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

```
titanic_data = pd.read_csv('tested.csv')
```

```
titanic_data.describe()
```

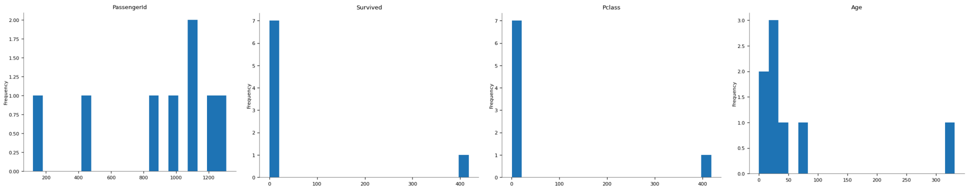
1 to 8 of 8 entries Filter

index	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	418.0	418.0	418.0	332.0	418.0	418.0	41
mean	1100.5	0.36363636363636365	2.2655502392344498	30.272590361445783	0.4473684210526316	0.3923444976076555	35.627188489208
std	120.81045760473994	0.48162214093223055	0.8418375519640519	14.18120923562442	0.8967595611217125	0.9814288785371684	55.90757617997
min	892.0	0.0	1.0	0.17	0.0	0.0	
25%	996.25	0.0	1.0	21.0	0.0	0.0	7.89
50%	1100.5	0.0	3.0	27.0	0.0	0.0	14.45
75%	1204.75	1.0	3.0	39.0	1.0	0.0	3
max	1309.0	1.0	3.0	76.0	8.0	9.0	512.32

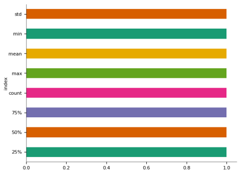
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

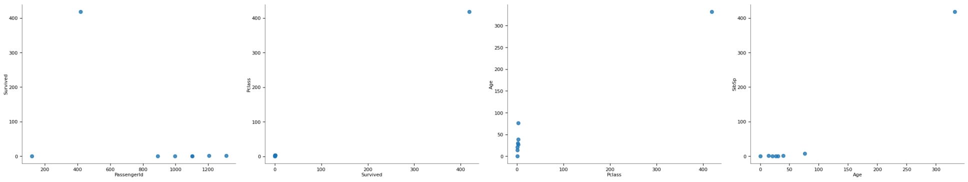
Distributions



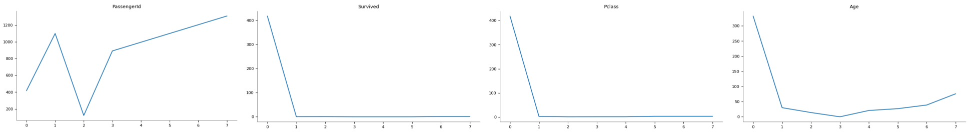
Categorical distributions



2-d distributions

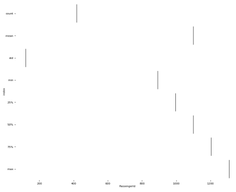


Values

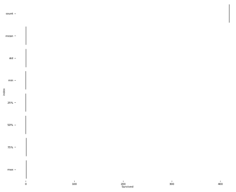


Faceted distributions

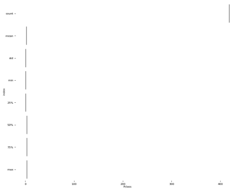
```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `le
```



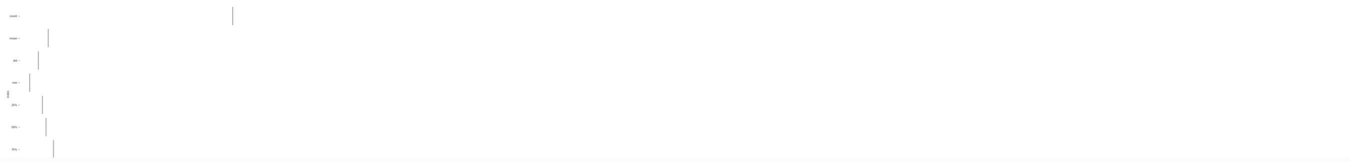
```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `le
```



```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `le
```



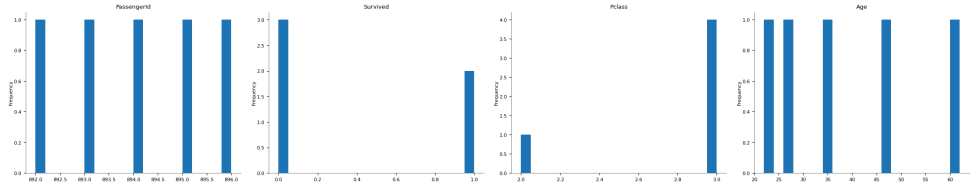
```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `le
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set `l
```



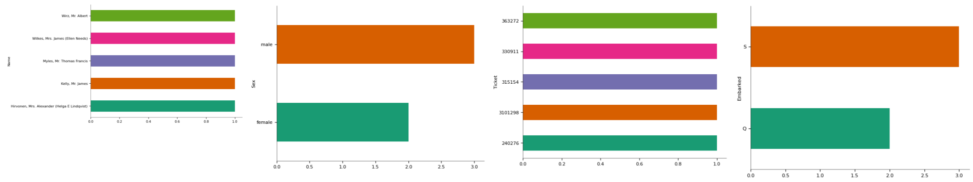
```
titanic_data.head()
```

0	892	0	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	1	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	0	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	0	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	1	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

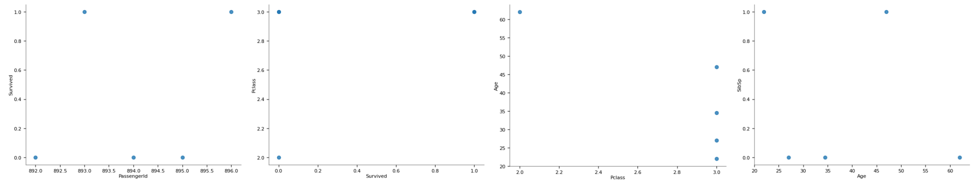
Distributions



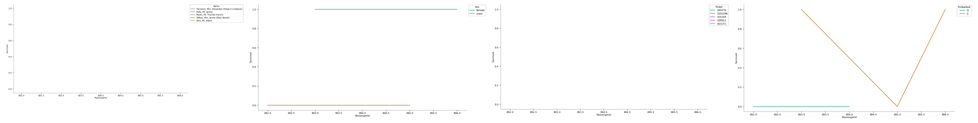
Categorical distributions



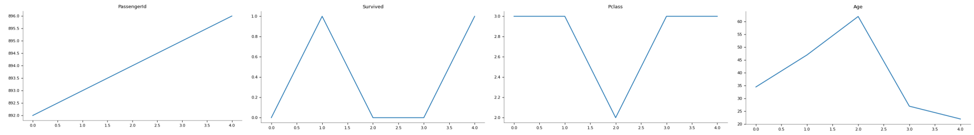
2-d distributions



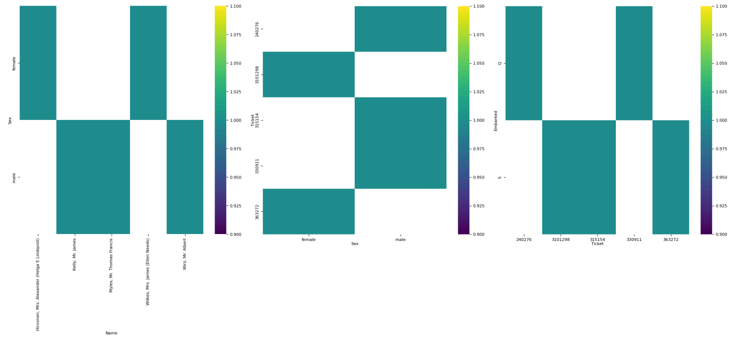
Time series



Values

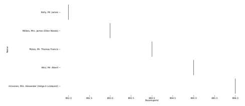


2-d categorical distributions

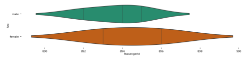


Faceted distributions

```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set
```

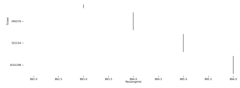


```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set
```

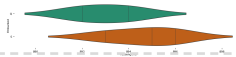


```
<string>:5: FutureWarning:

Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set
```

```
<string>:5: FutureWarning:
```



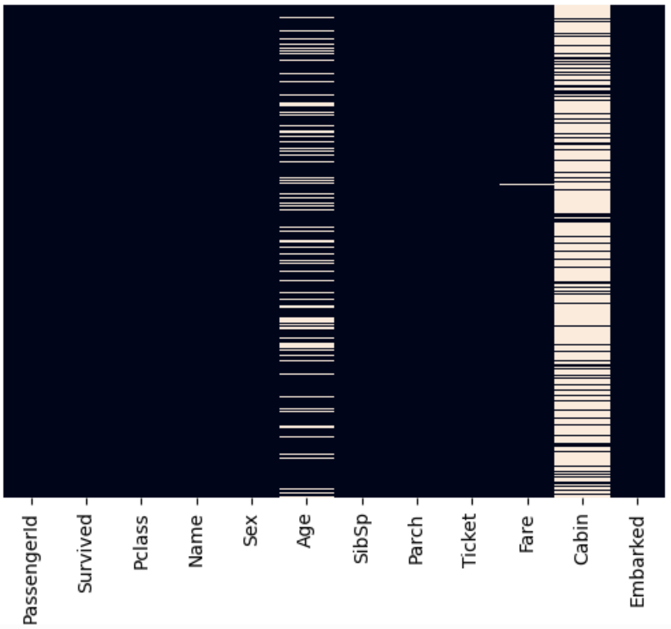
```
Passing `palette` without assigning `hue` is deprecated and will be removed in v0.14.0. Assign the `y` variable to `hue` and set
```



Next steps: Generate code with `titanic_data` | View recommended plots | New interactive sheet

Generate | create a dataframe with 2 columns and 10 rows | Close

```
sns.heatmap(titanic_data.isnull(),yticklabels=False,cbar=False)
plt.show()
```



```
titanic_data.isnull().sum().sort_values(ascending=False)
```

```
                0
        Sex    87
PassengerId     0
     Pclass     0
   Survived     0
       Name     0
        Age     0
      SibSp     0
      Parch     0
     Ticket     0
       Fare     0
 Embarked_Q     0
 Embarked_S     0

dtype: int64
```

```
(titanic_data.isnull().sum() / len(titanic_data) * 100 ).sort_values(ascending=False)
```

	0
Sex	100.0
PassengerId	0.0
Pclass	0.0
Survived	0.0
Name	0.0
Age	0.0
SibSp	0.0
Parch	0.0
Ticket	0.0
Fare	0.0
Embarked_Q	0.0
Embarked_S	0.0

dtype: float64

```
titanic_data.shape
```

```
(418, 12)
```

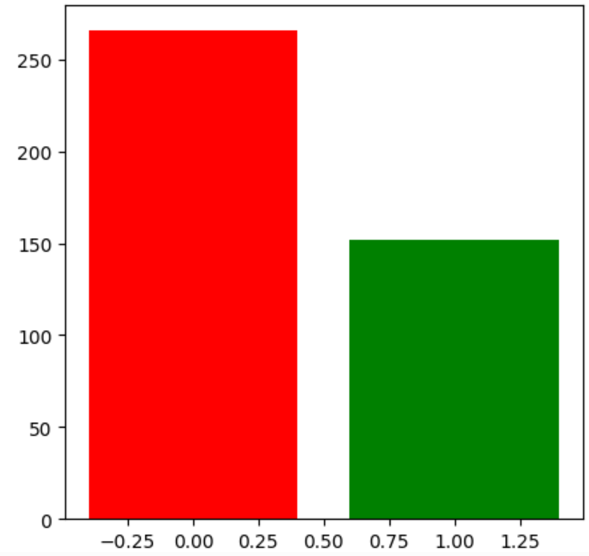
Double-click (or enter) to edit

```
titanic_data['Survived'].value_counts()
```

	count
Survived	
0	266
1	152

dtype: int64

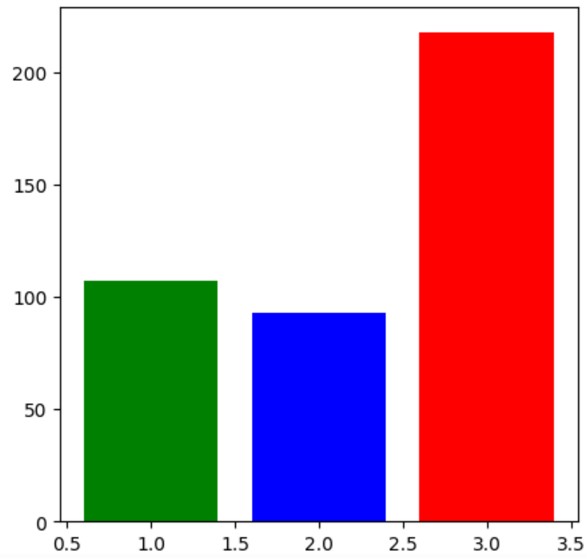
```
plt.figure(figsize=(5,5))
plt.bar(list(titanic_data['Survived'].value_counts().keys()),list(titanic_data['Survived'].value_counts()), color=["r","g"])
plt.show()
```



```
titanic_data['Pclass'].value_counts()
```

```
        count
Pclass
     3    218
     1    107
     2     93

dtype: int64
```

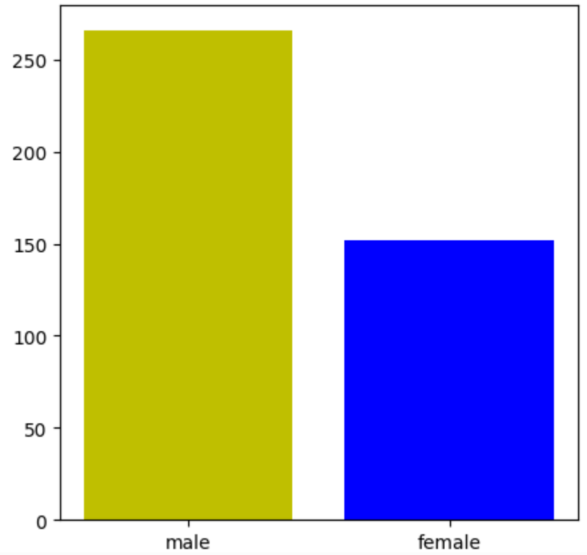
```
plt.figure(figsize=(5,5))
plt.bar(list(titanic_data['Pclass'].value_counts().keys()),list(titanic_data['Pclass'].value_counts()), color=["r","g","b"])
plt.show()
```



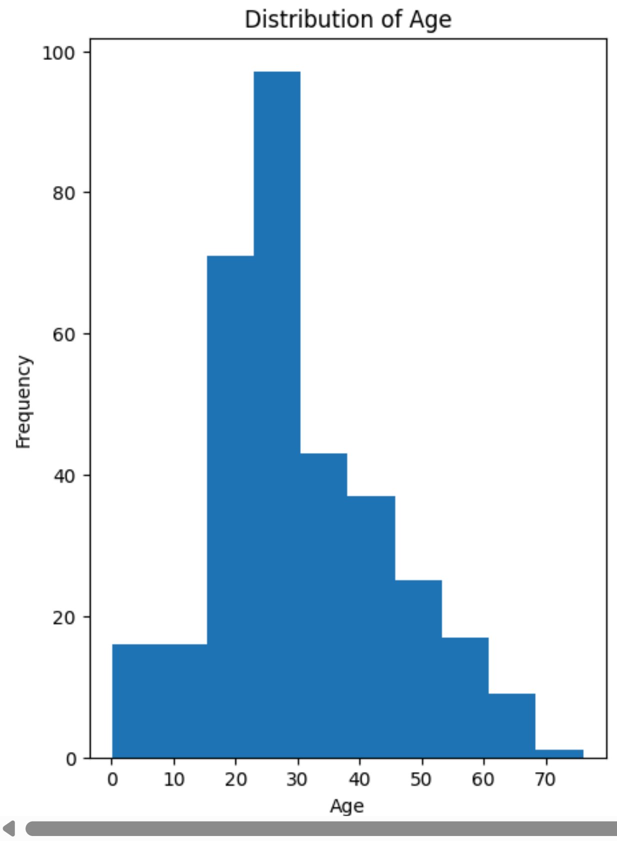
```
titanic_data['Sex'].value_counts()
```

```
        count
   Sex
  male    266
female    152

dtype: int64
```

```
plt.figure(figsize=(5,5))
plt.bar(list(titanic_data['Sex'].value_counts().keys()),list(titanic_data['Sex'].value_counts()), color=["y","b"])
plt.show()
```



```
plt.figure(figsize=(5,7))
plt.hist(titanic_data['Age'])
plt.title("Distribution of Age")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.show()
```



```
titanic_data['Survived'].isnull()
```

```
        Survived
   0       False
   1       False
   2       False
   3       False
   4       False
 ...         ...
 413       False
 414       False
 415       False
 416       False
 417       False
418 rows × 1 columns

dtype: bool
```

```
sum(titanic_data['Survived'].isnull())
```

```
0
```

```
titanic_data['Age'].isnull()
```

```
       Age
  0    False
  1    False
  2    False
  3    False
  4    False
 ...    ...
413     True
414    False
415    False
416     True
417     True
418 rows × 1 columns

dtype: bool
```

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

```
86
```

```
#building model
```

```
x_data=titanic_data[['Age']]
y_data=titanic_data[['Survived']]
```

```
from sklearn.tree import DecisionTreeClassifier
```

```
dtc = DecisionTreeClassifier()
```

```
dtc.fit(x_data,y_data)
```

```
▾ DecisionTreeClassifier  i  ?
DecisionTreeClassifier()
```

```
# Handling missing values
titanic_data['Age'].fillna(titanic_data['Age'].median(), inplace=True)
titanic_data['Embarked'].fillna(titanic_data['Embarked'].mode()[0], inplace=True)
titanic_data.drop(columns=['Cabin'], inplace=True)  # Dropping Cabin due to too many missing values
```

```
<ipython-input-42-5c0a890542ab>:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained ass
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col


  titanic_data['Age'].fillna(titanic_data['Age'].median(), inplace=True)
<ipython-input-42-5c0a890542ab>:3: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained ass
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col


  titanic_data['Embarked'].fillna(titanic_data['Embarked'].mode()[0], inplace=True)
```

Double-click (or enter) to edit

```
# Feature selection and train-test split
from sklearn.model_selection import train_test_split

features = ['Pclass', 'Age', 'Fare', 'Sex', 'Embarked_Q', 'Embarked_S']
X = titanic_data[features]
y = titanic_data['Survived']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```