Task 5

- task5-eda/titanic_dataset.csv — dataset used for EDA
- task5-eda/titanic_eda_notebook.ipynb — runnable Jupyter notebook documenting EDA steps
- task5-eda/titanic_eda_analysis.py — Python script reproducing charts
- task5-eda/describe.csv and missing_values.csv — summary outputs

How to run

1. Create a new folder on your computer and save the script below into a file named build_task5_eda_bundle.py.
2. In a terminal / command prompt, install dependencies:

```
python -m pip install --upgrade pip

pip install pandas seaborn matplotlib nbformat reportlab
```

3. Run the script:

```
python build_task5_eda_bundle.py
```

4. After the script finishes:
   - Open task5-eda/task5_eda_report.pdf — that's the professional PDF ready to upload.
   - Or upload task5-eda-bundle.zip to GitHub as your repository contents.

The script

```
#!/usr/bin/env python3

"""

build_task5_eda_bundle.py

Creates EDA deliverables for Task 5 (Titanic dataset):

- CSV dataset (seaborn titanic)

- Jupyter notebook (.ipynb)

- Analysis script (.py)
```

```
Usage:
    pip install pandas seaborn matplotlib nbformat reportlab
    python build_task5_eda_bundle.py
"""
import os, zipfile, shutil, textwrap
from pathlib import Path
from datetime import datetime
import pandas as pd, numpy as np, seaborn as sns, matplotlib.pyplot as plt
import nbformat as nbf

OUTDIR = Path("task5-eda")
if OUTDIR.exists():
    shutil.rmtree(OUTDIR)
OUTDIR.mkdir(parents=True)
IMGDIR = OUTDIR / "images"
IMGDIR.mkdir()

# 1) Load dataset (seaborn titanic sample)
df = sns.load_dataset("titanic")
(df).to_csv(OUTDIR / "titanic_dataset.csv", index=False)

# 2) Basic EDA artifacts
desc = df.describe(include='all').transpose()
desc.to_csv(OUTDIR / "describe.csv")
missing = df.isnull().sum().sort_values(ascending=False)
missing.to_csv(OUTDIR / "missing_values.csv", header=["missing_count"])

# Derived columns for useful groupings
```

```
df['age_group'] = pd.cut(df['age'], bins=[0,12,18,30,45,60,80], labels=["Child","Teen","Young Adult","Adult","Mid Age","Senior"])
df['fare_bin'] = pd.qcut(df['fare'].fillna(0)+0.01, q=4, labels=["Low","Medium","High","Very High"])

# 3) Save a few summary CSVs
df.groupby('pclass')['survived'].mean().reset_index().rename(columns={'survived':'survival_rate'}).to_csv(OUTDIR / "survival_by_pclass.csv", index=False)
df.groupby('sex')['survived'].mean().reset_index().rename(columns={'survived':'survival_rate'}).to_csv(OUTDIR / "survival_by_sex.csv", index=False)
df.groupby('age_group')['survived'].mean().reset_index().rename(columns={'survived':'survival_rate'}).to_csv(OUTDIR / "survival_by_agegroup.csv", index=False)

# 4) Figures (high-res)
sns.set()
plt.figure(figsize=(8,4))
sns.countplot(data=df, x='survived')
plt.title("Survival Count (0 = No, 1 = Yes)")
plt.tight_layout(); plt.savefig(IMGDIR / "survival_count.png", dpi=200); plt.close()

plt.figure(figsize=(10,4))
sns.histplot(data=df, x='age', bins=30, kde=True)
plt.title("Age Distribution")
plt.tight_layout(); plt.savefig(IMGDIR / "age_histogram.png", dpi=200); plt.close()

plt.figure(figsize=(10,5))
sns.boxplot(data=df, x='survived', y='age')
plt.title("Age by Survival")
plt.tight_layout(); plt.savefig(IMGDIR / "age_boxplot_by_survival.png", dpi=200); plt.close()
plt.figure(figsize=(10,5))
sns.countplot(data=df, x='pclass', hue='survived')
plt.title("Passenger Class vs Survival")
plt.tight_layout(); plt.savefig(IMGDIR / "pclass_survival.png", dpi=200); plt.close()

plt.figure(figsize=(8,6))
sns.heatmap(df.select_dtypes(include=[np.number]).corr(), annot=True, fmt=".2f", cmap="vlag")
plt.title("Correlation Matrix (numeric)")
plt.tight_layout(); plt.savefig(IMGDIR / "correlation_heatmap.png", dpi=200); plt.close()

# pairplot (sample)
pair_cols = ['survived','age','fare','pclass']
pp = sns.pairplot(df[pair_cols].dropna(), hue='survived', corner=True, plot_kws={'alpha':0.5})
pp.savefig(IMGDIR / "pairplot_sample.png", dpi=200)
plt.close()

# 5) Create Jupyter notebook documenting EDA
nb = nbf.v4.new_notebook()
cells = []
cells.append(nbf.v4.new_markdown_cell("# Titanic - Exploratory Data Analysis (EDA)\nThis notebook shows the EDA steps: data loading, missing values, visuals, and insights."))
cells.append(nbf.v4.new_code_cell("import pandas as pd\nimport seaborn as sns\nimport matplotlib.pyplot as plt\nsns.set()\ndf = pd.read_csv('titanic_dataset.csv')\ndf.head()"))
cells.append(nbf.v4.new_markdown_cell("## Summary statistics"))
cells.append(nbf.v4.new_code_cell("df.describe(include='all').transpose()"))
cells.append(nbf.v4.new_markdown_cell("## Missing values"))
cells.append(nbf.v4.new_code_cell("df.isnull().sum().sort_values(ascending=False)"))
cells.append(nbf.v4.new_markdown_cell("## Visualizations"))
cells.append(nbf.v4.new_code_cell("import seaborn as sns\nsns.countplot(data=df, x='survived')\nplt.show()"))
nb['cells'] = cells
with open(OUTDIR / "titanic_eda_notebook.ipynb", "w") as f:
    nbf.write(nb, f)

# 6) Analysis script (.py)
analysis_script = \"\"\"# titanic_eda_analysis.py
import pandas as pd, seaborn as sns, matplotlib.pyplot as plt
df = pd.read_csv('titanic_dataset.csv')
sns.countplot(data=df, x='survived')
plt.title('Survival Count')
plt.savefig('images/survival_count_script.png')
plt.close()
\"\"\"
with open(OUTDIR / "titanic_eda_analysis.py", "w") as f:
    f.write(analysis_script)

# 7) README.md
readme = f\"\"\"# Task 5 - Exploratory Data Analysis (Titanic)
This folder contains EDA outputs for Task 5 using the Titanic dataset (seaborn sample).
Files:
- titanic_dataset.csv
- titanic_eda_notebook.ipynb
- titanic_eda_analysis.py
- images/
- describe.csv
- missing_values.csv
- survival_by_pclass.csv, survival_by_sex.csv, survival_by_agegroup.csv
```