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

Choose Files Titanic-Dataset.csv

- **Titanic-Dataset.csv**(text/csv) - 61194 bytes, last modified: 6/9/2025 - 100% done

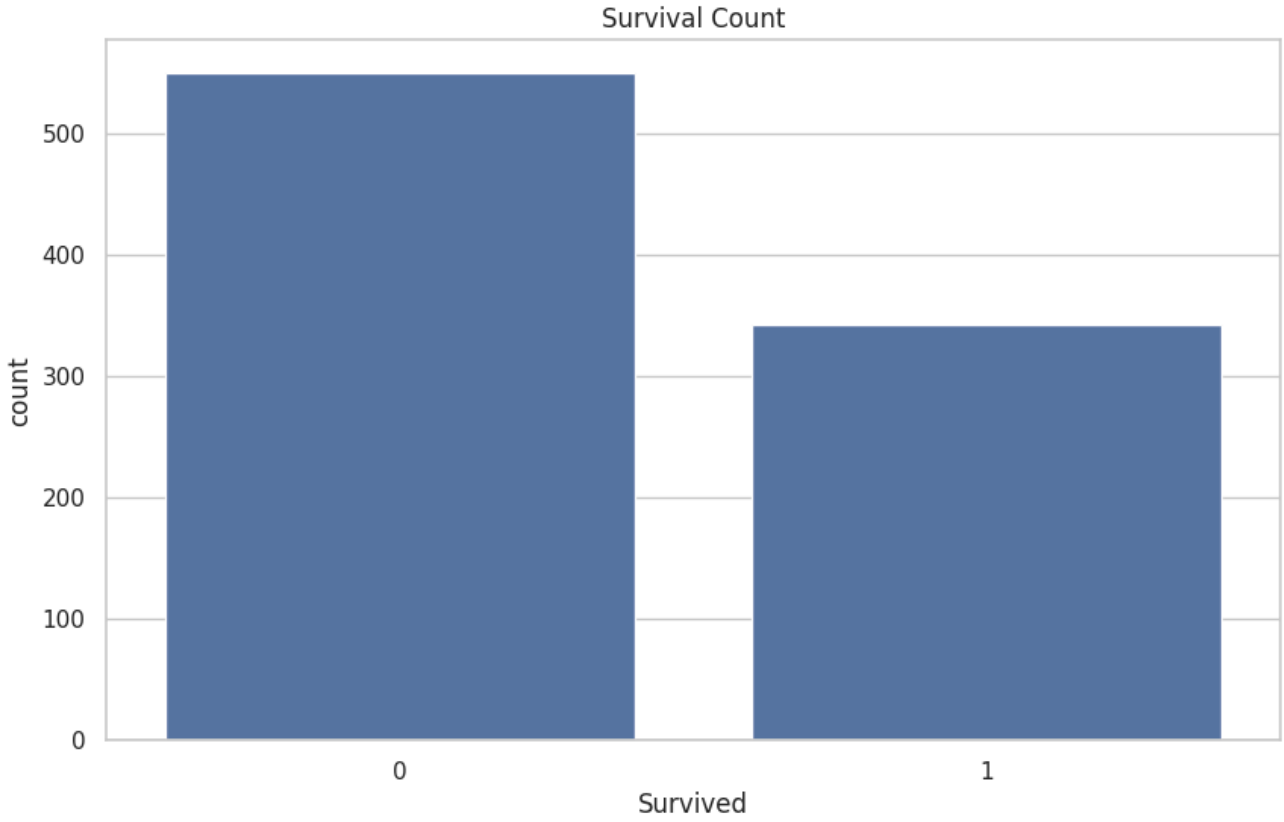
```
Saving Titanic-Dataset.csv to Titanic-Dataset.csv
```

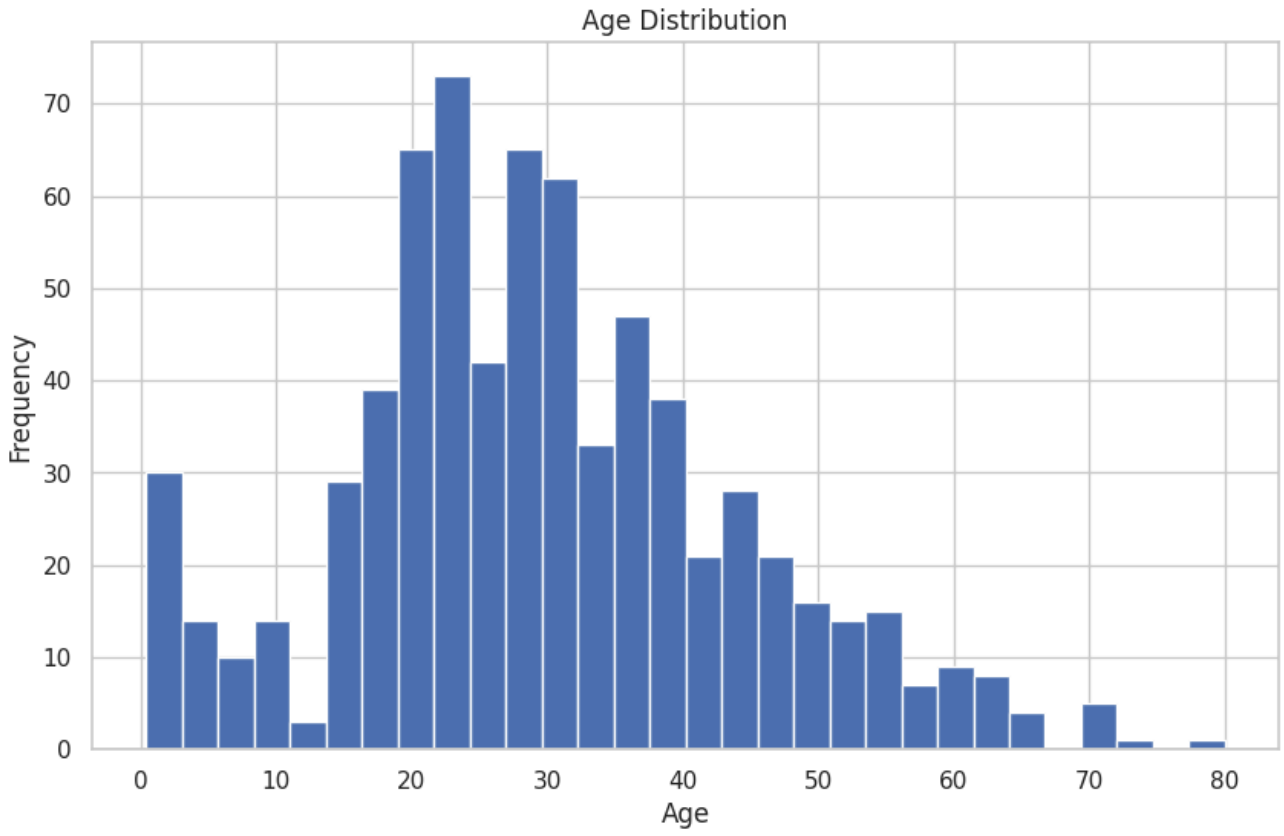
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

# Set plot styles
sns.set(style="whitegrid")
plt.rcParams["figure.figsize"] = (10, 6)

# Load the dataset
df = pd.read_csv('Titanic-Dataset.csv')

# 1. Basic Data Overview
print("First 5 rows:")
print(df.head())

print("\nData Info:")
print(df.info())

print("\nStatistical Summary:")
print(df.describe(include='all'))

print("\nMissing Values:")
print(df.isnull().sum())

print("\nDuplicate Rows:", df.duplicated().sum())
```

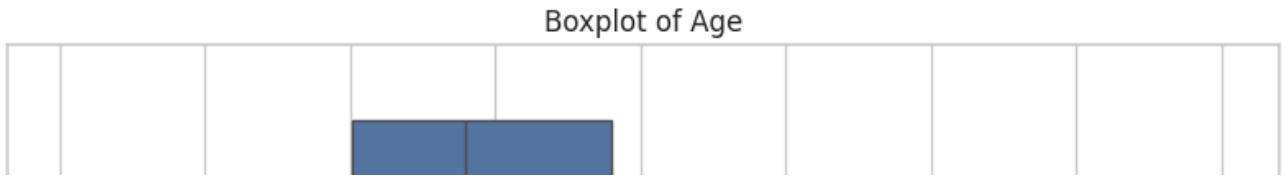
```
mean     446.000000    0.383838    2.308642                    NaN   NaN
std      257.353842    0.486592    0.836071                    NaN   NaN
min        1.000000    0.000000    1.000000                    NaN   NaN
25%      223.500000    0.000000    2.000000                    NaN   NaN
50%      446.000000    0.000000    3.000000                    NaN   NaN
75%      668.500000    1.000000    3.000000                    NaN   NaN
max      891.000000    1.000000    3.000000                    NaN   NaN

                Age       SibSp       Parch  Ticket        Fare Cabin Embarked
count   714.000000  891.000000  891.000000     891  891.000000   204      889
unique         NaN         NaN         NaN     681         NaN   147        3
top            NaN         NaN         NaN  347082         NaN    G6        S
freq           NaN         NaN         NaN       7         NaN     4      644
mean     29.699118    0.523008    0.381594     NaN   32.204208   NaN      NaN
std      14.526497    1.102743    0.806057     NaN   49.693429   NaN      NaN
min       0.420000    0.000000    0.000000     NaN    0.000000   NaN      NaN
25%      20.125000    0.000000    0.000000     NaN    7.910400   NaN      NaN
50%      28.000000    0.000000    0.000000     NaN   14.454200   NaN      NaN
75%      38.000000    1.000000    0.000000     NaN   31.000000   NaN      NaN
max      80.000000    8.000000    6.000000     NaN  512.329200   NaN      NaN

Missing Values:
PassengerId      0
Survived         0
Pclass           0
Name             0
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64

Duplicate Rows: 0
```

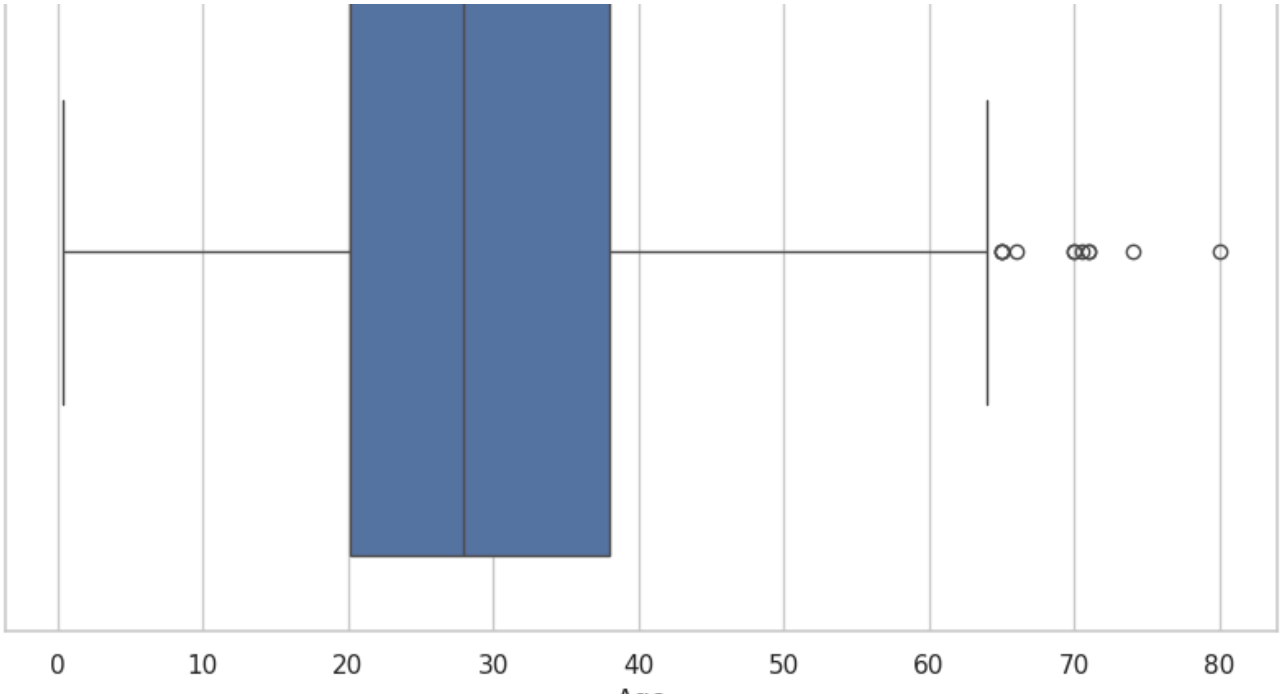
```
print("\nValue Counts for Survived:")
print(df['Survived'].value_counts())

sns.countplot(x='Survived', data=df)
plt.title("Survival Count")
plt.show()

# Numerical: Age
df['Age'].hist(bins=30)
plt.title("Age Distribution")
plt.xlabel("Age")
plt.ylabel("Frequency")
plt.show()

sns.boxplot(x='Age', data=df)
plt.title("Boxplot of Age")
plt.show()
```

```
Survived
0    549
1    342
Name: count, dtype: int64
```

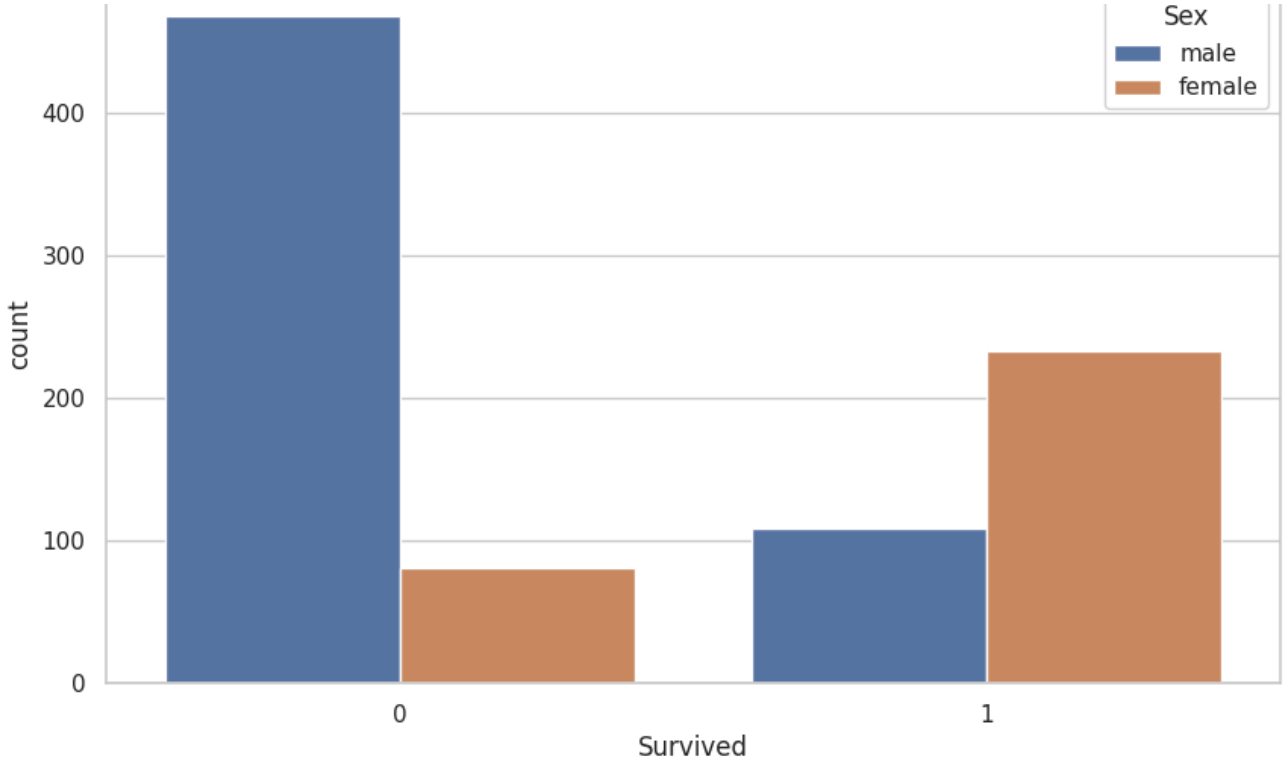


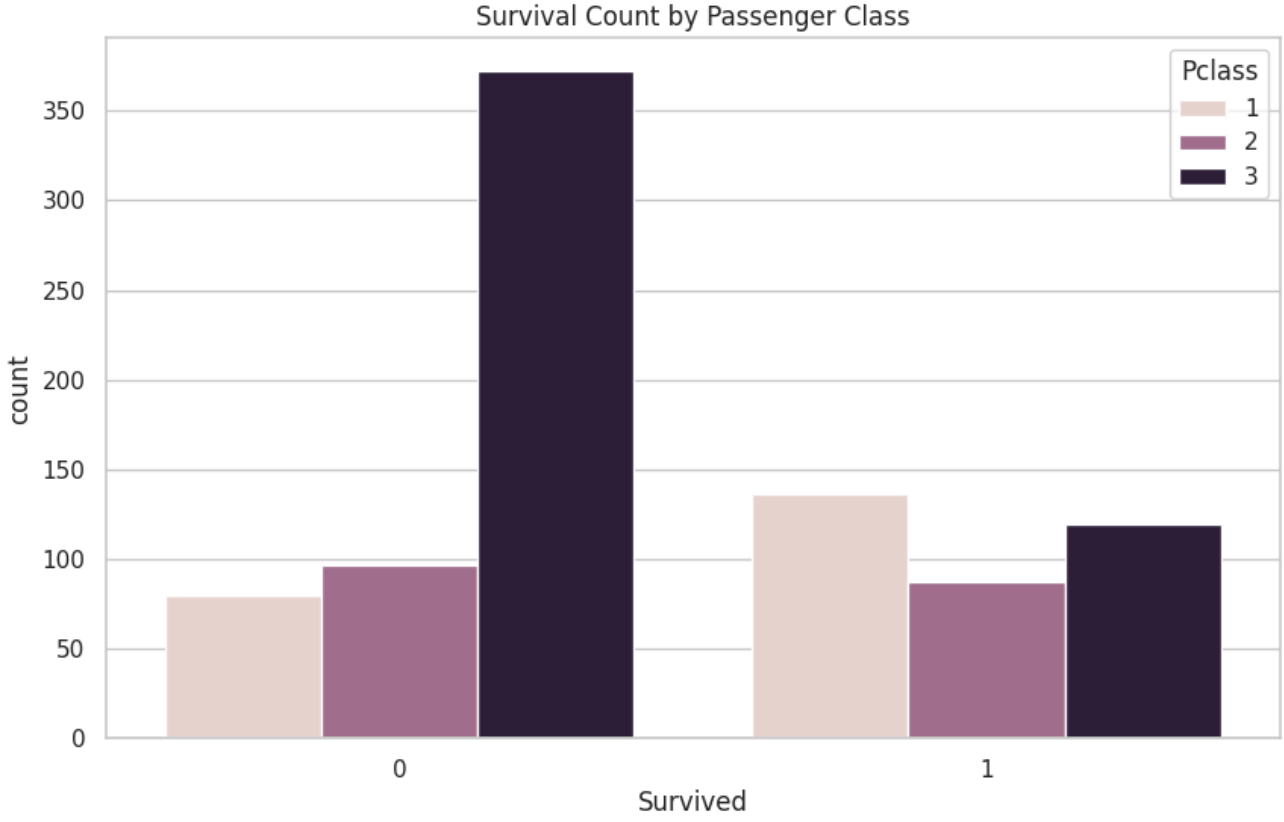


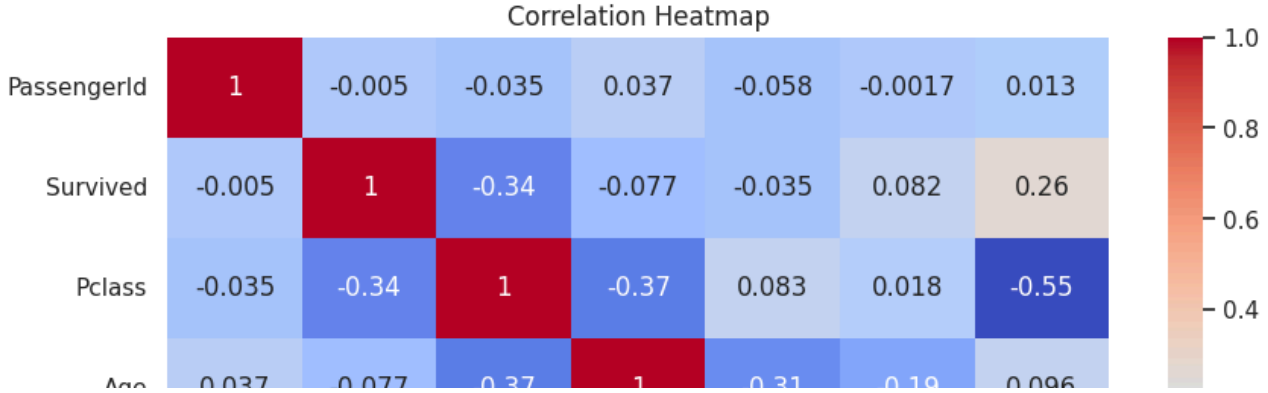





```
sns.countplot(x='Survived', hue='Sex', data=df)
plt.title("Survival Count by Gender")
plt.show()

# Survived vs Pclass
sns.countplot(x='Survived', hue='Pclass', data=df)
plt.title("Survival Count by Passenger Class")
plt.show()

# Heatmap of correlations (numeric features)
numeric_df = df.select_dtypes(include=['int64', 'float64'])
sns.heatmap(numeric_df.corr(), annot=True, cmap='coolwarm')
plt.title("Correlation Heatmap")
plt.show()
```


Sex
male
female
count
400
300
200
100
0
0
1
Survived


Survival Count by Passenger Class
Pclass
1
2
3
count
350
300
250
200
150
100
50
0
0
1
Survived


Correlation Heatmap
PassengerId
Survived
Pclass
Age
1
-0.005
-0.035
0.037
-0.058
-0.0017
0.013
-0.005
1
-0.34
-0.077
-0.035
0.082
0.26
-0.035
-0.34
1
-0.37
0.083
0.018
-0.55
0.037
0.077
0.37
1
0.31
0.19
0.096
1.0
0.8
0.6
0.4


```
print("\nAverage Age by Survival Status:")
print(df.groupby('Survived')['Age'].mean())

print("\nSurvival Rate by Passenger Class:")
print(df.groupby('Pclass')['Survived'].mean())
```

```
Average Age by Survival Status:
Survived
0    30.626179
1    28.343690
Name: Age, dtype: float64

Survival Rate by Passenger Class:
Pclass
1    0.629630
2    0.472826
3    0.242363
Name: Survived, dtype: float64
```

```
print("\nSummary:")
print("- More passengers did not survive (0) than survived (1).")
print("- Females had a higher survival rate than males.")
print("- Passengers in higher classes (Pclass=1) had better chances of survival."
print("- Age distribution is right-skewed, with many young passengers.")
```

```
Summary:
- More passengers did not survive (0) than survived (1).
- Females had a higher survival rate than males.
- Passengers in higher classes (Pclass=1) had better chances of survival.
- Age distribution is right-skewed, with many young passengers.
```

```
!apt-get install -y texlive-xetex texlive-fonts-recommended texlive-gene
!jupyter nbconvert --to pdf --output "Titanic_EDA_Report.pdf" "Titanic_E
```

```
        The nbformat version to write.
                Use this to downgrade notebooks.
            Choices: any of [1, 2, 3, 4]
            Default: 4
            Equivalent to: [--NotebookExporter.nbformat_version]

        Examples
        --------

            The simplest way to use nbconvert is

                    > jupyter nbconvert mynotebook.ipynb --to html

                    Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown

                    > jupyter nbconvert --to latex mynotebook.ipynb

                    Both HTML and LaTeX support multiple output templates. LaTeX incl
                    'base', 'article' and 'report'.  HTML includes 'basic', 'lab' and
                    'classic'. You can specify the flavor of the format used.

                    > jupyter nbconvert --to html --template lab mynotebook.ipynb

                    You can also pipe the output to stdout, rather than a file

                    > jupyter nbconvert mynotebook.ipynb --stdout

                    PDF is generated via latex

                    > jupyter nbconvert mynotebook.ipynb --to pdf

                    You can get (and serve) a Reveal.js-powered slideshow

                    > jupyter nbconvert myslides.ipynb --to slides --post serve

                    Multiple notebooks can be given at the command line in a couple o
                    different ways:

                    > jupyter nbconvert notebook*.ipynb
                    > jupyter nbconvert notebook1.ipynb notebook2.ipynb

                    or you can specify the notebooks list in a config file, containin

                        c.NbConvertApp.notebooks = ["my_notebook.ipynb"]

                    > jupyter nbconvert --config mycfg.py

        To see all available configurables, use `--help-all`.
```

```
from google.colab import files
files.download("Titanic_EDA_Report.pdf")
```