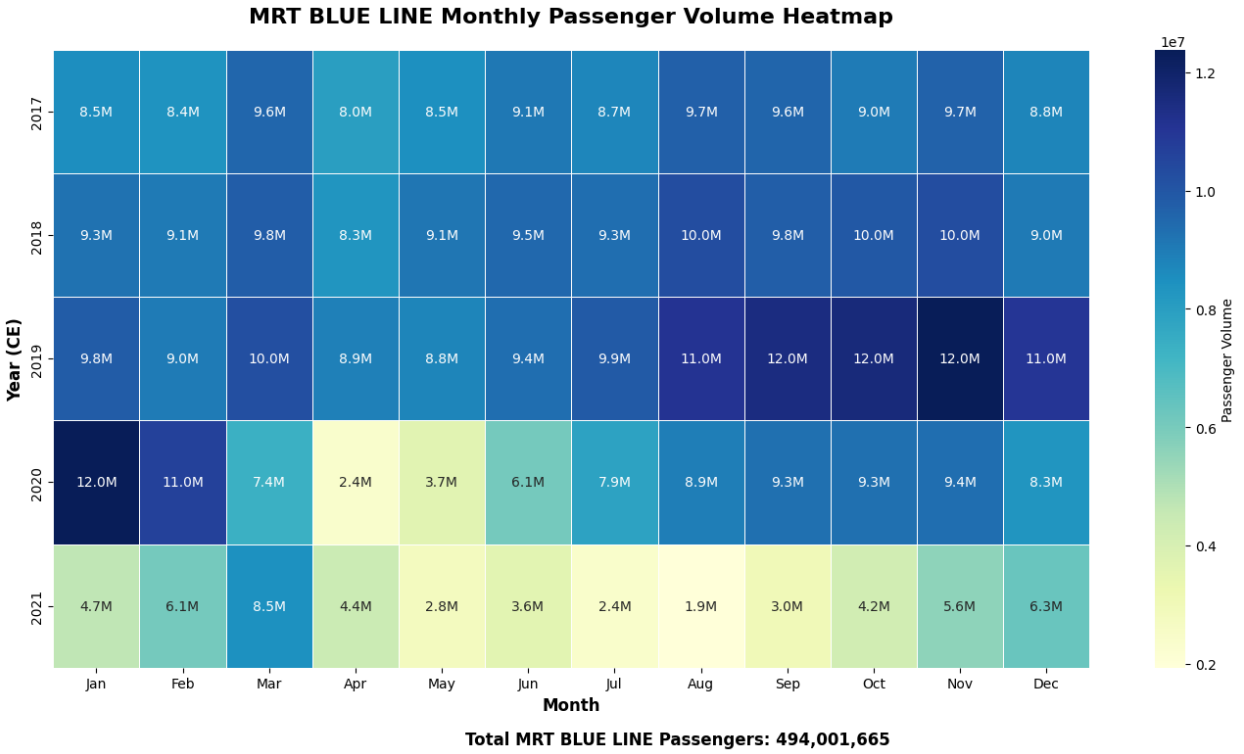


วิธีการคร่าวๆที่สำคัญในการสร้างกราฟ

1. Data Loading and Cleaning

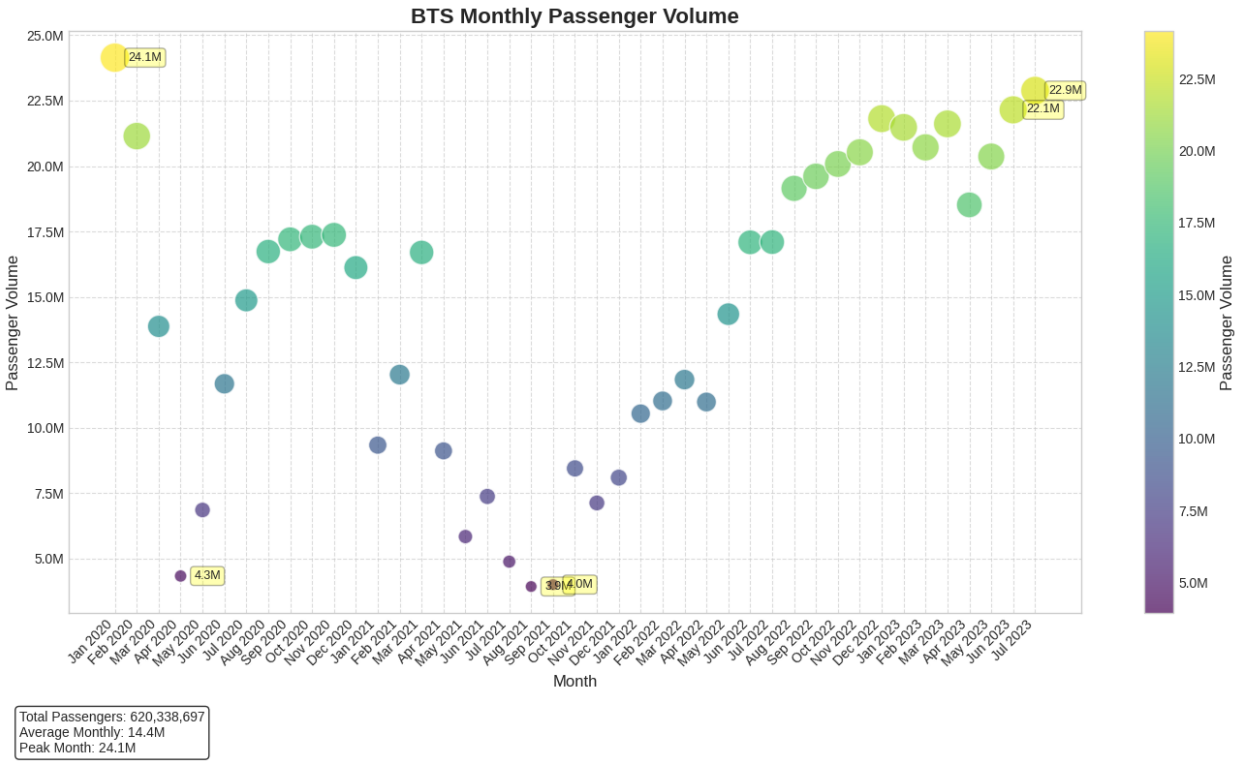
- ในคอลัมน์ ‘Passenger’ เป็นตัวเลขที่มีลูกน้ำจึงต้องลบออกแล้วเปลี่ยนtypeเป็นตัวเลข
- เปลี่ยนปี พศ เป็น คศ ด้วยการลบ 543
- สร้างคอลัมน์อีกอันนึงเพื่อmapเลขเดือนกับเดือน เพื่อเวลาเรียงข้อมูลจะได้สวยและอ่านเข้าใจง่าย

2. Data Preparation

- แยกdataframeของ Blue line และ Purple lineอ อกจากกัน
- mappingระหว่างเลขกำกับเดือนกับ ตัวย่อของเดือน เช่น 1 = Jan

3. Heatmap Creation Function

- สร้างตารางสรุปข้อมูลโดยใช้ปีเป็นแถวและเดือนเป็นคอลัมน์
- ปรับคำอธิบายประกอบกราฟโดยแสดงจำนวนผู้โดยสารเป็นล้าน (M) หรือเป็นพัน (K)



วิธีการสร้างกราฟนี้ (เน้นเฉพาะจุดที่สำคัญเท่านั้น)

1. มีการใช้ข้อมูลหลายdimension เช่น เดือน จำนวนผู้โดยสาร หาความหนาแน่นแบ่งตามสี

```
# Create scatter plot with sized points based on volume
scatter = plt.scatter(BTS['m_month'],
                      BTS['t_total'],
                      s=BTS['t_total']/50000,  # Size points based on passenger volume
                      alpha=0.7,
                      c=BTS['t_total'],  # Color based on passenger volume
                      cmap='viridis',
                      edgecolor='white',
                      linewidth=1)
```

ส่วนนี้จะสร้างแผนภาพscatterที่ช่วยให้สามารถรับรู้รูปแบบต่างๆ ผ่านกราฟได้ทั้งความเข้มข้นแบ่งตามสี ขนาดของจุด และ เทรนหรือแนวโน้มตามเดือน

2. การcustomรูปแบบตัวเลขหลักล้าน

```
# Format colorbar labels to show millions
def millions(x, pos):
    return f'{x/1000000:.1f}M'
cbar.ax.yaxis.set_major_formatter(FuncFormatter(millions))
```

ฟังก์ชันนี้ช่วยจัดรูปแบบตัวเลขโดยจะแปลงตัวเลขขนาดใหญ่ที่เป็นจำนวนมากๆแบบล้านๆ ให้เป็นรูปแบบที่อ่านง่ายขึ้นโดยแสดงค่าเป็นล้านพร้อมคำต่อท้าย "M" ทำให้สามารถตีความได้ง่ายขึ้นมาก

3. การชี้จุดข้อมูลและใส่คำอธิบายเฉพาะข้อมูลที่สำคัญที่ต้องการแสดง

```
# Add data point labels for key points
# Label top 3 and bottom 3 points
top_indices = BTS['t_total'].nlargest(3).index
bottom_indices = BTS['t_total'].nsmallest(3).index

for idx in list(top_indices) + list(bottom_indices):
    plt.annotate(f"{BTS.loc[idx, 't_total']/1000000:.1f}M",
                 xy=(BTS.loc[idx, 'm_month'], BTS.loc[idx, 't_total']),
                 xytext=(10, 0),
                 textcoords="offset points",
                 ha='left',
                 va='center',
                 fontsize=9,
                 bbox=dict(boxstyle='round,pad=0.3', fc='yellow', alpha=0.3))
```

เลือกใส่คำอธิบายจุดเฉพาะจุดที่จำนวนผู้โดยมากสุดและน้อยสุดอย่างละ 3ลำดับ โดยมันไม่ได้รับค่ามาอย่างเดียวแต่มันรับค่าตำแหน่งในdataframeที่ช่วยให้เรารู้พิกัดx yด้วย

4. การ handling data typeที่dynamic

```
# Format x-axis for better readability
if pd.api.types.is_datetime64_any_dtype(BTS['m_month']):
    ax.xaxis.set_major_formatter(mdates.DateFormatter('%b %Y'))
    plt.xticks(rotation=45, ha='right')
else:
    plt.xticks(rotation=45, ha='right')
```

ฟังก์ชั่น pd.api.types.is_datetime64_any_dtype() คือฟังก์ชันของpandasที่ช่วยเช็คว่าคอลัมน์นั้นๆมีข้อมูลที่อยู่ในรูป datetime หรือไม่ หลังจากนั้นเราก็ใส่condiftion if elseเข้าไป ถ้าเจอdatetime จะให้เปลี่ยน format เป็น mdates.DateFormatter('%b %Y') แสดงออกมาเป็น “Jan 2020” เป็นต้น