Unit-6:- *Linear Models, Simple Linear Regression, -Multiple Regression Generalized Linear Models,Logistic Regression, - Poisson Regression- other Generalized Linear Models-Survival Analysis,Nonlinear Models - Splines, Decision, Random Forests.*

Regression:- Regression analysis is a very widely used statistical tool to establish a relationship model between two variables. One of these variable is called predictor variable whose value is gathered through experiments. The other variable is called response variable whose value is derived from the predictor variable.

Linear Regression:- In Linear Regression these two variables are related through an equation, where exponent (power) of both these variables is 1. Mathematically a linear relationship represents a straight line when plotted as a graph. A non-linear relationship where the exponent of any variable is not equal to 1 creates a curve.

The general mathematical equation for a linear regression is − y = ax + b

Following is the description of the parameters used −

- y is the response variable.
- x is the predictor variable.
- a and b are constants which are called the coefficients.

lm() Function:-This function creates the relationship model between the predictor and the response variable.

Syntax: ***lm(formula,data)***

Following is the description of the parameters used −

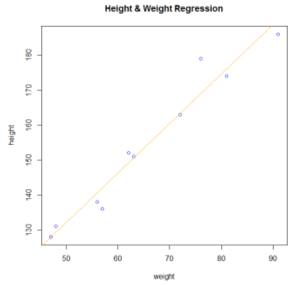
- **formula** is a symbol presenting the relation between x and y.
- **data** is the vector on which the formula will be applied.

Example:-

```
> height <- c(151, 174, 138, 186, 128, 136, 179, 163, 152, 131)
> weight <- c(63, 81, 56, 91, 47, 57, 76, 72, 62, 48)
> relation <- lm(weight~height)
> print(relation)

Call:
lm(formula = weight ~ height)

Coefficients:
(Intercept)       height
   -38.4551       0.6746
> plot(weight,height,col = "blue",main = "Height & Weight
Regression")
> abline(lm(height~weight),col="orange")
```



Advantages / Limitations of Linear Regression Model :

- Linear regression implements a statistical model that, when relationships between the independent variables and the dependent variable are almost linear, shows optimal results.
- Linear regression is often inappropriately used to model non-linear relationships.
- Linear regression is limited to predicting numeric output.
- A lack of explanation about what has been learned can be a problem.

Regression equation of x on y

$$x - \bar{x} = r.\frac{\sigma_x}{\sigma_y}(y - \overline{y})$$

Regression equation of y on x

$$y - \bar{y} = r.\frac{\sigma_y}{\sigma_x}(x - \overline{x})$$

where

$$r = \frac{\frac{1}{n}\sum xy - \bar{x} - \bar{y}}{\sqrt{\frac{1}{n}\sum (x-\bar{x})^2}\sqrt{\frac{1}{n}\sum (y-\bar{y})^2}}$$

Multiple Regression :- **Multiple regression** is an extension of simple **linear regression**. It is used when we want to predict the value of a variable based on the value of two or more other variables. The variable we want to predict is called the dependent variable

The general mathematical equation for multiple regression is – $x_1 = a_0 + a_1x_2 + a_2x_3$

Following is the description of the parameters used −

- x_1 is the response variable.
- a_0, a_1, a_2...bn are the coefficients.
- x_1, x_2, ...xn are the predictor variables.

The Normal equations for estimating a_0,a_1 and a_2 .

$$\sum x_1 = na_0 + a_1\sum x_2 + a_2\sum x_3$$
$$\sum x_1x_2 = a_0\sum x_2 + a_1\sum x_2^{\,2} + a_2\sum x_2x_3$$
$$\sum x_1x_3 = a_0\sum x_3 + a_1\sum x_2x_3 + a_2\sum x_3^{\,2}$$

We create the regression model using the lm() function in R. The model determines the value of the coefficients using the input data. Next we can predict the value of the response variable for a given set of predictor variables using these coefficients.

lm() Function :-This function creates the relationship model between the predictor and the response variable.

Syntax :- *lm(y ~ x1+x2+x3...,data)*

Following is the description of the parameters used −

- formula is a symbol presenting the relation between the response variable and predictor variables.
- data is the vector on which the formula will be applied.

Example

```
> lm(mpg~disp+hp+wt,data=mtcars)

Call:
lm(formula = mpg ~ disp + hp + wt, data = mtcars)

Coefficients:
(Intercept)         disp           hp           wt  
  37.105505    -0.000937    -0.031157    -3.800891  
```

Create Equation for Regression Model

Based on the above intercept and coefficient values, we create the mathematical equation.

$Y = a + disp.x_1 + hp.x_2 + wt.x_3$

or

$Y = 37.15 + (-0.000937)*x_1 + (-0.0311)*x_2 + (-3.8008)*x_3$

Logistic Regression : The Logistic Regression is a regression model in which the response variable (dependent variable) has categorical values such as True/False or 0/1. It actually measures the probability of a binary response as the value of response variable based on the mathematical equation relating it with the predictor variables.

The general mathematical equation for logistic regression is −

y = 1/(1+e^-(a+b_1x_1+b_2x_2+b_3x_3+...))

- **y** is the response variable.
- **x** is the predictor variable.
- **a** and **b** are the coefficients which are numeric constants.

The function used to create the regression model is the **glm()** function.

Syntax :- *glm(formula,data,family)*

Following is the description of the parameters used −

- **formula** is the symbol presenting the relationship between the variables.
- **data** is the data set giving the values of these variables.
- **family** is R object to specify the details of the model. It's value is binomial for logistic regression.

For example, in the built-in data set mtcars, the data column am represents the transmission type of the automobile model (0 = automatic, 1 = manual). With the logistic regression equation, we can model the probability of a manual transmission in a vehicle based on its engine horsepower and weight data.

```
> am.glm = glm(formula=am ~ hp + wt, data=mtcars, family=binomial)
> am.glm

Call:  glm(formula = am ~ hp + wt, family = binomial, data = mtcars)

Coefficients:
(Intercept)           hp           wt
   18.86630      0.03626     -8.08348

Degrees of Freedom: 31 Total (i.e. Null);  29 Residual
Null Deviance:      43.23
Residual Deviance: 10.06      AIC: 16.06
```

Poisson Regression:- Poisson Regression involves regression models in which the response variable is in the form of counts and not fractional numbers. For example, the count of number of births or number of wins in a football match series. Also the values of the response variables follow a Poisson distribution. The general mathematical equation for Poisson regression is −

$$\log(y) = a + b_1x_1 + b_2x_2 + b_nx_n.....$$

Following is the description of the parameters used −

- y is the response variable.
- a and b are the numeric coefficients.
- x is the predictor variable.

The function used to create the Poisson regression model is the glm()function.

We have the in-built data set "warpbreaks" which describes the effect of wool type (A or B) and tension (low, medium or high) on the number of warp breaks per loom. Let's consider "breaks" as the response variable which is a count of number of breaks. The wool "type" and "tension" are taken as predictor variables.

```
> output <-glm(formula = breaks ~ wool+tension,data = warpbreaks,
+ family = poisson)
> print(summary(output))

Call:
glm(formula = breaks ~ wool + tension, family = poisson, data = warpbreaks)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-3.6871  -1.6503  -0.4269   1.1902   4.2616

Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  3.69196    0.04541  81.302  < 2e-16 ***
woolB       -0.20599    0.05157  -3.994 6.49e-05 ***
tensionM    -0.32132    0.06027  -5.332 9.73e-08 ***
tensionH    -0.51849    0.06396  -8.107 5.21e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 297.37  on 53  degrees of freedom
Residual deviance: 210.39  on 50  degrees of freedom
AIC: 493.06

Number of Fisher Scoring iterations: 4
```

CURVE FITTING:- Curve fitting is the process of constructing a **curve**, or mathematical function, that has the best **fit** to a series of data points, possibly subject to constraints.

Type of curve	Equation	Normal equations
Fitting of a straight line	y = a + bx	$\sum y = na + b\sum x$ $\sum xy = a\sum x + b\sum x^2$
Fitting of a second degree polynomial	$y = a + bx + cx^2$	$\sum y = na + b\sum x + c\sum x^2$ $\sum xy = a\sum x + b\sum x^2 + c\sum x^3$ $\sum x^2 y = a\sum x^2 + b\sum x^3 + c\sum x^4$
Power curve	$y = a.b^x$	Applylog on both sides $\log y = \log a + x\log b$ $Y = A + Bx$ $\sum Y = nA + B\sum x$ $\sum xY = A\sum x + B\sum x^2$ where Y = log y, A = log a and B = log b
Exponential curve	$y = a.\ e^{bx}$	Applylog on both sides $\log y = \log a + bx$ $Y = A + bx$ $\sum Y = nA + b\sum x$ $\sum xY = A\sum x + b\sum x^2$ where Y = log y and A = log a
Exponential curve	$y = a.\ x^b$	Applylog on both sides $\log y = \log a + b\log x$ $Y = A + bX$ $\sum Y = nA + b\sum X$ $\sum XY = A\sum X + b\sum X^2$ where Y = log y , X=log x and A = log a

Problem:- Fit a straight line to the following data

x	*0*	*1*	*2*	*3*	*4*
y	*1*	*1.8*	*3.3*	*4.5*	*6.3*

Solution:-

Straight line is y = a+bx

The two normal equations are

$$\sum y = na + b\sum x$$

$$\sum xy = a\sum x + b\sum x^2$$

x	x^2	y	xy
0	0	1	0
1	1	1.8	1.8
2	4	3.3	13.2
3	9	4.5	40.5

4	16	6.3	100.8
$\sum x = 10$	$\sum x^2 = 30$	$\sum y = 16.9$	$\sum xy = 156.3$

Substituting the values, we get

5a+10b = 16.9 ....(1)

10a+30b = 156.3 ....(2)

Solving (1) and (2), we get

Multiply eq (1) with 2

10a+20b = 33.2 ....(3)

Subtract (3) and (2)

10a + 20b = 33.2

$\underline{10a + 30b = 156.3}$

0 -10b =-123.1

Therefore b=12.3 now substitute in (1) and a = -21.24.

Thus the equation of the straight line is y = a + bx

y = -21.24+12.3x

Problem:- **Fit a parabola to the following data**

x	1	2	3	4	5
y	10	12	8	10	14

Solution:-

Polynomial equation line is y = a + bx + cx²

The three normal equations are

$$\sum y = na + b\sum x + c\sum x^2$$

$$\sum xy = a\sum x + b\sum x^2 + c\sum x^3$$

$$\sum x^2 y = a\sum x^2 + b\sum x^3 + c\sum x^4$$

x	x^2	x^3	x^4	y	xy	x^2y
1	1	1	1	10	10	10
2	4	8	16	12	24	64
3	9	27	81	8	24	72
4	16	64	256	10	40	160
5	25	125	625	14	70	350
$\sum x = 15$	$\sum x^2 = 55$	$\sum x^3 = 225$	$\sum x^4 = 979$	$\sum y = 54$	$\sum xy = 168$	$\sum x^2 y = 656$

Substituting the values, we get

5a+15b+55c = 54 ....(1)

15a+55b+225c = 168 ....(2)

55a+225b+979c = 656 ....(3)

Solving (1) and (2), we get

Multiply eq (1) with 3 and subtract with (2)

15a+45b+165c = 162

$\underline{(-)15a+55b+225c = 168}$

0 -10b+60c = -6 ....(4)

Solving (1) and (3), we get

Multiply eq (1) with 11 and subtract with (3)

55a+165b+605c = 594

$\underline{(-)55a+225b+979c = 656}$

0 -60b-370c = -62

60 b+ 370 c = 62.......(5)

Solve (4) and (5)

Multiply eq (4) with 6 and add with (5)

-60b + 360c = -36

60b + 370c = 62

70 c =26

c =0.37

substitute in equation (4)

-10b + 60(0.37) = -6

b = 2.82

Substitute c and b values in equation (1)

5a+15b+55c = 54

5a+15(2.82)+55(0.37) = 54

a =-1.73

Thus the equation of the polynomial is $y = a + bx + cx^2$

$y = -1.73+2.82x + 0.37x^2$

Problem:- **Fit a curve of the type $y=ae^{bx}$ to the following data**

x	**0**	**1**	**2**	**3**
y	**1.05**	**2.1**	**3.85**	**8.3**

Solution:-Exponential curve equation line is $y = a . e^{bx}$

The two normal equations are

Apply logarithm on both sides

$$\log y = \log a + bx$$

$$Y = A + bx$$

$$\sum Y = nA + b\sum x$$

$$\sum xY = A\sum x + b\sum x^2$$

x	y	Y = log y	xY	x^2
0	1.05	0.021	0	0
1	2.1	0.324	0.32	1
2	3.85	0.585	1.17	4
3	8.3	0.919	2.75	9
$\sum x = 6$	$\sum y = 15.3$	$\sum Y = 1.849$	$\sum xY = 4.24$	$\sum x^2 = 14$

The equations are

4A + 6b = 1.84 …….(1)

6A + 14b = 4.24 …….(2)

Solve (1) and (2) equations

Multiply (1) with 3 and (2) with 2 then subtract them

12A+18b = 5.52

(-)12A+ 28b = 8.48

-10b =-2.96

b =0.296

Substitute b value in (1)

4A + 6(0.296) = 1.84

A = 0.016

a= antilog(A)

= antilog(0.016) =1.061

Therefore the exponential curve is $y = (1.061).e^{0.296x}$

Survival analysis: Survival analysis is generally defined as a set of methods for analyzing data where the outcome variable is the time until the occurrence of an event of interest. The event can be death, occurrence of a disease, marriage, divorce, etc.

In survival analysis, there is a special structure for right-censored survival data. To use this, one first must load the "survival" package, which is included in the main R distribution,

```
library(survival)
```

The basic syntax for creating survival analysis in R is −

```
Surv(time,event)
survfit(formula)
```

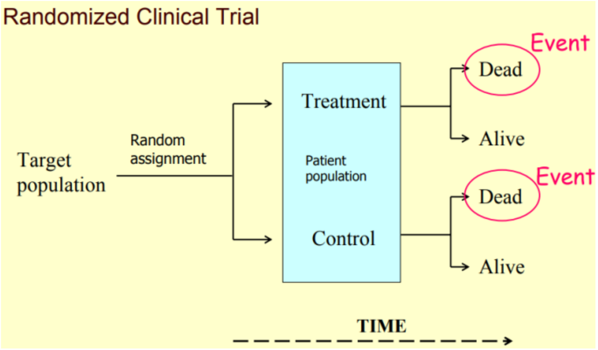
Following is the description of the parameters used −

- time is the follow up time until the event occurs.
- event indicates the status of occurrence of the expected event.
- formula is the relationship between the predictor variables.

Next, define the survival times "tt" and the censoring indicator "status", where "status = 1" indicates that the time is an observed event, and "status = 0" indicates that it is censored. Then the "Surv" function binds them into a single object. In the following example, time 6 is right censored, while the others are observed event times,

```
> tt <- c(2, 5, 6, 7, 8)
> status <- c( 1, 1, 0, 1, 1)
> Surv(tt, status) # Create a survival data structure
 [1] 2 5 6+ 7 8
```

Example:-



Nonlinear Models

Decision trees:- Decision tree is a graph to represent choices and their results in form of a tree. The nodes in the graph represent an event or choice and the edges of the graph represent the decision rules or conditions. It is mostly used in Machine Learning and Data Mining applications using R.

Examples:

- Predicting an email as spam or not spam,
- Predicting of a tumor is cancerous
- Predicting a loan as a good or bad credit risk based on the factors in each of these.

The package "party" has the function **ctree()** which is used to create and analyze decison tree.

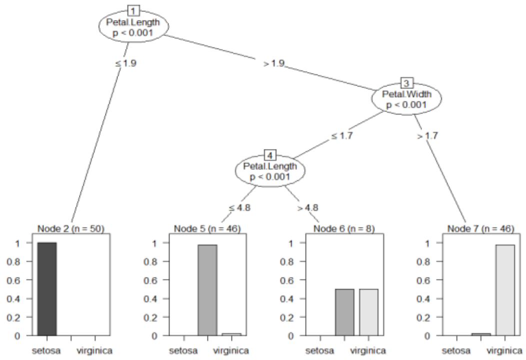
Syntax : *ctree(formula, data)*

Following is the description of the parameters used −

- **formula** is a formula describing the predictor and response variables.
- **data** is the name of the data set used.

Example:

```
library(party)
model2<-ctree(Species ~ Sepal.Length + Sepal.Width + Petal.Length + Petal.Width, data=mydata)
plot(model2)
```



Advantages of Decision Trees

- Simple to understand and interpret.
- Requires little data preparation.
- Works with both numerical and categorical data.
- Possible to validate a model using statistical tests. Gives you confidence it will work on new data sets.
- Robust.
- Scales to big data

Limitations of Decision Trees

- Learning globally optimal tree is NP-hard.
- Easy to overfit the tree
- Complex.

Random Forests:- In the random forest approach, a large number of decision trees are created. Every observation is fed into every decision tree. The most common outcome for each observation is used as the final output.

The package "randomForest" has the function **randomForest()** which is used to create and analyze random forests.

Syntax :- ***randomForest(formula, data)***

Following is the description of the parameters used −

- **formula** is a formula describing the predictor and response variables.
- **data** is the name of the data set used.

Advantages

- It is one of the most accurate learning algorithms available. For many data sets, it produces a highly accurate classifier.
- It runs efficiently on large databases.
- It can handle thousands of input variables without variable deletion.
- It gives estimates of what variables are important in the classification.
- It generates an internal unbiased estimate of the generalization error as the forest building progresses.
- It has an effective method for estimating missing data and maintains accuracy when a large proportion of the data are missing.

Disadvantages

- Random forests have been observed to overfit for some datasets with noisy classification/regression tasks.
- For data including categorical variables with different number of levels, random forests are biased in favor of those attributes with more levels. Therefore, the variable importance scores from random forest are not reliable for this type of data.

Splines: A linear spline is a continuous function formed by connecting linear segments. The points where the segments connect are called the knots of the spline.