```
from google.colab import files
uploaded = files.upload()
```


```
import pandas as pd
import sqlite3

# Load the CSV
df = pd.read_csv('train.csv')
df.head()
```

What can I help you build?

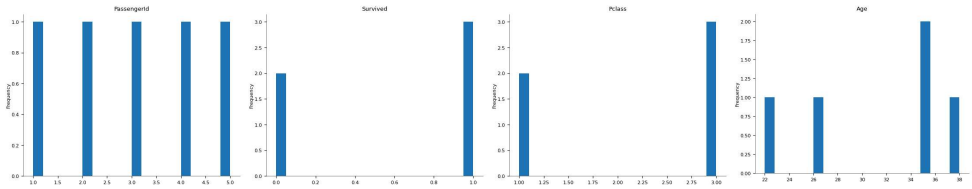
1 to 5 of 5 entries Filter

index	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Thayer)	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.05	NaN	S

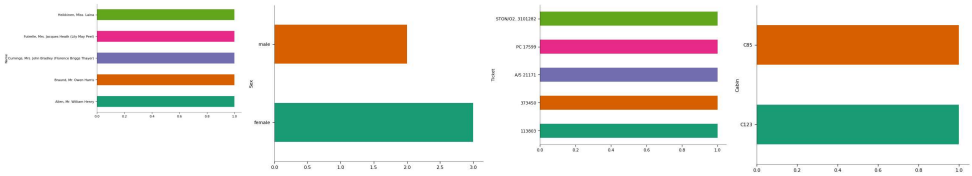
Show 25 per page

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

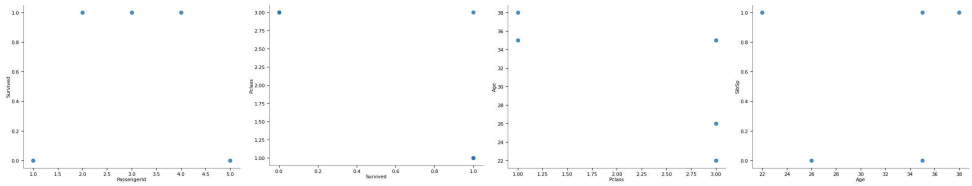
Distributions



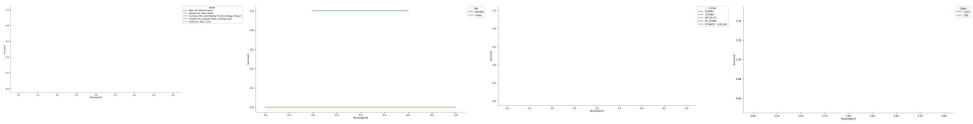
Categorical distributions



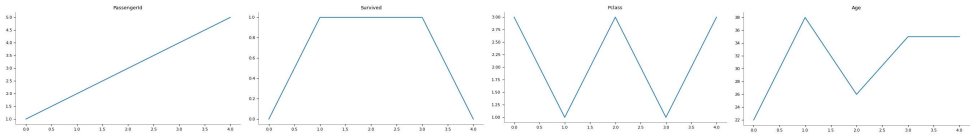
2-d distributions



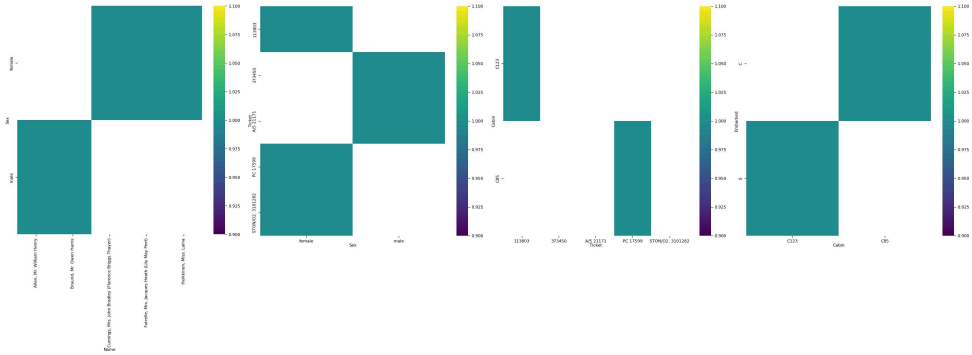
Time series



Values



2-d categorical distributions



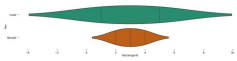
Faceted distributions

```
<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 `legend

                    <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 `legend
```



```
<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 `legend
```

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

```
# Create SQL DB
conn = sqlite3.connect('titanic.db')

# Store data into SQL table
df.to_sql('titanic_data', conn, if_exists='replace', index=False)
```

```
891
```

```
query = '''
SELECT Sex, COUNT(*) as Total,
SUM(Survived) as Survived,
ROUND(AVG(Survived)*100, 2) as Survival_Rate
FROM titanic_data
GROUP BY Sex
'''
pd.read_sql_query(query, conn)
```

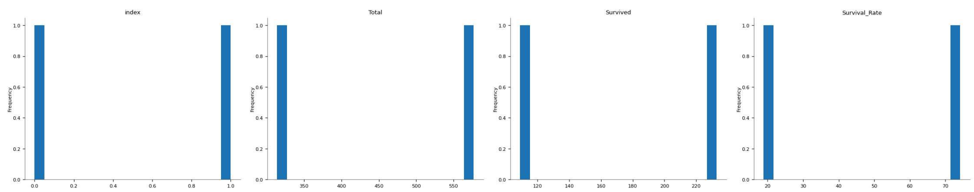
1 to 2 of 2 entries Filter

index	Sex	Total	Survived	Survival_Rate
0	female	314	233	74.2
1	male	577	109	18.89

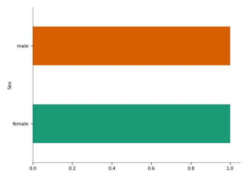
Show 25 per page

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

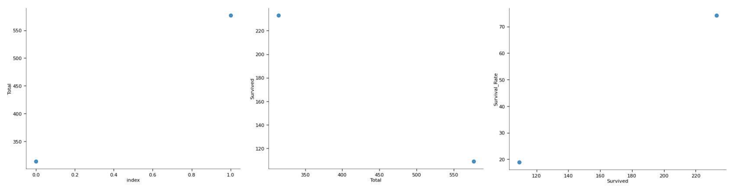
Distributions



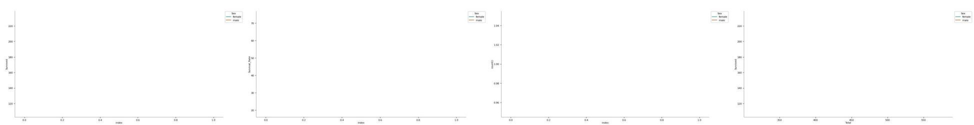
Categorical distributions



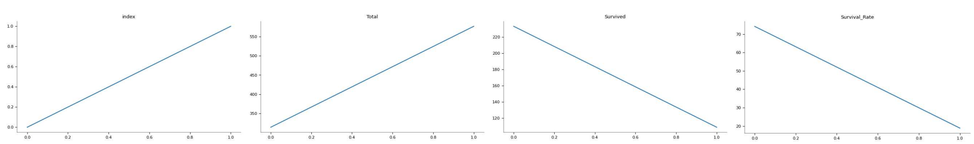
2-d distributions



Time series



Values



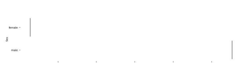
Faceted distributions

```
<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 `legend
```

```
<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 `legend
```



```
<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 `legend
```

```
<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 `legend
```

```
# Fill missing values
df['Age'].fillna(df['Age'].mean(), inplace=True)
df['Embarked'].fillna(df['Embarked'].mode()[0], inplace=True)

# Convert categorical to numeric
df['Sex'] = df['Sex'].map({'male': 0, 'female': 1})
```

```
df = pd.get_dummies(df, columns=['Embarked', 'Pclass'], drop_first=True)
```

```
/tmp/ipython-input-11-3309191636.py: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 valu

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


  df['Age'].fillna(df['Age'].mean(), inplace=True)
/tmp/ipython-input-11-3309191636.py: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 valu

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


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

```
df.drop(['Name', 'Ticket', 'Cabin', 'PassengerId'], axis=1, inplace=True)
```

```
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score

X = df.drop('Survived', axis=1)
y = df['Survived']

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

model = RandomForestClassifier()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)
print("Model Accuracy:", accuracy_score(y_test, y_pred))
```

```
Model Accuracy: 0.8100558659217877
```

```
def would_you_survive(age, sex, fare, sibsp, parch, embarked_Q, embarked_S, pclass_2, pclass_3):
    input_data = [[age, sex, fare, sibsp, parch, embarked_Q, embarked_S, pclass_2, pclass_3]]
    prediction = model.predict(input_data)
    return "🎉 You would survive!" if prediction[0] == 1 else "💔 You would not survive."

# Example: A 22-year-old female, fare 70, 0 siblings, 0 parents, embarked S, class 3
would_you_survive(22, 1, 70, 0, 0, 0, 1, 0, 1)
```

```
'🎉 You would survive!'
```