# BUSINESS ANALYTICS
# Using R

# CONTENTS

# PRINCIPAL COMPONENTS ANALYSIS

## PRINCIPAL COMPONENTS ANALYSIS

- A dimensionality reduction technique
- Reduces the dimensionality of multivariate data without compromising much on the variation in the original data set.
- Achieved by transforming the original varaible into a new set of variables namely prinipal components (PCAs)
- PCAs are uncorrelated and ordered
- Hence the first few of them account for most of the variation in the orignal variables

## PRINCIPAL COMPONENTS ANALYSIS

- Desribes the variation in a set of correlated variables $x = (x_1, x_2, - - -, x_q)$ by a set of uncorrelated variables $y = (y_1, y_2, - - -, y_q)$
- Each principal component is a linear combination of the x variables.
- The new variables are derived in decreasing order of importance.
- Hence $y_1$ account for maximum possible variation in x among all linear combinations of x
- $y_2$ account for maximum possible of the remaining variation subject to being uncorrelated to y1. and so on.

## PRINCIPAL COMPONENTS ANALYSIS

- A dimensionality reduction technique
- Large number of correlated variables can be reduced to a manageable number of uncorrelated or independent factors.
- The emphasis is on the identification of underlying factors that might explain the dimensions associated with large data sets

$$y_i = a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + - - - + a_{iq}x_q$$

Where $y_i$: estimate of $i^{th}$ principal component, $a_i$: weight or score coefficient, $x_i$: $i^{th}$ variable and k: number of variables

The coefficients are selected such that

- the first principal component explains largest portion of the total variation
- the second first principal component accounts for the most of the residual variance, etc.

# PRINCIPAL COMPONENTS ANALYSIS

- Helps to understand the variability in large data sets with inter correlated variables using a smaller number of uncorrelated factors.
- Explaining variability of a set of n variables using *m* factors where $m < n$
- The emphasis is on the identification of underlying factors that might explain the dimensions associated with large data

## Objectives

- Reduces the complexity of a large set of variables by summarizing them in a smaller set of components or factors
- Tries to improve the interpretation of complex data through logical factors

# PRINCIPAL COMPONENTS ANALYSIS

## Computation of sample Principal Components

- The first principal component is that linear combination of original variables whose sample variance is greatest amongst all possible such linear combinations
- The second principal component is the linear combination of original variables that account for maximum proportion of the remaining variance subject to being uncorrelated with the first principal component and so on
- The first principal component is

$$y_1 = a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + - - - + a_{1q}x_q$$

The variance of $y_1$ be increased by increasing $a_1 = (a_{11}, - - -, a_{1q})$, a restriction must be placed on these coefficients.

The sensible restriction or constraint is to ensure that sum of squares of the coefficients should be equal to one

$$a_1'a_1 = 1$$

# PRINCIPAL COMPONENTS ANALYSIS

## Computation of sample Principal Components

- To choose the elements of $a_1$ which maximizes the variance of $y_1$ subject to the constraint of $a_1'a_1 = 1$
- Since $y_1$ is a linear combination of x, the sample variance of $y_1$ is given by

$$Var(y_1) = a_1'Sa_1$$

where S is the sample covariance matrix of x

The coefficients $a_1$ of first principal component $y_1$ is computed by solving

Maximize

$$z = a_1'Sa_1$$

Subject to

$$a_1'a_1 = 1$$

The solution to the above problem (using Lagrange multiplier method) is the Eigen vector of S corresponding to the largest Eigen value of S denoted by $\lambda_1$

# PRINCIPAL COMPONENTS ANALYSIS

## Computation of sample Principal Components

In general, the coefficients $a_i$ of first principal component $y_i$ is computed by solving

Maximize

$$z = a_i'Sa_i$$

Subject to

$$a_i'a_i = 1$$

The solution to the above problem (using Lagrange multiplier method) is the Eigen vector of S corresponding to the $i^{th}$ largest Eigen value of S denoted by $\lambda_i$

Since $a_i'a_i = 1$, the variance of $i^{th}$ principal component $y_1$ will be $\lambda_i$

# PRINCIPAL COMPONENTS ANALYSIS

## Steps

- Prepare correlation matrix
- Extract a set of principal components using correlation matrix
- Determine the number of principal components
- Interpret results

## PRINCIPAL COMPONENTS ANALYSIS

Example: Suppose a researcher wants to determine the underlying benefits consumers seek from the purchase of a toothpaste. A sample of 30 respondents was interviewed. The respondents were asked to indicate their degree of agreement with the following statements using a 7 point scale (1: strongly disagree, 7: strongly agree)

1. It is important to buy a toothpaste that prevents cavities
2. I like a toothpaste that gives shiny teeth
3. A toothpaste should strengthen your gums
4. I prefer toothpaste that freshens breath
5. Prevention of tooth decay is not an important benefit offered by a toothpaste
6. The most important consideration in buying a toothpaste is attractive teeth

# PRINCIPAL COMPONENTS ANALYSIS

Step 1: Normalize the data

z transform:

Transformed data = (Data – Mean) / SD

Reading ghe file to R

```
>mydata = mydata[,2:7]
```

Transforming the variables

```
>myzdata = scale(mydata)
```

# PRINCIPAL COMPONENTS ANALYSIS

Step 2: Check for Correlation

- Variables must be correlated for data reduction

> cor(myzdata)

**Correlation Matrix**

| | | x1 | x2 | x3 | x4 | x5 | x6 |
|---|---|---|---|---|---|---|---|
| Correlation | x1 | 1.000 | -.053 | .873 | -.086 | -.858 | .004 |
| | x2 | -.053 | 1.000 | -.155 | .572 | .020 | .640 |
| | x3 | .873 | -.155 | 1.000 | -.248 | -.778 | -.018 |
| | x4 | -.086 | .572 | -.248 | 1.000 | -.007 | .640 |
| | x5 | -.858 | .020 | -.778 | -.007 | 1.000 | -.136 |
| | x6 | .004 | .640 | -.018 | .640 | -.136 | 1.000 |

High correlation between $x_1$, $x_3$ & $x_5$

Good correlation between $x_2$, $x_4$ & $x_6$

## PRINCIPAL COMPONENTS ANALYSIS

Step 4: Method used: Principle Component Analysis

```
> mymodel = princomp(myzdata)

>summary(mymodel)
```

# PRINCIPAL COMPONENTS ANALYSIS

Step 4: Method used: Principle Component Analysis

Used to identify minimum number of components accounting for maximum variance in the data

Eigen Values: Amount of variance attributed to a component

Total Variance = 6 (Sum of all Eigen values)

Prop. variance for PC1= Eigen value of PC1 / Total Variance (2.731/6 = 0.455)

| Component | SD | Variance | Proportion of Variance | Cumulative Proportion of Variance |
|---|---|---|---|---|
| PC 1 | 1.653 | 2.732 | 0.455 | 0.455 |
| PC 2 | 1.489 | 2.217 | 0.369 | 0.825 |
| PC 3 | 0.665 | 0.442 | 0.074 | 0.899 |
| PC 4 | 0.584 | 0.341 | 0.057 | 0.955 |
| PC 5 | 0.427 | 0.182 | 0.030 | 0.986 |
| PC 6 | 0.292 | 0.085 | 0.014 | 1.000 |
| Total | | 6.000 | | |

# PRINCIPAL COMPONENTS ANALYSIS

Step 4: Determine the number of Components

1. Based on Eigen Values: Only components with Eigen value > 1.0 or Eigen value > 0.7 are selected.

2. Based on cumulative % variance: Factors extracted should account for at least 65 % of variance

| Component | SD | Variance | Proportion of Variance | Cumulative Proportion of Variance |
|---|---|---|---|---|
| PC 1 | 1.653 | 2.732 | 0.455 | 0.455 |
| PC 2 | 1.489 | 2.217 | 0.369 | 0.825 |
| PC 3 | 0.665 | 0.442 | 0.074 | 0.899 |
| PC 4 | 0.584 | 0.341 | 0.057 | 0.955 |
| PC 5 | 0.427 | 0.182 | 0.030 | 0.986 |
| PC 6 | 0.292 | 0.085 | 0.014 | 1.000 |
| Total | | 6.000 | | |

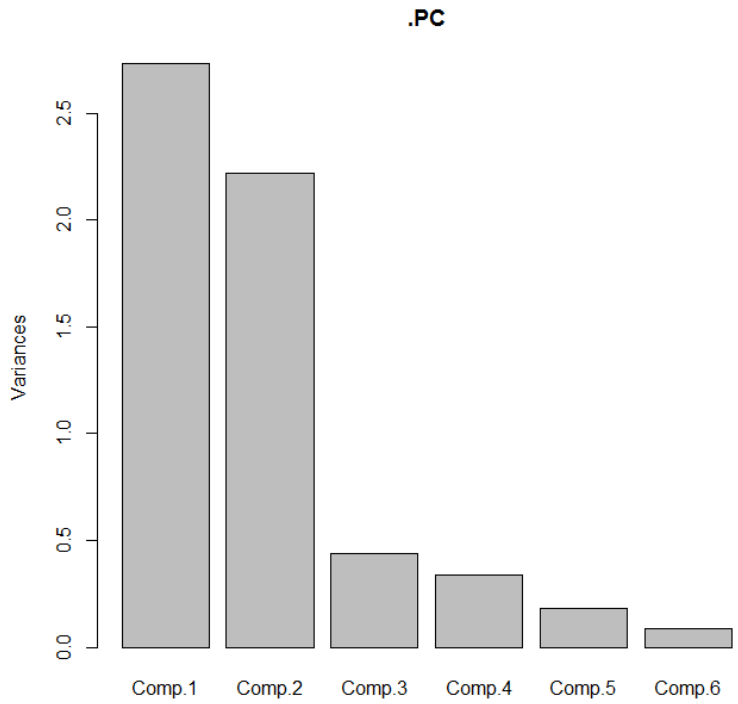
Number of factors selected : 2

# PRINCIPAL COMPONENTS ANALYSIS

Step 4: Determine the number of Factors

>plot(mymodel)

3. Based on Scree plot: Plot of the Eigen values against the number of factors in order of extraction. The number of components is identified based on slope change of scree plot




Number of factors selected : 2

# PRINCIPAL COMPONENTS ANALYSIS

## Step 5: Calculate Component Scores– Eigen Vectors

>loadings(mymodel)

$$y_i = a_{i1}x_1 + a_{i2}x_2 + a_{i3}x_3 + --- + a_{ik}x_k$$

| | Component | |
|---|---|---|
| | $y_1$ | $y_2$ |
| $x_1$ | 0.562 | -0.170 |
| $x_2$ | -0.182 | -0.534 |
| $x_3$ | 0.566 | -0.088 |
| $x_4$ | -0.207 | -0.530 |
| $x_5$ | -0.526 | 0.236 |
| $x_6$ | -0.107 | -0.585 |

# PRINCIPAL COMPONENTS ANALYSIS

Step 5: Interpret Components – Eigen Vectors

| | Component | |
|---|---|---|
| | $y_1$ | $y_2$ |
| $x_1$ | 0.562 | -0.170 |
| $x_2$ | -0.182 | -0.534 |
| $x_3$ | 0.566 | -0.088 |
| $x_4$ | -0.207 | -0.530 |
| $x_5$ | -0.526 | 0.236 |
| $x_6$ | -0.107 | -0.585 |

Component 1 is correlated with $x_1$, $x_3$ & $x_5$

Component 2 is correlated with $x_2$, $x_4$ & $x_6$

# PRINCIPAL COMPONENTS ANALYSIS

Step 5: Interpret Components

| | Component | |
|---|---|---|
| | $y_1$ | $y_2$ |
| Prevention of Cavities | 0.562 | -0.170 |
| $x_2$ | -0.182 | -0.534 |
| Strong Gum | 0.566 | -0.088 |
| $x_4$ | -0.207 | -0.530 |
| Non Prevention of Tooth Decay | -0.526 | 0.236 |
| $x_6$ | -0.107 | -0.585 |

Interpretation

Component 1 ($y_1$) represents the health related benefits

# PRINCIPAL COMPONENTS ANALYSIS

Step 5: Interpret Components

| | Component | |
|---|---|---|
| | $y_1$ | $y_2$ |
| Prevention of Cavities | 0.562 | -0.170 |
| Shiny Teeth | -0.182 | -0.534 |
| Strong Gum | 0.566 | -0.088 |
| Fresh Breath | -0.207 | -0.530 |
| Non Prevention of Tooth Decay | -0.526 | 0.236 |
| Attractive Teeth | -0.107 | -0.585 |

Interpretation

Component 2 ($y_2$) represents the social related benefits

# PRINCIPAL COMPONENTS ANALYSIS

## Step 6: Reduced Data Set

```
>pc = mymodel$scores

>cbind(pc[,1], pc[,2])
```

| Respondent | PC1 | PC2 | Respondent | PC1 | PC2 |
|---|---|---|---|---|---|
| 1 | 1.953 | -0.071 | 16 | 1.412 | 0.1352 |
| 2 | -1.6763 | 0.9852 | 17 | 1.261 | 0.6098 |
| 3 | 2.4298 | 0.6577 | 18 | 2.5041 | -0.2372 |
| 4 | -0.0908 | -1.6975 | 19 | -1.2981 | 1.3974 |
| 5 | -1.5154 | 2.7238 | 20 | -1.2777 | -1.7423 |
| 6 | 1.6696 | 0.0148 | 21 | -1.449 | 1.7912 |
| 7 | 1.0622 | 1.1536 | 22 | 0.9783 | -0.2455 |
| 8 | 2.0882 | -0.5402 | 23 | -1.4107 | 0.8217 |
| 9 | -1.29 | 1.3543 | 24 | -0.9281 | -2.6799 |
| 10 | -2.7958 | -1.6321 | 25 | 1.4305 | -0.0294 |
| 11 | 2.0398 | 0.3893 | 26 | -1.0791 | -2.2053 |
| 12 | -1.6682 | 0.9421 | 27 | 1.4698 | 0.106 |
| 13 | 2.4379 | 0.6146 | 28 | -1.5875 | -1.2162 |
| 14 | -0.4251 | -1.9974 | 29 | -0.8027 | -3.2699 |
| 15 | -1.6509 | 1.8801 | 30 | -1.7904 | 1.987 |

## PRINCIPAL COMPONENTS ANALYSIS

Exercise 1: Data on Customer satisfaction survey conducted by IT company is given below. Each customer is asked to were asked to indicate their degree of agreement with the following statements using a 7 point scale (1: strongly disagree, 7: strongly agree) . Can you reduce the 14 variables into less number of factors?

# EXPLORATORY FACTOR ANALYSIS

## FACTOR ANALYSIS

- Many times it may not be possible to measure some the concepts directly
- Such cases the concepts are examined indirectly by collecting information on variables which can be directly measurable and assumed to be indicators of the concepts of interest.

Example

It is difficult to conclude a student is interested in science or arts directly. The students scores on science subjects or arts subjects can be an indicator for the students interest in science or arts.

- Concepts which cannot be measured directly are called latent variables or factors
- The variables which can be directly measured and related to latent variables are called manifest variables

# FACTOR ANALYSIS

The method of analysis to uncover the relationship between latent variables and manifest variables is factor analysis

The method is based on multiple regression, except in factor analysis manifest variables is regressed on unobservable latent variables

Types of Factor Analysis: Exploratory and Confirmatory

## Exploratory Factor Analysis

Used to investigate the relationship between factors and manifest variables without making any assumption about which manifest variables is related to which factors

## Confirmatory Factor Analysis

Used to test whether a specific factor model postulated a priori on the relationship between factors and manifest variables is correct or not

# FACTOR ANALYSIS

## Factor analysis model

A regression model linking the manifest variables to a set of unobserved (or unobservable) latent variables

Assumes that the observed relationships between the manifest variables are the result of relationship between manifest variables and latent variables

The relationship between the manifest variables is measured using covariance matrix or correlation matrix.

# FACTOR ANALYSIS

## Factor analysis model

Let a set of observed or manifest variables x = ($x_1$, $x_2$, - - -, $x_q$) be linked to k unobserved latent variables or common factors $f_1$, $f_2$, - - -, $f_k$, where k < q by the regression model given by

$$x_1 = \lambda_{11} f_1 + \lambda_{12} f_2 + - - - + \lambda_{1k} f_k + \mu_1$$

$$x_2 = \lambda_{21} f_1 + \lambda_{22} f_2 + - - - + \lambda_{2k} f_k + \mu_2$$

- - - - - - - - - - - - - - - - - - - - - - - -

- - - - - - - - - - - - - - - - - - - - - - - -

$$x_q = \lambda_{q1} f_1 + \lambda_{q2} f_2 + - - - + \lambda_{qk} f_k + \mu_q$$

Where

$\lambda_j$ are regression coefficients of the x variables on the common factors known as factor loadings

Shows how each observed variable $x_i$, depends on the common factors

# FACTOR ANALYSIS

## Factor analysis model

The regression model is

$$x = \Lambda f + \mu$$

## Assumptions

The random disturbance terms $\mu_1$, $\mu_2$, - - -, $\mu_q$ are uncorrelated with each other and with the factors $f_1$, $f_2$, - - -, $f_k$.

Hence correlation between the observed variables arise from their relationship with the common factors.

The factors $f_1$, $f_2$, - - -, $f_k$ also uncorrelated and occur in the standardized form with mean zero and standard deviation one

# FACTOR ANALYSIS

## Principal Component Method of factor analysis model

Very similar to principal component analysis but not operating directly on S or R but on the reduced covariance matrix S*

$$S^* = S - \psi$$

# FACTOR ANALYSIS

## Steps

- Prepare correlation matrix
- Extract a set of factors using correlation matrix
- Determine the number of factors
- Rotate factors to increase interpretability
- Interpret results

## FACTOR ANALYSIS

Example: Suppose a researcher wants to determine the underlying benefits consumers seek from the purchase of a toothpaste. A sample of 30 respondents was interviewed. The respondents were asked to indicate their degree of agreement with the following statements using a 7 point scale (1: strongly disagree, 7: strongly agree)

1. It is important to buy a toothpaste that prevents cavities
2. I like a toothpaste that gives shiny teeth
3. A toothpaste should strengthen your gums
4. I prefer toothpaste that freshens breath
5. Prevention of tooth decay is not an important benefit offered by a toothpaste
6. The most important consideration in buying a toothpaste is attractive teeth

## FACTOR ANALYSIS

Step 1: Normalize the data

z transform:

Transformed data = (Data – Mean) / SD

Reading ghe file to R

```
>mydata = mydata[,2:7]
```

Transforming the variables

```
>myzdata = scale(mydata)
```

# FACTOR ANALYSIS

Step 2: Check for Correlation

- Variables must be correlated for data reduction

> cor(myzdata)

**Correlation Matrix**

| | | x1 | x2 | x3 | x4 | x5 | x6 |
|---|---|---|---|---|---|---|---|
| Correlation | x1 | 1.000 | -.053 | .873 | -.086 | -.858 | .004 |
| | x2 | -.053 | 1.000 | -.155 | .572 | .020 | .640 |
| | x3 | .873 | -.155 | 1.000 | -.248 | -.778 | -.018 |
| | x4 | -.086 | .572 | -.248 | 1.000 | -.007 | .640 |
| | x5 | -.858 | .020 | -.778 | -.007 | 1.000 | -.136 |
| | x6 | .004 | .640 | -.018 | .640 | -.136 | 1.000 |

High correlation between x1, x3 & x5

Good correlation between x2, x4 & x6

# FACTOR ANALYSIS

Step 3: Check for Sampling (factor) adequacy

```
>library(psych)
> KMO(myzdata)
```

| Statistics | Value | Criteria |
|---|---|---|
| Kaiser, Meyer, Olkin (KMO) | 0.66 | > 0.5 |

# FACTOR ANALYSIS

Step 4: Identifying the number of factors

Compute eigen values

Choose the factors with eigen values > 1

```
> s = cov(myzdata)
> s_eigen = eigen(s)
> variance = s_eigen$values
```

| Factor | Variance | % Variance | Cum % Variance |
|---|---|---|---|
| F1 | 2.731188 | 45.52 | 45.52 |
| F2 | 2.218119 | 36.97 | 82.49 |
| F3 | 0.441598 | 7.36 | 89.85 |
| F4 | 0.341258 | 5.69 | 95.54 |
| F5 | 0.182628 | 3.04 | 98.58 |
| F6 | 0.085209 | 1.42 | 100.00 |
| Total | 6 | | |

# FACTOR ANALYSIS

Step 4: Determine the number of Factors

1. Based on Eigen Values: Only factors with Eigen value > 1.0 are selected
2. Based on cumulative % variance: Factors extracted should account for at least 65 % of variance

| Factor | Variance | % Variance | Cum % Variance |
|---|---|---|---|
| F1 | 2.731188 | 45.52 | 45.52 |
| F2 | 2.218119 | 36.97 | 82.49 |
| F3 | 0.441598 | 7.36 | 89.85 |
| F4 | 0.341258 | 5.69 | 95.54 |
| F5 | 0.182628 | 3.04 | 98.58 |
| F6 | 0.085209 | 1.42 | 100.00 |
| Total | 6 | | |

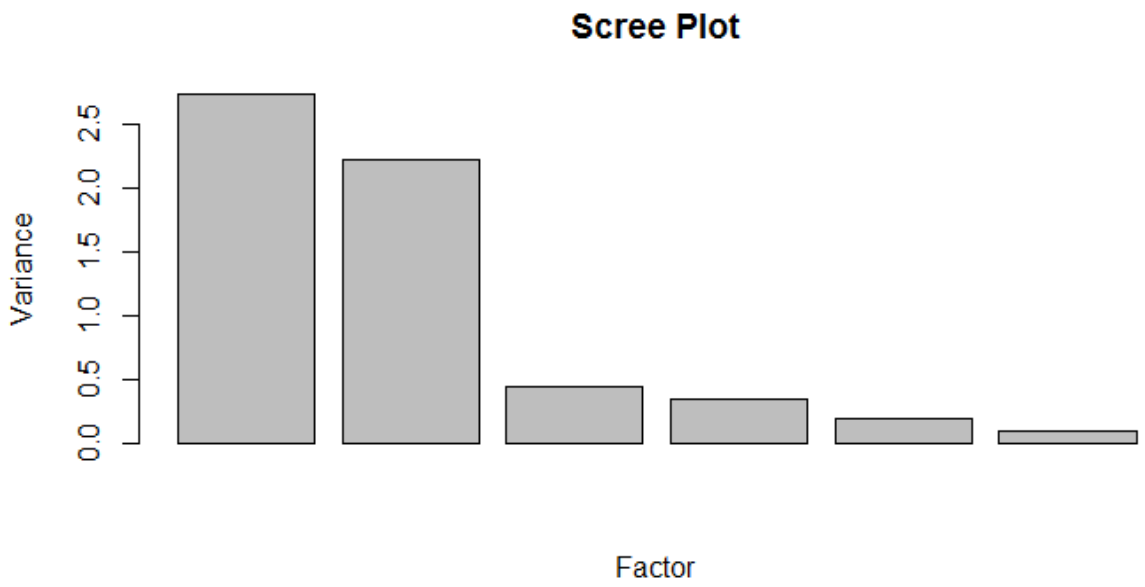
Number of factors selected : 2

## FACTOR ANALYSIS

Step 4: Determine the number of Factors

```
> barplot(variance, xlab = "Factor", ylab = 'Variance', main = "Scree Plot“)
```

3. Based on Scree plot: Plot of the eigen values against the number of factors in order of extraction. The number of factors is identified based on slope change of scree plot




Number of factors selected : 2

# FACTOR ANALYSIS

## Step 5: Calculate Factor Scores

```
> mymodel = factanal(myzdata, 2)
```

| | Component | |
|---|---|---|
| | 1 | 2 |
| x1 | 0.968 | 0.000 |
| x2 | 0.000 | 0.749 |
| x3 | 0.898 | -0.140 |
| x4 | 0.000 | 0.784 |
| x5 | -0.887 | 0.236 |
| x6 | 0.000 | 0.830 |

Interpretation is difficult when the variables are evenly loaded on many factors
Solution: Rotation

# FACTOR ANALYSIS

## Step 5: Calculate Factor Scores: Rotation

A process by which a solution is made more interpretable without changing its underlying mathematical properties.

Types of rotations

1. Orthogonal rotation
2. Oblique rotation

### Orthogonal rotation

Restricts the rotated factors to being uncorrelated

### Oblique rotation

allows the rotated factors to be correlated

# FACTOR ANALYSIS

Step 5: Calculate Factor Scores: Rotation

A process by which a solution is made more interpretable without changing its underlying mathematical properties.

Commonly used rotation : Orthogonal rotation

Commonly used orthogonal rotation : varimax rotation

Try to achieve factors with a few large loadings and as many near – zero loadings as possible

# FACTOR ANALYSIS

## Step 5: Calculate Factor Scores: Rotation

```
> myrotatedmodel = factanal(myzdata, 2, rotation = "varimax", scores = "regression")
> myrotatedmodel
```

| | Component | |
|---|---|---|
| | 1 | 2 |
| x1 | 0.968 | 0.000 |
| x2 | 0.000 | 0.749 |
| x3 | 0.898 | -0.140 |
| x4 | 0.000 | 0.784 |
| x5 | -0.887 | 0.236 |
| x6 | 0.000 | 0.830 |

## FACTOR ANALYSIS

Step 5: Interpret Components – Eigen Vectors

| | Component | |
|---|---|---|
| | 1 | 2 |
| x1 | 0.968 | 0.000 |
| x2 | 0.000 | 0.749 |
| x3 | 0.898 | -0.140 |
| x4 | 0.000 | 0.784 |
| x5 | -0.887 | 0.236 |
| x6 | 0.000 | 0.830 |

Component 1 is correlated with x1, x3 & x5

Component 2 is correlated with x2, x4 & x6

# FACTOR ANALYSIS

Step 5: Interpret Components

| | Component | |
|---|---|---|
| | 1 | 2 |
| Prevention of Cavities | 0.968 | 0.000 |
| x2 | 0.000 | 0.749 |
| Strong Gum | 0.898 | -0.140 |
| x4 | 0.000 | 0.784 |
| Non Prevention of Tooth Decay | -0.887 | 0.236 |
| x6 | 0.000 | 0.830 |

Interpretation

Component 1 represents the health related benefits

## FACTOR ANALYSIS

Step 5: Interpret Components

| | Component | |
|---|---|---|
| | 1 | 2 |
| Prevention of Cavities | 0.968 | 0.000 |
| Shiny Teeth | 0.000 | 0.749 |
| Strong Gum | 0.898 | -0.140 |
| Fresh Breath | 0.000 | 0.784 |
| Non Prevention of Tooth Decay | -0.887 | 0.236 |
| Attractive Teeth | 0.000 | 0.830 |

Interpretation

Component 2 represents the social related benefits

# FACTOR ANALYSIS

## Step 6: Reduced Data Set

> output = myrotatedmodel$scores

> output

| Respondent | Factor1 | Factor2 | Respondent | Factor1 | Factor2 |
|---|---|---|---|---|---|
| 1 | 1.3046 | -0.2413 | 16 | 0.8934 | -0.342 |
| 2 | -1.2952 | -0.2556 | 17 | 0.5714 | -0.5502 |
| 3 | 1.1629 | -0.7569 | 18 | 1.501 | -0.24 |
| 4 | 0.1747 | 1.0108 | 19 | -0.9845 | -0.6938 |
| 5 | -1.428 | -1.3608 | 20 | -0.4187 | 1.259 |
| 6 | 0.9864 | -0.2511 | 21 | -1.3132 | -0.8042 |
| 7 | 0.4605 | -0.9084 | 22 | 0.5706 | -0.0143 |
| 8 | 1.1867 | -0.0515 | 23 | -0.9855 | -0.1067 |
| 9 | -0.7678 | -0.6358 | 24 | 0.0209 | 1.7277 |
| 10 | -1.1191 | 1.3473 | 25 | 0.8821 | -0.1881 |
| 11 | 1.0738 | -0.6495 | 26 | -0.3011 | 1.5195 |
| 12 | -1.0785 | -0.1976 | 27 | 0.4675 | -0.2914 |
| 13 | 1.3796 | -0.6989 | 28 | -0.5606 | 0.7776 |
| 14 | 0.0978 | 1.2286 | 29 | 0.1111 | 2.1146 |
| 15 | -1.394 | -0.7847 | 30 | -1.1988 | -0.9623 |

## FACTOR ANALYSIS

Exercise 1: Data on Customer satisfaction survey conducted by IT company is given below. Each customer is asked to were asked to indicate their degree of agreement with the following statements using a 7 point scale (1: strongly disagree, 7: strongly agree) . Can you reduce the 14 variables into less number of factors?

# CLUSTER ANALYSIS

## CLUSTER ANALYSIS

A technique used to classify objects or cases into relatively homogeneous groups called clusters

Cluster

A collection of data objects similar to one another within the same cluster and dissimilar to the objects in other clusters

Cluster analysis

A procedure for grouping a set of data objects into clusters

## CLUSTER ANALYSIS

- A technique used to classify objects or cases into relatively homogeneous groups called clusters

Example: A survey was done to study the consumers attitude towards shopping. The consumers need to be clustered based on their attitude towards shopping. The respondents were asked to express their degree of agreement with the following statements on a 7 point scale (1: strongly disagree, 7: strongly agree).

x1: Shopping is fun

x2: Shopping is bad for your budget

x3: I combine shopping with eating out

x4: I try to get the best buys when shopping

x5: I don’t care about shopping

x6: You can save a lot os money by comparing prices

Step 1: Choose Type of clustering - Agglomerative Clustering

- Hierarchical Clustering – characterized by development of a hierarchy or tree like structure
- Starts with each object or record as separate clusters
- Clusters are formed by grouping objects in to bigger and bigger clusters until all objects are in one cluster.
- The objects grouped based on linkage measure
- Commonly used linkage measure is Euclidean distance d,
- Euclidean distance between two records i and j, $d_{ij}$ is defined as

$$d_{ij} = \sqrt{\sum_{k=1}^{q} (x_{ik} - x_{jk})^2}$$

Step 1: Choose Type of clustering - Agglomerative Clustering

- The data are not partitioned into a particular number of classes or groups at a single step
- Consists of a series of partitions that may run from a single cluster containing all individuals to n clusters, each contain a single individual
- Produce partitions by a series of successive fusions of the n individuals into groups
- Fusion once made are irreversible, when the algorithm has placed two individuals in the same group they cannot subsequently appear in different groups

## CLUSTER ANALYSIS

### Types of Linkage

1. Single Linkage:

Based on minimum distance

The first two objects clustered are those having minimum distance between them

$$d_{AB} = \min_{\substack{i \in A \\ j \in B}} (d_{ij})$$

2. Complete Linkage:

Based on maximum distance

The distance between two clusters is calculated as the distance between two furthest points

$$d_{AB} = \max_{\substack{i \in A \\ j \in B}} (d_{ij})$$

Where $d_{AB}$ is the distance between two clusters A and B and $d_{ij}$ is the distance between individuals i and j found from the initial inter – individual distance matrix

# CLUSTER ANALYSIS

## Types of Linkage

### 3. Average Linkage:

Based on average distance

The distance between two clusters is defined as the average of the distance between all pairs of points

Preferred method

$$d_{AB} = \frac{1}{n_A n_B} \sum_{i \in A} \sum_{j \in B} d_{ij}$$

Where $n_A$ and $n_B$ are the numbers of individuals in clusters A and B

## CLUSTER ANALYSIS

### Step 2: Choose Method

Variance method:

Generates clusters with minimum within cluster variance

Uses Ward's Procedure

Ward's Procedure

For each cluster means for all the variables are computed

For each object or record, the Euclidean distance to the cluster mean is computed

## CLUSTER ANALYSIS

R Code

Read data to mydata and compute distance

```
> distance = dist(mydata, method = “euclidean”)
```

Generate Clusters

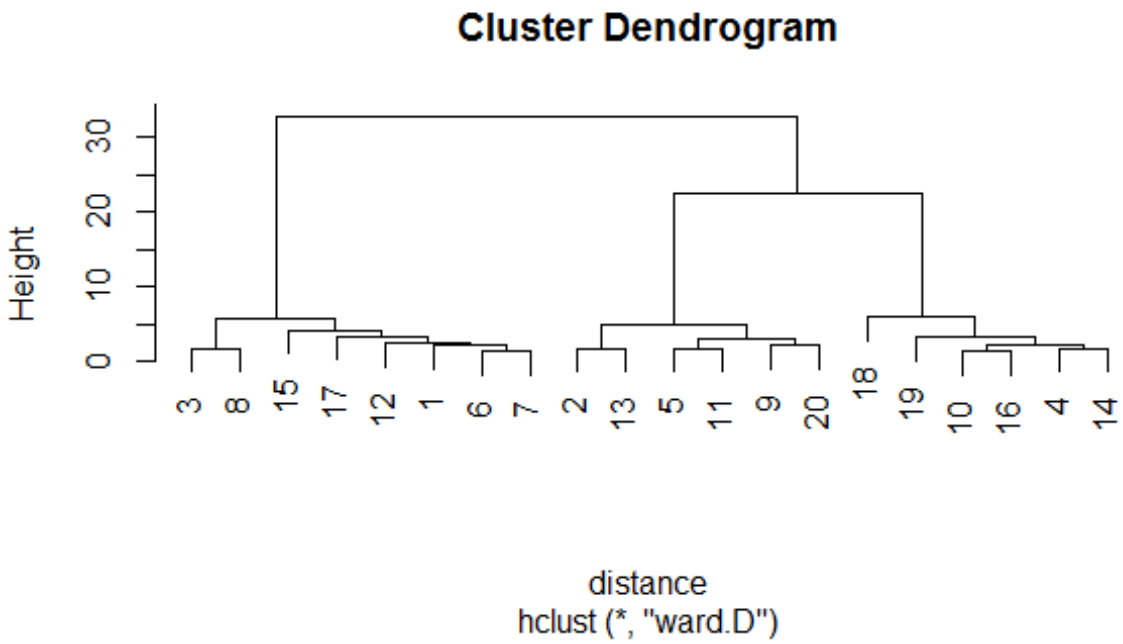
```
>mymodel = hclust(distance, method = “ward”)
```

Plot Dendogram

```
>plot(mymodel)
```

## CLUSTER ANALYSIS

Decide on number of clusters: Dendrogram



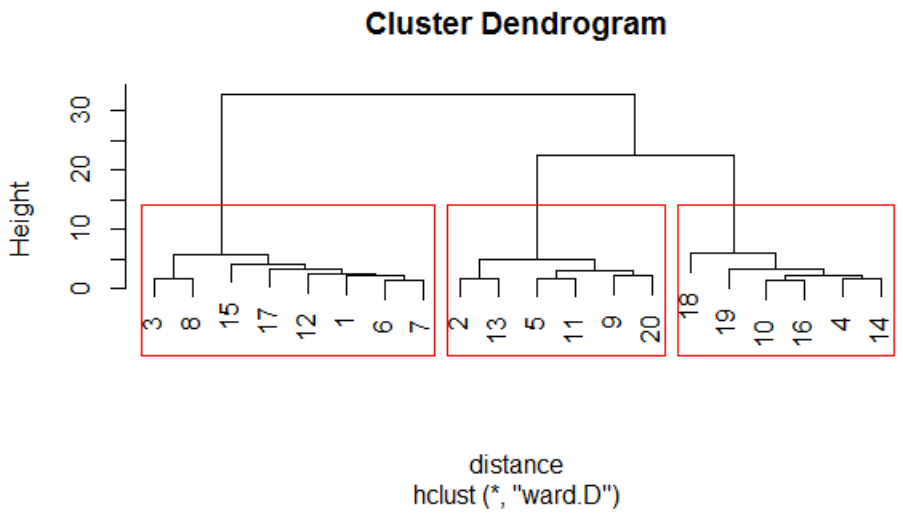
# CLUSTER ANALYSIS

## Decide on number of clusters: Dendrogram

Stages is given in x axis and distance in y axis

When one move from 3 cluster to 2 cluster the distance increases drastically. So 3 cluster may be appropriate

```
>groups = cutree(mymodel, k = 3)
> rect.hclust(mymodel, k = 3, border = "red")
```



# CLUSTER ANALYSIS

## Identification of cluster membership for each record

```
> output = cbind(mydata, groups
>write,csv(output, "E:/ISI_Mumbai/output.csv“)
```

| Respondent id | x1 | x2 | x3 | x4 | x5 | x6 | groups |
|---|---|---|---|---|---|---|---|
| 1 | 6 | 4 | 7 | 3 | 2 | 3 | 1 |
| 2 | 2 | 3 | 1 | 4 | 5 | 4 | 2 |
| 3 | 7 | 2 | 6 | 4 | 1 | 3 | 1 |
| 4 | 4 | 6 | 4 | 5 | 3 | 6 | 3 |
| 5 | 1 | 3 | 2 | 2 | 6 | 4 | 2 |
| 6 | 6 | 4 | 6 | 3 | 3 | 4 | 1 |
| 7 | 5 | 3 | 6 | 3 | 3 | 4 | 1 |
| 8 | 7 | 3 | 7 | 4 | 1 | 4 | 1 |
| 9 | 2 | 4 | 3 | 3 | 6 | 3 | 2 |
| 10 | 3 | 5 | 3 | 6 | 4 | 6 | 3 |
| 11 | 1 | 3 | 2 | 3 | 5 | 3 | 2 |
| 12 | 5 | 4 | 5 | 4 | 2 | 4 | 1 |
| 13 | 2 | 2 | 1 | 5 | 4 | 4 | 2 |
| 14 | 4 | 6 | 4 | 6 | 4 | 7 | 3 |
| 15 | 6 | 5 | 4 | 2 | 1 | 4 | 1 |
| 16 | 3 | 5 | 4 | 6 | 4 | 7 | 3 |
| 17 | 4 | 4 | 7 | 2 | 2 | 5 | 1 |
| 18 | 3 | 7 | 2 | 6 | 4 | 3 | 3 |
| 19 | 4 | 6 | 3 | 7 | 2 | 7 | 3 |
| 20 | 2 | 3 | 2 | 4 | 7 | 2 | 2 |

# CLUSTER ANALYSIS

## Cluster Profile

> aggregate(mydata, by = list(groups), FUN = mean)

| Variables | Cluster Mean | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| x1 (shopping is fun) | 5.750 | 1.667 | 3.500 |
| x2 (shopping upsets my budget) | 3.625 | 3.000 | 5.833 |
| x3 (I combine shopping with eating out) | 6.000 | 1.833 | 3.333 |
| x4 (I try to get best buys when shopping) | 3.125 | 3.500 | 6.000 |
| x5 (I don't care about shopping) | 1.875 | 5.500 | 3.500 |
| x6 (save a lot by comparing prices | 3.875 | 3.333 | 6.000 |

Cluster 1: High on x1 & x3 but low on x5
Fun loving and concerned

Cluster 2: Low on x1 & x3 but High on x5
Careless & no fun in shopping (apathetic)

Cluster 3: High on x2 x4 & x6
Concerned about spending money (Economical)

# CLUSTER ANALYSIS

## k mean clustering

Partitions n individuals in a set of multivariate data into k groups or clusters ($G_1$, $G_2$, - - -, $G_k$)

k is given or a possible range is specified

Common approach is to identify the k groups which minimizes the within – group sum of squares (WGSS)

$$WGSS = \sum_{j=1}^{q} \sum_{l=1}^{k} \sum_{i \in G_i} (x_{ij} - \bar{x}_j^{(1)})^2$$

Where $\bar{x}_j^{(l)} = \frac{1}{n_i} \sum_{i \in G_i} x_{ij}$ is the mean of the individuals in group $G_l$ on variable j

Computing WGSS for each value of k and choose that of value of k which minimize WGSS is almost impossible

One option is to plot WGSS for different values of k and choose the optimum k at which the slope of the curve changes

# CLUSTER ANALYSIS

## k mean clustering

Computing WGSS for each value of k and choose that of value of k which minimize WGSS is almost impossible

Moreover as number of cluster increases BGSS decreases or BGSS / Total SS will increase

One option is to plot BGSS/ Total SS for different values of k and choose the optimum k at which the curve flattens or slope changes

## CLUSTER ANALYSIS

Example: Cluster the data given in cluster_Analysis_example.csv using k mean method

```
>mynewmodel = kmeans(mydata,3)
> mynewmodel
>cluster = mynewmodel$cluster
>output = cbind(mydata, cluster)
> write.csv(output, "E:/MSQMS/Applied_Multivariate_Analysis/output.csv")
```

To find optimum k, compute BGSS / Total SS for different values of k

```
> kmeans(mydata,k, k =.1,2, - - -)
```

## CLUSTER ANALYSIS

Example: Cluster the data given in cluster_Analysis_example.csv using k mean method

optimum k,

> kmeans(mydata,k, k =.1,2, - - -)

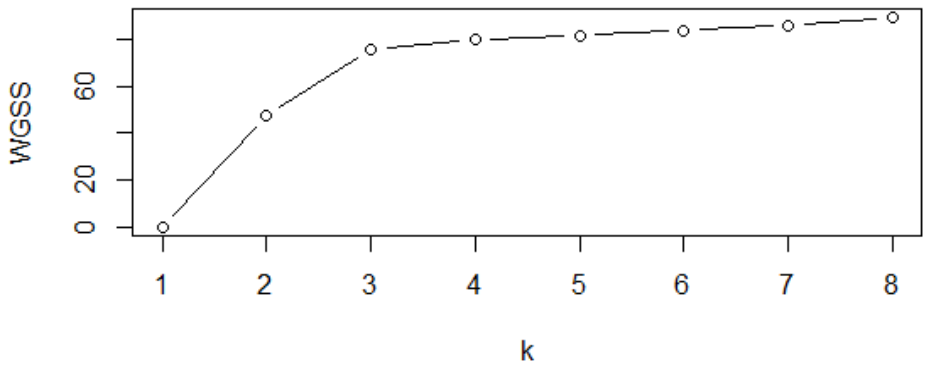
| k | WGSS/Total SS |
|---|---|
| 1 | 0.0 |
| 2 | 47.5 |
| 3 | 75.8 |
| 4 | 79.6 |
| 5 | 81.4 |
| 6 | 83.7 |
| 7 | 85.8 |
| 8 | 89.2 |

## CLUSTER ANALYSIS

Example: Cluster the data given in cluster_Analysis_example.csv using k mean method

optimum k,

```
> plot(k, WGSS, type = "b")
```




The curve flattens after k = 3, hence optimum k is 3

## CLUSTER ANALYSIS

Exercise 1: Data on Customer satisfaction survey conducted by IT company is given below. Each customer is asked to were asked to indicate their degree of agreement with the following statements using a 7 point scale (1: strongly disagree, 7: strongly agree) . Can you group the customers into meaningful groups?

# POISSON REGRESSION

## POISSON REGRESSION

Used to develop models when the output or response variable y is count

Models the logarithm of the output variable

$$y = e^{a+b_1x_1+b_2x_2+---+b_kx_k}$$

y: output variable
$x_i$'s : independent variables
a, $b_1$, $b_2$, ---: coefficients to be estimated

If estimate of $p \geq 0.5$, then classified as success, otherwise as failure

## POISSON REGRESSION

Usage: When the dependant variable (Y variable) is count

Example: Develop a model to predict the number of code review defects. The control factors identified are author skill (0: Low, 1: High), reviewer skill (0: Low, 1: High), review type (0: peer, 1: fagan), preparation time and size. The data is given in CR_Data.csv?

1. Reading the file and variables

```
> size = mydata$Size
> askill = mydata$Author.Skill
> rskill = mydata$Reviewer.skill
> rtype = mydata$Review.type
> ptime = mydata$Preparation.time
> defects = mydata$Defects
```


70

## POISSON REGRESSION

Usage: When the dependant variable (Y variable) is count

2. Converting control variables to factors

```
> askill = factor(askill)
> rskill = factor(rskill)
> rtype = factor(rtype)
```

## POISSON REGRESSION

3. Perform Poisson regression

```
> mymodel = glm(defects ~ size + askill + rskill + rtype + ptime, family = poisson())
> summary(mymodel)
```

| | Estimate | Std. Error | z | p value |
|---|---|---|---|---|
| (Intercept) | 6.01365 | 0.218261 | 27.553 | 0.0000 |
| size | 0.02265 | 0.002585 | 8.761 | 0.0000 |
| askill1 | -0.7701 | 0.067414 | -11.423 | 0.0000 |
| rskill1 | -0.0698 | 0.040083 | -1.74 | 0.0818 |
| rtype1 | 0.60998 | 0.067548 | 9.03 | 0.0000 |
| ptime | -6.6641 | 0.823795 | -8.09 | 0.0000 |

# POISSON REGRESSION

4. Fitted Values

```
> pred = predict(mymodel, type = "response")
> res = defects – pred
> output = cbind(defects, pred, res) > write.csv(output, "E:/Infosys/Part 2/output.csv")
```

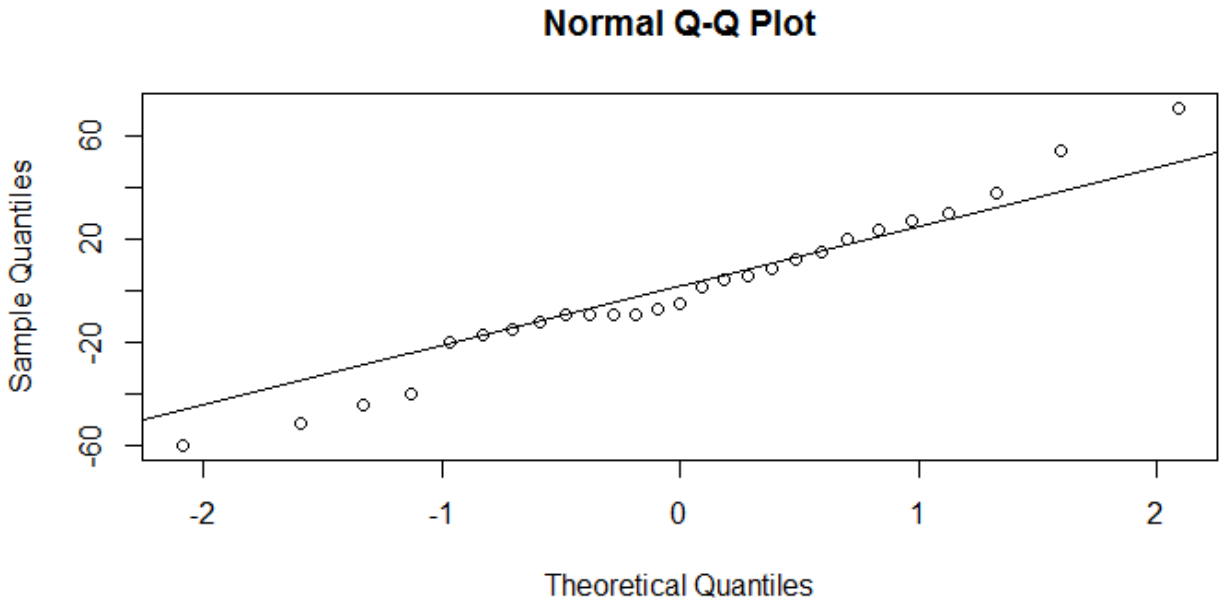
| SL No | Actual Defects | Predicted Defects | Residuals | SL No | Actual Defects | Predicted Defects | Residuals |
|---|---|---|---|---|---|---|---|
| 1 | 37 | 49.284 | -12.284 | 15 | 78 | 129.600 | -51.600 |
| 2 | 55 | 53.592 | 1.408 | 16 | 77 | 68.719 | 8.281 |
| 3 | 49 | 54.055 | -5.055 | 17 | 135 | 144.130 | -9.130 |
| 4 | 45 | 54.424 | -9.424 | 18 | 141 | 136.840 | 4.160 |
| 5 | 88 | 102.915 | -14.915 | 19 | 220 | 148.800 | 71.200 |
| 6 | 45 | 54.055 | -9.055 | 20 | 145 | 205.335 | -60.335 |
| 7 | 89 | 133.473 | -44.473 | 21 | 165 | 127.394 | 37.606 |
| 8 | 72 | 59.852 | 12.148 | 22 | 120 | 160.056 | -40.056 |
| 9 | 49 | 58.647 | -9.647 | 23 | 170 | 149.922 | 20.078 |
| 10 | 66 | 51.112 | 14.888 | 24 | 250 | 222.637 | 27.363 |
| 11 | 37 | 57.100 | -20.100 | 25 | 197 | 142.921 | 54.079 |
| 12 | 118 | 125.045 | -7.045 | 26 | 150 | 167.176 | -17.176 |
| 13 | 65 | 59.584 | 5.416 | 27 | 187 | 156.756 | 30.244 |
| 14 | 75 | 51.577 | 23.423 | | | | |

## POISSON REGRESSION

5. Normality test on residuals

> qqnorm(res)

> qqline(res)





74

## POISSON REGRESSION

5. Normality test on residuals

> shapiro.test(res)

| Statistic | Value |
|---|---|
| w | 0.9783 |
| p value | 0.8233 |

## POISSON REGRESSION

Exercise 2: The data on the number of awards received by different students based on the program to which they are admitted and their mathematics score in final examination is given in Poisson_Reg.csv. Develop a model to estimate the number of awards ?

If there is over dispersion (if residual deviance is much large then degrees of freedom) then use quasi  Poisson family

# NOMINAL LOGISTIC REGRESSION

# NOMINAL LOGISTIC REGRESSION

Used to develop models when the output or response variable y is nominal

The output variable will be categorical, having more than two categories

Models log odds of the outcomes as linear combination of the predictor variables

One of the outcome category is taken as baseline category

Develops models for estimating the probability that the output belongs to a category or not

# NOMINAL LOGISTIC REGRESSION

$$p_i = \frac{e^{a_i + b_{i1}x_1 + b_{i2}x_2 + --- + b_{ik}x_k}}{1 + e^{a_i + b_{i1}x_1 + b_{i2}x_2 + --- + b_{ik}x_k}}$$

$p_i$: probability that the output belongs to a particular category i
$x_i$'s : independent variables
$a_i$, $b_{i1}$, $b_{i2}$, ---: coefficients to be estimated

$$\sum_{i=1}^{n} p_i = 1$$

Where n is the number of categories

The predicted category is the one with highest probability

# NOMINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Read the data file and variables

```
> dd = mydata$DD
> effort = mydata$Effort
> coverage = mydata$Test.Coverage
```

# NOMINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Make one of the classes (say “Low”) of output variable as the baseline level

```
> library(nnet)
> dd = relevel(dd, ref = "Low“)
```

Develop the model and display summary result

```
> mymodel = multinom(dd ~ effort + coverage)
> result = summary(mymodel)
> result
```

# NOMINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Coefficients

| | (Intercept) | effort | coverage |
|---|---|---|---|
| High | 7.313595 | -0.0684 | -0.0882 |
| Medium | 3.865068 | -0.039 | -0.0485 |

Standard Errors

| | (Intercept) | effort | coverage |
|---|---|---|---|
| High | 1.326415 | 0.02498 | 0.02341 |
| Medium | 1.281011 | 0.02462 | 0.02235 |

# NOMINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Testing the significance of coefficients using z tests

```
> z = result$coefficients / result$standard.errors
> z
```

z statistics

| | (Intercept) | effort | coverage |
|---|---|---|---|
| High | 5.513807 | -2.740058 | -2.740058 |
| Medium | 3.017201 | -1.582807 | -2.172594 |

# NOMINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

## Computation of p value

```
> p = (1 - pnorm(abs(z), 0,1))*2
 > p
```

## P values

| | (Intercept) | effort | coverage |
|---|---|---|---|
| High | 0.0000 | 0.0061 | 0.00016 |
| Medium | 0.0000 | 0.1135 | 0.0298 |

Since p value < 0.05, all the factors have significant impact on defect density

# NOMINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Prediction Models

$$p(dd = High) = \frac{e^{7.3135 - 0.0684 \times Effort - 0.0882 \times Coverage}}{1 + e^{7.3135 - 0.0684 \times Effort - 0.0882 \times Coverage}}$$

$$p(dd = Medium) = \frac{e^{3.865 - 0.039 \times Effort - 0.0485 \times Coverage}}{1 + e^{3.865 - 0.039 \times Effort - 0.0485 \times Coverage}}$$

$$p(dd = Low) = 1 - P(dd = High) - P(dd = Medium)$$

Predicted class of dd will be k if p(dd = k) = max(P(dd= i)), i = Low, Medium or High

# NOMINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Predicted values

```
> pred = predict(mymodel)
> fit = fitted(mymodel)
>fit
> output = cbind(dd, pred)
>write.csv(output, "E:/Infosys/Part 2/output.csv")
```

# NOMINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Comparing Actual Vs Predicted

```
> mytable = table(dd, pred)
> mytable
> prop.table(mytable)
```

# NOMINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Comparing Actual Vs Predicted

| | | Predicted | | |
|---|---|---|---|---|
| Actual | | Low | High | Medium |
| | Low | 94 | 11 | 0 |
| | High | 23 | 27 | 0 |
| | Medium | 31 | 14 | 0 |

# NOMINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Comparing Actual Vs Predicted (in %)

| | | Predicted | | |
|---|---|---|---|---|
| Actual | | Low | High | Medium |
| | Low | 0.47 | 0.05 | 0.00 |
| | High | 0.12 | 0.14 | 0.00 |
| | Medium | 0.16 | 0.07 | 0.00 |

Accuracy = 0.47 + 0.14 + 0.00 = 0.61 = 61%

# SURVIVAL ANALYSIS

# SURVIVAL ANALYSIS

A collection of techniques for modeling the time to an event.

Data may be right censored

– the even may not have occurred till the end of the study period

- Incomplete information on an observation that up to certain time the event had not occured

Data are typically entered in

start time, end time and status( 1: event occurred, 0 : event did not occur) or

Time to event and status

# SURVIVAL ANALYSIS

Example 1: The data on time on treatment in months of a group of patients suffering from a particular disease is given in survival.csv file. The type of medicine (0: type 1, 1: type 2) and the age of the patient is also given in the file.

1. Compute the survival distribution (Kaplan – Meier estimator) of the patients?
2. Compare the survival distribution of patients who use medicine 1 and 2?
3. Is there any difference in the survival distribution of patients who treated with medicine and from those who treated with medicine 2?
4. Develop a model to predict the survival based on age?

# SURVIVAL ANALYSIS

Example 1: The data on time on treatment in months of a group of patients suffering from a particular disease is given in survival.csv file. The type of medicine (0: type 1, 1: type 2) and the age of the patient is also given in the file.

Reading the data and variables

```
> time = mydata$time

> age = mydata$age

> drug = mydata$drug

> status =mydata$censor
```

Create a survival object

```
>Library(survival)

>Survobject = Surv(time, status)

> survobject
```

# SURVIVAL ANALYSIS

Example 1: The data on time on treatment in months of a group of patients suffering from a particular disease is given in survival.csv file. The type of medicine (0: type 1, 1: type 2) and the age of the patient is also given in the file.

Computing Kaplan – Meier Estimator

```
> mymodel = survfit(survobject ~ 1)

> summary(mymodel)

> plot(mymodel)
```

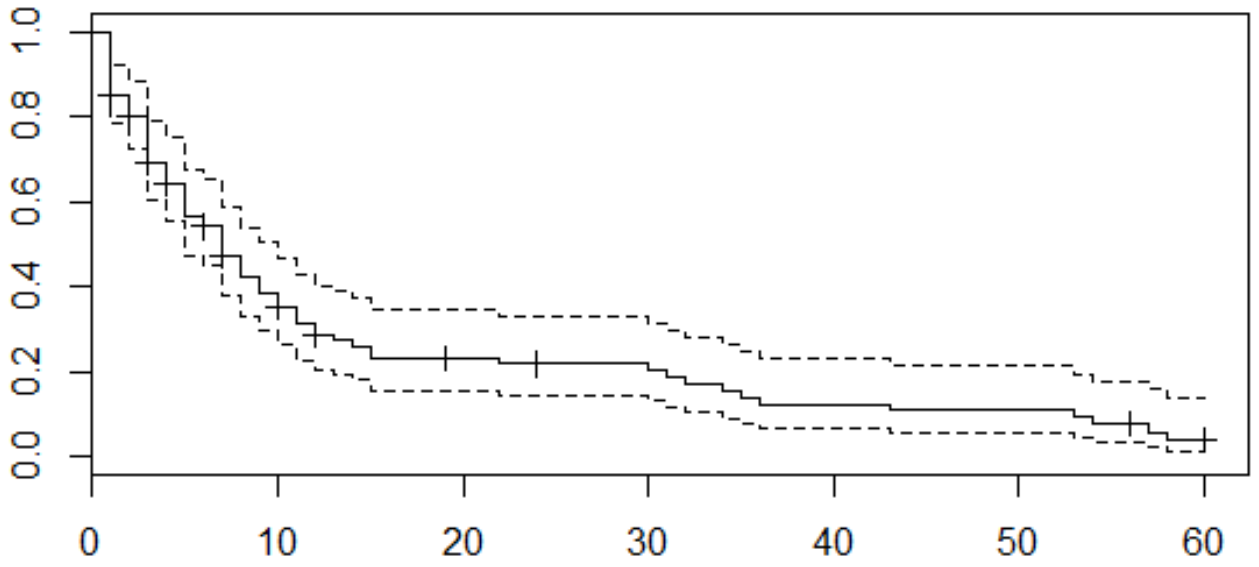
# SURVIVAL ANALYSIS

Example 1: The data on time on treatment in months of a group of patients suffering from a particular disease is given in survival.csv file. The type of medicine (0: type 1, 1: type 2) and the age of the patient is also given in the file.

| time | n.risk | n.event | survival | std.err | lower 95% CI | Upper 95% CI |
|---|---|---|---|---|---|---|
| 1 | 100 | 15 | 0.85 | 0.0357 | 0.7828 | 0.923 |
| 2 | 83 | 5 | 0.7988 | 0.0402 | 0.7237 | 0.882 |
| 3 | 73 | 10 | 0.6894 | 0.0473 | 0.6026 | 0.789 |
| 4 | 61 | 4 | 0.6442 | 0.0493 | 0.5544 | 0.748 |
| 5 | 56 | 7 | 0.5636 | 0.0517 | 0.4709 | 0.675 |
| 6 | 49 | 2 | 0.5406 | 0.0521 | 0.4476 | 0.653 |
| 7 | 46 | 6 | 0.4701 | 0.0526 | 0.3775 | 0.586 |
| 8 | 39 | 4 | 0.4219 | 0.0525 | 0.3306 | 0.538 |
| 9 | 35 | 3 | 0.3857 | 0.052 | 0.2962 | 0.502 |
| 10 | 32 | 3 | 0.3496 | 0.0511 | 0.2625 | 0.466 |
| 11 | 28 | 3 | 0.3121 | 0.05 | 0.228 | 0.427 |
| 12 | 25 | 2 | 0.2872 | 0.049 | 0.2055 | 0.401 |
| 13 | 21 | 1 | 0.2735 | 0.0486 | 0.1931 | 0.387 |
| 14 | 20 | 1 | 0.2598 | 0.048 | 0.1809 | 0.373 |
| 15 | 19 | 2 | 0.2325 | 0.0467 | 0.1568 | 0.345 |
| 22 | 16 | 1 | 0.2179 | 0.046 | 0.1441 | 0.33 |
| 30 | 14 | 1 | 0.2024 | 0.0453 | 0.1305 | 0.314 |
| 31 | 13 | 1 | 0.1868 | 0.0444 | 0.1173 | 0.298 |
| 32 | 12 | 1 | 0.1712 | 0.0433 | 0.1043 | 0.281 |
| 34 | 11 | 1 | 0.1557 | 0.0421 | 0.0916 | 0.264 |
| 35 | 10 | 1 | 0.1401 | 0.0407 | 0.0793 | 0.247 |
| 36 | 9 | 1 | 0.1245 | 0.039 | 0.0674 | 0.23 |
| 43 | 8 | 1 | 0.109 | 0.0371 | 0.0559 | 0.212 |
| 53 | 7 | 1 | 0.0934 | 0.0349 | 0.0449 | 0.194 |
| 54 | 6 | 1 | 0.0778 | 0.0324 | 0.0344 | 0.176 |
| 57 | 4 | 1 | 0.0584 | 0.0296 | 0.0216 | 0.157 |
| 58 | 3 | 1 | 0.0389 | 0.0253 | 0.0109 | 0.13 |

# SURVIVAL ANALYSIS

Example 1: The data on time on treatment in months of a group of patients suffering from a particular disease is given in survival.csv file. The type of medicine (0: type 1, 1: type 2) and the age of the patient is also given in the file.



# SURVIVAL ANALYSIS

Example 1: The data on time on treatment in months of a group of patients suffering from a particular disease is given in survival.csv file. The type of medicine (0: type 1, 1: type 2) and the age of the patient is also given in the file.

Comparison of survival distribution of patients with medicine 1 & 2

```
>mynewmodel = survfit(survobject ~ drug)
```

```
> summary(mynewmodel)
```

# SURVIVAL ANALYSIS

## Survival distribution of patients with medicine 1

| time | n.risk | n.event | survival | std.err | lower 95% CI | Upper 95% CI |
|---|---|---|---|---|---|---|
| 1 | 51 | 5 | 0.902 | 0.0416 | 0.8239 | 0.987 |
| 2 | 46 | 3 | 0.8431 | 0.0509 | 0.749 | 0.949 |
| 3 | 41 | 2 | 0.802 | 0.0561 | 0.6992 | 0.92 |
| 4 | 38 | 2 | 0.7598 | 0.0606 | 0.6498 | 0.888 |
| 5 | 35 | 3 | 0.6947 | 0.066 | 0.5766 | 0.837 |
| 6 | 32 | 1 | 0.673 | 0.0675 | 0.5529 | 0.819 |
| 7 | 31 | 1 | 0.6513 | 0.0687 | 0.5296 | 0.801 |
| 8 | 30 | 2 | 0.6078 | 0.0706 | 0.484 | 0.763 |
| 9 | 28 | 2 | 0.5644 | 0.072 | 0.4396 | 0.725 |
| 10 | 26 | 2 | 0.521 | 0.0727 | 0.3964 | 0.685 |
| 11 | 23 | 2 | 0.4757 | 0.0731 | 0.352 | 0.643 |
| 12 | 21 | 2 | 0.4304 | 0.0728 | 0.309 | 0.6 |
| 13 | 19 | 1 | 0.4077 | 0.0724 | 0.2879 | 0.577 |
| 14 | 18 | 1 | 0.3851 | 0.0718 | 0.2672 | 0.555 |
| 15 | 17 | 1 | 0.3624 | 0.0711 | 0.2468 | 0.532 |
| 22 | 15 | 1 | 0.3383 | 0.0703 | 0.225 | 0.508 |
| 30 | 13 | 1 | 0.3123 | 0.0696 | 0.2018 | 0.483 |
| 31 | 12 | 1 | 0.2862 | 0.0685 | 0.1791 | 0.457 |
| 32 | 11 | 1 | 0.2602 | 0.067 | 0.1571 | 0.431 |
| 34 | 10 | 1 | 0.2342 | 0.0652 | 0.1357 | 0.404 |
| 35 | 9 | 1 | 0.2082 | 0.0629 | 0.1151 | 0.376 |
| 36 | 8 | 1 | 0.1821 | 0.0602 | 0.0953 | 0.348 |
| 43 | 7 | 1 | 0.1561 | 0.0569 | 0.0764 | 0.319 |
| 53 | 6 | 1 | 0.1301 | 0.0531 | 0.0585 | 0.289 |
| 54 | 5 | 1 | 0.1041 | 0.0484 | 0.0418 | 0.259 |
| 57 | 4 | 1 | 0.0781 | 0.0427 | 0.0267 | 0.228 |
| 58 | 3 | 1 | 0.052 | 0.0355 | 0.0136 | 0.198 |

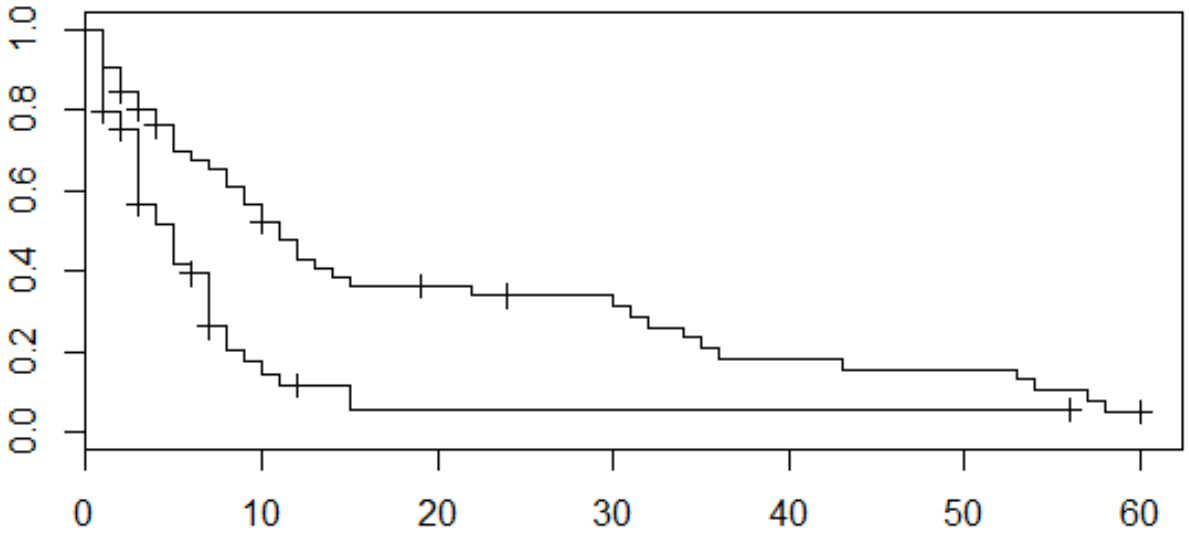
# SURVIVAL ANALYSIS

Survival distribution of patients with medicine 2

| time | n.risk | n.event | survival | std.err | lower 95% CI | Upper 95% CI |
|---|---|---|---|---|---|---|
| 1 | 49 | 10 | 0.7959 | 0.0576 | 0.6907 | 0.917 |
| 2 | 37 | 2 | 0.7529 | 0.062 | 0.6407 | 0.885 |
| 3 | 32 | 8 | 0.5647 | 0.074 | 0.4367 | 0.73 |
| 4 | 23 | 2 | 0.5156 | 0.0753 | 0.3872 | 0.686 |
| 5 | 21 | 4 | 0.4174 | 0.0753 | 0.2931 | 0.594 |
| 6 | 17 | 1 | 0.3928 | 0.0748 | 0.2705 | 0.57 |
| 7 | 15 | 5 | 0.2619 | 0.0691 | 0.1562 | 0.439 |
| 8 | 9 | 2 | 0.2037 | 0.0648 | 0.1092 | 0.38 |
| 9 | 7 | 1 | 0.1746 | 0.0618 | 0.0873 | 0.349 |
| 10 | 6 | 1 | 0.1455 | 0.0579 | 0.0667 | 0.317 |
| 11 | 5 | 1 | 0.1164 | 0.0531 | 0.0476 | 0.285 |
| 15 | 2 | 1 | 0.0582 | 0.049 | 0.0112 | 0.303 |

# SURVIVAL ANALYSIS

Comparison of survival distribution of patients with medicine 1 & 2



# SURVIVAL ANALYSIS

Testing for difference in survival distribution of patients used medicine 1 & 2

```
> survdiff(survobject ~ drug)
```

| Statistics | Value |
|---|---|
| Chi Square | 11.9 |
| Degrees of Freedom | 1 |
| P value | 0.000575 |

Since p value < 0.05, there is significant difference in the survival distribution of patients using medicine 1 & hat of medicine 2

# SURVIVAL ANALYSIS

Modeling survival in terms of age

>mymodel = coxph(survobject ~ age)

> mymodel

| | Coefficient | exp(coeff) | se(coeff) | z | p value |
|---|---|---|---|---|---|
| age | 0.0853 | 1.09 | 0.0174 | 4.89 | 0.00 |

| | |
|---|---|
| Total | 100 |
| Events | 80 |
| Likelihood ratio test | 23.3 |
| P value | 0.00 |

# SURVIVAL ANALYSIS

Modeling survival in terms of age

>mymodel = coxph(survobject ~ age)

> mymodel

| | rho | Chi sq | se(coeff) | p |
|---|---|---|---|---|
| age | 0.00129 | 0.00011 | 0.0174 | 0.992 |

# CLASSIFICATION *and* REGRESSION TREE

# CLASSIFICATION AND REGRESSION TREE

## Objective

To develop a predictive model to classify dependant or response metric (Y) in terms of independent or exploratory variables(Xs).

### When to Use

Xs : Continuous or discrete

Y : Discrete or continuous

# CLASSIFICATION AND REGRESSION TREE

## Classification Tree

When response Y is discrete

Method = “class”

## Regression Tree

When response Y is discrete

Method = “anova”

# CLASSIFICATION AND REGRESSION TREE

Classifies data (develops a model) based on the training data

Each sample is assumed to belong to a predefined class

Sample data set used for building the model is training set

Usage:

For classifying future or unknown data

# CLASSIFICATION AND REGRESSION TREE

Example:

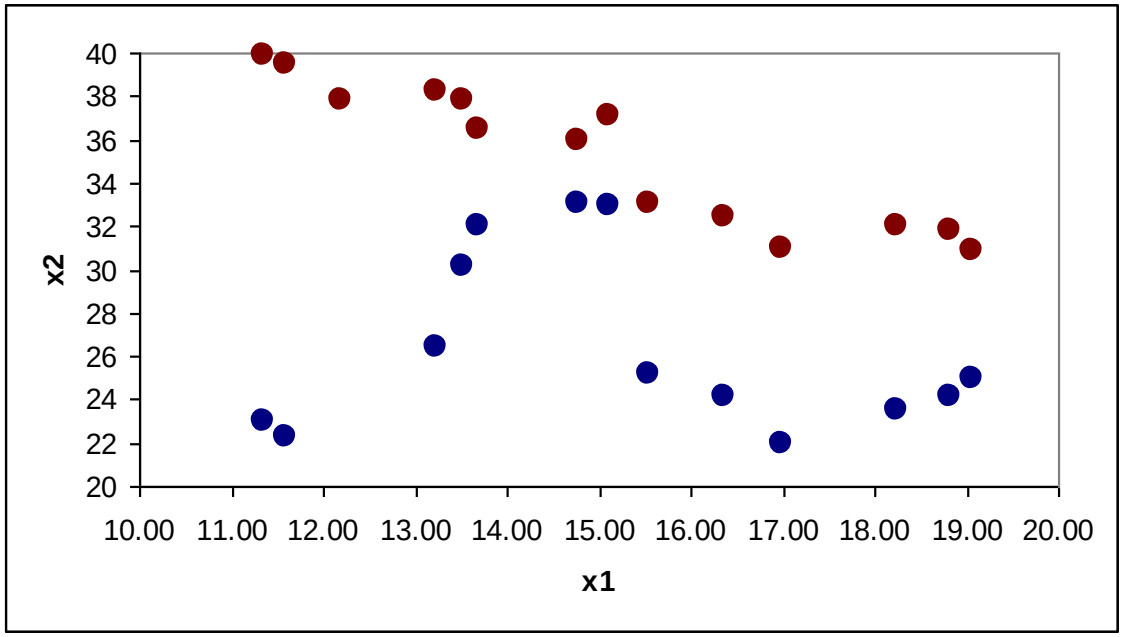
| | |
|---|---|
| Attribute 1 | x1 |
| Attribute 2 | x2 |
| Label : y | Y1 (Red) , y2 (Blue) |

| x1 | x2 | Y | x1 | x2 | Y |
|---|---|---|---|---|---|
| 11.35 | 23 | Blue | 11.85 | 39.9 | Red |
| 11.59 | 22.3 | Blue | 12.09 | 39.5 | Red |
| 12.19 | 24.5 | Blue | 12.69 | 37.8 | Red |
| 13.23 | 26.4 | Blue | 13.73 | 38.2 | Red |
| 13.51 | 30.2 | Blue | 14.01 | 37.8 | Red |
| 13.68 | 32 | Blue | 14.18 | 36.5 | Red |
| 14.78 | 33.1 | Blue | 15.28 | 36 | Red |
| 15.11 | 33 | Blue | 15.61 | 37.1 | Red |
| 15.55 | 25.2 | Blue | 16.05 | 33.1 | Red |
| 16.37 | 24.1 | Blue | 16.87 | 32.4 | Red |
| 16.99 | 22 | Blue | 17.49 | 31 | Red |
| 18.23 | 23.5 | Blue | 18.73 | 32 | Red |
| 18.83 | 24.1 | Blue | 19.33 | 31.8 | Red |
| 19.06 | 25 | Blue | 19.56 | 30.9 | Red |

# CLASSIFICATION AND REGRESSION TREE

Example:

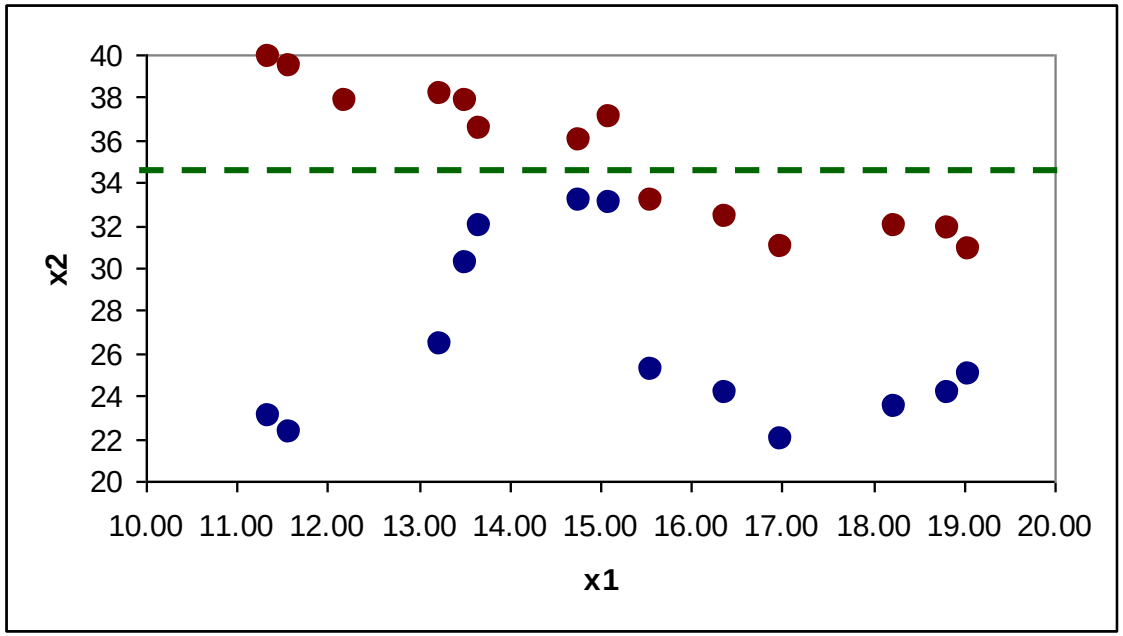
| | |
|---|---|
| Attribute 1 | x1 |
| Attribute 2 | x2 |
| Label : y | Y1 (Red) , y2 (Blue) |

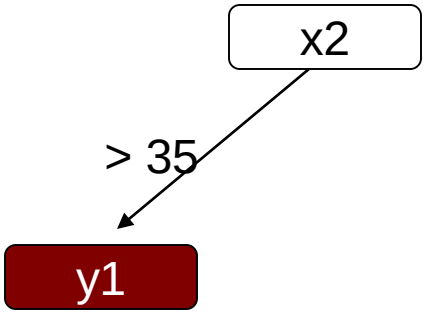


# CLASSIFICATION AND REGRESSION TREE

Example:

| | |
|---|---|
| Attribute 1 | x1 |
| Attribute 2 | x2 |
| Label : y | y1 (Red) , y2 (Blue) |

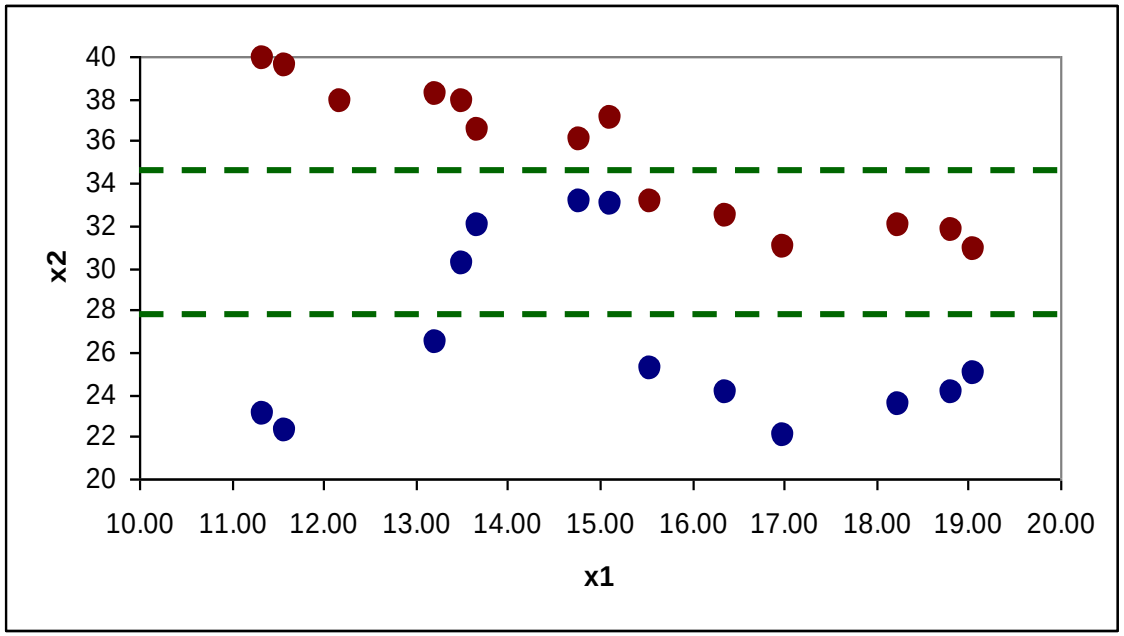


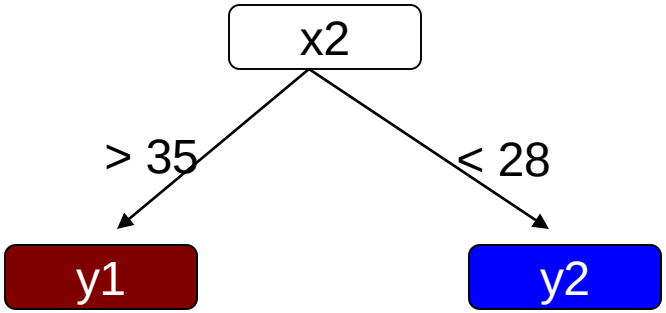



# CLASSIFICATION AND REGRESSION TREE

Example:

| | |
|---|---|
| Attribute 1 | x1 |
| Attribute 2 | x2 |
| Label : y | y1 (Red) , y2 (Blue) |

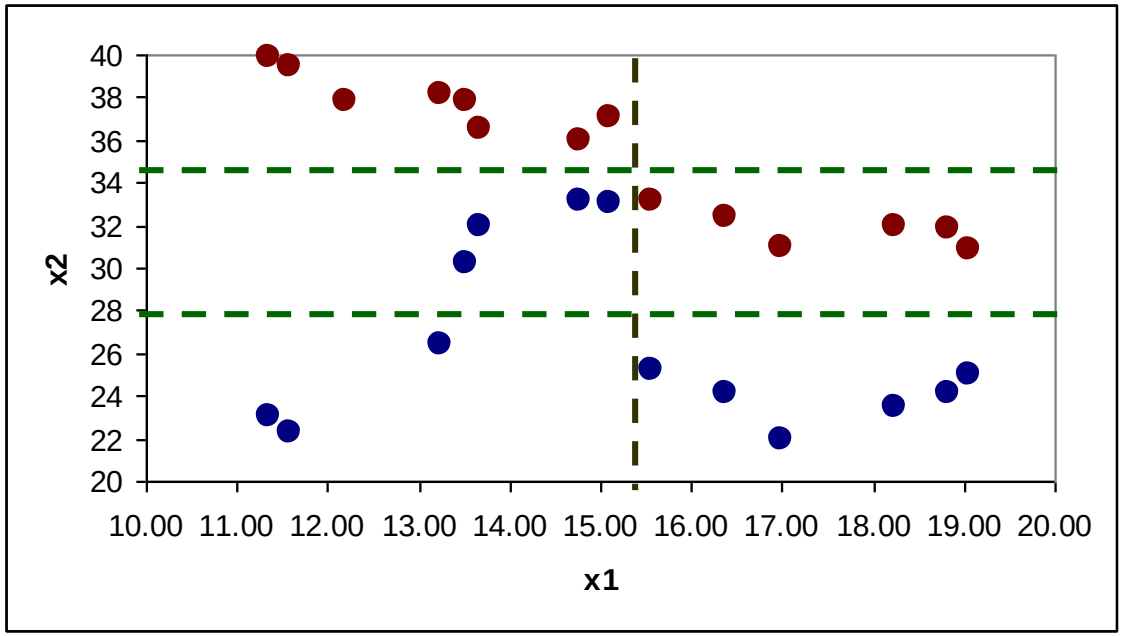


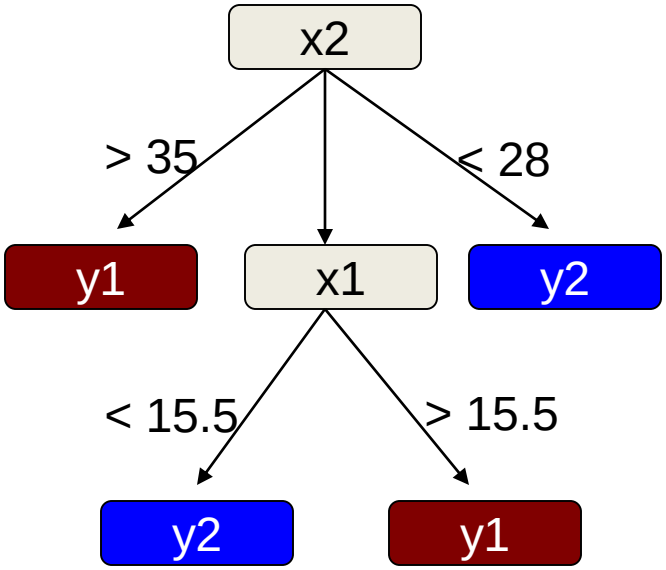



# CLASSIFICATION AND REGRESSION TREE

Example:

| | |
|---|---|
| Attribute 1 | x1 |
| Attribute 2 | x2 |
| Label : y | y1 (Red) , y2 (Blue) |



# CLASSIFICATION AND REGRESSION TREE

Example: Rules

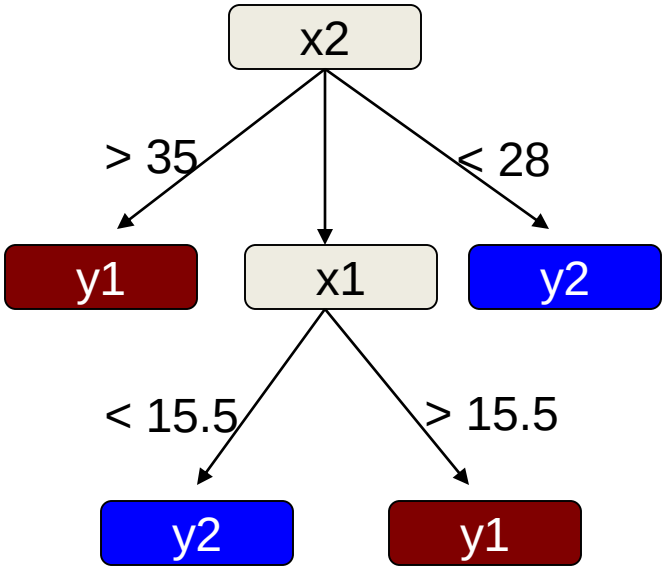
| | |
|---|---|
| Attribute 1 | x1 |
| Attribute 2 | x2 |
| Label : y | y1 (Red) , y2 (Blue) |

If x2 > 35 then y = y1

If x2 < 28, then y = y2

If 28 > x2 > 35 & x1 > 15.5, then y = y1

If 28 > x2 > 35 & x1 < 15.5, then y = y2



# CLASSIFICATION AND REGRESSION TREE

## Challenges

How to represent the entire information in the dataset using minimum number of rules?

How to develop the smallest tree?

## Solution

Select the variable with maximum information (highest relation with Y) for first split

# CLASSIFICATION AND REGRESSION TREE

Example: A marketing company wants to optimize their mailing campaign by sending the brochure mail only to those customers who responded to previous mail campaigns. The profile of customers are given below. Can you develop a rule to identify the profile of customers who are likely to respond (Mail_Respond.csv?

| SL No | District | House Type | Income | Previous_Customer | Outcome |
|---|---|---|---|---|---|
| 1 | Suburban | Detached | High | No | No Response |
| 2 | Suburban | Detached | High | Yes | No Response |
| 3 | Rural | Detached | High | No | Responded |
| 4 | Urban | Semi-detached | High | No | Responded |
| 5 | Urban | Semi-detached | Low | No | Responded |
| 6 | Urban | Semi-detached | Low | Yes | No Response |
| 7 | Rural | Semi-detached | Low | Yes | Responded |
| 8 | Suburban | Terrace | High | No | No Response |
| 9 | Suburban | Semi-detached | Low | No | Responded |
| 10 | Urban | Terrace | Low | No | Responded |
| 11 | Suburban | Terrace | Low | Yes | Responded |
| 12 | Rural | Terrace | High | Yes | Responded |
| 13 | Rural | Detached | Low | No | Responded |
| 14 | Urban | Terrace | High | Yes | No Response |

# CLASSIFICATION AND REGRESSION TREE

Example: A marketing company wants to optimize their mailing campaign by sending the brochure mail only to those customers who responded to previous mail campaigns. The profile of customers are given below? Can you develop a rule to identify the profile of customers who are likely to respond?

Number of variables = 4

| SL No | Variable Name | Number of values |
|---|---|---|
| 1 | District | 3 |
| 2 | House Type | 3 |
| 3 | Income | 2 |
| 4 | Previous Customer | 2 |

Total Combination of Customer Profiles = 3 x 3 x 2 x 2 = 36

# CLASSIFICATION AND REGRESSION TREE

Read file and variables

```
> mydata = Mail_Respond
> house = mydata$House_Type
> district = mydata$District
>  income = mydata$Income
> prev = mydata$Previous_Customer
> outcome = mydata$Outcome
```

# CLASSIFICATION AND REGRESSION TREE

Develop the model

```
> library(rpart)

> mymodel = rpart( outcome ~ district + house + income + prev, method = "class",
control = rpart.control(minsplit = 2))
```

Note: When response is categorical, method = "class", when response is numeric, methos = "anova"

```
>print(mymodel)
```

# CLASSIFICATION AND REGRESSION TREE

1) root 14 5 Responded (0.3571429 0.6428571)
    2) dist=Suburban,Urban 10 5 No Response (0.5000000 0.5000000)
        4) income=High 5 1 No Response (0.8000000 0.2000000)
            8) house=Detached,Terrace 4 0 No Response (1.0000000 0.0000000) *
            9) house=Semi-detached 1 0 Responded (0.0000000 1.0000000) *

        5) income=Low 5 1 Responded (0.2000000 0.8000000)
            10) prev=Yes 2 1 No Response (0.5000000 0.5000000)
                20) dist=Urban 1 0 No Response (1.0000000 0.0000000) *
                21) dist=Suburban 1 0 Responded (0.0000000 1.0000000) *

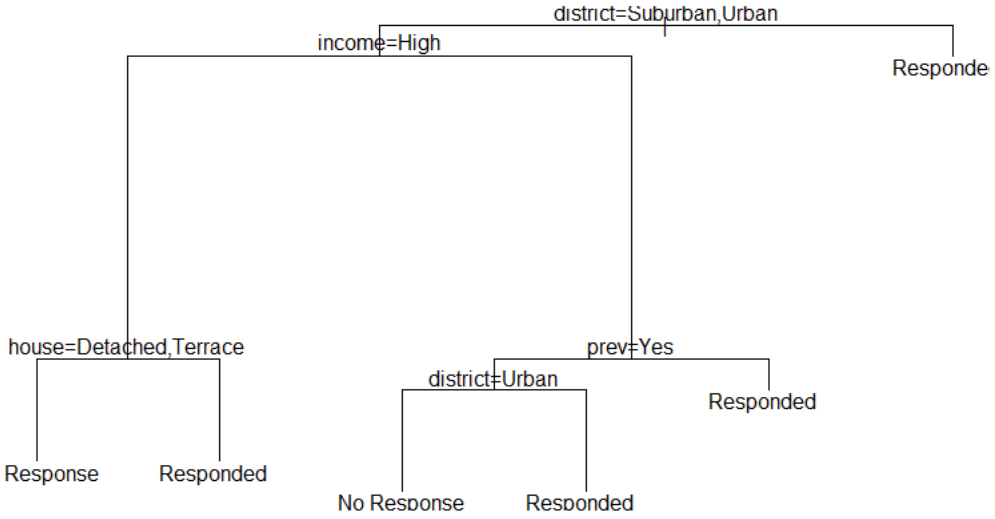
            11) prev=No 3 0 Responded (0.0000000 1.0000000) *

    3) dist=Rural 4 0 Responded (0.0000000 1.0000000) *

# CLASSIFICATION AND REGRESSION TREE

## Plot the tree

```
> plot(mymodel)
> text(mymodel, pretty = 0)
```



# CLASSIFICATION AND REGRESSION TREE

## Making predictions

```
> pred = predict(mymodel)
> Predclass = ifelse(pred[,1] > 0.5, "1", "2")
> mytable = table(outcome, predclass)
```

| | | Predicted | |
|---|---|---|---|
| | | Respond | No Respond |
| Outcome | Respond | 9 | 0 |
| | No Respond | 0 | 5 |

# CLASSIFICATION AND REGRESSION TREE

Exercise 1: Develop a tree based model for predicting whether the customer will take pep using ghe customer profile data given in bank-data.csv?

Exercise 2: Develop a tree based model for predicting conversion using temperature, time and kappa number as factors. The data is given in Mult_Reg_Conversion.csv?

# CLASSIFICATION AND REGRESSION TREE

## Random Forest

Improves predictive accuracy

Generates large number of bootstrapped trees

Classifies a new case using each tree in the new forest of trees

Final predicted outcome by combining the results across all of the trees

Regression tree – average

Classification tree – majority vote

# CLASSIFICATION AND REGRESSION TREE

## Random Forest

### Example

Develop a model to predict plant class using the data given in Iris.csv.\ using random forest method. Validate the model with Iris_test.csv data?

# CLASSIFICATION AND REGRESSION TREE

## Random Forest

### Example

Develop a model to predict plant class using the data given in Iris.csv.\ using random forest method. Validate the model with Iris_test.csv data?

# CLASSIFICATION AND REGRESSION TREE

## Random Forest

### Example

Develop a model to predict plant class using the data given in Iris.csv.\ using random forest method. Validate the model with Iris_test.csv data?

Read Iris data to mydata

```
> library(randomForest)
> mymodel = randomForest(Class ~ sepal.length + sepal.width + petal.length + petal.width, data
= mydata)
> mymodel
```

# CLASSIFICATION AND REGRESSION TREE

## Random Forest

### Example

Develop a model to predict plant class using the data given in Iris.csv.\ using random forest method. Validate the model with Iris_test.csv data?

| | Iris-setosa | Iris-versicolor | Iris-virginica | class.error |
|---|---|---|---|---|
| Iris-setosa | 50 | 0 | 0 | 0 |
| Iris-versicolor | 0 | 47 | 3 | 0.06 |
| Iris-virginica | 0 | 3 | 47 | 0.06 |

# CLASSIFICATION AND REGRESSION TREE

## Random Forest

### Example

Develop a model to predict plant class using the data given in Iris.csv.\ using random forest method. Validate the model with Iris_test.csv data?

### Model Validation

Read data to new data

```
> newdata <- read.csv("E:/Infosys/Part 2/Data/Iris_test.csv")
> pred = predict(mymodel, newdata = newdata)
> mytable = table(newdata$Class, pred)
> mytable
```

# CLASSIFICATION AND REGRESSION TREE

## Random Forest

### Example

Develop a model to predict plant class using the data given in Iris.csv.\ using random forest method. Validate the model with Iris_test.csv data?

| | Predicted | | |
|---|---|---|---|
| Actual | Iris-setosa | Iris-versicolor | Iris-virginica |
| Iris-setosa | 49 | 0 | 0 |
| Iris-versicolor | 0 | 15 | 0 |
| Iris-virginica | 0 | 0 | 2 |

# CLASSIFICATION AND REGRESSION TREE

## Random Forest

### Example

Develop a model to predict plant class using the data given in Iris.csv.\ using random forest method. Validate the model with Iris_test.csv data?

| | Predicted | | |
|---|---|---|---|
| Actual | Iris-setosa | Iris-versicolor | Iris-virginica |
| Iris-setosa | 74.24 | 0.00 | 0.00 |
| Iris-versicolor | 0.00 | 22.73 | 0.00 |
| Iris-virginica | 0.00 | 0.00 | 3.03 |

# ORDINAL LOGISTIC REGRESSION

# ORDINAL LOGISTIC REGRESSION

Used to develop models when the output or response variable y is prdinall

The output variable will be categorical, having more than two categories

# ORDINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Read the data file and variables

```
> dd = mydata$DD
> effort = mydata$Effort
> coverage = mydata$Test.Coverage
```

# ORDINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Make one of the classes (say “Low”) of output variable as the baseline level

```
> library(MASS)
> mymodel = polr(dd ~ effort + coverage)
> summary(mymodel)
```

# ORDINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

## Coefficients

| effort | coverage |
|---|---|
| 0.0234 | 0.0257 |

## Intercepts

| High \| Low | Low \| Medium |
|---|---|
| 1.4947 | 3.925 |

# ORDINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Predicted values

```
> pred = predict(mymodel)
> fit = fitted(mymodel)
>fit
> output = cbind(dd, pred)
>write.csv(output, "E:/Infosys/Part 2/output.csv")
```

# ORDINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Comparing Actual Vs Predicted

```
> mytable = table(dd, pred)
> mytable
> prop.table(mytable)
```

# ORDINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Comparing Actual Vs Predicted

| | | Predicted | | |
|---|---|---|---|---|
| Actual | | High | Low | Medium |
| | High | 8 | 42 | 0 |
| | Low | 0 | 105 | 0 |
| | Medium | 1 | 44 | 0 |

# ORDINAL LOGISTIC REGRESSION

Example 1: The data on system test defect density along with testing effort and test coverage is given in ST_Defects.csv. The defect density is classified as Low Medium High. Develop a model to estimate the system testing defect density class based on testing effort and test coverage ?

Comparing Actual Vs Predicted (in %)

| | | Predicted | | |
|---|---|---|---|---|
| Actual | | High | Low | Medium |
| | High | 4.0 | 21.0 | 0.00 |
| | Low | 0.00 | 52.50 | 0.00 |
| | Medium | 0.50 | 22.0 | 0.00 |

Accuracy = 4 + 52.5 + 0.00 = 0.565 = 56.5%