Bharati Vidyapeeth
Deemed to be University
Department of Engineering and Technology
Plot no. KC-1, Sector 3, Kharghar, Navi Mumbai-410210

Subject: Deep Learning | Experiment No - 01 (3rd YEAR CSE-AIML 2023-2024)

Roll No: 11	Name: Kamran Khan
Class: CSE-AIML	Batch: B1
PRN: 2143110133	Date of Experiment: __ / __ / 2024
Marks (Out of 25):	Date of Submission: __ / __ / 2024

Aim: Implement Linear Regression

Theory:

Machine Learning is a branch of Artificial intelligence that focuses on the development of algorithms and statistical models that can learn from and make predictions on data. Linear regression is also a type of machine-learning algorithm more specifically a supervised machine-learning algorithm that learns from the labelled datasets and maps the data points to the most optimized linear functions. which can be used for prediction on new datasets.

Linear regression is a type of supervised machine learning algorithm that computes the linear relationship between a dependent variable and one or more independent features. When the number of the independent feature, is 1 then it is known as Univariate Linear regression, and in the case of more than one feature, it is known as multivariate linear regression.

In [59]:
```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
```

In [60]:
```
# Load the data from the CSV file
data = pd.read_csv('Cars.csv')

data.head()
```

Out[60]:

	HP	MPG	VOL	SP	WT
0	49	53.700681	89	104.185353	28.762059
1	55	50.013401	92	105.461264	30.466833
2	55	50.013401	92	105.461264	30.193597
3	70	45.696322	92	113.461264	30.632114
4	53	50.504232	92	104.461264	29.889149

In [61]:
```
# Check for missing values
data.isnull().sum()
```

Out[61]:
```
HP      0
MPG     0
VOL     0
SP      0
WT      0
dtype: int64
```

In [62]:
```
# Descriptive statistics
data.describe()
```

Out[62]:

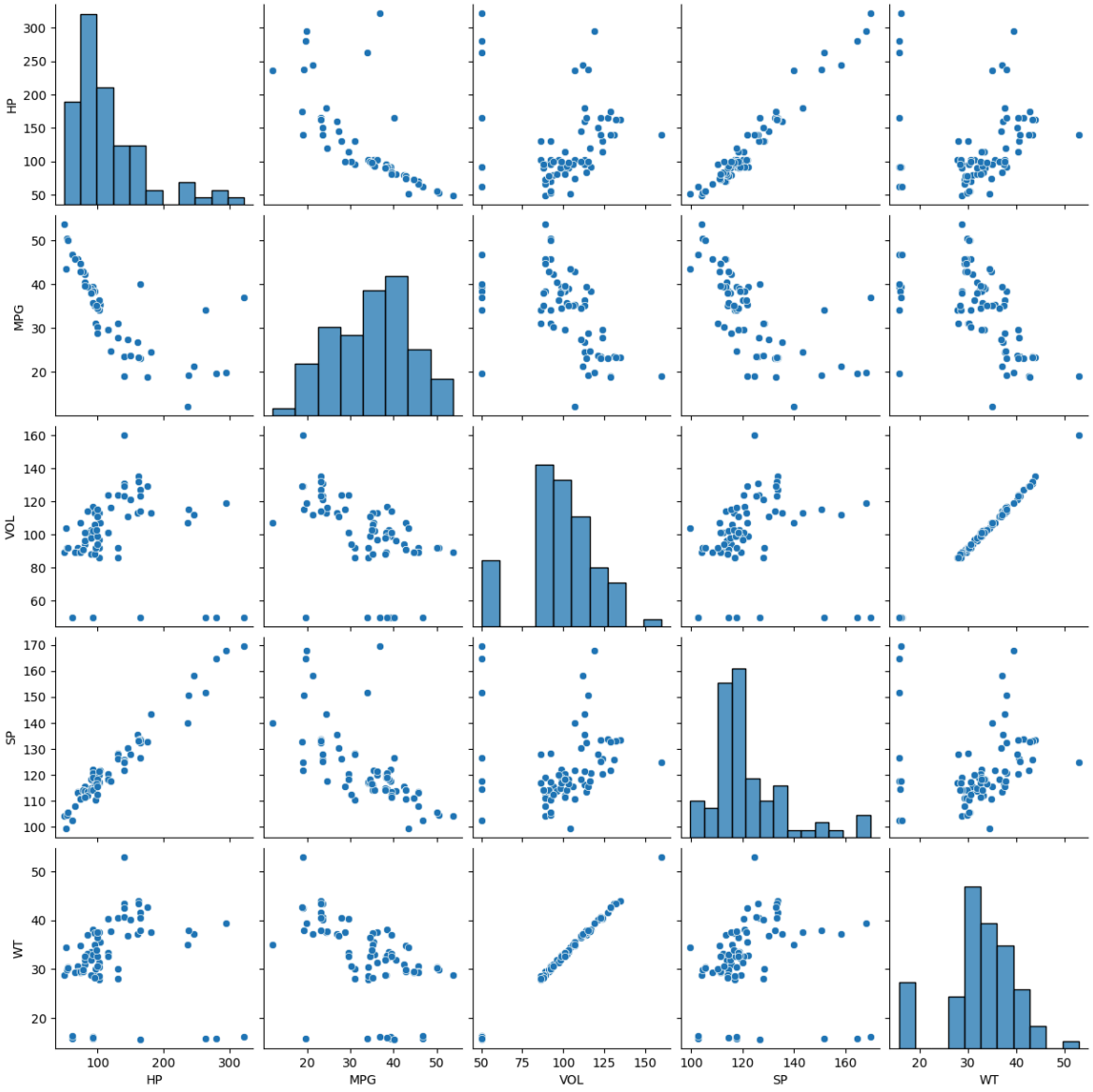
	HP	MPG	VOL	SP	WT
count	81.000000	81.000000	81.000000	81.000000	81.000000
mean	117.469136	34.422076	98.765432	121.540272	32.412577
std	57.113502	9.131445	22.301497	14.181432	7.492813
min	49.000000	12.101263	50.000000	99.564907	15.712859
25%	84.000000	27.856252	89.000000	113.829145	29.591768
50%	100.000000	35.152727	101.000000	118.208698	32.734518
75%	140.000000	39.531633	113.000000	126.404312	37.392524
max	322.000000	53.700681	160.000000	169.598513	52.997752

In [63]:
```
# Correlation matrix
corr_matrix = data.corr()
corr_matrix
```

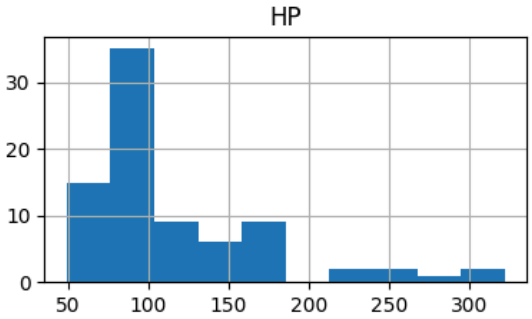
Out[63]:

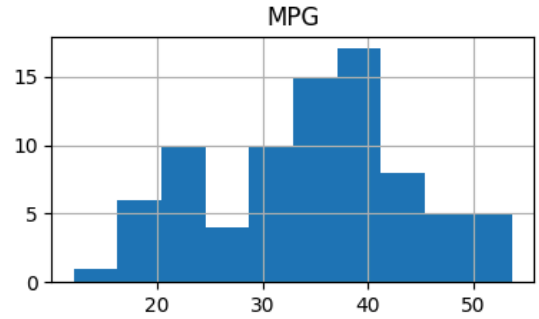
	HP	MPG	VOL	SP	WT
HP	1.000000	-0.725038	0.077459	0.973848	0.076513
MPG	-0.725038	1.000000	-0.529057	-0.687125	-0.526759
VOL	0.077459	-0.529057	1.000000	0.102170	0.999203
SP	0.973848	-0.687125	0.102170	1.000000	0.102439
WT	0.076513	-0.526759	0.999203	0.102439	1.000000

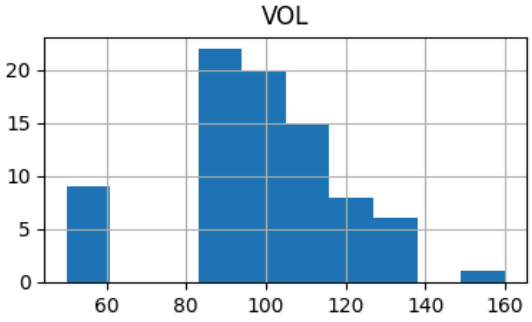
In [64]:
```
# Pairwise scatter plots
sns.pairplot(data)
plt.show()
```

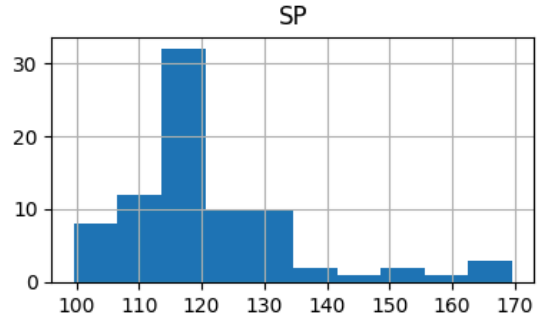


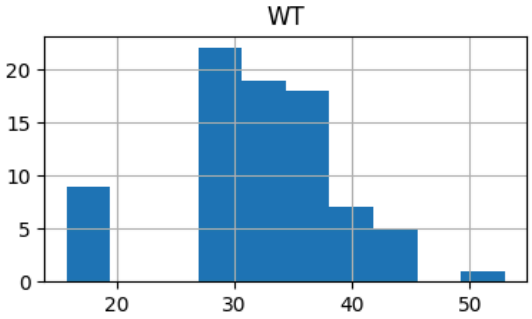
In [65]:
```
# Histograms for each feature
data.hist(figsize=(10, 8))
plt.show()
```



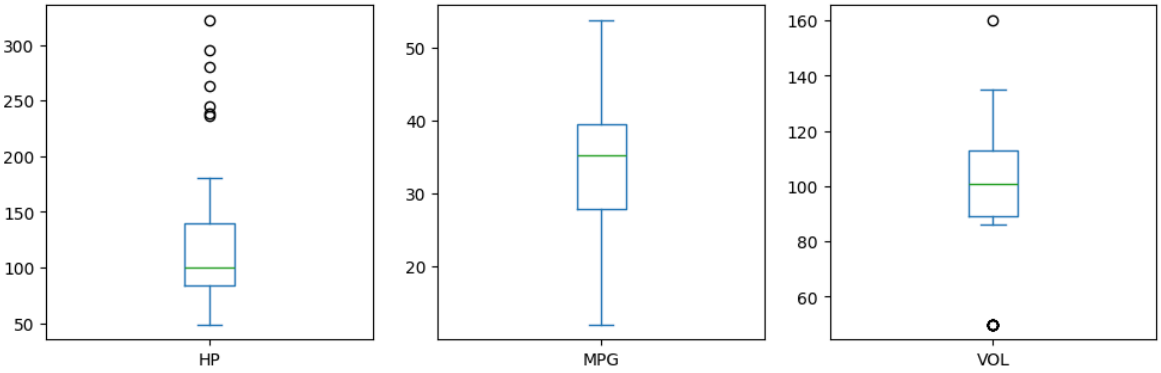


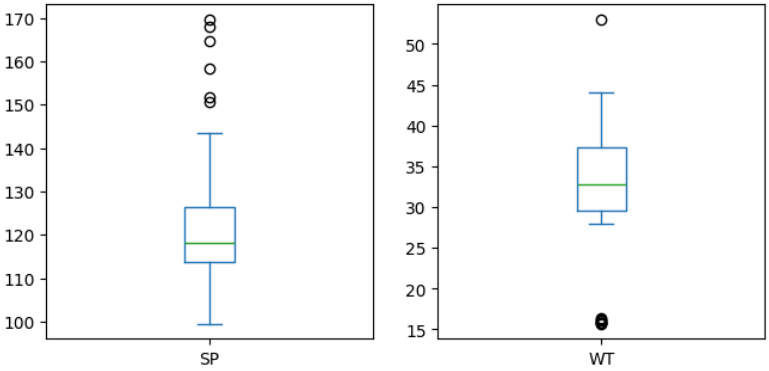






In [66]:
```
# Box plots for each feature
data.plot(kind='box', subplots=True, layout=(2, 3), figsize=(12, 8), sharex=Fals
plt.show()
```





In [67]:
```
# Drop any rows with missing values
data = data.dropna()
```

In [68]:
```
scaler = StandardScaler()
data[['HP', 'MPG', 'VOL', 'SP', 'WT']] = scaler.fit_transform(data[['HP', 'MPG',
```

In [69]:
```
# Assuming we want to predict 'WT' based on other features
X = data[['HP', 'MPG', 'VOL', 'SP']]  # Features
y = data['WT']  # Target variable
```

In [70]:
```
# Split the data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_
```

In [71]:
```
# Create a linear regression model
model = LinearRegression()
```

In [72]:
```
# Fit the model to the training data
model.fit(X_train, y_train)
```

Out[72]:

▾ LinearRegression

LinearRegression()

In [73]:
```
# Get the coefficients
print('Coefficients:', model.coef_)

# Get the intercept
print('Intercept:', model.intercept_)
```

```
Coefficients: [-0.02513822  0.00567949  1.00025576  0.03000924]
Intercept: 0.0015547195789852655
```

In [74]:
```
# Make predictions on the test set
y_pred = model.predict(X_test)
```

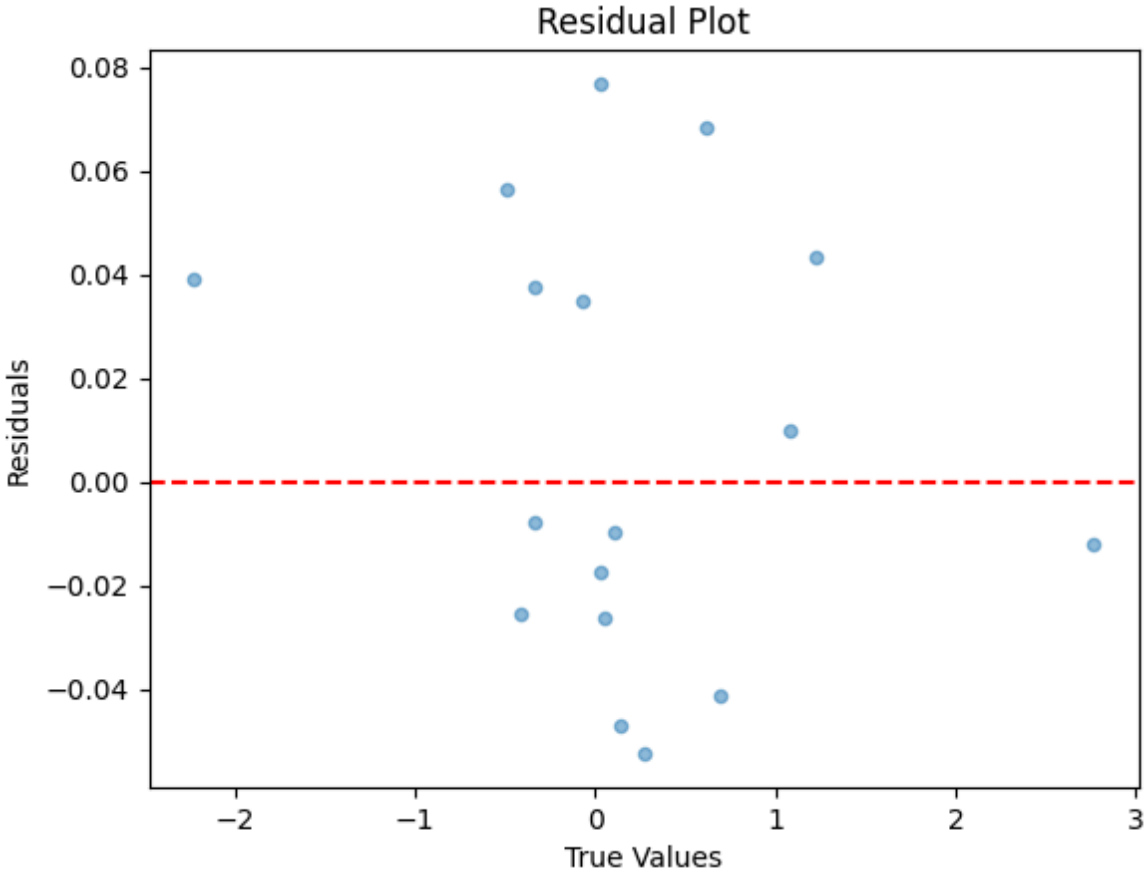
In [75]:
```
# Evaluate the model
from sklearn.metrics import mean_squared_error, r2_score

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print('Mean Squared Error:', mse)
print('R-squared:', r2)
```

```
Mean Squared Error: 0.0016730269289721442
R-squared: 0.9982571329976777
```

In [76]:
```
# Residual plot
plt.scatter(y_test, y_pred - y_test, alpha=0.5, s=20)
plt.axhline(y=0, color='r', linestyle='--')
plt.xlabel('True Values')
plt.ylabel('Residuals')
plt.title('Residual Plot')
plt.show()
```



Conclusion: Hence, we have successfully implemented Linear Regression

Signature of Lab Incharge
(Prof. Poonam Kapse)