..... Big Data Final Project .....

1. EDA Function

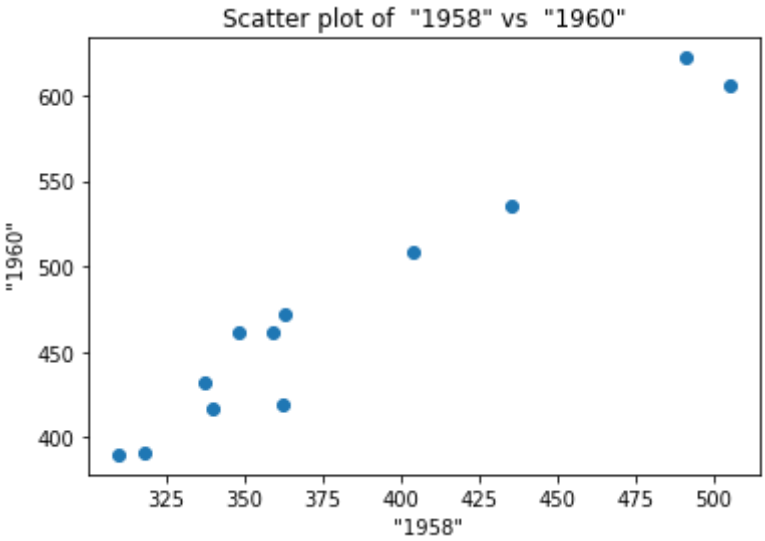
In [9]:
```
import pandas as pd
from pandas.auto_eda import produce_eda
```

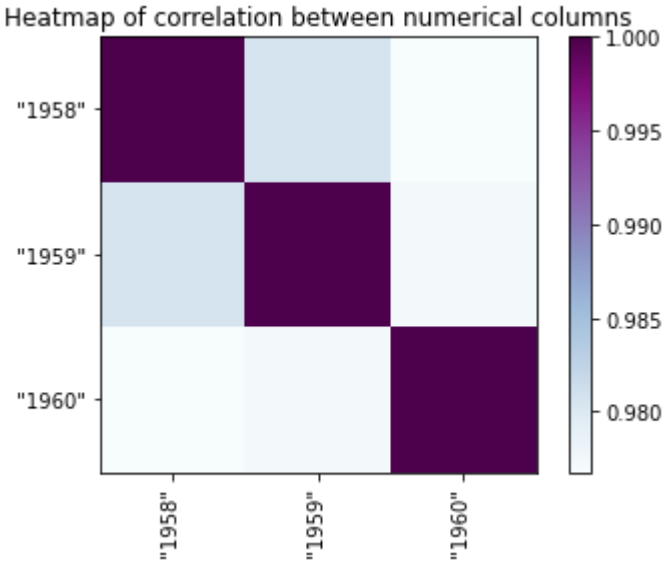
Test case on datasets

In [10]:
```
data_frame = pd.read_csv('airtravel.csv')
produce_eda(data_frame)
```

```
Missing values:
Month      0
 "1958"    0
 "1959"    0
 "1960"    0
dtype: int64
Percentage of missing values per column:
Month      0.0
 "1958"    0.0
 "1959"    0.0
 "1960"    0.0
dtype: float64

Data types of columns:
Month      object
 "1958"     int64
 "1959"     int64
 "1960"     int64
dtype: object
Number of unique values per column:
Month      12
 "1958"    12
 "1959"    12
 "1960"    11
dtype: int64

Summary statistics of numerical columns:
           "1958"      "1959"      "1960"
count   12.000000   12.000000   12.000000
mean   381.000000  428.333333  476.166667
std     64.530472   69.830097   77.737125
min    310.000000  342.000000  390.000000
25%    339.250000  387.500000  418.500000
50%    360.500000  406.500000  461.000000
75%    411.750000  465.250000  514.750000
max    505.000000  559.000000  622.000000
Top 5 most common values in Month:
Month
JAN    1
FEB    1
MAR    1
APR    1
MAY    1
Name: count, dtype: int64

Correlation between numerical columns:
```

```
           "1958"    "1959"    "1960"
 "1958"  1.000000  0.980656  0.976757
 "1959"  0.980656  1.000000  0.977292
 "1960"  0.976757  0.977292  1.000000
Mean of  "1958": 381.00
Standard deviation of  "1958": 64.53
Mean of  "1959": 428.33
Standard deviation of  "1959": 69.83
Mean of  "1960": 476.17
Standard deviation of  "1960": 77.74
```






```
Summary statistics of categorical columns:
       Month
count     12
unique    12
top      JAN
freq       1

T-test between  "1958" and  "1960": t-statistic = -3.2630, p-value = 0.0036
```

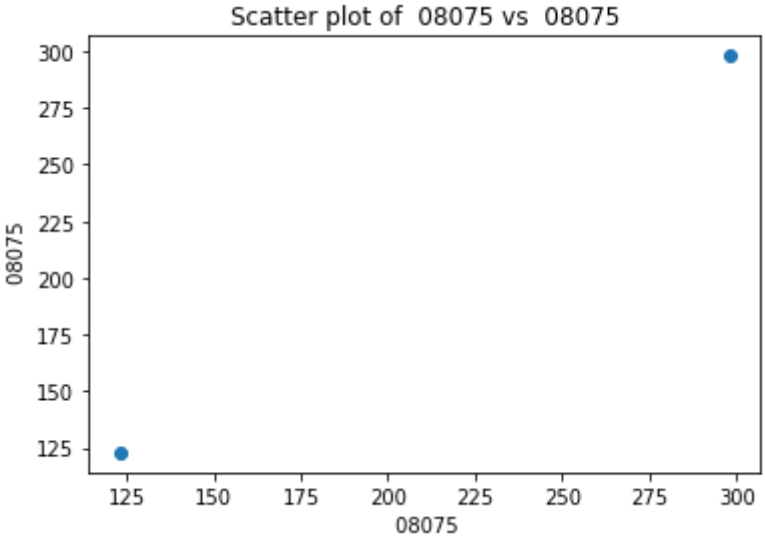
In [11]:

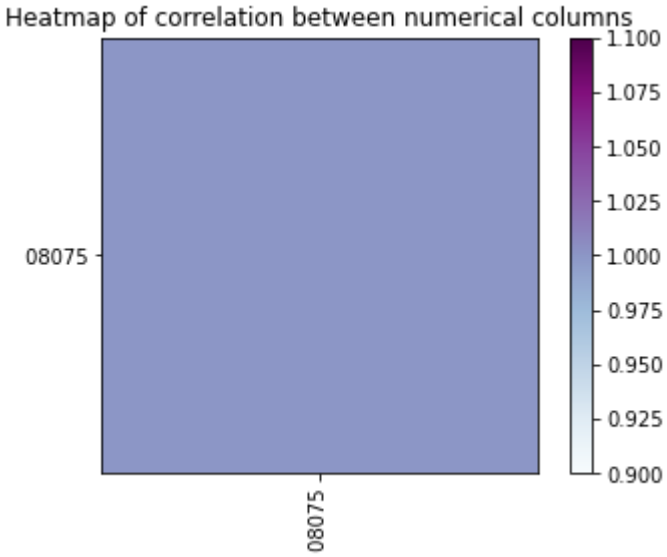
```
data_frame = pd.read_csv('addresses.csv')
produce_eda(data_frame)
```

```
Missing values:
John                1
Doe                 0
120 jefferson st.   1
Riverside           0
 NJ                 0
 08075              0
dtype: int64
```

```
Percentage of missing values per column:
John                 20.0
Doe                   0.0
120 jefferson st.    20.0
Riverside             0.0
 NJ                   0.0
 08075                0.0
dtype: float64

Data types of columns:
John                 object
Doe                  object
120 jefferson st.    object
Riverside            object
 NJ                  object
 08075                int64
dtype: object
Number of unique values per column:
John                 4
Doe                  5
120 jefferson st.    4
Riverside            4
 NJ                  5
 08075               5
dtype: int64

Summary statistics of numerical columns:
              08075
count      5.000000
mean   21769.800000
std    39059.209909
min      123.000000
25%      298.000000
50%     8075.000000
75%     9119.000000
max    91234.000000
Top 5 most common values in John:
John
Jack                     1
John "Da Man"            1
Stephen                  1
Joan "the bone", Anne    1
Name: count, dtype: int64
Top 5 most common values in Doe:
Doe
McGinnis    1
Repici      1
Tyler       1
Blankman    1
Jet         1
Name: count, dtype: int64
Top 5 most common values in 120 jefferson st.:
120 jefferson st.
220 hobo Av.                        1
120 Jefferson St.                   1
7452 Terrace "At the Plaza" road    1
9th, at Terrace plc                 1
Name: count, dtype: int64
Top 5 most common values in Riverside:
Riverside
SomeTown       2
Phila          1
Riverside      1
Desert City    1
Name: count, dtype: int64
Top 5 most common values in  NJ:
 NJ
 PA    1
 NJ    1
```

```
SD     1
 SD    1
CO     1
Name: count, dtype: int64

Correlation between numerical columns:
         08075
 08075     1.0
Mean of  08075: 21769.80
Standard deviation of  08075: 39059.21
```






```
Summary statistics of categorical columns:
        John       Doe 120 jefferson st. Riverside   NJ
count      4         5                 4         5    5
unique     4         5                 4         4    5
top     Jack  McGinnis      220 hobo Av.  SomeTown   PA
freq       1         1                 1         2    1

Chi-squared test between John and  NJ: chi-squared statistic = 12.0000, p-value = 0.
2133
```

Test Case - 1

In [12]:

```
# Test case 1
# Create data frame
data = {'col1': ['a', 'b', 'c', 'a', 'b'], 'col2': ['x', 'y', 'z', 'y', 'x'], 'col3'
df = pd.DataFrame(data)
```

```
# Call produce_eda function
produce_eda(df)
```

```
Missing values:
col1    0
col2    0
col3    0
dtype: int64
Percentage of missing values per column:
col1    0.0
col2    0.0
col3    0.0
dtype: float64

Data types of columns:
col1    object
col2    object
col3    object
dtype: object
Number of unique values per column:
col1    3
col2    3
col3    2
dtype: int64
Top 5 most common values in col1:
col1
a    2
b    2
c    1
Name: count, dtype: int64
Top 5 most common values in col2:
col2
x    2
y    2
z    1
Name: count, dtype: int64
Top 5 most common values in col3:
col3
yes    3
no     2
Name: count, dtype: int64

Summary statistics of categorical columns:
       col1 col2 col3
count     5    5    5
unique    3    3    2
top       a    x  yes
freq      2    2    3

Chi-squared test between col1 and col3: chi-squared statistic = 0.8333, p-value = 0.
6592
```

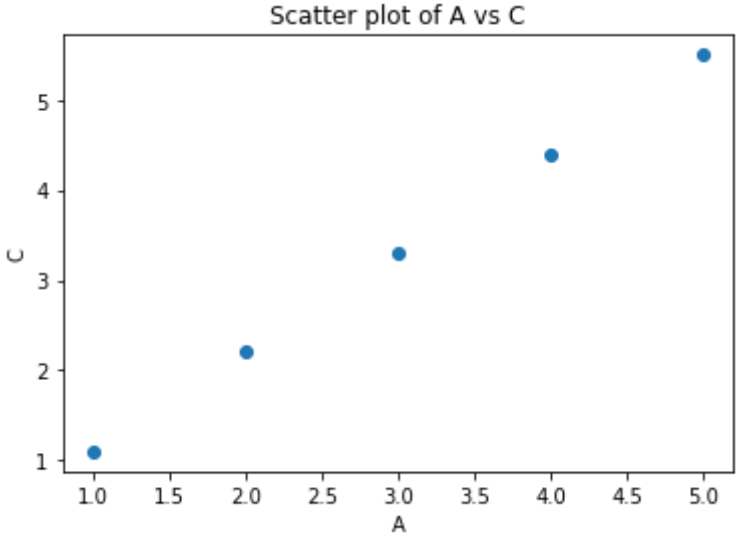
Test Case - 2

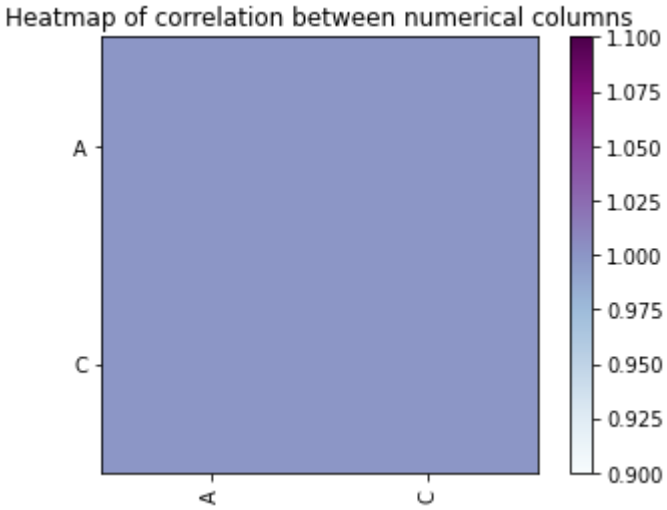
In [13]:

```
# Test case 2: DataFrame with no missing values
df = pd.DataFrame({
    'A': [1, 2, 3, 4, 5],
    'B': ['a', 'b', 'c', 'd', 'e'],
    'C': [1.1, 2.2, 3.3, 4.4, 5.5]
})

produce_eda(df)
```

```
Missing values:
A    0
B    0
```

```
C    0
dtype: int64
Percentage of missing values per column:
A    0.0
B    0.0
C    0.0
dtype: float64

Data types of columns:
A      int64
B     object
C    float64
dtype: object
Number of unique values per column:
A    5
B    5
C    5
dtype: int64

Summary statistics of numerical columns:
              A         C
count  5.000000  5.000000
mean   3.000000  3.300000
std    1.581139  1.739253
min    1.000000  1.100000
25%    2.000000  2.200000
50%    3.000000  3.300000
75%    4.000000  4.400000
max    5.000000  5.500000
Top 5 most common values in B:
B
a    1
b    1
c    1
d    1
e    1
Name: count, dtype: int64

Correlation between numerical columns:
     A    C
A  1.0  1.0
C  1.0  1.0
Mean of A: 3.00
Standard deviation of A: 1.58
Mean of C: 3.30
Standard deviation of C: 1.74
```






```
Summary statistics of categorical columns:
        B
count   5
unique  5
top     a
freq    1

T-test between A and C: t-statistic = -0.2854, p-value = 0.7826
```

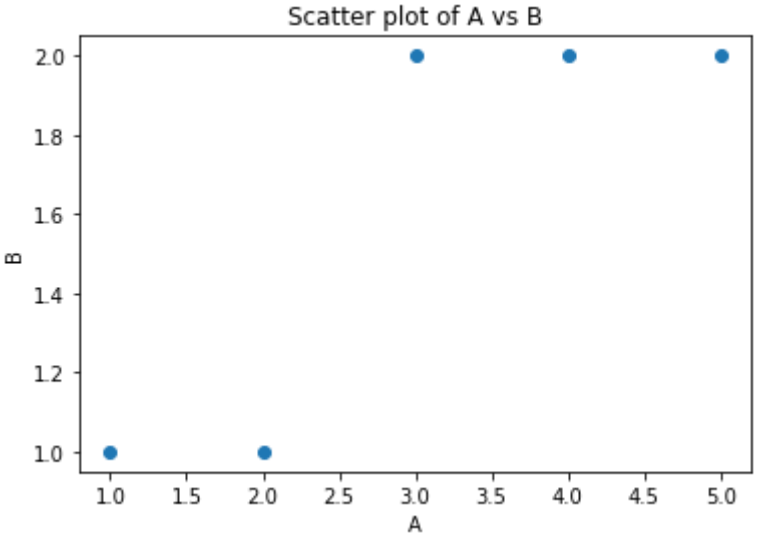
Test Case - 3

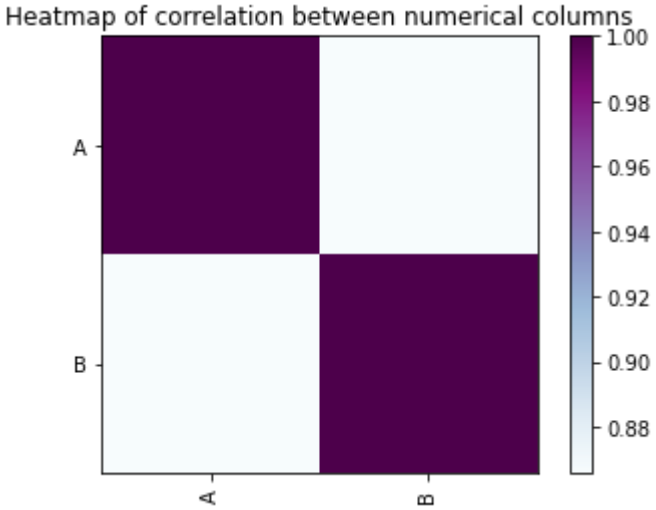
In [14]:

```
# Test case 3
df = pd.DataFrame({
    'A': [1, 2, 3, 4, 5],
    'B': [1, 1, 2, 2, 2],
    'C': ['apple', 'banana', 'apple', 'banana', 'banana'],
    'D': ['alpha', 'beta', 'gamma', 'alpha', 'gamma']
})
produce_eda(df)
```

```
Missing values:
A    0
B    0
C    0
D    0
dtype: int64
Percentage of missing values per column:
A    0.0
B    0.0
C    0.0
D    0.0
dtype: float64

Data types of columns:
A     int64
B     int64
C    object
D    object
dtype: object
Number of unique values per column:
A    5
B    2
C    2
D    3
dtype: int64

Summary statistics of numerical columns:
              A         B
```

```
count  5.000000  5.000000
mean   3.000000  1.600000
std    1.581139  0.547723
min    1.000000  1.000000
25%    2.000000  1.000000
50%    3.000000  2.000000
75%    4.000000  2.000000
max    5.000000  2.000000
Top 5 most common values in C:
C
banana    3
apple     2
Name: count, dtype: int64
Top 5 most common values in D:
D
alpha    2
gamma    2
beta     1
Name: count, dtype: int64

Correlation between numerical columns:
          A         B
A  1.000000  0.866025
B  0.866025  1.000000
Mean of A: 3.00
Standard deviation of A: 1.58
Mean of B: 1.60
Standard deviation of B: 0.55
```






```
Summary statistics of categorical columns:
              C      D
count         5      5
unique        2      3
top     banana  alpha
```

```
freq         3      2

T-test between A and B: t-statistic = 1.8708, p-value = 0.0983

Chi-squared test between C and D: chi-squared statistic = 0.8333, p-value = 0.6592
```

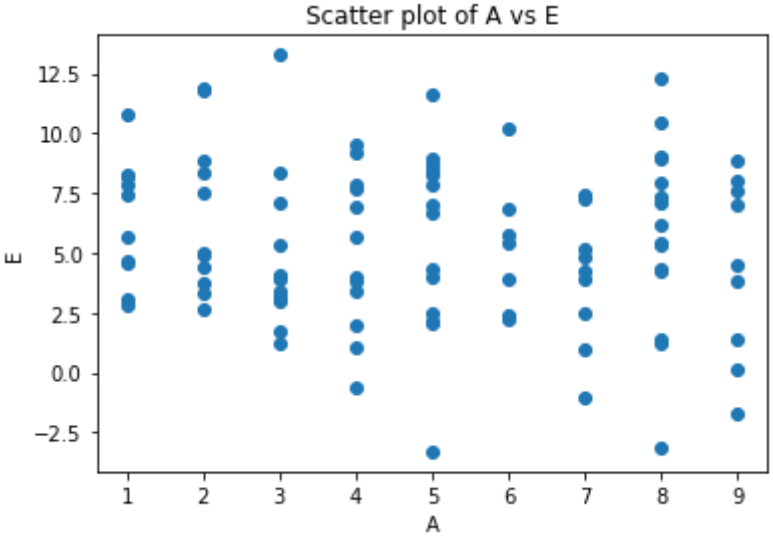
Test Case - 4

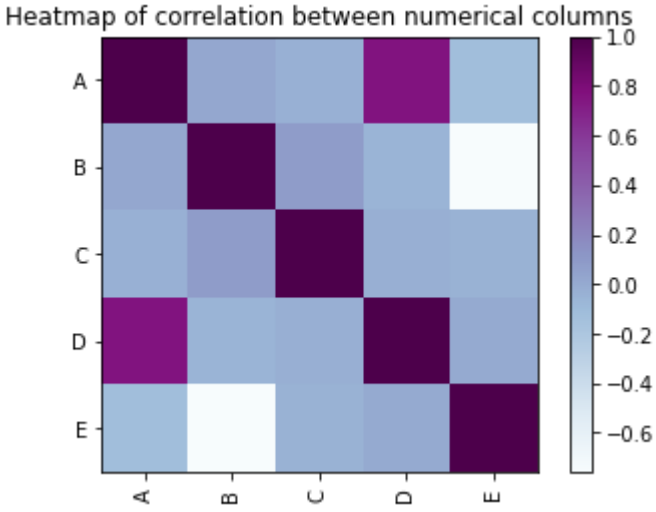
In [15]:

```
import numpy as np
import pandas as pd

# Generate a sample dataset with correlated columns
data = {'A': np.random.randint(1, 10, 100),
        'B': np.random.randint(1, 10, 100),
        'C': np.random.randint(1, 10, 100)}

df = pd.DataFrame(data)

# Add a column that is correlated with column A
df['D'] = df['A'] + np.random.normal(0, 2, 100)

# Add a column that is negatively correlated with column B
df['E'] = 10 - df['B'] + np.random.normal(0, 2, 100)

# Test the produce_eda function
produce_eda(df)
```

```
Missing values:
A    0
B    0
C    0
D    0
E    0
dtype: int64
Percentage of missing values per column:
A    0.0
B    0.0
C    0.0
D    0.0
E    0.0
dtype: float64

Data types of columns:
A      int32
B      int32
C      int32
D    float64
E    float64
dtype: object
Number of unique values per column:
A      9
B      9
C      9
D    100
E    100
dtype: int64

Summary statistics of numerical columns:
                A           B           C           D           E
count  100.000000  100.000000  100.000000  100.000000  100.000000
mean     4.980000    5.000000    5.460000    4.963933    5.355917
std      2.546457    2.628515    2.560066    3.211697    3.369355
min      1.000000    1.000000    1.000000   -1.552050   -3.307291
25%      3.000000    3.000000    4.000000    2.741817    3.133992
50%      5.000000    5.000000    6.000000    4.826338    5.054182
```

```
75%      7.000000    7.250000    8.000000    7.096281    7.832073
max      9.000000    9.000000    9.000000   12.912776   13.288098

Correlation between numerical columns:
          A         B         C         D         E
A  1.000000  0.024146 -0.032662  0.758887 -0.124160
B  0.024146  1.000000  0.082559 -0.053804 -0.758469
C -0.032662  0.082559  1.000000 -0.024646 -0.039058
D  0.758887 -0.053804 -0.024646  1.000000  0.002464
E -0.124160 -0.758469 -0.039058  0.002464  1.000000
Mean of A: 4.98
Standard deviation of A: 2.55
Mean of B: 5.00
Standard deviation of B: 2.63
Mean of C: 5.46
Standard deviation of C: 2.56
Mean of D: 4.96
Standard deviation of D: 3.21
Mean of E: 5.36
Standard deviation of E: 3.37
```






```
T-test between A and E: t-statistic = -0.8901, p-value = 0.3745
```

Test case 5

In [16]:

```
# Test case 1
# Create data frame
data = {'col1': ['a', 'b', 'c', 'a', 'b'], 'col2': ['x', 'y', 'z', 'y', 'x'], 'col3'
df = pd.DataFrame(data)
```

```
# Call produce_eda function
produce_eda(df)
```

```
Missing values:
col1    0
col2    0
col3    0
dtype: int64
Percentage of missing values per column:
col1    0.0
col2    0.0
col3    0.0
dtype: float64

Data types of columns:
col1    object
col2    object
col3    object
dtype: object
Number of unique values per column:
col1    3
col2    3
col3    2
dtype: int64
Top 5 most common values in col1:
col1
a    2
b    2
c    1
Name: count, dtype: int64
Top 5 most common values in col2:
col2
x    2
y    2
z    1
Name: count, dtype: int64
Top 5 most common values in col3:
col3
yes    3
no     2
Name: count, dtype: int64

Summary statistics of categorical columns:
       col1 col2 col3
count     5    5    5
unique    3    3    2
top       a    x  yes
freq      2    2    3

Chi-squared test between col1 and col3: chi-squared statistic = 0.8333, p-value = 0.
6592
```

In [ ]: