Analysis of Combined Cycle Power Plant Dataset

This analysis is based on the Power dataset which can be found at the website: https://archive.ics.uci.edu/ml/datasets/Combined+Cycle+Power+Plant.

This dataset contains 9568 records collected from a power plant facility. In this exercise, I will explore how various predictors can affect the Electrical Power Output:

Predictors:

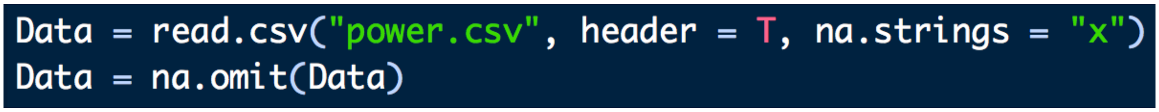
- **Ambient Temperature (AT)**: measured in Degrees Celsius
- **Vacuum (V)**: exhaust stream pressure measured in cmHG
- **Atmospheric Pressure (AP)**: measured in Minibars
- **Relative Humidity (RH)**: measured in a percentage

Response Variable:

- **Electrical Power Output (PE)**: target variable measured in Mega Watt.

1. Descriptive Analytics on various predictors

Firstly, to load data into a variable "Data" and discard entries with missing values, the following R codes are executed:

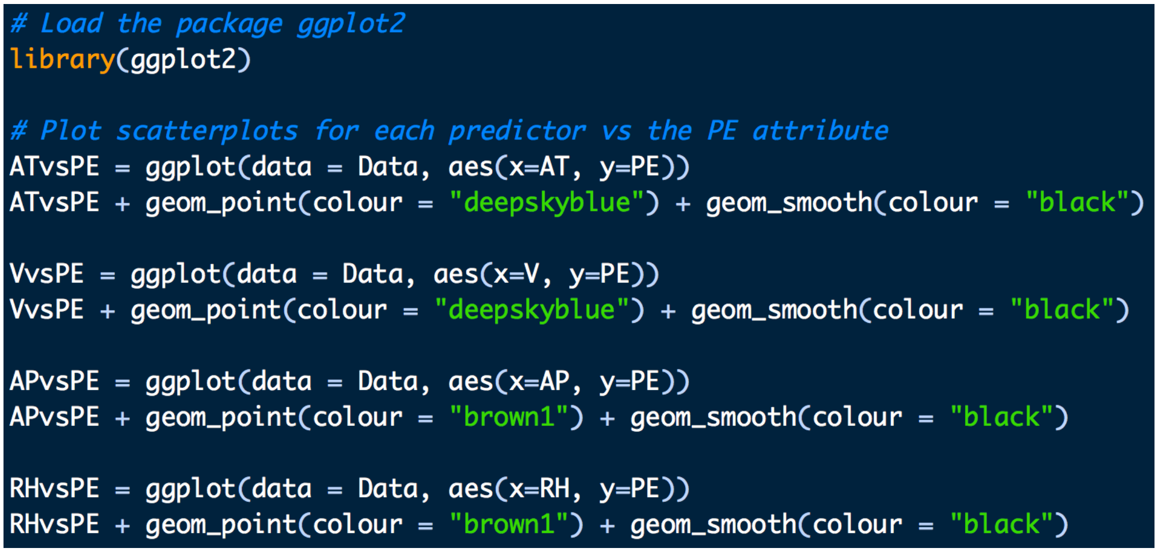


```
Data = read.csv("power.csv", header = T, na.strings = "x")
Data = na.omit(Data)
```

The first line reads the "power.csv" file, changes the "x" characters used to indicate missing values into NA values and stores the result into the "Data" variable. The second line removes rows with such NA values and overwrites the "Data" variable.

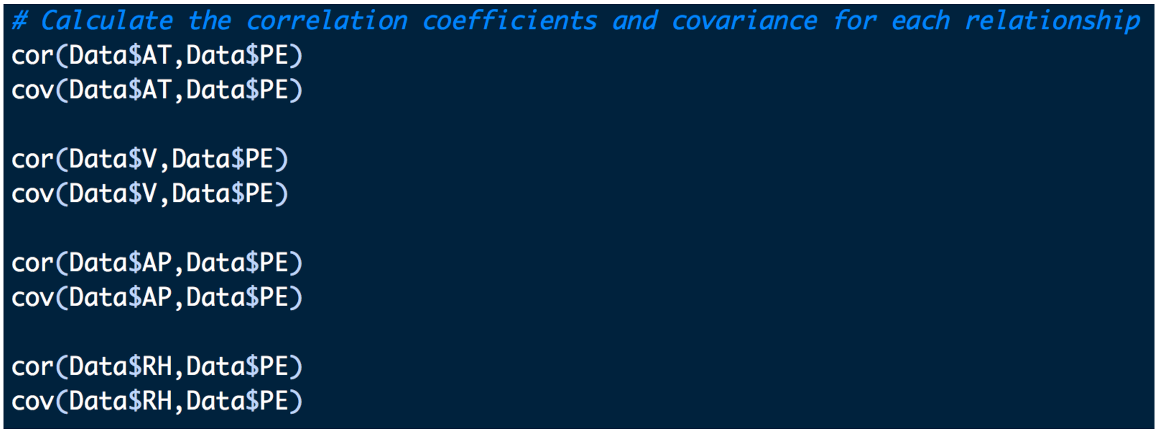
Next, the package "ggplot2" is loaded to perform descriptive analytics with charts. After which, various scatterplots are plotted to show relationships between the predictor attributes (AT, V, AP, RH) and the PE attribute individually.
For aesthetics, a "geom_smooth" layer is added to show trends in the scatterplot which may be difficult to see if there is no clear relationship between the attributes. After viewing the initial plots, the scatterplots are replotted with different colour schemes to show a positive (red) or negative (blue) relationship.
The correlation coefficients (cor) and covariance (cov) values for each relationship are also computed. The following screenshot shows the R codes used:

Plotting the scatterplots



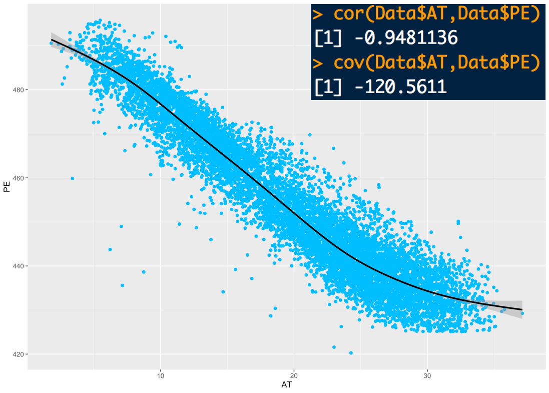
```
# Load the package ggplot2
library(ggplot2)

# Plot scatterplots for each predictor vs the PE attribute
ATvsPE = ggplot(data = Data, aes(x=AT, y=PE))
ATvsPE + geom_point(colour = "deepskyblue") + geom_smooth(colour = "black")

VvsPE = ggplot(data = Data, aes(x=V, y=PE))
VvsPE + geom_point(colour = "deepskyblue") + geom_smooth(colour = "black")

APvsPE = ggplot(data = Data, aes(x=AP, y=PE))
APvsPE + geom_point(colour = "brown1") + geom_smooth(colour = "black")

RHvsPE = ggplot(data = Data, aes(x=RH, y=PE))
RHvsPE + geom_point(colour = "brown1") + geom_smooth(colour = "black")
```

Calculating cor and cov for each relationship



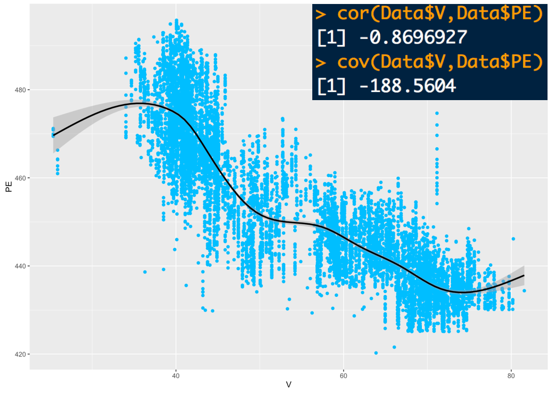
```
# Calculate the correlation coefficients and covariance for each relationship
cor(Data$AT,Data$PE)
cov(Data$AT,Data$PE)

cor(Data$V,Data$PE)
cov(Data$V,Data$PE)

cor(Data$AP,Data$PE)
cov(Data$AP,Data$PE)

cor(Data$RH,Data$PE)
cov(Data$RH,Data$PE)
```

This produces the following plots (with the corresponding cor and cov values):



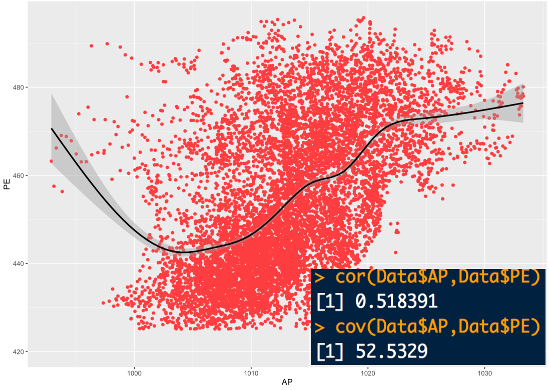
```
> cor(Data$AT,Data$PE)
[1] -0.9481136
> cov(Data$AT,Data$PE)
[1] -120.5611
```

Figure 1: Ambient Temperature (AT) vs Electrical Power Output (PE)



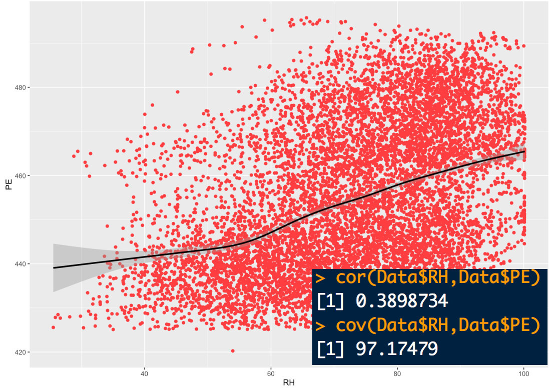
```
> cor(Data$V,Data$PE)
[1] -0.8696927
> cov(Data$V,Data$PE)
[1] -188.5604
```

Figure 2: Vacuum (V) vs Electrical Power Output (PE)



```
> cor(Data$AP,Data$PE)
[1] 0.518391
> cov(Data$AP,Data$PE)
[1] 52.5329
```

Figure 3: Atmospheric Pressure (AP) vs Electrical Power Output (PE)



```
> cor(Data$RH,Data$PE)
[1] 0.3898734
> cov(Data$RH,Data$PE)
[1] 97.17479
```

Figure 4: Relative Humidity (RH) vs Electrical Power Output (PE)

Analysis:

Figure 1 shows that with an increase in Ambient temperature, Electrical Power Output decreases. There is also minimal vertical spread of scatter points indicating a strong relationship. It can be said that there is a **strong inverse relationship between AT and PE**.
These conclusions from the plot are supported by a cor value of about -0.948 which is close to -1. The negative cov value of -120.56 also shows that there is an inverse relationship.

Figure 2 shows that with an increase in Vacuum, Electrical Power Output decreases. There is also little vertical spread of scatter points indicating a strong relationship. Like in figure 1, it can be said that there is a **strong inverse relationship between V and PE**.
However, the spread is slightly larger than that of figure 1, hinting at a slightly weaker relationship and the cor value of -0.869 compared to -0.948 confirms that this is indeed true. The negative cov value of -188.56 also shows that there is an inverse relationship.

Figure 3 shows that, for most of the data points, an increase in Atmospheric Pressure leads to an increase in Electrical Power Output. Unlike figures 1 and 2, there is huge

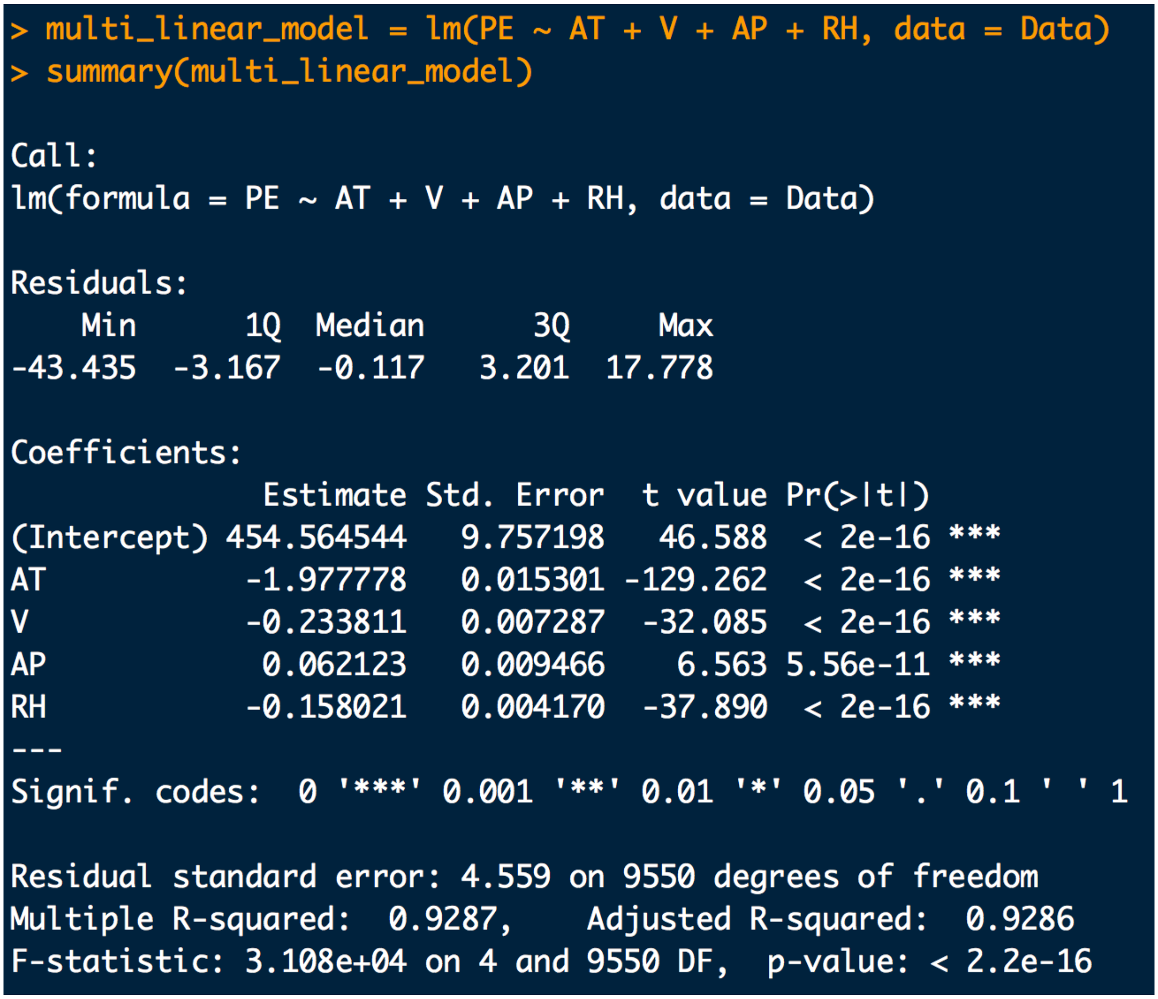
vertical spread of scatter points indicating a weak relationship. Hence, it can be concluded that there is a **weak positive relationship between AP and PE**.
These conclusions from the plot are supported by a cor value of about 0.518 which is much smaller than 1. The positive cov value of 52.5 also shows that there is a positive relationship.

Figure 4 shows that with an increase in Relative Humidity, Electrical Power Output increases. Like in figure 3, there is also a large vertical spread of scatter points indicating a weak relationship. It can thus be said that there is a **weak positive relationship between RH and PE**.

Yet, the spread is even larger in figure 4 than in figure 3, hinting at an even weaker relationship. This is confirmed by a cor value of about 0.389 which is even smaller in magnitude compared to 0.518. The positive cov value of 97.1 also shows that there is a positive relationship.

2. Simple Linear Regression on Predictors (Combined)

The following screenshot shows the R codes used to generate a multiple linear regression model using all the predictors against the target and its summary as well as the output:



```
> multi_linear_model = lm(PE ~ AT + V + AP + RH, data = Data)
> summary(multi_linear_model)

Call:
lm(formula = PE ~ AT + V + AP + RH, data = Data)

Residuals:
    Min      1Q  Median      3Q     Max 
-43.435  -3.167  -0.117   3.201  17.778 

Coefficients:
              Estimate Std. Error  t value Pr(>|t|)    
(Intercept) 454.564544   9.757198   46.588  < 2e-16 ***
AT           -1.977778   0.015301 -129.262  < 2e-16 ***
V            -0.233811   0.007287  -32.085  < 2e-16 ***
AP            0.062123   0.009466    6.563 5.56e-11 ***
RH           -0.158021   0.004170  -37.890  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 4.559 on 9550 degrees of freedom
Multiple R-squared:  0.9287,	Adjusted R-squared:  0.9286 
F-statistic: 3.108e+04 on 4 and 9550 DF,  p-value: < 2.2e-16
```

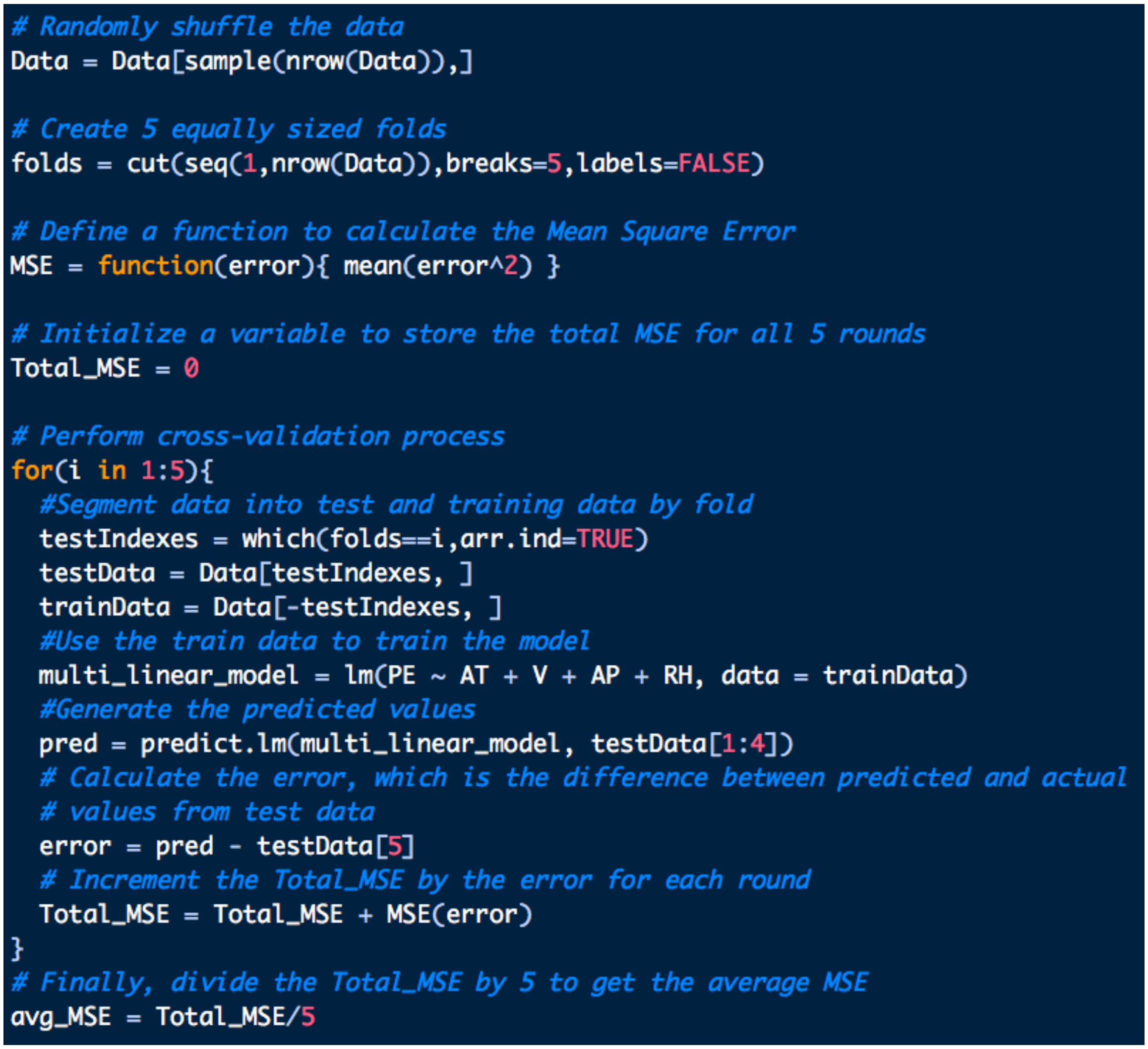
Since, multiple linear regression is being done, the Adjusted R-squared value is examined. For the model, the Adjusted R-squared value of 0.9286 is high, suggesting that all 4 predictors have a strong correlation with PE.

Furthermore, the coefficient of RH is negative which suggests that it has an inverse relationship with PE. This contradicts with the previous findings that RH has a positive relationship with PE.

The P-values of the various coefficients of the predictors suggest that they are significant. This means that the current findings which account for all the predictors working in tandem, is likely to be accurate and all the predictors are significant.

3. Evaluation of Model

The following screenshot demonstrates, with comments, a 5-fold cross-validation to calculate the average (unbiased estimate) Mean Square Error (MSE):



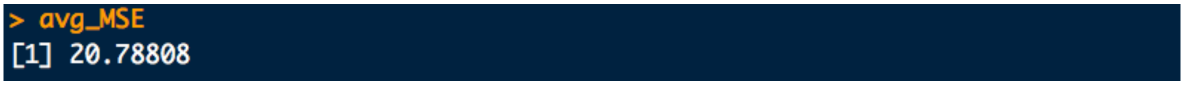
```
# Randomly shuffle the data
Data = Data[sample(nrow(Data)),]

# Create 5 equally sized folds
folds = cut(seq(1,nrow(Data)),breaks=5,labels=FALSE)

# Define a function to calculate the Mean Square Error
MSE = function(error){ mean(error^2) }

# Initialize a variable to store the total MSE for all 5 rounds
Total_MSE = 0

# Perform cross-validation process
for(i in 1:5){
  #Segment data into test and training data by fold
  testIndexes = which(folds==i,arr.ind=TRUE)
  testData = Data[testIndexes, ]
  trainData = Data[-testIndexes, ]
  #Use the train data to train the model
  multi_linear_model = lm(PE ~ AT + V + AP + RH, data = trainData)
  #Generate the predicted values
  pred = predict.lm(multi_linear_model, testData[1:4])
  # Calculate the error, which is the difference between predicted and actual
  # values from test data
  error = pred - testData[5]
  # Increment the Total_MSE by the error for each round
  Total_MSE = Total_MSE + MSE(error)
}
# Finally, divide the Total_MSE by 5 to get the average MSE
avg_MSE = Total_MSE/5
```

As mentioned in the code comments, the data is first shuffled to ensure that the 5 folds created later are random. Next, the data is coded into 5 intervals to be iterated through later in the cross-validation process.

Before beginning the process of cross-validation, a function to calculate MSE is defined and a variable is initialized to store the total mean squared error throughout the iterative process.

During the 5-fold cross-validation itself, the data is categorized into training and test data with each fold taking turns to act as the test data to validate the model. The training data is fed into the model and the model generates predictions to be later compared with the actual test data results. The MSE is calculated with the function earlier defined and the value is added into the total MSE variable. This process is repeated through 5 iterations.

Finally, the total MSE is divided by 5 to obtain the average MSE shown in the following screenshot below:



```
> avg_MSE
[1] 20.78808
```