Nonlinear & Logistic Regression using RStudio

Course Taught at SUAD

Dr. Tanujit Chakraborty
Ms. Madhurima Panja
@ Sorbonne
tanujitisi@gmail.com

This presentation includes…

- Regression Analysis
  - Non-linear regression
  - Regression Splines
- Logistic Regression
  - Binomial logistic regression
  - Multinomial logistic regression

Non-Linear Regression Model

Non-linear Regression Model

Definition and Formulation:

- **Non-linear Regression Model:** When the regression equation is in terms of r −degree, $r > 1$, then it is called non-linear regression model
- **Multiple Non-linear Regression Model:** When more than one independent variables are there, then it is called multiple non-linear regression model
  - It is alternatively termed as polynomial regression model.
- In general, it takes the form

$$y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_r x^r + \epsilon$$

- The estimated response is obtained as

$$\hat{y} = b_0 + b_1 x + b_2 x^2 + \ldots + b_r x^r$$

Solving for Polynomial Regression Model

Model formulation:

Given that (x_i, y_i); $i = 1,2,\ldots,n$ are n pairs of observations.

Each observations would satisfy the equations:

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 {x_i}^2 + \ldots + \beta_r {x_i}^r + \epsilon_i$$

and $$\hat{y}_i = b_0 + b_1 x_i + b_2 {x_i}^2 + \ldots + b_r {x_i}^r + e_i$$

where, r is the degree of polynomial
ϵ_i is the i^{th} random error
e_i is the i^{th} residual error

Note: The number of observations, n, must be at least as large as $r + 1$, the number of parameters to be estimated.

Solving for Polynomial Regression Model

Transformation to Linear Regression:

The polynomial model can be transformed into a general linear regression model setting $x_1 = x, x_2 = x^2, \ldots, x_n = x^r$.

Thus, the equation assumes the form:

$$y_i = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_r x_r + \epsilon_i$$

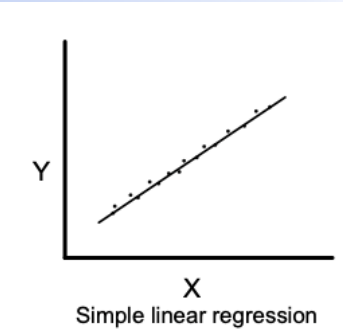
$$\hat{y}_i = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_r x_r + e_i$$

This model then can be solved using the procedure followed for multiple linear regression model.

Linear versus Non-Linear Regression

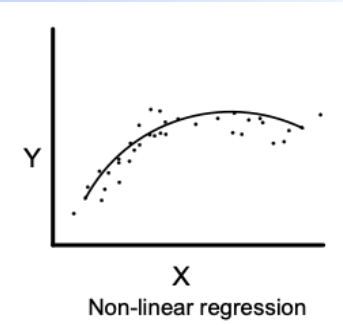
Simple linear regression model:

$$Y = \beta_0 + \beta_1 x$$



Simple non-linear regression model:

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_r x^r$$



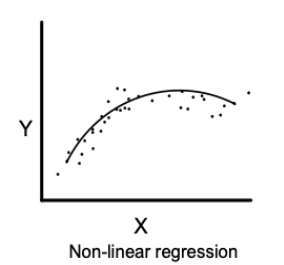
Linear versus Non-Linear Regression

Simple linear regression model:

$$Y = \beta_0 + \beta_1 x$$

Simple non-linear regression model:

$$Y = \beta_0 + \beta_1 x + \beta_2 x^2 + \ldots + \beta_r x^r$$



Issues:

a) Whether linear or non-linear model?
b) If non-linear, then what is its degree $r \geq 2$?

Solution:

Take the R^2 measures for all models (with r =1, 2, ...)
and then select that model with the higher value of R^2

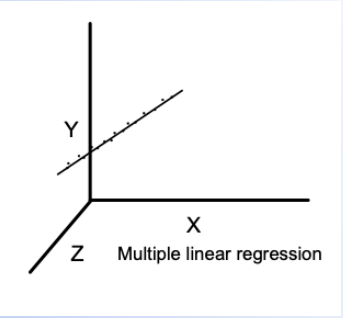
X	Y
x_i	y_i

Multiple Non-Linear Regression

Issues with Multiple Non-Linear Regression

Multiple non-linear regression model:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2^2 + \ldots + \beta_r x_k^r$$



Issues:

- Too complex to solve. Many parameters, many variations!
- Usually, used advanced machine learning models, such as SVM, kNN, ANN, etc.

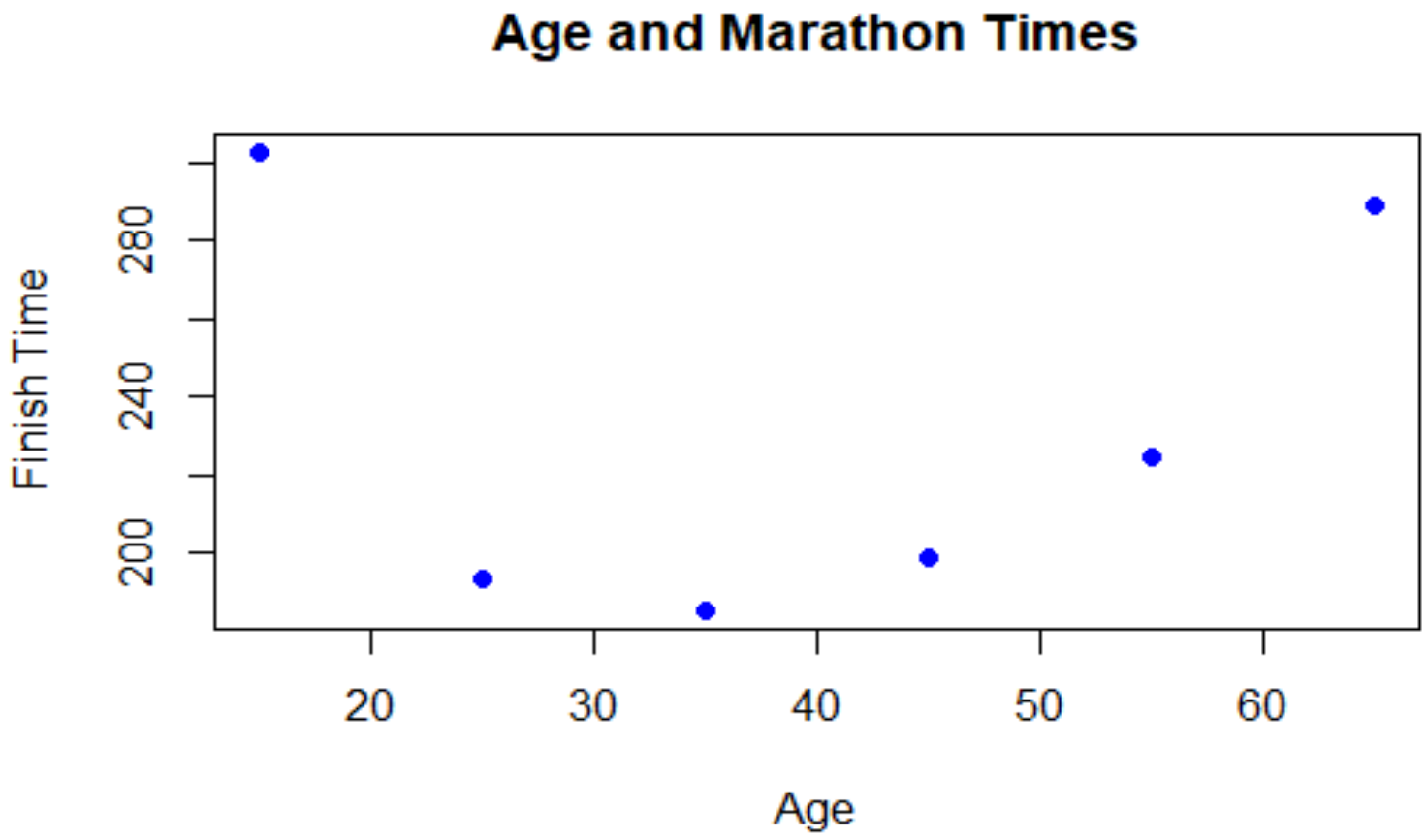
X	Y	Z

Nonlinear Regression: Marathon Data

- Example: The article **"Master's Performance in the New York City Marathon" (*British Journal of Sports Medicine* [2004]: 408–412)** gave the following data on the average finishing time by age group for female participants in the New York City marathon.

Age Group	Representative Age	Average Finish Time
10-19	15	302.38
20-29	25	193.63
30-39	35	185.46
40-49	45	198.49
50-59	55	224.30
60-69	65	288.71

Scatter Plot

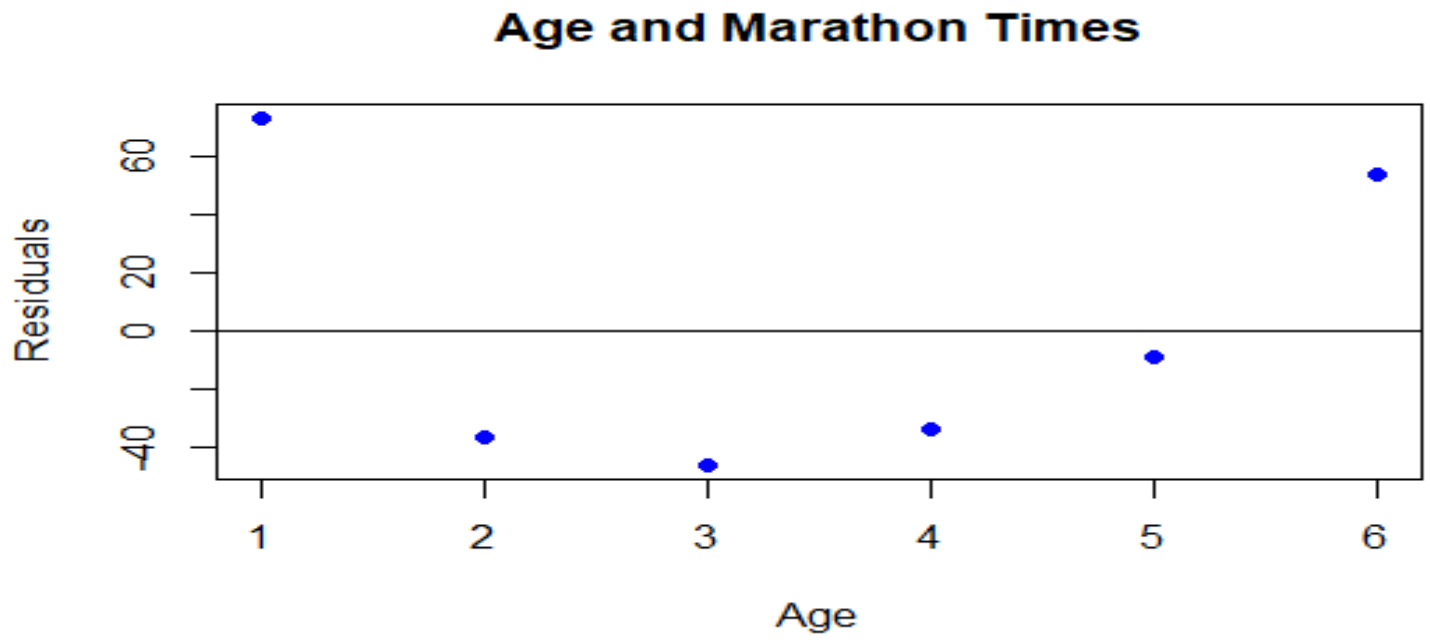


Correlation coefficient in Linear Case & Residual Plot

$r = 0.03847689$,

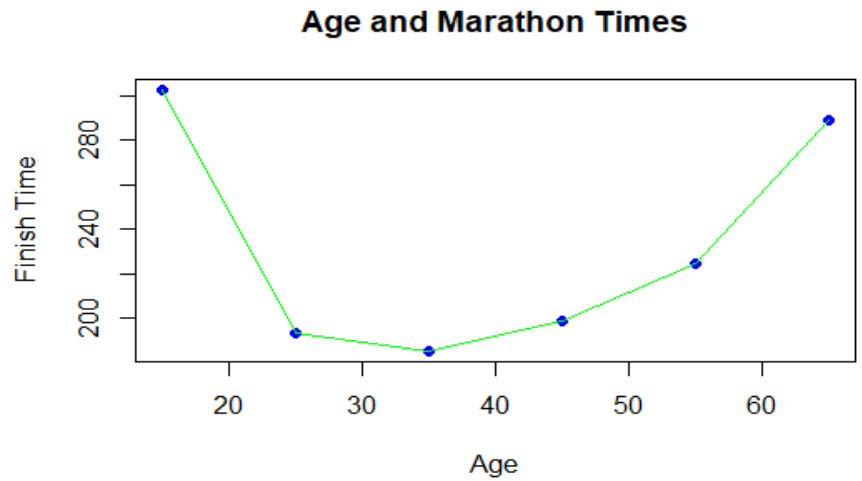
Linear Regression line: $\hat{y} = 227.97 + 0.10x$

Residual Plot:



Conclusion

- It is clear that no straight line can do a reasonable job of describing the relationship between x and y.
- However, the relationship can be described by a curve.
- Here the scatterplot looks like a parabola (the graph of a quadratic function).



- This suggests trying to find a quadratic function of the form

$$\hat{Y} = a + b_1 X + b_2 X^2$$

Polynomial Regression

- Fit the regression line using the quadratic equation: $\hat{Y} = a + b_1 X + b_2 X^2$
- $y_i - (a + b_1 x_i + b_2 x_i^2)$ is called the error of estimate or residual for y_i.
- Principle of least square: determine a, b_1 and b_2 so that

  $E = \sum (y_i - a - b_1 x_i - b_2 x_i^2)^2$, is minimum.

- Thus,

$$\frac{\partial E}{\partial a} = 0 = -2 \sum (y_i - a - b_1 x_i - b_2 x_i^2)$$

$$\frac{\partial E}{\partial b_1} = 0 = -2 \sum (y_i - a - b_1 x_i - b_2 x_i^2) x_i$$

$$\frac{\partial E}{\partial b_2} = 0 = -2 \sum (y_i - a - b_1 x_i - b_2 x_i^2) x_i^2$$

- Hence,

$$\sum y_i = na + b_1 \sum x_i + b_2 \sum x_i^2$$

$$\sum x_i y_i = a \sum x_i + b_1 \sum x_i^2 + b_2 \sum x_i^3$$

$$\sum x_i^2 y_i = a \sum x_i^2 + b_1 \sum x_i^3 + b_2 \sum x_i^4$$

Example: Marathon Data

- By solving the above system of equation, we get

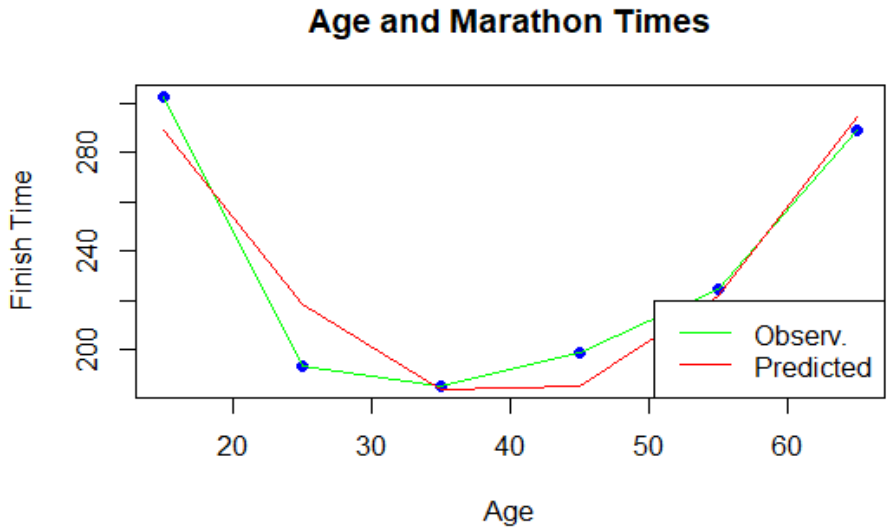
$$a = 462.0004453,$$
$$b_1 = -14.2054036$$
$$b_2 = 0.1788779$$

- Hence, the fitted line is : $\hat{y} = 462 - 14.2x + 0.179x^2$.
- The coefficient of Determination:

$$R^2_{XY} = 1 - \frac{SSResid}{SSTo} = 0.9211164$$

- Standard deviation about the least squares line:

$$S_e = \sqrt{\frac{SSResid}{n-3}} = 18.48125$$



Transformation

- An alternative to finding a curve to fit the data is to find a way to transform the x values and/or y values so that a scatterplot of the transformed data has a linear appearance.
- A **transformation** involves using a simple function of a variable in place of the variable itself.
- For example, instead of trying to describe the relationship between x and y, it might be easier to describe the relationship between $\sqrt{x}$ and y or between x and $\log(y)$.
- If we can describe the relationship between, say, $\sqrt{x}$ and y, we will still be able to predict the value of y for a given x value.
- Fitting the power curve: $Y = aX^b$.
- Fitting the exponential curve: $Y = ab^X, Y = ae^{bX}$

Commonly used Linear Transformation

Note: Standardization of the data is needed when units are different and large-scale variable value difference in the data.

Equation	Transformation		Changed Equation
	Y'	X'	
$Y = \beta_0 x^{\beta_1}$	$Y' = \log y$	$x' = \log x$	$Y' = \log \beta_0 + \beta_1 x'$
$Y = \beta_0 e^{\beta_1 x}$	$Y' = \ln y$	$x' = x$	$Y' = \ln \beta_0 + \beta_1 x'$
$Y = \beta_0 + \beta_1 \log x$	$Y' = y$	$x' = \log x$	$Y' = \beta_0 + \beta_1 x'$
$Y = \frac{x}{\beta_0 x - \beta_1}$	$Y' = 1/y$	$x' = 1/x$	$Y' = \beta_0 - \beta_1 x'$

MODELING NONLINEAR RELATIONS using R

MODELING NONLINEAR RELATIONS

The linear regression is fast and powerful tool to model complex phenomena.

But makes several assumptions about the data including the assumption of linear relationship exists between predictors and response variable.

When these assumptions are violated, the model breaks down quickly.

MODELING NONLINEAR RELATIONS

The linear model $y = x\beta + \varepsilon$ is general model

Can be used to fit any relationship that is linear in the unknown parameter β

Examples:

$$y = \beta_0 + \beta_1 x_1 + - - - + \beta_k x_k + \varepsilon$$

$$y = \beta_0 + \beta_1 x_1 + \beta_{11} x_1^2 + \varepsilon$$

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{11} x_1^2 + \beta_{22} x_2^2 + \beta_{12} x_1 x_2 + \varepsilon$$

In general

$$y = \beta_0 + \beta_1 f(x) + \varepsilon$$

where f(x) can be 1/x, $\sqrt{x}$, log(x), e^x , etc

MODELING NONLINEAR RELATIONS

Detection of non linear relation between predictor Xs and response variable Y

Scatter Plot:

The plotted points are not lying lie in a straight line is an indication of non linear relationship between predictor and dependant variable.

Component Residual Plots:

An extension of partial residual plots.

Partial residual plots are the plots of residuals of one predictor against dependant variable.

Component residual plots (crplots) adds a line indicating where the best fit line lies.

A significant difference between the residual line and the component line indicate that the predictor does not have a linear relationship with the dependent variable.

MODELING NONLINEAR RELATIONS

Example : The data given in Nonlinear_Thrust.csv represent the thrust of a jet turbine engine (y) and 3 predictor variables: x_1 = fuel flow rate, x_2 = pressure, and x_3 = exhaust temperature. Develop a suitable model for thrust in terms of the predictor variables.

Read Data

```
> attach(mydata)
> cor(mydata)
```

	x1	x2	x3	y
x1	1.00	0.40	-0.20	0.54
x2	0.40	1.00	-0.30	-0.36
x3	-0.20	-0.30	1.00	0.35
y	0.54	-0.36	0.35	1.00

There is no strong correlation between y and x's

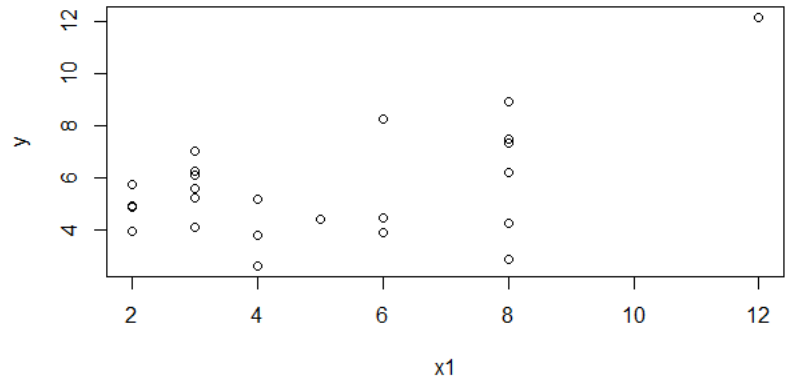
MODELING NONLINEAR RELATIONS

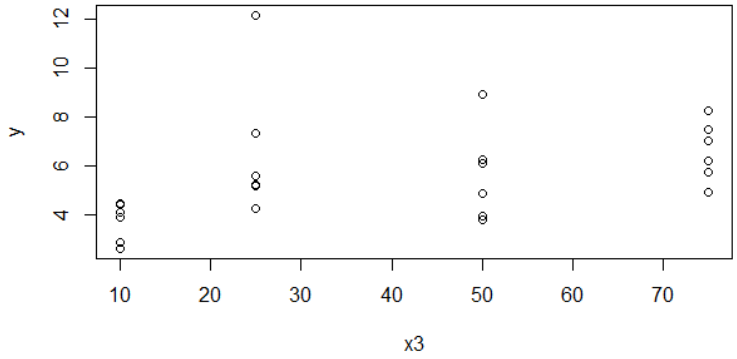
Draw Scatter plots

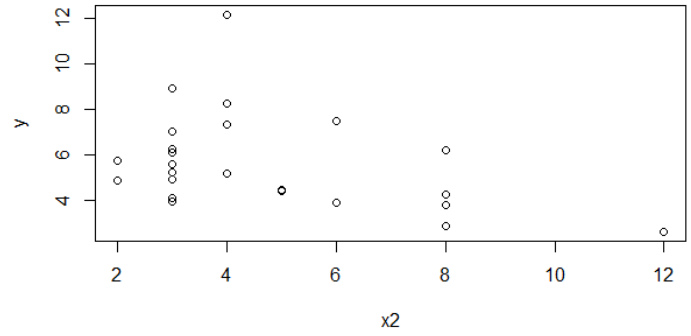
> plot(x1,y)

> plot(x2,y)

> plot(x3,y)







There is no strong correlation between y and x's

MODELING NONLINEAR RELATIONS

Develop the model

```
> mymodel = lm(y ~ x1 + x2 + x3, data = mydata)
> summary(mymodel)
```

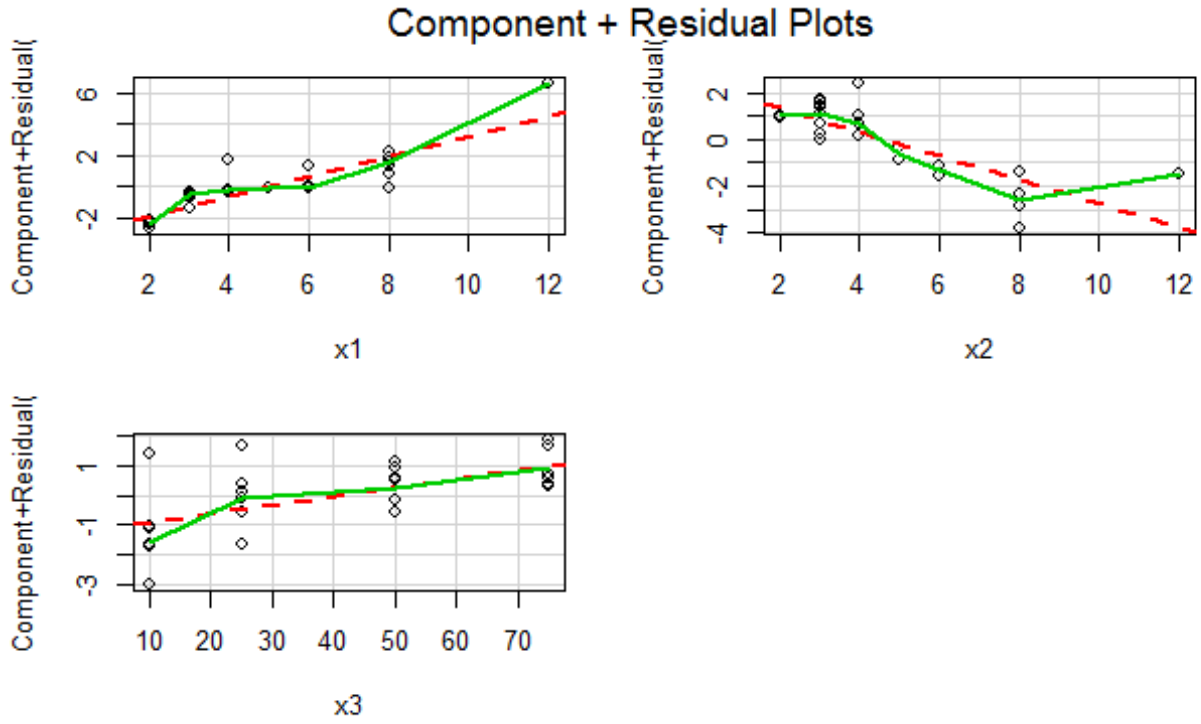
	Estimate	Std. Error	t	p value
(Intercept)	3.58315	0.726839	4.93	0.0001
x1	0.651547	0.0855	7.62	0.0000
x2	-0.509866	0.097132	-5.249	0.0000
x3	0.028888	0.009021	3.202	0.00428

R^2	0.786
Adjusted R^2	0.7563

MODELING NONLINEAR RELATIONS

Develop the model

```
> library(car)
> crPlots(mymodel)
```

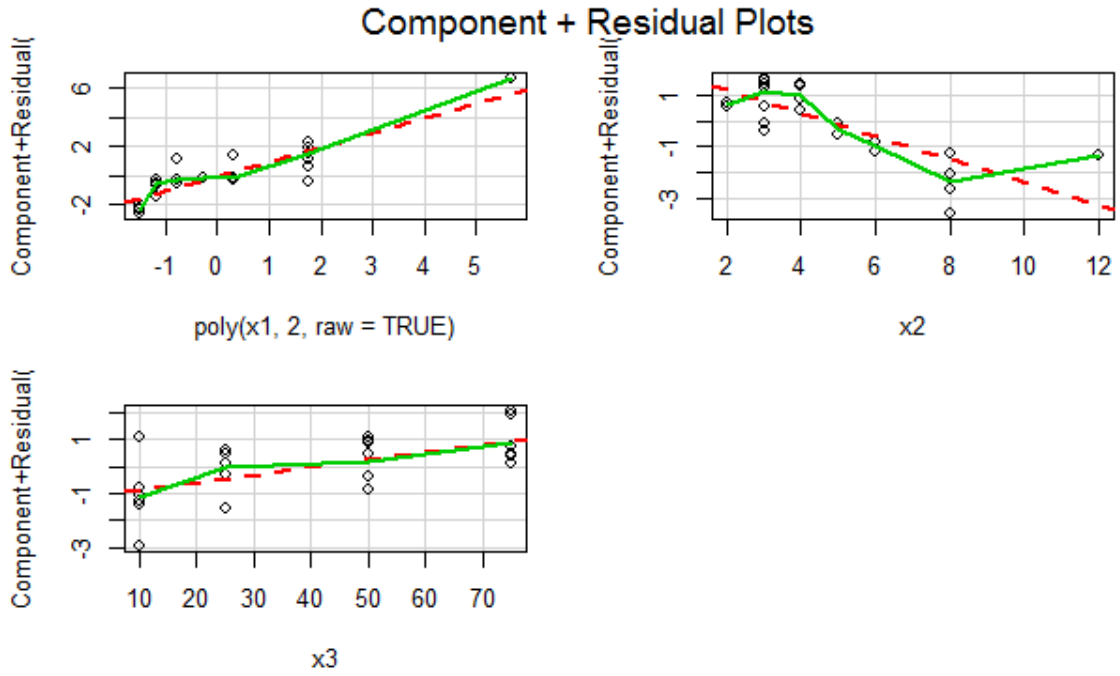


Since the best fit line different from residual line, it is possible improve the model by adding higher order terms

MODELING NONLINEAR RELATIONS

Develop the model

```
> mymodel = lm(y ~ poly(x1, 2, raw = TRUE) + x2 + x3, data = mydata)
> crPlots(mymodel)
```

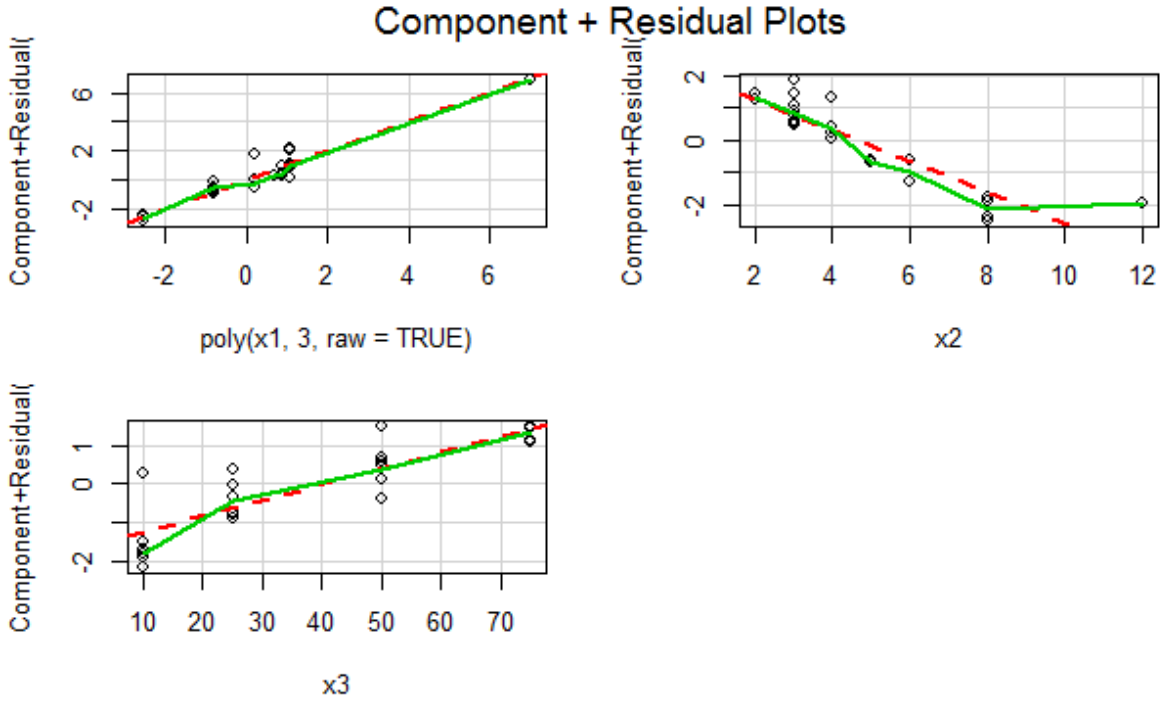


Since the best fit line different from residual line, it is possible improve the model by adding higher order terms

MODELING NONLINEAR RELATIONS

Develop the model

```
> mymodel = lm(y ~ poly(x1, 3, raw = TRUE) + x2 + x3, data = mydata)
> crPlots(mymodel)
```

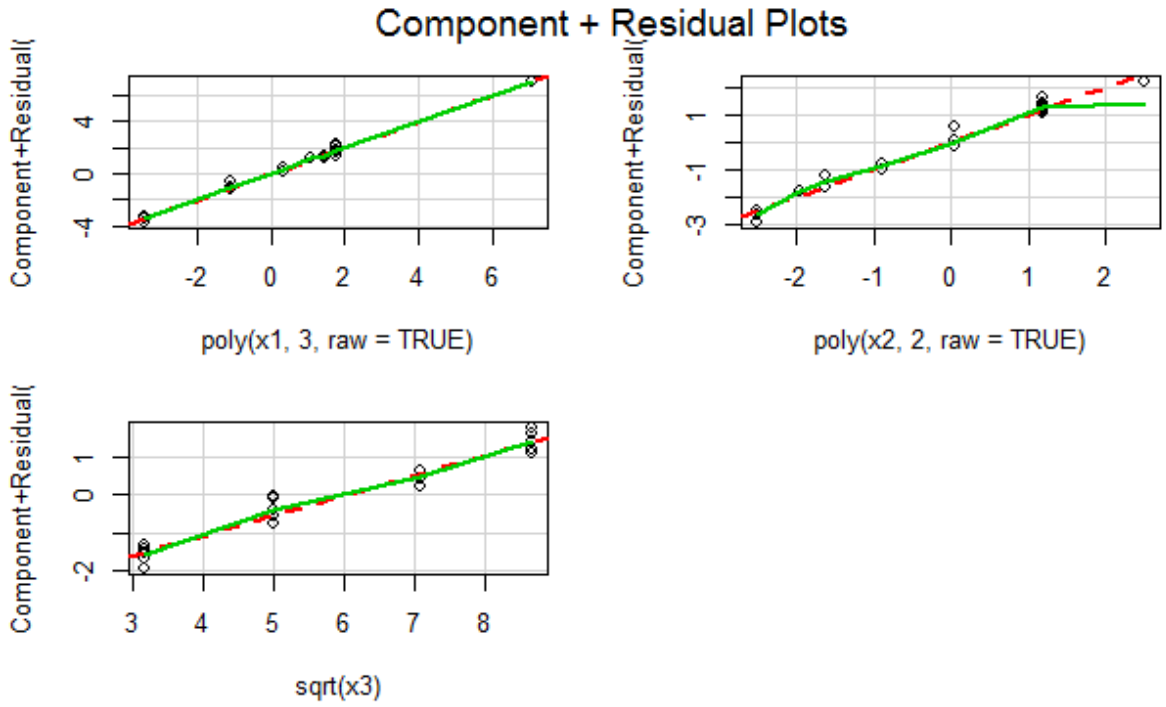


Since the best fit line is more or less overlapping residual line, hence adding square and cube terms of x1 will improve the model. Similarly add additional terms or functions of x2 and x3 to improve the model

MODELING NONLINEAR RELATIONS

Develop the model: Final Model

```
> mymodel = lm(y ~ poly(x1, 3, raw = TRUE) + poly(x2, 2, raw = TRUE) + sqrt(x3), data = mydata)
> crPlots(mymodel)
```



MODELING NONLINEAR RELATIONS

Develop the model: Final Model

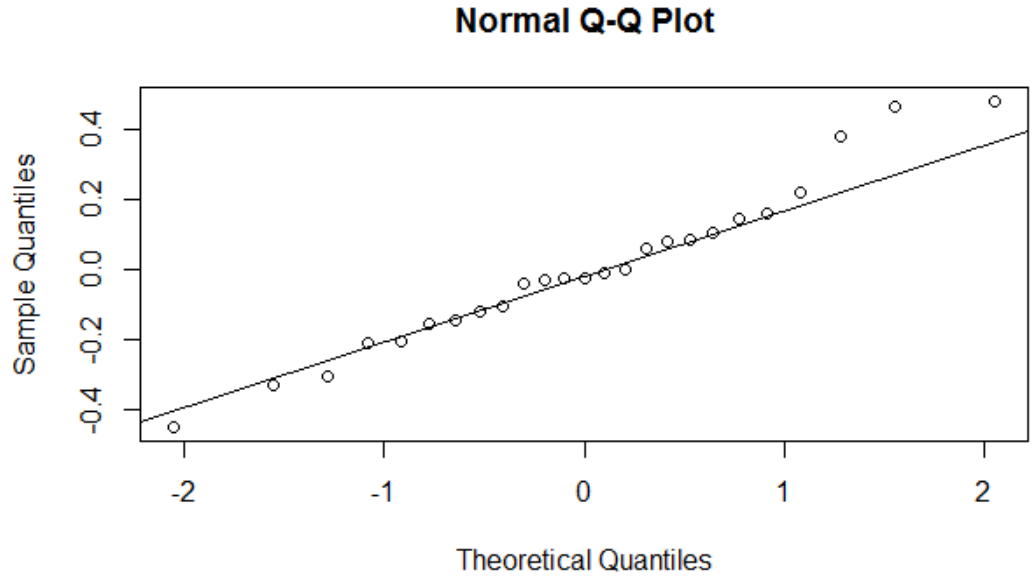
	Estimate	Std. Error	t	p value
(Intercept)	-3.48301	0.705793	-4.935	0.000107
x_1	5.503467	0.36278	15.17	0.0000
x_1^2	-0.77878	0.056814	-13.708	0.0000
x_1^3	0.037516	0.002685	13.971	0.0000
x_2	-1.81437	0.146304	-12.401	0.0000
x_2^2	0.097886	0.010374	9.435	0.0000
$\sqrt{x_3}$	0.527417	0.030664	17.2	0.0000

R^2	0.9881
Adjusted R^2	0.9841

MODELING NONLINEAR RELATIONS

Develop the model: Final Model

```
> res = residuals(mymodel)
> qqnorm(res)
> qqline(res)
> shapiro.test(res)
```



Shapiro-Wilk test for Normality	
w	0.9704
p value	0.6569

REGRESSION SPLINES

REGRESSION SPLINES

Spline

A continuous function formed by connecting linear segments

A function constructed piecewise from polynomial functions

Knots

The points where the segments are connected

Spline of degree D

A function formed by connecting polynomial segments of degree D so that

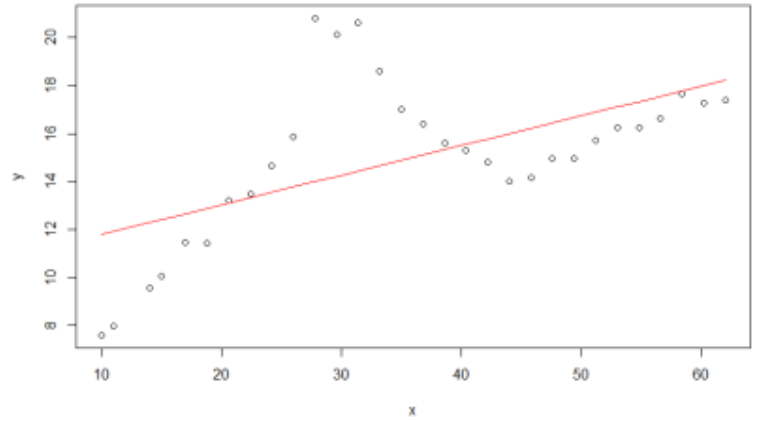
- Function is continuous
- Function has D – 1 continuous derivatives

Usage

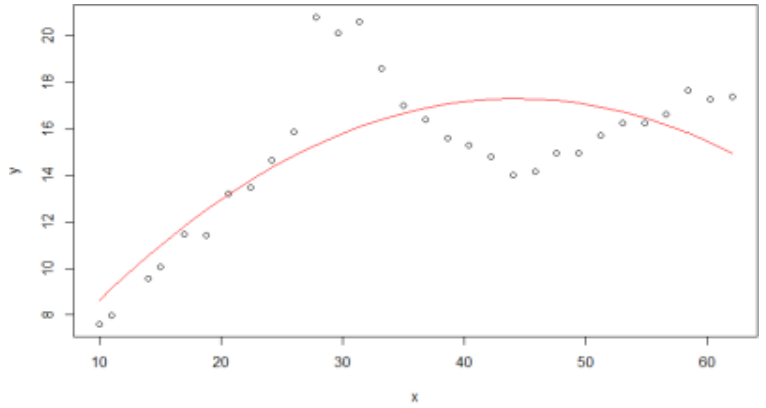
Develop models when relationship between y and x's is piecewise polynomial

REGRESSION SPLINES

y vs x (linear)

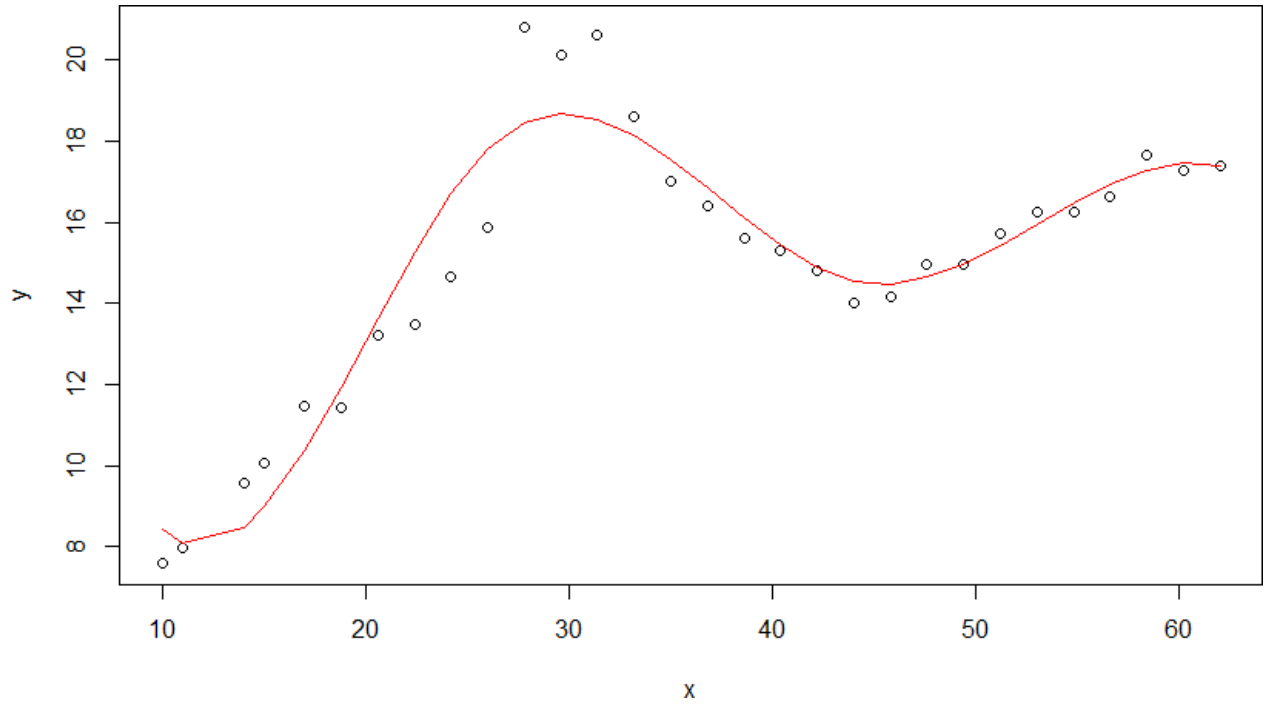


y vs x (Polynomial)



REGRESSION SPLINES

y vs x (Piecewise polynomial - Spline)



REGRESSION SPLINES

Example 1: The data on defect finding rate (design phase) and the corresponding defect finding rate (coding phase) of 20 similar projects is given in Reg_Spline_DFR.csv. Fit a suitable model to predict defect finding rate in coding phase in terms of defect finding rate in design phase?

Reading data

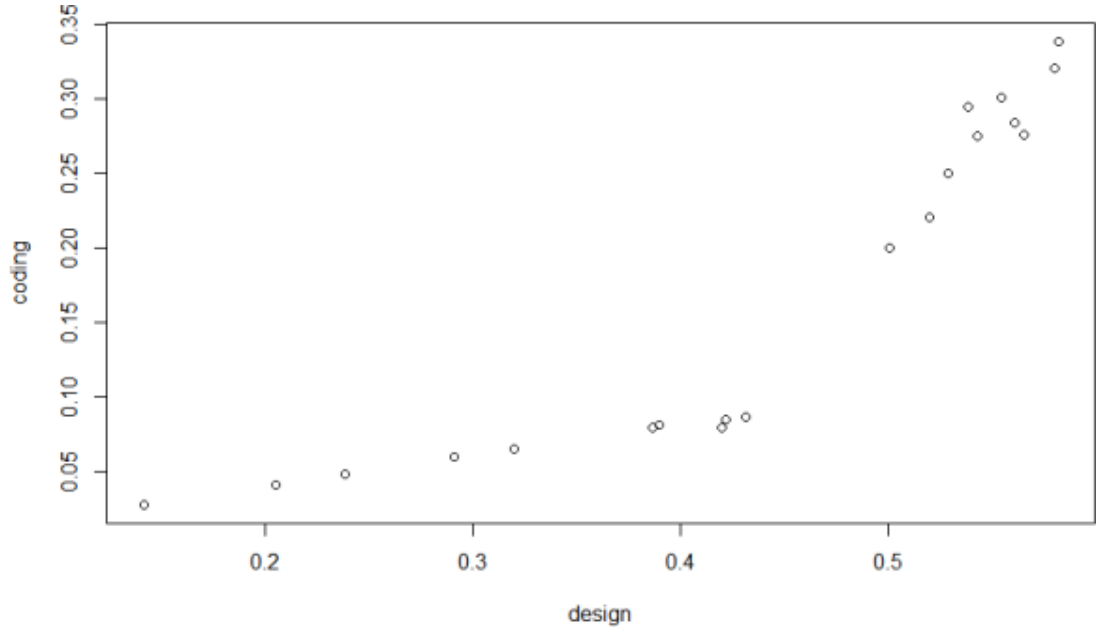
```
> design = mydata$Design
> coding = mydata$Coding
> plot(design, coding)
```

REGRESSION SPLINES

Example 1:

Exploring the relationship

> plot(design, coding)



REGRESSION SPLINES

Example 1:

Fitting a linear model

```
> mymodel = lm(coding ~ design)
> summary(mymodel)
```

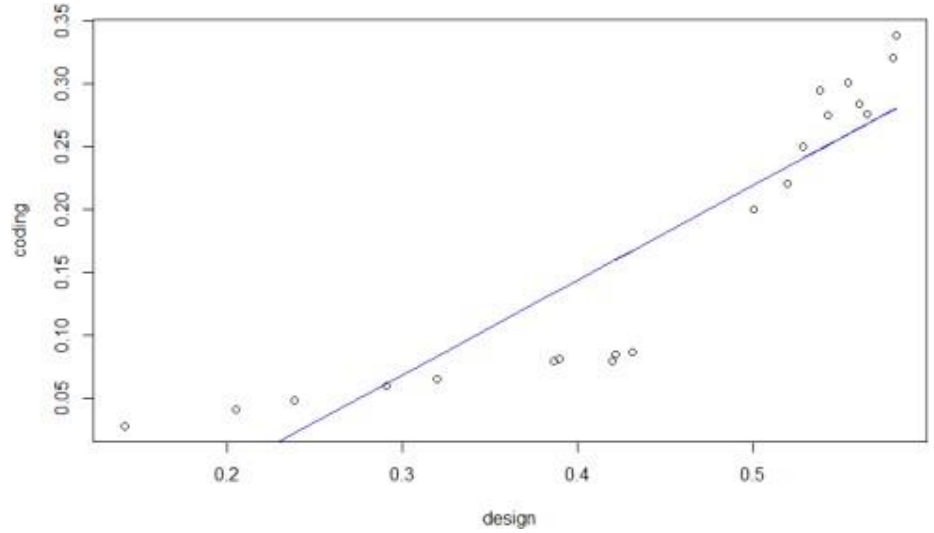
Statistics	Value
R^2	0.7862
R^2 adjusted	0.7744
F Statistics	66.21
P value	0.0000

REGRESSION SPLINES

Example 1:

Plotting the model

```
> pred = predict(mymodel)
> plot(design, coding)
> lines( design, pred, col ="blue")
```



REGRESSION SPLINES

Example 1:

Introducing knot at design = 0.44

```
> design44 = design - 0.44
> design44[design44 < 0] = 0
```

Fitting linear spline model

```
> mymodel = lm(coding ~ design +design44)
> summary(mymodel)
```

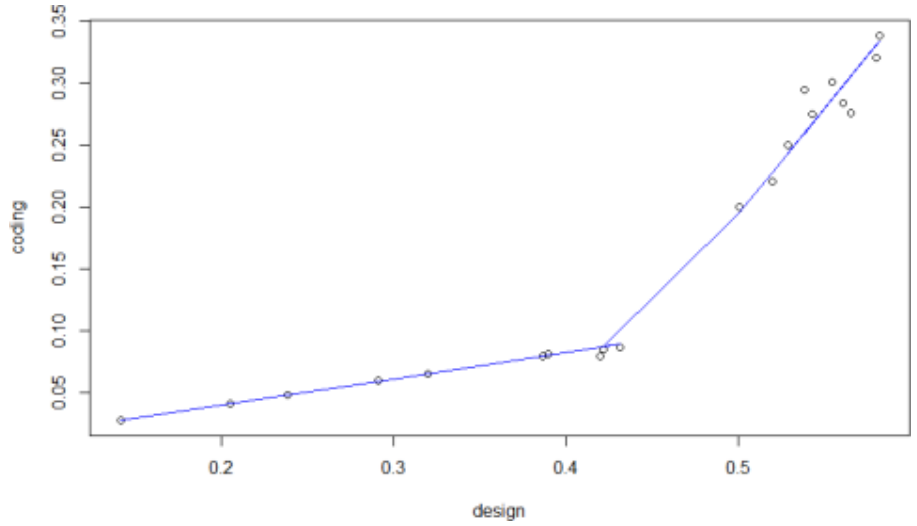
Statistics	Value
R^2	0.9823
R^2 adjusted	0.9802
F Statistics	472.2
P value	0.000

REGRESSION SPLINES

Example 1:

Plotting the linear spline model

```
> pred = predict(mymodel)
> plot(design, coding)
> lines(design, pred, col ="blue")
```



Note: Model is good but not a continuous function

REGRESSION SPLINES

Example 1:

Fitting cubic spline model

```
> designsq =design^2
> designcb =design^3
> design44cb = design44^3
> mymodel = lm(coding ~ poly(design, 3, raw =TRUE) + design44cb)
> summary(mymodel)
```

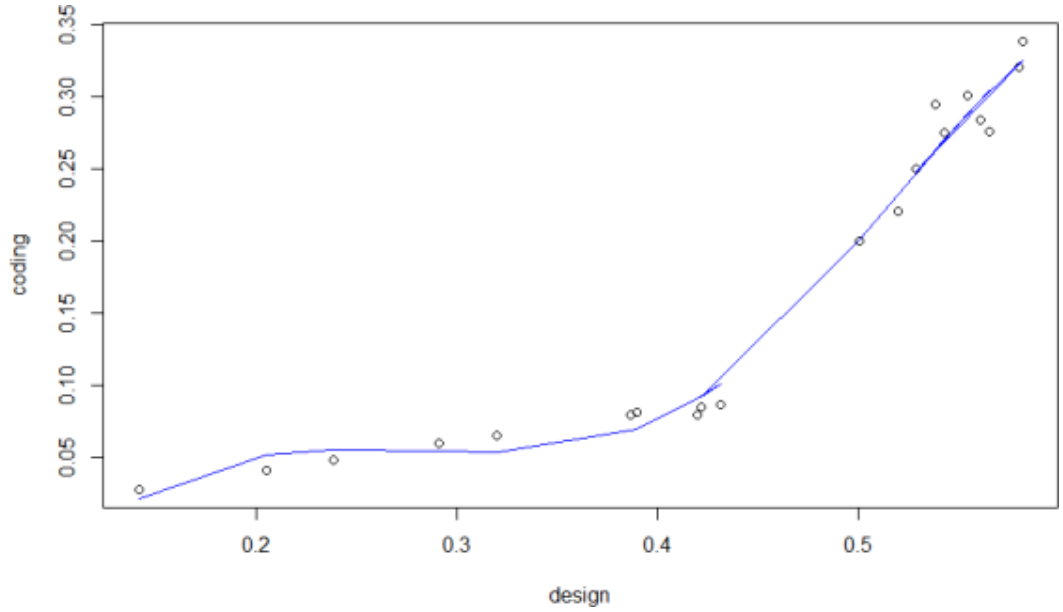
Statistics	Value
R^2	0.9782
R^2 adjusted	0.9724
F Statistics	168.5
P value	0.000

REGRESSION SPLINES

Example 1:

Plotting the linear spline model

```
> pred = predict(mymodel)
> plot(design, coding)
> lines(design, pred, col ="blue")
```

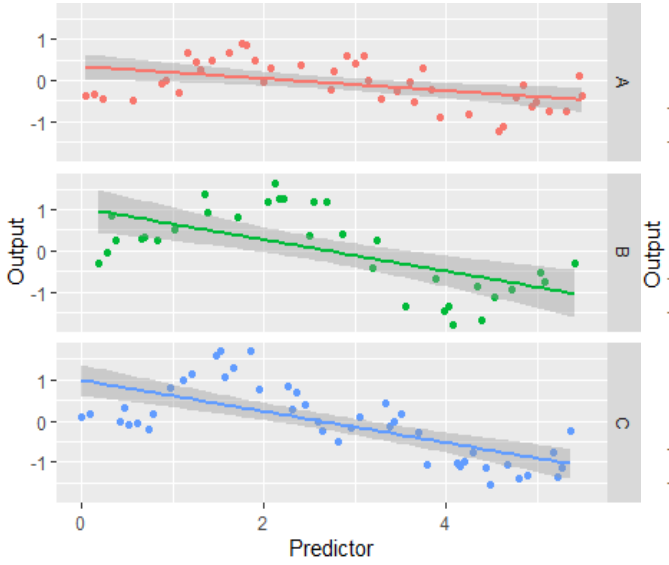


Homework: EXPLORE Multivariate Adaptive Regression Splines (**MARS**)

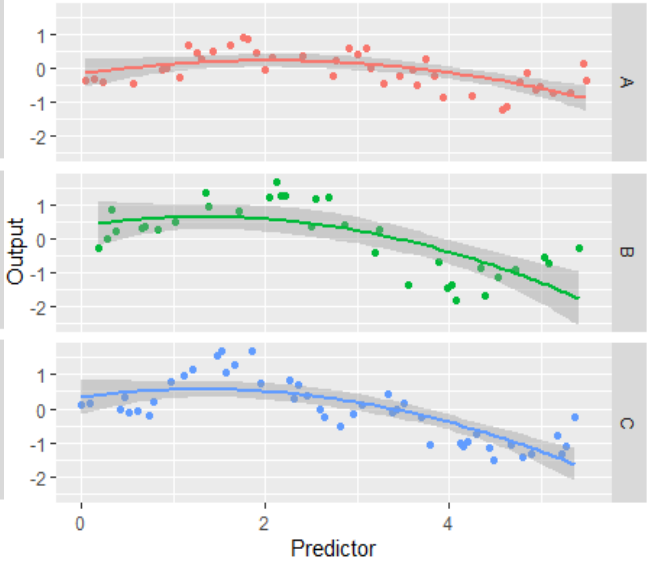
Regression in R

Type of regression	Number of knots in splines	Method used in R	Formula
Linear regression		`lm`	default, y ~ poly(x,1)
Quadratic regression		`lm`	Y ~ poly(x,2)
Cubic regression		`lm`	Y ~ poly(x,3)
Natural splines	2	`gam`	splines::ns(x, 2)
Natural splines	3	`gam`	splines::ns(x, 3)
Natural splines	30	`gam`	splines::ns(x, 30)
B-Splines	3	`gam`	splines::bs(x, 3)
B-Splines	30	`gam`	splines::bs(x, 30)

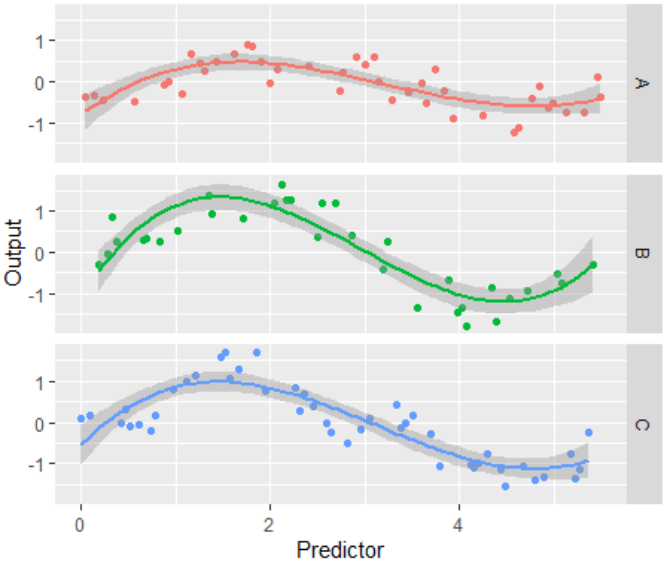
Linear Regression



Quadratic Regression

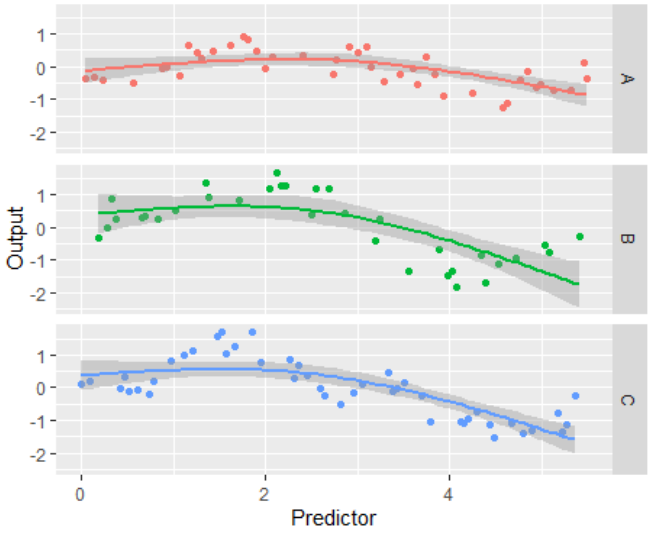


Cubic Regression

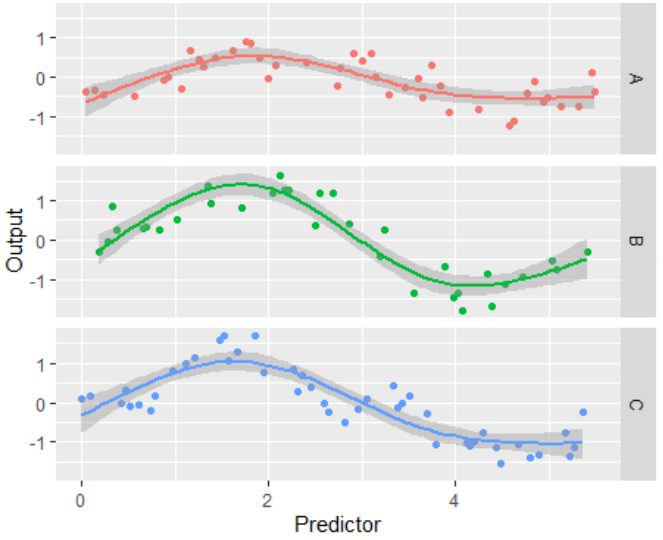


From the above graphs, we can conclude that the data is better modelled using a cubic regression function rather than a linear regression or a quadratic regression.

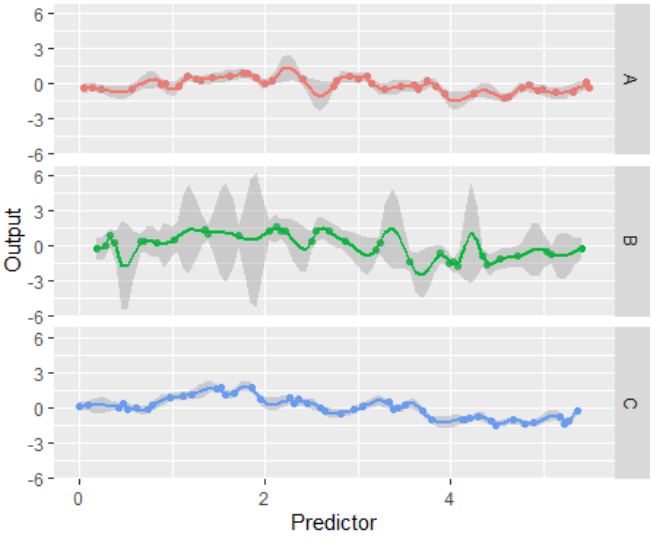
Natural splines with 2 knots



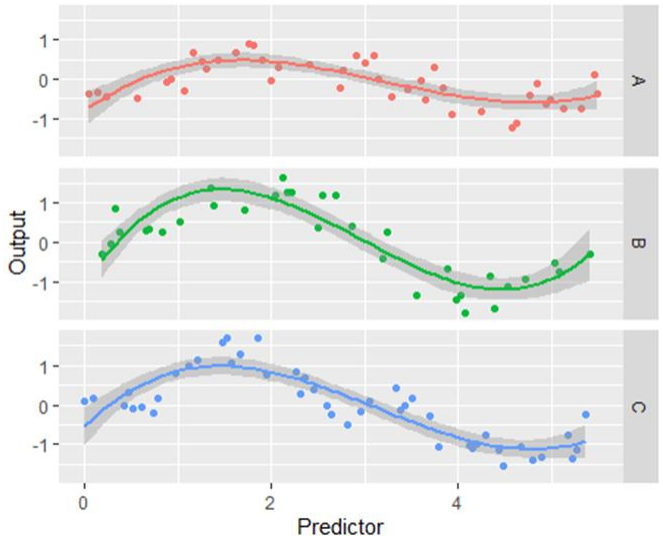
Natural splines with 3 knots



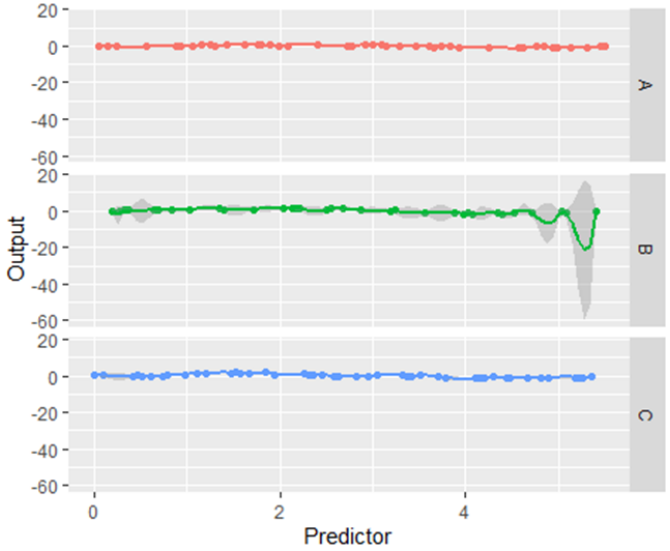
Natural splines with 30 knots



B splines with 3 knots



B splines with 30 knots



Logistic Regression

Introduction

- Regression analysis and logistic regression
- Concept of logistic regression
- Types of logistic regression

Regression Analysis and Logistic Regression

- **Regression analysis**
  - Based on the principle of **Least Square Estimation (LSE)**
    - The parameters are chosen to minimize the sum of squared errors (SSE)
  - Minimizes error in prediction
    - If the error distribution is normal with constant variance, the LSE estimates the parameters accurately; that is, model is the best possible and with the smallest standard errors
  - Applicable when dependent variable follows normal distribution
- **Logistic regression analysis**
  - When a dependent variable does not follow normal distribution
  - Value of a dependent variable may be with 2, 3 or a fewer more outcomes

Regression Analysis and Logistic Regression

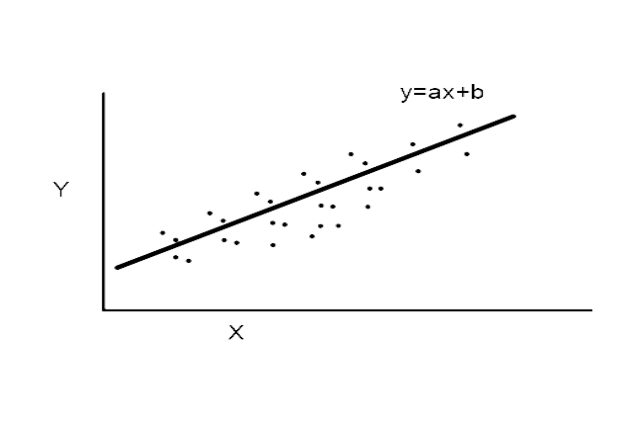
- **Logistic regression**
  - Considers maximum likelihood estimation (MLE) to give better result.
  - In MLE, the likelihood is the probability of the observed data set given a set of proposed values for the parameters.
  - The principle of MLE is to estimate parameters by choosing parameter values that give the largest possible likelihood.

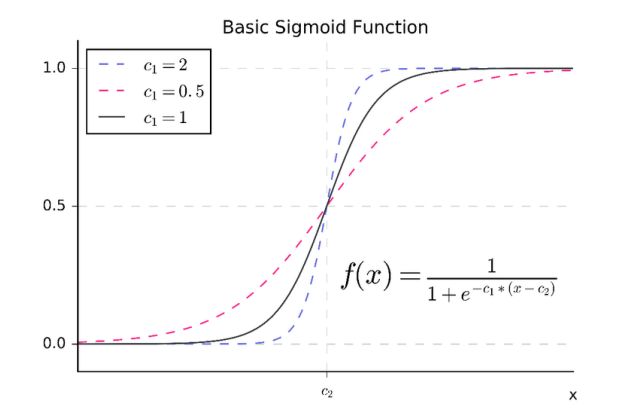
Note

- Regression analysis predicts a value of a dependent variable
- Logistic regression predicts the probability of a given value of a dependent variable
- Both estimates their respective model parameters

Regression Analysis and Logistic Regression

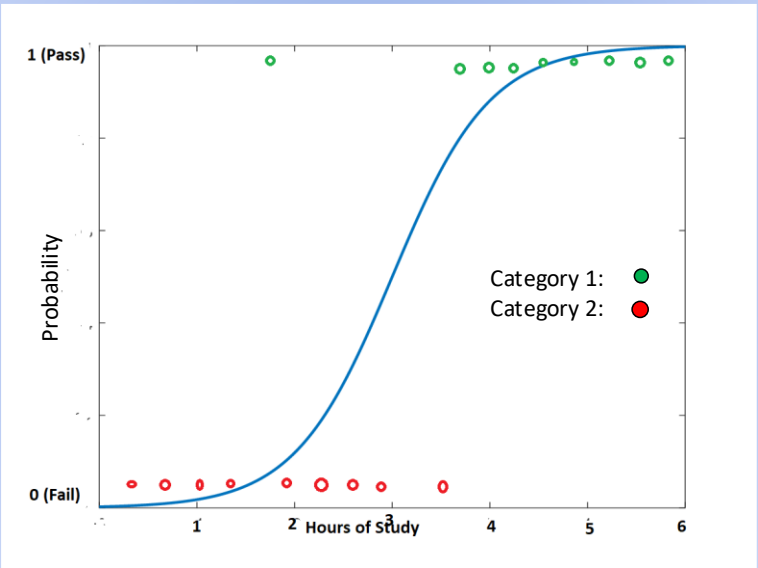
A Regression model and Logistic Regression model





An Example

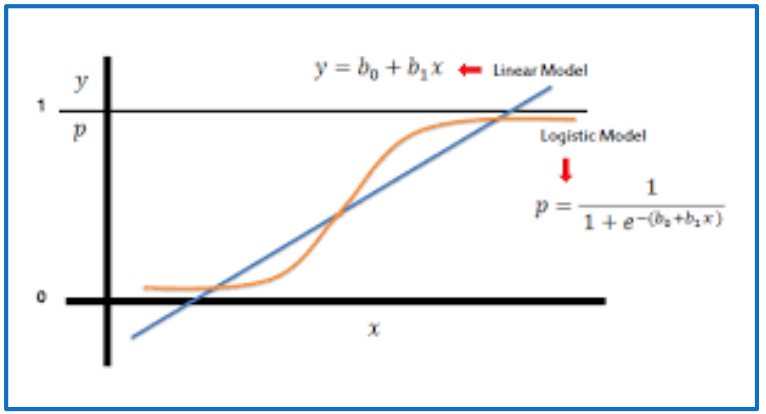
Hours (x_i)	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass (y_i)	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1



Concept of Logistic Regression

Introduction

- Developed and popularized primarily by *Joseph Berkson* in *(1944)*, where he coined the term **logit.**
- Logistic regression is a statistical method.
  - uses a logistic function to model a binary dependent variable.
  - although many more complex extensions exist.
- Logistic regression (or logit regression) estimates the parameters of a logistic model.



Concept of Logistic Regression

Introduction

- A binary logistic model has a dependent variable with two possible values, such as ***Pass*** ***or*** ***Fail***, ***Happy*** ***or*** ***Sad*** etc.
- It is represented by an indicator variable, where the two values are labeled '1' and '0'

1 0

Concept of Logistic Regression

What is logistic regression?

- The logistic function, takes the following form

$$p(x_1, \ldots, x_m) = p_x = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_m x_m}}{1+e^{\beta_0+\beta_1 x_1+\cdots+\beta_m x_m}}$$

- Binary regression: A case when p has two outcomes: success and failure
- It defines **odds**, which is the ratio of the probability of success and failure

$$odds = \frac{p_x}{1-p_x} = e^{\beta_0+\beta_a x_1+\cdots+\beta_m x_m}$$

- The logarithm of the odds (called logit) is

$$t = ln(odds) = \beta_0 + \beta_1 x_1 + \cdots + \beta_m x_m$$

$$t = ln\left(\frac{p_x}{1-p_x}\right)$$

$$p_x = \frac{e^t}{1+e^t}$$

An Illustration

A sample is collected to examine the effect of toxic substance on tumor. A subject is examined for the toxic content in the body and then the presence (1) or absence (0) of tumors. The independent variable is the concentration of the toxic substance "Conc" . The number of subjects at each concentration (N) and the number having tumors "Tumor" is shown in the table.

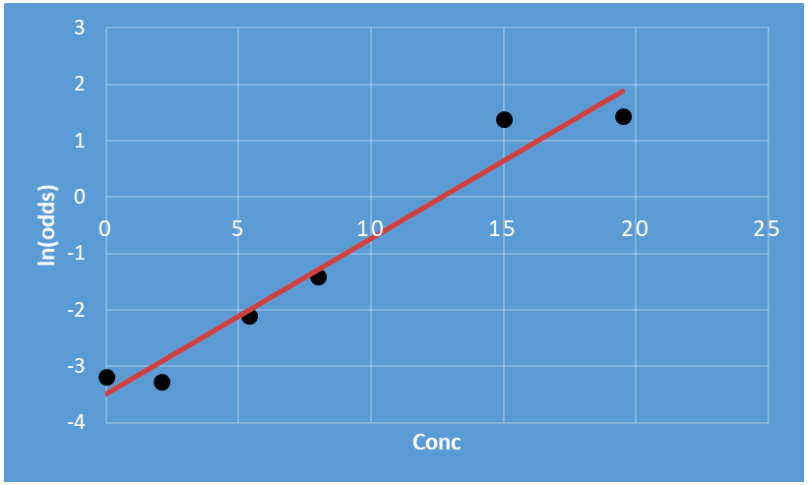
Odds and ln(odds) are also included in the table.

Conc	N	Tumor	Odds	ln(odds)
0.0	50	2	0.0417	-3.18
2.1	54	5	0.1020	-3.28
5.4	46	5	0.1220	-2.10
8.0	51	10	0.2439	-1.41
15.0	50	40	4.0000	+1.39
19.5	52	42	4.2000	+1.44

$$t = ln\frac{p}{1-p}$$

An Illustration

Conc	N	Tumor	Odds	ln(odds)
0.0	50	2	0.0417	-3.18
2.1	54	5	0.1020	-3.28
5.4	46	5	0.1220	-2.10
8.0	51	10	0.2439	-1.41
15.0	50	40	4.0000	+1.39
19.5	52	42	4.2000	+1.44



Here, we find a relation between t (ln(odds) and x (Conc).

$$t = \beta_0 + \beta_1 x$$

Thus, logit is a linear function

For this, it can be calculated as

β_0= -3.204 and β_1= 0.2628

An Illustration

Here, we find a relation between t (ln(odds) and x (Conc).

$$t = \beta_0 + \beta_1 x$$

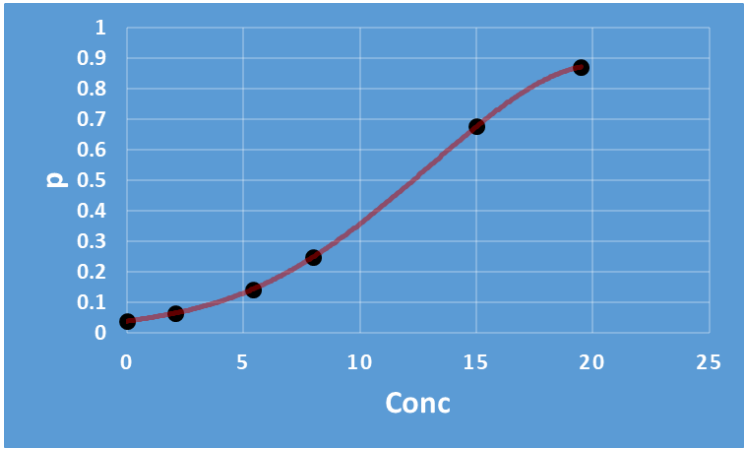
Thus, logit is a linear function

For this, it can be calculated as

β_0= -3.204 and β_1= 0.2628

For an example, a subject exposed to a concentration of 10, has an estimated probability of tumor is

$$p_{10} = \frac{e^t}{1+e^t} = 0.36$$



Concept of Logistic Regression

Logit in Logistic Regression

- The log-odds (the logarithm of the odds) is a linear combination of
  - one or more independent variables ("predictors").
  - the independent variables can each be a continuous variable (any real value).
- The probability of the value can vary between 0 (such as certainly false) and 1 (such as certainly true).

Concept of Logistic Regression

Output of logistic function

- Increasing one of the independent variables, multiplicatively scales the odds of the given outcome at a constant rate, with each independent variable having its own parameter;
  - for a binary dependent variable this generalizes the odds ratio.
- Binary logistic regression: The dependent variable has two levels (categorical).
- Multinomial logistic regression: Outputs with more than two levels.
- Ordinal logistic regression: if the multiple categories are ordered, then it is called ordinal logistic regression.

Logistic Regression as Classifier

Logistic regression models the probabilities for classification problems with possible outcomes.

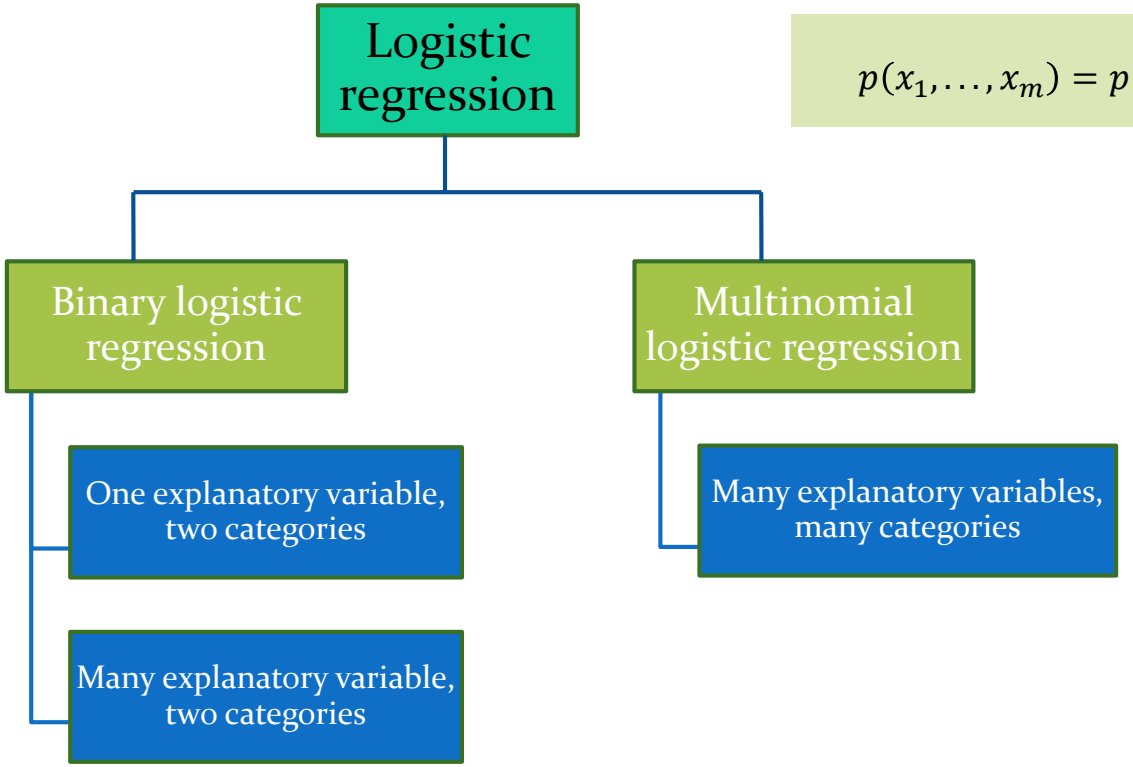
- It's an extension of the linear regression model for classification problems.

- The logistic regression simply models probability of output in terms of input and does not perform statistical classification (it is not a classifier).

- Although it can be used to make a classifier.
  - By choosing a cut-off value and classifying inputs with probability greater than the cutoff as one class, below the cut-off as the other

Logistic Regression Techniques

Types of Logistic Regression



$$p(x_1, \ldots, x_m) = p = \frac{e^{\beta_0+\beta_1 x_1+\cdots+\beta_m x_m}}{1 + e^{\beta_0+\beta_1 x_1+\cdots+\beta_m x_m}}$$

Types of Logistic Regression

Case 1: One explanatory variable, two categories

- Explanatory variable:
  X: Hours Study
- Outcome: Pass or Fail

Case 2: Many explanatory variable, two categories

- Explanatory variable
  X1: Hours Study X2: 12th % Marks
- Outcome: Pass or Fail

Case 3: Many explanatory variable, many categories

- Explanatory variable:
  X1: Hours Study X2: 12th % Marks X3: Age
- Outcome: Bad, Good, Excellent

Binary Logistic Regression

- Logistic Regression Analysis
  - Binary Logistic Regression
    - One explanatory variable, two categories

One explanatory variable, two categories

One Explanatory Variable, Two Categories

A group of 20 students spends between 0 and 6 hours studying for an exam.

The table shows the number of hours each student spent studying, and whether they passed (1) or failed (0).

Hours (x_i)	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass (y_i)	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

How does the number of hours spent studying affect the probability of the student passing the exam?

One Explanatory Variable, Two Categories

How does the number of hours spent studying affect the probability of the student passing the exam?

Hours (x_i)	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass (y_i)	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

Note

- The reason for using logistic regression for this problem is that the values of the dependent variable, pass and fail, while represented by "1" and "0", are not cardinal numbers.
- If the problem was changed so that pass/fail was replaced with the grade 0 to 100 (cardinal numbers), then simple regression analysis could be used.

One Explanatory Variable, Two Categories

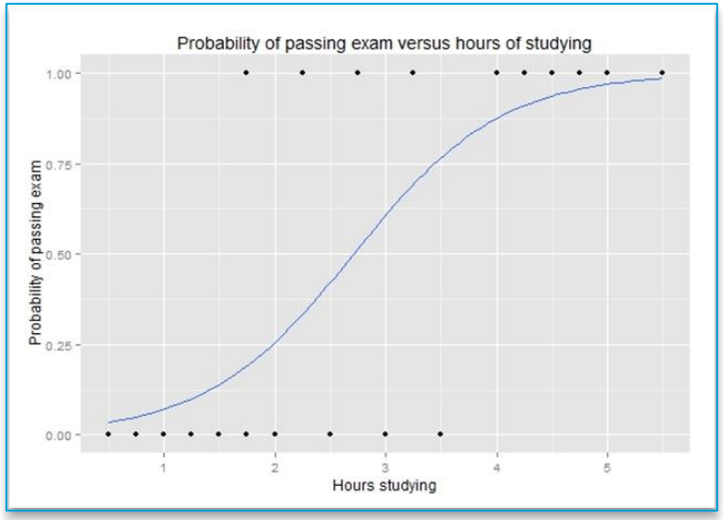
Hours (x_i)	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass (y_i)	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

- We wish to fit a logistic function to the data consisting of the hours studied (x_i) and the outcome of the test ($y_i = 1$ for pass, 0 for fail).
- The data points are indexed by the subscript i which runs from $i = 1$ to $20\ (= n)$. The x variable is called the "explanatory variable", and the y variable is called the "**categorical variable**" consisting of two categories: "**pass**" or "**fail**" corresponding to the categorical values 1 and 0, respectively.

One Explanatory Variable, Two Categories

$$p(x) = \frac{1}{1 + e^{-(x-\mu)/s}}$$

$$t = log_b \frac{p}{1-p}$$



- Graph of a logistic regression curve fitted to the (x_i, y_i) data. The curve shows the probability of passing an exam versus hours studying.

One Explanatory Variable, Two Categories

- The logistic function is of the form:

$$p(x) = \frac{1}{1 + e^{-(x-\mu)/s}}$$

where μ is a location parameter (the midpoint of the curve, where $p(\mu) = {}^1/_2$) and s is a scale parameter.

- This expression may be rewritten as:

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

where $\beta_0 = -{}^{\mu}/_{S}$ and is known as the intercept, and $\beta_1 = {}^1/_S$ known as slope.

- We may define the "fit" to y_i at a given x_i as:

$$p_i = p(x_i)$$

One Explanatory Variable, Two Categories

- The p_i are the probabilities that the corresponding y_i will be unity and $1 - pi$ are the probabilities that they will be zero.
- We wish to find the values of β_0 and β_1 which give the *"best fit"* to the data.
  - In the case of linear regression, the sum of the squared deviations of the fit from the data points (y_i) is taken as a measure of the goodness of fit, and the best fit is obtained when that function is minimized.

One Explanatory Variable, Two Categories

- In the case of logistic regression, the measure of goodness of fit is given by the likelihood function, which is the probability that the given data set is produced by a particular logistic function:

$$L = \prod_{i:y_i=1} p_i \prod_{i:y_i=0} (1 - p_i)$$

and the best fit is obtained for those choices of β_0 and β_1 where L is maximized.

One Explanatory Variable, Two Categories

- The maximum of L will also be the maximum of the log-likelihood ℓ, defined as the logarithm of L :

$$\ell = \sum_{i:y_i=1} \ln(p_i) + \sum_{i:y_i=0} \ln(1-p_i)$$

$$= \sum_{i=1}^{n} \left(y_i \ln(p_i) + (1-y_i)\ln(1-p_i)\right)$$

$$p_i = p(xi) = \frac{1}{1+e^{-}(\beta_0 + \beta_1 x_i)}$$

- Here, ℓ is nonlinear in β_0 and β_1. Determining their optimum values will require numerical methods.

One Explanatory Variable, Two Categories

- Note that one method of maximizing ℓ is to require the derivatives of ℓ with respect to β_0 and β_1 to be zero:

$$0 = \frac{\partial \ell}{\partial \beta_0} = \sum_{i=1}^{n} (y_i - p_i)$$

$$0 = \frac{\partial \ell}{\partial \beta_1} = \sum_{i=1}^{n} (y_i - p_i)x_i$$

and the maximization procedure can be accomplished by solving the above two equations for β_0 and β_1, which again, will generally require the use of numerical methods.

$$p_i = p(x_i) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_i)}}$$

One Explanatory Variable, Two Categories

Hours (x_i)	0.50	0.75	1.00	1.25	1.50	1.75	1.75	2.00	2.25	2.50	2.75	3.00	3.25	3.50	4.00	4.25	4.50	4.75	5.00	5.50
Pass (y_i)	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

The values of β_0 and β_1 which maximize ℓ using the above data are found to be:

$$\beta_0 = -4.07771 \text{ and } \beta_1 = 1.50465$$

which yields a value for μ of:

$$\mu = -\beta_0/\beta_1 = 2.71008$$

The logistic regression analysis gives the following output.

	Coefficient	Std. Error	z-value	p-value (Wald)
Intercept (β_0)	−4.0777	1.7610	−2.316	0.0206
Slope (β_1)	1.5046	0.6287	2.393	0.0167

- By the Wald test, the output indicates that hours studying is significantly associated with the probability of passing the exam ($p = 0.0167$)

One Explanatory Variable, Two Categories

	Coefficient	Std. Error	z-value	p-value (Wald)
Intercept (β_0)	−4.0777	1.7610	−2.316	0.0206
Slope (β_1)	1.5046	0.6287	2.393	0.0167

- The β_0 and β_1 coefficients may be entered into the logistic regression equation to estimate the probability of passing the exam.
- For example, for a student who studies 2 hours, entering the value $x = 2$ into the equation gives the estimated probability of passing the exam of 0.26 :

$$t = \beta_0 + 2\beta_1 = -4.0777 + 2 \times 1.5046 = -1.0685$$

$$p = \frac{1}{1 + e^{-t}} = 0.26 = \text{ Probability of passing exam}$$

One Explanatory Variable, Two Categories

- Similarly, for a student who studies 4 hours, the estimated probability of passing the exam is 0.87:

$$t = \beta_0 + 4\beta_1 = -4.0777 + 4 \times 1.5046 = 1.9407$$

$$p = \frac{1}{1+e^{-t}} = 0.87 = \text{Probability of passing exam}$$

- This table shows the probability of passing the exam for several values of hours studying.

Hours of study (x)	Passing exam		
	Log-odds (t)	Odds (e^t)	Probability (p)
1	−2.57	0.076 ≈ 1:13.1	0.07
2	−1.07	0.34 ≈ 1:2.91	0.26
μ=2.71…	0	1	0.5
3	0.44	1.55	0.61
4	1.94	6.96	0.87
5	3.45	31.4	0.97

One Explanatory Variable, Two Categories

- This simple model is known as *"binary logistic regression"*, and has one explanatory variable and a categorical variable which can assume one of two categorical values.

- The above example of binary logistic regression on one explanatory variable can be generalized to binary logistic regression on any number of explanatory variables $\boldsymbol{x_1}, \boldsymbol{x_2}, \ldots$ and any number of categorical values $\boldsymbol{y = 0, 1, 2, \ldots}$.

Many explanatory variable, two categories

- Logistic Regression Analysis
  - Binary Logistic Regression
    - Many explanatory variable, two categories

Many Explanatory Variable, Two Categories

- To begin with, we may consider a logistic model with M explanatory variables, $x_1, x_2 \ \ldots \ x_M$ and two categorical values ($y = 0$ and 1).
- For the simple binary logistic regression model, we assumed a linear relationship between the predictor variable and the log-odds (also called logit) of the event that $y = 1$.
- This linear relationship may be extended to the case of M explanatory variables:

$$t = \log_b \frac{p}{1-p} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_M x_M$$

where t is the log-odds and β_i are parameters of the model.

- An additional generalization has been introduced in which the base of the model ***b*** is not restricted to the **Euler number *e***.
  - In most applications, the base ***b*** of the logarithm is usually taken to be ***e***.
  - However, in some cases it can be easier to communicate results by working in **base 2** or **base 10**.

Many Explanatory Variable, Two Categories

- For a more compact notation, we will specify the explanatory variables and the $\boldsymbol{\beta}$ coefficients as $(\boldsymbol{M+1})$-dimensional vectors:

$$\boldsymbol{x} = \{x_0, x_1, x_2, \dots, x_M\}$$

$$\boldsymbol{\beta} = \{\beta_0, \beta_1, \beta_2, \dots, \beta_M\}$$

with an added explanatory variable $\boldsymbol{x_0 = 1}$. The logit may now be written as:

$$t = \sum_{m=0}^{M} \beta_m x_m = \beta \cdot x$$

Many Explanatory Variable, Two Categories

- Solving for the probability $\boldsymbol{p}$ that $\boldsymbol{y = 1}$ yields:

$$p(\boldsymbol{x}) = \frac{b^{\beta \cdot \boldsymbol{x}}}{1 + b^{\beta \cdot x}} = \frac{1}{1 + b^{-\beta \cdot x}} = S_b(t)$$

  where $\boldsymbol{S_b}$ is the sigmoid function with base $\boldsymbol{b}$.

- The above formula shows that once the β is fixed, we can easily compute the log-odds that $y = 1$ for a given observation.

- The main use-case of a logistic model is given an observation $\boldsymbol{x}$, estimate the probability $p(\boldsymbol{x})$ that $y = 1$.

Many Explanatory Variable, Two Categories

- The optimum beta coefficients may again be found by maximizing the log-likelihood.
- For K measurements, defining x_k as the explanatory vector of the $k-th$ measurement, and y_k as the categorical outcome of that measurement, the log likelihood may be written in a form very similar to the simple case of $M = 1$:

$$\ell = \sum_{k=1}^{K} y_k \log_b\left(p(\boldsymbol{x}_k)\right) + \sum_{k=1}^{K} (1 - y_k) \log_b\left(1 - p(\boldsymbol{x}_k)\right)$$

- Finding the optimum β parameters will require numerical methods.

Many Explanatory Variable, Two Categories

- One useful technique is to equate the derivatives of the log likelihood with respect to each of the β parameters to zero yielding a set of equations which will hold at the maximum of the log likelihood:

$$\frac{\partial \ell}{\partial \beta_m} = 0 = \sum_{k=1}^{K} y_k x_{mk} - \sum_{k=1}^{K} p(\boldsymbol{x}_k) x_{mk}$$

where x_{mk} is the value of the x_m explanatory variable from the *k-th* measurement.

Many Explanatory Variable, Two Categories

- Consider an example with $M = 2$ explanatory variables, $b = 10$, and coefficients $\beta_0 = -3$, $\beta_1 = 1$, and $\beta_2 = 2$ which have been determined.
- To be concrete, the model is:

$$t = \log_{10} \frac{p}{1-p} = -3 + x_1 + 2x_2$$

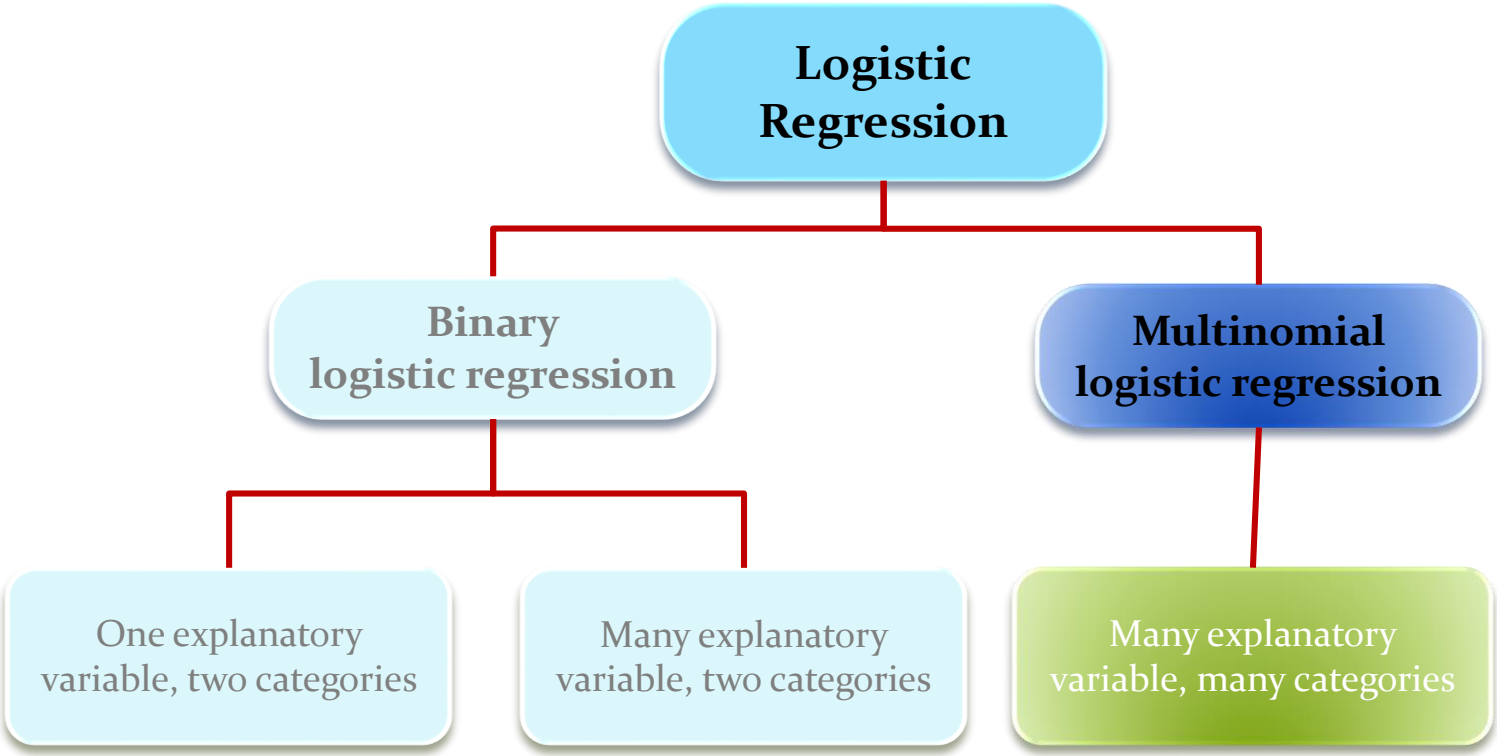
$$p = \frac{b^{\beta \cdot x}}{1 + b^{\beta \cdot x}} = \frac{b^{\beta_0 + \beta_1 x_1 + \beta_2 x_2}}{1 + b^{\beta_0 + \beta_1 x_1 + \beta_2 x_2}} = \frac{1}{1 + b^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2)}}$$

where p is the probability of the event that $y = 1$.

- This can be interpreted as follows:
  - $\beta_0 = -3$ is the *y*-intercept.
  - $\beta_1 = 1$ means that increasing x_1 by 1 increases the log-odds by 1.
  - $\beta_2 = 2$ means that increasing x_2 by 1 increases the log-odds by 2.

Multinomial Logistic Regression

Multinomial Logistic Regression



Many Explanatory Variable, Many Categories

- In the cases of two categories *(binomial logistic regression)*, the categories were indexed by "0" and "1", and we had two probability distributions:
  - The probability that the outcome was in *category 1* was given by $p(x)$ and
  - The probability that the outcome was in *category 0* was given by $1 - p(x)$.
  - The sum of both probabilities is equal to unity, as they must be.

- In general, if we have $M + 1$ explanatory variables (including x_0) and $N + 1$ categories, we will need $N + 1$ separate probability distributions, one for each category, indexed by n, which describe the probability that the categorical outcome y for explanatory vector x will be in category y_n.

Many Explanatory Variable, Many Categories

- It will also be required that the sum of these probabilities over all categories be equal to unity.
- Using the mathematically convenient base $\boldsymbol{e}$, these probabilities are:

$$p_n(\boldsymbol{x}) = \frac{e^{\beta_n \cdot \boldsymbol{x}}}{1 + \sum_{u=1}^{N} e^{\beta_u \cdot x}} \text{ for } n = 1,2, \ldots, \text{N}$$

$$p_0(\boldsymbol{x}) = 1 - \sum_{n=1}^{N} p_n(\boldsymbol{x}) = \frac{1}{1 + \sum_{u=1}^{N} e^{\beta_u \cdot x}}$$

- It can be seen that, as required, the sum of the $p_n(\boldsymbol{x})$ over all categories is unity.
- Note that the selection of $p_0(\boldsymbol{x})$ to be defined in terms of the other probabilities is artificial.
- Any of the probabilities could have been selected to be so defined.

Many Explanatory Variable, Many Categories

- This special value of n is termed the "pivot index," and the log-odds (t_n) are expressed in terms of the pivot probability and are again expressed as a linear combination of the explanatory variables:

$$t_n = \ln\left(\frac{p_n(\boldsymbol{x})}{p_0(\boldsymbol{x})}\right) = \boldsymbol{\beta}_n \cdot \boldsymbol{x}$$

Note :

For the simple case of $N = 1$, the two-category case is recovered, with

$$p(x) = p_1(x) \text{ and}$$

$$p_0(x) = 1 - p_1(x)$$

Many Explanatory Variable, Many Categories

- The log-likelihood that a particular set of K measurements or data points will be generated by the above probabilities can now be calculated.
- Indexing each measurement by k, let the $k-th$ set of measured explanatory variables be denoted by x_k and their categorical outcomes be denoted by y_k which can be equal to any integer in $[0, N]$.

Many Explanatory Variable, Many Categories

- The log-likelihood is then:

$$\ell = \sum_{k=1}^{K} \sum_{n=0}^{N} \Delta(n, y_k) ln\left(p_n(x_k)\right)$$

where $\Delta(n, y_k)$ is an indicator function which is equal to unity if $y_k = n$ and zero otherwise.

- In the case of two explanatory variables, this indicator function was defined as y_k when $n = 1$ and $1 - yk$ when $n = 0$.

Many Explanatory Variable, Many Categories

- Again, the optimum beta coefficients may be found by maximizing the log-likelihood function generally using numerical methods.
- A possible method of solution is to set the derivatives of the log-likelihood with respect to each beta coefficient equal to zero and solve for the beta coefficients:

$$\frac{\partial \ell}{\partial \beta_{nm}} = 0 = \sum_{k=1}^{K} \Delta(n, y_k) x_{mk} - \sum_{k=1}^{K} p_n(\boldsymbol{x}_k) x_{mk}$$

where β_{nm} is the m-th coefficient of the β_n vector and x_{mk} is the $m-th$ explanatory variable of the $k-th$ measurement.

Applications of Logistic Regression

- **In medical domains:**
  - The Trauma and Injury Severity Score (TRISS), which is widely used to predict mortality in injured patients, was originally developed by **Boyd *et al***. using logistic regression.
  - Many other medical scales used to assess severity of a patient have been developed using logistic regression.
  - Logistic regression may be used to predict the risk of developing a given disease (e.g. diabetes; coronary heart disease), based on observed characteristics of the patient (age, gender, body mass index, results of various blood tests, etc.)

Applications of Logistic Regression

- **In social sciences:**
  - Another example might be to predict whether an Indian voter will vote National Congress or Communist Party or BJP or Any other party, based on age, income, gender, race, state of residence, votes in previous elections, etc.
- **In engineering:**
  - The technique can also be used in engineering, especially for predicting the probability of failure of a given process, system or product.
- **In marketing:**
  - It is also used in marketing applications, such as prediction of a customer's propensity to purchase a product or halt a subscription, etc.

Logistic Regression using RStudio

Introduction

- We now study approaches for predicting qualitative responses, a process that is known as classification.
- Predicting a qualitative response for an observation can be referred to as classifying that observation, since it involves assigning the observation to a category, or class.
- On the other hand, often the methods used for classification first predict the probability of each of the categories of a qualitative variable, as the basis for making the classification.
- In this sense they also behave like regression methods.
- There are many possible classification techniques, or classifiers, that one might use to predict a qualitative response.
- We shall discuss two of the most widely-used classifiers: logistic regression and linear/quadratic discriminant analysis.
- There are more computing intensive methods, including generalized additive models, trees, random forests, boosting methods, and support vector machines, among others.

Classification examples

- A financial institution is devising a software to determine whether customers who have applied for a loan will be a 'good' customer or not, based on several factors including income, assets, and liabilities.
- A person arrives at the emergency room with a set of symptoms that could possibly be attributed to one of three medical conditions. Which of the three conditions does the individual have?
- An online banking service must be able to determine whether or not a transaction being performed on the site is fraudulent, on the basis of the user's IP address, past transaction history, and so forth.
- Can visual features from chest CT-scans reveal whether a person is infected by COVID-19 or is it just common flu?

Owl or Apple?

Generalized Linear Models

- The Generalized Linear Model (GLM) allows the incorporation of non-normal response variable distribution.
- The response variable distribution includes the exponential family of distributions only, which encompasses distributions like normal, Binomial, Poisson, exponential, gamma, among others.
- The normal-error linear model is a special case of GLM.
- We begin our presentation on such models that can perform classification task - ***logistic regression***.

Consider the *Default* data in the *ISLR* package.

- This dataset gives information on 10000 customers with 4 variables.
- The aim is to predict which customers will default on their credit card debt.
- The default variable indicates whether the customer defaulted or not.

Logistic Regression Models

```
library(ISLR2)
Default %>% head()
##   default student   balance    income
## 1      No      No  729.5265 44361.625
## 2      No     Yes  817.1804 12106.135
## 3      No      No 1073.5492 31767.139
## 4      No      No  529.2506 35704.494
## 5      No      No  785.6559 38463.496
## 6      No     Yes  919.5885  7491.559
```

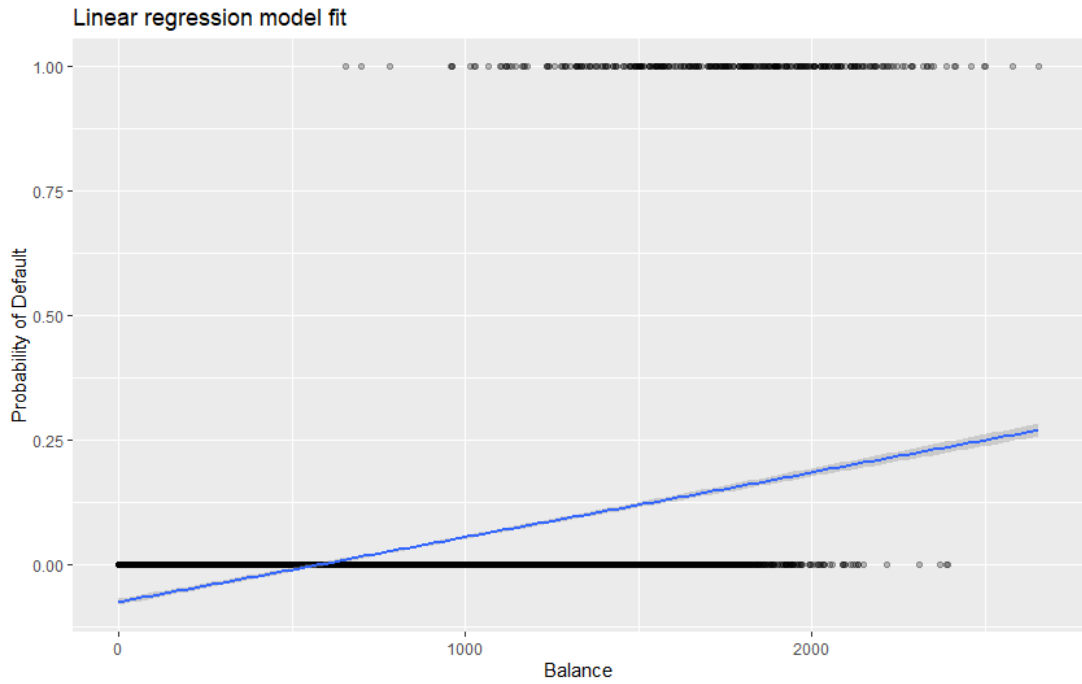
- The default variable is a factor variable with levels Yes and No indicating whether the customer had defaulted or not.
- We cannot model 0s and 1s directly. Rather, in logistic regression, we model the probability that the response variable Y belongs to a particular category.
- For the Default data, logistic regression models the probability of a customer defaulting. For example, the probability of a customer defaulting given balance is given by P(default="Yes"|balance). Let us abbreviate this probability by p(balance).
- Then, for any given value of balance, one can predict the default.

Logistic Regression Models

For example, one may predict a customer defaulting if p(balance)>0.5.
If someone is more conservative, they might choose a lower threshold, say, p(balance)>0.2.
How should we model the relationship between X and P(Y=1|X)?

Let us consider a simple linear regression model, that is, $p(X)=\beta_0+\beta_1 X$.

```
Default %>%
  mutate(prob = ifelse(default == "Yes", 1, 0)) %>%
  ggplot(aes(balance, prob)) +
  geom_point(alpha = .25) +
  geom_smooth(method = "lm") +
  ggtitle("Linear regression model fit") +
  xlab("Balance") +
  ylab("Probability of Default")
```



Logistic Regression Models

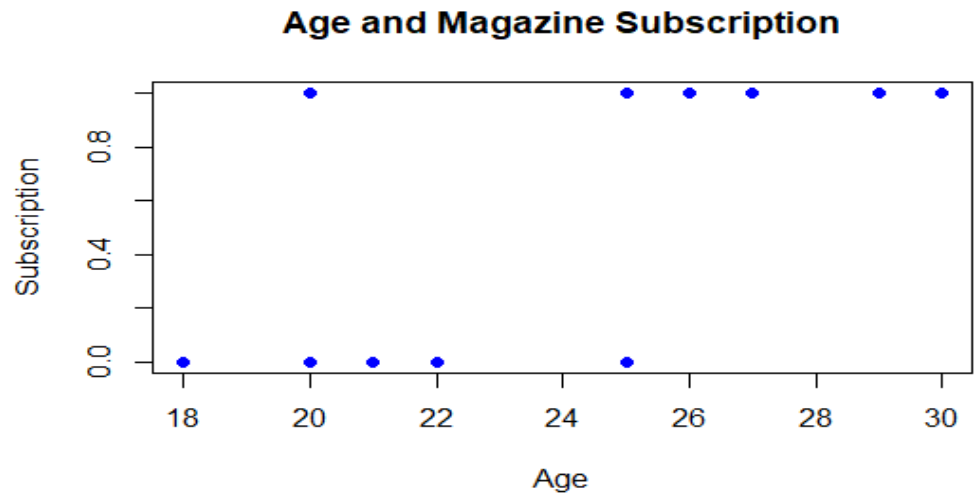
- The normal-error linear model for the probability makes little sense, as the probability function is restricted to lie between 0 and 1 only.
- To avoid this problem, we must model p(X) using a function that gives outputs between 0 and 1.
- The logistic regression model uses the function $p(X) = \frac{\exp(\beta_0+\beta_1 x)}{1+\exp(\beta_0+\beta_1 x)}$, which is known as the logistic function.
- The logistic regression function can be easily presented in the form of a linear regression model.
- Consider the transformation $\eta = ln\left(\frac{p(x)}{1-p(x)}\right)$.
- Then, we can write the logistic regression model as $\eta = \beta_0 + \beta_1 x$.
- This transformation is often called the **logit transformation** of the probability p(X), and the ratio $\frac{p(x)}{1-p(x)}$ in the transformation is called the **Odds**.
- The logit transformation is also known as the **log-odds** transformation.

Example: Logistic Regression

- The following data on some random customers from a given city. Determine the customer's decision to subscribe to a magazine.

Age (x)	Subscription (y)
18	0
18	0
20	0
20	1
21	0
21	0
22	0
22	0
22	0
25	1
25	0
25	1
26	1
26	1
27	1
29	1
30	1
30	1

Example:



- Fitted linear model:

$$\hat{y} = P(Subscribe = 1) = \hat{p} = -1.8 + 0.098x$$

- At age 18, $\hat{y} = -0.06$,
- At age 20, $\hat{y} = 0.13$
- At age 30, $\hat{y} = 1.11$

Fitting a simple Logistic Regression Models

- We now illustrate the fitting of a logistic regression model with one single predictor X corresponding to a binary response variable Y.
- The *glm* function fits generalized linear models
- The syntax of the glm function is similar to that of lm, except that we must pass the argument family = binomial in order to tell R to run a logistic regression rather than some other type of generalized linear model.
- Estimates of the coefficients in the model are obtained by a method called maximum likelihood method.

```
library(modelr)

set.seed(100)

# Split into training and testing data
default_split <- resample_partition(Default, c(test = 0.3, train = 0.7))

default_train <- as_tibble(default_split$train)
default_test  <- as_tibble(default_split$test)
```

Fitting a simple Logistic Regression Models

```
# Fitting a simple logistic regression model
model1 <- glm(default ~ balance, family = binomial, data = default_train)
summary(model1)
```

```
## Call:
## glm(formula = default ~ balance, family = binomial, data = default_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1606  -0.1420  -0.0558  -0.0203   3.7201
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.089e+01  4.432e-01  -24.57   <2e-16 ***
## balance      5.683e-03  2.714e-04   20.94   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2043.8  on 7000  degrees of freedom
## Residual deviance: 1093.5  on 6999  degrees of freedom
## AIC: 1097.5
##
## Number of Fisher Scoring iterations: 8
```

Fitting a simple Logistic Regression Models

```
# Quick check whether "default = Yes" is coded as "1" by R
contrasts(default_train$default)
##     Yes
## No    0
## Yes   1
```

Interpretation of the model coefficients

- Consider the fitted value of the linear predictor at a particular value of X, say x_0, given by $\hat{\eta}(x_0) = \hat{\beta}_0 + \hat{\beta}_1 x_0$.
- The fitted value at $X = x_0 + 1$ is $\hat{\eta}(x_0 + 1) = \hat{\beta}_0 + \hat{\beta}_1(x_0 + 1)$.
- The difference in the two predicted values is $\hat{\eta}(x_0 + 1) - \hat{\eta}(x_0) = \hat{\beta}_1$.
- Now, $\hat{\eta}(x_0)$ and $\hat{\eta}(x_0 + 1)$ are the log-odds corresponding to respective predictor values of $X = x_0$ and $X = x_0 + 1$. Therefore, the estimated coefficient $\hat{\beta}_1$ captures the difference in the respective log-odds, that is, $\hat{\beta}_1 = ln\left(\frac{odds(x_0+1)}{odds(x_0)}\right)$.
- Taking antilogs, we obtain the **Odds Ratio** as $\hat{O}_R = \frac{odds(x_0+1)}{odds(x_0)} = \exp(\hat{\beta}_1)$**.**

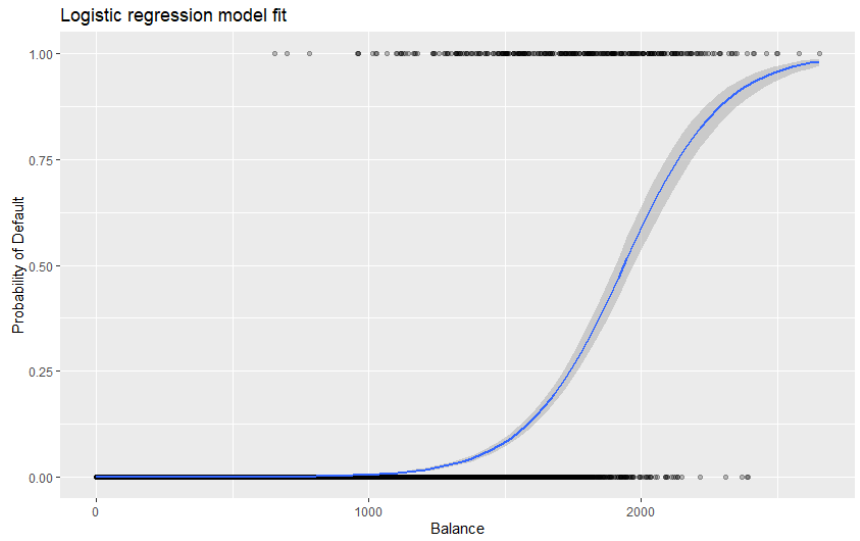
Fitting a simple Logistic Regression Models

```
exp(coef(model1))
##  (Intercept)      balance
## 1.868042e-05 1.005699e+00
```

We can interpret the balance coefficient as: for every one unit increase in balance, the odds of the customer defaulting goes up by 1.006.

Let us visualize the model fit.

```
Default %>%
  mutate(prob = ifelse(default == "Yes", 1, 0)) %>%
  ggplot(aes(balance, prob)) +
  geom_point(alpha = .25) +
  geom_smooth(method = "glm", method.args = list(family = "binomial")) +
  ggtitle("Logistic regression model fit") +
  xlab("Balance") +
  ylab("Probability of Default")
```



Logistic Regression Model Predictions

Once we have the estimates of the model coefficients, it is very easy to arrive at the predictions.
For example, the default probability for an individual with a balance of $1,000 can be calculated as $\hat{p}(1000) = \frac{\exp(\hat{\beta}_0+\hat{\beta}_1 1000)}{1+\exp(\hat{\beta}_0+\hat{\beta}_1 1000)}$. We can make predictions using the fitted model by using the predict function on the test data.

```
test.predictions.m1 <- predict(model1, newdata = default_test, type = "response")
test.predictions.m1 <- tibble(balance = default_test$balance,
                              true.default.status = default_test$default,
                              pred.default = test.predictions.m1)
test.predictions.m1
## # A tibble: 2,999 x 3
##    balance true.default.status pred.default
##      <dbl> <fct>                     <dbl>
##  1    817. No                   0.00194
##  2      0  No                   0.0000187
##  3   1221. No                    0.0189
##  4   1113. No                   0.0103
##  5    286. No                    0.0000950
##  6      0  No                   0.0000187
##  7   1095. No                    0.00933
##  8    643. No                    0.000721
##  9    914. No                   0.00335
## 10   1500. No                    0.0858
## # ... with 2,989 more rows
```

Fitting a simple Logistic Regression Models

Take a look at the predicted default probabilities who actually defaulted

```
test.predictions.m1 %>%
  filter(true.default.status == "Yes")
## # A tibble: 100 x 3
##    balance true.default.status pred.default
##      <dbl> <fct>                     <dbl>
##  1   1899. Yes                     0.476
##  2   1573. Yes                     0.125
##  3   1530. Yes                     0.100
##  4   1119. Yes                     0.0107
##  5   1764. Yes                     0.296
##  6   1532. Yes                     0.101
##  7    780. Yes                     0.00157
##  8   1889. Yes                     0.462
##  9   1244. Yes                     0.0214
## 10   1753. Yes                     0.284
## # ... with 90 more rows
```

Notice the type = "response" argument in the predict function. This gives the actual probabilities. The other type, type = "link" will give us the log-odds.

Fitting a Multiple Logistic Regression Models

We can generalize the simple logistic regression model to multiple logistic regression model by incorporating more predictors as $\ln\left(\frac{p(X_1,\ldots,X_p)}{1-p(X_1,\ldots,X_p)}\right)=\beta_0+\beta_1X_1+\cdots+\beta_pX_p$.

This gives $p(X_1,\ldots,X_p) = \frac{\exp(\beta_0+\beta_1X_1+\cdots+\beta_pX_p)}{1+\exp(\beta_0+\beta_1X_1+\cdots+\beta_pX_p)}$.

Let us now demonstrate the procedure with the Default dataset taking into account all the available predictors, namely, student, balance and income.

```
model2 <- glm(default ~ student + balance + income, family = binomial, data = default_train)
summary(model2)
## Call:
## glm(formula = default ~ student + balance + income, family = binomial,
##     data = default_train)
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0676  -0.1380  -0.0535  -0.0193   3.6937
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.081e+01  5.934e-01 -18.219   <2e-16 ***
## studentYes  -6.348e-01  2.857e-01  -2.221   0.0263 *
## balance      5.855e-03  2.816e-04  20.791   <2e-16 ***
## income      -3.099e-06  1.005e-05  -0.308   0.7577
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## (Dispersion parameter for binomial family taken to be 1)
##     Null deviance: 2043.8  on 7000  degrees of freedom
## Residual deviance: 1082.6  on 6997  degrees of freedom
## AIC: 1090.6
## Number of Fisher Scoring iterations: 8
```

Fitting a Multiple Logistic Regression Models

We can do predictions similarly.

```
test.predictions.m2 <- predict(model2, newdata = default_test, type = "response")
test.predictions.m2 <- tibble(default_test, pred.default = test.predictions.m2)
test.predictions.m2
```

```
## # A tibble: 2,999 x 5
##    default student balance income pred.default
##    <fct>   <fct>     <dbl>  <dbl>       <dbl>
##  1 No      Yes       817. 12106.    0.00123
##  2 No      No          0  29275.    0.0000184
##  3 No      Yes      1221. 13269.    0.0128
##  4 No      No       1113. 23810.    0.0125
##  5 No      No        286. 45042.    0.0000937
##  6 No      No          0  50265.    0.0000172
##  7 No      No       1095. 26465.    0.0112
##  8 No      No        643. 41474.    0.000764
##  9 No      No        914. 46907.    0.00365
## 10 No      Yes      1500. 13191.    0.0626
## # ... with 2,989 more rows
```

Take a look at the predicted default probabilities who actually defaulted

```
test.predictions.m2 %>% filter(default == "Yes")
```

```
## # A tibble: 100 x 5
##    default student balance income pred.default
##    <fct>   <fct>     <dbl>  <dbl>       <dbl>
##  1 Yes     Yes      1899. 20655.      0.404
##  2 Yes     Yes      1573. 14930.      0.0924
##  3 Yes     No       1530. 30004.      0.125
##  4 Yes     Yes      1119. 21848.      0.00693
##  5 Yes     No       1764. 46227.      0.348
##  6 Yes     No       1532. 43930.      0.121
##  7 Yes     No        780. 51657.      0.00165
##  8 Yes     Yes      1889. 22652.      0.388
##  9 Yes     No       1244. 37634.      0.0254
## 10 Yes     No       1753. 48965.      0.332
## # ... with 90 more rows
```

Model validation via Classification performance assessment

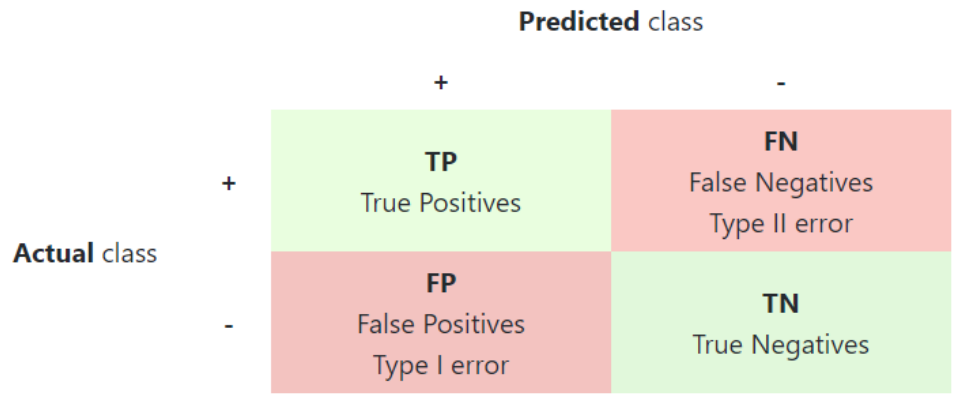
Now we can compare the predicted target variable versus the observed values for the classification models.

We can start by using the confusion matrix, which is a table that describes the classification performance for each model on the test data.

The confusion matrix is used to have a more complete picture when assessing the performance of a model.

We define the following:

- **True Positives**: these are cases in which we predicted the customer would default and they did.
- **True Negatives**: We predicted no default, and the customer did not default.
- **False Positives**: We predicted yes, but they didn't actually default. (Also known as a "Type I error.")
- **False Negatives**: We predicted no, but they did default. (Also known as a "Type II error.")



Model validation via Classification performance assessment

First model: only predictor is balance

```
glm.pred.m1 <- rep("No", dim(default_test)[1])
glm.pred.m1[test.predictions.m1$pred.default > 0.5] <- "Yes"
table(glm.pred.m1, default_test$default)
## glm.pred.m1   No  Yes
##         No  2879   70
##         Yes   20   30
```

Second model: all predictors included

```
glm.pred.m2 <- rep("No", dim(default_test)[1])
glm.pred.m2[test.predictions.m2$pred.default > 0.5] <- "Yes"
table(glm.pred.m2, default_test$default)
## glm.pred.m2   No  Yes
##         No  2884   67
##         Yes   15   33
```

Classification Metrics

Metric	Formula	Interpretation
Accuracy	$\frac{TP + TN}{TP + TN + FP + FN}$	Overall performance of model
Precision	$\frac{TP}{TP + FP}$	How accurate the positive predictions are
Recall Sensitivity	$\frac{TP}{TP + FN}$	Coverage of actual positive sample
Specificity	$\frac{TN}{TN + FP}$	Coverage of actual negative sample
F1 score	$\frac{2TP}{2TP + FP + FN}$	Hybrid metric useful for unbalanced classes

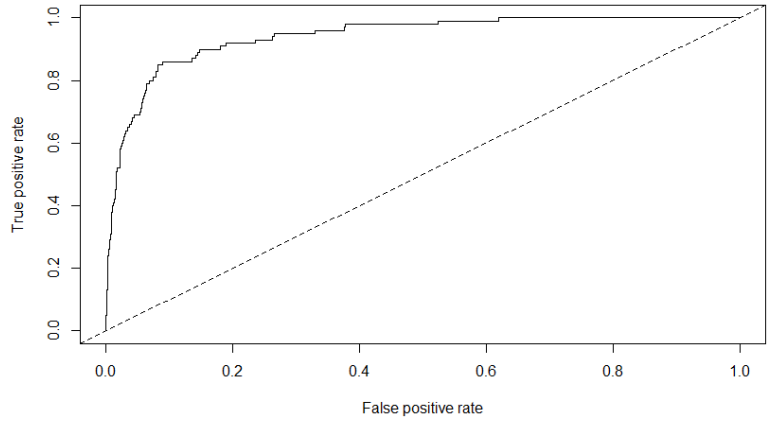
ROC Curve and AUC

The ROC curve is a popular graphic for simultaneously displaying the two types of errors for all possible thresholds.

The name “ROC” is historic, and comes from communications theory. It is an acronym for receiver operating characteristics.

The following plot displays the ROC curve for the logistic regression classifier on the training data.

```
library(ROCR)
test.predictions.m1 <- predict(model1, newdata = default_test, type = "response")
prediction(test.predictions.m1, default_test$default) %>%
performance(measure = "tpr", x.measure = "fpr") %>%
plot() %>%
abline(a=0,b=1, lty = "dashed")
```



ROC Curve and AUC

The overall performance of a classifier, summarized over all possible thresholds, is given by the Area Under the ROC Curve (AUC).

An ideal ROC curve will hug the top left corner, so the larger the AUC the better the classifier.

```
# AUC
prediction(test.predictions.m1, default_test$default) %>%
  performance(measure = "auc") %>%
  .@y.values
## [[1]]
## [1] 0.9409003
```

Note that if we want to reduce the false positive rate by restricting the predicted class to be negative, the false negative rate will also go up. This is saying that in order to reduce Type-I error, the Type-II error goes up.

Example:

- The hypothetical data below are from a toxicity study designed to measure the effectiveness of different doses of a pesticide on mosquitoes. The table below summarizes the concentration of the pesticide, the sample sizes, and the number of critters dispatched.

Concentration (g/cc):	0.10	0.15	0.20	0.30	0.50	0.70	0.95
No. of mosquitoes :	48	52	56	51	47	53	51
No. killed :	10	13	25	31	39	51	49

  - Make a scatterplot of the proportions of mosquitoes killed versus the pesticide concentration.
  - Using the techniques introduced in this section, calculate $y' = \ln\left(\frac{p}{1-p}\right)$ for each of the concentrations and fit the line $y' = a + b\ (Concentration)$. What is the significance of a positive slope for this line?
  - The point at which the dose kills 50% of the pests is sometimes called LD50, for "Lethal dose 50%." What would you estimate to be LD50 for this pesticide when used on mosquitoes?

SUMMARIZE: BINARY LOGISTIC REGRESSION

- It is a multiple regression with an outcome variable (or dependent variable) to be a categorical dichotomic and explanatory variables that can be either continuous or categorical.

- In other words, the interest is in predicting which of two possible events are going to happen given certain other information.

- For example in Drug efficacy testing, logistic regression could be used to analyze the factors that determine whether the drug cures a particular disease or not.

- The Logistic Curve will relate the explanatory variable X to the probability of the event occurring.

SUMMARIZE: BINARY LOGISTIC REGRESSION

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

The output variable will be binary, coded as either success or failure

Models probability of success p which lies between 0 and 1

Linear model is not appropriate

$$p = \frac{e^{a+b_1x_1+b_2x_2+---+b_kx_k}}{1+e^{a+b_1x_1+b_2x_2+---+b_kx_k}}$$

p: probability of success
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

BINARY LOGISTIC REGRESSION

Usage: When the dependent variable (Y variable) is binary

Example: Develop a model to predict the number of visits of family to a vacation resort based on the salient characteristics of the families. The data collected from 30 households is given in Resort_Visit.csv

1. Reading the file and variables

```
> mydata =  read.csv('Resort_Visit.csv',header = T,sep = ",")
> visit = mydata$Resort_Visit
> income = mydata$Family_Income
> attitude = mydata$Attitude.Towards.Travel
> importance = mydata$Importance_Vacation
> size = mydata$House_Size
> age = mydata$Age._Head
```

2. Converting response variable to discrete

```
> visit = factor(visit)
```

BINARY LOGISTIC REGRESSION

3. Correlation Matrix

> cor(mydata)

	Resort_Visit	Family_Income	Attitude_Travel	Importance_Vacation	House_Size	Age_Head
Resort_Visit	1.00	-0.60	-0.27	-0.42	-0.59	-0.21
Family_Income	-0.60	1.00	0.30	0.23	0.47	0.21
Attitude_Travel	-0.27	0.30	1.00	0.19	0.15	-0.13
Importance_Vacation	-0.42	0.23	0.19	1.00	0.30	0.11
House_Size	-0.59	0.47	0.15	0.30	1.00	0.09
Age_Head	-0.21	0.21	-0.13	0.11	0.09	1.00

Interpretation: Correlation between X variables should be low

BINARY LOGISTIC REGRESSION

4. Checking relation between Xs and Y

> aggregate(income ~visit, FUN = mean)

> aggregate(attitude ~visit, FUN = mean)

> aggregate(importance ~visit, FUN = mean)

> aggregate(size ~visit, FUN = mean)

> aggregate(age ~visit, FUN = mean)

Resort_Visit	Mean				
	Family_Income	Attitude_Travel	Importance_Vacation	House_Size	Age_Head
0	58.5200	5.4000	5.8000	4.3333	53.7333
1	41.9133	4.3333	4.0667	2.8000	50.1333

Higher the difference in means, stronger will be the relation to response variable.

BINARY LOGISTIC REGRESSION

5. Checking relation between Xs and Y – box plot

> boxplot(income ~ visit)

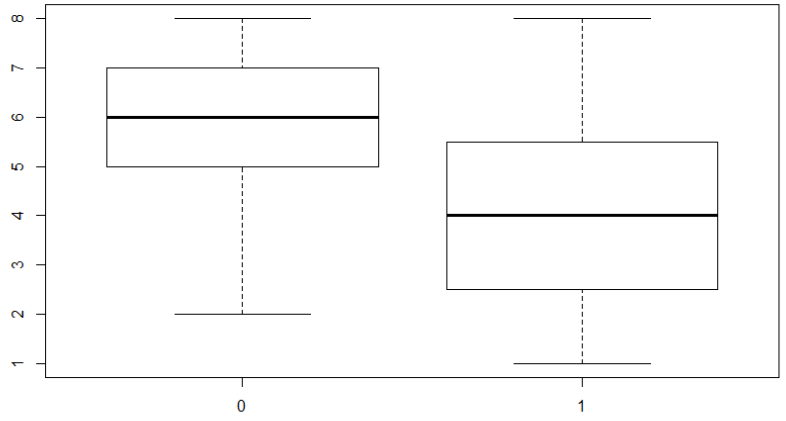
> boxplot(attitude ~ visit)

> boxplot(importance ~ visit)

> boxplot(size ~ visit)

> boxplot(age ~ visit)

Income Vs visit



BINARY LOGISTIC REGRESSION

6. Perform Logistic regression

> model = glm(visit ~ income + attitude + importance + size + age, family = binomial(logit))

> summary(model)

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	15.49503	6.68017	2.32	0.0204
Income	-0.11698	0.06605	-1.771	0.0766
attitude	-0.28129	0.33919	-0.829	0.4069
importance	-0.46157	0.32006	-1.442	0.1493
size	-0.80699	0.49314	-1.636	0.1018
age	-0.07019	0.07199	-0.975	0.3295

BINARY LOGISTIC REGRESSION

6. Perform Logistic regression - Anova

> anova(model,test = 'Chisq')

	Df	Deviance	Resid.Df	Resid.Dev	Pr(>Chi)
NULL	29	41.589			
income	1	12.9813	28	28.608	0.00031
attitude	1	0.4219	27	28.186	0.51598
importance	1	3.8344	26	24.351	0.05021
size	1	3.4398	25	20.911	0.06364
age	1	1.0242	24	19.887	0.31152

Since p value < 0.05 for Income redo the modelling with important factor (income) only.

BINARY LOGISTIC REGRESSION

7. Perform Logistic regression - Modified

	Estimate	Std Error	z value	p value
(Intercept)	6.36727	2.32544	2.738	0.00618
Income	-0.12778	0.04634	-2.758	0.00582

Since p value < 0.05 for Income, the response variable can be modelled in terms of those two factors

The logistic model is:

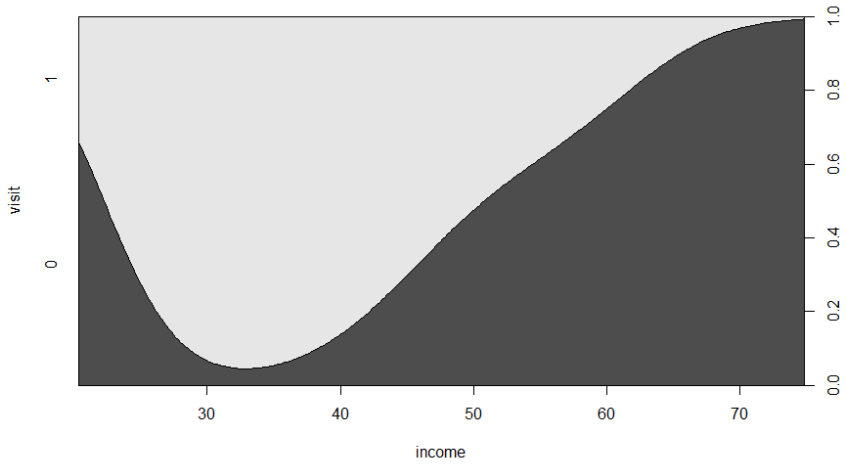
$$y = \frac{e^{6.36727 - 0.12778 * Annual\_Income}}{1 + e^{6.36727 - 0.12778 * Annual\_Income}}$$

BINARY LOGISTIC REGRESSION

8. Conditional Density plots (Response Vs Factors)

Describing how the conditional distribution of a categorical variable y changes over a numerical variable x

> cdplot(visit ~ income)



BINARY LOGISTIC REGRESSION

9. Fitted Values and residuals

```
> predict(model,type = 'response')
> residuals(model,type = 'deviance')
> predclass = ifelse(predict(model, type ='response')>0.5,"1","0")
```

SL No.	Actual	Fitted	Residuals	Predicted Class	SL No.	Actual	Fitted	Residuals	Predicted Class
1	0	0.970979	-2.66073	1	16	1	0.904132	0.448954	1
2	0	0.059732	-0.35097	0	17	1	0.939523	0.353222	1
3	0	0.021049	-0.20627	0	18	1	0.880611	0.50426	1
4	0	0.202309	-0.67236	0	19	1	0.345537	1.457845	0
5	0	0.292461	-0.83182	0	20	1	0.724535	0.802777	1
6	0	0.014893	-0.17324	0	21	1	0.925508	0.393479	1
7	0	0.677783	-1.50501	1	22	1	0.677559	0.882337	1
8	0	0.038723	-0.28105	0	23	1	0.680103	0.878079	1
9	0	0.109432	-0.48145	0	24	1	0.516151	1.150092	1
10	0	0.030543	-0.24908	0	25	1	0.680326	0.877704	1
11	0	0.017609	-0.1885	0	26	1	0.77062	0.721887	1
12	0	0.050856	-0.32309	0	27	1	0.629425	0.962235	1
13	0	0.04202	-0.29301	0	28	1	0.954395	0.305541	1
14	0	0.601981	-1.35739	1	29	1	0.841493	0.587498	1
15	0	0.499424	-1.17643	0	30	1	0.900286	0.45835	1

BINARY LOGISTIC REGRESSION

10. Model Evaluation

> mytable = table(visit, predclass)

> mytable

> prop.table(mytable)

	Predicted Count		Total
Actual Count	0	1	
0	11	4	15
1	3	12	15
Total	14	16	30

Statistics	Value
Accuracy %	76.666
Error %	23.333

Accuracy of ≥ 75 % is considerably good.

BINARY LOGISTIC REGRESSION

10. Model Evaluation

> mytable = table(visit, predclass)

> mytable

> prop.table(mytable)

	Predicted Count		Total
Actual Count	0	1	
0	11	4	15
1	3	12	15
Total	14	16	30

Statistics	Value
Accuracy %	76.666
Error %	23.333

Accuracy of ≥ 75 % is considerably good.

ORDINAL LOGISTIC REGRESSION

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

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

```
> mydata = read.csv('ST_Defects.csv', header = T, sep = ",")
> dd = mydata$DD
> effort = mydata$Effort
```

- coverage = mydata$Test.Coverage
- dd = factor(dd)

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:/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)
```

		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%

Exercise

Assess the classification performance of the models with a more conservative threshold in the Default dataset.

Find the associated classification metrics from the models.

Load the Smarket data from the ISLR2 package. Use the variable Direction as the response variable. Use a logistic regression model to predict the market performance, but do not use the variable Today as a predictor (why?)

Problems to Ponder

Problem–1

Statement: In a study of urban planning in a country, a survey was taken of 50 cities; 24 used Happiness Index (HI) and 26 did not. One part of the study was to investigate the relationship between the presence or absence of HI and the median family income of the city(x). The data are given in Table 1, with median income in order of $1000s.

Table 1: Data from Happiness Study

HI	x	HI	x	HI	x	HI	x
0	9.2	0	10.5	1	9.6	1	12.5
0	9.2	0	10.5	1	10.1	1	12.6
0	9.3	0	10.9	1	10.3	1	12.6
0	9.4	0	11.0	1	10.9	1	12.6
0	9.5	0	11.2	1	10.9	1	12.9
0	9.5	0	11.2	1	11.1	1	12.9
0	9.5	0	11.5	1	11.1	1	12.9
0	9.6	0	11.7	1	11.1	1	12.9
0	9.7	0	11.8	1	11.5	1	13.1
0	9.7	0	12.1	1	11.8	1	13.2
0	9.8	0	12.3	1	11.9	1	13.5
0	9.8	0	12.5	1	12.1		
0	9.9	0	12.9	1	12.2		

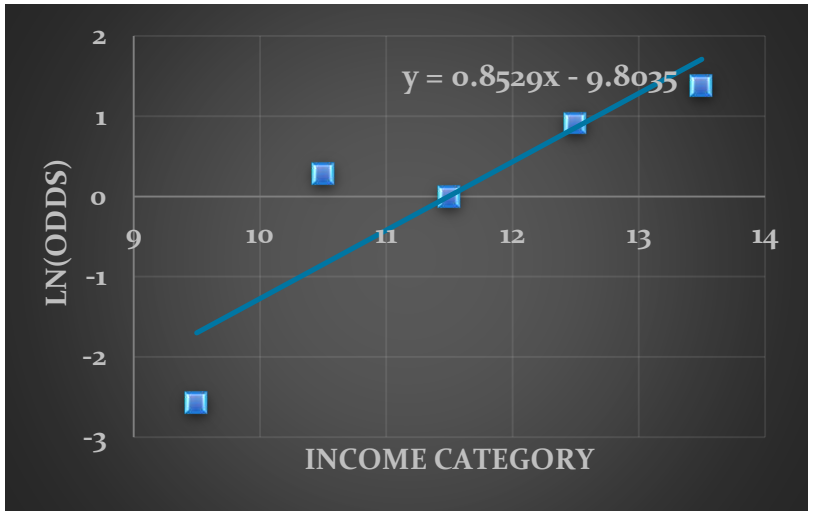
Problem−1: Solution

To understand whether a logistic model is appropriate, let us first categorized all the incomes in classes of unit length ($9 \leq x < 10, 10 \leq x < 11, etc.$). We then calculate odds within each class. However, one of the classes has zero occurrences of 0, creating an undefined odds ratio ($3 \div 0$). Since the ln(odds) is undefined, we followed a common practice of adding 1 to both the numerator and denominator counts in the calculation of all the ln(odds).

Statement: In a study of urban planning in Florida, a survey was taken of 50 cities; 24 used tax increment funding (TF) and 26 did not. One part of the study was to investigate the relationship between the presence or absence of TF and the median family income of the city(x). The data are given in the Table, with median income in order $1000s.

Income category	Income Category	Number of 0	Number of 1	Odds	ln(Odds))
$9 \leq X < 10$	9.5	13	1	$1/13 = 0.077$	-2.56395
$10 \leq X < 11$	10.5	3	4	$4/3 = 1.333$	0.287432
$11 \leq X < 12$	11.5	6	6	$6/6 = 1$	0
$12 \leq X < 13$	12.5	4	10	$10/4 = 2.5$	0.916291
$13 \leq X < 14$	13.5	0	3	$(3/0 =)4/1 = 4$	1.386294

Problem−1 : Solution



Income category	Mid-point of Income Category	NUMBER OF 0	NUMBER OF 1	ODDS	LN(ODDS)
$9 \le X < 10$	9.5	13	1	$1/13 = 0.077$	-2.56395
$10 \le X < 11$	10.5	3	4	$4/3 = 1.333$	0.287432
$11 \le X < 12$	11.5	6	6	$\frac{6}{6} = 1$	0
$12 \le X < 13$	12.5	4	10	$10/4 = 2.5$	0.916291
$13 \le X < 14$	13.5	0	3	$(3/0 =)4/1 = 4$	1.386294

Here, we find a relation between ln(Odds) of HI (i.e., t) and the midpoint of income category (x).

The equation we obtain is, $t = -9.8035 + 0.8529x$

Problem−1: Solution

- So, we find that there is a strong evidence of a relationship between the use of HI and median income. More wealthy cities have a higher probability of happiness index HI.

- Let's predict the probability of HI given an average family income

$$p = p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}}$$

- If one city has a median income of $12500, the probability of Happiness Index HI is

$$x = 12.5$$
$$t = -9.8035 + 0.8529 \times 12.5 = 0.858$$

$$\text{The probability} = \frac{e^t}{1+e^t} = \frac{e^{0.858}}{1+e^{0.858}} = 0.704$$

- So, there is a 70.4% probability that a city with $12500 median income will have happiness index HI is high.

Problem–2

Statement: Time Magazine (2006) used data from the USA to compare whites and blacks opinions of the death penalty. The data consisted of responses from 32,937 participants collected between 1972 and 1996. The outcome variable was whether the respondent did or did not support the death penalty. The survey provided a table of the percentage of whites and blacks each year that supported the death penalty.

Year	White (%)	Black (%)
1972	57.4	28.8
1973	63.6	35.8
1974	66.3	36.3
1975	63.2	31.9
1976	67.5	41.1
1977	70	41.6
1978	69.4	43
1980	70.3	39.1
1982	76.9	48.4
1983	76.2	45
1984	74.5	43.5
1985	79	49.7
1986	75.3	42.7

Year	White (%)	Black (%)
1987	73.7	42.9
1988	76	42.5
1989	76.5	56.1
1990	77.7	52.3
1991	71.4	42.7
1993	75.4	51.5
1994	78.3	50.7
1996	75.5	50.3

Problem–2

Convert the percentages given in the table to the ln(odds) within each race and year, and plot ln(odds) versus year. Comment on any patterns you see. If there is a trend in time, does it appear linear or quadratic?

Year	White (%)	Black (%)
1972	57.4	28.8
1973	63.6	35.8
1974	66.3	36.3
1975	63.2	31.9
1976	67.5	41.1
1977	70	41.6
1978	69.4	43
1980	70.3	39.1
1982	76.9	48.4
1983	76.2	45
1984	74.5	43.5
1985	79	49.7
1986	75.3	42.7

Year	White (%)	Black (%)
1987	73.7	42.9
1988	76	42.5
1989	76.5	56.1
1990	77.7	52.3
1991	71.4	42.7
1993	75.4	51.5
1994	78.3	50.7
1996	75.5	50.3

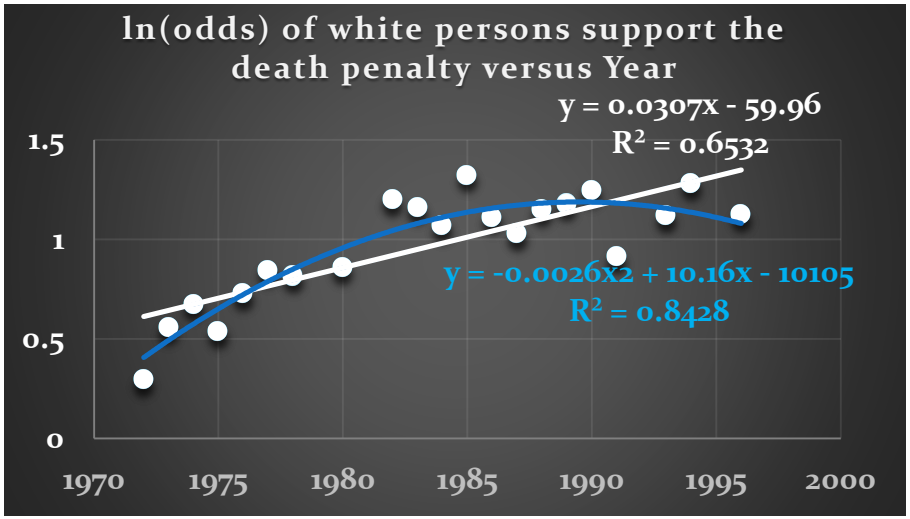
Problem–2 : Solution

Year	White (%)	Odds	ln(odds)
1972	57.4	1.347418	0.29819005
1973	63.6	1.747253	0.558044696
1974	66.3	1.967359	0.67669206
1975	63.2	1.717391	0.540806456
1976	67.5	2.076923	0.730887509
1977	70	2.333333	0.84729786
1978	69.4	2.267974	0.818886859
1980	70.3	2.367003	0.861624753
1982	76.9	3.329004	1.202673259
1983	76.2	3.201681	1.163675882
1984	74.5	2.921569	1.072120673
1985	79	3.761905	1.324925415
1986	75.3	3.048583	1.114676891

Year	White (%)	Odds	ln(odds)
1987	73.7	2.802281	1.03043386
1988	76	3.166667	1.15267951
1989	76.5	3.255319	1.18029032
1990	77.7	3.484305	1.248268579
1991	71.4	2.496503	0.914891152
1993	75.4	3.065041	1.120060832
1994	78.3	3.608295	1.283235342
1996	75.5	3.081633	1.125459539

$$Odds\ (white\ supporting\ death\ penalty) = \frac{\%\ of\ white\ persons\ supporting}{\%\ of\ white\ persons\ opposing} \left(= \frac{57.4}{100 - 57.4}\right)$$

Problem−2 : Solution



Given figure is a scatter plot of ln(odds) of white persons support the death penalty versus year. Two regression lines, one linear and one quadratic is fitted through the data. Visually we can observe that the quadratic line fits the data more. It is also evident from the R^2 value:

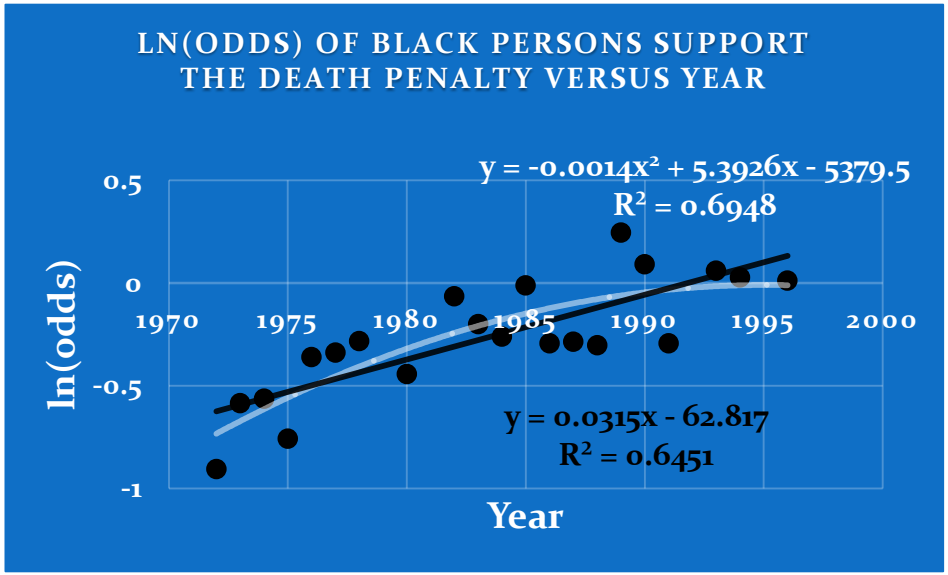
$$\boldsymbol{R^2(\text{Quadratic}) = 0.8428 > R^2(linear) = 0.6532}$$

Problem–2: Solution

Year	Black (%)	Odds	ln(odds)
1972	28.8	0.404494	-0.90512
1973	35.8	0.557632	-0.58406
1974	36.3	0.569859	-0.56237
1975	31.9	0.468429	-0.75837
1976	41.1	0.697793	-0.35983
1977	41.6	0.712329	-0.33922
1978	43	0.754386	-0.28185
1980	39.1	0.642036	-0.44311
1982	48.4	0.937984	-0.06402
1983	45	0.818182	-0.20067
1984	43.5	0.769912	-0.26148
1985	49.7	0.988072	-0.012
1986	42.7	0.745201	-0.2941

Year	Black (%)	Odds	ln(odds)
1987	42.9	0.751313	-0.28593
1988	42.5	0.73913	-0.30228
1989	56.1	1.277904	0.245221
1990	52.3	1.096436	0.092065
1991	42.7	0.745201	-0.2941
1993	51.5	1.061856	0.060018
1994	50.7	1.028398	0.028002
1996	50.3	1.012072	0.012

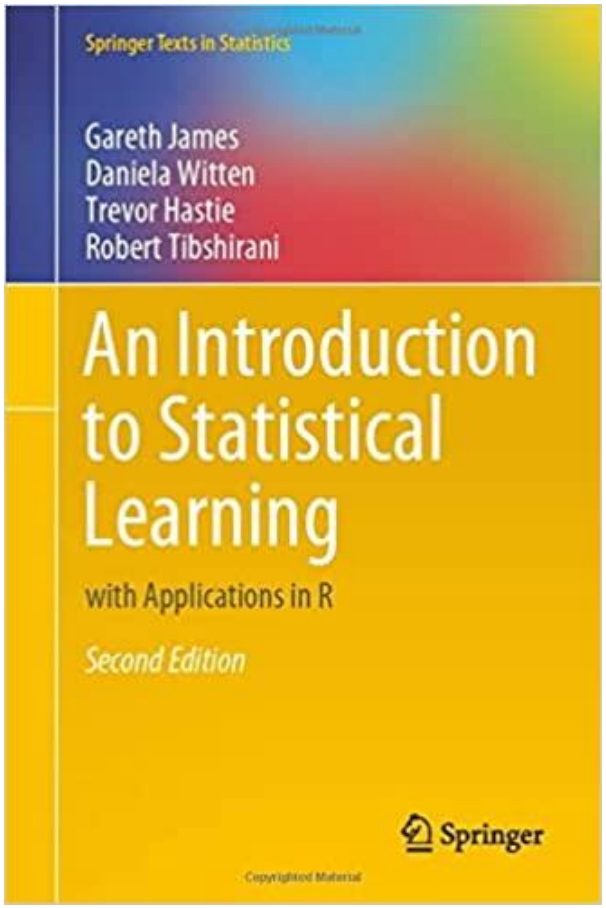
Problem−2: Solution



Given figure is a scatter plot of ln(odds) of black persons support the death penalty versus year. Two regression lines, one linear and one quadratic is fitted through the data. Visually we can observe that the quadratic line fits the data more. It is also evident from the R^2 value:

$$R^2(\text{Quadratic}) = 0.6948 > R^2(linear) = 0.6451$$

References



The author would like to acknowledge several resources:

1. https://www.statlearning.com/
2. https://hastie.su.domains/ElemStatLearn/
3. https://nptel.ac.in/courses/111105042
4. https://cse.iitkgp.ac.in/~dsamanta/courses/da/
5. https://sebastianraschka.com/blog/2021/ml-course.html