Chapter 2: Generalised Linear Model (GLM)

Noryanti Muhammad
Centre for Mathematical Sciences
College of Computing and Applied Sciences
Universiti Malaysia Pahang

Centre of Excellence (CoE) for Data Science & Artificial Intelligence
Research & Innovation Department
Universiti Malaysia Pahang

Expected Outcomes:

By the end of this chapter, students should be able:

- ✓To determine the normality of the model and data.
- ✓To investigate the model by using model adequacy checking including residual analysis, detection outliers, lack of fit.
- ✓To solve the problem occurs in the model using transformation and weighting.
  - Understand the concept of a variance stabilizing transformation.
  - Use transformations of the response or predictors to improve regression models.
- ✓To analyse more details the model by investigate the leverage and influence variables.
- ✓To investigate the multicollinearity in the model and solve the problem.

Content:

2.1 Normality Test

- Assessment of the appropriate **residual plots** is sufficient to diagnose deviations from normality.
- However, more rigorous and formal quantification of normality may be requested. So, this section provides a discussion of some common testing procedures (of which there are many) for normality.
- For each test discussed, the **formal hypothesis test** is written as:
  - ➢Kolmogorov Smirnov (K-S)
  - ➢Anderson-Darling Test
  - ➢Shapiro-Wilk Test

Hypothesis
H0: The errors/data follows Normal Distribution
H1: The errors/data does not follow Normal Distribution

2.1 Normality Test

Kolmogorov Smirnov (K-S)

- There is the **one-sample** K–S test which is used to test the normality of a selected continuous variable.
- There is the **two-sample** K–S test which is used to test whether two samples have the same distribution or not.

Example: The results show that D(200) = 0.9949, $p < .01$, meaning that there is a statistically significant deviation from normality. Therefore, we can reject the null hypothesis of no deviation from normality in relation to the variable sales. Sales is not normally distributed, which confirms the earlier interpretation of the histogram of sales.

```
#One-Sample K–S Test
ks.test(marketing$sales, "pnorm")

#Two-Sample K–S Test
ks.test(marketing$sales, marketing$youtube)
```

```
> #One-Sample K-S Test
> ks.test(marketing$sales, "pnorm")

        One-sample Kolmogorov-Smirnov test

data:  marketing$sales
D = 0.99494, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(marketing$sales, "pnorm") :
  ties should not be present for the Kolmogorov-Smirnov test
> #Two-Sample K-S Test
> ks.test(marketing$youtube, marketing$facebook)

        Two-sample Kolmogorov-Smirnov test

data:  marketing$youtube and marketing$facebook
D = 0.82, p-value < 2.2e-16
alternative hypothesis: two-sided

Warning message:
In ks.test(marketing$youtube, marketing$facebook) :
  p-value will be approximate in the presence of ties
>
```

2.1 Normality Test

Anderson-Darling Test

- The Anderson–Darling test is a statistical test of whether a given sample of data is drawn from a given probability distribution.
- When applied to testing whether a normal distribution adequately describes a set of data, it is one of the most powerful statistical tools for detecting most departures from normality.

```
#Anderson-Darling Test to test for normality
ad.test(marketing$sales)
```

```
> #conduct Anderson-Darling Test to test for normality
> ad.test(marketing$sales)

        Anderson-Darling normality test

data:  marketing$sales
A = 1.7373, p-value = 0.0001831
```

Example:
A: is a test statistics values
p-value < 0.01, reject H0, the data does not follow a Normal Distribution

2.1 Normality Test

Shapiro-Wilk Test

- Similar like the Anderson–Darling test.

```
#conduct shapiro wilk Test to test for normality
shapiro.test(marketing$youtube)
```

```
> shapiro.test(marketing$youtube)

        Shapiro-Wilk normality test

data:  marketing$youtube
W = 0.94951, p-value = 1.693e-06
```

Example:
W: is a test statistics values
p-value < 0.01, reject H0, the data does not follow a Normal Distribution

2.2 Model Adequacy Checking

The **fitting** of the linear regression model, estimation of parameters testing of hypothesis properties of the estimator, is based on the following major assumptions:

- The relationship between the study variable and explanatory variables is linear, at least approximately.
- The error term has zero mean.
- The error term has a constant variance.
- The errors are uncorrelated (if exist correlation, called autocorrelation).
- The errors are normally distributed.

For instance, we should routinely plot the residuals against:

- the fitted values (to look for heteroscedasticity);
- the explanatory variables (to look for evidence of curvature);
- the sequence of data collection (to took for temporal correlation);
- standard normal deviates (to look for non-normality of errors).

2.2 Model Adequacy Checking

Linear regression model makes several assumptions about the data, such as :

Linearity of the data. The relationship between the predictor (x) and the outcome (y) is assumed to be linear.

Normality of residuals**.** The residual errors are assumed to be normally distributed.

Homogeneity of residuals variance. The residuals are assumed to have a constant variance (**homoscedasticity**)

Independence of residuals error terms.

Potential problems include:

Non-linearity of the outcome - predictor relationships

Heteroscedasticity: Non-constant variance of error terms.

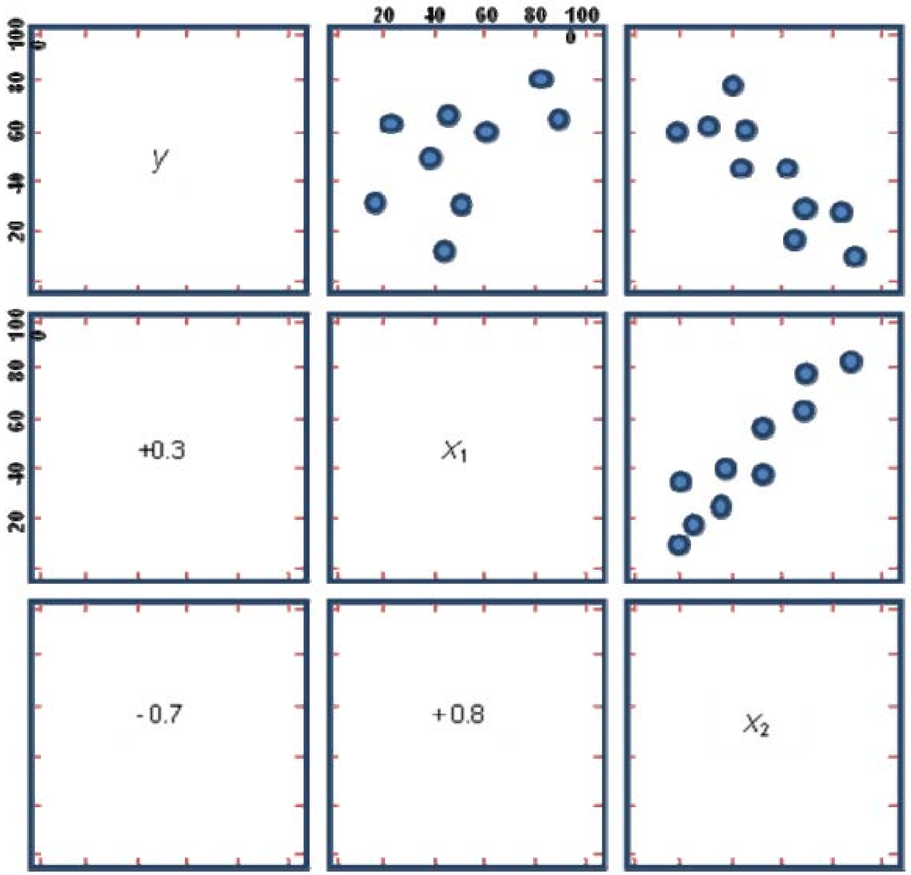
Presence of influential values in the data that can be:

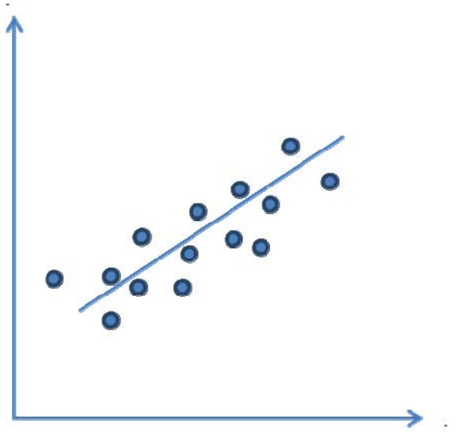
Outliers: extreme values in the outcome (y) variable

High-leverage points: extreme values in the predictors (x) variable

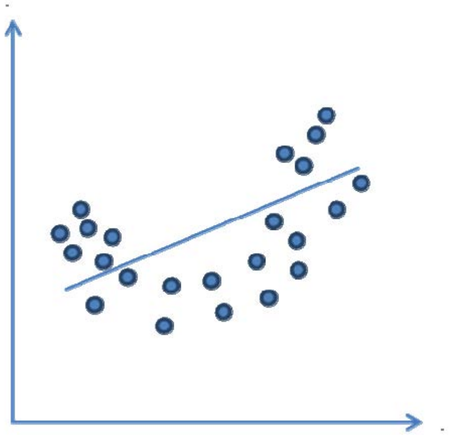
Linearity of the data

The existence of the linear relationship between y and X by scatter diagram of the available data.





Linear

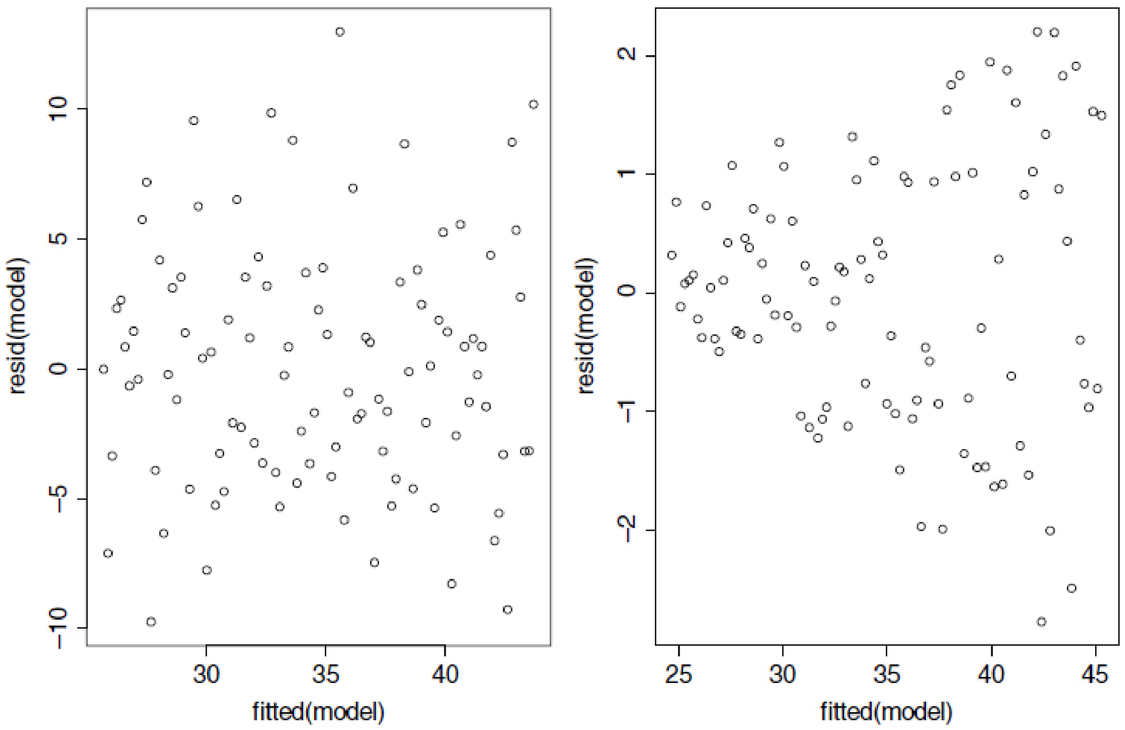


Not Linear

Heteroscedasticity: Non-constant variance of error terms

A plot of standardized residuals/residual against fitted values.

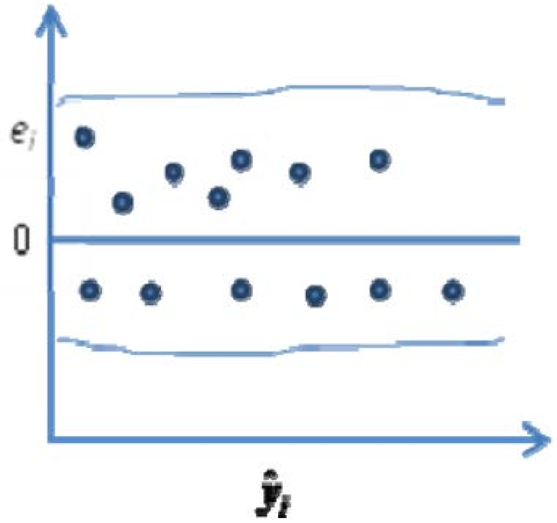
A common problem is that the variance increases with the mean, so that we obtain an expanding, fan-shaped pattern of residuals (right-hand panel).



homoscedasticity

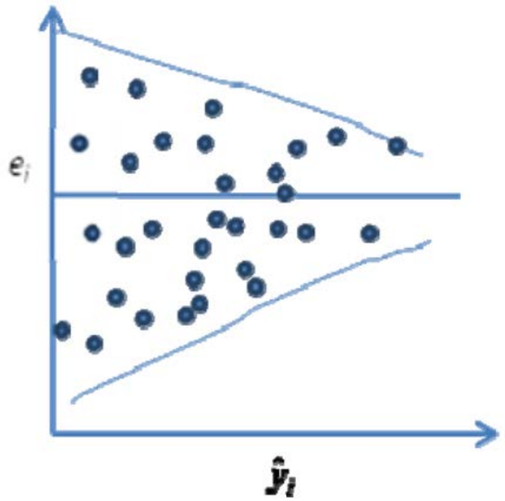
heteroscedasticity

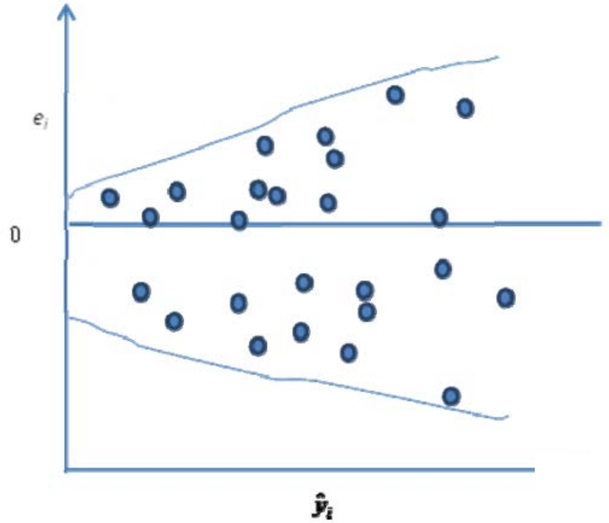
Type of Residual Plot Pattern



no visible model defects

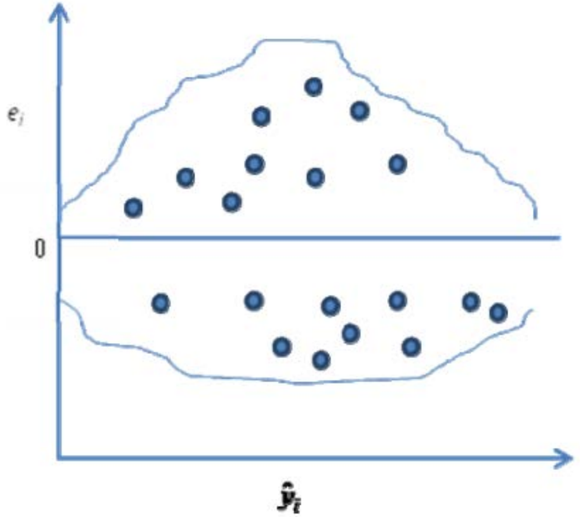
The variance of errors is not constant, but it is decreasing function of y





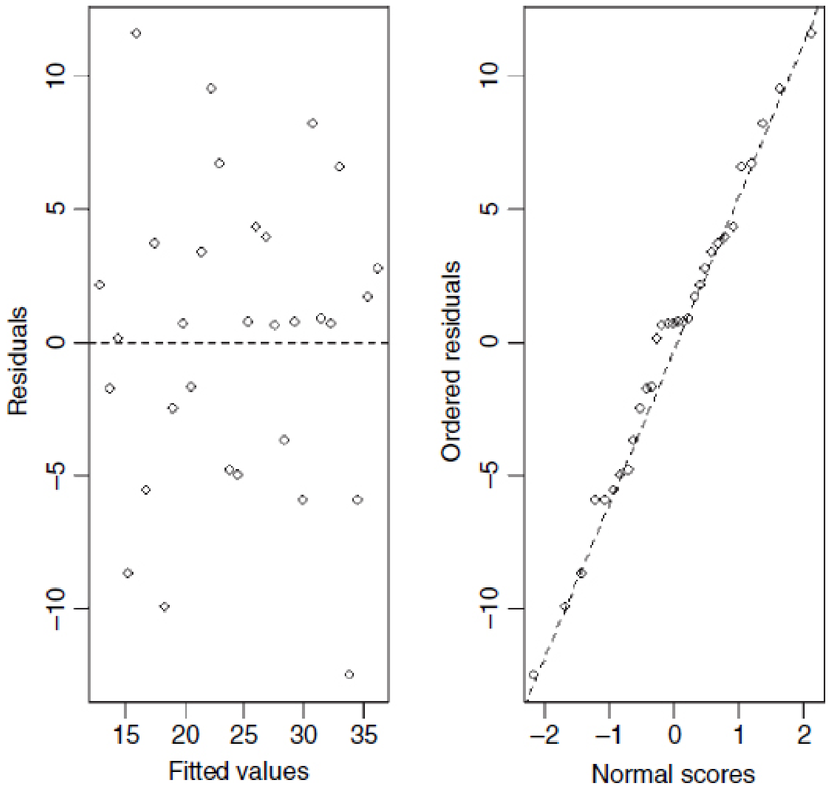
The variance of errors is not constant, but it is an increasing function of y

The variance of errors is not constant, but y is a proportion between 0 and 1

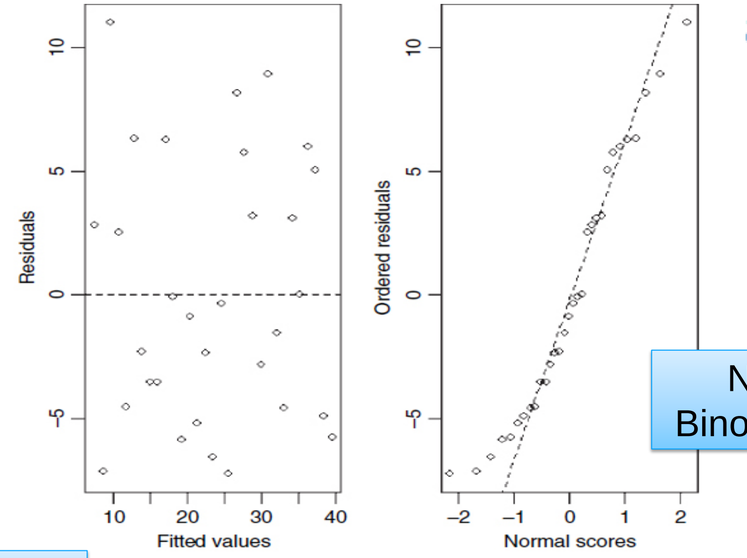


Non-normality of errors

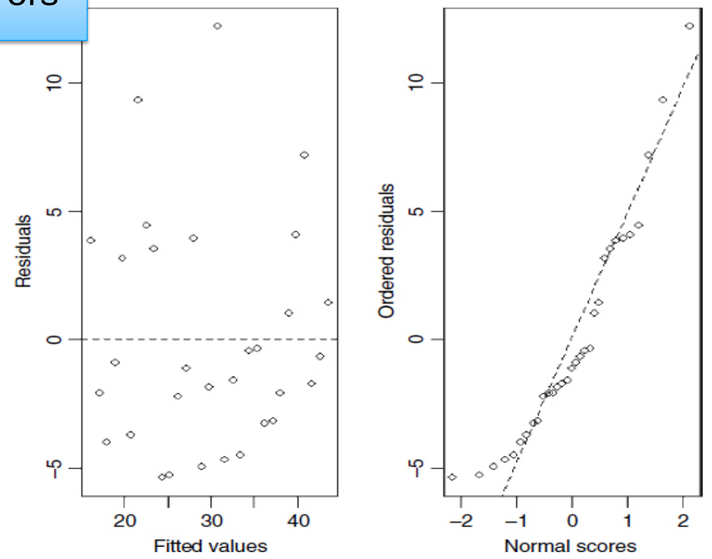
Residual Plots vs Fitted Values



Negative Binomial errors



Gamma errors



Plot Residual in Time sequence

If the time sequence in which the data were collected is known, then the residuals can be plotted against the time order.

- The graph is plot as Residual vs Time order.
- The interpretation of the plots is the same as in the case of plots of residuals versus fitted values.
- The **correlation between model errors** at different time periods is called **autocorrelation**.

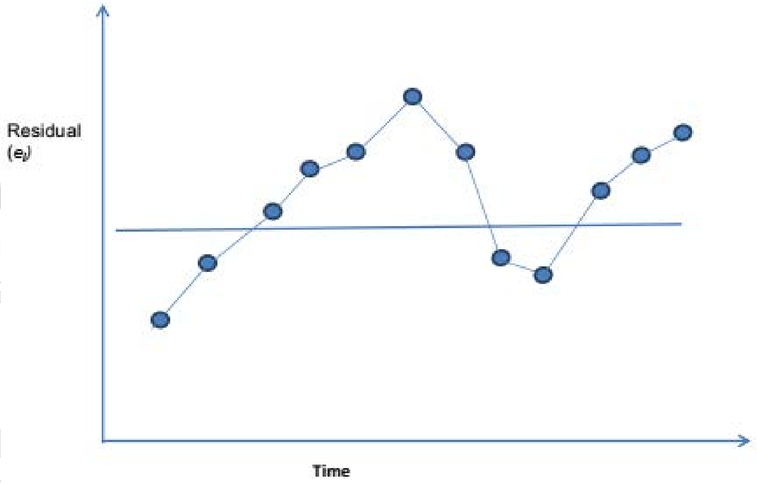
Plot Residual in Time sequence

- If all the residuals are contained in

  - a horizontal band, and the residuals fluctuat more or less in a random fashion within th band, then it is desirable and indicates th there are no obvious model deflects.

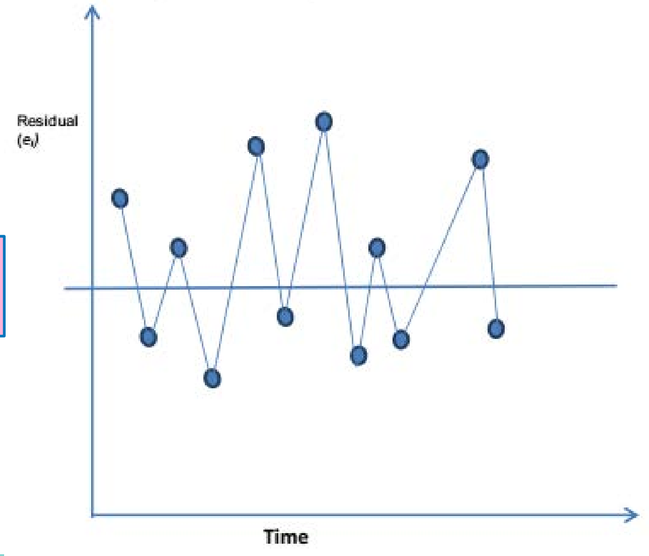
  - An outward opening funnel shape or inwar opening funnel shape, then it indicates th the variance is not constant but changing with time.

  - Double bow pattern or nonlinear pattern, then it indicates that the assumed relationship is nonlinear. In such a case, the linear or quadratic terms in time should be added to the model.



Positive Correlations

Negative Correlations

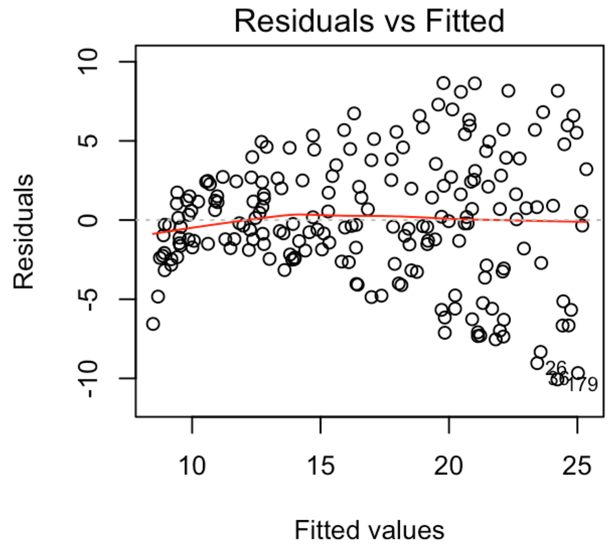


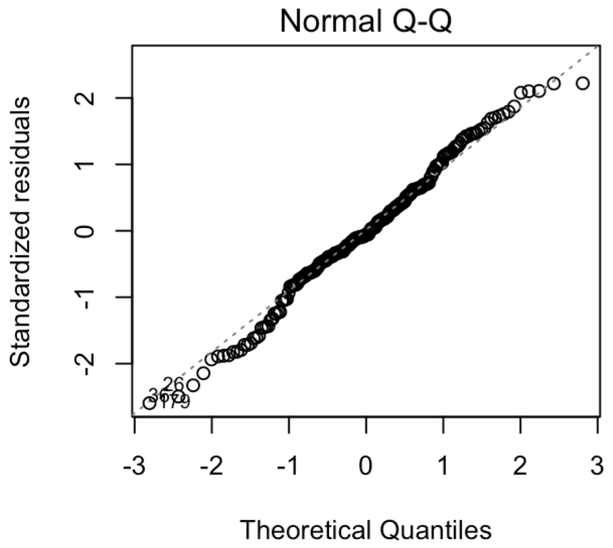
Alternative Plot

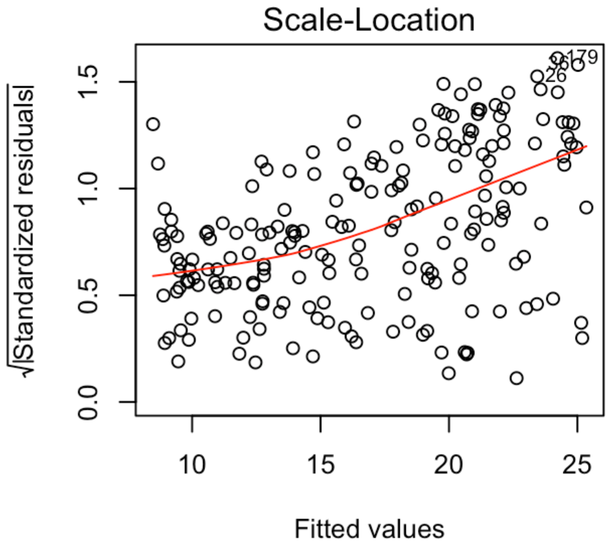
```
#summary plot
plot(model)
```

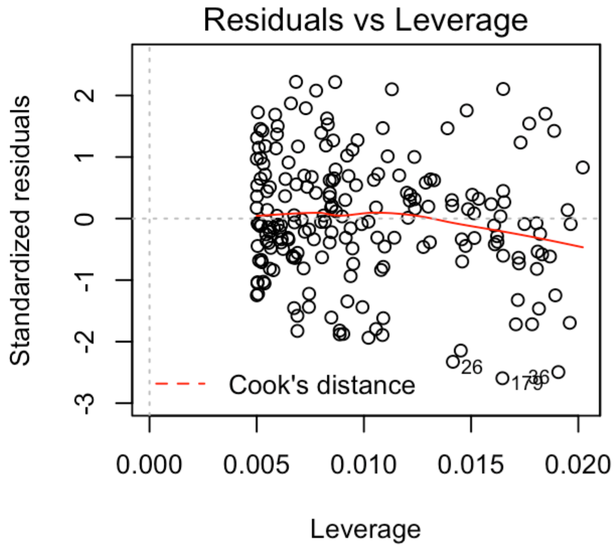
The diagnostic plots show residuals in four different ways:

- **Residuals vs Fitted**. Used to check the **linear relationship assumptions**. A horizontal line, without distinct patterns is an indication for a linear relationship, what is good.
- **Normal Q-Q**. Used to examine whether the **residuals are normally distributed**. It's good if residuals points follow the straight dashed line.
- **Scale-Location** (or Spread-Location). Used to check the **homogeneity of variance of the residuals** (homoscedasticity). Horizontal line with equally spread points is a good indication of homoscedasticity. This is not the case in our example, where we have a heteroscedasticity problem.
- **Residuals vs Leverage**. Used to **identify influential cases**, that is extreme values that might influence the regression results when included or excluded from the analysis. This plot will be described further in the next sections.









Detection Outliers

-extreme values

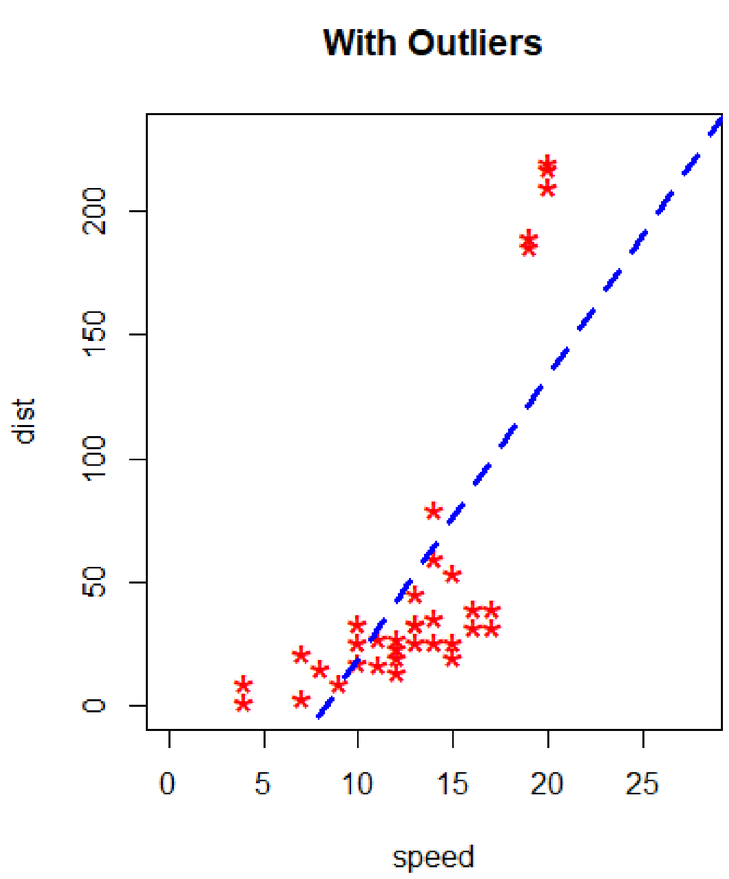
- Outliers can drastically bias/change the fit estimates and predictions.
- There are many ways to detect the outliers, can be categorise into two category i.e univariate and multivariate model approach.
- Declaring an observation as an outlier based on a just one (rather unimportant) feature could lead to unrealistic inferences.

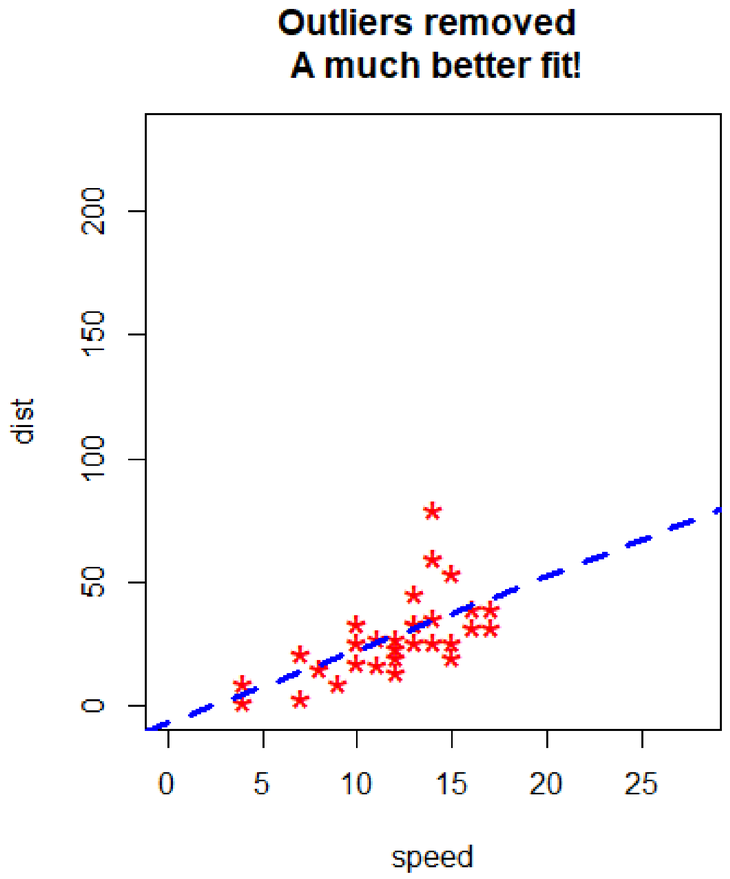
Univariate & Bivariate	Multivariate
•Boxplot •Scatter Plot	•Cooks Distance •Outliers Test

Outliers Example

-Visualize in box-plot of the X and Y

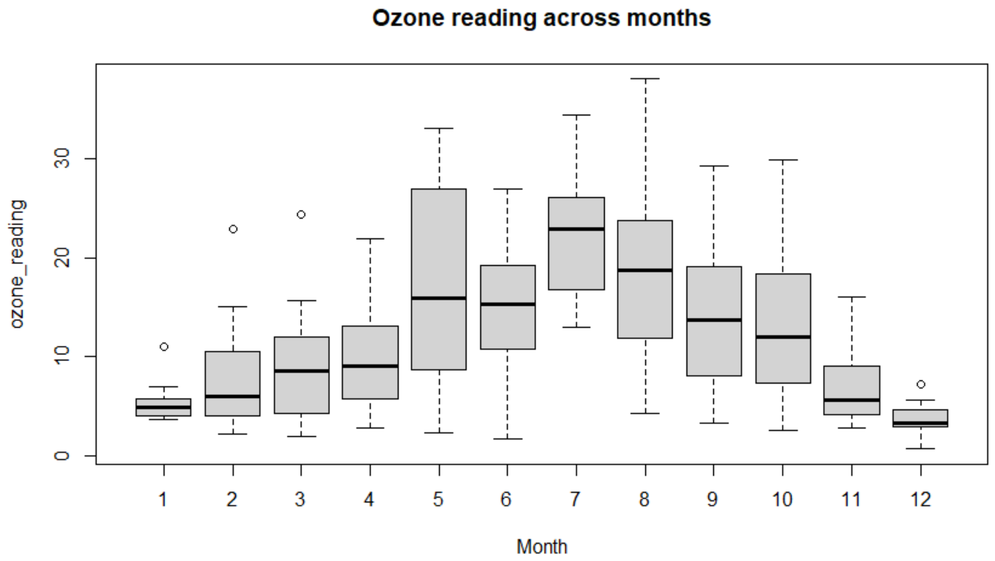
```
# Plot of data with outliers.
par(mfrow=c(1, 2))
plot(cars2$speed, cars2$dist, xlim=c(0, 28), ylim=c(0, 230), main="With Outliers", xlab="speed", ylab="dist", pch="*", col="red", cex=2)
abline(lm(dist ~ speed, data=cars2), col="blue", lwd=3, lty=2)

# Plot of original data without outliers. Note the change in slope (angle) of best fit line.
plot(cars1$speed, cars1$dist, xlim=c(0, 28), ylim=c(0, 230), main="Outliers removed \n A much better fit!", xlab="speed", ylab="dist", pch="*", col="red", cex=2)
abline(lm(dist ~ speed, data=cars1), col="blue", lwd=3, lty=2)
```

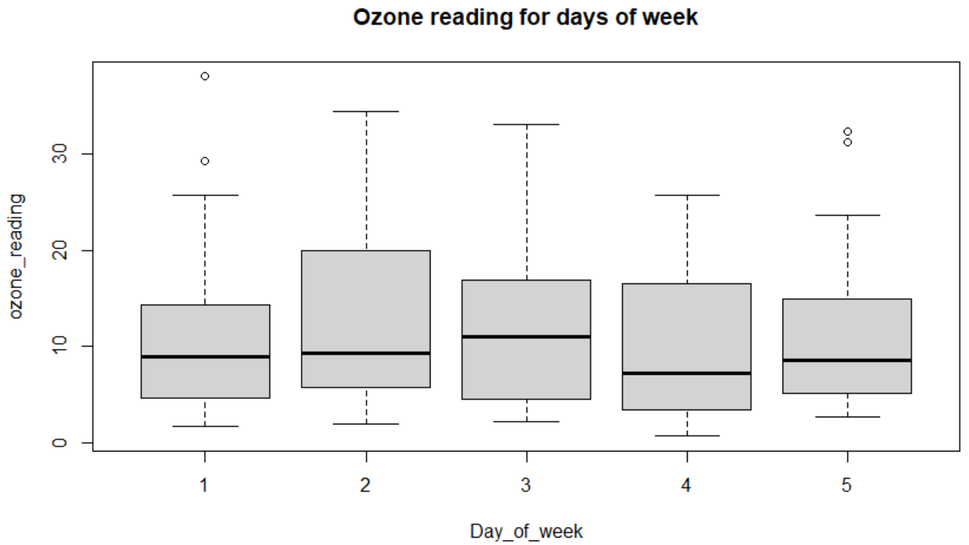




Boxplot

-Visualize in box-plot of the X and Y, for categorical X's



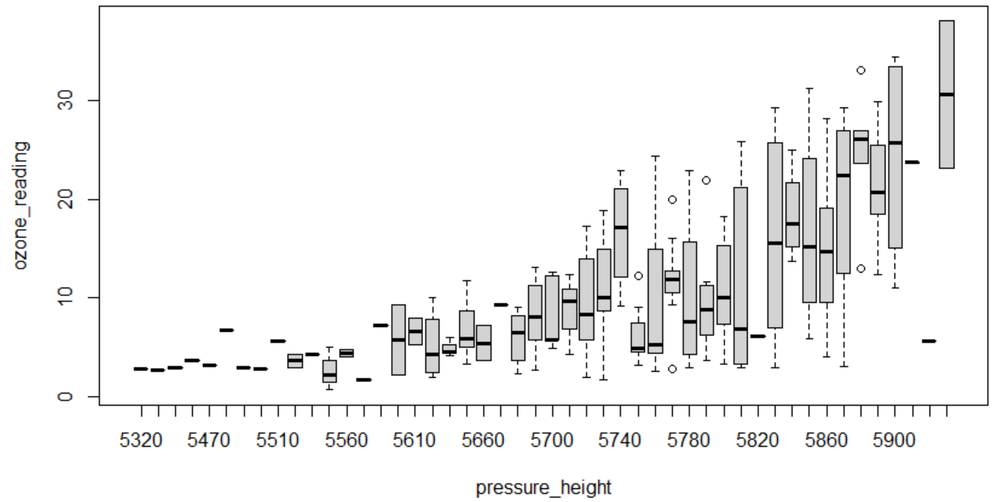


 # For categorical variable
boxplot(ozone_reading ~ Month, data=inputData, main="Ozone reading across months") # clear pattern is noticeable.
boxplot(ozone_reading ~ Day_of_week, data=inputData, main="Ozone reading for days of week") # this may not be significant, as day of week variable is a subset of the month var.

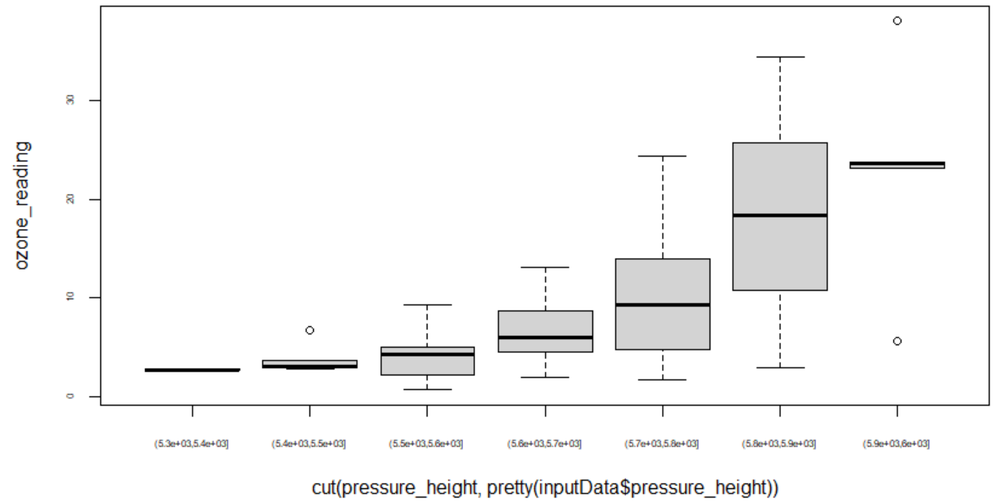
Boxplot

-Visualize in box-plot of the X and Y, (convert to categorical if needed.)

Continuous



Categorical

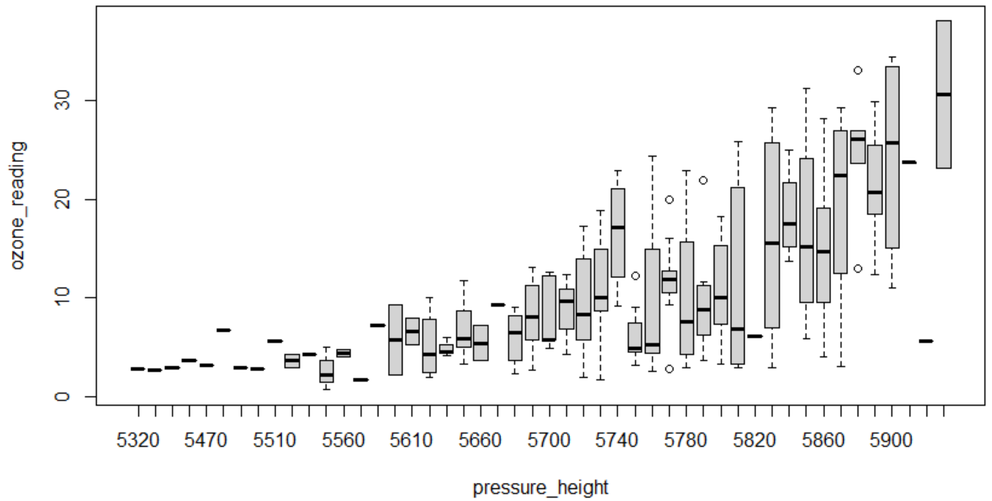



```
# For continuous variable (convert to categorical if needed.)
boxplot(ozone_reading ~ pressure_height, data=inputData, main="Boxplot for Pressure height (continuos var) vs Ozone")
boxplot(ozone_reading ~ cut(pressure_height, pretty(inputData$pressure_height)), data=inputData, main="Boxplot for Pressure height (categorial) vs Ozone", cex.axis=0.5)
```

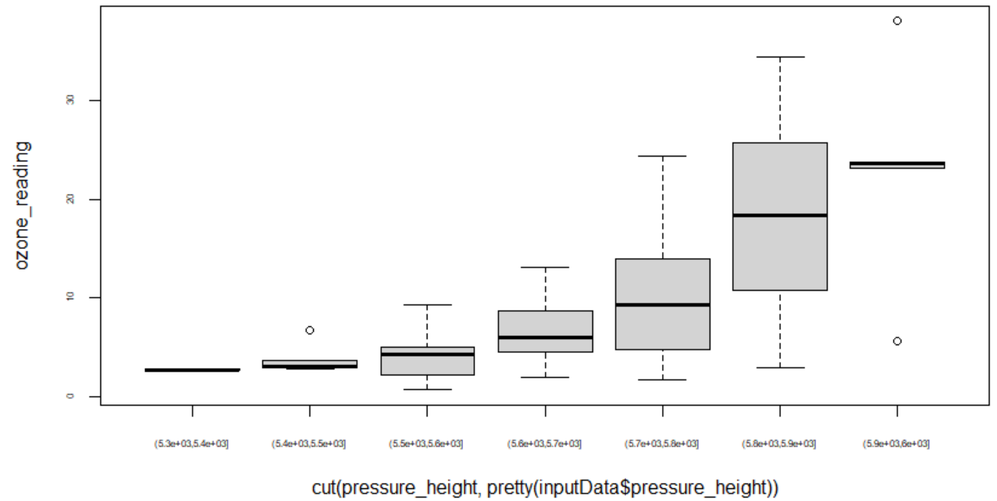
Boxplot

-Visualize in box-plot of the X and Y, (convert to categorical if needed.)

Continuous



Categorical




```
# For continuous variable (convert to categorical if needed.)
boxplot(ozone_reading ~ pressure_height, data=inputData, main="Boxplot for Pressure height (continuos var) vs Ozone")
boxplot(ozone_reading ~ cut(pressure_height, pretty(inputData$pressure_height)), data=inputData, main="Boxplot for Pressure height (categorial) vs Ozone", cex.axis=0.5)
```

Cooks Distance

Cook's distance is a measure computed with respect to a given regression model and therefore is impacted only by the X variables included in the model.

- Computes the influence exerted by each data point (row) on the predicted outcome.
- The cook's distance for each observation *i* measures the change in $\hat{Y}$ (fitted Y) for all observations with and without the presence of observation *i*, so we know how much the observation *i* impacted the fitted values.
- Mathematically, cook's distance Di for observation i is computed as:

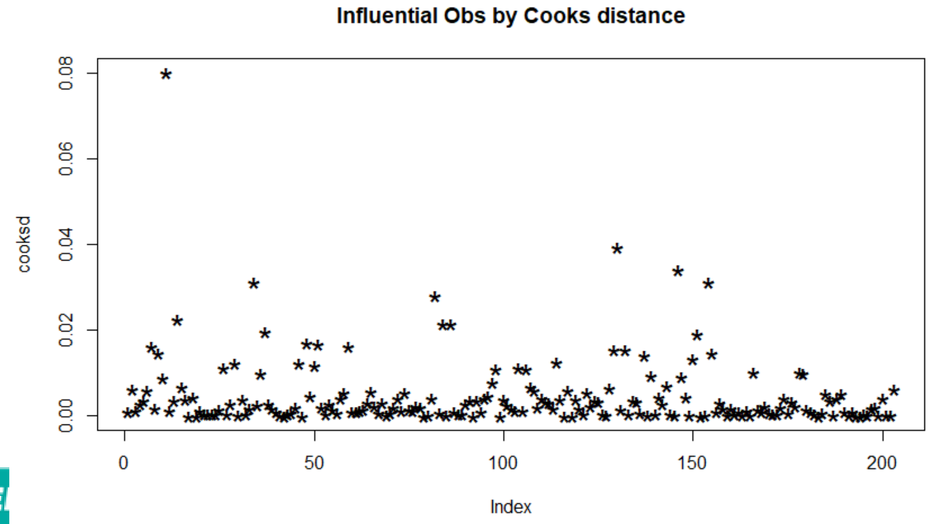
$$D_i = \frac{\sum_{j=1}^{n} \left(\hat{Y}_j - \hat{Y}_{j(i)} \right)^2}{p \times MSE}$$

where,

- $\hat{Y}_j$ is the value of j_{th} fitted response when all the observations are included.
- $\hat{Y}_{j(i)}$ is the value of j_{th} fitted response, where the fit does not include observation *i*.
- *MSE* is the mean squared error.
- *p* is the number of coefficients in the regression model.

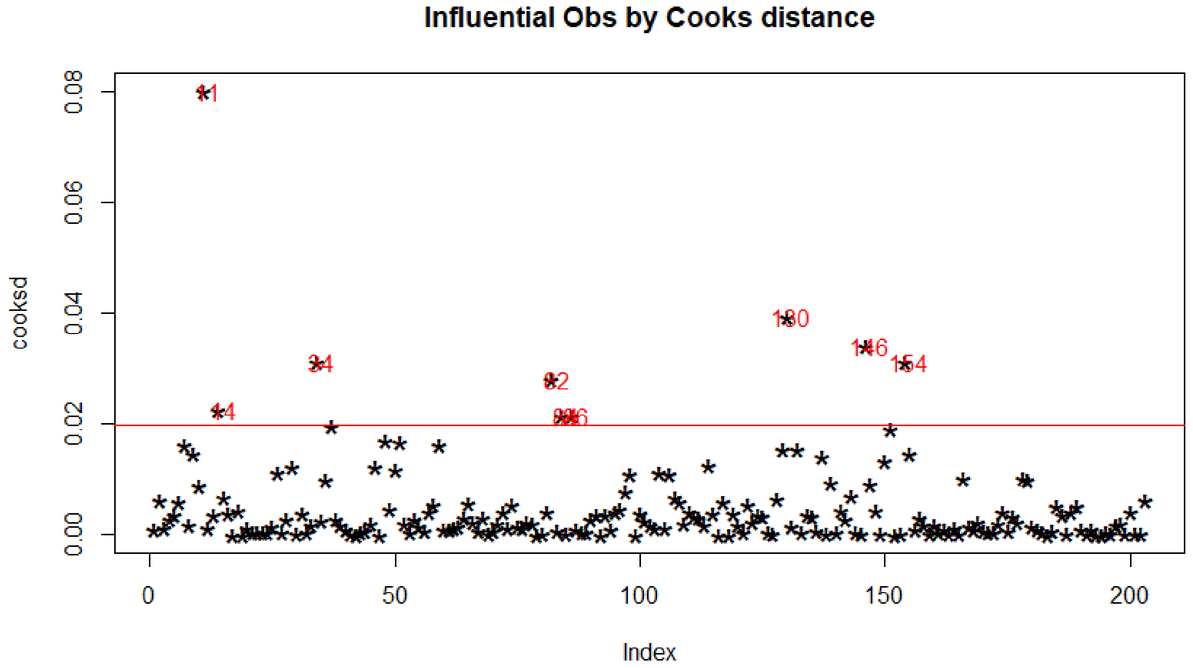
Cooks Distance

```
#Cook distance
model <- lm(ozone_reading ~ ., data=inputData)
cooksd <- cooks.distance(model)

plot(cooksd, pch="*", cex=2, main="Influential Obs by Cooks
distance")  # plot cook's distance
abline(h = 4*mean(cooksd, na.rm=T), col="red")  # add cutoff line
text(x=1:length(cooksd)+1, y=cooksd,
labels=ifelse(cooksd>4*mean(cooksd,
na.rm=T),names(cooksd),""), col="red")  # add labels
```



Influence measures

- In general use, those observations that have a cook's distance **greater than 4 times the mean** may be classified as influential.



Cooks Distance

Examine the influential rows

- Extract and examine each influential row 1-by-1 (from output), Then, we be able to reason out why that row turned out influential.
- It is likely that one of the X variables included in the model had extreme values.

```
# influential row numbers
influential <- as.numeric(names(cooksd)[(cooksd > 4*mean(cooksd, na.rm=T
head(inputData[influential, ])  # influential observations.
```

By examine the first 6 rows from the output to find out why these rows could be tagged as ***influential observations***.

•Row 32, 82, 84 - have very high ozone_reading.

•Rows 14, 84 and 86 - have very high Inversion_base_height.

•Row 11 has very low Pressure_gradient.

```
> # influential row numbers
> influential <- as.numeric(names(cooksd)[(cooksd > 4*mean(cooksd, na.rm=T))])
> influential
[1]  11  14  34  82  84  86 130 146 154
> head(inputData[influential, ])  # influential observations.
   Month Day_of_month Day_of_week ozone_reading pressure_height Wind_speed Humidity Temperature_Sandburg
11     1           19           1          4.07            5680          5       73                   52
14     1           23           5          4.90            5700          5       59                   69
34     2           27           5         22.89            5740          3       47                   53
82     5           12           3         33.04            5880          3       80                   80
84     5           14           5         31.15            5850          4       76                   78
86     5           28           5          4.82            5750          3       76                   65
   Temperature_ElMonte Inversion_base_height Pressure_gradient Inversion_temperature Visibility
11               56.48                  393                -68                69.80         10
14               51.08                 3044                 18                52.88        150
34               58.82                  885                 -4                67.10         80
82               73.04                  436                  0                86.36         40
84               71.24                 1181                 50                79.88         17
86               51.08                 3644                 86                59.36         70
>
```

Outliers Test

The function *outlierTest* from car

```
# Outlier Test
car::outlierTest(model)
```

```
> # Outlier Test
> car::outlierTest(model)
No Studentized residuals with Bonferroni p < 0.05
Largest |rstudent|:
    rstudent unadjusted p-value Bonferroni p
130 3.045756         0.0026525      0.53845
>
```

- This output suggests that observation in row 130 is most extreme.

Other Tests

- Grubbs's test, Dixon's test and Rosner's test
- Note that the 3 tests are appropriate only when the data (without any outliers) are **approximately normally distributed**. The normality assumption must thus be verified before applying these tests for outliers.

Treating Outliers

- **Imputation**

Imputation with mean / median / mode. This method has been dealt with in detail in the discussion about treating missing values.

```
#imputation
install.packages("Hmisc")
library(Hmisc)
impute(inputData$ozone_reading, mean)  # replace with mean
impute(inputData$pressure_height, median)  # median
impute(inputData$pressure_height, 20)  # replace specific number
```

- **Capping**

For missing values that lie outside the 1.5 * IQR limits, we could cap it by replacing those observations outside the lower limit with the value of 5th %ile and those that lie above the upper limit, with the value of 95th %ile.

```
#capping
x <- inputData$pressure_height
qnt <- quantile(x, probs=c(.25, .75), na.rm = T)
caps <- quantile(x, probs=c(.05, .95), na.rm = T)
H <- 1.5 * IQR(x, na.rm = T)
x[x < (qnt[1] - H)] <- caps[1]
x[x > (qnt[2] + H)] <- caps[2]
```

- **Prediction**

The outliers can be replaced with missing values (NA) and then can be predicted by considering them as a response variable.

Lack-of-Fit Test

- To determine the model **adequately** describe the data.
- A lack of fit test is used to determine whether a **full regression model** offers a **significantly better fit** to a dataset than some reduced version of the model.
- The *F* test-statistic turns out to be **0.7558** and the corresponding *p*-value is **0.5553**.
- Since this p-value is greater than .05, we cannot reject the null hypothesis of the test and conclude that the full model (model1) do not offers a statistically significantly better fit than the reduced model (model2).

```
#Lack of Fit Test
# Data
inputData<-read.csv("G:\\My Drive\\2. Course Teaching\\SEM I20212022\\BSD3443 Statistical Modelling\\Slide\\Sharing\\ozonedata")

#Develop model and consider the significant variables
model1 <- lm(ozone_reading ~ pressure_height+Wind_speed+Humidity+Temperature_Sandburg+Temperature_ElMonte+Inversion_base_height+Pressure_gradient+Inversion_temperature+Visibility, data = inputData)
summary(model1)

model2 <- lm(ozone_reading ~ pressure_height+Humidity+Temperature_Sandburg+Temperature_ElMonte+Inversion_base_height, data = inputData)
summary(model2)

#lack of fit test
anova(model1, model2)
```

```
> #lack of fit test
> anova(model1, model2)
Analysis of Variance Table

Model 1: ozone_reading ~ pressure_height + Wind_speed + Humidity + Temperature_Sandburg +
    Temperature_ElMonte + Inversion_base_height + Pressure_gradient +
    Inversion_temperature + Visibility
Model 2: ozone_reading ~ pressure_height + Humidity + Temperature_Sandburg +
    Temperature_ElMonte + Inversion_base_height
  Res.Df    RSS Df Sum of Sq      F Pr(>F)
1    193 3881.7
2    197 3942.5 -4     -60.8 0.7558 0.5553
>
```

2.3 Transformation and Weighting

- Data Transformation – the choice should be made by the engineer/scientist with subject matters knowledge.
- Weighting – can be used to overcome the non constant variance.

Variance stabilizing Transformations

- A common reason for the violation of assumption id for the response variable (y) to follow the probability distribution in which the variance is functionally related to the mean.
- For example, if y is a Poisson random variable in a model, then the variance of y is equal to the mean. (mean y related to the regressor/explanatory (x), the variance y will be proportional to x).
- Thus, if the distribution of y is Poisson, we could regress $y' = \sqrt{y}$ against x since the variance od the square root of a Poisson random variable is independent of the mean.
- The strength of the transformation depends on the amount of curvature that is includes.
- We also can use prior experience or theoretical considerations to guide us in selecting an appropriate transformation.

Variance stabilizing Transformations

Common and useful Variance stabilizing Transformations.

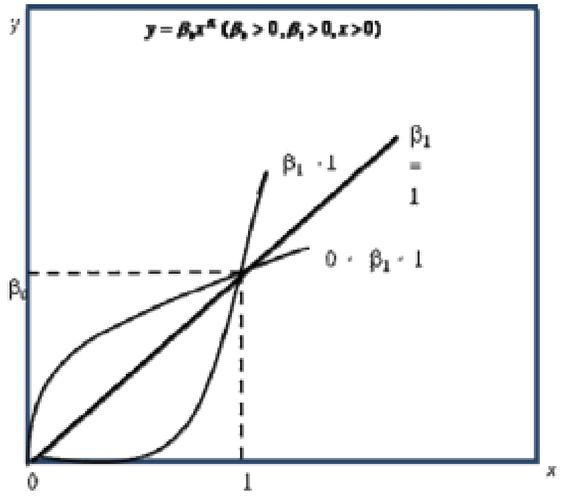
Relation of σ^2 to $E(y)$	Transformation
$\sigma^2 \propto$ constant	$y^* = y$ (no transformation)
$\sigma^2 \propto E(y)$	$y^* = \sqrt{y}$ (Poisson data)
$\sigma^2 \propto E(y)[1-E(y)]$	$y^* = \sin^{-1}(\sqrt{y})$ (Binomial proportion $0 \le y_i \le 1$)
$\sigma^2 \propto [E(y)]^2$	$y^* = \ln(y)$
$\sigma^2 \propto [E(y)]^3$	$y^* = 1/\sqrt{y}$
$\sigma^2 \propto [E(y)]^4$	$y^* = \frac{1}{y}$

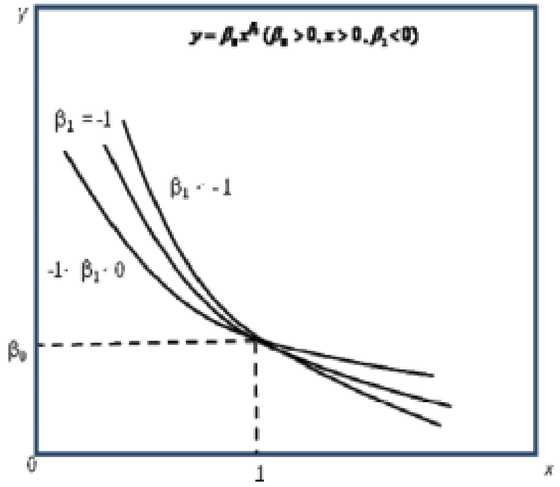
Transformation to Linearize Model

- In some cases, a nonlinear model can be linearized by using a suitable transformation.
- Such nonlinear models are called **intrinsically** or **transformable linear**.
- A**dvantage** of transforming the nonlinear function into the linear function is that the **statistical tools** are developed for the case of a linear regression model.

Transformation to Linearize Model

Example:



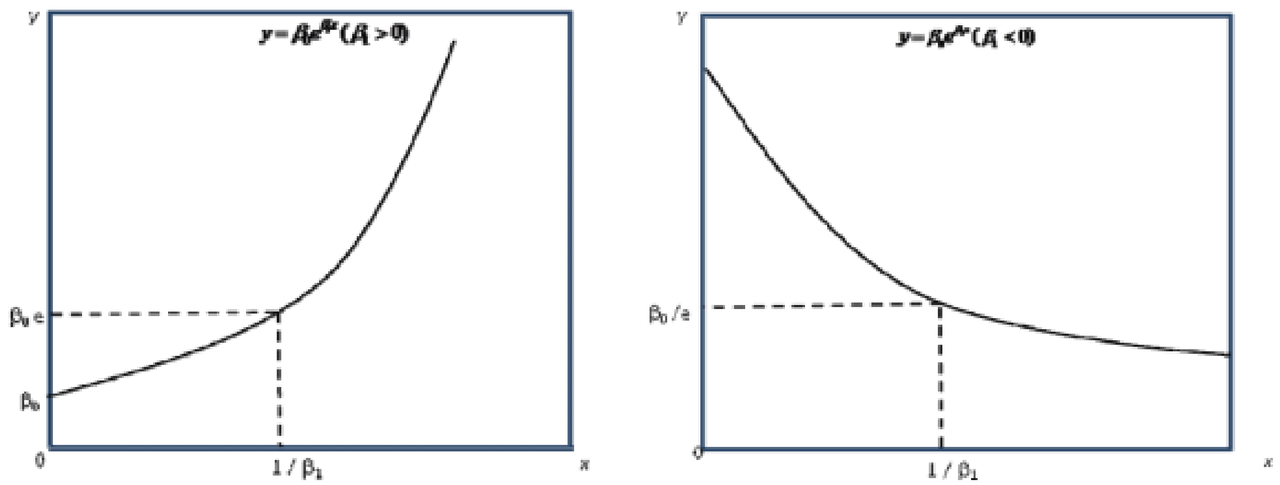


$$y = \beta_0 x^{\beta_i} \quad \Rightarrow \quad \log y = \log \beta_0 + \beta_1 \log x$$

$$y = \ln y$$
$$x = \ln x$$

Transformation to Linearize Model

Example:

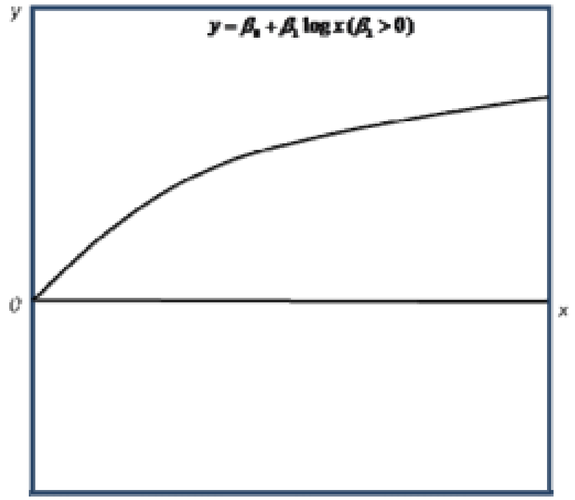


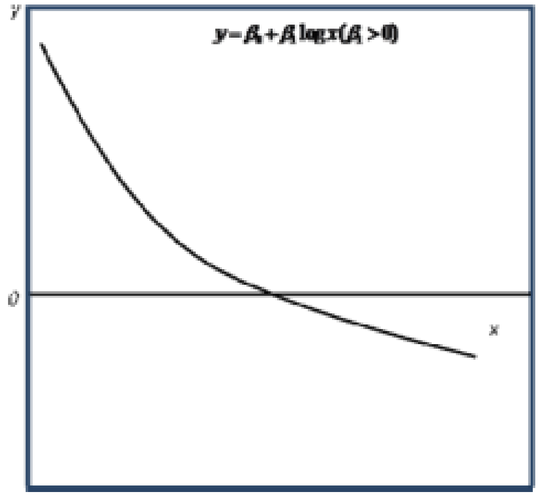
$$y = \beta_0 \exp(\beta_1 x) \quad \Rightarrow \quad \ln y = \ln \beta_0 + \beta_1 x$$

$log_e(ln)$ both sides

Transformation to Linearize Model

Example:



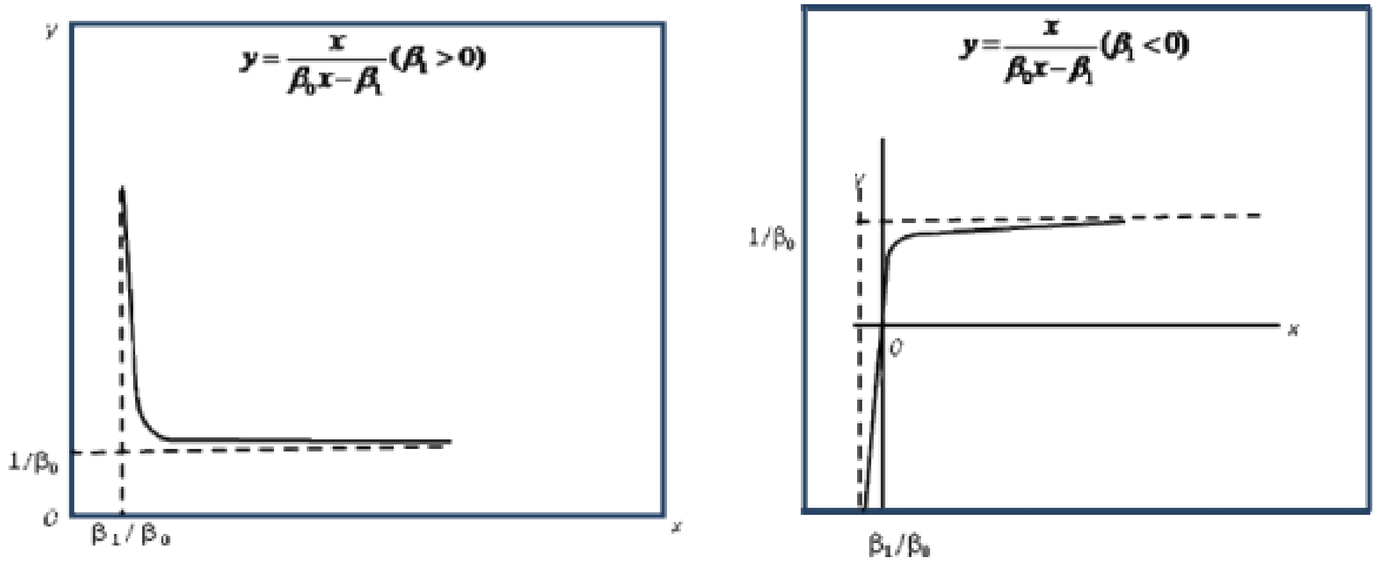


$$y = \beta_0 + \beta_1 \log x \quad \Rightarrow \quad y = \beta_0 + \beta_1 x$$

$$x = \log x$$

Transformation to Linearize Model

Example:



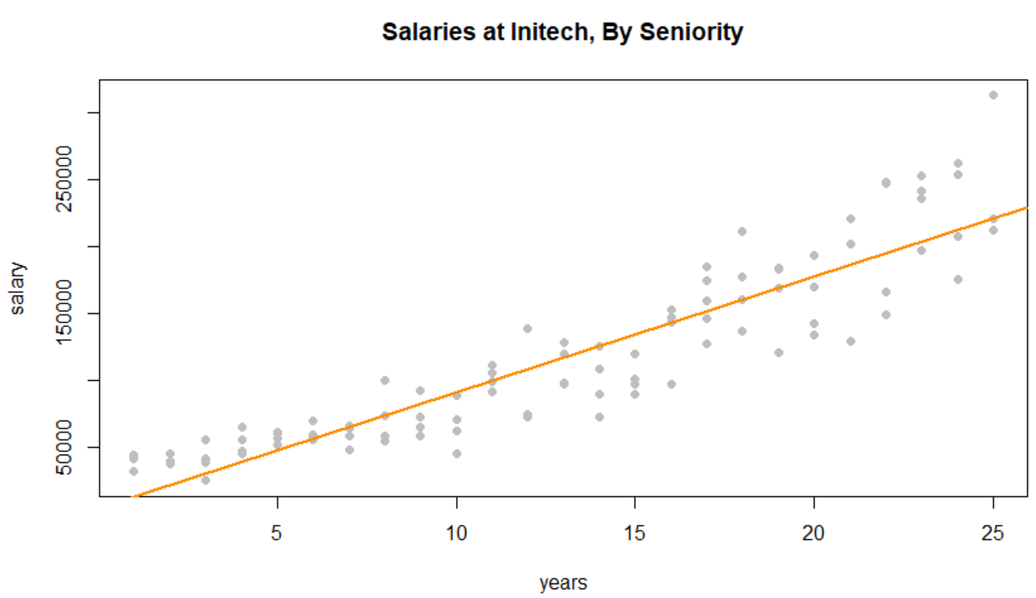
$$y = \frac{x}{\beta_0 x - \beta_1} \quad \Rightarrow \quad y = \beta_0 + \beta_1 x$$

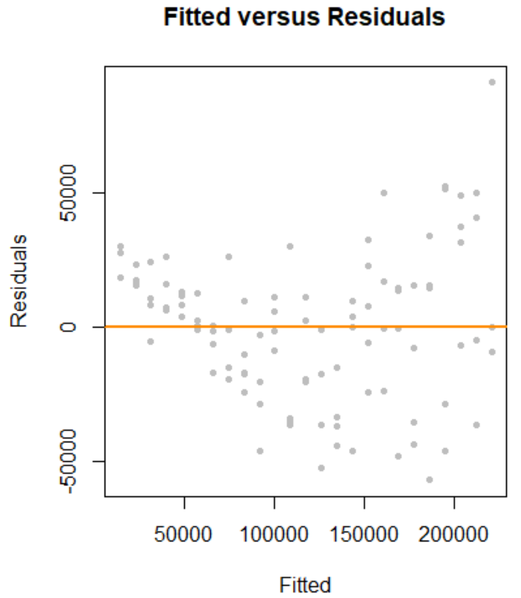
$$x = \frac{1}{x} \text{ and } y = \frac{1}{y}$$

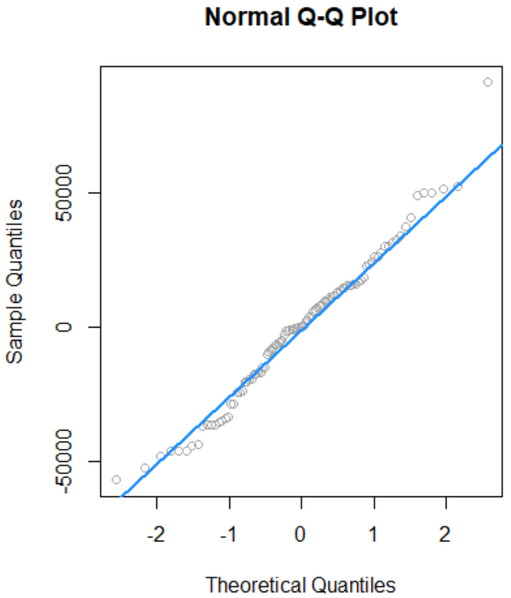
Example 1: Years and Salary
-Response transformation

Recall the fitted value is our estimate of the mean at a particular value of x. Under our usual assumptions,

$$\epsilon \sim N(0, \sigma^2) \quad \Rightarrow \quad \mathrm{Var}[Y|X = x] = \sigma^2$$



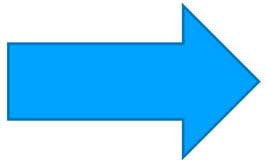




Example 1: Years and Salary

Using transformation

$$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$$



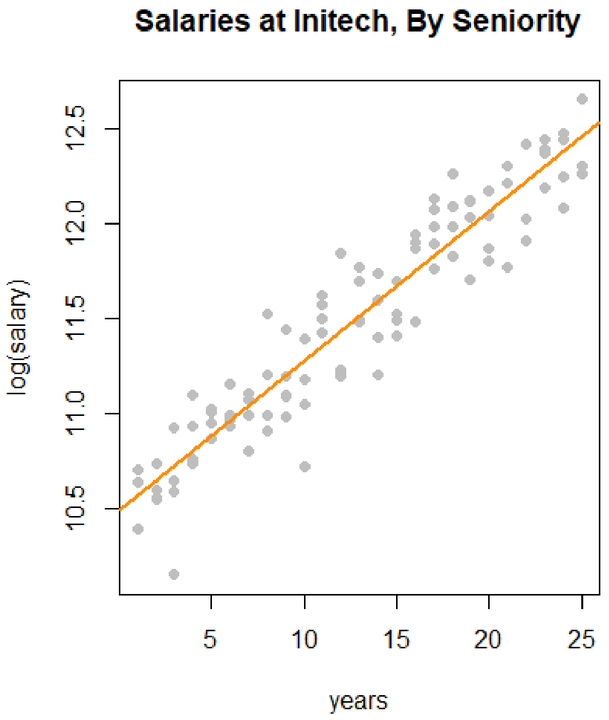
$$\log(Y_i) = \beta_0 + \beta_1 x_i + \epsilon_i.$$

Rescale the model from a log scale back to original scale of the data

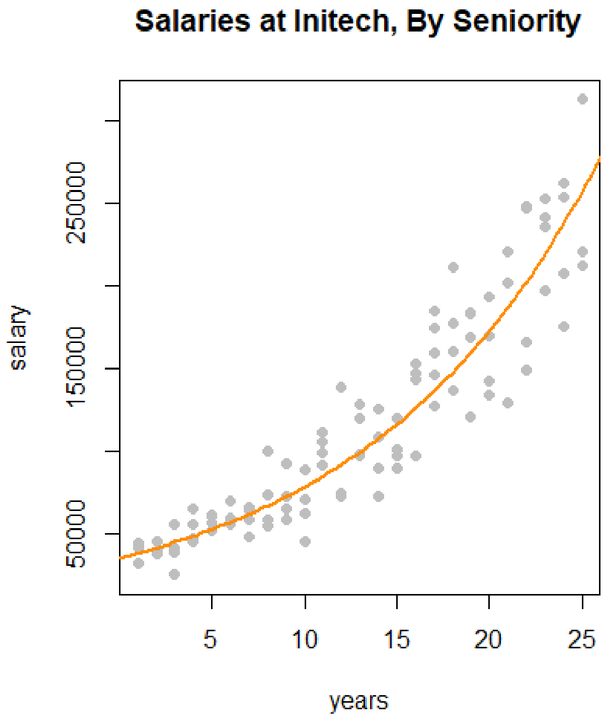
$$Y_i = \exp(\beta_0 + \beta_1 x_i) \cdot \exp(\epsilon_i)$$

```
#new model
initech_fit_log = lm(log(salary) ~ years, data = initech)
summary(initech_fit_log)
```

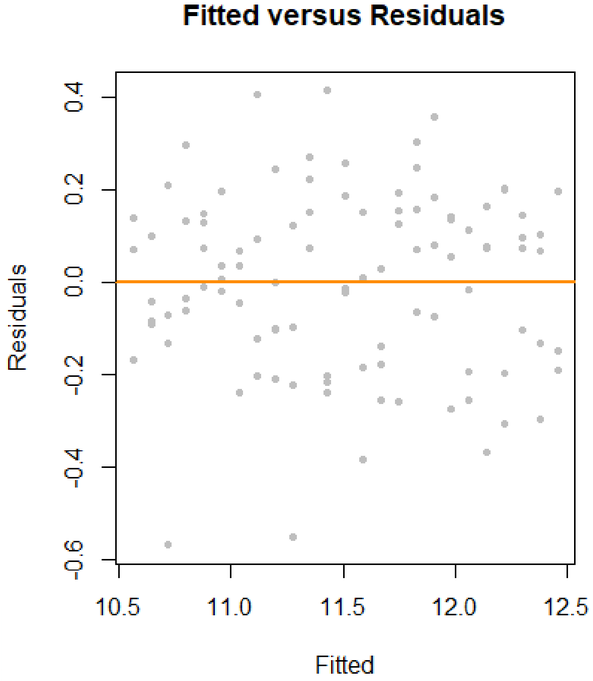
Example 1: Years and Salary

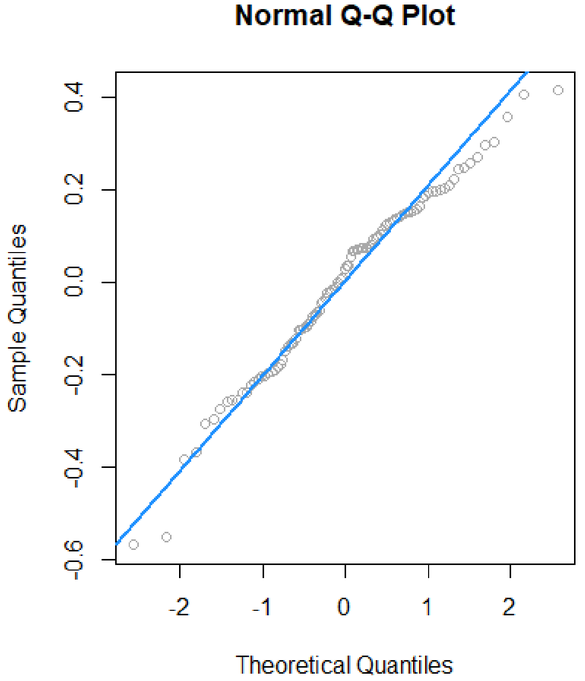


not transform scale



transformed log scale





Example 1: Years and Salary

Not transform scale

$$\log(\hat{y}(x)) = \hat{\beta}_0 + \hat{\beta}_1 x = 10.484 + 0.079x.$$

Transformed log scale

$$\hat{y}(x) = \exp(\hat{\beta}_0)\exp(\hat{\beta}_1 x) = \exp(10.484)\exp(0.079x).$$

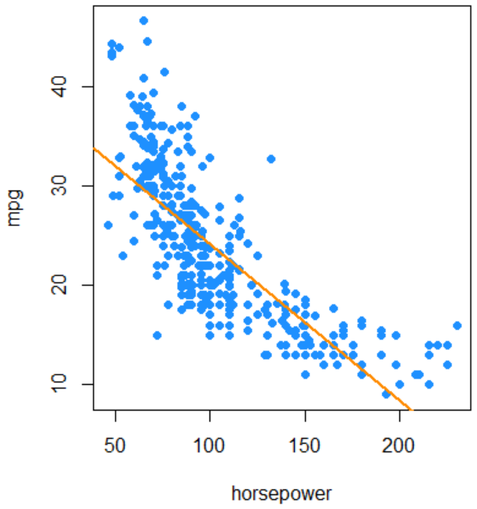
Every or each one additional year of experience, average salary increases $\exp(0.079) = 1.0822$ times (not adding anymore).

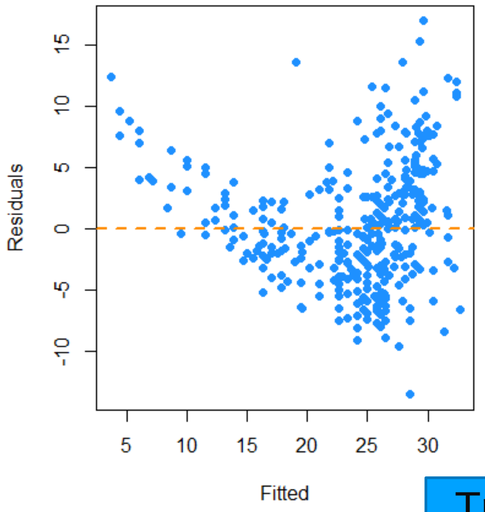
Example 2: City-cycle fuel consumption Data
-Predictor transformation

- "The data concerns city-cycle fuel consumption in miles per gallon, to be predicted in terms of 3 multivalued discrete and 5 continuous attributes." (Quinlan, 1993).

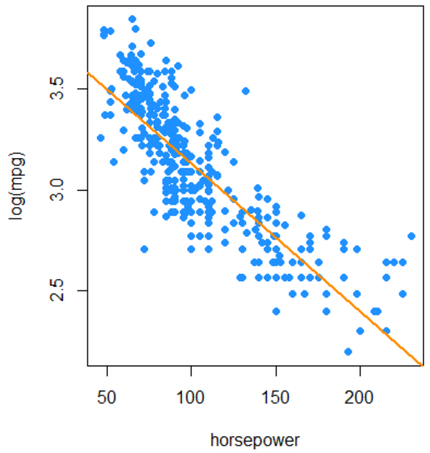
- Attribute Information:
  - mpg: continuous
  - cylinders: multi-valued discrete
  - displacement: continuous
  - horsepower: continuous
  - weight: continuous
  - acceleration: continuous
  - model year: multi-valued discrete
  - origin: multi-valued discrete
  - car name: string (unique for each instance)

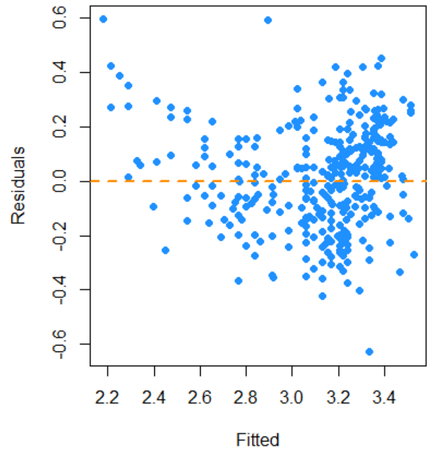
Example 2: City-cycle fuel consumption Data





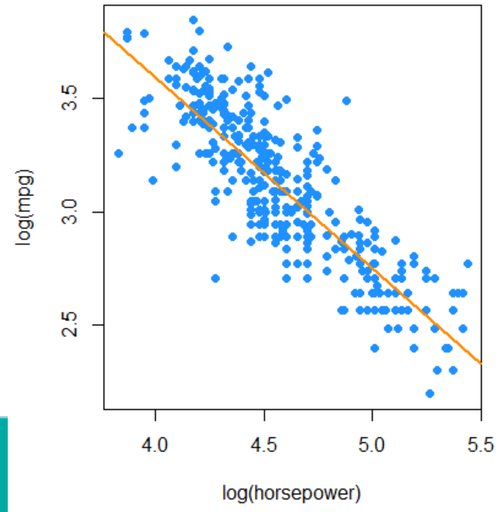
Original Data

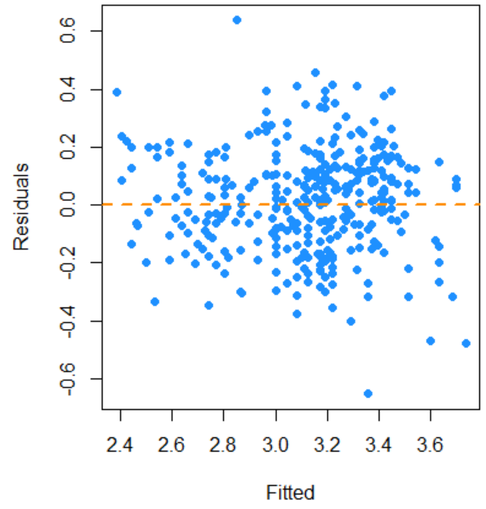




Transformation response and predictor variables

Transformation response variable





Weighted Least Squares

places weights on the observations such that those with small error variance are given more weight since they contain more information compared to observations with larger error variance

#Example for Wighted Least Squares

#define weights to use
wt <- 1 / lm(abs(model$residuals) ~ model$fitted.values)$fitted.values^2

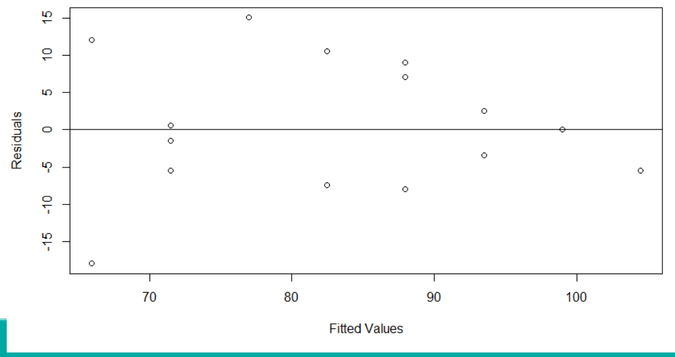
#perform weighted least squares regression
wls_model <- lm(score ~ hours, data = df, weights=wt)

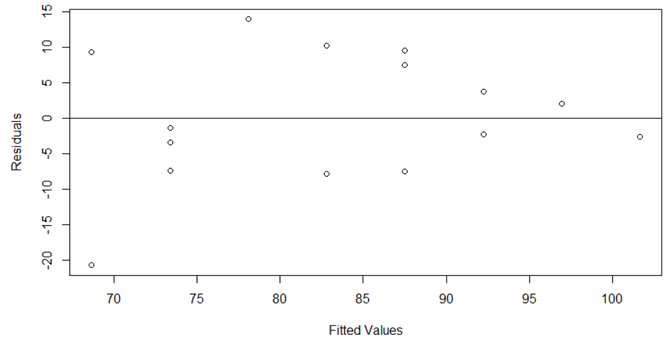
#view summary of model
summary(wls_model)

```
> #view summary of model
> summary(wls_model)

Call:
lm(formula = score ~ hours, data = df, weights = wt)

Weighted Residuals:
    Min      1Q  Median      3Q     Max
-2.0167 -0.9263 -0.2589  0.9873  1.6977

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  63.9689     5.1587  12.400 6.13e-09 ***
hours         4.7091     0.8709   5.407 9.24e-05 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.199 on 14 degrees of freedom
Multiple R-squared:  0.6762,    Adjusted R-squared:  0.6531
F-statistic: 29.24 on 1 and 14 DF,  p-value: 9.236e-05
```

```
> summary(model)

Call:
lm(formula = score ~ hours, data = df)

Residuals:
    Min      1Q  Median      3Q     Max
-17.967  -5.970  -0.719   7.531  15.032

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)   60.467      5.128  11.791 1.17e-08 ***
hours          5.500      1.127   4.879 0.000244 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 9.224 on 14 degrees of freedom
Multiple R-squared:  0.6296,    Adjusted R-squared:  0.6032
F-statistic:  23.8 on 1 and 14 DF,  p-value: 0.0002438

> #create residual vs. fitted plot
> plot(fitted(model), resid(model), xlab='Fitted values', ylab='Residuals')
```

```
> #perform Breusch-Pagan test
> bptest(model)

        studentized Breusch-Pagan test

data:  model
BP = 3.9597, df = 1, p-value = 0.0466
```





Selecting a transformation

-using Box-Cox Method

- The Box-Cox method considers a family of transformations on strictly positive response variables,

$$g_\lambda(y) = \begin{cases} \dfrac{y^\lambda - 1}{\lambda} & \lambda \neq 0 \\ \log(y) & \lambda = 0 \end{cases}$$

- The λ parameter is chosen by numerically maximizing the log-likelihood,

$$L(\lambda) = -\frac{n}{2}\log(RSS_\lambda/n) + (\lambda - 1)\sum \log(y_i).$$

- A $100(1-\alpha)\%$ confidence interval for λ is,

$$\left\{\lambda : L(\lambda) > L(\hat{\lambda}) - \frac{1}{2}\chi^2_{1,\alpha}\right\}$$

Selecting a transformation

-using Box-Cox Method

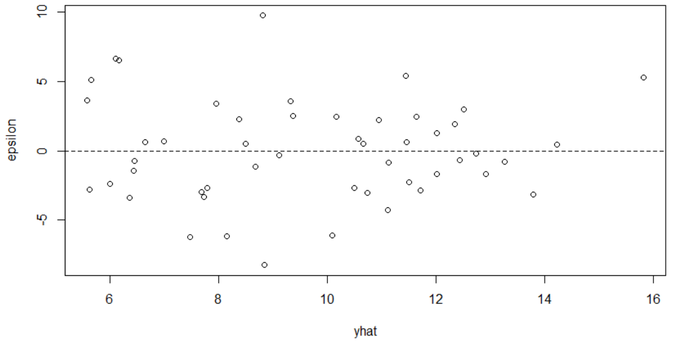
Steps

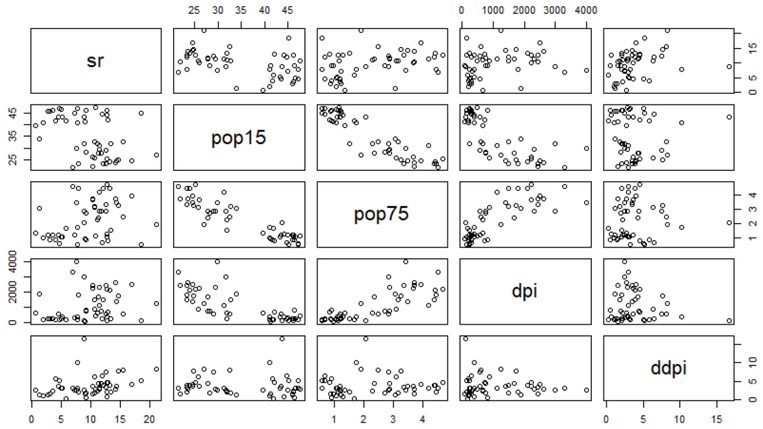
1. Fit the model as usual and check the assumption.
2. Then use the *boxcox()* function to find the best transformation of the form considered by the Box-Cox method.
3. Choose the best λ, which give maximise likelihood.
4. Find the confidence interval of the λ, which give more options.
5. Then, use the λ values to transform the model.
6. Fit the model again which considering the transformation which been choose, and check again the assumption.

Example 1

```
#selecting the best transformation

library(MASS)
library(faraway)

#We fit an additive multiple regression model with sr as the response and
each of the other variables as predictors.
savings_model<-lm(sr ~ ., data = savings)
summary(savings_model)

#We then use the boxcox() function to find the best transformation of the
form considered by the Box-Cox method.
boxcox(savings_model, plotit = TRUE)

#R automatically plots the log-Likelihood as a function of possible λ values.
#It indicates both the value that maximizes the log-likelihood, as well as a
confidence interval for the λ
#value that maximizes the log-likelihood.
boxcox(savings_model, plotit = TRUE, lambda = seq(0.5, 1.5, by = 0.1))

plot(fitted(savings_model), resid(savings_model), col = "dodgerblue",
     pch = 20, cex = 1.5, xlab = "Fitted", ylab = "Residuals")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)

library(lmtest)
#To formally test for heteroscedasticity, we can perform a Breusch-Pagan
test
bptest(savings_model)
#to check the normality
shapiro.test(resid(savings_model))
```

```
> summary(savings_model)

Call:
lm(formula = sr ~ ., data = savings)

Residuals:
    Min      1Q  Median      3Q     Max
-8.2422 -2.6857 -0.2488  2.4280  9.7509

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 28.5660865  7.3545161   3.884 0.000334 ***
pop15       -0.4611931  0.1446422  -3.189 0.002603 **
pop75       -1.6914977  1.0835989  -1.561 0.125530
dpi         -0.0003369  0.0009311  -0.362 0.719173
ddpi         0.4096949  0.1961971   2.088 0.042471 *
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 3.803 on 45 degrees of freedom
Multiple R-squared:  0.3385,    Adjusted R-squared:  0.2797
F-statistic: 5.756 on 4 and 45 DF,  p-value: 0.0007904

>
```



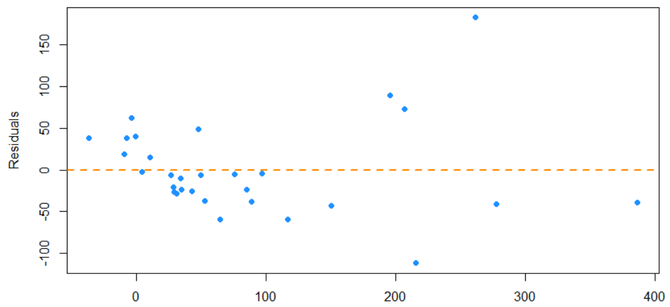


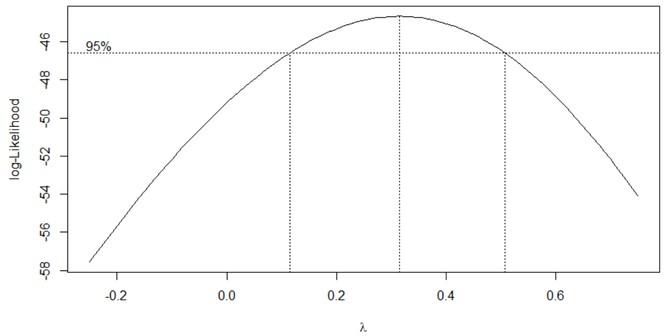
Example 2

```
#Example 2
gala_model<-lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
data = gala)
summary(gala_model)

#check assumption constant variance
plot(fitted(gala_model), resid(gala_model), col = "dodgerblue",
     pch = 20, cex = 1.5, xlab = "Fitted", ylab = "Residuals")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)

library(lmtest)
#To formally test for heteroscedasticity, we can perform a Breusch-Pagan test
bptest(gala_model)
#to check the normality
shapiro.test(resid(gala_model))

#use boc cox to find the best λ values
boxcox(gala_model, lambda = seq(-0.25, 0.75, by = 0.05), plotit = TRUE)

#updated model after transformation/weight
gala_model_cox1<-lm((((Species ^ 0.3) - 1) / 0.3) ~ Area + Elevation + Nearest
+ Scruz + Adjacent, data = gala)
summary(gala_model_cox1)

plot(fitted(gala_model_cox1), resid(gala_model_cox1), col = "dodgerblue",
     pch = 20, cex = 1.5, xlab = "Fitted", ylab = "Residuals")
abline(h = 0, lty = 2, col = "darkorange", lwd = 2)

library(lmtest)
#To formally test for heteroscedasticity, we can perform a Breusch-Pagan test
bptest(gala_model_cox1)
#to check the normality
shapiro.test(resid(gala_model_cox1))
```

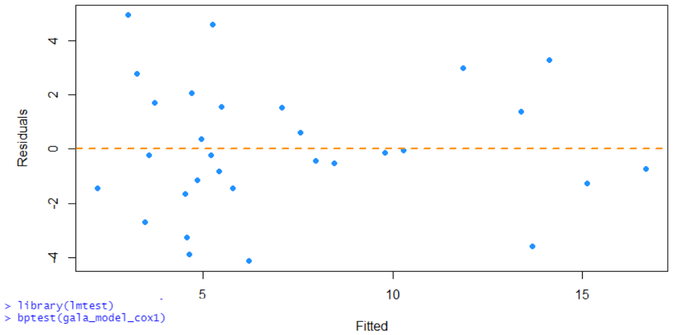
```
> gala_model<-lm(Species ~ Area + Elevation + Nearest + Scruz + Adjacent, data = gala)
> summary(gala_model)

Call:
lm(formula = Species ~ Area + Elevation + Nearest + Scruz + Adjacent,
    data = gala)

Residuals:
     Min       1Q   Median       3Q      Max
-111.679  -34.898   -7.862   33.460  182.584

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  7.068221  19.154198   0.369 0.715351
Area        -0.023938   0.022422  -1.068 0.296318
Elevation    0.319465   0.053663   5.953 3.82e-06 ***
Nearest      0.009144   1.054136   0.009 0.993151
Scruz       -0.240524   0.215402  -1.117 0.275208
Adjacent    -0.074805   0.017700  -4.226 0.000297 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 60.98 on 24 degrees of freedom
Multiple R-squared:  0.7658,    Adjusted R-squared:  0.7171
F-statistic:  15.7 on 5 and 24 DF,  p-value: 6.838e-07
```

```
> library(lmtest)
> bptest(gala_model)

        studentized Breusch-Pagan test

data:  gala_model
BP = 9.7959, df = 5, p-value = 0.08123

> shapiro.test(resid(gala_model))

        Shapiro-Wilk normality test

data:  resid(gala_model)
W = 0.91351, p-value = 0.01826
```






```
> summary(gala_model_cox1)

Call:
lm(formula = (((Species^0.3) - 1)/0.3) ~ Area + Elevation + Nearest +
    Scruz + Adjacent, data = gala)

Residuals:
    Min      1Q  Median      3Q     Max
-4.1301 -1.4007 -0.2357  1.5423  4.9260

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)  3.5618689  0.8144515   4.373 0.000204 ***
Area        -0.0019671  0.0009534  -2.063 0.050074 .
Elevation    0.0142730  0.0022818   6.255 1.83e-06 ***
Nearest      0.0329434  0.0448227   0.735 0.469478
Scruz       -0.0120948  0.0091591  -1.321 0.199114
Adjacent    -0.0027477  0.0007526  -3.651 0.001267 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.593 on 24 degrees of freedom
Multiple R-squared:  0.7457,    Adjusted R-squared:  0.6927
F-statistic: 14.07 on 5 and 24 DF,  p-value: 1.779e-06
```




```
> library(lmtest)
> bptest(gala_model_cox1)

        studentized Breusch-Pagan test

data:  gala_model_cox1
BP = 6.1213, df = 5, p-value = 0.2946

>
> shapiro.test(resid(gala_model_cox1))

        Shapiro-Wilk normality test

data:  resid(gala_model_cox1)
W = 0.9749, p-value = 0.6798
```

Exercise Questions

For each question check the check the assumption of the model, potential outliers, then find the best model.

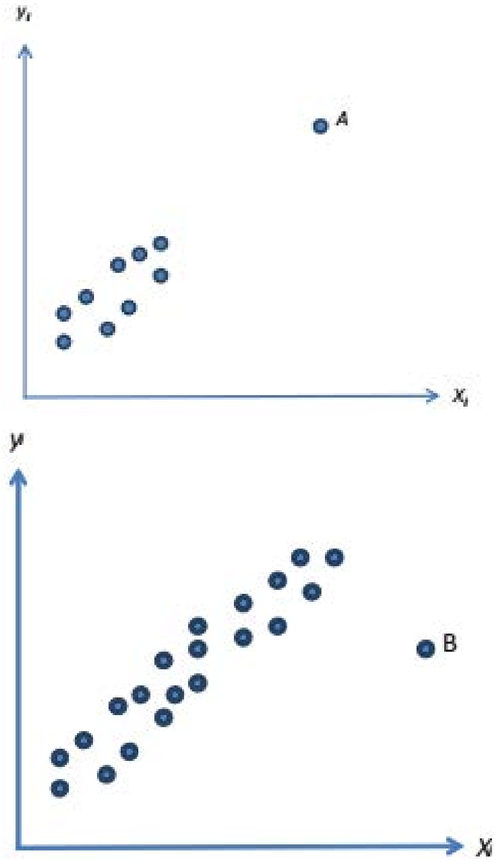
1. A designed experiment is done to assess how moisture content and sweetness of a pastry product affect a taster's rating of the product (*Pastry dataset*). In a designed experiment, the eight possible combinations of four moisture levels and two sweetness levels are studied. Two pastries are prepared and rated for each of the eight combinations, so the total sample size is n = 16. The *y*-variable is the rating of the pastry. The two *x*-variables are moisture and sweetness. The values (and sample sizes) of the *x*-variables were designed so that the *x*-variables were not correlated.

2. The data are from n = 214 females in statistics classes at the University of California at Davis (*Stat Females dataset*). The variables are y = student's self-reported height, x1 = student's guess at her mother's height, and x2 = student's guess at her father's height. All heights are in inches.

3. Data from n = 113 hospitals in the United States are used to assess factors related to the likelihood that a hospital patients acquires an infection while hospitalized. The variables here are y = infection risk, x1 = average length of patient stay, x2 = average patient age, x3= measure of how many x-rays are given in the hospital (*Hospital Infection dataset*).

2.4 Diagnostic Leverage and Influence

- Leverage
  - ➢Has an unusual x-value and may control certain model properties.
  - ➢This point does not affect the coefficient of the model, but certainly will have a dramatic effect on the model summary statistics such as R^2, and standard errors of the coefficient.
- Influence
  - ➢A moderately unusual x -coordinate and the y -value is also unusual.
  - ➢It has a noticeable impact on the model coefficients, and it pulls the regression model in its direction.



Leverage

- The location of points in x -space affects the model properties like parameter estimates, standard errors, predicted values, summary statistics etc.
- The hat matrix $H = X(X'X)^{-1}X'$ plays an important role in identifying influential observations.
- Since

$$V(\hat{y}) = \sigma^2 H$$

$$V(e) = \sigma^2 (I - H),$$

- The i^{th} diagonal element of H is $h_{ii} = x_i'(X'X)^{-1}x_i$ where x_i' is the i^{th} row of X –matrix.
- The hat matrix diagonal is a standardized measure of the distance of i^{th} an observation from the centre (or centroid) of the x –space.
- Thus, **large hat diagonals** reveal observations that are **potentially influential** because they are remote in x –space from the rest of the sample.

Leverage

- Average size of hat diagonal $(\bar{\bar{h}})$

$$\bar{\bar{h}} = \frac{\sum h_{ii}}{n} = \frac{rank(H)}{n}$$
$$= \frac{rank(X)}{n}$$
$$= \frac{tr(H)}{n} = \frac{k}{n}$$

- If $h_{ii} > 2\bar{\bar{h}} = \frac{2k}{n} \Rightarrow$ the point is remote enough from rest of the data to be considered as a leverage point.
- Care is needed in using cutoff value $\frac{2k}{n}$ and magnitudes of k and n are to be assessed. There can be situations where $\frac{2k}{n} > 1$ and then this cut off does not apply.

Measure of Influence

Cook's D-statistics

- is a measure of the distance between the least squares estimate based on all *n* observations in *b* and the estimate obtained by deleting the i^{th} point, say $b_{(i)}$.
- Points with large D_i -the points have considerable influence of OLSE *b* .

It is given by

$$D_i(M,C)=\frac{(b_{(i)}-b)'M(b_{(i)}-b)}{C};\quad i=1,2,...,n.$$

The usual choice of M and C are

$$M=X'X$$
$$C=kMS_{res}$$

So

$$D_i(X'X,kMS_{res})=\frac{(b_{(i)}-b)'X'X(b_{(i)}-b)}{kMS_{res}};\quad i=1,2,...,n$$
$$=\frac{(\hat{y}-\hat{y}_{(i)})'(\hat{y}-\hat{y}_{(i)})}{kMS_{res}}$$

This displacement is large and indicates that the OLSE is sensitive to the i^{th} data point.

- Since $F_{0.5}(k,n-k)\approx 1$, we usually consider that points for which $D_i>1$ to be influential.
- Ideally, each $b_{(i)}$ is expected to stay within the boundary of a 10-20% confidence region.
- D_i is not an F-statistic but cut off of 1 work very well in practice.

where

$$\hat{y}=Xb$$
$$\hat{y}_{(i)}=Xb_{(i)}$$
$$b=(X'X)^{-1}X'y.$$

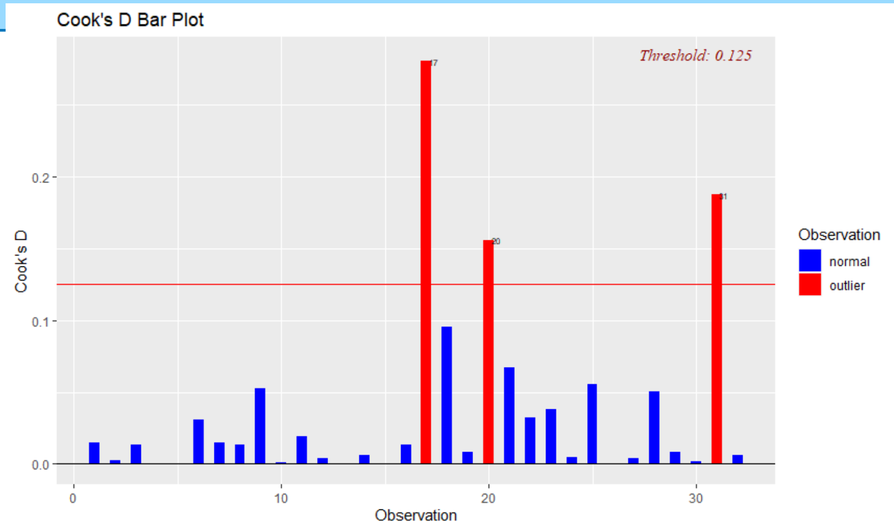
Cook’s D-statistics

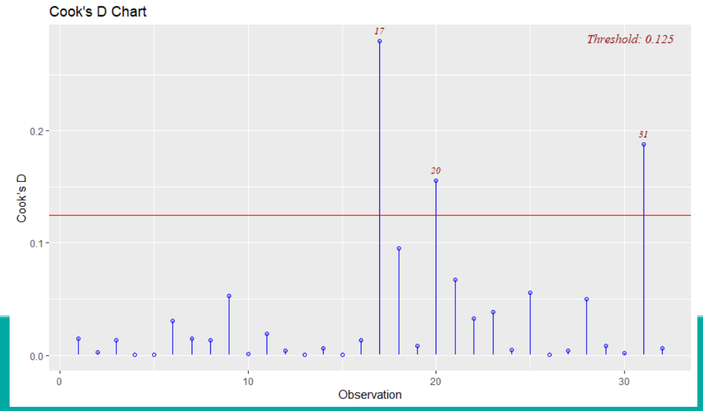
```
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_plot_cooksd_bar(model)

model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_plot_cooksd_chart(model)
```

Steps to compute Cook’s distance:

- delete observations one at a time.
- refit the regression model on remaining (n−1) observations
- examine how much all of the fitted values change when the ith observation is deleted.

A data point having a large cook’s d indicates that the data point strongly influences the fitted values.





Measure of Influence

DFFITS and DFBETAS

DFBETAS which indicates how much the regression coefficient changes if the i^{th} observation were deleted. Such change is measured in terms of standard deviation units. This statistic is

$$DFBETAS_{j,i} = \frac{b_j - b_{j(i)}}{\sqrt{S^2_{(i)} C_{jj}}}$$

where C_{jj} is the j^{th} diagonal element of $(X'X)^{-1}$

$b_{j(i)}$ regression coefficient computed without the use of i^{th} observation.

- Large (in magnitude) value of $DFBETAS_{j,i}$, indicates that i^{th} observation has considerable influence on the j^{th} regression coefficient.

- If $|DFBETAS|_{j,i} > \frac{2}{\sqrt{n}}$, then i^{th} observation warrants examination.

Measure of Influence

DFFITS and DFBETAS

DFFITS: The deletion influence of i^{th} observation on the predicted or fitted value can be investigated by using diagnostic by Belsley, Kuh and Welsch as

$$DFFITS_i = \frac{\hat{y}_i - \hat{y}_{(i)}}{\sqrt{S^2_{(i)} h_{ii}}},\ i = 1, 2, \ldots, n$$

where $\hat{y}_{(i)}$ is the fitted value of y_i obtained without the use of the i^{th} observation.

The denominator is just a standardization, since $Var(\hat{y}_i) = \sigma^2 h_{ii}$.

- DFFITS is the number of standard deviations that the fitted value $\hat{y}$ changes of i^{th} observation is removed.
- If $|DFBETAS|_{j,i} > \frac{2}{\sqrt{n}}$, then i^{th} observation warrants examination.

Measure of Influence

DFFITS and DFBETAS

Computationally,

$$DFFITS_i = \sqrt{\frac{h_{ii}}{1-h_{ii}}}\frac{e_i}{S_{(i)}\sqrt{1-h_{ii}}}$$

$$= t_i\sqrt{\frac{h_{ii}}{1-h_{ii}}}$$

$$= R-\text{student} \times \text{leverage of } i^{th} \text{ observation}$$

where t_i is R-student.

- If the data point is an outlier, then R-student will be large is magnitude.
- If the data point has high leverage, then h_{ii} will be close to unity.
- In either of these cases, $DFFITS_i$ can be large.
- If $h_{ii} \approx 0$, then the effect of R-student will be moderated.
- If R-student is near to zero, then combined with high leverage point, then $DFFITS_i$ can be a small value.
- Thus $DFFITS_i$ is affected by both leverage and prediction error. Belsley, Kuh and Welsch suggest that any observation for which

$$|DFFITS_i| > 2\sqrt{\frac{k}{n}}$$

warrants attention.

Note: The cutoff values of $DFFITS_{1,i}$ and $DFFITS_i$ are only guidelines. It is very difficult to provide cutoffs that are correct for all cases. So analyst is recommended to utilize information about both what is diagnostic means and the application environment in selecting a cutoff.

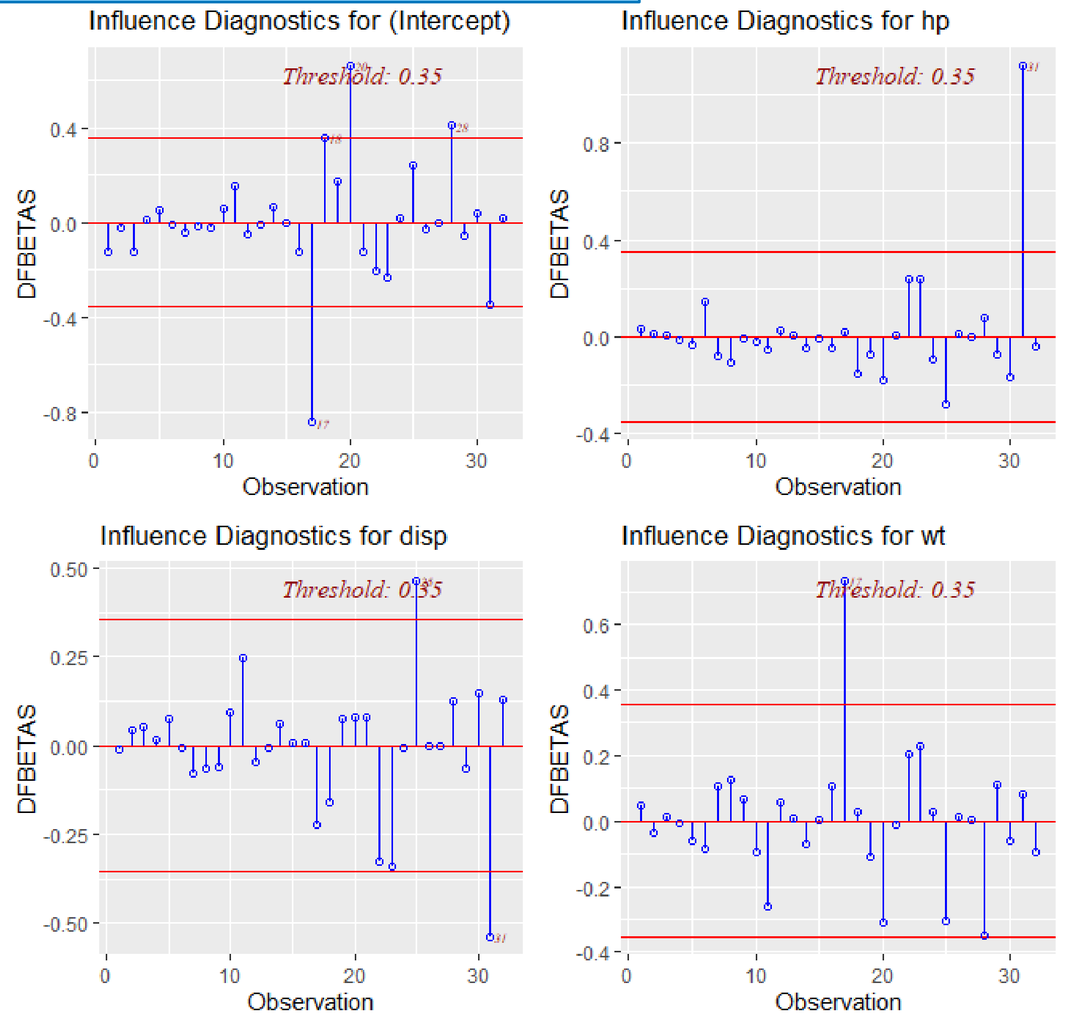
DFBETAS

```
#DFBETAs Panel
model <- lm(mpg ~ disp + hp + wt, data = mtcars)
ols_plot_dfbetas(model)
```

DFBETA measures the difference in each parameter estimate with and without the influential point. There is a DFBETA for each data point i.e if there are n observations and k variables, there will be n∗k DFBETAs.

In general, **large values** of DFBETAS indicate observations that are **influential** in estimating a given parameter.

Belsley, Kuh, and Welsch recommend 2 as a general cutoff value to indicate influential observations and $|DFBETAS|_{j,i} > \frac{2}{\sqrt{n}}$ as a size-adjusted cutoff.



DFFITS

Proposed by Welsch and Kuh (1977). It is the scaled difference between the ith fitted value obtained from the full data and the ith fitted value obtained by deleting the ith observation.

DFFIT - difference in fits, is used to identify influential data points. It quantifies the number of standard deviations that the fitted value changes when the ith data point is omitted.

Steps to compute DFFITs:

- delete observations one at a time.
- refit the regression model on remaining observations
- examine how much all of the fitted values change when the ith observation is deleted.

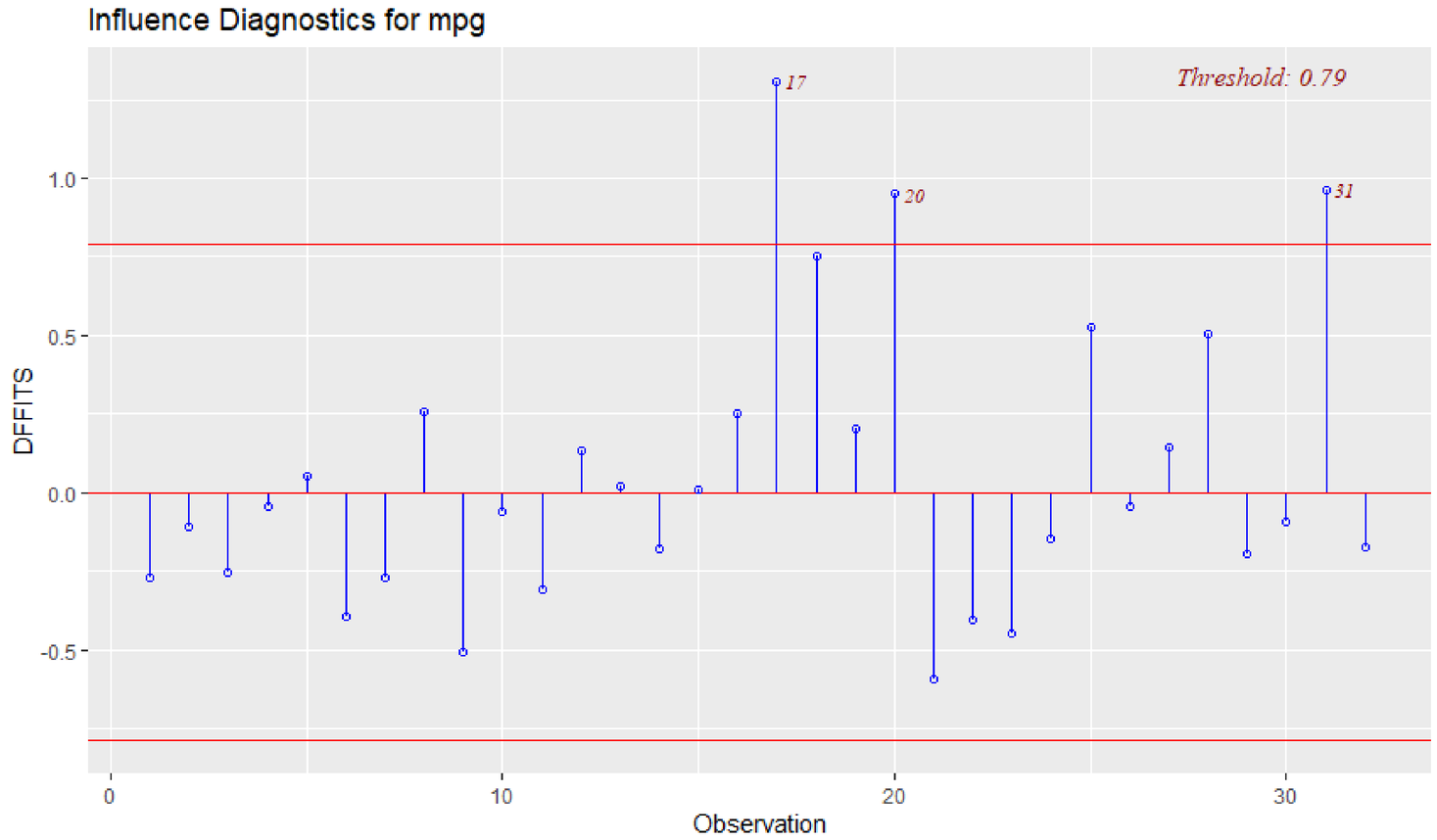
An observation is deemed influential if the absolute value of its DFFITS value is greater than:

$$2 * \frac{\sqrt{(p+1)}}{(n-p-1)}$$

where n is the number of observation and p is the number of predictors including intercept

DFFITS

```
#DFFITS Plot
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_plot_dffits(model)
```



Studentized Residual Plot

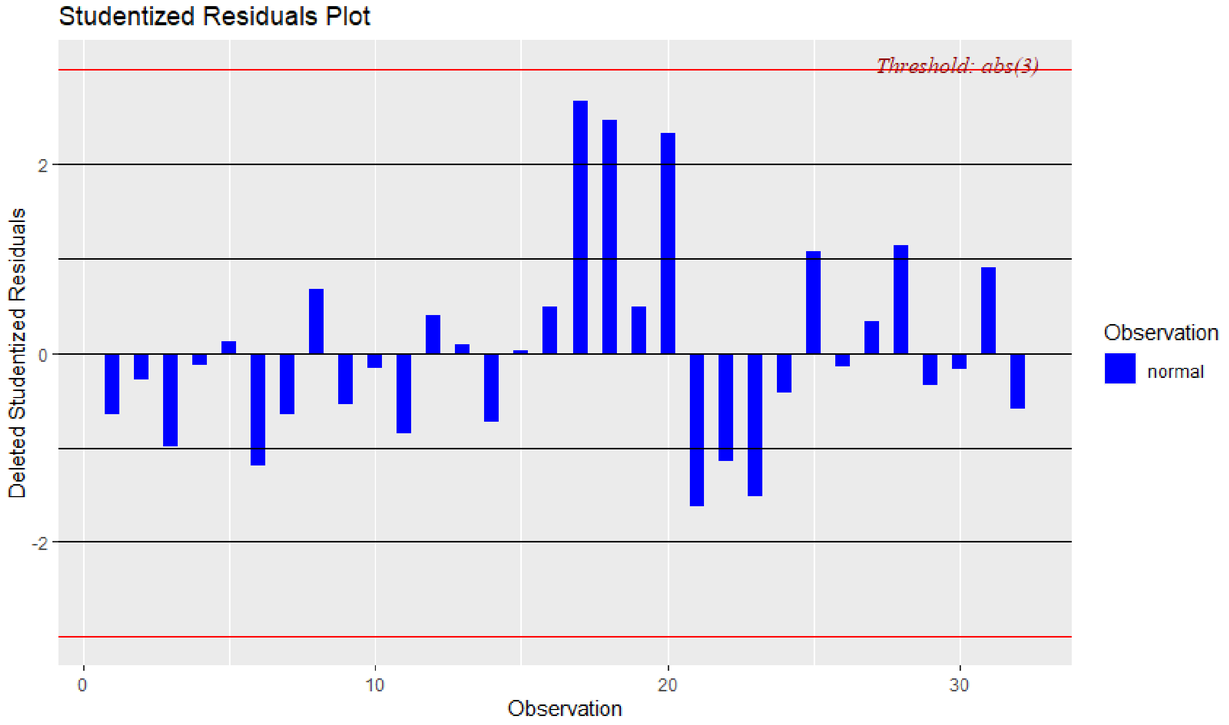
Plot for detecting outliers.

Studentized deleted residuals (or externally studentized residuals) is the deleted residual divided by its estimated standard deviation.

Studentized residuals are going to be more effective for detecting outlying Y observations than standardized residuals.

If an observation has an externally studentized residual that is larger than 3 (in absolute value) we can call it an outlier.

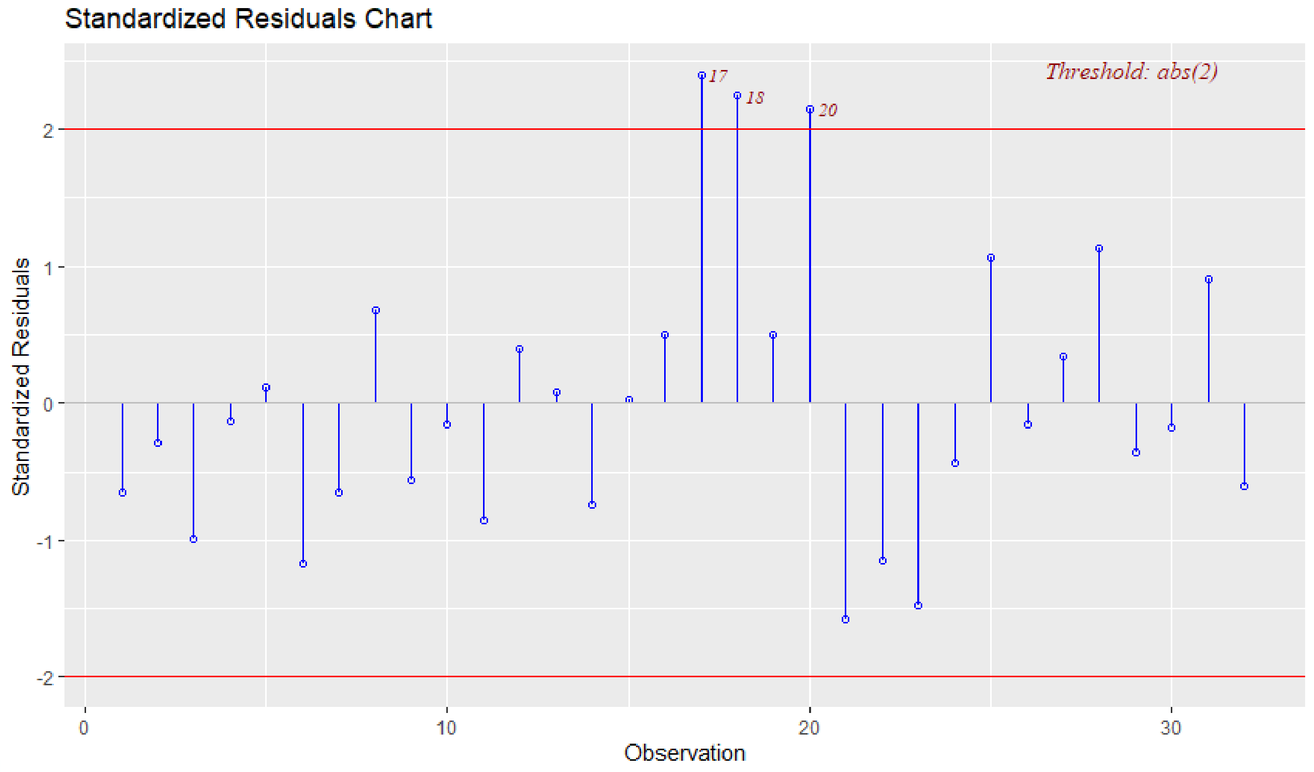
```
#Studentized Residual Plot
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_plot_resid_stud(model)
```



Standardized Residual Chart

- Chart for detecting outliers.
- Standardized residual (internally studentized) is the residual divided by estimated standard deviation.

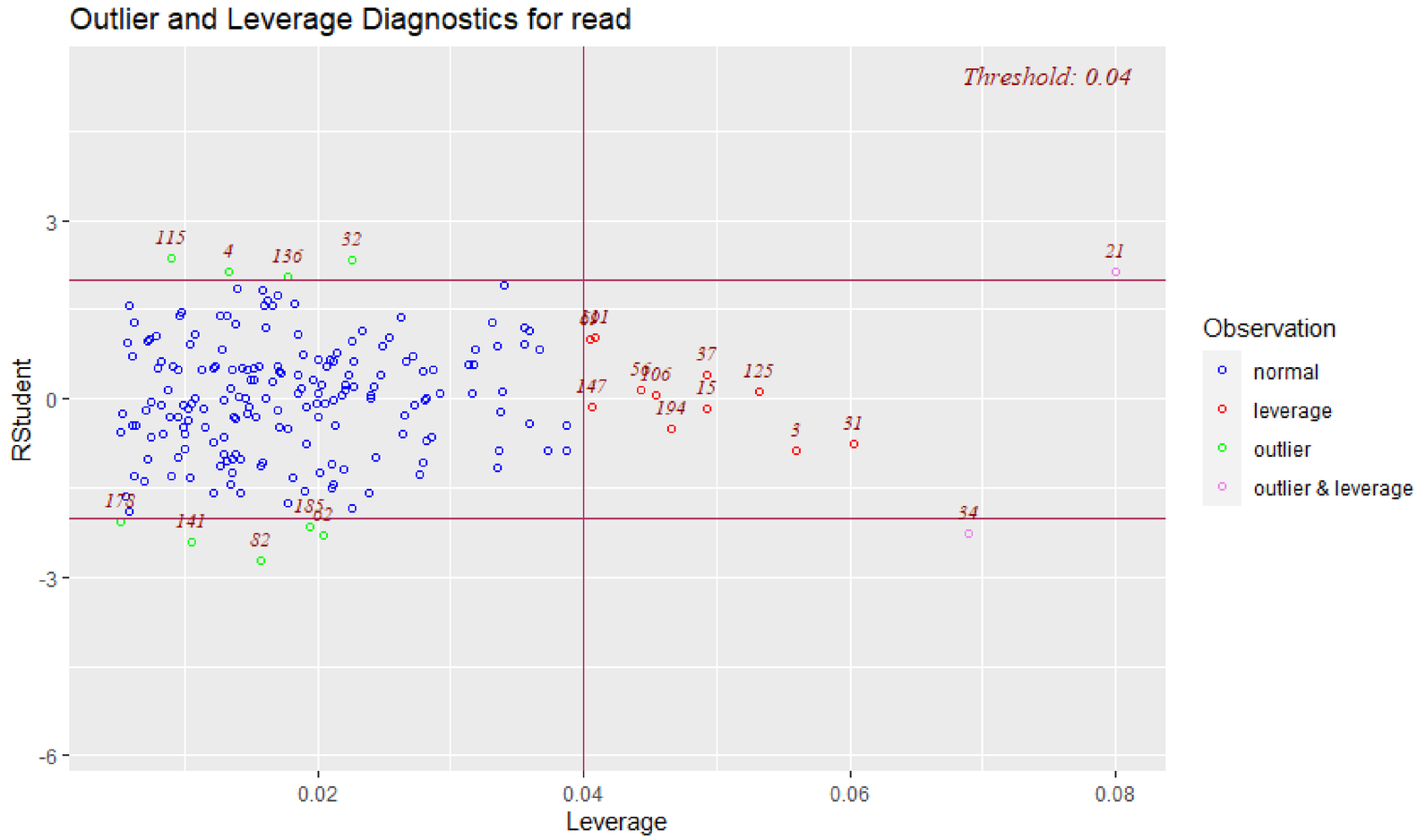
```
#Standardized Residual Chart
model <- lm(mpg ~ disp + hp + wt + qsec, data = mtcars)
ols_plot_resid_stand(model)
```



Studentized Residuals vs Leverage Plot

-Graph for detecting influential observations.

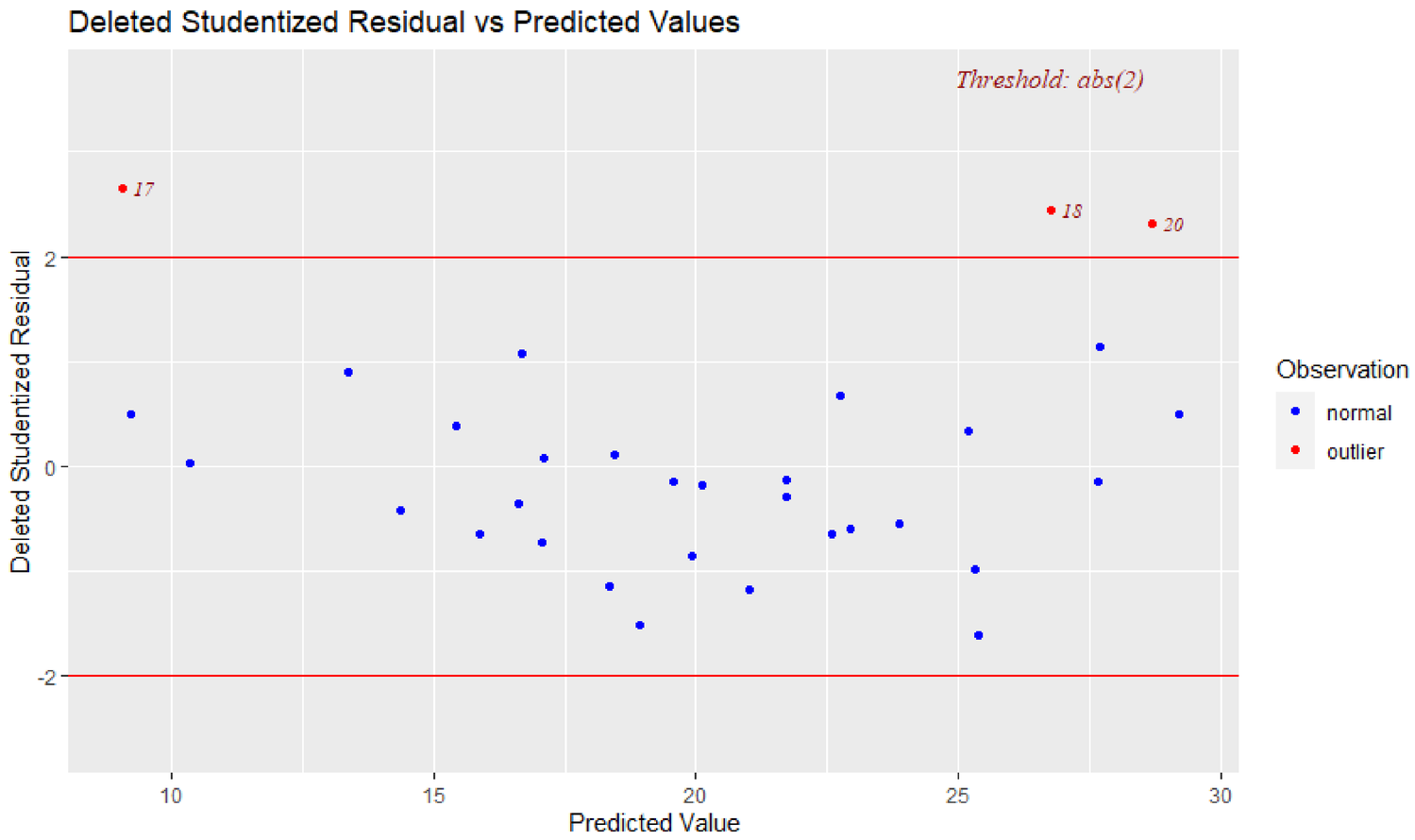
```
#Studentized Residuals vs Leverage Plot
model <- lm(read ~ write + math + science, data = hsb)
ols_plot_resid_lev(model)
```



Deleted Studentized Residual vs Fitted Values Plot

-Graph for detecting influential observations.

```
#Deleted Studentized
Residual vs Fitted Values Plot
model <- lm(mpg ~ disp + hp
+ wt + qsec, data = mtcars)
ols_plot_resid_stud_fit(model)
```



2.5 Multicollinearity

- Multicollinearity occurs when independent variables in a model are correlated.
- This correlation is a problem because independent variables should be independent.
- If the degree of correlation between variables is high enough, it can cause problems when you fit the model and interpret the results.
- **Source** of multicollinearity:
  - The data collection method employed
  - Constraint on the model or in the population
  - Model specification
  - An over defined model (*parsimony model)

2.5 Multicollinearity

Why is Multicollinearity a Potential Problem?

- The interpretation of a regression coefficient is that it represents the mean change in the dependent variable for each 1 unit change in an independent variable when *you hold all of the other independent variables constant*.
- The idea is that you can change the value of one independent variable and not the others.
- However, when independent variables are correlated, it indicates that changes in one variable are associated with shifts in another variable.
- The stronger the correlation, the more difficult it is to change one variable without changing another.
- It becomes difficult for the model to estimate the relationship between each independent variable and the dependent variable independently because the independent variables tend to change in unison.

2.5 Multicollinearity

Effects

Multicollinearity causes the following two basic types of problems:

- The **coefficient estimates** can swing wildly based on which other independent variables are in the model. The coefficients become **very sensitive** to small changes in the model.

- Multicollinearity **reduces the precision of the estimated coefficients**, which weakens the statistical power of your regression model. You might not be able to trust the p-values to identify independent variables that are statistically significant.

2.5 Multicollinearity

Do I Have to Fix Multicollinearity?

Multicollinearity makes it hard to interpret your coefficients, and it reduces the power of your model to identify independent variables that are statistically significant. These are definitely serious problems.

The need to reduce multicollinearity depends on **its severity** and your **primary goal** for your regression model. Keep the following three points in mind:

- The severity of the problems increases with the degree of the multicollinearity. Therefore, if you have only moderate multicollinearity, you may not need to resolve it.
- Multicollinearity affects only the specific independent variables that are correlated. Therefore, if multicollinearity is not present for the independent variables that you are particularly interested in, you may not need to resolve it. (*Suppose your model contains the experimental variables of interest and some control variables. If high multicollinearity exists for the control variables but not the experimental variables, then you can interpret the experimental variables without problems.)*
- Multicollinearity affects the coefficients and p-values, but it does not influence the predictions, precision of the predictions, and the goodness-of-fit statistics. If your primary goal is to make predictions, and you don't need to understand the role of each independent variable, you don't need to reduce severe multicollinearity.

2.5 Multicollinearity

Multicollinearity Diagnostic

- There are many ways to detect the multicollinearity such as scatter plot, correlation of coefficient, and Variance Inflation Factor (VIF).
- Variance Inflation Factor (VIF) is defined as $\frac{1}{1-R_j^2}$

VIFs=1

- There is no correlation between this independent variable and any others

1<=VIFs <= 5

- There is a moderate correlation, but it is not severe enough to warrant corrective measures.

VIFs>5

- Represent critical levels of multicollinearity where the coefficients are poorly estimated, and the p-values are questionable

Example

The *seatpos* dataset from the *faraway* package.

The **predictors** in this dataset are various attributes of car drivers, such as their height, weight and age. The **response** variable hipcenter measures the "horizontal distance of the midpoint of the hips from a fixed location in the car in mm."

Essentially, it measures the position of the seat for a given driver. This is potentially useful information for car manufacturers considering comfort and safety when designing vehicles.

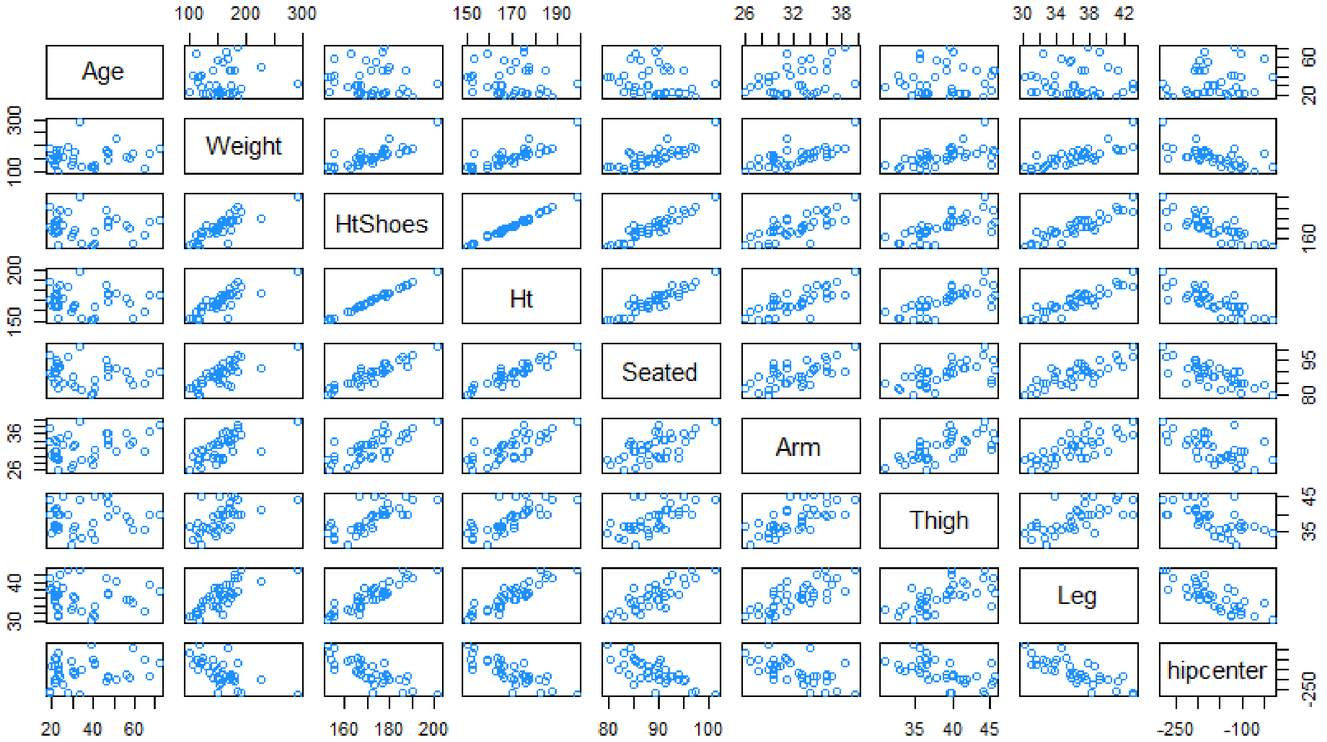
Attempt to fit a model that predicts hipcenter.

- Two predictor variables are immediately interesting: HtShoes and Ht.
- Certainly, expect a person's height to be highly correlated to their height when wearing shoes.

Example

```
#Multicolinearity
library(faraway)
pairs(seatpos, col = "dodgerblue")
round(cor(seatpos), 2)
hip_model = lm(hipcenter ~ ., data = seatpos)
summary(hip_model)
```

```
> summary(hip_model)

Call:
lm(formula = hipcenter ~ ., data = seatpos)

Residuals:
    Min      1Q  Median      3Q     Max
-73.827 -22.833  -3.678  25.017  62.337

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 436.43213  166.57162   2.620   0.0138 *
Age           0.77572    0.57033   1.360   0.1843
Weight        0.02631    0.33097   0.080   0.9372
HtShoes      -2.69241    9.75304  -0.276   0.7845
Ht            0.60134   10.12987   0.059   0.9531
Seated        0.53375    3.76189   0.142   0.8882
Arm          -1.32807    3.90020  -0.341   0.7359
Thigh        -1.14312    2.66002  -0.430   0.6706
Leg          -6.43905    4.71386  -1.366   0.1824
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 37.72 on 29 degrees of freedom
Multiple R-squared:  0.6866,    Adjusted R-squared:  0.6001
F-statistic:  7.94 on 8 and 29 DF,  p-value: 1.306e-05
```




```
> round(cor(seatpos), 2)
           Age Weight HtShoes    Ht Seated   Arm Thigh   Leg hipcenter
Age       1.00   0.08   -0.08 -0.09  -0.17  0.36  0.09 -0.04       0.21
Weight    0.08   1.00    0.83  0.83   0.78  0.70  0.57  0.78      -0.64
HtShoes  -0.08   0.83    1.00  1.00   0.93  0.75  0.72  0.91      -0.80
Ht       -0.09   0.83    1.00  1.00   0.93  0.75  0.73  0.91      -0.80
Seated   -0.17   0.78    0.93  0.93   1.00  0.63  0.61  0.81      -0.73
Arm       0.36   0.70    0.75  0.75   0.63  1.00  0.67  0.75      -0.59
Thigh     0.09   0.57    0.72  0.73   0.61  0.67  1.00  0.65      -0.59
Leg      -0.04   0.78    0.91  0.91   0.81  0.75  0.65  1.00      -0.79
hipcenter 0.21  -0.64   -0.80 -0.80  -0.73 -0.59 -0.59 -0.79       1.00
>
```

Example

```
#Multicolinearity
#VIF
vif(hip_model)
```

```
> #VIF
> vif(hip_model)
      Age    Weight   HtShoes        Ht    Seated       Arm     Thigh       Leg
 1.997931  3.647030 307.429378 333.137832  8.951054  4.496368  2.762886  6.694291
> |
```

2.5 Multicollinearity

Methods for dealing with multicollinearity

- Center the Independent Variables to Reduce Structural Multicollinearity

  - Centering the variables is also known as standardizing the variables by subtracting the mean. This process involves calculating the mean for each continuous independent variable and then subtracting the mean from all observed values of that variable. Then, use these centered variables in your model.

  - The advantage of just subtracting the mean is that the interpretation of the coefficients remains the same.

2.5 Multicollinearity

Methods for dealing with multicollinearity

The potential solutions include the following:

- Remove some of the highly correlated independent variables.
- Linearly combine the independent variables, such as adding them together.
- Perform an analysis designed for highly correlated variables, such as principal components analysis or partial least squares regression.
- LASSO and Ridge regression are advanced forms of regression analysis that can handle multicollinearity (*If you know how to perform linear least squares regression, you'll be able to handle these analyses with just a little additional study*)

Summary

- ✓The best model should be free from all potentials problems which been discussed in this chapter.
- ✓All potential ways to solve the problems occurs should be considered and taken, to improve the models.
- ✓Then the model could be used to predict better and used to simulate data based on the model.


Thank You!