# استيراد المكتبات

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sns

# تحميل البيانات

df = pd.read\_csv(r"railway 2 .csv")

# نظافة البيانات

print(df.head())

print(df.tail())

print(df.describe())

print(df.info())

print(df.isnull().sum())

print(df.dtypes)

print(df.duplicated().sum())

# إزالة المسافات الزائدة

df.columns = df.columns.str.strip()

df = df.applymap(lambda x: x.strip() if isinstance(x, str) else x)

# إزالة التكرارات

df = df.drop\_duplicates()

# تنضيف النصوص

str\_cols = df.select\_dtypes(include=['object']).columns

df[str\_cols] = df[str\_cols].apply(lambda x: x.str.strip())

# تحويل التواريخ

df['Date of Journey'] = pd.to\_datetime(df['Date of Journey'], errors='coerce')

df['Date of Purchase'] = pd.to\_datetime(df['Date of Purchase'], errors='coerce')

# معالجة القيم الفارغة

df.loc[(df['Journey Status'] == 'On Time') & (df['Reason for Delay'].isna()), 'Reason for Delay'] = 'No Delay'

df.loc[(df['Journey Status'] == 'Delayed') & (df['Reason for Delay'].isna()), 'Reason for Delay'] = 'Unknown'

df['Actual Arrival Time'] = df['Actual Arrival Time'].fillna('Cancelled')

df['Railcard'] = df['Railcard'].fillna('No Railcard')

# تعديل بعض القيم الموحدة

df['Reason for Delay'] = df['Reason for Delay'].replace({

    'Staffing': 'Staff Shortage',

    'Weather Conditions': 'Weather'

})

# توحيد كتابة 'Signal failure' إلى 'Signal Failure'

df['Reason for Delay'] = df['Reason for Delay'].replace('Signal failure', 'Signal Failure')

print(df.isnull().sum())

print(df.dtypes)

# حفظ نسخة من الداتا بعد التنظيف

df.to\_csv(r"D:\\railway 2\_cleaned.csv", index=False)

# --------------------------------------

# عمل وعرض وحفظ THE DASHBOARS

sns.set\_style("whitegrid")

plt.style.use('default')

# ألوان متدرجة

custom\_palette = ['#004c6d', '#2176b7', '#57a0d3', '#a9c9eb', '#dceefb']

# --------------------------------------

# داش بورد 1

fig1, axs1 = plt.subplots(2, 2, figsize=(18, 12))

axs1 = axs1.flatten()

#  Journey Status Distribution

sns.countplot(x='Journey Status', data=df, palette=custom\_palette, ax=axs1[0])

axs1[0].set\_title('Journey Status Distribution', fontsize=16, color='#004c6d')

axs1[0].set\_xlabel('Status')

axs1[0].set\_ylabel('Number of Journeys')

#  Reasons for Delay

delay\_reasons = df[df['Reason for Delay'] != 'No Delay']['Reason for Delay'].value\_counts()

sns.barplot(x=delay\_reasons.values, y=delay\_reasons.index, palette=custom\_palette, ax=axs1[1])

axs1[1].set\_title('Reasons for Delay', fontsize=16, color='#004c6d')

axs1[1].set\_xlabel('Number of Journeys')

axs1[1].set\_ylabel('Reason')

#  Top 5 Departure Stations

top\_departure = df['Departure Station'].value\_counts().head(5)

sns.barplot(x=top\_departure.values, y=top\_departure.index, palette=custom\_palette, ax=axs1[2])

axs1[2].set\_title('Top 5 Departure Stations', fontsize=16, color='#004c6d')

axs1[2].set\_xlabel('Number of Journeys')

axs1[2].set\_ylabel('Departure Station')

#  Top 5 Arrival Destinations

top\_arrival = df['Arrival Destination'].value\_counts().head(5)

sns.barplot(x=top\_arrival.values, y=top\_arrival.index, palette=custom\_palette, ax=axs1[3])

axs1[3].set\_title('Top 5 Arrival Destinations', fontsize=16, color='#004c6d')

axs1[3].set\_xlabel('Number of Journeys')

axs1[3].set\_ylabel('Arrival Destination')

plt.suptitle('Railway Dashboard - Part 1', fontsize=20, color='#004c6d', y=1.02)

plt.tight\_layout(rect=[0, 0, 1, 0.96])

fig1.savefig('railway\_dashboard\_part1.png', dpi=300, bbox\_inches='tight')

plt.show()

# --------------------------------------

# داش بورد 2

fig2, axs2 = plt.subplots(2, 2, figsize=(18, 12))

axs2 = axs2.flatten()

#  Payment Method Distribution

payment\_methods = df['Payment Method'].value\_counts()

axs2[0].pie(payment\_methods.values, labels=payment\_methods.index, autopct='%1.1f%%', startangle=140,

            colors=custom\_palette, textprops={'color':"black"})

axs2[0].set\_title('Payment Method Distribution', fontsize=16, color='#004c6d')

axs2[0].axis('equal')

#  Railcard Usage

railcard\_usage = df['Railcard'].value\_counts()

sns.barplot(x=railcard\_usage.values, y=railcard\_usage.index, palette=custom\_palette, ax=axs2[1])

axs2[1].set\_title('Railcard Usage', fontsize=16, color='#004c6d')

axs2[1].set\_xlabel('Number of Journeys')

axs2[1].set\_ylabel('Railcard Type')

#  Tickets Sold Over Time

sales\_over\_time = df['Date of Purchase'].value\_counts().sort\_index()

axs2[2].plot(sales\_over\_time.index, sales\_over\_time.values, marker='o', color='#00BFFF')

axs2[2].set\_title('Tickets Sold Over Time', fontsize=16, color='#004c6d')

axs2[2].set\_xlabel('Date')

axs2[2].set\_ylabel('Number of Tickets')

axs2[2].tick\_params(axis='x', rotation=45)

#  Top Departure Stations with Delays

delayed\_stations = df[df['Journey Status'] == 'Delayed']['Departure Station'].value\_counts().head(5)

sns.barplot(x=delayed\_stations.values, y=delayed\_stations.index, palette=custom\_palette, ax=axs2[3])

axs2[3].set\_title('Top Departure Stations with Delays', fontsize=16, color='#004c6d')

axs2[3].set\_xlabel('Number of Delayed Journeys')

axs2[3].set\_ylabel('Departure Station')

plt.suptitle('Railway Dashboard - Part 2', fontsize=20, color='#004c6d', y=1.02)

plt.tight\_layout(rect=[0, 0, 1, 0.96])

fig2.savefig('railway\_dashboard\_part2.png', dpi=300, bbox\_inches='tight')

plt.show()

# --------------------------------------

# داش بورد 3

# ضبط ستايل العرض

sns.set\_style("whitegrid")

plt.style.use('default')

# إنشاء الشكل

fig3, axs3 = plt.subplots(2, 2, figsize=(18, 12))

axs3 = axs3.flatten()

#  تحليل نسبة التأخير حسب يوم الأسبوع

df['Day\_of\_Week'] = df['Date of Journey'].dt.day\_name()

delay\_by\_day = df.groupby('Day\_of\_Week')['Journey Status'].apply(lambda x: (x == 'Delayed').mean() \* 100)

delay\_by\_day = delay\_by\_day.reindex([

    'Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'

])  # ترتيب الأيام بشكل صحيح

delay\_by\_day.plot(kind='barh', color='#2176b7', ax=axs3[0])

axs3[0].set\_title('Delay Percentage by Day of the Week', fontsize=16, color='#004c6d')

axs3[0].set\_xlabel('Delay Percentage (%)')

axs3[0].set\_ylabel('Day')

axs3[0].grid(axis='x')

#  توزيع أسعار التذاكر

sns.histplot(df['Price'], bins=30, kde=True, color='#004c6d', ax=axs3[1])

axs3[1].set\_title('Ticket Price Distribution', fontsize=16, color='#004c6d')

axs3[1].set\_xlabel('Price')

axs3[1].set\_ylabel('Number of Tickets')

axs3[1].grid(True)

#  تحليل وقت المغادرة مقابل نسبة التأخير

df['Departure\_Hour'] = pd.to\_datetime(df['Departure Time'], errors='coerce').dt.hour

delay\_by\_hour = df.groupby('Departure\_Hour')['Journey Status'].apply(lambda x: (x == 'Delayed').mean() \* 100)

delay\_by\_hour.plot(kind='line', marker='o', color='#00BFFF', ax=axs3[2])

axs3[2].set\_title('Delay Percentage by Departure Hour', fontsize=16, color='#004c6d')

axs3[2].set\_xlabel('Hour of Departure')

axs3[2].set\_ylabel('Delay Percentage (%)')

axs3[2].grid(True)

axs3[2].set\_xticks(range(0, 24))

#  تحليل العلاقة بين سعر التذكرة وحالة الرحلة

sns.boxplot(x='Journey Status', y='Price', data=df, palette='Blues', ax=axs3[3])

axs3[3].set\_title('Ticket Price vs Journey Status', fontsize=16, color='#004c6d')

axs3[3].set\_xlabel('Journey Status')

axs3[3].set\_ylabel('Ticket Price')

axs3[3].grid(True)

# تنسيق الشكل العام

plt.suptitle('Railway Dashboard - Part 3', fontsize=20, color='#004c6d', y=1.02)

plt.tight\_layout(rect=[0, 0, 1, 0.96])

# حفظ الصورة

fig3.savefig('railway\_dashboard\_part3.png', dpi=300, bbox\_inches='tight')

plt.show()