```
import pandas as pd
import seaborn as sns

df = pd.read_excel("/content/drive/MyDrive/insurance_claim_updated.xlsx")



from google.colab import drive
drive.mount('/content/drive')


#get count of rows and columns in our dataset
df.shape
```

```
(10211, 39)
```

Understanding the data

Glimpse Of Data

```
#list of columns in our dataset
col_list = df.columns
print(col_list)
```

```
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'],
      dtype='object')
```

```
# Top 5 entries of the dataset
df.head()
```

	months_as_customer	age	policy_number	policy_bind_date	policy_state	policy_cs
0	5	37	939011	16-07-2002	IN	250/50

```
# Bottom 5 entries of the dataset
df.tail()
```

	months_as_customer	age	policy_number	policy_bind_date	policy_state	polic
10206	91	16	524932	23-02-2014	IN	1(
10207	266	29	128125	12-10-2006	OH	2!
10208	332	40	494839	21-03-2008	OH	1(
10209	316	48	122384	28-02-2004	IL	50(
10210	113	20	614699	05-10-1999	OH	1(

```
df.describe()
```

	months_as_customer	age	policy_number	policy_deductable	policy_anr
count	10211.000000	10211.000000	1.021100e+04	10211.000000	
mean	213.467927	39.050142	5.474680e+05	1159.044168	
std	133.639732	11.508964	3.034069e+05	621.773731	
min	0.000000	2.000000	4.410000e+02	500.000000	
25%	106.000000	31.000000	3.095050e+05	500.000000	
50%	202.000000	38.000000	5.364750e+05	1000.000000	
75%	303.000000	47.000000	7.717955e+05	2000.000000	
max	747.000000	79.000000	1.615353e+06	2000.000000	

```
#unique entries in each column of dataframe
df.nunique()
```

```
months_as_customer              614
age                              76
policy_number                 10163
policy_bind_date                951
policy_state                      3
policy_csl                        3
policy_deductable                 3
policy_annual_premium          9706
umbrella_limit                   11
```

```
insured_zip                   10045
insured_sex                       2
insured_education_level           7
insured_occupation               14
insured_hobbies                  20
insured_relationship              6
capital.gains                  4940
capital.loss                   4871
incident_date                    60
incident_type                     4
collision_type                    4
incident_severity                 4
authorities_contacted             5
incident_state                    7
incident_city                     7
incident_location              1000
incident_hour_of_the_day         25
number_of_vehicles_involved       6
property_damage                   3
bodily_injuries                   5
witnesses                         7
police_report_available           3
total_claim_amount             4816
injury_claim                   1825
property_claim                 1858
vehicle_claim                  4147
auto_make                        14
auto_model                       39
auto_year                        35
fraud_reported                    2
dtype: int64
```

```
cols = ['policy_state','policy_csl','policy_deductable','insured_sex','insured_education_l
        'incident_type','collision_type','incident_severity','authorities_contacted','prop
        'police_report_available','auto_make']
for i in cols:
  temp = df[i].unique()
  print(i,":",temp)
```

```
policy_state : ['IN' 'IL' 'OH']
policy_csl : ['250/500' '500/1000' '100/300']
policy_deductable : [ 500 1000 2000]
insured_sex : ['FEMALE' 'MALE']
insured_education_level : ['Associate' 'MD' 'High School' 'PhD' 'JD' 'Masters' 'Colle
insured_occupation : ['priv-house-serv' 'exec-managerial' 'farming-fishing' 'transpor
 'armed-forces' 'tech-support' 'protective-serv' 'prof-specialty'
 'machine-op-inspct' 'other-service' 'adm-clerical' 'handlers-cleaners'
 'craft-repair' 'sales']
incident_type : ['Single Vehicle Collision' 'Multi-vehicle Collision' 'Parked Car'
 'Vehicle Theft']
collision_type : ['Front Collision' 'Rear Collision' '?' 'Side Collision']
incident_severity : ['Minor Damage' 'Total Loss' 'Trivial Damage' 'Major Damage']
authorities_contacted : ['Other' 'None' 'Police' 'Fire' 'Ambulance']
property_damage : ['?' 'NO' 'YES']
police_report_available : ['YES' '?' 'NO']
auto_make : ['Saab' 'Chevrolet' 'Toyota' 'Honda' 'Accura' 'BMW' 'Audi' 'Suburu'
 'Dodge' 'Nissan' 'Jeep' 'Ford' 'Mercedes' 'Volkswagen']
```

Cleaning Data

```
df = df.replace(to_replace="Y",value="YES")
df = df.replace(to_replace="N",value="NO")


#df = df.drop(df[df.score < 50].index)
df = df.drop(['insured_hobbies'],axis=1)


#df.isnull().sum()
#looking for outliers
df['age'].describe()
```

```
count    10211.000000
mean        39.050142
std         11.508964
min          2.000000
25%         31.000000
50%         38.000000
75%         47.000000
max         79.000000
Name: age, dtype: float64
```

```
corelation = df.corr()


sns.set(rc = {'figure.figsize':(19,10)})
sns.heatmap(corelation,xticklabels=corelation.columns,
            yticklabels=corelation.columns,annot=True)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7f573f705e10>
```