#Importing the essential libraries

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

df=pd.read_csv('Insurance.csv')

df.columns
```

```
Index(['INSR_BEGIN', 'INSR_END', 'EFFECTIVE_YR', 'INSR_TYPE',
'INSURED_VALUE',
       'PREMIUM', 'OBJECT_ID', 'PROD_YEAR', 'SEATS_NUM',
'CARRYING_CAPACITY',
       'TYPE_VEHICLE', 'CCM_TON', 'MAKE', 'USAGE', 'CLAIM_PAID'],
      dtype='object')
```

#Transform Process

```
df.columns=df.columns.str.lower()

df.head(10)
```

```
  insr_begin   insr_end effective_yr  insr_type  insured_value
premium  \
0  08-Aug-17  07-Aug-18            8       1202      519755.22
5097.83
1  08-Aug-16  07-Aug-17            8       1202      519755.22
6556.52
2  08-Aug-15  07-Aug-16            8       1202      519755.22
6556.52
3  08-Aug-14  07-Aug-15            8       1202      519755.22
5102.83
4  08-Aug-17  07-Aug-18            8       1202     1400000.00
13304.87
5  08-Aug-16  07-Aug-17            8       1202     1400000.00
16438.15
6  08-Aug-15  07-Aug-16            8       1202     1400000.00
16438.15
7  08-Aug-14  07-Aug-15            8       1202      285451.24
3931.23
8  24-Nov-17  23-Nov-18           12       1202     3400000.00
26804.72
9  24-Nov-16  23-Nov-17           12       1202     3400000.00
26804.72

    object_id  prod_year  seats_num  carrying_capacity type_vehicle
ccm_ton  \
0  5000029885     2007.0        4.0                6.0      Pick-up
3153.0
```

```
1  5000029885     2007.0        4.0                6.0      Pick-up
3153.0
2  5000029885     2007.0        4.0                6.0      Pick-up
3153.0
3  5000029885     2007.0        4.0                6.0      Pick-up
3153.0
4  5000029901     2010.0        4.0                7.0      Pick-up
2494.0
5  5000029901     2010.0        4.0                7.0      Pick-up
2494.0
6  5000029901     2010.0        4.0                7.0      Pick-up
2494.0
7  5000029901     2010.0        4.0                7.0      Pick-up
2494.0
8  5000030358     2012.0        0.0              220.0        Truck
12880.0
9  5000030358     2012.0        0.0              220.0        Truck
12880.0
```

```
     make             usage  claim_paid
0  NISSAN         Own Goods         NaN
1  NISSAN         Own Goods         NaN
2  NISSAN         Own Goods         NaN
3  NISSAN         Own Goods         NaN
4  TOYOTA         Own Goods         NaN
5  TOYOTA         Own Goods         NaN
6  TOYOTA         Own Goods   365250.00
7  TOYOTA         Own Goods    12152.73
8   IVECO  General Cartage         NaN
9   IVECO  General Cartage         NaN
```

```
df.tail(10)
```

```
        insr_begin   insr_end effective_yr  insr_type
insured_value  \
1389100  08-Dec-12  07-Dec-13           11       1201            0.0

1389101  08-Jul-13  07-Jul-14           11       1202       100000.0

1389102  08-Jul-12  07-Jul-13           11       1202       100000.0

1389103  08-Jul-11  07-Jul-12           11       1202       100000.0

1389104  11-Aug-13  10-Aug-14           88       1201            0.0

1389105  11-Aug-12  10-Aug-13           88       1201            0.0

1389106  11-Aug-11  10-Aug-12           88       1201            0.0

1389107  01-Jun-13  31-May-14           13       1201       250000.0
```

```
1389108  08-Jul-12  07-Jul-13           85       1202             0.0

1389109  08-Jul-11  07-Jul-12           85       1202             0.0


          premium   object_id  prod_year  seats_num  carrying_capacity
\
1389100   541.949  5000889594     2012.0        4.0                NaN

1389101  3137.800  5000051125     2000.0        6.0                5.0

1389102  3298.870  5000051125     2000.0        6.0                5.0

1389103  3673.697  5000051125     2000.0        6.0                5.0

1389104   541.950  5000047311     1985.0        4.0                NaN

1389105   577.700  5000047311     1985.0        4.0                NaN

1389106   695.318  5000047311     1985.0        4.0                NaN

1389107  4778.450  5000949222     1993.0        4.0                NaN

1389108  1057.427  5000049231     2017.0        2.0                0.0

1389109  1001.537  5000049231     2017.0        2.0                0.0


            type_vehicle  ccm_ton     make            usage
claim_paid
1389100       Automobile   1200.0  HYUNDAI          Private
NaN
1389101  Station Wagones   4164.0   TOYOTA          Private
NaN
1389102  Station Wagones   4164.0   TOYOTA          Private
NaN
1389103  Station Wagones   4164.0   TOYOTA          Private
NaN
1389104       Automobile   1295.0   TOYOTA          Private
NaN
1389105       Automobile   1295.0   TOYOTA          Private
NaN
1389106       Automobile   1295.0   TOYOTA          Private
NaN
1389107       Automobile   1295.0   TOYOTA          Private
NaN
1389108            Truck   4570.0    ISUZU  General Cartage
NaN
1389109            Truck   4570.0    ISUZU  General Cartage
NaN
```

```
df.shape
```

```
(1389110, 15)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1389110 entries, 0 to 1389109
Data columns (total 15 columns):
 #   Column             Non-Null Count    Dtype  
---  ------             --------------    -----  
 0   insr_begin         1389110 non-null  object 
 1   insr_end           1389110 non-null  object 
 2   effective_yr       1389104 non-null  object 
 3   insr_type          1389110 non-null  int64  
 4   insured_value      1389110 non-null  float64
 5   premium            1389073 non-null  float64
 6   object_id          1389110 non-null  int64  
 7   prod_year          1388729 non-null  float64
 8   seats_num          1388595 non-null  float64
 9   carrying_capacity  1028181 non-null  float64
 10  type_vehicle       1389110 non-null  object 
 11  ccm_ton            1389098 non-null  float64
 12  make               1389105 non-null  object 
 13  usage              1389110 non-null  object 
 14  claim_paid         104891 non-null   float64
dtypes: float64(7), int64(2), object(6)
memory usage: 159.0+ MB
```

```
df.duplicated().sum()
```

```
587279
```

```
df.drop_duplicates(inplace= True)
```

```
df.shape
```

```
(801831, 15)
```

```
df.isnull().sum()
```

```
insr_begin               0
insr_end                 0
effective_yr             4
insr_type                0
insured_value            0
premium                 21
object_id                0
prod_year              169
seats_num              235
carrying_capacity    198162
```

```
type_vehicle              0
ccm_ton                   8
make                      5
usage                     0
claim_paid           741686
dtype: int64
```

```
df['type_vehicle'].value_counts()
```

```
type_vehicle
Truck                        151055
Pick-up                      144859
Motor-cycle                  143136
Automobile                   125960
Bus                          105962
Station Wagones               60782
Trailers and semitrailers     35939
Special construction          12077
Tractor                       11411
Tanker                        10632
Trade plates                     18
Name: count, dtype: int64
```

```
df['type_vehicle'].value_counts().sum()
```

```
801831
```

```
seat_counts = df['seats_num'].value_counts().sort_index().to_dict()
print(seat_counts)
```

```
{0.0: 59805, 1.0: 152046, 2.0: 99028, 3.0: 82646, 4.0: 233340, 5.0:
14112, 6.0: 10500, 7.0: 7620, 8.0: 29476, 9.0: 5924, 10.0: 2483, 11.0:
41415, 12.0: 5331, 13.0: 1316, 14.0: 5627, 15.0: 4744, 16.0: 865,
17.0: 100, 18.0: 194, 19.0: 273, 20.0: 525, 21.0: 308, 22.0: 1294,
23.0: 277, 24.0: 11932, 25.0: 1374, 26.0: 482, 27.0: 1475, 28.0: 2318,
29.0: 2200, 30.0: 792, 31.0: 119, 32.0: 669, 33.0: 158, 34.0: 85,
35.0: 78, 36.0: 101, 37.0: 184, 38.0: 49, 39.0: 54, 40.0: 158, 41.0:
11, 42.0: 18, 43.0: 231, 44.0: 3441, 45.0: 1398, 46.0: 394, 47.0: 117,
48.0: 72, 49.0: 34, 50.0: 860, 51.0: 115, 52.0: 15, 53.0: 42, 54.0:
88, 55.0: 207, 56.0: 22, 57.0: 27, 58.0: 37, 59.0: 176, 60.0: 2345,
61.0: 762, 62.0: 654, 63.0: 532, 64.0: 41, 65.0: 154, 66.0: 39, 69.0:
5, 70.0: 20, 71.0: 1, 72.0: 1, 80.0: 11, 85.0: 624, 90.0: 14, 93.0: 9,
94.0: 4, 95.0: 29, 99.0: 4643, 100.0: 159, 102.0: 502, 105.0: 12,
110.0: 34, 112.0: 1, 120.0: 176, 125.0: 1, 132.0: 2, 139.0: 846,
150.0: 40, 158.0: 8, 160.0: 1113, 161.0: 4, 170.0: 2, 175.0: 13,
178.0: 1, 198.0: 11, 199.0: 1}
```

```
df['seats_num'].mode()
```

```
0    4.0
Name: seats_num, dtype: float64
```

```
filtered_seat = df[df['seats_num'] == 4.0]
grouped =
filtered_seat.groupby('type_vehicle').size().reset_index(name='count')
print(grouped)
```

```
                type_vehicle   count
0                 Automobile  110973
1                        Bus     879
2                 Motor-cycle    8568
3                     Pick-up  102452
4       Special construction     341
5            Station Wagones    9822
6                     Tanker      24
7                    Tractor       4
8  Trailers and semitrailers      16
9                      Truck     261
```

```
filtered_seat = df[pd.isna(df['seats_num'])]
grouped =
filtered_seat.groupby('type_vehicle').size().reset_index(name='count')
print(grouped)
```

```
           type_vehicle  count
0            Automobile    169
1                   Bus     13
2  Special construction      1
3               Tractor      8
4          Trade plates      8
5                 Truck     36
```

```
def fill_seats(df):
    # List of vehicle types for which SEATS_NUM should be set to 4.0
    vehicle_types = ['Automobile', 'Bus', 'Special construction',
'Tractor', 'Trade plates', 'Truck']

    # Fill SEATS_NUM with 4.0 for the specific vehicle types
    df.loc[df['type_vehicle'].isin(vehicle_types), 'seats_num'] = 4.0

    return df
df = fill_seats(df)
print(df)
```

```
       insr_begin   insr_end effective_yr  insr_type  insured_value  \
0       08-Aug-17  07-Aug-18            8       1202      519755.22
1       08-Aug-16  07-Aug-17            8       1202      519755.22
2       08-Aug-15  07-Aug-16            8       1202      519755.22
3       08-Aug-14  07-Aug-15            8       1202      519755.22
4       08-Aug-17  07-Aug-18            8       1202     1400000.00
...           ...        ...          ...        ...            ...
802031  11-Aug-12  10-Aug-13           88       1201           0.00
```

```
802032  11-Aug-11  10-Aug-12           88       1201           0.00
802033  01-Jun-13  31-May-14           13       1201      250000.00
802034  08-Jul-12  07-Jul-13           85       1202           0.00
802035  08-Jul-11  07-Jul-12           85       1202           0.00
```

```
           premium   object_id  prod_year  seats_num  carrying_capacity
\
0         5097.830  5000029885     2007.0        4.0                6.0

1         6556.520  5000029885     2007.0        4.0                6.0

2         6556.520  5000029885     2007.0        4.0                6.0

3         5102.830  5000029885     2007.0        4.0                6.0

4        13304.870  5000029901     2010.0        4.0                7.0

...            ...         ...        ...        ...                ...

802031     577.700  5000047311     1985.0        4.0                NaN

802032     695.318  5000047311     1985.0        4.0                NaN

802033    4778.450  5000949222     1993.0        4.0                NaN

802034    1057.427  5000049231     2017.0        4.0                0.0

802035    1001.537  5000049231     2017.0        4.0                0.0


       type_vehicle  ccm_ton    make            usage  claim_paid
0           Pick-up   3153.0  NISSAN        Own Goods         NaN
1           Pick-up   3153.0  NISSAN        Own Goods         NaN
2           Pick-up   3153.0  NISSAN        Own Goods         NaN
3           Pick-up   3153.0  NISSAN        Own Goods         NaN
4           Pick-up   2494.0  TOYOTA        Own Goods         NaN
...             ...      ...     ...              ...         ...
802031   Automobile   1295.0  TOYOTA          Private         NaN
802032   Automobile   1295.0  TOYOTA          Private         NaN
802033   Automobile   1295.0  TOYOTA          Private         NaN
802034        Truck   4570.0   ISUZU  General Cartage         NaN
802035        Truck   4570.0   ISUZU  General Cartage         NaN

[801831 rows x 15 columns]
```

```
df.isnull().sum()
```

```
insr_begin                0
insr_end                  0
effective_yr              4
insr_type                 0
```

```
insured_value              0
premium                   21
object_id                  0
prod_year                169
seats_num                  0
carrying_capacity    198162
type_vehicle               0
ccm_ton                    8
make                       5
usage                      0
claim_paid            741686
dtype: int64
```

```
df["carrying_capacity"].median()
```

```
6.0
```

```
filtered_carry = df[df['carrying_capacity'] == 6.0]
grouped =
filtered_carry.groupby('type_vehicle').size().reset_index(name='count'
)
print(grouped)
```

```
                type_vehicle  count
0                 Automobile    149
1                        Bus    565
2                Motor-cycle     38
3                    Pick-up  22383
4       Special construction     89
5            Station Wagones    382
6                    Tractor      1
7  Trailers and semitrailers      7
8                      Truck     11
```

```
filtered_carry = df[pd.isna(df['carrying_capacity'])]
grouped =
filtered_carry.groupby('type_vehicle').size().reset_index(name='count'
)
print(grouped)
```

```
                 type_vehicle   count
0                  Automobile  111583
1                         Bus     460
2                 Motor-cycle   31775
3                     Pick-up     674
4        Special construction      76
5             Station Wagones   52542
6                      Tanker       4
7                     Tractor     174
8                Trade plates      17
```

```
9   Trailers and semitrailers      83
10                      Truck     774
```

```
def carry_capacity(df):
    # List of vehicle types for which SEATS_NUM should be set to 4.0
    vehicle_types = ['Automobile', 'Bus', 'Motor-cycle', 'Pick-up',
'Special construction', 'Station Wagones','Tanker',  'Tractor', 'Trade
plates','Trailers and semitrailers', 'Truck']

    # Fill SEATS_NUM with 4.0 for the specific vehicle types
    df.loc[df['type_vehicle'].isin(vehicle_types),
'carrying_capacity'] = 6.0

    return df
df = carry_capacity(df)
print(df)
```

```
       insr_begin   insr_end effective_yr  insr_type  insured_value  \
0       08-Aug-17  07-Aug-18            8       1202      519755.22
1       08-Aug-16  07-Aug-17            8       1202      519755.22
2       08-Aug-15  07-Aug-16            8       1202      519755.22
3       08-Aug-14  07-Aug-15            8       1202      519755.22
4       08-Aug-17  07-Aug-18            8       1202     1400000.00
...           ...        ...          ...        ...            ...
802031  11-Aug-12  10-Aug-13           88       1201           0.00
802032  11-Aug-11  10-Aug-12           88       1201           0.00
802033  01-Jun-13  31-May-14           13       1201      250000.00
802034  08-Jul-12  07-Jul-13           85       1202           0.00
802035  08-Jul-11  07-Jul-12           85       1202           0.00

          premium   object_id  prod_year  seats_num  carrying_capacity
\
0        5097.830  5000029885     2007.0        4.0                6.0

1        6556.520  5000029885     2007.0        4.0                6.0

2        6556.520  5000029885     2007.0        4.0                6.0

3        5102.830  5000029885     2007.0        4.0                6.0

4       13304.870  5000029901     2010.0        4.0                6.0

...           ...         ...        ...        ...                ...

802031    577.700  5000047311     1985.0        4.0                6.0

802032    695.318  5000047311     1985.0        4.0                6.0

802033   4778.450  5000949222     1993.0        4.0                6.0

802034   1057.427  5000049231     2017.0        4.0                6.0
```

```
802035   1001.537  5000049231     2017.0        4.0                 6.0


       type_vehicle  ccm_ton    make            usage  claim_paid
0           Pick-up   3153.0  NISSAN        Own Goods         NaN
1           Pick-up   3153.0  NISSAN        Own Goods         NaN
2           Pick-up   3153.0  NISSAN        Own Goods         NaN
3           Pick-up   3153.0  NISSAN        Own Goods         NaN
4           Pick-up   2494.0  TOYOTA        Own Goods         NaN
...             ...      ...     ...              ...         ...
802031   Automobile   1295.0  TOYOTA          Private         NaN
802032   Automobile   1295.0  TOYOTA          Private         NaN
802033   Automobile   1295.0  TOYOTA          Private         NaN
802034        Truck   4570.0   ISUZU  General Cartage         NaN
802035        Truck   4570.0   ISUZU  General Cartage         NaN

[801831 rows x 15 columns]

df.isnull().sum()

insr_begin                0
insr_end                  0
effective_yr              4
insr_type                 0
insured_value             0
premium                  21
object_id                 0
prod_year               169
seats_num                 0
carrying_capacity         0
type_vehicle              0
ccm_ton                   8
make                      5
usage                     0
claim_paid           741686
dtype: int64

df['claim_paid'] = pd.to_numeric(df['claim_paid'],
errors='coerce').fillna(0)

make_value = df['make'].mode()[0]
print(make_value)
df['make'] = df['make'].fillna(make_value)

TOYOTA

df.head(3)

  insr_begin   insr_end effective_yr  insr_type  insured_value
premium  \
```

```
0  08-Aug-17  07-Aug-18            8       1202      519755.22
5097.83
1  08-Aug-16  07-Aug-17            8       1202      519755.22
6556.52
2  08-Aug-15  07-Aug-16            8       1202      519755.22
6556.52

    object_id  prod_year  seats_num  carrying_capacity type_vehicle
ccm_ton  \
0  5000029885     2007.0        4.0                6.0      Pick-up
3153.0
1  5000029885     2007.0        4.0                6.0      Pick-up
3153.0
2  5000029885     2007.0        4.0                6.0      Pick-up
3153.0

     make      usage  claim_paid
0  NISSAN  Own Goods         0.0
1  NISSAN  Own Goods         0.0
2  NISSAN  Own Goods         0.0
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 801831 entries, 0 to 802035
Data columns (total 15 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   insr_begin         801831 non-null  object
 1   insr_end           801831 non-null  object
 2   effective_yr       801827 non-null  object
 3   insr_type          801831 non-null  int64
 4   insured_value      801831 non-null  float64
 5   premium            801810 non-null  float64
 6   object_id          801831 non-null  int64
 7   prod_year          801662 non-null  float64
 8   seats_num          801831 non-null  float64
 9   carrying_capacity  801831 non-null  float64
 10  type_vehicle       801831 non-null  object
 11  ccm_ton            801823 non-null  float64
 12  make               801831 non-null  object
 13  usage              801831 non-null  object
 14  claim_paid         801831 non-null  float64
dtypes: float64(7), int64(2), object(6)
memory usage: 97.9+ MB
```

```
df.fillna({
    'premium': df['premium'].mean(),
    'prod_year': df['prod_year'].mode()[0],
```

```
    'ccm_ton': df['ccm_ton'].median()
}, inplace=True)

df.isnull().sum()
```

```
insr_begin            0
insr_end              0
effective_yr          4
insr_type             0
insured_value         0
premium               0
object_id             0
prod_year             0
seats_num             0
carrying_capacity     0
type_vehicle          0
ccm_ton               0
make                  0
usage                 0
claim_paid            0
dtype: int64
```

```
df['insr_begin'] = pd.to_datetime(df['insr_begin'], format='mixed',
dayfirst=True)
df['insr_end'] = pd.to_datetime(df['insr_end'], format='mixed',
dayfirst=True)
df['prod_year'] = pd.to_datetime(df['prod_year'], format='%Y',
errors='coerce')

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 801831 entries, 0 to 802035
Data columns (total 15 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   insr_begin         801831 non-null  datetime64[ns]
 1   insr_end           801831 non-null  datetime64[ns]
 2   effective_yr       801827 non-null  object
 3   insr_type          801831 non-null  int64
 4   insured_value      801831 non-null  float64
 5   premium            801831 non-null  float64
 6   object_id          801831 non-null  int64
 7   prod_year          801831 non-null  datetime64[ns]
 8   seats_num          801831 non-null  float64
 9   carrying_capacity  801831 non-null  float64
 10  type_vehicle       801831 non-null  object
 11  ccm_ton            801831 non-null  float64
 12  make               801831 non-null  object
 13  usage              801831 non-null  object
```

```
 14  claim_paid         801831 non-null  float64
dtypes: datetime64[ns](3), float64(6), int64(2), object(4)
memory usage: 97.9+ MB
```

```
df['effective_yr'].mode()[0]
```

```
'11'
```

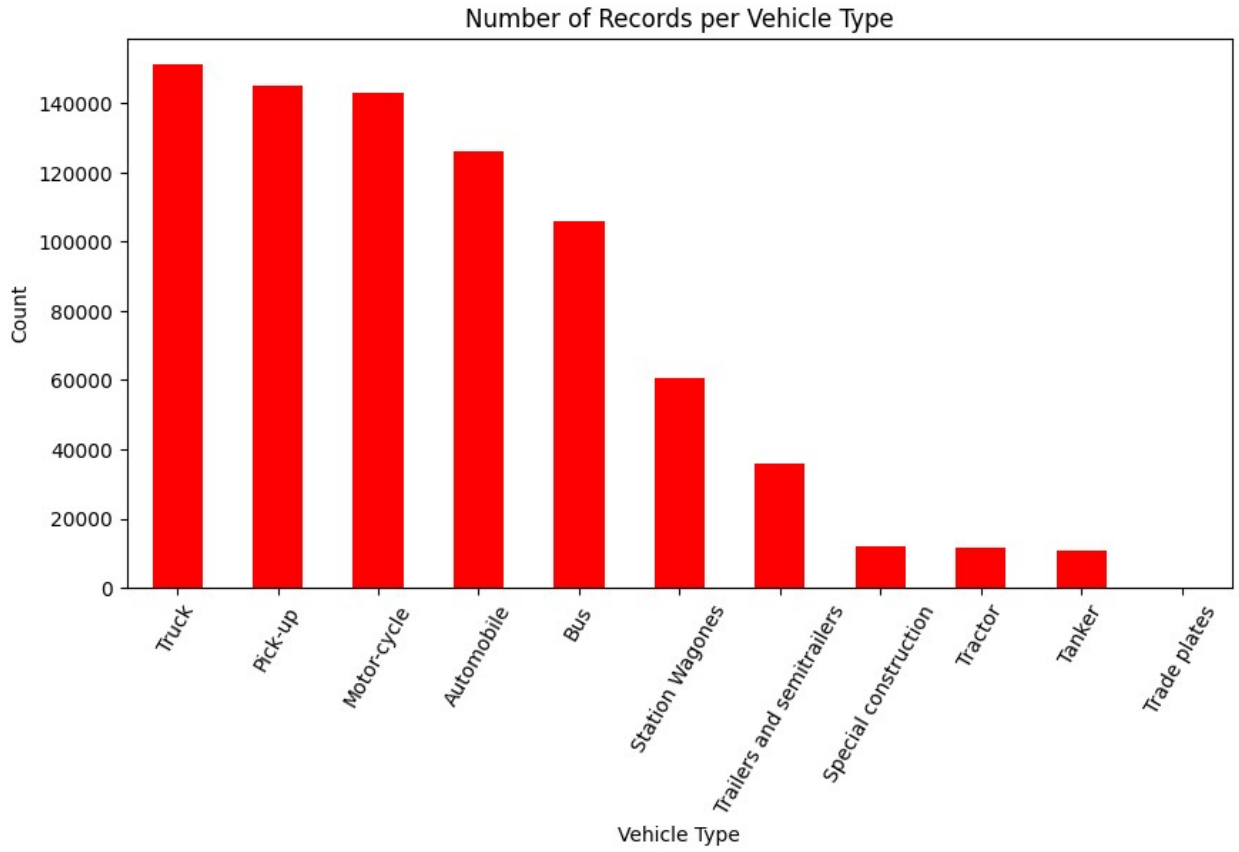
```
df.fillna({'effective_yr': df['effective_yr'].mode()[0]}, inplace=
True)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 801831 entries, 0 to 802035
Data columns (total 15 columns):
 #   Column             Non-Null Count   Dtype
---  ------             --------------   -----
 0   insr_begin         801831 non-null  datetime64[ns]
 1   insr_end           801831 non-null  datetime64[ns]
 2   effective_yr       801831 non-null  object
 3   insr_type          801831 non-null  int64
 4   insured_value      801831 non-null  float64
 5   premium            801831 non-null  float64
 6   object_id          801831 non-null  int64
 7   prod_year          801831 non-null  datetime64[ns]
 8   seats_num          801831 non-null  float64
 9   carrying_capacity  801831 non-null  float64
 10  type_vehicle       801831 non-null  object
 11  ccm_ton            801831 non-null  float64
 12  make               801831 non-null  object
 13  usage              801831 non-null  object
 14  claim_paid         801831 non-null  float64
dtypes: datetime64[ns](3), float64(6), int64(2), object(4)
memory usage: 97.9+ MB
```

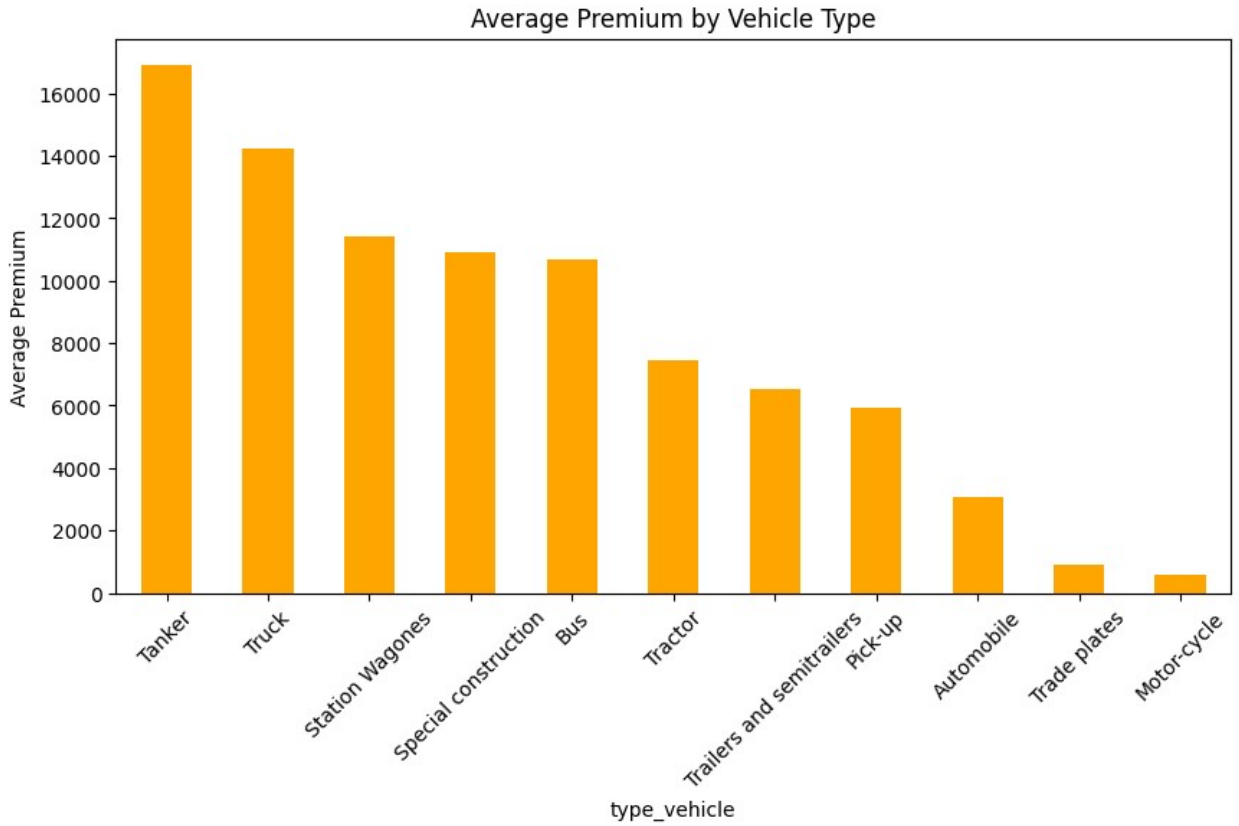
Top Vehicle Types by Count

```
df['type_vehicle'].value_counts().plot(kind='bar', color='red',
figsize=(10, 5))
plt.title('Number of Records per Vehicle Type')
plt.xlabel('Vehicle Type')
plt.ylabel('Count')
plt.xticks(rotation=60)
plt.show()
```



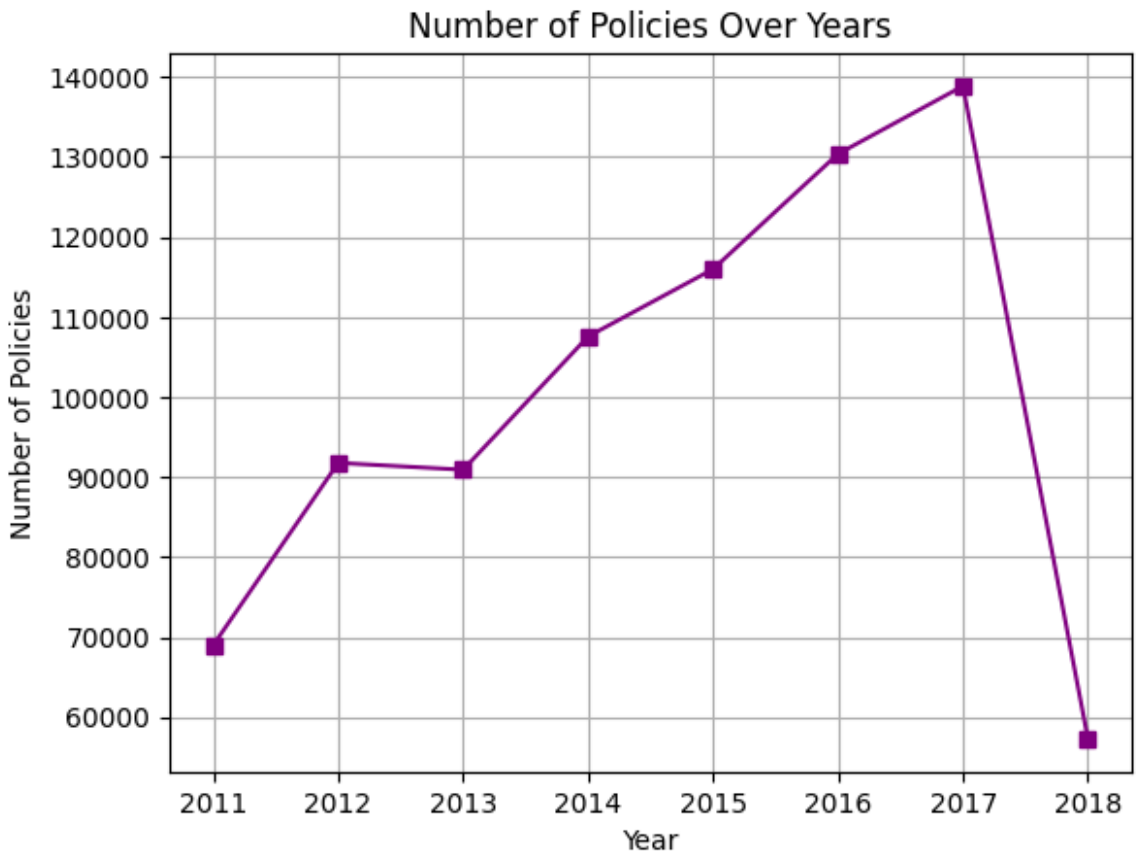
Average Premium by Vehicle Type

```
avg_premium = df.groupby('type_vehicle')
['premium'].mean().sort_values(ascending=False)
avg_premium.plot(kind='bar', color='orange', figsize=(10, 5))
plt.title('Average Premium by Vehicle Type')
plt.ylabel('Average Premium')
plt.xticks(rotation=45)
plt.show()
```



Insurance Policies Over Time

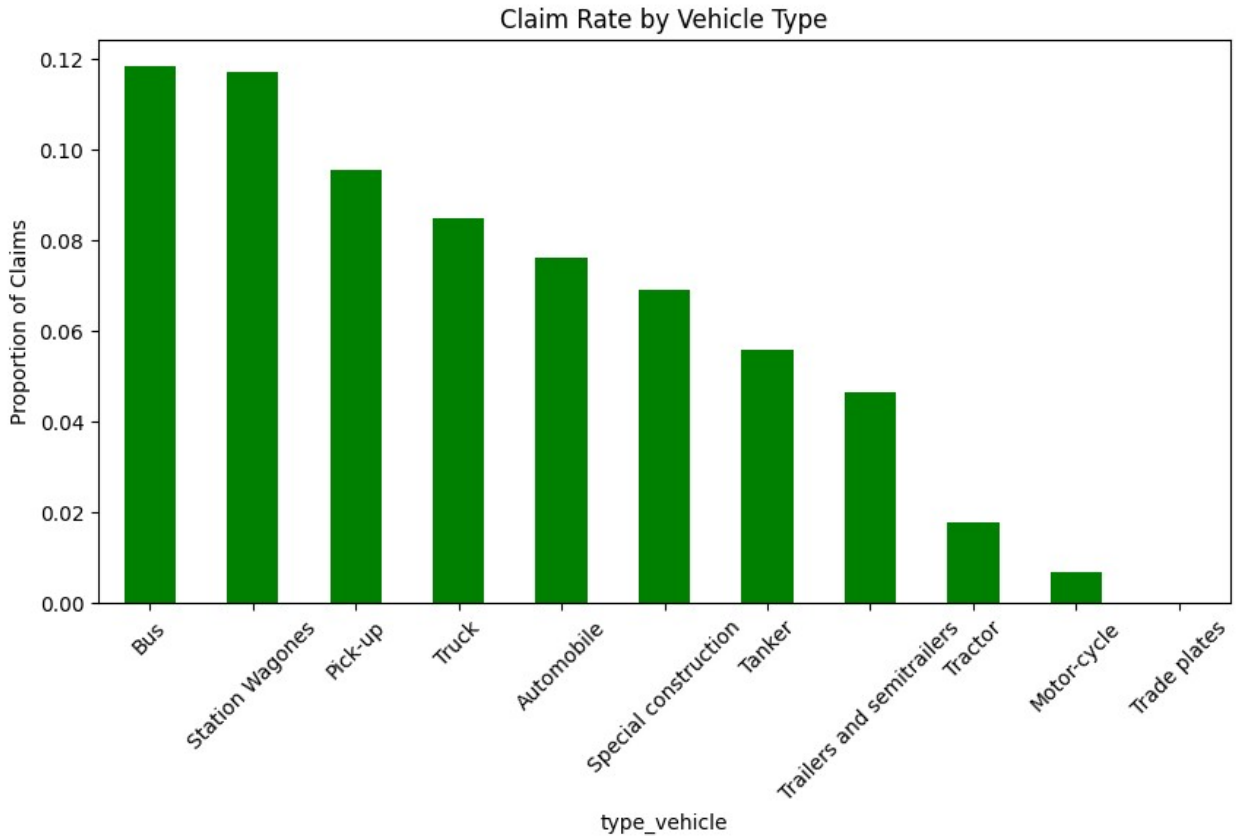
```
df['year'] = df['insr_begin'].dt.year
policy_counts = df['year'].value_counts().sort_index()

policy_counts.plot(kind='line', marker='s', color='purple')
plt.title('Number of Policies Over Years')
plt.xlabel('Year')
plt.ylabel('Number of Policies')
plt.grid(True)
plt.show()
```



Claim Paid Analysis

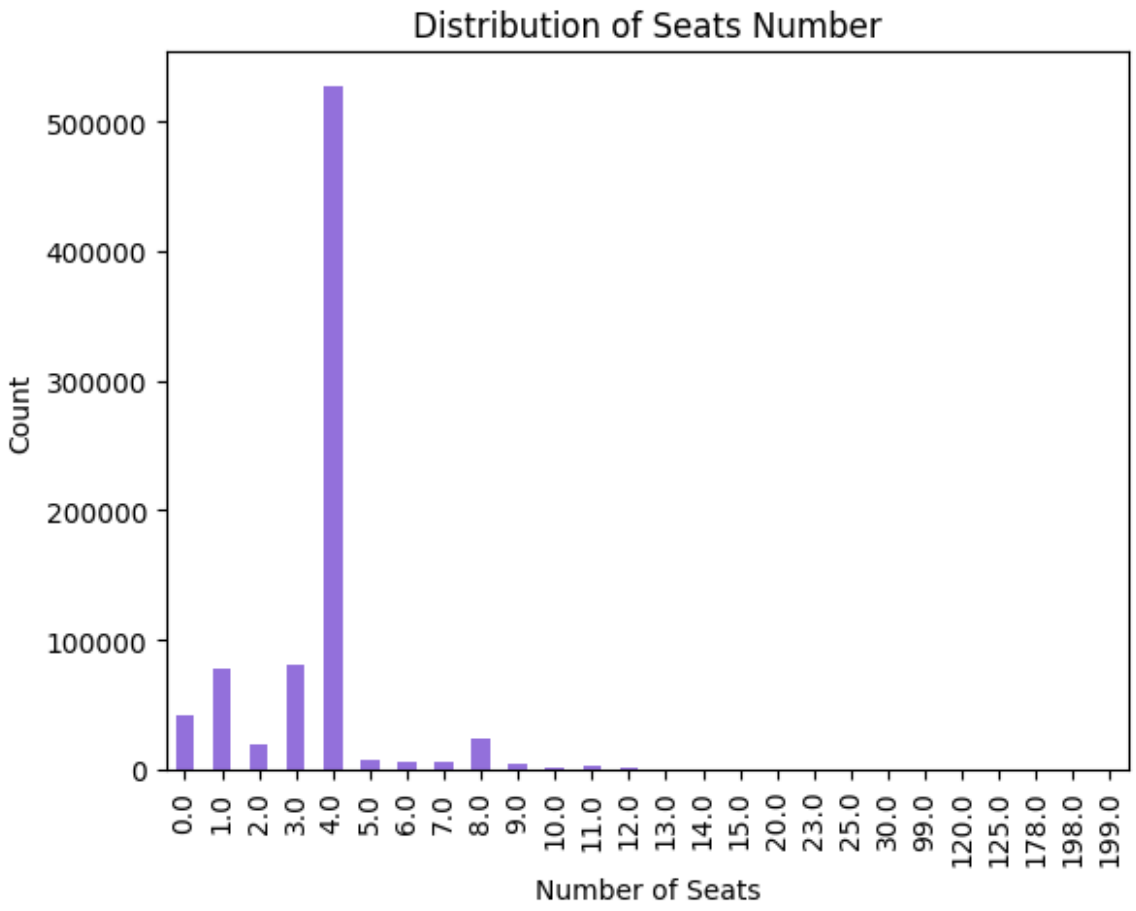
```
df['claim_paid_binary'] = np.where(df['claim_paid'] > 0, 1, 0)
claim_rate = df['claim_paid_binary'].mean()
print(f"Claim Rate: {claim_rate:.2%}")

claim_by_vehicle = df.groupby('type_vehicle')
['claim_paid_binary'].mean().sort_values(ascending=False)
claim_by_vehicle.plot(kind='bar', color='green', figsize=(10, 5))
plt.title('Claim Rate by Vehicle Type')
plt.ylabel('Proportion of Claims')
plt.xticks(rotation=45)
plt.show()
```

```
Claim Rate: 7.50%
```



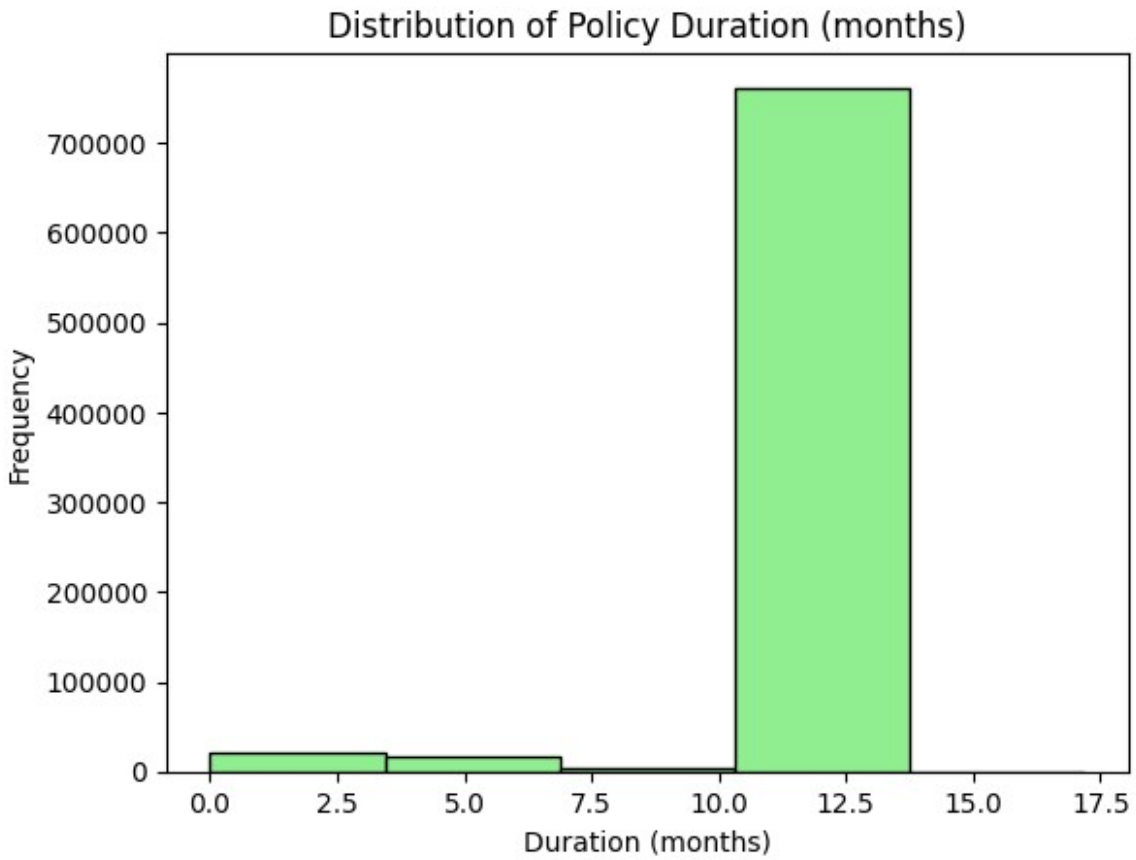
Distribution of Seats Number

```
df['seats_num'].value_counts().sort_index().plot(kind='bar',
color='mediumpurple')
plt.title('Distribution of Seats Number')
plt.xlabel('Number of Seats')
plt.ylabel('Count')
plt.show()
```



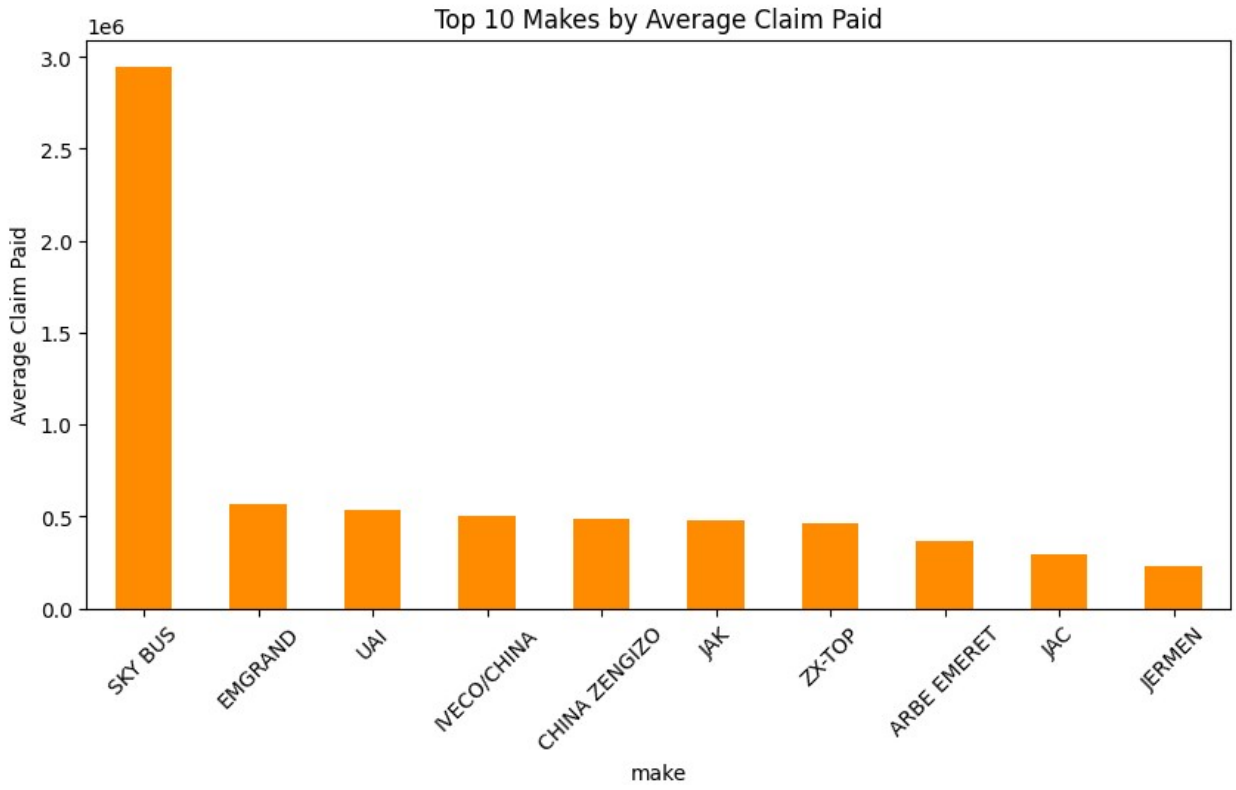
Policy Duration

```
df['policy_duration_months'] = (df['insr_end'] -
df['insr_begin']).dt.days/30.42

plt.hist(df['policy_duration_months'], bins=5, color='lightgreen',
edgecolor='black')
plt.title('Distribution of Policy Duration (months)')
plt.xlabel('Duration (months)')
plt.ylabel('Frequency')
plt.show()
```



Average Claim Paid by Make

```
avg_claim_by_make = df.groupby('make')
['claim_paid'].mean().sort_values(ascending=False).head(10)
avg_claim_by_make.plot(kind='bar', color='darkorange', figsize=(10,5))
plt.title('Top 10 Makes by Average Claim Paid')
plt.ylabel('Average Claim Paid')
plt.xticks(rotation=45)
plt.show()
```


Top 10 Makes by Average Claim Paid
1e6
3.0
2.5
2.0
1.5
1.0
0.5
0.0
Average Claim Paid
SKY BUS
EMGRAND
UAI
IVECO/CHINA
CHINA ZENGIZO
JAK
ZX-TOP
ARBE EMERET
JAC
JERMEN
make