📦 Import Required Libraries

```
import numpy as np
import pandas as pd

from sklearn.model_selection import train_test_split

import matplotlib.pyplot as plt
import seaborn as sns
```

```
df=pd.read_csv("/content/titanic_toy.csv",usecols=['Age','Fare','Survived'])
```

```
df.sample(5)
```

	Age	Fare	Survived
639	NaN	16.1000	0
326	61.0	6.2375	0
627	21.0	77.9583	1
887	19.0	30.0000	1
665	32.0	73.5000	0

🔀 Feature–Target Split & Train–Test Split

```
x=df.drop(columns='Survived')
y=df['Survived']
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

🧾 Create Imputed Feature Copies

```
x_train['age_imputed']=x_train['Age']
x_test['age_imputed']=x_test['Age']
x_train['fare_imputed']=x_train['Fare']
x_test['fare_imputed']=x_test['Fare']
```

🎲 Missing Value Imputation (Random Sampling)

```
x_train['age_imputed'][x_train['age_imputed'].isnull()]=x_train['Age'].dropna().sample(x_trai
x_test['age_imputed'][x_test['age_imputed'].isnull()]=x_test['Age'].dropna().sample(x_test['A
x_train['fare_imputed'][x_train['fare_imputed'].isnull()]=x_train['Fare'].dropna().sample(x_t
x_test['fare_imputed'][x_test['fare_imputed'].isnull()]=x_test['Fare'].dropna().sample(x_test
```

```
/tmp/ipython-input-3530151679.py:1: FutureWarning: ChainedAssignmentError: behaviour will cha
You are setting values through chained assignment. Currently this works in certain cases, but
A typical example is when you are setting values in a column of a DataFrame, like:
```

```
df["col"][row_indexer] = value

Use `df.loc[row_indexer, "col"] = values` instead, to perform the assignment in a single step

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide

  x_train['age_imputed'][x_train['age_imputed'].isnull()]=x_train['Age'].dropna().sample(x_tr
/tmp/ipython-input-3530151679.py:2: FutureWarning: ChainedAssignmentError: behaviour will cha
You are setting values through chained assignment. Currently this works in certain cases, but
A typical example is when you are setting values in a column of a DataFrame, like:

df["col"][row_indexer] = value

Use `df.loc[row_indexer, "col"] = values` instead, to perform the assignment in a single step

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide

  x_test['age_imputed'][x_test['age_imputed'].isnull()]=x_test['Age'].dropna().sample(x_test[
/tmp/ipython-input-3530151679.py:3: FutureWarning: ChainedAssignmentError: behaviour will cha
You are setting values through chained assignment. Currently this works in certain cases, but
A typical example is when you are setting values in a column of a DataFrame, like:

df["col"][row_indexer] = value

Use `df.loc[row_indexer, "col"] = values` instead, to perform the assignment in a single step

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide

  x_train['fare_imputed'][x_train['fare_imputed'].isnull()]=x_train['Fare'].dropna().sample(x
/tmp/ipython-input-3530151679.py:4: FutureWarning: ChainedAssignmentError: behaviour will cha
You are setting values through chained assignment. Currently this works in certain cases, but
A typical example is when you are setting values in a column of a DataFrame, like:

df["col"][row_indexer] = value

Use `df.loc[row_indexer, "col"] = values` instead, to perform the assignment in a single step

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_guide

  x_test['fare_imputed'][x_test['fare_imputed'].isnull()]=x_test['Fare'].dropna().sample(x_te
```

```
x_train
```

	Age	Fare	age_imputed	fare_imputed
140	NaN	15.2458	36.0	15.2458
439	31.0	10.5000	31.0	10.5000
817	31.0	37.0042	31.0	37.0042
378	20.0	NaN	20.0	26.5500
491	21.0	7.2500	21.0	7.2500
...	...	...	...	...
835	39.0	83.1583	39.0	83.1583
192	19.0	7.8542	19.0	7.8542
629	NaN	7.7333	28.0	7.7333
559	36.0	17.4000	36.0	17.4000
684	60.0	39.0000	60.0	39.0000

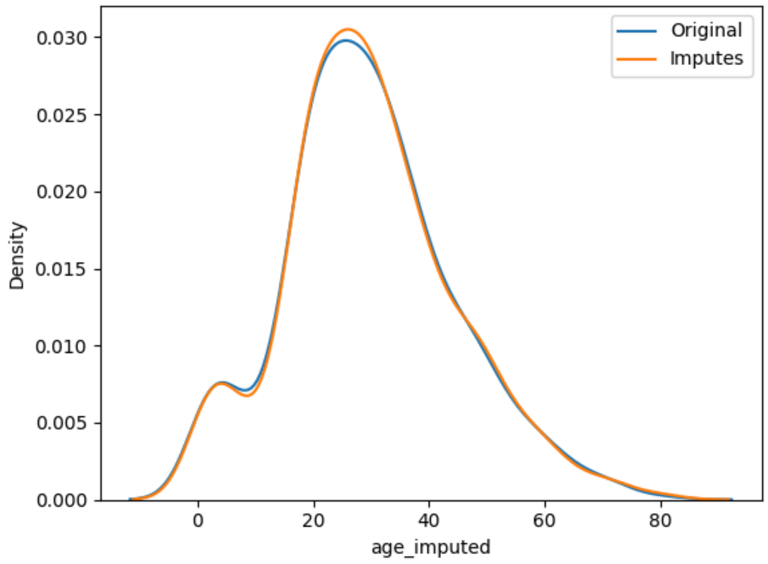
712 rows × 4 columns

Next steps: Generate code with x_train New interactive sheet

📊 Distribution Comparison: Before vs After Imputation

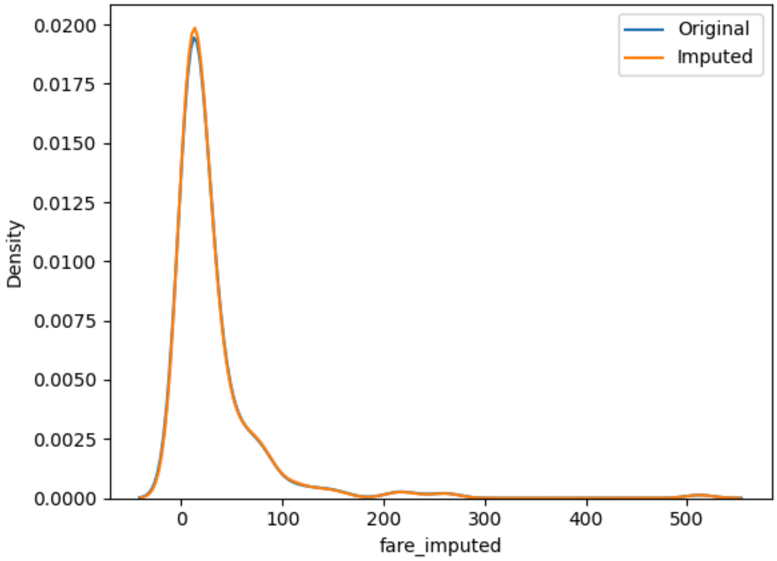
```
sns.distplot(x_train['Age'],label='Original',hist=False)
sns.distplot(x_train['age_imputed'],label='Imputes',hist=False)
plt.legend()
plt.show()
```

```
/tmp/ipython-input-2255413705.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(x_train['Age'],label='Original',hist=False)
/tmp/ipython-input-2255413705.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(x_train['age_imputed'],label='Imputes',hist=False)
```



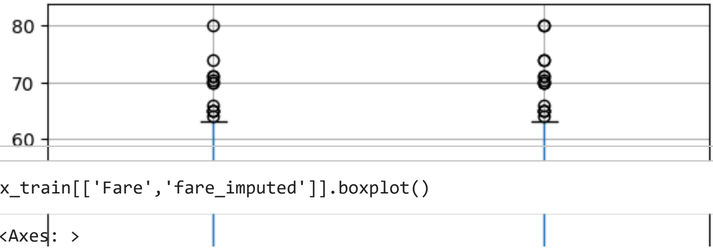
```
sns.distplot(x_train['Fare'],label="Original",hist=False)
sns.distplot(x_train['fare_imputed'],label="Imputed",hist=False)
plt.legend()
plt.show()
```

```
/tmp/ipython-input-4221924458.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(x_train['Fare'],label="Original",hist=False)
/tmp/ipython-input-4221924458.py:2: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `kdeplot` (an axes-level function for kernel density plots).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(x_train['fare_imputed'],label="Imputed",hist=False)
```



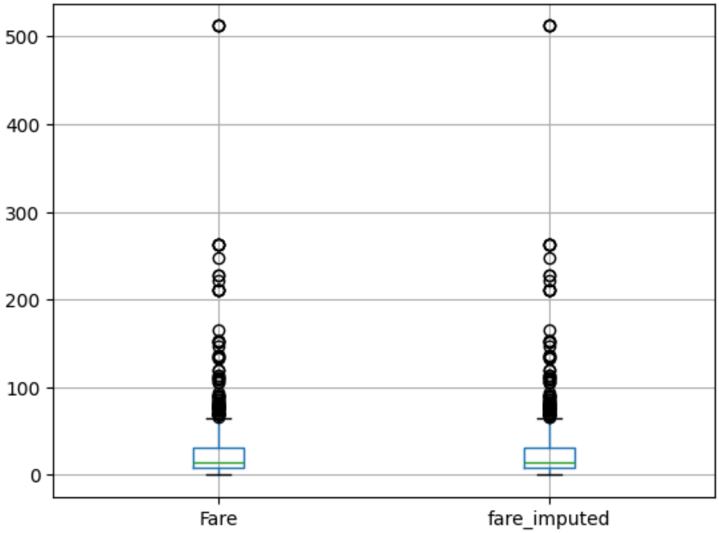
```
x_train[['Age','age_imputed']].boxplot()
```

```
<Axes: >
```



```
x_train[['Fare','fare_imputed']].boxplot()
```

```
<Axes: >
```



```
Start coding or generate with AI.
```