Introduction to

Binary Logistic Regression - I

Contents

Binary Logistic Regression

DEPENDENT VARIABLE

Categorical (Binary)

0 or 1
Death or Survival
Absence or Presence
⋮

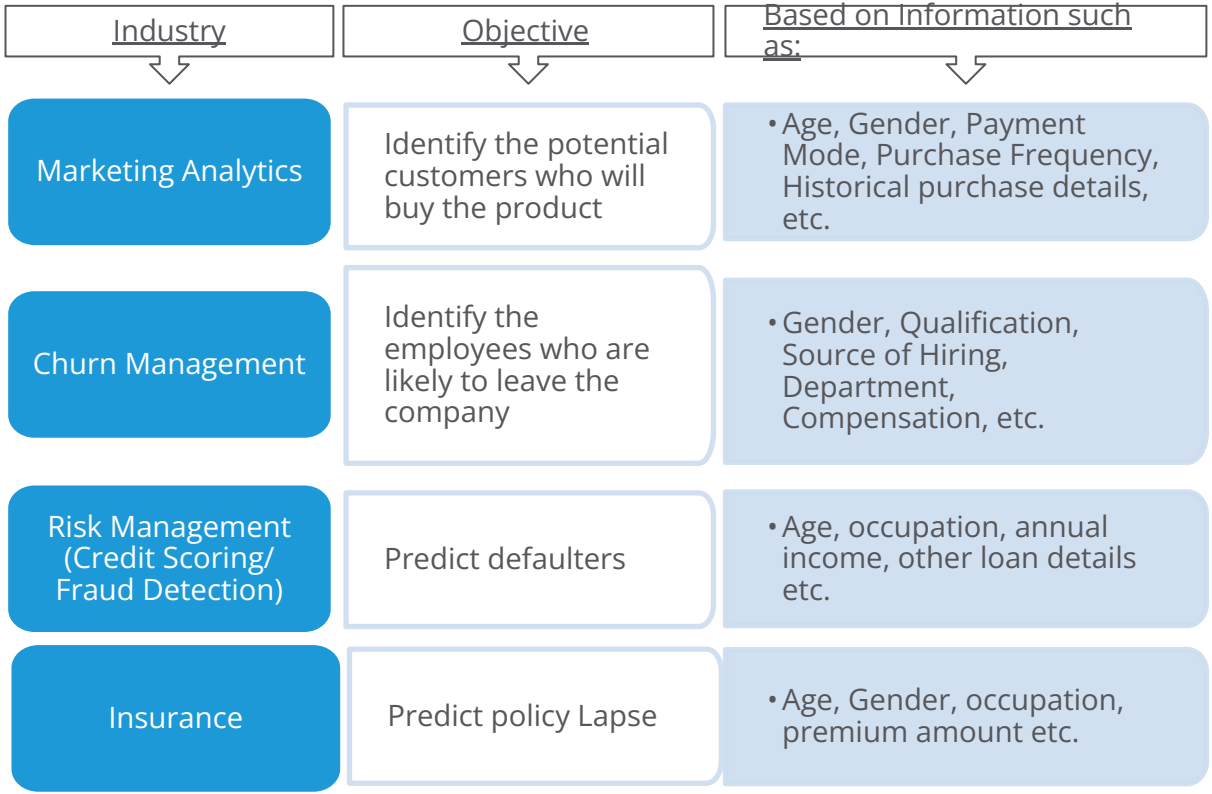
INDEPENDENT VARIABLE

Categorical or Continuous

Gender
Geographical Regions
Age
Income
Debt Ratio
⋮

Binary logistic regression **models the dependent variable as a logit of p, where p is the probability that dependent variable takes value 1**

Application Areas



Why Not Use Linear Regression Model?

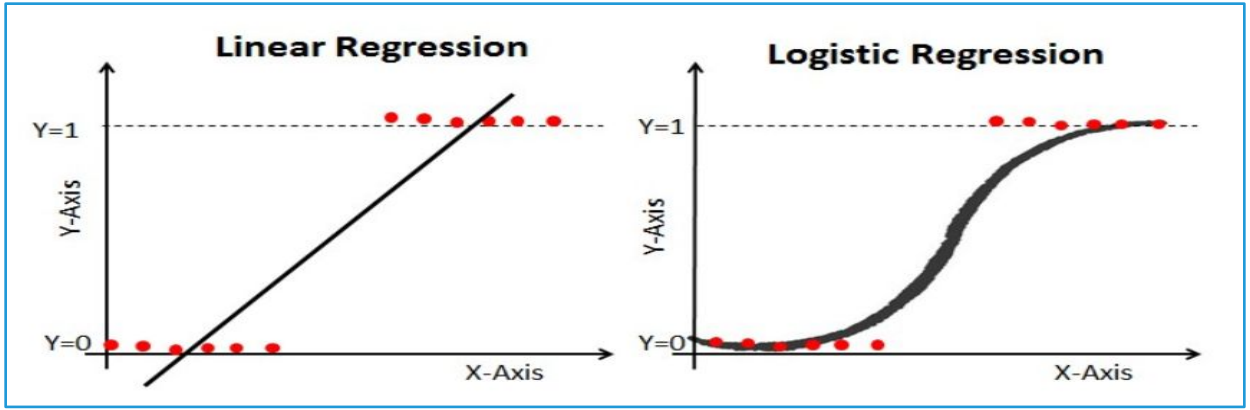
The statistical model for multiple linear regression is,

$$Y = b_0 + b_1X_1 + b_2X_2 + \ \ldots + b_pX_p + e$$

- If binary variable Y is used on left hand side of the model, then the two sides are not comparable. Right hand side is a continuous term.
- If probability 'P' is used instead of Y then linearity may not hold true. The relationship assumed in logistic regression is a 'S' shaped curve.

Why Not Use Linear Regression Model?

- Linear regression is suitable for predicting outcome which is continuous value.
  For example, predicting the price of a property based on area in Sq. Feet.
- The regression line is a **straight line**.
- Whereas logistic regression is for classification problems, which predicts a probability range between 0 to 1 (or predicts categories Yes or no).
  For example, predict whether a customer will make a purchase or not.
- The regression curve is a **sigmoid curve**.



Statistical Model

Statistical model for single predictor

$$p = \frac{e^{b_0 + b_1 x_1}}{1 + e^{b_0 + b_1 x_1}}$$

p is the Pr[Y=1/X] and X is the independent variable

$$1 - p = 1 - \frac{e^{b_0 + b_1 x_1}}{1 + e^{b_0 + b_1 x_1}} = \frac{1}{1 + e^{b_0 + b_1 x_1}}$$

$$\frac{p}{1 - p} = e^{b_0 + b_1 x_1} \rightarrow \log\left(\frac{p}{1 - p}\right) = b_0 + b_1 X_1$$

The left hand side uses 'link function'

Statistical Model – For k Predictors

The model can be extended for k independent variables

$$\log\left(\frac{p}{1-p}\right) = b_0 + b_1X_1 + \cdots + b_kX_k$$

where,

p	:	Probability that Y=1 given X
Y	:	Dependent Variable
$X_1, X_2, \ldots, X_k$	:	Independent Variables
$b_0, b_1, \ldots, b_k$	:	Parameters of Model

Note that **LHS of the model can lie between - ∞ to ∞**

Parameters of the model are estimated by Maximum Likelihood Method

Binary regression model is used to derive predicted probability of outcome. Error, by definition, is the difference between observed and predicted value. There is no such thing as comparable "observed probability" and hence **the model does not have any error component.**

Case Study – Modeling Loan Defaults

Background

- A bank possesses demographic and transactional data of its loan customers. If the bank has a model to predict defaulters it can help in loan disbursal decision making.

Objective

- To predict whether the customer applying for the loan will be a defaulter or not.

Available Information

- Sample size is 700
- **Independent Variables**: Age group, Years at current address, Years at current employer, Debt to Income Ratio, Credit Card Debts, Other Debts. The information on predictors was collected at the time of loan application process.
- **Dependent Variable**: Defaulter (=1 if defaulter ,0 otherwise). The status is observed after loan is disbursed.

Data Snapshot

Bank Loan Data

Independent Variables

Dependent Variable

SN	AGE	EMPLOY	ADDRESS	DEBTINC	CREDDEBT	OTHDEBT	DEFAULTE
1	3	17	12	9.3	11.36	5.01	1
2	1	10	6	17.3	1.36	4	0
3	2	15	14	5.5	0.86	2.17	0
4	3	15	14	2.9	2.66	0.82	0

Column	Description	Type	Measurement	Possible Values
SN	Serial Number		-	-
AGE	Age Groups	Categorical	1(<28 years), 2(28-40 years), 3(>40 years)	3
EMPLOY	Number of years customer working at current employer	Continuous	-	Positive value
ADDRESS	Number of years customer staying at current address	Continuous	-	Positive value
DEBTINC	Debt to Income Ratio	Continuous	-	Positive value
CREDDEBT	Credit Card Debt	Continuous	-	Positive value
OTHDEBT	Other Debt	Continuous	-	Positive value
DEFAULTER	Whether customer defaulted on loan	Binary	1(Defaulter), 0(Non-Defaulter)	2

Exploratory Data Analysis

- Before moving to modeling we can undertake some exploratory data analysis
- Depending upon the type of variable (Whether continuous or categorical) we can perform bivariate analysis

Cross Table 1: Relationship Between Defaulter and Age Group

Age Group	Defaulter	
	Yes	No
1	86	156
2	61	223
3	36	138

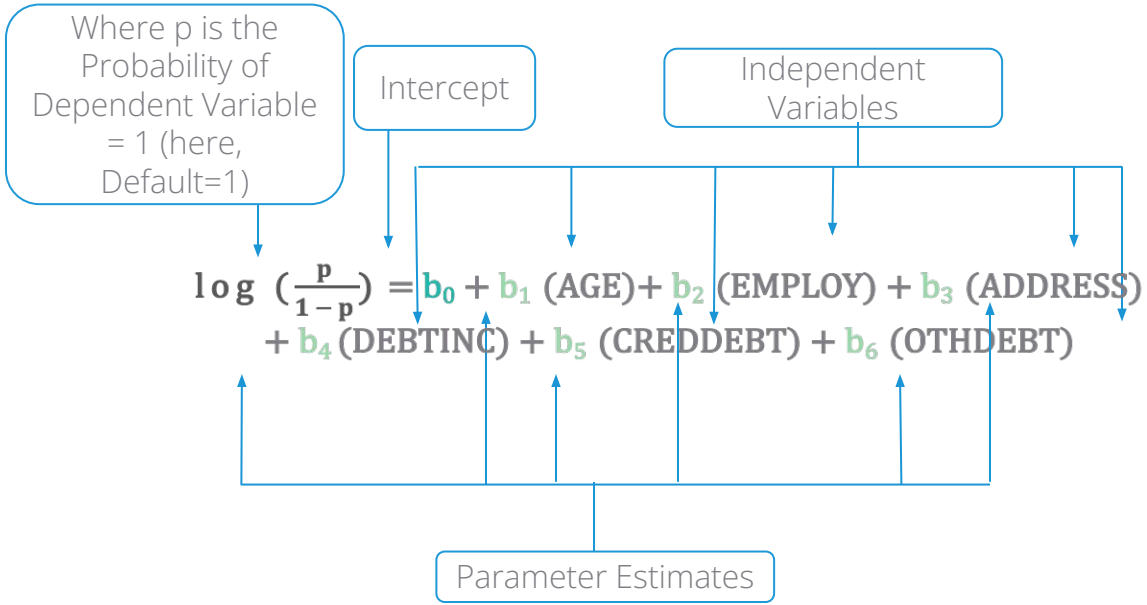
Cross Table 2: Transactional Behaviour of Defaulters v/s Non Defaulters

Types of Liabilities	**Average Liabilities**	
	Defaulter	Non-Defaulter
Credit Card Debt	2.42	1.25
Debt To Income Ratio	14.73	8.68
Other Debt	3.86	2.77

- Such data insights are also an important aspect of modeling

Binary Logistic Regression Model for the bank loan data

- Model of default on the predictors will look like this:



* Note that, this is not the final model. The equation is showed just for the understanding purpose. Only the significant variables will be part of the model.

Likelihood Function

- The parameters of the logistic model are estimated using **maximum likelihood estimation (MLE)**.
- The Likelihood function is as below:

$$L = \prod_{i=1}^{n} p^{y_i}(1-p)^{1-y_i}$$

n is the number of observations

- The likelihood function is a **joint probability** of Yi's.
- It is expressed as a function of regression parameters after substituting known X and Y value.
- Parameters are estimated by maximizing L.
- Two commonly used iterative maximum likelihood algorithms are **Fisher scoring method** and **Newton-Raphson method**. Both algorithms give the same parameter estimates; however, the estimated covariance matrix of the parameter estimators can differ slightly.

Maximum Likelihood Estimates of Parameters

	Coefficients
Intercept	-0.78821
AGE2	0.25202
AGE3	0.62707
EMPLOY	-0.26172
ADDRESS	-0.09964
DEBTINC	0.08506
CREDDEBT	0.56336
OTHDEBT	0.02315

log ((p/(1-p)) = -0.78821+ 0.25202 (AGE2)+ 0.62707 (AGE3)
-0.26172 (EMPLOY) - 0.09964 (ADDRESS) + 0.08506(DEBTINC) + 0.56336(CREDDEBT) +
0.02315 (OTHDEBT)

Parameters, Probability and Odds

Odds = Probability of Success (p) / Probability of Failure (1-p)

- In binary logistic regression, **model LHS is logit(p),** which is the **log of odds.** Hence, estimated parameter gives the **change in log of odds given one unit change in the independent variable.**

  Estimated coefficient of EMPLOY is -0.26172. This means that one unit change in EMPLOY will result in a change of -0.26172 in log of odds. The negative sign implies customer with a relatively steady job, is less likely to default.
- In order to get a more **straightforward and usable association between the independent and dependent variables,** Odds Ratio is calculated.

What is Odds Ratio

- Odds Ratio is a **measure of association between the independent variable and the outcome.**
- It represents the factor by which the **odds (event) change for a one-unit change in the independent variable.**

The odds of outcome being present when $X = x$ is $e^{b_0+b_1x}$

The odds of outcome being present when $X = x+1$ is $e^{b_0+b_1(x+1)}$

Odds Ratio

$$\frac{e^{b_0+b_1(x+1)}}{e^{b_0+b_1x}}$$

$$= e^{b_1} = \mathrm{EXP}(b_1)$$

Odds Ratio – Case Study

	Coefficients	Odds Ratio
Intercept	-0.78821	0.4546572
AGE2	0.25202	1.2866254
AGE3	0.62707	1.8721087
EMPLOY	**-0.26172**	**0.7697228**
ADDRESS	-0.09964	0.9051601
DEBTINC	0.08506	1.0887859
CREDDEBT	0.56336	1.7565703
OTHDEBT	0.02315	1.0234175

- When association between dependent and independent variable is
  - Positive: OR > 1
  - Negative: OR < 1
- OR = 1 indicates no association between variables

- For one unit change in EMPLOY the odds of default will change by 0.7697228 folds.

Individual testing using Wald's test

Individual testing is used for checking significance of each independent variable separately.

Objective	To test the **null hypothesis** that **each variable is insignificant**

Null Hypothesis (H_0): $b_i = 0$

Alternate Hypothesis (H_1): : $bi \neq 0$

i=1,2...,k

Test Statistic	Z =(Estimate of bi)/(Standard Error of estimated bi) Under H0, Z is assumed to follow standard normal distribution.
Decision Criteria	Reject the null hypothesis **if p-value < 0.05**

* Note that, few softwares like SAS provide Wald's chi square since, $z2 \sim \chi2(1)$)

Quick Recap

In this session, we learned about **Binary Logistic Regression** :

Binary logistic regression	• Dependent variable is binary and independent variables are categorical or continuous or mix of both. • Regression line is sigmoid curve. • Parameters are estimated using MLE.
ODDS Ratio	• measure of association between the independent variable and the outcome.

Introduction to

Binary Logistic Regression - II

Contents

Data Snapshot

Bank Loan Data

Independent Variables

Dependent Variable

SN	AGE	EMPLOY	ADDRESS	DEBTINC	CREDDEBT	OTHDEBT	DEFAULTE
1	3	17	12	9.3	11.36	5.01	1
2	1	10	6	17.3	1.36	4	0

Column	Description	Type	Measurement	Possible Values
SN	Serial Number		-	-
AGE	Age Groups	Categorical	1(<28 years), 2(28-40 years), 3(>40 years)	3
EMPLOY	Number of years customer working at current employer	Continuous	-	Positive value
ADDRESS	Number of years customer staying at current address	Continuous	-	Positive value
DEBTINC	Debt to Income Ratio	Continuous	-	Positive value
CREDDEBT	Credit Card Debt	Continuous	-	Positive value
OTHDEBT	Other Debt	Continuous	-	Positive value
DEFAULTER	Whether customer defaulted on loan	Binary	1(Defaulter), 0(Non-Defaulter)	2

Binary Logistic Regression in R

```
# Import data and check data structure before running model
data<-read.csv("BANK LOAN.csv",header=TRUE)
str(data)
```

```
# Output:
$ SN       : int  1 2 3 4 5 6 7 8 9 10 ...
$ AGE      : int  3 1 2 3 1 3 2 3 1 2 ...
$ EMPLOY   : int  17 10 15 15 2 5 20 12 3 0 ...
$ ADDRESS  : int  12 6 14 14 0 5 9 11 4 13 ...
$ DEBTINC  : num  9.3 17.3 5.5 2.9 17.3 10.2 30.6 3.6 24.4 19.7 ...
$ CREDDEBT : num  11.36 1.36 0.86 2.66 1.79 ...
$ OTHDEBT  : num  5.01 4 2.17 0.82 3.06 ...
$ DEFAULTER: int  1 0 0 0 1 0 0 0 1 0 ...
```

```
data$AGE<-factor(data$AGE)
str(data)
```

```
# Output:
'data.frame':   700 obs. of  8 variables:
$ SN       : int  1 2 3 4 5 6 7 8 9 10 ...
$ AGE      : Factor w/ 3 levels "1","2","3": 3 1 2 3 1 3 2 3 1 2 ...
$ EMPLOY   : int  17 10 15 15 2 5 20 12 3 0 ...
$ ADDRESS  : int  12 6 14 14 0 5 9 11 4 13 ...
$ DEBTINC  : num  9.3 17.3 5.5 2.9 17.3 10.2 30.6 3.6 24.4 19.7 ...
$ CREDDEBT : num  11.36 1.36 0.86 2.66 1.79 ...
$ OTHDEBT  : num  5.01 4 2.17 0.82 3.06 ...
$ DEFAULTER: int  1 0 0 0 1 0 0 0 1 0 ...
```

Age is an integer and needs to be converted into a factor, since, it is a categorical variable.

Logistic Regression in R

```
# Using glm function to develop binary logistic regression model

riskmodel<-glm(DEFAULTER~AGE+EMPLOY+ADDRESS+DEBTINC+CREDDEBT+OTHDEBT,
               family=binomial,data=data)
```

- **glm** is Generalized Linear Model. Logistic regression is type of GLM.
- LHS of ~ is the dependent variable and independent variables on RHS are separated by '+'.
- **riskmodel** is the model object
- By setting the **family =binomial**, **glm()** it fits a logistic regression model

Individual Hypothesis Testing in R

```
# Individual Testing
summary(riskmodel)
# Output:
```

- **summary()** function gives the output of glm.

```
Call:
glm(formula = DEFAULTER ~ AGE + EMPLOY + ADDRESS + DEBTINC +
    CREDDEBT + OTHDEBT, family = binomial, data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.3495  -0.6601  -0.2974   0.2509   2.8583

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.78821    0.26407  -2.985  0.00284 **
AGE2         0.25202    0.26651   0.946  0.34433
AGE3         0.62707    0.36056   1.739  0.08201 .
EMPLOY      -0.26172    0.03188  -8.211  < 2e-16 ***
ADDRESS     -0.09964    0.02234  -4.459 8.22e-06 ***
DEBTINC      0.08506    0.02212   3.845  0.00012 ***
CREDDEBT     0.56336    0.08877   6.347 2.20e-10 ***
OTHDEBT      0.02315    0.05709   0.405  0.68517
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 804.36  on 699  degrees of freedom
Residual deviance: 553.41  on 692  degrees of freedom
AIC: 569.41

Number of Fisher Scoring iterations: 6
```

Interpretation :

- Since p-value is <0.05 for Employ, Address, Debtinc, Creddebt, these independent variables are significant.

Individual Testing in R

- Validating the signs of coefficients:

  - Once the coefficients are obtained, they are checked for their signs based on business logic. Variable should be reconsidered if its sign does not match with the business logic.

  - For Ex. in our case study, sign of coefficient of Debtinc is positive which indicates that if debt to income ratio increases, chances of default increases.

Re-run Model in R

- Once variables to be retained are finalized ,re-run the model with these final variables and obtain revised coefficients for the model.
- Re-run the model with employ, address, debtinc, creddebt.

```
riskmodel<-glm(DEFAULTER~EMPLOY+ADDRESS+DEBTINC+CREDDEBT,
               family=binomial,data=data)

summary(riskmodel)
```

Re-run Model in R

```
# Output:

Call:
glm(formula = DEFAULTER ~ EMPLOY + ADDRESS + DEBTINC + CREDDEBT,
    family = binomial, data = data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.4483  -0.6396  -0.3108   0.2583   2.8496

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -0.79107    0.25154  -3.145  0.00166 **
EMPLOY      -0.24258    0.02806  -8.646  < 2e-16 ***
ADDRESS     -0.08122    0.01960  -4.144 3.41e-05 ***
DEBTINC      0.08827    0.01854   4.760 1.93e-06 ***
CREDDEBT     0.57290    0.08725   6.566 5.17e-11 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 804.36  on 699  degrees of freedom
Residual deviance: 556.74  on 695  degrees of freedom
AIC: 566.74

Number of Fisher Scoring iterations: 6
```

Interpretation :

- Since p-value is <0.05 for Employ, Address, Debtinc, Creddebt, these independent variables are significant and sign of the coefficients are also logical.

Final Model

- Final Model is :

$$\log\left(\frac{p}{1-p}\right) = -0.79107 - 0.24258 * (\text{EMPLOY}) - 0.08122 * (\text{ADDRESS}) + 0.08827 * (\text{DEBTINC}) + 0.57290\ (\text{CREDDEBT})$$

- This model is used for predicting the probabilities.

Odds Ratio in R

```
coef(riskmodel)
exp(coef(riskmodel))
exp(confint(riskmodel))
cbind(coef(riskmodel),odds_ratio=exp(coef(riskmodel)),exp(confint
(riskmodel)))
```

- **coef(riskmodel):** identify the model coefficients.
- **exp(coef(riskmodel)):** find odds ratio.
- **exp(confint(riskmodel)):** calculates confidence interval for odds ratio.

Odds Ratio in R

```
# Output:
                          odds_ratio     2.5 %    97.5 %
(Intercept) -0.79107079  0.4533591 0.2756574 0.7400939
EMPLOY      -0.24258492  0.7845971 0.7408645 0.8271278
ADDRESS     -0.08122146  0.9219895 0.8863345 0.9572345
DEBTINC      0.08826530  1.0922779 1.0536134 1.1332029
CREDDEBT     0.57289682  1.7733968 1.5097676 2.1242860
```

Interpretation :

- Note that, confidence interval for odds ratio does not include ‘1’ for all variables retained in the model. Which means that all of these variables are significant.
- The odds ratio for CREDDEBT is approximately 1.77
- For one unit change CREDDEBT, the odds of being a defaulter will change by 1.77 folds.

Predicting Probabilities in R

```
# Predicting Probabilities
data$predprob<-round(fitted(riskmodel),2)
head(data,n=10)
```

- **fitted** function generates the predicted probabilities based on the final riskmodel.
- **round** function helps rounding the probabilities to 2 decimal
- **data$predprob:** Predicted probabilities are saved in the same dataset ‘data’ in new variable ‘predprob’.

Predicting Probabilities in R

```
# Output:
```

	SN	AGE	EMPLOY	ADDRESS	DEBTINC	CREDDEBT	OTHDEBT	DEFAULTER	predprob
1	1	3	17	12	9.3	11.36	5.01	1	0.81
2	2	1	10	6	17.3	1.36	4.00	0	0.20
3	3	2	15	14	5.5	0.86	2.17	0	0.01
4	4	3	15	14	2.9	2.66	0.82	0	0.02
5	5	1	2	0	17.3	1.79	3.06	1	0.78
6	6	3	5	5	10.2	0.39	2.16	0	0.22
7	7	2	20	9	30.6	3.83	16.67	0	0.19
8	8	3	12	11	3.6	0.13	1.24	0	0.01
9	9	1	3	4	24.4	1.36	3.28	1	0.75
10	10	2	0	13	19.7	2.78	2.15	0	0.82

Interpretation :

- Last column in the data ‘predprob;’ is the probabilities generated using final model.

Classification Table

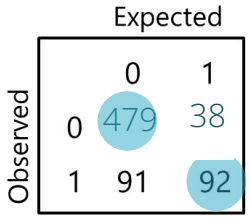
- Based on **cut-off value** of p, Y is estimated to be either 1 or 0

  Ex. p>0.5 ; Y=1

  p≤0.5 ; Y=0

- **Cross tabulation** of observed values of Y and predicted values of Y is called as **Classification Table.**
- The predictive success of the logistic regression can be assessed by looking at the classification table, but classification table is not a good measure of goodness fit since it **varies with the cut off value set.**
- Accuracy Rate measures **how accurate a model is in predicting outcomes.**
- In the adjoining table, 479 times Y=0 was observed as well as predicted. Similarly, Y=1 was observed and predicted 92 times.

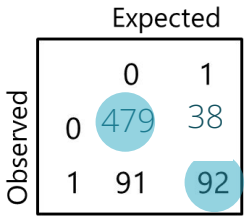
  Accuracy Rate = (479+92)/700 = 81.571



Misclassification

- Misclassification Rate □ Percentage of wrongly predicted observations
- Note that misclassification rate depends on cut off used for predictions

Suppose our classification table looks as follows:



- Here misclassification rate is : (38 +91) / 700=18.43%

Classification Table Terminology

Sensitivity	% of occurrences correctly predicted P(Ypred=1/Y=1)
Specificity	% of non occurrences correctly predicted P(Ypred=0/Y=0)
False Positive Rate (1 – Specificity)	% of non occurrences which are incorrectly predicted. P(Ypred=1/Y=0)
False Negative Rate (1- Sensitivity)	% of occurrences which are incorrectly predicted. P(Ypred=0/Y=1)

		Predicted	
		0	1
Observed	0	**Specificity**	**False Positive (1-Specificity)**
	1	**False Negative (1-Sensitivity)**	**Sensitivity**

Sensitivity and Specificity calculations

Cut-off Value		Accuracy	Sensitivity	Specificity
0.1	FALSE TRUE 0 252 265 1 12 171	(245+171)/700 = 60.4%	171/183=93.4%	245/517=48.7%
0.2	FALSE TRUE 0 352 165 1 28 155	(352+155)/700 = 72.4%	155/183=84.7%	352/517=68.1%
0.3	FALSE TRUE 0 415 102 1 46 137	(415+137)/700 = 78.9%	137/183=74.9%	415/517=80.3%
0.4	FALSE TRUE 0 449 68 1 70 113	(449+113)/700 = 80.14%	113/183=61.7%	449/517=86.8%
0.5	FALSE TRUE 0 479 38 1 91 92	(479+92)/700 =81. 57%	92/183=50.3%	479/517=92.6%

* Note : Here we are trying to find out the best cut-off value based on accuracy, sensitivity & specificity.

Classification and Sensitivity and Specificity table in R

```
# Predicting Probabilities
classificationtable<-table(data$DEFAULTER,data$predprob > 0.5)
classificationtable
```

- **table** function will create a cross table of observed Y (defaulter) vs. predicted Y (predprob).

```
# Output:
```

	FALSE	TRUE
0	479	38
1	91	92

Interpretation :

- True indicates predicted defaulters and False indicates predicted non-defaulters.
- There are 479 correctly predicted non-defaulters and 92 correctly predicted defaulters.
- There are 38 wrongly predicted as defaulters and 91 wrongly predicted as non-defaulters.

Sensitivity and Specificity in R

```
# Sensitivity and Specificity
sensitivity<-(classificationtable[2,2]/(classificationtable[2,2]+class
ificationtable[2,1]))*100
sensitivity

specificity<-(classificationtable[1,1]/(classificationtable[1,1]+class
ificationtable[1,2]))*100
specificity
```

```
# Output:
sensitivity
[1] 50.27322

specificity
[1] 92.6499
```

Interpretation :
The Sensitivity is at 50.3% and the Specificity is at 92.7% . This is when the cutoff was set at 0.5

Quick Recap

In this session, we learned how to execute **Binary Logistic Regression in R** :

Binary logistic regression	• Dependent variable is binary and independent variables are categorical or continuous or mix of both. • Regression line is sigmoid curve. • Parameters are estimated using MLE.
Classification table	• percentage of correctly predicted observations =accuracy. • Percentage of wrongly predicted observations =misclassification rate
Sensitivity/True Positive rate	• % of occurrences correctly predicted
Specificity/True Negative rate	• % of non occurrences correctly predicted
False Positive Rate	• % of non occurrences which are incorrectly predicted
False Negative Rate	• % of occurrences which are incorrectly predicted

Binary Logistic Regression

Checking Model Performance

Contents

Receiver Operating Characteristic Curve

- The **R**eceiver **O**perating **C**haracteristic (ROC) curve is

> A graphical representation of the trade off between the false positive and true positive rates for various cut off values

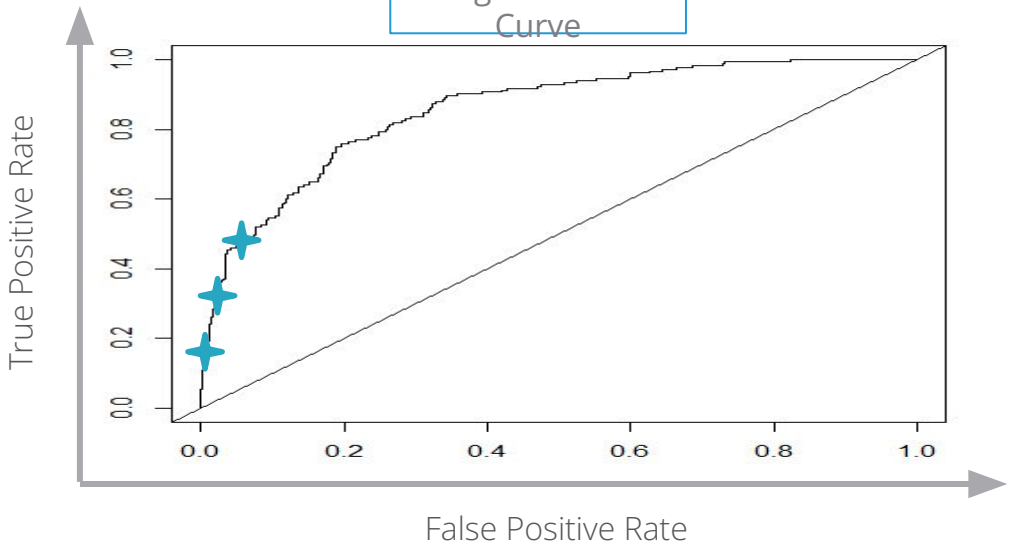
Y- axis: Sensitivity (true positive rate)

X-axis: 1-Specificity (false positive rate)

The performance of the classification model can be assessed by area under the ROC curve (C).

ROC Curve and Area Under ROC Curve

Fig. 01: ROC Curve



High TPR with low FPR is indicative of a good model. This will result in a curve that is closer to the Y-axis and top left corner of the plot. It implies a higher Area Under the ROC Curve.

ROC Curve and Area Under ROC Curve

Interpreting different versions of an ROC curve

Critical Points	Interpretations
TPR = 0 and FPR = 0	Model predicts every instance to be Non-event
TPR = 1 and FPR = 1	Model predicts every instance to be Event
TPR = 1 and FPR = 0	The Perfect Model

- If the model is perfect, AUC = 1
- If the model is guessing randomly, AUC = 0.5
- Thumb rule: Area Under ROC Curve > 0.65 is considered acceptable

ROC in R

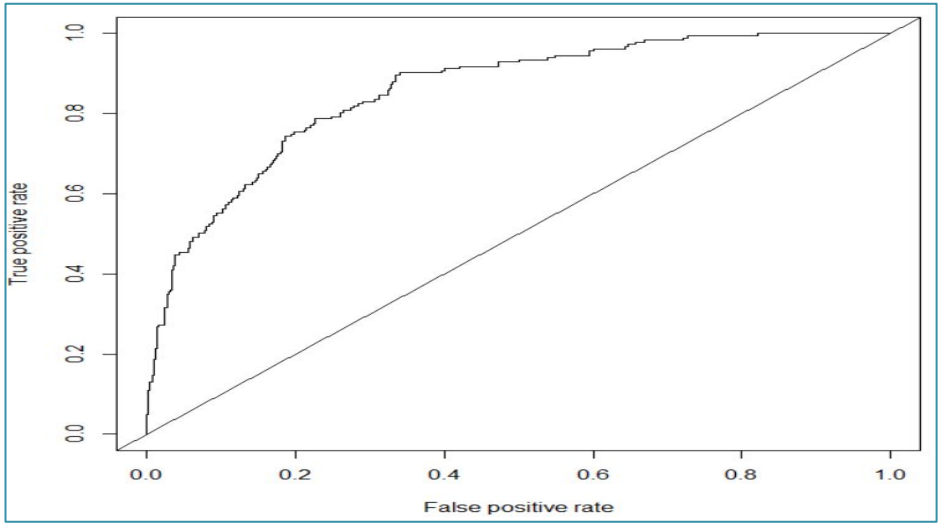
```
# Importing bank loan data & Fitting final Binary logistic model as
obtained in BLR02
data<-read.csv("BANK LOAN.csv",header=TRUE)
data$AGE<-factor(data$AGE)

riskmodel<-glm(DEFAULTER~EMPLOY+ADDRESS+DEBTINC+CREDDEBT,
family=binomial, data=data)
```

```
# Install and Load "ROCR" package.
install.packages("ROCR")
library(ROCR)

data$predprob<-fitted(riskmodel)
pred<-prediction(data$predprob,data$DEFAULTER)

perf<-performance(pred,"tpr","fpr")
plot(perf)
abline(0,1)
```

- **prediction()** function prepares data required for ROC curve.
- **performance()** function creates performance objects, "tpr" (True positive rate), "fpr" (False positive rate).
- **plot()** function plots the objects created using performance
- **abline()** adds a straight line to the plot.

ROC in R

```
# Output:
```




```
auc<-performance(pred,"auc")
auc@y.values
[1] 0.8556193
```

- Gives area under curve (AUC)

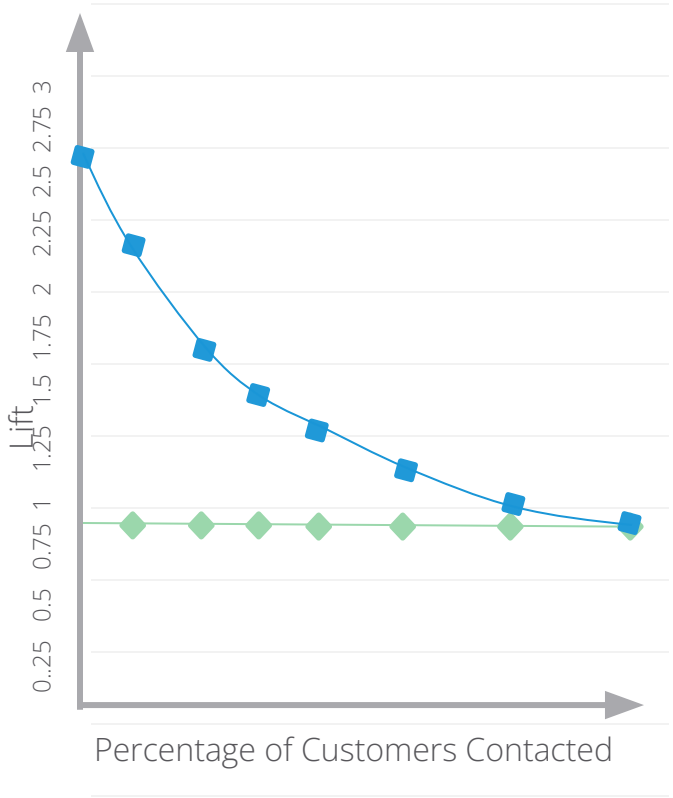
Interpretation :

- Area under the curve is 0.8556 which means model is performing well.

Lift Curve

- The idea is to quantify and compare two scenarios- one uses the model to identify certain cases and second using random selection of cases for a specific purpose such as a marketing campaign.
- Lift is the ratio of results obtained **with and without a model**.
- Although primarily used in marketing analytics, the concept finds applicability in other domains as well, such as risk modeling, supply chain analytics, etc.

Lift Curve



Lift Curve: After contacting X% of customers, Y% of respondents will be identified if a statistical model is used.
Ratio Y/X is plotted

Baseline: After contacting X% of customers, X% of respondents will be identified if random method is used.
Ratio X/X is plotted

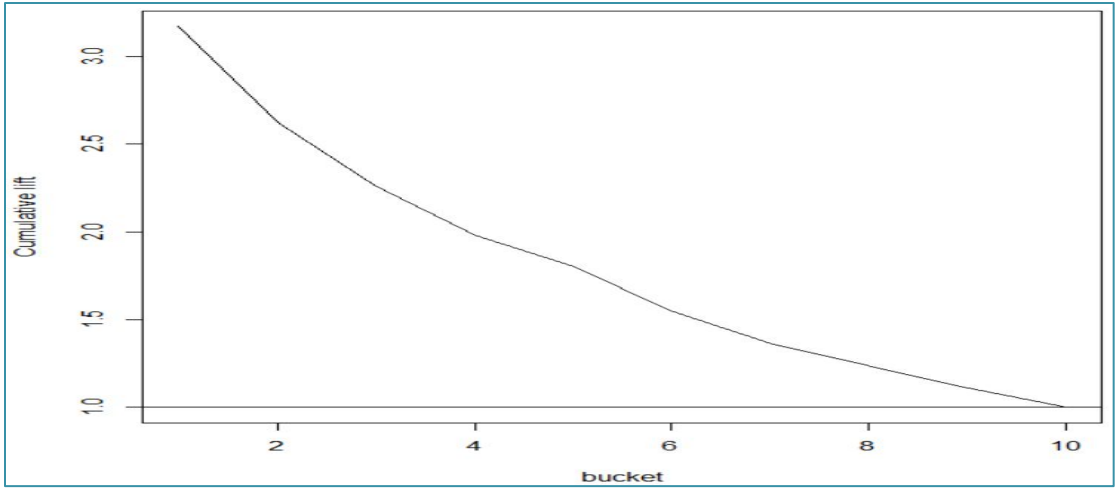
Lift Chart in R

```
# Install and load package “lift”
install.packages("lift")
library(lift)

data$predprob<-round(fitted(riskmodel),2)

plotLift(data$predprob,data$DEFAULTER, cumulative = TRUE,
n.buckets = 10)

abline(1,0)
```

- **fitted()** generates predicted probabilities.
- **plotLift()** plots a Lift curve by ordering the data by predicted probabilities and computing proportion of positives for each bucket.
- **cumulative=T** logical for specifying whether cumulative lift curve should be plotted
- **n.buckets=** how many buckets should be used
- **abline()** adds a straight line to the plot.

Lift Chart in R

```
# Output:
```



Interpretation :

- Model is performing better. As more defaulters identified in earlier buckets.

Kolmogorov-Smirnov Statistic

Kolmogorov-Smirnov (KS) Statistic is one of the most commonly used measures to assess predictive power for marketing or credit risk models.

> KS is the maximum difference between % cumulative Goods and Bads distribution across probability bands.
> The gains table typically has % cumulative Goods (or Non-Event) and % Cumulative Bads (Or Event) across 10 or 20 probability bands

- **KS is a point estimate**, which means it is only one value and indicates the probability band where separation between Goods (or Non-Event) and Bads (or Event) is maximum.
- Theoretically **K-S can range from 0-100**. **KS less than 25, may not indicate a good model**. Too high value should also be evaluated carefully.

Kolmogorov-Smirnov Statistic

BAND	Count	Percent	Count(bad)	%(bad)	Count(good)	%(good)	cum% bad	cum% good	KS
0.95-1	10	1.4%	9	4.9%	1	0.2%	4.9%	0.2%	4.7%
0.90-0.95	7	1.0%	7	3.8%	0	0.0%	8.7%	0.2%	8.5%
0.85-0.90	7	1.0%	6	3.3%	1	0.2%	12.0%	0.4%	11.6%
0.80-0.85	7	1.0%	5	2.7%	2	0.4%	14.8%	0.8%	14.0%
0.75-0.80	11	1.6%	9	4.9%	2	0.4%	19.7%	1.2%	18.5%
0.70-0.75	17	2.4%	14	7.7%	3	0.6%	27.3%	1.7%	25.6%
0.65-0.70	17	2.4%	12	6.6%	5	1.0%	33.9%	2.7%	31.2%
0.60-0.65	10	1.4%	7	3.8%	3	0.6%	37.7%	3.3%	34.4%
0.55-0.6	24	3.4%	14	7.7%	10	1.9%	45.4%	5.2%	40.1%
0.5-0.55	21	3.0%	9	4.9%	12	2.3%	50.3%	7.5%	42.7%
0.45-0.5	22	3.1%	9	4.9%	13	2.5%	55.2%	10.1%	45.1%
0.40-0.45	31	4.4%	13	7.1%	18	3.5%	62.3%	13.5%	48.8%
0.35-0.4	29	4.1%	11	6.0%	18	3.5%	68.3%	17.0%	51.3%
0.3-0.35	27	3.9%	13	7.1%	14	2.7%	75.4%	19.7%	55.7%
0.25-0.3	40	5.7%	7	3.8%	33	6.4%	79.2%	26.1%	53.1%
0.2-0.25	45	6.4%	12	6.6%	33	6.4%	85.8%	32.5%	53.3%
0.15-0.2	52	7.4%	10	5.5%	42	8.1%	91.3%	40.6%	50.6%
0.10-0.15	66	9.4%	4	2.2%	62	12.0%	93.4%	52.6%	40.8%
0.05-0.1	80	11.4%	8	4.4%	72	13.9%	97.8%	66.5%	31.3%
0-0.05	177	25.3%	4	2.2%	173	33.5%	100.0%	100.0%	0.0%
Total	**700**	**100%**	183	**100%**	517	**100%**			

Pearson Residuals

- The Pearson residual is defined as the standardized difference between observed and predicted frequency. It measures relative deviations between observed and fitted values. :

$$r_j = \frac{(Y_j - M_j p_j)}{\sqrt{M_j p_j (1 - p_j)}}$$

where
Mj : number of observations with jth covariate pattern
Y_j : Observed value (1 or 0) for jth covariate pattern
p_j : Predicted probability for j^{th} covariate pattern

- Binary Logistic Regression does not require 'Normality' of residuals

Pearson Residuals in R

```
# Getting Pearson Residuals:

data$resi<-residuals(riskmodel,"pearson")
head(data)
```

```
# Output:
```

	SN	AGE	EMPLOY	ADDRESS	DEBTINC	CREDDEBT	OTHDEBT	DEFAULTER	predprob	resi
1	1	3	17	12	9.3	11.36	5.01	1	0.80834673	0.4869219
2	2	1	10	6	17.3	1.36	4.00	0	0.19811470	-0.4970525
3	3	2	15	14	5.5	0.86	2.17	0	0.01006281	-0.1008221
4	4	3	15	14	2.9	2.66	0.82	0	0.02215972	-0.1505387
5	5	1	2	0	17.3	1.79	3.06	1	0.78180810	0.5282862
6	6	3	5	5	10.2	0.39	2.16	0	0.21646839	-0.5256165

- Residuals

- Pearson residuals are calculated by simply adding the argument **“pearson”** in the **residuals()** function.

Influence plot

- Influence plots are used to identify extreme values and their influence on a model.
- If removal of an observation causes substantial change in estimates of coefficients or predicted probabilities, then the observation is called an influential observation.
- Influential observations are analysed separately.

Influence plot in R

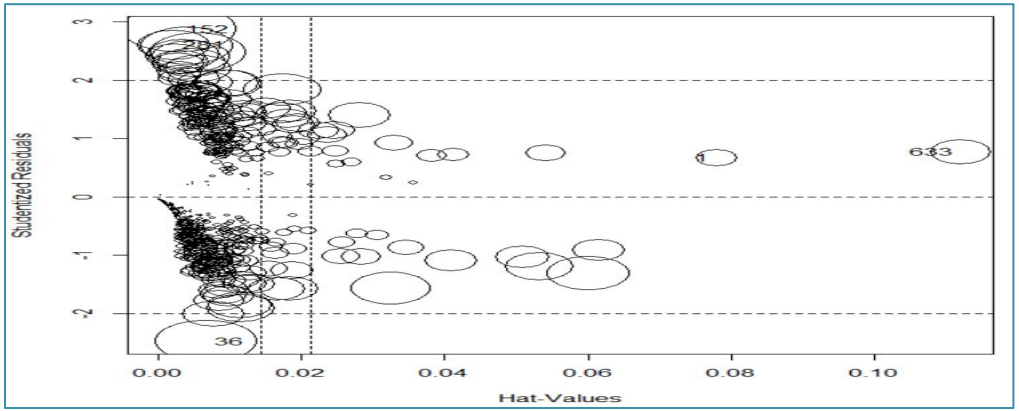
```
# Install and load “car” package.
install.packages("car")
library(car)

influencePlot(riskmodel)
```

- **influencePlot()** creates a bubble plot of Studentised residuals by hat values, with the areas of the circles representing the observations proportional to Cook’s distances.

Influence plot in R

```
# Output:
```

	StudRes	Hat	CookD
1	0.6675108	0.077944303	0.004347290
36	-2.4744534	0.006728529	0.025951104
152	2.8779760	0.002847547	0.032633681
281	2.6041504	0.002354813	0.013123240
633	0.7685420	0.112165052	0.009019769



- Large value of CookD indicates an influential observation
- Plot is for studentized residuals against hat-values, and the size of circle is proportional to Cook's distance

Multicollinearity

- Multicollinearity exists if there is a strong linear relationship among the continuous independent variables.
- Do not ignore multicollinearity in Binary Logistic Regression .
- Use variance inflation factors to detect multicollinearity.

Multicolinearity is explained in detail in MLR module.

Quick Recap

ROC Curve	• Graphical representation of the trade off between the false positive (FPR) and true positive (TPR) rates for various cut off values.
Lift Curve	• Compare model results with baseline without model
K-S statisitc	• KS is the maximum difference between % cumulative Goods (event/Y=1) and cumulative Bads (non events/Y=0) distribution across probability groups.
Residual	• Pearson's residual is used for binary logistic regression
Influence Plot	• Influence plots are used to identify the extreme values and their contribution to the model

Binary Logistic Regression

Model Validation

Contents

Cross Validation in Predictive Modeling

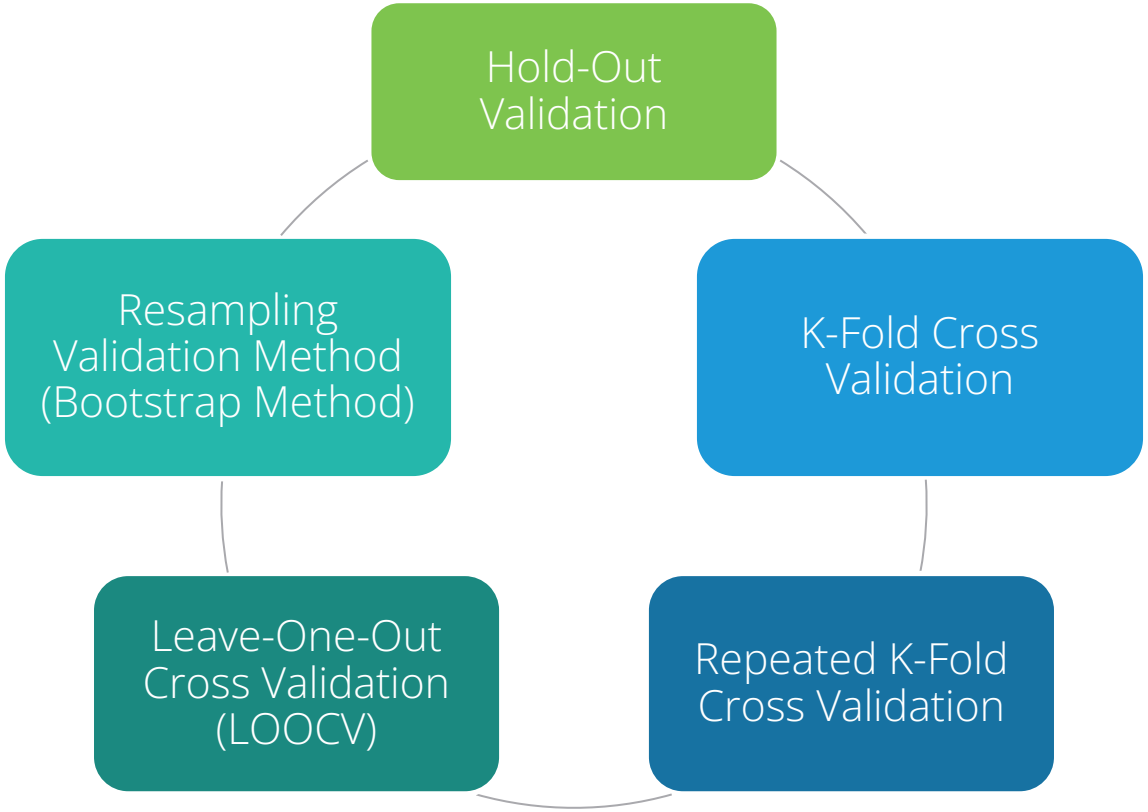
Cross Validation is a
process of evaluating the model on
'Out of Sample' data

- **Model performance measures** for binary logistic regression such as Accuracy rate, Sensitivity, Specificity **tend to be optimistic on 'In Sample Data'**
- More realistic measures of model performance are calculated using "Out of Sample' data
- Cross-validation is a procedure for estimating the generalization performance in this context

Cross validation is important because although a model is built on historical data, ultimately it is to be used on future data. However good the model, if it fails on out of sample data then it defeats the purpose of predictive modeling

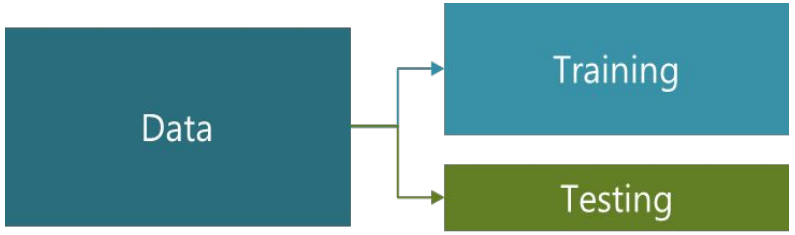
Cross Validation in Predictive Modeling

There are different approaches for cross validation. Five most significant of them are:



We will focus on **Hold Out** and **K-Fold** Cross validation methods.

Hold-Out Validation



In Hold-Out validation method, available data is split into two non-overlapped parts: 'Training Data' and 'Testing Data'

- The model is
  - Developed using training data
  - Evaluated using testing data
- Training data should have more sample size. Typically 70%-80% data is used for model development

* Here we continue to use previous data of bank loan for our further analysis.

Hold Out Validation in R

```
# Install and Load “caret” package
# Create 2 groups of the data: Training and Testing
```

```
install.packages("caret")
library(caret)

data <- read.csv("BANK LOAN.csv", header=TRUE)
index<-createDataPartition(data$DEFAULTER, p=0.7, list=FALSE)

dim(index)

traindata<-data[index,]
testdata<-data[-index,]
```

- **createDataPartition()** generates list of observation numbers to be included in training data.
- **p=** is the percentage of data that goes into training.
- **list=** specifies if results should be in a list format or matrix.

Hold Out Validation in R

```
# Check the dimensions of
```

```
dim(traindata)
[1] 490   8

dim(testdata)
[1] 210   8
```

Interpretation :

- The data of 700 observations are partitioned into 2 parts:
  With 490 observations in training (model development) part and remaining 210 observations in testing data part (out of sample).

Hold Out Validation

- Model will be run on the training data and predicted probabilities will be generated.
- Same model will be applied to test data to get the predicted probabilities.
- Confusion matrix will be used to check the performance of the model in training and testing data.

Confusion Matrix

	Observed	
Predicted	Event	No Event
Event	A	B
No Event	C	D

- Sensitivity = A/(A + C)
- Specificity = D/(B + D)
- Prevalence = (A + C)/(A + B + C + D)
- Positive Predicted Value = A / (A + B)
- Negative Predicted Value = D / (C + D)
- Detection Rate = A/(A + B + C + D)
- Detection Prevalence = (A + B)/(A + B + C + D)
- Balanced Accuracy = (Sensitivity + Specificity)/2
- Precision = A/(A + B)
- Recall = A/(A + C)

Confusion Matrix in R

```
# Generate confusion matrix for training data
riskmodel<-glm(DEFAULTER~EMPLOY+ADDRESS+DEBTINC+CREDDEBT,
               family=binomial,data=traindata)

traindata$predprob<-predict(riskmodel,traindata,type='response')
traindata$predY<-ifelse(traindata$predprob>0.30,1,0)
traindata$predY<-factor(traindata$predY)
traindata$DEFAULTER<-factor(traindata$DEFAULTER)
confusionMatrix(traindata$predY,traindata$DEFAULTER,positive="1")
```

```
# Generate confusion matrix for test data
testdata$predprob<-predict(riskmodel,testdata,type='response')
testdata$predY<-ifelse(testdata$predprob>0.3,1,0)
testdata$predY<-factor(testdata$predY)
testdata$DEFAULTER<-factor(testdata$DEFAULTER)
confusionMatrix(testdata$predY,testdata$DEFAULTER,positive="1")
```

- **confusionMatrix()** creates cross-tabulation of observed and predicted classes with associated statistics. The function contains data, reference.
- **positive=** factor level that corresponds to a “positive” result (Y=1).

Confusion Matrix in R

```
# Output:
```

For Training data

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 286  31
         1  73 100

               Accuracy : 0.7878
                 95% CI : (0.7488, 0.8232)
    No Information Rate : 0.7327
    P-Value [Acc > NIR] : 0.002877

                  Kappa : 0.5083
 Mcnemar's Test P-Value : 5.81e-05

            Sensitivity : 0.7634
            Specificity : 0.7967
         Pos Pred Value : 0.5780
         Neg Pred Value : 0.9022
             Prevalence : 0.2673
         Detection Rate : 0.2041
   Detection Prevalence : 0.3531
      Balanced Accuracy : 0.7800

       'Positive' Class : 1
```

For Testing data

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 127  14
         1  31  38

               Accuracy : 0.7857
                 95% CI : (0.724, 0.8392)
    No Information Rate : 0.7524
    P-Value [Acc > NIR] : 0.14903

                  Kappa : 0.4817
 Mcnemar's Test P-Value : 0.01707

            Sensitivity : 0.7308
            Specificity : 0.8038
         Pos Pred Value : 0.5507
         Neg Pred Value : 0.9007
             Prevalence : 0.2476
         Detection Rate : 0.1810
   Detection Prevalence : 0.3286
      Balanced Accuracy : 0.7673

       'Positive' Class : 1
```

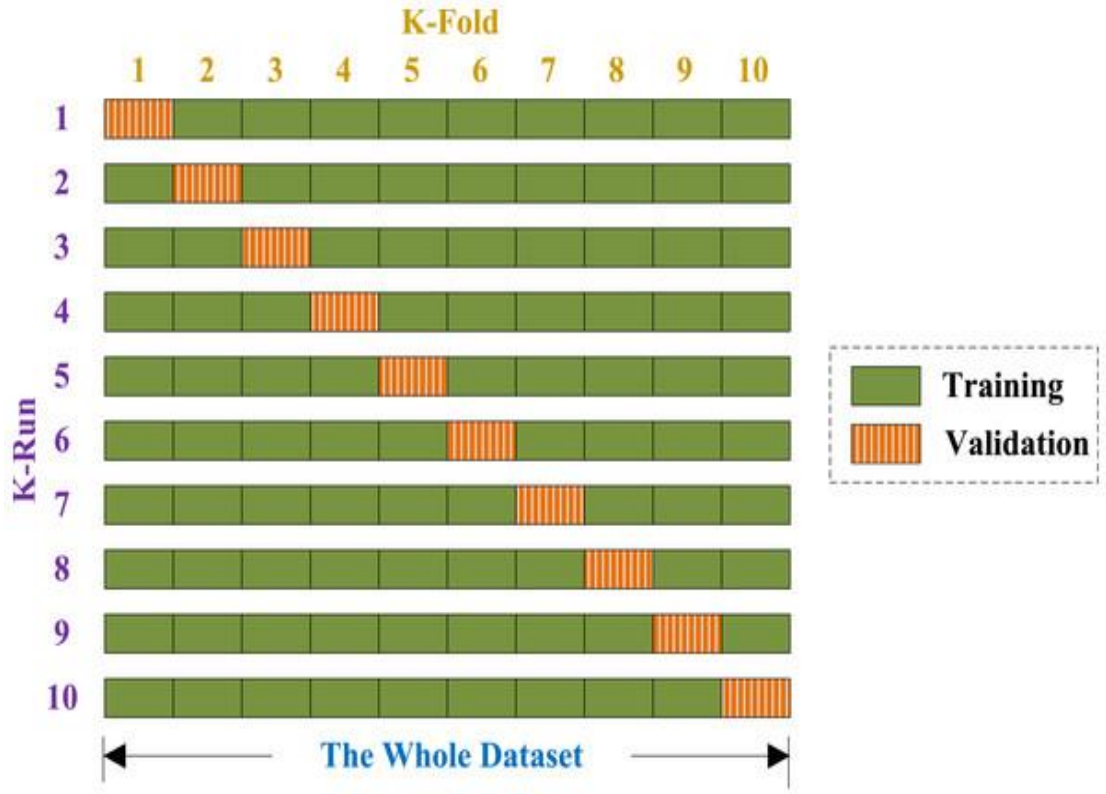
Interpretation :

- Accuracy of both the data is almost same. Sensitivity is also similar of both the datasets. Model is performing well for test data.

* Note : Since the Train –test data is chosen randomly, output may vary slightly on different devices.

K fold Cross Validation

- In k-fold cross-validation the data is first partitioned into k equally (or nearly equally) sized segments or folds.
- Then k iterations of training and testing are performed such that each time one fold is kept aside for testing and model is developed using k-1 folds.



K-fold Validation in R

```
# Create k-folds
library(caret)
kfolds<-trainControl(method="cv",number=4)

riskmodel<-train(as.factor(DEFAULTER)~EMPLOY+ADDRESS+
                   DEBTINC+CREDDEBT,data=data,method="glm",
                 family=binomial,trControl=kfolds)
riskmodel
```

- **trainControl()** control the computational nuances of the train function.
- **method="cv"** tells R to use Cross Validation method.
- **number=** specifies the number of folds.
- **train ()** fits predictive models over different tuning parameters.
- It performs a number of classification and regression routines, fits each model and calculates a resampling based performance measure.
- **trControl=** specifies the train function.

K-fold Validation in R

```
# Output:
```

```
Generalized Linear Model

700 samples
  4 predictor
  2 classes: '0', '1'

No pre-processing
Resampling: Cross-Validated (4 fold)
Summary of sample sizes: 525, 524, 525, 526
Resampling results:

  Accuracy  Kappa
  0.810054  0.4595083
```

Interpretation : accuracy of 0.81 indicates the good model.

Note : Since observations are assigned randomly in kfolds, output may vary slightly on different devices.

K-fold Validation in R

```
# Generate confusion matrix for k-fold validation
library(caret)
data$pred<-riskmodel$finalModel$fitted.values
data$predY<-ifelse(data$pred>0.3,1,0)

data$predY<-factor(data$predY)
data$DEFAULTER<-factor(data$DEFAULTER)

confusionMatrix(data$predY,data$DEFAULTER,positive="1")
```

- **riskmodel$finalModel$fitted.values:** Extract fitted model values from “riskmodel”.

K-fold Validation in R

```
# Output:
```

```
Confusion Matrix and Statistics

          Reference
Prediction   0   1
         0 415  45
         1 102 138

               Accuracy : 0.79
                 95% CI : (0.7579, 0.8196)
    No Information Rate : 0.7386
    P-Value [Acc > NIR] : 0.0009121

                  Kappa : 0.5059
 Mcnemar's Test P-Value : 3.86e-06

            Sensitivity : 0.7541
            Specificity : 0.8027
         Pos Pred Value : 0.5750
         Neg Pred Value : 0.9022
             Prevalence : 0.2614
         Detection Rate : 0.1971
   Detection Prevalence : 0.3429
      Balanced Accuracy : 0.7784

       'Positive' Class : 1
```

Interpretation : sensitivity and accuracy indicates stable model.

Quick Recap

In this session, we studied model validation for Binary Logistic :

Cross Validation	• Cross Validation is a process of evaluating the model on 'Out of Sample' data.
Hold out validation	• In Hold-Out validation method, available data is split into two non-overlapped parts: 'Training Data' and 'Testing Data'.
Confusion matrix	• It is used to check the performance of the model in training and testing data. • It has performance measures as Accuracy, sensitivity, specificity, etc..
K-fold validation	• In k-fold cross-validation the data is first partitioned into k equally (or nearly equally) sized segments or folds. • Then k iterations of training and testing are performed such that each time one fold is kept aside for testing and model is developed using k-1 folds.

Introduction to

Multinomial Logistic Regression

Contents

Multinomial Logistic Regression

DEPENDENT VARIABLE

Nominal

(With more than two mutually exclusive and exhaustive categories)

INDEPENDENT VARIABLE

Categorical or Continuous

- If there are k categories for the dependent variable, then (k-1) logit functions are defined with remaining 1 category as base level.

Application Areas

Industry / Function	Model	Based on Information such as:
Marketing	Buyers' Brand Preference	• Age, gender, payment mode, purchase frequency, historical purchase details, etc.
Education	Electives Chosen by Students	• Student gender, subjects chosen at the current Level, current academic score, etc.
Healthcare	Common painkiller used	• Demographics, type of ailment, socio-economic background, etc.

Statistical Model

- Let Y be the **dependent variable with 3 categories as A,B,C** and X1 ,X2,...Xk are k Independent variables.
- There will be 2 logit functions: one for **Y=B versus Y=A** and other **Y=C versus Y=A** Assuming A as the base category.

$$g_1(x) = \text{logit function for Y=B versus Y=A}$$
$$g_1(x) = \log\left(\frac{P_B}{P_A}\right)$$
$$= b_{01} + b_{11}x_1 + b_{21}x_2 + \ldots\ldots + b_{k1}x_k$$

where,
$P_B = P[Y = B \mid x]$
$P_A = P[Y = A \mid x]$

$$g_2(x) = \text{logit function for Y=C versus Y=A}$$
$$g_2(x) = \log\left(\frac{P_C}{P_A}\right)$$
$$= b_{02} + b_{12}x_1 + b_{22}x_2 + \ldots\ldots + b_{k2}x_k$$

where.
$P_C = P[Y = C \mid x]$

- Parameters of the model are estimated by the **Maximum Likelihood Estimation(MLE) Method**.

Case Study – High School Program Choice

Background

- At the time of entering high school, students make program choices among **general program**, **vocational program** and **academic program**. Their choice can be modeled using their writing score and their socio-economic status.

Objective

- To model student's choice of programs.

Available Information

- Data source: https://stats.idre.ucla.edu/
- Sample size is 200
- **Independent Variables**: Socio-Economic Status (SES) and Writing Score.
- **Dependent Variable**: Program Chosen (General, Vocational or Academic)

Data Snapshot

High School Data

Independent Variables

Dependent Variable

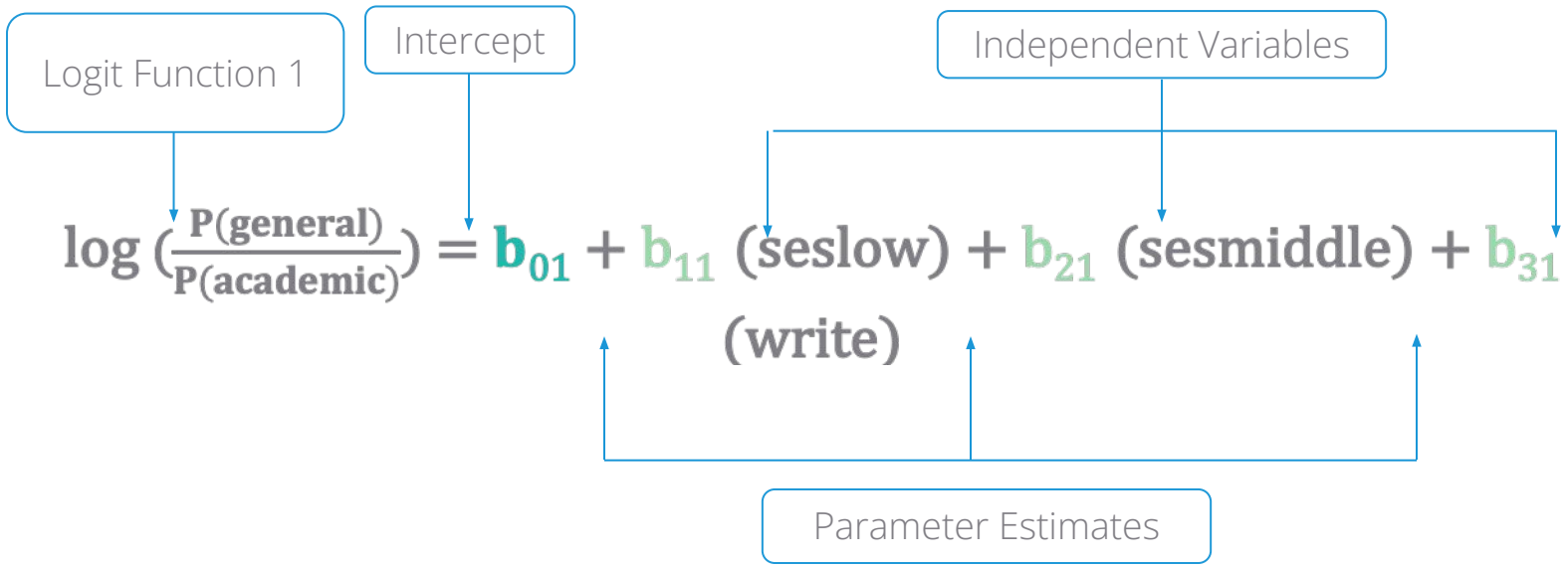
sn	id	ses	write	prog
1	45	low	35	vocation
2	108	middle	33	general

Column	Description	Type	Measurement	Possible Values
sn	serial number	numeric	-	-
id	student id	numeric	-	-
ses	scoio-economic status	Categorical	low, middle, high	3
write	writing score of the students	continuous	-	positive value
prog	program chosen by students	categorical	vocational, general, academic	3

Model for the case study

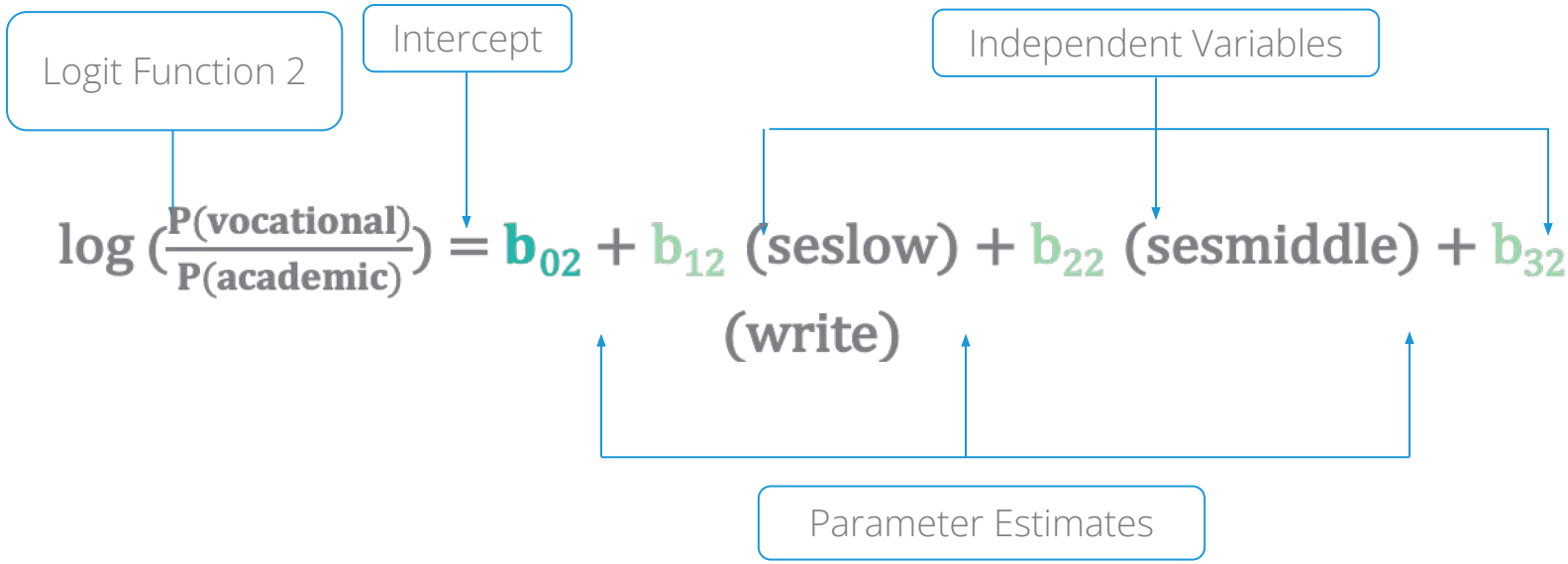
- There are two categorical variables in the data: 'prog' and 'ses'.
  - For the Dependent variable '**prog**', '**academic'** is taken as **base category**.
  - For the Independent variable '**ses**', **high**' is taken as **base category**.

- Model for the **general vs academic** is given as:



Model for the case study

- Model for the **vocational vs academic** is given as:



Maximum Likelihood Estimates of Parameters

Coefficients				
	Intercept	**seslow**	**sesmiddle**	**write**
general	1.689478	1.1628411	0.6295638	-0.05793086
vocation	4.235574	0.9827182	1.2740985	-0.11360389

Standard Errors				
	Intercept	**seslow**	**sesmiddle**	**write**
general	1.226939	0.5142211	0.4650289	0.02141101
vocation	1.204690	0.5955688	0.5111119	0.02222000

$$\log\left(\frac{P(general)}{P(academic)}\right) = 1.689478 + 1.1628411(seslow) + 0.629568$$
$$(sesmiddle) + (-0.05793086)(write)$$

- Similar to this, there will be another model equation for the category 'vocation' with 'academic' as base category.

Model Fitting in R

```
#Import the data
data<-read.csv("High School Data.csv", header=TRUE)
data$prog<-relevel(data$prog, ref="academic")

# Install and load package ‘nnet’.
install.packages("nnet")
library(nnet)
```

- **relevel()** tells R to re-order levels of a factor so that the level specified by ref is first and the others are moved down. First level is then taken as reference (base) category.

Model Fitting in R

```
#Run Multinomial Logistic Model

choicemodel<-multinom(prog~ses+write,data=data)

m<-summary(choicemodel)
m
```

- **mulinom()** fits a Multinomial Logistic Regression. Dependent variable is followed by '~' and independent variables are separated by plus signs.
- The output of **multinom()** function does not contain all the parameters required for further testing.
- In order to be able to extract specific components from the output and perform more actions on them, an object is created from **summary**().

Model Fitting in R

```
# Output
```

```
> choicemodel<-multinom(prog~ses+write,data=data)
# weights:  15 (8 variable)
initial  value 219.722458
iter  10 value 179.983731
final  value 179.981726
converged
> m<-summary(choicemodel)
> m
Call:
multinom(formula = prog ~ ses + write, data = data)

Coefficients:
          (Intercept)    seslow sesmiddle       write
general     1.689478 1.1628411 0.6295638 -0.05793086
vocation    4.235574 0.9827182 1.2740985 -0.11360389

Std. Errors:
          (Intercept)    seslow sesmiddle      write
general     1.226939 0.5142211 0.4650289 0.02141101
vocation    1.204690 0.5955688 0.5111119 0.02222000

Residual Deviance: 359.9635
AIC: 375.9635
```

- Output gives coefficients and standard errors of variables for each logit.

Individual Testing Using Wald's Test

- Individual testing is used for checking significance of each independent variable separately.

Objective	To test the **null hypothesis** that **each variable is insignificant**

Null Hypothesis (H_0): $b_{i1} = 0$ (for 1st logit)

Alternate Hypothesis (H_1): $b_{i1} \neq 0$ ((for 1st logit)

i=1,2...,k

Null Hypothesis (H_0): $b_{i2} = 0$ (for 2nd logit)

Alternate Hypothesis (H_1): $b_{i2} \neq 0$ (for 2nd logit)

i=1,2...,k

Test Statistic	$\mathbf{Z^2 = (b_{i1} / \text{Std. Error of } b_{i1})^2}$ Under H0, $Z^2 \sim \chi^2_{(1)}$
Decision Criteria	Reject the null hypothesis **if p-value < 0.05**

Individual Testing- Case study

Table of p-values				
	Intercept	seslow	sesmiddle	write
general	0.1685163893	0.02373673	0.1757949	6.816914e-03
vocational	0.0004382601	0.09893276	0.0126741	3.176088e-07

- p-value for seslow (general), sesmiddle (vocational) and write (general and vocational) < 0.05

Interpretation of Results

Coefficients				
	Intercept	**seslow**	**sesmiddle**	**write**
general	1.689478	**1.1628411**	0.6295638	**-0.05793086**
vocational	4.235574	0.9827182	**1.2740985**	**-0.11360389**
P-values				
general	0.1685163893	**0.02373673**	0.1757949	**6.816914e-03**
vocational	0.0004382601	0.09893276	**0.0126741**	**3.176088e-07**

- 'write' is a significant variable. Higher the writing score, less preference to 'general' or 'vocational'(as academic is base category and coefficient sign is negative).
- 'Low' SES category prefer 'general' over 'academic' more than 'high' SES category (as high SES is base category).
- 'middle' SES category prefer 'vocation' over 'academic' more than 'high' SES category.

Individual Testing in R

```
#Individual Testing

z<-m$coefficients/m$standard.errors

pvalue <-1-pchisq(z^2,df=1)

pvalue
```

- ‘z’ creates a dataframe of Z values as coefficients divided by standard errors
- **pchisq()** is used to calculate p-values using square of Z and degrees of freedom as arguments
- **pvalue** stores table of p-values.

Individual Testing in R

```
# Output:
```

	(Intercept)	seslow	sesmiddle	write
general	0.1685163893	0.02373673	0.1757949	6.816914e-03
vocation	0.0004382601	0.09893276	0.0126741	3.176088e-07

Interpretation :

- seslow(general), write(general), sesmiddle (vocation), write(vocation) are significant, as p-value <0.05.

Classification Table

- **Cross tabulation** of observed values of Y and estimated values of Y is called as Classification Table.
- The predictive success of the logistic regression can be assessed by looking at the classification table

Classification				
Observed	Predicted			
	academic	**general**	**vocation**	Percent Correct
academic	**92**	**4**	**9**	87.61%
general	**27**	**7**	**11**	15.56%
vocation	**23**	**4**	**23**	46.00%
Overall Percentage	71.0%	7.5%	21.5%	61.0%

- Table shows that, model is predicting 61%=(92+7+23)/ 200 correctly.

Predicted Probabilities and Classification Table in R

```
# Predicted Probabilities

data$predprob<-round(fitted(choicemodel),2)

head(data)
```

- **fitted**() generates predicted probabilities for program choice.

```
# Output:
```

	sn	id	ses	write	prog	predprob.academic	predprob.general	predprob.vocation
1	1	45	low	35	vocation	0.15	0.34	0.51
2	2	108	middle	33	general	0.12	0.18	0.70
3	3	15	high	39	vocation	0.42	0.24	0.34
4	4	67	low	37	vocation	0.17	0.35	0.48
5	5	153	middle	31	vocation	0.10	0.17	0.73
6	6	51	high	36	general	0.35	0.24	0.41

Predicted category is Vocation since it has highest probability 0.51

Interpretation :

- Predicted probabilities are given for each outcome (academic, general, vocation).
- Category of the maximum of these probabilities is taken as predicted category of that observation.

Predicted Probabilities and Classification Table in R

```
# Classification Table

expected<-predict(choicemodel,data, type="class")

ctable<-table(data$prog,expected)

ctable
```

- **predict**() returns predicted values.
- **type="class"** returns a factor of classifications based on the responses (frequency).
  type="probs" returns matrix of probabilities.
- **table**() function simply gives the true positive and negative rates of the model (in the form of counts), which are key to deciding power of the model.

```
# Output:
```

	expected		
	academic	general	vocation
academic	92	4	9
general	27	7	11
vocation	23	4	23

Interpretation :

- Classification table of predicted and expected counts.

Quick Recap

In this session, we learned about **Multinomial Logistic Regression** :

Multinomial Logistic Regression	• Dependent variable is nominal with more than two categories and independent variables are categorical or continuous or mix of both. • Parameters are estimated using MLE. • If there are k categories for the dependent variable then (k-1) logit functions are defined with remaining 1 category as base level.
Multinomial Logistic regression in R	• **`relevel()`** used to define base category. • **`nnet()`** library required for multinomial regression • **`multinom()`** performs multinomial logistic regression • Use **`summary()`** function to extract more details from **`multinom()`** function.

Introduction to

Ordinal Logistic Regression

Contents

Ordinal Logistic Regression

DEPENDENT VARIABLE

↓

Ordinal

(With two or more mutually exclusive and exhaustive categories)

INDEPENDENT VARIABLE

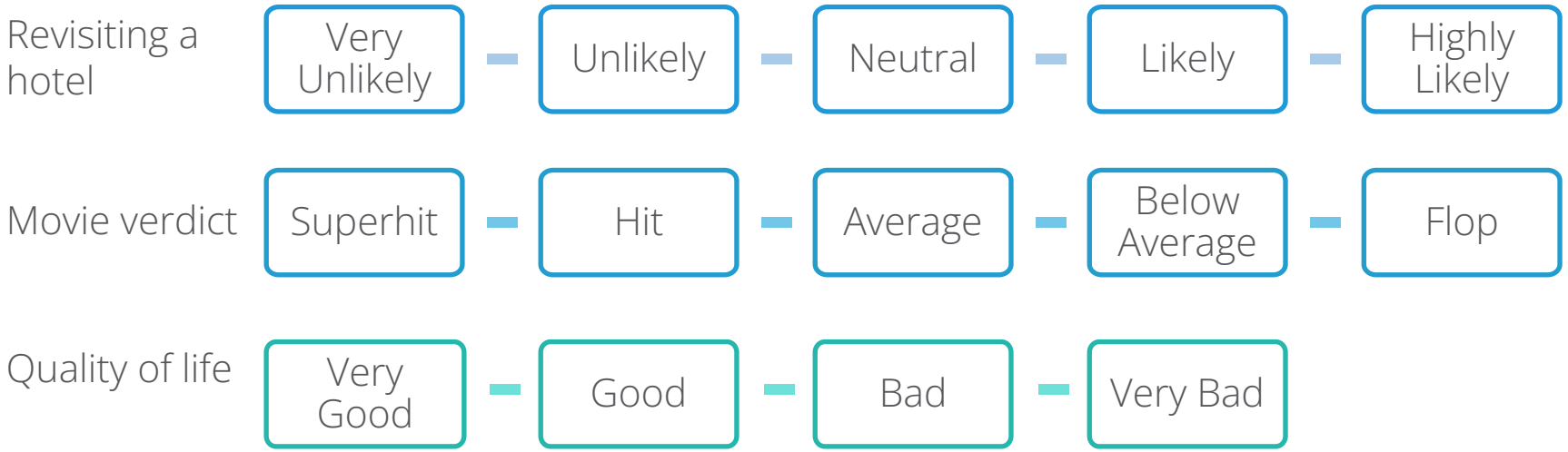
↓

Categorical or Continuous

- If there are **k categories** for the dependent variable then **(k-1) logit functions** are defined with remaining 1 category as base level.
- Here **coefficient of the variable is assumed to be same for each logit function** but **intercepts in logit functions differ**.

Ordinal Logistic Regression

Typical Examples of Ordinal and Scaled Variables

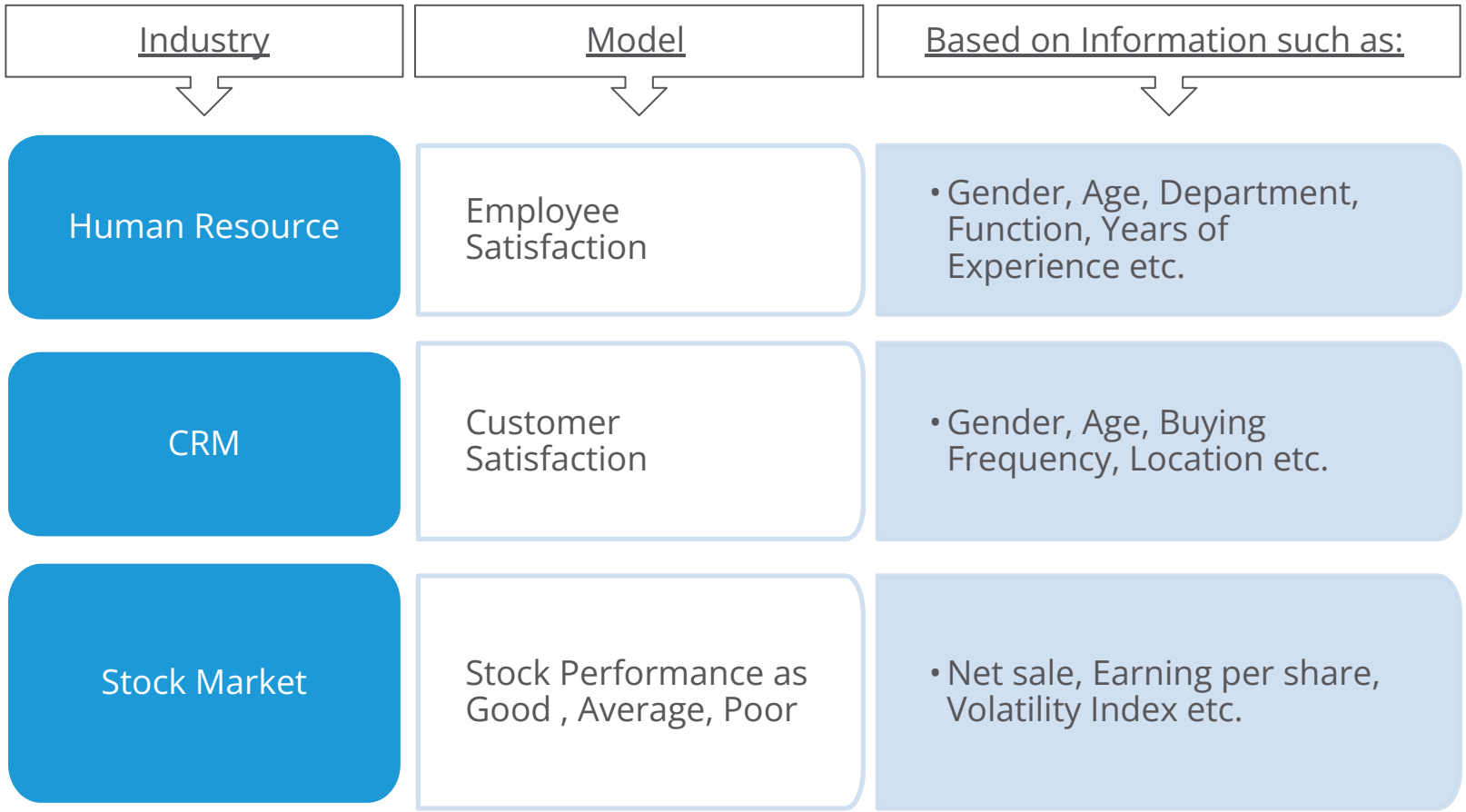


Generally ordinal dependent variable is represented numerically, i.e. it is coded.

Eg. Quality of life can be coded as Very Good = 4, Good = 3, Bad = 2 and Very Bad = 1.

* Note that though the difference in Y values, between Very Good & Good and Good & Bad is equal to 1,it does not mean the difference in level of satisfaction is equal.

Application Areas



Case Study – Brand Preference

Background

- A study was conducted to understand the customers' preference towards a brand. Data collected was customer demographics and their brand preference on a Likert scale.

Objective

- To study the factors that influence the brand preference.

Available Information

- Sample size is 259
- **Independent Variables**: Gender, Age, Location
- **Dependent Variable**: Brand Preference 1- Not Likely, 2-Likely, 3-Most Likely

Data Snapshot

Dependent Variables Independent Variables

Observations

id	Preference	Gender	Location	Age
1	3	MALE	CITY	<=25
2	2	MALE	CITY	25-40
3	1	MALE	CITY	40-55
4	2	FEMALE	CITY	25-40
5	2	FEMALE	CITY	40-55
6	1	FEMALE	CITY	25-40
7	2	FEMALE	CITY	<=25
8	2	FEMALE	SUBURBS	<=25

Column	Description	Type	Measurement	Possible Values
Id	Customer ID	numeric	-	-
Preference	Preference to the Brand	Categorical	1- Not Likely, 2-Likely, 3-Most Likely	3
Gender	Gender	Categorical	Male, Female	2
Location	Location	Categorical	City, Suburbs	2
Age	Age of the Customer	Categorical	<25, 25-40, 40-55	3

Model fitting in R

```
#Import the data
data<-read.csv("Brand Preference Study.csv", header=TRUE)

data$Preference<-as.ordered(data$Preference)

#Install and load package 'MASS'
install.packages("MASS")
library(MASS)

prefmodel <- polr(Preference~Gender+Location+Age,data=data,Hess=TRUE)
effect<-summary(prefmodel)
effect
```

- **as.ordered**() tells R to treat variable "Preference" as Ordinal variable.

- **polr**() fits a Proportional Odds Logistic Regression. Dependent variable is followed by a '~' and independent variables are separated by plus signs.
- **Hess=TRUE** ensures that the Hessian (the observed information matrix) is returned.

Model fitting in R

```
# Output:
Call:
polr(formula = Preference ~ Gender + Location + Age, data = data,
    Hess = TRUE)

Coefficients:
                  Value Std. Error t value
GenderMALE        1.1872     0.3420  3.4710
LocationSUBURBS -2.3863     0.2962 -8.0560
Age25-40        -0.2174     0.3141 -0.6923
Age40-55        -0.7511     0.3531 -2.1268

Intercepts:
    Value   Std. Error t value
1|2 -1.4568  0.3135    -4.6468
2|3  1.1904  0.3063     3.8859

Residual Deviance: 397.5779
AIC: 409.5779
```

- Output gives coefficient, standard error and t value for variables in each logit.

Individual Testing Using Wald's Test

- Individual testing is used for checking significance of each independent variable separately.

Objective	To test the **null hypothesis** that **each variable is insignificant**

Null Hypothesis (H_0): $b_i = 0$

Alternate Hypothesis (H_1): $b_i \neq 0$

i=1,2...,k

Test Statistic	$Z^2 = (b_i / \text{Std. Error of } b_i)^2$ Under H0, $Z^2 \sim \chi^2_{(1)}$
Decision Criteria	Reject the null hypothesis **if p-value < 0.05**

Individual Testing in R

```
#Individual Testing

ptable<-data.frame(effect$coefficients)

ptable$pvalue<- 1-pchisq(ptable$t.value^2,df=1)

ptable$pvalue<-round(ptable$pvalue,4)
ptable
```

- ptable stores coefficients along with t values
- **pchisq()** is used to calculate p-values.
- pvalue stores table of p-values.

Individual Testing in R

```
# Output:
```

	Value	Std..Error	t.value	pvalue
GenderMALE	1.1871541	0.3420191	3.4710171	0.0005
LocationSUBURBS	-2.3862520	0.2962095	-8.0559606	0.0000
Age25-40	-0.2174104	0.3140560	-0.6922664	0.4888
Age40-55	-0.7510563	0.3531452	-2.1267637	0.0334
1|2	-1.4567802	0.3135024	-4.6467910	0.0000
2|3	1.1903988	0.3063378	3.8859027	0.0001

Interpretation :

- Gender, Location and age40-55 are significant, as p-value <0.05.

Classification Table

- **Cross tabulation** of observed values of Y and estimated values of Y is called as Classification Table.
- The predictive success of the ordinal logistic regression can be assessed by looking at the classification table

Classification			
Observed	Predicted		
	Not Likely	**Likely**	**Most Likely**
Not Likely	**108**	**24**	**1**
Likely	**34**	**56**	**4**
Most Likely	**2**	**24**	**6**

- Table shows that, model is predicting 66%=(108+56+6)/ 259 correctly.

Predicted Probabilities and Classification Table in R

```
# Predicted Probabilities
data$predprob<-round(fitted(prefmodel),2)
head(data)
```

- **fitted**() generates predicted probabilities for brand preference.

```
# Output:
  id Preference Gender Location   Age predprob.1 predprob.2 predprob.3
1  1          3   MALE     CITY  <=25       0.07       0.43       0.50
2  2          2   MALE     CITY 25-40       0.08       0.47       0.45
3  3          1   MALE     CITY 40-55       0.13       0.55       0.32
4  4          2 FEMALE     CITY 25-40       0.22       0.58       0.20
5  5          2 FEMALE     CITY 40-55       0.33       0.54       0.13
6  6          1 FEMALE     CITY 25-40       0.22       0.58       0.20
```

Predicted category is 3(most likely) since it has highest probability 0.50

Interpretation :

- Predicted probabilities are given for each outcome (not likely, likely, most likely).
- Category with maximum of these probabilities is taken as predicted category of that observation.

Predicted Probabilities and Classification Table in R

```
# Classification Table
expected<-predict(prefmodel,data,type="class")

ctable<-table(data$Preference,expected)
ctable
```

- **predict**() returns predicted values.
- **type="class"** returns a factor of classifications based on the responses (frequency). **type="probs"** returns matrix of probabilities.
- **table**() function simply gives the true positive and negative rates of the model (in the form of counts), which are key for deciding power of the model.

```
# Output:
  expected
     1   2   3
1 108  24   1
2  34  56   4
3   2  24   6
```

Interpretation :

- Classification table of predicted and expected shows that, model is predicting 66%=(108+56+6)/ 259 correctly

Quick Recap

In this session, we learned about Ordinal Logistic Regression :

Ordinal Logistic Regression

- Generally ordinal dependent variable is represented numerically, i.e. it is coded.
- If there are k categories for the dependent variable then (k-1) logit functions are defined with remaining 1 category as base level.
- Coefficient of the variable is assumed to be same for each logit function but intercepts in logit function differ.

Ordinal Logistic regression in R

- **`MASS`** library required for ordinal regression
- **`polr()`** fits a Proportional Odds Logistic Regression.
- **`predict()`** function with **`type=class`** returns predicted category,

Poisson Regression

Contents

Understanding Poisson Distribution

- Suppose we wish to study the number of accidents taking place on a busy highway in a year. Number of accidents is a count variable and the event 'accident' is considered as rare event
- There are several phenomena where a variable is a 'count' and is observed in a specific time period; such as,
  - Number of deaths caused by lightning in six months
  - Number of visits to a dentist per year
- **Such random variables do not follow normal distribution** and hence cannot be modeled using multiple linear regression
- The probability distribution best suited for such data is **Poisson distribution** and the regression model is **Poisson regression**

The distribution is named after French mathematician Siméon Denis Poisson

Understanding Poisson Distribution

Poisson distribution is a limiting case of Binomial distribution where

- n (Number of trials) is very large ($n \to \infty$) and
- p (Probability of success) is very small ($p \to 0$) such that
- np is finite (say λ)

In other words, **chance of a success is very small and trial is repeated large number of times**

The Probability Mass Function is :

$$P(Y = y|\lambda) = \frac{e^{-\lambda}\lambda^{y}}{y!} \qquad y = 0,1,2,3 \ldots$$

Poisson distribution is specified with a single parameter λ .

For Poisson distribution Mean = Variance = λ

Poisson Regression

DEPENDENT VARIABLE

Count

Often it is the count of the rare event. Counts are all positive integers

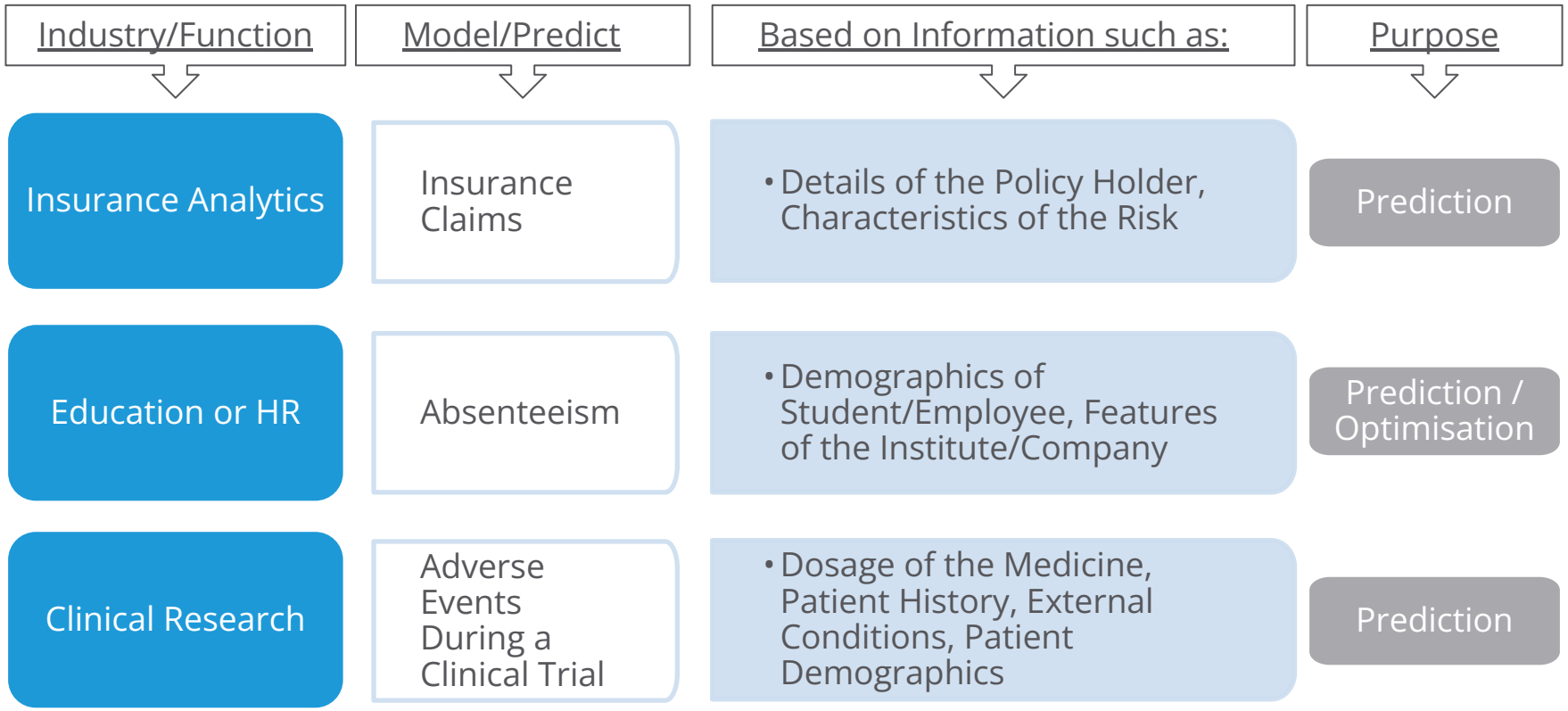
INDEPENDENT VARIABLE

Categorical or Continuous

Poisson regression is **most suitable in the case of rare events**

Poisson regression is a type of Generalised Linear Model, where the link function is a logarithm and the underlying distribution is Poisson

Application Areas



Statistical Model

$$\log(\lambda) = b_0 + b_1 x_1 + b_2 x_2 + \ldots\ldots + b_k x_k$$

Where, λ is the conditional mean of Y given X

- Here ordinary least square method used in multiple linear regression is not appropriate as Y is discrete and RHS is continuous
- log(λ) is the link function used in Poisson Regression which establishes link between 'Y' and linear combination of X's
- Note that λ is greater than zero and its log will be negative if it lies between 0 and 1
- The regression coefficients are estimated using the method of maximum likelihood

Case Study – Modeling Number of Complaints

Background

- A company has recently launched a loyalty program under which they collected information about their customers. The next leg of the program aims to add 20,000 customers to their loyalty pool. The company wants to understand if the number of complaints by a customer can be modeled in order to set up a call centre with optimum strength.

Objective

- To model number of complaints to prepare a road map for the call centre in the next leg of loyalty program

Available Information

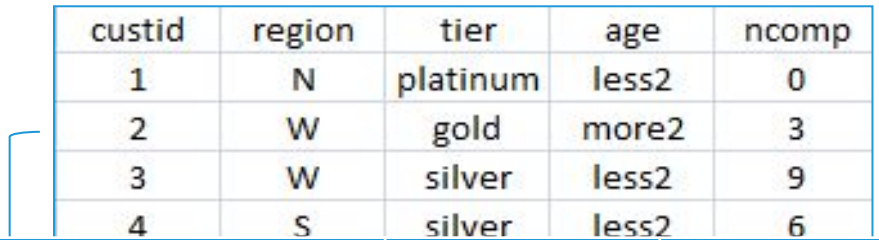
- Sample size is 113
- Information is available about **Region, Loyalty Tier, Complaints and Customer's Association with the Company**

Data Snapshot

Complaints

Independent variables

Dependent variable



custid	region	tier	age	ncomp
1	N	platinum	less2	0
2	W	gold	more2	3
3	W	silver	less2	9
4	S	silver	less2	6

Columns	Description	Type	Measurement	Possible values
custid	Customer ID	character	-	-
region	Region to which the customer belongs	categorical	E,W,N,S	4
tier	Loyalty program tier of the customer	categorical	platinum, gold, silver	3
age	Representing customer's association with the company	categorical	less2, more2	2
ncomp	Number of complaints	Integer(count)	-	positive values

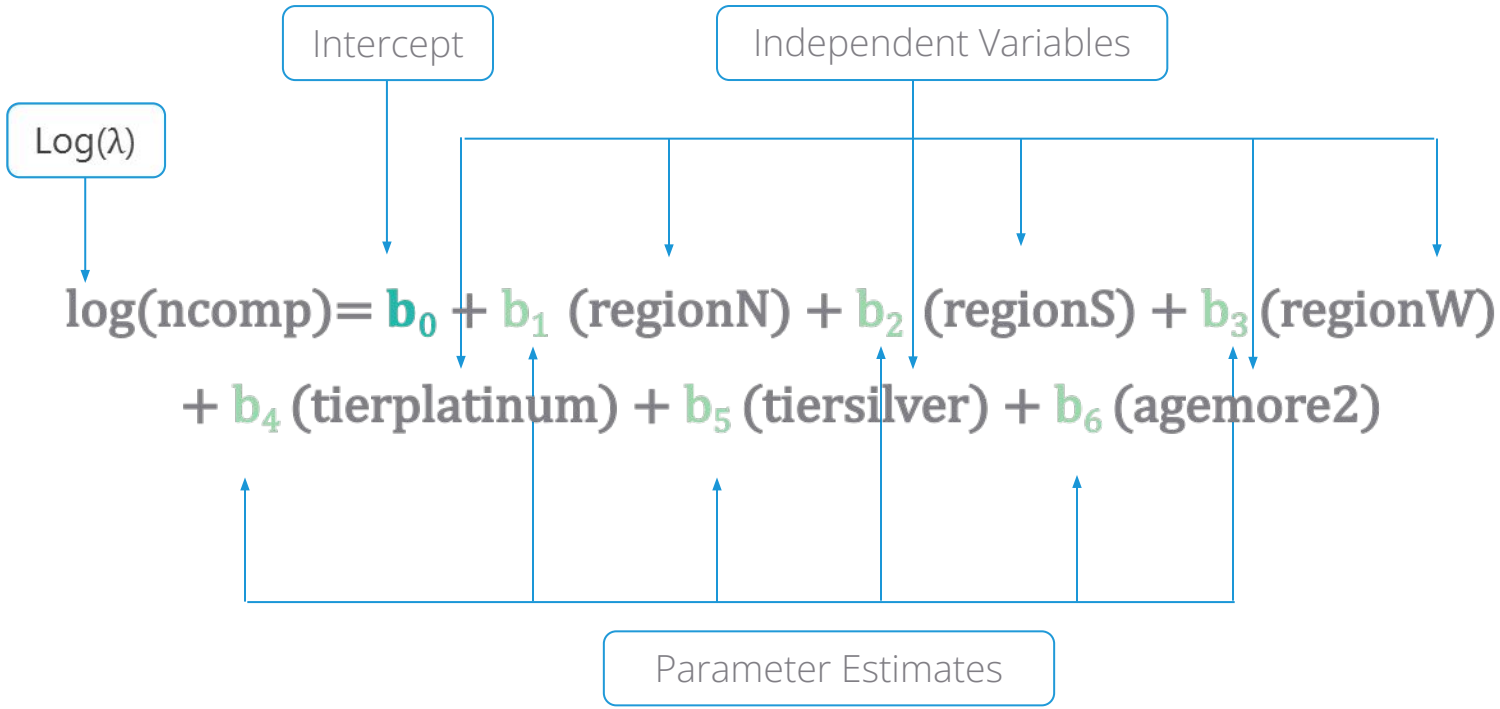
Model for the Case Study

All independent variables (region, tier and age) in the case study are categorical

Independent variables	Categories	Base category
region	East(E), West(W), North(N), South(S)	East(E)
tier	platinum, gold, silver	gold
age	less2, more2	less2

Model for the Case Study

The Poisson regression model is :



Model Fitting in R

```
#Importing the Data
calldata<-read.csv("Complaints.csv",header=TRUE)

#Model Fitting
compmodel<-glm(formula=ncomp~region+tier+age,data=calldata,
               family='poisson')

summary(compmodel)
```

- **glm()** fits a generalised linear model.
- **family=poisson** ensures that a Poisson regression is used.

summary() yields model summary.

Model Fitting in R

```
# Output
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.2919     0.1389   9.302  < 2e-16 ***
regionN       -0.1096     0.1420  -0.772 0.439968
regionS       -0.2286     0.1489  -1.535 0.124786
regionW       -0.4498     0.1613  -2.789 0.005290 **
tierplatinum  -0.6883     0.1754  -3.925 8.69e-05 ***
tiersilver     0.4410     0.1137   3.878 0.000105 ***
agemore2       0.1767     0.1077   1.641 0.100785
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

    Null deviance: 186.08  on 112  degrees of freedom
Residual deviance: 116.01  on 106  degrees of freedom
AIC: 456.86

Number of Fisher Scoring iterations: 5
```

Interpretation:

The **Estimate** column gives the estimates of coefficients of the independent variables in the model.

Individual Testing in R

```
# Identifying significant variables
summary(compmodel)
# Output
```

```
Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.2919     0.1389   9.302  < 2e-16 ***
regionN       -0.1096     0.1420  -0.772 0.439968
regionS       -0.2286     0.1489  -1.535 0.124786
regionW       -0.4498     0.1613  -2.789 0.005290 **
tierplatinum  -0.6883     0.1754  -3.925 8.69e-05 ***
tiersilver     0.4410     0.1137   3.878 0.000105 ***
agemore2       0.1767     0.1077   1.641 0.100785
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)
```

Interpretation:

- The p-values for **regionW, tierplatinum, tiersilver are <0.05**
- **regionW** is significant, with a negative coefficient : likelihood of complaints coming from West region is lower by -0.4498 compared to complaints from East region
- **tierplatinum** and **tiersilver** are significant, with a negative and positive coefficients respectively : compared to the base tier Gold, customers from Silver category tend to complain more whereas complaints from Platinum customers are the least

Goodness of Fit

Objective	To test the **null hypothesis** that **the model is a good fit**

Null Hypothesis (H_0): Model is a good fit

Alternate Hypothesis (H_1): There is significant lack of fit

Test Statistic	Pearson's chi − sq, $\chi^2 = \sum_{i=1}^{n} \frac{(y_i - \hat{\lambda}_i)^2}{\hat{\lambda}_i} \sim \chi^2_{(n-k)}$ or Deviance: $G^2 = 2\sum_{i=1}^{n} y_i \log\left(\frac{y_i}{\hat{\lambda}_i}\right) - (y_i - \hat{\lambda}_i) \sim \chi^2_{(n-k)}$
Decision Criteria	Reject the null hypothesis if p-value < 0.05

Goodness of fit in R

```
#Goodness of Fit
s1<-summary(compmodel)
res_deviance<-s1$deviance
df<-s1$df.residual
pvalue<-1-pchisq(res_deviance,df)
pvalue
```

- creating an object s1 to store summary of **compmodel**
- storing residual deviance in **res_deviance**
- storing the corresponding degrees of freedom in **df**
- **pchisq()** calculates Chi-square value by using (**Residual**, **degrees of freedom**) as the arguments

```
# Output
[1] 0.2380154
```

Interpretation:
p-value > 0.05, Do not reject H_0.
The model can be considered to be a good fit .

Predictions in R

```
#Predictions
calldata$ncomppred<-round(predict(compmodel,calldata,type='response'))
head(calldata)
```

- **predict()** requires model object, data and type.
- **type='response'** gives the predicted probabilities

```
# Output
```

	custid	region	tier	age	ncomp	ncomppred
1	1	N	platinum	less2	0	2
2	2	W	gold	more2	3	3
3	3	W	silver	less2	9	4
4	4	S	silver	less2	6	5
5	5	E	silver	less2	7	6
6	6	N	silver	less2	5	5

Interpretation:
The last two columns are observed and predicted vales of "ncomp"

Introduction to Zero-Inflated Poisson Regression

- One common cause of over-dispersion is excess zeros, which in turn are generated by an additional data generating process. In this situation, zero-inflated poisson regression should be considered
- Zero-inflated models attempt to account for excess zeros. In other words, two kinds of zeros are thought to exist in the data, "true zeros" and "excess zeros". Zero-inflated models estimate two equations simultaneously, one for the count model and one for the excess zeros.

```
# Zero-Inflated Poisson Regression
install.packages("pscl")
library(pscl)

zip_model<-zeroinfl(formula= dependent variable~ independent
variables| variable causing zero inflation, data= data)

summary(zip_model)
```

Offset Variable in Poisson Regression

- Poisson regression can be used to analyze **not only the count data but also the rate data**
- **Rates are simply counts divided by a measure like total count or time**. For ex. Insurance claim rate is measured as number of claims divided by the total number of the policy-holders (say N)
- The **log transformed regression variable** with the **constant coefficient of 1** for each observation is known as **'Offset'**
- The Poisson model is fit to the counts & **uses the log of the denominator** (usually N) **as an offset variable**
- Model in R using offset variable (C is number of claims, CAR is vehicle type ,AGE is vehicle age and N is number of policies)

- claimmodel<-glm(formula=C~offset(log(N))+CAR+AGE,data=claimdata, family=poisson)

Quick Recap

In this session, we learnt the basics of Poisson regression.

Poisson Regression	• Is used to model count of a rare event
Model Building	• glm function is used to perform Poisson Regression • Family="poisson" is used inside glm function
Parameter estimation	• Parameters are estimated by maximum likelihood estimation method
Check Variable Significance	• Hypothesis Testing

Quick Recap

Measure Goodness of Fit and Predictions	• Check Residual Deviance and Degrees of Freedom as a measure of goodness of fit • Deviance follows a chi-squared distribution, with degrees of freedom equal to the difference in the number of parameters • Predict the count using the estimated model parameters
Zero-Inflated Poisson Regression	• Is used when there are excess zeros in the count which are generated by an additional data generating process

Survival Analysis and Cox Regression

Contents

Introduction to Survival Analysis

- Survival analysis is the study of time taken for an event to occur.
- The **analysis variable** is the time between a **time origin and an end point**.
- The end point is either the occurrence of the event of interest, referred to as a death or failure, or the end of the subject's participation in the study.
- Two functions are of fundamental **interest—the survival function and the hazard function**
- One of the earliest applications of survival analysis was by Christiaan Huygens in 1669, showing how many out of 100 people survive until 86 years
- The name 'survival' analysis stems from the usage of this method for modeling 'time to death'; however the concept can be extended to several different areas and event can be defined as occurrence of a disease, lapse of a policy, etc.

Time to Event and Censoring

Consider that **time to occurrence of an event T is a random variable**. In order to define time-to-event following terms must be clearly defined:

Time Origin

The time origin must be specified such that individuals are as much as possible on an equal footing.
e.g. time point when treatment starts for a particular disease

Time Scale

Usually, observation time is used as the time scale for both clinical and observational studies
e.g. months, years, age

Definition of an Event

Based on the study objective, the event should be defined
e.g. death, disease occurrence etc.

If rate of occurrence of an event is λ then the expected time - to – event is $1/\lambda$

Subjects are said to be **censored** in case of either of the following outcomes:

- If they are lost to follow up
- If they drop out of the study
- If the study ends before they have an outcome of interest

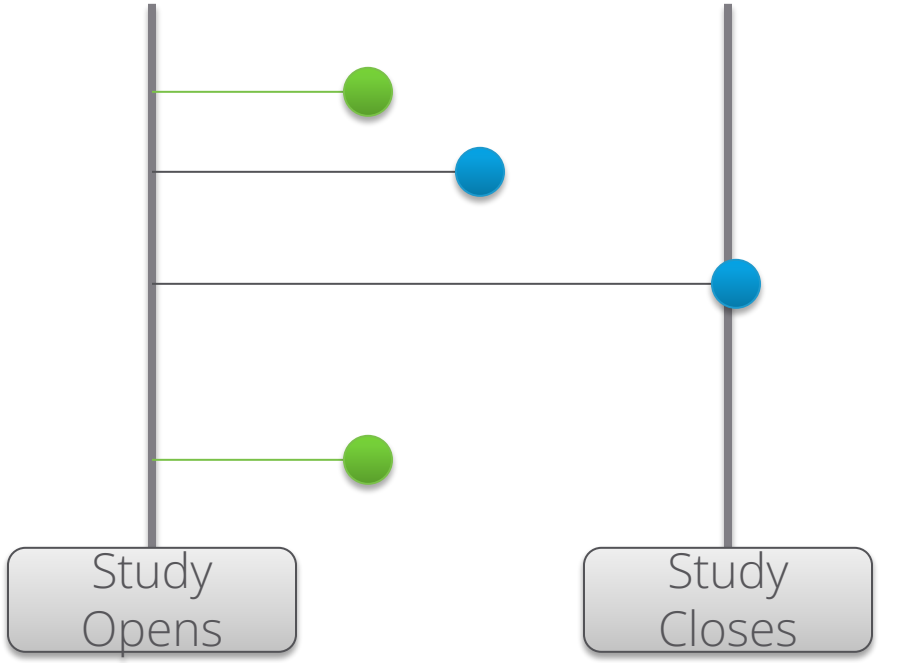
Concept of Censoring

Two-variable outcome :

Time variable: t_i = time at event, is a random variable having probability distribution

Status variable: c_i =1 if event occurred;

c_i =0 no event by time *t(Censored)*



Objectives of Survival Analysis

To estimate time-to-event for a group of individuals

- Eg. Time until recovery from low back pain

To assess the impact of factors/covariates on time-to-event

- Eg. Age, Gender, Occupation and treatment

Statistical Modeling

To compare time-to-event between two or more groups

- Eg. Time until recovery from low back pain in active vs. placebo groups

Statistical Inference

Concept of Survival Function

The goal of survival analysis is to estimate and compare **survival experiences of different groups.**

Survival experience is described by the cumulative survival function:

$$S(t) = P(T > t)$$

$$= 1 - P(T \leq t)$$

$$= 1 - F(t)$$

F(t) is the Cumulative Distribution Function (CDF)

Example: If t=40 years, S(t=40) = Probability of surviving beyond 40 years.

Hazard Function

Hazard Function (Instantaneous Failure Rate)

It is the ratio of conditional probability that the failure/death will occur in the interval t+Δt given that it has not occurred before time t and width of the interval (Δt).

Hazard function, $h(t) = \lim_{\Delta t \to 0} \frac{P(t \leq T < t + \Delta t | T > t)}{\Delta t}$, T is a random variable denoting **'Survival Time'**

$$= \lim_{\Delta t \to 0} \frac{F(t + \Delta t) - F(t)}{\Delta t}$$

$$= \frac{f(t)}{S(t)}$$

Hazard vs Density Function

- When one is born, there exists a certain probability of dying at any age; that is the probability density (Marginal probability)

  Example: A baby girl born today has, say, a 1% chance of dying at 80 years.

- However, as one survives for a while, the probabilities keep changing (Conditional probability)

  Example: A woman who is 79 today has, say, a 5% chance of dying at 80 years.

Cox Regression

DEPENDENT VARIABLE

Time to Event

Time variable must be quantitative

INDEPENDENT VARIABLES

Categorical or Continuous

Cox regression produces a survival function that predicts the probability of survival till time t for given values of the predictor variables

Statistical Model

$$h(t|x) = h_0(t)\exp(b_1x_1 + b_2x_2 + \cdots + b_kx_k)$$

$$\ln\left(\frac{h(t|x)}{h_{0(t)}}\right) = b_1x_1 + b_2x_2 + \cdots + b_kx_k$$

where

$h_{0(t)}$: Baseline hazard function (All x variables = 0)

$x_1, x_2, \ldots, x_k$: Independent variables

$b_1, b_2, \ldots, b_k$: Unknown parameters of the model

Cox Regression model is semi-parametric method

(No assumption about specific distribution but parametric form of the model)

Case Study – Predicting Time Taken to Default

Background

- The bank possesses demographic and transactional data of its loan customers. If the bank has a robust model to predict defaulters it can undertake better resource allocation.

Objective

- To predict whether the customer applying for the loan will be a defaulter and to identify early defaulters.

Available Information

- Sample size is 700
- **Age group, Years at current address, Years at current employer, Debt to Income Ratio, Credit Card Debts, Other Debts** are the independent variables
- **Status and Time** are used to create survival objects. Status =1 if customer defaulted before 36 months, and 0 if no default was observed in 36 months

Data Snapshot

BANK LOAN (COX)

Independent variables

Survival objects

Columns	Description	Type	Measurement	Possible values
AGE	Age Groups 1 (<28 years), 2(28-40 years), 3 (>40 years)	Factor	1,2,3	3
EMPLOY	No. of Years the Customer is Employed	Numerical	Years	positive value
ADDRESS	No. of Years the Customer is Staying at their Current Address	Numerical	Years	positive value
DEBTINC	Debt to Income Ratio	Numerical	-	positive value
CREDDEBT	Credit to Debt Ratio	Numerical	-	positive values
OTHERDEBT	Other Debt	Numerical	-	Positive value
STATUS	Whether the Customer Defaulted on the Loan (1) or 0 (Censored at 36 Months)	Binary	0,1	2
TIME	Indicates Time of 'Default'	Numerical	In months	positive value

Model Fitting in R

```
#Importing the Data
bankloan<-read.csv("BANK LOAN (COX).csv",header=TRUE)
bankloan$AGE<-as.factor(bankloan$AGE)

#Creating a Survival Object
library(survival)

surv.object<-Surv(bankloan$TIME,bankloan$STATUS)

#Model Fitting
timemodel<-coxph(surv.object~AGE+EMPLOY+ADDRESS+DEBTINC+CREDDEBT
                 +OTHDEBT, data=bankloan, x=TRUE)

summary(timemodel)
```

- **AGE** is converted to factor.

Surv() creates a survival object which will be used as the response variable in Cox regression. It requires time to event and event variable

- **coxph()** from package **survival** fits a Cox Regression.
- Dependent variable (Survival object) is followed by a tilde and independent variables are separated by plus signs.
- x logical value: if TRUE, the x matrix is returned in component x

Model Fitting in R

```
# Output
```

```
> summary(timemodel)
Call:
coxph(formula = surv.object ~ AGE + EMPLOY + ADDRESS + DEBTINC +
    CREDDEBT + OTHDEBT, data = bankloan)

  n= 700, number of events= 183

             coef exp(coef) se(coef)       z Pr(>|z|)
AGE2      0.30668   1.35891  0.18701   1.640   0.1010
AGE3      0.54006   1.71611  0.25293   2.135   0.0327 *
EMPLOY   -0.24177   0.78524  0.02238 -10.803  < 2e-16 ***
ADDRESS  -0.09825   0.90643  0.01634  -6.011 1.84e-09 ***
DEBTINC   0.05859   1.06034  0.01308   4.478 7.53e-06 ***
CREDDEBT  0.58482   1.79468  0.05020  11.649  < 2e-16 ***
OTHDEBT   0.06465   1.06679  0.03166   2.042   0.0411 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

         exp(coef) exp(-coef) lower .95 upper .95
AGE2        1.3589     0.7359    0.9419    1.9605
AGE3        1.7161     0.5827    1.0453    2.8173
EMPLOY      0.7852     1.2735    0.7515    0.8204
ADDRESS     0.9064     1.1032    0.8779    0.9359
DEBTINC     1.0603     0.9431    1.0335    1.0879
CREDDEBT    1.7947     0.5572    1.6265    1.9802
OTHDEBT     1.0668     0.9374    1.0026    1.1351

Concordance= 0.833  (se = 0.014 )
Likelihood ratio test= 336.9  on 7 df,   p=<2e-16
Wald test            = 282.4  on 7 df,   p=<2e-16
Score (logrank) test = 322  on 7 df,   p=<2e-16
```

Interpretation :
The **coef** column gives the estimates of the parameters in the model.

Individual Testing in R

```
# Output
```

	coef	exp(coef)	se(coef)	z	Pr(>|z|)	
AGE2	0.30668	1.35891	0.18701	1.640	0.1010	
AGE3	0.54006	1.71611	0.25293	2.135	0.0327	*
EMPLOY	-0.24177	0.78524	0.02238	-10.803	< 2e-16	***
ADDRESS	-0.09825	0.90643	0.01634	-6.011	1.84e-09	***
DEBTINC	0.05859	1.06034	0.01308	4.478	7.53e-06	***
CREDDEBT	0.58482	1.79468	0.05020	11.649	< 2e-16	***
OTHDEBT	0.06465	1.06679	0.03166	2.042	0.0411	*

Interpretation:

- Except AGE2, all variables are significant and have an impact on time taken by a customer to default (p-values <0.05)
- Higher the number of years spent at one address or employee, lesser is the probability to default (as the coefficients are negative)
- Higher the amount of liabilities, higher is the probability to default (as the coefficients are positive)

Predicted Probabilities in R

```
#Importing New Data for Predictions & check if the structure is same as
#the train data

bankloantest<-read.csv("BANK LOAN (COX) TEST.csv",header=TRUE)
bankloantest$AGE <- as.factor(bankloantest$AGE)

#Predicted Probabilities

install.packages("pec")
library(pec)
bankloantest$prob24<-predictSurvProb(timemodel,bankloantest,times=24)

head(bankloantest)
```

- **predictSurvProb()** extracts probability predictions from different modeling approaches, most commonly used for Cox regression.
- **times=** is a vector of times in the range of the response variable, e.g. times when the response is a survival object, at which to return the survival probabilities
- Here, it is used to give probability that the customer will survive (Remain non-defaulter) for at least 24 months.

Predicted Probabilities in R

```
#Output
```

	SN	AGE	EMPLOY	ADDRESS	DEBTINC	CREDDEBT	OTHDEBT	prob24
1	701	3	17	12	9.4	11.38	5.01	0.1262493
2	702	2	10	6	17.3	1.36	4.00	0.9341567
3	703	3	15	13	5.5	0.86	2.17	0.9957209
4	704	2	15	14	2.9	2.66	0.82	0.9930838
5	705	1	2	0	17.6	1.79	3.06	0.4630305
6	706	1	5	5	10.2	0.35	2.16	0.9416758

Interpretation :
Predicted probabilities that the customer will default before 24 months

Proportional Hazards Model

For any two cases, the ratio of hazard function at any time point is constant

Consider simple example in which only independent variable is X=1 for group 1 and X=0 for group 2

For X=1,hazrad function is

$$h(t|x)=h_0(t)\ exp^{(b_1)}$$

For X=0,hazrad function is

$$h(t|x)=h_0(t)$$

Therefore, hazard ratio is

$$exp^{(b_1)} = Constant$$

Quick Recap

Survival Analysis,	• Survival analysis is the study of time taken for an event to occur
Cox regression	• Cox regression is used to model "Time to Event" variable
Cox Regression Model Fitting,	• **`coxph()`** from package **survival** fits a Cox regression • Survival Object is the response variable in **`coxph()`** • **`summary()`** of **`coxph()`** object returns Likelihood Ratio test results and p-values for checking variable significance
Predictions on New Data	• **`predictSurvProb()`** from package **pec** generates predicted probabilities
Proportional Hazards Model	• For any two cases, the ratio of hazard function at any time point is constant