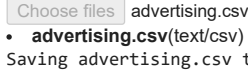
```
from google.colab import files


uploaded = files.upload()
```



```
Saving advertising.csv to advertising.csv
```

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error, r2_score

# Load the advertising dataset
advertising = pd.read_csv("advertising.csv")

# Display the first few rows of the dataset and basic information
print("First few rows of the dataset:")
print(advertising.head())
print("\nDataset shape:", advertising.shape)
print("\nDataset information:")
print(advertising.info())
print("\nSummary statistics:")
print(advertising.describe())

# Check for null values
null_percent = advertising.isnull().sum() * 100 / advertising.shape[0]
print("\nPercentage of null values in each column:")
print(null_percent)
print("\nThere are no NULL values in the dataset.")

# Visualize outliers using box plots
fig, axs = plt.subplots(3, figsize=(5, 5))
sns.boxplot(advertising['TV'], ax=axs[0])
sns.boxplot(advertising['Newspaper'], ax=axs[1])
sns.boxplot(advertising['Radio'], ax=axs[2])
plt.tight_layout()
plt.show()

sns.boxplot(advertising['Sales'])
plt.show()

# Explore relationships between variables using pair plots and a correlation heatmap
sns.pairplot(advertising, x_vars=['TV', 'Newspaper', 'Radio'], y_vars='Sales', height=4, aspect=1, kind='scatter')
plt.show()

sns.heatmap(advertising.corr(), cmap="YlGnBu", annot=True)
plt.show()

# Prepare data for modeling
X = advertising['TV']
y = advertising['Sales']

X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7, test_size=0.3, random_state=100)

# Perform linear regression using statsmodels
X_train_sm = sm.add_constant(X_train)
lr = sm.OLS(y_train, X_train_sm).fit()
print("Regression parameters:")
print(lr.params)
print("\nRegression summary:")
print(lr.summary())

# Visualize regression line and errors
plt.scatter(X_train, y_train)
plt.plot(X_train, lr.params[0] + lr.params[1] * X_train, 'r')
plt.show()

y_train_pred = lr.predict(X_train_sm)
residuals = (y_train - y_train_pred)
sns.distplot(residuals, bins=15)
plt.xlabel('y_train - y_train_pred', fontsize=15)
plt.show()

plt.scatter(X_train, residuals)
plt.show()
```

```
# Prepare test data for predictions
X_test_sm = sm.add_constant(X_test)
y_pred = lr.predict(X_test_sm)

# Calculate and display evaluation metrics
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
r_squared = r2_score(y_test, y_pred)
print("Root Mean Squared Error:", rmse)
print("R-squared:", r_squared)

# Visualize predictions and regression line for test data
plt.scatter(X_test, y_test)
plt.plot(X_test, lr.params[0] + lr.params[1] * X_test, 'r')
plt.show()
```

```
First few rows of the dataset:
      TV  Radio  Newspaper  Sales
0  230.1   37.8       69.2   22.1
1   44.5   39.3       45.1   10.4
2   17.2   45.9       69.3   12.0
3  151.5   41.3       58.5   16.5
4  180.8   10.8       58.4   17.9

Dataset shape: (200, 4)

Dataset information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 4 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   TV         200 non-null    float64
 1   Radio      200 non-null    float64
 2   Newspaper  200 non-null    float64
 3   Sales      200 non-null    float64
dtypes: float64(4)
memory usage: 6.4 KB
None

Summary statistics:
               TV       Radio   Newspaper       Sales
count  200.000000  200.000000  200.000000  200.000000
mean   147.042500   23.264000   30.554000   15.130500
std     85.854236   14.846809   21.778621    5.283892
min      0.700000    0.000000    0.300000    1.600000
25%     74.375000    9.975000   12.750000   11.000000
50%    149.750000   22.900000   25.750000   16.000000
75%    218.825000   36.525000   45.100000   19.050000
max    296.400000   49.600000  114.000000   27.000000

Percentage of null values in each column:
TV           0.0
Radio        0.0
Newspaper    0.0
Sales        0.0
dtype: float64

There are no NULL values in the dataset.
```