1. Changed column types manually to avoid errors:
    - Date >> Date
    - Ridership >> whole number
    - Percentage columns >> Decimal number

2. Converting percentage values from 0-100 scale to 0-1 scale:

M Code

```
// Converting percentage columns to numbers first
    #"Changed Type" = Table.TransformColumns(#"Promoted Headers",
        {
            {"Subways: % of Comparable Pre-Pandemic Day", each Number.FromText(_), type number},
            {"Buses: % of Comparable Pre-Pandemic Day", each Number.FromText(_), type number},
            {"LIRR: % of Comparable Pre-Pandemic Day", each Number.FromText(_), type number},
            {"Metro-North: % of Comparable Pre-Pandemic Day", each Number.FromText(_), type number},
            {"Access-A-Ride: % of Comparable Pre-Pandemic Day", each Number.FromText(_), type
number},
            {"Bridges and Tunnels: % of Comparable Pre-Pandemic Day", each Number.FromText(_), type
number},
            {"Staten Island Railway: % of Comparable Pre-Pandemic Day", each Number.FromText(_),
type number}
        }
    ),

    // Now dividing by 100
    #"Converted Percentages" = Table.TransformColumns(#"Changed Type",
        {
            {"Subways: % of Comparable Pre-Pandemic Day" , each _ / 100, type number},
            {"Buses: % of Comparable Pre-Pandemic Day", each _ / 100, type number},
            {"LIRR: % of Comparable Pre-Pandemic Day", each _ / 100, type number},
            {"Metro-North: % of Comparable Pre-Pandemic Day", each _ / 100, type number},
            {"Access-A-Ride: % of Comparable Pre-Pandemic Day", each _ / 100, type number},
            {"Bridges and Tunnels: % of Comparable Pre-Pandemic Day", each _ / 100, type number},
            {"Staten Island Railway: % of Comparable Pre-Pandemic Day", each _ / 100, type number}
})
```

3. Ensuring the Date column is correctly formatted as Date type

M Code

```
#"Changed Type1" = Table.TransformColumnTypes(#"Converted Percentages",{{"Date", type date}}),
```

4. Calculating total ridership for each day by summing all transport modes:

M Code

```
#"Added Total Ridership Column" = Table.AddColumn(#"Changed Type5", "Total Ridership", each

    [#"Subways: Total Estimated Ridership"] +

    [#"Buses: Total Estimated Ridership"] +

    [#"LIRR: Total Estimated Ridership"] +

    [#"Metro-North: Total Estimated Ridership"] +

    [#"Bridges and Tunnels: Total Traffic"]+

    [#"Access-A-Ride: Total Scheduled Trips"] +

    [#"Staten Island Railway: Total Estimated Ridership"], type number)
```

5. **Calculating a weighted recovery percentage considering ridership proportions:**

   The weighted recovery percentage calculation in Power Query M assesses the overall recovery of a transportation system by considering the proportionate contribution of ridership from different modes. It calculates a weighted sum of recovery percentages for each mode, adjusted by their respective ridership volumes, and divides this sum by the total ridership across all modes. This metric provides a holistic view of the system's recovery status, ensuring that no single mode disproportionately influences the overall recovery percentage.

   Final Calculation: Weighted Recovery % = WeightedSum / TotalWeight

M Code

```
#"Added Weighted Recovery %" = Table.AddColumn(#"Added Ridership Category", "Weighted Recovery %",
each

   let

     TotalWeight =

        [#"Subways: Total Estimated Ridership"] +

        [#"Buses: Total Estimated Ridership"] +

        [#"LIRR: Total Estimated Ridership"] +

        [#"Metro-North: Total Estimated Ridership"] +

        [#"Access-A-Ride: Total Scheduled Trips"] +

        [#"Staten Island Railway: Total Estimated Ridership"],


      WeightedSum =

        ([#"Subways: % of Comparable Pre-Pandemic Day"] * [#"Subways: Total Estimated Ridership"] +

        [#"Buses: % of Comparable Pre-Pandemic Day"] * [#"Buses: Total Estimated Ridership"] +

        [#"LIRR: % of Comparable Pre-Pandemic Day"] * [#"LIRR: Total Estimated Ridership"] +

        [#"Metro-North: % of Comparable Pre-Pandemic Day"] * [#"Metro-North: Total Estimated
        Ridership"] +

        [#"Access-A-Ride: % of Comparable Pre-Pandemic Day"] * [#"Access-A-Ride: Total Scheduled
        Trips"] +

        [#"Staten Island Railway: % of Comparable Pre-Pandemic Day"] * [#"Staten Island Railway:
        Total Estimated Ridership"])

in

    if TotalWeight = 0 then null else WeightedSum / TotalWeight, type number),
```

6. Categorizing ridership levels based on thresholds:

M Code

```
#"Added Ridership Category" = Table.AddColumn(#"Added Total Ridership Column", "Ridership Category",
each

    if [Total Ridership] > 5000000 then "Very High"

    else if [Total Ridership] > 3000000 then "High"

    else if [Total Ridership] > 1000000 then "Moderate"

    else "Low", type text),
```

7. Classifying recovery levels based on weighted recovery percentage:

M Code

```
#"Added Weighted Recovery Classification" = Table.AddColumn( #"Added Weighted     Recovery %",
"Recovery Classification", each

    if [#"Weighted Recovery %"] >= 1 then "Fully Recovered"

    else if [#"Weighted Recovery %"] >= 0.8 then "High Recovery"

    else if [#"Weighted Recovery %"] >= 0.5 then "Medium Recovery"

    else "Low Recovery", type text)
```

Callender Table

The CalendarTable generates a structured date table that helps in time-based analysis. It includes additional attributes for each date to enable easier grouping and filtering in reports.

- `CALENDAR ( MIN('MTA_Daily_Ridership'[Date]), MAX('MTA_Daily_Ridership'[Date]))`

  Creates a continuous date range from the **earliest** to the **latest** date in the MTA_Daily_Ridership dataset.

- **Adding Date Attributes:**
  - **Year**: Extracts the year from each date.
  - **Month**: Extracts the numerical month (1–12).
  - **Month Name**: Converts the month number into its full name (e.g., January).
  - **Quarter**: Groups months into their respective quarters (Q1–Q4).
  - **Season**: Assigns a season based on the month.
  - **Weekday**: Returns the full weekday name (e.g., Monday).
  - **Weekday Number**: Numbers weekdays from Monday (1) to Sunday (7).
  - **Week of Year**: Determines the week number within the year, following ISO 8601 standards.
  - **Week of Month**: Calculates which week of the month a given date falls into.
  - **Day Type**: Classifies the date as either a "Weekday" or "Weekend" based on the day of the week.

```
CalendarTable =
ADDCOLUMNS (
    CALENDAR ( MIN('MTA_Daily_Ridership'[Date] ), MAX('MTA_Daily_Ridership'[Date] )),
    "Year", YEAR([Date]),
    "Month", MONTH([Date]),
    "Month Name", FORMAT([Date], "MMMM"),
    "Quarter", "Q" & FORMAT(QUARTER([Date]), "0"),
    "Season",
        SWITCH ( TRUE(),
            MONTH([Date]) IN {12, 1, 2}, "Winter",
            MONTH([Date]) IN {3, 4, 5}, "Spring",
            MONTH([Date]) IN {6, 7, 8}, "Summer",
            "Fall"
        ),
    "Weekday", FORMAT([Date], "dddd"),
    "Weekday Number", WEEKDAY([Date], 2), -- Monday = 1, Sunday = 7
    "Week of Year", WEEKNUM([Date], 2),  -- ISO 8601 week numbering
    "Week of Month", INT((DAY([Date]) - 1) / 7) + 1,
    "Day Type",
        IF ( WEEKDAY([Date], 1) IN {1, 7}, "Weekend", "Weekday" ) -- Sunday (1) & Saturday (7) are weekends
)
```

Measures

Month-over-Month (MoM) Growth Calculation

This formula calculates the percentage change in total ridership compared to the previous month.

```
MoM Growth =
VAR CurrentMonthValue = SUM('MTA_Daily_Ridership'[Total Ridership])
VAR PreviousMonthValue =
    CALCULATE(
        SUM('MTA_Daily_Ridership'[Total Ridership]),
        DATEADD('CalendarTable'[Date], -1, MONTH)
    )
RETURN
    IF(
        NOT ISBLANK(PreviousMonthValue) && PreviousMonthValue <> 0,
        (CurrentMonthValue - PreviousMonthValue) / PreviousMonthValue,
        BLANK()
    )
```

VAR CurrentMonthValue

- Calculates the **total ridership** for the current month.

VAR PreviousMonthValue

- Retrieves the **total ridership** from the previous month using DATEADD(), which shifts the date one month backward.

RETURN

- If the previous month's value is not blank or zero, the formula computes the MoM growth as:

  (Current Month Value - Previous Month Value) / Previous Month Value

- If the previous month's value is blank or zero, it returns BLANK() to prevent errors or misleading results.

Year-over-Year (YoY) Growth Calculation

This formula measures the percentage change in total ridership compared to the same month in the previous year.

```
YoY Growth =
VAR CurrentYearValue = SUM('MTA_Daily_Ridership'[Total Ridership])
VAR PreviousYearValue =
    CALCULATE(
        SUM('MTA_Daily_Ridership'[Total Ridership]),
        DATEADD('CalendarTable'[Date], -1, YEAR)
    )
RETURN
    IF(
        NOT ISBLANK(PreviousYearValue) && PreviousYearValue <> 0,
        (CurrentYearValue - PreviousYearValue) / PreviousYearValue,
        BLANK()
    )
```

VAR CurrentYearValue

- Computes the **total ridership** for the current year.

VAR PreviousYearValue

- Retrieves the **total ridership** from the **same month in the previous year** using DATEADD() with a -1 YEAR shift.

RETURN

- If the previous year's value is not blank or zero, the formula calculates the YoY growth as:
  (Current Year Value - Previous Year Value) / Previous Year Value

- If the previous year's value is blank or zero, it returns BLANK() to avoid incorrect calculations.