Regression Analysis

Introduction:

Regression analysis is a set of statistical methods used for estimating the relationships between a dependent variable and one or more independent variables.
Some common forms of regression analysis are Linear regression, Logistic Regression, Ridge Regression, Lasso Regression, etc.

DATASET – 1

(CO2 – Carbon Dioxide Uptake in Grass Plants)

About the dataset :

The CO2 data frame has 84 rows and 5 columns of data from an experiment on the cold tolerance of the grass species *Echinochloa crus-galli*.

It has 5 attributes:

- **Plant :** An ordered factor which is also a unique identifier for each plant.
- **Type :** It contains the origin of the plant (Mississippi or Quebec)
- **Treatment :** Factors with levels chilled or nonchilled.
- **conc :** a numeric vector of ambient carbon dioxide concentrations
- **uptake :** a numeric vector of carbon dioxide uptake rates

Objective:

We'll try to fit a regression model to predict Treatment type based on the conc (CO2 concentration) and uptake (CO2 uptake).

Methodology:

We'll try to fit a logistic regression model as the attribute **Treatment** is having only 2 values i.e. chilled and nonchilled.

1. Installing and Loading Library

```
> install.packages("ISLR")
> library(ISLR)
```

2. Attaching dataset and initial processing

```
> attach(CO2)
```

As we know that our target attribute (i.e. Treatment) is having 2 categorical variables so, we'll first be converting it to binary values (i.e. 0 and 1).

```
> CO2$Treatment<-ifelse(CO2$Treatment=="nonchilled",0,1)
```

3. Model Fitting

We try to fit the model for dependent variable (i.e. Treatment) on independent variables (conc and uptake).

```
> model=glm(Treatment~ conc + uptake,data = CO2,family="binomi
al")
```

4. Model Summary

```
> summary(model)
```

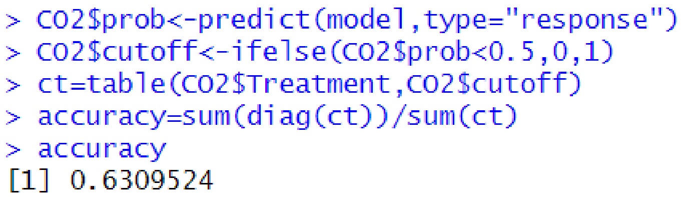
```
Call:
glm(formula = Treatment ~ conc + uptake, family = "binomial",
    data = CO2)

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)  1.6506858  0.6551051   2.520  0.01174 *
conc         0.0015974  0.0009655   1.655  0.09802 .
uptake      -0.0859501  0.0270428  -3.178  0.00148 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 116.45  on 83  degrees of freedom
Residual deviance: 104.70  on 81  degrees of freedom
AIC: 110.7

Number of Fisher Scoring iterations: 4
```

From summary, we can see the estimated values of regression coefficients and intercept.Also, we can see that uptake is significant attribute in predicting our target variable whereas conc is not having enough significance.

5. Predicting Values and Calculating Accuracy

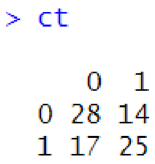
```
> CO2$prob<-predict(model,type="response")
> CO2$cutoff<-ifelse(CO2$prob<0.5,0,1)
> ct=table(CO2$Treatment,CO2$cutoff)
> accuracy=sum(diag(ct))/sum(ct)
```

```
> accuracy
```



```
> CO2$prob<-predict(model,type="response")
> CO2$cutoff<-ifelse(CO2$prob<0.5,0,1)
> ct=table(CO2$Treatment,CO2$cutoff)
> accuracy=sum(diag(ct))/sum(ct)
> accuracy
[1] 0.6309524
```

We are getting an accuracy score of 0.6309524 (i.e. we can say our model is predicting target variable with almost 63% accuracy)



```
> ct

     0  1
  0 28 14
  1 17 25
```

We can see the contingency table also. Correct predictions for 0 (i.e nonchiled) is 28 whereas for 1 (i.e. chilled) is 25. The number of correct prediction is more than wrong predictions.

Results:

We used CO2 dataset and tried to fit a logistic regression model for predicting whether the grass went under chilled treatment or nonchilled ones on the basis of carbon dioxide concentration and CO2 uptake. We got an accuracy of 63% with our model and found that CO2 concentration isn't singnificantly affecting the target variable during the prediction.

DATASET – 2

(BOD – Biochemical Oxygen Demand)

About the dataset :

The BOD dataframe has 6 rows and 2 columns giving the biochemical oxygen demand versus time in an evaluation of water quality.

It has 2 attributes:

- **Time :** A numeric vector giving the time of measurement (days)
- **demand :** A numeric vector giving the biochemical oxygen demand (mg/l)

Objective:

We'll try to fit a Simple Linear Regression Model to predict Oxygen demand based on the Time.

Methodology:

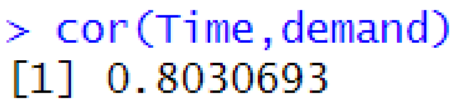
We'll try to fit a Simple Linear regression model for demand on Time.

1. **Attaching Dataset**

```
> attach(BOD)
```

2. **Checking Correlation Between variables**

```
> cor(Time,demand)
```

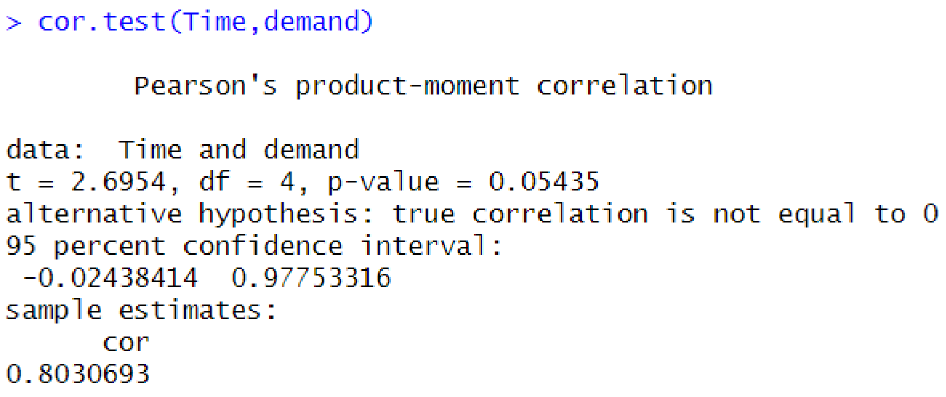


```
> cor(Time,demand)
[1] 0.8030693
```

We check whether the 2 variables are correlated or not and we found that correlation between 2 variables is 0.8 which means they are significantly positively correlated.

3. **Testing the correlation**

```
> cor.test(Time,demand)
```



```
> cor.test(Time,demand)

        Pearson's product-moment correlation

data:  Time and demand
t = 2.6954, df = 4, p-value = 0.05435
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.02438414  0.97753316
sample estimates:
      cor
0.8030693
```

We can see that alternative hypothesis is true. It means variables are correlated.

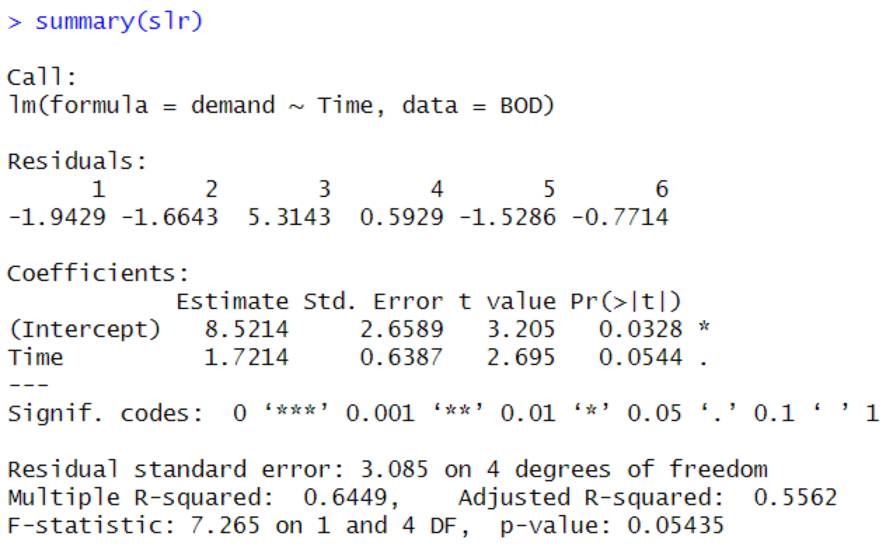
4 . Model Fitting

We try to fit the model for predicting demand based on time.

```
> slr=lm(demand ~ Time, data=BOD)
```

5. Model Summary

```
> summary(slr)
```

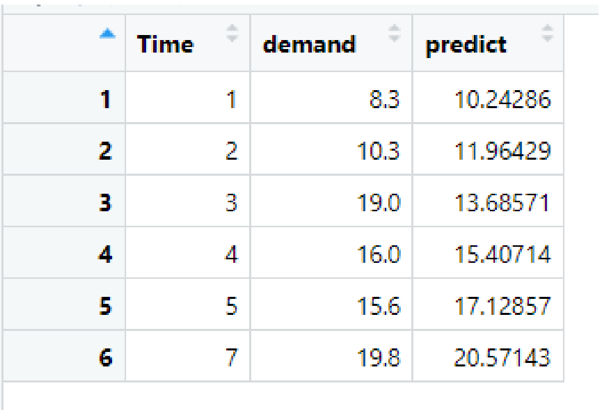


```
> summary(slr)

Call:
lm(formula = demand ~ Time, data = BOD)

Residuals:
      1       2       3       4       5       6 
-1.9429 -1.6643  5.3143  0.5929 -1.5286 -0.7714 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)  
(Intercept)   8.5214     2.6589   3.205   0.0328 *
Time          1.7214     0.6387   2.695   0.0544 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.085 on 4 degrees of freedom
Multiple R-squared:  0.6449,	Adjusted R-squared:  0.5562 
F-statistic: 7.265 on 1 and 4 DF,  p-value: 0.05435
```

R-square value is coming out to be 0.6449. So, we can say that time variable will predict/affect demand by 64.49%. It means if given Time, we can predict demand with 64% accuracy.

6. Viewing Predicted values

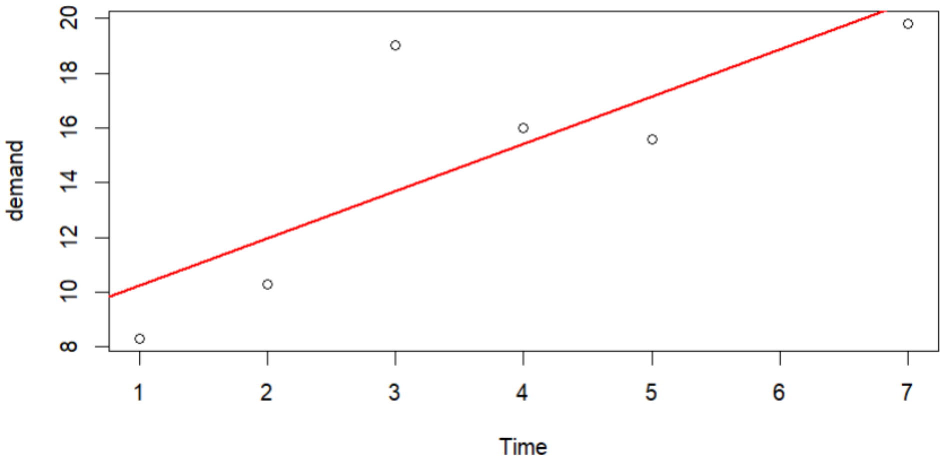
```
> BOD$predict<-predict(slr,type="response")
> View(BOD)
```



	Time	demand	predict
1	1	8.3	10.24286
2	2	10.3	11.96429
3	3	19.0	13.68571
4	4	16.0	15.40714
5	5	15.6	17.12857
6	7	19.8	20.57143

7. Fitting Regression Line

```
> plot(Time,demand)
> abline(slr,col='red',lwd=2)
```



Result:

We can say that Time & demand are correlated and we can predict demand with 64% accuracy when Time is given. Still, we can’t be very confident about our predictions as there are not enough datapoints in our dataset.