# Task 2: Exploratory Data Analysis (EDA) - Titanic Dataset (Stanford Version)

# 1. Import Libraries

import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from scipy import stats

sns.set(style="whitegrid")

plt.rcParams["figure.figsize"] = (8, 5)

# 2. Load Dataset

url = "https://web.stanford.edu/class/archive/cs/cs109/cs109.1166/stuff/titanic.csv"

df = pd.read\_csv(url)

# Initial Inspection

print("Data Shape:", df.shape)

print("\nData Types:\n", df.dtypes)

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

# 3. Summary Statistics

print("\nSummary Stats:\n", df.describe(include='all'))

# 4. Handle Missing Values (only Age in this dataset)

if "Age" in df.columns:

df['Age'].fillna(df['Age'].median(), inplace=True)

# ===============================

# Distribution Analysis

# ===============================

# Numerical Features

if "Age" in df.columns and "Fare" in df.columns and "Pclass" in df.columns:

fig, ax = plt.subplots(1, 2, figsize=(14, 6))

sns.histplot(df['Age'], bins=30, kde=True, ax=ax[0]).set\_title('Age Distribution')

sns.boxplot(x='Pclass', y='Fare', data=df, ax=ax[1]).set\_title('Fare by Class')

plt.show()

# Categorical Features

if "Sex" in df.columns and "Survived" in df.columns:

plt.figure(figsize=(8, 4))

sns.countplot(x='Sex', hue='Survived', data=df).set\_title('Survival by Gender')

plt.show()

# ===============================

# Correlation & Relationships

# ===============================

plt.figure(figsize=(10, 6))

sns.heatmap(df.corr(numeric\_only=True), annot=True, cmap='coolwarm')

plt.title("Correlation Heatmap")

plt.show()

if "Pclass" in df.columns and "Survived" in df.columns:

print("\nSurvival by Pclass (%):\n")

print(pd.crosstab(df['Pclass'], df['Survived'], normalize='index') \* 100)

# ===============================

# Outlier Detection

# ===============================

if "Fare" in df.columns:

plt.figure(figsize=(8, 4))

sns.boxplot(x=df['Fare']).set\_title('Fare Outliers')

plt.show()

z\_scores = np.abs(stats.zscore(df['Fare']))

outliers = df[z\_scores > 3]

print(f"\nFound {len(outliers)} fare outliers")

# ===============================

# Advanced Visualizations

# ===============================

if "Age" in df.columns and "Pclass" in df.columns and "Survived" in df.columns:

g = sns.FacetGrid(df, col='Survived', row='Pclass', height=3)

g.map(sns.histplot, 'Age', bins=20)

plt.show()

if {"Age","Fare","Parch","Survived"}.issubset(df.columns):

sns.pairplot(df[['Age', 'Fare', 'Parch', 'Survived']], hue='Survived')

plt.show()

OUTPUT:











