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

```
import pandas as pd
df = pd.read_csv('/content/test.csv')
df.head()
```

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

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

```
import numpy as np
print(df.isnull().sum())

for col in df.select_dtypes(include=np.number).columns:
    if df[col].isnull().any():
        df[col].fillna(df[col].mean(), inplace=True)

for col in df.select_dtypes(include='object').columns:
    if df[col].isnull().any():
        df[col].fillna(df[col].mode()[0], inplace=True)

print("\nMissing values after filling:")
print(df.isnull().sum())
```

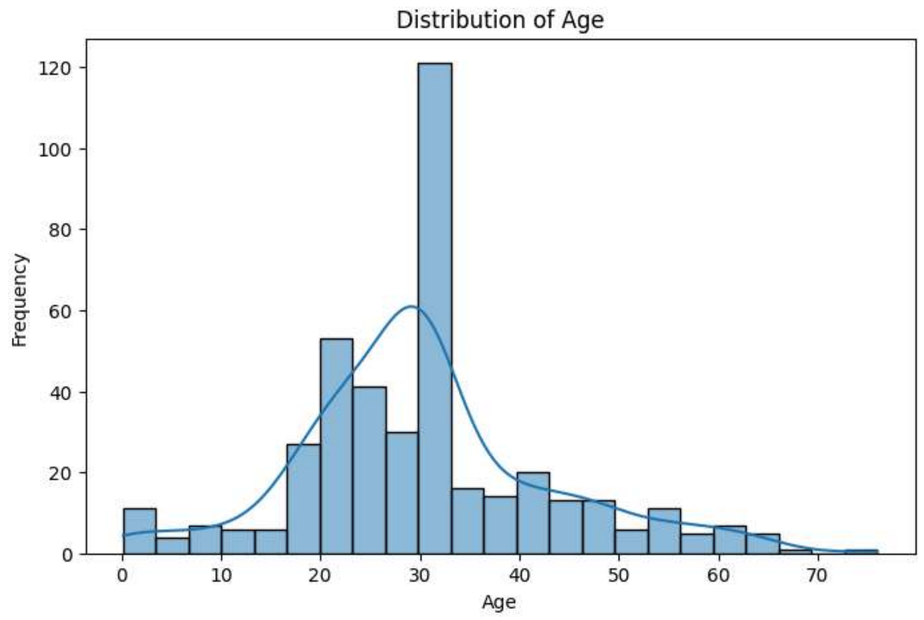
```
PassengerId      0
Pclass           0
Name             0
Sex              0
Age             86
SibSp            0
Parch            0
Ticket           0
Fare             1
Cabin          327
Embarked         0
dtype: int64

Missing values after filling:
PassengerId    0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
/tmp/ipython-input-3-562877466.py:6: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained a
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[co


  df[col].fillna(df[col].mean(), inplace=True)
/tmp/ipython-input-3-562877466.py:10: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained
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[co


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

```
print("\nDescriptive Statistics
print(df.describe(include='all'
print("\nMissing Values:")
print(df.isnull().sum())
```

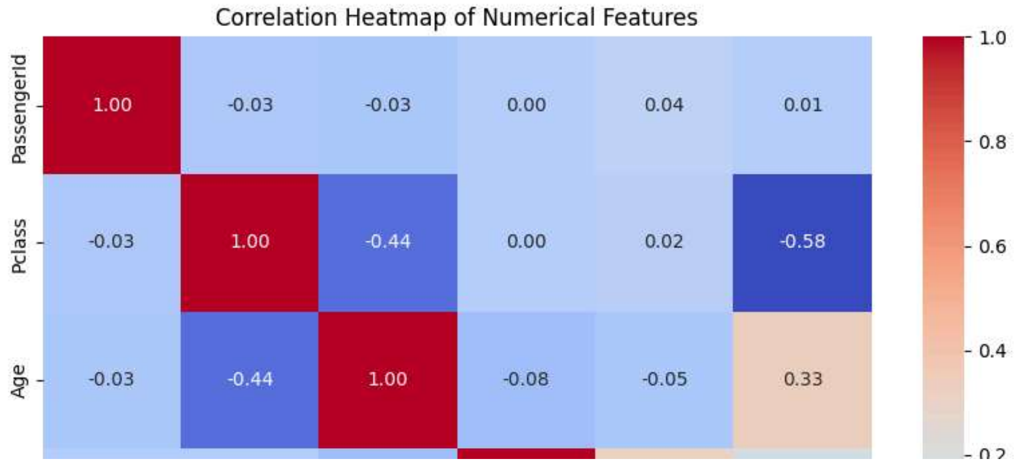
What can I help you build?

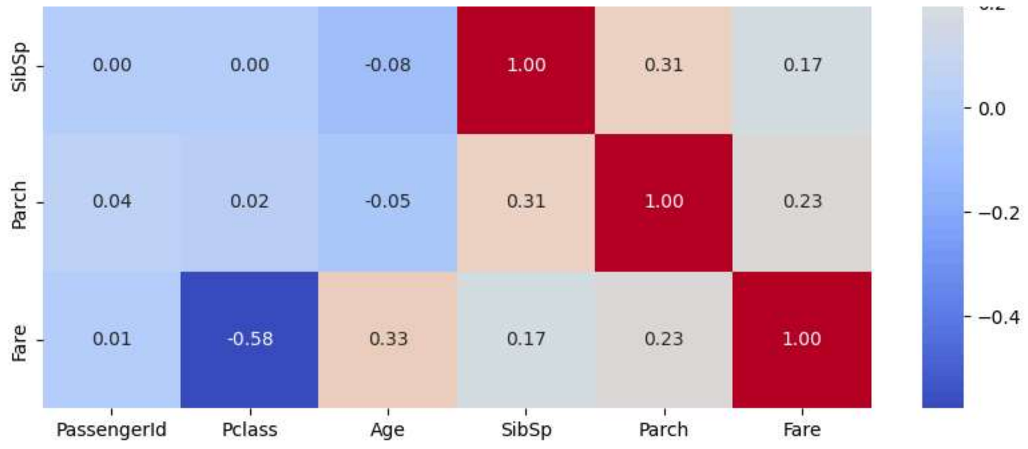
```
if 'Age' in df.columns and pd.api.types.is_numeric_dtype(df['Age']):
    plt.figure(figsize=(8, 5))
    sns.histplot(df['Age'].dropna(), kde=True)
    plt.title('Distribution of Age')
    plt.xlabel('Age')
    plt.ylabel('Frequency')
    plt.show()
else:
    print("\n'Age' column not found or is not numeric. Please replace 'Age' with a valid numerical column name for visualization.")

if 'Gender' in df.columns and pd.api.types.is_object_dtype(df['Gender']):
    plt.figure(figsize=(8, 5))
    sns.countplot(x='Gender', data=df)
    plt.title('Distribution of Gender')
    plt.xlabel('Gender')
    plt.ylabel('Count')
    plt.show()
else:
     print("\n'Gender' column not found or is not categorical. Please replace 'Gender' with a valid categorical column name for visuali

numerical_df = df.select_dtypes(include=np.number)
if not numerical_df.empty:
    plt.figure(figsize=(10, 8))
    sns.heatmap(numerical_df.corr(), annot=True, cmap='coolwarm', fmt=".2f")
    plt.title('Correlation Heatmap of Numerical Features')
    plt.show()
else:
    print("\nNo numerical columns found for correlation heatmap.")
```

```
Descriptive Statistics:
        PassengerId      Pclass                      Name   Sex         Age  \
count    418.000000  418.000000                       418   418  418.000000
unique          NaN         NaN                       418     2         NaN
top             NaN         NaN  Peter, Master. Michael J  male         NaN
freq            NaN         NaN                         1   266         NaN
mean    1100.500000    2.265550                       NaN   NaN   30.272590
std      120.810458    0.841838                       NaN   NaN   12.634534
min      892.000000    1.000000                       NaN   NaN    0.170000
25%      996.250000    1.000000                       NaN   NaN   23.000000
50%     1100.500000    3.000000                       NaN   NaN   30.272590
75%     1204.750000    3.000000                       NaN   NaN   35.750000
max     1309.000000    3.000000                       NaN   NaN   76.000000

             SibSp       Parch    Ticket        Fare            Cabin Embarked
count   418.000000  418.000000       418  418.000000              418      418
unique         NaN         NaN       363         NaN               76        3
top            NaN         NaN  PC 17608         NaN  B57 B59 B63 B66        S
freq           NaN         NaN         5         NaN              330      270
mean      0.447368    0.392344       NaN   35.627188              NaN      NaN
std       0.896760    0.981429       NaN   55.840500              NaN      NaN
min       0.000000    0.000000       NaN    0.000000              NaN      NaN
25%       0.000000    0.000000       NaN    7.895800              NaN      NaN
50%       0.000000    0.000000       NaN   14.454200              NaN      NaN
75%       1.000000    0.000000       NaN   31.500000              NaN      NaN
max       8.000000    9.000000       NaN  512.329200              NaN      NaN

Missing Values:
PassengerId    0
Pclass         0
Name           0
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin          0
Embarked       0
dtype: int64
```



```
'Gender' column not found or is not categorical. Please replace 'Gender' with a valid categorical column name for visualization.
```





```
print("\nShape before outlier removal:", df.shape)
for col in df.select_dtypes(include=np.number).columns:
    Q1 = df[col].quantile(0.25)
    Q3 = df[col].quantile(0.75)
    IQR = Q3 - Q1
    lower_bound = Q1 - 1.5 * IQR
    upper_bound = Q3 + 1.5 * IQR
    df = df[(df[col] >= lower_bound) & (df[col] <= upper_bound)]

print("Shape after outlier removal:", df.shape)

if 'SibSp' in df.columns and 'Parch' in df.columns:
    df['FamilySize'] = df['SibSp'] + df['Parch'] + 1
    print("\nAdded 'FamilySize' column.")

categorical_cols_to_encode = df.select_dtypes(include='object').columns.tolist()
if 'Name' in categorical_cols_to_encode:
  categorical_cols_to_encode.remove('Name')

if categorical_cols_to_encode:
    df = pd.get_dummies(df, columns=categorical_cols_to_encode, drop_first=True)
    print(f"\nOne-hot encoded columns: {categorical_cols_to_encode}")
    print("DataFrame after one-hot encoding:")
    print(df.head())
else:
    print("\nNo suitable categorical columns found for one-hot encoding.")


print("\nFinal DataFrame Info:")
df.info()
print("\nFinal DataFrame Head:")
print(df.head())
```

```
1  7.0000          2     False          False  ...      False      False
3  8.6625          1      True          False  ...      False      False
5  9.2250          1      True          False  ...      False      False
6  7.6292          1     False          False  ...      False      False

   Cabin_F  Cabin_F E46  Cabin_F E57  Cabin_F G63  Cabin_F2  Cabin_F33  \
0    False        False        False        False     False      False
1    False        False        False        False     False      False
3    False        False        False        False     False      False
5    False        False        False        False     False      False
6    False        False        False        False     False      False

   Embarked_Q  Embarked_S
0        True       False
1       False        True
3       False        True
```

```
5          897         3         Svensson, Mr. Johan Cervin  14.0        0        0
```

```
5         897       3        Svensson, Mr. Johan Cervin  14.0      0      0
6         898       3              Connolly, Miss. Kate  30.0      0      0

     Fare  FamilySize  Sex_male  Ticket_110489  ...  Cabin_E46  Cabin_E60  \
0  7.8292           1      True          False  ...      False      False
1  7.0000           2     False          False  ...      False      False
3  8.6625           1      True          False  ...      False      False
5  9.2250           1      True          False  ...      False      False
6  7.6292           1     False          False  ...      False      False

   Cabin_F  Cabin_F E46  Cabin_F E57  Cabin_F G63  Cabin_F2  Cabin_F33  \
0    False        False        False        False     False      False
1    False        False        False        False     False      False
3    False        False        False        False     False      False
5    False        False        False        False     False      False
6    False        False        False        False     False      False

   Embarked_Q  Embarked_S
0        True       False
1       False        True
3       False        True
5       False        True
6        True       False

[5 rows x 301 columns]
```

```python
import pandas as pd
import matplotlib.pyplot as plt
if 'Survived' in df.columns:
    plt.figure(figsize=(8, 5))
    sns.countplot(x='Survived', data=df)
    plt.title('Distribution of Survival')
    plt.xlabel('Survived')
    plt.ylabel('Count')
    plt.show()
else:
    print("\n'Survived' column not found.")

if 'Pclass' in df.columns:
    plt.figure(figsize=(8, 5))
    sns.countplot(x='Pclass', data=df, hue='Survived' if 'Survived' in df.columns else None)
    plt.title('Survival Count by Pclass')
    plt.xlabel('Pclass')
    plt.ylabel('Count')
    if 'Survived' in df.columns:
        plt.legend(title='Survived')
    plt.show()
else:
    print("\n'Pclass' column not found.")

if 'Age' in df.columns and 'Survived' in df.columns and pd.api.types.is_numeric_dtype(df['Age']):
    plt.figure(figsize=(10, 6))
    sns.boxplot(x='Survived', y='Age', data=df)
    plt.title('Age Distribution by Survival')
    plt.xlabel('Survived')
    plt.ylabel('Age')
    plt.show()
else:
    print("\n'Age' or 'Survived' column not found or 'Age' is not numeric for Age vs Survival boxplot.")

if 'Fare' in df.columns and 'Survived' in df.columns and pd.api.types.is_numeric_dtype(df['Fare']):
    plt.figure(figsize=(10, 6))
    sns.boxplot(x='Survived', y='Fare', data=df)
    plt.title('Fare Distribution by Survival')
    plt.xlabel('Survived')
    plt.ylabel('Fare')
    plt.show()
else:
    print("\n'Fare' or 'Survived' column not found or 'Fare' is not numeric for Fare vs Survival boxplot.")

if 'Embarked' in df.columns and 'Survived' in df.columns and pd.api.types.is_object_dtype(df['Embarked']):
     plt.figure(figsize=(8, 5))
     sns.countplot(x='Embarked', data=df, hue='Survived')
     plt.title('Survival Count by Embarked')
     plt.xlabel('Embarked')
     plt.ylabel('Count')
     plt.legend(title='Survived')
     plt.show()
else:
     print("\n'Embarked' or 'Survived' column not found or 'Embarked' is not categorical for Embarked vs Survival countplot.")
```