Part_2: Initial Analysis

Farhana Islam

1. As the performance metrics we are working with are based on duration you will need to make sure all the fields containing datetime data are identified and appropriately formatted.   (2 Mark)

Answer: Four columns contain date and time data which are shown below**:**

OR Booking Req DT/Tm (Operating Room Booking Request Date/Time): This field indicates the date and time when the operating room booking was requested.

Proc DT (Procedure Date): This field represents the date when the procedure was performed.

Pt OR Chk In (Patient Operating Room Check-in): This field indicates the time when the patient checked into the operating room.

Pt In OR (Patient In Operating Room): This field indicates the time when the patient entered the operating room

```python
import pandas as pd
from datetime import datetime
df = pd.read_csv("ORBooking.csv")

# Convert the "OR Booking Req DT/Tm" column to datetime format
df["OR Booking Req DT/Tm"] = pd.to_datetime(df["OR Booking Req DT/Tm"])

# Convert "Proc DT" column to datetime format
df["Proc DT"] = pd.to_datetime(df["Proc DT"])

# Convert time columns to strings with a consistent format
df["Pt OR Chk In"] = pd.to_datetime(df["Pt OR Chk In"], format='%H%M',
errors='coerce').dt.strftime('%H:%M:%S')
df["Pt In OR"] = pd.to_datetime(df["Pt In OR"], format='%H%M',
errors='coerce').dt.strftime('%H:%M:%S')

# Convert time columns to time objects
df['Pt OR Chk In'] = pd.to_datetime(df['Pt OR Chk In']).dt.time
df['Pt In OR'] = pd.to_datetime(df['Pt In OR']).dt.time

# Create new columns with datetime format
df['Pt OR Chk In_DATETIME'] = df['Proc DT'] + pd.to_timedelta(df['Pt OR Chk In'].astype(str))
df['Pt In OR_DATETIME'] = df['Proc DT'] + pd.to_timedelta(df['Pt In OR'].astype(str))

df.dtypes
```

3. The PT priority column reflects the number of hours before they should be operated on based on their condition. (for example, E-06H means the should be operated on with 6 hours after their

diagnosis and subsequent booking). Create a new column with a date time for when each patient should have their procedure started by based on the PT priority column compared to the OR Booking Req DT/Tm. (2 Marks)

```python
# Convert OR Booking Req DT/Tm to datetime format
df["OR Booking Req DT/Tm"] = pd.to_datetime(df["OR Booking Req DT/Tm"])

# Extract the numeric part from the PT Priority column
df["PT Priority Hours"] = df["Pt Priority"].str.extract(r'(\d+)').astype(int)

# Calculate the expected procedure start datetime
df["Expected Procedure Start DT"] = df["OR Booking Req DT/Tm"] - pd.to_timedelta(df["PT Priority Hours"], unit='h')

# Display the DataFrame with the new column
df.head()
```

4. In order to start looking at the amount of time patients are waiting before surgery you will need to compare the OR Booking Req DT/Tm to the actual time of procedure. The OR is a 24 hour operation, you may need to account for this in the below questions when creating new fields.

a. You will need to make a choice between the Pt OR Chk In or the Pt In OR time columns as the time to pair with the date for surgery for this new column. Explain how you will make this choice. (2 Mark)

Answer:

To decide between the "Pt OR Chk In" and "Pt In OR" time columns as the time to pair with the date for surgery in the new column, we need to consider the significance of each event and how it relates to the actual start time of the procedure. Here's how we can make this choice:

Pt OR Chk In: This represents the time when the patient checks into the operating room.

Pt In OR: This represents the time when the patient is in the operating room and the procedure is about to start.

Consideration:

Pt OR Chk In: While this time indicates when the patient officially enters the operating room, it doesn't necessarily represent the exact start time of the procedure. Delays can occur between check-in and the actual start of the procedure due to various reasons such as preparation, setup, and coordination.

Pt In OR: This time represents when the patient is ready to undergo the procedure and is typically closer to the actual start time of the surgery.

Decision:

Considering that we are interested in analyzing the amount of time patients are waiting before surgery, it's more appropriate to use the "Pt In OR" time column. This time better reflects the readiness of the patient for the procedure and is likely closer to the actual start time of the surgery.

Justification:

By using the "Pt In OR" time column, we capture a more accurate representation of the time patients spend waiting before surgery, as it reflects the moment when they are prepared and ready for the procedure to begin. This choice aligns better with the objective of analyzing patient wait times before surgery.

Therefore, for the new column comparing the OR Booking Req DT/Tm to the actual time of the procedure, we will use the "Pt In OR" time column.

b. Create a new column combining the procedure date and either of the Pt OR Check In or the Pt In OR time to a datetime format. (2 Marks)

Ans: b

```python
import pandas as pd

# Load the dataset
df = pd.read_csv("ORBooking.csv")

# Convert "Proc DT" column to datetime format
df["Proc DT"] = pd.to_datetime(df["Proc DT"])

# Convert time columns to strings with a consistent format
df["Pt In OR"] = pd.to_datetime(df["Pt In OR"], format='%H%M',
errors='coerce').dt.strftime('%H:%M:%S')

# Convert the "Pt In OR" column to datetime format
df['Pt In OR_DATETIME'] = df['Proc DT'].dt.strftime('%Y-%m-%d') + ' ' + df['Pt In OR']

# Convert the "Pt In OR_DATETIME" column to datetime format
df['Pt In OR_DATETIME'] = pd.to_datetime(df['Pt In OR_DATETIME'])

# Display the DataFrame with the new column
df.head()
```

5. You will now start to analyze the performance for getting the patients completed in their specified time.

a. Calculate the time difference between the OR Booking Req DT/Tm and your new column with the procedure date & time. (1 Mark)

```python
# Calculate the time difference
df['Time Difference'] = df['OR Booking Req DT/Tm'] - df['Pt In OR_DATETIME']
```

b. Determine whether each procedure was started on time and what the overall rate at which procedures were started on time (1 Mark)

```
# Determine if each procedure was started on time
df['Started On Time'] = df['Time Difference'] <= pd.Timedelta(0)

# Calculate the overall rate of procedures started on time
overall_rate_on_time = df['Started On Time'].mean()

# Display the overall rate
print("Overall rate of procedures started on time:", overall_rate_on_time)
```

c. Measure the difference in time between the time the procedure was due and when it was started. (1 Mark)

```
print("Difference in time between the procedure due time and actual start time:")
df['Time Difference']
```

6. Using other techniques to aggregate the data, what insights can you gain (2-3 points max). (4 Marks)

Using alternative techniques to aggregate data can unveil additional insights into the operational performance and underlying trends within the dataset. Here are few potential insights:

- Aggregate Time Difference by Procedure Type:

Group the data by procedure type and calculate summary statistics (e.g., mean, median, standard deviation) of the time difference between the OR Booking Req DT/Tm and the procedure start time. This analysis can reveal if certain types of procedures tend to experience consistently longer or shorter wait times, indicating potential areas for improvement in scheduling or resource allocation.

- Time Series Analysis of Procedure Start Times:

Analyze the distribution of procedure start times over different time intervals (e.g., hourly, or daily) to identify patterns or trends in the scheduling of surgeries throughout the day or week. This analysis can help optimize staffing levels and resource utilization based on demand patterns.

- Percentage of Procedures Started on Time Over Time:

Calculate the percentage of procedures started on time for each time (e.g., hourly, or daily) to track performance trends over time. This analysis can highlight changes in operational efficiency or identify periods of high or low performance, allowing for targeted interventions or adjustments to improve overall timeliness.

These insights can provide valuable information for healthcare administrators and practitioners to optimize scheduling processes, allocate resources effectively, and ensure timely delivery of care to patients.