```
import os
import numpy as np
import pandas as pd
from scipy.stats import skew, kurtosis
from sklearn.datasets import load_diabetes
from sklearn.linear_model import LinearRegression, LogisticRegression
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import r2_score, mean_squared_error, accuracy_score, roc_auc_score,
confusion_matrix, classification_report
import statsmodels.api as sm

uci = load_diabetes()
df_uci = pd.DataFrame(uci.data, columns=uci.feature_names)
df_uci["target"] = uci.target

pima_path = "diabetes.csv"
if not os.path.exists(pima_path):
    raise FileNotFoundError("Please place 'diabetes.csv' in the same folder as this script.")
df_pima = pd.read_csv(pima_path)

if "Outcome" not in df_pima.columns:
    df_pima.rename(columns={df_pima.columns[-1]: "Outcome"}, inplace=True)

def univariate_analysis(df, name):
    print(f"\n===== Univariate Analysis: {name} =====")
    results = []
    for col in df.select_dtypes(include=[np.number]).columns:
        data = df[col].dropna()
        results.append({
            "Feature": col,
```

```
            "Count": len(data),
            "Mean": data.mean(),
            "Median": data.median(),
            "Mode": data.mode().iloc[0] if not data.mode().empty else np.nan,
            "Variance": data.var(),
            "Std Dev": data.std(),
            "Skewness": skew(data),
            "Kurtosis": kurtosis(data)
        })
    return pd.DataFrame(results)

uni_uci = univariate_analysis(df_uci, "UCI Diabetes")
uni_pima = univariate_analysis(df_pima, "Pima Indians Diabetes")

print("\n===== Bivariate Analysis: UCI Diabetes (Linear Regression) =====")
X = df_uci[["bmi"]]  # using 'bmi' as strongest predictor
y = df_uci["target"]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
lin_reg = LinearRegression().fit(X_train, y_train)
y_pred = lin_reg.predict(X_test)
print(f"R²: {r2_score(y_test, y_pred):.3f}")
print(f"RMSE: {np.sqrt(mean_squared_error(y_test, y_pred)):.3f}")

print("\n===== Bivariate Analysis: Pima (Logistic Regression) =====")
X_pima = df_pima[["Glucose"]]  # using Glucose as single predictor
y_pima = df_pima["Outcome"]
Xp_train, Xp_test, yp_train, yp_test = train_test_split(X_pima, y_pima, test_size=0.2,
random_state=42, stratify=y_pima)
log_reg = LogisticRegression(max_iter=1000).fit(Xp_train, yp_train)
yp_pred = log_reg.predict(Xp_test)
yp_prob = log_reg.predict_proba(Xp_test)[:, 1]
```

```
print(f"Accuracy: {accuracy_score(yp_test, yp_pred):.3f}")
print(f"AUC: {roc_auc_score(yp_test, yp_prob):.3f}")

print("\n===== Multiple Regression: UCI Diabetes =====")
X_all = sm.add_constant(df_uci.drop(columns=["target"]))
model = sm.OLS(df_uci["target"], X_all).fit()
print(model.summary())

print("\n===== Multiple Logistic Regression: Pima =====")
X_pima_all = df_pima.drop(columns=["Outcome"])
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X_pima_all)
Xp_train, Xp_test, yp_train, yp_test = train_test_split(X_scaled, y_pima, test_size=0.2,
random_state=42, stratify=y_pima)
log_reg_full = LogisticRegression(max_iter=2000).fit(Xp_train, yp_train)
yp_pred_full = log_reg_full.predict(Xp_test)
yp_prob_full = log_reg_full.predict_proba(Xp_test)[:, 1]
print(f"Accuracy: {accuracy_score(yp_test, yp_pred_full):.3f}")
print(f"AUC: {roc_auc_score(yp_test, yp_prob_full):.3f}")
print(confusion_matrix(yp_test, yp_pred_full))
print(classification_report(yp_test, yp_pred_full))
print("\n===== COMPARISON =====")
print(f"UCI R² (Multiple Regression): {model.rsquared:.3f}")
print(f"Pima Accuracy (Multiple Logistic): {accuracy_score(yp_test, yp_pred_full):.3f}, AUC:
{roc_auc_score(yp_test, yp_prob_full):.3f}")
```

Output:-

```
===== Bivariate Analysis: UCI Diabetes (Linear Regression) =====
R²: 0.233
RMSE: 63.732

===== Bivariate Analysis: Pima (Logistic Regression) =====
```

Accuracy: 0.708

AUC: 0.767

```
===== Multiple Regression: UCI Diabetes =====
                            OLS Regression Results
==============================================================================
Dep. Variable:                 target   R-squared:                       0.518
Model:                            OLS   Adj. R-squared:                  0.507
Method:                 Least Squares   F-statistic:                     46.27
Date:                Thu, 11 Sep 2025   Prob (F-statistic):           3.83e-62
Time:                        14:14:01   Log-Likelihood:                -2386.0
No. Observations:                 442   AIC:                             4794.
Df Residuals:                     431   BIC:                             4839.
Df Model:                          10
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const        152.1335      2.576     59.061      0.000     147.071     157.196
age          -10.0099     59.749     -0.168      0.867    -127.446     107.426
sex         -239.8156     61.222     -3.917      0.000    -360.147    -119.484
bmi          519.8459     66.533      7.813      0.000     389.076     650.616
bp           324.3846     65.422      4.958      0.000     195.799     452.970
s1          -792.1756    416.680     -1.901      0.058   -1611.153      26.802
s2           476.7390    339.030      1.406      0.160    -189.620    1143.098
s3           101.0433    212.531      0.475      0.635    -316.684     518.770
s4           177.0632    161.476      1.097      0.273    -140.315     494.441
s5           751.2737    171.900      4.370      0.000     413.407    1089.140
s6            67.6267     65.984      1.025      0.306     -62.064     197.318
==============================================================================
Omnibus:                        1.506   Durbin-Watson:                   2.029
Prob(Omnibus):                  0.471   Jarque-Bera (JB):                1.404
Skew:                           0.017   Prob(JB):                        0.496
Kurtosis:                       2.726   Cond. No.                         227.
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

===== Multiple Logistic Regression: Pima =====
Accuracy: 0.714
AUC: 0.823
[[82 18]
 [26 28]]
              precision    recall  f1-score   support

           0       0.76      0.82      0.79       100
           1       0.61      0.52      0.56        54

    accuracy                           0.71       154
   macro avg       0.68      0.67      0.67       154
weighted avg       0.71      0.71      0.71       154
```

===== COMPARISON =====

UCI R^2 (Multiple Regression): 0.518

Pima Accuracy (Multiple Logistic): 0.714, AUC: 0.823