Exploratory Data Analysis (EDA)

Dress Sales

In [1]:

```
import pandas as pd

# Load the dataset
data = pd.read_csv('DressSales.csv')

# Display the first few rows of the dataset
print(data.head())

# Check for missing values
print(data.isnull().sum())
```

```
     Dress_ID  29-08-2013  31-08-2013  09-02-2013  09-04-2013  09-06-2013  \
0  1006032852        2114        2274        2491        2660        2727
1  1212192089         151         275         570         750         813
2  1190380701           6           7           7           7           8
3   966005983        1005        1128        1326        1455        1507
4   876339541         996        1175        1304        1396        1432

   09-08-2013  09-10-2013 09-12-2013 14-09-2013  ... 24-09-2013 26-09-2013  \
0        2887        2930       3119       3204  ...       3554     3624.0
1        1066        1164       1558       1756  ...       2710     2942.0
2           8           9         10         10  ...         11       11.0
3        1621        1637       1723       1746  ...       1878     1892.0
4        1559        1570       1638       1655  ...       2032     2156.0

  28-09-2013 30-09-2013  10-02-2013  10-04-2013  10-06-2013  10-08-2013  \
0       3706     3746.0      3795.0      3832.0        3897      3923.0
1       3258     3354.0      3475.0      3654.0        3911      4024.0
2         11       11.0        11.0        11.0          11        11.0
3       1914     1924.0      1929.0      1941.0        1952      1955.0
4       2252     2312.0      2387.0      2459.0        2544      2614.0

   10-10-2013  10-12-2013
0      3985.0        4048
1      4125.0        4277
2        11.0          11
3      1959.0        1963
4      2693.0        2736

[5 rows x 24 columns]
Dress_ID        0
29-08-2013      0
31-08-2013      0
09-02-2013      0
09-04-2013      0
09-06-2013      0
09-08-2013      0
09-10-2013      0
09-12-2013      0
14-09-2013      0
16-09-2013      0
18-09-2013      0
20-09-2013      0
22-09-2013      0
24-09-2013      0
26-09-2013    222
28-09-2013      0
30-09-2013    257
10-02-2013    259
10-04-2013    258
10-06-2013      0
```

```
10-08-2013    255
10-10-2013    255
10-12-2013      0
dtype: int64
```

Summary Statistics

Let's calculate summary statistics for the numeric columns to get a sense of the data's central tendency and dispersion:

In [2]:

```
# Summary statistics
print(data.describe())
```

```
           Dress_ID   29-08-2013   31-08-2013   09-02-2013   09-04-2013  \
count  4.790000e+02   479.000000   479.000000   479.000000   479.000000
mean   9.022420e+08   198.085595   209.776618   223.551148   299.791232
std    1.822352e+08   579.189322   590.836166   603.098222   601.716515
min    1.234568e+08     0.000000     0.000000     0.000000     1.000000
25%    7.666611e+08     0.000000     0.000000     0.000000    28.500000
50%    9.096250e+08     2.000000     3.000000     4.000000   110.000000
75%    1.039684e+09   138.500000   165.500000   194.500000   308.500000
max    1.253973e+09  7455.000000  7467.000000  7479.000000  7374.000000

        09-06-2013   09-08-2013   09-10-2013   24-09-2013   26-09-2013  \
count   479.000000   479.000000   479.000000   479.000000   257.000000
mean    304.745303   316.534447   320.100209   372.939457   295.501946
std     603.854257   609.070537   610.360681   631.674995   696.941427
min       1.000000     1.000000     1.000000     0.000000     0.000000
25%      31.500000    36.000000    37.000000    53.000000    19.000000
50%     116.000000   124.000000   129.000000   178.000000    60.000000
75%     319.500000   334.000000   334.000000   435.000000   227.000000
max    7351.000000  7255.000000  7240.000000  6644.000000  6528.000000

        28-09-2013   30-09-2013   10-02-2013   10-04-2013   10-06-2013  \
count   479.000000   222.000000   220.000000   221.000000   479.000000
mean    389.590814   240.914414   247.572727   251.058824   415.340292
std     646.989727   697.151163   707.881500   713.630310   666.827441
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%      55.500000    13.250000    13.750000    14.000000    62.000000
50%     186.000000    52.500000    53.000000    52.000000   200.000000
75%     465.500000   112.750000   112.250000   111.000000   489.000000
max    6476.000000  6327.000000  6285.000000  6142.000000  6049.000000

        10-08-2013   10-10-2013   10-12-2013
count   224.000000   224.000000   479.000000
mean    258.437500   262.611607   434.048017
std     724.092886   732.867748   684.146593
min       0.000000     0.000000     0.000000
25%      14.750000    14.750000    65.000000
50%      57.000000    59.500000   216.000000
75%     131.250000   133.500000   526.500000
max    5912.000000  5862.000000  5753.000000
```

Bivariate Analysis

Bivariate analysis involves exploring the relationships between pairs of variables. Since your dataset contains dates and 'Dress_ID', you may want to examine how 'Dress_ID' varies over time

In [3]:

```
import matplotlib.pyplot as plt

# Example: Bivariate analysis between Dress_ID and a specific date column (e.g., '29-08-2013')
plt.figure(figsize=(10, 6))
plt.scatter(data['29-08-2013'], data['Dress_ID'], alpha=0.5)
plt.xlabel('29-08-2013')
```

```
plt.ylabel('Dress_ID')
plt.title('Relationship between Dress_ID and 29-08-2013')
plt.show()
```