

# Data Visualization with PowerBI

Mohamed Kassem

2021-04-27



# Contents

<b>1</b>	<b>Intorduction</b>	<b>5</b>
<b>2</b>	<b>Setting and Objectives</b>	<b>7</b>
2.1	Setting . . . . .	7
2.2	Objectives . . . . .	7
<b>3</b>	<b>Data</b>	<b>9</b>
<b>4</b>	<b>Dataset</b>	<b>11</b>
4.1	Step 1 . . . . .	12
4.2	Step 2 . . . . .	12
4.3	Step 3 . . . . .	13
4.4	Step 5 . . . . .	14
4.5	Step 6 . . . . .	14
4.6	Step 7 . . . . .	15
<b>5</b>	<b>DataModel</b>	<b>17</b>
5.1	Step 1 . . . . .	17
5.2	Step 2 . . . . .	18
5.3	Step 3 . . . . .	18
<b>6</b>	<b>Report page</b>	<b>21</b>
6.1	information . . . . .	21
6.2	Size . . . . .	21

6.3	Background . . . . .	21
6.4	Alignment . . . . .	21
6.5	Wallpaper . . . . .	21
<b>7</b>	<b>Visualization types</b>	<b>23</b>
7.1	Comparison . . . . .	23
7.2	Change over time . . . . .	23
7.3	Ranking . . . . .	23
7.4	Spatial . . . . .	23
7.5	Part-to-whole . . . . .	23
7.6	Distribution . . . . .	23
7.7	Correlation . . . . .	23
7.8	Single . . . . .	23
7.9	Filter & Slicer . . . . .	23
<b>8</b>	<b>PowerBI native visuals</b>	<b>25</b>
8.1	Comparison visuals . . . . .	25
8.2	Change over time visuals . . . . .	26
8.3	Combo charts . . . . .	26
<b>9</b>	<b>Utilizing PowerBI native visualizations</b>	<b>27</b>
9.1	NYC TLC Dashboard design: . . . . .	27
9.2	Building our dashboard: . . . . .	28
<b>10</b>	<b>Costum R and Python visuals</b>	<b>31</b>
<b>11</b>	<b>Visuals interactions - Slicers &amp; filters.</b>	<b>33</b>

# Chapter 1

## Intorduction

This course is intended for educational purposes & preparing Data analysts for Exam DA-100: Analyzing Data with Microsoft Power BI : **Creating reports**

- All the needed resources to follow along can be found here <https://github.com/Mkassem16/NycTaxiPBI>.
- Creating basic PowerBI reports knowledge is prerequisite for this course.

**By the end of this course Data Analysts should be able to:**

- + Customize Report pages.
- + Decide for appropriate visualizations type.
- + To utilize PowerBI native visualizations.
- + Format and configure PowerBI native visuals.
- + Configure conditional formatting.
- + Import their own custom R or Python Visuals.
- + Utilize Slicers and Filters.
- + Adjust visuals for accessibility.
- + Configure automatic page refresh.
- + Create paginated reports.



## Chapter 2

# Setting and Objectives

### 2.1 Setting

The New York City Taxi and Limousine Commission (TLC), created in 1971, is the agency responsible for licensing and regulating New York City's Medallion (Yellow) taxi cabs. Over 200,000 TLC licensees complete approximately 1,000,000 trips each day.

For the sake of educational purposes Data Analysts should think of an Imaginary situation where NYC TLC has asked them to prepare a PowerBI report for their c-suite executives and decision managers to better understand their monthly operations in general and track their daily and monthly KPIs.

### 2.2 Objectives

#### 2.2.1 General

- Creating an easy to read and understand descriptive dashboard.
- Quick one look dynamic status updates.
- Tracking daily and weekly KPIs.
- Understanding revenue breakdown.

#### 2.2.2 Requested

- Is there a specific day of week that NYC TLC should deploy more licensees?
- Is there a specific time of day that NYC TLC should deploy more licensees?

### 2.2.3 Extra

- Mining for relations between trip duration, distance and fare.



## Chapter 3

# Data

Data is publicly available at <https://www1.nyc.gov/site/tlc/about/tlc-trip-record-data.page>

Please Find Data catalog explaining columns and categories here: [https://github.com/Mkassem16/NycTaxiPBI/blob/main/data/data\\_dictionary\\_trip\\_records\\_yellow.pdf](https://github.com/Mkassem16/NycTaxiPBI/blob/main/data/data_dictionary_trip_records_yellow.pdf)

For the purpose of building a demonstration dashboard, I extracted a sample of 10,000 rows of 2020 data. This sample dataset is hosted publicly on Google Cloud storage and can be queried directly from PowerBi. [https://storage.googleapis.com/powerbi\\_datacamp/nyc\\_taxi\\_db.csv](https://storage.googleapis.com/powerbi_datacamp/nyc_taxi_db.csv)

```
glimpse(df)
```

```
## Rows: 10,018
## Columns: 17
## $ vendor_id           <dbl> 1, 1, 2, 1, 2, 1, 4, 2, 1, 2, 1, 2, 2, 2, 2, ...
## $ pickup_datetime     <dtm> 2021-03-27 12:47:16, 2021-06-10 19:02:02, 2021...
## $ dropoff_datetime    <dtm> 2021-03-27 13:39:54, 2021-06-10 19:31:53, 2021...
## $ passenger_count     <dbl> 1, 1, 1, 1, 2, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, ...
## $ trip_distance       <dbl> 2.70, 15.10, 7.92, 6.50, 6.44, 10.00, 7.24, 36....
## $ rate_code           <dbl> 1, 1, 1, 1, 1, 1, 1, 5, 1, 1, 1, 1, 1, 1, 1, ...
## $ store_and_fwd_flag  <chr> "N", "N", "N", "N", "N", "N", "N", "N", "N", "N...
## $ payment_type        <dbl> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, ...
## $ fare_amount         <dbl> 29.0, 42.0, 26.0, 22.5, 24.5, 29.0, 22.0, 83.5,...
## $ extra               <dbl> 0.0, 0.0, 0.5, 0.5, 0.5, 0.5, 0.5, 0.0, 0.0, 0....
## $ mta_tax             <dbl> 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.5, 0.0, 0.5, 0....
## $ tip_amount          <dbl> 5.95, 12.10, 5.46, 4.75, 3.87, 6.05, 4.66, 21.5...
## $ tolls_amount        <dbl> 0.00, 5.76, 0.00, 0.00, 0.00, 0.00, 0.00, 23.76...
## $ imp_surcharge       <dbl> 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0.3, 0....
```

```
## $ total_amount      <dbl> 35.75, 60.66, 32.76, 28.55, 29.67, 36.35, 27.96...
## $ pickup_location_id <dbl> 68, 138, 261, 262, 261, 100, 7, 132, 264, 170, ...
## $ dropoff_location_id <dbl> 162, 88, 41, 231, 162, 127, 53, 265, 264, 236, ...
```

## Chapter 4

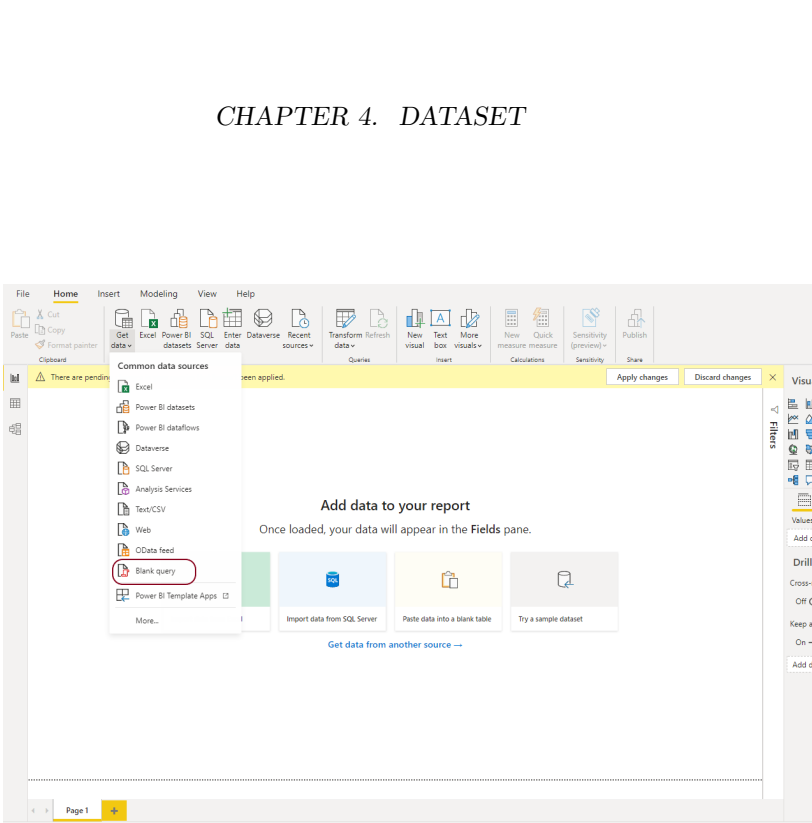
# Dataset

*Dataset* in PowerBI vocabulary is: the set of queries that the user will perform in order to import all of his data tables and all the transformations that will take place before data can be ready for the model and visualization.

**We will perform 2 queries to create 2 data tables:**

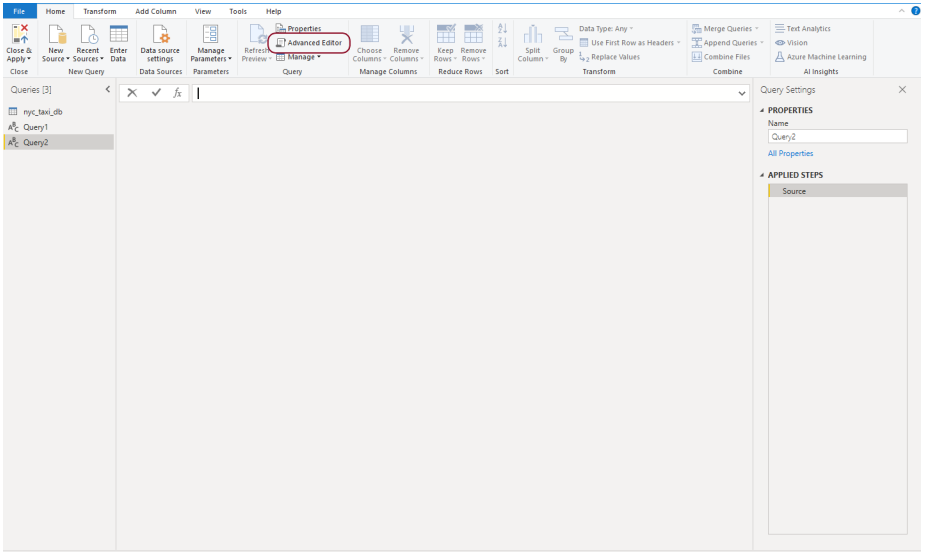
- Table 1 (**nyc\_taxi\_21**) : will include NYC TLC data. [https://github.com/Mkassem16/NycTaxiPBI/blob/main/queries/nyc\\_taxi\\_21.pq](https://github.com/Mkassem16/NycTaxiPBI/blob/main/queries/nyc_taxi_21.pq)
- Table 2 (**Calendar**) : Calendar table that will control the date dimension <https://github.com/Mkassem16/NycTaxiPBI/blob/main/queries/Calendar.pq>

4.1 Step 1



Getting data through Blank query

4.2 Step 2



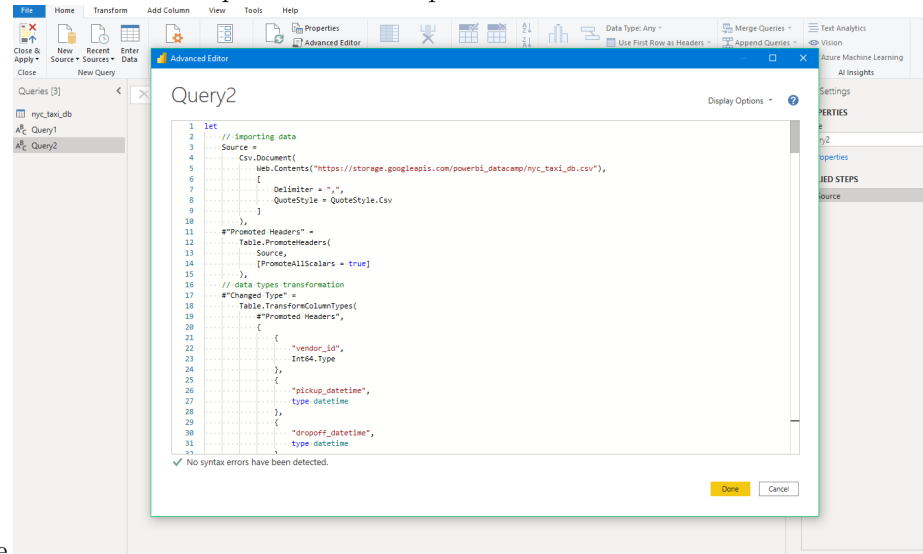
Choose Advanced editor

#### 4.3. STEP 3

13

### 4.3 Step 3

Get the query of Table 1 from the provided link and paste it here instead of what



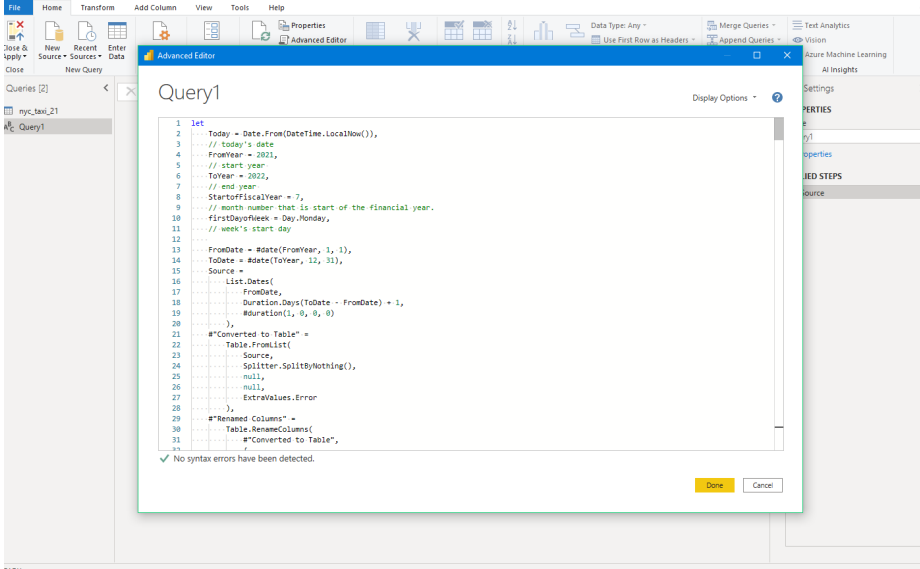
is already there  
## Step 4

1	vendor_id	pickup_datetime	dropoff_datetime	passenger_count	trip_distance	rate_code
2	1	2021-03-27T12:47:16	2021-03-27T13:39:54	1	2.7	1
3	1	2021-06-20T19:02:02	2021-06-20T19:31:53	1	15.1	1
4	2	2021-10-20T20:35:45	2021-10-20T21:05:05	1	7.92	1
5	1	2021-09-11T20:13:27	2021-09-11T20:36:56	1	6.5	1
6	2	2021-06-20T22:20:22	2021-06-20T22:45:24	2	6.44	1
7	1	2021-03-12T01:29:24	2021-03-12T01:46:51	1	10	1
8	4	2021-10-24T03:42:17	2021-10-24T03:00:51	1	7.24	1
9	2	2021-08-04T15:00:47	2021-08-04T16:08:47	1	36.97	5
10	1	2021-04-06T10:05:58	2021-04-06T10:50:18	1	9.6	1
11	2	2021-10-24T12:07:34	2021-10-24T12:46:35	1	3.51	1
12	1	2021-03-21T11:40:26	2021-03-21T11:57:36	1	8	1
13	2	2021-06-07T16:12:53	2021-06-07T17:07:23	1	9.62	1
14	2	2021-04-19T15:10:53	2021-04-19T16:33:06	1	19.22	1
15	2	2021-01-08T13:23:41	2021-01-08T13:45:41	2	6.73	1
16	2	2021-09-07T09:02:10	2021-09-07T09:38:43	1	10.08	1
17	2	2021-09-08T15:20:47	2021-09-08T15:46:43	2	10.29	1
18	1	2021-03-25T15:59:17	2021-03-25T16:38:21	1	2.5	1
19	1	2021-07-31T18:11:35	2021-07-31T18:41:44	1	8.4	1
20	1	2021-01-17T00:57:23	2021-01-17T02:24:43	1	4.4	1
21	1	2021-03-17T23:25:38	2021-03-17T23:50:23	3	8.1	1
22	1	2021-10-10T10:06:39	2021-10-10T10:49:39	2	14.3	1
23	1	2021-11-10T19:45:00	2021-11-10T20:14:49	1	5.3	1
24	1	2021-09-08T23:32:40	2021-09-08T23:28:54	1	12.8	1
25	1	2021-03-24T01:24:44	2021-03-24T01:55:22	1	10.3	1
26	1	2021-06-13T16:53:07	2021-06-13T17:26:20	2	4.4	1
27	2	2021-12-02T19:17:45	2021-12-02T19:32:58	2	7.53	1
28	2	2021-05-16T22:22:54	2021-05-16T23:07:59	1	5.32	1
29	2	2021-06-14T22:46:36	2021-06-14T23:07:59	1	6.17	1
30	2	2021-09-25T11:59:24	2021-09-25T12:55:08	2	11.14	1
31	2	2021-11-28T18:57:58	2021-11-28T18:58:01	6	0.07	5
32						

Make sure that query name is **nyc\_taxi\_21**

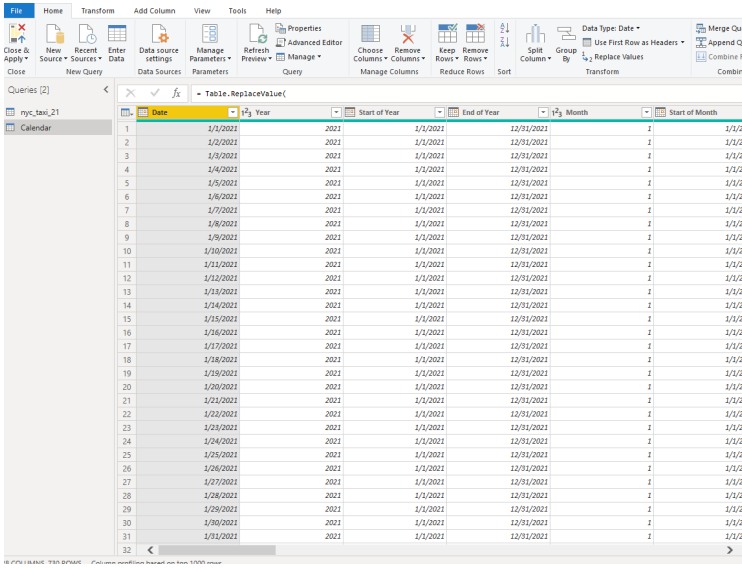
4.4 Step 5

Get the query of Table 2 from the provided link and paste it here instead of what



is already there

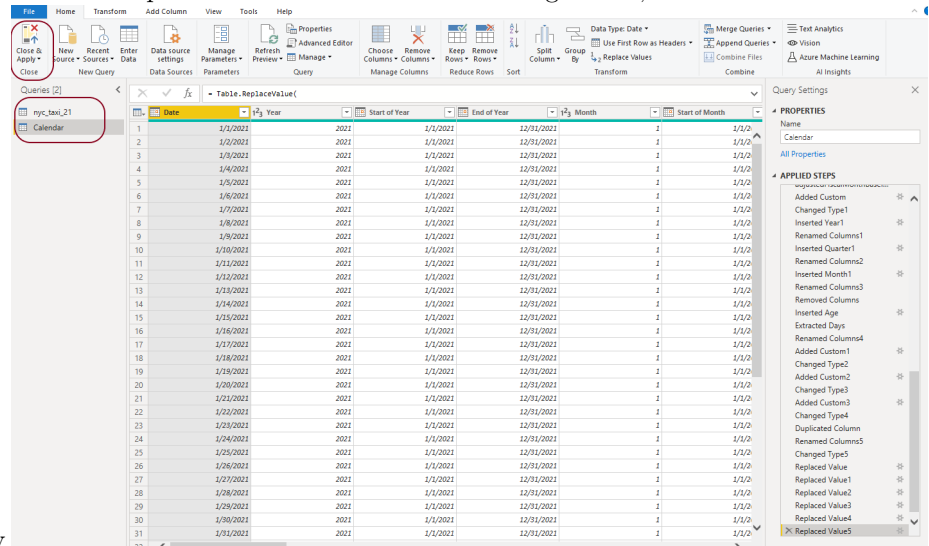
4.5 Step 6



Make sure that query name is **Calendar**

## 4.6 Step 7

Double check queries names as its important for Measures created through DAX,



Then click close & apply





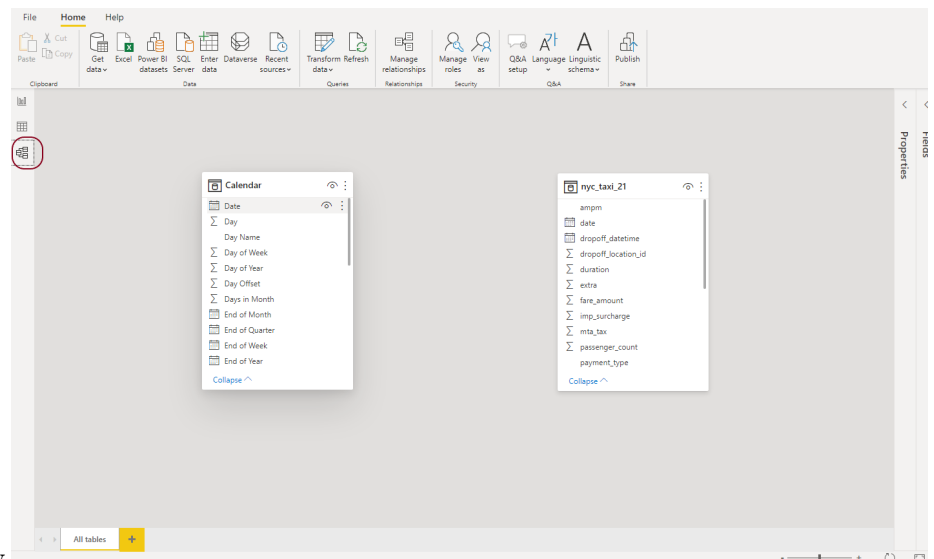
## Chapter 5

# DataModel

*Data Modeling* in PowerBI vocabulary is: the step of creating relations between tables before proceeding and building slicers and filters on our visuals we need to make sure that the relationships are with the right direction, cardinality and that they are active.

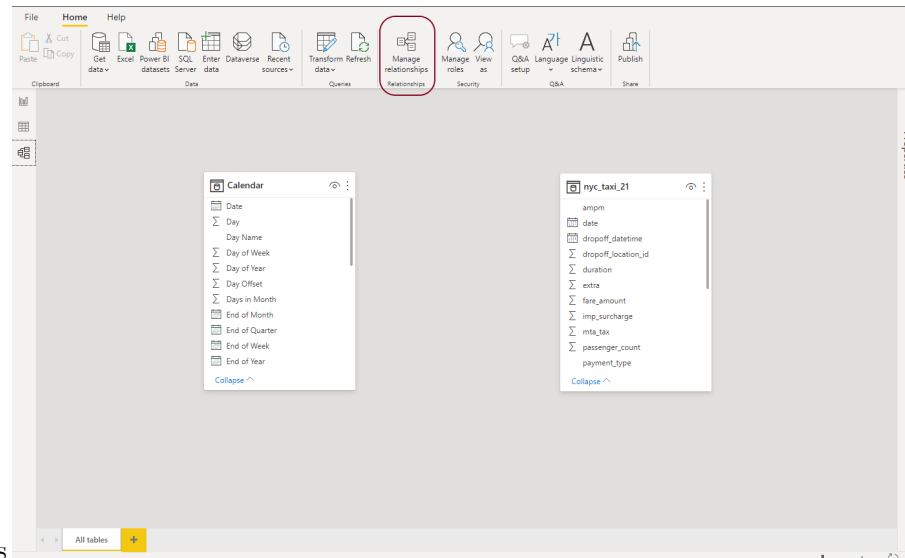
We will create 1 relation between Calendar table and the data table

### 5.1 Step 1



Navigate to data modeling view

## 5.2 Step 2



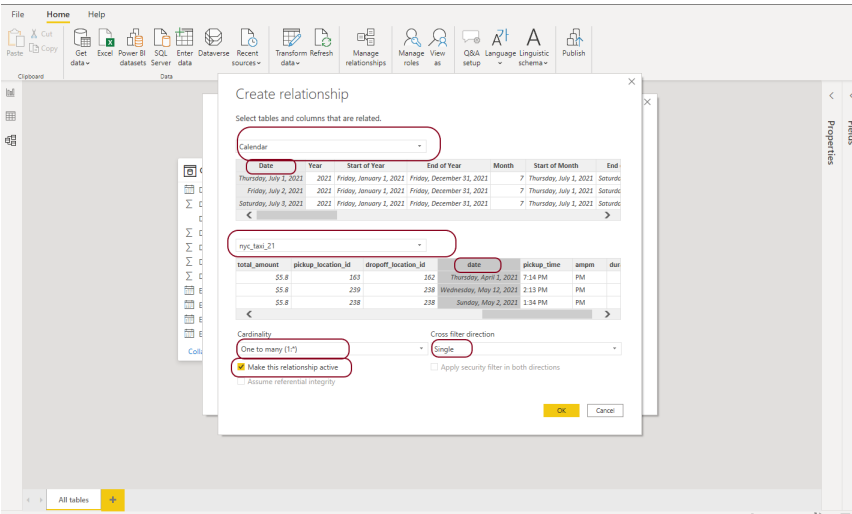
Click on manage relations

## 5.3 Step 3

Create new relation and make sure of the following:

- Select the right tables
- Select the right columns (by clicking on the column name in the display)
- Select the right Cardinality (usually with Calendar tables its One to many)
- Select the right direction (**Calendar** table to filter **nyc\_tax\_21** table)

5.3. STEP 3





## Chapter 6

# Report page

We can start customizing report page by clicking on the format icon of the Visualization pane without choosing any visual.

**Options include:**

**6.1 information**

**6.2 Size**

**6.3 Background**

**6.4 Alignment**

**6.5 Wallpaper**

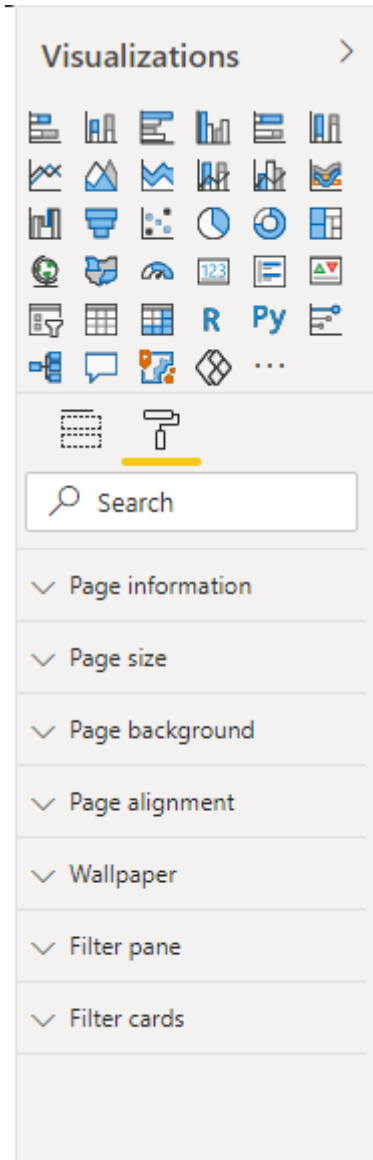


Figure 6.1: format

## Chapter 7

# Visualization types

7.1 Comparison

7.2 Change over time

7.3 Ranking

7.4 Spatial

7.5 Part-to-whole

7.6 Distribution

7.7 Correlation

7.8 Single

7.9 Filter & Slicer





## Chapter 8

# PowerBI native visuals

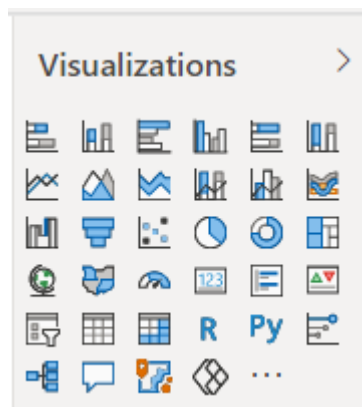


Figure 8.1: native

PowerBI offers a group of native visualizations that we will build most of the dashboard using them.

### 8.1 Comparison visuals

#### 8.1.1 Comparing Numerical values of 1 or more categories

##### 8.1.1.1 Clustered bar/Column chart

#### 8.1.2 Comparing Numerical values of 1 or more categories and Comparing their totals

#### 8.1.2.1 Stacked bar/Column chart

### 8.1.3 Comparing Percentages

#### 8.1.3.1 100% Stacked Bar/Coumn chart

## 8.2 Change over time visuals

### 8.2.1 Displaying Numerical values of 1 or more categories over time dimension

#### 8.2.1.1 Line/Area chart

### 8.2.2 Displaying Numerical values of 1 or more categories over time dimension with focus on Ranking and changes

#### 8.2.2.1 Ribbon chart

### 8.2.3 Displaying Numerical values of 1 or more categories over time dimension with focus on their total

## 8.3 Combo charts

Adding time dimension to comparison visuals or Adding categorical values to change over time visuals

## Chapter 9

# Utilizing PowerBI native visualizations

Clicking on any of the native visuals from the visualization pane will draw the visual on the report page. The lower half of the visualization pane will be divided into 3 sections

**Fields:** (where you can drag and drop data to the right dimension)

**Format:** (to change the look and add conditional formatting)

**Analytics:** (extra set of tools that vary according to the type of the visual)

### 9.1 NYC TLC Dashboard design:

- Monthly dynamic view that can be filtered to weekly and daily views.

Header should contain:

- Logo
- Date controls
- Navigation buttons.

Body should contain:

- The requested 2 KPI tracking visuals. (conditional formatting)
- Revenue breakdown

- Daily performance chart
- Timings chart
- Status updates
- Explaining relation between distance and duration and Fare. (R & Python visuals)

## 9.2 Building our dashboard:

### 9.2.1 Header:

#### 9.2.1.1 3 status update cards

##### **Format :**

Will be the same for all of them, only title is changing.

You can create one with the right format then copy and paste it.

You can use the format painter to apply the same format to multiple visuals.

##### **General**

*Width:* 200

*Height:* 116

##### **Data label**

*Color:* #666666

*Display units:* None

*Value decimal:* 0

*Text size:* 30

*Font family:* Segoe UI Light

##### **Title**

*Font color:* #666666

*Background color:* No fill

*Text size:* 15

*Alignment:* Center

*font family:* Segoe UI Light

##### **Background**

*Color:* #666666

*Transparency:* 84%

##### **Border**

*Color:* #CCCCCC

*Radius:* 4

**Shadow***Color:* #B3B3B3*Shadow position:* Outside*Preset:* Center**9.2.1.1.1 Total fare Amount (Card 1) Title:** Total fare**Field:** nyc\_taxi\_21[Fare] -> sum**9.2.1.1.2 Pick ups Count (Card 2) Title:** Pick ups**Field:** nyc\_taxi\_21[pickup\_location] -> count**9.2.1.1.3 Passengers Count (Card 3) Title:** Passengers**Field:** nyc\_taxi\_21[passenger\_count] -> sum**9.2.1.2 1 Slicer**

After Clicking on the slicer visual, click on the small arrow on the right top corner of the visual and choose droplist.

**Format :****General***Outline color:* #888888*Outline weight:* 1*Width:* 265*Height:* 115**Selection Control***Multi-select with CTRL:* Off**Data label***Color:* #666666*Display units:* None*Value decimal:* 0*Text size:* 30*Font family:* Segoe UI Light**Title***Font color:* #666666*Background color:* No fill*Text size:* 15*Alignment:* Center*font family:* Segoe UI Light**Background***Color:* #666666*Transparency:* 84%

**Border***Color:* #CCCCCC*Radius:* 4**Shadow***Color:* #B3B3B3*Shadow position:* Outside*Preset:* Center

We need to be able to group the data by the month name. To achieve this we need to create a Data group. Creating Data group can be achieved by selecting the needed column to be grouped from the Fields pane, then on the top tool bar new options will appear under name Column tools. Select Data groups and choose New data group.

Name: Months

Bin size: 1 - Months

**Fields:****Field:** [Calendar]Months**Field:** [Calendar]Week of Month\_name**9.2.1.3 1 Image (Logo)****9.2.1.4 2 Images (Navigation)****9.2.1.5 2 Shapes (Navigation)****9.2.2 Body:**

## Chapter 10

# Costum R and Python visuals





## Chapter 11

### Visuals interactions - Slicers & filters.