Project Name : Vehicles Insurance Claim Data

Aim :

To describe and analyze and Draw the plots of all dataset

Libraries Info:

i will describe and analyze all the data with the help of Two libraries like....,

i used pandas library import pandas as pd

i used matplotlib.pyplot library import matplotlib.pyplot as plt

In [3]:
```
import pandas as pd
import matplotlib.pyplot as plt
```

I used pd.read_excel for collect the excel file with thier link from file manager

In [51]:
```
df = pd.read_excel("C:\Project Files (Shital)\Vehicle Insurance Claims Data.xlsx
df
```

2	29	687698	2000-09-06	OH	1413.14	430632
3	41	227811	1990-05-25	IL	1415.74	608117
4	44	367455	2014-06-06	IL	1583.91	610706
...	...	...	...	...	...	...
995	38	941851	1991-07-16	OH	1310.80	431289
996	41	186934	2014-01-05	IL	1436.79	608177
997	34	918516	2003-02-17	OH	1383.49	442797
998	62	533940	2011-11-18	IL	1356.92	441714
999	60	556080	1996-11-11	OH	766.19	612260

1000 rows × 22 columns

df.shape is used for show the shape of whole data such as 1000 rows and 24 columns are their

In [52]:
```
df.shape
```

Out[52]:
```
(1000, 22)
```

the len() function is used for length of whole data

In [53]:
```
print(len(df))
```

```
1000
```

the replace() function is use to replace the values

In [55]:
```
df['Policy_state'] = df['Policy_state'].replace({
    "OH":"Ohio",
    "IN":"in",
    "IL":"Illinois"


},regex=True)

df.head()
```

Out[55]:

	Age	Policy_number	Policy_bind_date	Policy_state	Policy_annual_premium	Insured_zip	Insure
0	48	521585	2014-10-17	Ohio	1406.91	466132	
1	42	342868	2006-06-27	in	1197.22	468176	
2	29	687698	2000-09-06	Ohio	1413.14	430632	
3	41	227811	1990-05-25	Illinois	1415.74	608117	
4	44	367455	2014-06-06	Illinois	1583.91	610706	

5 rows × 22 columns

the replace() function is use to replace the multiple values at a time

In [56]:
```
df['Incident_state'] = df['Incident_state'].replace({
    "OH":"Ohio",
    "IN":"in",
    "IL":"Illinois",
    "NY":"New York",
    "SC":"Sorth Carolina",
    "WV":"West Verginia",
    "VA":"Virginia",
    "NC":"North Carolina",
    "PA":"Pennsylvania",
    "OH":"Ohio",
    "IN":"in",
    "IL":"Illinois"


},regex=True)

df.head()
```

Out[56]:

	Age	Policy_number	Policy_bind_date	Policy_state	Policy_annual_premium	Insured_zip	Insure
0	48	521585	2014-10-17	Ohio	1406.91	466132	
1	42	342868	2006-06-27	in	1197.22	468176	
2	29	687698	2000-09-06	Ohio	1413.14	430632	
3	41	227811	1990-05-25	Illinois	1415.74	608117	
4	44	367455	2014-06-06	Illinois	1583.91	610706	

5 rows × 22 columns

the info() function is used for collect all info about data

In [57]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 22 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   Age                           1000 non-null   int64
 1   Policy_number                 1000 non-null   int64
 2   Policy_bind_date              1000 non-null   datetime64[ns]
 3   Policy_state                  1000 non-null   object
 4   Policy_annual_premium         1000 non-null   float64
 5   Insured_zip                   1000 non-null   int64
 6   Insured_gender                1000 non-null   object
 7   Insured_occupation            1000 non-null   object
 8   Insured_relationship          1000 non-null   object
 9   Incident_date                 1000 non-null   datetime64[ns]
 10  Incident_type                 1000 non-null   object
 11  Collision_type                1000 non-null   object
 12  Incident_severity             1000 non-null   object
 13  Incident_state                1000 non-null   object
 14  Incident_city                 1000 non-null   object
 15  Incident_location             1000 non-null   object
 16  Number_of_vehicles_involved   1000 non-null   int64
 17  Witnesses                     1000 non-null   int64
 18  Total_claim_amount            1000 non-null   int64
 19  Injury_claim                  1000 non-null   int64
 20  Property_claim                1000 non-null   int64
 21  Vehicle_claim                 1000 non-null   int64
dtypes: datetime64[ns](2), float64(1), int64(9), object(10)
memory usage: 172.0+ KB
```

the column function is used for showing the number of column

In [58]:
```
column_name = df.columns

print(column_name)
```

```
Index(['Age', 'Policy_number', 'Policy_bind_date', 'Policy_state',
       'Policy_annual_premium', 'Insured_zip', 'Insured_gender',
       'Insured_occupation', 'Insured_relationship', 'Incident_date',
       'Incident_type', 'Collision_type', 'Incident_severity',
       'Incident_state', 'Incident_city', 'Incident_location',
       'Number_of_vehicles_involved', 'Witnesses', 'Total_claim_amount',
       'Injury_claim', 'Property_claim', 'Vehicle_claim'],
      dtype='object')
```

the head() function is used to shown initial information about data of excel file

In [59]:
```
df.head(1)
```

Out[59]:

	Age	Policy_number	Policy_bind_date	Policy_state	Policy_annual_premium	Insured_zip	Insured
0	48	521585	2014-10-17	Ohio	1406.91	466132	

1 rows × 22 columns

In [60]:
```
df[['Insured_gender','Insured_relationship','Total_claim_amount']]
```

Out[60]:

	Insured_gender	Insured_relationship	Total_claim_amount
0	MALE	husband	71610
1	MALE	other-relative	5070
2	FEMALE	own-child	34650
3	FEMALE	unmarried	63400
4	MALE	unmarried	6500
...	...	...	...
995	FEMALE	unmarried	87200
996	FEMALE	wife	108480
997	FEMALE	other-relative	67500
998	MALE	wife	46980
999	FEMALE	husband	5060

1000 rows × 3 columns

the count()function is used to shown data of Insured gender such as .....,

there are 537 Female and 463 Male

In [61]:
```
df['Insured_gender'].value_counts()
```

Out[61]:
```
FEMALE    537
MALE      463
Name: Insured_gender, dtype: int64
```

the count() is used to shown data of all Policy State such as...,

the highest policy state of Ohio's count is 352

the lowest policy state of in's count is 310

In [62]:
```
df['Policy_state'].value_counts()
```

Out[62]:
```
Ohio        352
Illinois    338
in          310
Name: Policy_state, dtype: int64
```

The count() function is used to shown data of all Incident State such as...,

the highest Incidents occurs in New York state and it's count is 262

the lowest Incident occures in Ohio state and it's count is 23

In [63]:
```
df['Incident_state'].value_counts()
```

Out[63]:
```
New York          262
Sorth Carolina    248
West Verginia     217
Virginia          110
North Carolina    110
Pennsylvania       30
Ohio               23
Name: Incident_state, dtype: int64
```

In [64]:
```
Incident_severity= df.groupby("Incident_severity")

Incident_severity.size()
```

Out[64]:
```
Incident_severity
Major Damage      276
Minor Damage      354
Total Loss        280
Trivial Damage     90
dtype: int64
```

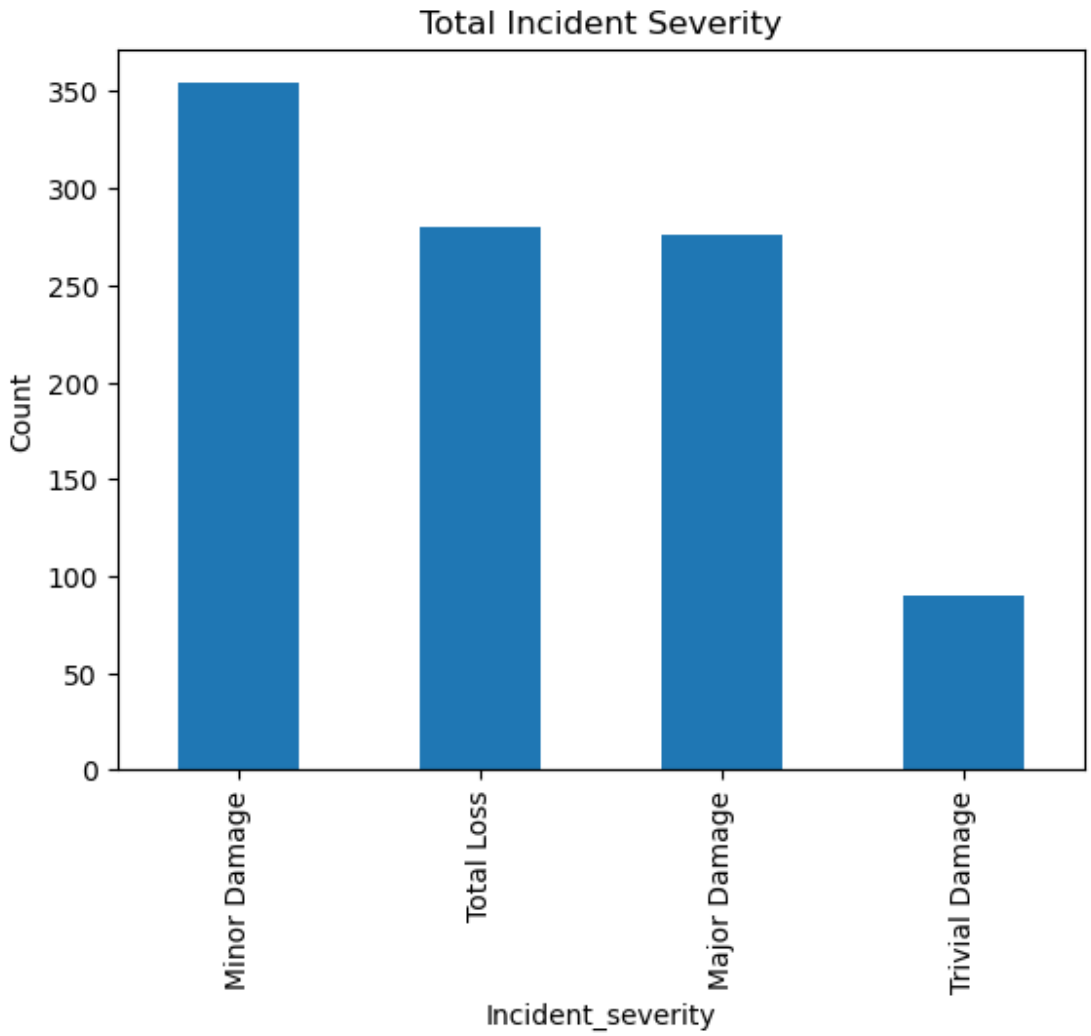
in this plot thier are Minor Damage vehicles are 350

the major Damage vehicles are 275

the total loss vehicles are 280

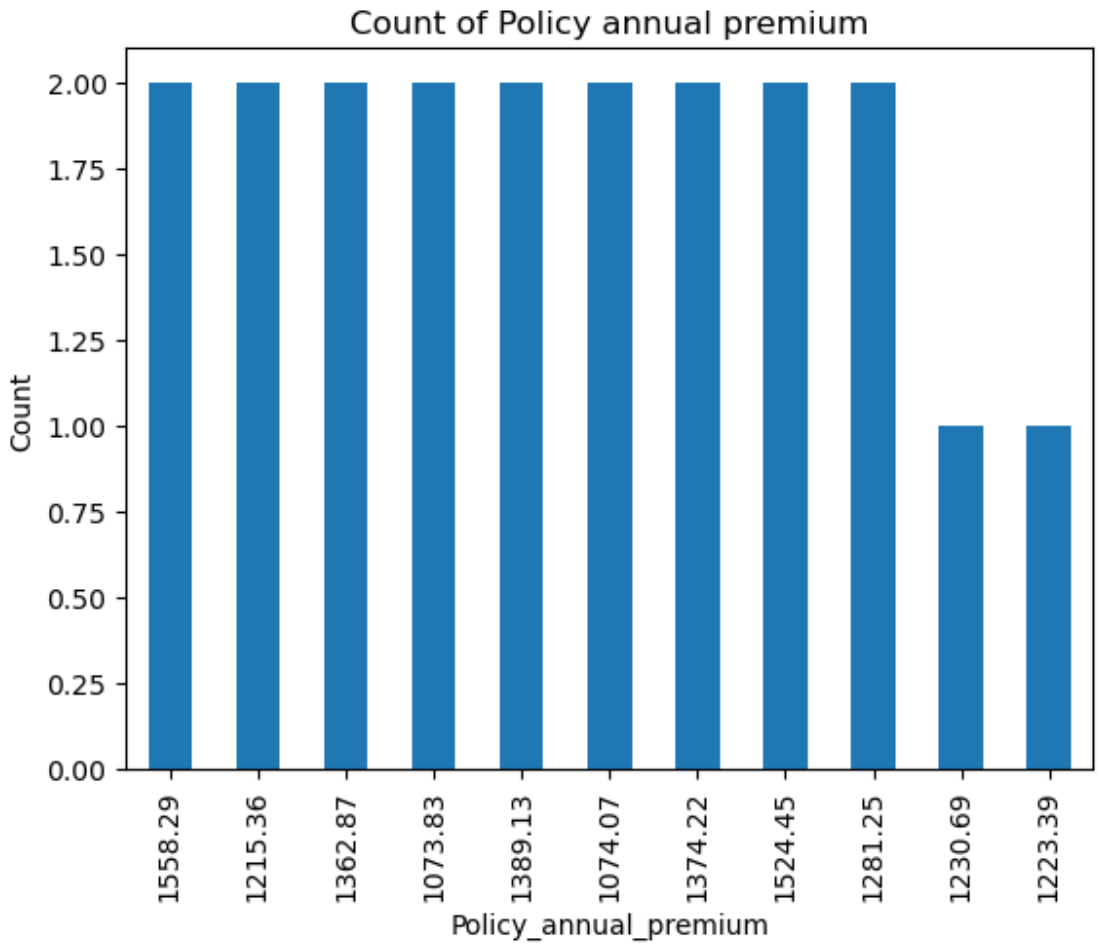
the trivial damage vehicles are 70

In [65]:
```
df['Incident_severity'].value_counts().nlargest(11).plot(kind='bar')
plt.title('Total Incident Severity')
plt.xlabel('Incident_severity')
plt.ylabel('Count')
plt.show()
```



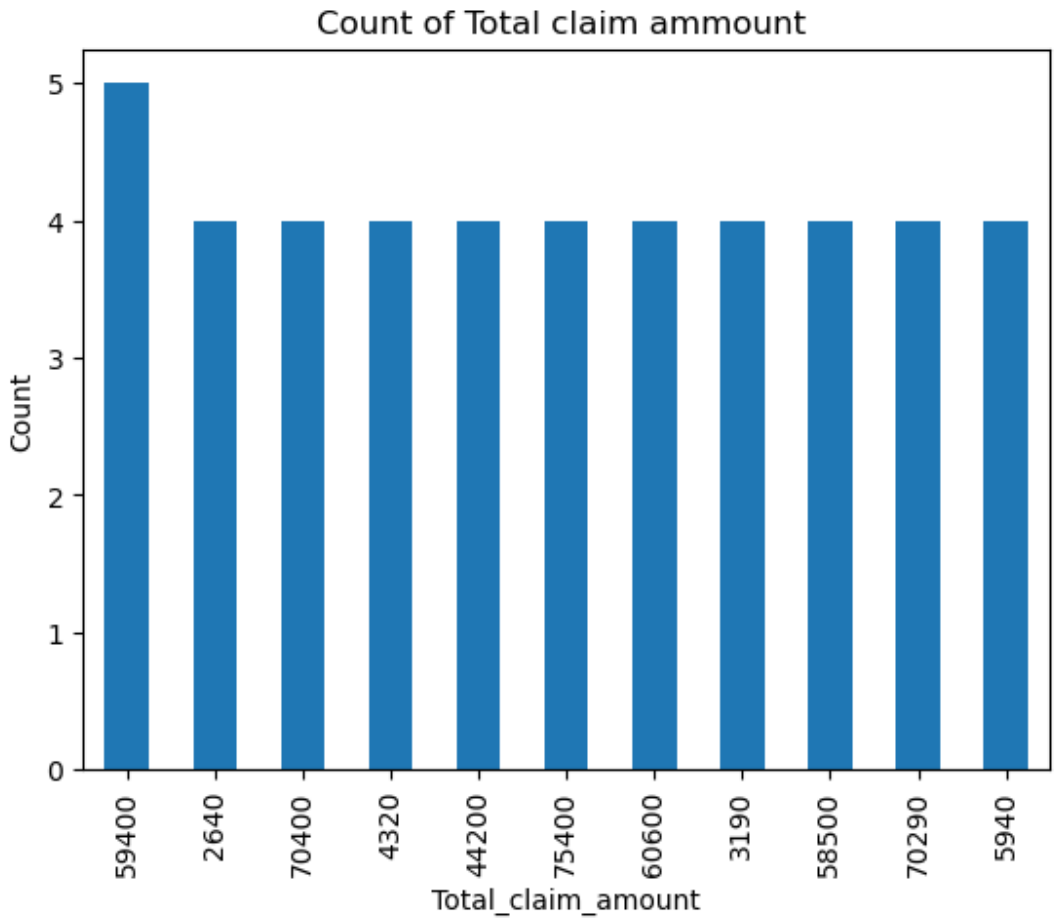
in this graph , the lowest Policy_annual_premium is 1230.69 and 1223.39 as compair to others

In [66]:
```
df['Policy_annual_premium'].value_counts().nlargest(11).plot(kind='bar')
plt.title('Count of Policy annual premium')
plt.xlabel('Policy_annual_premium')
plt.ylabel('Count')
plt.show()
```



in this graph , the highest Total_claim_amount is 59400 among the others

In [67]:
```
df['Total_claim_amount'].value_counts().nlargest(11).plot(kind='bar')
plt.title('Count of Total claim ammount ')
plt.xlabel('Total_claim_amount')
plt.ylabel('Count')
plt.show()
```



In [68]:
```
age = df.groupby("Age")

age.size()
```

Out[68]:
```
Age
19     1
20     1
21     6
22     1
23     7
24    10
25    14
26    26
27    24
28    30
29    35
30    42
31    42
32    38
33    39
34    44
35    32
36    32
37    41
38    42
39    48
40    38
41    45
42    32
43    49
44    32
45    26
46    33
47    24
48    25
49    14
50    13
51     9
52     4
53    13
54    10
55    14
56     8
57    16
58     8
59     5
60     9
61    10
62     4
63     2
64     2
dtype: int64
```

this is a group about Policy_state

in this group ...,

the highest policy state illinois is 338 and

the middlest policy state Ohio is 352 ,

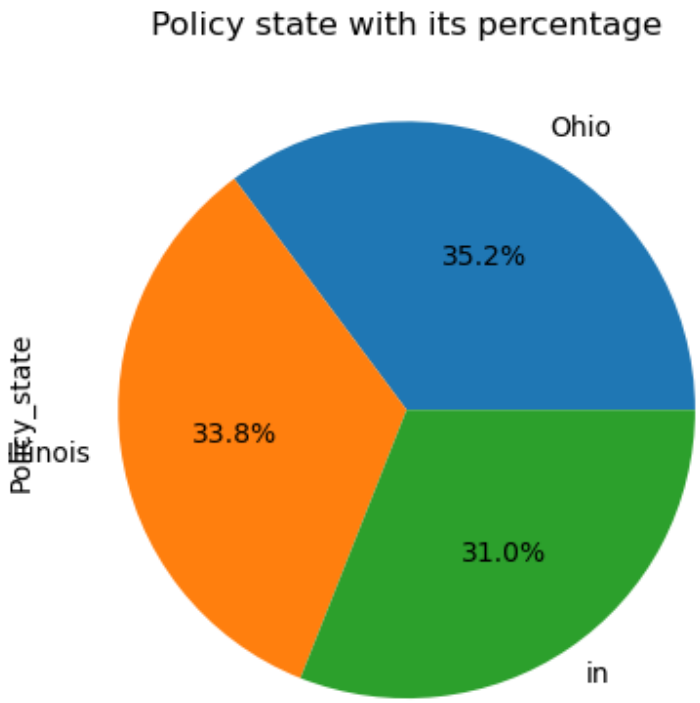
the lowest policy state in is 310

In [69]:
```
Policy_state = df.groupby("Policy_state")

Policy_state.size()
```

Out[69]:
```
Policy_state
Illinois    338
Ohio        352
in          310
dtype: int64
```

In [70]:
```
df['Policy_state'].value_counts().plot(kind='pie',autopct='%1.1f%%')
plt.title('Policy state with its percentage')
plt.show()
```



this is a group about Incident_state

in this group ...,

the highest Incident state New York is 262 and

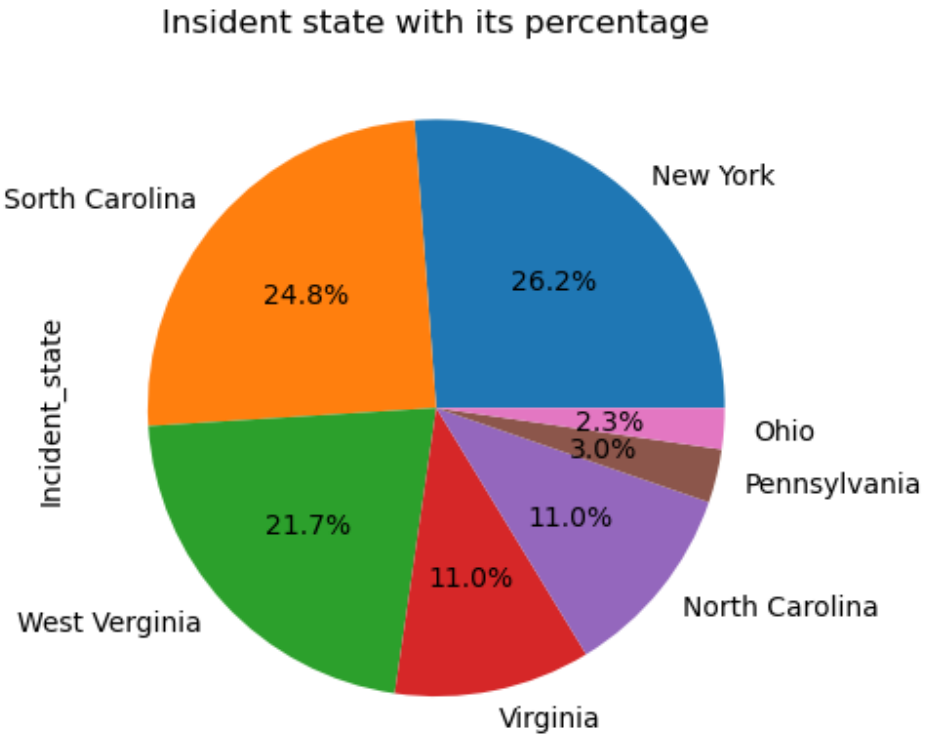
the lowest Incident state West Verginia is 217

In [71]:
```
Incident_state = df.groupby("Incident_state")

Incident_state.size()
```

Out[71]:
```
Incident_state
New York          262
North Carolina    110
Ohio               23
Pennsylvania       30
Sorth Carolina    248
Virginia          110
West Verginia     217
dtype: int64
```

In [72]:
```
df['Incident_state'].value_counts().plot(kind='pie',autopct='%1.1f%%')
plt.title('Insident state with its percentage')
plt.show()
```



this is a group about Insured_gender

in this group ...,

the Insured genders are:

the female is 537

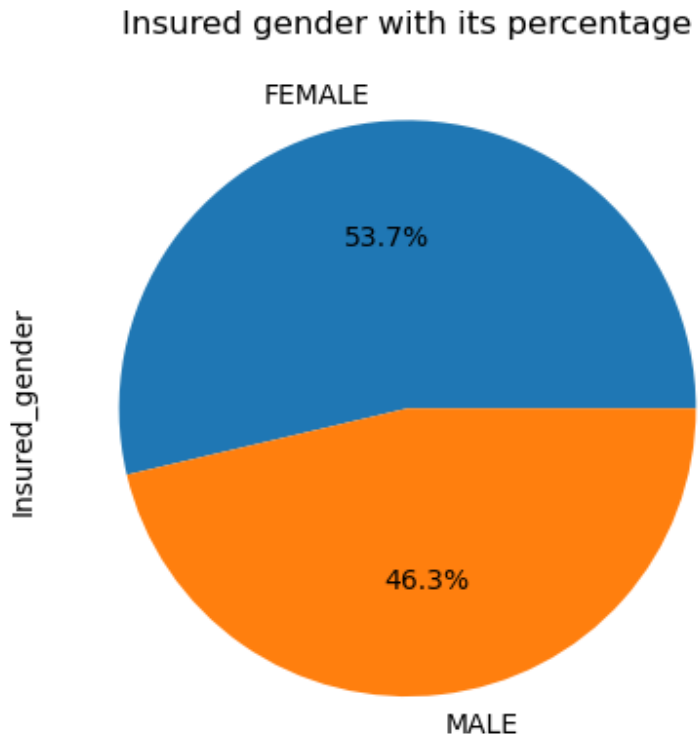
the male is 463

In [73]:
```
Insured_gender= df.groupby("Insured_gender")

Insured_gender.size()
```

Out[73]:
```
Insured_gender
FEMALE    537
MALE      463
dtype: int64
```

In [74]:
```
df['Insured_gender'].value_counts().plot(kind='pie',autopct='%1.1f%%')
plt.title('Insured gender with its percentage')
plt.show()
```



in this, the groupby is used for type of Collision

the ? is 178

the Front Collision is 254

the Rear Collision is 292

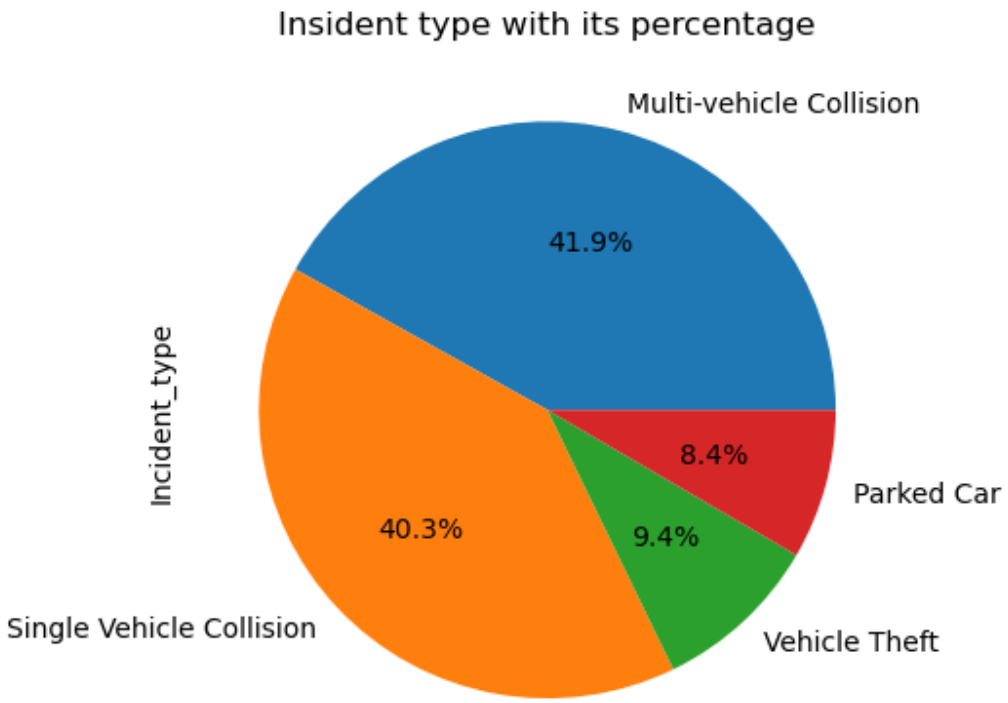
the Side Collision is 276

In [75]:
```
Collision_type= df.groupby("Collision_type")

Collision_type.size()
```

Out[75]:
```
Collision_type
?                 178
Front Collision   254
Rear Collision    292
Side Collision    276
dtype: int64
```

In [76]:
```
df['Incident_type'].value_counts().plot(kind='pie',autopct='%1.1f%%')
plt.title('Insident type with its percentage')
plt.show()
```



in this, the groupby is used for type of incident

the Multi-vehicle Collision is 419

the Parked Cars are 84

the Single Vehicle Collision is 403

the Vehicle Theft is 94

In [77]:
```
Incident_type= df.groupby("Incident_type")

Incident_type.size()
```

Out[77]:
```
Incident_type
Multi-vehicle Collision     419
Parked Car                   84
Single Vehicle Collision    403
Vehicle Theft                94
dtype: int64
```

In [78]:
```
Incident_city= df.groupby("Incident_city")

Incident_city.size()
```

Out[78]:
```
Incident_city
Arlington      152
Columbus       149
Hillsdale      141
Northbend      145
Northbrook     122
Riverwood      134
Springfield    157
dtype: int64
```

Conclusion:

In this project, I analyzed the all data like the highest Incidents occurs in Springfield city and it's count is 157 and also the lowest Incident occures in Northbrook city and it's count is 122.

The Policy_annual_premium is 1230.69 and 1223.39 as compair to others

The highest Incident state New York is 262 and The lowest Incident state West Verginia is 217

The Insured genders are: Female is 537 and Male is 463

In this, type of incident The Multi-vehicle Collision is 419 , the Parked Cars are 84,the Single Vehicle Collision is 403,the Vehicle Theft is 94

In this, the groupby is used for type of Collision ,.the ? is 178, the Front Collision is 254, the Rear Collision is 292, the Side Collision is 276