```
from google.colab import files
uploaded = files.upload()
import pandas as pd
df = pd.read_csv("Caravan.csv")
print(df.head())
```

Choose Files Caravan.csv

- **Caravan.csv**(text/csv) - 1041013 bytes, last modified: 22/11/2024 - 100% done

```
Saving Caravan.csv to Caravan.csv
   rownames  MOSTYPE  MAANTHUI  MGEMOMV  MGEMLEEF  MOSHOOFD  MGODRK  MGODPR  \
0         1       33         1        3         2         8       0       5
1         2       37         1        2         2         8       1       4
2         3       37         1        2         2         8       0       4
3         4        9         1        3         3         3       2       3
4         5       40         1        4         2        10       1       4

   MGODOV  MGODGE  ...  APERSONG  AGEZONG  AWAOREG  ABRAND  AZEILPL  APLEZIER  \
0       1       3  ...         0        0        0       1        0         0
1       1       4  ...         0        0        0       1        0         0
2       2       4  ...         0        0        0       1        0         0
3       2       4  ...         0        0        0       1        0         0
4       1       4  ...         0        0        0       1        0         0

   AFIETS  AINBOED  ABYSTAND  Purchase
0       0        0         0        No
1       0        0         0        No
2       0        0         0        No
3       0        0         0        No
4       0        0         0        No

[5 rows x 87 columns]
```

```
print("First five rows of the dataset:")
print(df.head())
```

```
First five rows of the dataset:
   rownames  MOSTYPE  MAANTHUI  MGEMOMV  MGEMLEEF  MOSHOOFD  MGODRK  MGODPR  \
0         1       33         1        3         2         8       0       5
1         2       37         1        2         2         8       1       4
2         3       37         1        2         2         8       0       4
3         4        9         1        3         3         3       2       3
4         5       40         1        4         2        10       1       4

   MGODOV  MGODGE  ...  APERSONG  AGEZONG  AWAOREG  ABRAND  AZEILPL  APLEZIER  \
0       1       3  ...         0        0        0       1        0         0
1       1       4  ...         0        0        0       1        0         0
2       2       4  ...         0        0        0       1        0         0
3       2       4  ...         0        0        0       1        0         0
4       1       4  ...         0        0        0       1        0         0

   AFIETS  AINBOED  ABYSTAND  Purchase
0       0        0         0        No
1       0        0         0        No
2       0        0         0        No
3       0        0         0        No
4       0        0         0        No

[5 rows x 87 columns]
```

```
print("Dataset Dimensions:")
print(f"Rows: {df.shape[0]}, Columns: {df.shape[1]}")
```

```
Dataset Dimensions:
Rows: 5822, Columns: 87
```

```
print("Summary Statistics:")
print(df.describe())
```

```
Summary Statistics:
          rownames      MOSTYPE     MAANTHUI      MGEMOMV     MGEMLEEF  \
count  5822.000000  5822.000000  5822.000000  5822.000000  5822.000000
mean   2911.500000    24.253349     1.110615     2.678805     2.991240
std    1680.810965    12.846706     0.405842     0.789835     0.814589
min       1.000000     1.000000     1.000000     1.000000     1.000000
25%    1456.250000    10.000000     1.000000     2.000000     2.000000
50%    2911.500000    30.000000     1.000000     3.000000     3.000000
75%    4366.750000    35.000000     1.000000     3.000000     3.000000
max    5822.000000    41.000000    10.000000     5.000000     6.000000

          MOSHOOFD       MGODRK       MGODPR       MGODOV       MGODGE  ...  \
count  5822.000000  5822.000000  5822.000000  5822.000000  5822.000000  ...
mean      5.773617     0.696496     4.626932     1.069907     3.258502  ...
std       2.856760     1.003234     1.715843     1.017503     1.597647  ...
min       1.000000     0.000000     0.000000     0.000000     0.000000  ...
25%       3.000000     0.000000     4.000000     0.000000     2.000000  ...
```

```
50%       7.000000     0.000000     5.000000     1.000000     3.000000  ...
75%       8.000000     1.000000     6.000000     2.000000     4.000000  ...
max      10.000000     9.000000     9.000000     5.000000     9.000000  ...

            ALEVEN     APERSONG      AGEZONG      AWAOREG       ABRAND  \
count  5822.000000  5822.000000  5822.000000  5822.000000  5822.000000
mean      0.076606     0.005325     0.006527     0.004638     0.570079
std       0.377569     0.072782     0.080532     0.077403     0.562058
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     1.000000
75%       0.000000     0.000000     0.000000     0.000000     1.000000
max       8.000000     1.000000     1.000000     2.000000     7.000000

           AZEILPL     APLEZIER       AFIETS      AINBOED     ABYSTAND
count  5822.000000  5822.000000  5822.000000  5822.000000  5822.000000
mean      0.000515     0.006012     0.031776     0.007901     0.014256
std       0.022696     0.081632     0.210986     0.090463     0.119996
min       0.000000     0.000000     0.000000     0.000000     0.000000
25%       0.000000     0.000000     0.000000     0.000000     0.000000
50%       0.000000     0.000000     0.000000     0.000000     0.000000
75%       0.000000     0.000000     0.000000     0.000000     0.000000
max       1.000000     2.000000     3.000000     2.000000     2.000000

[8 rows x 86 columns]
```

```
print("Missing values per column:")
print(df.isnull().sum())
```

```
Missing values per column:
rownames    0
MOSTYPE     0
MAANTHUI    0
MGEMOMV     0
MGEMLEEF    0
           ..
APLEZIER    0
AFIETS      0
AINBOED     0
ABYSTAND    0
Purchase    0
Length: 87, dtype: int64
```

```
print("Column names in the dataset:")
print(df.columns)
print("Numerical Columns in the Dataset:")
print(df.select_dtypes(include=['int64', 'float64']).columns)
```

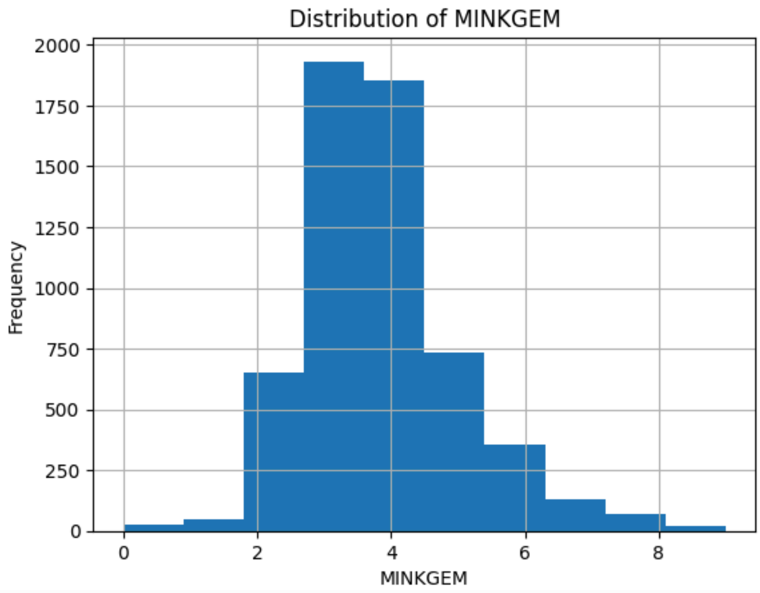
```
Column names in the dataset:
Index(['rownames', 'MOSTYPE', 'MAANTHUI', 'MGEMOMV', 'MGEMLEEF', 'MOSHOOFD',
       'MGODRK', 'MGODPR', 'MGODOV', 'MGODGE', 'MRELGE', 'MRELSA', 'MRELOV',
       'MFALLEEN', 'MFGEKIND', 'MFWEKIND', 'MOPLHOOG', 'MOPLMIDD', 'MOPLLAAG',
       'MBERHOOG', 'MBERZELF', 'MBERBOER', 'MBERMIDD', 'MBERARBG', 'MBERARBO',
       'MSKA', 'MSKB1', 'MSKB2', 'MSKC', 'MSKD', 'MHHUUR', 'MHKOOP', 'MAUT1',
       'MAUT2', 'MAUT0', 'MZFONDS', 'MZPART', 'MINKM30', 'MINK3045',
       'MINK4575', 'MINK7512', 'MINK123M', 'MINKGEM', 'MKOOPKLA', 'PWAPART',
       'PWABEDR', 'PWALAND', 'PPERSAUT', 'PBESAUT', 'PMOTSCO', 'PVRAAUT',
       'PAANHANG', 'PTRACTOR', 'PWERKT', 'PBROM', 'PLEVEN', 'PPERSONG',
       'PGEZONG', 'PWAOREG', 'PBRAND', 'PZEILPL', 'PPLEZIER', 'PFIETS',
       'PINBOED', 'PBYSTAND', 'AWAPART', 'AWABEDR', 'AWALAND', 'APERSAUT',
       'ABESAUT', 'AMOTSCO', 'AVRAAUT', 'AAANHANG', 'ATRACTOR', 'AWERKT',
       'ABROM', 'ALEVEN', 'APERSONG', 'AGEZONG', 'AWAOREG', 'ABRAND',
       'AZEILPL', 'APLEZIER', 'AFIETS', 'AINBOED', 'ABYSTAND', 'Purchase'],
      dtype='object')
Numerical Columns in the Dataset:
Index(['rownames', 'MOSTYPE', 'MAANTHUI', 'MGEMOMV', 'MGEMLEEF', 'MOSHOOFD',
       'MGODRK', 'MGODPR', 'MGODOV', 'MGODGE', 'MRELGE', 'MRELSA', 'MRELOV',
       'MFALLEEN', 'MFGEKIND', 'MFWEKIND', 'MOPLHOOG', 'MOPLMIDD', 'MOPLLAAG',
       'MBERHOOG', 'MBERZELF', 'MBERBOER', 'MBERMIDD', 'MBERARBG', 'MBERARBO',
       'MSKA', 'MSKB1', 'MSKB2', 'MSKC', 'MSKD', 'MHHUUR', 'MHKOOP', 'MAUT1',
       'MAUT2', 'MAUT0', 'MZFONDS', 'MZPART', 'MINKM30', 'MINK3045',
       'MINK4575', 'MINK7512', 'MINK123M', 'MINKGEM', 'MKOOPKLA', 'PWAPART',
       'PWABEDR', 'PWALAND', 'PPERSAUT', 'PBESAUT', 'PMOTSCO', 'PVRAAUT',
       'PAANHANG', 'PTRACTOR', 'PWERKT', 'PBROM', 'PLEVEN', 'PPERSONG',
       'PGEZONG', 'PWAOREG', 'PBRAND', 'PZEILPL', 'PPLEZIER', 'PFIETS',
       'PINBOED', 'PBYSTAND', 'AWAPART', 'AWABEDR', 'AWALAND', 'APERSAUT',
       'ABESAUT', 'AMOTSCO', 'AVRAAUT', 'AAANHANG', 'ATRACTOR', 'AWERKT',
       'ABROM', 'ALEVEN', 'APERSONG', 'AGEZONG', 'AWAOREG', 'ABRAND',
       'AZEILPL', 'APLEZIER', 'AFIETS', 'AINBOED', 'ABYSTAND'],
      dtype='object')
```

```
numerical_cols = df.select_dtypes(include=['int64', 'float64']).columns
df[numerical_cols] = df[numerical_cols].fillna(df[numerical_cols].mean())

numerical_cols = ['MINKM30', 'MINK3045', 'MINK4575', 'MINK7512', 'MINK123M', 'MINKGEM']
```

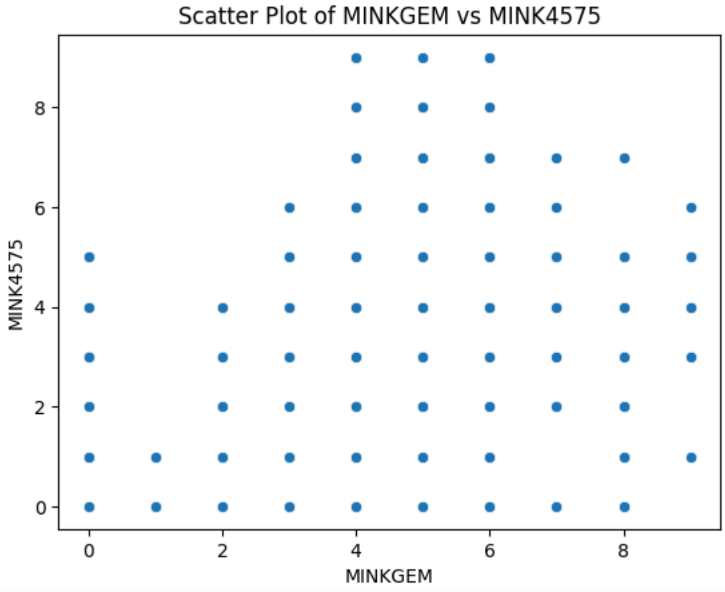
```
for col in numerical_cols:
    if col in df.columns:
        df[col] = df[col].fillna(df[col].mean())

categorical_cols = ['MOSTYPE', 'MOSHOOFD', 'MGODRK', 'MGODPR', 'MGODOV', 'MGODGE', 'MRELGE']
for col in categorical_cols:
    if col in df.columns:
        df[col] = df[col].fillna(df[col].mode()[0])
print("Remaining missing values:")
print(df.isnull().sum())
```

```
Remaining missing values:
rownames    0
MOSTYPE     0
MAANTHUI    0
MGEMOMV     0
MGEMLEEF    0
           ..
APLEZIER    0
AFIETS      0
AINBOED     0
ABYSTAND    0
Purchase    0
Length: 87, dtype: int64
```

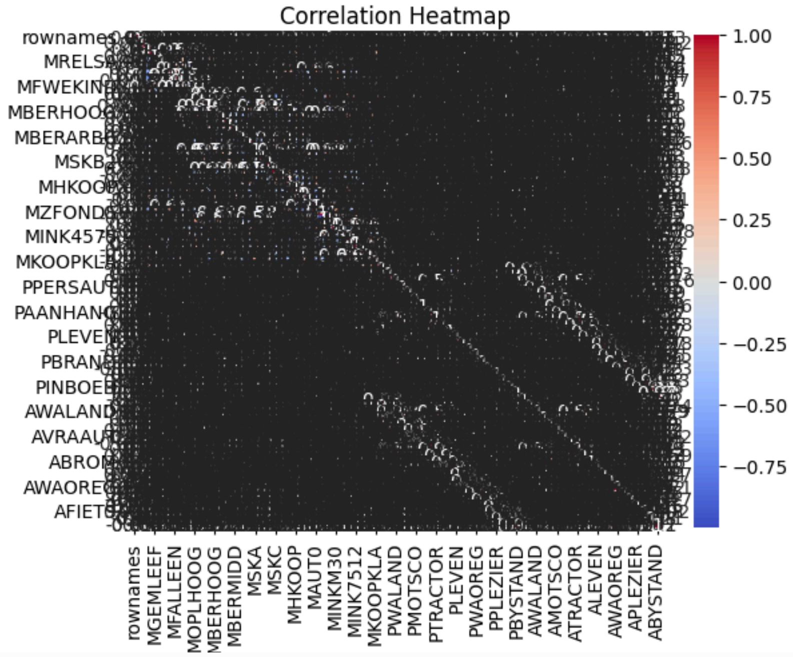
```
import matplotlib.pyplot as plt
df['MINKGEM'].hist(bins=10)
plt.title('Distribution of MINKGEM')
plt.xlabel('MINKGEM')
plt.ylabel('Frequency')
plt.show()
```



```
import seaborn as sns
sns.scatterplot(x='MINKGEM', y='MINK4575', data=df)
plt.title('Scatter Plot of MINKGEM vs MINK4575')
plt.show()
```



```
numeric_df = df.select_dtypes(include=['int64', 'float64'])
correlation = numeric_df.corr()
import seaborn as sns
import matplotlib.pyplot as plt
sns.heatmap(correlation, annot=True, cmap='coolwarm')
plt.title('Correlation Heatmap')
plt.show()
```



```
from sklearn.model_selection import train_test_split
X = df.drop('Purchase', axis=1)
y = df['Purchase']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report

rf = RandomForestClassifier(random_state=42)
rf.fit(X_train, y_train)

y_pred = rf.predict(X_test)

print("Classification Report:")
print(classification_report(y_test, y_pred))
```

```
Classification Report:
              precision    recall  f1-score   support

          No       0.93      0.99      0.96      1628
         Yes       0.33      0.04      0.07       119
```

```
    accuracy                           0.93      1747
   macro avg       0.63      0.52      0.52      1747
weighted avg       0.89      0.93      0.90      1747
```

```
df.to_csv(r"C:\Users\Dhruv\Downloads\Cleaned_Caravan.csv", index=False)
print("Cleaned dataset saved!")
```

```
Cleaned dataset saved!
```