Nonlinear Regression using RStudio

Course Taught at SUAD

Dr. Tanujit Chakraborty

@ Sorbonne

tanujitisi@gmail.com

This presentation includes…

- Regression Analysis
  - Non-linear regression
  - Regression Splines

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:

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

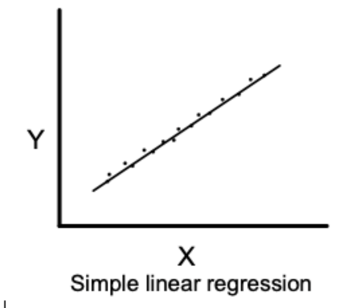
$$\hat{y}_i = b_0 + b_1 x_1 + b_2 x_2 + \ldots + b_r x_r + e_i$$

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

Linear versus Non-Linear Regression

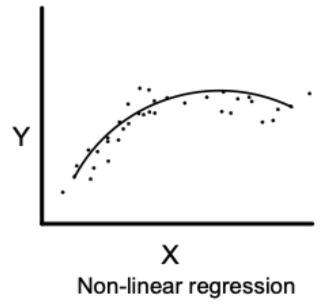
Simple linear regression model:

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



Simple non-linear regression model:

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



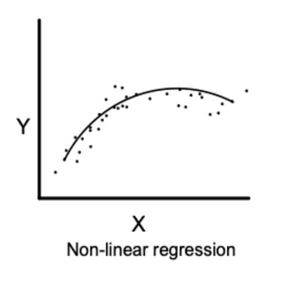
Linear versus Non-Linear Regression

Simple linear regression model:

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

Simple non-linear regression model:

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



Non-linear regression

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

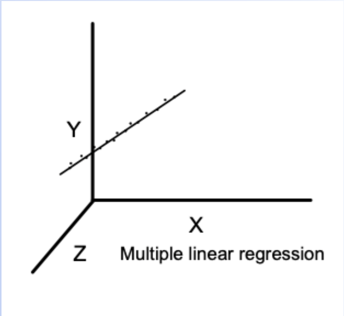
X	Y
x_i	y_i

Multiple Non-Linear Regression

Issues with Multiple Non-Linear Regression

Multiple non-linear regression model:

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



X	Y	Z

Issues:

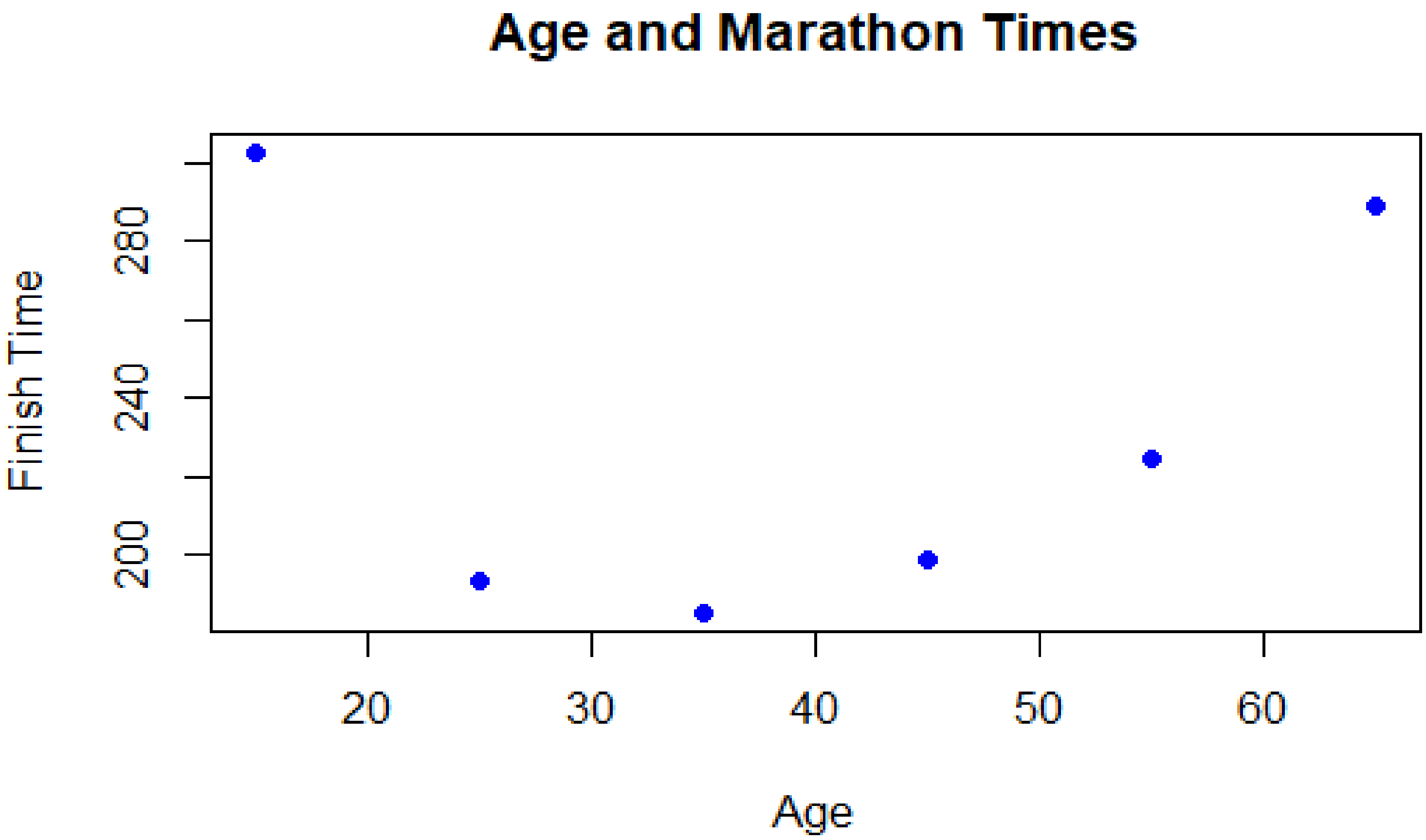
- Too complex to solve. Many parameters, many variations!
- Usually, used advanced machine learning models, such as SVM, kNN, ANN, etc.

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

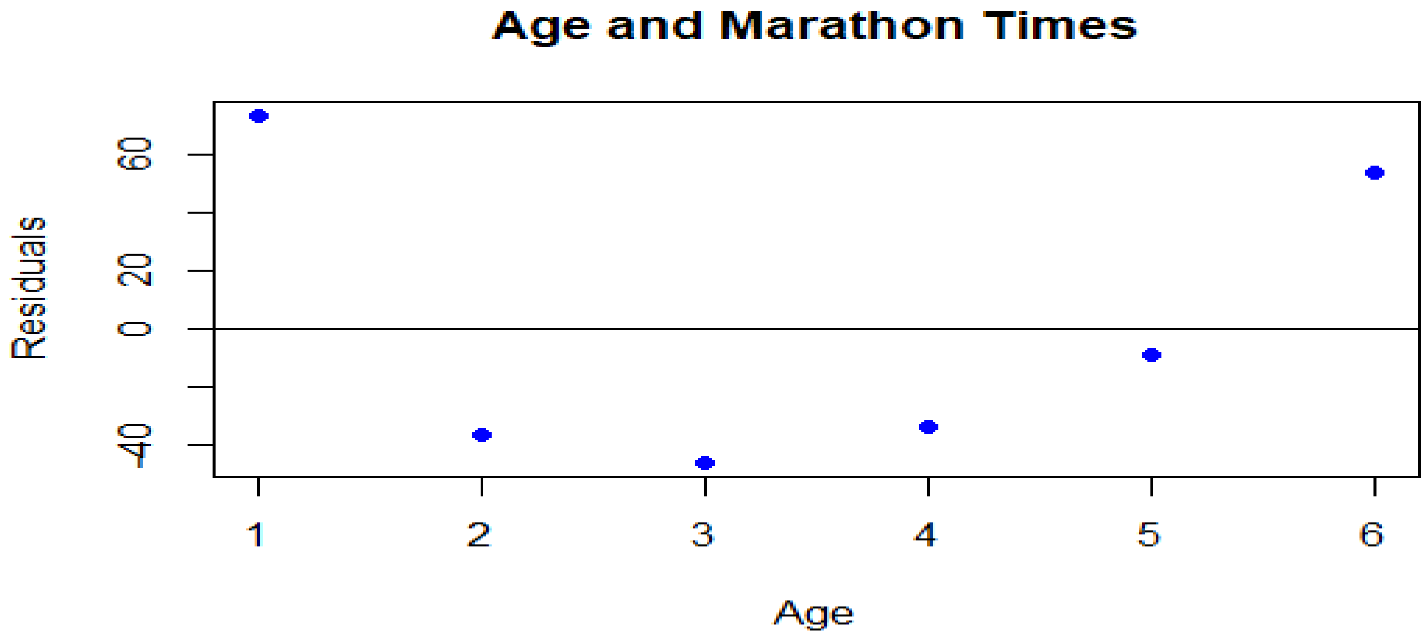


Correlation coefficient in Linear Case & Residual Plot

$r = 0.03847689$,

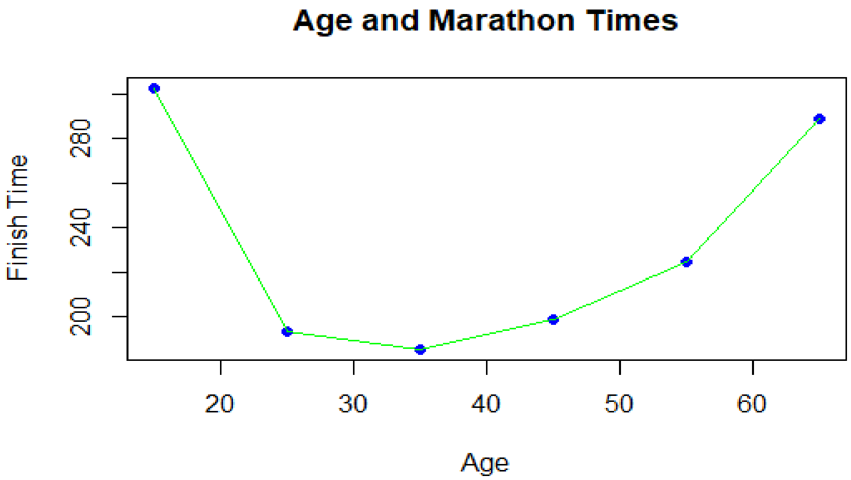
Linear Regression line: $\hat{y} = 227.97 + 0.10x$

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: $\hat{Y} = a + b_1X + b_2X^2$
- $y_i - (a + b_1x_i + b_2x_i^2)$ is called the error of estimate or residual for y_i.
- Principle of least square: determine a, b_1 and b_2 so that

$E = \sum\left(y_i - a - b_1x_i - b_2x_i^2\right)^2$, is minimum.

- Thus,

$$\frac{\partial E}{\partial a} = 0 = -2\sum(y_i - a - b_1x_i - b_2x_i^2)$$

$$\frac{\partial E}{\partial b_1} = 0 = -2\sum(y_i - a - b_1x_i - b_2x_i^2)x_i$$

$$\frac{\partial E}{\partial b_2} = 0 = -2\sum(y_i - a - b_1x_i - b_2x_i^2)x_i^2$$

- Hence,

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

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

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

Example: Marathon Data

- By solving the above system of equation, we get

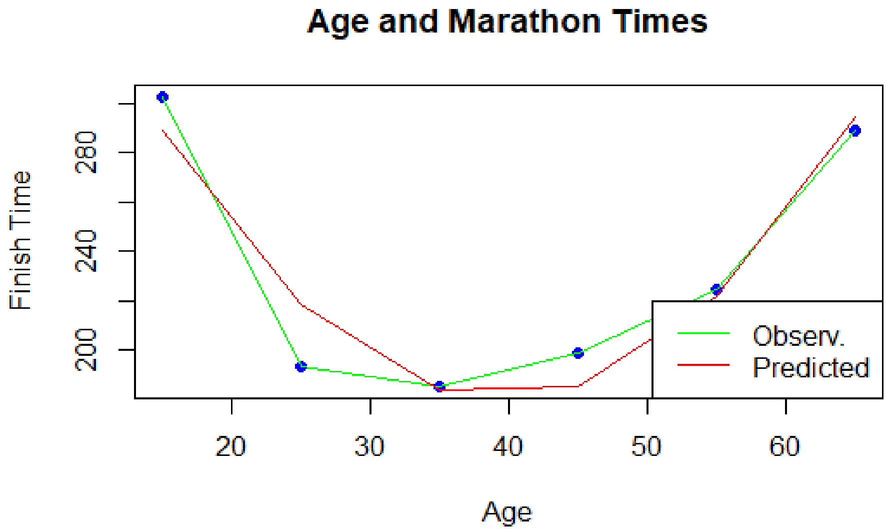
$$a = 462.0004453,$$
$$b_1 = -14.2054036$$
$$b_2 = 0.1788779$$

- 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'	
$Y = \beta_0 x^{\beta_1}$	$Y' = \log y$	$x' = \log x$	$Y' = \log \beta_0 + \beta_1 x'$
$Y = \beta_0 e^{\beta_1 x}$	$Y' = \ln y$	$x' = x$	$Y' = \ln \beta_0 + \beta_1 x'$
$Y = \beta_0 + \beta_1 \log x$	$Y' = y$	$x' = \log x$	$Y' = \beta_0 + \beta_1 x'$
$Y = \dfrac{x}{\beta_0 x - \beta_1}$	$Y' = {}^{1}/_{y}$	$x' = {}^{1}/_{x}$	$Y' = \beta_0 - \beta_1 x'$

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 = xβ + ε 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, √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)
```

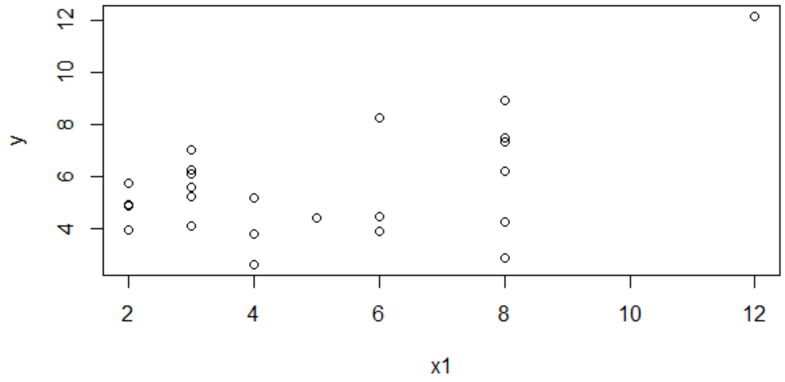
	x1	x2	x3	y
x1	1.00	0.40	-0.20	0.54
x2	0.40	1.00	-0.30	-0.36
x3	-0.20	-0.30	1.00	0.35
y	0.54	-0.36	0.35	1.00

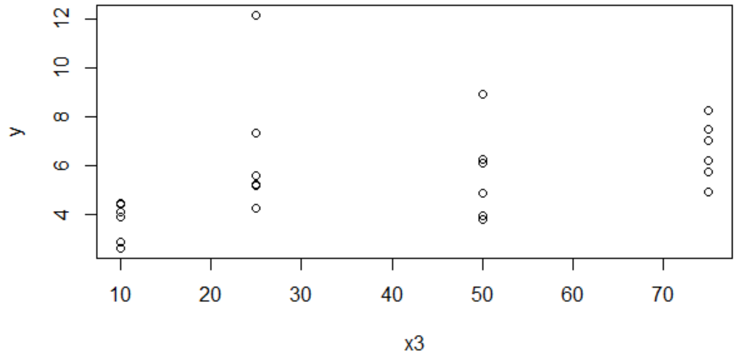
There is no strong correlation between y and x’s

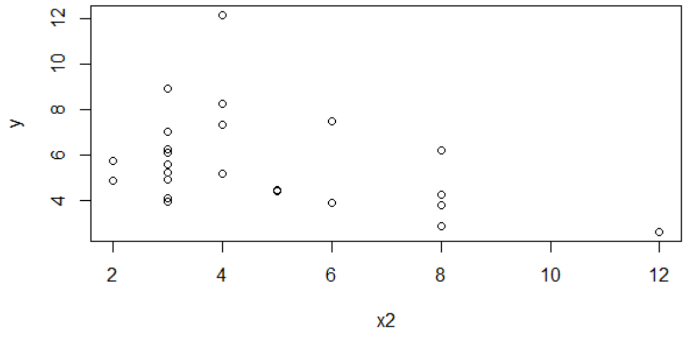
MODELING NONLINEAR RELATIONS

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)
```

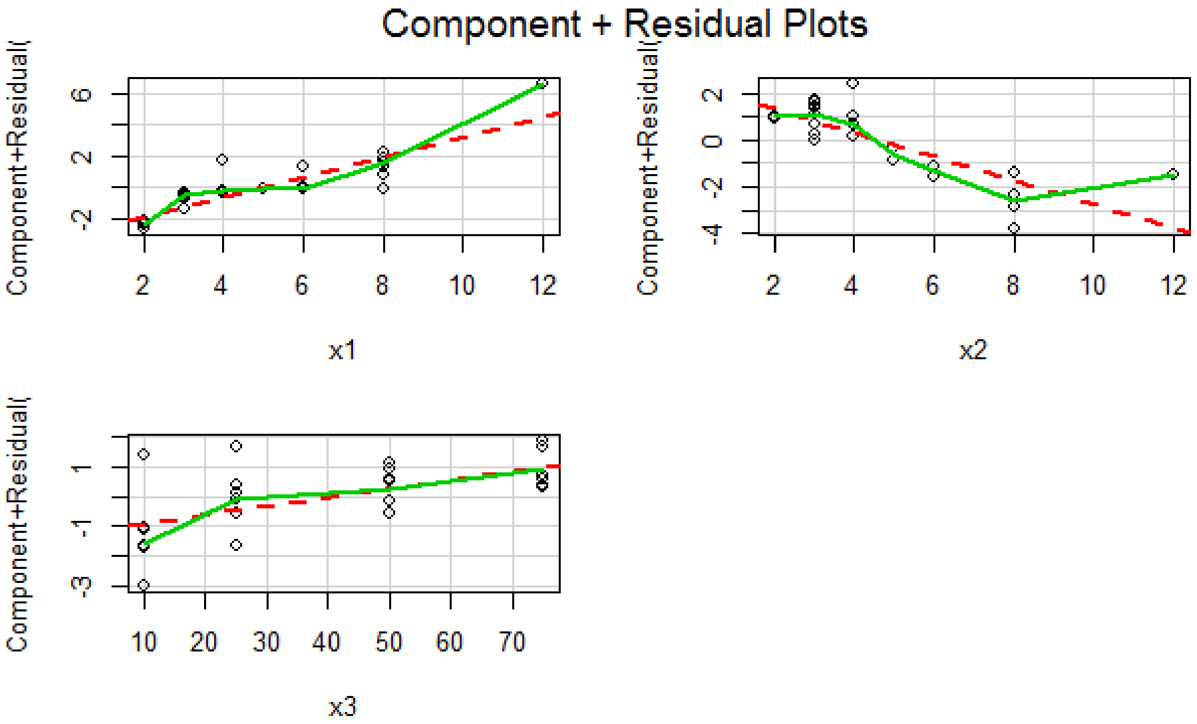
	Estimate	Std. Error	t	p value
(Intercept)	3.58315	0.726839	4.93	0.0001
x1	0.651547	0.0855	7.62	0.0000
x2	-0.509866	0.097132	-5.249	0.0000
x3	0.028888	0.009021	3.202	0.00428

R^2	0.786
Adjusted R^2	0.7563

MODELING NONLINEAR RELATIONS

Develop the model

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

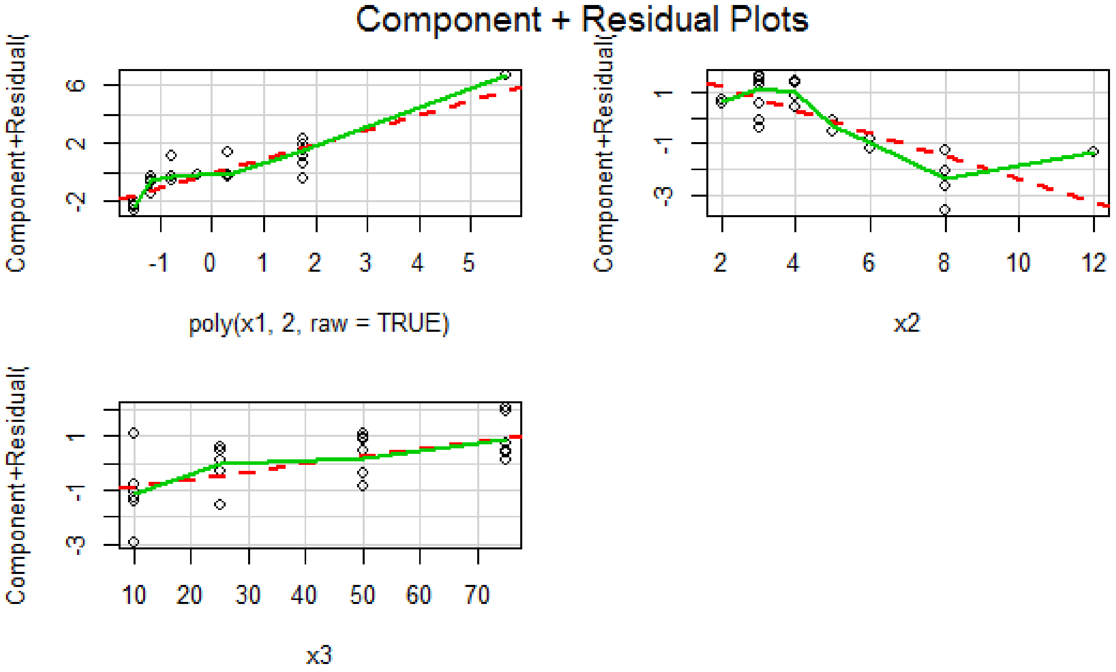


Since the best fit line different from residual line, it is possible improve the model by adding higher order terms

MODELING NONLINEAR RELATIONS

Develop the model

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

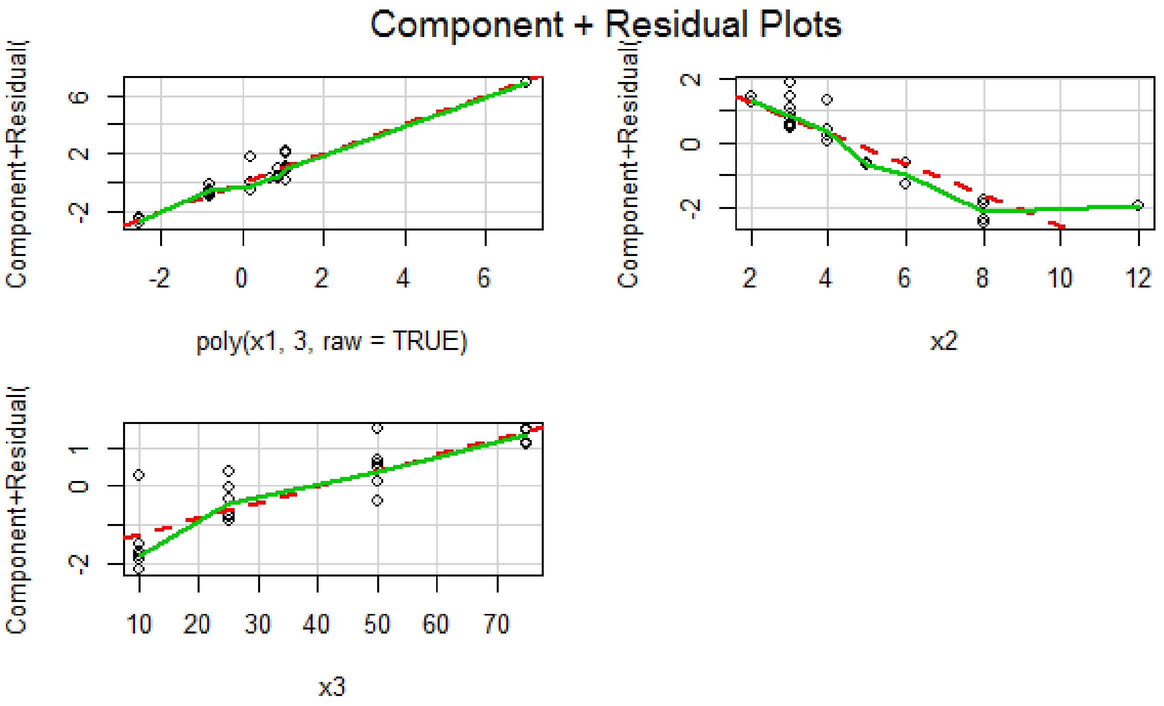


Since the best fit line different from residual line, it is possible improve the model by adding higher order terms

MODELING NONLINEAR RELATIONS

Develop the model

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

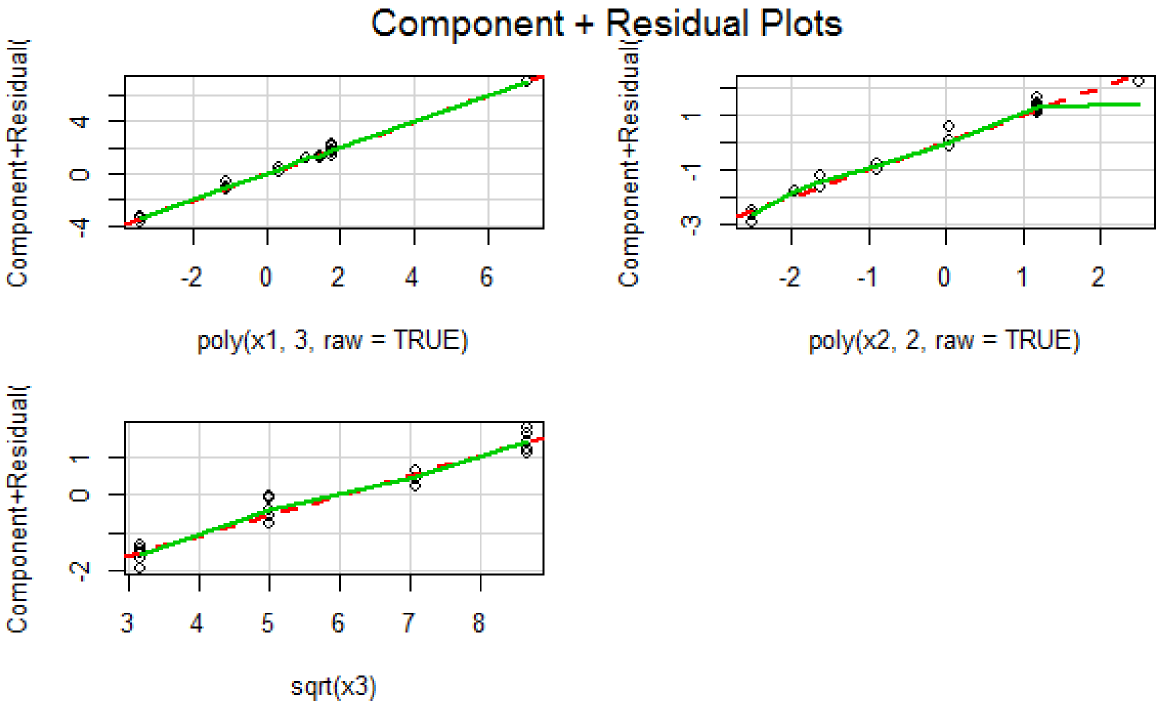


Since the best fit line is more or less overlapping residual line, hence adding square and cube terms of x1 will improve the model. Similarly add additional terms or functions of x2 and x3 to improve the model

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

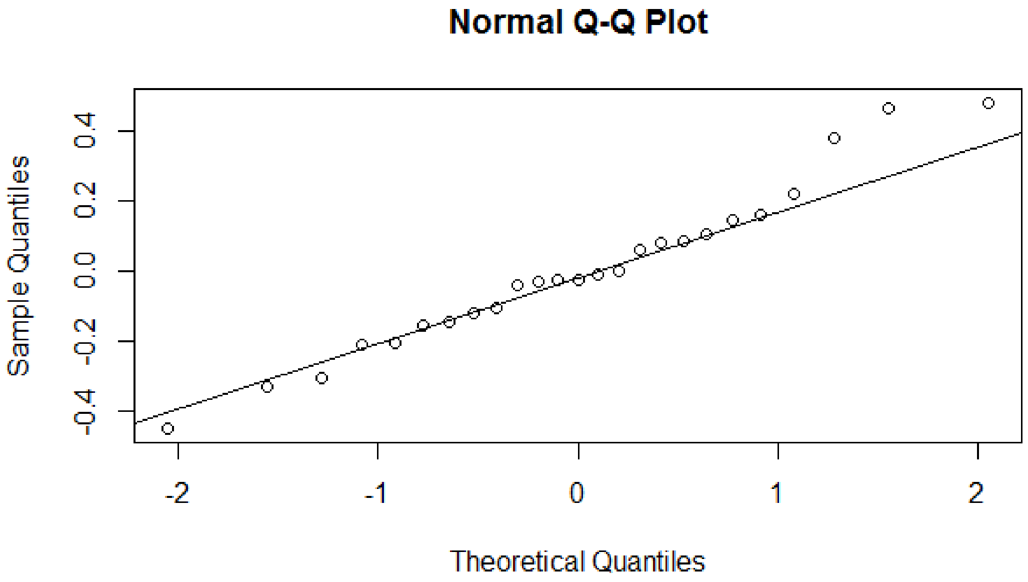
	Estimate	Std. Error	t	p value
(Intercept)	-3.48301	0.705793	-4.935	0.000107
x_1	5.503467	0.36278	15.17	0.0000
x_1^2	-0.77878	0.056814	-13.708	0.0000
x_1^3	0.037516	0.002685	13.971	0.0000
x_2	-1.81437	0.146304	-12.401	0.0000
x_2^2	0.097886	0.010374	9.435	0.0000
$\sqrt{x_3}$	0.527417	0.030664	17.2	0.0000

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

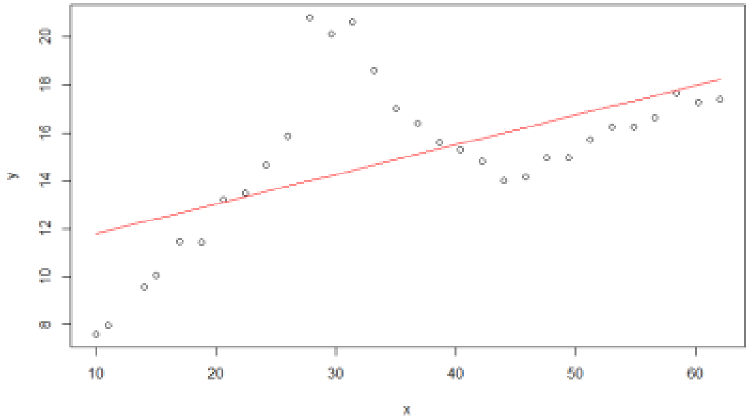
- Function is continuous
- Function has D – 1 continuous derivatives

Usage

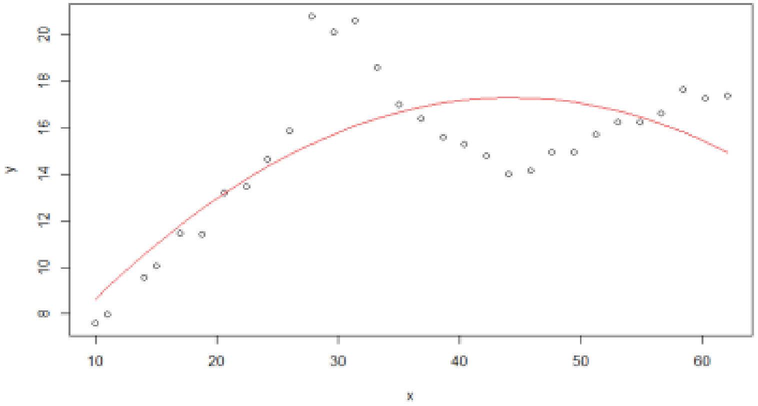
Develop models when relationship between y and x's is piecewise polynomial

REGRESSION SPLINES

y vs x (linear)

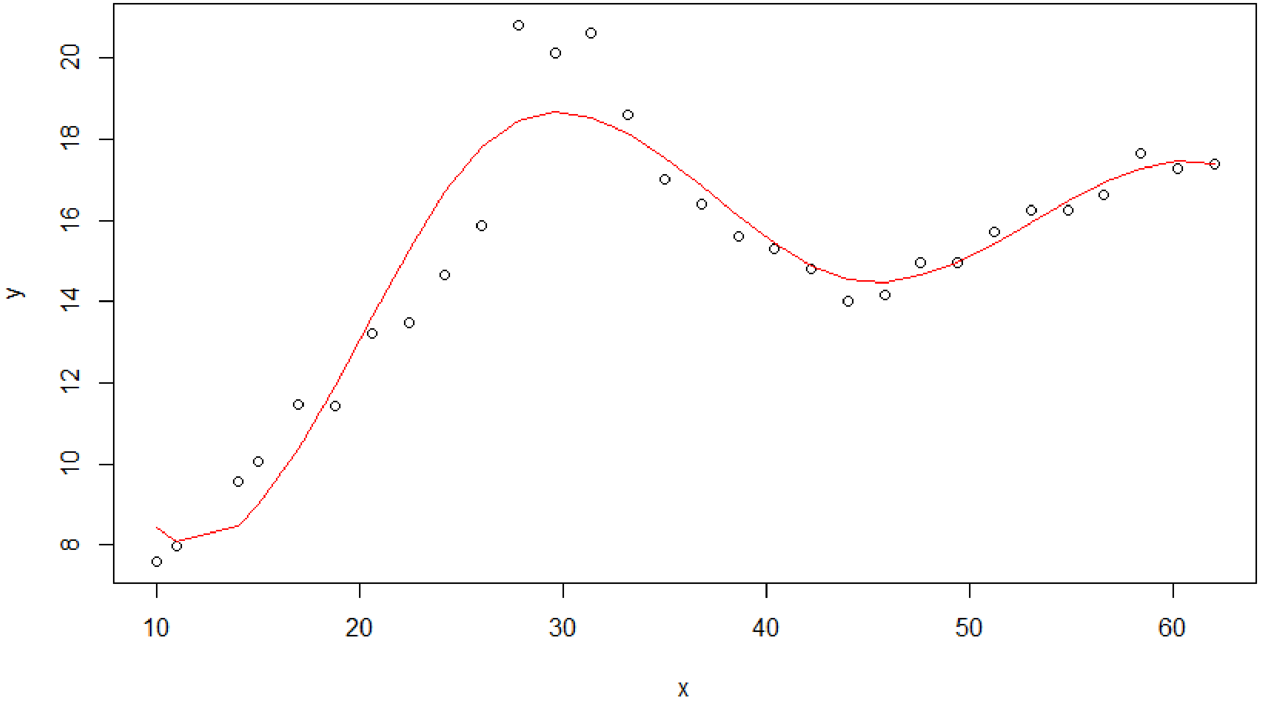


y vs x (Polynomial)



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?

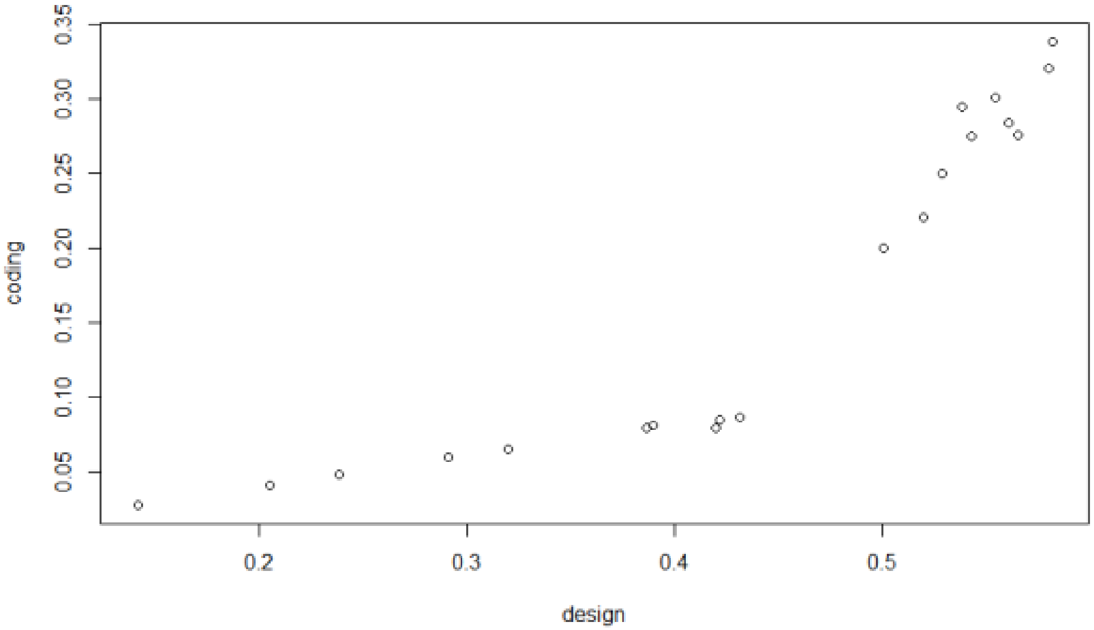
Reading data

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

Example 1:

Exploring the relationship

```
> plot(design, coding)
```



REGRESSION SPLINES

Example 1:

Fitting a linear model

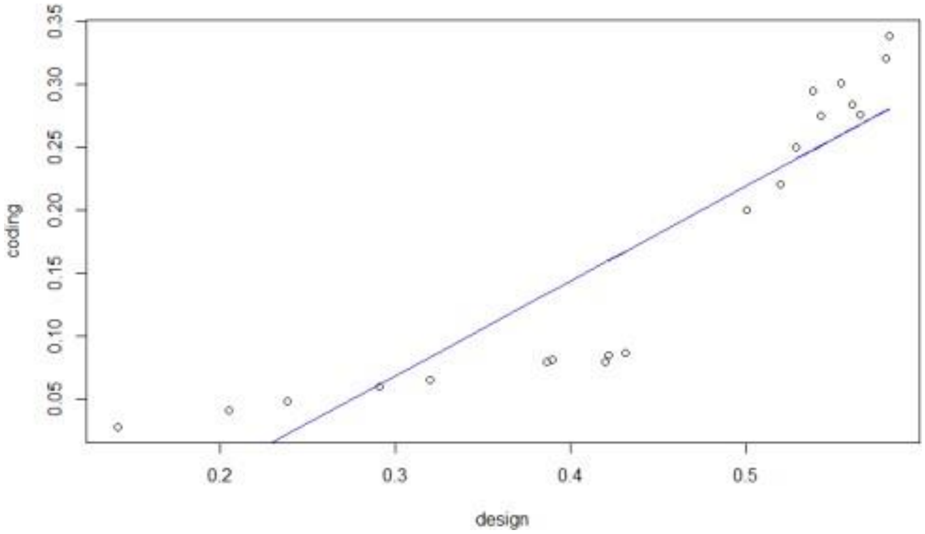
```
> mymodel = lm(coding ~ design)
> summary(mymodel)
```

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

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)
```

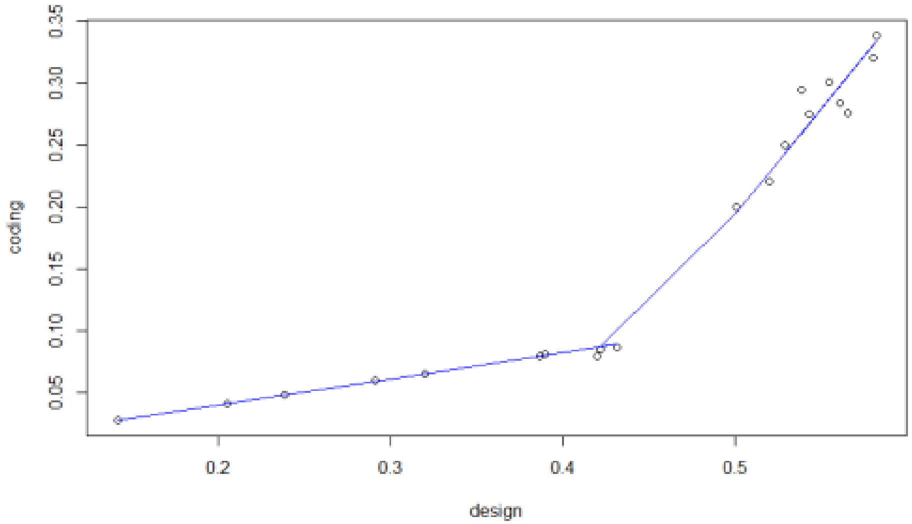
Statistics	Value
R^2	0.9823
R^2 adjusted	0.9802
F Statistics	472.2
P value	0.000

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

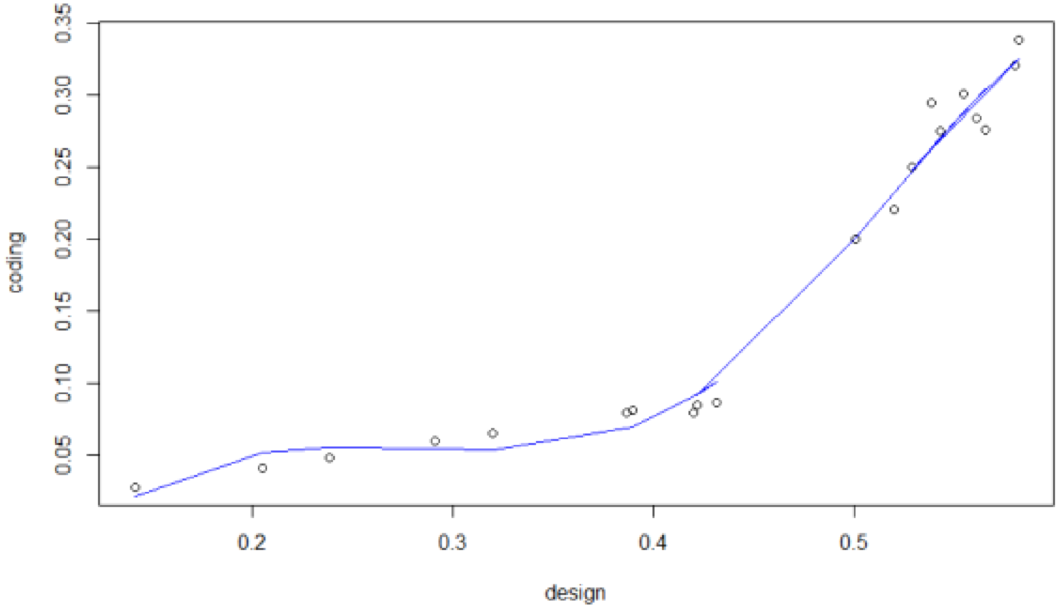
```
> designsq = design^2
> designcb = design^3
> design44cb = design44^3
> mymodel = lm(coding ~ poly(design, 3, raw =TRUE) + design44cb)
> summary(mymodel)
```

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

Example 1:

Plotting the linear spline model

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

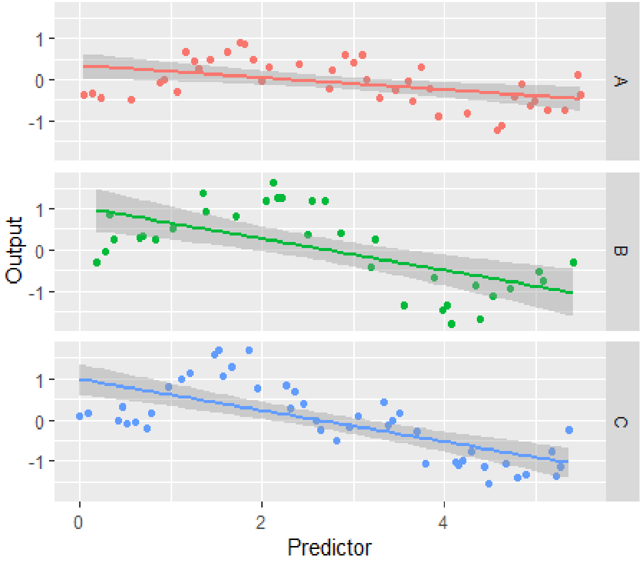


Homework: EXPLORE Multivariate Adaptive Regression Splines (**MARS**)

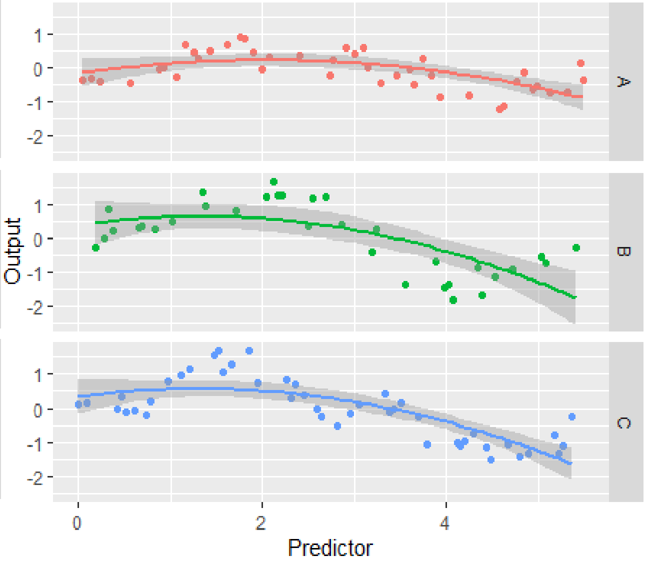
Regression in R

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)

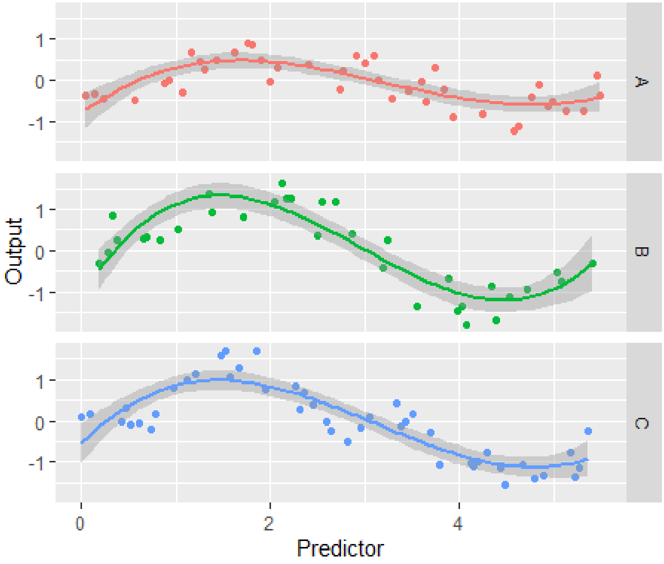
Linear Regression



Quadratic Regression

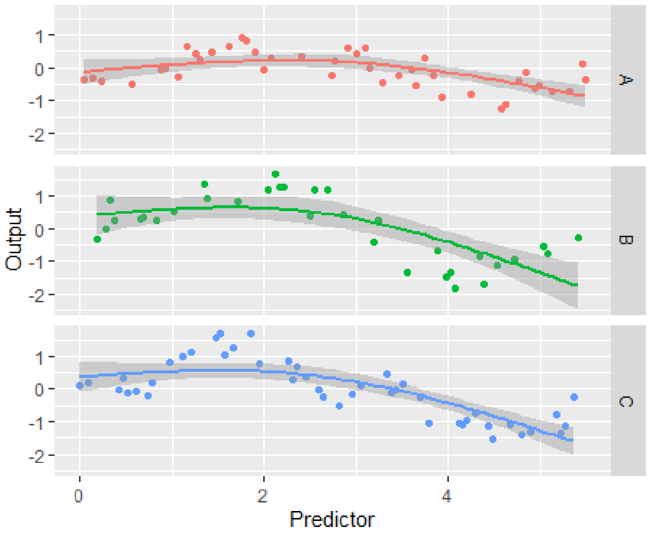


Cubic Regression

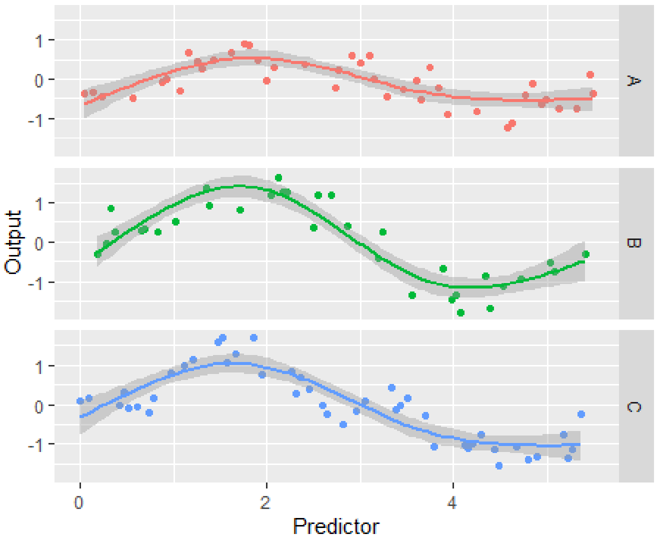


From the above graphs, we can conclude that the data is better modelled using a cubic regression function rather than a linear regression or a quadratic regression.

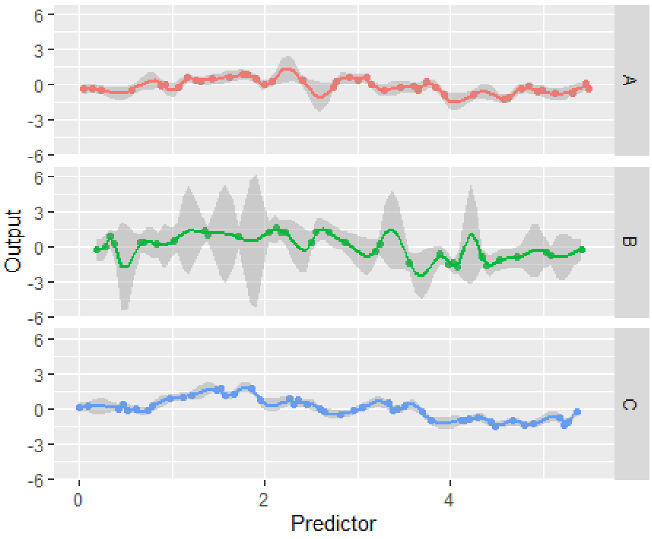
Natural splines with 2 knots



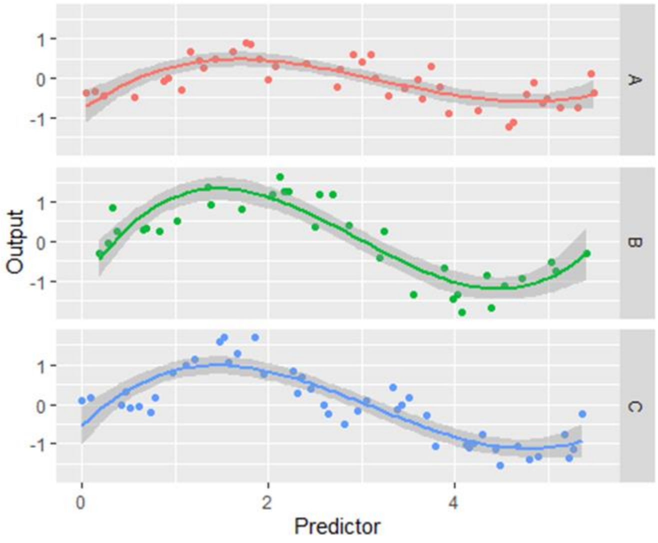
Natural splines with 3 knots



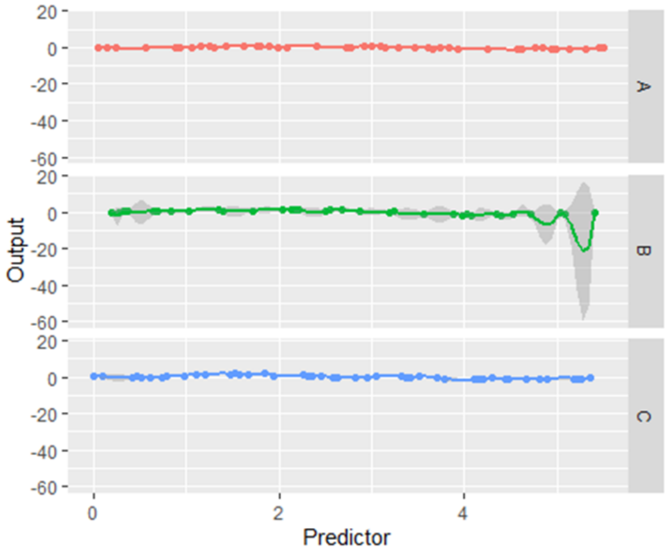
Natural splines with 30 knots



B splines with 3 knots



B splines with 30 knots


Output
A
B
C
Predictor