Start coding or generate with AI.

How to handle missing value

Imputation Technique for Data Cleaning

1 Mean Imputation Technique

2 Median Imputation Technique

3 Mode Imputation Technique

```
import pandas as pd
import seaborn as sns
```

```
# import titanic dataset from seaborn
df = sns.load_dataset('titanic')
```

```
df
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	NaN	Southampton	no	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	B	Southampton	yes	True
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	NaN	Southampton	no	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	C	Cherbourg	yes	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	NaN	Queenstown	no	True

891 rows × 15 columns

Next steps: Generate code with df View recommended plots

```
df.head()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True

Next steps: Generate code with df View recommended plots

```
df.tail()
```

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_
886	0	2	male	27.0	0	0	13.00	S	Second	man	
887	1	1	female	19.0	0	0	30.00	S	First	woman	
888	0	3	female	NaN	1	2	23.45	S	Third	woman	
889	1	1	male	26.0	0	0	30.00	C	First	man	

```
df.shape
```

```
(891, 15)
```

```
print("Number of rows are",df.shape[0])
print("Number of colums are",df.shape[1])
```

```
Number of rows are 891
Number of colums are 15
```

```
df.isnull().sum()
```

```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

```
# Handlin missing values by deleting row and colums but this is not good appreach we lose too mane data
df.dropna().shape
```

```
(182, 15)
```

```
df.shape
```

```
(891, 15)
```

```
# Imputation Technique for Data Cleaning
# 1 Mean Imputation Technique
# 2 Median Imputation Technique
# 3 Mode Imputation Technique
```

```
# 1 Mean Imputation Technique
# This techniqu work well when data is normally distributed and this technique work with numerical data
```

```
df['Age_mean'] = df['age'].fillna(df['age'].mean())
# This techniqu work well when data is normally distributed
```

```
df[['Age_mean','age']]
```

	Age_mean	age
0	22.000000	22.0
1	38.000000	38.0
2	26.000000	26.0
3	35.000000	35.0
4	35.000000	35.0
...	...	...
886	27.000000	27.0
887	19.000000	19.0
888	29.699118	NaN
889	26.000000	26.0
890	32.000000	32.0

891 rows × 2 columns

```
# 2 Median Imputation Technique
# This techniqu also work well when data is not normally distributed and many outlier and this technique work with numerical data
```

```
df['Age_median'] = df['age'].fillna(df['age'].median())
```

```
df[['Age_mean','age','Age_median']]
```

	Age_mean	age	Age_median
0	22.000000	22.0	22.0
1	38.000000	38.0	38.0
2	26.000000	26.0	26.0
3	35.000000	35.0	35.0
4	35.000000	35.0	35.0
...	...	...	...
886	27.000000	27.0	27.0
887	19.000000	19.0	19.0
888	29.699118	NaN	28.0
889	26.000000	26.0	26.0
890	32.000000	32.0	32.0

891 rows × 3 columns

```
# 1 Mode Imputation Technique
# This techniqu work well when data is categorical and this technique work with non numerical data
```

```
df.isnull().sum()
```

```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
Age_mmean        0
Age_mean         0
Age_median       0
```

```
deck_mode      687
dtype: int64
```

```
df['deck'].isnull().sum()
```

```
688
```

```
df[df['age'].notna()]['embarked'].mode()[0]
```

```
'S'
```

```
mode = df[df['age'].notna()]['embarked'].mode()[0]
```

```
df['embarked_mode'] = df['embarked'].fillna(mode)
```

```
df[['Age_mean','age','Age_median','embarked_mode']]
```

	Age_mean	age	Age_median	embarked_mode
0	22.000000	22.0	22.0	S
1	38.000000	38.0	38.0	C
2	26.000000	26.0	26.0	S
3	35.000000	35.0	35.0	S
4	35.000000	35.0	35.0	S
...	...	...	...	...
886	27.000000	27.0	27.0	S
887	19.000000	19.0	19.0	S
888	29.699118	NaN	28.0	S
889	26.000000	26.0	26.0	C
890	32.000000	32.0	32.0	Q

891 rows × 4 columns

Start coding or generate with AI.