import pandas as pd

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

titan = sns.load\_dataset('titanic')

titan.info()

print(titan.head())

print(titan.isnull().sum())

titan['age'].fillna(titan['age'].mean(), inplace=True)

# Fill missing values in 'embarked' with the forward fill method

titan['embarked'].fillna(method='ffill', inplace=True)

# Fill missing values in 'deck' with the mode

titan['deck'].fillna(titan['deck'].mode()[0], inplace=True)

# Fill missing values in 'embark\_town' with the forward fill method

titan['embark\_town'].fillna(method='ffill', inplace=True)

# Confirm no missing values remain

print(titan.isnull().sum())

# Descriptive statistics

print(titan.describe())

# Data Visualization

# Countplot for Gender

plt.figure(figsize=(8, 6))

sns.countplot(x='sex', data=titan, palette='pastel', hue='alive')

plt.title('Count of Passengers by Gender')

plt.xlabel('Gender')

plt.ylabel('Count')

plt.show()

# Countplot for Passenger Class

plt.figure(figsize=(8, 6))

sns.countplot(x='class', data=titan, palette='pastel', hue='alive')

plt.title('Count of Passengers by Class')

plt.xlabel('Class')

plt.ylabel('Count')

plt.show()

# Countplot for Embarked Town

plt.figure(figsize=(8, 6))

sns.countplot(x='embark\_town', data=titan, palette='pastel', hue='alive')

plt.title('Count of Passengers by Embarked Town')

plt.xlabel('Embarked Town')

plt.ylabel('Count')

plt.show()

# Heatmap for Survival by Passenger Class

group = titan.groupby(['pclass', 'survived'])

pclass\_survived = group.size().unstack()

sns.heatmap(pclass\_survived, annot=True, fmt="d")

plt.title('Survival Counts by Passenger Class')

plt.xlabel('Survived')

plt.ylabel('Passenger Class')

plt.show()

# Violinplot for Age by Gender and Survival

sns.violinplot(x='sex', y='age', hue='survived', data=titan, split=True)

plt.title('Age Distribution by Gender and Survival')

plt.xlabel('Gender')

plt.ylabel('Age')

plt.show()

# Catplot for Survival Rate by Family Size

titan['family\_Size'] = titan['parch'] + titan['sibsp']

sns.catplot(x='family\_Size', y='survived', data=titan, kind='point')

plt.title('Survival Rate by Family Size')

plt.xlabel('Family Size')

plt.ylabel('Survival Rate')

plt.show()

# Catplot for Survival Rate by Alone vs. with Family

titan['alone'] = 0

titan.loc[titan['family\_Size'] == 0, 'alone'] = 1

sns.catplot(x='alone', y='survived', data=titan, kind='point')

plt.title('Survival Rate for Passengers Traveling Alone vs. with Family')

plt.xlabel('Alone (1: Alone, 0: With Family)')

plt.ylabel('Survival Rate')

plt.show()

# Bar Plot for Fare (Continuous Feature)

titan['fare\_Range'] = pd.qcut(titan['fare'], 4)

sns.barplot(x='fare\_Range', y='survived', data=titan, order=titan['fare\_Range'].unique())

plt.title('Survival Rate by Fare Range')

plt.xlabel('Fare Range')

plt.ylabel('Survival Rate')

plt.show()

# Categorical Count Plots for Embarked Feature

sns.catplot(x='embarked', hue='survived', kind='count', col='pclass', data=titan)

plt.show()

# Histogram for Ticket Prices (Fare)

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

sns.histplot(titan['fare'], bins=30, kde=True, color='lightgreen')

plt.title('Distribution of Ticket Prices (Fare)')

plt.xlabel('Fare')

plt.ylabel('Frequency')

plt.show()

sns.catplot(x ="Glucose",hue ="Pregnancies", kind ="count",data = df)