Advanced Real-Life EDA Assignment

Part 1: Numeric Data Analysis

Problem 1: Real Estate Price Analysis

- **Problem Statement**: Analyze a real estate dataset to determine the central tendency and dispersion of house prices in a particular city. Calculate the mean, median, mode, standard deviation, variance, and range of house prices.
- **Hint**: Use Python libraries like NumPy and Pandas for calculations. The dataset can be found here.

python
```
import numpy as np
import pandas as pd

# Load dataset
url = 'https://raw.githubusercontent.com/ageron/handson-
ml/master/datasets/housing/housing.csv'
data = pd.read_csv(url)

# Calculations
mean = data['median_house_value'].mean()
median = data['median_house_value'].median()
mode = data['median_house_value'].mode()[0]
std_dev = data['median_house_value'].std()
variance = data['median_house_value'].var()
data_range = data['median_house_value'].max() -
data['median_house_value'].min()

print(f"Mean: {mean}, Median: {median}, Mode: {mode}, Std Dev: {std_dev},
Variance: {variance}, Range: {data_range}")
```

Part 2: Categorical Data Analysis

Problem 2: Customer Purchase Behavior

- **Problem Statement**: Analyze customer purchase behavior from an e-commerce dataset to identify the frequency of different product categories purchased. Create a frequency table for the product categories.
- **Hint**: Use the `value_counts()` function in Pandas. The dataset can be found here.

python
```
import pandas as pd

# Load dataset
url = 'https://raw.githubusercontent.com/databricks/learning-
spark/master/data/retail-data/all/2010-12-01.csv'
data = pd.read_csv(url)

# Frequency table
frequency_table = data['StockCode'].value_counts()

print(frequency_table.head(10))
```

Part 3: Applied Visualization for EDA

Problem 3: Car Data Visualization

- **Problem Statement**: Visualize the relationship between car horsepower and miles per gallon (MPG) using scatterplots. Identify any patterns or correlations.
- **Hint**: Use the `scatterplot()` function from Seaborn. The dataset can be found here.

python
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load dataset
url = 'https://archive.ics.uci.edu/ml/machine-learning-
databases/autos/imports-85.data'
columns = ['symboling', 'normalized_losses', 'make', 'fuel_type',
'aspiration', 'num_doors',
           'body_style', 'drive_wheels', 'engine_location', 'wheel_base',
'length', 'width',
           'height', 'curb_weight', 'engine_type', 'num_cylinders',
'engine_size', 'fuel_system',
           'bore', 'stroke', 'compression_ratio', 'horsepower', 'peak_rpm',
'city_mpg', 'highway_mpg', 'price']
data = pd.read_csv(url, names=columns)

# Scatterplot
sns.scatterplot(x='horsepower', y='city_mpg', data=data)
plt.title('Horsepower vs. City MPG')
plt.xlabel('Horsepower')
plt.ylabel('City MPG')
plt.show()
```

Part 4: Correlation Analysis

Problem 4: Health Data Correlation

- **Problem Statement**: Analyze a health dataset to determine the correlation between BMI (Body Mass Index) and various health indicators such as cholesterol level and blood pressure. Create a heatmap to visualize the correlation matrix.
- **Hint**: Use the `heatmap()` function from Seaborn. The dataset can be found here.

python
```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

# Load dataset
url = 'https://raw.githubusercontent.com/stedy/Machine-Learning-with-R-
datasets/master/diabetes.csv'
data = pd.read_csv(url)

# Correlation matrix
corr_matrix = data[['BMI', 'BloodPressure', 'Cholesterol',
'Glucose']].corr()
```

```
# Heatmap
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm')
plt.title('Correlation Matrix Heatmap')
plt.show()
```

Part 5: Normality and Distribution

Problem 5: Financial Data Normality

- **Problem Statement**: Analyze a financial dataset to check the normality of stock returns. Create a QQ plot and calculate Z-scores for the returns.
- **Hint**: Use the `probplot()` function from SciPy. The dataset can be found here.

python

```
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as stats

# Load dataset
url = 'https://raw.githubusercontent.com/plotly/datasets/master/finance-
charts-apple.csv'
data = pd.read_csv(url)

# Calculate returns
data['Return'] = data['AAPL.Close'].pct_change().dropna()

# QQ Plot
stats.probplot(data['Return'].dropna(), dist="norm", plot=plt)
plt.title('QQ Plot for Stock Returns')
plt.show()

# Z-scores
z_scores = stats.zscore(data['Return'].dropna())
print(f"Z-scores:\n{z_scores}")
```