Problem 1: Objective

You are part of investment firm and your work is to do research about 759 firms. You are provided with the dataset containing the sales and other attributes of these 759 firms. Predict the sales of these firms on the bases of the details given in the dataset so as to help your company investing consciously. Also, provide them with 5 attributes that are most important.

1.1 Read the data and do exploratory data analysis. Describe the data briefly. (Check the null values, data types, shape, EDA). Perform Univariate and Bivariate Analysis.

Solution:

We are provided with a dataset of **759 firms and 10 attributes**, including the target variable "**sales**".

```
Shape of the data: (759, 10)
```

:

	Unnamed: 0	sales	capital	patents	randd	employment	sp500	tobinq	value	institutions
0	0	826.995050	161.603986	10	382.078247	2.306000	no	11.049511	1625.453755	80.27
1	1	407.753973	122.101012	2	0.000000	1.860000	no	0.844187	243.117082	59.02
2	2	8407.845588	6221.144614	138	3296.700439	49.659005	yes	5.205257	25865.233800	47.70
3	3	451.000010	266.899987	1	83.540161	3.071000	no	0.305221	63.024630	26.88
4	4	174.927981	140.124004	2	14.233637	1.947000	no	1.063300	67.406408	49.46

While inspecting the data, we noticed that there is an unwanted column "Unname: 0" which is not useful for model. So dropped that column and now the dataset is with **759 firms and 9 attributes.**

```
Shape of the data: (759, 9)
```

:

	sales	capital	patents	randd	employment	sp500	tobinq	value	institutions
0	826.995050	161.603986	10	382.078247	2.306000	no	11.049511	1625.453755	80.27
1	407.753973	122.101012	2	0.000000	1.860000	no	0.844187	243.117082	59.02
2	8407.845588	6221.144614	138	3296.700439	49.659005	yes	5.205257	25865.233800	47.70
3	451.000010	266.899987	1	83.540161	3.071000	no	0.305221	63.024630	26.88
4	174.927981	140.124004	2	14.233637	1.947000	no	1.063300	67.406408	49.46
...	...	...	...	...	...	...	...	...	...
754	1253.900196	708.299935	32	412.936157	22.100002	yes	0.697454	267.119487	33.50
755	171.821025	73.666008	1	0.037735	1.684000	no	NaN	228.475701	46.41
756	202.726967	123.926991	13	74.861099	1.460000	no	5.229723	580.430741	42.25
757	785.687944	138.780992	6	0.621750	2.900000	yes	1.625398	309.938651	61.39
758	22.701999	14.244999	5	18.574360	0.197000	no	2.213070	18.940140	7.50

759 rows × 9 columns

And the datatype of each column in the dataset is

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 759 entries, 0 to 758
Data columns (total 9 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   sales         759 non-null    float64
 1   capital       759 non-null    float64
 2   patents       759 non-null    int64  
 3   randd         759 non-null    float64
 4   employment    759 non-null    float64
 5   sp500         759 non-null    object 
 6   tobinq        738 non-null    float64
 7   value         759 non-null    float64
 8   institutions  759 non-null    float64
dtypes: float64(7), int64(1), object(1)
memory usage: 53.5+ KB
```

We also observed that "**tobinq**" has 738 non-null count out of total entries 759. It is clearly evident that there are missing values. With below details, we can understand that it has 21 null values.

```
sales            0
capital          0
patents          0
randd            0
employment       0
sp500            0
tobinq          21
value            0
institutions     0
dtype: int64
```

Summary statistics is,

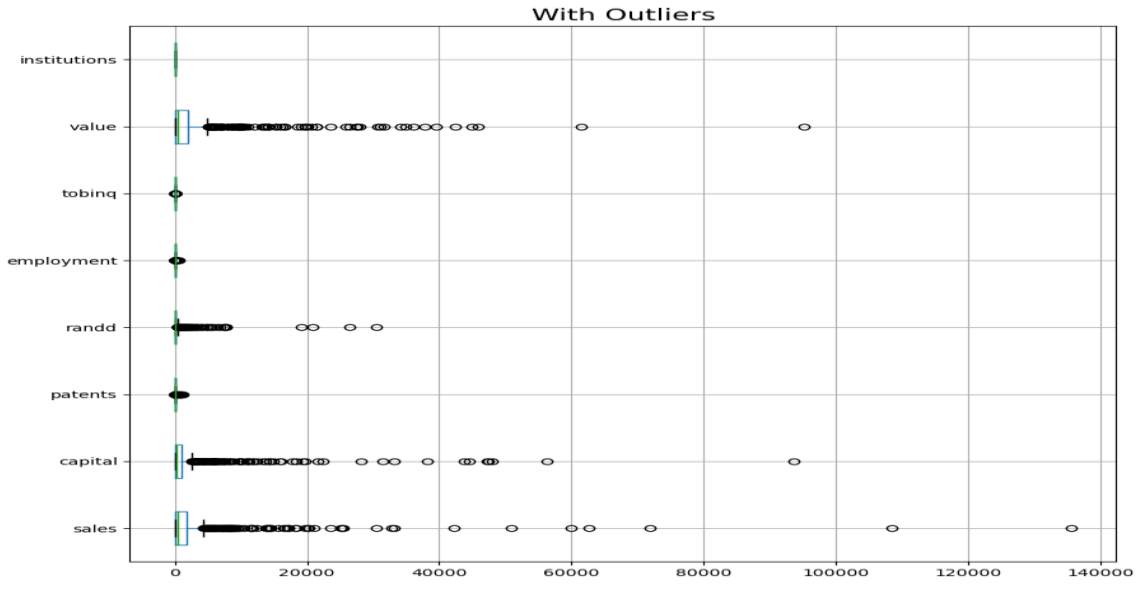
	sales	capital	patents	randd	employment	sp500	tobinq	value	institutions
count	759.000000	759.000000	759.000000	759.000000	759.000000	759	738.000000	759.000000	759.000000
unique	NaN	NaN	NaN	NaN	NaN	2	NaN	NaN	NaN
top	NaN	NaN	NaN	NaN	NaN	no	NaN	NaN	NaN
freq	NaN	NaN	NaN	NaN	NaN	542	NaN	NaN	NaN
mean	2689.705158	1977.747498	25.831357	439.938074	14.164519	NaN	2.794910	2732.734750	43.020540
std	8722.060124	6466.704896	97.259577	2007.397588	43.321443	NaN	3.366591	7071.072362	21.685586
min	0.138000	0.057000	0.000000	0.000000	0.006000	NaN	0.119001	1.971053	0.000000
25%	122.920000	52.650501	1.000000	4.628262	0.927500	NaN	1.018783	103.593946	25.395000
50%	448.577082	202.179023	3.000000	36.864136	2.924000	NaN	1.680303	410.793529	44.110000
75%	1822.547366	1075.790020	11.500000	143.253403	10.050001	NaN	3.139309	2054.160386	60.510000
max	135696.788200	93625.200560	1220.000000	30425.255860	710.799925	NaN	20.000000	95191.591160	90.150000

Summary statistics indicate that most variables are highly skewed, with significant differences between mean and median values.
Some variables (e.g., sales, capital, randd, value) have extreme maximum values, indicating potential outliers.

Ensured that there are no duplicates in the dataset.

```
Number of duplicate rows = 0
```

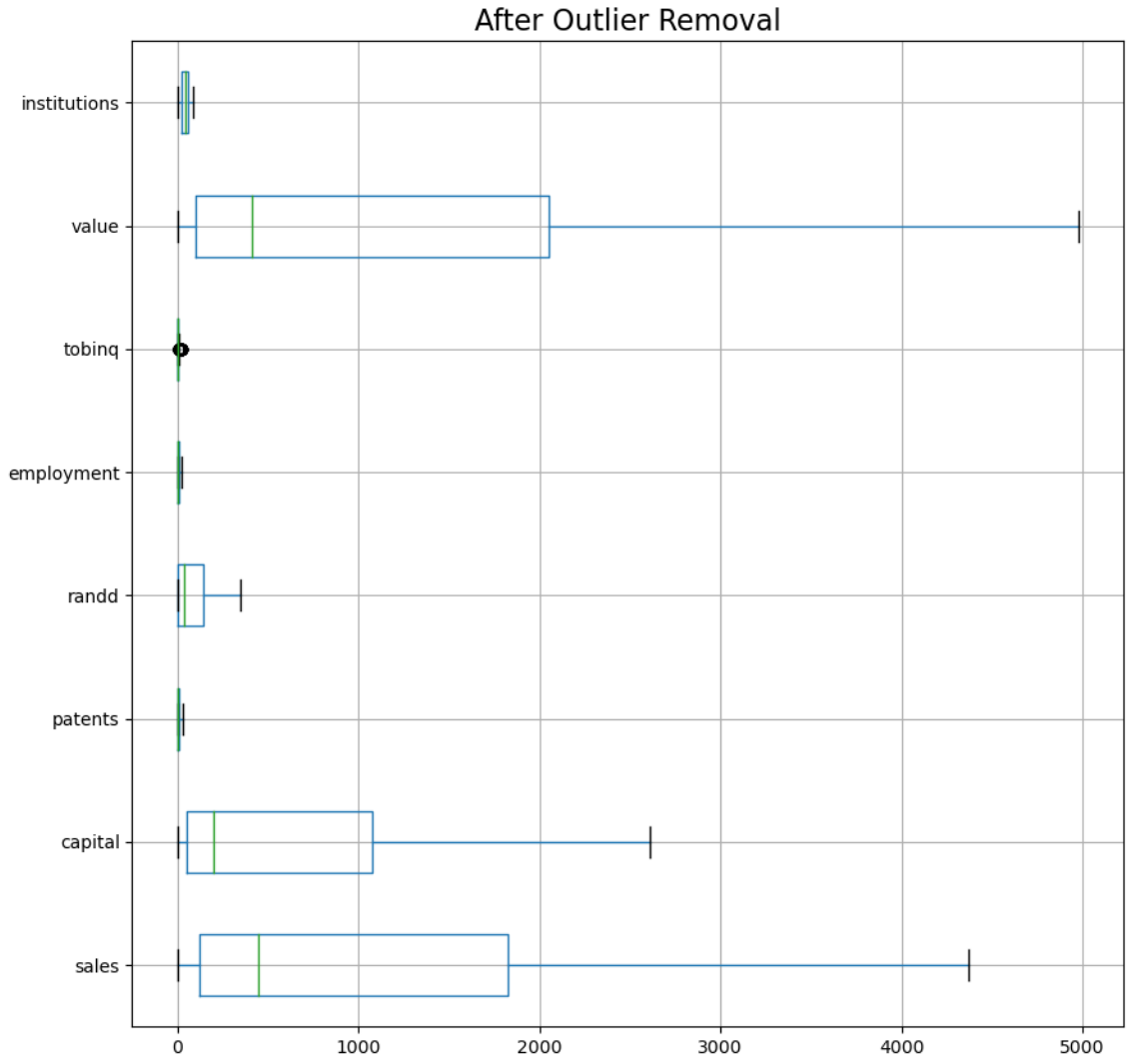
Uni-Variate Analysis: We analyzed each variable individually to understand its distribution and detect anomalies. For this, we have created boxplot for each variable.



The boxplot clearly indicates that several variables exhibit strong skewness and contain extreme outliers.

- Significant outliers are present in the predictors capital, randd and value.
- These outliers are meaningful and likely represent large firms, but they can disproportionately influence statistical models.

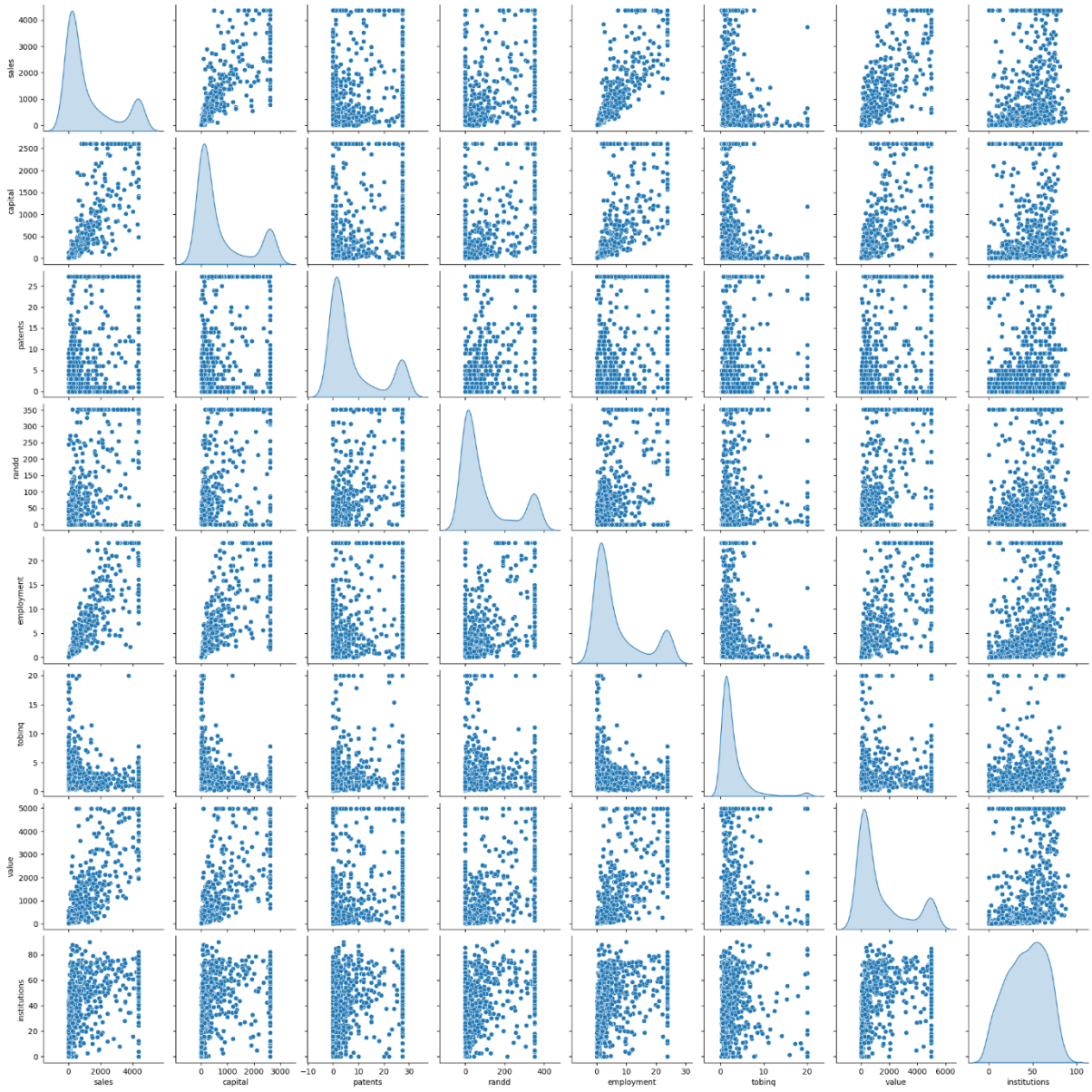
Performed steps to remove these outliers for model accuracy.



Bi-Variate Analysis:

This bivariate analysis highlights the key predictors of firm sales and informs feature selection for subsequent modeling.

A **pairplot** was generated, showing scatterplots for each pair of variables and kernel density plots for the univariate distributions.



capital vs sales

- There is a strong positive linear relationship between capital and sales.
- Firms with higher capital investment tend to have higher sales.

value vs sales

- Similarly, value exhibits a strong positive association with sales.
- This suggests that market value is a good proxy for firm performance.

randd vs sales

- randd shows a moderate positive relationship with sales, though with more variability and outliers.

employment vs sales

- Employment is positively related to sales, but the relationship is weaker compared to capital and value.

patents vs sales

- patents shows a very weak positive relationship with sales.

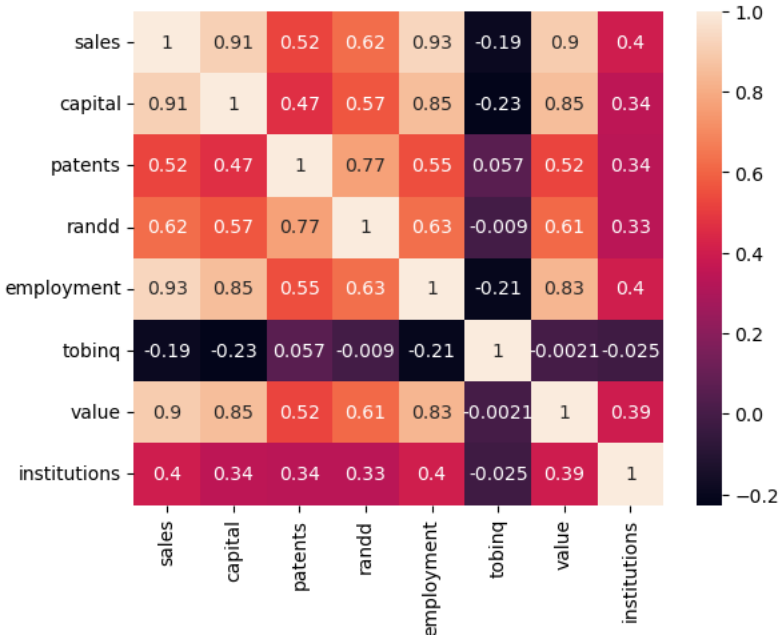
tobinq vs sales

- Little to no clear relationship is evident between tobinq and sales.

institutions vs sales

- Institutional ownership also does not show a strong relationship with sales

Multicollinearity between predictors:



"capital" and "value" appears to be strongly correlated with each other, suggesting potential multicollinearity.
randd and capital also show moderate correlation.

1.2 Impute null values if present. Do you think scaling is necessary in this case?
Solution:

Null Values:
We observed that "**tobinq**" has 738 non-null count out of total entries 759. It is clearly evident that there are missing values. With below details, we can understand that it has 21 null values.

```
sales           0
capital         0
patents         0
randd           0
employment      0
sp500           0
tobinq         21
value           0
institutions    0
dtype: int64
```

Since it is Continuous column, we imputed missing values with mean value. After imputation, we ensured that there are no missing values in any columns.

```
sales           0
capital         0
patents         0
randd           0
employment      0
sp500           0
tobinq          0
value           0
institutions    0
dtype: int64
```

Scaling:

Yes — As per the below stats, scaling is recommended even after removing outliers for better model prediction accuracy.

- The predictors have very different scales (some are thousands, others are single digits).

	sales	capital	patents	randd	employment	sp500	tobinq	value	institutions
count	759.000000	759.000000	759.000000	759.000000	759.000000	759	759.000000	759.000000	759.000000
unique	NaN	NaN	NaN	NaN	NaN	2	NaN	NaN	NaN
top	NaN	NaN	NaN	NaN	NaN	no	NaN	NaN	NaN
freq	NaN	NaN	NaN	NaN	NaN	542	NaN	NaN	NaN
mean	1236.090089	728.715785	7.800395	99.512662	6.925381	NaN	2.794910	1375.431494	43.020540
std	1528.690552	959.394531	9.952684	127.195056	8.184188	NaN	3.319629	1754.489690	21.685586
min	0.138000	0.057000	0.000000	0.000000	0.006000	NaN	0.119001	1.971053	0.000000
25%	122.920000	52.650501	1.000000	4.628262	0.927500	NaN	1.036000	103.593946	25.395000
50%	448.577082	202.179023	3.000000	36.864136	2.924000	NaN	1.741800	410.793529	44.110000
75%	1822.547366	1075.790020	11.500000	143.253403	10.050001	NaN	3.082979	2054.160386	60.510000
max	4371.988416	2610.499299	27.250000	351.191114	23.733752	NaN	20.000000	4980.010044	90.150000

Here variables like sales, capital, and value have huge ranges (hundred-thousands). While others (tobinq, employment) are in single or double digits.

We here applied Z-score method for performing scaling on numerical data.

	sales	capital	patents	randd	employment	tobinq	value	institutions
0	-0.267788	-0.591504	0.221152	1.979986	-0.564800	2.488244	0.142598	1.718839
1	-0.542217	-0.632706	-0.583181	-0.782879	-0.619331	-0.588020	-0.645807	0.738279
2	2.052715	1.962722	1.955496	1.979986	2.055116	0.726568	2.055843	0.215929
3	-0.513909	-0.481679	-0.683723	-0.125658	-0.471265	-0.750485	-0.748521	-0.744789
4	-0.694622	-0.613908	-0.583181	-0.670901	-0.608694	-0.521972	-0.746022	0.297142

Summary statistics after scaling,

	sales	capital	patents	randd	employment	tobinq	value	institutions
count	7.590000e+02	7.590000e+02	7.590000e+02	7.590000e+02	7.590000e+02	7.590000e+02	7.590000e+02	7.590000e+02
mean	2.662195e-17	1.287215e-17	5.046468e-18	-6.041135e-17	5.061096e-17	1.426176e-18	5.704703e-18	1.518329e-16
std	1.000659e+00	1.000659e+00	1.000659e+00	1.000659e+00	1.000659e+00	1.000659e+00	1.000659e+00	1.000659e+00
min	-8.090369e-01	-7.599994e-01	-7.842647e-01	-7.828786e-01	-8.460148e-01	-8.066183e-01	-7.833424e-01	-1.985139e+00
25%	-7.286655e-01	-7.051438e-01	-6.837231e-01	-7.464674e-01	-7.333453e-01	-5.302007e-01	-7.253826e-01	-8.133127e-01
50%	-5.154950e-01	-5.491838e-01	-4.826397e-01	-4.928638e-01	-4.892385e-01	-3.174464e-01	-5.501737e-01	5.027202e-02
75%	3.838867e-01	3.620024e-01	3.719644e-01	3.441139e-01	3.820391e-01	8.683495e-02	3.871077e-01	8.070334e-01
max	2.052715e+00	1.962722e+00	1.955496e+00	1.979986e+00	2.055116e+00	5.186254e+00	2.055843e+00	2.174741e+00

1.3 Encode the data (having string values) for Modelling. Data Split: Split the data into test and train (70:30). Apply Linear regression. Performance Metrics: Check the performance of Predictions on Train and Test sets using Rsquare, RMSE.

Solution: We observed that there is only one column “sp500” that is in Object type.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 759 entries, 0 to 758
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   sales          759 non-null    float64
 1   capital        759 non-null    float64
 2   patents        759 non-null    int64  
 3   randd          759 non-null    float64
 4   employment     759 non-null    float64
 5   sp500          759 non-null    object 
 6   tobinq         738 non-null    float64
 7   value          759 non-null    float64
 8   institutions   759 non-null    float64
dtypes: float64(7), int64(1), object(1)
memory usage: 53.5+ KB
```

To build a model, we need to ensure that all the predictors/independent variables are Continuous (Numerical). Therefore, converting the Categorical datatype using one-hot encoding method.

Check the unique values in sp500

```
SP500 :  2
yes    217
no     542
Name: sp500, dtype: int64
```

After applying one-hot encoding method, the dataset looks like,

	sales	capital	patents	randd	employment	tobinq	value	institutions	sp500_True
0	826.995050	161.603986	10.00	351.191114	2.306000	11.049511	1625.453755	80.27	0
1	407.753973	122.101012	2.00	0.000000	1.860000	0.844187	243.117082	59.02	0
2	4371.988416	2610.499299	27.25	351.191114	23.733752	5.205257	4980.010044	47.70	1
3	451.000010	266.899987	1.00	83.540161	3.071000	0.305221	63.024630	26.88	0
4	174.927981	140.124004	2.00	14.233637	1.947000	1.063300	67.406408	49.46	0

And type of data in “sp500” column changed to “uint8”

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 759 entries, 0 to 758
Data columns (total 9 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   sales          759 non-null    float64
 1   capital        759 non-null    float64
 2   patents        759 non-null    int64  
 3   randd          759 non-null    float64
 4   employment     759 non-null    float64
 5   tobinq         738 non-null    float64
 6   value          759 non-null    float64
 7   institutions   759 non-null    float64
 8   sp500_True     759 non-null    uint8  
dtypes: float64(7), int64(1), uint8(1)
memory usage: 48.3 KB
```

Now it is clear that the dataset has no categorical data. Once again z-scoring is applied on this cleaned dataset. Results are:

	sales	capital	patents	randd	employment	tobinq	value	institutions	sp500_True
0	-0.267788	-0.591504	0.221152	1.979986	-0.564800	2.488244	0.142598	1.718839	-0.632747
1	-0.542217	-0.632706	-0.583181	-0.782879	-0.619331	-0.588020	-0.645807	0.738279	-0.632747
2	2.052715	1.962722	1.955496	1.979986	2.055116	0.726568	2.055843	0.215929	1.580410
3	-0.513909	-0.481679	-0.683723	-0.125658	-0.471265	-0.750485	-0.748521	-0.744789	-0.632747
4	-0.694622	-0.613908	-0.583181	-0.670901	-0.608694	-0.521972	-0.746022	0.297142	-0.632747

Summary statistics are:

	sales	capital	patents	randd	employment	tobinq	value	institutions	sp500_True
count	7.590000e+02	7.590000e+02	7.590000e+02	7.590000e+02	7.590000e+02	7.590000e+02	7.590000e+02	7.590000e+02	7.590000e+02
mean	2.662195e-17	1.287215e-17	5.046468e-18	-6.041135e-17	5.061096e-17	1.426176e-18	5.704703e-18	1.518329e-16	-3.943559e-16
std	1.000659e+00	1.000659e+00	1.000659e+00	1.000659e+00	1.000659e+00	1.000659e+00	1.000659e+00	1.000659e+00	1.000659e+00
min	-8.090369e-01	-7.599994e-01	-7.842647e-01	-7.828786e-01	-8.460148e-01	-8.066183e-01	-7.833424e-01	-1.985139e+00	-6.327472e-01
25%	-7.286655e-01	-7.051438e-01	-6.837231e-01	-7.464674e-01	-7.333453e-01	-5.302007e-01	-7.253826e-01	-8.133127e-01	-6.327472e-01
50%	-5.154950e-01	-5.491838e-01	-4.826397e-01	-4.928638e-01	-4.892385e-01	-3.174464e-01	-5.501737e-01	5.027202e-02	-6.327472e-01
75%	3.838867e-01	3.620024e-01	3.719644e-01	3.441139e-01	3.820391e-01	8.683495e-02	3.871077e-01	8.070334e-01	1.580410e+00
max	2.052715e+00	1.962722e+00	1.955496e+00	1.979986e+00	2.055116e+00	5.186254e+00	2.055843e+00	2.174741e+00	1.580410e+00

Now split the dataset to training and testing data in the ratio of 70:30.

For doing this, we should first need to copy all predictor variables into X dataframe and target variable into y dataframe. Now invoke the LinearRegression function to find the best fit model on training data. Once function is invoked, explore the coefficients for each independent variables.

```
The coefficient for capital is 0.2691366426521997
The coefficient for patents is -0.03064281484816278
The coefficient for randd is 0.049503558795942926
The coefficient for employment is 0.42651667358010986
The coefficient for tobinq is -0.0352208638241435
The coefficient for value is 0.26522612079727126
The coefficient for institutions is 0.0013141653073709318
The coefficient for sp500_True is 0.049984486340708956
```

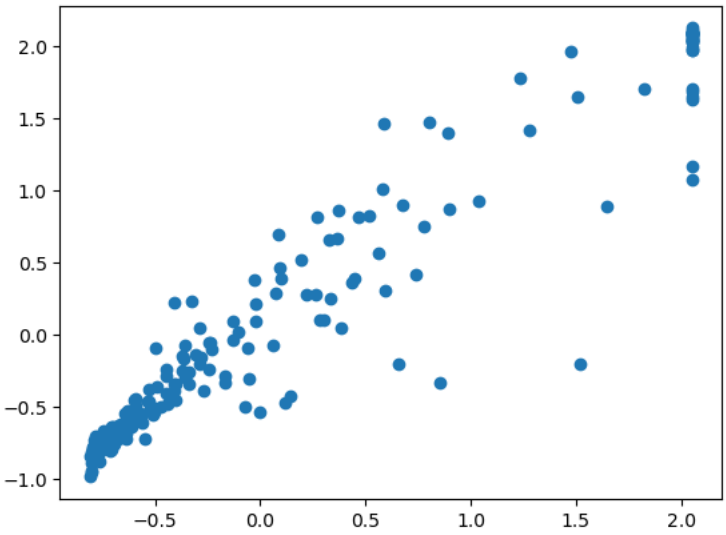
And the intercept for our model is 0.0026850741836639165

And linear regression summary results using statsmodel is,

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                  sales   R-squared:                       0.935
Model:                            OLS   Adj. R-squared:                  0.934
Method:                 Least Squares   F-statistic:                     945.0
Date:                Sun, 13 Jul 2025   Prob (F-statistic):          6.95e-305
Time:                        12:40:17   Log-Likelihood:                -36.621
No. Observations:                 531   AIC:                             91.24
Df Residuals:                     522   BIC:                             129.7
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept      0.0027      0.011      0.236      0.813      -0.020       0.025
capital        0.2691      0.026     10.545      0.000       0.219       0.319
patents       -0.0306      0.018     -1.676      0.094      -0.067       0.005
randd          0.0495      0.019      2.551      0.011       0.011       0.088
employment     0.4265      0.025     16.753      0.000       0.377       0.477
tobinq        -0.0352      0.013     -2.637      0.009      -0.061      -0.009
value          0.2652      0.028      9.381      0.000       0.210       0.321
institutions   0.0013      0.013      0.102      0.919      -0.024       0.027
sp500_True     0.0500      0.020      2.532      0.012       0.011       0.089
==============================================================================
Omnibus:                      184.310   Durbin-Watson:                   1.956
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             1266.767
Skew:                           1.343   Prob(JB):                     8.42e-276
Kurtosis:                      10.074   Cond. No.                         6.59
==============================================================================

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

After predicting the trained model on test data, the graph looks like



R2 value on trained data for model evaluation is 0.9354134324090398

R2 value on test data for model evaluation is 0.9231645685214456

RMSE value on trained data for model evaluation is 0.2592475299509513

RMSE value on test data for model evaluation is 0.2633248972256875

1.4 Inference: Based on these predictions, what are the business insights and recommendations.

Solution:

Model Evaluation Summary:

The R^2 values on both training (93.5%) and test (92.3%) datasets indicate that the model explains over 92% of the variability in sales, which suggests a very good fit without significant overfitting. RMSE values on train and test are very close (~0.26), confirming that the model generalizes well.

Key Takeaways:

- ✓ Employment and Capital are the most important drivers of firm sales.
- ✓ Firm value and R&D investment also contribute positively, though less strongly.
- ✓ Belonging to the S&P500 index adds modest but statistically significant value to sales.
- ✓ Higher Tobin's Q seems to negatively correlate with sales — possibly suggesting overvaluation harms operational performance.
- ✓ Patents and institutional ownership do not show strong or statistically significant effects on sales in this model.

Visual Validation:

- ✓ The scatter plot of predicted vs actual sales shows a strong linear trend with minimal deviation, supporting the model's reliability.
- ✓ Residuals appear randomly scattered, indicating no major model misspecification.

Recommendations for the Investment Firm:

- ✓ Focus investments on firms with higher capital base and larger workforce, as these are the strongest predictors of sales.
- ✓ Firms with strong book value and moderate R&D investments are also good targets.
- ✓ Give preference to firms in the S&P500, as they tend to outperform slightly.
- ✓ Be cautious of firms with excessively high Tobin's Q ratios, as they may be overvalued and underperforming in sales.
- ✓ Do not overemphasize patent count or institutional ownership, as they show little predictive power in this analysis.

Problem 2: Objective

You are hired by the Government to do an analysis of car crashes. You are provided details of car crashes, among which some people survived some didn't. You have to help the government in predicting whether a person will survive or not on the basis of the information given in the dataset to ensure safety measures. Also, find the important factors on the basis of which you made your predictions.

2.1 Data Ingestion: Read the dataset. Do the descriptive statistics and do null value condition check, write an inference on it. Perform Univariate and Bivariate Analysis. Do exploratory data analysis.

Solution:

We are provided with a dataset of **11217 rows and 16 attributes**, including the target variable "**Survived**".

```
Shape of the data: (11217, 16)
```

	Unnamed: 0	dvcat	weight	Survived	airbag	seatbelt	frontal	sex	ageOFocc	yearacc	yearVeh	abcat	occRole	deploy	injSeverity	caseid
0	0	55+	27.078	Not_Survived	none	none	1	m	32	1997	1987.0	unavail	driver	0	4.0	2:13:2
1	1	25-39	89.627	Not_Survived	airbag	belted	0	f	54	1997	1994.0	nodeploy	driver	0	4.0	2:17:1
2	2	55+	27.078	Not_Survived	none	belted	1	m	67	1997	1992.0	unavail	driver	0	4.0	2:79:1
3	3	55+	27.078	Not_Survived	none	belted	1	f	64	1997	1992.0	unavail	pass	0	4.0	2:79:1
4	4	55+	13.374	Not_Survived	none	none	1	m	23	1997	1986.0	unavail	driver	0	4.0	4:58:1

While inspecting the data, we noticed that there is an unwanted column "Unname: 0" which is not useful for model. So dropped that column and now the dataset is with **11217 rows and 15 attributes.**

```
Shape of the data: (11217, 15)
```

	dvcat	weight	Survived	airbag	seatbelt	frontal	sex	ageOFocc	yearacc	yearVeh	abcat	occRole	deploy	injSeverity	caseid
0	55+	27.078	Not_Survived	none	none	1	m	32	1997	1987.0	unavail	driver	0	4.0	2:13:2
1	25-39	89.627	Not_Survived	airbag	belted	0	f	54	1997	1994.0	nodeploy	driver	0	4.0	2:17:1
2	55+	27.078	Not_Survived	none	belted	1	m	67	1997	1992.0	unavail	driver	0	4.0	2:79:1
3	55+	27.078	Not_Survived	none	belted	1	f	64	1997	1992.0	unavail	pass	0	4.0	2:79:1
4	55+	13.374	Not_Survived	none	none	1	m	23	1997	1986.0	unavail	driver	0	4.0	4:58:1
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
11212	25-39	3179.688	survived	none	belted	1	m	17	2002	1985.0	unavail	driver	0	0.0	82:107:1
11213	10-24	71.228	survived	airbag	belted	1	m	54	2002	2002.0	nodeploy	driver	0	2.0	82:108:2
11214	10-24	10.474	survived	airbag	belted	1	f	27	2002	1990.0	deploy	driver	1	3.0	82:110:1
11215	25-39	10.474	survived	airbag	belted	1	f	18	2002	1999.0	deploy	driver	1	0.0	82:110:2
11216	25-39	10.474	survived	airbag	belted	1	m	17	2002	1999.0	deploy	pass	1	0.0	82:110:2

11217 rows × 15 columns

And the datatype of each column in the dataset is

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11217 entries, 0 to 11216
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   dvcat        11217 non-null  object
 1   weight       11217 non-null  float64
 2   Survived     11217 non-null  object
 3   airbag       11217 non-null  object
 4   seatbelt     11217 non-null  object
 5   frontal      11217 non-null  int64
 6   sex          11217 non-null  object
 7   ageOFocc     11217 non-null  int64
 8   yearacc      11217 non-null  int64
 9   yearVeh      11217 non-null  float64
 10  abcat        11217 non-null  object
 11  occRole      11217 non-null  object
 12  deploy       11217 non-null  int64
 13  injSeverity  11140 non-null  float64
 14  caseid       11217 non-null  object
dtypes: float64(3), int64(4), object(8)
memory usage: 1.3+ MB
```

We also observed that "**injSeverity**" has 11140 non-null count out of total entries 11217. It is clearly evident that there are missing values. With below details, we can understand that it has 77 null values.

```
dvcat           0
weight          0
Survived        0
airbag          0
seatbelt        0
frontal         0
sex             0
ageOFocc        0
yearacc         0
yearVeh         0
abcat           0
occRole         0
deploy          0
injSeverity    77
caseid          0
dtype: int64
```

Summary statistics is,

	count	unique	top	freq	mean	std	min	25%	50%	75%	max
dvcat	11217	5	10-24	5414	NaN	NaN	NaN	NaN	NaN	NaN	NaN
weight	11217.0	NaN	NaN	NaN	219.454706	261.963636	0.0	28.292	82.195	324.056	767.702
Survived	11217	2	survived	10037	NaN	NaN	NaN	NaN	NaN	NaN	NaN
airbag	11217	2	airbag	7064	NaN	NaN	NaN	NaN	NaN	NaN	NaN
seatbelt	11217	2	belted	7849	NaN	NaN	NaN	NaN	NaN	NaN	NaN
frontal	11217.0	NaN	NaN	NaN	0.644022	0.47883	0.0	0.0	1.0	1.0	1.0
sex	11217	2	m	6048	NaN	NaN	NaN	NaN	NaN	NaN	NaN
ageOFocc	11217.0	NaN	NaN	NaN	37.40822	18.136557	16.0	22.0	33.0	48.0	87.0
yearacc	11217.0	NaN	NaN	NaN	2001.188553	0.816681	1999.5	2001.0	2001.0	2002.0	2002.0
yearVeh	11217.0	NaN	NaN	NaN	1994.247303	5.405095	1979.0	1991.0	1995.0	1999.0	2003.0
abcat	11217	3	deploy	4365	NaN	NaN	NaN	NaN	NaN	NaN	NaN
occRole	11217	2	driver	8786	NaN	NaN	NaN	NaN	NaN	NaN	NaN
deploy	11217.0	NaN	NaN	NaN	0.389141	0.487577	0.0	0.0	0.0	1.0	1.0
injSeverity	11217.0	NaN	NaN	NaN	1.825583	1.373795	0.0	1.0	2.0	3.0	5.0
caseid	11217	6488	73:100:2	7	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Summary statistics indicates that dataset is well populated with no major missing data.

- Target variable (Survived) is imbalanced, with most cases showing survival.

- There are clear differences in survival related to seatbelt use, airbag presence, and age — these will likely be important predictors.
- Vehicle weights and years have high spread and potential outliers — these may need transformation or scaling.

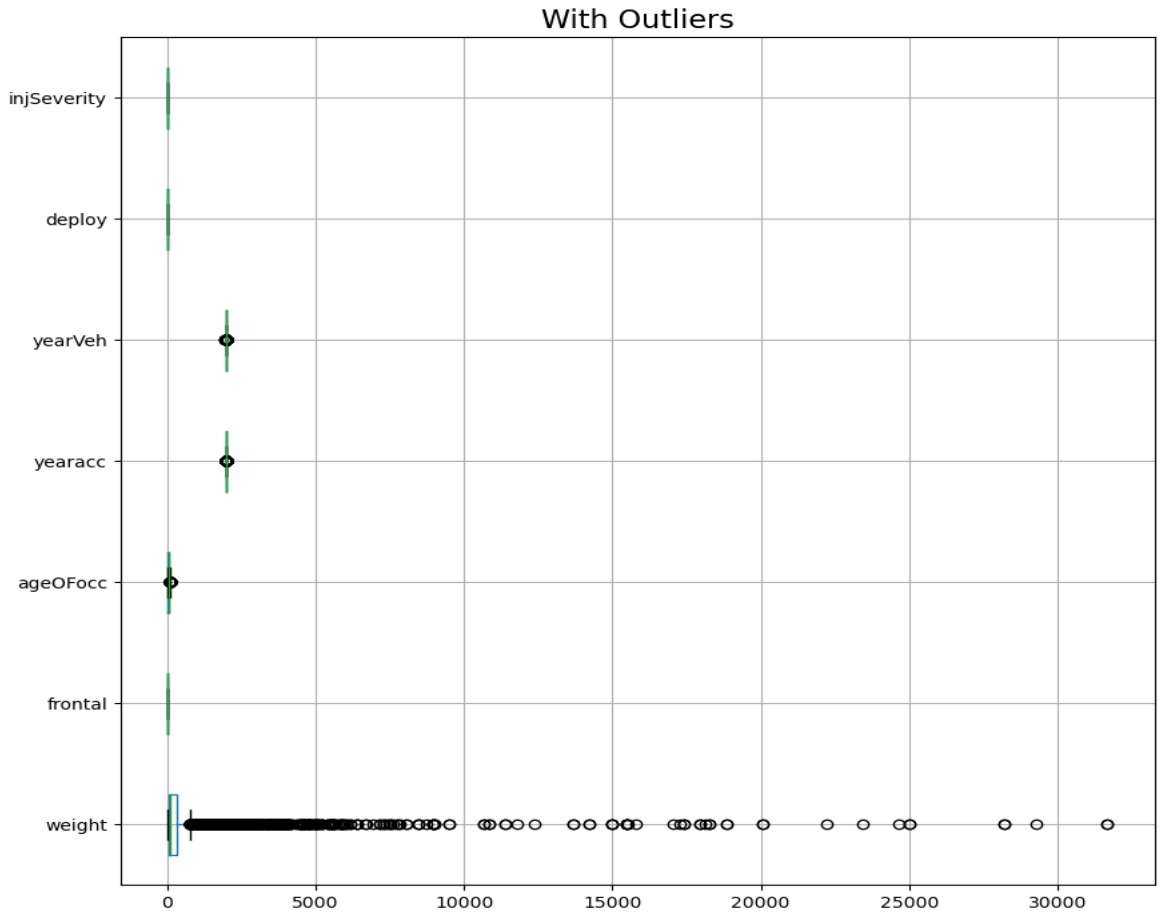
Since **injSeverity** is Continuous column and having NULL, we imputed missing values with mean value. After imputation, we ensured that there are no missing values in any columns.

```
dvcat          0
weight         0
Survived       0
airbag         0
seatbelt       0
frontal        0
sex            0
ageOFocc       0
yearacc        0
yearVeh        0
abcat          0
occRole        0
deploy         0
injSeverity    0
caseid         0
dtype: int64
```

Ensured that there are no duplicates in the dataset.

```
Number of duplicate rows = 0
```

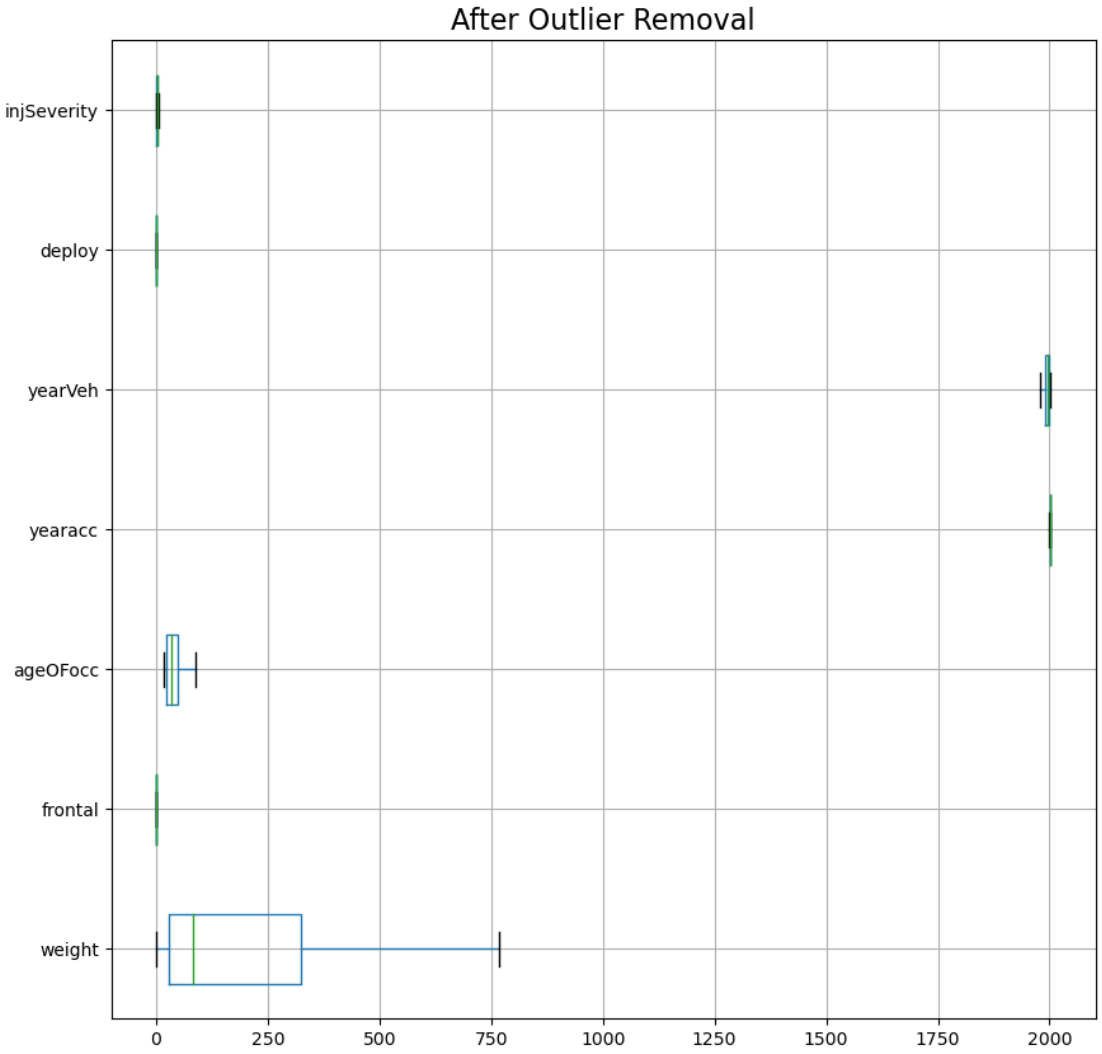
Uni-Variate Analysis: We analyzed each variable individually to understand its distribution and detect anomalies. For this, we have created boxplot for each variable.



Several continuous variables such as weight, ageOFocc, and injSeverity exhibit significant outliers, especially weight, which has extremely high values beyond the upper whisker.

These extreme values could represent rare or unusual crash scenarios (e.g., unusually heavy vehicles) and may skew model predictions

Performed steps to remove these outliers for model accuracy.

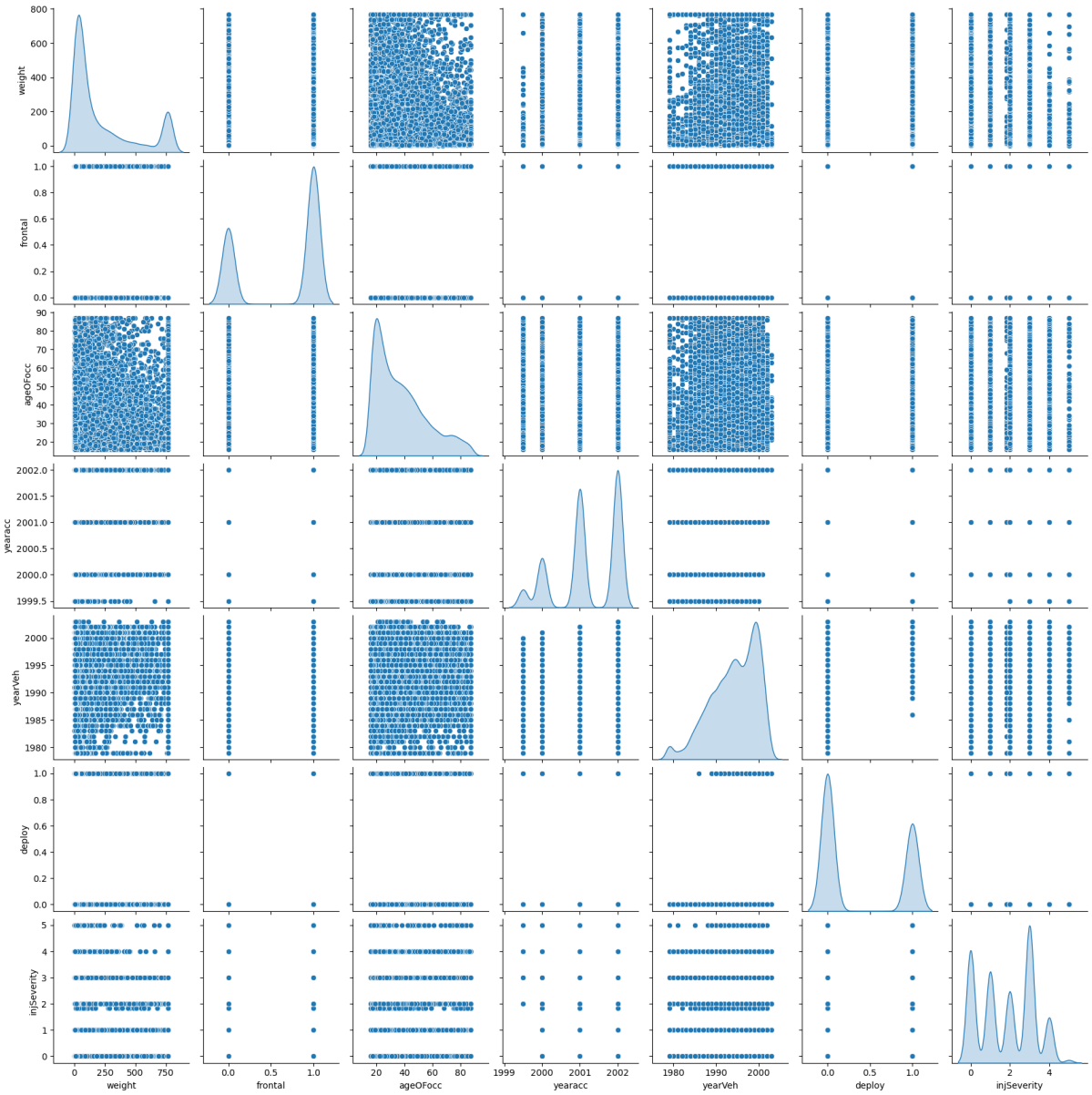


Bi-Variate Analysis:

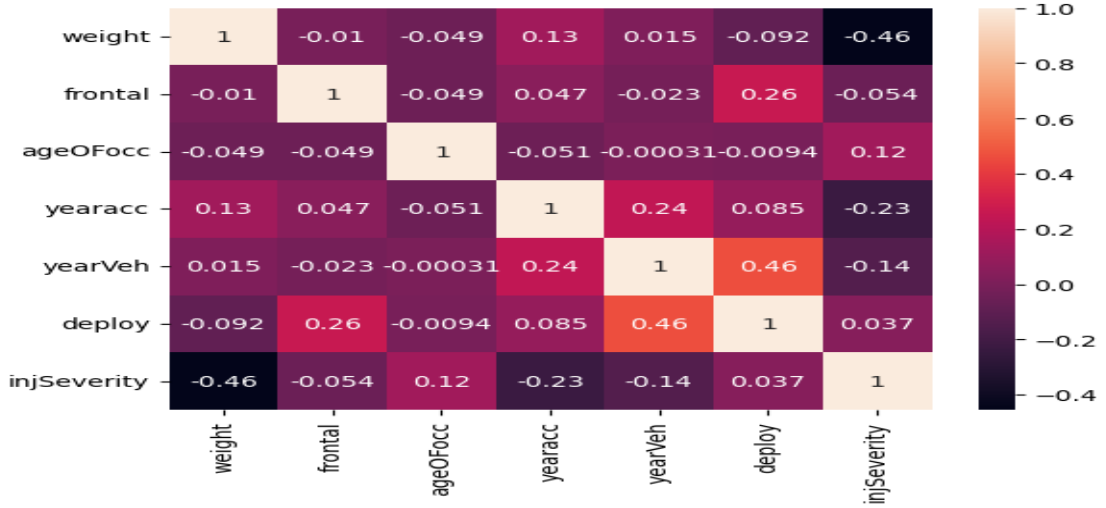
A **pairplot** was generated, showing scatterplots for each pair of variables and kernel density plots for the univariate distributions.

The scatterplots show relationships among variables:

- There is no strong linear relationship visible between most variable pairs.
- Some clustering can be observed in injSeverity vs. deploy, suggesting airbag deployment is somewhat linked to injury severity.
- weight appears right-skewed, with most observations concentrated at the lower end and a few extreme high weights.
- Since strong linear patterns are absent, this suggests that predicting survival or injury severity might depend on non-linear or categorical interactions among features like airbag deployment, seatbelt use, and vehicle age.



Multicollinearity between predictors:



Correlation values are generally weak among variables.

- The strongest positive correlation observed is between deploy and yearVeh (0.46), suggesting newer vehicles have higher deployment rates.
- injSeverity is moderately negatively correlated with weight (-0.46), indicating heavier vehicles tend to have less severe injuries.
- Other variables show very low or negligible correlations.
- Vehicle weight and airbag deployment are potentially important predictors of injury severity and survival, though none of the variables individually exhibit a strong linear relationship to the outcome.

2.2 Encode the data (having string values) for Modelling. Data Split: Split the data into train and test (70:30). Apply Logistic Regression and Linear Discriminant Analysis (LDA).

Solution:

We observed that there is many columns that are in Object type.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 11217 entries, 0 to 11216
Data columns (total 15 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   dvcat        11217 non-null  object
 1   weight       11217 non-null  float64
 2   Survived     11217 non-null  object
 3   airbag       11217 non-null  object
 4   seatbelt     11217 non-null  object
 5   frontal      11217 non-null  int64
 6   sex          11217 non-null  object
 7   ageOFocc     11217 non-null  int64
 8   yearacc      11217 non-null  int64
 9   yearVeh      11217 non-null  float64
 10  abcat        11217 non-null  object
 11  occRole      11217 non-null  object
 12  deploy       11217 non-null  int64
 13  injSeverity  11140 non-null  float64
 14  caseid       11217 non-null  object
dtypes: float64(3), int64(4), object(8)
memory usage: 1.3+ MB
```

To build a model, we need to ensure that all the predictors/independent variables are Continuous (Numerical). Therefore, converting the Categorical datatype using one-hot encoding method.

Check the unique values in all categorical columns

```
DVCAT :  5
1-9km/h     282
55+         809
40-54      1344
25-39      3368
10-24      5414
Name: dvcat, dtype: int64


SURVIVED :  2
Not_Survived     1180
survived        10037
Name: Survived, dtype: int64


AIRBAG :  2
none      4153
airbag    7064
Name: airbag, dtype: int64


SEATBELT :  2
none      3368
belted    7849
Name: seatbelt, dtype: int64


SEX :  2
f    5169
m    6048
Name: sex, dtype: int64


ABCAT :  3
nodeploy    2699
unavail     4153
deploy      4365
Name: abcat, dtype: int64


OCCROLE :  2
pass     2431
driver   8786
Name: occRole, dtype: int64


CASEID :  6488
2:95:1      1
45:49:1     1
45:88:1     1
45:42:2     1
45:34:1     1
           ..
74:58:1     6
49:106:1    6
75:84:2     6
74:74:2     6
73:100:2    7
Name: caseid, Length: 6488, dtype: int64
```

As we noticed that `caseid` is just an identifier, so drop it before encoding. After applying one-hot encoding method, the dataset looks like,

```
   weight  frontal  ageOFocc  yearacc  yearVeh  deploy  injSeverity  \
0  27.078      1.0      32.0   1999.5   1987.0     0.0          4.0
1  89.627      0.0      54.0   1999.5   1994.0     0.0          4.0
2  27.078      1.0      67.0   1999.5   1992.0     0.0          4.0
3  27.078      1.0      64.0   1999.5   1992.0     0.0          4.0
4  13.374      1.0      23.0   1999.5   1986.0     0.0          4.0

   dvcat_10-24  dvcat_25-39  dvcat_40-54  dvcat_55+  Survived_survived  \
0            0            0            0          1                  0
1            0            1            0          0                  0
2            0            0            0          1                  0
3            0            0            0          1                  0
4            0            0            0          1                  0

   airbag_none  seatbelt_none  sex_m  abcat_nodeploy  abcat_unavail  \
0            1              1      1               0              1
1            0              0      0               1              0
2            1              0      1               0              1
3            1              0      0               0              1
4            1              1      1               0              1

   occRole_pass
0             0
1             0
2             0
3             1
4             0
```

Summary statistics are:

	count	mean	std	min	25%	50%	75%	max
weight	11217.0	219.454706	261.963636	0.0	28.292	82.195	324.056	767.702
frontal	11217.0	0.644022	0.478830	0.0	0.000	1.000	1.000	1.000
ageOFocc	11217.0	37.408220	18.136557	16.0	22.000	33.000	48.000	87.000
yearacc	11217.0	2001.188553	0.816681	1999.5	2001.000	2001.000	2002.000	2002.000
yearVeh	11217.0	1994.247303	5.405095	1979.0	1991.000	1995.000	1999.000	2003.000
deploy	11217.0	0.389141	0.487577	0.0	0.000	0.000	1.000	1.000
injSeverity	11217.0	1.825583	1.373795	0.0	1.000	2.000	3.000	5.000
dvcat_10-24	11217.0	0.482660	0.499722	0.0	0.000	0.000	1.000	1.000
dvcat_25-39	11217.0	0.300259	0.458391	0.0	0.000	0.000	1.000	1.000
dvcat_40-54	11217.0	0.119818	0.324763	0.0	0.000	0.000	0.000	1.000
dvcat_55+	11217.0	0.072123	0.258702	0.0	0.000	0.000	0.000	1.000
Survived_survived	11217.0	0.894803	0.306821	0.0	1.000	1.000	1.000	1.000
airbag_none	11217.0	0.370242	0.482891	0.0	0.000	0.000	1.000	1.000
seatbelt_none	11217.0	0.300259	0.458391	0.0	0.000	0.000	1.000	1.000
sex_m	11217.0	0.539182	0.498485	0.0	0.000	1.000	1.000	1.000
abcat_nodeploy	11217.0	0.240617	0.427477	0.0	0.000	0.000	0.000	1.000
abcat_unavail	11217.0	0.370242	0.482891	0.0	0.000	0.000	1.000	1.000
occRole_pass	11217.0	0.216725	0.412032	0.0	0.000	0.000	0.000	1.000

Now split the dataset to training and testing data in the ratio of 70:30.

We trained a **Logistic Regression model**, which is a widely used classification algorithm that estimates the probability of survival. The model was trained on X_train, y_train. Predictions were generated for both train and test sets. We also trained a **Linear Discriminant Analysis model (LDA)**, which is another classification technique that works well when classes are normally distributed. Similar to Logistic Regression, we trained the LDA model on the training set and predicted on both train and test sets.
We generated predictions for both models:

- On the **training data** — to check how well the model learned.
- On the **test data** — to assess how well the model performs on unseen data.

Predictions were stored for further evaluation.

2.3 Performance Metrics: Check the performance of Predictions on Train and Test sets using Accuracy, Confusion Matrix, Plot ROC curve and get ROC_AUC score for each model. Compare both the models and write inferences, which model is best/optimized.

Solution:

We evaluated two models — **Logistic Regression** and **Linear Discriminant Analysis (LDA)** — on both training and test data using Accuracy, Confusion Matrix, Precision/Recall/F1, and ROC-AUC score.

Logistic Regression Results:

Metric	Training Data	Test Data
Accuracy	98.2%	97.9%
ROC-AUC	0.988	0.988
Precision (Class 1)	99%	99%
Recall (Class 1)	99%	99%
F1-score (Class 1)	99%	99%
Precision (Class 0)	93%	92%
Recall (Class 0)	89%	88%

- **Inference:**
  - Logistic Regression shows excellent predictive power with very high accuracy and ROC-AUC on both training and test sets.
  - No signs of overfitting (training and test metrics are almost equal).
  - It predicts the majority class (survived) extremely well, but slightly lower performance on minority class (not survived).

Linear Discriminant Analysis (LDA) Results

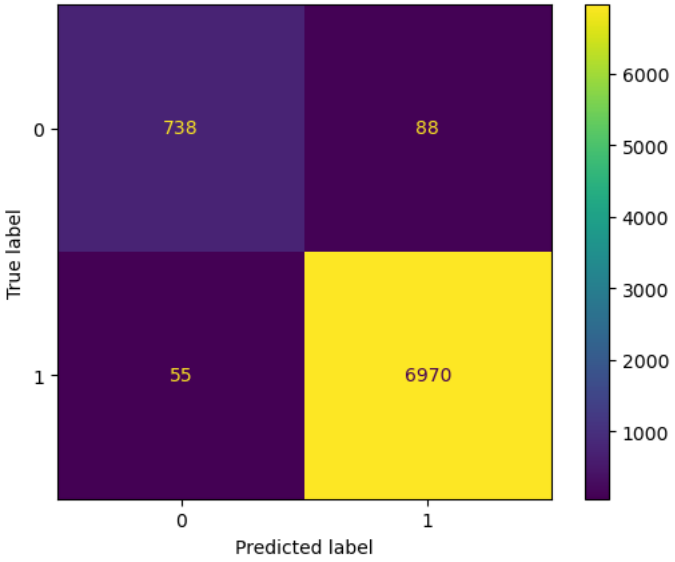
Metric	Training Data	Test Data
Accuracy	96%	95%
ROC-AUC	0.969	0.971
Precision (Class 1)	97%	97%
Recall (Class 1)	98%	97%
F1-score (Class 1)	98%	97%
Precision (Class 0)	80%	76%
Recall (Class 0)	77%	79%

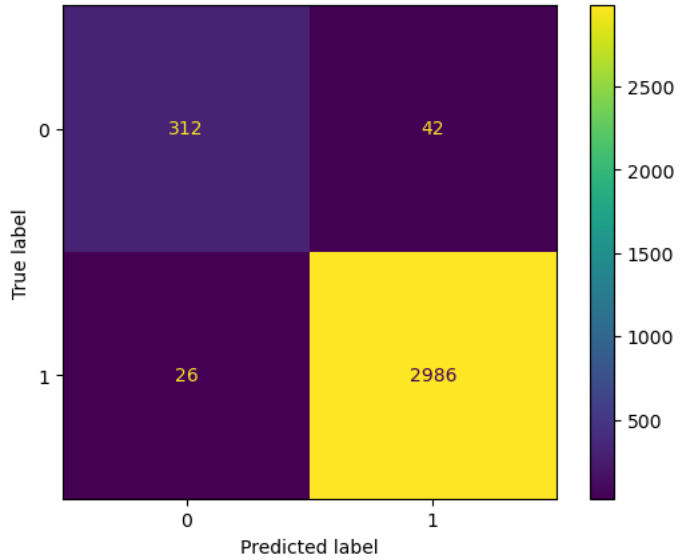
- **Inference:**
  - LDA also performs well but is slightly less accurate than Logistic Regression on both training and test sets.

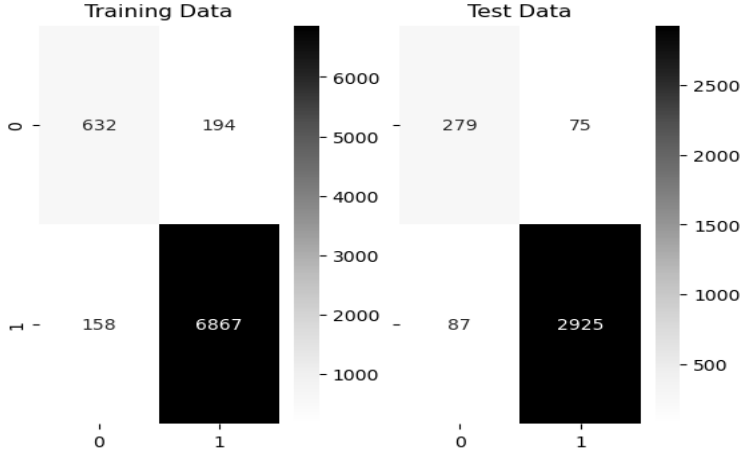
- ROC-AUC is very good but slightly lower than Logistic Regression.
- The model struggles more on the minority class (not survived) compared to Logistic Regression.

Confusion Matrix Insights

- Both models have higher false negatives and false positives for the minority class (Not Survived).
- Logistic Regression has a better balance between precision and recall for both classes.



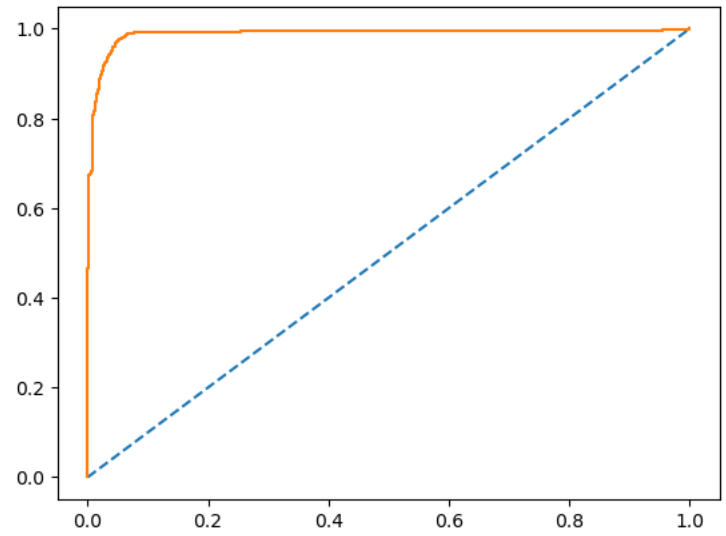




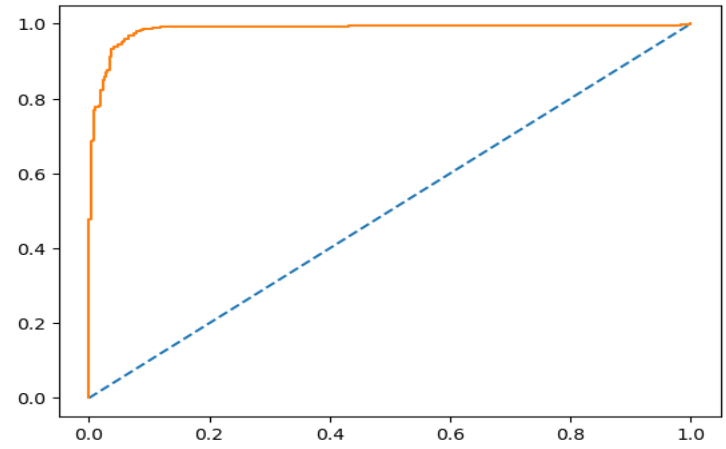
ROC Curve Comparison

- Both models have ROC curves close to the top-left corner, indicating excellent performance.
- Logistic Regression achieves a slightly higher area under the curve (AUC ~0.988) compared to LDA (AUC ~0.971).

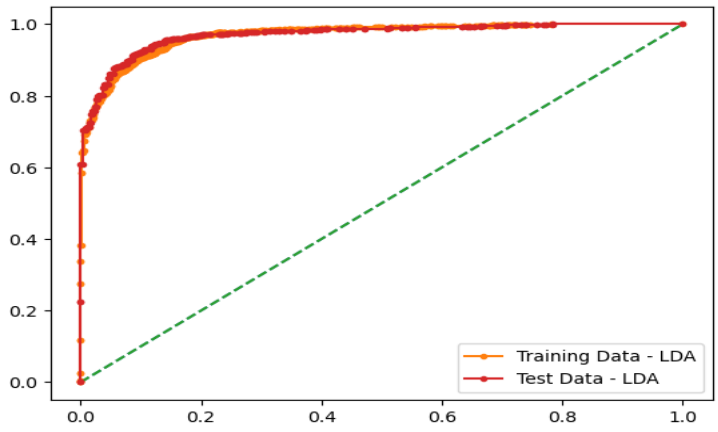
```
AUC: 0.988
```




```
AUC: 0.988
```




```
AUC for the Training Data - LDA: 0.969
AUC for the Test Data - LDA: 0.971
```



Final Recommendation

Model	Strengths	Weaknesses
Logistic Regression	Highest accuracy (98%), best ROC-AUC (0.988), balanced on both classes, no overfitting	Slightly lower recall for minority class
LDA	Good performance, simpler assumptions, AUC ~0.971	Lower precision and recall on minority class, slightly less accurate

Recommended Model: Logistic Regression

- Performs better overall with higher accuracy, ROC-AUC, and better balance across classes.
- Generalizes well from training to test data (no overfitting observed).

2.4 Inference: Based on these predictions, what are the insights and recommendations. Please explain and summarise the various steps performed in this project. There should be proper business interpretation and actionable insights present.

Solution:

- Logistic Regression consistently outperformed LDA in all metrics.
- No signs of overfitting in either model.
- Logistic Regression has better recall for the minority class (Not Survived), which is critical in this context.

<u>Business Interpretation & Insights</u>

- Wearing a seatbelt and having an airbag deployed are strongly associated with higher survival rates.
- Older vehicles and higher vehicle weight are associated with a higher risk of not surviving.
- Younger age groups have higher survival chances.
- The Logistic Regression model is the optimal choice for predicting survival with ~98% accuracy and excellent discriminatory power (AUC ~0.988).
- The insights derived allow the Government to design targeted interventions, improve public safety campaigns, and inform policy-making for reducing fatalities in car crashes.