----------------------INSURANCE DATASET----------------------

IMPORTING THE LIBRARIES

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import scipy as sp
import warnings
import os
warnings.filterwarnings("ignore")
```

Loading the dataset

```
data=pd.read_csv(r"C:\Users\Admin\Downloads\insurance_claims.csv")
data.head()
```

```
   months_as_customer  age  policy_number policy_bind_date
policy_state  \
0                 328   48         521585       17-10-2014
OH
1                 228   42         342868       27-06-2006
IN
2                 134   29         687698       06-09-2000
OH
3                 256   41         227811       25-05-1990
IL
4                 228   44         367455       06-06-2014
IL

  policy_csl  policy_deductable  policy_annual_premium  umbrella_limit
\
0    250/500               1000                1406.91               0

1    250/500               2000                1197.22         5000000

2    100/300               2000                1413.14         5000000

3    250/500               2000                1415.74         6000000

4   500/1000               1000                1583.91         6000000
```

```
   insured_zip  ... police_report_available total_claim_amount
injury_claim  \
0       466132  ...                     YES              71610
6510
1       468176  ...                       ?               5070
780
2       430632  ...                      NO              34650
7700
3       608117  ...                      NO              63400
6340
4       610706  ...                      NO               6500
1300

  property_claim vehicle_claim  auto_make  auto_model auto_year  \
0          13020         52080       Saab         92x      2004
1            780          3510   Mercedes        E400      2007
2           3850         23100      Dodge         RAM      2007
3           6340         50720  Chevrolet       Tahoe      2014
4            650          4550     Accura         RSX      2009

  fraud_reported _c39
0              Y  NaN
1              Y  NaN
2              N  NaN
3              Y  NaN
4              N  NaN

[5 rows x 40 columns]

data.shape

(1000, 40)

data.replace('?',' ',inplace=True)
data
data["fraud_reported"] = data["fraud_reported"].apply(lambda x:
x.replace("Y", "Yes"))
data
data["fraud_reported"] = data["fraud_reported"].apply(lambda x:
x.replace("N", "No"))
data

     months_as_customer  age  policy_number policy_bind_date
policy_state  \
0                   328   48         521585       17-10-2014
OH
1                   228   42         342868       27-06-2006
IN
2                   134   29         687698       06-09-2000
```

```
OH
3                    256   41         227811       25-05-1990
IL
4                    228   44         367455       06-06-2014
IL
..                   ...  ...            ...              ...
...
995                    3   38         941851       16-07-1991
OH
996                  285   41         186934       05-01-2014
IL
997                  130   34         918516       17-02-2003
OH
998                  458   62         533940       18-11-2011
IL
999                  456   60         556080       11-11-1996
OH

    policy_csl  policy_deductable  policy_annual_premium
umbrella_limit  \
0      250/500               1000                1406.91
0
1      250/500               2000                1197.22
5000000
2      100/300               2000                1413.14
5000000
3      250/500               2000                1415.74
6000000
4     500/1000               1000                1583.91
6000000
..         ...                ...                    ...             .
..
995   500/1000               1000                1310.80
0
996    100/300               1000                1436.79
0
997    250/500                500                1383.49
3000000
998   500/1000               2000                1356.92
5000000
999    250/500               1000                 766.19
0

     insured_zip  ... police_report_available total_claim_amount
injury_claim  \
0         466132  ...                     YES              71610
6510
1         468176  ...                                       5070
780
```

```
2       430632  ...                       NO          34650
7700
3       608117  ...                       NO          63400
6340
4       610706  ...                       NO           6500
1300
..         ...  ...                      ...            ...
...
995     431289  ...                                   87200
17440
996     608177  ...                                  108480
18080
997     442797  ...                      YES          67500
7500
998     441714  ...                      YES          46980
5220
999     612260  ...                                    5060
460

    property_claim vehicle_claim   auto_make  auto_model auto_year  \
0            13020         52080        Saab         92x      2004
1              780          3510    Mercedes        E400      2007
2             3850         23100       Dodge         RAM      2007
3             6340         50720   Chevrolet       Tahoe      2014
4              650          4550      Accura         RSX      2009
..             ...           ...         ...         ...       ...
995           8720         61040       Honda      Accord      2006
996          18080         72320  Volkswagen      Passat      2015
997           7500         52500      Suburu     Impreza      1996
998           5220         36540        Audi          A5      1998
999            920          3680    Mercedes        E400      2007

    fraud_reported _c39
0              Yes  NaN
1              Yes  NaN
2               No  NaN
3              Yes  NaN
4               No  NaN
..             ...  ...
995             No  NaN
996             No  NaN
997             No  NaN
998             No  NaN
999             No  NaN

[1000 rows x 40 columns]
```

```
data.describe()
```

```
       months_as_customer         age  policy_number
policy_deductable  \
count         1000.000000  1000.000000    1000.000000
1000.000000
mean           203.954000    38.948000  546238.648000
1136.000000
std            115.113174     9.140287  257063.005276
611.864673
min              0.000000    19.000000  100804.000000
500.000000
25%            115.750000    32.000000  335980.250000
500.000000
50%            199.500000    38.000000  533135.000000
1000.000000
75%            276.250000    44.000000  759099.750000
2000.000000
max            479.000000    64.000000  999435.000000
2000.000000

       policy_annual_premium  umbrella_limit    insured_zip  capital-
gains  \
count            1000.000000    1.000000e+03    1000.000000
1000.000000
mean             1256.406150    1.101000e+06  501214.488000
25126.100000
std               244.167395    2.297407e+06   71701.610941
27872.187708
min               433.330000   -1.000000e+06  430104.000000
0.000000
25%              1089.607500    0.000000e+00  448404.500000
0.000000
50%              1257.200000    0.000000e+00  466445.500000
0.000000
75%              1415.695000    0.000000e+00  603251.000000
51025.000000
max              2047.590000    1.000000e+07  620962.000000
100500.000000

        capital-loss  incident_hour_of_the_day
number_of_vehicles_involved  \
count    1000.000000               1000.000000
1000.00000
mean   -26793.700000                 11.644000
1.83900
std     28104.096686                  6.951373
1.01888
min   -111100.000000                  0.000000
1.00000
25%    -51500.000000                  6.000000
1.00000
```

```
50%    -23250.000000                 12.000000
1.00000
75%         0.000000                 17.000000
3.00000
max         0.000000                 23.000000
4.00000

       bodily_injuries    witnesses  total_claim_amount  injury_claim
\
count      1000.000000  1000.000000          1000.00000   1000.000000

mean          0.992000     1.487000         52761.94000   7433.420000

std           0.820127     1.111335         26401.53319   4880.951853

min           0.000000     0.000000           100.00000      0.000000

25%           0.000000     1.000000         41812.50000   4295.000000

50%           1.000000     1.000000         58055.00000   6775.000000

75%           2.000000     2.000000         70592.50000  11305.000000

max           2.000000     3.000000        114920.00000  21450.000000


       property_claim  vehicle_claim    auto_year  _c39
count     1000.000000    1000.000000  1000.000000   0.0
mean      7399.570000   37928.950000  2005.103000   NaN
std       4824.726179   18886.252893     6.015861   NaN
min          0.000000      70.000000  1995.000000   NaN
25%       4445.000000   30292.500000  2000.000000   NaN
50%       6750.000000   42100.000000  2005.000000   NaN
75%      10885.000000   50822.500000  2010.000000   NaN
max      23670.000000   79560.000000  2015.000000   NaN
```

```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 40 columns):
 #   Column                       Non-Null Count  Dtype
---  ------                       --------------  -----
 0   months_as_customer           1000 non-null   int64
 1   age                          1000 non-null   int64
 2   policy_number                1000 non-null   int64
 3   policy_bind_date             1000 non-null   object
 4   policy_state                 1000 non-null   object
 5   policy_csl                   1000 non-null   object
 6   policy_deductable            1000 non-null   int64
 7   policy_annual_premium        1000 non-null   float64
```

```
 8   umbrella_limit               1000 non-null   int64
 9   insured_zip                  1000 non-null   int64
 10  insured_sex                  1000 non-null   object
 11  insured_education_level      1000 non-null   object
 12  insured_occupation           1000 non-null   object
 13  insured_hobbies              1000 non-null   object
 14  insured_relationship         1000 non-null   object
 15  capital-gains                1000 non-null   int64
 16  capital-loss                 1000 non-null   int64
 17  incident_date                1000 non-null   object
 18  incident_type                1000 non-null   object
 19  collision_type               1000 non-null   object
 20  incident_severity            1000 non-null   object
 21  authorities_contacted        1000 non-null   object
 22  incident_state               1000 non-null   object
 23  incident_city                1000 non-null   object
 24  incident_location            1000 non-null   object
 25  incident_hour_of_the_day     1000 non-null   int64
 26  number_of_vehicles_involved  1000 non-null   int64
 27  property_damage              1000 non-null   object
 28  bodily_injuries              1000 non-null   int64
 29  witnesses                    1000 non-null   int64
 30  police_report_available      1000 non-null   object
 31  total_claim_amount           1000 non-null   int64
 32  injury_claim                 1000 non-null   int64
 33  property_claim               1000 non-null   int64
 34  vehicle_claim                1000 non-null   int64
 35  auto_make                    1000 non-null   object
 36  auto_model                   1000 non-null   object
 37  auto_year                    1000 non-null   int64
 38  fraud_reported               1000 non-null   object
 39  _c39                         0 non-null      float64
dtypes: float64(2), int64(17), object(21)
memory usage: 312.6+ KB
```

```
data.duplicated().sum()
```

```
0
```

```
data.drop(data.index[39], inplace=True)
data
```

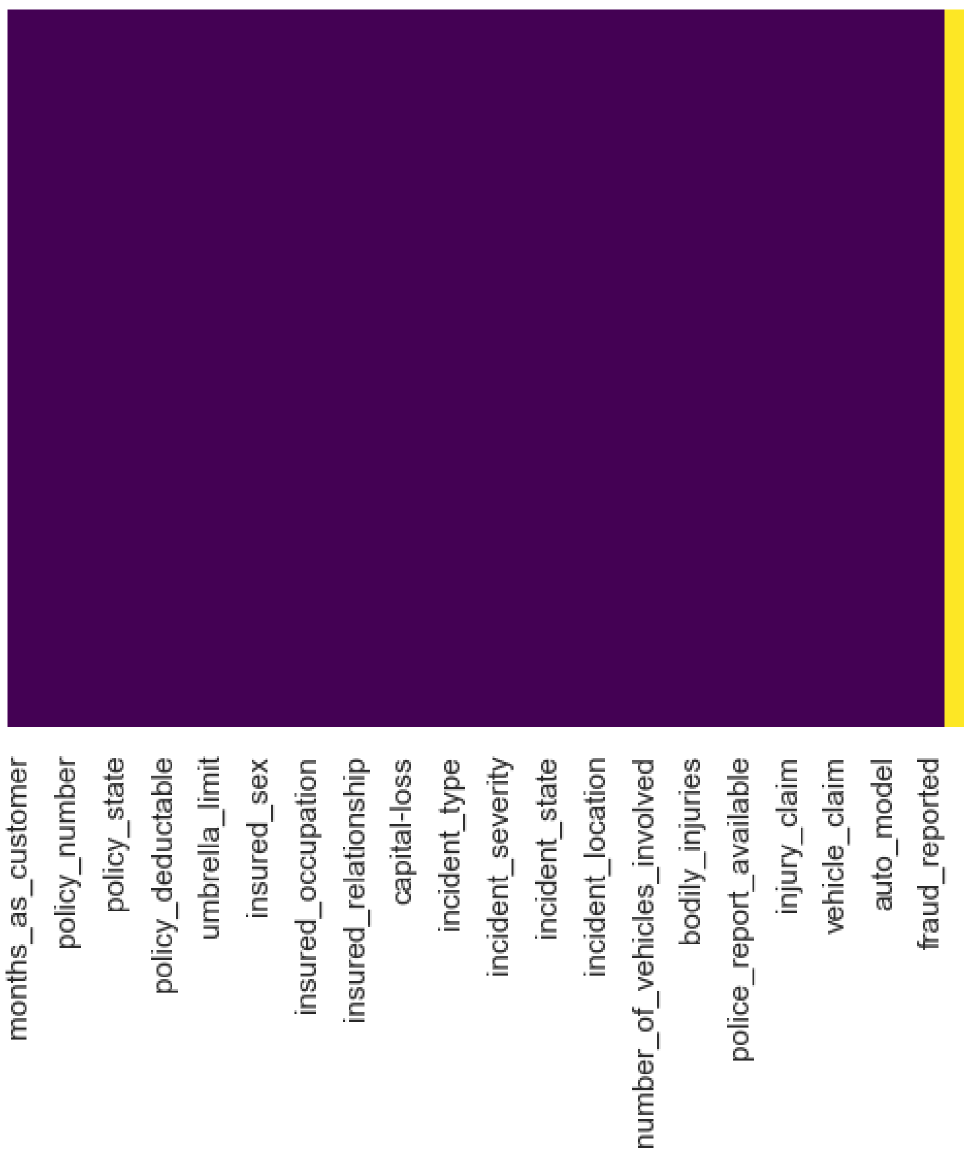
```
     months_as_customer  age  policy_number policy_bind_date
policy_state  \
0                   328   48         521585       17-10-2014
OH
1                   228   42         342868       27-06-2006
IN
2                   134   29         687698       06-09-2000
OH
```

```
3                    256   41          227811       25-05-1990
IL
4                    228   44          367455       06-06-2014
IL
..                   ...  ...             ...              ...
...
995                    3   38          941851       16-07-1991
OH
996                  285   41          186934       05-01-2014
IL
997                  130   34          918516       17-02-2003
OH
998                  458   62          533940       18-11-2011
IL
999                  456   60          556080       11-11-1996
OH

    policy_csl  policy_deductable  policy_annual_premium
umbrella_limit  \
0      250/500               1000                1406.91
0
1      250/500               2000                1197.22
5000000
2      100/300               2000                1413.14
5000000
3      250/500               2000                1415.74
6000000
4     500/1000               1000                1583.91
6000000
..         ...                ...                    ...             .
..
995   500/1000               1000                1310.80
0
996    100/300               1000                1436.79
0
997    250/500                500                1383.49
3000000
998   500/1000               2000                1356.92
5000000
999    250/500               1000                 766.19
0

     insured_zip  ... police_report_available total_claim_amount
injury_claim  \
0         466132  ...                     YES              71610
6510
1         468176  ...                                       5070
780
2         430632  ...                      NO              34650
```

```
7700
3        608117  ...                     NO            63400
6340
4        610706  ...                     NO             6500
1300
..          ...  ...                    ...              ...
...
995      431289  ...                                   87200
17440
996      608177  ...                                  108480
18080
997      442797  ...                    YES            67500
7500
998      441714  ...                    YES            46980
5220
999      612260  ...                                    5060
460

    property_claim vehicle_claim   auto_make  auto_model auto_year  \
0            13020         52080        Saab         92x      2004
1              780          3510    Mercedes        E400      2007
2             3850         23100       Dodge         RAM      2007
3             6340         50720   Chevrolet       Tahoe      2014
4              650          4550      Accura         RSX      2009
..             ...           ...         ...         ...       ...
995           8720         61040       Honda      Accord      2006
996          18080         72320  Volkswagen      Passat      2015
997           7500         52500      Suburu     Impreza      1996
998           5220         36540        Audi          A5      1998
999            920          3680    Mercedes        E400      2007

    fraud_reported _c39
0              Yes  NaN
1              Yes  NaN
2               No  NaN
3              Yes  NaN
4               No  NaN
..             ...  ...
995             No  NaN
996             No  NaN
997             No  NaN
998             No  NaN
999             No  NaN

[999 rows x 40 columns]
```

VISUALIZING THE DATA

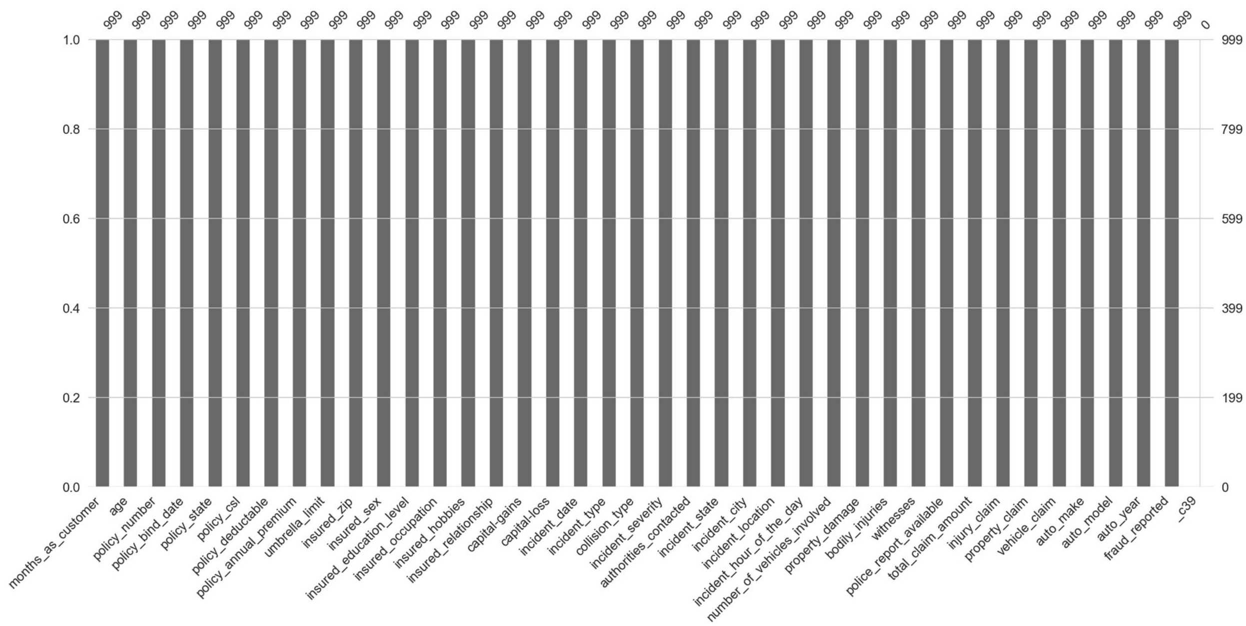
```
sns.heatmap(data.isnull(),yticklabels=False,cbar=False,cmap='viridis')
plt.show()
```



VISUALIZING THE NULL VALUES USING MISSINGNO

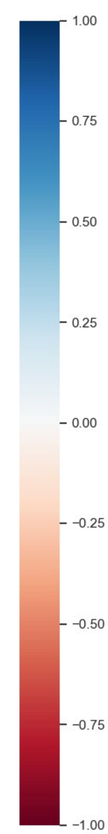
```
import missingno as msno

msno.bar(data)
```

```
<AxesSubplot:>
```



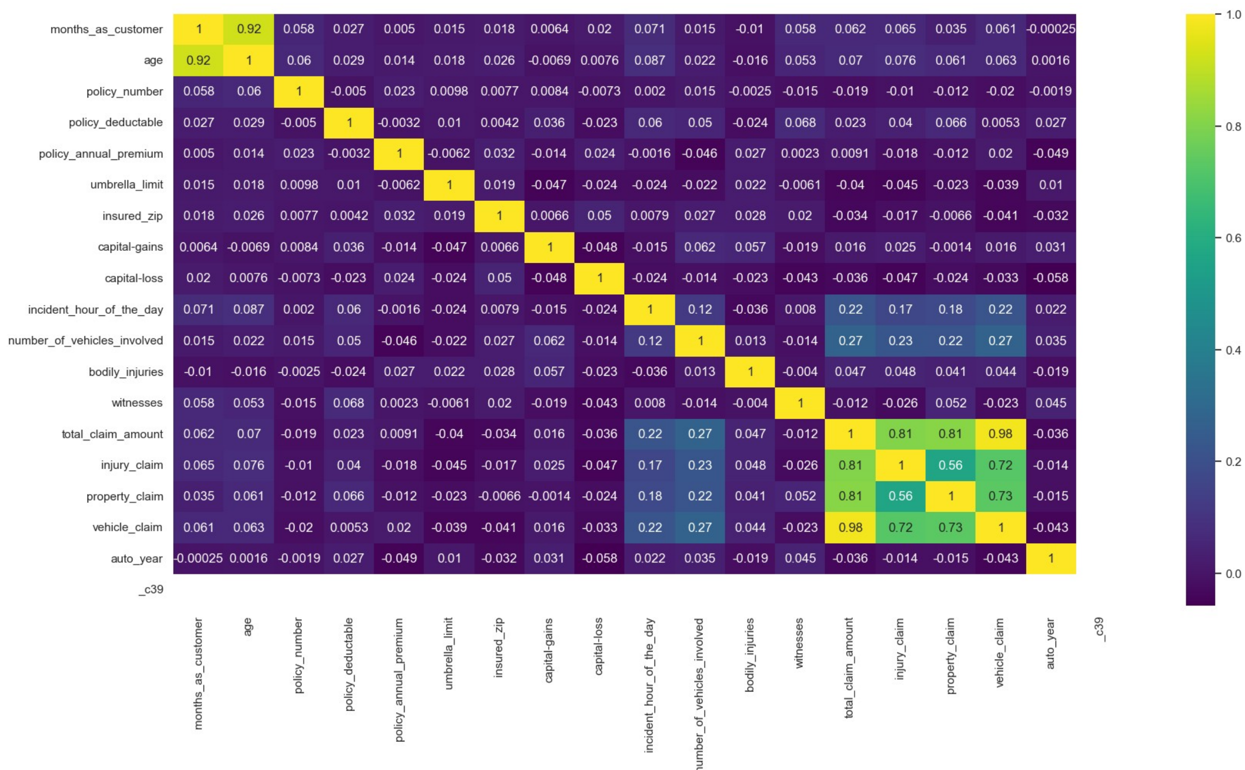

```
msno.heatmap(data)
```

```
<AxesSubplot:>
```

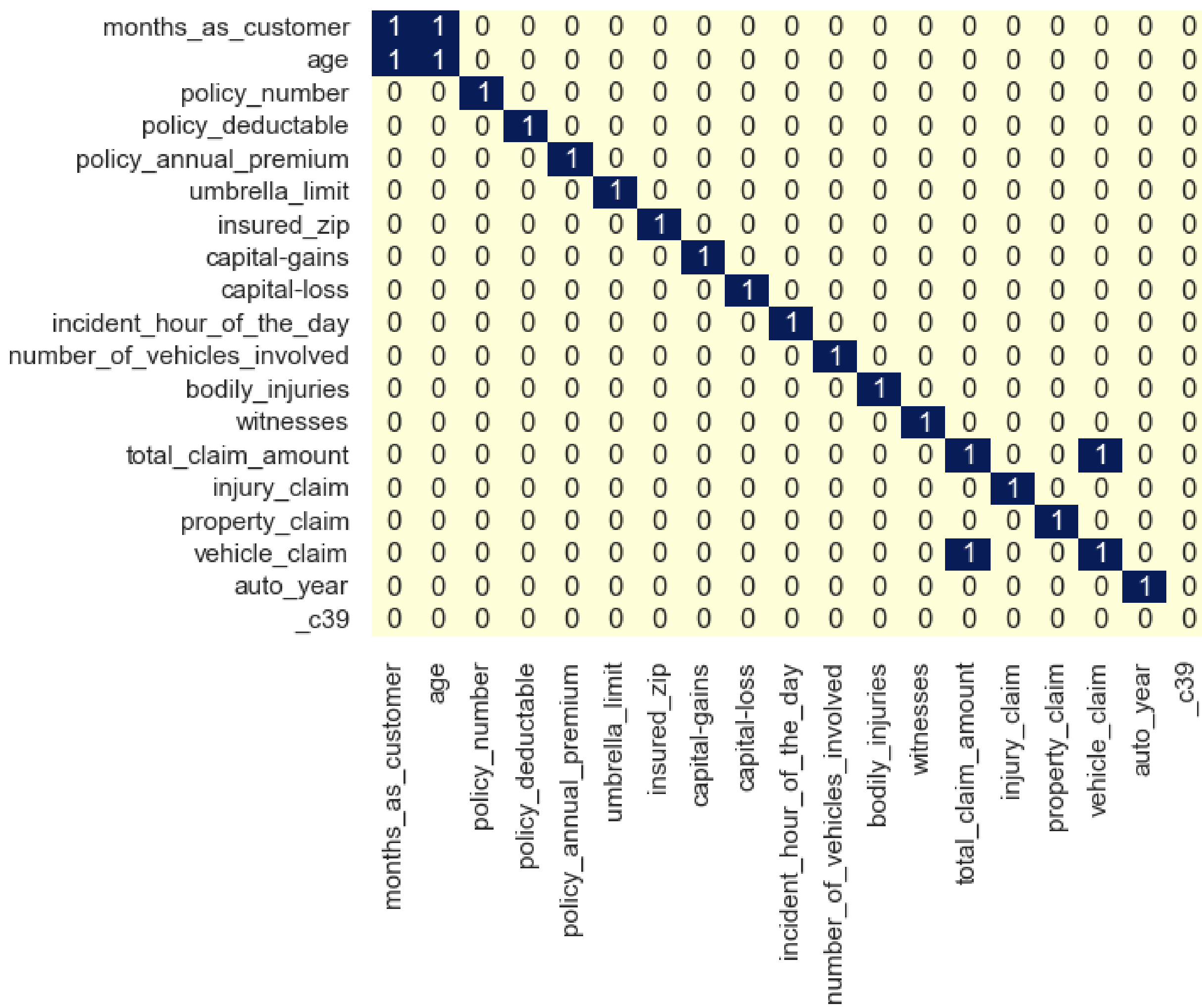


Data correlation

```
plt.figure(figsize=(20,10))
corr = data.corr()
sns.heatmap(data.corr(), cmap="viridis", annot=True)
plt.show()
```



```
sns.heatmap(data.corr() > 0.9, annot=True, cbar=False,cmap="YlGnBu")
plt.show()
```

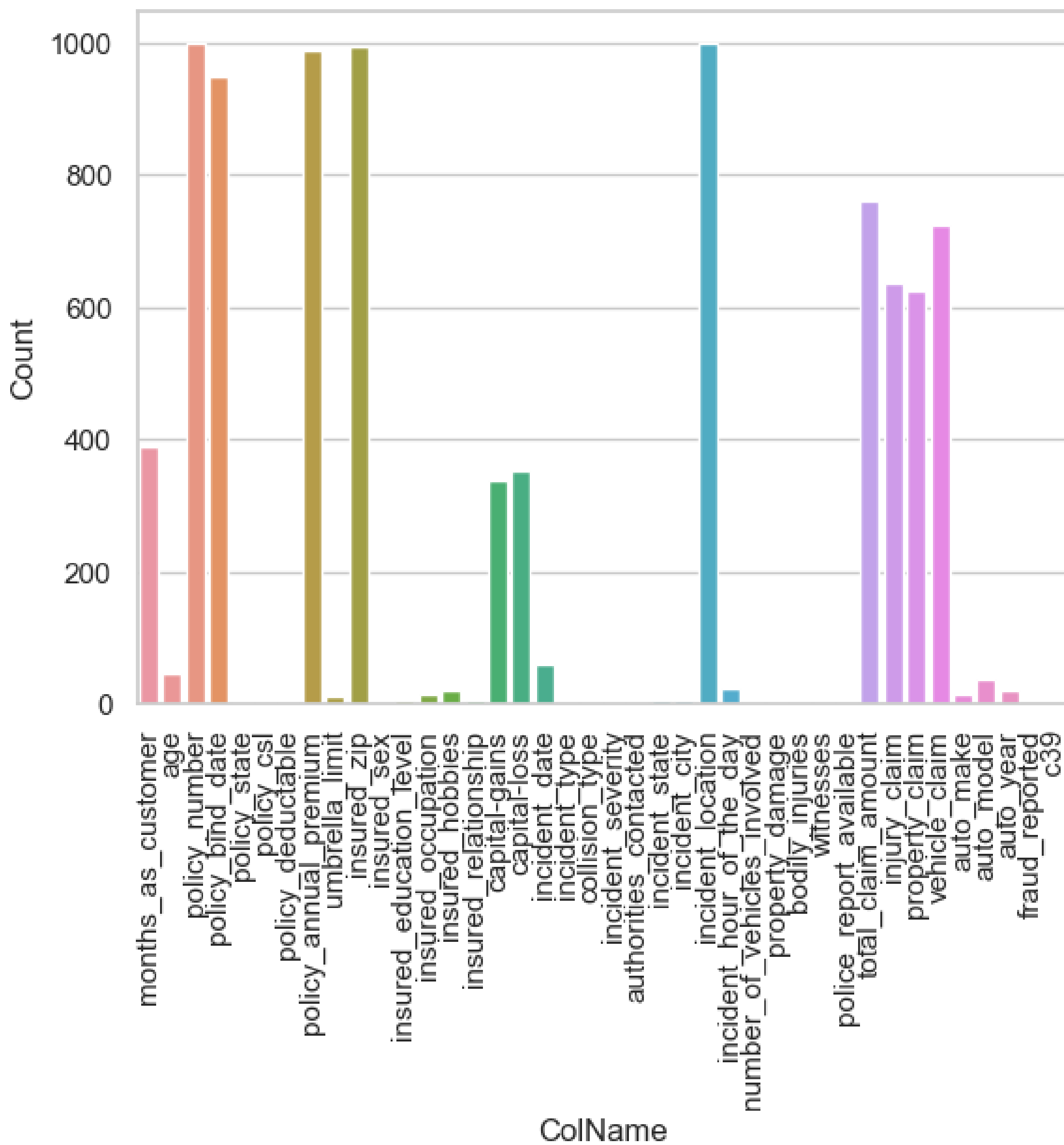



```
data.corr()['policy_deductable']
```

```
months_as_customer             0.026777
age                            0.028982
policy_number                 -0.004978
policy_deductable              1.000000
policy_annual_premium         -0.003223
umbrella_limit                 0.010377
insured_zip                    0.004161
capital-gains                  0.036117
capital-loss                  -0.022574
incident_hour_of_the_day       0.059858
number_of_vehicles_involved    0.050401
bodily_injuries               -0.024055
witnesses                      0.068157
total_claim_amount             0.023145
injury_claim                   0.039887
property_claim                 0.065609
vehicle_claim                  0.005292
```

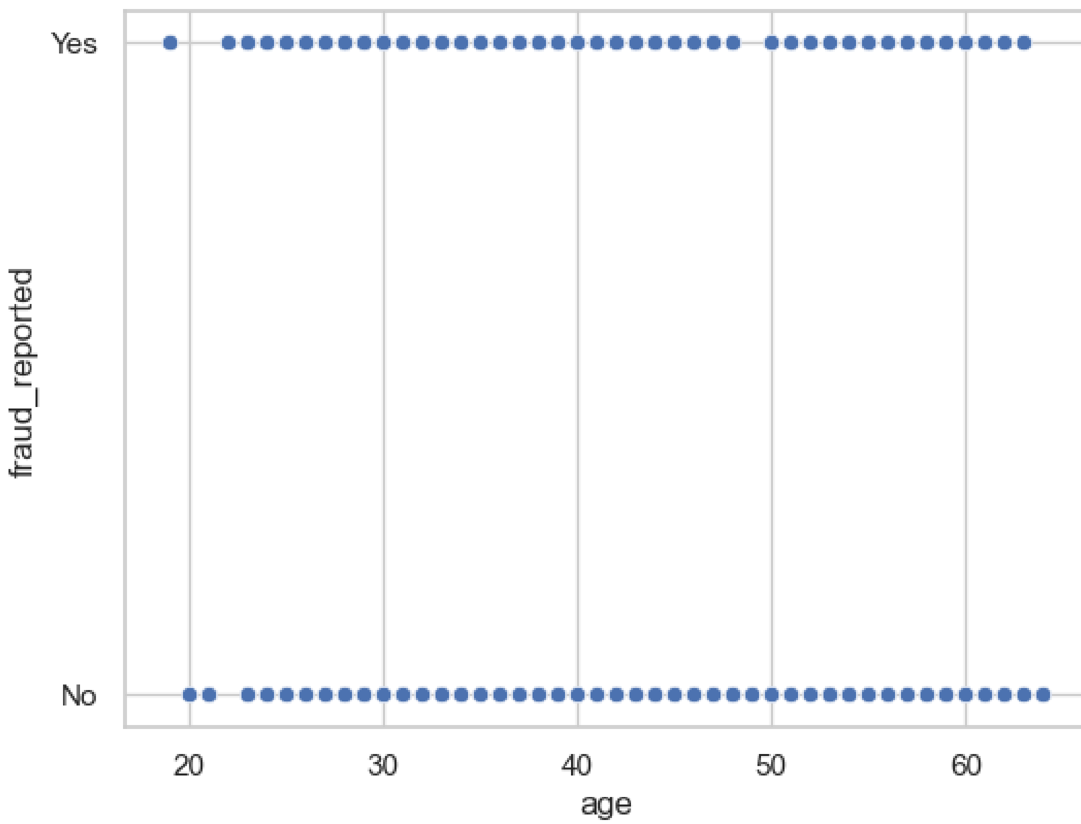
```
auto_year                          0.027153
_c39                                    NaN
Name: policy_deductable, dtype: float64
```

```
unique=data.nunique().to_frame()
unique.columns=['Count']
unique.index.names=['ColName']
unique=unique.reset_index()
sns.set(style='whitegrid',color_codes='True')
sns.barplot(x='ColName', y = 'Count', data = unique)
plt.xticks(rotation=90)
plt.show()
```



```
sns.scatterplot(x=data['age'],y=data['fraud_reported'])
```

```
<AxesSubplot:xlabel='age', ylabel='fraud_reported'>
```




```
#drop columns that are not used in our project


df=data.drop(columns
=['policy_number','age','policy_bind_date','policy_state','insured_zip
','incident_date','incident_state',

'incident_city','incident_location','insured_hobbies','auto_make','aut
o_model','auto_year','_c39','total_claim_amount'])


df
```

```
ColName  months_as_customer policy_csl  policy_deductable  \
0                       328    250/500               1000
1                       228    250/500               2000
2                       134    100/300               2000
```

```
3                       256    250/500               2000
4                       228   500/1000               1000
..                      ...        ...                ...
995                       3   500/1000               1000
996                     285    100/300               1000
997                     130    250/500                500
998                     458   500/1000               2000
999                     456    250/500               1000

ColName  policy_annual_premium  umbrella_limit insured_sex  \
0                      1406.91               0        MALE
1                      1197.22         5000000        MALE
2                      1413.14         5000000      FEMALE
3                      1415.74         6000000      FEMALE
4                      1583.91         6000000        MALE
..                         ...             ...         ...
995                    1310.80               0      FEMALE
996                    1436.79               0      FEMALE
997                    1383.49         3000000      FEMALE
998                    1356.92         5000000        MALE
999                     766.19               0      FEMALE

ColName insured_education_level insured_occupation
insured_relationship  \
0                            MD       craft-repair
husband
1                            MD  machine-op-inspct       other-
relative
2                           PhD              sales             own-
child
3                           PhD       armed-forces
unmarried
4                     Associate              sales
unmarried
..                          ...                ...                  ..
.
995                     Masters       craft-repair
unmarried
996                         PhD     prof-specialty
wife
997                     Masters       armed-forces       other-
relative
998                   Associate  handlers-cleaners
wife
999                   Associate              sales
husband

ColName  capital-gains  ...  incident_hour_of_the_day  \
0                53300  ...                         5
1                    0  ...                         8
```

```
2                 35100  ...                             7
3                 48900  ...                             5
4                 66000  ...                            20
..                  ...  ...                           ...
995                   0  ...                            20
996               70900  ...                            23
997               35100  ...                             4
998                   0  ...                             2
999                   0  ...                             6

ColName number_of_vehicles_involved property_damage bodily_injuries
witnesses  \
0                                 1             YES               1
2
1                                 1                               0
0
2                                 3              NO               2
3
3                                 1                               1
2
4                                 1              NO               0
1
..                              ...             ...             ...
...
995                               1             YES               0
1
996                               1             YES               2
3
997                               3                               2
3
998                               1                               0
1
999                               1                               0
3

ColName  police_report_available  injury_claim property_claim
vehicle_claim  \
0                            YES          6510          13020
52080
1                                          780            780
3510
2                             NO          7700           3850
23100
3                             NO          6340           6340
50720
4                             NO          1300            650
4550
..                           ...           ...            ...
...
```

```
995                                          17440          8720
61040
996                                          18080         18080
72320
997                        YES                7500          7500
52500
998                        YES                5220          5220
36540
999                                            460           920
3680

ColName  fraud_reported
0                   Yes
1                   Yes
2                    No
3                   Yes
4                    No
..                  ...
995                  No
996                  No
997                  No
998                  No
999                  No

[999 rows x 25 columns]
```

```
df.dtypes
```

```
ColName
months_as_customer              int64
policy_csl                     object
policy_deductable               int64
policy_annual_premium         float64
umbrella_limit                  int64
insured_sex                    object
insured_education_level        object
insured_occupation             object
insured_relationship           object
capital-gains                   int64
capital-loss                    int64
incident_type                  object
collision_type                 object
incident_severity              object
authorities_contacted          object
incident_hour_of_the_day        int64
number_of_vehicles_involved     int64
property_damage                object
bodily_injuries                 int64
witnesses                       int64
police_report_available        object
```

```
injury_claim                           int64
property_claim                         int64
vehicle_claim                          int64
fraud_reported                        object
dtype: object
```

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

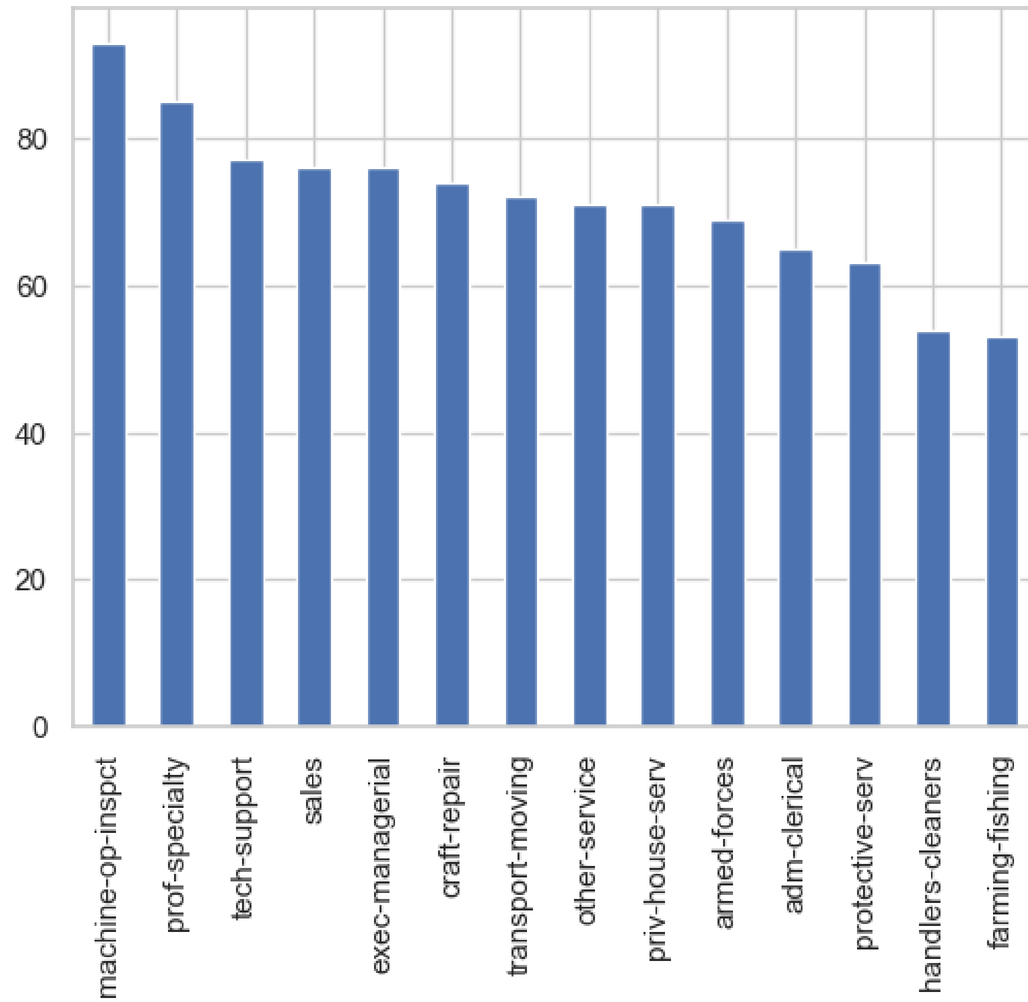
```
250/500     350
100/300     349
500/1000    300
Name: policy_csl, dtype: int64
```

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

```
High School    160
JD             160
Associate      145
MD             144
Masters        143
PhD            125
College        122
Name: insured_education_level, dtype: int64
```

```
df['insured_occupation'].value_counts().plot(kind='bar')
```

```
<AxesSubplot:>
```




```
df.corr()

ColName                      months_as_customer  policy_deductable  \
ColName
months_as_customer                     1.000000           0.026777
policy_deductable                      0.026777           1.000000
policy_annual_premium                  0.005019          -0.003223
umbrella_limit                         0.015480           0.010377
capital-gains                          0.006438           0.036117
capital-loss                           0.020260          -0.022574
incident_hour_of_the_day               0.070635           0.059858
number_of_vehicles_involved            0.014705           0.050401
bodily_injuries                       -0.010221          -0.024055
witnesses                              0.058496           0.068157
injury_claim                           0.065377           0.039887
property_claim                         0.034981           0.065609
vehicle_claim                          0.061014           0.005292
```

```
ColName                      policy_annual_premium  umbrella_limit  \
ColName
months_as_customer                        0.005019        0.015480
policy_deductable                        -0.003223        0.010377
policy_annual_premium                     1.000000       -0.006236
umbrella_limit                           -0.006236        1.000000
capital-gains                            -0.013763       -0.046887
capital-loss                              0.023535       -0.023611
incident_hour_of_the_day                 -0.001554       -0.023802
number_of_vehicles_involved              -0.045988       -0.021675
bodily_injuries                           0.026828        0.022181
witnesses                                 0.002302       -0.006091
injury_claim                             -0.017654       -0.045083
property_claim                           -0.011674       -0.023447
vehicle_claim                             0.020246       -0.038580

ColName                      capital-gains  capital-loss  \
ColName
months_as_customer                0.006438      0.020260
policy_deductable                 0.036117     -0.022574
policy_annual_premium            -0.013763      0.023535
umbrella_limit                   -0.046887     -0.023611
capital-gains                     1.000000     -0.047744
capital-loss                     -0.047744      1.000000
incident_hour_of_the_day         -0.015496     -0.024029
number_of_vehicles_involved       0.062378     -0.014120
bodily_injuries                   0.056907     -0.023290
witnesses                        -0.018819     -0.042690
injury_claim                      0.025345     -0.046780
property_claim                   -0.001397     -0.023582
vehicle_claim                     0.015826     -0.032698

ColName                      incident_hour_of_the_day  \
ColName
months_as_customer                           0.070635
policy_deductable                            0.059858
policy_annual_premium                       -0.001554
umbrella_limit                              -0.023802
capital-gains                               -0.015496
capital-loss                                -0.024029
incident_hour_of_the_day                     1.000000
number_of_vehicles_involved                  0.120000
bodily_injuries                             -0.035944
witnesses                                    0.008039
injury_claim                                 0.166704
property_claim                               0.180502
vehicle_claim                                0.215777

ColName                      number_of_vehicles_involved
```

```
bodily_injuries  \
ColName

months_as_customer                              0.014705        -
0.010221
policy_deductable                               0.050401        -
0.024055
policy_annual_premium                          -0.045988
0.026828
umbrella_limit                                 -0.021675
0.022181
capital-gains                                   0.062378
0.056907
capital-loss                                   -0.014120        -
0.023290
incident_hour_of_the_day                        0.120000        -
0.035944
number_of_vehicles_involved                     1.000000
0.013046
bodily_injuries                                 0.013046
1.000000
witnesses                                      -0.013563        -
0.003962
injury_claim                                    0.225378
0.048237
property_claim                                  0.219824
0.040680
vehicle_claim                                   0.269500
0.043504

ColName                      witnesses  injury_claim
property_claim  \
ColName

months_as_customer            0.058496      0.065377        0.034981

policy_deductable             0.068157      0.039887        0.065609

policy_annual_premium         0.002302     -0.017654       -0.011674

umbrella_limit               -0.006091     -0.045083       -0.023447

capital-gains                -0.018819      0.025345       -0.001397

capital-loss                 -0.042690     -0.046780       -0.023582

incident_hour_of_the_day      0.008039      0.166704        0.180502

number_of_vehicles_involved  -0.013563      0.225378        0.219824
```

```
bodily_injuries                         -0.003962      0.048237         0.040680

witnesses                                1.000000     -0.025854         0.051701

injury_claim                            -0.025854      1.000000         0.563636

property_claim                           0.051701      0.563636         1.000000

vehicle_claim                           -0.022611      0.723051         0.732274


ColName                                      vehicle_claim
ColName
months_as_customer                                0.061014
policy_deductable                                 0.005292
policy_annual_premium                             0.020246
umbrella_limit                                   -0.038580
capital-gains                                     0.015826
capital-loss                                     -0.032698
incident_hour_of_the_day                          0.215777
number_of_vehicles_involved                       0.269500
bodily_injuries                                   0.043504
witnesses                                        -0.022611
injury_claim                                      0.723051
property_claim                                    0.732274
vehicle_claim                                     1.000000
```

```
df.columns
```

```
Index(['months_as_customer', 'policy_csl', 'policy_deductable',
       'policy_annual_premium', 'umbrella_limit', 'insured_sex',
       'insured_education_level', 'insured_occupation',
'insured_relationship',
       'capital-gains', 'capital-loss', 'incident_type',
'collision_type',
       'incident_severity', 'authorities_contacted',
       'incident_hour_of_the_day', 'number_of_vehicles_involved',
       'property_damage', 'bodily_injuries', 'witnesses',
       'police_report_available', 'injury_claim', 'property_claim',
       'vehicle_claim', 'fraud_reported'],
      dtype='object', name='ColName')
```

```
df=df.replace('%',' ',regex=True)
df.head()
```

```
ColName  months_as_customer policy_csl  policy_deductable  \
0                       328    250/500               1000
1                       228    250/500               2000
2                       134    100/300               2000
3                       256    250/500               2000
4                       228   500/1000               1000
```

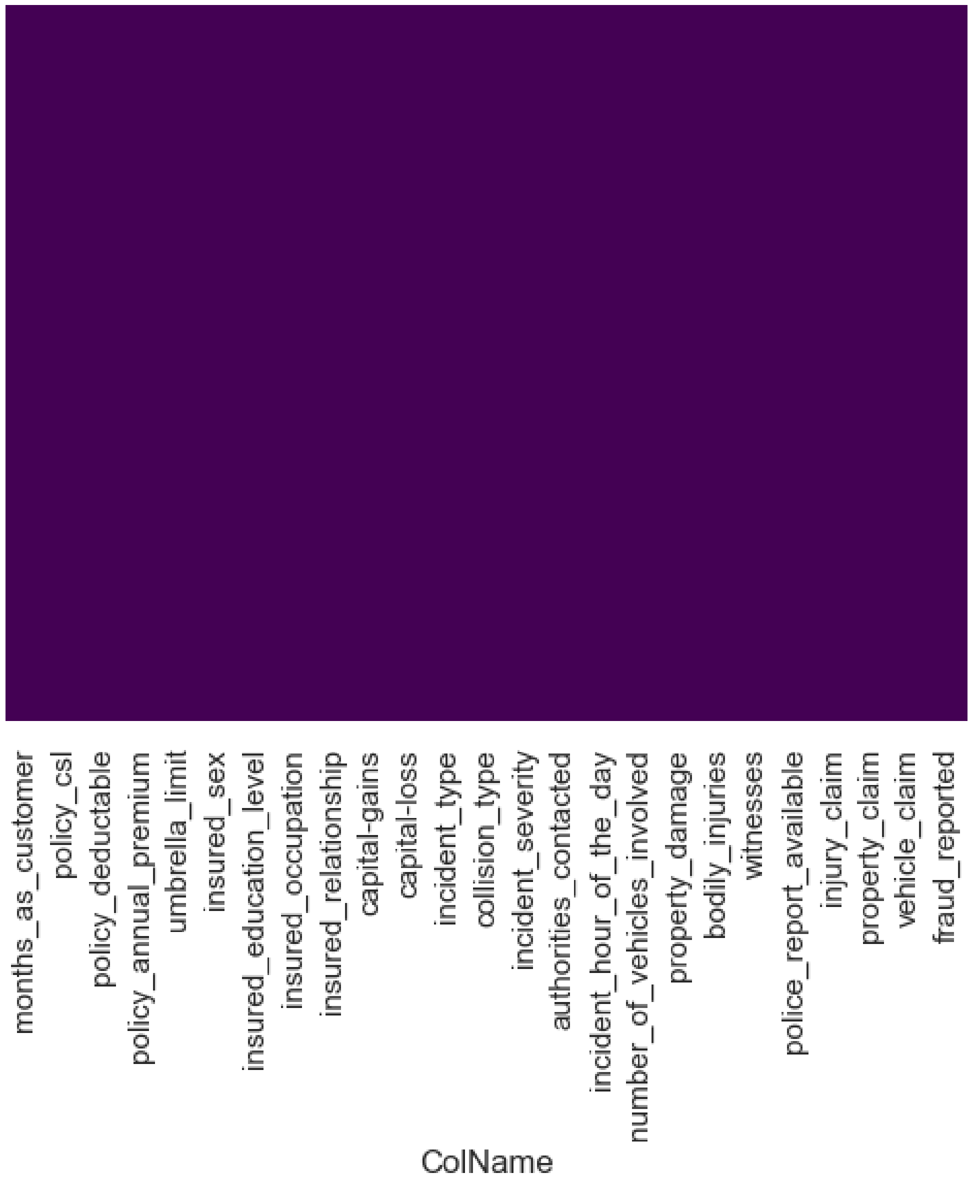
```
ColName  policy_annual_premium  umbrella_limit insured_sex  \
0                      1406.91               0        MALE
1                      1197.22         5000000        MALE
2                      1413.14         5000000      FEMALE
3                      1415.74         6000000      FEMALE
4                      1583.91         6000000        MALE

ColName insured_education_level insured_occupation
insured_relationship  \
0                            MD       craft-repair
husband
1                            MD  machine-op-inspct       other-
relative
2                           PhD              sales           own-
child
3                           PhD       armed-forces
unmarried
4                     Associate              sales
unmarried

ColName  capital-gains  ...  incident_hour_of_the_day  \
0                53300  ...                         5
1                    0  ...                         8
2                35100  ...                         7
3                48900  ...                         5
4                66000  ...                        20

ColName number_of_vehicles_involved property_damage bodily_injuries
witnesses  \
0                                 1             YES               1
2
1                                 1                               0
0
2                                 3              NO               2
3
3                                 1                               1
2
4                                 1              NO               0
1

ColName  police_report_available  injury_claim property_claim
vehicle_claim  \
0                            YES          6510          13020
52080
1                                          780            780
3510
2                             NO          7700           3850
23100
3                             NO          6340           6340
```

```
50720
4                              NO          1300            650
4550

ColName  fraud_reported
0                   Yes
1                   Yes
2                    No
3                   Yes
4                    No

[5 rows x 25 columns]
```

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

```
ColName
months_as_customer             0
policy_csl                     0
policy_deductable              0
policy_annual_premium          0
umbrella_limit                 0
insured_sex                    0
insured_education_level        0
insured_occupation             0
insured_relationship           0
capital-gains                  0
capital-loss                   0
incident_type                  0
collision_type                 0
incident_severity              0
authorities_contacted          0
incident_hour_of_the_day       0
number_of_vehicles_involved    0
property_damage                0
bodily_injuries                0
witnesses                      0
police_report_available        0
injury_claim                   0
property_claim                 0
vehicle_claim                  0
fraud_reported                 0
dtype: int64
```

```
sns.heatmap(df.isnull(),yticklabels=False,cbar=False,cmap='viridis')
plt.show()
```




```
from sklearn.preprocessing import LabelEncoder

le=LabelEncoder()
df['insured_sex']=le.fit_transform(df['insured_sex'])
df['insured_sex'].unique()

df['policy_csl']=le.fit_transform(df['policy_csl'])
df['policy_csl'].unique().astype(float)

df['insured_education_level']=le.fit_transform(df['insured_education_l
evel'])
df['insured_education_level'].unique().astype(float)
```

```
df['insured_occupation']=le.fit_transform(df['insured_occupation'])
df['insured_occupation'].unique().astype(float)



df['insured_relationship']=le.fit_transform(df['insured_relationship']
)
df['insured_relationship'].unique().astype(float)

df['incident_type']=le.fit_transform(df['incident_type'])
df['incident_type'].unique().astype(float)

df['collision_type']=le.fit_transform(df['collision_type'])
df['collision_type'].unique().astype(float)

df['incident_severity']=le.fit_transform(df['incident_severity'])
df['incident_severity'].unique().astype(float)

df['authorities_contacted']=le.fit_transform(df['authorities_contacted
'])
df['authorities_contacted'].unique()

df['property_damage']=le.fit_transform(df['property_damage'])
df['property_damage'].unique().astype(float)


df['police_report_available']=le.fit_transform(df['police_report_avail
able'])
df['police_report_available'].unique().astype(float)


df['fraud_reported']=le.fit_transform(df['fraud_reported'])
df['fraud_reported'].unique().astype(float)
```

```
array([1., 0.])
```

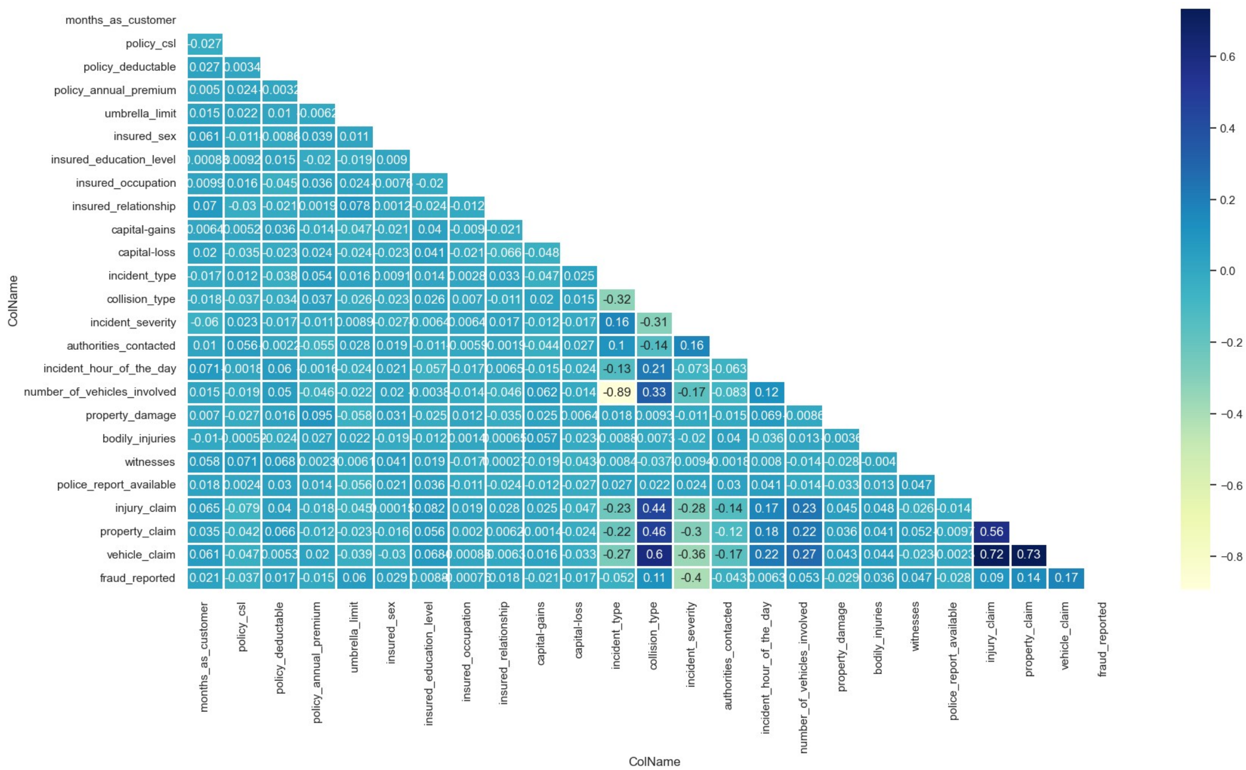
```
df=df.dropna(how='any')
```

```
df.shape
```

```
(999, 25)
```

```
#sns.heatmap(df.isnull(), cbar=False)
#plt.show()




plt.figure(figsize=(20,10))
corr = df.corr()
```

```
mask=np.triu(np.ones_like(corr,dtype=bool))

sns.heatmap(data=corr, mask=mask,
cmap="YlGnBu",annot=True,linewidth=2)
plt.show()
```




```
df['collision_type'].replace('',np.nan, inplace=True)

df['property_damage'].replace('',np.nan, inplace=True)

df['police_report_available'].replace('',np.nan, inplace=True)

df.dropna(subset=['collision_type'],inplace=True)

df.dropna(subset=['property_damage'],inplace=True)

df.dropna(subset=['police_report_available'],inplace=True)

#df['fraud_reported']=pd.to_numeric(df['fraud_reported'],errors='coerc
e')

x=df.drop('fraud_reported',axis=1).astype(np.float)

y=df['fraud_reported']

from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
```

```
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import LabelEncoder
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import
accuracy_score,classification_report,confusion_matrix,r2_score
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
import warnings
warnings.filterwarnings("ignore")
```

```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,rando
m_state=1)
```

```
from sklearn.linear_model import LogisticRegression
reg = LogisticRegression()
reg.fit(x_train,y_train)
```

```
LogisticRegression()
```

```
y_pred=reg.predict(x_test)
from sklearn.metrics import
accuracy_score,classification_report,confusion_matrix,r2_score
print(classification_report(y_test,y_pred))
print(confusion_matrix(y_test,y_pred))
print("Training Score: ",reg.score(x_train,y_train)*100)
```

```
              precision    recall  f1-score   support

           0       0.76      0.99      0.86       152
           1       0.50      0.02      0.04        48

    accuracy                           0.76       200
   macro avg       0.63      0.51      0.45       200
weighted avg       0.70      0.76      0.67       200

[[151   1]
 [ 47   1]]
Training Score:  75.21902377972467
```

```
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

```
     Actual  Predicted
508       0          0
609       0          0
453       0          0
369       0          0
243       0          0
..      ...        ...
431       0          0
```

```
588        0          0
551        0          0
608        0          0
208        0          0

[200 rows x 2 columns]
```

```
print(accuracy_score(y_test,y_pred)*100)
```

```
76.0
```

```
from sklearn.model_selection import GridSearchCV
param = {
         'penalty':['l1','l2'],
         'C':[0.001, 0.01, 0.1, 1, 10, 50,70, 100]
}
lr= LogisticRegression(penalty='l1')
cv=GridSearchCV(reg,param,cv=5,n_jobs=-1)
cv.fit(x_train,y_train)
cv.predict(x_test)
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0,
       0, 0])
```

```
print("Best CV score", cv.best_score_*100)
```

```
Best CV score 75.22091194968554
```

```
import pickle
```

```
pickle.dump(reg,open('C:/Flask/reg_saved','wb'))
```

```
reg_load=pickle.load(open("C:/Flask/reg_saved",'rb'))
```

```python
import joblib

joblib.dump(reg,'model2')
```

```
['model2']
```

```python
a=joblib.load('model2')

a.predict(x_test)
```

```
array([0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
       0, 0])
```