

Import Library

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

Numerical column imputation(Titanic datasets)

```
df = pd.read_csv('/content/titanic_datasets.csv',usecols=['Age' , 'Fare' , 'Survived'])
df.head(2)
```

	Survived	Age	Fare
0	0	34.5	7.8292
1	1	47.0	7.0000

Next steps:

[Generate code with df](#)

[New interactive sheet](#)

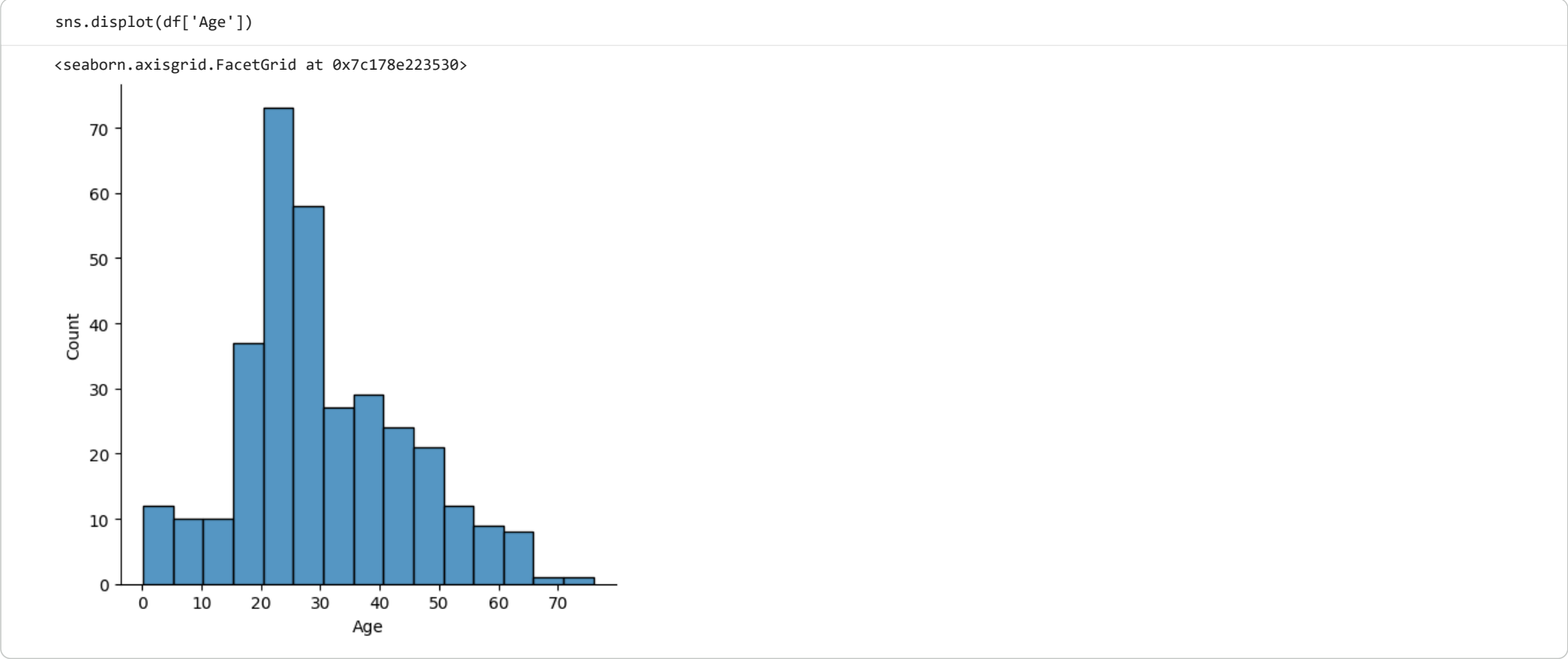
```
df.isnull().mean()*100
```

	0
Survived	0.000000
Age	20.574163
Fare	0.239234

dtype: float64

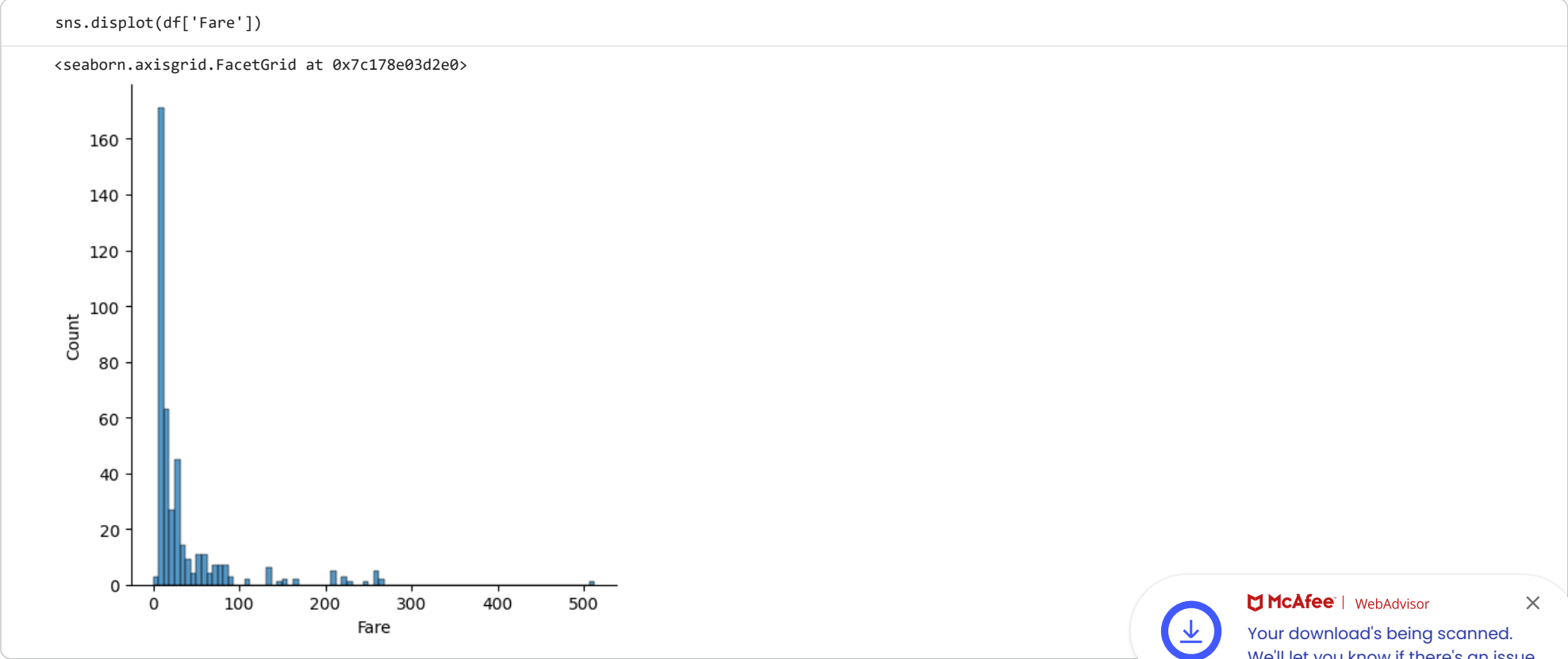
```
df['Age'].skew()
```

```
np.float64(0.4573612871503845)
```



```
df['Fare'].skew()
```

```
np.float64(3.6872133081121405)
```



```
x = df.drop('Survived',axis=1)
y = df['Survived']
```

```
from sklearn.model_selection import train_test_split
from sklearn.compose import ColumnTransformer
```

```
x_train , x_test , y_train , y_test = train_test_split(x,y,test_size=0.2,random_state=2)
```

x_train

	Age	Fare	
280	23.0	8.6625	
284	2.0	20.2125	
40	39.0	13.4167	
17	21.0	7.2250	
362	31.0	21.0000	
...	
299	29.0	7.8542	
22	NaN	31.6833	
72	29.0	7.9250	
15	24.0	27.7208	
168	NaN	27.7208	

334 rows × 2 columns

Next steps: [Generate code with x_train](#) [New interactive sheet](#)

```
from sklearn.impute import SimpleImputer
```

```
mean_imputer = SimpleImputer(strategy='mean')
median_imputer = SimpleImputer(strategy='median')
```

```
trf1 = ColumnTransformer([
    ('mean_imputer',mean_imputer,['Fare']),
    ('median_imputer' , median_imputer , ['Age'])
],remainder='passthrough')
```

```
x_train = trf1.fit_transform(x_train)
x_test  = trf1.transform(x_test)
```

x_train

[Show hidden output](#)

```
trf1.named_transformers_['mean_imputer'].statistics_
```

```
array([33.66278799])
```

```
x_train = pd.DataFrame(
    x_train,
    columns=['Fare', 'Age']
)
x_test = pd.DataFrame(
    x_test,
    columns = ['Fare' , "Age"]
)
```


x_train

	Fare	Age	
0	8.6625	23.0	
1	20.2125	2.0	
2	13.4167	39.0	
3	7.2250	21.0	
4	21.0000	31.0	
...	
329	7.8542	29.0	
330	31.6833	27.0	
331	7.9250	29.0	
332	27.7208	24.0	
333	27.7208	27.0	

334 rows × 2 columns

Next steps: [Generate code with x_train](#) [New interactive sheet](#)

x_test



McAfee | WebAdvisor

×

Your download's being scanned.
We'll let you know if there's an issue.

	Fare	Age	
0	51.8625	27.0	
1	7.7958	21.0	
2	12.3500	61.0	
3	26.5500	64.0	
4	69.5500	27.0	
...	
79	13.5000	24.0	
80	26.0000	50.0	
81	31.3875	40.0	
82	28.5375	31.0	
83	10.5000	25.0	
84 rows × 2 columns			

Next steps: [Generate code with x_test](#) [New interactive sheet](#)

```
x_train.isnull().sum()
```

Start coding or [generate](#) with AI.

Categorical column imputation

```
import requests
from io import StringIO

url = "https://raw.githubusercontent.com/campusx-official/100-days-of-machine-learning/refs/heads/main/day37-handling-missing-categorical-data/train.csv"
headers = {"User-Agent": "Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:66.0) Gecko/20100101 Firefox/66.0"}
req = requests.get(url, headers=headers)
data = StringIO(req.text)
df = pd.read_csv(data)
```

```
df = df[['FireplaceQu', 'GarageQual', 'SalePrice']]
df.head()
```

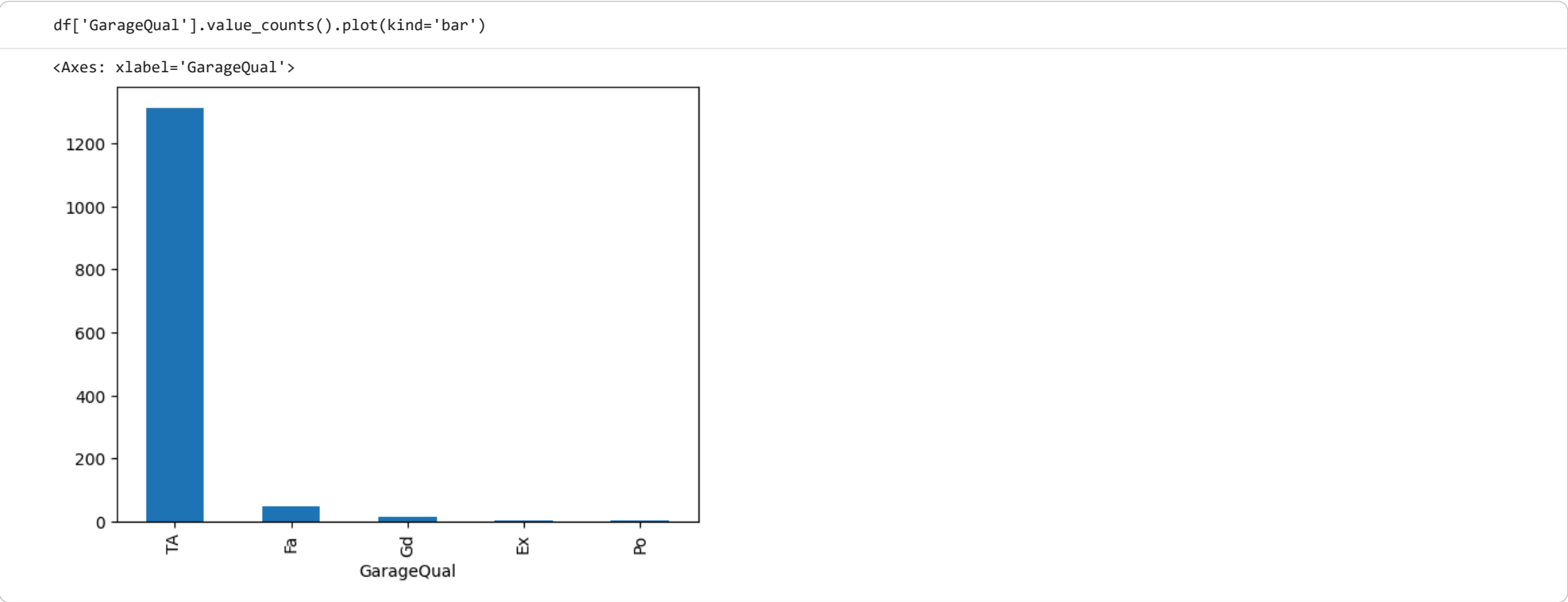
	FireplaceQu	GarageQual	SalePrice	
0	NaN	TA	208500	
1	TA	TA	181500	
2	TA	TA	223500	
3	Gd	TA	140000	
4	TA	TA	250000	

Next steps: [Generate code with df](#) [New interactive sheet](#)

```
df.isnull().mean()*100
```

	0
FireplaceQu	47.260274
GarageQual	5.547945
SalePrice	0.000000

dtype: float64



```
df['GarageQual'].mode()
```

McAfee | WebAdvisor
Your download's being scanned.
We'll let you know if there's an issue.

×

```

    GarageQual
0          TA

dtype: object

x_train , x_test , y_train , y_test = train_test_split(df.drop(columns=['SalePrice']),df['SalePrice'])


from sklearn.impute import SimpleImputer

imputer = SimpleImputer(strategy='most_frequent')

x_train = imputer.fit_transform(x_train)
x_test = imputer.transform(x_test)

imputer.statistics_

array(['Gd', 'TA'], dtype=object)
```



 | WebAdvisor

×

Your download's being scanned.
We'll let you know if there's an issue.