Presentation 1

• Links to presentation(s) and code(s) on GitHub:

- Presentation Link
    - Note that if the Power BI embedded in the slides fails, static visuals can be found at the bottom of the slide deck
- All coding was done in Power BI: see below // Power BI Sharing Link

• What did you do?

I created visuals for the Time Trends Dashboard. I began by compiling all CAR data and All Calls data into two CSV files using the import code provided by Prof. Krishna. I then loaded both files into Power BI for analysis. I then performed the following transformations *within Power BI***.**

CAR Data:

I transformed the data using the *Group By* function to ensure each *Contact Session ID* corresponded to only one row—preventing double counting caused by multiple actions per session. I used the *Advanced Group By* option to create multiple aggregations that support deeper analysis and visualization. Specifically, I created four new columns:

- Count — *Operation:* Count Rows

  Purpose: Determines the number of actions per session.
- Call Start Time — *Operation:* Minimum of *Activity Start Timestamp*

  Purpose: Shows timestamp of the first row in a session to identify when the call began.
- Starting Hour — *Operation:* Minimum of *Hour*

  Purpose: Records the hour of day when the session started.
- All Rows — *Operation:* All Rows

  Purpose: Retains information from each session in nested tables for detailed reference.
- I also chose to “Expand All Rows” in order to access nested information in the visuals

I also created a column from call start time for weekday (in numbers) where Monday = 1 and Sunday = 7 using the following code:

```
DayOfWeekNum = WEEKDAY('combined_calls'[Call Start Time], 2)
```

All Calls Data:

I created a new column called DateTime_CST which converts the "Answer time" to CST using the following code:

```
DateTime_CST =
VAR DateGMT = 'combined_all_calls'[Answer time]
VAR YearNum = YEAR(DateGMT)
VAR DSTStart =
    DATE(YearNum, 3, 10 - WEEKDAY(DATE(YearNum, 3, 8))) + TIME(2, 0, 0)
VAR DSTEnd =
    DATE(YearNum, 11, 3 - WEEKDAY(DATE(YearNum, 11, 1))) + TIME(2, 0, 0)
VAR OffsetHours =
    IF(DateGMT >= DSTStart && DateGMT < DSTEnd, -5, -6)
RETURN
    DateGMT + ( OffsetHours / 24 )
```

I then created a column including just the hour of day using the following code:

```
HourOfDay_CST = HOUR('combined_all_calls'[DateTime_CST])
```

I also created a column for Weekday (in numbers) where Monday = 1 and Sunday = 7

```
DayOfWeekNum_CST = WEEKDAY('combined_all_calls'[DateTime_CST], 2)
```

After these transformations, I created the following visualizations:

- Data: CAR
  Type: Bar Chart
  X-axis:Starting Hour
  Y-axis: Contact Session ID
  Legend: DayofWeekNum
- Data: CAR
  Type: Line Graph
  X-axis: Starting Hour
  Y-axis: Average of Count for Contact Session ID (# of rows for a call)
- Data: All Calls

Type: Line Graph
X-axis: Hour of Day (CST)
Y-axis: Call Duration

*On each visual, you can filter by: EP Name, Flow Name, Termination Reason, and the axes and legend for the visual. You can see which filters are on while viewing the graphics.

• How does it help the project?

This analysis of peak call time trends with various filters can help the team uncover operational inefficiencies and develop strategies to improve overall intake performance, as seen in my presentation. Based on my analysis, changing the messages for times users are encouraged to call could increase the quality of service for users and allow agents to reach more people overall.

• Issues faced, Attempts to resolve issues, and Issues resolved

I did not face any significant issues with this contribution.

• Next steps

Going forward, I plan to:

- *Refine the dashboard visuals* to clearly display trends in peak call times by current filters.
    - Use heatmaps or line charts to highlight patterns across hours and days.
- *Incorporate additional filters* to enable more granular insights into when and why certain termination reasons occur.
- *Collaborate with team members* who are focusing on other metrics so that the time trend insights can be connected with them.

• References (Mention if you built up on someone else's work)

Code import files from Prof. Krishna