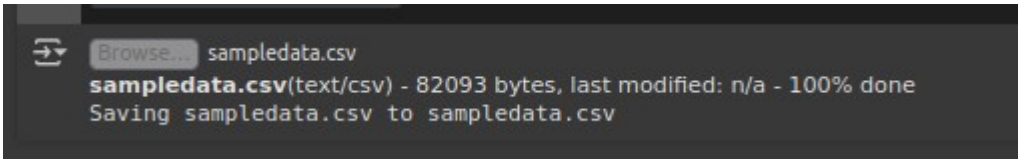
```
from google.colab import files
uploaded=files.upload()
```



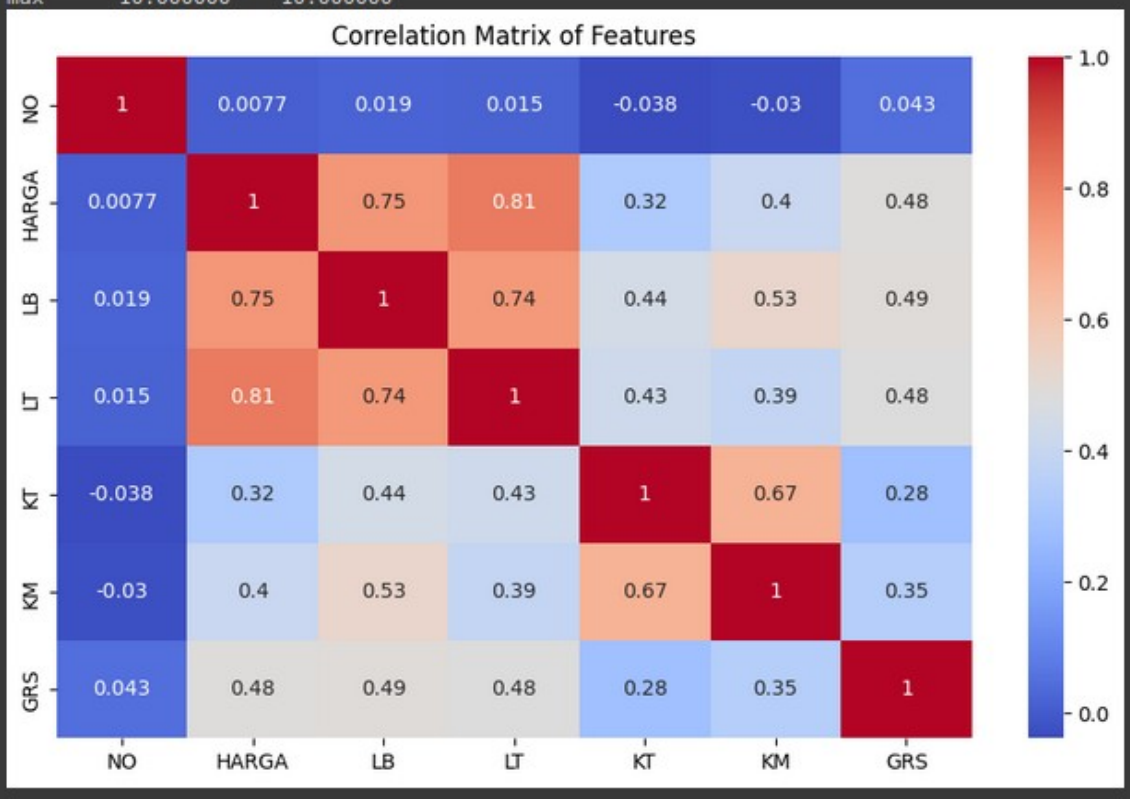

```
import matplotlib.pyplot as plt
import seaborn as sns

# Basic statistics
print(df.describe())

# Correlation matrix
plt.figure(figsize=(10, 6))
sns.heatmap(df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Matrix of Features")
plt.show()
```

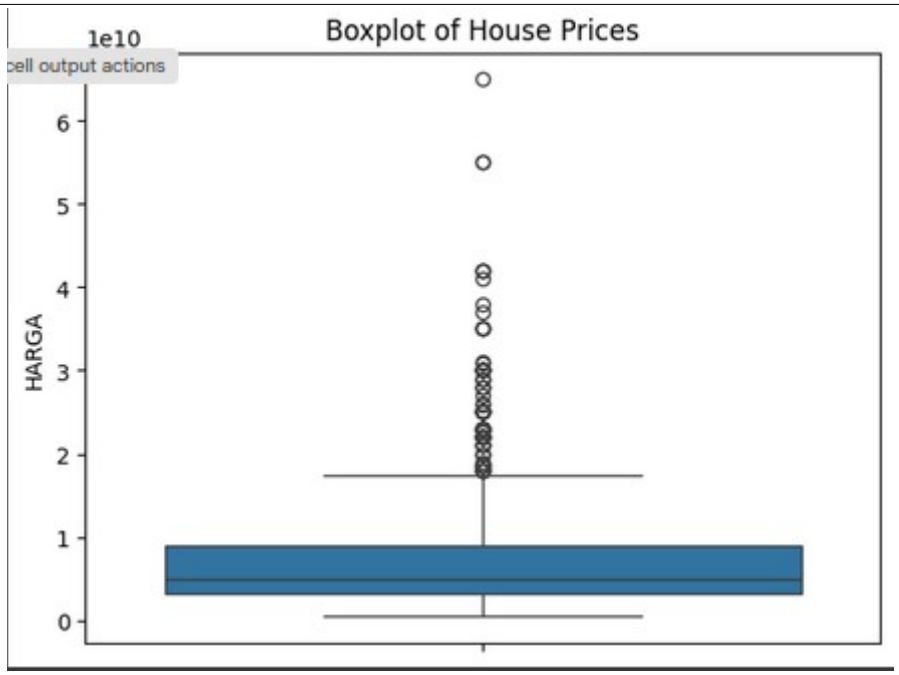
```
                NO         HARGA           LB           LT           KT  \
count  1010.000000  1.010000e+03  1010.000000  1010.000000  1010.000000
mean    505.500000  7.628987e+09   276.539604   237.432673     4.668317
std     291.706188  7.340946e+09   177.864557   179.957604     1.572776
min       1.000000  4.300000e+08    40.000000    25.000000     2.000000
25%     253.250000  3.262500e+09   150.000000   130.000000     4.000000
50%     505.500000  5.000000e+09   216.500000   165.000000     4.000000
75%     757.750000  9.000000e+09   350.000000   290.000000     5.000000
max    1010.000000  6.500000e+10  1126.000000  1400.000000    10.000000

                KM          GRS
count  1010.000000  1010.000000
mean      3.607921     1.920792
std       1.420066     1.510998
min       1.000000     0.000000
25%       3.000000     1.000000
50%       3.000000     2.000000
75%       4.000000     2.000000
max      10.000000    10.000000
```




```
# Boxplot to detect price outliers
sns.boxplot(df['HARGA'])
plt.title("Boxplot of House Prices")
plt.show()

# Remove extreme outliers
df_clean = df[df['HARGA'] < df['HARGA'].quantile(0.95)]
```




```
from sklearn.linear_model import LinearRegression
import numpy as np

X = df[['LB']]
y = df['HARGA']

model = LinearRegression()
model.fit(X, y)

print("R^2 Score:", model.score(X, y))
```

```
R^2 Score: 0.5581327856561413
```

```
features = ['LB', 'LT', 'KT', 'KM', 'GRS']
X = df[features]
y = df['HARGA']

model = LinearRegression()
model.fit(X, y)

print("Model coefficients:", model.coef_)
print("R^2 score:", model.score(X, y))
```

```
Model coefficients: [ 1.23187516e+07  2.36590867e+07 -6.19514797e+08  4.55486747e+08
  3.09965160e+08]
R^2 score: 0.7162361438094645
```

```
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline

poly_model = make_pipeline(PolynomialFeatures(2), LinearRegression())
poly_model.fit(X, y)

print("R^2 score (poly):", poly_model.score(X, y))
```

```
R^2 score (poly): 0.7384637627019939
```

```
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

model = LinearRegression()
model.fit(X_train, y_train)
pred = model.predict(X_test)

print("Test RMSE:", np.sqrt(mean_squared_error(y_test, pred)))
```

```
Test RMSE: 2986616943.9349093
```

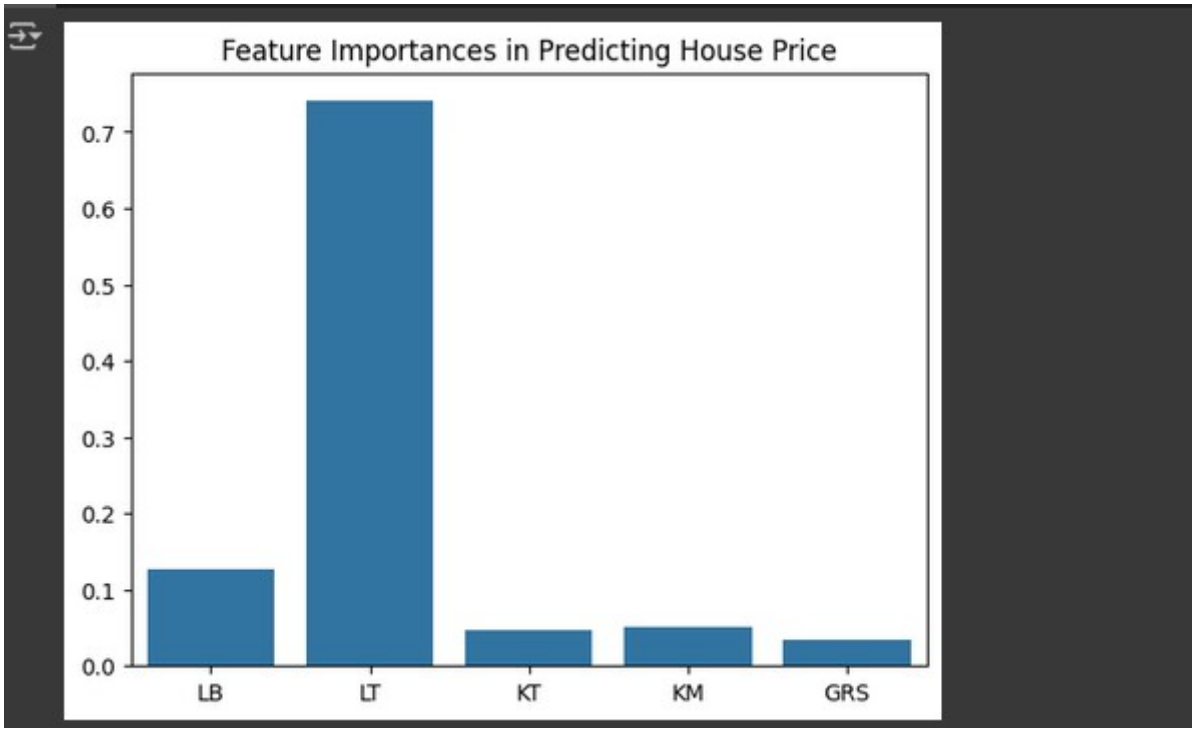
```
from sklearn.ensemble import RandomForestRegressor

rf = RandomForestRegressor(n_estimators=100)
rf.fit(X_train, y_train)


print("Random Forest R^2:", rf.score(X_test, y_test))
```

```
Random Forest R^2: 0.8498985772208579
```

```
importances = rf.feature_importances_
sns.barplot(x=features, y=importances)
plt.title("Feature Importances in Predicting House Price")
plt.show()
```




```
sample = pd.DataFrame({
    'LB': [200],
    'LT': [150],
    'KT': [4],
    'KM': [3],
    'GRS': [1]
})

predicted_price = rf.predict(sample)
print("Predicted House Price:", predicted_price[0])
```

```
Predicted House Price: 4644500000.0
```

```
import joblib

joblib.dump(rf, 'house_price_model.pkl')
print("Model saved as 'house_price_model.pkl'")
```