DECEMBER 2021

This Business Report shall provide detailed explanation of how we approached each problem given in the assignment. It shall also provide relative resolution and explanation with regards to the problems

LIFE INSURANCE SALES - CAPSTONE BUSINESS REPORT

THAKUR ARUN SINGH

CONTENTS

Problem 1: Introduction of the business problem

The dataset belongs to a leading life insurance company. The company wants to predict the bonus for its agents so that it may design appropriate engagement activity for their high performing agents and up skill programs for low performing agents

PROBLEM 1.A

Defining problem statement

Resolution:

EXECUTIVE SUMMARY

Academics and practitioners have studied over the years models for predicting firm's various aspects, using statistical and machine-learning approaches. We are going to discuss one of the various aspects. An earlier sign that company employees are dissatisfied is the firm policies, work load, pay scale and bonuses. We are going to dive deeper and predict the bonus for its employees, so that we may designed an appropriate engagement activity for their high performing employees and up skill programs for low performing employees.

Project Approach

The work that we have completed:

- Merged data from other sources like, demographic information, account details etc. for further deep analysis
- Data Quality and preparation activities were performed like missing value treatment, imputation, data type conversions for homogeneity in the data set
- Performed EDA on the data to understand the data and to determine any outlier and treatment for the same
- Also used ANNOVA to understand if the model performance can be improved

PROBLEM 1.B

Need of the study/project

Resolution:

All Life Insurance companies offer various incentive plans for their employees, to boost their sales and to have higher balance of insurance plans, **"The higher the Insurance amount the higher the bonus pay out"**. However, not all companies get successful in the above Mantra. Some companies fail to have a better plan for bonus payouts. For such companies we should study the data and need to understand the requirements to implement the above plan. Where in, the company can predict the bonus for its employees and design appropriate engagement activity for their high performing employees and also have training programs for non/low performing employees.

PROBLEM 1.C

Understanding business/social opportunity

Resolution:

Based on our analyses, once we thoroughly study the data we will be able to understand the business better and take decisions. Here, after all the analyses which were perform we are able to come to collusion whether the given approach will work or not.

Problem 2: Data Report

PROBLEM 2.A

Understanding how data was collected in terms of time, frequency and methodology

Resolution:

- Looking at the data we can see that the data collected with a wide verity of age range from 18 years to 58 years.
- There is good mix of gender where we have 40% Female 60% and 60% Male
- 50% of the data consist of married people
- From the entire data we have about 35% of the people who are at Manager Level.
- About 49% of the people who took Life Insurance policy are salaried employees.

Describing the data:

- First we import all the necessary libraries in Python, and then import the data file which is 'LifeInsuranceSales'. Once we import the file we confirm whether the data has been uploaded correctly or not using 'head' function. Using this function we can view the data and all the columns and headers whether they are aligning correctly or not.
- Then using the 'shape' function we can understand how many row and columns are there in our data set.
- To check the data type of all the columns and also to check the null values, 'info' function. Has been used.
- To see the detail description of the data such as, Count, Mean, Median, Min, Max, Standard Deviations etc,
- Using the 'isnull' function, one can understand if there are any null values in the data set. And we do not have any null values in the existing data set.
- Using the 'dups' function we check for the duplicates and there were no duplicate values.
- We also identified the unique values in categorical data.

PROBLEM 2.B

Visual inspection of data (rows, columns, descriptive details)

Resolution:

To see if the data has been imported or not.

Out[3]:

	CustID	AgentBonus	Age	CustTenure	Channel	Occupation	EducationField	Gender	ExistingProdType	Designation	NumberOfPolicy	MaritalStatus	Montl
0	7000000	4409	22.0	4.0	Agent	Salaried	Graduate	Female	3	Manager	2.0	Single	
1	7000001	2214	11.0	2.0	Third Party Partner	Salaried	Graduate	Male	4	Manager	4.0	Divorced	
2	7000002	4273	26.0	4.0	Agent	Free Lancer	Post Graduate	Male	4	Exe	3.0	Unmarried	
3	7000003	1791	11.0	NaN	Third Party Partner	Salaried	Graduate	Fe male	3	Executive	3.0	Divorced	
4	7000004	2955	6.0	NaN	Agent	Small Business	UG	Male	3	Executive	4.0	Divorced	

To know the shape of the data, we can see that we have 4520 Rows and 23 Coloms

In [91]:
```
sales_df.shape
```
Out[91]: (4520, 23)

We can see the date distribution

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
AgentBonus	4520	NaN	NaN	NaN	4077.84	1403.32	1605	3027.75	3911.5	4867.25	9608
Age	4251	NaN	NaN	NaN	14.4947	9.03763	2	7	13	20	58
CustTenure	4294	NaN	NaN	NaN	14.469	8.96367	2	7	13	20	57
Channel	4520	3	Agent	3194	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Occupation	4520	5	Salaried	2192	NaN	NaN	NaN	NaN	NaN	NaN	NaN
EducationField	4520	7	Graduate	1870	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Gender	4520	3	Male	2688	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ExistingProdType	4520	NaN	NaN	NaN	3.68894	1.01577	1	3	4	4	6
Designation	4520	6	Manager	1620	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NumberOfPolicy	4475	NaN	NaN	NaN	3.56536	1.45593	1	2	4	5	6
MaritalStatus	4520	4	Married	2268	NaN	NaN	NaN	NaN	NaN	NaN	NaN
MonthlyIncome	4284	NaN	NaN	NaN	22890.3	4885.6	16009	19683.5	21606	24725	38456
Complaint	4520	NaN	NaN	NaN	0.287168	0.452491	0	0	0	1	1
ExistingPolicyTenure	4336	NaN	NaN	NaN	4.13007	3.34639	1	2	3	6	25
SumAssured	4366	NaN	NaN	Na	620000	246235	16853	43944	57897	75823	1.84E+

				N			6	3	6	6	06
Zone	4520	4	West	2566	NaN	NaN	NaN	NaN	NaN	NaN	NaN
PaymentMethod	4520	4	Half Yearly	2656	NaN	NaN	NaN	NaN	NaN	NaN	NaN
LastMonthCalls	4520	NaN	NaN	NaN	4.62699	3.62013	0	2	3	8	18
CustCareScore	4468	NaN	NaN	NaN	3.06759	1.38297	1	2	3	4	5

PROBLEM 2.C

Understanding of attributes (variable info, renaming if required)

Resolution:

Below we can see the variable info about the data – data types of respective columns comprises of float, integer and object.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4520 entries, 0 to 4519
Data columns (total 20 columns):
 #   Column                Non-Null Count  Dtype  
---  ------                --------------  -----  
 0   CustID                4520 non-null   int64  
 1   AgentBonus            4520 non-null   int64  
 2   Age                   4251 non-null   float64
 3   CustTenure            4294 non-null   float64
 4   Channel               4520 non-null   object 
 5   Occupation            4520 non-null   object 
 6   EducationField        4520 non-null   object 
 7   Gender                4520 non-null   object 
 8   ExistingProdType      4520 non-null   int64  
 9   Designation           4520 non-null   object 
 10  NumberOfPolicy        4475 non-null   float64
 11  MaritalStatus         4520 non-null   object 
 12  MonthlyIncome         4284 non-null   float64
 13  Complaint             4520 non-null   int64  
 14  ExistingPolicyTenure  4336 non-null   float64
 15  SumAssured            4366 non-null   float64
 16  Zone                  4520 non-null   object 
 17  PaymentMethod         4520 non-null   object 
 18  LastMonthCalls        4520 non-null   int64  
 19  CustCareScore         4468 non-null   float64
dtypes: float64(7), int64(5), object(8)
memory usage: 706.4+ KB
```

Converting some of the data types to Object data.

```
Channel : ['Agent' 'Third Party Partner' 'Online']
Occupation : ['Salaried' 'Free Lancer' 'Small Business' 'Laarge Business'
 'Large Business']
EducationField : ['Graduate' 'Post Graduate' 'UG' 'Under Graduate' 'Engineer' 'Diploma'
 'MBA']
Gender : ['Female' 'Male' 'Fe male']
Designation : ['Manager' 'Exe' 'Executive' 'VP' 'AVP' 'Senior Manager']
MaritalStatus : ['Single' 'Divorced' 'Unmarried' 'Married']
Zone : ['North' 'West' 'East' 'South']
PaymentMethod : ['Half Yearly' 'Yearly' 'Quarterly' 'Monthly']
```

Replacing the duplicate words in Gender, Occupation, EducationField & Designation variable

```
['Female' 'Male']
['Salaried' 'Free Lancer' 'Small Business' 'Large Business']
['Graduate' 'Post Graduate' 'Under Graduate' 'Engineer' 'Diploma' 'MBA']
['Single/Unmarried' 'Divorced' 'Married']
['Manager' 'Executive' 'AVP-VP' 'Senior Manager']
```

Problem 3: Exploratory data analysis

PROBLEM 3.A

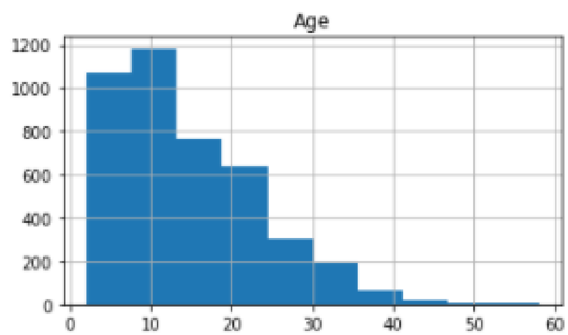
Univariate analysis (distribution and spread for every continuous attribute, distribution of data in categories for categorical ones)

Resolution: First we can see the data distribution

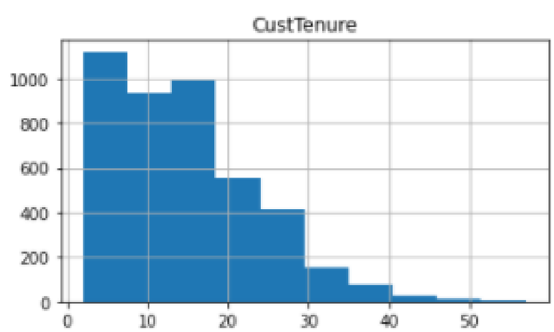
	count	unique	top	freq	mean	std	min	25%	50%	75%	max
AgentBonus	4520	NaN	NaN	NaN	4077.84	1403.32	1605	3027.75	3911.5	4867.25	9608
Age	4251	NaN	NaN	NaN	14.4947	9.03763	2	7	13	20	58
CustTenure	4294	NaN	NaN	NaN	14.469	8.96367	2	7	13	20	57
Channel	4520	3	Agent	3194	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Occupation	4520	5	Salaried	2192	NaN	NaN	NaN	NaN	NaN	NaN	NaN
EducationField	4520	7	Graduate	1870	NaN	NaN	NaN	NaN	NaN	NaN	NaN
Gender	4520	3	Male	2688	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ExistingProdType	4520	NaN	NaN	NaN	3.68894	1.01577	1	3	4	4	6
Designation	4520	6	Manager	1620	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NumberOfPolicy	4475	NaN	NaN	NaN	3.56536	1.45593	1	2	4	5	6
MaritalStatus	4520	4	Married	2268	NaN	NaN	NaN	NaN	NaN	NaN	NaN
MonthlyIncome	4284	NaN	NaN	NaN	22890.3	4885.6	16009	19683.5	21606	24725	38456
Complaint	4520	NaN	NaN	NaN	0.287168	0.452491	0	0	0	1	1
ExistingPolicyTenure	4336	NaN	NaN	NaN	4.13007	3.34639	1	2	3	6	25
SumAssured	4366	NaN	NaN	NaN	620000	246235	168536	439443	578976	758236	1.8385e+06
Zone	4520	4	West	2566	NaN	NaN	NaN	NaN	NaN	NaN	NaN
PaymentMethod	4520	4	Half Yearly	2656	NaN	NaN	NaN	NaN	NaN	NaN	NaN
LastMonthCalls	4520	NaN	NaN	NaN	4.62699	3.62013	0	2	3	8	18
CustCareScore	4468	NaN	NaN	NaN	3.06759	1.38297	1	2	3	4	5

Agent Bonus: From the below graph we can see that the majority of the bonus falls between 3,000 to 4,000.

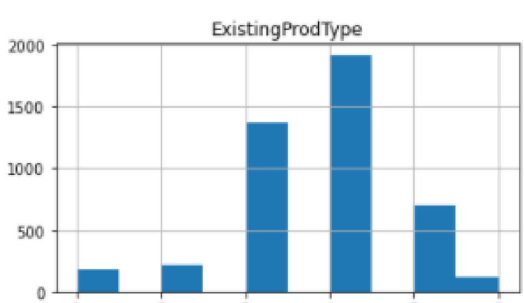
Age: Below the graph shows that the average age of the customer is about 30-35 years.



Customer Tenure: Below graph shows that there are over 1000 customer who are loyal customers for at least 10 years.

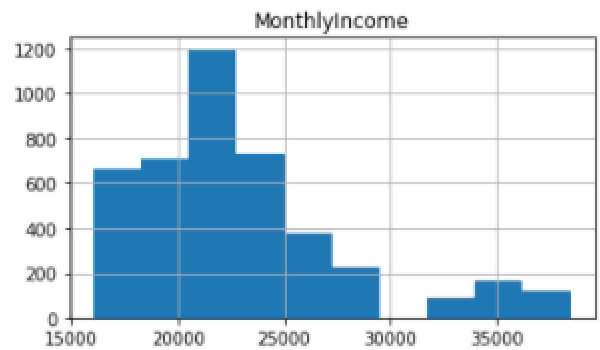


Existing Product Type: Below graph shows that there at about 2000 people who hold at least 4 products

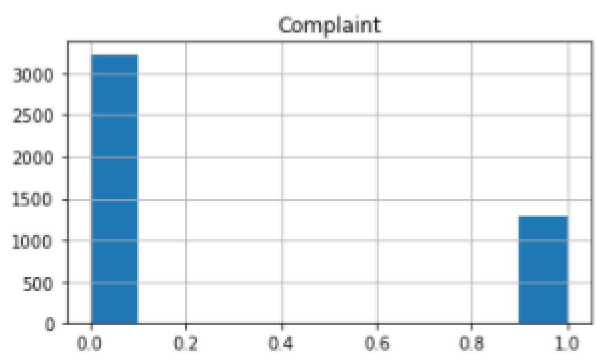


Number of Policies: Below graph shows that at least 1000 people hold 4 policies each.

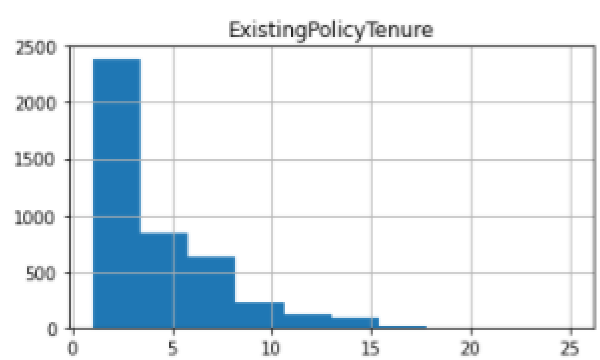
Monthly Income: About 1200 people have a monthly income between 20,000 – 25,000.



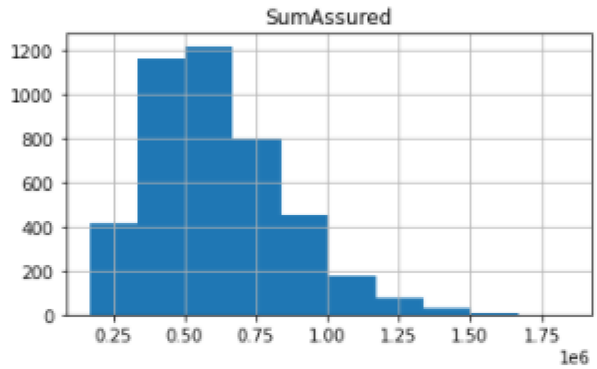
Complaints: There were about 3000+ people who had complained between 0-1 times.



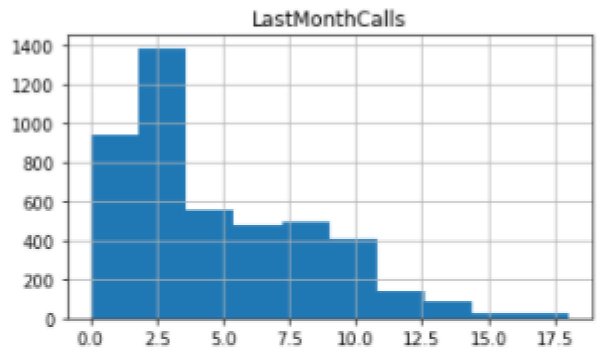
Existing Policy Tenure: There are about 2500 customer who took policies at least a year ago.



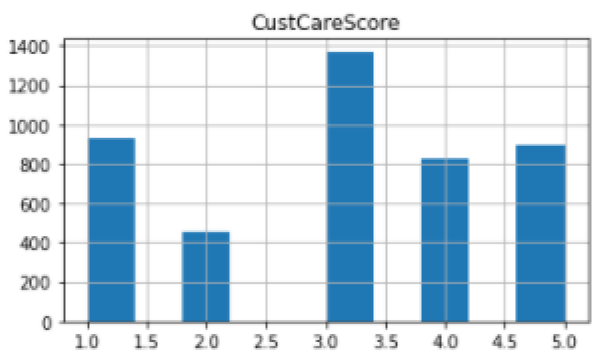
Sum Assured: The major count of the sum assured is between 400K and 600K



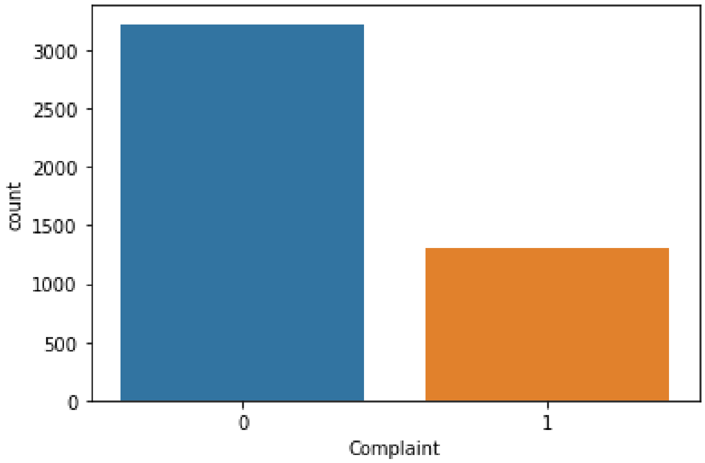
Last Month Calls: We have received at least 2 calls from nearly 1400 customers



Customer Care Score: Marjory of the customer lies between the Customer care score of 3 – 3.5.

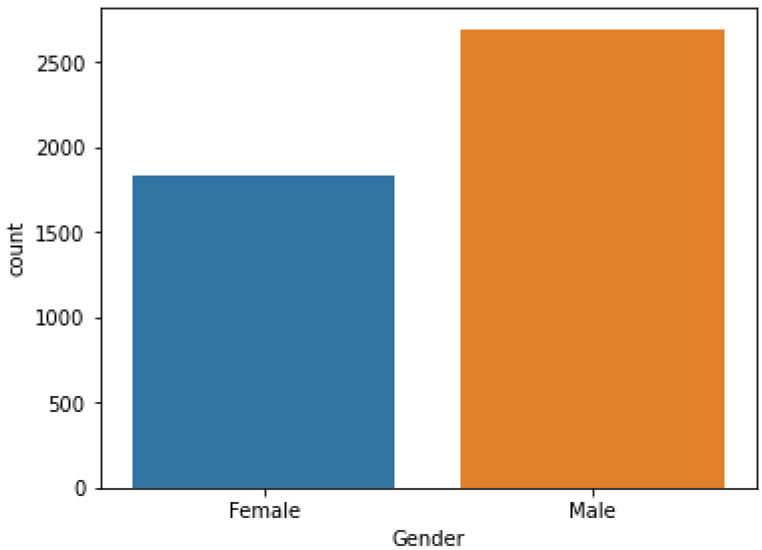


Complaints: Below graph shows the count of the customers who complained and not complained.



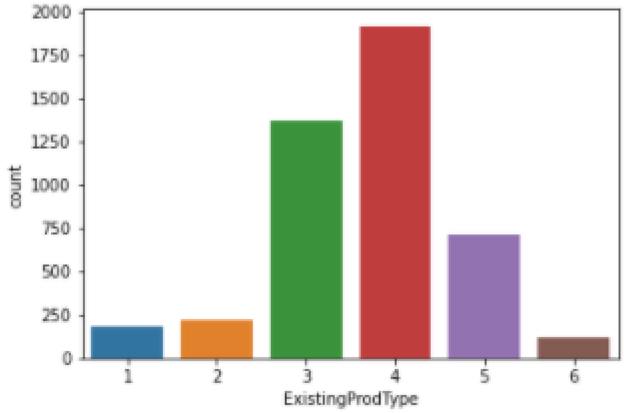

```
0    0.712832
1    0.287168
Name: Complaint, dtype: float64
```

Gender: Below graph gives me the count of Gender



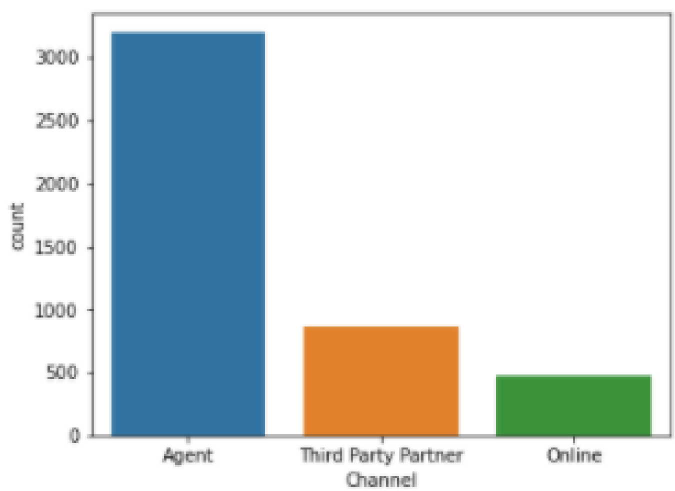

```
Male      0.59469
Female    0.40531
Name: Gender, dtype: float64
```

Existing Policy Type: Below Count Plot shows the existing policy types



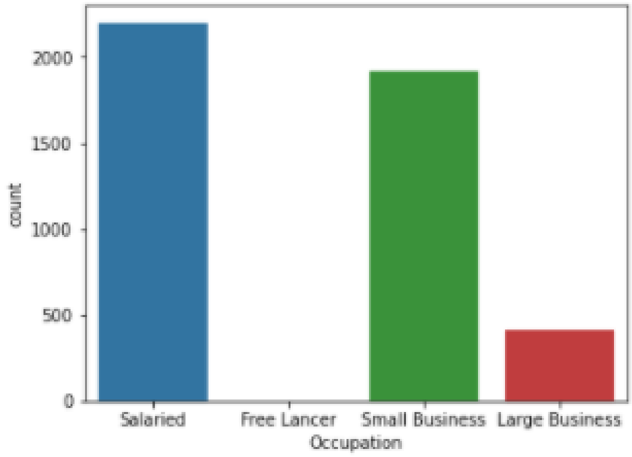

```
4    0.423894
3    0.302876
5    0.156637
2    0.048894
1    0.040487
6    0.027212
Name: ExistingProdType, dtype: float64
```

Blow graph shows the channel data



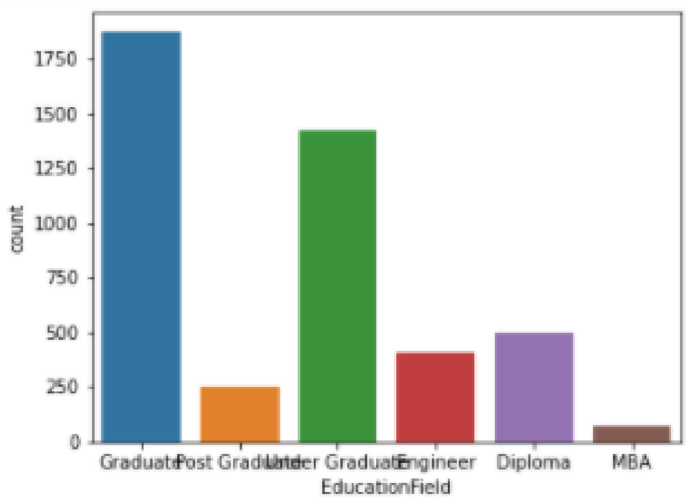

```
Agent                  0.706637
Third Party Partner    0.189823
Online                 0.103540
Name: Channel, dtype: float64
```

Below graph shows the occupation



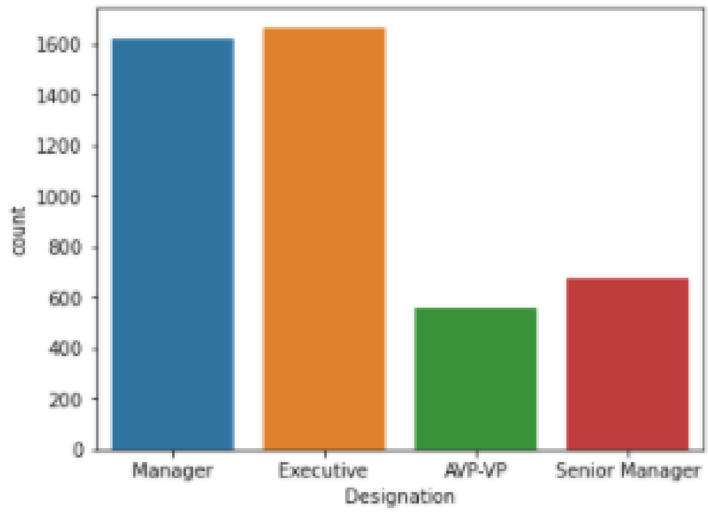

```
Salaried          0.484956
Small Business    0.424336
Large Business    0.090265
Free Lancer       0.000442
Name: Occupation, dtype: float64
```

Below graph shows the education field



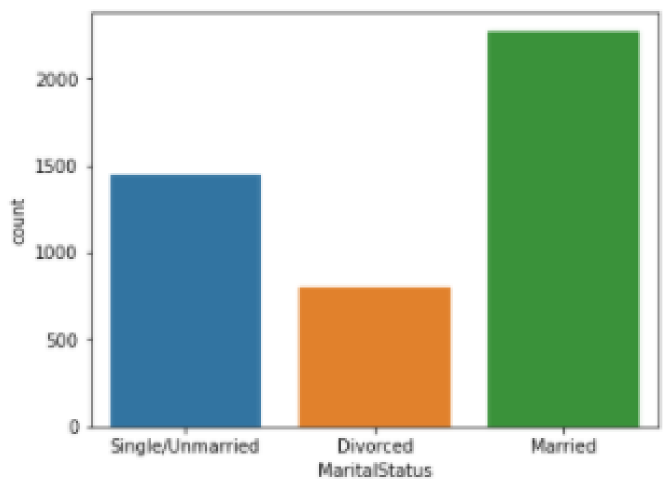

```
Graduate          0.413717
Under Graduate    0.314159
Diploma           0.109735
Engineer          0.090265
Post Graduate     0.055752
MBA               0.016372
Name: EducationField, dtype: float64
```

Below graph shows the designation



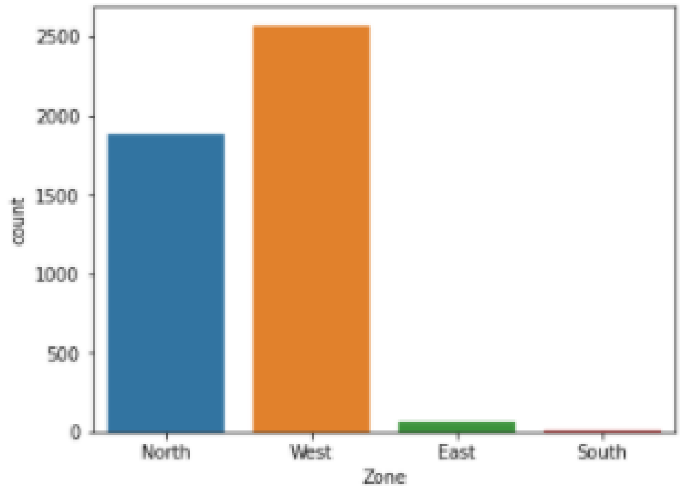

```
Executive         0.367699
Manager           0.358407
Senior Manager    0.149558
AVP-VP            0.124336
Name: Designation, dtype: float64
```

Below graph shows the marital status



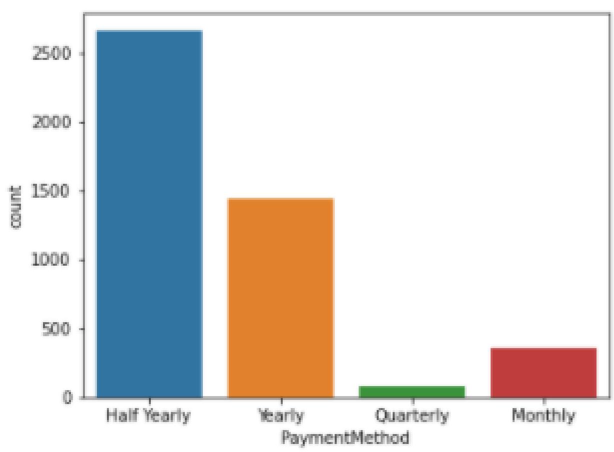

```
Married             0.501770
Single/Unmarried    0.320354
Divorced            0.177876
Name: MaritalStatus, dtype: float64
```

Below graph shows the various Zones




```
West     0.567699
North    0.416814
East     0.014159
South    0.001327
Name: Zone, dtype: float64
```

Below graph shows various payment methods




```
Half Yearly    0.587611
Yearly         0.317257
Monthly        0.078319
Quarterly      0.016814
Name: PaymentMethod, dtype: float64
```

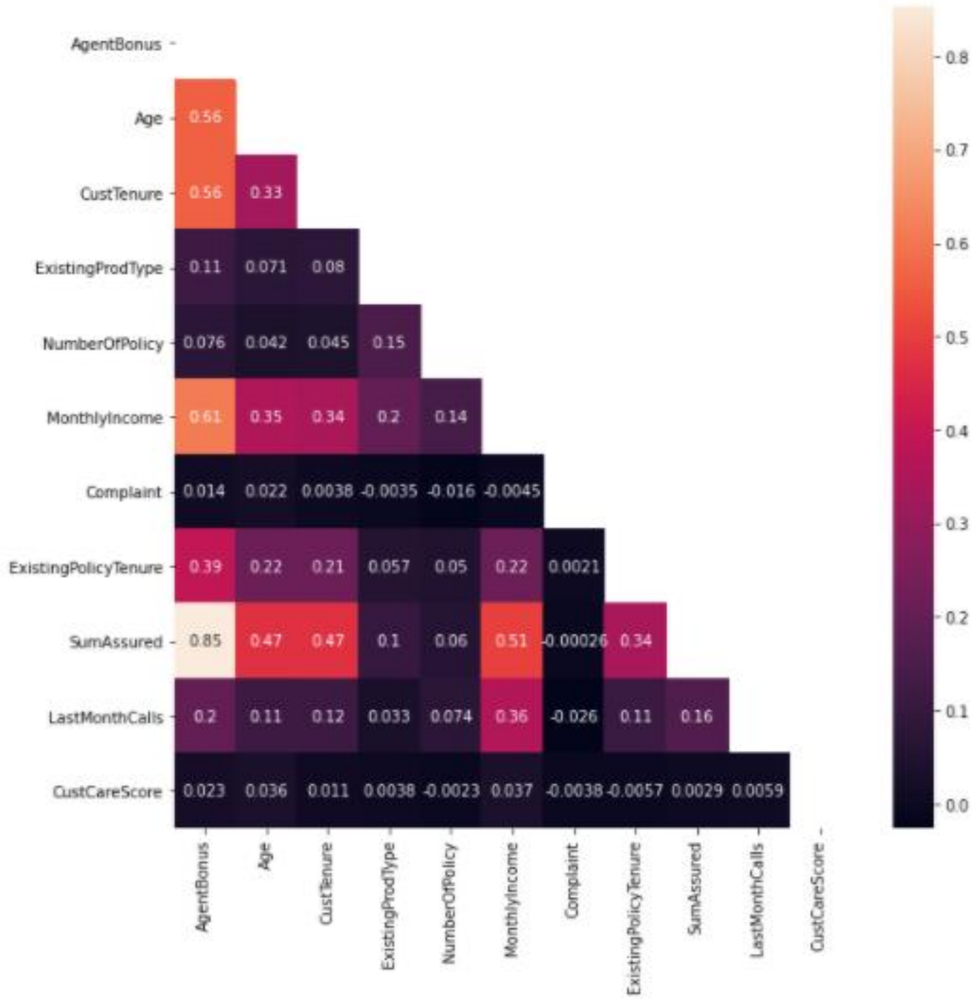
PROBLEM 3.B

Bivariate analysis (relationship between different variables, correlations)

Resolution:

Below Pair plot shows the pairwise relationships in a dataset

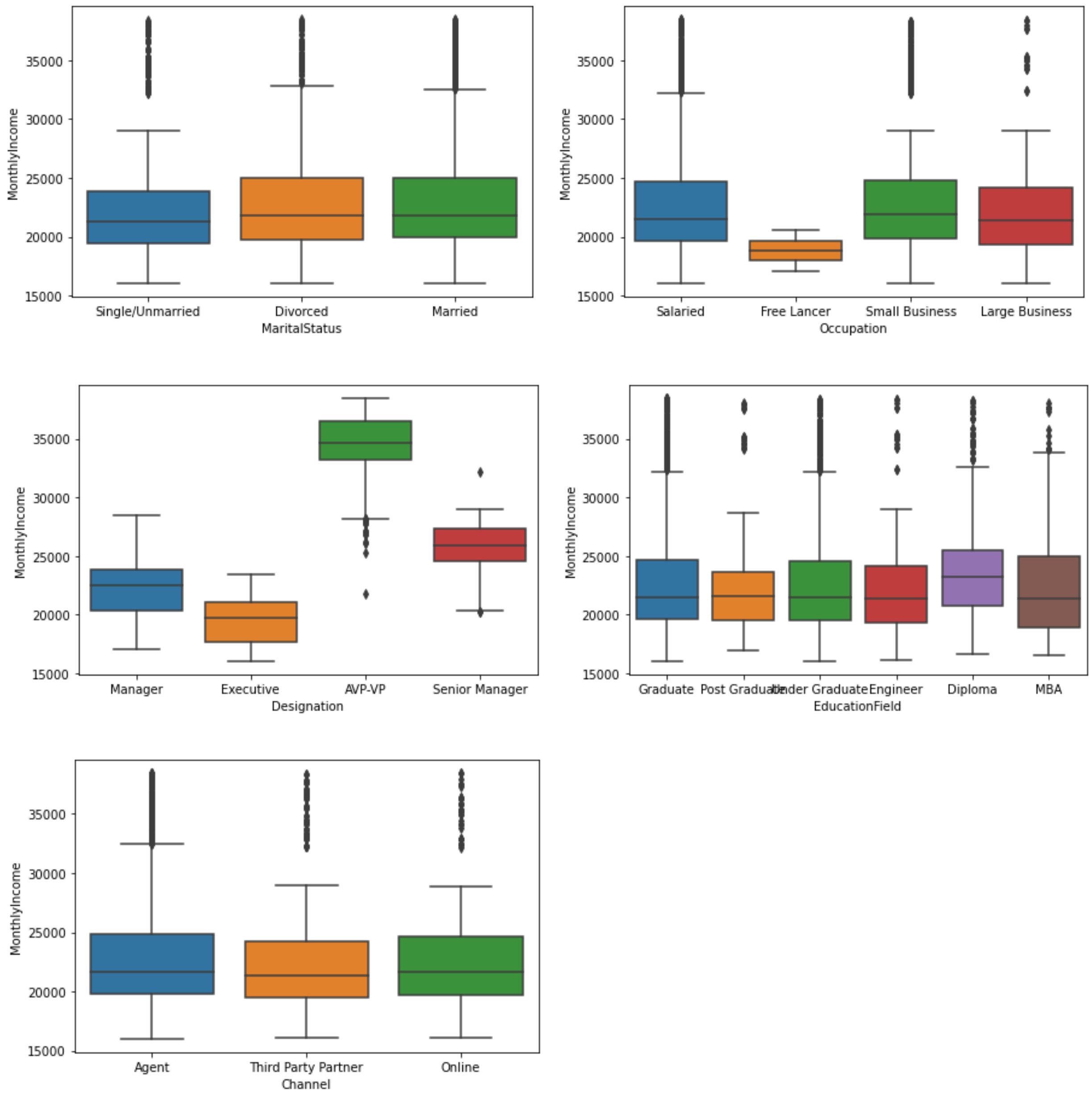
Below Correlation / Heat Map shows Strong Correlation



Top 5 strong correlations:

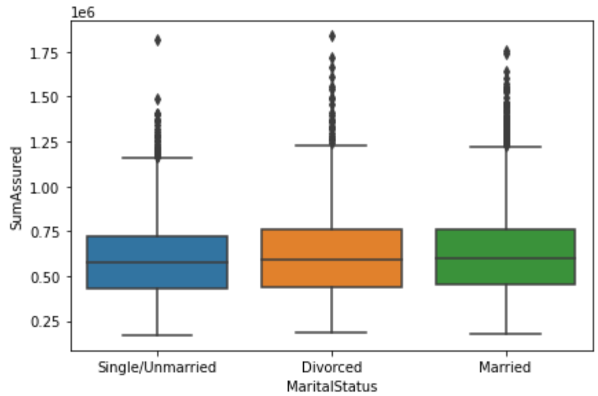
- SumAssured & AgentBonus
- MonthlyIncome & AgentBonus
- CustTenure & AgentBonus
- Age & AgentBonus
- MonthlyIncome & SumAssured

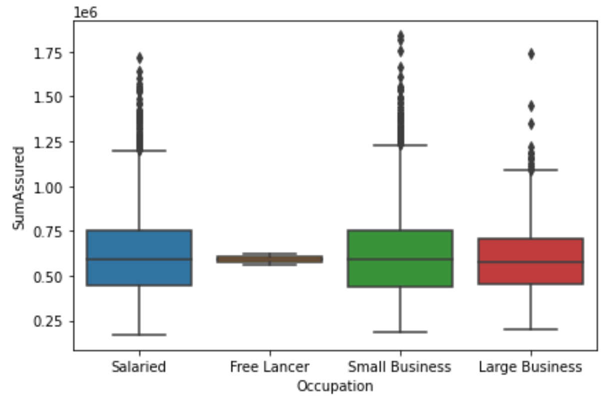
Below box plots shows relationship between MonthlyIncome & categorical variables

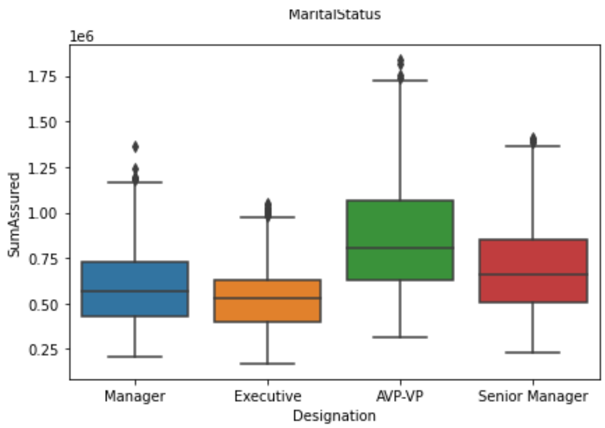


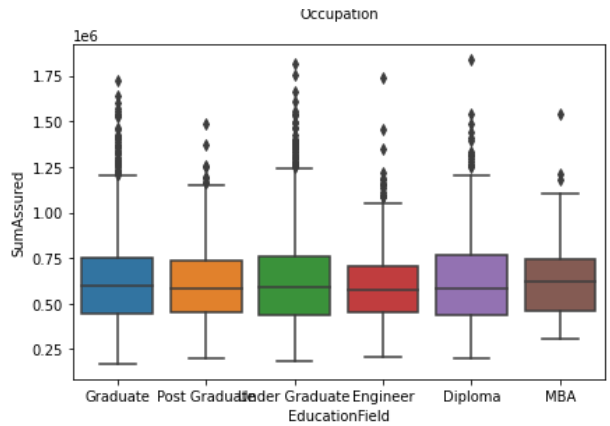
Customer Designation creates clear groups for MonthlyIncome of the customer so Missing Values in MonthlyIncome will be filled considering means of every group

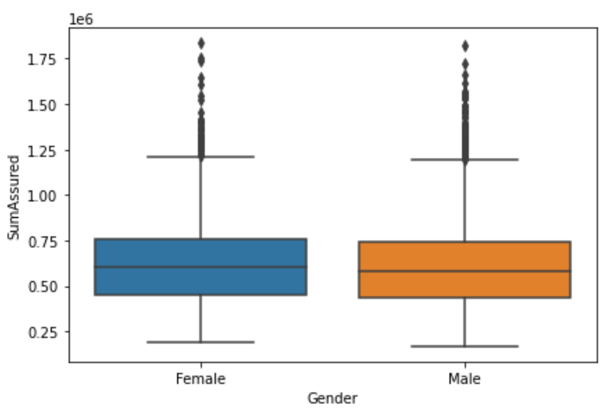
Below box plots shows relation between SumAssured & categorical variables

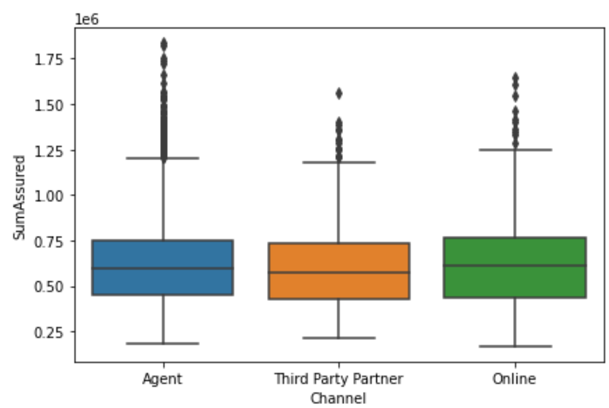




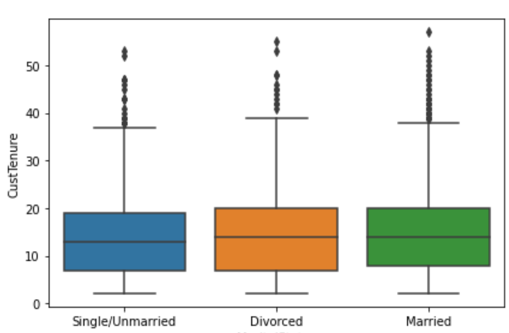


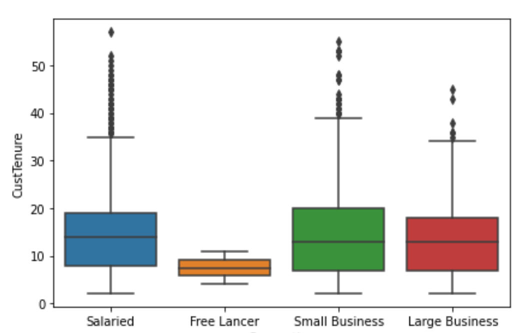


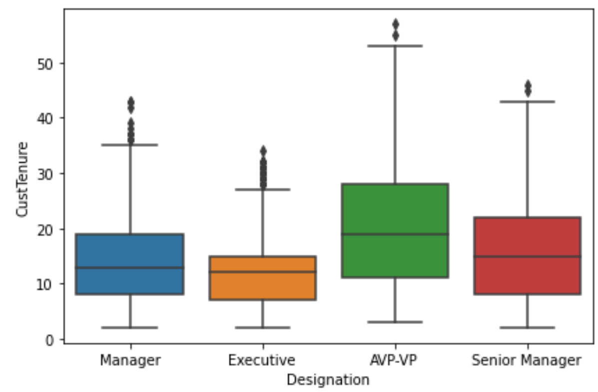


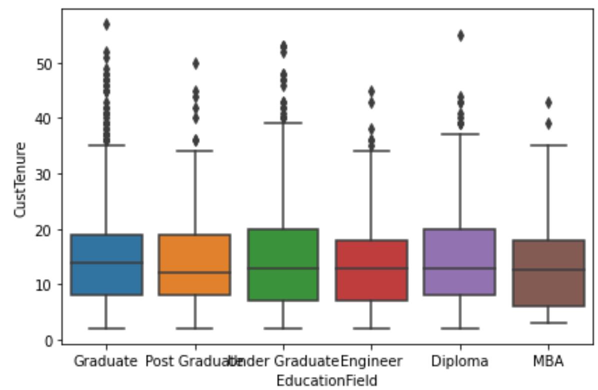


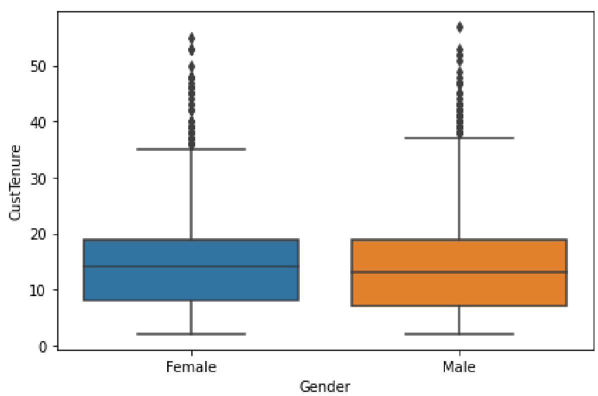
Below box plots shows the relationship between CustTenure & categorical variables

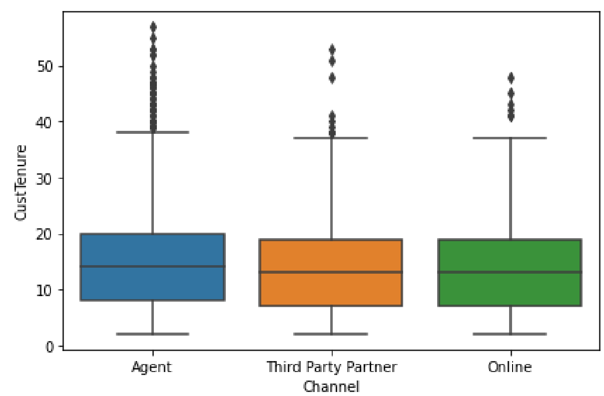




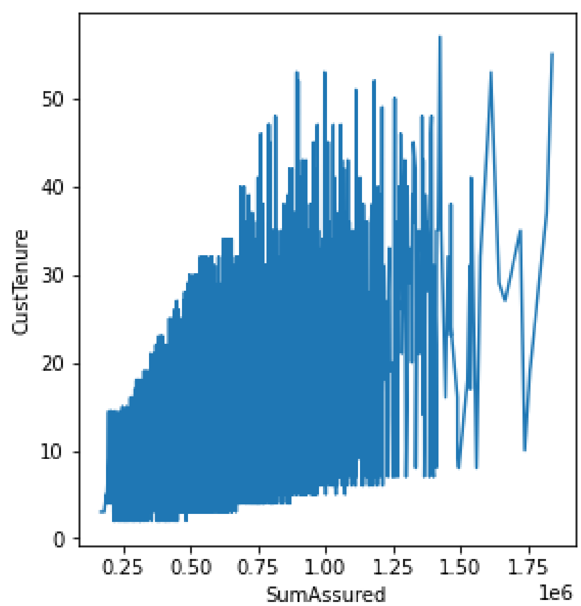


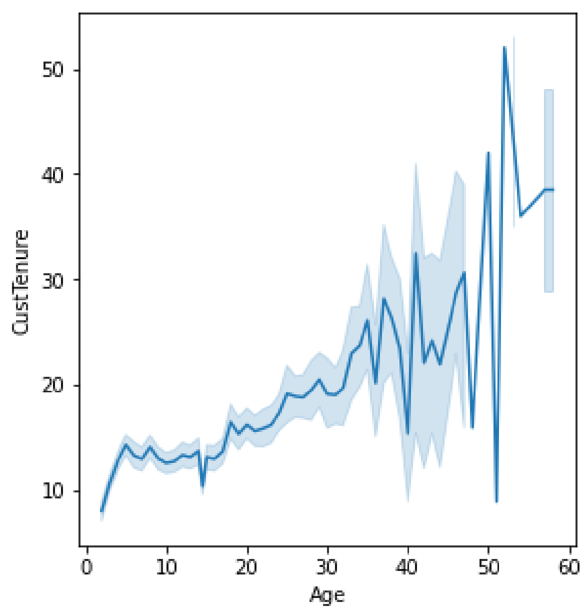




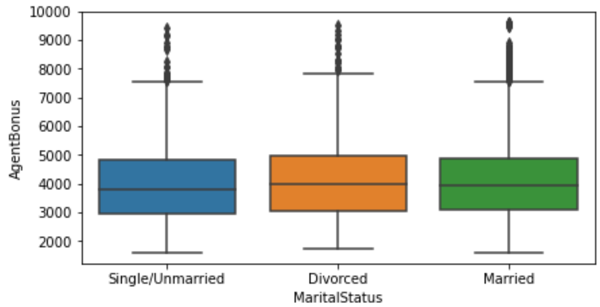


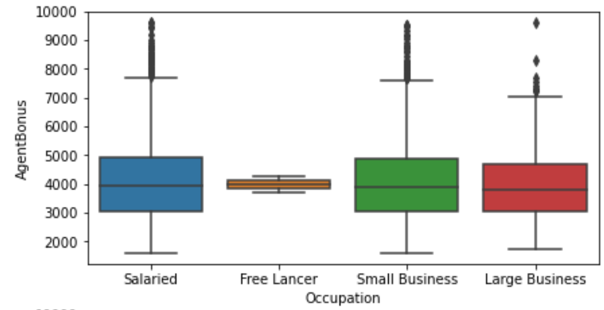
Below graphs shows the relationship between Customer tenure VS Sum Assured and age

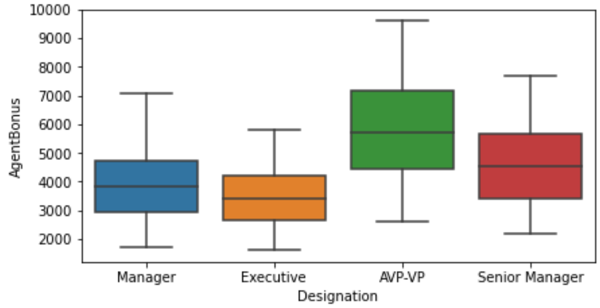


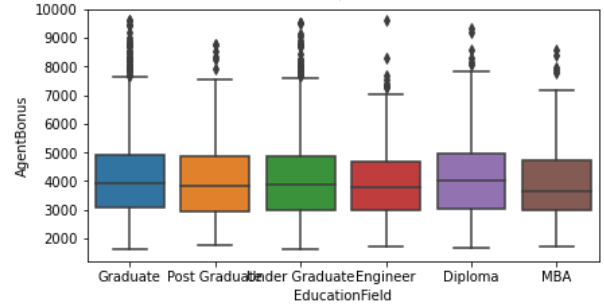


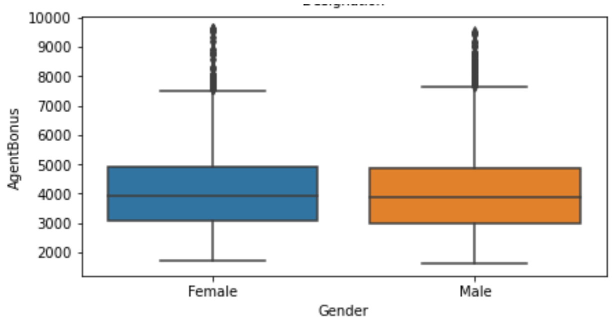
Below Box plots shows relationship between AgentBonus & categorical variables

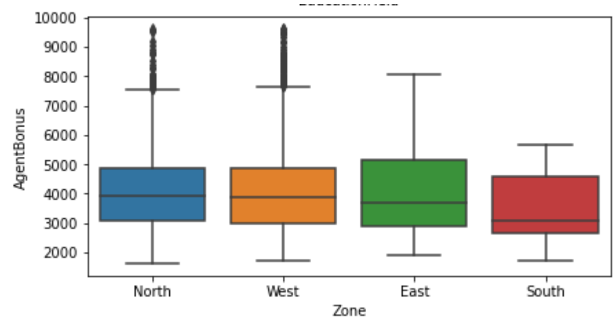


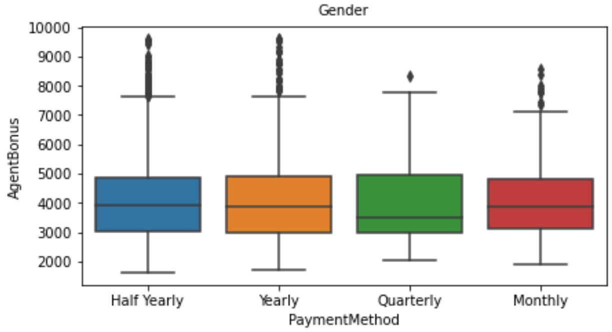


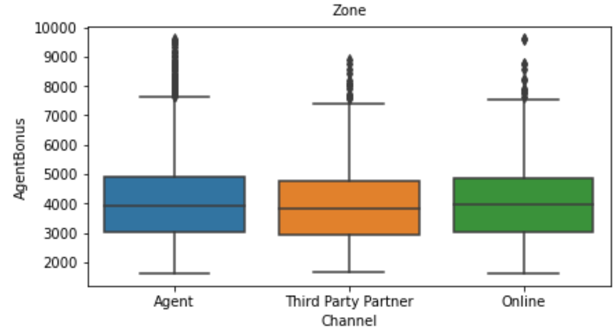


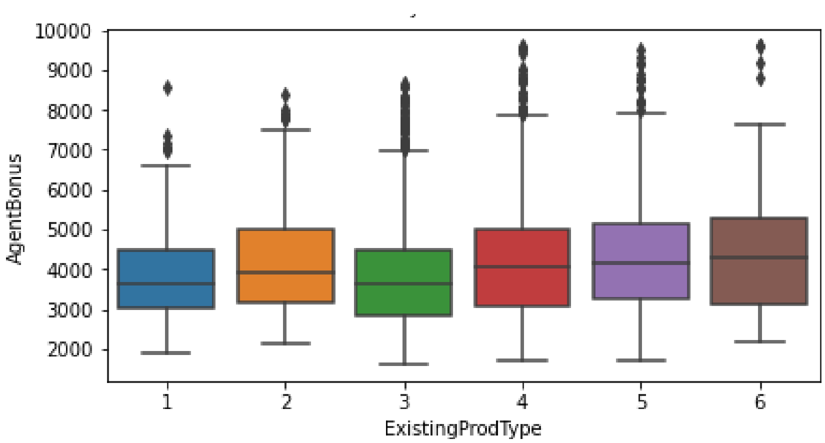












The dependent variable AgentBonus show some variation with Designation from above Boxplots, but doesn't seem to show any relationship otherwise with other categorical variable. We will test it further with ANOVA

PROBLEM 3.C

Removal of unwanted variables (if applicable)

Resolution:

We have no removed any unwanted variables as there were not many, there was only customer ID and it was not making any difference so did not remove any.

PROBLEM 3.D

Missing Value treatment (if applicable)

Resolution:

First we check for missing values : We can see that there are lot of missing values in multiple columns

```
Out[9]: CustID                   0
        AgentBonus               0
        Age                    269
        CustTenure             226
        Channel                  0
        Occupation               0
        EducationField           0
        Gender                   0
        ExistingProdType         0
        Designation              0
        NumberOfPolicy          45
        MaritalStatus            0
        MonthlyIncome          236
        Complaint                0
        ExistingPolicyTenure   184
        SumAssured             154
        Zone                     0
        PaymentMethod            0
        LastMonthCalls           0
        CustCareScore           52
        dtype: int64
```

Then we are substituting Missing values for MonthlyIncome

```
Out[17]: Designation
         AVP-VP            34377.114416
         Executive         19509.678099
         Manager           22228.965432
         Senior Manager    25846.513274
         Name: MonthlyIncome, dtype: float64
```

```
In [20]: #Mean value imputation for missing values
         sales_df.ExistingPolicyTenure.fillna(sales_df.ExistingPolicyTenure.mean(), inplace=True)
         sales_df.SumAssured.fillna(sales_df.SumAssured.mean(), inplace=True)
         sales_df.Age.fillna(sales_df.Age.mean(), inplace=True)
         sales_df.CustTenure.fillna(sales_df.CustTenure.mean(), inplace=True)
```

```
In [21]: #Mode value imputation for missing values
         sales_df.NumberOfPolicy.fillna(4, inplace=True)
         sales_df.CustCareScore.fillna(3, inplace=True)
```

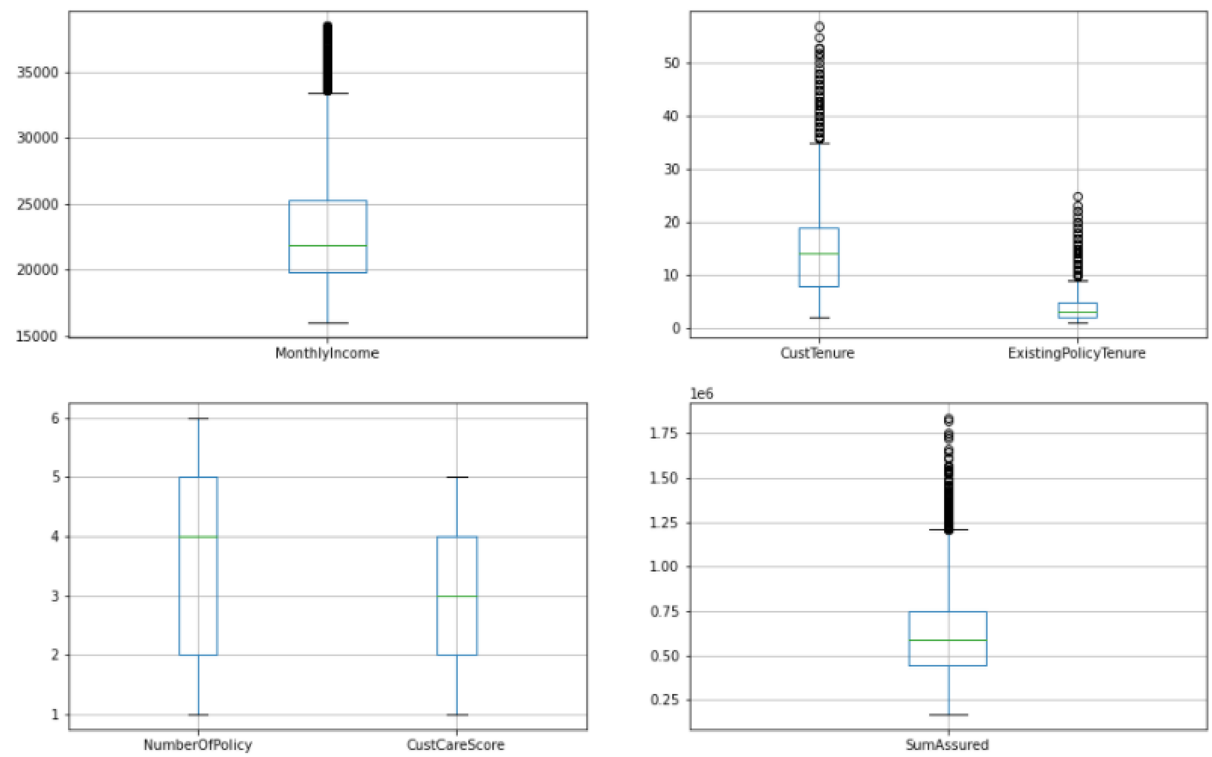
Once the imputation is done we can check for data and there are no missing values

```
Out[22]: CustID                  0
         AgentBonus              0
         Age                     0
         CustTenure              0
         Channel                 0
         Occupation              0
         EducationField          0
         Gender                  0
         ExistingProdType        0
         Designation             0
         NumberOfPolicy          0
         MaritalStatus           0
         MonthlyIncome           0
         Complaint               0
         ExistingPolicyTenure    0
         SumAssured              0
         Zone                    0
         PaymentMethod           0
         LastMonthCalls          0
         CustCareScore           0
         dtype: int64
```

PROBLEM 3.E
Outlier treatment (if required)

Resolution:

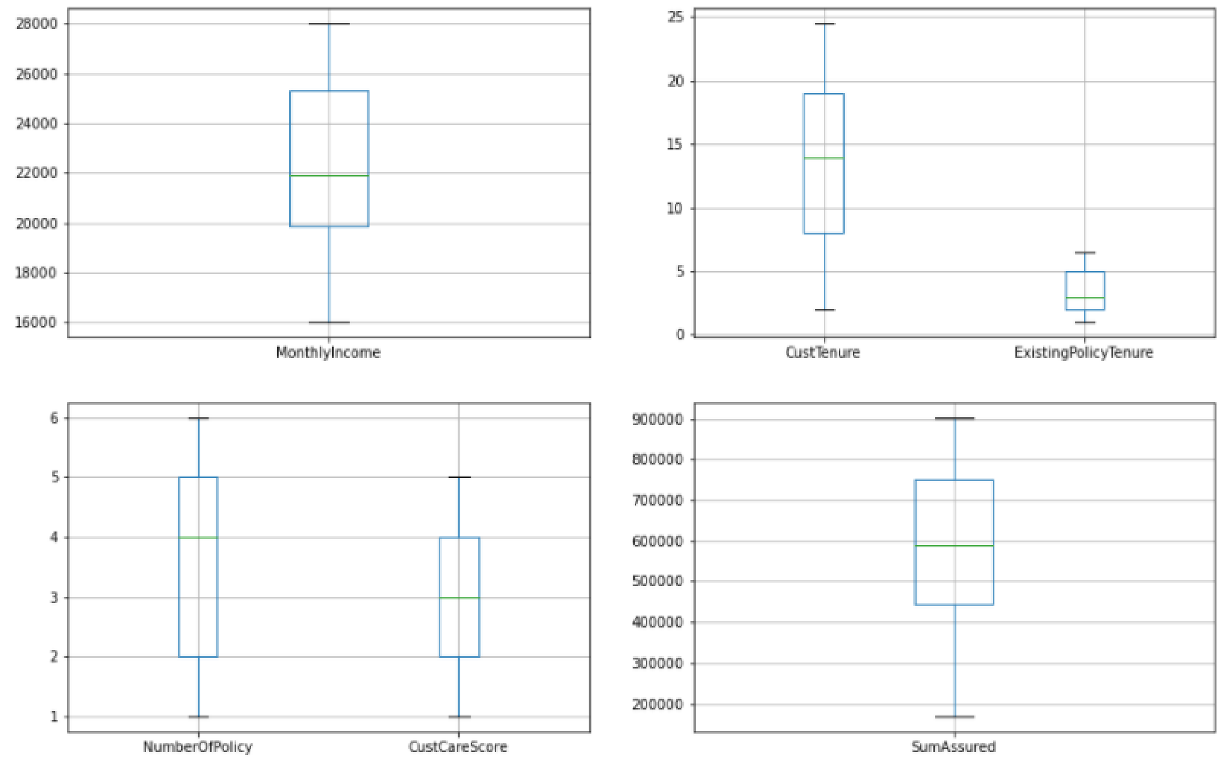
First we check for the outliers and box plots shows the outliers



Based on the above graph we remove the below outliers

```
In [29]: remove_outliers('MonthlyIncome')
         remove_outliers('CustTenure')
         remove_outliers('ExistingPolicyTenure')
         remove_outliers('SumAssured')
```

After removing the outliers, we can see from the below box plots that there are no outliers



PROBLEM 3.F

Variable transformation (if applicable)

Resolution:

Yes, there were few variables which were transformation below are the transformed variables.

Categories created for age:

```
Out[38]: ['21-39', '1-20', '40-60']
         Categories (3, object): ['1-20' < '21-39' < '40-60']
```

Encoding categorical variables

```
In [39]: #encoding categorical variables
         encoded_df = sales_df.copy()

         encoded_df['Age'] = pd.Categorical(encoded_df['Age']).codes
         encoded_df['Age'].unique()

Out[39]: array([1, 0, 2], dtype=int8)
```

```python
In [40]: #encoding categorical variables
         for col in encoded_df:
             if encoded_df[col].dtype == 'object':
                 encoded_df[col] = pd.Categorical(encoded_df[col]).codes
                 print(col, ": ", encoded_df[col].unique())

Channel :  [0 2 1]
Occupation :  [2 0 3 1]
EducationField :  [2 4 5 1 0 3]
Gender :  [0 1]
Designation :  [2 1 0 3]
MaritalStatus :  [2 0 1]
Zone :  [1 3 0 2]
PaymentMethod :  [0 3 2 1]
```

PROBLEM 3.G

Addition of new variables (if required)

Resolution:

There were 2 new variables added in the data set

Out[50]:

Type	Designation	...	MonthlyIncome	Complaint	ExistingPolicyTenure	SumAssured	Zone	PaymentMethod	LastMonthCalls	CustCareScore	SilWidth	Cluster
3	Manager	...	20993.0	1	2.0	806761.000000	North	Half Yearly	5	2.0	0.073427	1
4	Manager	...	20130.0	0	3.0	294502.000000	North	Yearly	7	3.0	0.040148	2
4	Executive	...	17090.0	1	2.0	619999.699267	North	Yearly	0	3.0	0.050278	2
3	Executive	...	17909.0	1	2.0	268635.000000	West	Half Yearly	0	5.0	0.111666	1
3	Executive	...	18468.0	0	4.0	366405.000000	West	Half Yearly	2	5.0	0.137513	1

We will consider 3 ultimate clusters as that is giving us very fewer negative silhouette widths than 4 clusters

Note: Positive silhouette width suggests that the observation belong to the correct cluster, negative would be opposite.

Problem 4: Business insights from EDA

PROBLEM 4.A

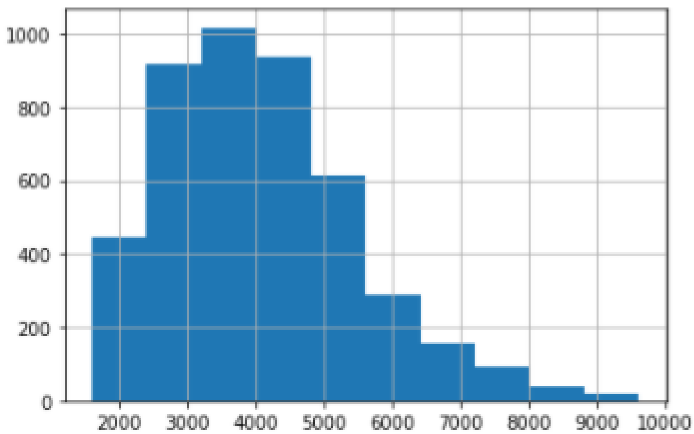
Is the data unbalanced? If so, what can be done? Please explain in the context of the business

Resolution:

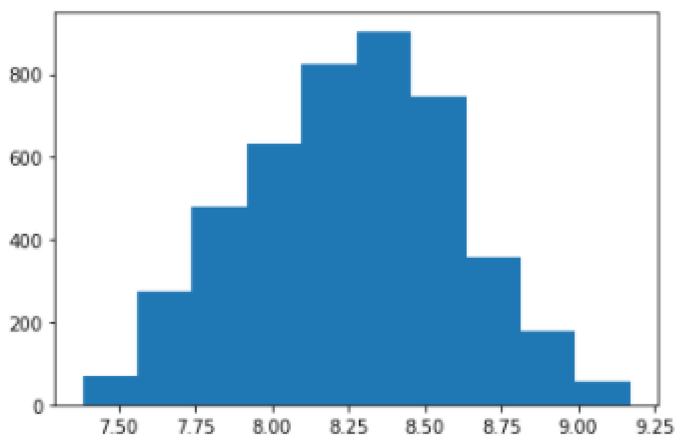
To check whether the data is unbalanced, we checked using Shapiro-wilk to test the normality of the continues variables

H0- Data is normal H1 - Data is not normal

```
In [31]: t, p = stats.shapiro(sales_df['AgentBonus'])
         t1, p1 = stats.shapiro(stats.zscore(sales_df['AgentBonus']))
         print(t," ", p)
         print(t1," ", p1)

         0.9570314884185791   1.4187508533160892e-34
         0.9570330381393433   1.4203979229396827e-34
```




```
(array([ 69., 276., 480., 634., 824., 904., 745., 355., 177.,  56.]),
 array([7.38087904, 7.55982627, 7.7387735 , 7.91772073, 8.09666797,
        8.2756152 , 8.45456243, 8.63350967, 8.8124569 , 8.99140413,
        9.17035136]),
 <BarContainer object of 10 artists>)
```




```
In [34]: t, p = stats.shapiro(sales_df_ANNOVA['AgentBonus'])
         print(t ,", ", p)

         0.9950266480445862 ,  2.562266371297639e-11
```

The dependent variable was tried to convert into a normal distribution, however the results were still unsuccessful. It will be assumed Normal for further ANNOVA test

ANNOVA

In [35]:
```
formula = 'AgentBonus ~ C(Channel)+ C(Occupation) + C(LastMonthCalls) +C(Complaint) + C(ExistingProdType) + C(MaritalStatus) + C(
model = ols(formula, sales_df_ANNOVA).fit()
aov_table = anova_lm(model)
print(aov_table)
```

```
                        df     sum_sq    mean_sq           F         PR(>F)
C(Channel)             2.0   0.747771   0.373886    4.462671   1.158297e-02
C(Occupation)          3.0   0.482862   0.160954    1.921135   1.238708e-01
C(LastMonthCalls)     18.0  32.956076   1.830893   21.853413   4.098847e-69
C(Complaint)           1.0   0.148066   0.148066    1.767308   1.837835e-01
C(ExistingProdType)    5.0   4.363999   0.872800   10.417678   5.916756e-10
C(MaritalStatus)       2.0   1.039424   0.519712    6.203244   2.040326e-03
C(EducationField)      5.0   0.122326   0.024465    0.292015   9.176025e-01
C(NumberOfPolicy)      5.0   1.638901   0.327780    3.912362   1.533478e-03
C(Zone)                3.0   0.302482   0.100827    1.203467   3.068547e-01
C(CustCareScore)       4.0   0.603304   0.150826    1.800248   1.258464e-01
C(Gender)              1.0   0.390978   0.390978    4.666682   3.080631e-02
C(Designation)         3.0 106.997424  35.665808  425.704600  3.067919e-243
Residual            4467.0 374.248163   0.083781         NaN            NaN
```

In [36]:
```
model.summary()
```

Out[36]:

OLS Regression Results

Dep. Variable:	AgentBonus	**R-squared:**	0.286
Model:	OLS	**Adj. R-squared:**	0.278
Method:	Least Squares	**F-statistic:**	34.38
Date:	Sun, 26 Dec 2021	**Prob (F-statistic):**	2.89e-282
Time:	15:35:11	**Log-Likelihood:**	-783.16
No. Observations:	4520	**AIC:**	1672.
Df Residuals:	4467	**BIC:**	2012.
Df Model:	52		
Covariance Type:	nonrobust		

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

Out[37]:

OLS Regression Results

Dep. Variable:	AgentBonus	**R-squared:**	0.772
Model:	OLS	**Adj. R-squared:**	0.772
Method:	Least Squares	**F-statistic:**	3053.
Date:	Sun, 26 Dec 2021	**Prob (F-statistic):**	0.00
Time:	15:35:29	**Log-Likelihood:**	1795.1
No. Observations:	4520	**AIC:**	-3578.
Df Residuals:	4514	**BIC:**	-3540.
Df Model:	5		
Covariance Type:	nonrobust		

Notes:

[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.

[2] The condition number is large, 4.07e+06. This might indicate that there are

Strong multicollinearity or other numerical problems.

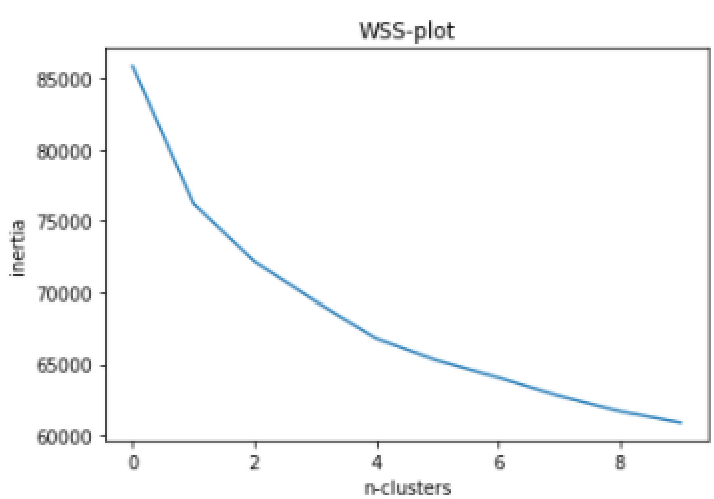
PROBLEM 4.B

Any business insights using clustering (if applicable)

Resolution:

Below are the findings after using the clustering

```
In [45]: wss

Out[45]: [85880.00000000023,
          76220.24458786246,
          72152.83670812742,
          69390.5008559819,
          66808.48464330098,
          65268.89090621684,
          64077.749512616705,
          62777.56207706078,
          61704.63072514283,
          60906.99922904601]
```



We can see the elbow at 2 places cluster 1 and cluster 4

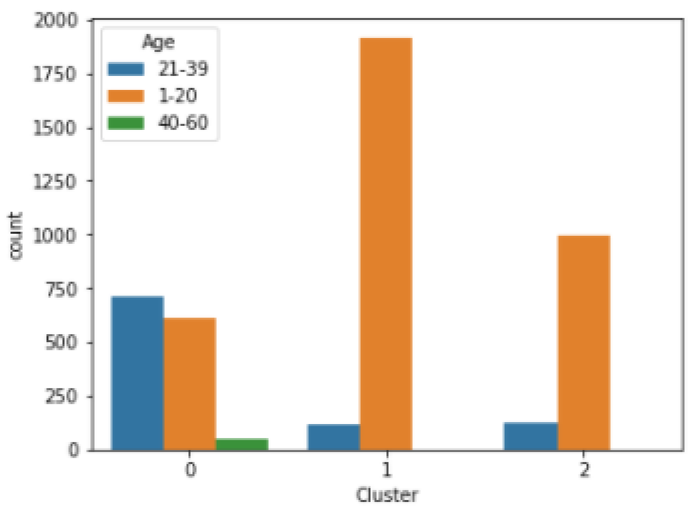
Below is the silhouette score

```
In [49]: from sklearn.metrics import silhouette_samples, silhouette_score

         silhouette_score(sales_df_scaled, kmean.labels_)

Out[49]: 0.08330856477373955
```

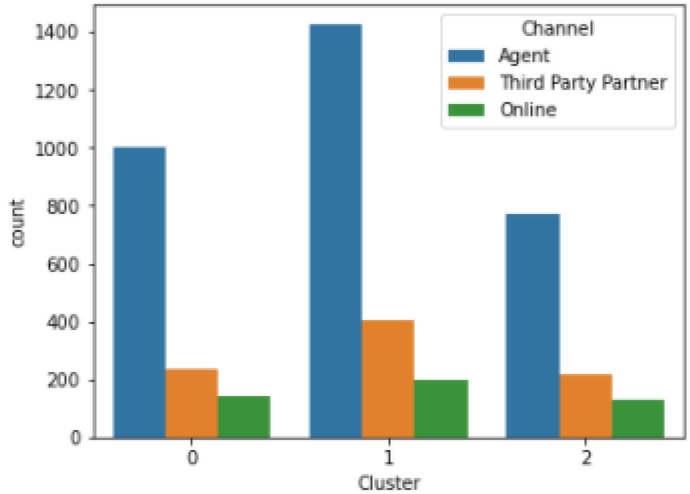
We will consider 3 ultimate clusters as that is giving us very fewer negative silhouette widths than 4 clusters

Note: Positive silhouette width suggests that the observation belong to the correct cluster, negative would be opposite.

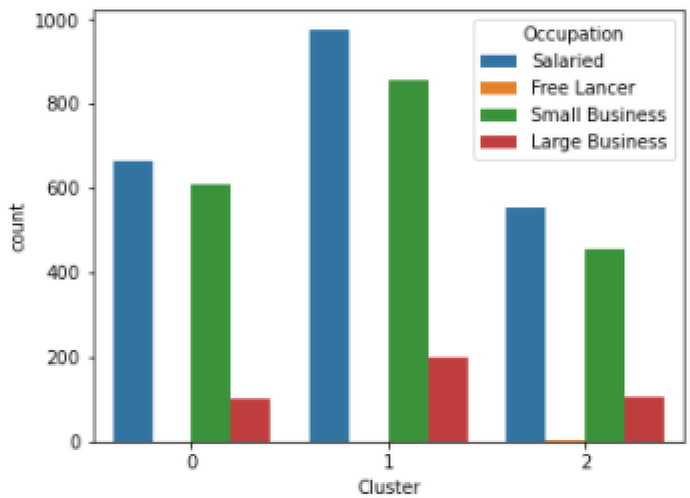
Below graphs shows the findings with variables VS Clusters



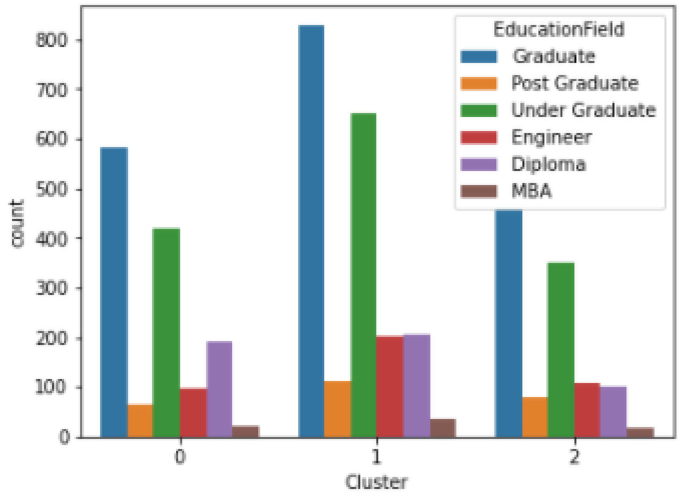

```
1-20     0.778097
21-39    0.210841
40-60    0.011062
Name: Age, dtype: float64
```



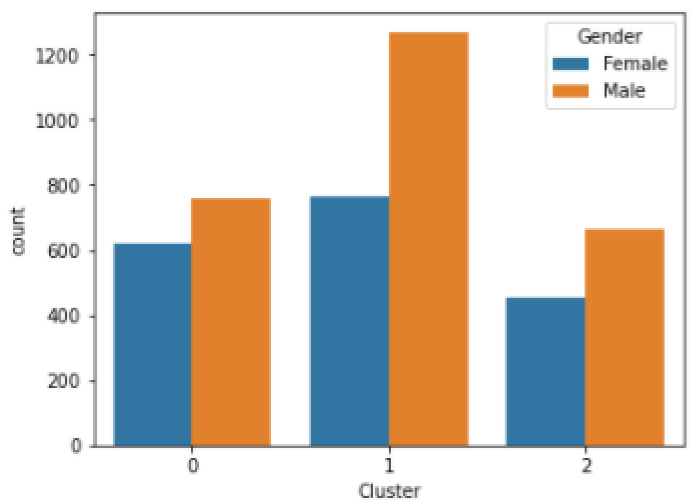

```
Agent                 0.706637
Third Party Partner   0.189823
Online                0.103540
Name: Channel, dtype: float64
```



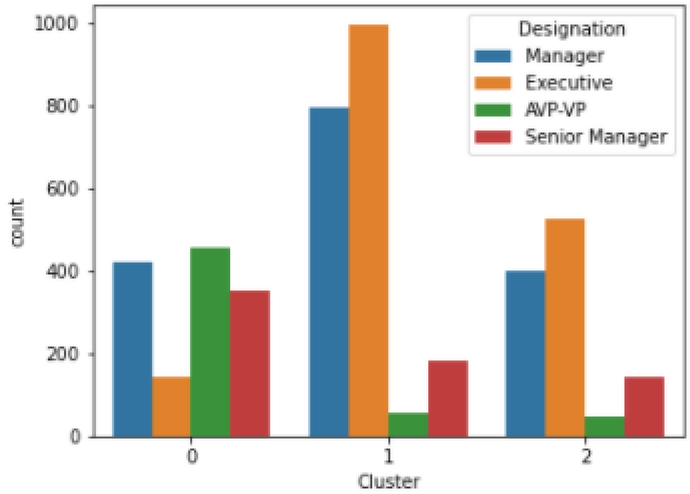

```
Salaried          0.484956
Small Business    0.424336
Large Business    0.090265
Free Lancer       0.000442
Name: Occupation, dtype: float64
```



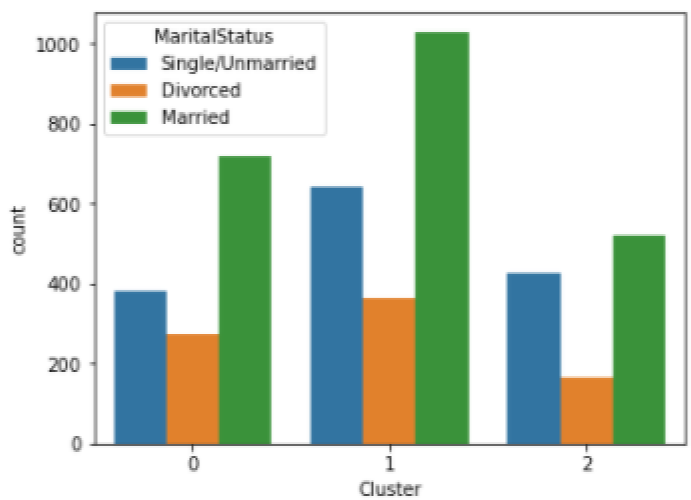

```
Graduate          0.413717
Under Graduate    0.314159
Diploma           0.109735
Engineer          0.090265
Post Graduate     0.055752
MBA               0.016372
Name: EducationField, dtype: float64
```



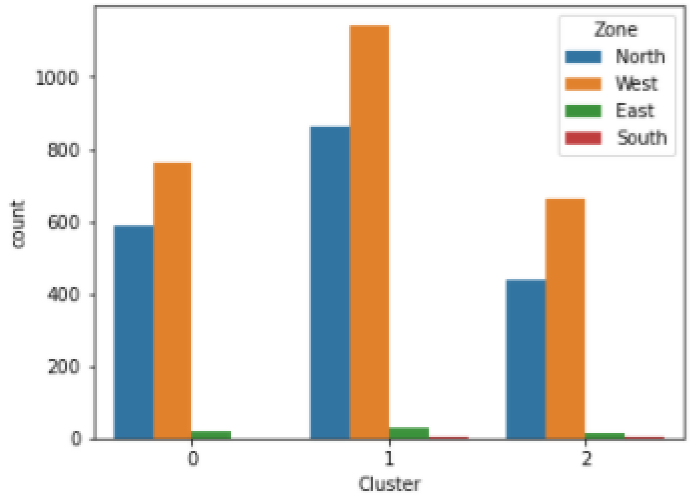

```
Male      0.59469
Female    0.40531
Name: Gender, dtype: float64
```



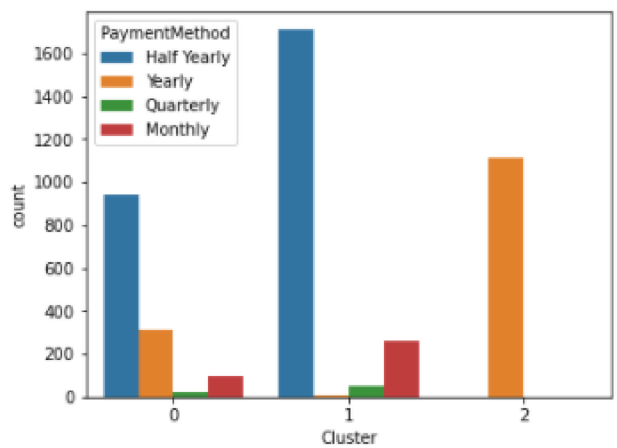

```
Executive         0.367699
Manager           0.358407
Senior Manager    0.149558
AVP-VP            0.124336
Name: Designation, dtype: float64
```




```
Married             0.501770
Single/Unmarried    0.320354
Divorced            0.177876
Name: MaritalStatus, dtype: float64
```




```
West     0.567699
North    0.416814
East     0.014159
South    0.001327
Name: Zone, dtype: float64
```




```
Half Yearly    0.587611
Yearly         0.317257
Monthly        0.078319
Quarterly      0.016814
Name: PaymentMethod, dtype: float64
```

PROBLEM 4.C

Any other business insights

Resolution:

We have performed Train and Test as well for 'sales_df_encoded' data and below are the additional insights.

In [83]:
```
display(x_train.shape)
display(y_train.shape)
display(x_test.shape)
display(y_test.shape)
```

```
(3390, 39)

(3390,)

(1130, 39)

(1130,)
```

```
'Dtree:'

{'max_depth': 10,
 'max_features': 36,
 'min_samples_leaf': 20,
 'min_samples_split': 60}

'RF:'

{'max_depth': 14,
 'max_features': 20,
 'min_samples_leaf': 20,
 'min_samples_split': 60,
 'n_estimators': 300}

'Grad Boost:'

{'learning_rate': 0.1, 'max_features': 20, 'n_estimators': 200}
```

In [86]:
```
display(grid_search_dt.score(x_train, y_train))
display(grid_search_dt.score(x_test, y_test))

display(grid_search_rf.score(x_train, y_train))
display(grid_search_rf.score(x_test, y_test))
```

```
0.8574452911593478

0.8021059978450223

0.8617561756225306

0.824304752131246
```

Out[87]:

	Train_Score	Test_Score	Train_RMSE	TEST_RMSE
Linear Regression	0.787786	0.775144	650.972831	651.019330
Ridge Regression	0.787585	0.774918	651.279691	651.346810
Lasso Regression	0.787036	0.775015	652.121090	651.206848
Elastic-Net	0.750951	0.730779	705.208277	712.354132
Decesion Tree	0.845111	0.798891	556.141562	615.684229
Random Forest	0.862355	0.824523	524.271003	575.110693
Bagging	0.861590	0.823214	525.725131	577.251782
Adaptive Boosting	0.771176	0.733724	675.968547	708.447844
Gradient Boosting	0.889957	0.833672	468.766033	559.917743
ANN	0.679062	0.666024	800.545024	793.413410
VotingRegressor	0.873732	0.830410	502.136244	565.381432

The End

Thakur Arun Singh

***************************^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^************************