titanic

August 28, 2025

[30]:

```python
from sklearn.datasets import fetch_openml

# Load Titanic dataset from OpenML
titanic = fetch_openml(name="titanic", version=1, as_frame=True, parser="auto")
df_original = titanic.frame
print(df_original.head())
print(df_original.info())
print(f"Rows: {len(df_original)}")
```

```
   pclass survived                                             name     sex  \
0       1        1                    Allen, Miss. Elisabeth Walton  female
1       1        1                   Allison, Master. Hudson Trevor    male
2       1        0                     Allison, Miss. Helen Loraine  female
3       1        0             Allison, Mr. Hudson Joshua Creighton    male
4       1        0  Allison, Mrs. Hudson J C (Bessie Waldo Daniels)  female

       age  sibsp  parch  ticket      fare    cabin embarked boat   body  \
0  29.0000      0      0   24160  211.3375       B5        S    2    NaN
1   0.9167      1      2  113781  151.5500  C22 C26        S   11    NaN
2   2.0000      1      2  113781  151.5500  C22 C26        S  NaN    NaN
3  30.0000      1      2  113781  151.5500  C22 C26        S  NaN  135.0
4  25.0000      1      2  113781  151.5500  C22 C26        S  NaN    NaN

                         home.dest
0                     St Louis, MO
1  Montreal, PQ / Chesterville, ON
2  Montreal, PQ / Chesterville, ON
3  Montreal, PQ / Chesterville, ON
4  Montreal, PQ / Chesterville, ON
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1309 entries, 0 to 1308
Data columns (total 14 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   pclass     1309 non-null   int64
 1   survived   1309 non-null   category
 2   name       1309 non-null   object
 3   sex        1309 non-null   category
 4   age        1046 non-null   float64
 5   sibsp      1309 non-null   int64
 6   parch      1309 non-null   int64
 7   ticket     1309 non-null   object
 8   fare       1308 non-null   float64
 9   cabin      295 non-null    object
 10  embarked   1307 non-null   category
 11  boat       486 non-null    object
 12  body       121 non-null    float64
 13  home.dest  745 non-null    object
dtypes: category(3), float64(3), int64(3), object(5)
memory usage: 116.8+ KB
None
Rows: 1309
```

[31]:

```python
#1. Remove Duplicates
# Drop duplicate rows if any
df = df_original.drop_duplicates()
print("Any duplicates?", df.duplicated().any())
print("Number of duplicates:", df.duplicated().sum())
print(f"Rows: {len(df)}")
```

```
Any duplicates? False
Number of duplicates: 0
Rows: 1309
```

[32]:

```python
#2. Handle Missing Values

# Check missing values
print(df.isnull().sum())
```

```
pclass          0
survived        0
name            0
sex             0
age           263
sibsp           0
parch           0
ticket          0
fare            1
cabin        1014
embarked        2
boat          823
body         1188
home.dest     564
dtype: int64
```

[33]:
```
# fill with mean/median:
df["age"].fillna(df["age"].median(), inplace=True)
df["fare"].fillna(df["fare"].median(), inplace=True)
```

[34]:
```
# Only 2 missing → fill with the mode (most common port).
df["embarked"].fillna(df["embarked"].mode()[0], inplace=True)
```

[35]:
```
# Drop cabin, boat, body, home.dest:
# Since they are >70% missing, usually dropped.
df.drop(columns=["cabin", "boat", "body", "home.dest"], inplace=True)
```

[36]:
```
df
```

[36]:
```
      pclass survived                                            name  \
0          1        1                    Allen, Miss. Elisabeth Walton
1          1        1                   Allison, Master. Hudson Trevor
2          1        0                     Allison, Miss. Helen Loraine
3          1        0             Allison, Mr. Hudson Joshua Creighton
4          1        0  Allison, Mrs. Hudson J C (Bessie Waldo Daniels)
…        …      …                                               …
1304       3        0                             Zabour, Miss. Hileni
1305       3        0                            Zabour, Miss. Thamine
1306       3        0                        Zakarian, Mr. Mapriededer
1307       3        0                              Zakarian, Mr. Ortin
1308       3        0                               Zimmerman, Mr. Leo

         sex      age  sibsp  parch  ticket      fare embarked
0     female  29.0000      0      0   24160  211.3375        S
1       male   0.9167      1      2  113781  151.5500        S
2     female   2.0000      1      2  113781  151.5500        S
3       male  30.0000      1      2  113781  151.5500        S
4     female  25.0000      1      2  113781  151.5500        S
…        …      …    …     …      …       …        …
1304  female  14.5000      1      0    2665   14.4542        C
1305  female  28.0000      1      0    2665   14.4542        C
1306    male  26.5000      0      0    2656    7.2250        C
1307    male  27.0000      0      0    2670    7.2250        C
1308    male  29.0000      0      0  315082    7.8750        S

[1309 rows x 10 columns]
```

[9]:
```
# Check missing values
print(df.isnull().sum())
```

```
pclass      0
survived    0
name        0
sex         0
```

```
age         0
sibsp       0
parch       0
ticket      0
fare        0
embarked    0
dtype: int64
```

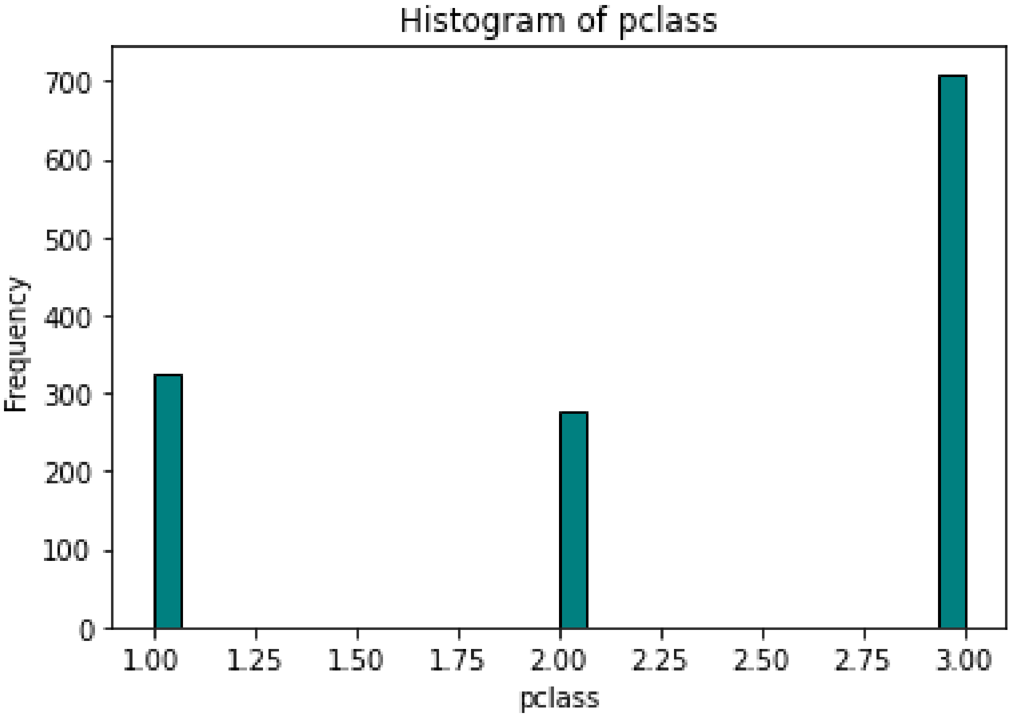
[37]:
```
#3. Check inconsistent formats

print(df.isin(["N/A", "na", "NA", "null", "NULL", "?", "--"]).sum())
```

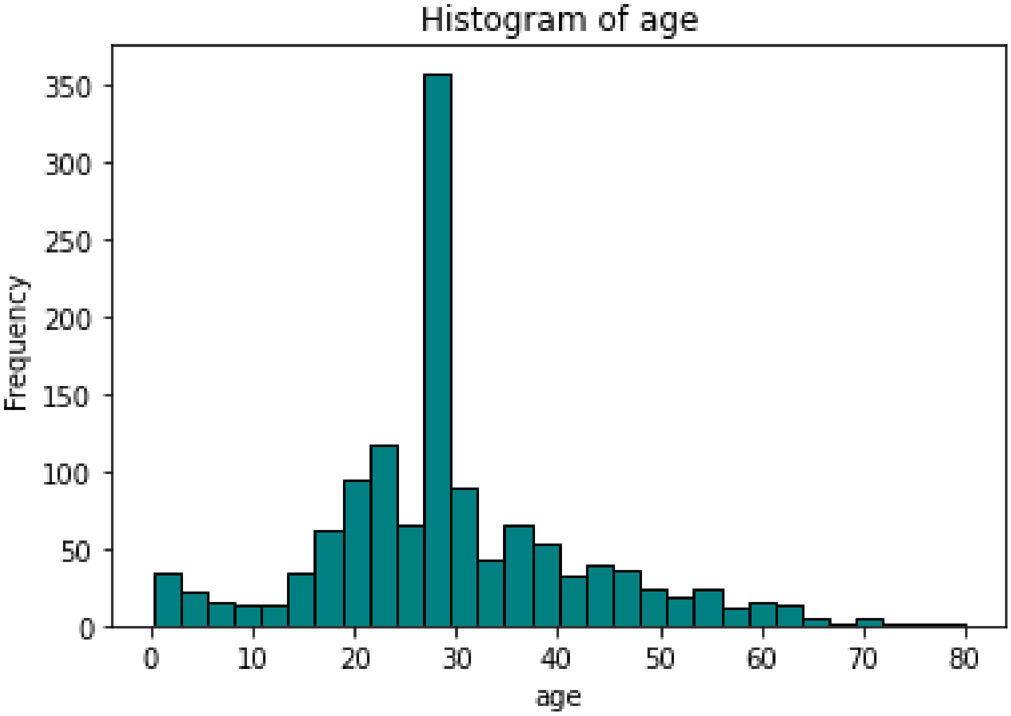
```
pclass      0
survived    0
name        0
sex         0
age         0
sibsp       0
parch       0
ticket      0
fare        0
embarked    0
dtype: int64
```

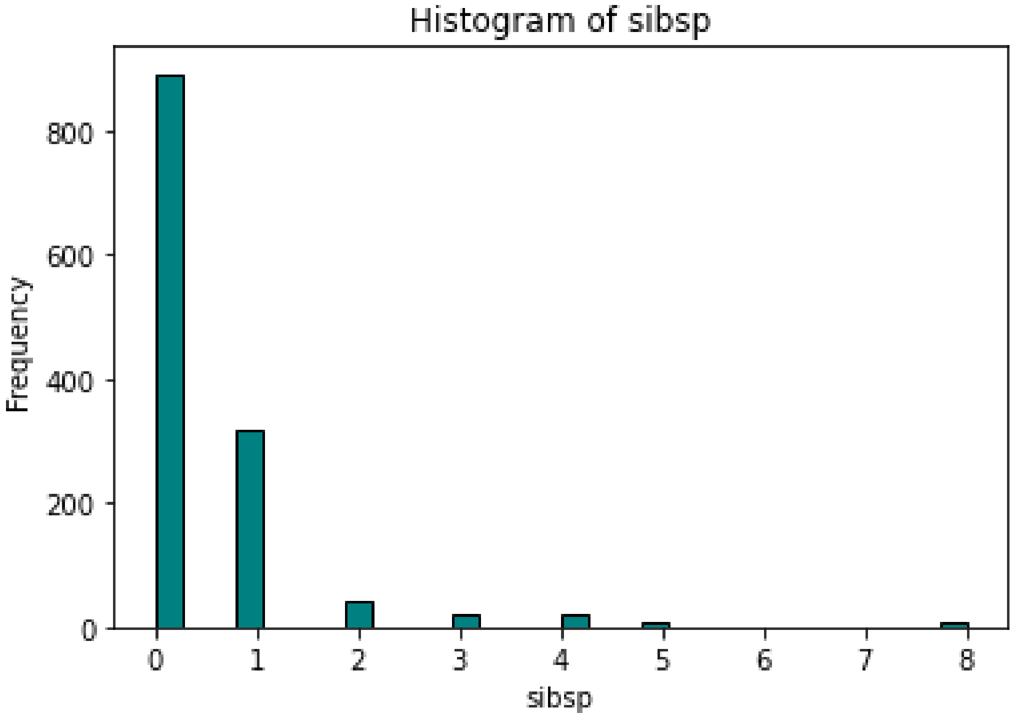
[38]:
```
# check unique values:
print(df['sex'].unique())
```

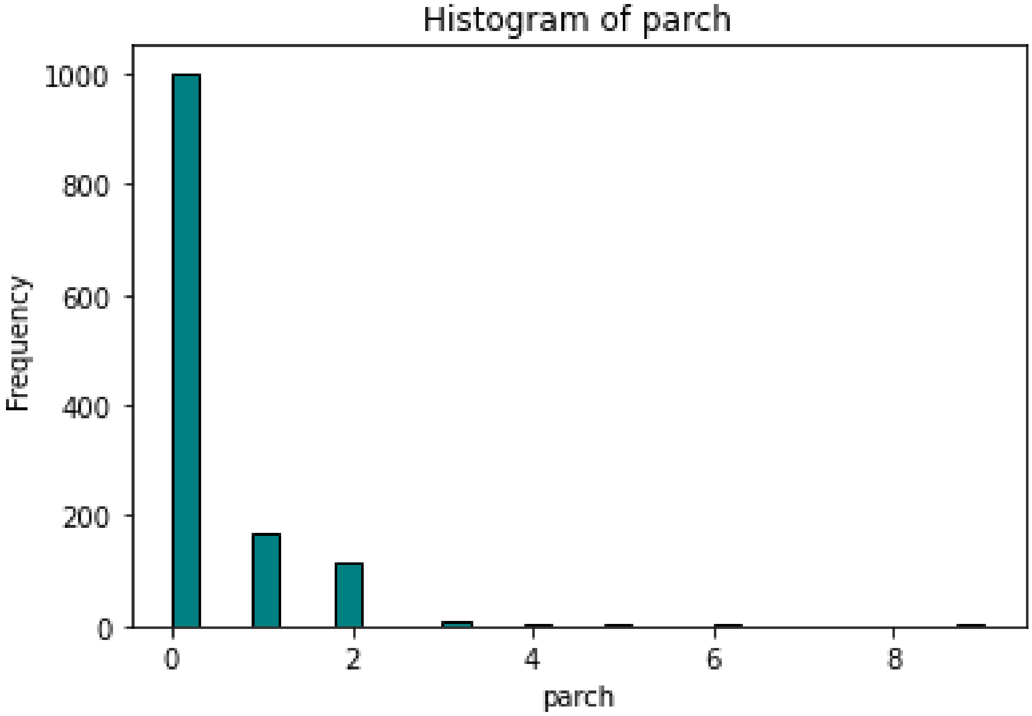
```
['female', 'male']
Categories (2, object): ['female', 'male']
```

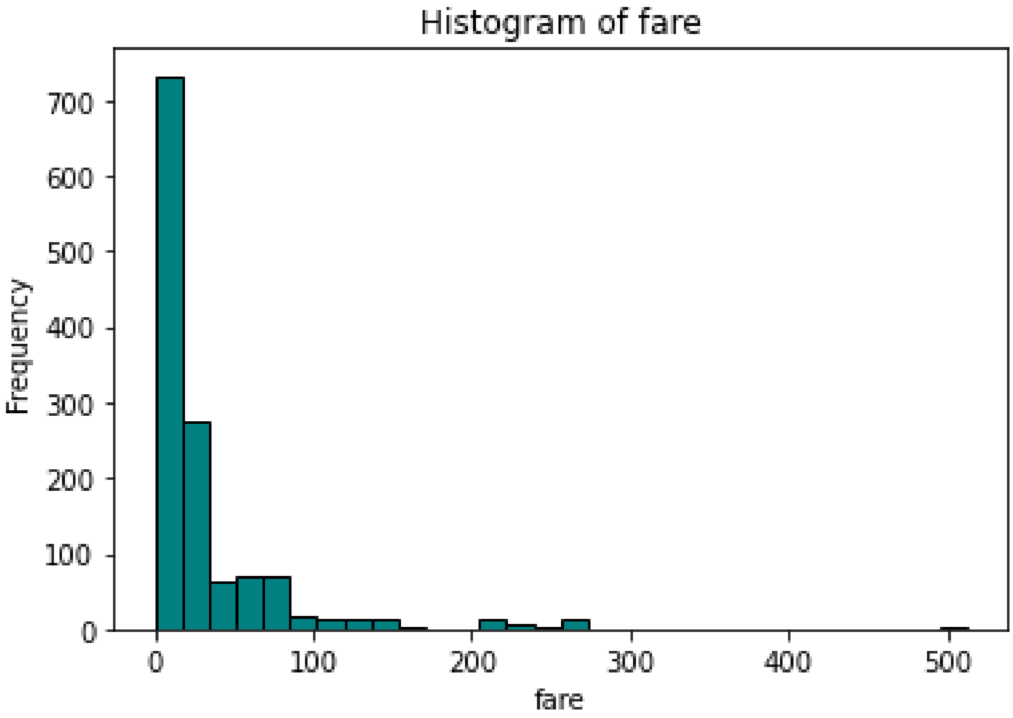
[39]:
```
#4. Check Outlier with histogram
import matplotlib.pyplot as plt

# selects only the numeric columns from the DataFrame df.
numeric_cols = df.select_dtypes(include=["int64", "float64"]).columns

for col in numeric_cols:
    plt.figure(figsize=(6,4)) # create a new figure width = 6 inches, height =␣
 ↪4 inches
    plt.hist(df[col].dropna(), bins=30, color="teal", edgecolor="black")␣
 ↪#divides the range into 30 equal-width "buckets."
    plt.title(f"Histogram of {col}")
    plt.xlabel(col)
    plt.ylabel("Frequency")
    plt.show()
```


Histogram of pclass
Frequency
0
100
200
300
400
500
600
700
1.00
1.25
1.50
1.75
2.00
2.25
2.50
2.75
3.00
pclass


Histogram of age
Frequency
0
50
100
150
200
250
300
350
0
10
20
30
40
50
60
70
80
age


Histogram of sibsp
Frequency
0
200
400
600
800
0
1
2
3
4
5
6
7
8
sibsp


Histogram of parch
Frequency
0
200
400
600
800
1000
0
2
4
6
8
parch



```
[27]: # 5. check Data Types
      print(df.dtypes)
```

```
pclass         int64
survived    category
name          object
sex         category
age          float64
sibsp          int64
parch          int64
ticket        object
fare         float64
embarked    category
dtype: object
```

```
[28]: df['survived'] = df['survived'].astype('int64')   # keep 0/1
```

```
[ ]:
```