Future Intern Project of Data Analytics Task 2

Task 1: Calculate summary statistics (Mean, Median, Mode, Standard Deviation

For a dataset

Steps:

1. Import Packages and Train dataset and load and display

1. Train Dataset

```
[1]: #Import all required libraries
     import pandas as pd
     import numpy as np
```

```
[9]: #Read the Dataset
     data = pd.read_csv('train (1).csv')
```

```
[11]: #First rows of Dataset
      data.head()
```

[11]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S

2. First rows of Dataset

```
[15]: #Check for Data types
      data.dtypes
```

```
[15]: PassengerId      int64
      Survived         int64
      Pclass           int64
      Name            object
      Sex             object
      Age            float64
      SibSp            int64
      Parch            int64
      Ticket          object
      Fare           float64
      Cabin           object
      Embarked        object
      dtype: object
```

3. Get Identify Categorical and Numerical Columns

[17]:
```
#Idetify the Categorical Columns
categorical_columns = data.select_dtypes (include=['object','category']).columns.tolist()

#Identify the Numerical Columns
numerical_columns = data.select_dtypes(include=['int64','float64']).columns.tolist()
```

[23]:
```
print("Categorical_columns:")
print(categorical_columns)
```

```
Categorical_columns:
['Name', 'Sex', 'Ticket', 'Cabin', 'Embarked']
```

[25]:
```
print("Numerical_columns")
print(numerical_columns)
```

```
Numerical_columns
['PassengerId', 'Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']
```

[35]:
```
columns_used = ['PassengerId', 'Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']
selected_columns = data.loc[:, columns_used]
```

[37]:
```
selected_columns.head(5)
```

[37]:

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
0	1	0	3	22.0	1	0	7.2500
1	2	1	1	38.0	1	0	71.2833
2	3	1	3	26.0	0	0	7.9250
3	4	1	1	35.0	1	0	53.1000
4	5	0	3	35.0	0	0	8.0500

4. Lets check for mean for each column

```
#Lets check for mean for each column
column_means = selected_columns.mean()
print("Mean for each column:",column_means)
```

```
Mean for each column: PassengerId    446.000000
Survived           0.383838
Pclass             2.308642
Age               29.699118
SibSp              0.523008
Parch              0.381594
Fare              32.204208
dtype: float64
```

5. Lets check for mode for each column

```
#Lets check for mode for each column
column_modes = selected_columns.mode().iloc[0]
print("Mode for each column:",column_modes)
```

```
Mode for each column: PassengerId     1.00
Survived         0.00
Pclass           3.00
Age             24.00
SibSp            0.00
Parch            0.00
Fare             8.05
Name: 0, dtype: float64
```

6. Lets check for median for each column

```
#Lets check for median for each column
column_median = selected_columns.median()
print("Median for each column:",column_median)
```

```
Median for each column: PassengerId    446.0000
Survived         0.0000
Pclass           3.0000
Age             28.0000
SibSp            0.0000
Parch            0.0000
Fare            14.4542
dtype: float64
```

7. Lets check for standard deviation for each column

```
#Lets check for standard deviation for each column
column_std_deviation = selected_columns.std()
print("Standard Deviation for each column:",column_std_deviation)
```

```
Standard Deviation for each column: PassengerId    257.353842
Survived         0.486592
Pclass           0.836071
Age             14.526497
SibSp            1.102743
Parch            0.806057
Fare            49.693429
dtype: float64
```

8. Check for statistics for each numeric column all together

```
#Check for statistics for each nnumeric column all together
selected_columns.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

9. Check for Stats data

```
[51]: #Check for Stats data
data.head(4)
```

[51]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S

[53]:
```
#Check for two unique variables
data['Sex'].unique()
```

[53]:
```
array(['male', 'female'], dtype=object)
```

[55]:
```
#We have two variables
data['Survived'].unique()
```

[55]:
```
array([0, 1], dtype=int64)
```

[59]:
```
data[["Sex","Age"]].groupby("Sex").mean()
```

[59]:

Sex	Age
female	27.915709
male	30.726645

[61]:
```
data[["Survived","Age"]].groupby("Survived").mean()
```

[61]:

Survived	Age
0	30.626179
1	28.343690

[63]:
```
data.groupby(["Sex","Survived"])["Age"].mean()
```

[63]:
```
Sex     Survived
female  0           25.046875
        1           28.847716
male    0           31.618056
        1           27.276022
Name: Age, dtype: float64
```

By

Shaikh Jasmin Kauser