Code Lipis Infotech Pvt. Ltd.

Historical Data Losses

Full Project Report

- DIPANSHU RANGA

-dipanshuranga@gmail.com

19

Historical Losses Dataset

Analysis Report

ABOUT:

Analysis Report on the past data of **losses** from an **insurance company** on **automobile** insurance policy, to **predict future losses** for same featured data as this dataset to decide **amount of premium of new customer's.**

1. DATA CONSOLIDATION:

Original dataset named with **"Historical_Losses_Data.csv".**

Dataset comprises of 15,290 observations and 9 variables

1st variable is **"Policy Number"** i.e., Not-**Significant** for making predictions so it's **dropped**.

Now, Dataset has **15,290 observations** and **8 variables.**

2. EXPLORATORY DATA ANALYSIS:

2.1. VARIABLE - IDENTIFICATION:

Out of 8 variables there is **one dependent** variable and other **seven** are **independent** variables.

Dependent Variable: --

1. Losses

Independent Variables: --

1. Age
2. Years_of_Driving_Experience
3. Number_of_Vehicles
4. Gender
5. Married
6. Vehicle_Age
7. Fuel_Type

*There are no NULL Values in dataset.

Data Type:	
Numeric:	**Character:**
1. Age	1. Gender
2. Years_of_Driving_Experience	2. Married
3. Number_of_Vehicles	3. Fuel_Type
4. Vehicle_Age	
5. Losses	

*All Numeric variables are integers.

Variable Category:	
Continuous:	**Categorical:**
1. Age	1. Gender
2. Years_of_Driving_Experience	2. Married
3. Number_of_Vehicles	3. Fuel_Type
4. Vehicle_Age	
5. Losses	

2.2. UNIVARIATE ANALYSIS:

Range Information:

Variable	Range/Category:
Age	16 – 70
Years_of_Driving_Experience	0 - 53
Number_of_Vehicles	1 - 4
Gender	'M' and 'F'
Married	'Married' and 'Single'
Vehicle_Age	0 – 15
Fuel_Type	'P' and 'D'
Losses	13 - 3500

*'M' and 'F' stands for 'Male' and 'Female respectively. **'P' and 'D' stands for 'Petrol' and 'Diesel' respectively.

Numerical-Data Variable Description:

	Age	Years_of_Driving_Experience	Number_of_Vehicles	Vehicle_Age	Losses
Count	15290	15290	15290	15290	15290
Mean	42.32	23.73	2.49	8.65	389.85
Std	18.28	17.85	0.95	4.34	253.72
Min	16.0	0.0	1.0	0.0	13.0
25%	24.0	6.0	2.0	6.0	226.0
50%	42.0	23.0	2.0	9.0	355.0
75%	61.0	42.0	3.0	12.0	489.0
Max	70.0	53.0	4.0	15.0	3500.0

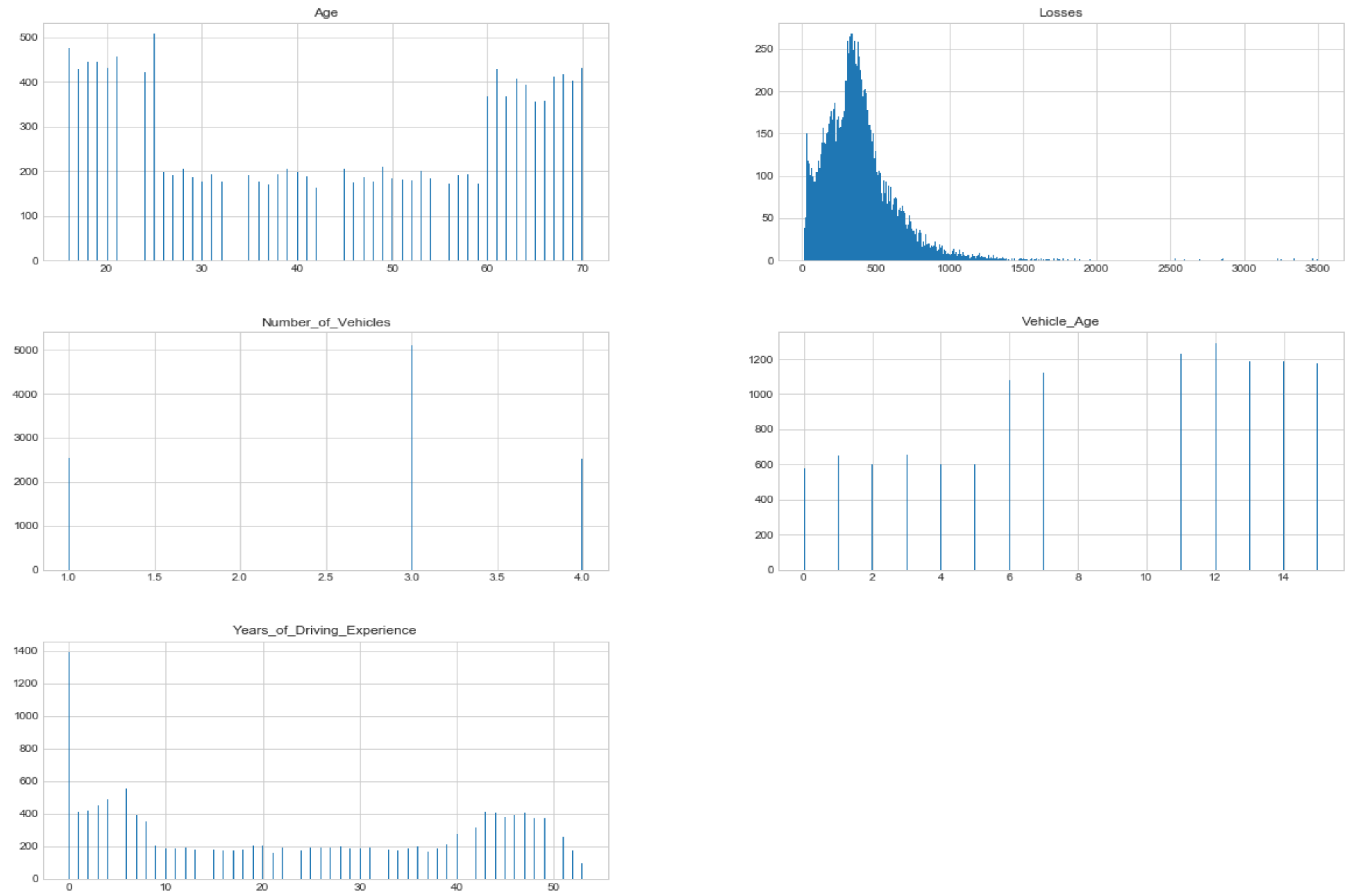
- There are no missing values in numerical columns.
- Maximum observation of "Losses" is 3500 and 75 percentile is 489 which tell us that **"Losses"** contain **outlier** values.

Skewness:

Variable	Skewness
Age	+0.05
Years_of_Driving_Experience	+0.097
Number_of_Vehicles	+0.0065
Vehicle_Age	-0.344
Losses	+2.55

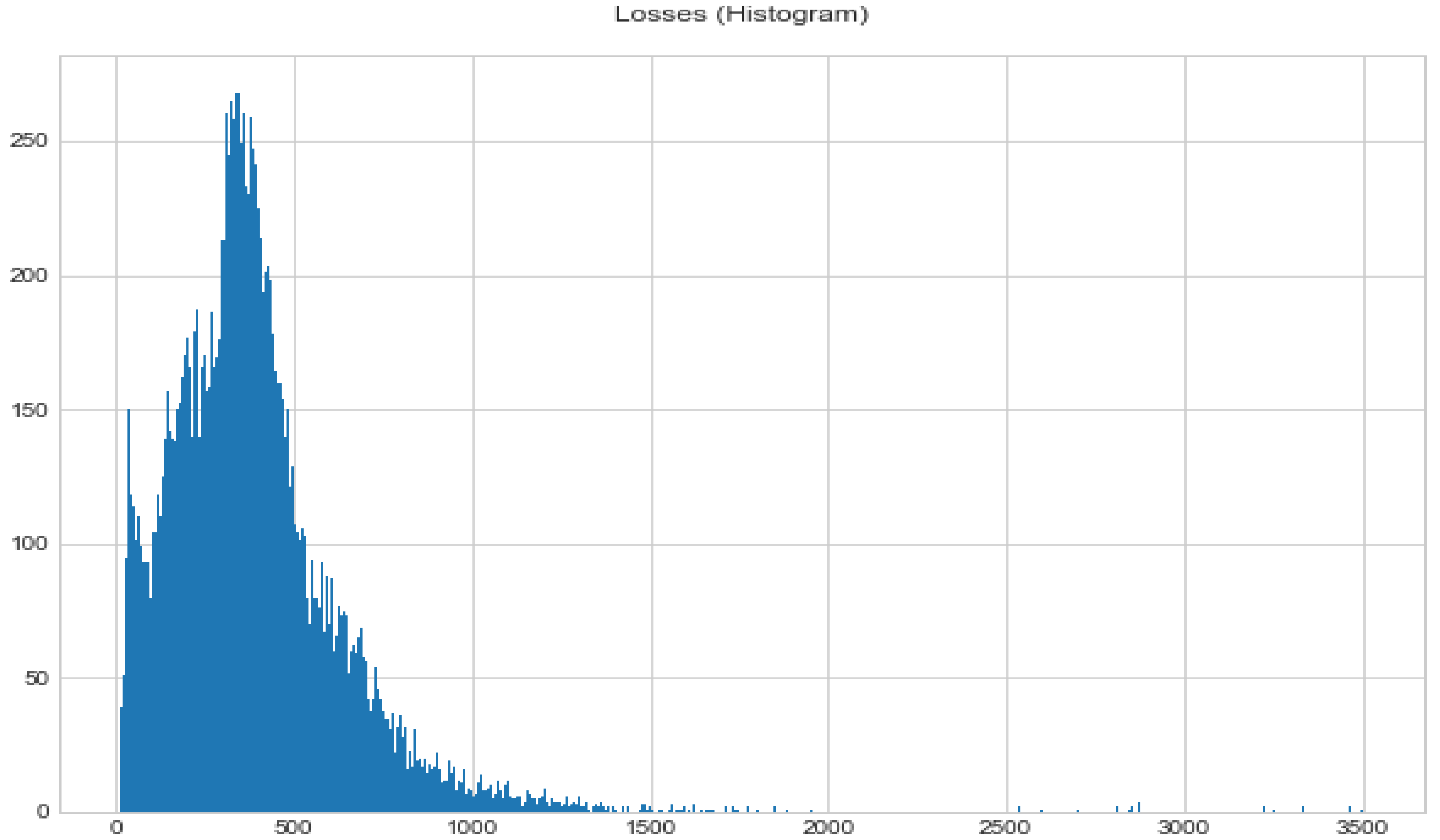
- "Losses" is **highly positive skewed** with a skewness of over +2.0
- "Vehicle_Age" is **negatively skewed** with skewness of over -0.3.

Histograms:



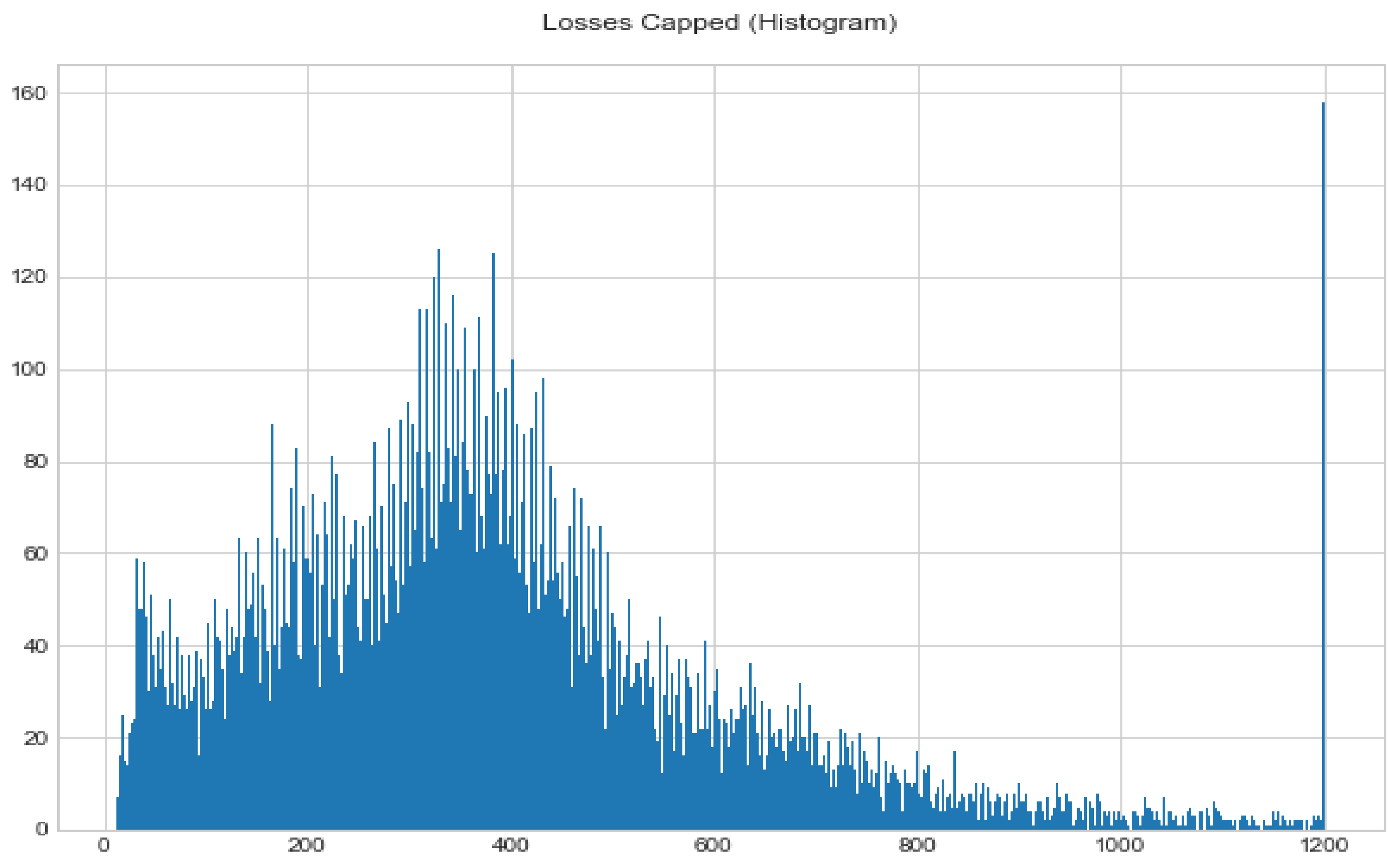
- Numbers of population in age group of **'30 to 60'** are **lower** in **number** in comparison to other age group.
- People with **zero** driving **experience** are **high** in **number**.

- Taking a separate look at Losses Histogram:
-



- Already observed high skewness of 2+ in Skewness-Table.
- There are people with **high losses**, but they are **low** in **number**.
- We can put all the **high losses** observations in **one category**.

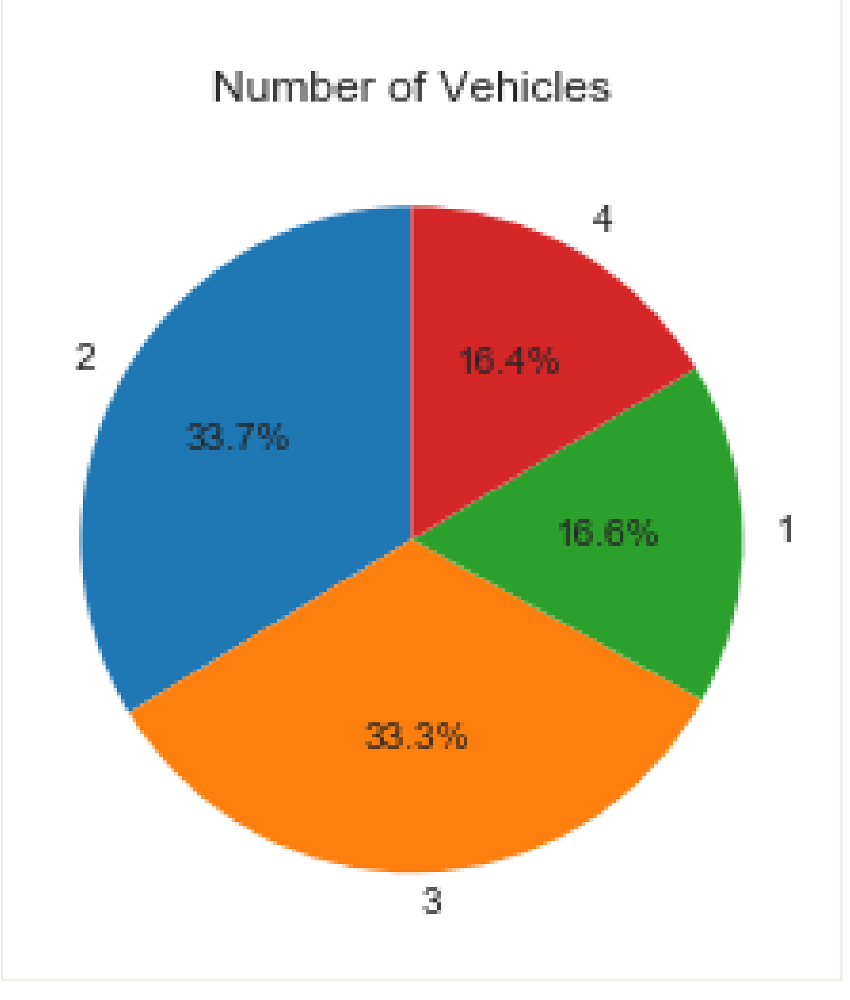
- After upper capping of “Losses” at 1200:



- The skewness is gradually decreased to 1.05 (before capping it was 2+).

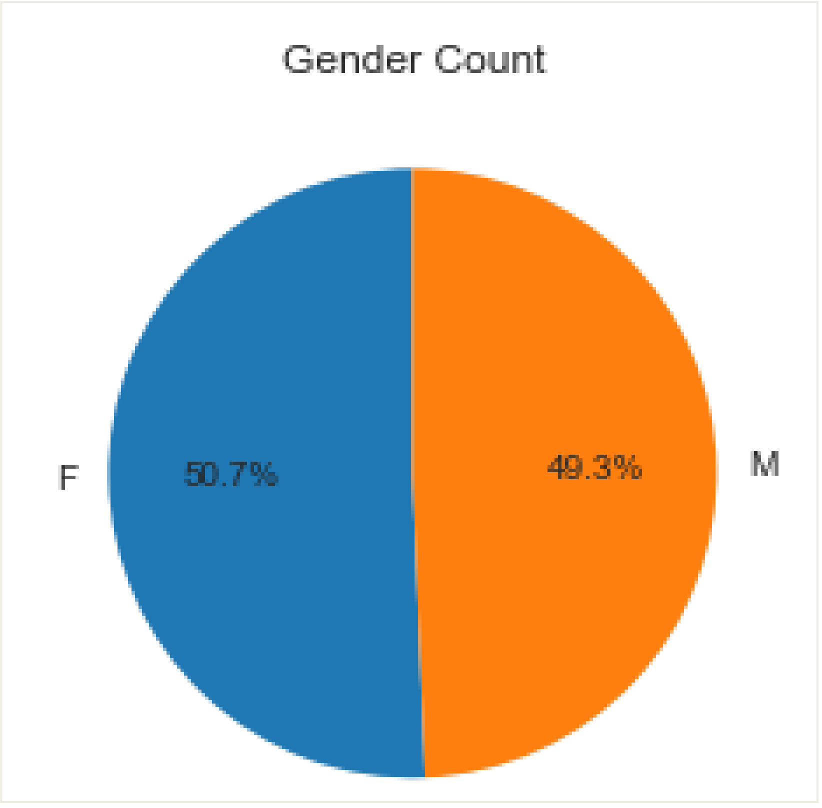
Pie-Plots:

- “Number_of_Vehicles”:

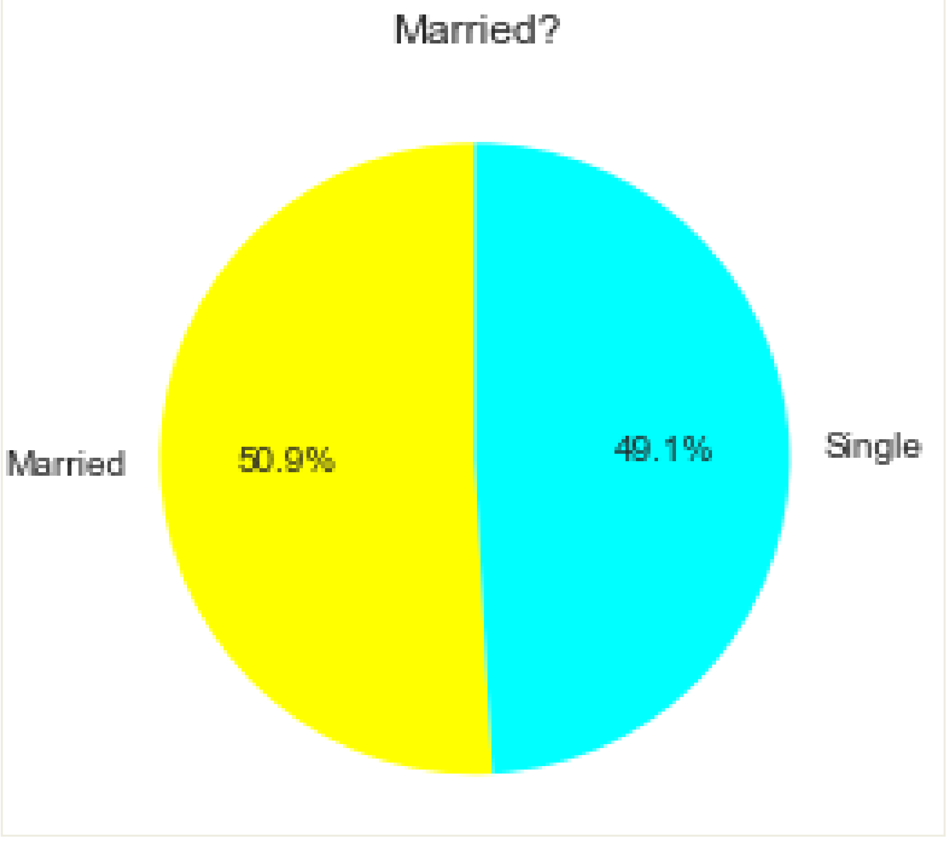


- People owning ‘two’ and ‘three’ vehicles are nearly **double** in number who owns ‘one’ or ‘four‘ vehicles.

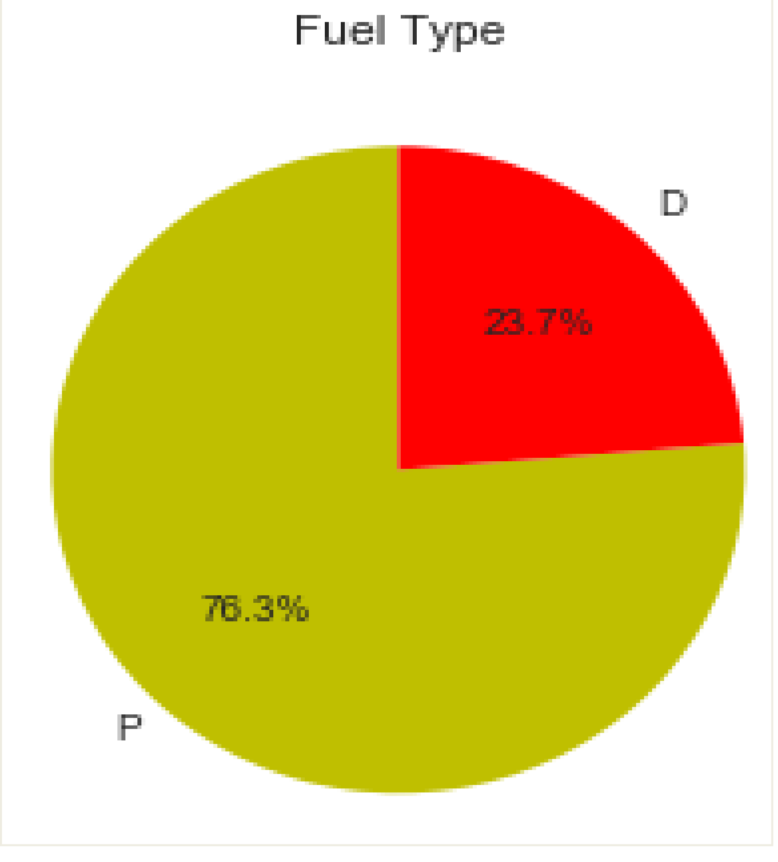
- **Categorical Data** (Gender, Married & Fuel_Type):



■ **Male** & **Female** have nearly equal share in observations.



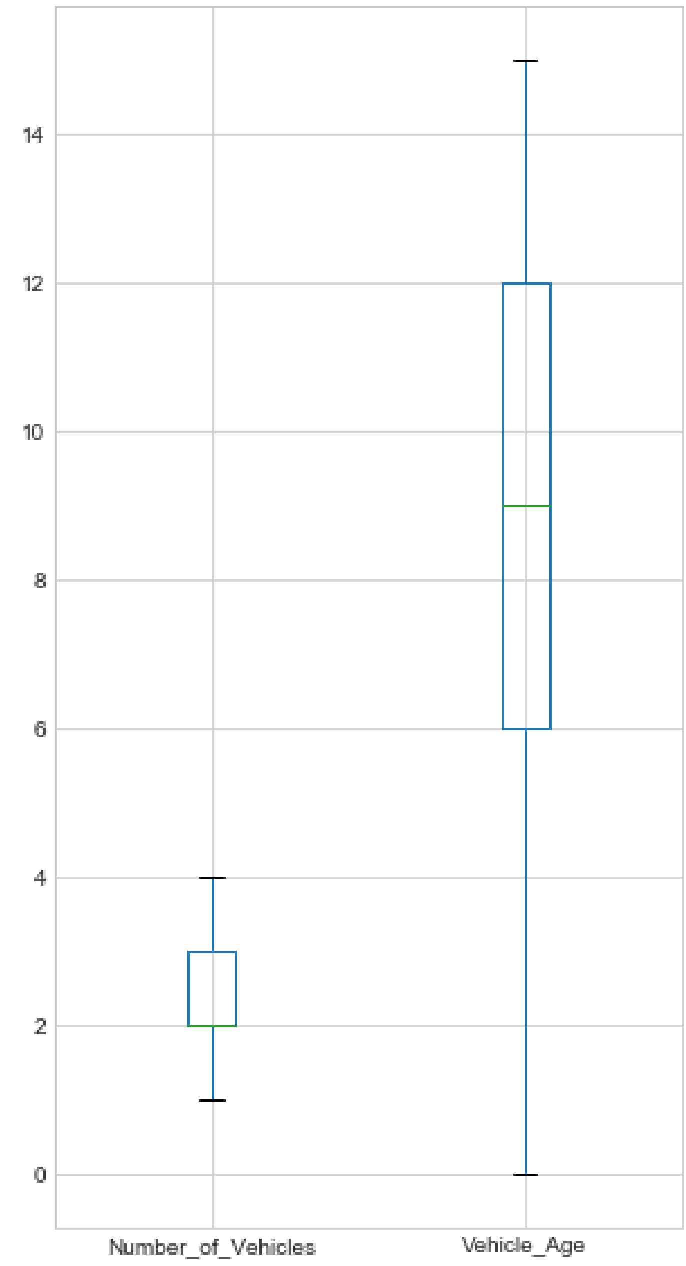
■ **Married** & **Single** population both have nearly equal number of observation.

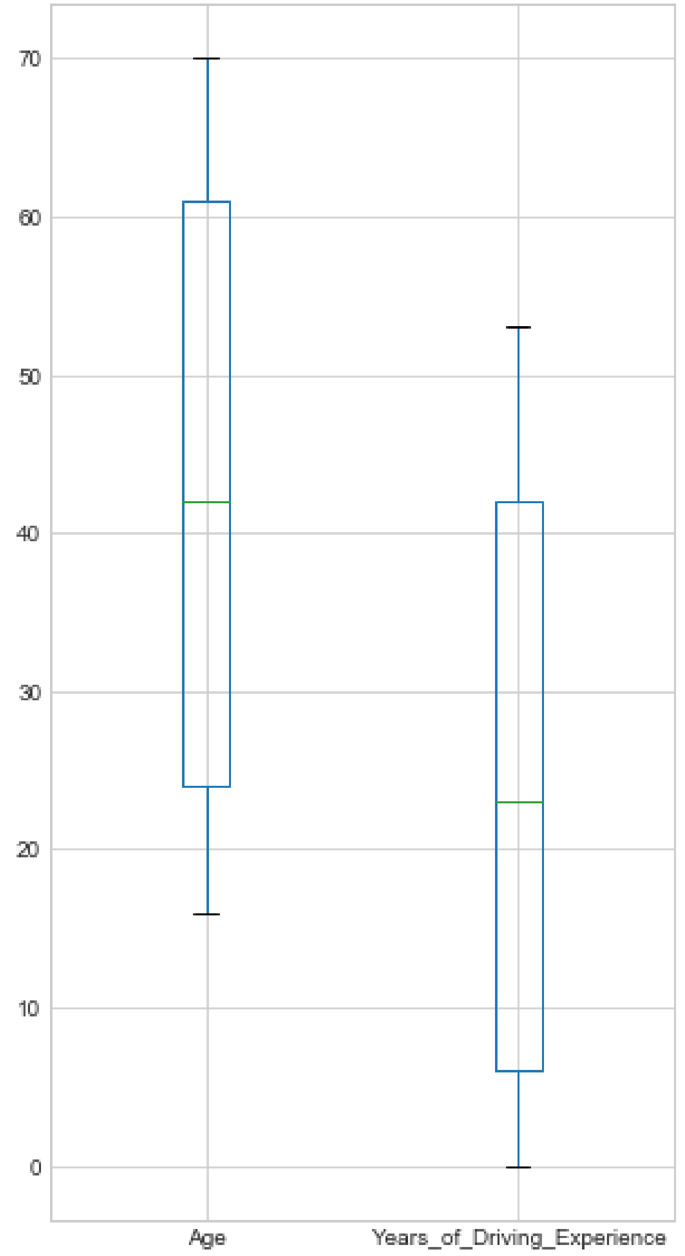


■ **Petrol** type vehicles are **more** in number.

■ **Petrol** Type vehicles are nearly **3x** of **Diesel** Type Vehicles.

Box-Plots:





- There are **No Outliers** in **Age, Years_of_Driving_Experience, Number_of_Vehicles** and **Vehicle_Age**.

- **"Age"** & "**Years_of_Driving_Experience"** has nearly **same scale.**

- **Univariate view for Bivariate Analysis:**

As seen in **Univariate Analysis** only **"Gender"** & "**Married**" have nearly **same number** of **observation** for every **unique value**. But other variables have different number of observations for different unique values.

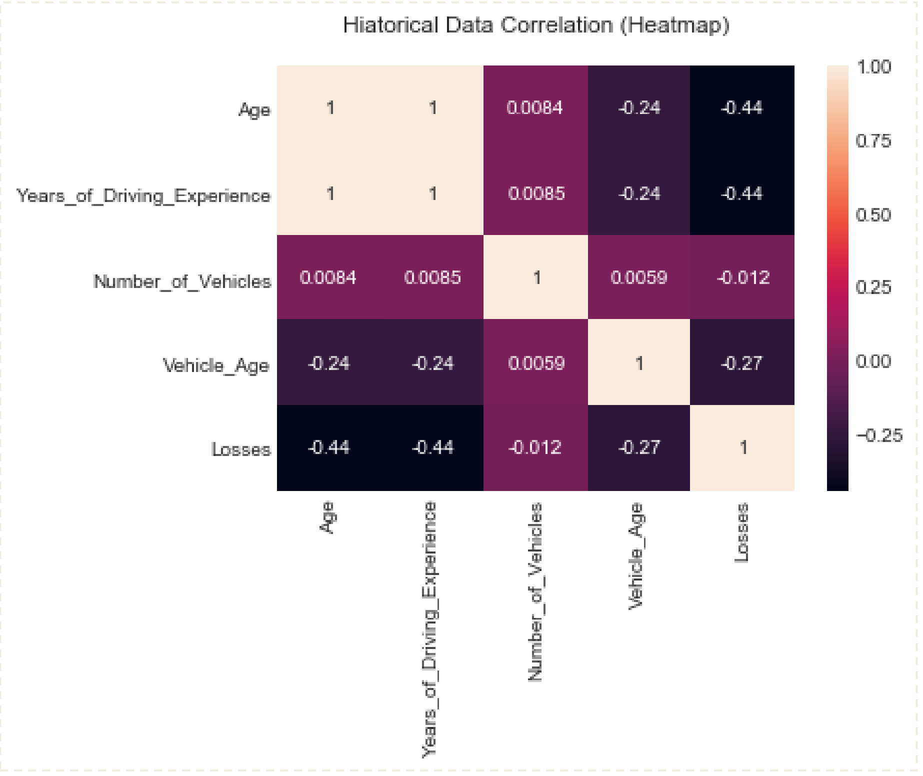
For Example:

1. There are more number of people who own two & three vehicles then who own one & four
2. People who own Petrol Vehicle are three times than the people who own Diesel Vehicle.

So, for **Bivariate Analysis** we are **taking plot of Average of Losses** (dependent variable) with every other variable (**instead of sum of losses)**.

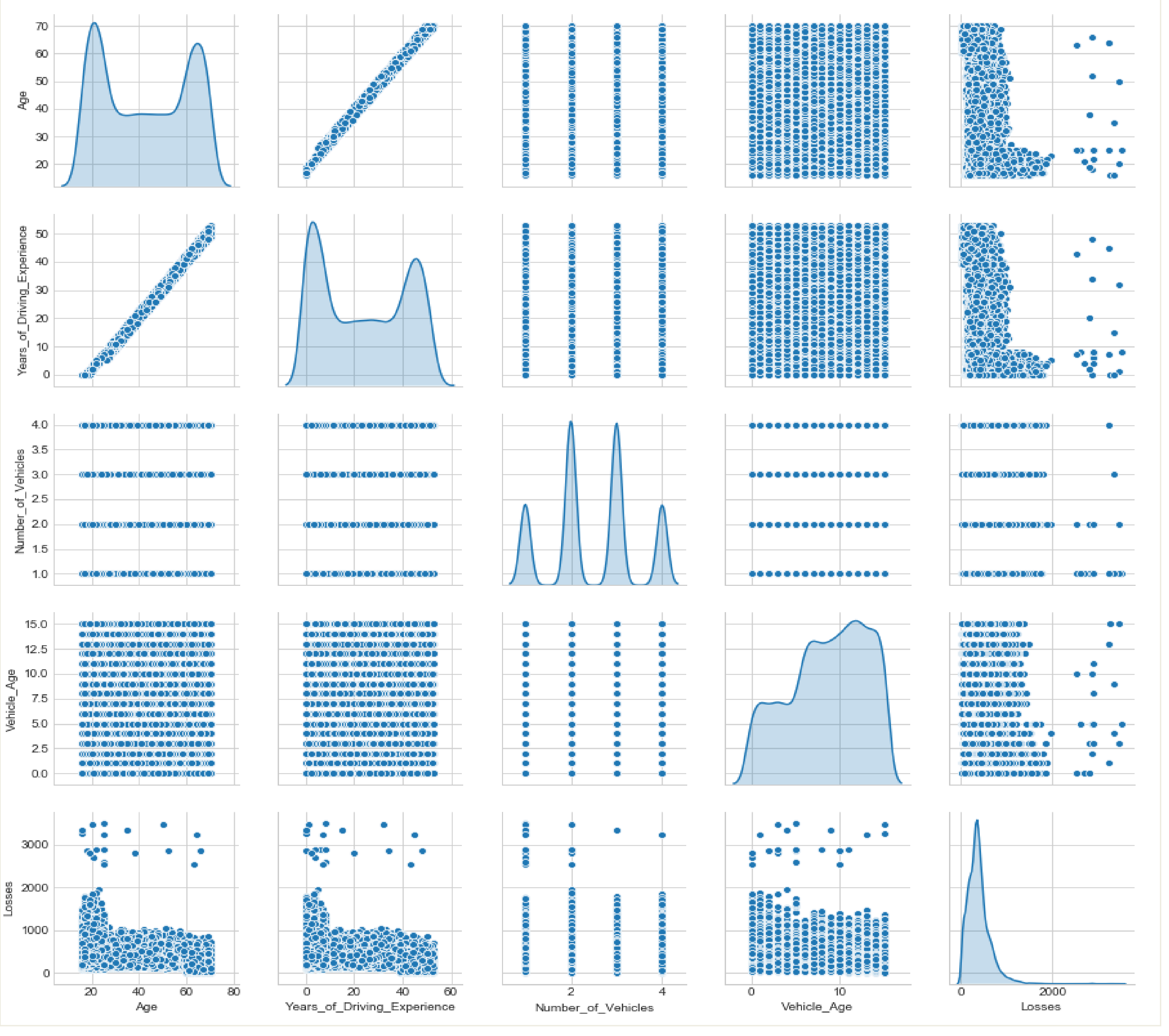
2.3. **BIVARIATE ANALYSIS:**

Correlation Heat-Map:



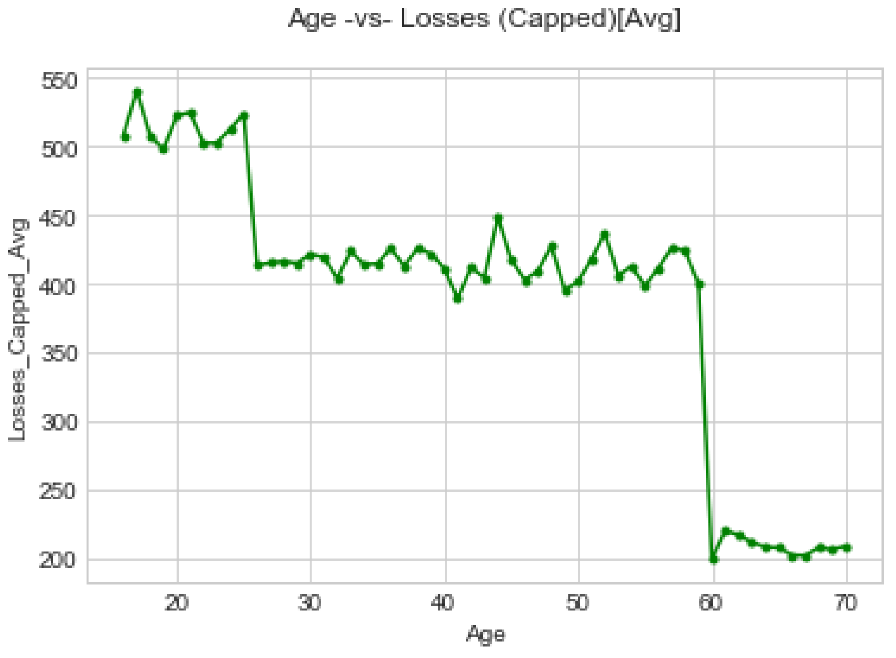
- There is **high positive correlation** between "Age" and "Years_of_Driving_Experience"

Pair-Plot:



- There are observations in population with **high losses**, but they are very **low in number**.
- It also **confirms** the **high correlation** between "Age" and "Years_of_Driving_Experience"
- The skewness of **"Vehicle_Age"** is natural because **most** number of **populations** owns **old vehicles**.

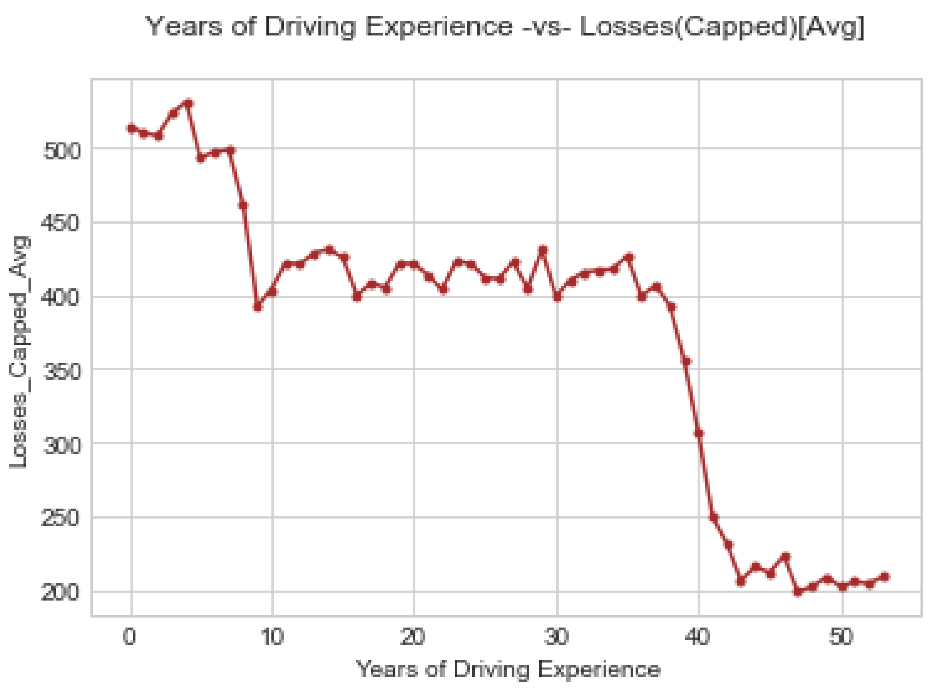
2.3.1. Age vs. Losses:



- People with young **Age group** like '16 to 25' have **very high losses**.
- People with **Age group** like '26 to 59' have **moderate high losses**.
- People with young **Age group** like '60 to 70' have **low losses**.

*Bucketing Needed.

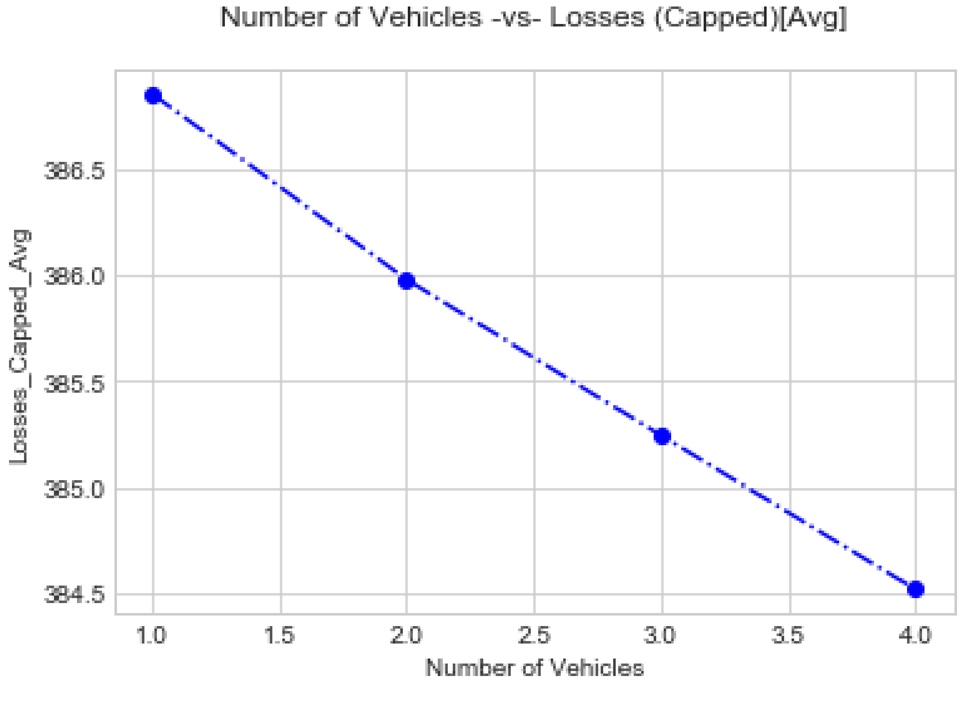
2.3.2. Years_of_Driving_Experience vs. Losses:



- People with low Driving **Experience** like **0 to 8** have **very high losses**.
- People with Driving **Experience** of **9 to 40** have **moderate losses.**
- People with Driving **Experience** of **41 to 53** have **low losses.**

*Bucketing Needed.

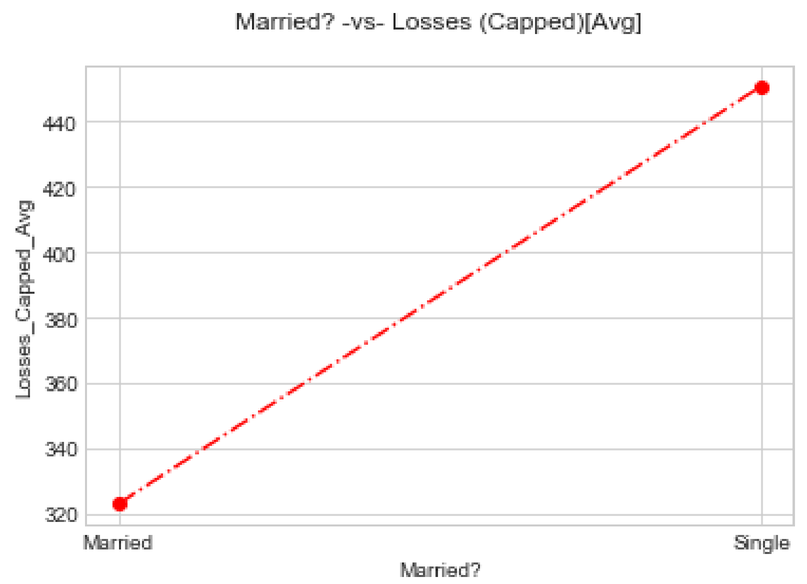
2.3.3. Number_of_Vehicles vs. Losses:



- People with **one vehicle** have more **losses** than others.

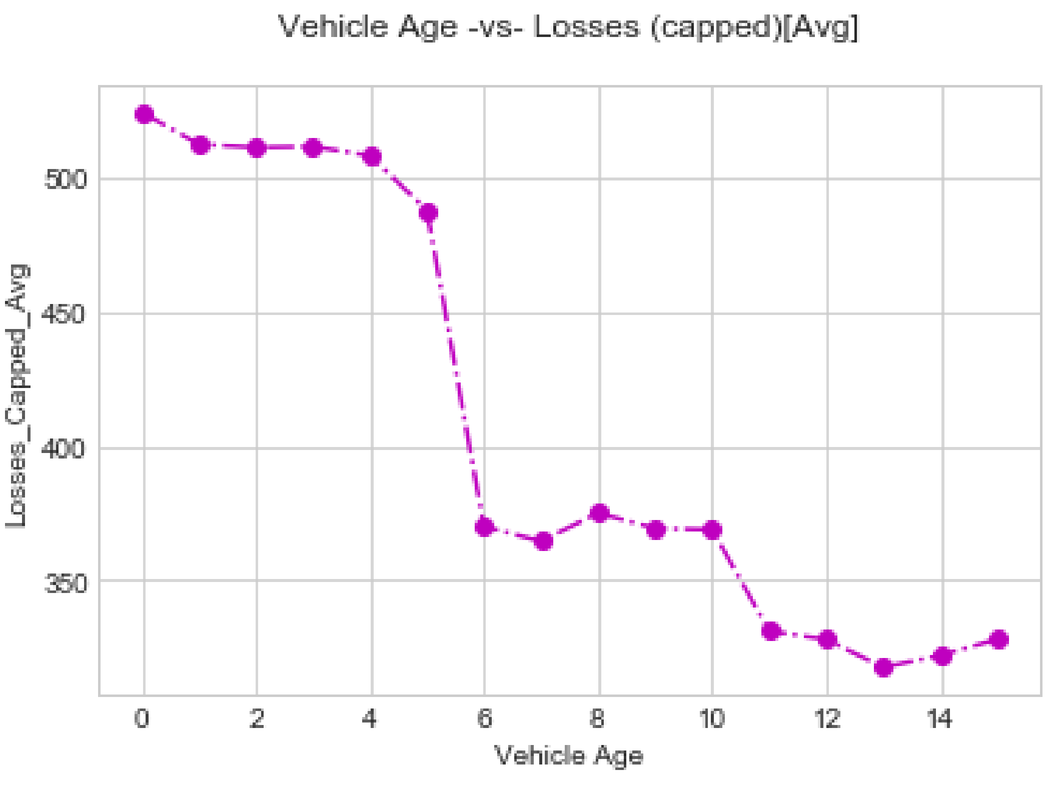
2.3.4. Gender vs. Losses:

2.2.5. Married vs. Losses:



- **Male** population has **higher** number of **losses**.
- **Single** people have **higher** number of **losses**.

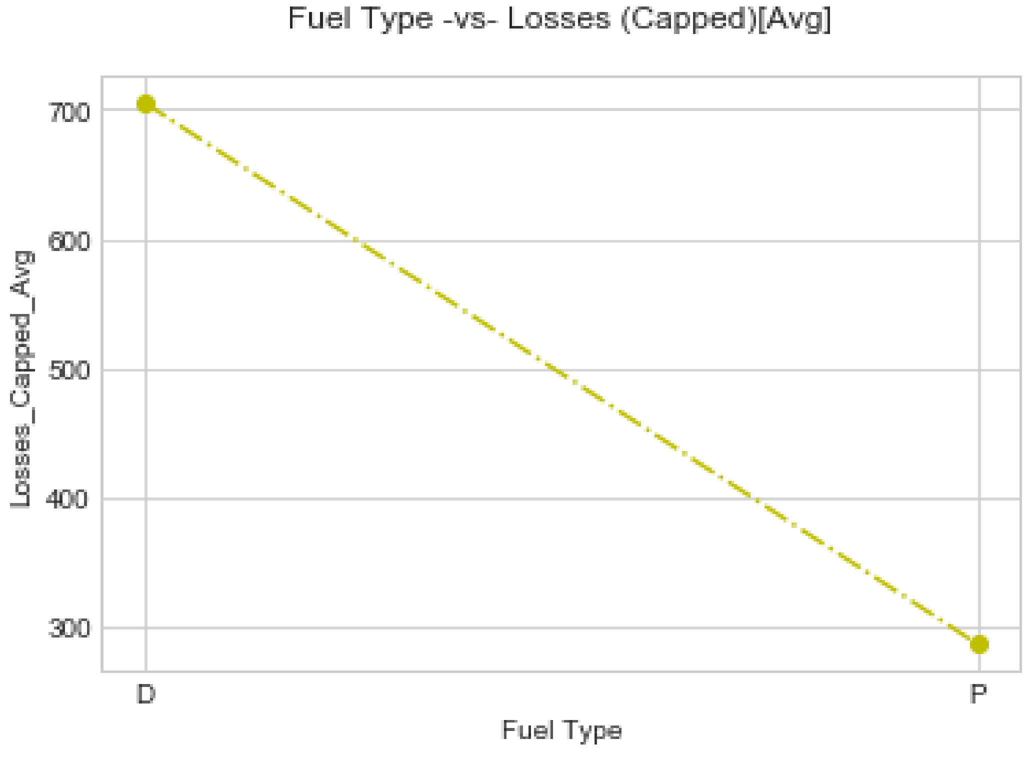
2.2.6. Vehicle_Age vs. Losses:



- People who own **new vehicles (0 to 5 years old)** have **high losses**.
- People who own **vehicles** aging from '**6 to 10**' has comparably **low losses**. **But,** people with vehicle age group of **'11 to 15'** have **even more lower losses**.

*Bucketing Needed.

2.2.7. Fuel_Type vs. Losses:



- People with **Diesel** Type vehicles have **high** amount of **losses**.

- **EDA Points Summary:**

- There are no missing values in numerical columns.
- Maximum observation of “Losses” is 3500 and 75 percentile is 489 which tell us that **“Losses”** contain **outlier** values.
- “Losses” is **highly positive skewed** with a skewness of over +2.0
- “Vehicle_Age” is **negatively skewed** with skewness of over -0.3.
- Numbers of population in age group of **‘30 to 60’** are **lower** in **number** in comparison to other age group.
- People with **zero** driving **experience** are **high** in **number**.
- Already observed high skewness of 2+ in Skewness-Table.
- There are people with **high losses**, but they are **low** in **number**.
- We can put all the **high losses** observations in **one category**.
- The skewness is gradually decreased to 1.05 (before capping it was 2+).
- People owning ‘two’ and ‘three’ vehicles are nearly **double** in number who owns ‘one’ or ‘four‘ vehicles.
- **Male** & **Female** have nearly equal share in observations.
- **Married** & **Single** population both have nearly equal number of observation.
- **Petrol** type vehicles are **more** in number.
- **Petrol** Type vehicles are nearly **3x** of **Diesel** Type Vehicles.
- There are **no outliers** in **Age, Years_of_Driving_Experience, Number_of_Vehicles** and **Vehicle_Age**.
- **Age** & **Years_of_Driving_Experience** has nearly **same scale.**
- There is **high positive correlation** between “Age” and “Years_of_Driving_Experience”
- There are observations in population with **higher losses**, but they are very **low in number**.
- It(pair-plot) also **confirms** the **high correlation** between “Age” and “Years_of_Driving_Experience”
- The skewness of **“Vehicle_Age”** is natural because **most** number of **populations** owns **old vehicles**.
- People with young **Age group** like ‘16 to 25’ have **very high losses**.
- People with **Age group** like ‘26 to 59’ have **moderate high losses**.
- People with young **Age group** like ’60 to 70’ have **low losses**.
- People with low Driving **Experience** like **0 to 8** have **very high losses**.
- People with Driving **Experience** of **9 to 40** have **moderate losses.**
- People with Driving **Experience** of **41 to 53** have **low losses.**
- People with **one vehicle** have more **losses** than others.
- **Male** population has **higher** number of **losses**.
- **Single** people have **higher** number of **losses**.
- People who own **new vehicles (0 to 5 years old)** have **high losses**.
- People who own **vehicles** aging from ‘**6 to 10’** has comparably **low losses**. **But,** people with vehicle age group of **’11 to 15’** have **even more lower losses**.
- People with **Diesel** Type vehicles have **high** amount of **losses**.

2.3. Conclusion:

- From the above **EDA Report** we get to know that **“Age”, “Years_of_Driving_Experience”** and **“Vehicle_Age” needs bucketing.**

2.4. Bucketing:

2.4.1. Age Bucketing:

Age Group **16** to **25** as → **21**
Age Group **26** to **59** as → **43**
Age Group **60** to **70** as → **65**

2.4.2. Years_of_Driving_Experience Bucketing:

YODE Group **0** to **8** as → **4**
YODE Group **9** to **40** as → **25**
YODE Group **41** to **53** as → **47**

2.4.3. Vehicle_Age Bucketing:

Vehicle_Age **0** to **5** as → **3**
Vehicle_Age **6** to **10** as → **8**
Vehicle_Age **11** to **15** as → **13**

3. Confirmatory Data Analysis:

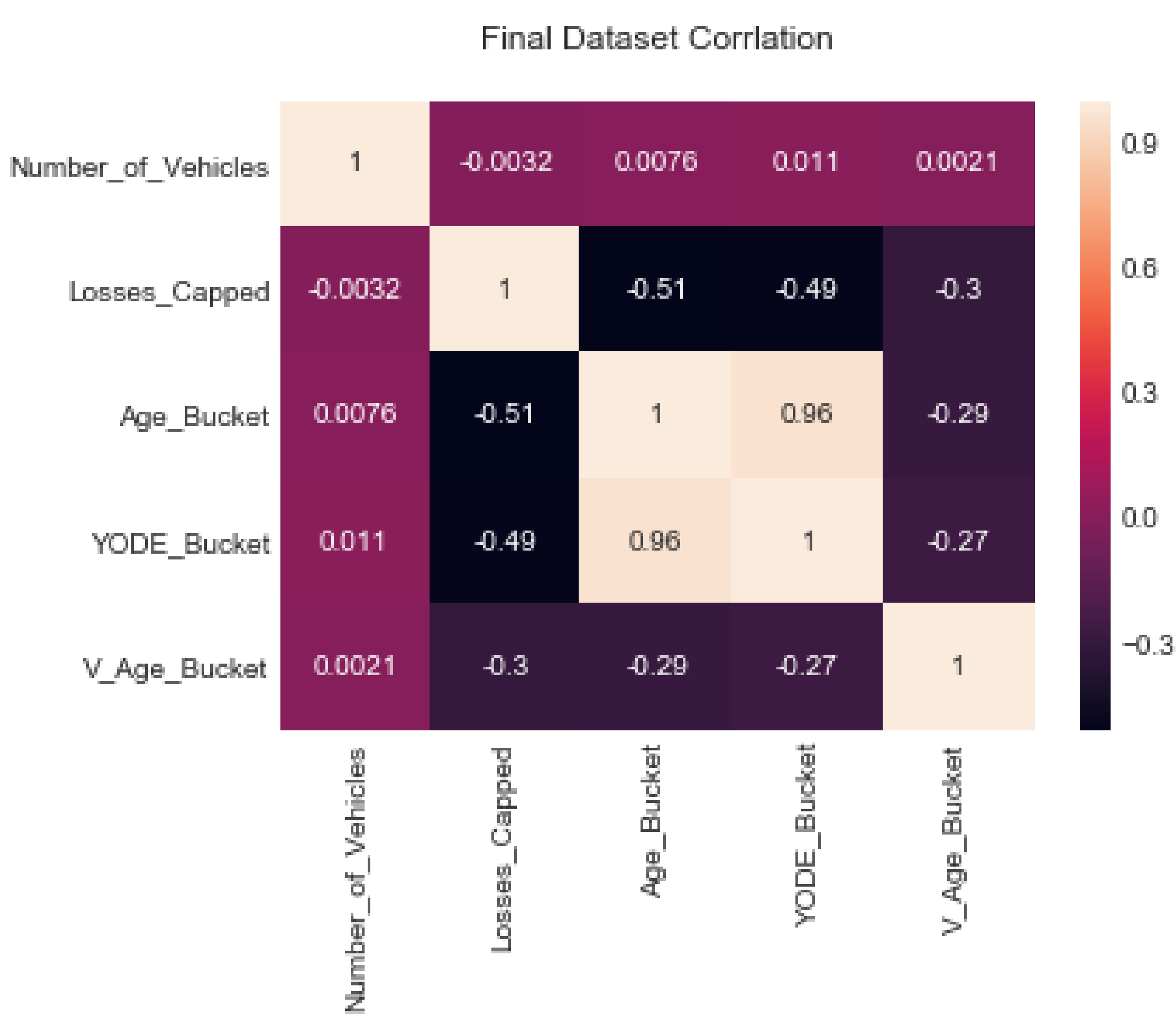
3.1. Data Information:

	Number_of_Vehicles	Losses_Capped	Age_Bucket	YODE_Bucket	V_Age_Bucket
count	15290.0	15290.0	15290.0	15290.0	15290.0
mean	2.49	385.63	42.68	24.266	8.781
std	0.953	228.77	16.78	16.61	3.9
min	1.0	13.0	21.0	4.0	3.0
25%	2.0	226.0	21.0	4.0	8.0
50%	2.0	355.0	43.0	25.0	8.0
75%	3.0	489.0	65.0	47.0	13.0
max	4.0	1200.0	65.0	47.0	13.0

Columns	Skewness
Number_of_Vehicles	0.0065
Losses_Capped	1.055
Age_Bucket	0.024
YODE_Bucket	0.118
V_Age_Bucket	-0.281

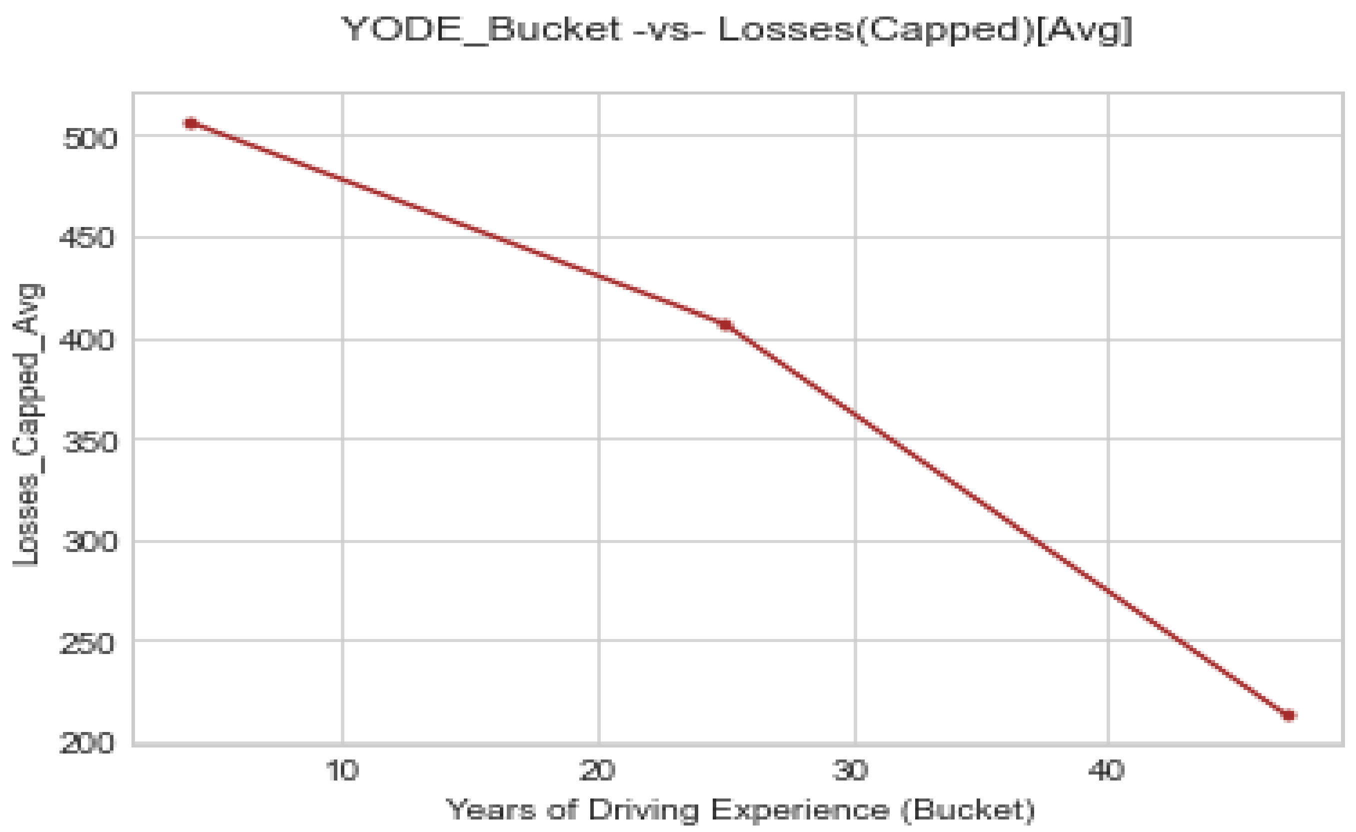
3.2. Bivariate((CDA}:

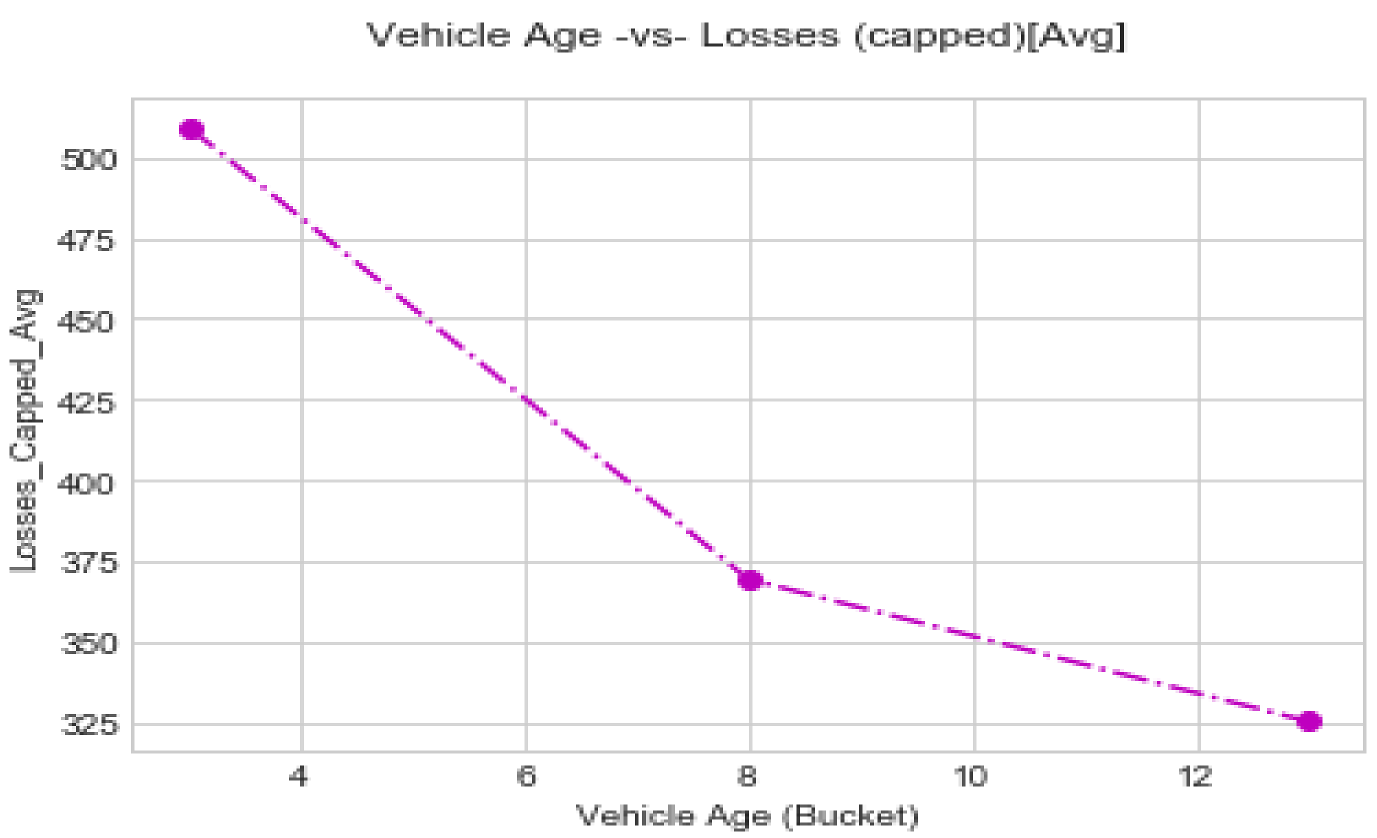
.2.1 Correlation:

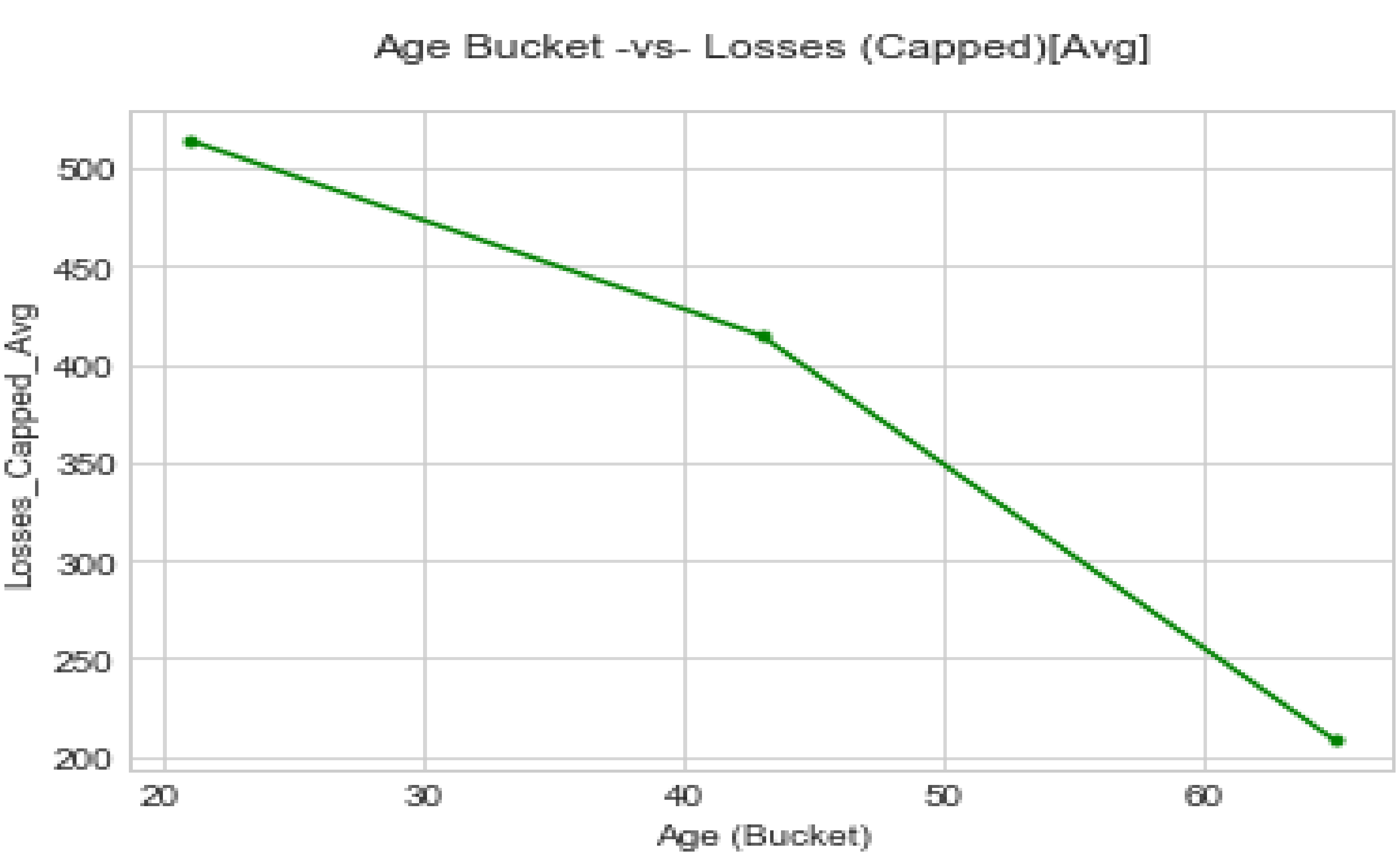


- As you van see “Age_Bucket” & “YODE_Bucket” still have high correlation. So, we will make two separate models in model preparation.
- YODE is used here and further in report as an abbreviation of Years_of_Driving_Experience.

3.2.2. **Bucketed datasets Plots:**







- All three graphs showing **Linear Relationship** and a negative line.

4. Data Pre-Processing:

4.1. Missing Value Treatment:

- There are no missing values in this dataset.

4.2. Oulier Treatment:

- Outliers have been treated above with **capping** & **bucketing.**

4.3. Variable Transformation:

- Converting categorical data into numerical data.

4.4. Variable Creation:

- Creating two different data because Age and Years of Driving Experience still have correlation.

5. Model Development:

5.1. Model-1:

After the OLS through Backward Elimination on "**X_Age**" (the dataset contains 'Age_Bucketed' and others but not 'YODE_Bucketed' we got these results with a significance level of 5% (0.05)

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          Losses_Capped   R-squared:                       0.775
Model:                            OLS   Adj. R-squared:                  0.775
Method:                 Least Squares   F-statistic:                     8784.
Date:                Fri, 15 Mar 2019   Prob (F-statistic):               0.00
Time:                        05:56:56   Log-Likelihood:            -1.0364e+05
No. Observations:               15290   AIC:                         2.073e+05
Df Residuals:                   15284   BIC:                         2.073e+05
Df Model:                           6
Covariance Type:            nonrobust
======================================================================================
                         coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------
Number_of_Vehicles    95.8454      1.552     61.754      0.000      92.803      98.888
Gender               119.5762      3.450     34.655      0.000     112.813     126.340
Married              144.0847      3.513     41.010      0.000     137.198     150.971
Fuel_Type           -269.0490      5.092    -52.837      0.000    -279.030    -259.068
Age_Bucket             1.6484      0.102     16.235      0.000       1.449       1.847
V_Age_Bucket          13.2384      0.435     30.467      0.000      12.387      14.090
==============================================================================
Omnibus:                      185.030   Durbin-Watson:                   1.180
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              265.037
Skew:                           0.148   Prob(JB):                     2.81e-58
Kurtosis:                       3.573   Cond. No.                         144.
==============================================================================

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

- $R^2 = 0.775$
- AIC = 2.073
- Durbin Watson = 1.80
- "Number_of_Vehicles" is eliminated as its p-values > our significance level

5.2. Model-2:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:          Losses_Capped   R-squared:                       0.772
Model:                            OLS   Adj. R-squared:                  0.772
Method:                 Least Squares   F-statistic:                     8644.
Date:                Fri, 15 Mar 2019   Prob (F-statistic):               0.00
Time:                        05:56:57   Log-Likelihood:            -1.0374e+05
No. Observations:               15290   AIC:                         2.075e+05
Df Residuals:                   15284   BIC:                         2.075e+05
Df Model:                           6
Covariance Type:            nonrobust
======================================================================================
                         coef    std err          t      P>|t|      [0.025      0.975]
--------------------------------------------------------------------------------------
Number_of_Vehicles   109.7161      1.475     74.368      0.000     106.824     112.608
Gender               139.8706      3.422     40.876      0.000     133.163     146.578
Married              167.3355      3.478     48.113      0.000     160.518     174.153
Fuel_Type           -195.3807      4.983    -39.211      0.000    -205.148    -185.614
YODE_Bucket           -0.9699      0.115     -8.399      0.000      -1.196      -0.744
V_Age_Bucket          10.5786      0.450     23.486      0.000       9.696      11.461
==============================================================================
Omnibus:                      197.836   Durbin-Watson:                   1.269
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              245.423
Skew:                           0.205   Prob(JB):                     5.10e-54
Kurtosis:                       3.466   Cond. No.                         90.3
==============================================================================

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

OLS with **Backward Elimination** on **X_YODE** where **Y** is dependent variable containing **Losses_Capped.**

- R^2 = 0.772
- AIC = 2.075
- Durbin-Watson = 1.269
- “Number_of_Vehicles” is removed by Backward Elimination because of its high p-value.

- →→→ We **selected Model-1** because of its **high R^2** value & **low AIC** value.

CODE:

```
import pandas as pd

################################################################################

# 1. Data Consolidation:
dataset = pd.read_csv('Historical_Losses_Data.csv')

# 1.1 Gathering Information about the dataset
dataset.shape
dataset.head(6)
dataset.tail(6)
dataset.info()

# 1.2 Removing NON-Significant Variables
dataset = dataset.drop(['Policy_Number'], axis=1)

# 1.3 Creating a list to gather variable/feature name
columns = dataset.columns.tolist()

################################################################################

# 2. Exploratory Data Analysis -EDA Report
import seaborn as sns
import matplotlib.pyplot as plt
import pandasql as ps
plt.style.use('seaborn-whitegrid')   ### setting up Grid-Lines Style

##### 2.1 Univariate Analysis:  #####

# 2.1.1. finding ranges and catgories in variables
dataset.Age.sort_values().unique()
dataset.Years_of_Driving_Experience.sort_values().unique()
dataset.Number_of_Vehicles.sort_values().unique()
dataset.Gender.sort_values().unique()
dataset.Married.sort_values().unique()
dataset.Vehicle_Age.sort_values().unique()
dataset.Fuel_Type.sort_values().unique()
dataset.Losses.sort_values().unique()

# 2.1.2. Description:
Description = dataset.describe()
skew=dataset.skew()

# 2.1.3. Histogram of dataset
dataset.hist(bins=500, figsize=(20,15))
plt.savefig('Historical_Data(Histogram).png')
plt.show()
```

```

# taking a seperate look at 'Losses' variable because of high positive skewness:
dataset['Losses'].hist(bins=500, figsize=(10,7))
plt.title('Losses (Histogram)\n')
plt.savefig('Losses(Histogram).png')
plt.show()

# upper-capping 'losses' at 1200 according to industrial standards to deal with skewness:
Losses_Capped = dataset['Losses'].clip(upper=1200)
Losses_Capped.hist(bins=500, figsize=(10,7))
plt.title('Losses Capped (Histogram)\n')
plt.savefig('Losses_Capped(Histogram).png')
plt.show()
Losses_Capped.skew()

# making a single DataFrame of 'dataset' and 'Losses_Capped' for further analysis:
q1 = """ SELECT *,
      (CASE WHEN Losses > 1200 THEN 1200 ELSE Losses END)
      AS Losses_Capped
      FROM dataset """
Capped_Losses_dataset = ps.sqldf(q1, locals())

# 2.1.4. Pie-Plots:
# "Number_of_Vehicles" variable:
Number_of_Vehicles_Count = pd.DataFrame(
      dataset.Number_of_Vehicles.value_counts().reset_index())
Number_of_Vehicles_Count.columns=['Number of Vehicles','Count']
plt.pie(Number_of_Vehicles_Count['Count'],
      labels=Number_of_Vehicles_Count['Number of Vehicles'],
      autopct='%.1f%%', startangle=90)
plt.title('Number of Vehicles')
plt.savefig('Number_of_Vehicles_Count(Pie-Plot).png')
plt.show()

# taking a seperate view at charachter categorical data (Demographical Variables):

#Gender:
Gender_Count = pd.DataFrame(dataset.Gender.value_counts().reset_index())
Gender_Count.columns=['Gender','Count']
plt.pie(Gender_Count['Count'], labels=Gender_Count['Gender'],
         autopct='%.1f%%', startangle=90)
plt.title('Gender Count')
plt.savefig('Gender_Count(Pie-Plot).png')
plt.show()

#Married:
Married_Count = pd.DataFrame(dataset.Married.value_counts().reset_index())
Married_Count.columns=['Married?','Count']
```

```
plt.pie(Married_Count['Count'], labels=Married_Count['Married?'],
        colors=['Yellow','Cyan'], autopct='%.1f%%', startangle=90)
plt.title('Married?')
plt.savefig('Married_Count(Pie-Plot).png')
plt.show()

#Fuel_Type:
Fuel_Type_Count=pd.DataFrame(dataset.Fuel_Type.value_counts().reset_index())
Fuel_Type_Count.columns=['Fuel_Type','Count']
plt.pie(Fuel_Type_Count['Count'], labels=Fuel_Type_Count['Fuel_Type'],
        colors=['y','Red'], autopct='%.1f%%', startangle=90)
plt.title('Fuel Type')
plt.savefig('Fuel_Type_Count(Pie-Plot).png')
plt.show()

# 2.1.5. Box-Plot of the variable for oulier detection:
dataset.boxplot(column=['Age','Years_of_Driving_Experience'], figsize=(5,10))
plt.savefig('Historical_dataset(Box-Plot)1.png', bbox_inches='tight')
plt.show()
dataset.boxplot(column=['Number_of_Vehicles','Vehicle_Age'], figsize=(5,10))
plt.savefig('Historical_dataset(Box-Plot)2.png', bbox_inches='tight')
plt.show()


##### 2.2 Bivariate Analysis: #####

# Dependent Variables are Losses & Losses_Capped
# although we are performing analysis on Losses_Capped

#2.2.0. Correlation & Pair_Plots:
sns.heatmap(dataset.corr(), annot=True)
plt.title('Hiatorical Data Correlation (Heatmap)\n')
plt.savefig('Historical_Data_Correlation(Heatmap).png', bbox_inches='tight')
plt.show()

sns.pairplot(dataset, diag_kind='kde')
plt.savefig('Historical_Data(Pair-Plot).png')
plt.show()

# 2.2.1 Age vs Losses_Capped (Average) Analysing:
q1 = """ SELECT Age,
     AVG(Losses_Capped)
     AS Losses_Capped_Avg
     FROM Capped_Losses_dataset
     GROUP BY Age
     ORDER BY Age """

Age_vs_Losses_Cap = ps.sqldf(q1, locals())
plt.plot(Age_vs_Losses_Cap['Age'], Age_vs_Losses_Cap['Losses_Capped_Avg'], 'g.-')
```

```
plt.title('Age -vs- Losses (Capped)[Avg]\n')
plt.xlabel('Age')
plt.ylabel('Losses_Capped_Avg')
plt.savefig('Age-vs-Losses(Plot).png', bbox_inches='tight')
plt.show()

# 2.2.2 Years_of_Driving_Experience vs Losses_Capped(Avg) Analysing:
# YODE is used as an abbrivation of Years_of_Driving_Experience here.
q1= """ SELECT Years_of_Driving_Experience,
      AVG(Losses_Capped)
      AS Losses_Capped_Avg
      FROM Capped_Losses_dataset
      GROUP BY Years_of_Driving_Experience
      ORDER BY Years_of_Driving_Experience """

YODE_vs_Losses_Cap = ps.sqldf(q1, locals())

plt.plot(YODE_vs_Losses_Cap['Years_of_Driving_Experience'],
          YODE_vs_Losses_Cap['Losses_Capped_Avg'],
          color='Brown', marker='.',linestyle='-')

plt.title('Years of Driving Experience -vs- Losses(Capped)[Avg]\n')
plt.xlabel('Years of Driving Experience')
plt.ylabel('Losses_Capped_Avg')
plt.savefig('Years_of_Driving_Experience-vs-Losses(Plot).png', bbox_inches='tight')
plt.show()

# 2.2.3  Number_of_Vehicles vs Losses_Capped(Avg) Analysing:
q1= """ SELECT Number_of_Vehicles,
      AVG(Losses_Capped)
      AS Losses_Capped_Avg
      FROM Capped_Losses_dataset
      GROUP BY Number_of_Vehicles
      ORDER BY NUmber_of_Vehicles """

Number_of_Vehicles_vs_Losses_Cap = ps.sqldf(q1, locals())

plt.plot(Number_of_Vehicles_vs_Losses_Cap['Number_of_Vehicles'],
          Number_of_Vehicles_vs_Losses_Cap['Losses_Capped_Avg'], 'bo-.')
plt.title('Number of Vehicles -vs- Losses (Capped)[Avg]\n')
plt.xlabel('Number of Vehicles')
plt.ylabel('Losses_Capped_Avg')
plt.savefig('Number_of_Vehicles-vs-Losses(Plot).png', bbox_inches='tight')
plt.show()

# 2.2.4 Gender vs Losses_Capped(Avg) Analysis:
q1= """ SELECT Gender,
      AVG(Losses_Capped)
      AS Losses_Capped_Avg
```

```
        FROM Capped_Losses_dataset
        GROUP BY Gender
        ORDER BY Gender """

plt.plot(Gender_vs_Losses_Cap['Gender'],
Gender_vs_Losses_Cap = ps.sqldf(q1, locals())
plt.plot(Gender_vs_Losses_Cap['Gender'],
            Gender_vs_Losses_Cap['Losses_Capped_Avg'], 'co-.')
plt.title('Gender -vs- Losses (Capped)[Avg]\n')
plt.xlabel('Gender')
plt.ylabel('Losses_Capped_Avg')
plt.savefig('Gender-vs-Losses(Plot).png', bbox_inches='tight')
plt.show()

#2.2.5 Married vs Losses_Capped(Avg) Analysis:
q1= """ SELECT Married,
        AVG(Losses_Capped)
        AS Losses_Capped_Avg
        FROM Capped_Losses_dataset
        GROUP BY Married
        ORDER BY Married """

Married_vs_Losses_Cap = ps.sqldf(q1, locals())
plt.plot(Married_vs_Losses_Cap['Married'],
            Married_vs_Losses_Cap['Losses_Capped_Avg'], 'ro-.')
plt.title('Married? -vs- Losses (Capped)[Avg]\n')
plt.xlabel('Married?')
plt.ylabel('Losses_Capped_Avg')
plt.savefig('Married_vs_Losses(Plot).png', bbox_inches='tight')
plt.show()

# 2.2.6 Vehicle_Age vs Losses_Capped(Avg) Analysis:
q1= """ SELECT Vehicle_Age,
        AVG(Losses_Capped)
        AS Losses_Capped_Avg
        FROM Capped_Losses_dataset
        GROUP BY Vehicle_Age
        ORDER BY Vehicle_Age """

Vehicle_Age_vs_Losses_Cap = ps.sqldf(q1, locals())
plt.plot(Vehicle_Age_vs_Losses_Cap['Vehicle_Age'],
            Vehicle_Age_vs_Losses_Cap['Losses_Capped_Avg'], 'mo-.')
plt.title('Vehicle Age -vs- Losses (capped)[Avg]\n')
plt.xlabel('Vehicle Age')
plt.ylabel('Losses_Capped_Avg')
plt.savefig('Vehicle_Age-vs-Losses(Plot).png', bbox_inches='tight')
plt.show()

# 2.2.7 Fuel_Type vs Losses_Capped(Avg) Analysis:
q1= """ SELECT Fuel_Type,
```

```
        AVG(Losses_Capped)
        AS Losses_Capped_Avg
        FROM Capped_Losses_dataset
        GROUP BY Fuel_Type
        ORDER BY Fuel_Type """

Fuel_Type_vs_Losses_Cap = ps.sqldf(q1, locals())
plt.plot(Fuel_Type_vs_Losses_Cap['Fuel_Type'],
             Fuel_Type_vs_Losses_Cap['Losses_Capped_Avg'], 'yo-.')
plt.title('Fuel Type -vs- Losses (Capped)[Avg]\n')
plt.xlabel('Fuel Type')
plt.ylabel('Losses_Capped_Avg')
plt.savefig('Fuel_Type-vs-Losses(Plot).png', bbox_inches='tight')
plt.show()

# EDA Report Ends.
###############################################

### 2.3 EDA Conclusion: ###

""" From the above EDA Report we get to know that
    “Age”, “Years_of_Driving_Experience” and “Vehicle_Age” needs bucketing"""

# 2.4 Bucketing:
# 2.4.1. Age Bucketing:

q1 = """ SELECT *,
        (CASE
            WHEN Age BETWEEN 16 AND 25 THEN 21
            WHEN Age BETWEEN 26 AND 59 THEN 43
            WHEN Age BETWEEN 60 AND 70 THEN 65
            ELSE Age
            END)
        AS Age_Bucket
        FROM Capped_Losses_dataset
        ORDER BY Age"""

CLD_Age_Bucket = ps.sqldf(q1, locals())

#2.4.2. Years_of_Driving_Experirncr Bucketing:

q1 = """ SELECT *,
        (CASE
            WHEN Years_of_Driving_Experience BETWEEN 0 AND 8 THEN 4
            WHEN Years_of_Driving_Experience BETWEEN 9 AND 40 THEN 25
            WHEN Years_of_Driving_Experience BETWEEN 41 and 53 THEN 47
            ELSE Years_of_Driving_Experience
            END)
        AS YODE_Bucket
```

```
        FROM CLD_Age_Bucket
        ORDER BY Years_of_Driving_Experience"""

CLD_YODE_Bucket = ps.sqldf(q1, locals())

#2.4.3. Vehicle_Age Bucketing:

q1= """ SELECT *,
        (CASE
             WHEN Vehicle_Age BETWEEN 0 AND 5 THEN 3
             WHEN Vehicle_Age BETWEEN 6 AND 10 THEN 8
             WHEN Vehicle_Age BETWEEN 11 AND 15 THEN 13
             ELSE Vehicle_Age
             END)
        AS V_Age_Bucket
        FROM CLD_YODE_Bucket
        ORDER BY Age"""

CLD_V_Age_Bucket = ps.sqldf(q1, locals())


# dropping variables that we have bucketed & capped
dataset_final = CLD_V_Age_Bucket.drop(columns=['Age',
                                 'Years_of_Driving_Experience',
                                 'Vehicle_Age','Losses'], axis=1)

###############################################################################

##### 3 Confirmatory Data Analysis: #####

# 3.1 gathering information about 'dataset_final':
Description_final = dataset_final.describe()
Skew_final = dataset_final.skew()

# 3.2 Bivariate Analysis (CDA):

# 3.2.1 Correlation Check:
sns.heatmap(dataset_final.corr(), annot=True)
plt.title('Final Dataset Corrlation\n')
plt.savefig('dataset_final_correlation(Heatmap).png', bbox_inches='tight')
plt.show()

# "Years_of_Driving_Experience" & "Age" still have high correlation even after bucketing
# there should be two models in model preperation because of this correlation

# 3.2.2 Age_Bucket vs Losses_Capped(Avg) :

```

```
q1 = """ SELECT Age_Bucket,
         AVG(Losses_Capped)
         AS Losses_Capped_Avg
         FROM dataset_final
         GROUP BY Age_Bucket
         ORDER BY Age_Bucket """

Age_Bucket_vs_Losses_Cap = ps.sqldf(q1, locals())
plt.plot(Age_Bucket_vs_Losses_Cap['Age_Bucket'],
             Age_Bucket_vs_Losses_Cap['Losses_Capped_Avg'], 'g.-')
plt.title('Age Bucket -vs- Losses (Capped)[Avg]\n')
plt.xlabel('Age (Bucket)')
plt.ylabel('Losses_Capped_Avg')
plt.savefig('Age_Bucket-vs-Losses_Cap(Plot).png', bbox_inches='tight')
plt.show()

#3.2.3 YODE_Bucket vs Losses_Capped(Avg) :

q1 =""" SELECT YODE_Bucket,
        AVG(Losses_Capped)
        AS Losses_Capped_Avg
        FROM dataset_final
        GROUP BY YODE_bucket
        ORDER BY YODE_Bucket """

YODE_Bucket_vs_Losses_Cap = ps.sqldf(q1, locals())
plt.plot(YODE_Bucket_vs_Losses_Cap['YODE_Bucket'],
             YODE_Bucket_vs_Losses_Cap['Losses_Capped_Avg'],
             color='Brown', marker='.',linestyle='-')

plt.title('YODE_Bucket -vs- Losses(Capped)[Avg]\n')
plt.xlabel('Years of Driving Experience (Bucket)')
plt.ylabel('Losses_Capped_Avg')
plt.savefig('YODE_Bucket-vs-Losses_Cap(Plot).png', bbox_inches='tight')
plt.show()

# 3.2.4 V_Age_Bucket vs Losses_Capped(Avg):

q1 = """ SELECT V_Age_Bucket,
        AVG(Losses_Capped)
        AS Losses_Capped_Avg
        FROM dataset_final
        GROUP BY V_Age_Bucket
        ORDER BY V_Age_Bucket """

V_Age_Bucket_vs_Losses_Cap = ps.sqldf(q1, locals())
plt.plot(V_Age_Bucket_vs_Losses_Cap['V_Age_Bucket'],
         V_Age_Bucket_vs_Losses_Cap['Losses_Capped_Avg'],
         'mo-.')
```

```
plt.title('Vehicle Age -vs- Losses (capped)[Avg]\n')
plt.xlabel('Vehicle Age (Bucket)')
plt.ylabel('Losses_Capped_Avg')
plt.savefig('V_Age_Bucket-vs-Losses_Cap(Plot).png', bbox_inches='tight')
plt.show()

################################################################################

##### 4.  Data Pre-Processing: ##### (on 'dataset_final')

dataset_final_OLS = dataset_final.copy() # making a copy

# 4.1 Missing Value Treatment:
" There are no missing value so skipping this step"

# 4.2 Oulier Treatment :
" Outlier Values have been treated above with the help of 'Capping' and 'Bucketing' "

# 4.3 Variable Transformation :
from sklearn.preprocessing import LabelEncoder

# 4.3.1 converting categorical data into binary/numerical form:
myencoder = LabelEncoder()

dataset_final_OLS['Gender'] = myencoder.fit_transform(dataset_final_OLS['Gender'])
dataset_final_OLS['Married'] = myencoder.fit_transform(dataset_final_OLS['Married'])
dataset_final_OLS['Fuel_Type'] = myencoder.fit_transform(dataset_final_OLS['Fuel_Type'])

# 4.4 Variable Creation :

# Segregation between Independent and Dependent Vaariables:
# X = Independent Variables ; Y = Dependent Variables

X = pd.DataFrame(dataset_final_OLS.drop(columns='Losses_Capped')).copy()
Y = pd.DataFrame(dataset_final_OLS['Losses_Capped']).copy()

# from CDA we know "YODE_Bucket" & "Age_Bucket" have correlation nearly to 1
# making two seperate models for these variables
X_Age = pd.DataFrame(X.drop(columns='YODE_Bucket')).copy()
X_YODE = pd.DataFrame(X.drop(columns='Age_Bucket')).copy()

################################################################################

###### 5. Model Development:  #####
import statsmodels.formula.api as smfa
import statsmodels.tools as smt

# 5.1 Creating Linear Formula:
```

```
# b0X0 + b1X1 + b2X2 + ......... + b8X8
# where X0 should be constant
X = smt.add_constant(X)

# 5.2 Creating Different Models with OLS (Ordinary Least Square)
####   & using Backward Elimination Approach

import selectionprocess as sp   ## user generated module
# setting a significance level of 5% (i.e, 0.05)
sig_level = 0.05

# 5.2.1 Model-1 : on X_Age
X_Age_Modeled, Age_Model_Summary = sp.BackwardElimination_OLS_DataFrame(X_Age, Y, sig_level)

# "Number of_Vehicles" is removed by backward elimination through OLS
# because its p-value > significance level

font_dict = {'family' : 'monospace',
             'size'   : 'large',
             'weight' : 'semibold'}

plt.text(0, 0, str(Age_Model_Summary),font_dict)
plt.axis('off')
plt.savefig('OLS_report_X_Age_Modeled.png', bbox_inches='tight')
plt.show()

# 5.2.2 Model-2 : on X_YODE
X_YODE_Modeled, YODE_Model_Summary = sp.BackwardElimination_OLS_DataFrame(X_YODE, Y, sig_level)

# "Number of_Vehicles" is removed by backward elimination through OLS
# because its p-value > significance level

plt.text(0, 0, str(YODE_Model_Summary),font_dict)
plt.axis('off')
plt.savefig('OLS_report_X_YODE_Modeled.png', bbox_inches='tight')
plt.show()


""" selecting Model-1 for further prediction
           as it has higher R^2 value and lower AIC value """

coeff = smfa.OLS(Y, X_Age_Modeled).fit().params

# 5.2.3 Splitting into Train Test Values:
from sklearn.model_selection import train_test_split

X_Train, X_Test, Y_Train, Y_Test = train_test_split(X_Age_Modeled, Y,
```

```
                                    test_size=0.2, random_state=66)

from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_Train, Y_Train)
################################################################################
##### 6. Model Performance:  #####

Y_Predicted = regressor.predict(X_Test)

# 6.1 Creating a single Analysis DataFrame:

Y_Predicted_temp = pd.DataFrame(Y_Predicted, columns=['Losses_Predicted'])
Y_Test_temp = pd.DataFrame(Y_Test.reset_index().drop('index',axis=1))

temp = dataset_final.copy()
tempTrain, tempTest=train_test_split(temp,test_size=0.2,random_state=66)

temp = tempTest.drop(columns=['Losses_Capped','YODE_Bucket'])
temp = pd.DataFrame(temp.reset_index().drop('index',axis=1))

Analysis = pd.concat([temp, Y_Test_temp, Y_Predicted_temp], axis=1)
Error_Values = pd.DataFrame(Analysis['Losses_Predicted']-Analysis['Losses_Capped'],
                 columns=['Error (Losses_Predicted-Losses_Capped)'])

Analysis = pd.concat([Analysis, Error_Values], axis=1)

#END.

```

selectionprocess.py

```
# Slection Process all in one module

import numpy as np
import pandas as pd
import statsmodels.formula.api as smfa

# 1 Backward Elimination: for array-like
# independent_array == array-like, independent dataset/variables (normally denoted by X in program)
# dependent_array == array-like(1d), dependent dataset/variable
# significance level == float, if value of significance is 5% set this value as 0.05

# returns(arrray-like) ==  moduled or processed final array after removing all
#                       in-significant variables through OLS
# returns (summary) == summary of final model as 'statsmodels.iolib.summary.Summary'
# also prints final summary

def Backward_Elimination_OLS_Array(independent_array, dependent_array, significance):
    count = len(independent_array[0])
    for num1 in range(count):
        OLS_Regressor = smfa.OLS(dependent_array, independent_array).fit()
        Max_P_value = max(OLS_Regressor.pvalues).astype(float)

        if Max_P_value > significance:
            for num2 in range(count-num1):
                if(OLS_Regressor.pvalues[num2].astype(float) == Max_P_value):
                    independent_array = np.delete(arr=independent_array,
                                                  obj=num2, axis=1)
    print(OLS_Regressor.summary())
    return independent_array, OLS_Regressor.summary()


# 2 Backward Elimination: for DataFrame-like
# independent_DF == DataFrame-like, independent dataset/variables (normally denoted by X in program)
# dependent_DF == DataFrame-like(1d), dependent dataset/variable
# significance level == float, if value of significance is 5% set this value as 0.05

# returns (DataFrame-like)==   moduled or processed final DataFrame after removing all
#                          in-significant variables through OLS
# returns (summary) == summary of final model as 'statsmodels.iolib.summary.Summary'

```

```
# also prints final summary


def BackwardElimination_OLS_DataFrame(independent_DF, dependent_DF, significance):
    count = len(independent_DF.columns)
    for num1 in range(count):
        OLS_Regressor = smfa.OLS(dependent_DF, independent_DF).fit()
        Max_P_value = max(OLS_Regressor.pvalues)

        if Max_P_value > significance:
            for num2 in range(count-num1):
                if(OLS_Regressor.pvalues[num2] == Max_P_value):
                    independent_DF = pd.DataFrame.drop(independent_DF,
                                     columns=independent_DF.columns[num2],
                                     axis=1)
    print(OLS_Regressor.summary())
    return independent_DF, OLS_Regressor.summary()

```