IBAT COLLEGE DUBLIN

OPTION 1: DATA EXPLORATION, VISUALISATION AND DATA ANALYSIS

Lecturer's name:	John Rowley
STUDENT NAME:	Aida Zadgari
STUDENT ID:	22118966
Course:	Diploma in Predictive Data Analytics

Table of Contents

Table of Figures

Chapter 1: Introduction

Data set definition

The term data set refers to a file that contains one or more records. The record is the basic unit of information. The term field refers to a specific portion of a record used for a particular category of data, such as an employee's name or department. Any named group of records is called a data set. Data sets can hold information such as medical records or insurance records, to be used by a program running on the system. Data sets are also used to store information needed by applications or the operating system itself, such as source programs, macro libraries, or system variables or parameters (IBM, 2021a).

What are the different types of data sets?

There are different types of data sets One type is numerical. It has data represented with numbers instead of words. Categorical data set is another and it displays data based on characteristics or qualities of an item. Bivariate data sets consist of two variable that have a relationship. Multivariate data sets have three or more variables that all depend on each other. Correlation data sets are data that have 1 of 3 relationships. Positive relationships , negative relationships, zero relationships occur when the variable have no effect on each other (Wells, 2022).

The purpose of dataset

The purpose of a data set is organizing the collected data so that is easier to understand and places the data into columns and rows for comparison (Wells, 2022).

Predictive data analytics

Understanding Predictive Analytics

Predictive modeling is also known as predictive analytics. Generally, the term "predictive modelling" is favoured in academic settings, while "predictive analytics" is the preferred term for commercial applications of predictive modelling (Ali, 2020).

The term predictive analytics refers to the use of statistics and modeling techniques to make predictions about future outcomes and performance. Predictive analytics looks at current and historical data patterns to determine if those patterns are likely to emerge again. This allows businesses and investors to adjust where they use their resources to take advantage of possible future events. Predictive analysis can also be used to improve operational efficiencies and reduce risk. On the whole, Predictive analytics is a decision-making tool in a variety of industries (Halton, 2021).

Successful use of predictive analytics depends heavily on unfettered access to sufficient volumes of accurate, clean and relevant data.(Ali, 2020)

Top 5 Types of Predictive Models

Predictive modeling techniques have been perfected over time. As we add more data, more muscular computing, AI and machine learning and see overall advancements in analytics, we're able to do more with these models. The top five predictive analytics models are:

1. Classification model: Considered the simplest model, it categorizes data for simple and direct query response. An example use case would be to answer the question "Is this a fraudulent transaction?"

2. Clustering model: This model nests data together by common attributes. It works by grouping things or people with shared characteristics or behaviours and plans strategies for each group at a larger scale. An example is in determining credit risk for a loan applicant based on what other people in the same or a similar situation did in the past.

3. Forecast model: This is a very popular model, and it works on anything with a numerical value based on learning from historical data. For example, in answering how much lettuce a restaurant should order next week or how many calls a customer support agent should be able to handle per day or week, the system looks back to historical data.

4. Outliers model: This model works by analysing abnormal or outlying data points. For example, a bank might use an outlier model to identify fraud by asking whether a transaction is outside of the customer's normal buying habits or whether an expense in a given category is normal or not. For example, a $1,000 credit card charge for a washer and dryer in the cardholder's preferred big box store would not be alarming, but $1,000 spent on designer clothing in a location where the customer has never charged other items might be indicative of a breached account.

5. Time series model: This model evaluates a sequence of data points based on time. For example, the number of stroke patients admitted to the hospital in the last four months is used to predict how many patients the hospital might expect to admit next week, next month or the rest of the year. A single metric measured and compared over time is thus more meaningful than a simple average (Ali, 2020).

Chapter 2: Business and Data understanding

The first step in CRISP-DM is **Business Understanding OR Organizational Understanding**, this step is crucial to a successful data mining outcome (Rowley J, 2022).

In this assignment, I imported data "insurance_claims" from Kaggle and analysed it according to some important factor in insurance industry.

The quantity, frequency of damages and the amount of claim payments, as well as the locations and factors that led to the majority of claims being paid, are the most crucial factors for insurance firms to consider. Then, we begin to concentrate on the locations where the majority of occurrences have occurred and the greatest amount of payment has been made and the possible reason behind this assumption. We started with Python and imported the CSV file as figure 1.

1. Import the CSV and Print Dataframe

```
In [12]: # No need for a directory path if in the same directory, index_col = 'Year'
         import pandas as pd
         df = pd.read_csv("insurance_claims (3).csv")

In [13]: df

Out[13]:
```

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_deductable	policy_annual_premium	umbrella_limit	insured_zip	...
0	328	48	521585	2014-10-17	OH	250/500	1000	1406.91	0	466132	...
1	228	42	342868	2006-06-27	IN	250/500	2000	1197.22	5000000	468176	...
2	134	29	687698	2000-09-06	OH	100/300	2000	1413.14	5000000	430632	...
3	256	41	227811	1990-05-25	IL	250/500	2000	1415.74	6000000	608117	...
4	228	44	367455	2014-06-06	IL	500/1000	1000	1583.91	6000000	610706	...
...	...	...	...	...	...	...	...	...	...	...	...
995	3	38	941851	1991-07-16	OH	500/1000	1000	1310.80	0	431289	...
996	285	41	186934	2014-01-05	IL	100/300	1000	1436.79	0	608177	...
997	130	34	918516	2003-02-17	OH	250/500	500	1383.49	3000000	442797	...
998	458	62	533940	2011-11-18	IL	500/1000	2000	1356.92	5000000	441714	...
999	456	60	556080	1996-11-11	OH	250/500	1000	766.19	0	612260	...

1000 rows × 40 columns

Figure 1: Import CSV file into python

As figure 2, we can see the number of rows and columns and the name of columns.

```
In [5]: # the number of rows and columns
        df.shape

Out[5]: (1000, 40)

In [6]: # to see the titles for all columns
        df.columns

Out[6]: Index(['months_as_customer', 'age', 'policy_number', 'policy_bind_date',
               'policy_state', 'policy_csl', 'policy_deductable',
               'policy_annual_premium', 'umbrella_limit', 'insured_zip', 'insured_sex',
               'insured_education_level', 'insured_occupation', 'insured_hobbies',
               'insured_relationship', 'capital-gains', 'capital-loss',
               'incident_date', 'incident_type', 'collision_type', 'incident_severity',
               'authorities_contacted', 'incident_state', 'incident_city',
               'incident_location', 'incident_hour_of_the_day',
               'number_of_vehicles_involved', 'property_damage', 'bodily_injuries',
               'witnesses', 'police_report_available', 'total_claim_amount',
               'injury_claim', 'property_claim', 'vehicle_claim', 'auto_make',
               'auto_model', 'auto_year', 'fraud_reported', '_c39'],
              dtype='object')
```

Figure 2: overall look to the data 1

With the help of df.max() in figure 3, we are able to take a broad view and see the most significant data for each column. For example, we have a customer with 479 months, and the oldest customer is 64. The highest level of education is a Ph.D., and Springfield city has experienced the most accidents. Males are more likely to have accidents than females, the maximum number of involved vehicles in accident is 4, most of the cars are Volkswagen and etc. Also, we give more attention to the month of accident later, as the information is just belong to year 2015.

```
In [32]: df.max()

Out[32]: months_as_customer                        479
         age                                        64
         policy_number                          999435
         policy_bind_date                   2015-02-22
         policy_state                               OH
         policy_csl                           500/1000
         policy_deductable                        2000
         policy_annual_premium                 2047.59
         umbrella_limit                       10000000
         insured_zip                            620962
         insured_sex                              MALE
         insured_education_level                   PhD
         insured_occupation           transport-moving
         insured_hobbies                       yachting
         insured_relationship                     wife
         capital-gains                          100500
         capital-loss                                0
         incident_date                      2015-03-01
         incident_type                   Vehicle Theft
         collision_type                 Side Collision
         incident_severity              Trivial Damage
         authorities_contacted                  Police
         incident_city                     Springfield
         incident_location              9988 Rock Ridge
         incident_hour_of_the_day                   23
         number_of_vehicles_involved                 4
         property_damage                           YES
         bodily_injuries                             2
         witnesses                                   3
         police_report_available                   YES
         total_claim_amount                     114920
         injury_claim                            21450
         property_claim                          23670
         vehicle_claim                           79560
         auto_make                          Volkswagen
         auto_model                                 X6
         auto_year                                2015
         fraud_reported                              Y
         _c39                                      NaN
         yearincident                             2015
         monthincident                               3
         yearpolicy                               2015
         monthpolicy                                12
         dtype: object
```

Figure 3: overall look to the data 2

We also used descriptive statistic to explain our data more. The policy annual premium mean is 1256 with the max of 2047 and the min of 433. The total claim amount has a mean of 52761, the max of 21450 and the min of 100. The mean of total claim is much more than the mean of total premium and it could be a warning for insurer.

```
In [33]: #descriptive statistic: we can have an overall look to the data
         df[['policy_annual_premium','total_claim_amount','injury_claim','property_claim','vehicle_claim']].describe().round(2)

Out[33]:
```

	policy_annual_premium	total_claim_amount	injury_claim	property_claim	vehicle_claim
count	1000.00	1000.00	1000.00	1000.00	1000.00
mean	1256.41	52761.94	7433.42	7399.57	37928.95
std	244.17	26401.53	4880.95	4824.73	18886.25
min	433.33	100.00	0.00	0.00	70.00
25%	1089.61	41812.50	4295.00	4445.00	30292.50
50%	1257.20	58055.00	6775.00	6750.00	42100.00
75%	1415.70	70592.50	11305.00	10885.00	50822.50
max	2047.59	114920.00	21450.00	23670.00	79560.00

Figure 4: overall look to the data 3

Chapter 3: Data preparation, modelling and evaluation

3.1) Data preparation

For the data preparation, I used MySQL Workbench, created a SCHEMAS first and import the data into the table, then read the file (Figure 5).

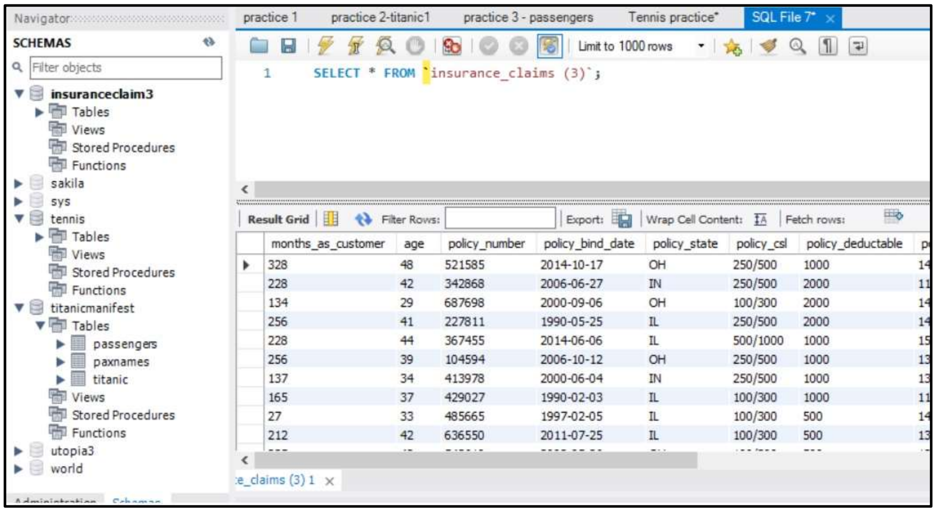

Figure 5: Import data into MySQL Workbench

There are some missing data in columns "property_damage" and "police_report_available" as figure 6, I filled the "?" cells in "property_damage" according to "property_claim" column. Wherever we had amount for 'property_claim, I filled with yes, and else =NO. No missing data was observed in other columns (Figure 7).

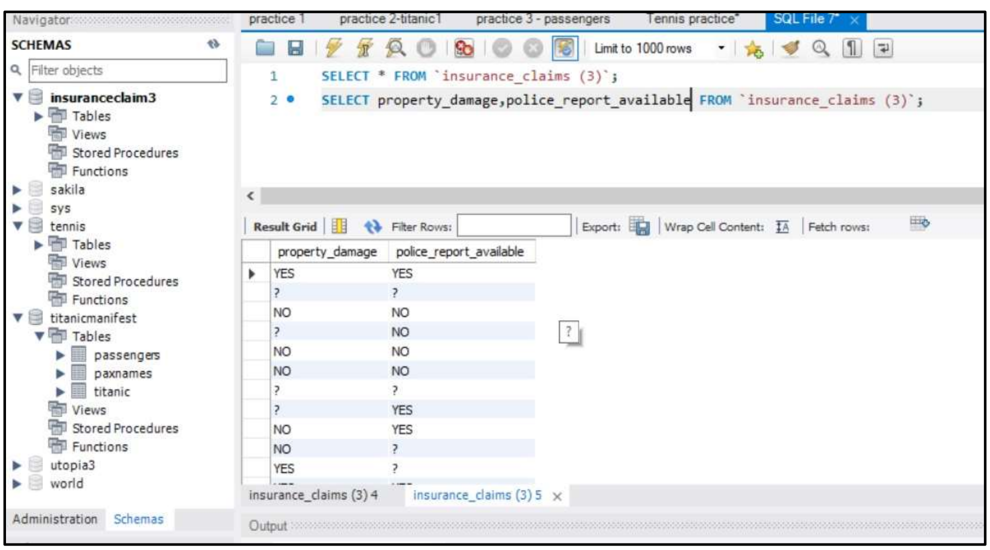

Figure 6: Missing data

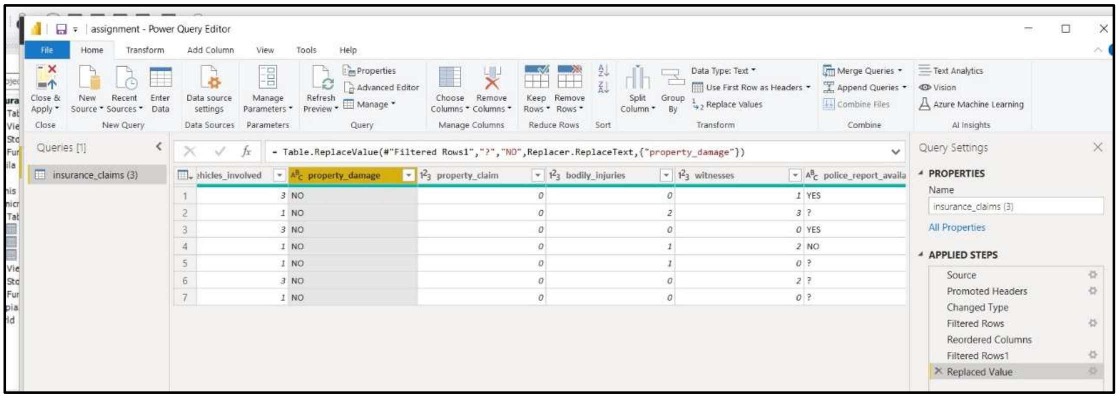

Figure 7: data cleaning

3.2) Modelling and Evaluation

Now, it is time to model our data and do analysis, as we discussed before, we are going to focus on which place has the most claim amount and the most accident with considering the policy premiums at the same time. First, we set index according to state as figure 8 and then grouped the data according to state and select some columns.

```
In [19]: #set incident_state as an index
df.set_index('incident_state', inplace = True)
df
```

Out[19]:

incident_state	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_csl	policy_ded
SC	328	48	521585	2014-10-17	OH	250/500	
VA	228	42	342868	2006-06-27	IN	250/500	
NY	134	29	687698	2000-09-06	OH	100/300	
OH	256	41	227811	1990-05-25	IL	250/500	
NY	228	44	367455	2014-06-06	IL	500/1000	
...	...	...	...	...	...	...	
NC	3	38	941851	1991-07-16	OH	500/1000	
SC	285	41	186934	2014-01-05	IL	100/300	
NC	130	34	918516	2003-02-17	OH	250/500	
NY	458	62	533940	2011-11-18	IL	500/1000	
WV	456	60	556080	1996-11-11	OH	250/500	

Figure 8: Index-Python

According to figure 9 most "witnesses" belong to NY state and the most "involved vehicles in accidents" belong to NY too. The maximum “total_claim_amount” has been paid in NY. We can conclude that because there is more accident there, so there are more witnesses and more paid claim amount. We can also prove this result in figure 10. Then, we can determine which NY city has the most accidents. Additionally, we may examine "incident_city" in more details such as "sex", “age”, “education” and etc and determine why the majority of accidents have occurred there.

```
In [20]: # grouping by State and see how many witnessess and how many involved vehicle in each state
         df_claims = df.groupby("incident_state")["witnesses","number_of_vehicles_involved",
                                                   "total_claim_amount","policy_annual_premium"].sum()
         df_claims

         C:\Users\Aida\AppData\Local\Temp\ipykernel_23588\3691469565.py:2: FutureWarning: Indexing with
         ted to a tuple of keys) will be deprecated, use a list instead.
           df_claims = df.groupby("incident_state")["witnesses","number_of_vehicles_involved",
Out[20]:
```

incident_state	witnesses	number_of_vehicles_involved	total_claim_amount	policy_annual_premium
NC	154	200	5421230	136251.24
NY	388	497	14773770	328573.78
OH	43	49	1144720	27841.66
PA	42	59	1487150	37132.90
SC	371	454	13538150	307657.91
VA	168	188	5499770	142291.77
WV	321	392	10897150	276656.89

Figure 9: grouping state- python

Considering the amount of claims and premium together as claim to premium ratio in figure 10, we can confirm our first conclusion that NY state has the most loss, the reason we compared the total claim amount for each state with total premium is that sometimes we cannot decide according to claims without considering the premiums, as it is possible that some insured or in this case states has a high loss but also a huge amount of premium, so we can ignore them as a risky insured, when we compare these both amount at the same time, we can have a right decision and look at the claim to premium ratio and select the insured that are really risky. Now we can continue to investigate the possible reasons behind that.

```
In [21]: # compare claim amount to premium amount for each state seperatly and add it to df_claims as a new column "claimtopremium"
         # use def function

         def claimratio ():
             df_claims['claimtopremium'] = ((df_claims.total_claim_amount/df_claims.policy_annual_premium)*100).round(2)
             return df_claims
         claimratio ()
Out[21]:
```

incident_state	witnesses	number_of_vehicles_involved	total_claim_amount	policy_annual_premium	claimtopremium
NC	154	200	5421230	136251.24	3978.85
NY	388	497	14773770	328573.78	4496.33
OH	43	49	1144720	27841.66	4111.54
PA	42	59	1487150	37132.90	4004.94
SC	371	454	13538150	307657.91	4400.39
VA	168	188	5499770	142291.77	3865.14
WV	321	392	10897150	276656.89	3938.87

Figure 10: claim to premium ratio

```
In [17]: # calculate the number of claims for each state and add it to df_claims
         df.groupby('incident_state')[['total_claim_amount']].count()
Out[17]:
```

incident_state	total_claim_amount
NC	110
NY	262
OH	23
PA	30
SC	248
VA	110
WV	217

Figure 11: number of claims for each state

As I explained before, we can see the total claim amount for Columbus is the largest amount, but considering premiums we can conclude that Riverwood is the riskiest city in NY (Figure 12).

```
In [26]: # Grouping by state and city
         df_NY_group = df_state.groupby(['incident_state','incident_city'])['witnesses',"number_of_vehicles_involved",
                                                                      "total_claim_amount","policy_annual_premium"]
         # Use function for calculing the claim to premium ratio
         def claimratio2 ():
             df_NY_group['claimtopremium2'] = ((df_NY_group.total_claim_amount/df_NY_group.policy_annual_premium)*100).r
             return df_NY_group
         claimratio2 ()

         # Use highlight for max values
         df_NY_group.style.highlight_max(color = 'lightgreen', axis = 0)

C:\Users\Aida\AppData\Local\Temp\ipykernel_23588\1072776821.py:2: FutureWarning: Indexing with multiple keys (i
ted to a tuple of keys) will be deprecated, use a list instead.
  df_NY_group = df_state.groupby(['incident_state','incident_city'])['witnesses',"number_of_vehicles_involved",
Out[26]:
```

incident_state	incident_city	witnesses	number_of_vehicles_involved	total_claim_amount	policy_annual_premium	claimtopremium2
NY	Arlington	49	74	2016230	48548.340000	4153.040000
	Columbus	71	85	2758240	58672.340000	4701.090000
	Hillsdale	43	61	1705400	42312.720000	4030.470000
	Northbend	59	73	1993560	42462.570000	4694.860000
	Northbrook	45	50	1770750	38656.890000	4580.680000
	Riverwood	61	69	2153930	43451.280000	4957.120000
	Springfield	60	85	2375660	54469.640000	4361.440000

Figure 12: grouping incident city in NY

3.2.1) Analysing the NY state & dive into more details

For making some visualization we imported the data in Power BI as figure 13.

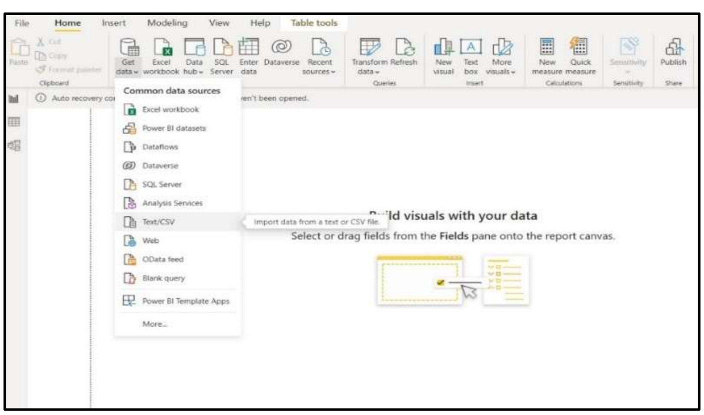

Figure 13: Import data in Power BI

We go to the report section and bring the needed data to the table from the field section, and use slicer for filtering base on incident_state (NY) and incident_city (Riverwood).

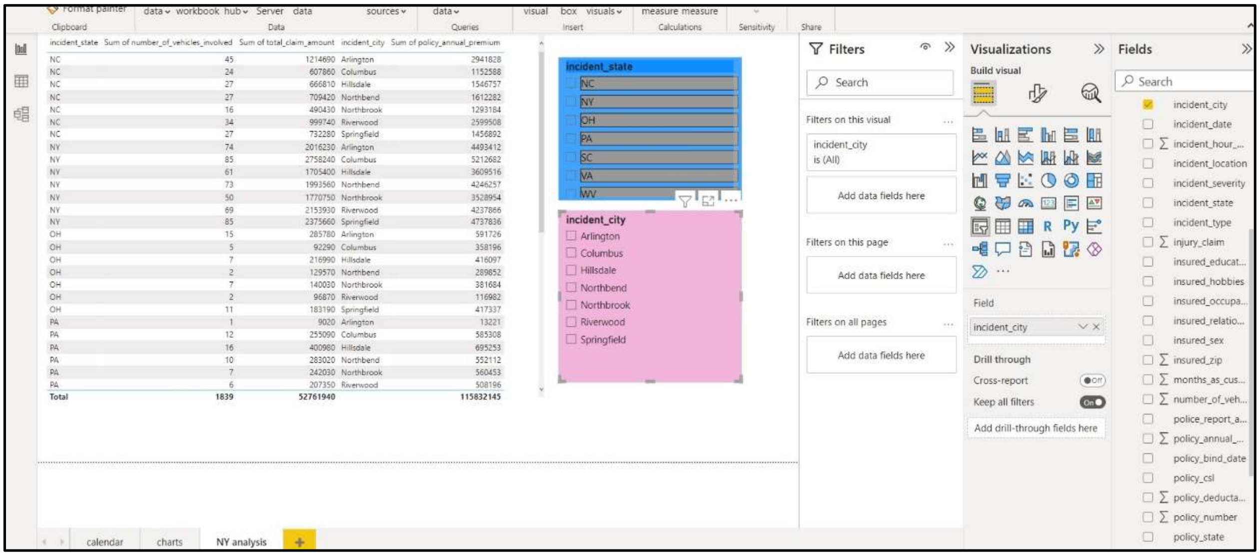

Figure 14: filtering states and cities in Power BI

Let's do some analysis for insured characteristics in NY and Riverwood, in NY the number of females are 138 and the number of males are 124 as figure 15.

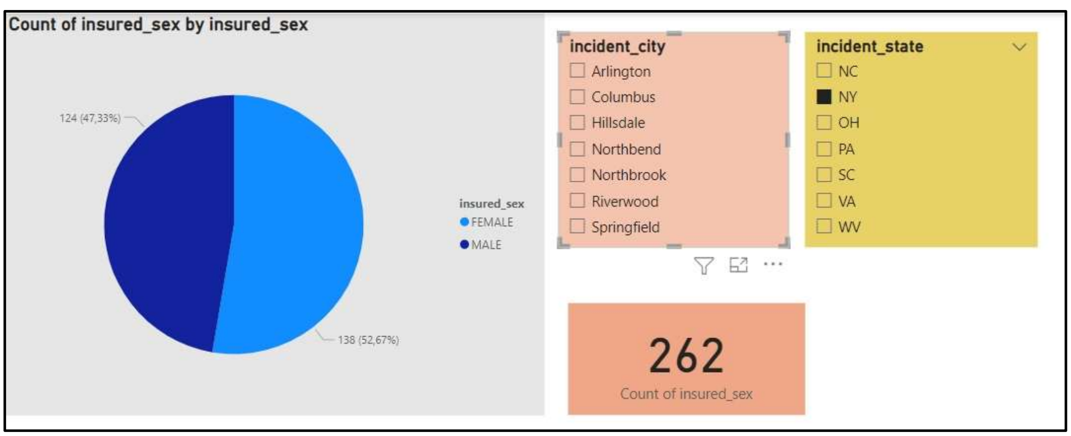

Figure 15: Power BI analysis 1

When we add the filter for incident_city, and select Riverwood, we can also see the percentage for female is 55.88% and for male is 44.12% as figure 16.

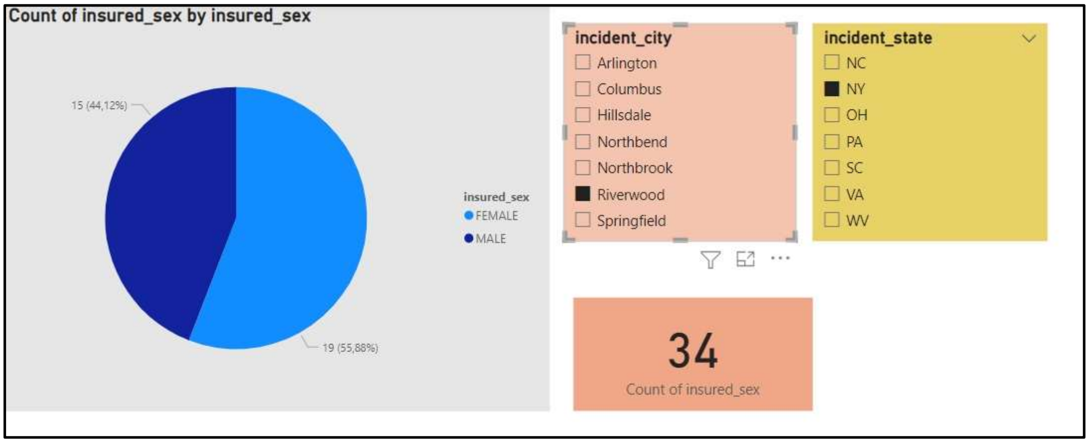

Figure 16: Power BI analysis 2

As we see, Female insured are more than male insured in accident reports, we can conclude that female are more careless but we cannot conclude it by certain, because the total claims amount and the severity of damages for male can be more than female. Figure 17 shows the total claims amount for female which is 88920 and more than male, now we can say that female makes more damages and accident in Riverwood.

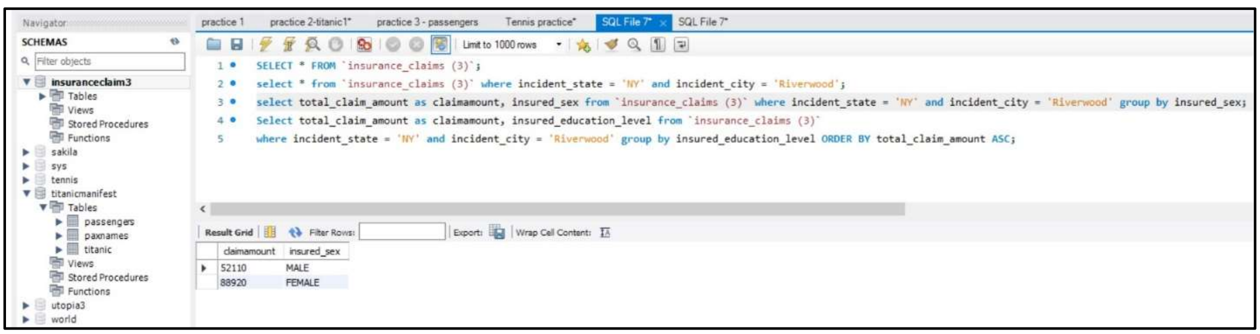

Figure 17: SQL analysis 1 - insured_sex

we can do the same process about education level and age, to get more view about what is happening in Riverwood.

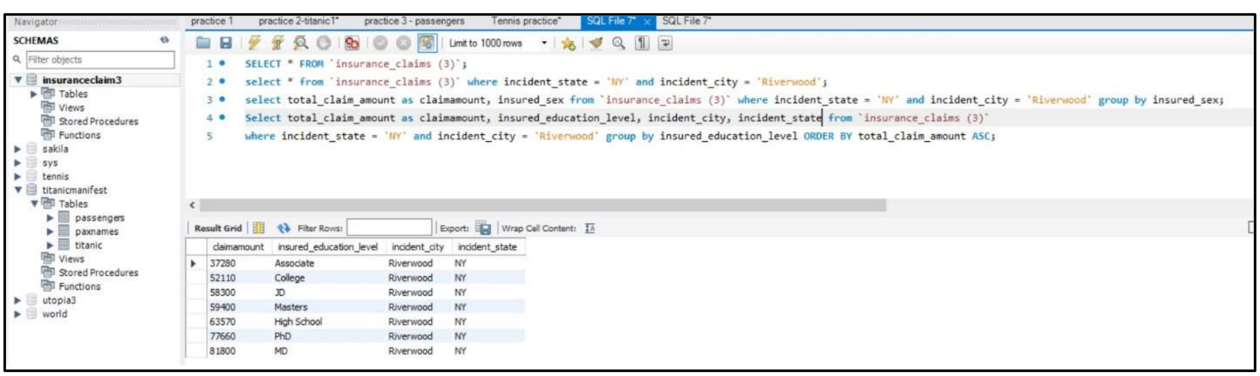

Figure 18: SQL analysis 2- education level

Considering the education level, and as we see in figure 18, insured with a high level of education (MD and PhD) have made more accident. Moreover, the age of most insured is more than 35 years old (16) and the number of insured <=35 is half of the first group (8) (figure 19).

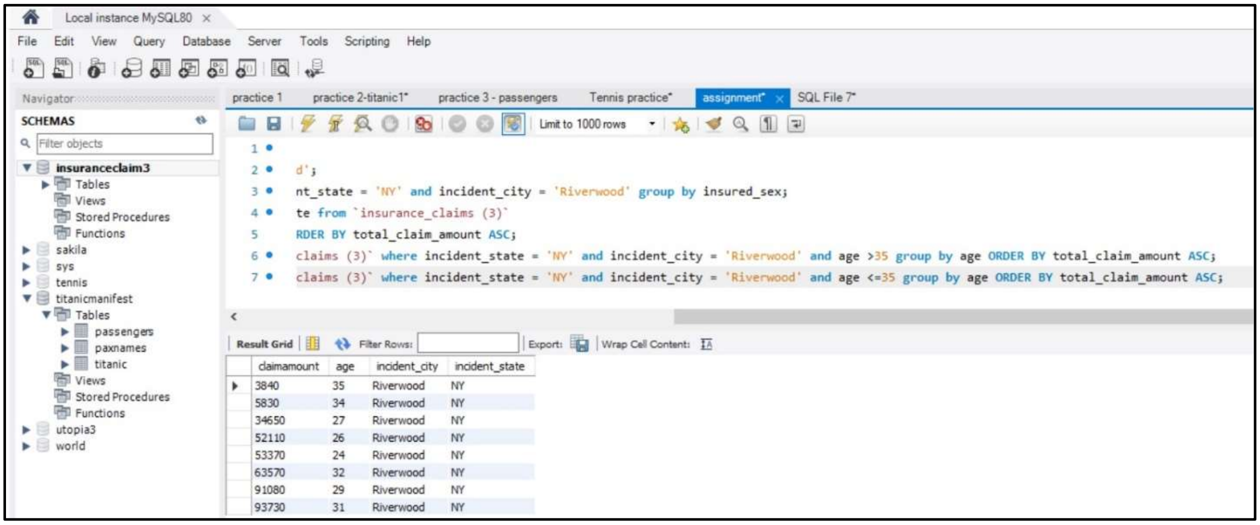

Figure 19: SQL analysis 3 - age

3.2.2) Some general analysis

How would be the relation between customer with more months and total_claim_amount?

The hypothesis for plotting the aforementioned relation is that: customers with more months (older customers) can be more sensitive to their insurance records and thus cause fewer damages. According to the figure 20, we see even insured with more months can experience significant losses, maybe we can say that there is not a direct and clear relation between these two variables but generally and after 250 months, the below "bar chart" shows a downward trend when the months of contract for insurers increase.

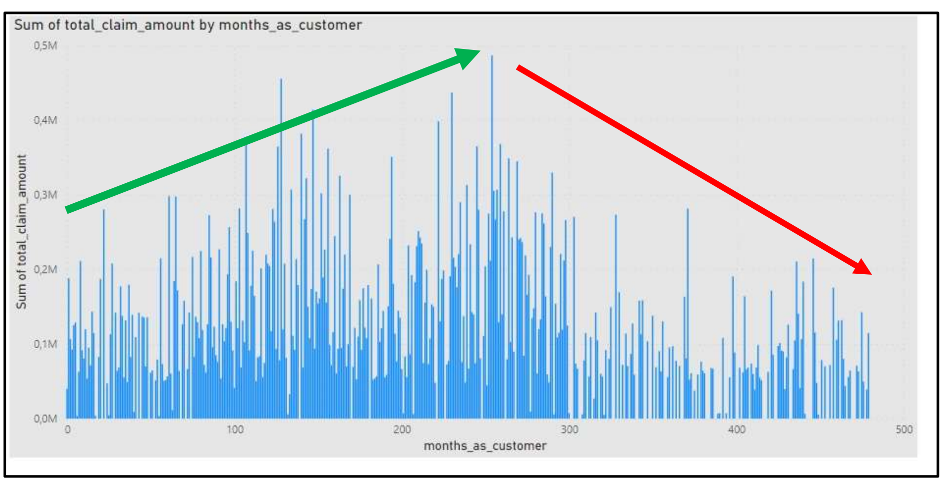

Figure 20: Power BI analysis – months as customer

Compare total amount of claims for different months

First, we should check the date format to analyze according to date. We went to the transform data, a window as figure 21 opened, checked the "Date type" and it was "Date".

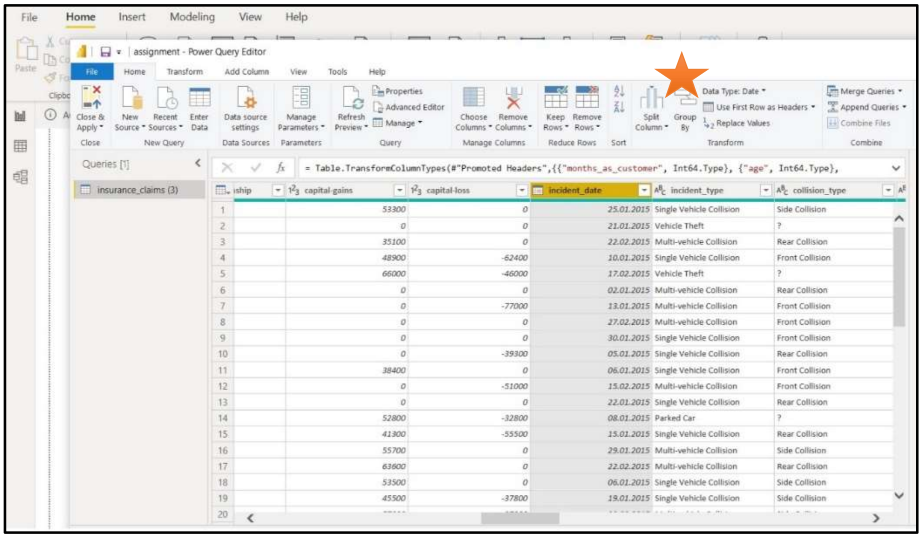

Figure 21: Power BI analysis - using DAX 1

Then, I change format of "incident date" in Data as figure 22.

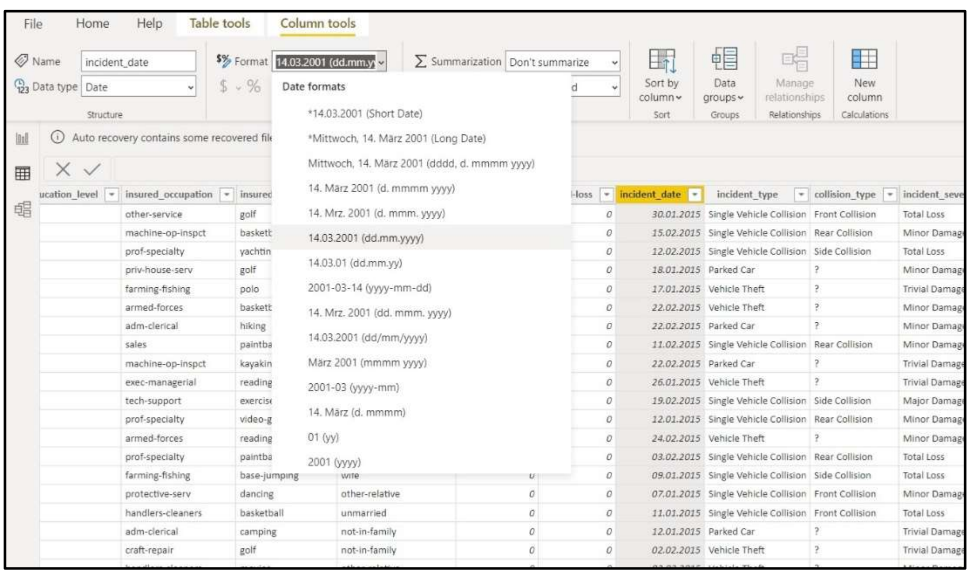

Figure 22: Power BI analysis - using DAX 2

We selected table from visualization part and then go to the Fields part to select the considered data, in this case we selected incident city, incident date, state and incident type and incident hour to understand what is going on (see figure 23).

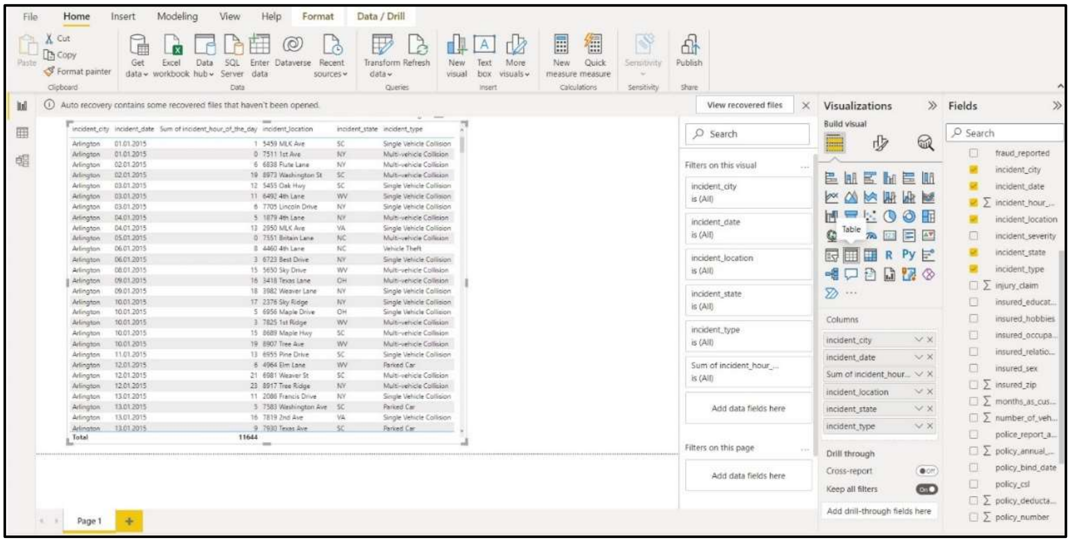

Figure 23:Power BI analysis - using DAX 3

Then we wrote formula as figure 24 to separate the day and month and year and made a new table to create “Mycalendar” and mark it as date table.

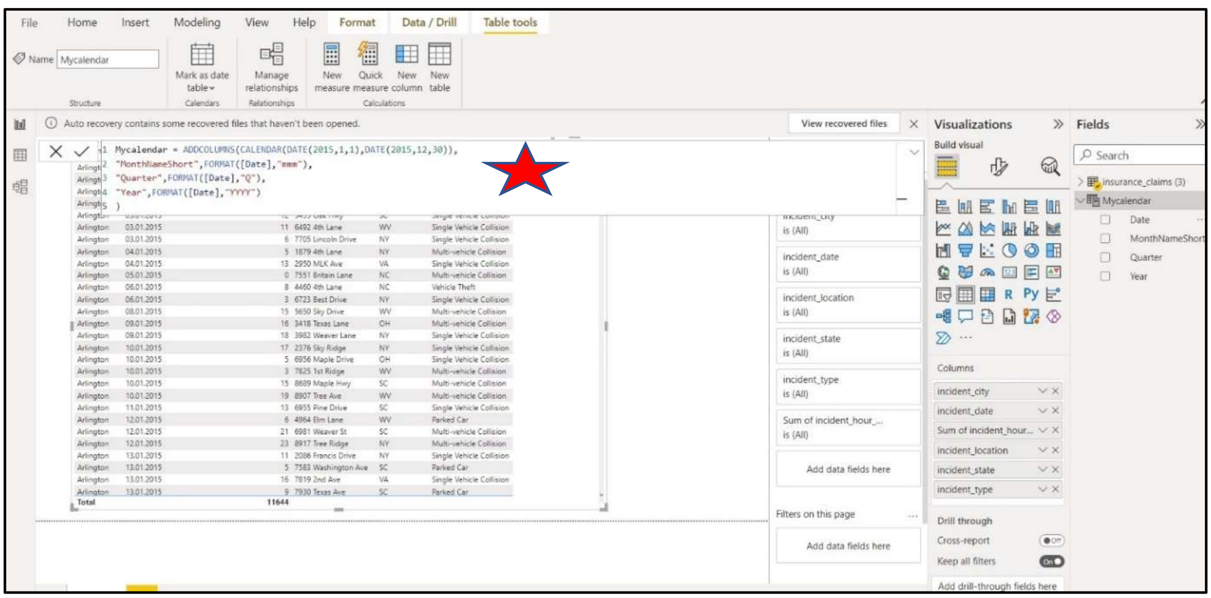

Figure 24: Power BI analysis - using DAX 4

Then we went to the modelling section and built a relation between Mycalendar and main data and join them together as figure 25.

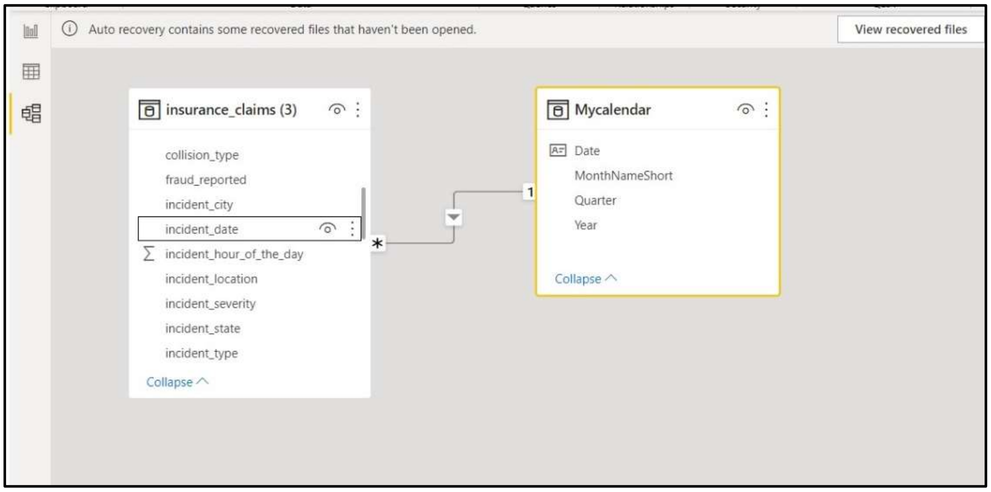

Figure 25: Power BI analysis - using DAX 5

Back to report and add month, quarter and year to the table (Figure 26).

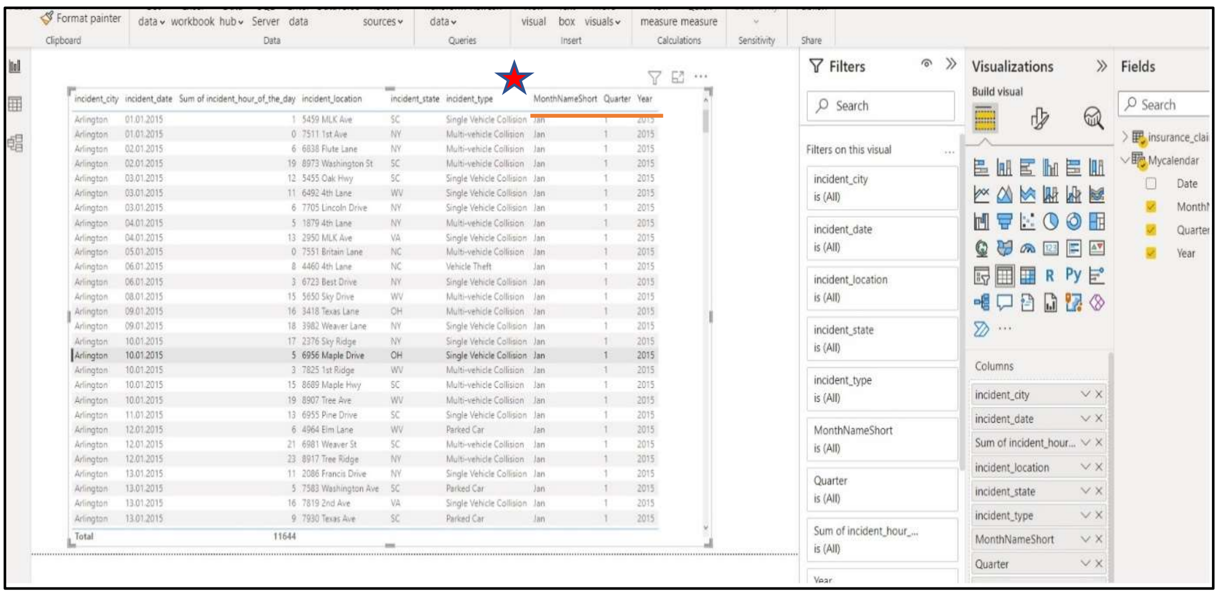

Figure 26: Power BI analysis - using DAX 6

Now, we can analyze the data according to the months. We just have data for three first months and after selecting pie chart, I used smart narratives to add explanation to the chart. We cannot compare these 3 months together, as the data for third month is not complete, then we just compare 2 first months together and January has the first level with 52.62% of total (Figure 27).

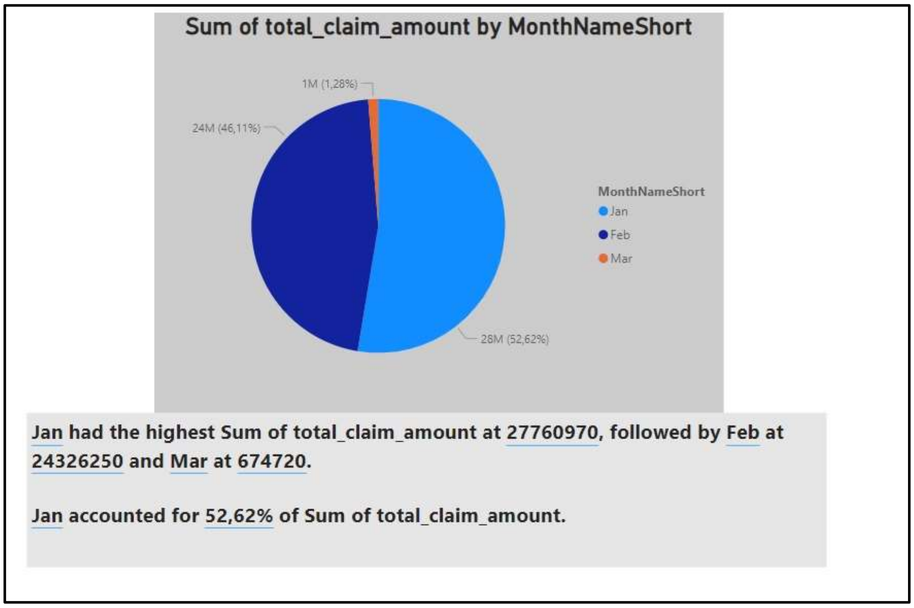

Figure 27: Power BI analysis - using DAX 7

Now, we are going to have a forecast for the remained months by the end of year 2015, we selected line chart and incident date and total claim amount. We can see the forecasting amount by clicking everywhere in the chart and we can also export the data or show it as a table (Figure 28).

Figure 28: Power BI analysis - using DAX 8

References

1. Ali, R. (2020) *Predicting Your Business's Future*, *Oracle NetSuite*. Available at: https://www.netsuite.com/portal/resource/articles/financial-management/predictive-modeling.shtml (Accessed: 12 October 2022).
2. Halton, C. (2021) *Predictive Analytics Definition*, *Investopedia*. Available at: https://www.investopedia.com/terms/p/predictive-analytics.asp (Accessed: 9 October 2022).
3. IBM (2021a) *IBM Documentation*. Available at: https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/en/zos-basic-skills?topic=more-what-is-data-set (Accessed: 9 October 2022).
4. IBM (2021b) *IBM Documentation*. Available at: https://prod.ibmdocs-production-dal-6099123ce774e592a519d7c33db8265e-0000.us-south.containers.appdomain.cloud/docs/en/spss-modeler/saas?topic=dm-crisp-help-overview (Accessed: 12 October 2022).
5. Wells, D. (2022) *Data Set Types & Examples | What Is a Data Set in Math? - Video & Lesson Transcript*, *study.com*. Available at: https://study.com/learn/lesson/data-set-in-math-types-examples.html (Accessed: 9 October 2022).