1337 non-null   object
 19  POSTALCODE        2747 non-null   object
 20  COUNTRY           2823 non-null   object
 21  TERRITORY         1749 non-null   object
 22  CONTACTLASTNAME   2823 non-null   object
 23  CONTACTFIRSTNAME  2823 non-null   object
 24  DEALSIZE          2823 non-null   object
dtypes: float64(2), int64(7), object(16)
memory usage: 551.5+ KB
```

[76]:
```
df.shape
```

[76]: (2823, 25)

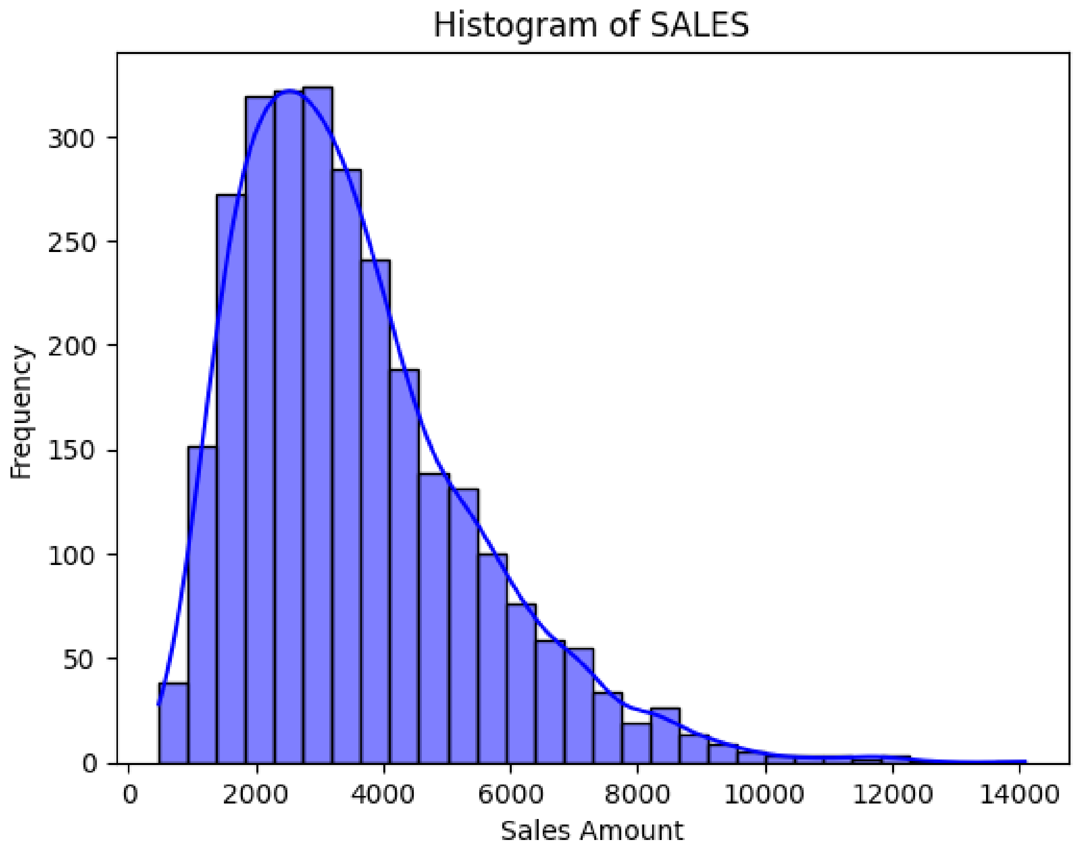
[77]:
```
df.isnull().sum()
```

[77]:
```
ORDERNUMBER            0
QUANTITYORDERED        0
PRICEEACH              0
ORDERLINENUMBER        0
SALES                  0
ORDERDATE              0
STATUS                 0
QTR_ID                 0
MONTH_ID               0
YEAR_ID                0
PRODUCTLINE            0
MSRP                   0
PRODUCTCODE            0
CUSTOMERNAME           0
PHONE                  0
ADDRESSLINE1           0
ADDRESSLINE2        2521
CITY                   0
STATE               1486
POSTALCODE            76
COUNTRY                0
TERRITORY           1074
CONTACTLASTNAME        0
CONTACTFIRSTNAME       0
DEALSIZE               0
dtype: int64
```

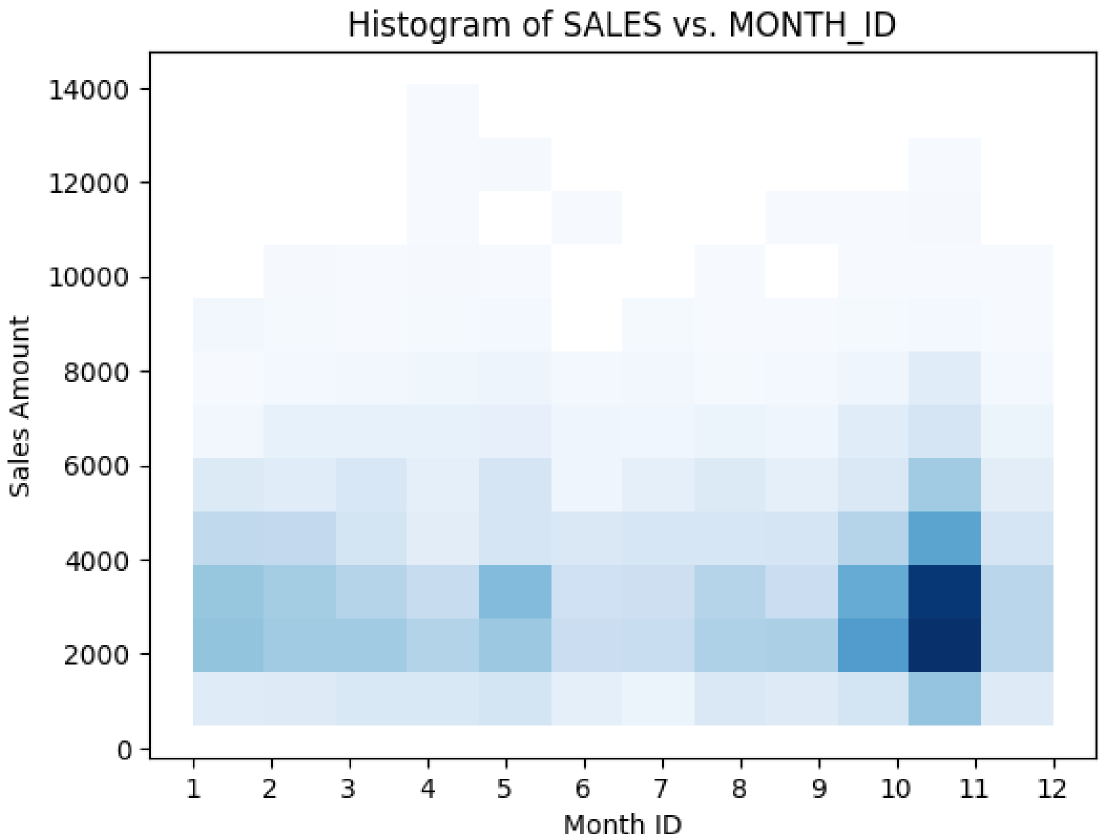
[78]:
```
df = df.drop('STATE', axis=1)
df = df.drop('POSTALCODE', axis=1)
df = df.drop('TERRITORY', axis=1)
df = df.drop('ADDRESSLINE2', axis=1)
```

[79]:
```
import matplotlib.pyplot as plt
import seaborn as sns
```

[80]:
```
sns.histplot(df["SALES"], bins=30, kde=True, color="blue")
plt.title("Histogram of SALES")
plt.xlabel("Sales Amount")
plt.ylabel("Frequency")
plt.show()
```

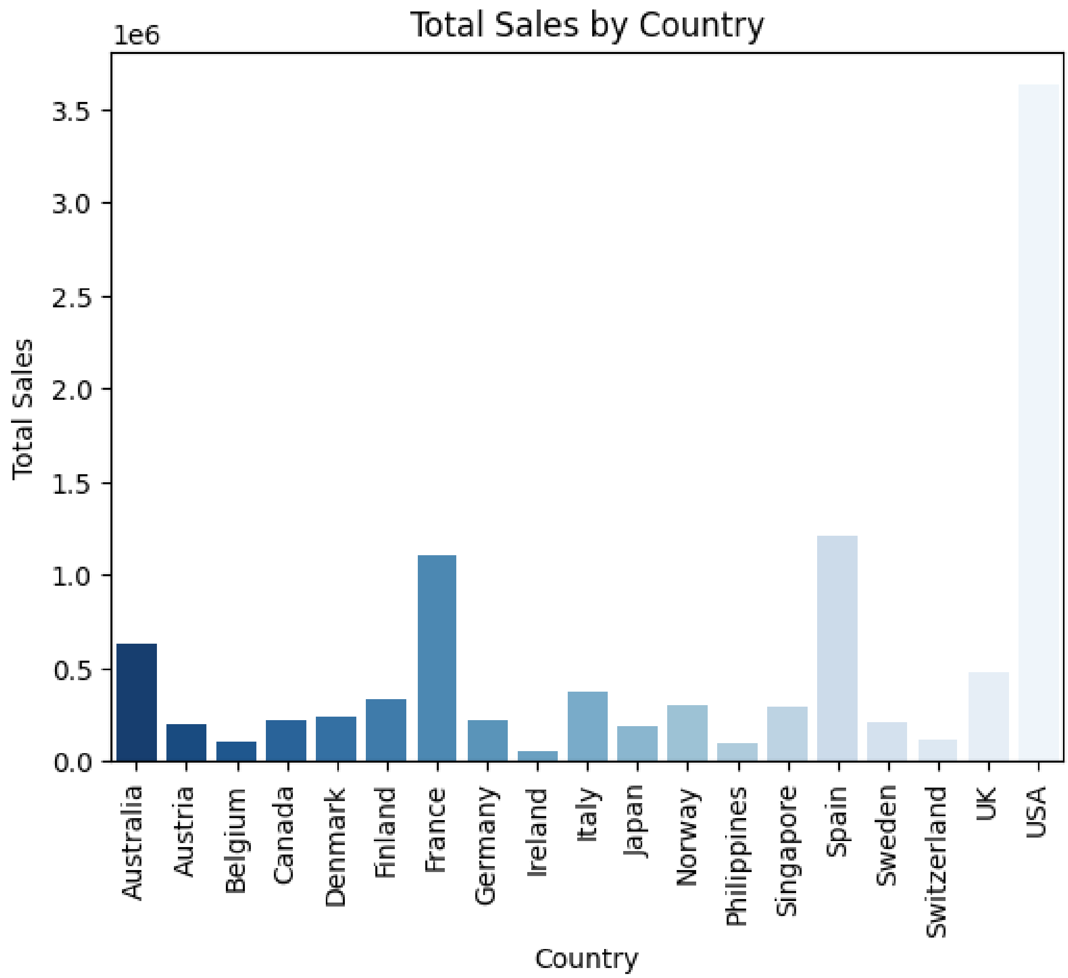


[81]:
```
sns.histplot(x=df["MONTH_ID"], y=df["SALES"], bins=12, kde=True, cmap="Blues")
plt.title("Histogram of SALES vs. MONTH_ID")
plt.xlabel("Month ID")
plt.ylabel("Sales Amount")
plt.xticks(range(1, 13))
plt.show()
```

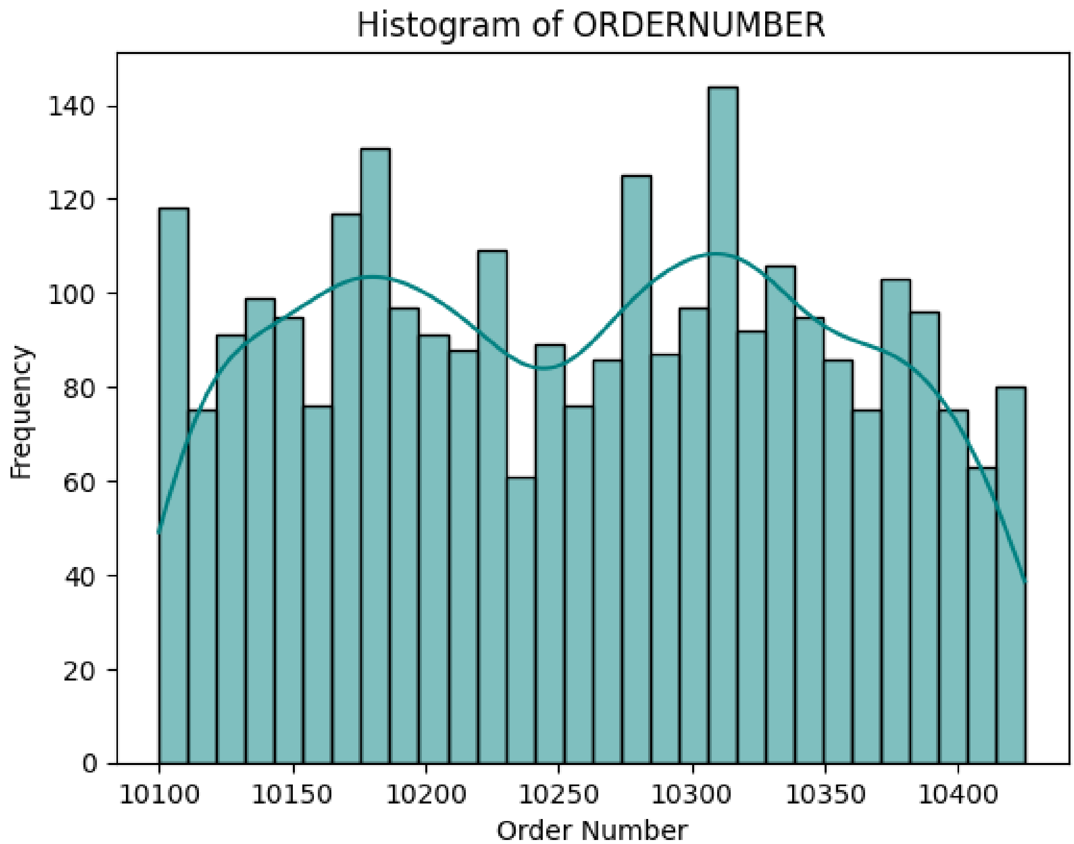


[82]:
```
sns.barplot(x=df.groupby("COUNTRY")["SALES"].sum().index,
            y=df.groupby("COUNTRY")["SALES"].sum().values,
            palette="Blues_r")
plt.xticks(rotation=90)
plt.title("Total Sales by Country")
plt.xlabel("Country")
plt.ylabel("Total Sales")
plt.show()
```

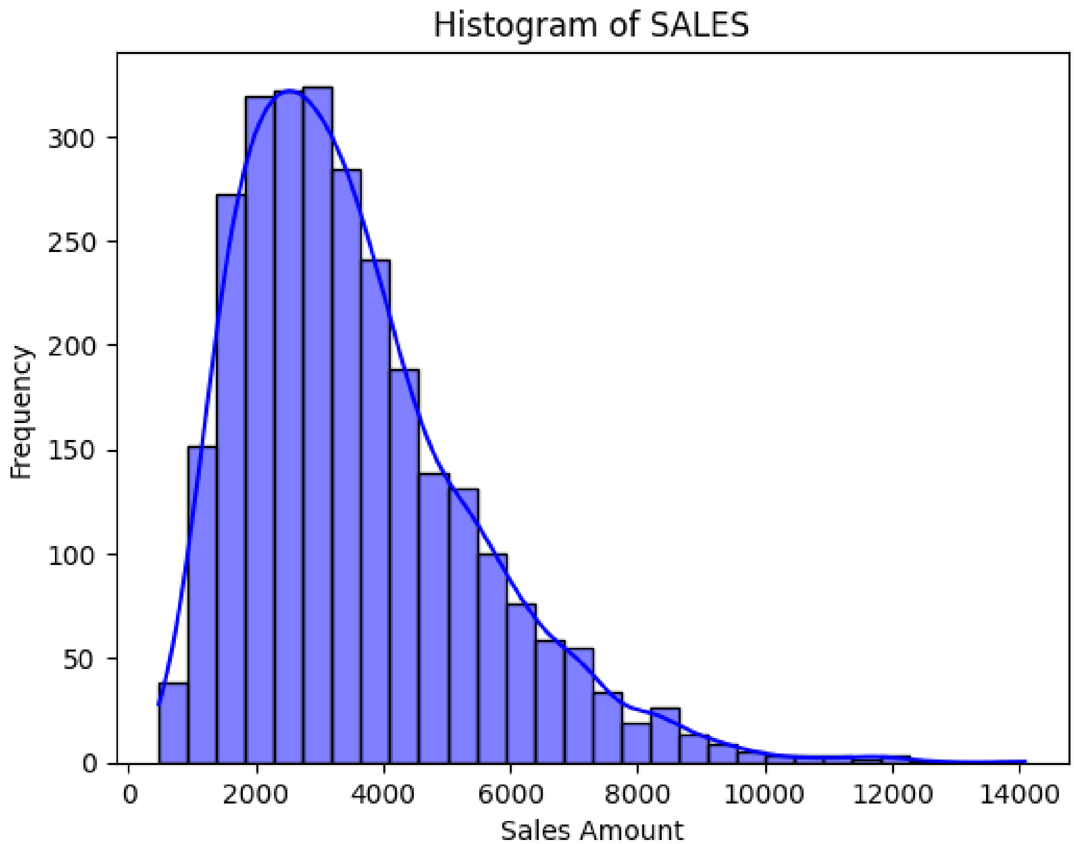
```
<ipython-input-82-6da0283484d0>:1: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in
v0.14.0. Assign the `x` variable to `hue` and set `legend=False` for the same
effect.

  sns.barplot(x=df.groupby("COUNTRY")["SALES"].sum().index,
```



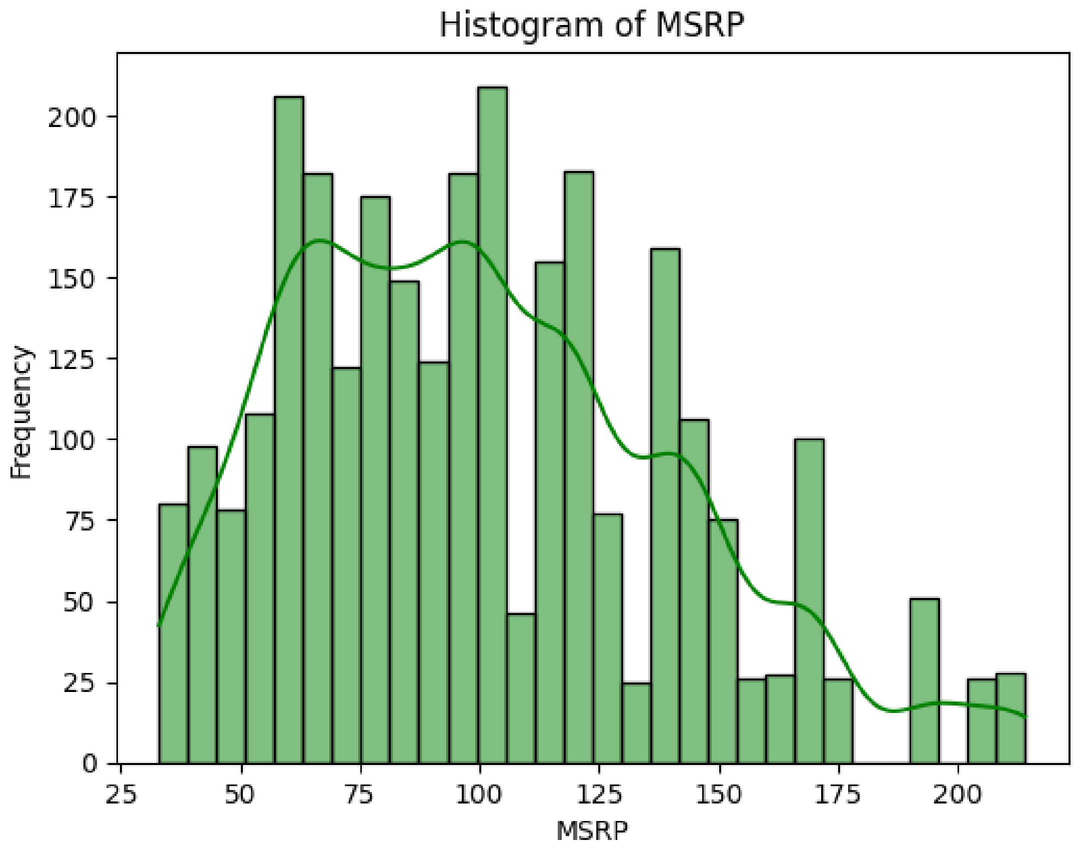
[83]:
```
sns.histplot(df["ORDERNUMBER"], bins=30, kde=True, color="teal")
plt.title("Histogram of ORDERNUMBER")
plt.xlabel("Order Number")
plt.ylabel("Frequency")
plt.show()
```



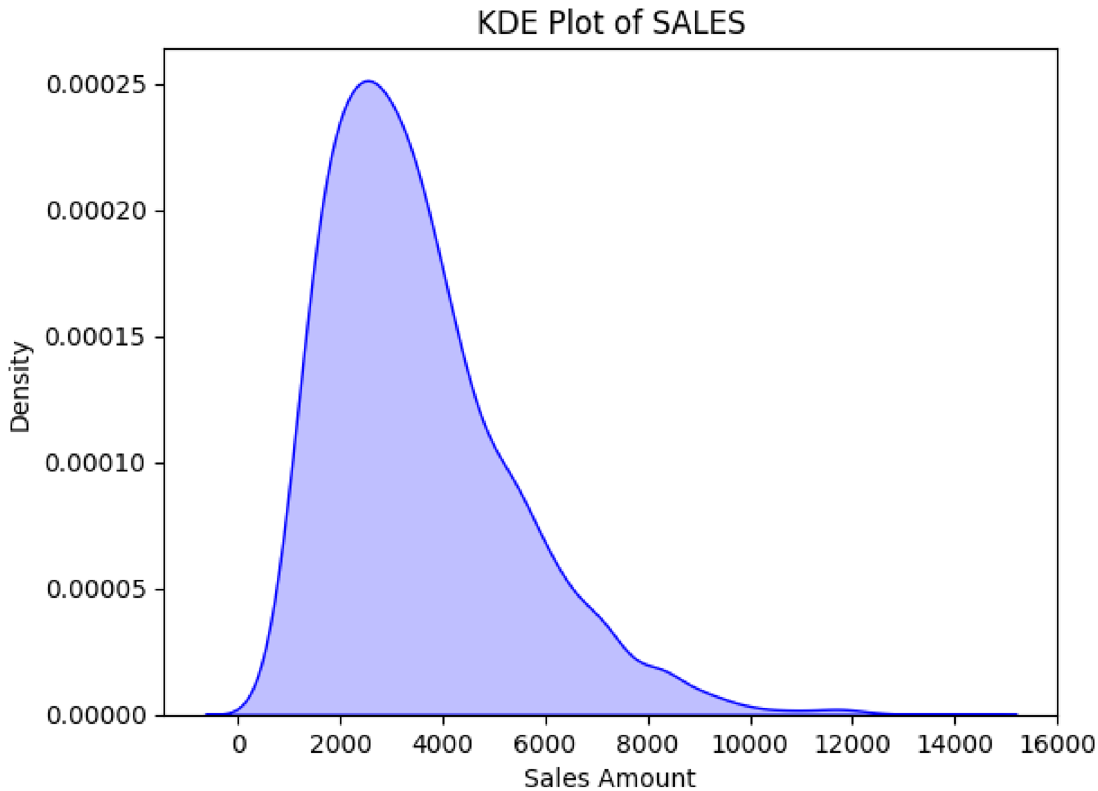
[84]:
```
sns.histplot(df["SALES"], bins=30, kde=True, color="blue")
plt.title("Histogram of SALES")
plt.xlabel("Sales Amount")
plt.ylabel("Frequency")
plt.show()
```



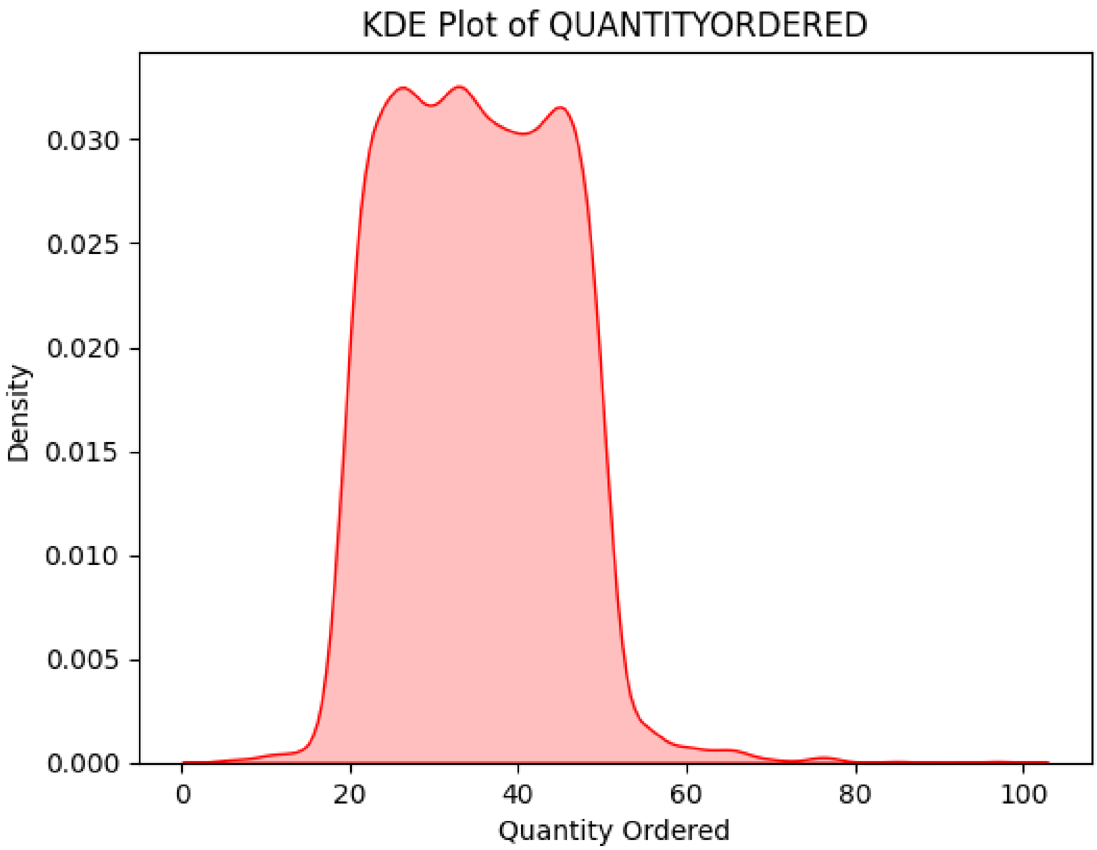
[85]:
```
sns.histplot(df["MSRP"], bins=30, kde=True, color="green")
plt.title("Histogram of MSRP")
plt.xlabel("MSRP")
plt.ylabel("Frequency")
plt.show()
```



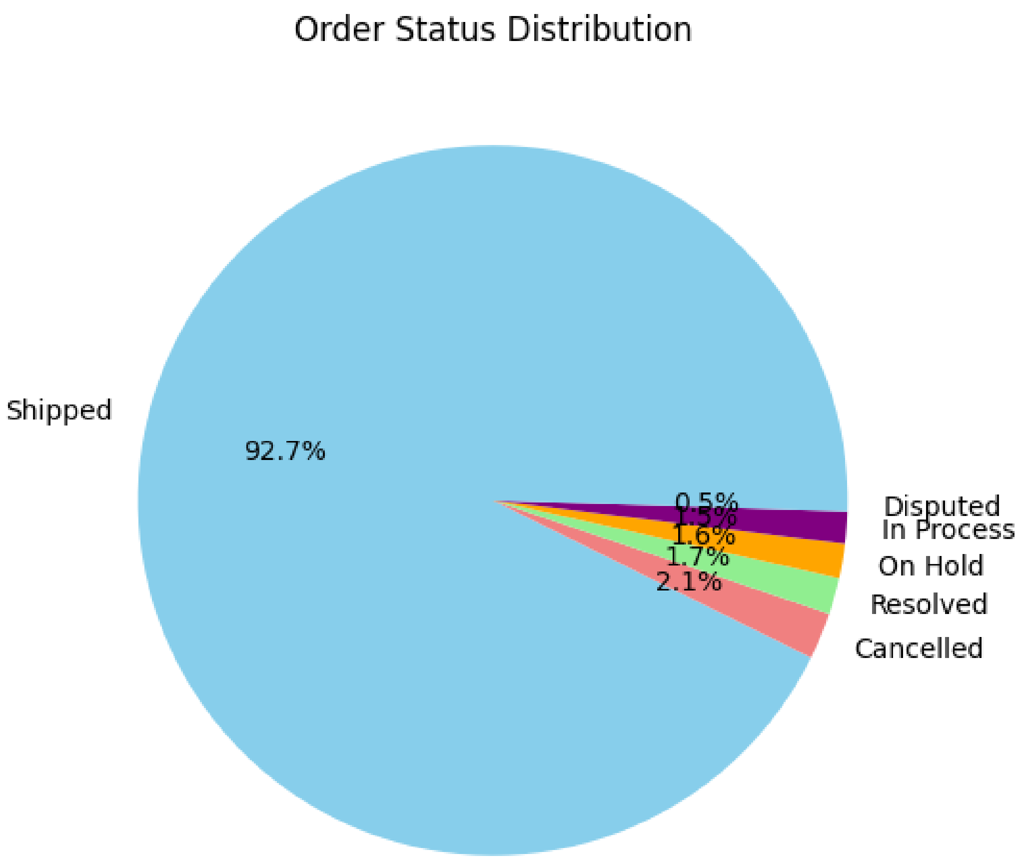
[86]:
```
sns.kdeplot(df["SALES"], fill=True, color="blue")
plt.title("KDE Plot of SALES")
plt.xlabel("Sales Amount")
plt.ylabel("Density")
plt.show()
```



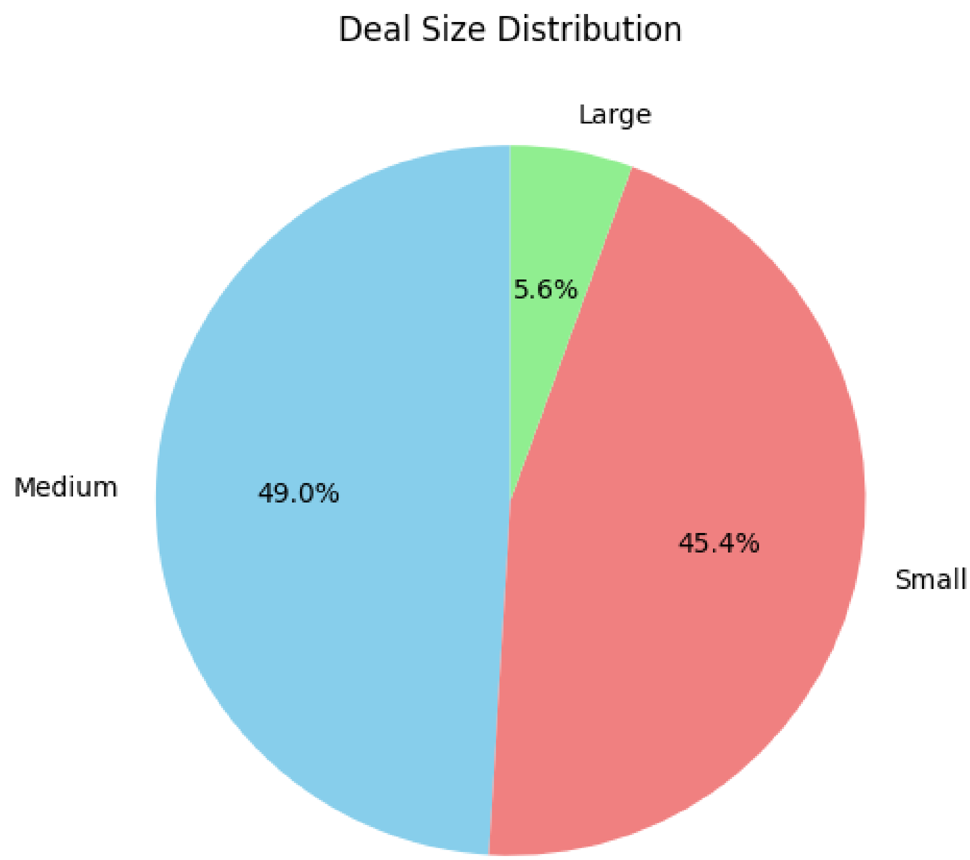
[87]:
```
sns.kdeplot(df["QUANTITYORDERED"], fill=True, color="red")
plt.title("KDE Plot of QUANTITYORDERED")
plt.xlabel("Quantity Ordered")
plt.ylabel("Density")
plt.show()
```



[88]:
```
plt.figure(figsize=(6, 6))
df["STATUS"].value_counts().plot.pie(autopct="%1.1f%%", colors=["skyblue",␣
↪"lightcoral", "lightgreen", "orange", "purple"])
plt.title("Order Status Distribution")
plt.ylabel("")
plt.show()
```



[89]:
```
plt.figure(figsize=(6, 6))
df["DEALSIZE"].value_counts().plot.pie(autopct="%1.1f%%", colors=["skyblue",␣
↪"lightcoral", "lightgreen"], startangle=90)
plt.title("Deal Size Distribution")
plt.ylabel("")
plt.show()
```



- Generate a small dataset from a non-normal distribution (e.g., uniform distribution).

[90]:
```
uniform_data = np.random.uniform(low=10, high=50, size=100)
```

[91]:
```
df_uniform = pd.DataFrame(uniform_data, columns=["Uniform_Distribution"])
```

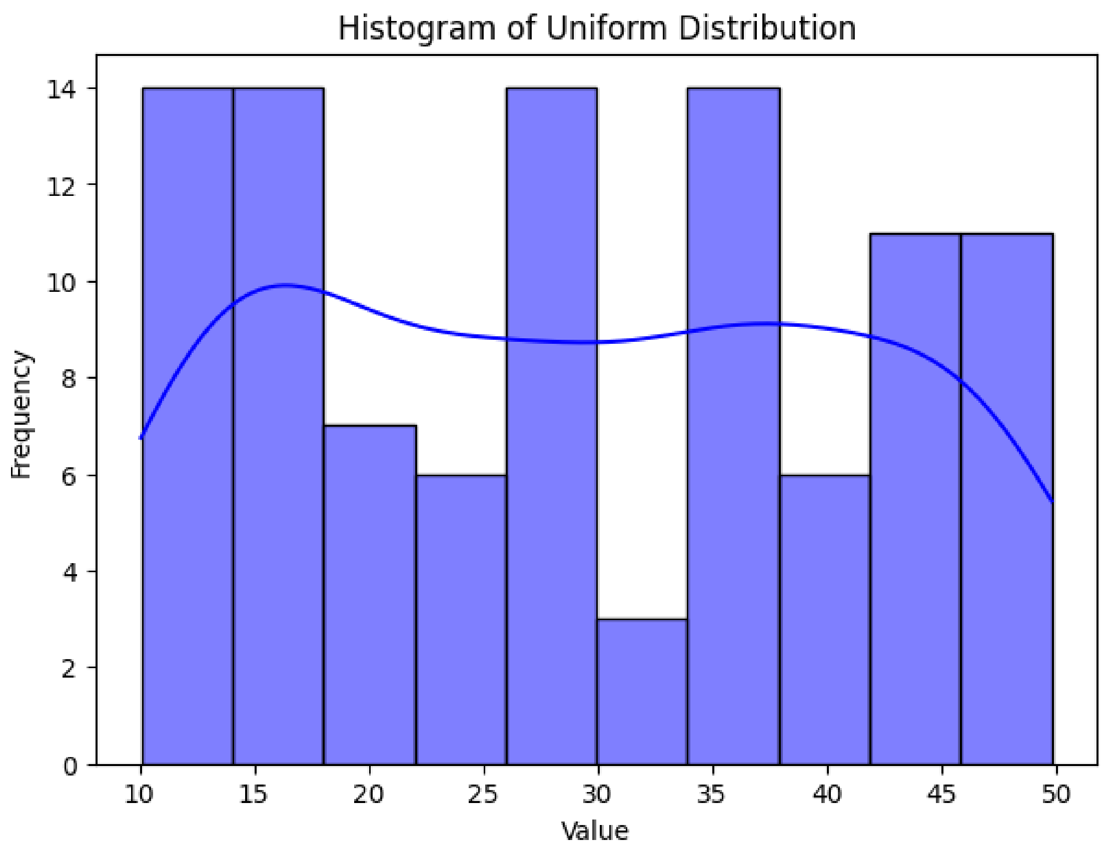
[92]:
```
df_uniform.head()
```

[92]:

	Uniform_Distribution
0	36.052119
1	44.958219
2	36.071830
3	44.038639
4	28.605493

[93]:
```
plt.figure(figsize=(7, 5))
```

```
sns.histplot(df_uniform["Uniform_Distribution"], bins=10, kde=True,␣
 ↪color="blue")
plt.title("Histogram of Uniform Distribution")
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.show()
```



- Take multiple random samples from the dataset and calculate their means.

[94]:
```
num_samples = 30
sample_size = 10
```

[95]:
```
sample_means = []

for _ in range(num_samples):
    sample = df_uniform["Uniform_Distribution"].sample(sample_size,␣
 ↪replace=True)
    sample_means.append(sample.mean())
```

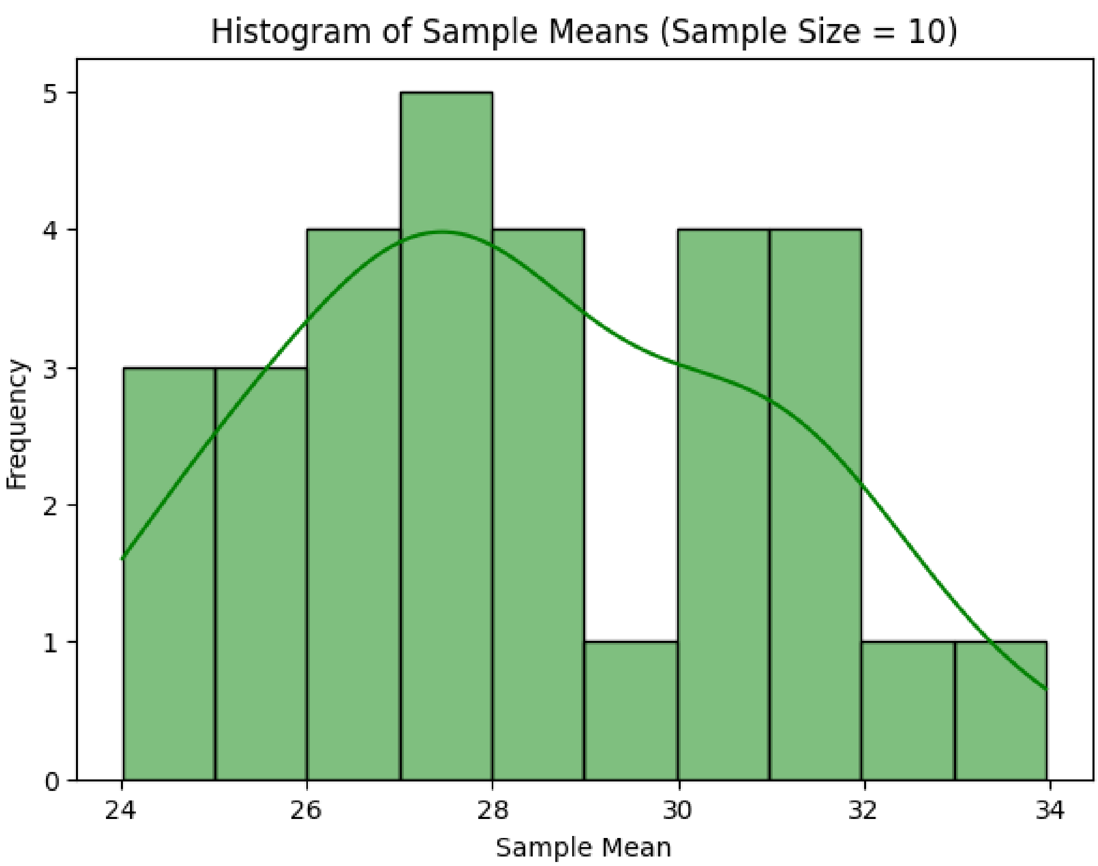
[96]:
```
df_sample_means = pd.DataFrame(sample_means, columns=["Sample_Means"])
```

[97]:
```
df_sample_means.head()
```

[97]:

	Sample_Means
0	31.461911
1	25.024023
2	26.445996
3	24.220957
4	31.559834

[98]:
```
plt.figure(figsize=(7, 5))
sns.histplot(df_sample_means["Sample_Means"], bins=10, kde=True, color="green")
plt.title(f"Histogram of Sample Means (Sample Size = {sample_size})")
plt.xlabel("Sample Mean")
plt.ylabel("Frequency")
plt.show()
```



Hypothesis Testing :

[99]:
```
from scipy.stats import ttest_ind
```

```
sales_small = df[df["DEALSIZE"] == "Small"]["SALES"]
sales_large = df[df["DEALSIZE"] == "Large"]["SALES"]

t_stat, p_value = ttest_ind(sales_small, sales_large, equal_var=False)

print(f"T-statistic: {t_stat:.4f}")
print(f"P-value: {p_value:.4f}")

alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis: There is a significant difference in␣
↪SALES between Small and Large deal sizes.")
else:
    print("Fail to reject the null hypothesis: No significant difference in␣
↪SALES between Small and Large deal sizes.")
```

```
T-statistic: -59.5907
P-value: 0.0000
Reject the null hypothesis: There is a significant difference in SALES between
Small and Large deal sizes.
```

[100]:
```
contingency_table = pd.crosstab(df["STATUS"], df["DEALSIZE"])

chi2_stat, p_value, dof, expected = stats.chi2_contingency(contingency_table)

print(f"Chi-Square Statistic: {chi2_stat:.4f}")
print(f"Degrees of Freedom: {dof}")
print(f"P-value: {p_value:.4f}")

alpha = 0.05
if p_value < alpha:
    print("Reject the null hypothesis: STATUS and DEALSIZE are dependent.")
else:
    print("Fail to reject the null hypothesis: No significant relationship␣
↪between STATUS and DEALSIZE.")
```

```
Chi-Square Statistic: 34.3444
Degrees of Freedom: 10
P-value: 0.0002
Reject the null hypothesis: STATUS and DEALSIZE are dependent.
```

[101]:
```
len(sales_data)
```

[101]: 2823

[102]:
```
mean_sales = np.mean(sales_data)
std_error = stats.sem(sales_data)
```

[103]:
```
mean_sales
```

[103]:
```
np.float64(3553.889071909316)
```

[104]:
```
std_error
```

[104]:
```
np.float64(34.66589211924902)
```

[105]:
```
confidence = 0.95
margin_of_error = stats.t.ppf((1 + confidence) / 2, len(sales_data) - 1) *␣
↪std_error

lower_bound = mean_sales - margin_of_error
upper_bound = mean_sales + margin_of_error

print(f"95% Confidence Interval for Mean SALES: ({lower_bound:.2f},␣
↪{upper_bound:.2f})")

print("Interpretation: We are 95% confident that the true mean SALES value lies␣
↪within this range.")
```

```
95% Confidence Interval for Mean SALES: (3485.92, 3621.86)
Interpretation: We are 95% confident that the true mean SALES value lies within
this range.
```

[106]:
```
import scipy.stats as stats

usa_sales = df[df["COUNTRY"] == "USA"]["SALES"].dropna()
france_sales = df[df["COUNTRY"] == "France"]["SALES"].dropna()

t_stat, p_value = stats.ttest_ind(usa_sales, france_sales, equal_var=False)  #␣
↪Welch's t-test

print(f"T-Statistic: {t_stat:.4f}")
print(f"P-Value: {p_value:.4f}")

alpha = 0.05
if p_value < alpha:
    print("Result: Reject the Null Hypothesis (Significant difference in SALES␣
↪between USA and France)")
else:
    print("Result: Fail to Reject the Null Hypothesis (No significant␣
↪difference in SALES between USA and France)")
```

```
T-Statistic: 0.6063
P-Value: 0.5446
Result: Fail to Reject the Null Hypothesis (No significant difference in SALES
```

between USA and France)

[107]:
```
from scipy.stats import skew, kurtosis

sales_skewness = skew(df["SALES"], nan_policy='omit')
print(f"Skewness of SALES: {sales_skewness:.4f}")

sales_kurtosis = kurtosis(df["SALES"], nan_policy='omit')
print(f"Kurtosis of SALES: {sales_kurtosis:.4f}")
```

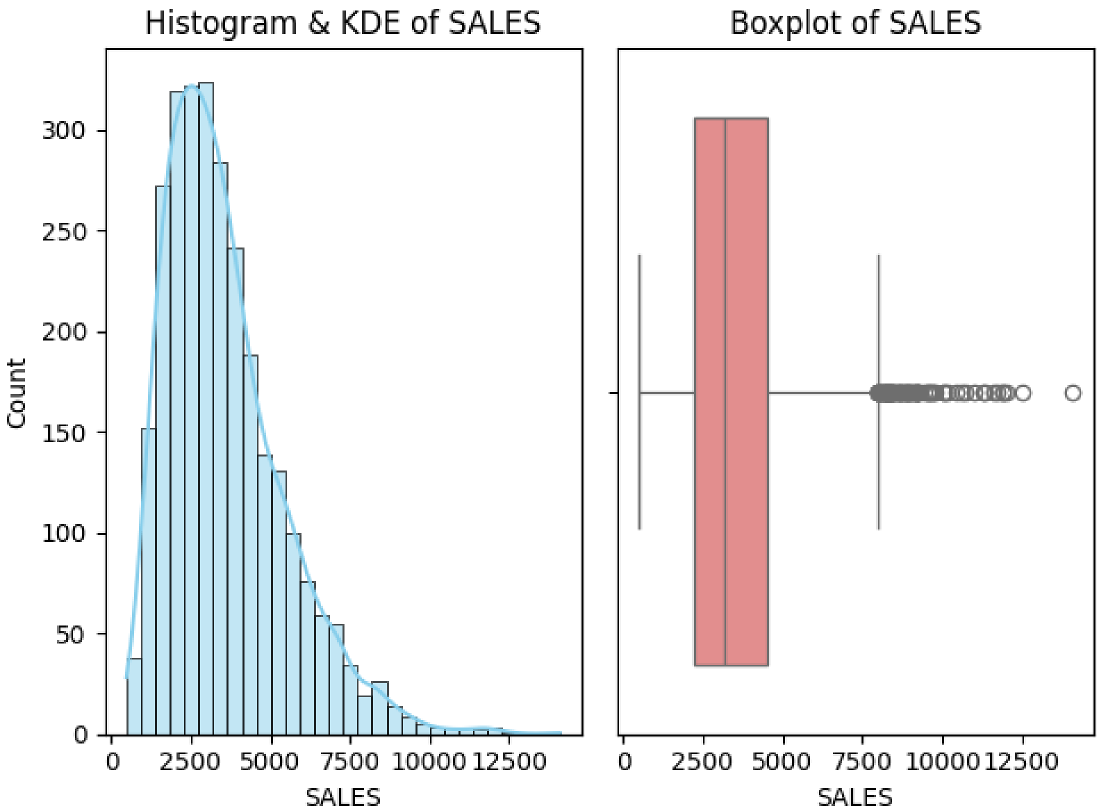
```
Skewness of SALES: 1.1605
Kurtosis of SALES: 1.7874
```

[108]:
```
import pandas as pd
from scipy.stats import skew, kurtosis

numerical_cols = df.select_dtypes(include=['number'])

skewness_values = numerical_cols.apply(lambda x: skew(x, nan_policy='omit'))
kurtosis_values = numerical_cols.apply(lambda x: kurtosis(x, nan_policy='omit'))

stats_df = pd.DataFrame({'Skewness': skewness_values, 'Kurtosis':␣
↪kurtosis_values})
print(stats_df)
```

```
                 Skewness  Kurtosis
ORDERNUMBER      0.013816 -1.173357
QUANTITYORDERED  0.362393  0.412883
PRICEEACH       -0.946146 -0.376279
ORDERLINENUMBER  0.590427 -0.562285
SALES            1.160459  1.787378
QTR_ID          -0.255815 -1.498237
MONTH_ID        -0.272757 -1.382951
YEAR_ID          0.271307 -0.951001
MSRP             0.579867 -0.133706
```

[109]:
```
highly_skewed_cols = stats_df[stats_df['Skewness'].abs() > 1].index

for i, col in enumerate(highly_skewed_cols):
    plt.subplot(len(highly_skewed_cols), 2, 2*i+1)
    sns.histplot(df[col], bins=30, kde=True, color="skyblue")
    plt.title(f"Histogram & KDE of {col}")

    plt.subplot(len(highly_skewed_cols), 2, 2*i+2)
    sns.boxplot(x=df[col], color="lightcoral")
    plt.title(f"Boxplot of {col}")

plt.tight_layout()
```

```
plt.show()
```



Histogram & KDE of SALES
Count
300
250
200
150
100
50
0
0
2500
5000
7500
10000
12500
SALES
Boxplot of SALES
0
2500
5000
7500
10000
12500
SALES