```
# -*- coding: utf-8 -*-
"""Assignment_2_Group_ECE14.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1rF9HFV9BOp-
EkHDB0XiFvO1KxEaKIOPd

# Assignment 2
# Data Visualization and Pre-processing

## 1. Perform Below Visualizations.
### Univariate Analysis
#### 1. Summary Statistics
"""

file_data = pd.read_csv('C:\Kavin\Churn_Modelling.csv')
file_data

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm

file_data['Balance'].mean()

file_data['Balance'].median()

file_data['Balance'].std()

"""#### 2. Frequency Table"""

file_data['Surname'].value_counts()

"""#### 3. Create Charts"""

file_data.boxplot(column=['Balance'], grid=False)

file_data.hist(column='Balance', grid=False, edgecolor='black')

sns.kdeplot(file_data['Balance'])

"""### Bi - Variate Analysis

#### 1. Scatterplots
"""

plt.scatter(file_data.CreditScore.head(100), file_data.Age.head(100))
plt.title('Scatter')
plt.xlabel('CreditScore')
plt.ylabel('Age')
```

```
"""#### 2. Correlation Coefficients"""

file_data.corr()

"""#### 3. Simple Linear Regression"""

y = file_data['CustomerId']
x = file_data['HasCrCard']
x = sm.add_constant(x)
model = sm.OLS(y,x).fit()
model.summary()

plt.plot(file_data['RowNumber'].head() ,file_data['CreditScore'].head(),
)

plt.title('Line plot')
plt.xlabel('RowNumber')
plt.ylabel('CreditScore')

"""### Multi - Variate Analysis"""

f = plt.subplots(figsize=(12,10))
sns.heatmap(file_data.head().corr(), cmap="YlGnBu")

corrmat = file_data.corr(method='spearman')
cg = sns.clustermap(corrmat, cmap="YlGnBu", linewidths=0.1);
plt.setp(cg.ax_heatmap.yaxis.get_majorticklabels(), rotation=0)
cg

"""## 4. Perform descriptive statistics on the dataset.

"""

file_data.shape

file_data.info()

file_data.describe()

file_data.head()

file_data.tail()

file_data.mean(numeric_only=True)

file_data.median(numeric_only=True)

file_data.mode()

file_data.var(numeric_only=True)

file_data.std(numeric_only=True)

file_data.skew(numeric_only=True)
```

```
file_data.kurt(numeric_only=True)

quantile = file_data['Balance'].quantile(q=[0.75, 0.25])
quantile

x = file_data.Balance
sns.boxplot(x=x)

"""## 5. Handle the Missing values."""

print(file_data.isnull())

print(file_data.isnull().sum())

file_data.isna().any()

"""## 6. Find the outliers and replace the outliers"""

x = sns.boxplot(x=file_data["Age"])
x

x = file_data.Age
sns.boxplot(x=x)

x = np.where(file_data['Age']>57,39, file_data['Age'])

sns.boxplot(x=x)

"""## 7. Check for Categorical columns and perform encoding."""

pd.Categorical(file_data["Geography"])

# One Hot Encoding

pd.get_dummies(file_data["Geography"]).head(10)

pd.get_dummies(file_data).head(10)

"""## 8. Split the data into dependent and independent variables."""

# Splitting the Dataset into the Independent

X = file_data.iloc[:, :-1].values
print(X)

# Extracting the Dataset to Get the Dependent

Y = file_data.iloc[:, -1].values
print(Y)

"""## 9. Scale the independent variables"""

from sklearn.preprocessing import scale
```

```
x = scale(file_data["EstimatedSalary"])
x

"""## 10. Split the data into training and testing"""

from sklearn.model_selection import train_test_split

x = file_data.drop("EstimatedSalary", axis=1)
x

y = file_data.EstimatedSalary
y

x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2)

print(x_train.shape, x_test.shape)
```