In [1]:

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

In [2]:

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

In [3]:

```
df = pd.read_csv(r"C:\Users\lenovo\Downloads\titanic.csv")
```

In [4]:

```
df.head()
```

Out[4]:

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

In [5]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Survived     418 non-null    int64  
 2   Pclass       418 non-null    int64  
 3   Name         418 non-null    object 
 4   Sex          418 non-null    object 
 5   Age          332 non-null    float64
 6   SibSp        418 non-null    int64  
 7   Parch        418 non-null    int64  
 8   Ticket       418 non-null    object 
 9   Fare         417 non-null    float64
 10  Cabin        91 non-null     object 
 11  Embarked     418 non-null    object 
dtypes: float64(2), int64(5), object(5)
memory usage: 39.3+ KB
```

In [6]:

```
df.isnull().mean()
```

Out[6]:

```
PassengerId    0.000000
Survived       0.000000
Pclass         0.000000
Name           0.000000
Sex            0.000000
Age            0.205742
SibSp          0.000000
Parch          0.000000
Ticket         0.000000
Fare           0.002392
Cabin          0.782297
Embarked       0.000000
dtype: float64
```

In [7]:

```
X = df.drop(columns=['PassengerId','Survived','Pclass','Name','Sex','SibSp','Parch','Emba
```

In [8]:

```
y= df['Survived']
```

In [9]:

```
X.head()
```

Out[9]:

| | Age | Fare | Cabin |
|---|---|---|---|
| **0** | 34.5 | 7.8292 | NaN |
| **1** | 47.0 | 7.0000 | NaN |
| **2** | 62.0 | 9.6875 | NaN |
| **3** | 27.0 | 8.6625 | NaN |
| **4** | 22.0 | 12.2875 | NaN |

In [10]:

```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=2)
```

In [11]:

```
X_train.shape, X_test.shape
```

Out[11]:

```
((334, 3), (84, 3))
```

In [12]:

```
X_train.isnull().mean()
```

Out[12]:

```
Age       0.215569
Fare      0.002994
Cabin     0.784431
dtype: float64
```

In [13]:

```
mean_age = X_train['Age'].mean()
median_age = X_train['Age'].median()

mean_fare = X_train['Fare'].mean()
median_fare = X_train['Fare'].median()
```

In [14]:

```
X_train['Age_median'] = X_train['Age'].fillna(median_age)
X_train['Age_mean'] = X_train['Age'].fillna(mean_age)

X_train['Fare_median']= X_train['Fare'].fillna(median_fare)
X_train['Fare_mean'] = X_train['Fare'].fillna(mean_fare)
```

In [15]:

```
X_train.sample(10)
```

Out[15]:

| | Age | Fare | Cabin | Age_median | Age_mean | Fare_median | Fare_mean |
|---|---|---|---|---|---|---|---|
| **406** | 23.0 | 10.5000 | NaN | 23.0 | 23.000000 | 10.5000 | 10.5000 |
| **233** | NaN | 7.8792 | NaN | 27.0 | 29.307252 | 7.8792 | 7.8792 |
| **181** | 37.0 | 83.1583 | E52 | 37.0 | 37.000000 | 83.1583 | 83.1583 |
| **21** | 9.0 | 3.1708 | NaN | 9.0 | 9.000000 | 3.1708 | 3.1708 |
| **58** | NaN | 16.1000 | NaN | 27.0 | 29.307252 | 16.1000 | 16.1000 |
| **13** | 63.0 | 26.0000 | NaN | 63.0 | 63.000000 | 26.0000 | 26.0000 |
| **316** | 57.0 | 146.5208 | B78 | 57.0 | 57.000000 | 146.5208 | 146.5208 |
| **318** | 27.0 | 7.8542 | NaN | 27.0 | 27.000000 | 7.8542 | 7.8542 |
| **197** | 18.0 | 7.7750 | NaN | 18.0 | 18.000000 | 7.7750 | 7.7750 |
| **291** | 30.0 | 6.9500 | NaN | 30.0 | 30.000000 | 6.9500 | 6.9500 |

In [16]:

```
print('Orginal Age variable variance: ', X_train['Age'].var())
print('Age Varience after median imputation: ', X_train['Age_median'].var())
print('Age varience after mean imputation: ', X_train['Age_mean'].var())

print('Original Fare variable varience: ', X_train['Fare'].var())
print('Fare varience after median imputation: ', X_train['Fare_median'].var())
print('Fare varience after mean imputation: ', X_train['Fare_mean'].var())
```

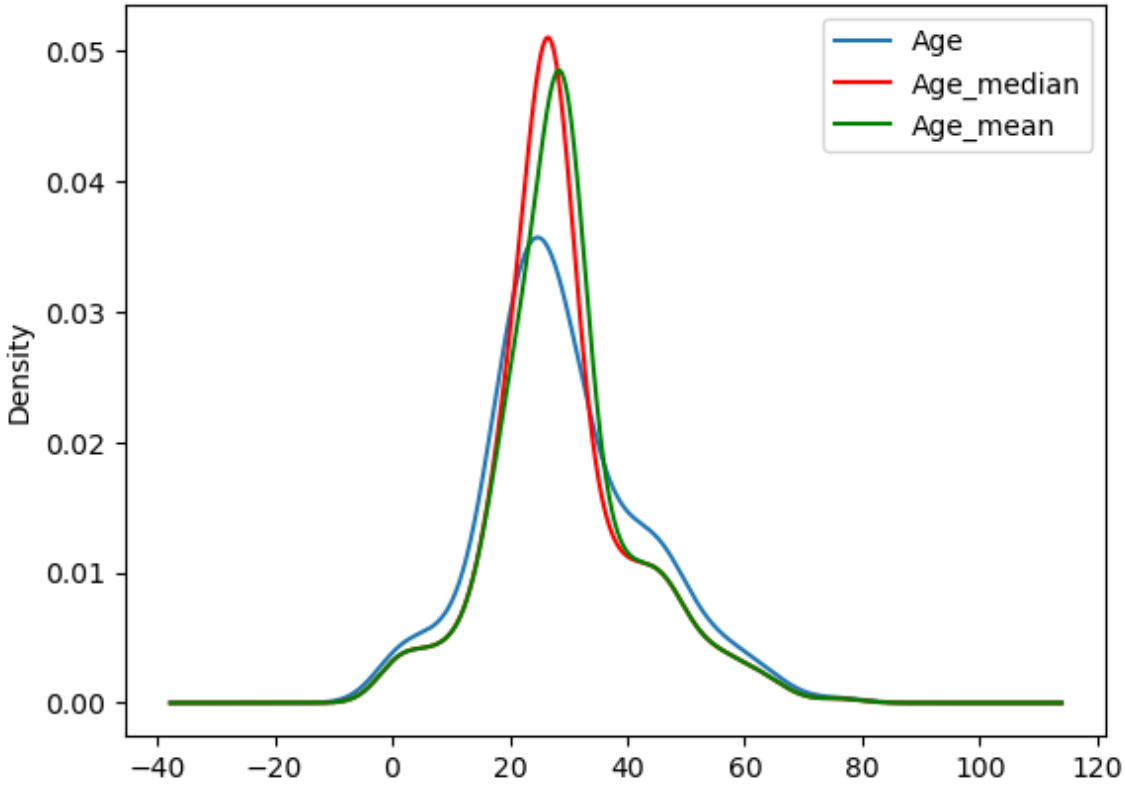
```
Orginal Age variable variance:  184.7040299669505
Age Varience after median imputation:  145.67090989552415
Age varience after mean imputation:  144.76802348760975
Original Fare variable varience:  2333.007047160699
Fare varience after median imputation:  2327.1458643048404
Fare varience after mean imputation:  2326.0010199920484
```

In [17]:

```
fig = plt.figure()
ax = fig.add_subplot(111)

# original variable distributin
X_train['Age'].plot(kind= 'kde', ax=ax)

# variable imputed with median
X_train['Age_median'].plot(kind= 'kde', ax=ax, color='red')

# variable imputed with mean
X_train['Age_mean'].plot(kind='kde', ax=ax, color='green')

# legends
lines, labels = ax.get_legend_handles_labels()
ax.legend(lines, labels, loc='best')
```

Out[17]:

```
<matplotlib.legend.Legend at 0x22766291850>
```



In [18]:

```
X_train.cov()
```

Out[18]:

| | Age | Fare | Age_median | Age_mean | Fare_median | Fare_mean |
|---|---|---|---|---|---|---|
| **Age** | 184.704030 | 216.034928 | 184.704030 | 184.704030 | 212.196215 | 214.533220 |
| **Fare** | 216.034928 | 2333.007047 | 178.883204 | 168.654128 | 2333.007047 | 2333.007047 |
| **Age_median** | 184.704030 | 178.883204 | 145.670910 | 144.768023 | 176.485104 | 178.346017 |
| **Age_mean** | 184.704030 | 168.654128 | 144.768023 | 144.768023 | 166.315953 | 168.147659 |
| **Fare_median** | 212.196215 | 2333.007047 | 176.485104 | 166.315953 | 2327.145864 | 2326.001020 |
| **Fare_mean** | 214.533220 | 2333.007047 | 178.346017 | 168.147659 | 2326.001020 | 2326.001020 |

In [19]:

```
X_train.corr()
```

Out[19]:

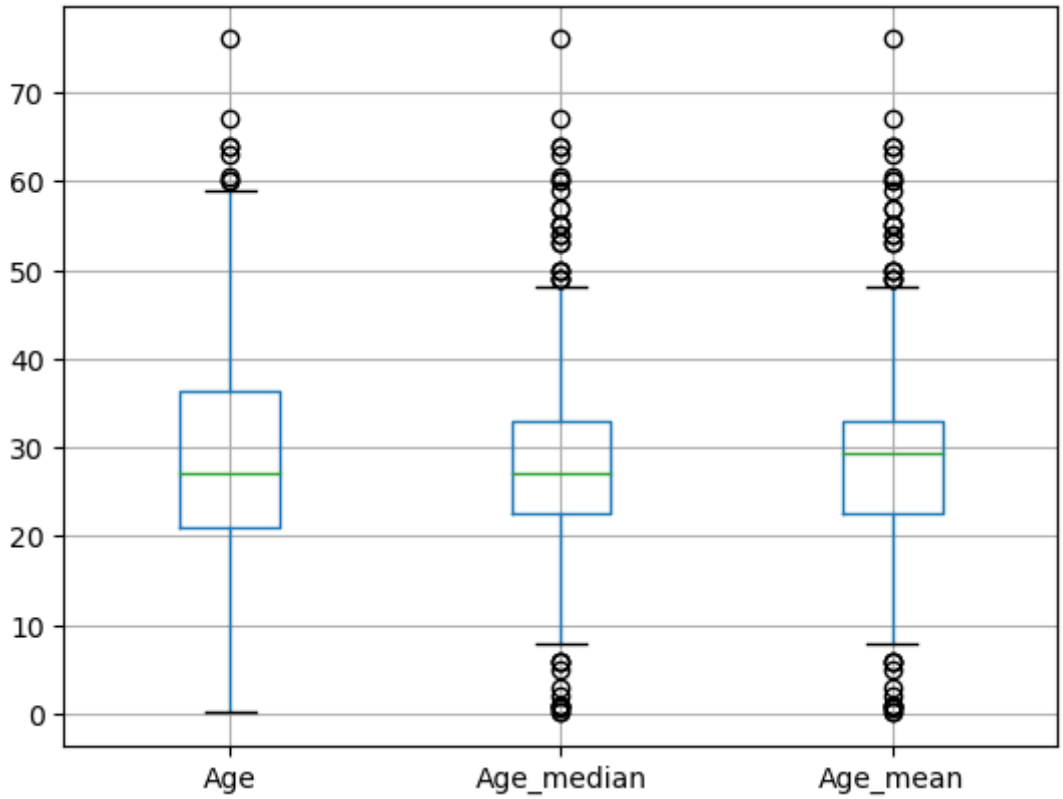
| | Age | Fare | Age_median | Age_mean | Fare_median | Fare_mean |
|---|---|---|---|---|---|---|
| **Age** | 1.000000 | 0.302479 | 1.000000 | 1.000000 | 0.295082 | 0.298455 |
| **Fare** | 0.302479 | 1.000000 | 0.309620 | 0.292746 | 1.000000 | 1.000000 |
| **Age_median** | 1.000000 | 0.309620 | 1.000000 | 0.996896 | 0.303117 | 0.306388 |
| **Age_mean** | 1.000000 | 0.292746 | 0.996896 | 1.000000 | 0.286540 | 0.289767 |
| **Fare_median** | 0.295082 | 1.000000 | 0.303117 | 0.286540 | 1.000000 | 0.999754 |
| **Fare_mean** | 0.298455 | 1.000000 | 0.306388 | 0.289767 | 0.999754 | 1.000000 |

In [20]:

```
X_train[['Age','Age_median','Age_mean']].boxplot()
```

Out[20]:

<AxesSubplot:>

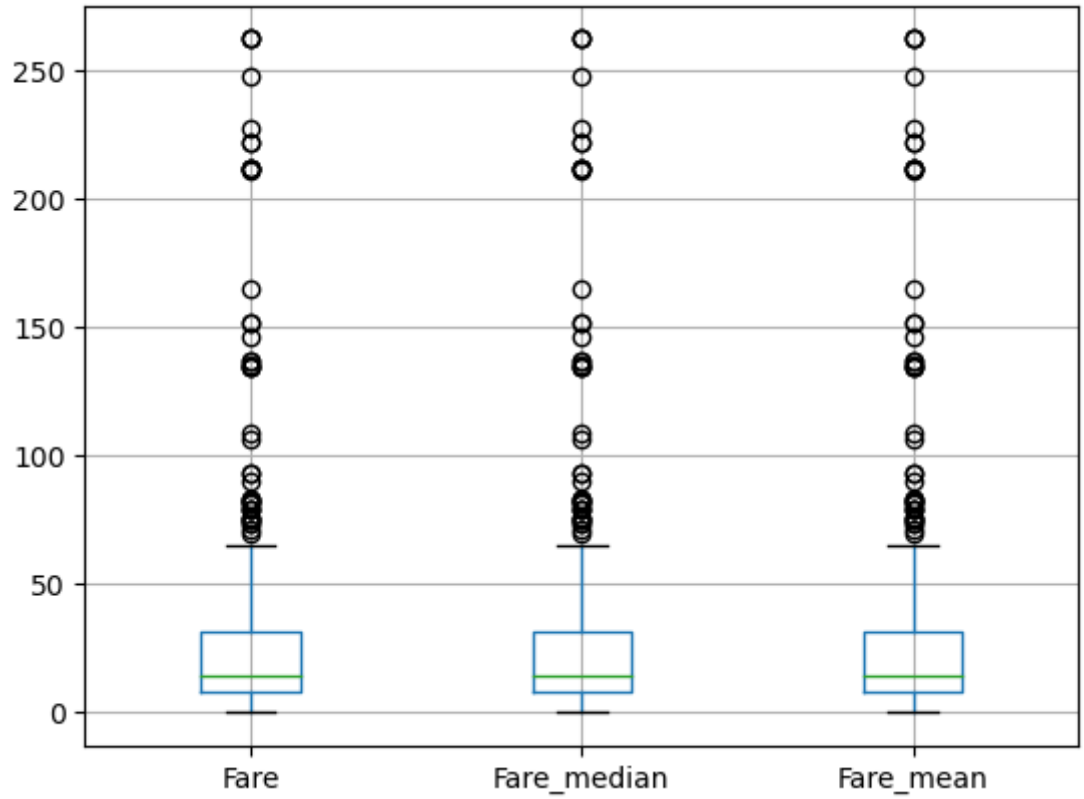


In [21]:

```
X_train[['Fare','Fare_median','Fare_mean']].boxplot()
```

Out[21]:

<AxesSubplot:>




In [ ]:

```

```

## Random Imputation on numerical data

In [22]:

```
df1 = pd.read_csv(r"C:\Users\lenovo\Downloads\titanic.csv", usecols=['Age','Fare','Surviv
```

In [23]:

```
df1.head()
```

Out[23]:

| | Survived | Age | Fare |
|---|---|---|---|
| **0** | 0 | 34.5 | 7.8292 |
| **1** | 1 | 47.0 | 7.0000 |
| **2** | 0 | 62.0 | 9.6875 |
| **3** | 0 | 27.0 | 8.6625 |
| **4** | 1 | 22.0 | 12.2875 |

In [24]:

```
df1.isnull().mean()*100
```

Out[24]:

```
Survived     0.000000
Age         20.574163
Fare         0.239234
dtype: float64
```

In [28]:

```
X = df1.drop(columns=['Survived'])
y = df1['Survived']
```

In [29]:

```
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.2, random_state=2)
```

In [30]:

```
X_train
```

Out[30]:

| | 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

In [31]:

```
X_train['Age_imputed'] = X_train['Age']
X_test['Age_imputed'] = X_test['Age']
```

In [33]:

```
X_test.tail(10)
```

Out[33]:

| | Age | Fare | Age_imputed |
|---|---|---|---|
| **173** | NaN | 7.2292 | NaN |
| **70** | 24.0 | 7.7500 | 24.0 |
| **37** | 21.0 | 8.6625 | 21.0 |
| **217** | 57.0 | 164.8667 | 57.0 |
| **117** | 1.0 | 16.7000 | 1.0 |
| **348** | 24.0 | 13.5000 | 24.0 |
| **30** | 50.0 | 26.0000 | 50.0 |
| **174** | 40.0 | 31.3875 | 40.0 |
| **68** | 31.0 | 28.5375 | 31.0 |
| **204** | 25.0 | 10.5000 | 25.0 |

In [34]:

```
X_train['Age'] [X_train['Age_imputed'].isnull()]
```

Out[34]:

```
163   NaN
41    NaN
29    NaN
65    NaN
382   NaN
       ..
116   NaN
124   NaN
47    NaN
22    NaN
168   NaN
Name: Age, Length: 72, dtype: float64
```

In [35]:

```
X_train['Age'].isnull().sum()
```

Out[35]:

```
72
```

In [37]:

```
X_train['Age'].dropna().sample(X_train['Age'].isnull().sum()).values
```

Out[37]:

```
array([45.  , 45.  ,  6.  , 27.  , 30.  , 18.  , 24.  , 21.  , 18.  ,
       36.  , 25.  , 21.  , 46.  , 26.  , 41.  , 22.  , 39.  , 24.  ,
       23.  , 30.  , 27.  , 35.  , 24.  ,  0.75, 28.  , 18.  , 18.  ,
       16.  , 37.  , 25.  , 24.  ,  1.  , 24.  , 55.  , 63.  , 32.  ,
       43.  ,  0.33, 43.  , 64.  ,  3.  , 47.  , 17.  , 13.  , 46.  ,
       45.  , 27.  ,  8.  , 27.  , 30.  , 33.  , 36.  , 21.  , 17.  ,
       17.  , 45.  , 33.  , 59.  , 20.  , 45.  , 20.  , 30.  , 60.  ,
       29.  , 38.  , 45.  , 34.5 , 31.  , 24.  , 31.  , 21.  , 48.  ])
```

In [38]:

```
X_train['Age_imputed'][X_train['Age_imputed'].isnull()] = X_train['Age'].dropna().sample(
X_test['Age_imputed'][X_test['Age_imputed'].isnull()]= X_train['Age'].dropna().sample(X_t
```

In [39]:

```
X_train
```

Out[39]:

| | Age | Fare | Age_imputed |
|---|---|---|---|
| **280** | 23.0 | 8.6625 | 23.0 |
| **284** | 2.0 | 20.2125 | 2.0 |
| **40** | 39.0 | 13.4167 | 39.0 |
| **17** | 21.0 | 7.2250 | 21.0 |
| **362** | 31.0 | 21.0000 | 31.0 |
| **...** | ... | ... | ... |
| **299** | 29.0 | 7.8542 | 29.0 |
| **22** | NaN | 31.6833 | 21.0 |
| **72** | 29.0 | 7.9250 | 29.0 |
| **15** | 24.0 | 27.7208 | 24.0 |
| **168** | NaN | 27.7208 | 25.0 |

334 rows × 3 columns

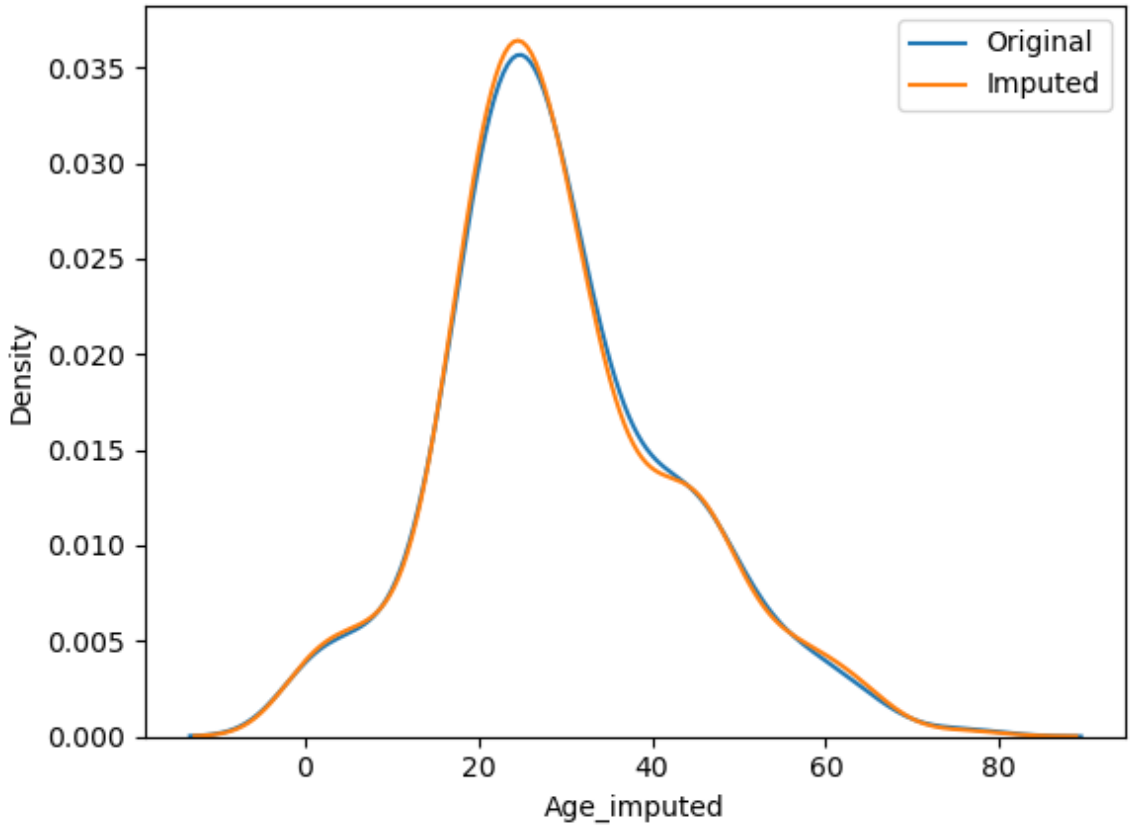
In [41]:

```
import seaborn as sns
```

In [42]:

```
sns.distplot(X_train['Age'], label='Original', hist=False)
sns.distplot(X_train['Age_imputed'], label= 'Imputed', hist= False)

plt.legend()
plt.show()
```

```
C:\Users\lenovo\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `kdeplot` (an axes-level func
tion for kernel density plots).
  warnings.warn(msg, FutureWarning)
C:\Users\lenovo\anaconda3\lib\site-packages\seaborn\distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in
a future version. Please adapt your code to use either `displot` (a figure
-level function with similar flexibility) or `kdeplot` (an axes-level func
tion for kernel density plots).
  warnings.warn(msg, FutureWarning)
```




In [43]:

```
print('Original variable varience: ', X_train['Age'].var())
print('variable varience after random imputation: ', X_train['Age_imputed'].var())
```

```
Original variable varience:  184.7040299669505
variable varience after random imputation:  187.21043133642635
```

In [46]:

```
X_train[['Age','Age_imputed','Fare']].cov()
```

Out[46]:

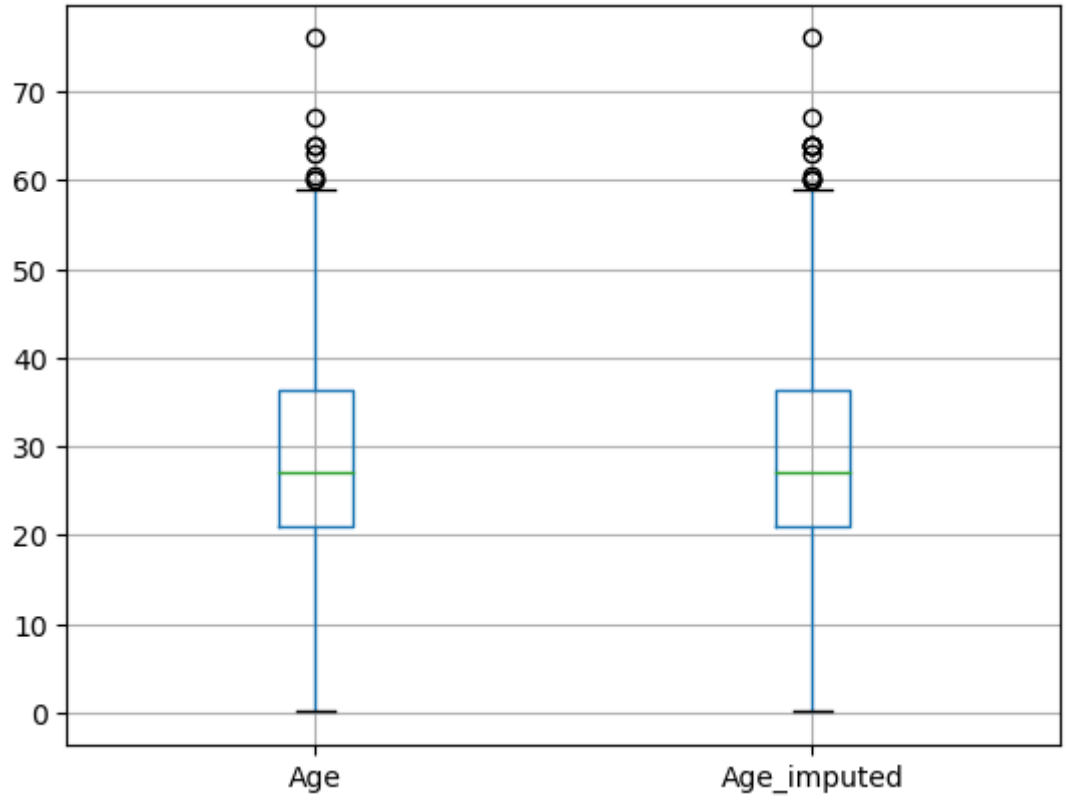
| | Age | Age_imputed | Fare |
|---|---|---|---|
| **Age** | 184.704030 | 184.704030 | 216.034928 |
| **Age_imputed** | 184.704030 | 187.210431 | 171.823575 |
| **Fare** | 216.034928 | 171.823575 | 2333.007047 |

In [47]:

```
X_train[['Age','Age_imputed']].boxplot()
```

Out[47]:

<AxesSubplot:>




In [ ]:

```

```