Future Intern as Data Analysis

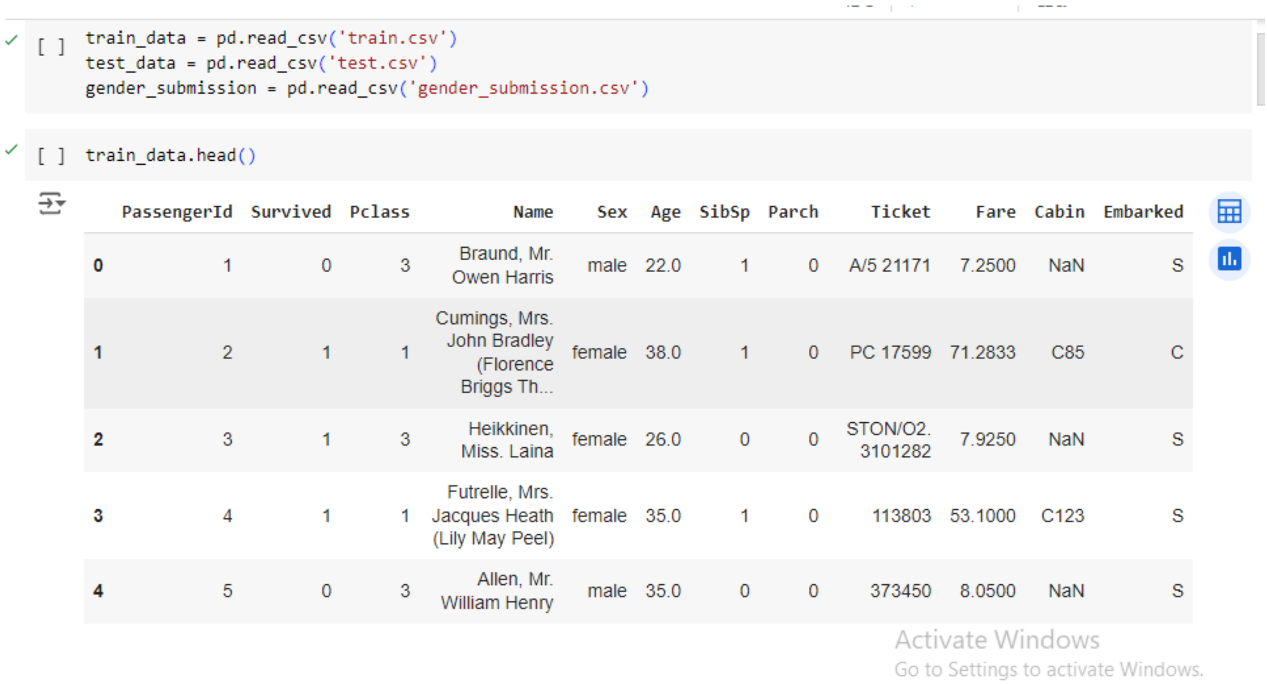
Task 2

Task: Calculate summary statistics (Mean, Mode ,Median And Standard Deviation for a dataset

Solution:

Step:

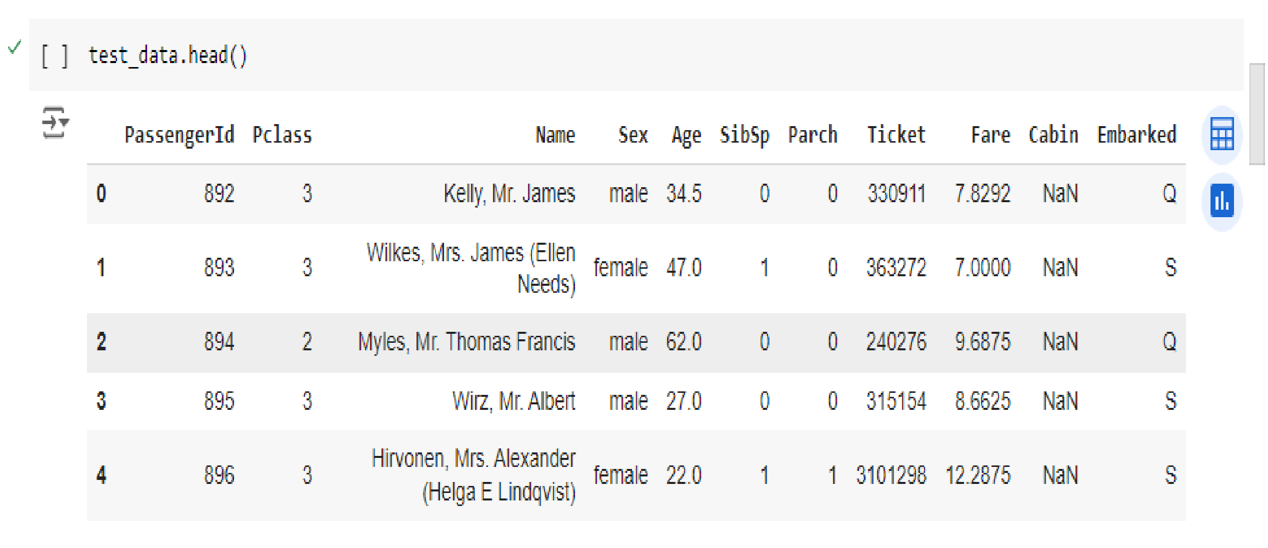
1. Import packages and display train dataset and test dataset



```
[ ] train_data = pd.read_csv('train.csv')
    test_data = pd.read_csv('test.csv')
    gender_submission = pd.read_csv('gender_submission.csv')
```

```
[ ] train_data.head()
```

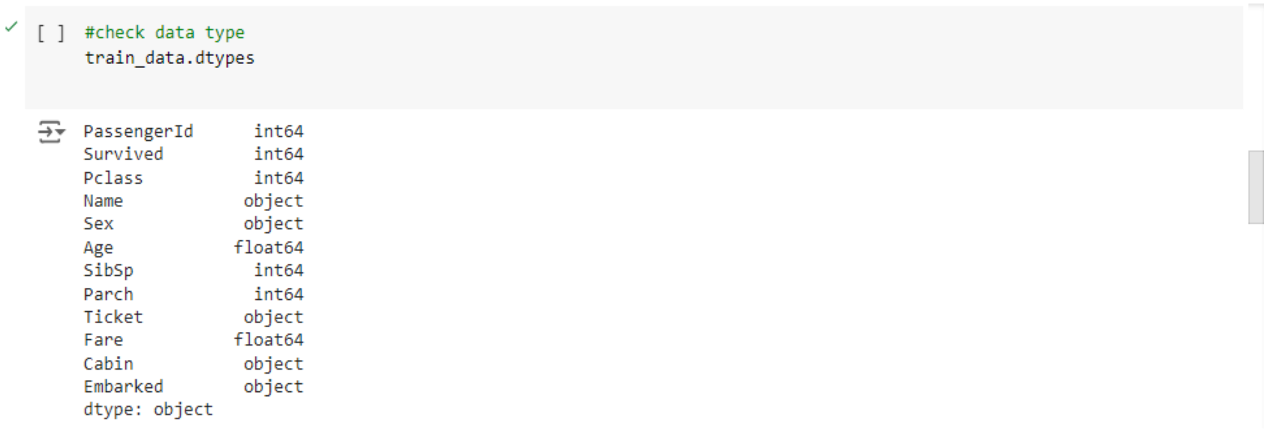
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 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



```
[ ] test_data.head()
```

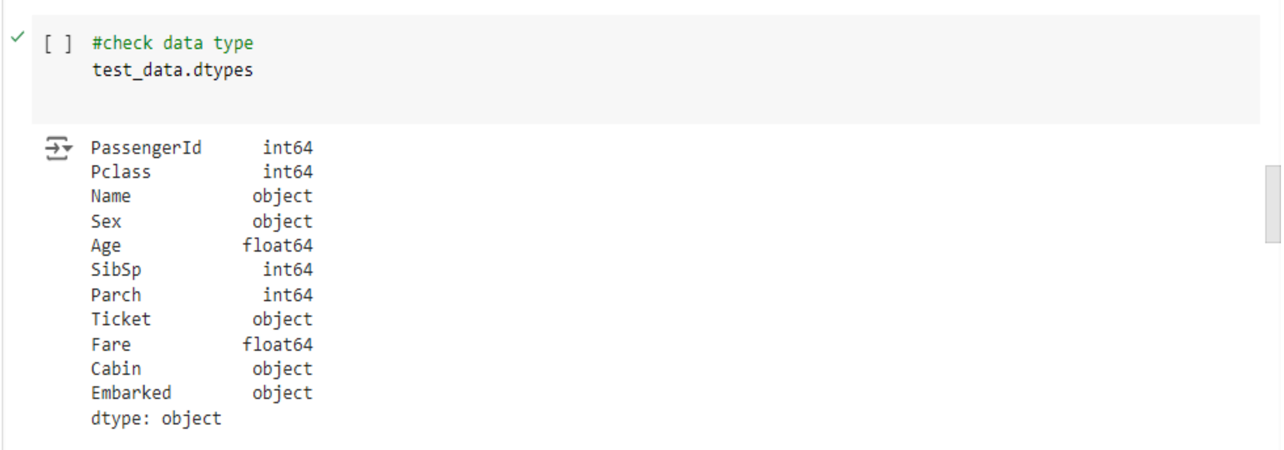
	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

2. Check data type of train dataset and test dataset



```
#check data type
train_data.dtypes
```

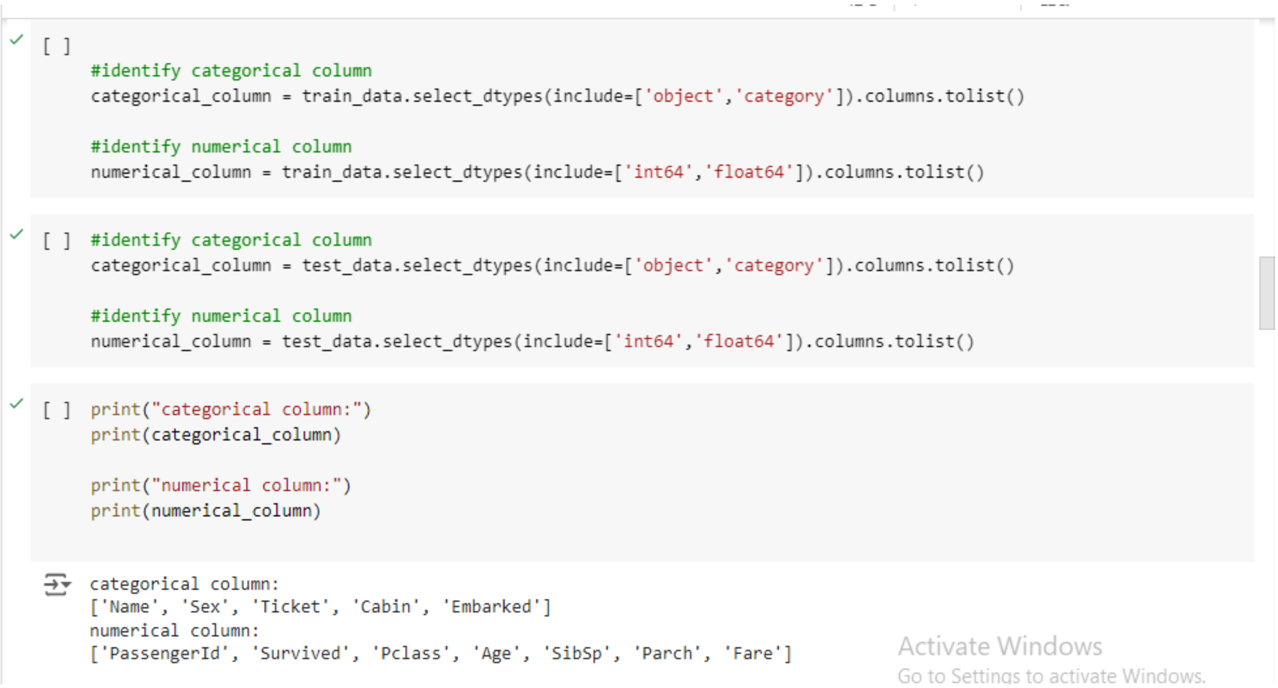
```
PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```



```
#check data type
test_data.dtypes
```

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

3. Identify categorical and numerical columns



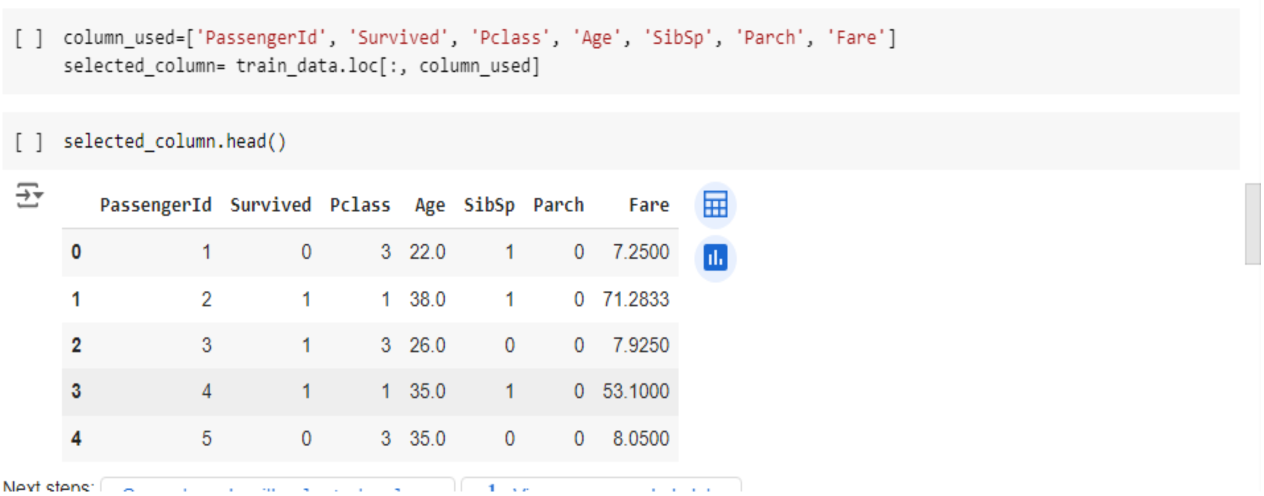
```
#identify categorical column
categorical_column = train_data.select_dtypes(include=['object','category']).columns.tolist()

#identify numerical column
numerical_column = train_data.select_dtypes(include=['int64','float64']).columns.tolist()
```

```
#identify categorical column
categorical_column = test_data.select_dtypes(include=['object','category']).columns.tolist()

#identify numerical column
numerical_column = test_data.select_dtypes(include=['int64','float64']).columns.tolist()
```

```
print("categorical column:")
print(categorical_column)

print("numerical column:")
print(numerical_column)
```

```
categorical column:
['Name', 'Sex', 'Ticket', 'Cabin', 'Embarked']
numerical column:
['PassengerId', 'Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']
```

4. Select columns where we perform staticstics



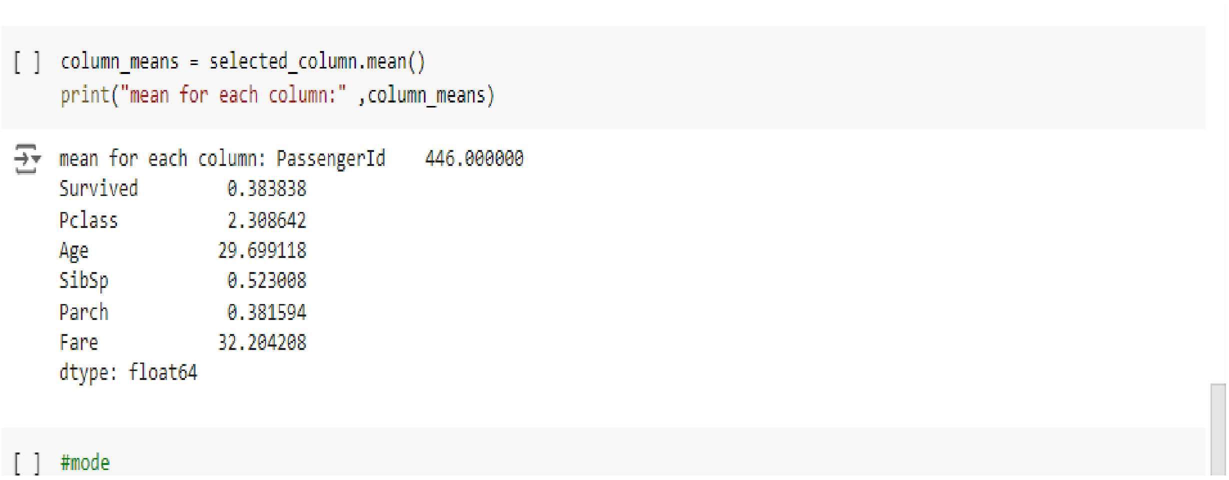
```
[ ]  column_used=['PassengerId', 'Survived', 'Pclass', 'Age', 'SibSp', 'Parch', 'Fare']
     selected_column= train_data.loc[:, column_used]
```

```
[ ]  selected_column.head()
```

	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

Next steps:

5. Check Mean for each column

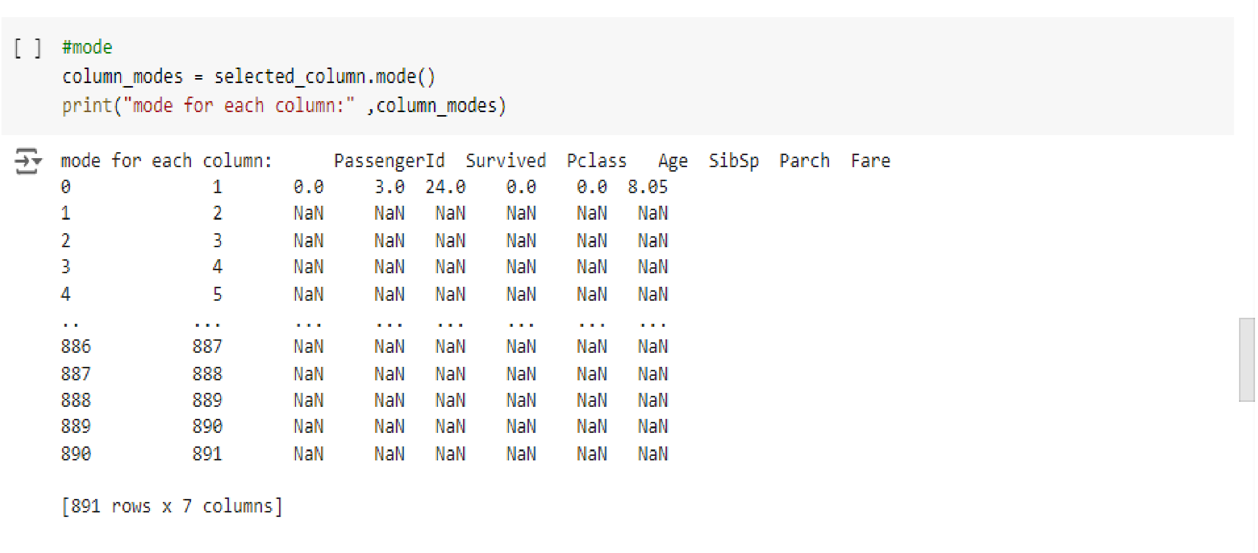


```
[ ]  column_means = selected_column.mean()
     print("mean for each column:" ,column_means)
```

```
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
```

```
[ ]  #mode
```

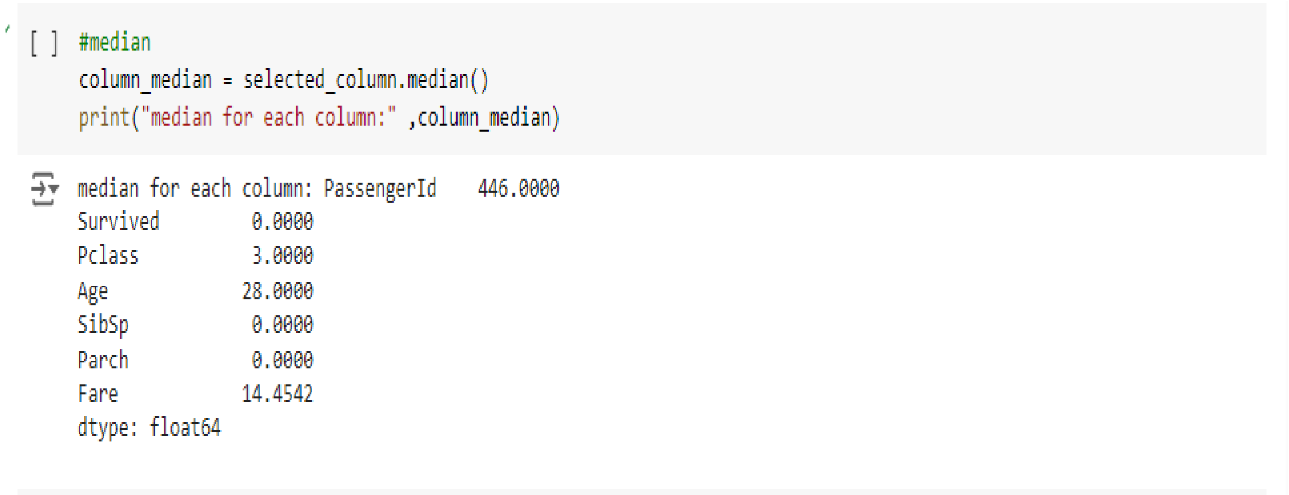
6. Check Mode for each column



```
[ ]  #mode
     column_modes = selected_column.mode()
     print("mode for each column:" ,column_modes)
```

```
mode for each column:      PassengerId  Survived  Pclass   Age  SibSp  Parch  Fare
0              1       0.0     3.0  24.0    0.0    0.0  8.05
1              2       NaN     NaN   NaN    NaN    NaN   NaN
2              3       NaN     NaN   NaN    NaN    NaN   NaN
3              4       NaN     NaN   NaN    NaN    NaN   NaN
4              5       NaN     NaN   NaN    NaN    NaN   NaN
..           ...       ...     ...   ...    ...    ...   ...
886          887       NaN     NaN   NaN    NaN    NaN   NaN
887          888       NaN     NaN   NaN    NaN    NaN   NaN
888          889       NaN     NaN   NaN    NaN    NaN   NaN
889          890       NaN     NaN   NaN    NaN    NaN   NaN
890          891       NaN     NaN   NaN    NaN    NaN   NaN

[891 rows x 7 columns]
```

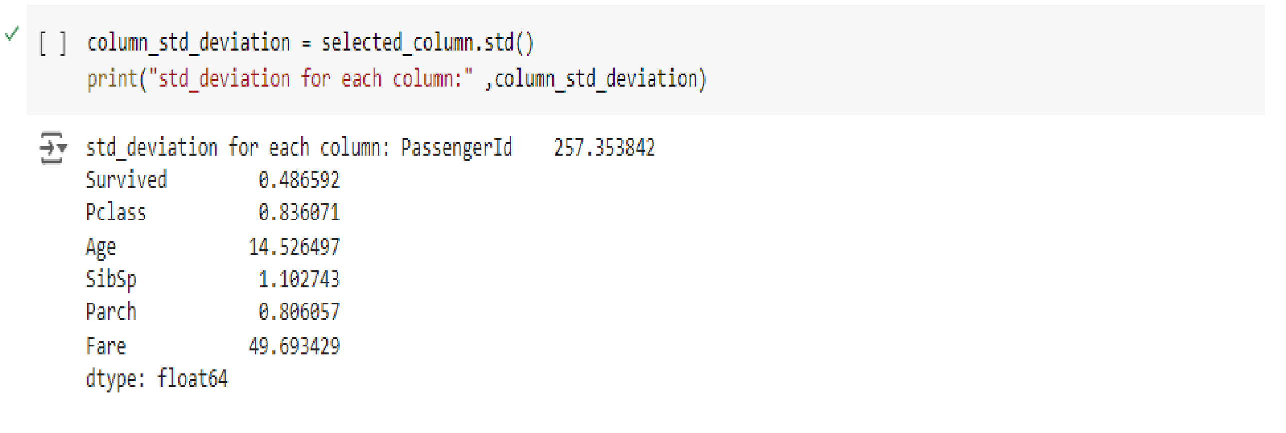
7. Check Median for each column



```
[ ] #median
    column_median = selected_column.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
```

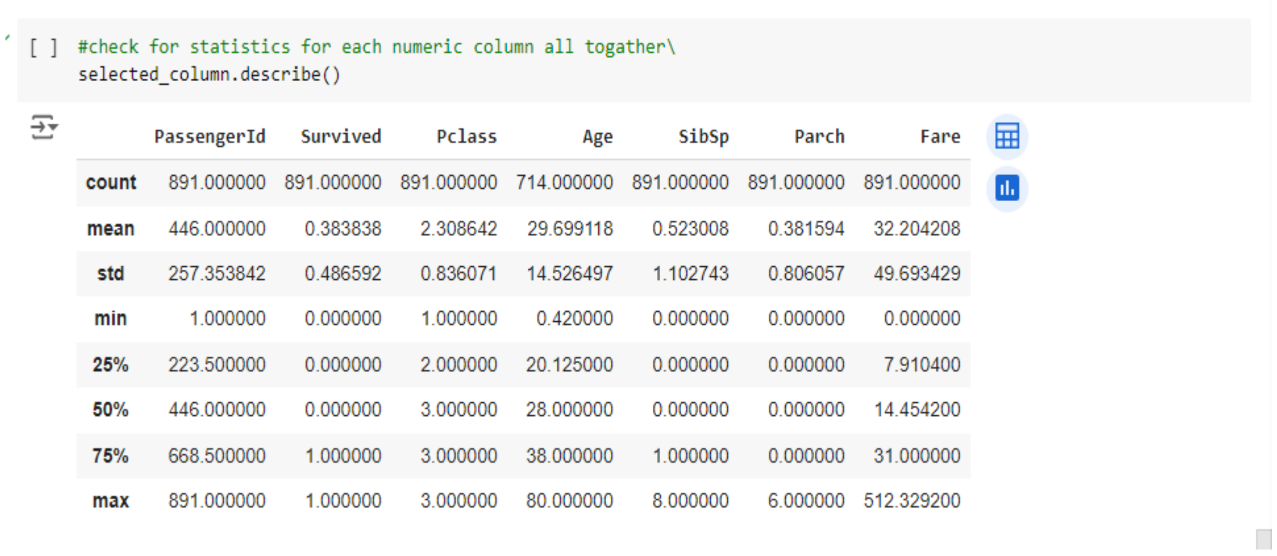
8. Check Mean for each column



```
[ ] column_std_deviation = selected_column.std()
    print("std_deviation for each column:" ,column_std_deviation)
```

```
std_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
```

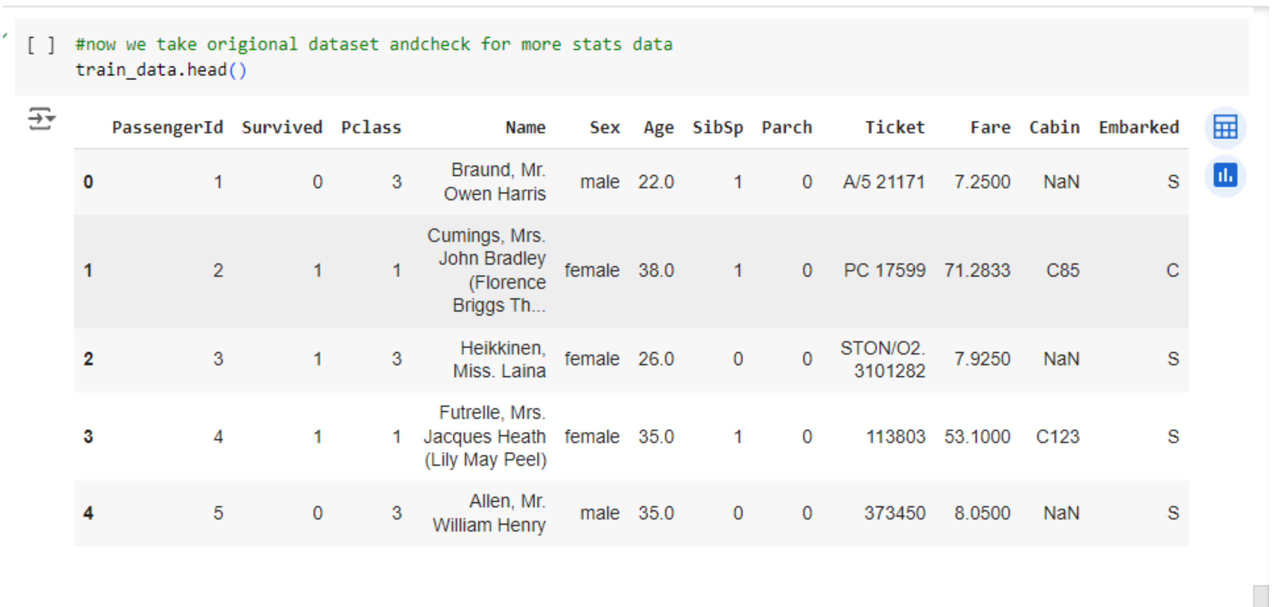
9. Check the statistics for each numeric column



```
[ ] #check for statistics for each numeric column all togather\
    selected_column.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

10. Check stats data



```
[ ] #now we take origional dataset andcheck for more stats data
    train_data.head()
```

	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