# MID TERM PROJECT

BDAT 1007 Business Intelligence

Prof. Diego Carbajal

Aakanksha Patel 200533845 | Rajvi Chavda 200537026

# Table of Contents

## Introduction:

Auto insurance provides financial protection to customers against physical damage, resulting from traffic collisions and theft of vehicles. In addition, it covers the cost associated with injuries, death, or property damage caused by the insured owner of the vehicle to another driver, vehicle, or property such as a fence, building, or utility pole. Although auto insurance requirements vary from state to state, bodily injury liability and property damage liability coverage has been mandated in many jurisdictions before using or keeping a vehicle on public roads. The auto insurance market exhibits high growth potential, as the number of road accidents is increasing in most countries across the globe.

The purpose of this project is to understand the main factors that drive customer tenure within the auto insurance industry for one or more years. The analysis is based on 12 months of united state Auto Insurance survey data.

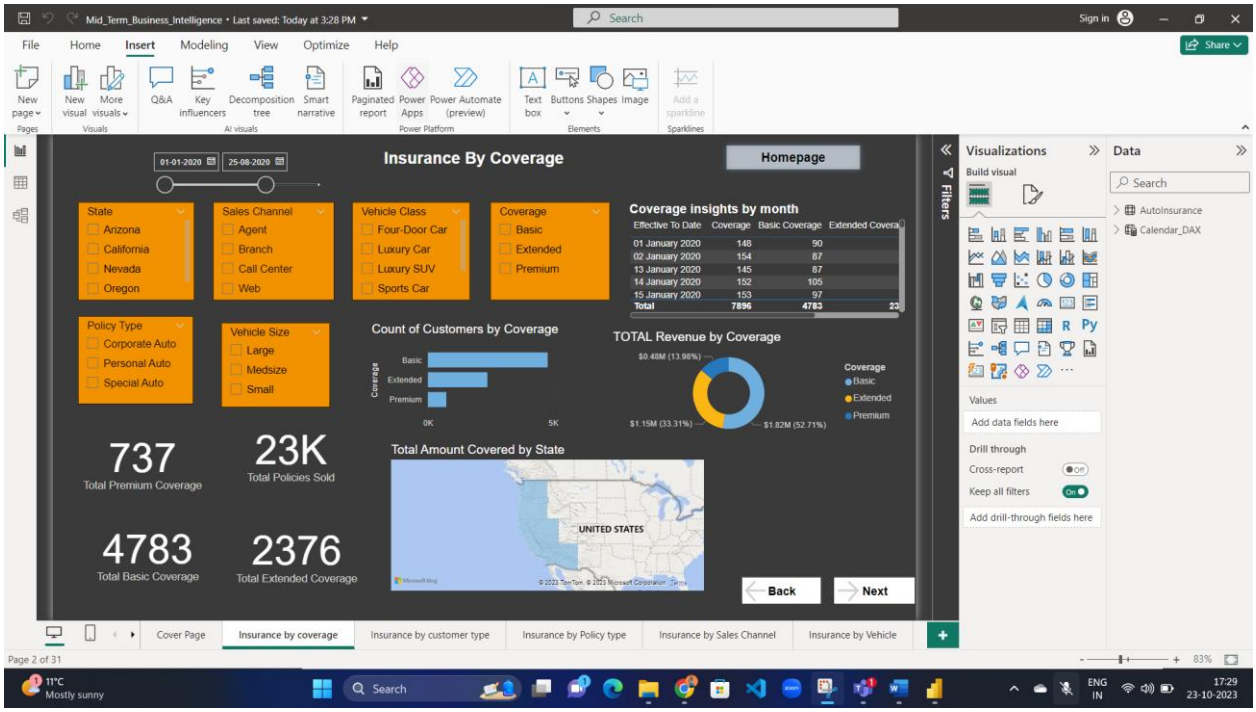
## Objective:

The project involves creating a dashboard (report page) based on the supplied data set, the objective of the report is-

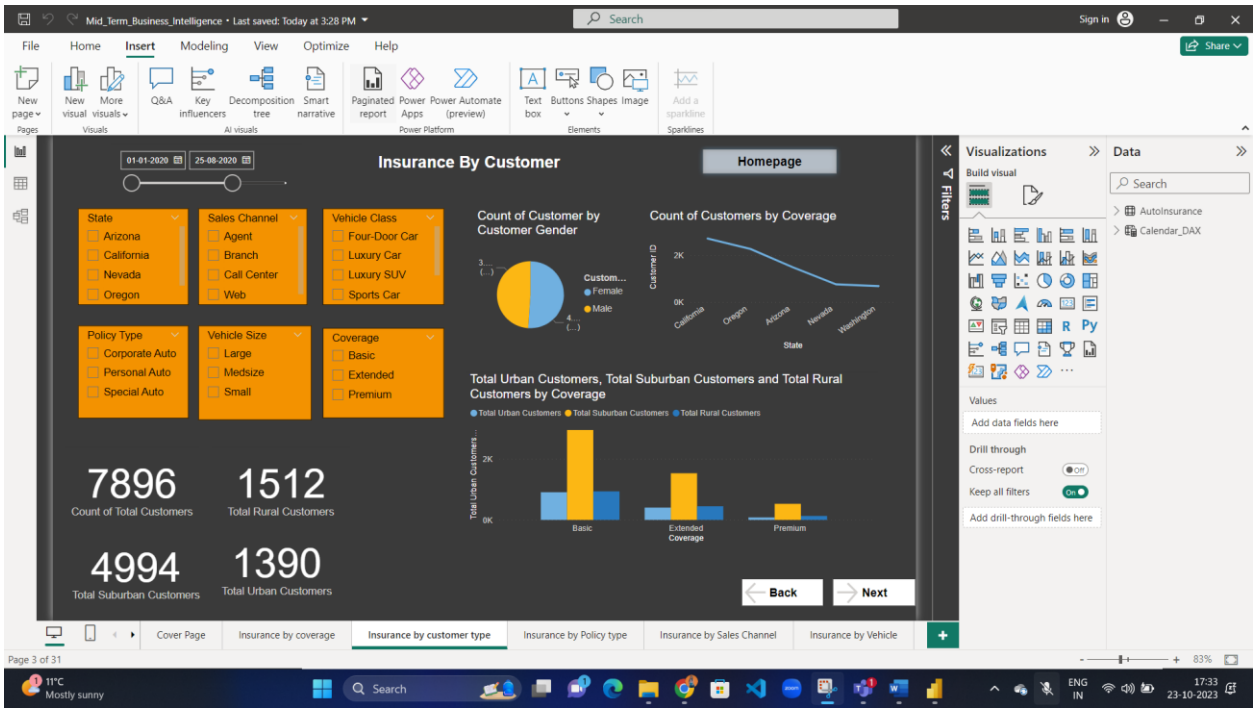
- Loading the datasets in the Power BI desktop.
- Perform data cleaning and data transformation if required.
- Creating the required table Using Power BI DAX.
- Building the data model and creating the relationship based on the key attribute.
- Creating multiple report page (Dashboard) based on the KPI's.
- Apply different type of formatting options if required.
- Publish the report into Power BI services.
- Configure for auto refresh.

Note: Below we have our all screenshots for this project.

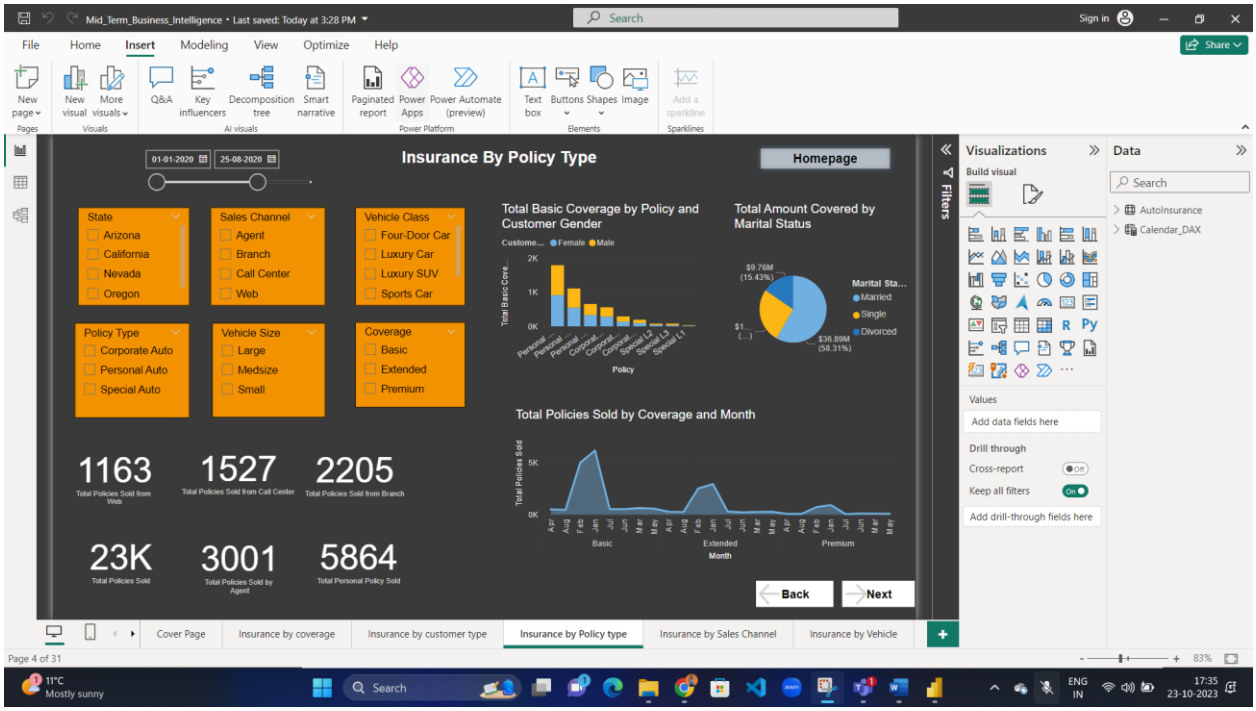
## Insurance by Coverage:



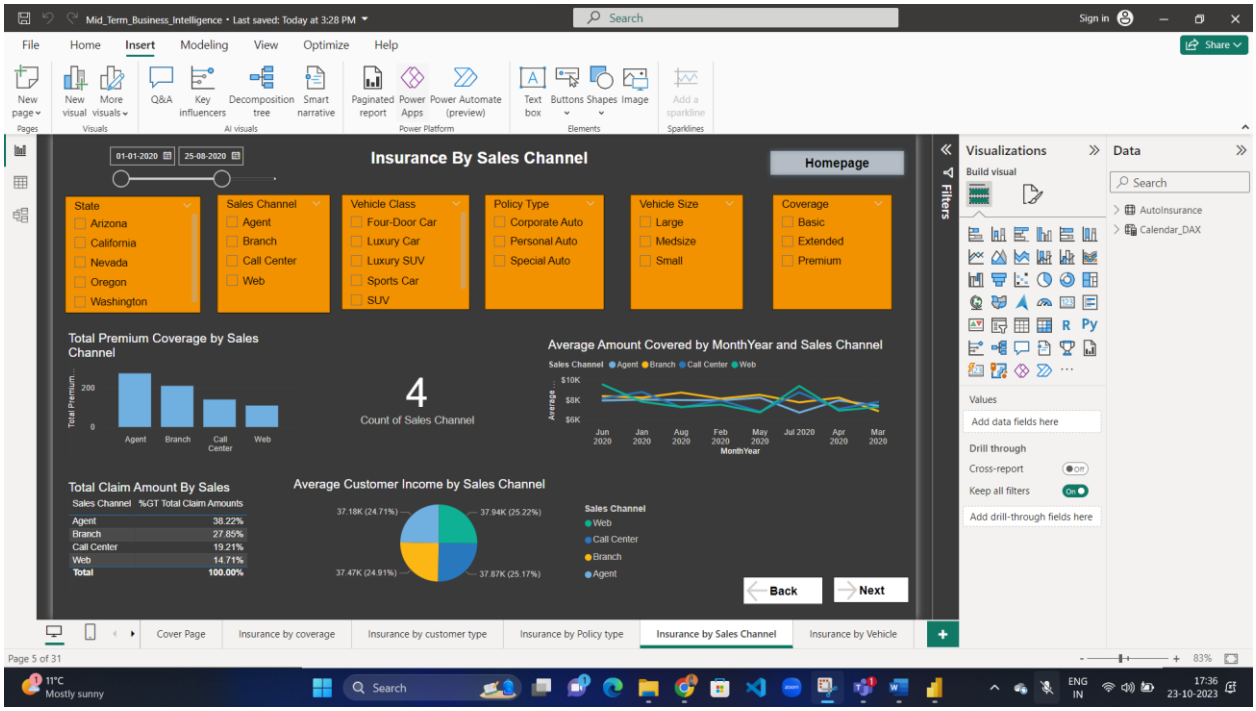

## Insurance by Customer type:



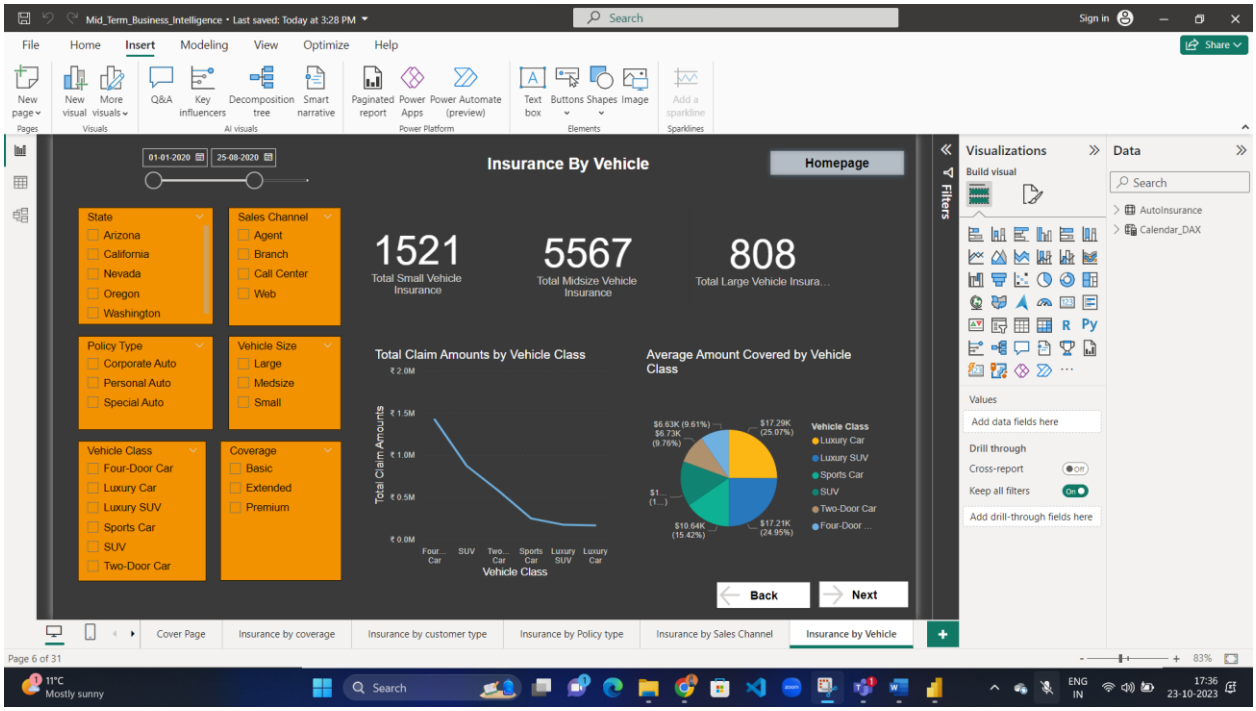
## Insurance by Policy type:




## Insurance by Sales Channel:



## Insurance by Vehicle:




## The KPI's value using DAX formula

1. Total Number of Customers:
    a. Formula: `COUNTROWS(AutoInsurance)`
    b. Description: This KPI counts the total number of customers in the "AutoInsurance" dataset, providing a clear picture of your customer base size.
2. Total Number of Policies Sold:
    a. Formula: `SUM(AutoInsurance[Number of Policies])`
    b. Description: This KPI calculates the sum of the "Number of Policies" from the "AutoInsurance" dataset, representing the total count of insurance policies sold.
3. Total Amount Covered:
    a. Formula: `SUM(AutoInsurance[Customer Lifetime Value])`
    b. Description: This KPI sums up the "Customer Lifetime Value" from the "AutoInsurance" dataset, showing the total monetary amount covered by insurance policies for all customers.
4. Average Amount Covered:
    a. Formula: `AVERAGE(AutoInsurance[Customer Lifetime Value])`
    b. Description: This KPI calculates the average "Customer Lifetime Value" from the "AutoInsurance" dataset, providing the typical coverage amount per customer.
5. Average Policy Sold:
    a. Formula: `AVERAGE(AutoInsurance[Number of Policies])`

   b. Description: This KPI calculates the average "Number of Policies" from the "AutoInsurance" dataset, offering insights into the average number of policies held by your customers.

   c. 6.Total Number of Open Complaints:
   d. Formula: `SUM(AutoInsurance[Number of Open Complaints])`
   e. Description: This KPI sums up the "Number of Open Complaints" from the "AutoInsurance" dataset, representing the overall count of open customer complaints.

6. Total Claim Amounts:
   a. Formula: `SUM(AutoInsurance[Total Claim Amount])`
   b. Description: This KPI calculates the total amount claimed by customers by summing up the "Total Claim Amount" from the "AutoInsurance" dataset.

7. Average Claim Amount:
   a. Formula: `AVERAGE(AutoInsurance[Total Claim Amount])`
   b. Description: This KPI calculates the average "Total Claim Amount" from the "AutoInsurance" dataset, providing insights into the typical claim amount per customer.

8. Average Monthly Premium:
   a. Formula: `AVERAGE(AutoInsurance[Monthly Premium Auto])`
   b. Description: This KPI calculates the average monthly premium paid by customers, based on the "Monthly Premium Auto" column in the "AutoInsurance" dataset.

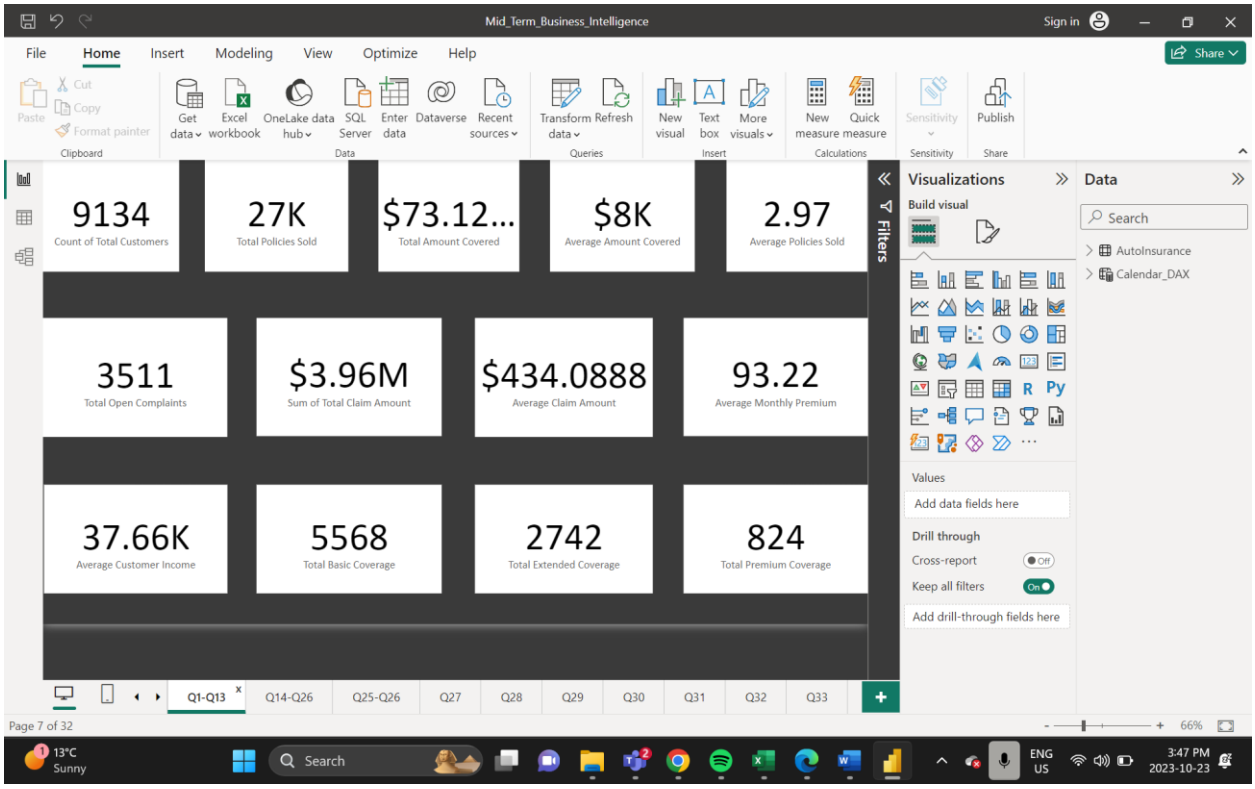
9. Average Customer Income:
   a. Formula: `AVERAGE(AutoInsurance[Income])`
   b. Description: This KPI calculates the average customer income, based on the "Income" column in the "AutoInsurance" dataset, providing insights into the typical income level of your customers.

10. Total Basic Coverage:
    a. Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Coverage] = "Basic")`
    b. Description: This KPI counts the number of customers with "Basic" coverage in the "AutoInsurance" dataset, indicating the total count of customers with basic insurance coverage.

11. Total Extended Coverage:

a. Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Coverage] = "Extended"))`
b. Description: This KPI counts the number of customers with "Extended" coverage in the "AutoInsurance" dataset, representing the total count of customers with extended insurance coverage.

12. Total Premium Coverage:
    a. Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Coverage] = "Premium"))`
    b. Description: This KPI counts the number of customers with "Premium" coverage in the "AutoInsurance" dataset, indicating the total count of customers with premium-level insurance coverage.




14. Total Suburban Customers:
- Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Location Code] = "Suburban"))`
- Description: This KPI counts the number of customers whose "Location Code" is labeled as "Suburban" in the "AutoInsurance" dataset, indicating the total count of suburban customers.

15. Total Rural Customers:
- Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Location Code] = "Rural"))`
- Description: This KPI counts the number of customers with a "Location Code" of "Rural" in the "AutoInsurance" dataset, representing the total count of rural customers.

16. Total Urban Customers:
  - Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Location Code] = "Urban"))`
  - Description: This KPI counts the number of customers with an "Location Code" of "Urban" in the "AutoInsurance" dataset, indicating the total count of urban customers.

17. Total Corporate Policy Sold:
  - Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Policy Type] = "Corporate Auto"))`
  - Description: This KPI counts the number of customers with a "Policy Type" of "Corporate Auto" in the "AutoInsurance" dataset, showing the total count of corporate policies sold.

18. Total Personal Policy Sold:
  - Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Policy Type] = "Personal Auto"))`
  - Description: This KPI counts the number of customers with a "Policy Type" of "Personal Auto" in the "AutoInsurance" dataset, representing the total count of personal policies sold.

19. Total Special Policy Sold:
  - Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Policy Type] = "Special Auto"))`
  - Description: This KPI counts the number of customers with a "Policy Type" of "Special Auto" in the "AutoInsurance" dataset, indicating the total count of special policies sold.

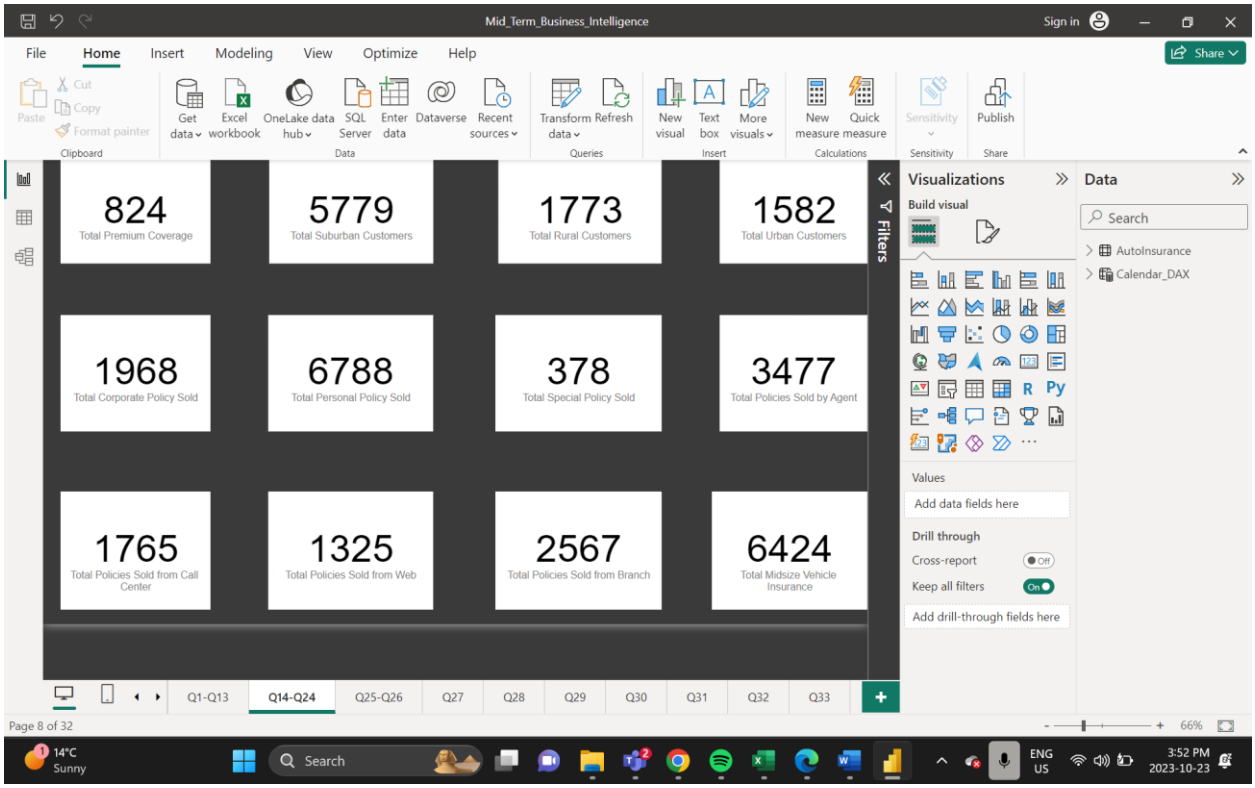
20. Total Policy Sold by Agent:
  - Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Sales Channel] = "Agent"))`
  - Description: This KPI counts the number of customers whose "Sales Channel" is labeled as "Agent" in the "AutoInsurance" dataset, showing the total count of policies sold by agents.

21. Total Policy Sold from Call Center:
  - Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Sales Channel] = "Call Center"))`
  - Description: This KPI counts the number of customers whose "Sales Channel" is "Call Center" in the "AutoInsurance" dataset, representing the total count of policies sold through call centers.

22. Total Policy Sold from Web:
  - Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Sales Channel] = "Web"))`
  - Description: This KPI counts the number of customers whose "Sales Channel" is "Web" in the "AutoInsurance" dataset, indicating the total count of policies sold online.

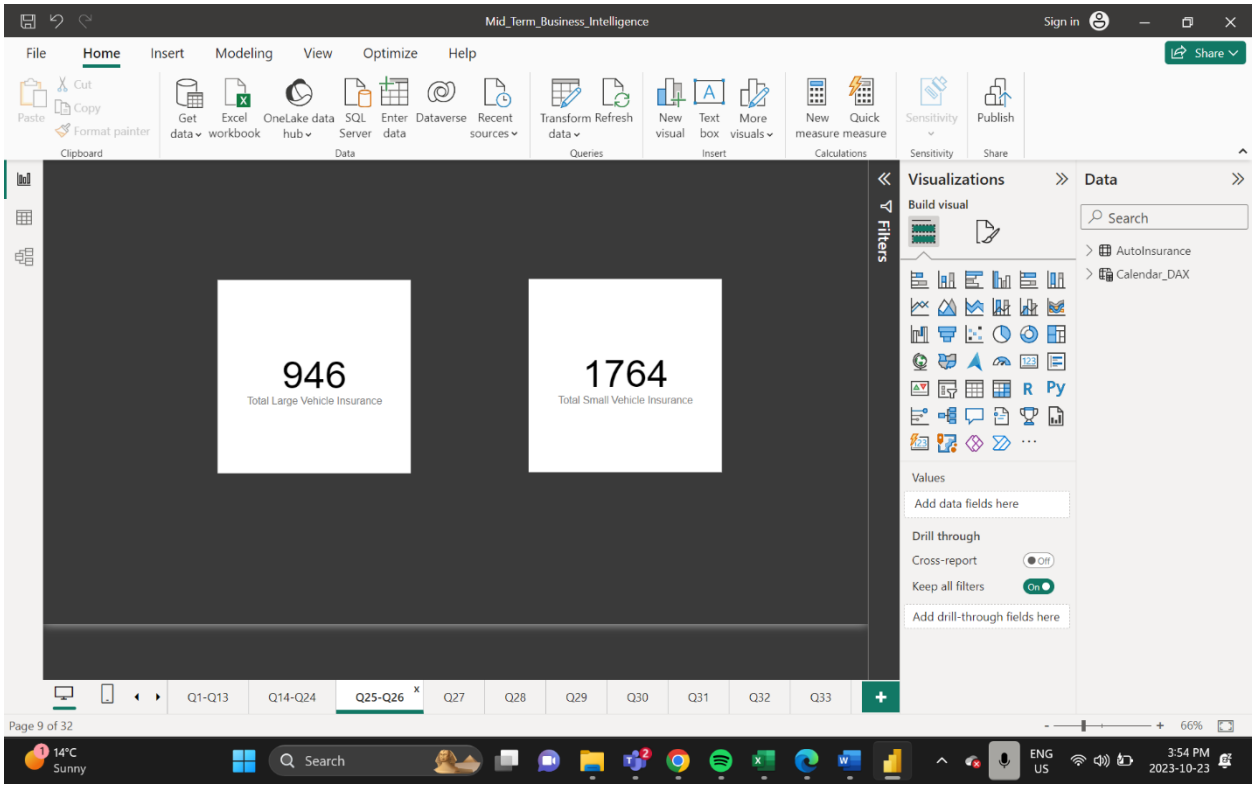
23. Total Policy Sold from Branch:
  - Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Sales Channel] = "Branch"))`
  - Description: This KPI counts the number of customers whose "Sales Channel" is "Branch" in the "AutoInsurance" dataset, showing the total count of policies sold through branch locations.

24. Total Midsize Vehicle Insurance:
- Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Vehicle Size] = "Medsize"))`
- Description: This KPI counts the number of customers with "Vehicle Size" labeled as "Medsize" in the "AutoInsurance" dataset, indicating the total count of midsize vehicle insurance policies.

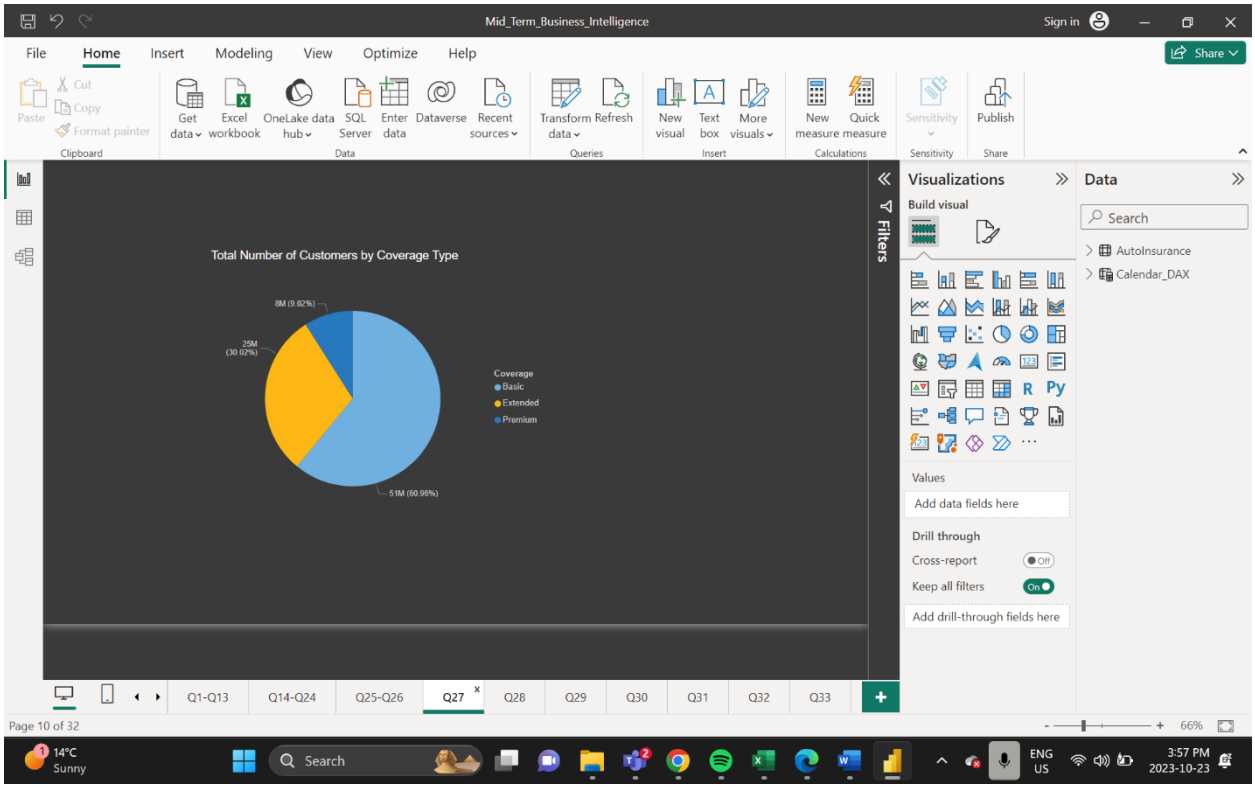



25. Total Large Vehicle Insurance:
- Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Vehicle Size] = "Large"))`
- Description: This KPI counts the number of customers with a "Vehicle Size" of "Large" in the "AutoInsurance" dataset, representing the total count of large vehicle insurance policies.

26. Total Small Vehicle Insurance:
- Formula: `COUNTROWS(FILTER(AutoInsurance, AutoInsurance[Vehicle Size] = "Small"))`
- Description: This KPI counts the number of customers with a "Vehicle Size" of "Small" in the "AutoInsurance" dataset, indicating the total count of small vehicle insurance policies.

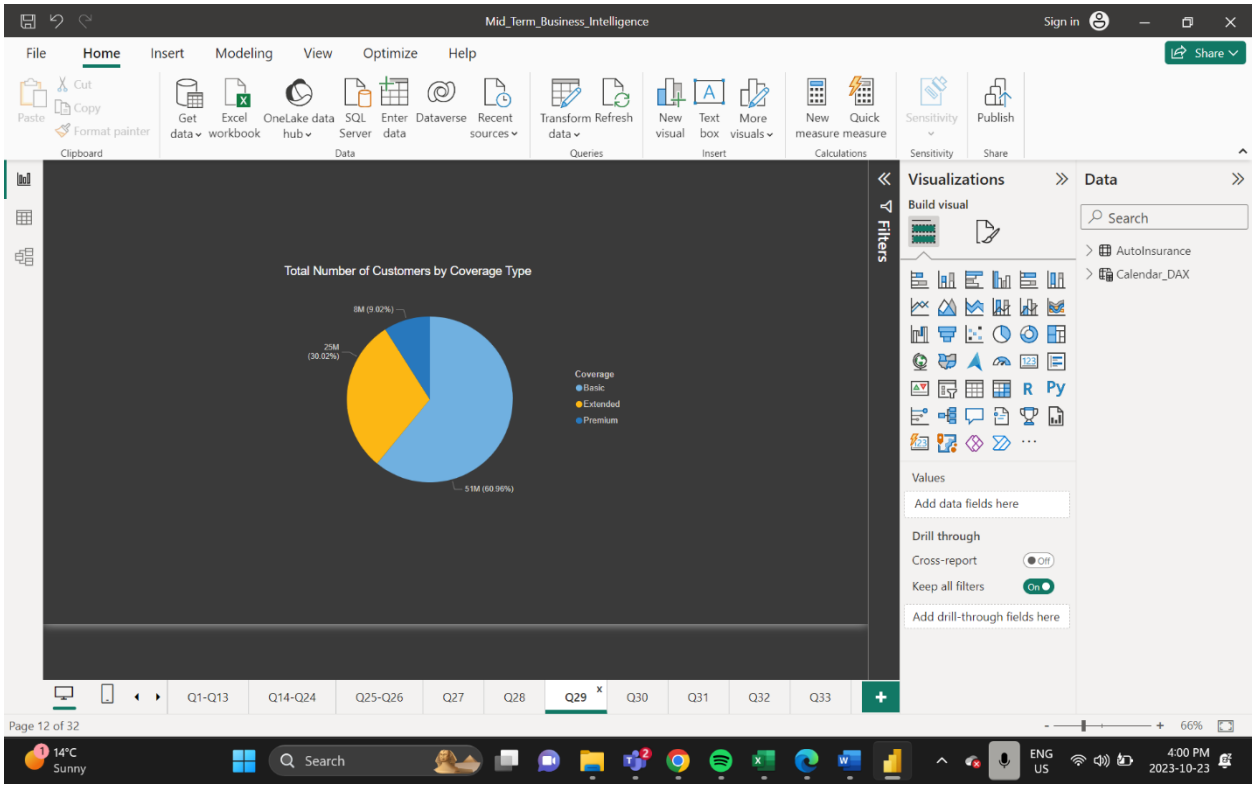



27. Pie chart: Total Number of Customers by Coverage Type.
    - Select the relevant data fields, including "Coverage" and "Customer" or "Customer ID."
    - Write DAX formulas to count customers for each coverage type using `COUNTROWS` and `FILTER`.
    - Create a Pie Chart in your data visualization tool (e.g., Power BI).
    - Assign the "Coverage" field to the chart's category/legend section and customer counts to the values section.
    - Label the chart appropriately, such as "Total Number of Customers by Coverage Type."
    - Customize and format the chart for clarity and aesthetics.
    - The resulting Pie Chart visually displays the distribution of customers across different coverage types.




28. Donut chart: Total Number of Customers by Customer Gender.

- Select relevant data fields, including "Customer Gender" and a unique identifier.
- Write DAX formulas to count customers for each gender category using `COUNTROWS` and `FILTER`.
- Create a Donut Chart in your data visualization tool.
- Assign "Customer Gender" to the chart's category/legend section and customer counts to the values section.
- Label the chart as "Total Number of Customers by Customer Gender."
- Customize the chart for clarity and aesthetics.
- The resulting Donut Chart visually displays the customer distribution by gender.




29 .To create a Pie Chart showing the "Total Number of Customers by Coverage Type":

1. Select data fields, including "Coverage" and a unique customer identifier.

2. Calculate the count of customers for each coverage type using DAX formulas with `COUNTROWS` and `FILTER`.

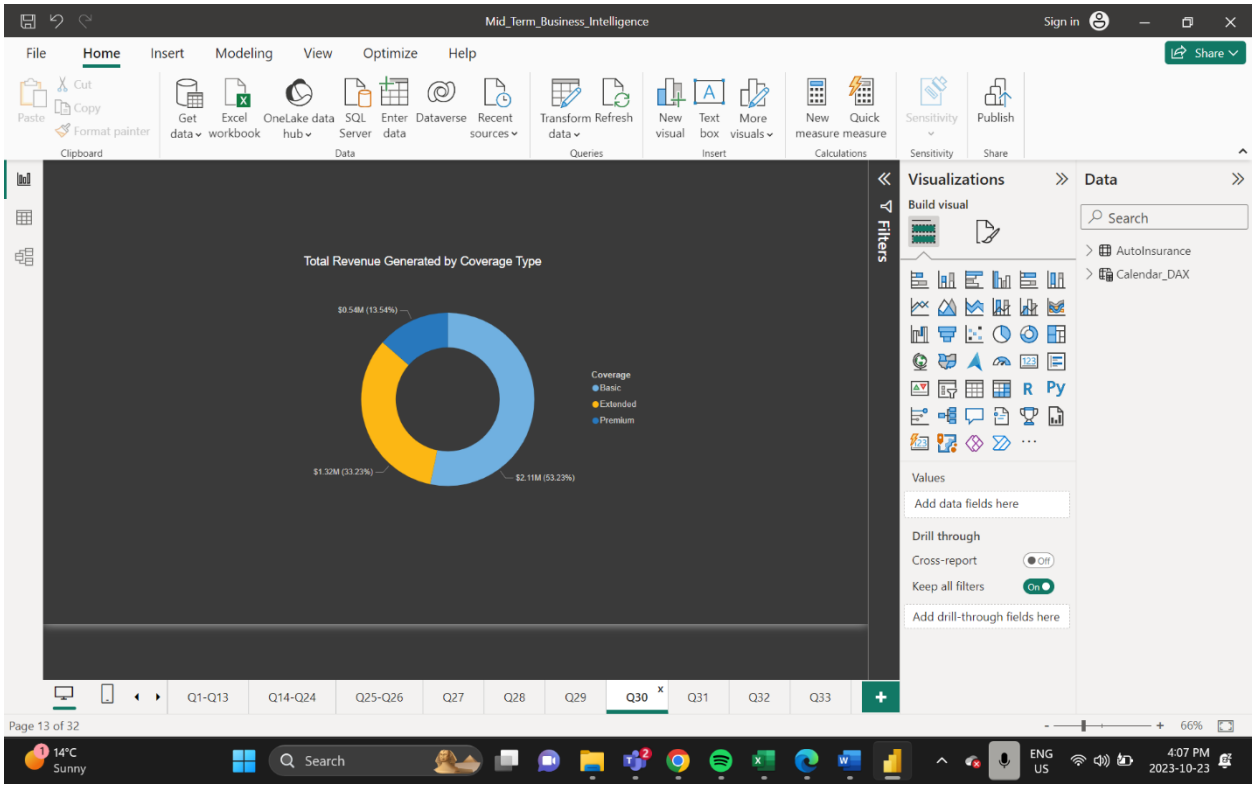
3. Create a Pie Chart in your data visualization tool.

4. Assign "Coverage" to the category/legend section and customer counts to the values section.

5. Label the chart as "Total Number of Customers by Coverage Type."

6. Customize the chart for clarity and aesthetics.

7. The Pie Chart visually illustrates the distribution of customers across different coverage types.




30. To create a Donut Chart representing the "Total Revenue Generated by Coverage Type":

1. Select relevant data fields, including "Coverage Type" and a field with revenue data.

2. Use DAX formulas to calculate the total revenue for each coverage type, typically with `SUMX` and `FILTER` functions.

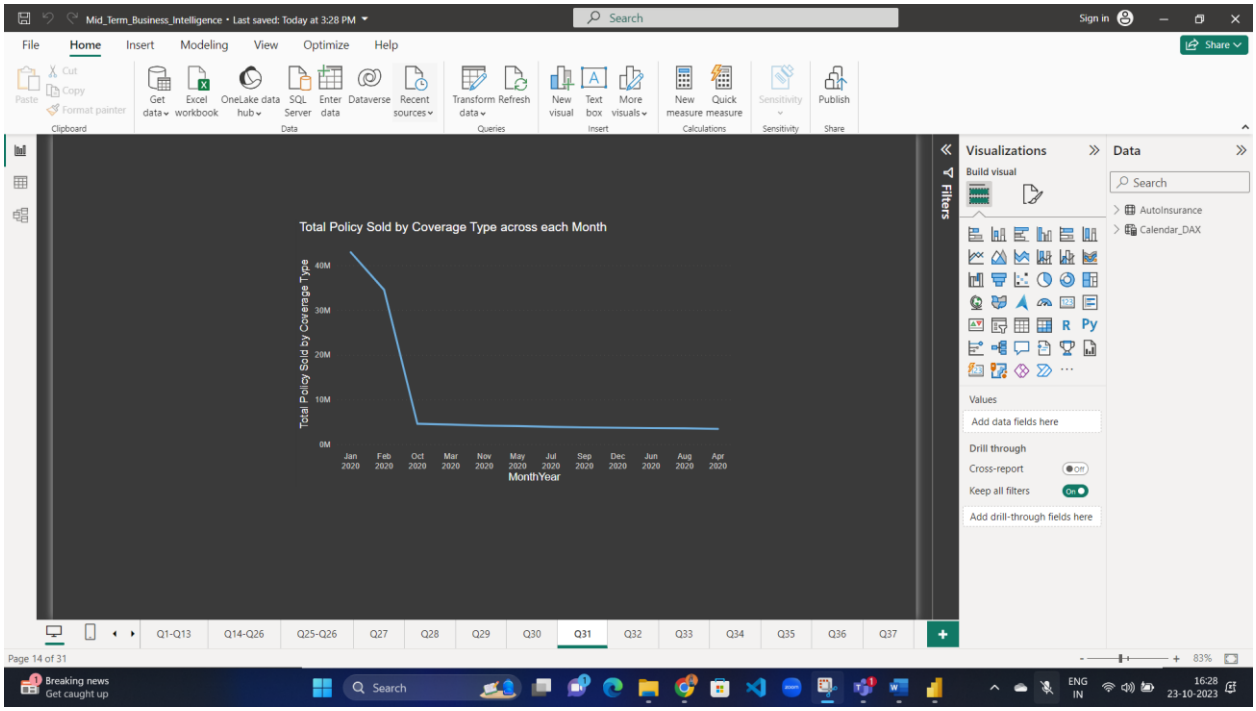
3. Create a Donut Chart using your data visualization tool.

4. Assign "Coverage Type" to the category/legend section and the calculated revenue to the values section.

5. Label the chart as "Total Revenue Generated by Coverage Type."

6. Customize the chart for clarity and aesthetics.

7. The resulting Donut Chart visually shows the contribution of each coverage type to the total revenue.




31.Line Chart: Total Policy Sold by Coverage Type across each Month.

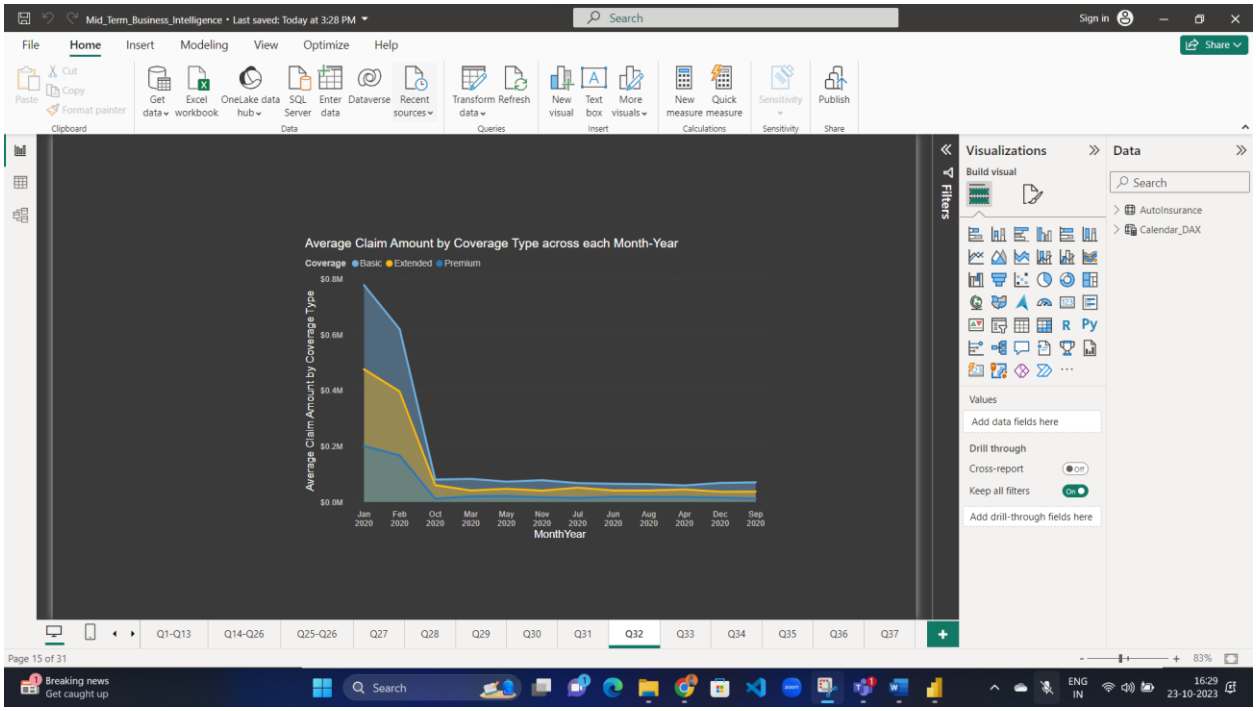
1. Import your data, including "Month," "Coverage Type," and "Policies Sold."

2. Create a new line chart visual.

3. Configure the visual by placing "Month" on the X-axis and "Coverage Type" in the Legend section.

4. If needed, create a measure to calculate the total policies sold.

5. Add the measure to the Values section of the visual.




32. Area Chart: Average Claim Amount by Coverage Type across each Month-Year.

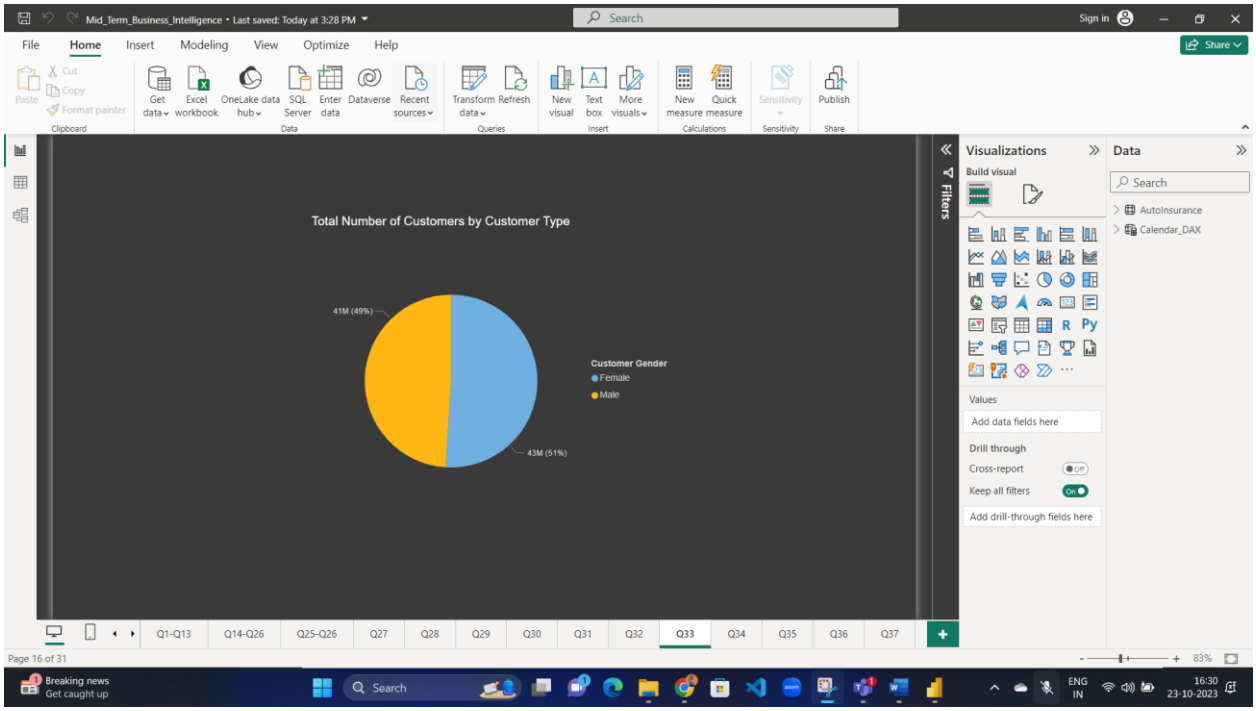
1. Import your data, including "Month-Year," "Coverage Type," and "Claim Amount."

2. Create a new area chart visual.

3. Configure the visual by placing "Month-Year" on the X-axis and "Coverage Type" in the Legend section.

4. If needed, create a measure to calculate the average claim amount.

5. Add the measure to the Values section of the visual.



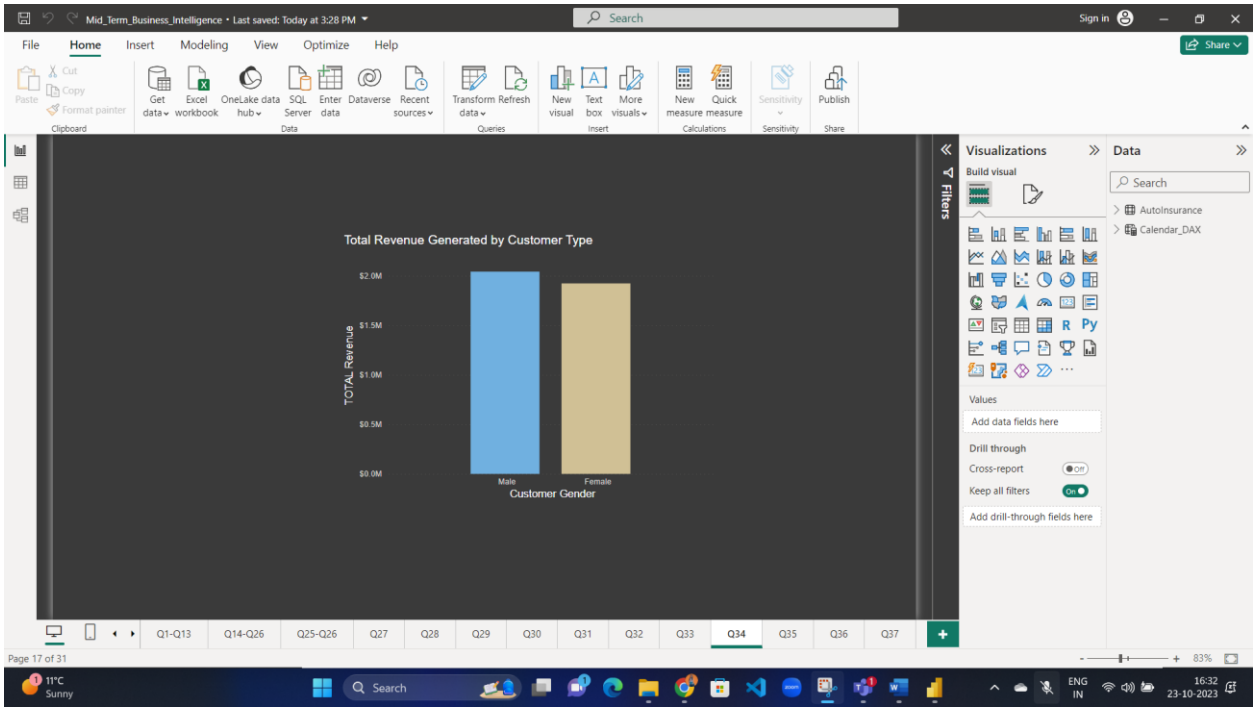

33.Pie Chart: Total Number of Customers by Customer Type.

1. Import your data, including a "Customer Type" column and unique customer identifiers.

2. Create a new pie chart visual.

3. Configure the visual by dragging the "Customer Type" field into the Values section.

4. Interact with the visual to see the distribution of customers by type.



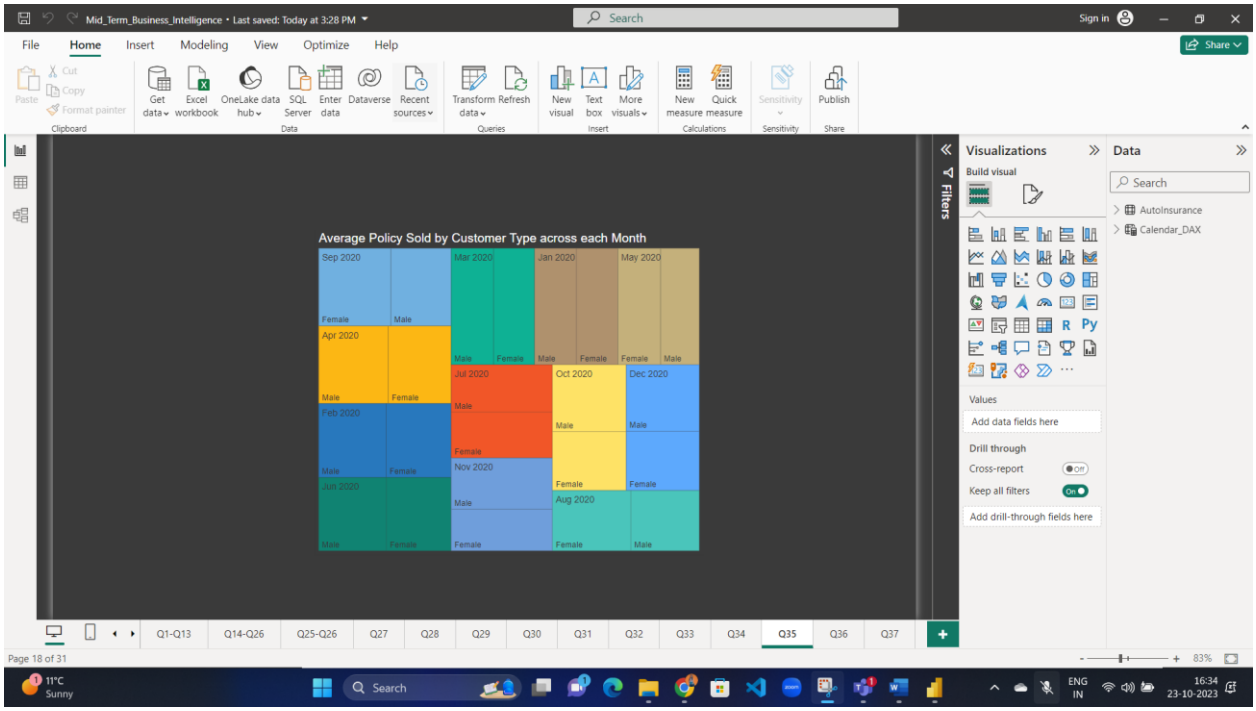

34.Bar Chart: Total Revenue Generated by Customer Type.

1. Import your data with "Customer Type" and "Revenue" columns.

2. Create a Bar Chart visual.

3. Configure the chart with "Customer Type" on the X-axis and "Revenue" as the Values.

4. Interact with the visual to see revenue by customer type.




35.Tree Map: Average Policy Sold by Customer Type across each Month.

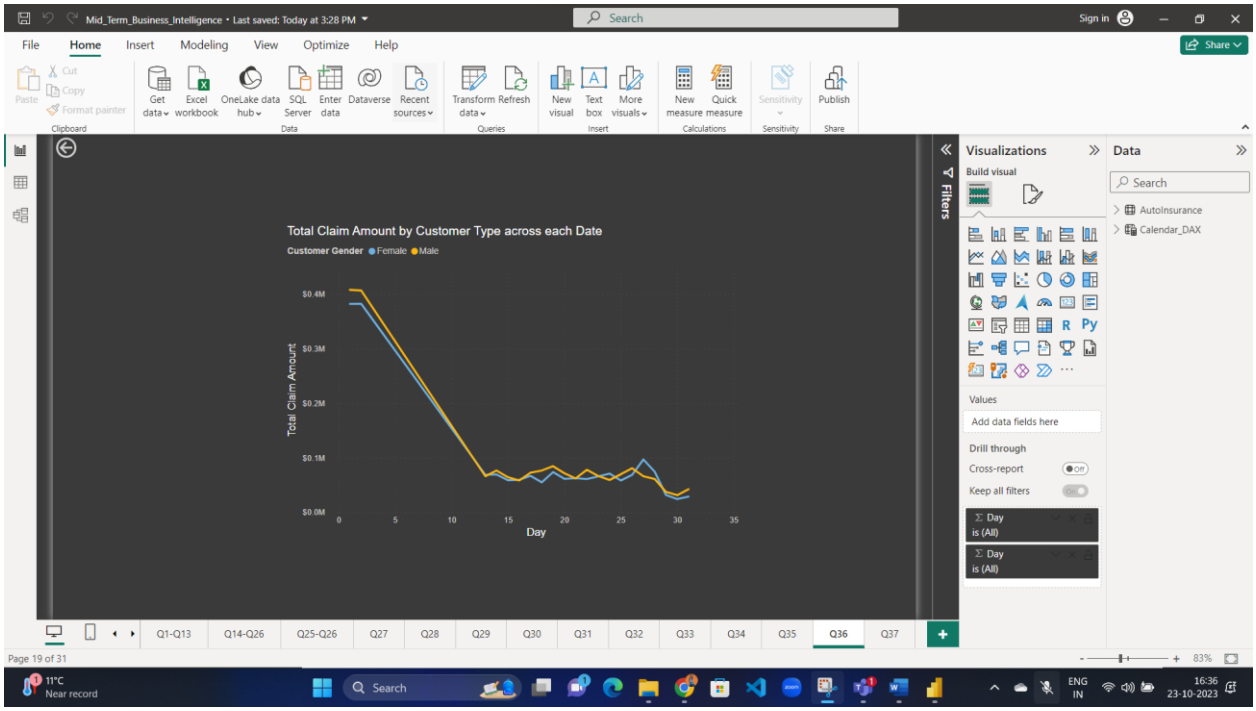
1. Import your data, including "Month," "Customer Type," and "Policies Sold."

2. Create a Tree Map visual.

3. Configure the visual with "Month" in the "Group" section and "Customer Type" in the "Details" or "Color saturation" section.

4. Optionally, create a measure to calculate the average policies sold.

5. Add the measure to the "Values" section of the visual.




36.Line Chart: Total Claim Amount by Customer Type across each Date (Year/Qtr/Month/Day).

1. Import your data, including "Date," "Customer Type," and "Claim Amount."

2. Create a Line Chart visual.

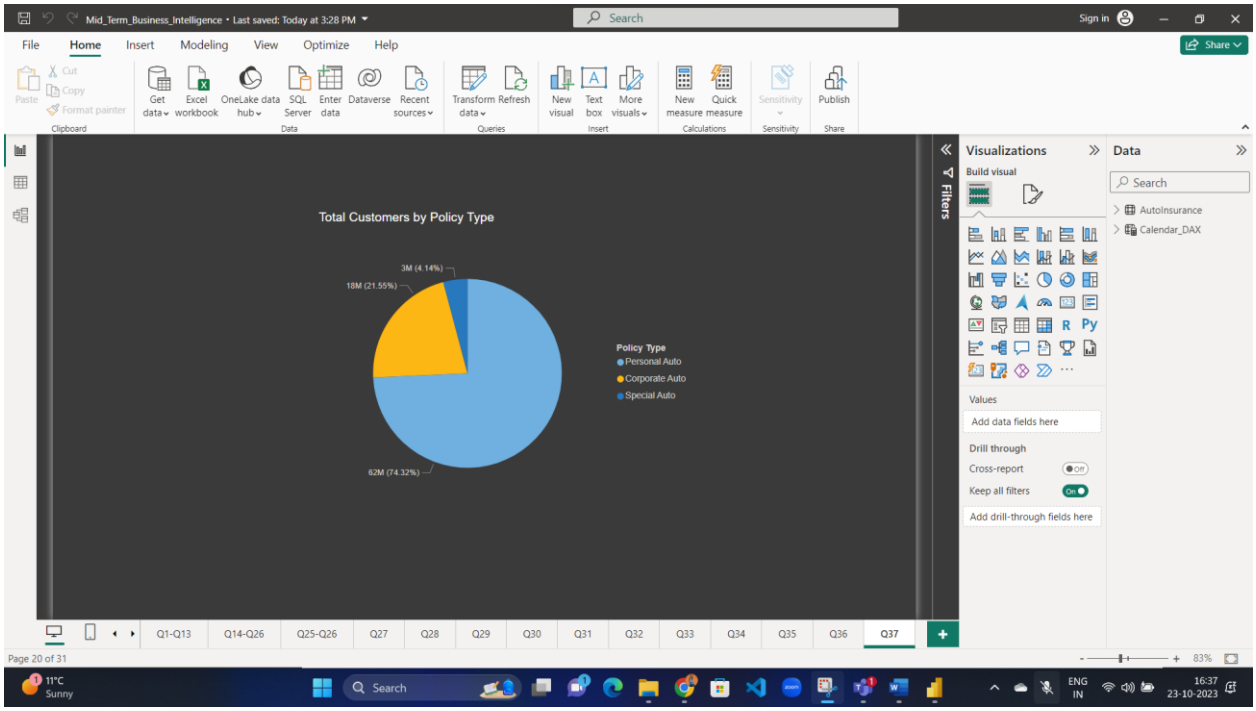
3. Configure the visual with "Date" on the X-axis and "Customer Type" in the Legend section.

4. Optionally, create a measure to calculate the total claim amount.

5. Add the measure to the Values section of the visual.

6. Use the date hierarchy to select the desired time scale (Year, Quarter, Month, Day).

7. Interact with the Line Chart to view the total claim amount by customer type over the selected time scale.



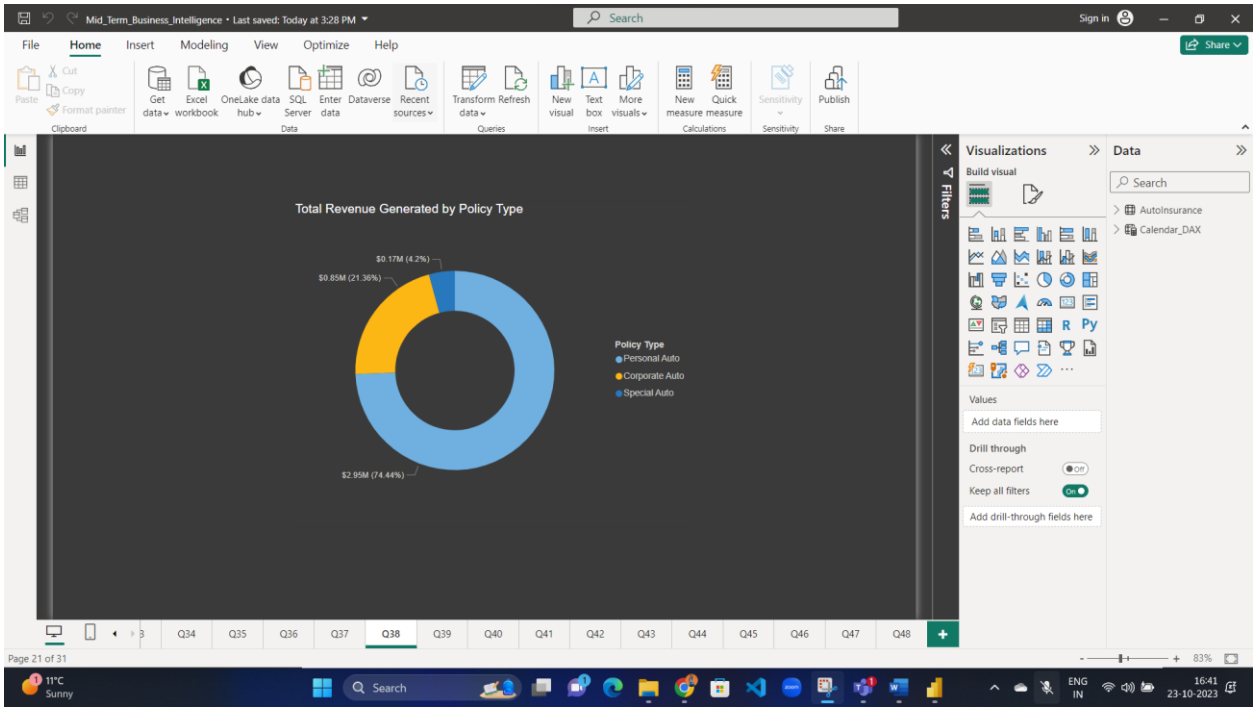

37. Pie Chart: Total Number of Customers by Policy type.

1. Import your data with "Policy Type" and customer information.

2. Create a Pie Chart visual.

3. Configure the visual with "Policy Type" in the "Values" section.

4. Interact with the Pie Chart to see the distribution of customers by policy type.



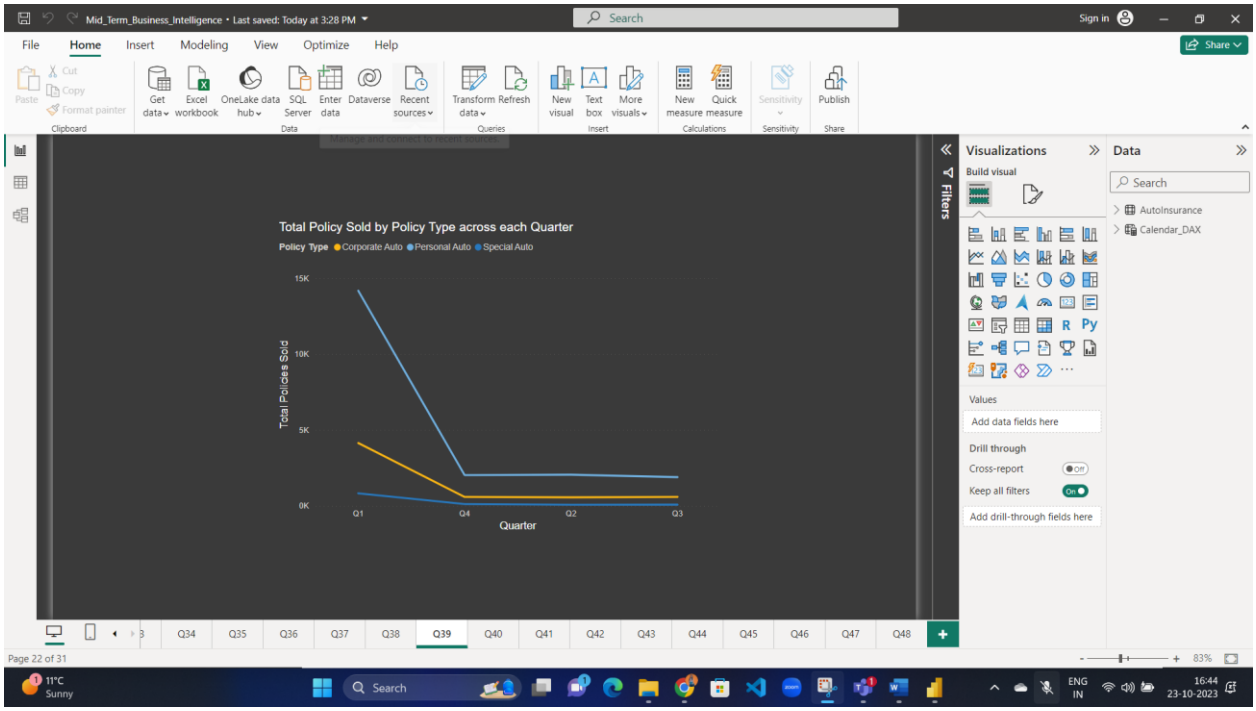

38.Donut Chart: Total Revenue Generated by Policy Type.

1. Import your data, including "Policy Type" and "Revenue" columns.

2. Create a Donut Chart visual.

3. Configure the visual with "Policy Type" in the Legend section and "Revenue" in the Values section.

4. Interact with the Donut Chart to see the distribution of revenue by policy type.




39. Line Chart: Total Policy Sold by Policy Type across each Quarter

1. Import your data with columns for "Quarter," "Policy Type," and "Policies Sold."

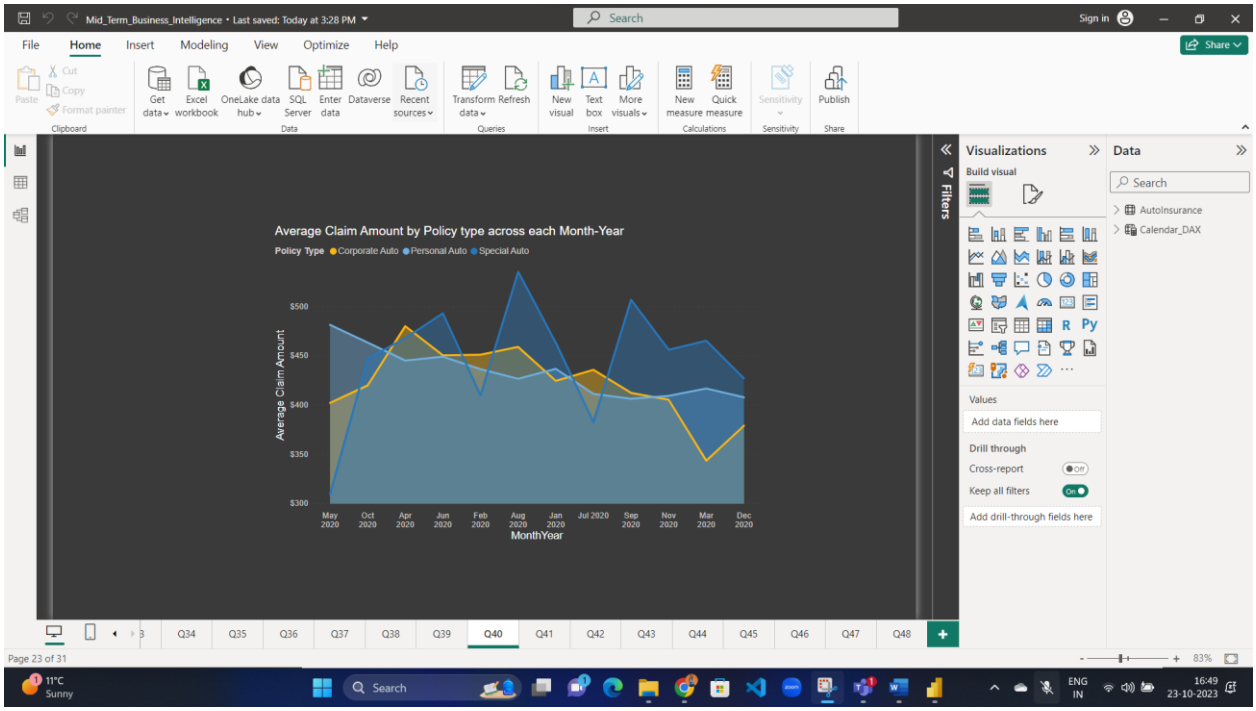
2. Create a Line Chart visual.

3. Configure the visual with "Quarter" on the X-axis and "Policy Type" in the Legend section.

4. Optionally, create a measure for total policies sold.

5. Add the measure to the Values section of the visual.

6. Interact with the Line Chart to see policy sales by type across each quarter.




40.Area Chart: Average Claim Amount by Policy type across each Month-Year.

1. Import your data with columns for "Month-Year," "Policy Type," and "Claim Amount."

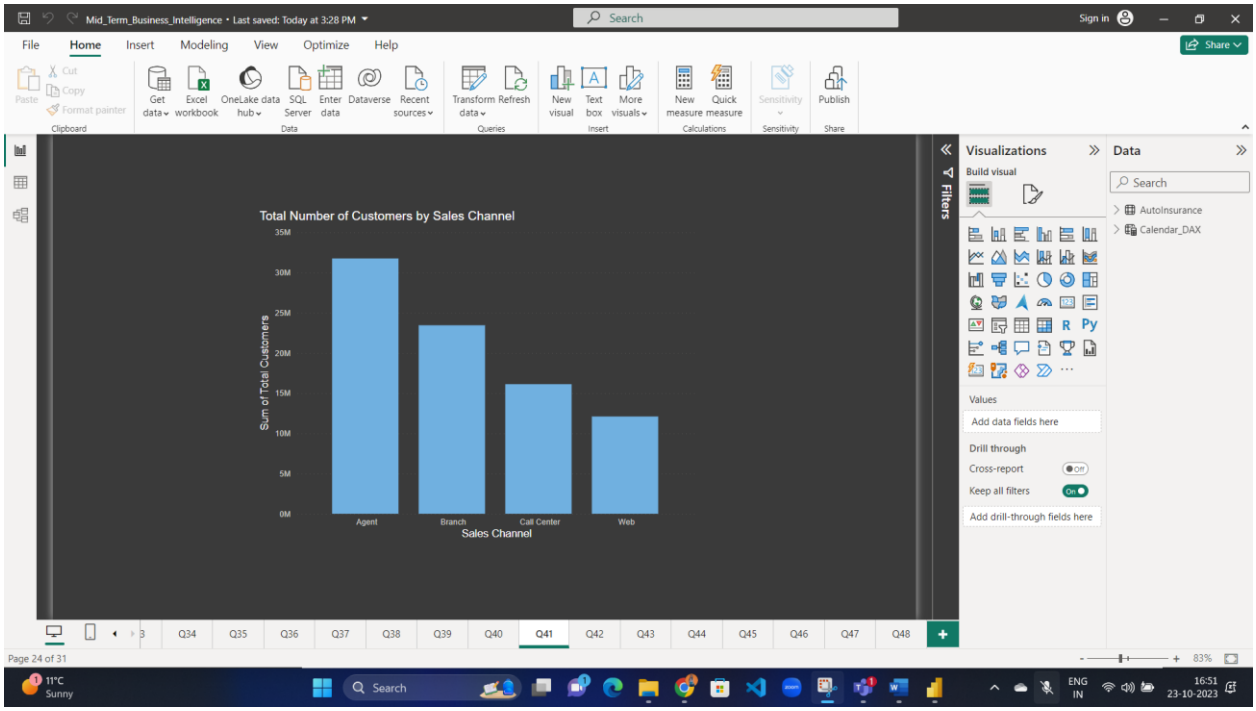
2. Create an Area Chart visual.

3. Configure the visual with "Month-Year" on the X-axis and "Policy Type" in the Legend section.

4. Optionally, create a measure to calculate the average claim amount.

5. Add the measure to the Values section of the visual.

6. Interact with the Area Chart to view the average claim amount by policy type over each Month-Year.

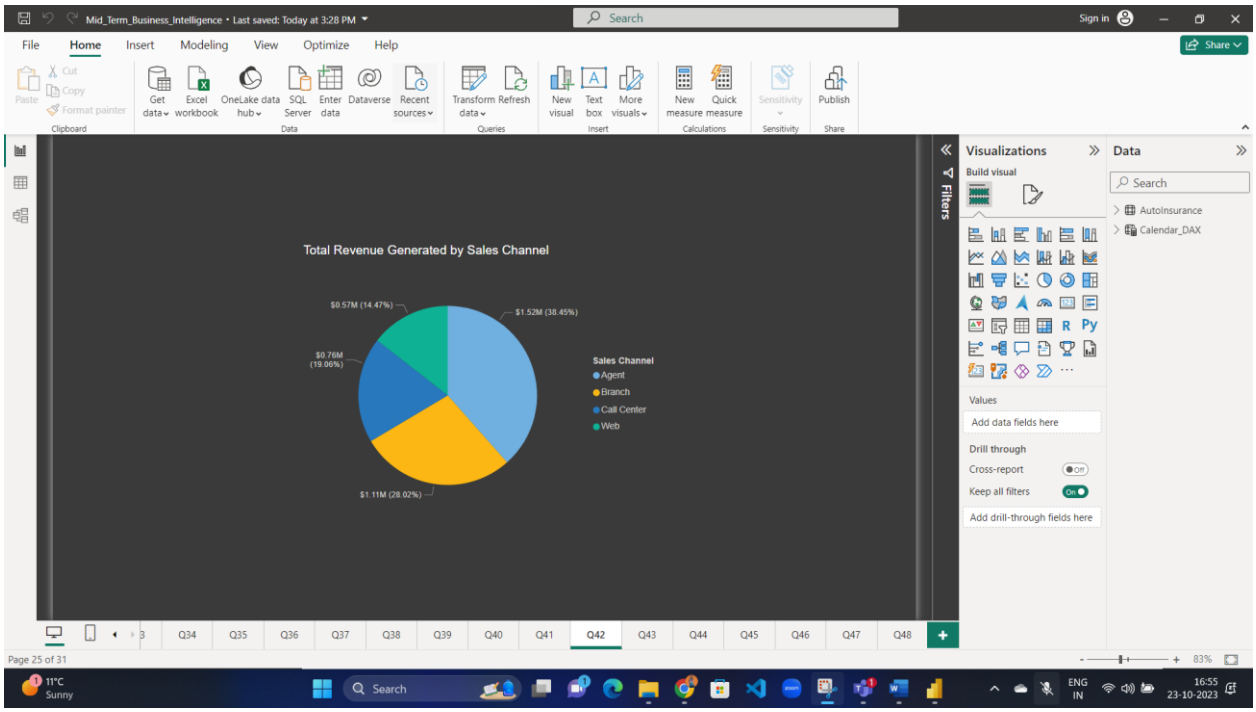



41.Bar Chart: Total Number of Customers by Sales Channel.

1. Import your data with columns for "Sales Channel" and customer identifiers.

2. Create a Bar Chart visual.

3. Configure the visual with "Sales Channel" on the X-axis.

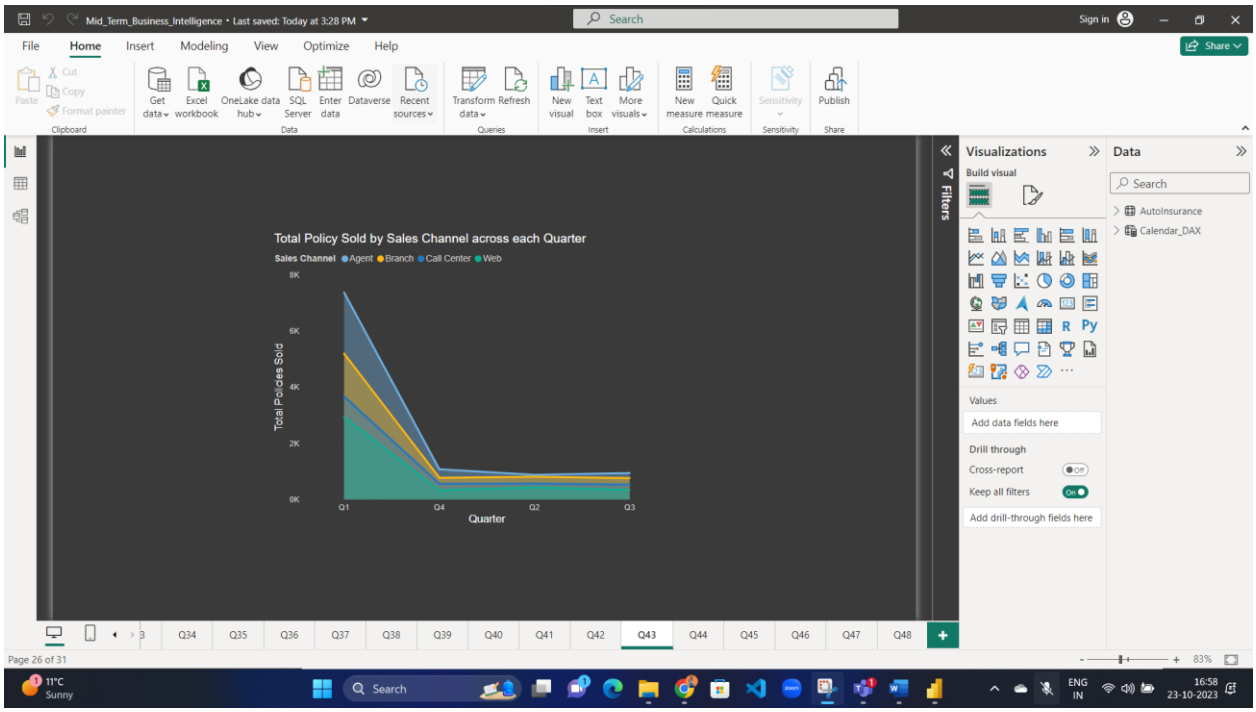



42.Pie Chart: Total Revenue Generated by Sales Channel.

1. Import your data with columns for "Sales Channel" and "Revenue."

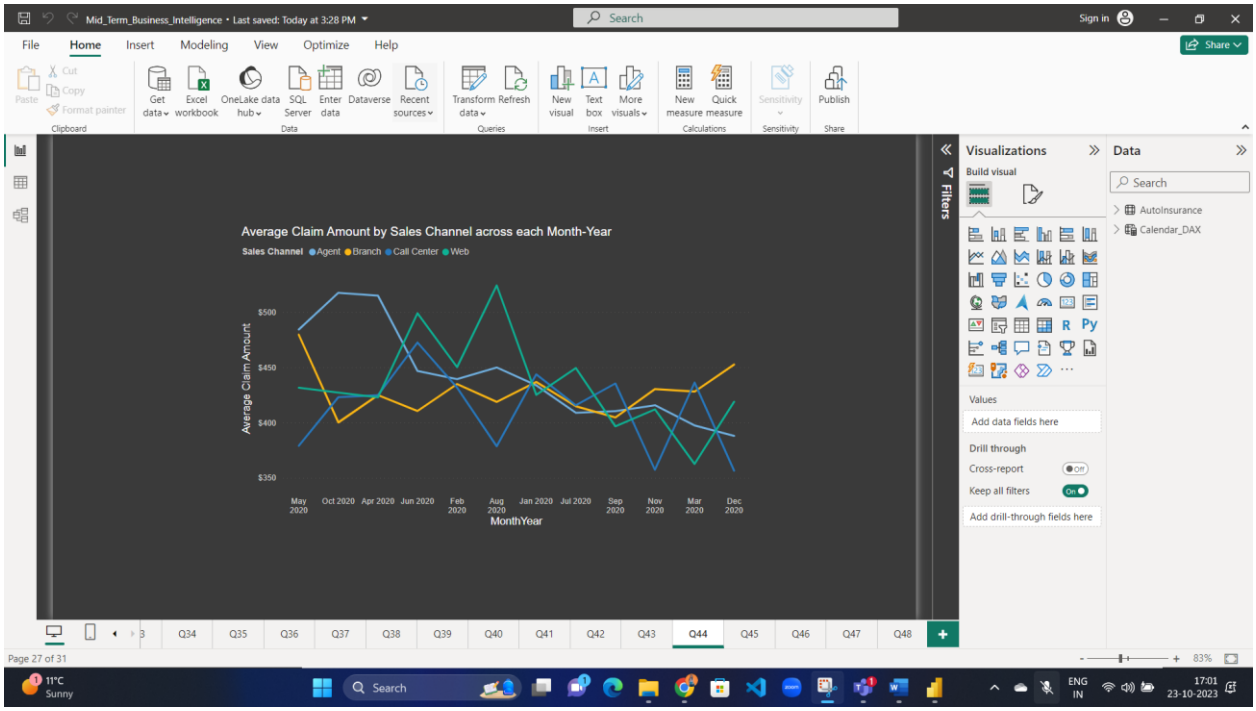
2. Create a Pie Chart visual.

3. Configure the visual with "Sales Channel" in the Legend section and "Revenue" in the Values section.




43.Area Chart: Total Policy Sold by Sales Channel across each Quarter.

1. Import your data with columns for "Quarter," "Sales Channel," and "Policies Sold."
2. Create an Area Chart visual.
3. Configure the visual with "Quarter" on the X-axis and "Sales Channel" in the Legend section.
4. Optionally, create a measure to calculate the total policies sold.

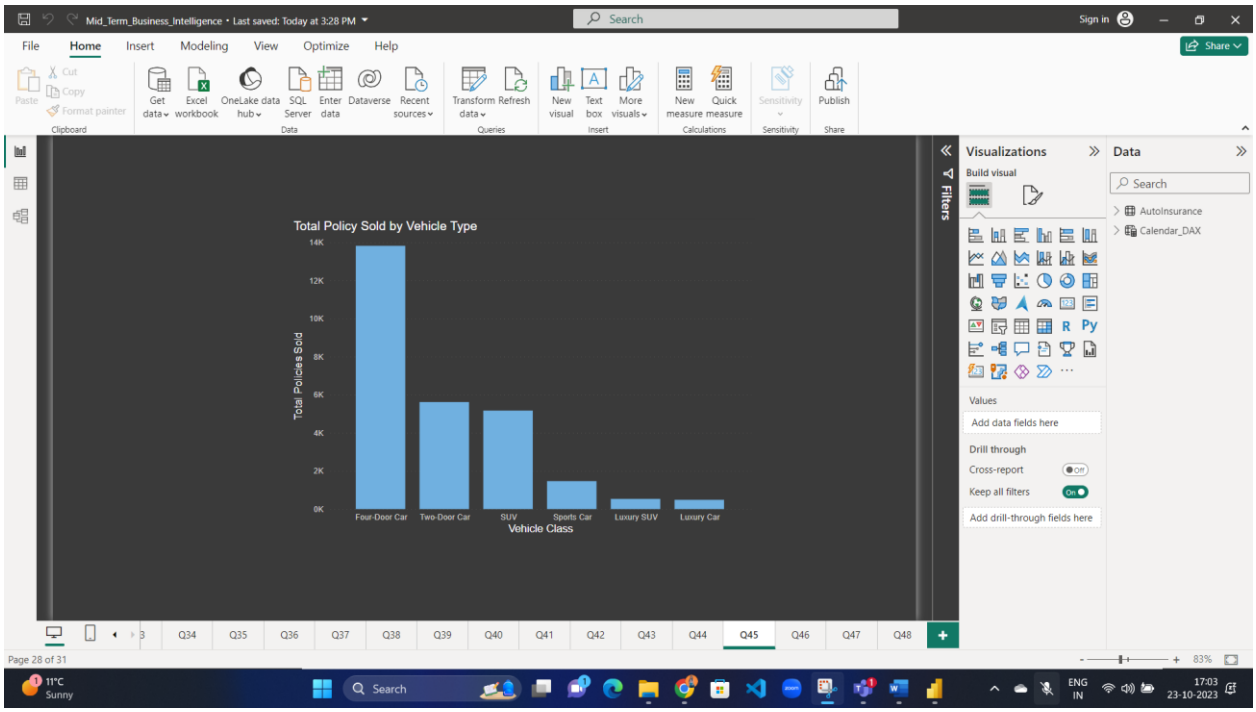



44.Line Chart: Average Claim Amount by Sales Channel across each Month-Year.

1. Import your data with columns for "Month-Year," "Sales Channel," and "Claim Amount."

2. Create a Line Chart visual.

3. Configure the visual with "Month-Year" on the X-axis and "Sales Channel" in the Legend section.

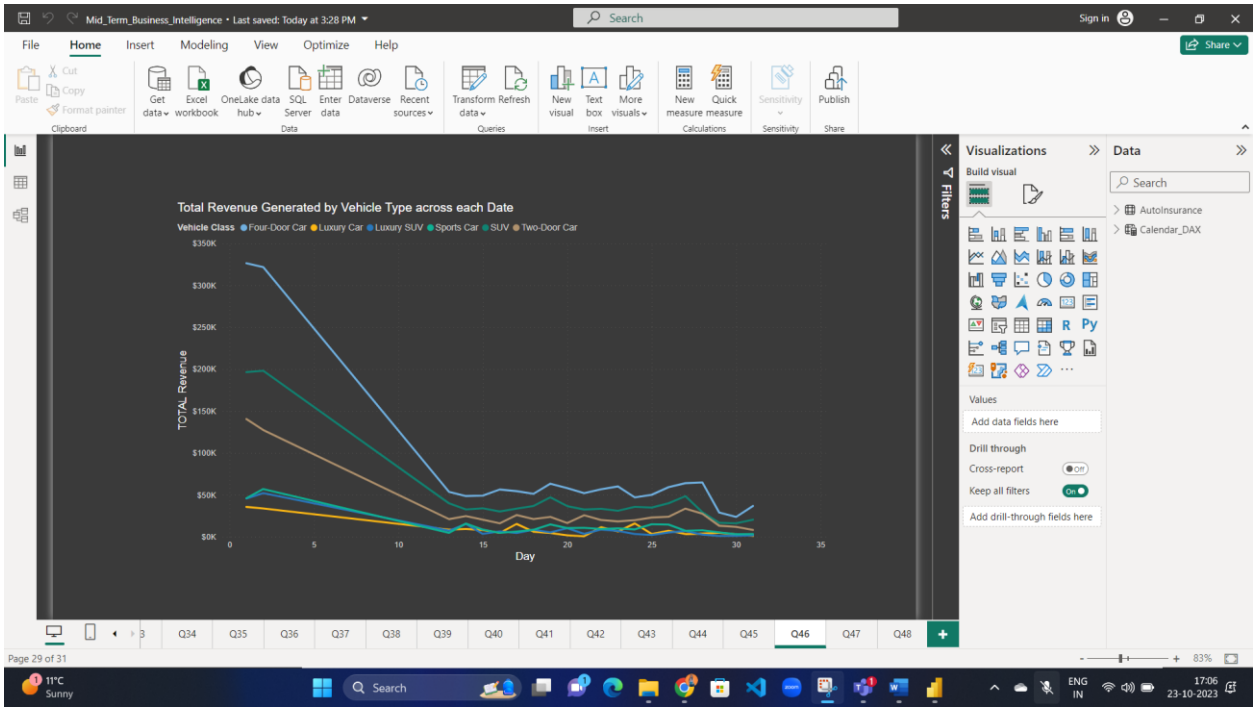


45.Bar Chart: Total Policy Sold by Vehicle Type.

1. Import your data with columns for "Vehicle Type" and "Policies Sold."

2. Create a Bar Chart visual.

3. Configure the visual with "Vehicle Type" on the X-axis.

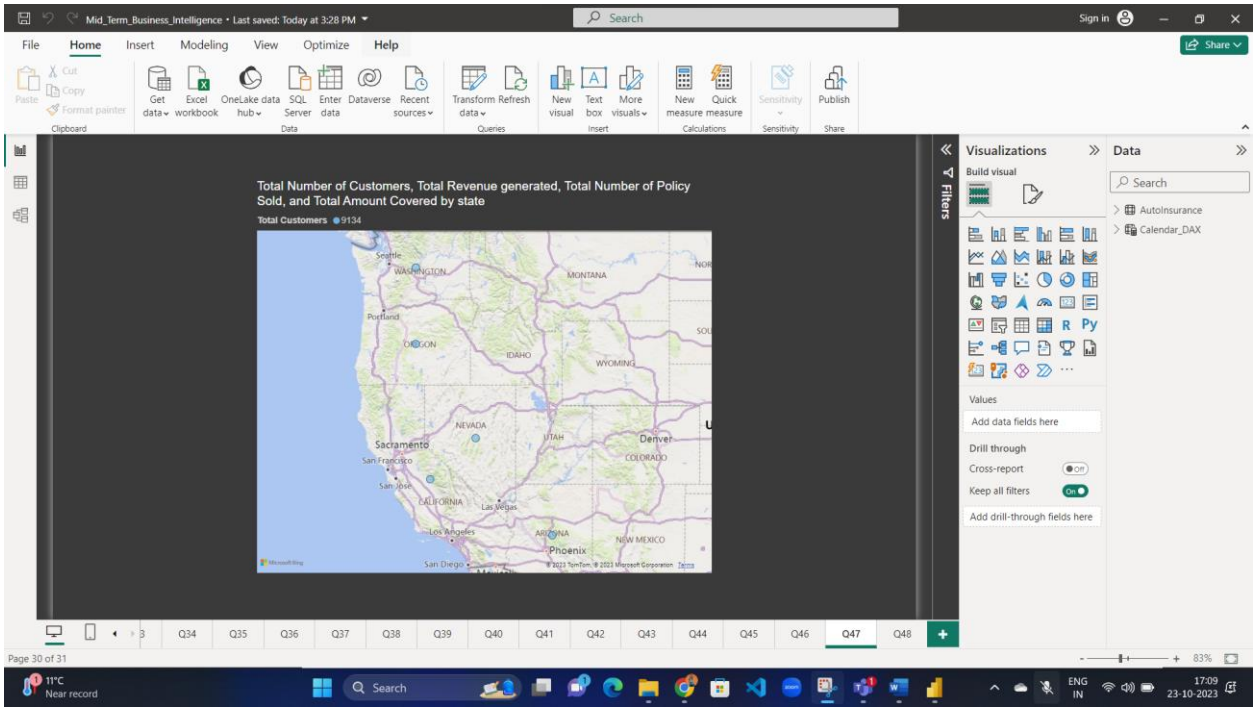



46.Line Chart: Total Revenue Generated by Vehicle Type across each Date.

1. Import your data with columns for "Date," "Vehicle Type," and "Revenue."

2. Create a Line Chart visual.

3. Configure the visual with "Date" on the X-axis and "Vehicle Type" in the Legend section.

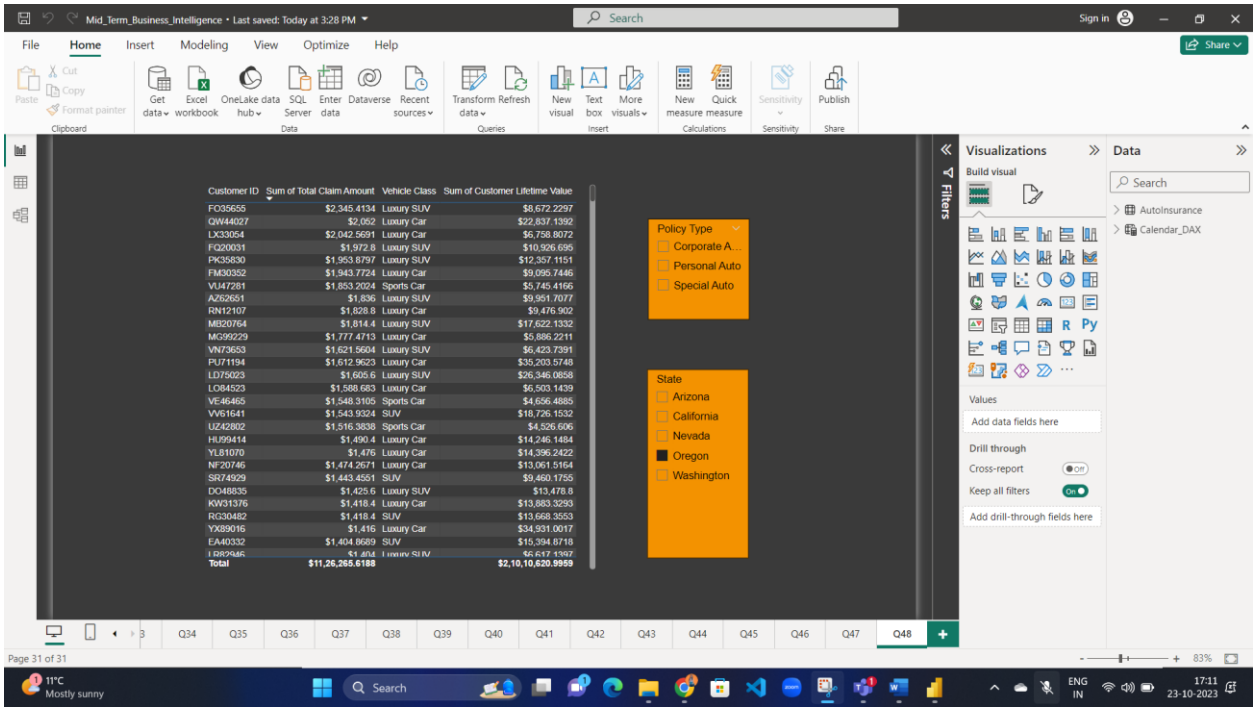



47.Map Visual: Display State wise Total Number of Customers, Total Revenue generated, Total Number of Policy Sold, and Total Amount Covered.

1. Import your data with a table containing state-wise data, including "State," "Total Customers," "Total Revenue," "Total Policies Sold," and "Total Amount Covered."

2. Create a Map visual.

3. Configure the visual by assigning "State" to the location and the other metrics to values (e.g., color, size, labels) as needed.




| Customer ID | Sum of Total Claim Amount | Vehicle Class | Sum of Customer Lifetime Value |
|---|---|---|---|
| FO35655 | $2,345.4134 | Luxury SUV | $8,672.2297 |
| QW44027 | $2,052 | Luxury Car | $22,837.1392 |
| LX33054 | $2,042.5691 | Luxury Car | $6,758.8072 |
| FQ20031 | $1,972.8 | Luxury SUV | $10,926.695 |
| PK35830 | $1,953.8797 | Luxury SUV | $12,357.1151 |
| FM30352 | $1,943.7724 | Luxury Car | $9,095.7446 |
| VU47281 | $1,853.2024 | Sports Car | $5,745.4166 |
| AZ62651 | $1,836 | Luxury SUV | $9,951.7077 |
| RN12107 | $1,828.8 | Luxury Car | $9,476.902 |
| MB20764 | $1,814.4 | Luxury SUV | $17,622.1332 |
| MG99229 | $1,777.4713 | Luxury Car | $5,886.2211 |
| VN73653 | $1,621.5604 | Luxury SUV | $6,423.7391 |
| PU71194 | $1,612.9623 | Luxury Car | $35,203.5748 |
| LD75023 | $1,605.6 | Luxury SUV | $26,346.0858 |
| LO84523 | $1,588.683 | Luxury Car | $6,503.1439 |
| VE46465 | $1,548.3105 | Sports Car | $4,656.4885 |
| VV61641 | $1,543.9324 | SUV | $18,726.1532 |
| UZ42802 | $1,516.3838 | Sports Car | $4,526.606 |
| HU99414 | $1,490.4 | Luxury Car | $14,246.1484 |
| YL81070 | $1,476 | Luxury Car | $14,396.2422 |
| NF20746 | $1,474.2671 | Luxury Car | $13,061.5164 |
| SR74929 | $1,443.4551 | SUV | $9,460.1755 |
| DO48835 | $1,425.6 | Luxury SUV | $13,478.8 |
| KW31376 | $1,418.4 | Luxury Car | $13,883.3293 |
| RG30482 | $1,418.4 | SUV | $13,668.3553 |
| YX89016 | $1,416 | Luxury Car | $34,931.0017 |
| EA40332 | $1,404.8689 | SUV | $15,394.8718 |
| Total | $11,26,265.6188 | | $2,10,10,620.9959 |

49.Table Visual: Display the granular level information's.

1. Ensure your granular-level data is imported into Power BI.

2. Create a Table visual.

3. Configure the visual by adding the specific fields or information you want to display in the table.

4. Optionally, format the table for better visual presentation.

5. Interact with the Table visual to view and explore the granular-level data.

6. Save and share your Power BI report to effectively communicate granular details from your dataset.