



## Data Science with Python Internship – Task 4

Mini Visualization Dashboard (Matplotlib + Seaborn) on Titanic or Student Performance



### Task Description

In Task-3, you performed a deeper EDA on the Titanic dataset (cleaning, grouping, and basic visualizations). Now you will create a **mini data visualization dashboard** inside a single Jupyter Notebook. Your goal is to communicate key insights with **well-chosen charts** and short, clear interpretations beneath each figure.

You may use **Titanic** (recommended) or **Student Performance** (UCI). Focus on: data cleaning, feature engineering (simple), multiple plot types, and concise insights.

## What You'll Build (Requirements)

### 1. Data Prep

- Load CSV into pandas.
- Handle missing values (e.g., fill Age by mean/median; drop columns like Cabin if using Titanic).
- Create 1–2 helpful features (e.g., **AgeGroup**, **FamilySize** = SibSp + Parch, or **Pass/Fail** for Student dataset).

### 2. Dashboard Plots ( $\geq 5$ distinct charts)

- **Histogram** of a numeric variable (e.g., Age).
- **Bar chart** comparing categories (e.g., Survival by Sex / Class; or Average score by study time).
- **Boxplot** to show distribution and outliers (e.g., Fare by Class; or Math score by parental education).
- **Scatterplot** for relationship (e.g., Age vs. Fare colored by Survival).
- **Heatmap** of correlations (numeric columns).

### 3. Narration & Layout

- Use Markdown headings to separate sections: *Overview, Cleaning, Features, Visualizations, Insights, Conclusion.*
- Add 1–2 lines of interpretation under each plot (what does the chart show? what's the takeaway?).

### 4. Good Practices

- Label axes, add titles, and legends where needed.
- Keep code cells short and readable (one chart per cell).
- Avoid hardcoding paths; keep `data/` folder or mount via Colab.

## Deliverable

- A single **Jupyter Notebook (.ipynb)** that:
  - Loads & cleans data, creates features, and renders  $\geq 5$  charts.
  - Includes **clear Markdown commentary** and **insights** below each figure.
- Optional: export key figures as `.png` into an `images/` folder.
- Submit notebook + a brief **README** (dataset used, how to run).

## Sample Starter Code (Step-by-Step & Easy to Extend)

### Step 1: Import Libraries & Setup

```
python

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

sns.set() # Apply Seaborn styling
```

### Step 2: Load the Dataset

(Titanic example – change filename if using Student dataset)

```
python

df = pd.read_csv("titanic.csv")
print(df.head()) # preview first 5 rows
print(df.info()) # check columns & null values
```

### Step 3: Data Cleaning

```
python

# Fill missing Age values with median
df["Age"] = df["Age"].fillna(df["Age"].median())

# Fill missing Embarked with mode
df["Embarked"] = df["Embarked"].fillna(df["Embarked"].mode()[0])

# Drop unused column
df = df.drop(columns=["Cabin"], errors="ignore")
```

### Step 4: Feature Engineering

```
python

# Add FamilySize = SibSp + Parch
df["FamilySize"] = df["SibSp"] + df["Parch"]

# Create Age Groups
df["AgeGroup"] = pd.cut(
    df["Age"], bins=[0,12,18,30,50,80],
    labels=["Child","Teen","YoungAdult","Adult","Senior"]
)
```

### Step 5: Visualizations

#### Histogram – Age Distribution

```
python

sns.histplot(df["Age"], bins=20, kde=True)
plt.title("Age Distribution")
plt.show()
```

## Bar Chart – Survival Rate by Sex

```
python

sns.barplot(x="Sex", y="Survived", data=df)
plt.title("Survival Rate by Gender")
plt.show()
```

## Boxplot – Fare by Passenger Class

```
python

sns.boxplot(x="Pclass", y="Fare", data=df)
plt.title("Fare Distribution by Class")
plt.show()
```

## Scatterplot – Age vs. Fare

```
python

sns.scatterplot(x="Age", y="Fare", hue="Survived", data=df, alpha=0.7)
plt.title("Age vs Fare (colored by Survival)")
plt.show()
```

## Heatmap – Correlations

```
python

sns.heatmap(df.corr(), annot=True, cmap="coolwarm")
plt.title("Correlation Heatmap")
plt.show()
```

**Tip:** After each plot, add 1–2 clear sentences in Markdown summarizing the insight.

**Optional:** Try facet grids, e.g., `sns.catplot(col="Sex", x="Pclass", y="Survived", kind="bar", data=df)` to compare subgroups.

---

## Helpful Resources (Free)

- Seaborn Documentation: <https://seaborn.pydata.org/>
- Matplotlib Gallery: <https://matplotlib.org/stable/gallery/index.html>
- Pandas User Guide: <https://pandas.pydata.org/docs/>
- Kaggle Titanic Dataset: <https://www.kaggle.com/c/titanic/data>
- Student Performance (UCI): <https://archive.ics.uci.edu/> (Student Performance dataset)

---

## Tools to Use

- Google Colab (recommended) or Jupyter Notebook locally
- Python Libraries: pandas, numpy, seaborn, matplotlib
- Version Control (optional): GitHub for sharing your notebook

## Why This Task Is Important

- Trains you to **tell a data story visually**, which is the core of DS/DA roles.
- Builds muscle memory for **pandas + Matplotlib/Seaborn** workflows.
- Teaches you to document findings clearly, improving **communication skills**.