Chapter 11: Model Diagnostics

Wang Shujia

Contents

1 Assumptions Checking

Regression Model Assumptions

1. *Inference and prediction rely on our assumptions being "true"*
   - Otherwise, estimates are wrong!
   - Predictions are systematically biased
   - Confidence intervals don't have the right coverage
2. *The following assumptions to be checking*:
   - Normality
   - Linearity
   - Homoscedasticity
   - Independence
   - No outliers, high leverage points and influential observations in your dataset

1.1 A Typical Approach: plot(model)

Basic Tool for Diagnaostics: Residuals

Three types of residuals:

1. **Residuals**: are defined as the deviation of each point from the fitted regression line
$$e_i = Y_i - \hat{Y}_i$$
2. **Standardised Residuals**:
$$r_i = \frac{e_i}{s\sqrt{1-h_i}}$$
   where h_i is the ith diagonal element of the hat matrix H, s is the sample standard deviation.
3. **Studentised Residuals** (also called "jackknifed residuals"):
$$r_i = \frac{e_i}{s_{(-i)}\sqrt{1-h_i}}$$
   where $s_{(-i)}$ is the sample standard deviation without the i^{th} observation.

How to Check the Regression Model Assumptions?

A Typical Approach: check the 4 in 1 residual plot

```
model <- lm(y~x)
plot(model)
```

Example 1: Anscombe dataset

```
> ansc<-read.csv("R/anscombe.csv", header=TRUE); ansc

##    x1    y1 x2   y2 x3    y3 x4    y4
## 1  10  8.04 10 9.14 10  7.46  8  6.58
## 2   8  6.95  8 8.14  8  6.77  8  5.76
## 3  13  7.58 13 8.74 13 12.74  8  7.71
## 4   9  8.81  9 8.77  9  7.11  8  8.84
## 5  11  8.33 11 9.26 11  7.81  8  8.47
## 6  14  9.96 14 8.10 14  8.84  8  7.04
## 7   6  7.24  6 6.13  6  6.08  8  5.25
## 8   4  4.26  4 3.10  4  5.39 19 12.50
## 9  12 10.84 12 9.13 12  8.15  8  5.56
## 10  7  4.82  7 7.26  7  6.42  8  7.91
## 11  5  5.68  5 4.74  5  5.73  8  6.89
```

Regression Model For Set 1

```
> fit1<-lm(y1~x1, data=ansc)
> coef(fit1)

## (Intercept)          x1
##   3.0000909   0.5000909

> anova(fit1)

## Analysis of Variance Table
##
## Response: y1
##           Df Sum Sq Mean Sq F value  Pr(>F)
## x1         1 27.510 27.5100   17.99 0.00217 **
## Residuals  9 13.763  1.5292
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Our model is: $Y = 3.0000909 + 0.5000909X$

Regression Model For Set 2

```
> fit2<-lm(y2~x2,data=ansc)
> coef(fit2)

## (Intercept)          x2
##    3.000909    0.500000

> anova(fit2)

## Analysis of Variance Table
##
## Response: y2
##           Df Sum Sq Mean Sq F value   Pr(>F)
## x2         1 27.500 27.5000  17.966 0.002179 **
## Residuals  9 13.776  1.5307
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Again our model is: $Y = 3.0009091 + 0.5X$

Regression Model For Set 3

```
> fit3<-lm(y3~x3, data=ansc)
> coef(fit3)

## (Intercept)          x3
##   3.0024545   0.4997273

> anova(fit3)

## Analysis of Variance Table
##
## Response: y3
##           Df Sum Sq Mean Sq F value   Pr(>F)
## x3         1 27.470 27.4700  17.972 0.002176 **
## Residuals  9 13.756  1.5285
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Again and again our model is: $Y = 3.0024545 + 0.4997273X$

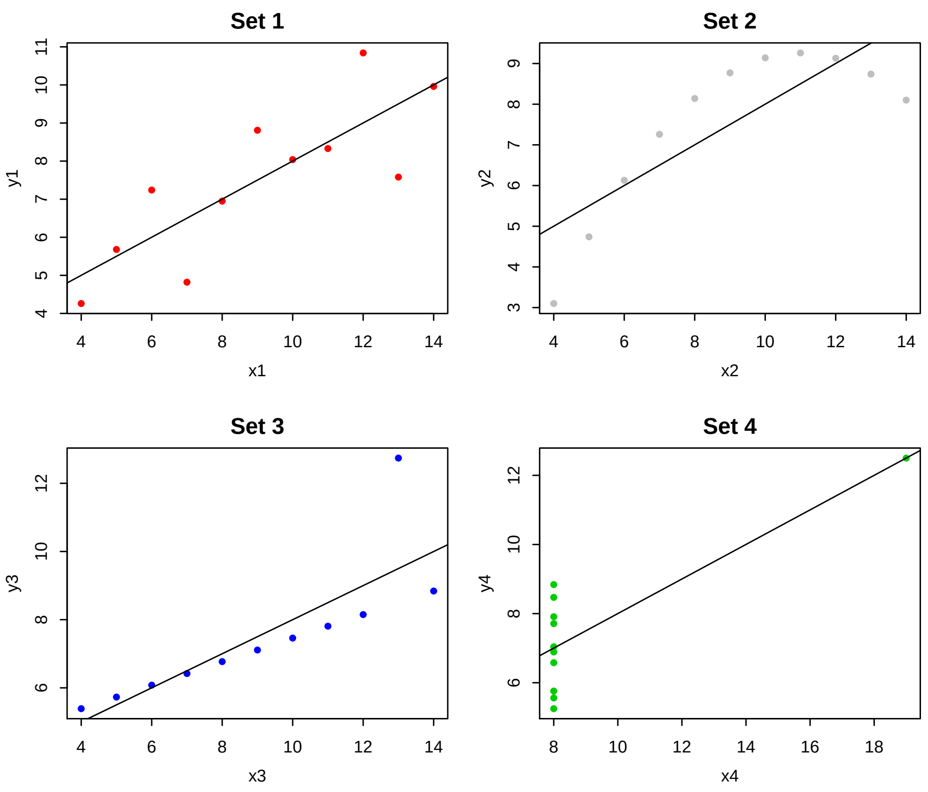
Regression Model For Set 4

```
> fit4<-lm(y4~x4, data=ansc)
> coef(fit4)

## (Intercept)          x4
##   3.0017273   0.4999091

> anova(fit4)
```

```
## Analysis of Variance Table
##
## Response: y4
##           Df Sum Sq Mean Sq F value   Pr(>F)
## x4         1 27.490 27.4900  18.003 0.002165 **
## Residuals  9 13.742  1.5269
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

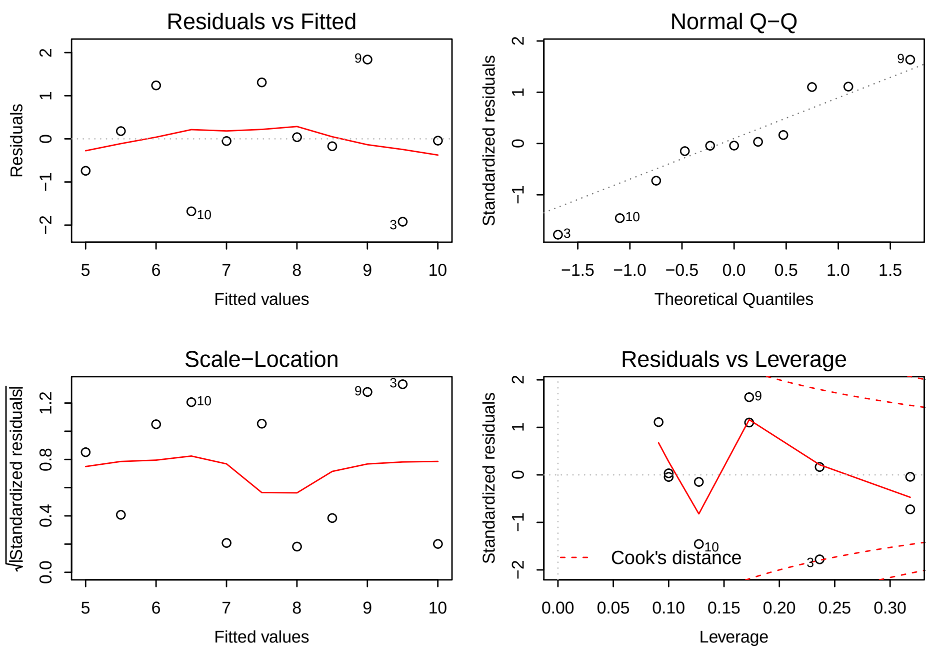
Again and again and again our model is: $Y = 3.0017273 + 0.4999091X$

Raw Data Scatter Plots



How to Check the Model Assumptions?

```
> par(mfrow=c(2,2)); plot(fit1)
```



How to Check the Model Assumptions?

- *Normality*—Normal Q-Q plot

  If you've met the normality assumption, the points should fall on the straight 45-degree line

- *Linearity*—Residuals vs Fitted

  If the dependent variable is linearly related to the independent variables, the model should capture all the systematic variance present in the data, leaving nothing but random noise

- *Homoscedasticity*—Scale-Location

  If you've met the constant variance assumption, the points in the Scale-Location graph should be a random band around a horizontal line.

- *Independence* — these plots tells little information

- *Data Checking* —Residual versus Leverage

  The graph identifies outliers, high-leverage points, and influential observations

Outliers, High Leverage Points and Influential Observations

Three kinds of anomalous data: **Outlier**, **high leverage point** and **influential point**
Concepts:

- Outliers: The observation that isn't predicted well by the fitted regression model. Large standardised residuals
- Influential points: can greatly affect our regression equation
- High leverage points: Observation that is very different from all the others. With high leverage value
- Data points can be both outliers, high leverage and influential

Methods:

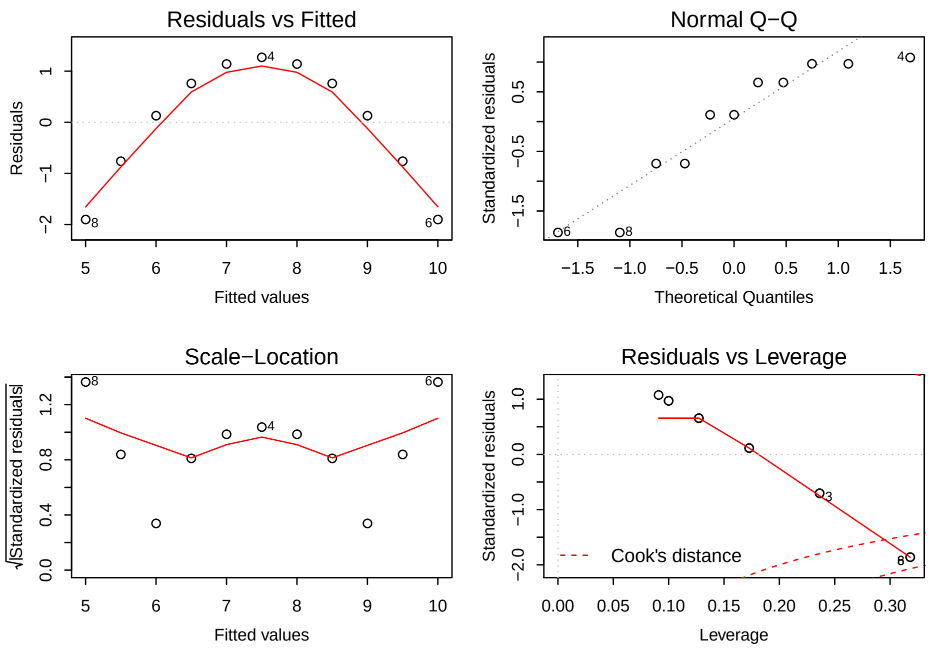
- Outlier: A standardized residual $|r_i| > 2$ happens about 5% of the time
- High leverage: Computed from hat matrix, $h_i > 2p/n$ or $3p/n$
- Influential: Cook's Distance greater than 1

Cook's Distance and Leverage

- *Leverage*: When X_i is far from $\bar{X}$, the greater the leverage h_i of case i
- *hatvalues(model)*: displays the hat values
- *Cook's Distance*: Combinesr_i and h_i to identify influential observations
  Technically, we have
  $$CookD_i = \frac{h_i r_i^2}{(1-h_i)p}$$
  where p =number of variables. Significant if $CookD_i \ggg 1$.
- *plot(model, which=4)*: Plot the Cook's Distance

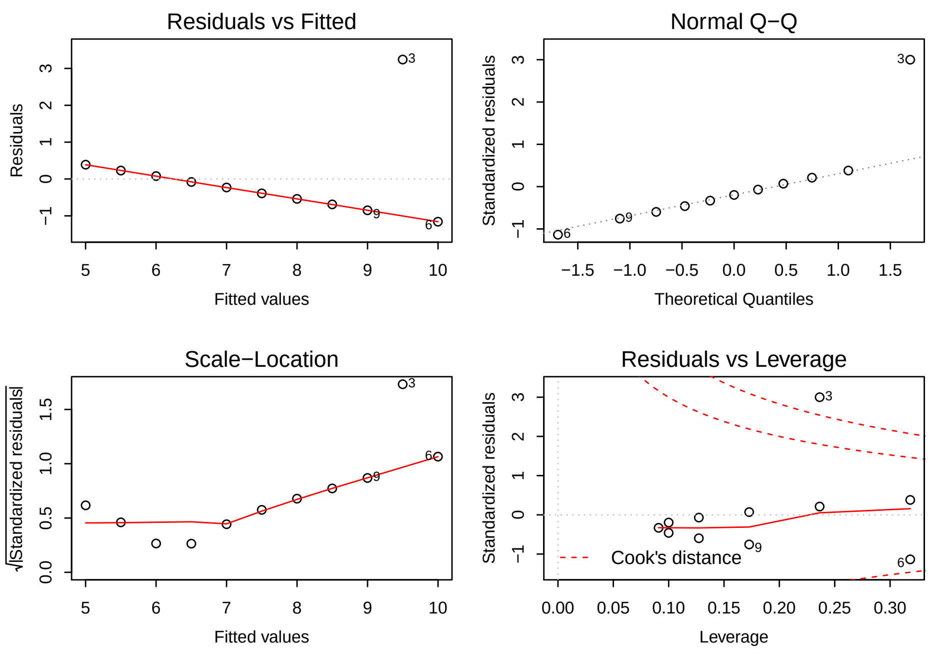
Residuals Plot For Set 2

```
> par(mfrow=c(2,2)); plot(fit2)
```



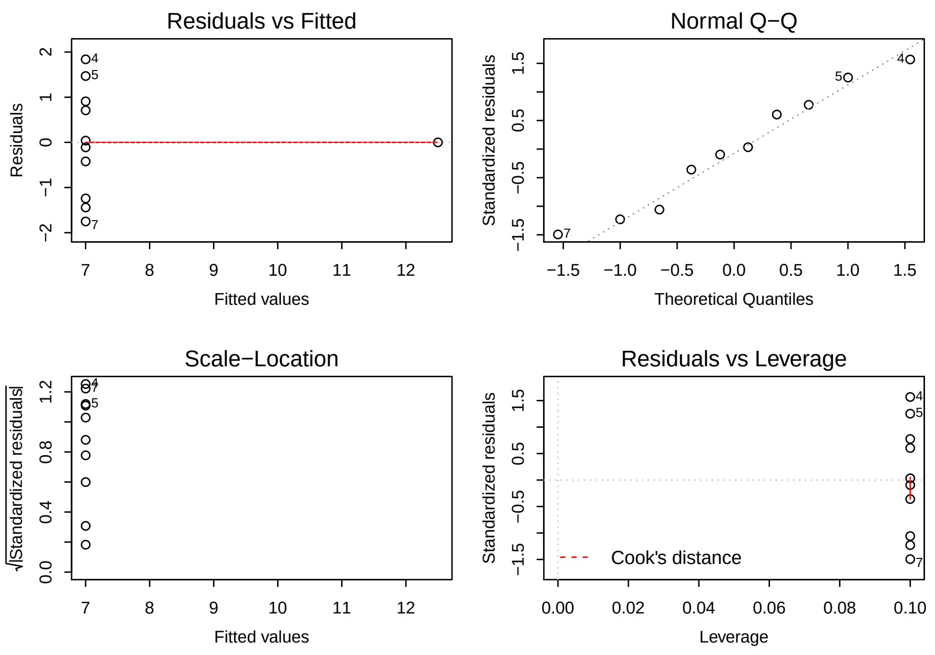
Residuals Plot For Set 3

```
> par(mfrow=c(2,2)); plot(fit3)
```



Residuals Plot For Set 4

```
> par(mfrow=c(2,2)); plot(fit4)
```



Example 2: Cancer and Smoking

$Y=$ lung cancer deaths/million in 1950

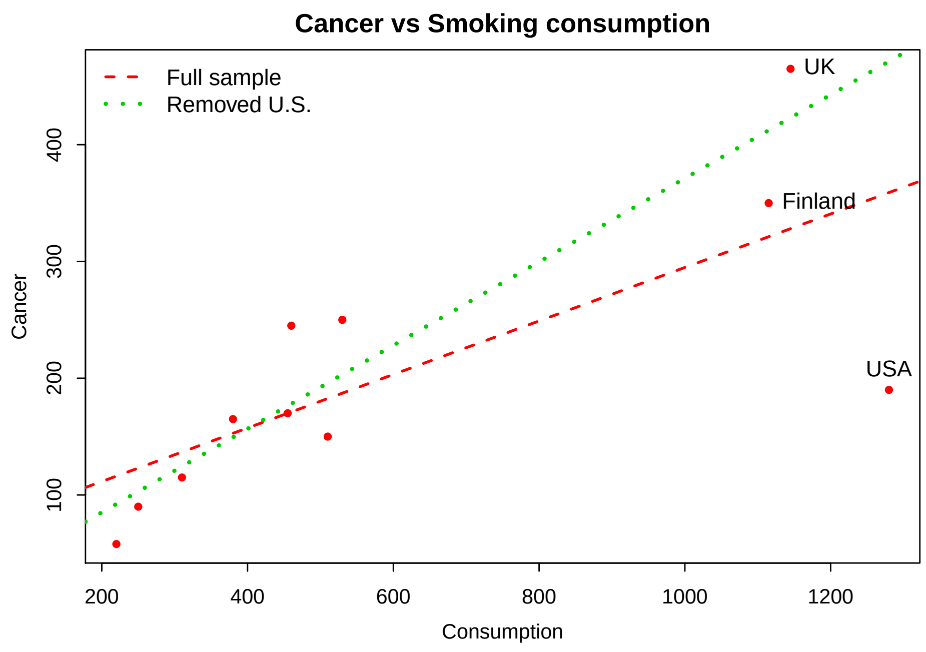
$X=$ cigarette consumption/capita in 1930

```
> smoke<-read.table("R/smoking.txt",header=TRUE)
> colnames(smoke)<-c("Country", "Cancer","Consumption")
> attach(smoke); smoke

##          Country Cancer Consumption
## 1        Iceland     58         220
## 2         Norway     90         250
## 3         Sweden    115         310
## 4         Canada    150         510
## 5        Denmark    165         380
## 6      Australia    170         455
## 7  United States    190        1280
## 8        Holland    245         460
## 9    Switzerland    250         530
## 10       Finland    350        1115
## 11 Great Britain    465        1145
```

Correlation Matrix

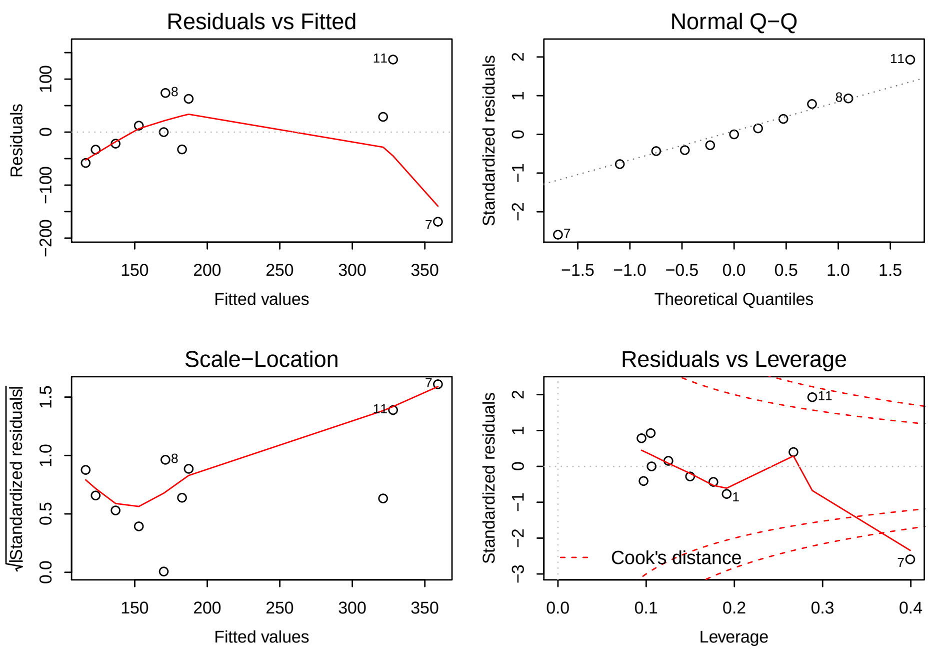
```
> cor(cbind(Cancer,Consumption))

##                Cancer Consumption
## Cancer      1.0000000   0.7409723
## Consumption 0.7409723   1.0000000
```

Smoking Data Plot



4 in 1 Residual Plots

```
> par(mfrow=c(2,2)); plot(fitsmoke)
```



Model Coefficients

```
> coef(fitsmoke)

## (Intercept) Consumption
##  65.7488570   0.2291153

> coef(fitsmoke2)

##     (Intercept) Consumption[-7]
##       13.553435        0.357668

> summary(fitsmoke)$adj.r.squared

## [1] 0.4989333

> summary(fitsmoke2)$adj.r.squared

## [1] 0.8712149
```

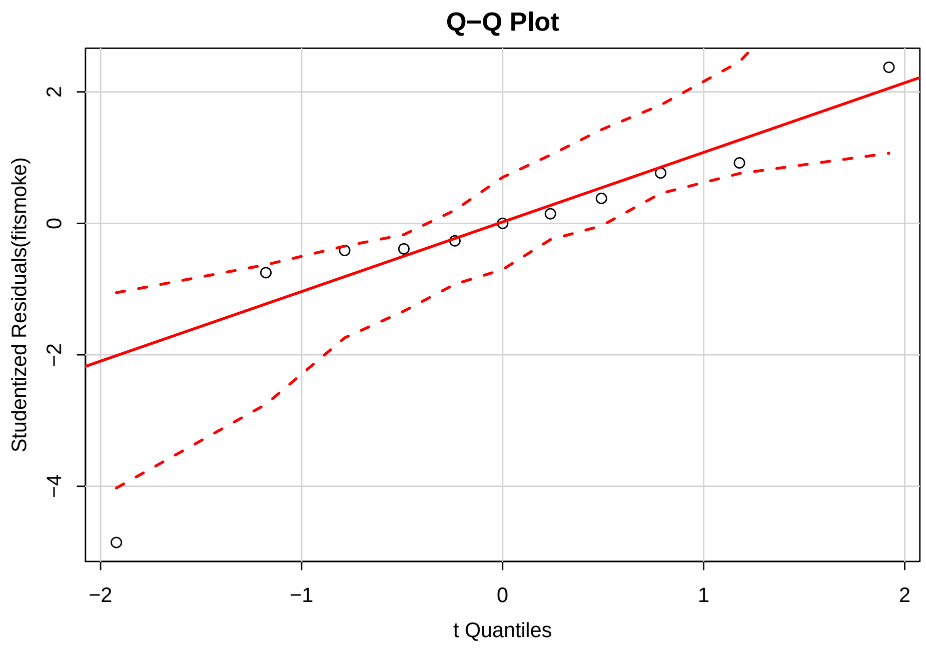
1.2 An Enhanced Approach: Use car Package

Normality

- The *qqPlot( )* function in car provides a more accurate method of assessing the normality assumption than provided by the 4 in 1 *plot( )* function in the base package
- It plots the **studentized residuals** against a $t-$distribution with $n-p-1$ degrees of freedom, where n is the sample size and$p$ is the number of regression parameters (including the intercept)
- *plot(model, which=2)* also provides a Q-Q normal plot

Normality Testing: qqPlot()

```
> qqPlot(fitsmoke,simulate=TRUE,main="Q-Q Plot")
```



Independence of Errors

- Time series data will often display autocorrelation
- The best way to assess the independence is from your knowledge of how the data were collected
- Durbin–Watson test: *durbinWatsonTest()* in car package
- DW statistic:
$$DW = \frac{\sum_{i=2}^{n}(e_i - e_{i-1})^2}{\sum_{i=1}^{n} e_i^2}$$
- DW test model: $e_i = \rho e_{i-1} + \epsilon_i$, $\epsilon_i \sim N(0, \sigma^2)$

DW Test for Independence

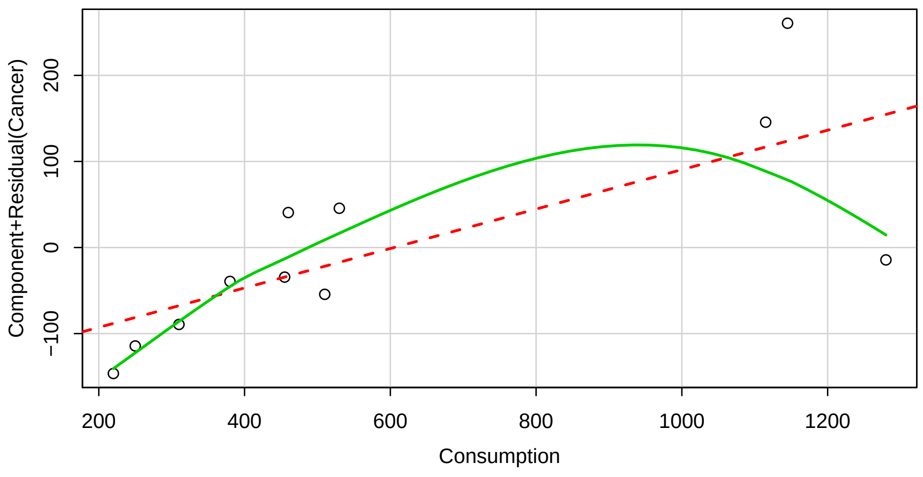
```
> durbinWatsonTest(fitsmoke)
 lag   Autocorrelation   D-W Statistic   p-value
   1        0.01347248        1.625687     0.436
Alternative hypothesis:  rho != 0
```

Conclusion:
The Durbin-Watson Test shows that there is no 1 lag autocorrelation between model errors; the null hypothesis $H_0 : \rho = 0$ is accepted with a p-value 0.436.

Linearity

Component plus residual plots, also known as **partial residual plots**
Use *crPlots( )* function in the car package

```
> crPlots(fitsmoke)
```



Homoscedasticity

The car package also provides two useful functions for identifying *non-constant error variance*

- *ncvTest( ):* Non-constant variance test
- *spreadLevelPlot( ) :* a scatter plot of the absolute standardized residuals versus the fitted values, and superimposes a line of best fit
- *plot(model, which=3)* in plot() also

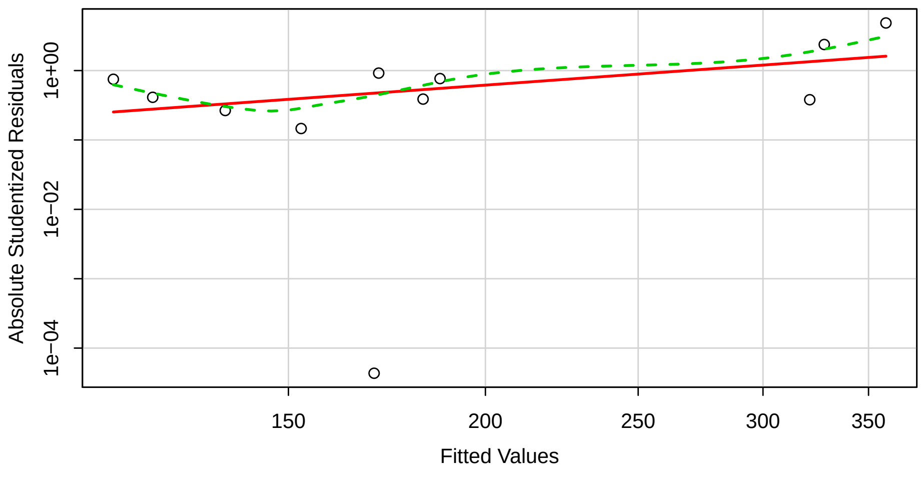
Homoscedasticity Test

> ncvTest(fitsmoke)

```
Non-constant Variance Score
Test Variance formula:  ~ fitted.values
Chisquare = 7.236117 Df = 1
p = 0.007145134
```

Spread Level Plot

```
> spreadLevelPlot(fitsmoke,main="")
```



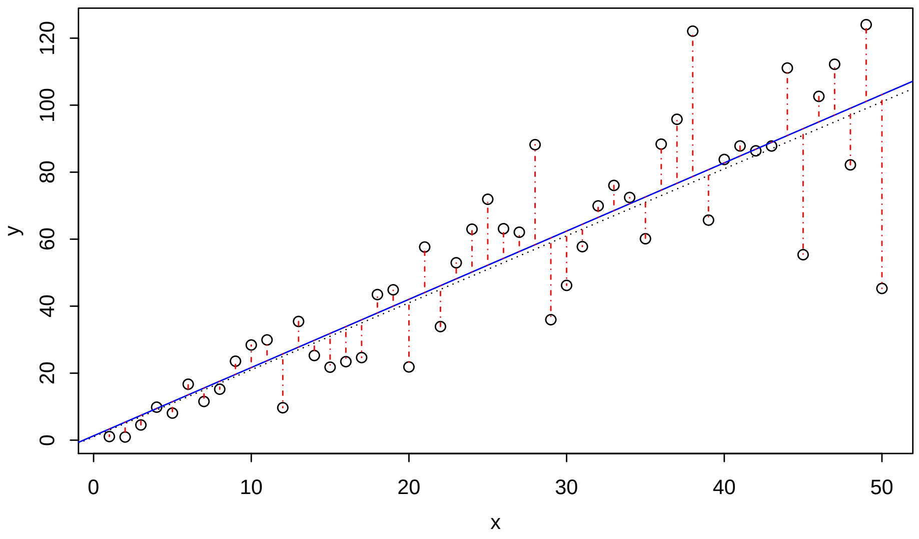
```
##
## Suggested power transformation:  -0.6408373

> detach(smoke)
```

A simulation example of non-constant variance

```
> x <- 1:50
> y = 1 + 2 * x + rnorm(x) * (1 + x/2)
> fm <- lm(y ~ x)
> summary(fm)

##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -57.867  -9.215   1.448   7.594  43.386
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.2936     4.7792   0.271    0.788
## x             2.0372     0.1631  12.490   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.64 on 48 degrees of freedom
## Multiple R-squared:  0.7647,Adjusted R-squared:  0.7598
## F-statistic:   156 on 1 and 48 DF,  p-value: < 2.2e-16
```

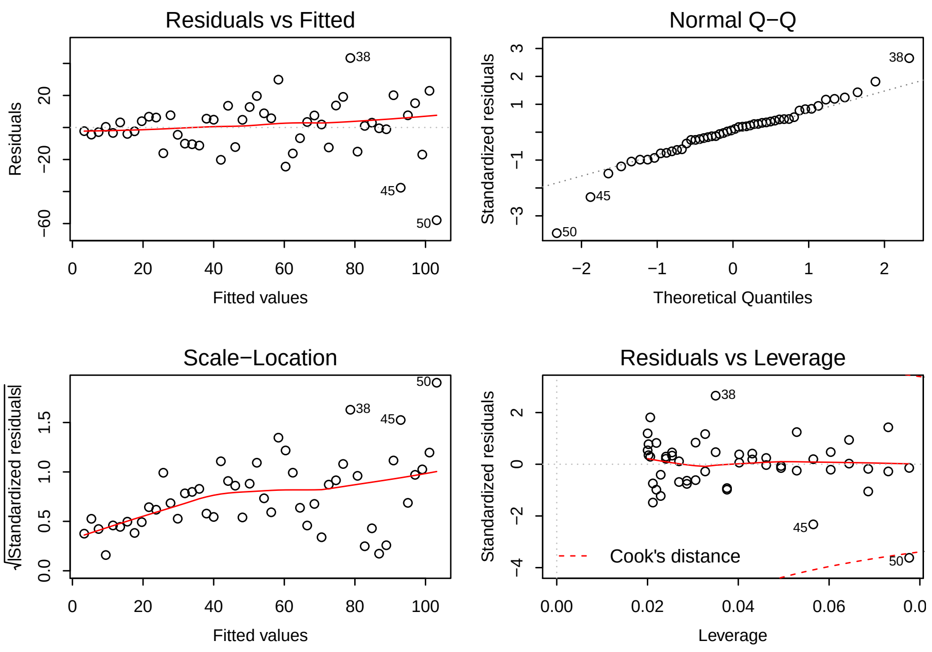
Simulation: scatter plot

```
> plot(x, y, data = rdata)
> abline(1, 2, lty = 3)
> abline(coef(fm), col = "blue")
> segments(x, fitted(fm), x, y, lty = 4, col = "red")
```



Simulation: 4 in 1 plot

```
> par(mfrow=c(2,2)); plot(fm)
```



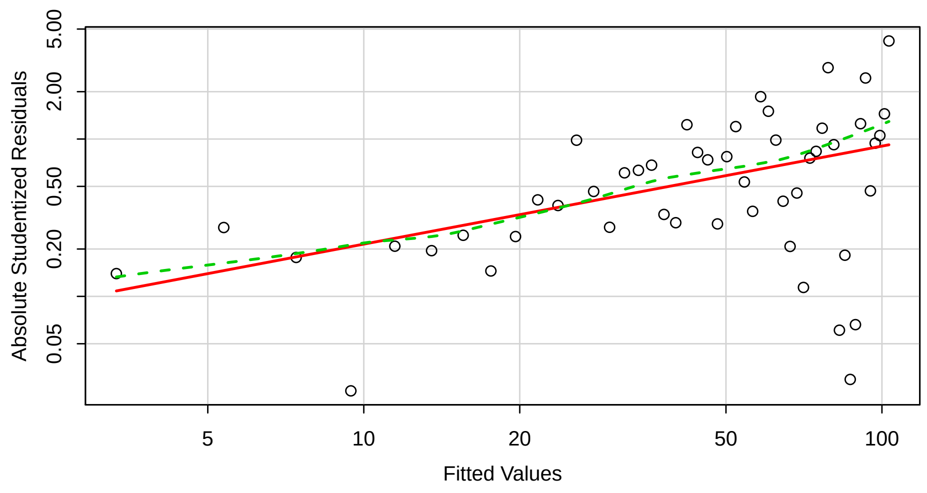
Simulation: ncv test

```
> ncvTest(fm)

## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 19.07177    Df = 1     p = 1.258936e-05
```

Conclusion: The homogeneity of variance assumption is rejected (the p-value is 1.3×10^{-5})

Simulation: Spread Level Plot

```
> spreadLevelPlot(fm,main="")
```



```
##
## Suggested power transformation:  0.3771262

> detach(rdata)
```

Unusual Observations

- *Outlier*: Observations that aren't predicted well by the model
  - Residuals plot
  - Standardized residuals that are larger than 2 or less than –2
  - outlierTest( ) in the car package

OutlierTest

```
> outlierTest( fitsmoke)
  data    rstudent    unadjusted p-value    Bonferonni p
    7    -4.854434         0.0012647           0.013912
```

Conclusion:

The United States (data 7) is an outlier, this is supported with a strong Bonferonni p-value 0.013912.

High Leverage Points

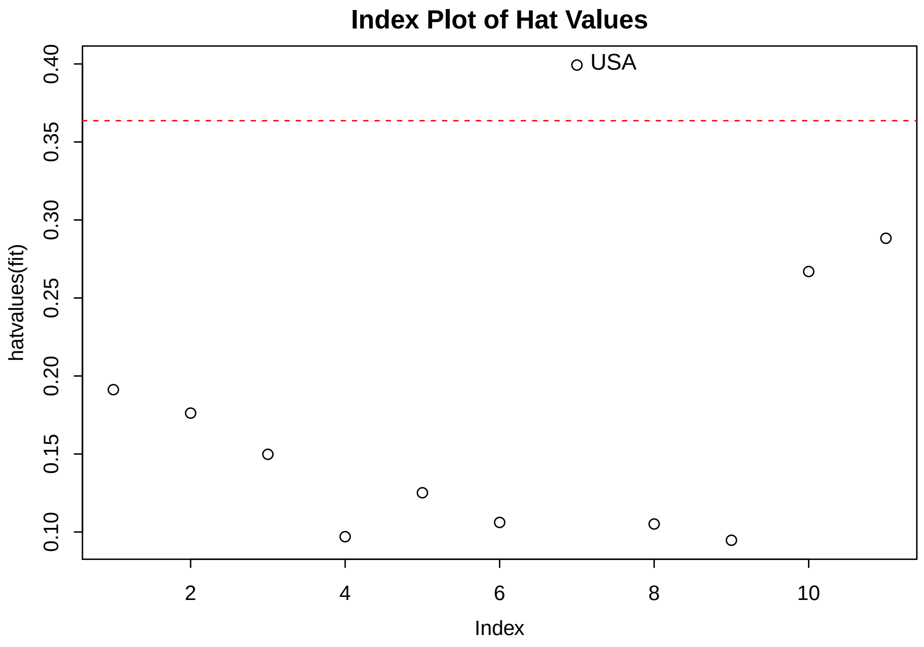
A **high leverage observation** is that very different from all the other observations

- Roughly speaking, an observation with a hat value greater than $2p/n$ or $3p/n$ should be examined
- *hatvalues(model)* extract the hat values
- *plot(model,which=5)* show the residuals by leverage plot
- You can define a *hat.plot( )* function to plot

***hat.plot* Function**

```
hat. plot <- function( fit) {
p <- length( coefficients( fit) )
n <- length( fitted( fit) )
plot( hatvalues( fit) , main=" Index Plot of Hat Values" )
abline( h=c( 2, 3) *p/n, col=" red" , lty=2)
identify( 1: n, hatvalues( fit) , names( hatvalues( fit) ) )
}
```

Hat Plot for Smoking Data



Influential Observations

Influential observations are observations that have a disproportionate impact on the values of the model parameters

- Methods for identifying influential observations:
  1. *plot(model,which=4)* displays Cook's distance for all observations
  2. *avPlots( )*: Added variable plots in car
- Combine the information from outlier, leverage, and influence plots into one highly informative plot: *influencePlot( )*

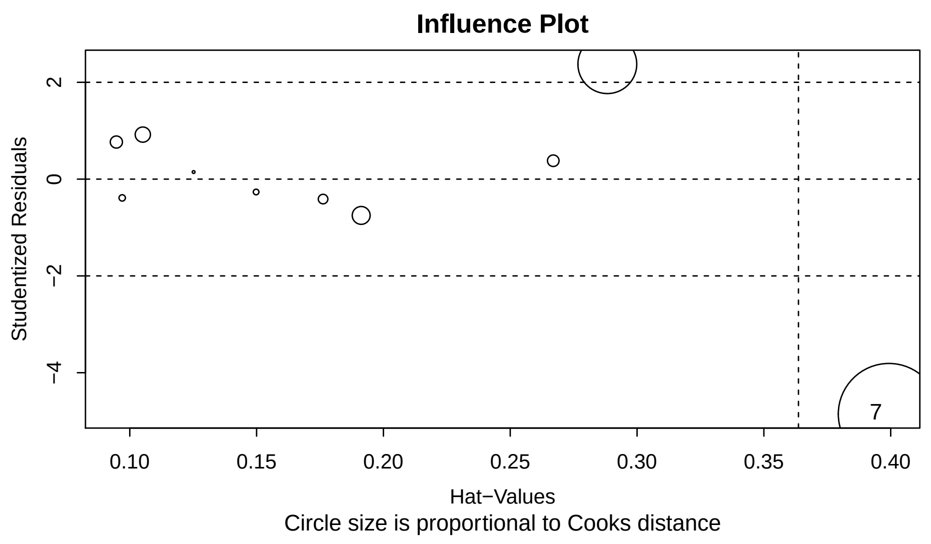
  The plot and outlier will turn out together.

The Unusual Observations Identified By *influencePlot( )*

data	StudRes	Hat	CookD
7	-4.854434	0.3993056	1.494391

InfluencePlot for Smoking Data

```
> influencePlot(fitsmoke,main="Influence Plot",sub="Circle size is proportional to Cooks distance")
```



```
##     StudRes       Hat    CookD
## 7 -4.854434 0.3993056 2.233204

> detach(smoke)

## Error in detach(smoke):  invalid 'name' argument
```

Summary

Model assumptions checking:

- Basic approach: 4 in 1 plot(model) or plot(model, k), k can be 1 to 6.
- Enhanced Approach:
  - Normality: qqPlot() and residplot()
  - Independence: Durbin–Watson test, durbinWatsonTest()
  - Homogeneity (Non constant of variance): ncvTest( ) and spreadLevelPlot( )
  - Linearity: crPlots( )
  - Unusual Observations:
    - Outlier: outlierTest( )
    - High Leverage Points: hat. plot( )
    - Influential observations: Cook's distance and avPlots( )
    - Combine three: influencePlot( )

2 Dealing With Violations

General Approaches

"What do you do if you identify problems?"

There are four approaches to dealing with violations of regression assumptions:

- Deleting observations
- Transforming variables
- Adding or deleting variables
- Using another regression approach

Deleting observations: should be careful

This section we will mainly discuss the transformation of variables.

2.1 Transformations

Transformations

When models don't meet the normality, linearity, or homoscedasticity assumptions, transforming one or more variables can often improve or correct the situation.

- Select an appropriate transformation for the variables
- Re-run the regression on the transformed
- If all is fine then the model holds on the transformed scale
- You can also transform back to the original nonlinear scale
- If a transformation doesn't make sense, you should avoid it

The most common cases are

- Power relationship
- Exponential relationship
- Logit transformation $\ln[Y/(1-Y)]$ if Y is a proportion

The Log-Log Model (Power Model)

A common power model: $Y = AX^b$, where $A = e^a$

- Take logs of both sides to get:

$$\ln Y = \ln A + \ln X = a + b \ln X$$

- Regress $\ln Y$ versus $\ln X$
- The slope, b is an elasticity. % change in Y versus % change in X
- Use when variables are related on a multiplicative, or percentage, scale

Exponential Model

An exponential relationship: $Y = Ae^{bX}$ where $A = e^a$.

Taking logs of both sides gives

$$\ln Y = \ln A + bX = a + bX$$

We run a regression of $\ln Y$ on X

- Logging the Y variable helps with large outliers
- Not all variables can be logged
  - Y needs to be positive
  - Dummy variables $X = 0$ or 1 can't be logged
  - Counting variables are usually left alone as well

Response or Predictor?

Apply a transformation to the response variable:

- When the model violates the normality assumption
- When appears heteroscedasticity (nonconstant error variance)
  - *Box–Cox* transformation: $f(x) = (x^\lambda - 1)/\lambda$, for $\lambda \neq 0$; $f(x) = \ln(x)$ if $\lambda = 0$
  - *powerTransform( )* in the car package, generates a maximum-likelihood estimation of the power λ

Apply a transformation to the predictor variable:

- When the assumption of linearity is violated: *boxTidwell( )*

Common transformations:

λ	-2	-1	-0.5	0	0.5	1	2
Transformation	$1/Y^2$	$1/Y$	$1/\sqrt{Y}$	$\log(Y)$	$\sqrt{Y}$	None	Y^2

Question: how to interpret the slope parameter?

1. $\ln Y = a + bX$
2. $Y = a + b\ln X$
3. $\ln Y = a + b\ln X$

2.2 Mammals Example

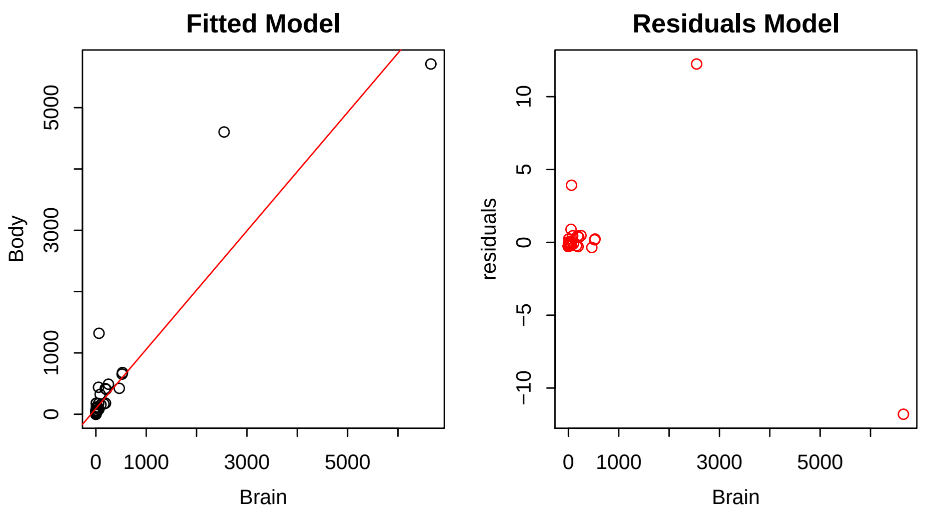
Mammals Data

Brain and Body weight for 62 mammals. Whose the smartest animal?

```
> mam<-read.table("R/mammals.txt",header=TRUE)
> head(mam)

##                      Mammal    Brain   Body
## 1          African_elephant 6654.000 5712.0
## 2 African_giant_pouched_rat    1.000    6.6
## 3                Arctic_Fox    3.385   44.5
## 4    Arctic_ground_squirrel    0.920    5.7
## 5            Asian_elephant 2547.000 4603.0
## 6                    Baboon   10.550  179.5
```

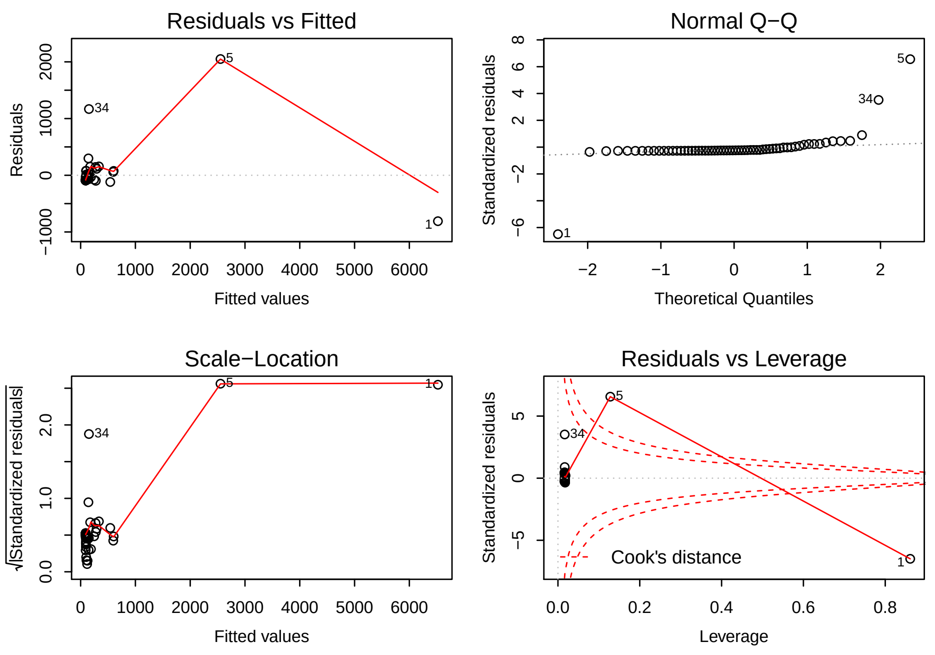
Residuals Analysis

```
> attach(mam)
> par(mfrow = c(1, 2)); fitmam<-lm(Body~Brain)
> plot(Brain,Body,main="Fitted Model")
> abline(fitmam,col="red"); residuals<-rstudent(fitmam)
> plot(Brain,residuals, main="Residuals Model",col="red")
```



Residual Plots: Transformation Required

```
> par( mfrow=c( 2, 2) ); plot(fitmam)
```

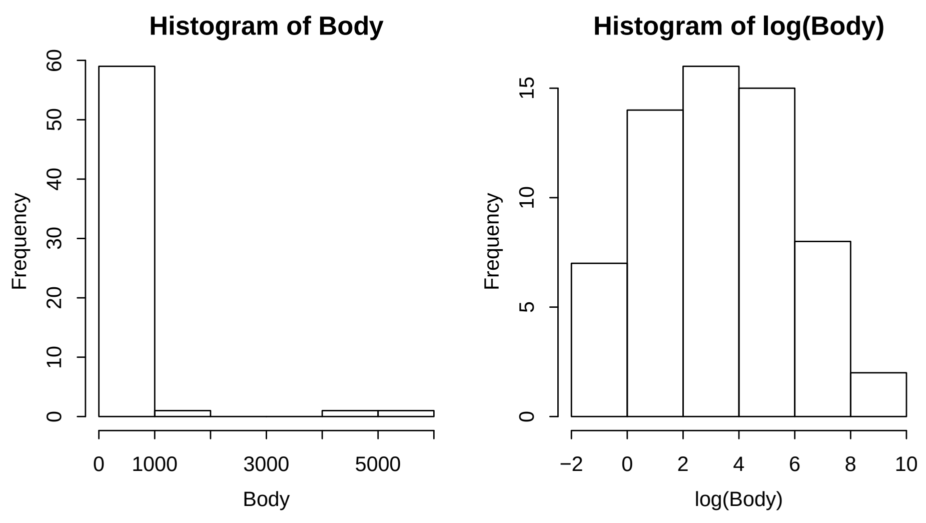


Transformation for Response Variable

```
> summary(powerTransform(Body))

## bcPower Transformation to Normality
##      Est Power Rounded Pwr Wald Lwr bnd Wald Upr Bnd
## Body    -4e-04           0      -0.0945       0.0938
##
## Likelihood ratio tests about transformation parameters
##                                LRT df      pval
## LR test, lambda = (0) 6.743806e-05  1 0.9934478
## LR test, lambda = (1) 3.427569e+02  1 0.0000000
```

Are the Body a Normal Distribution?

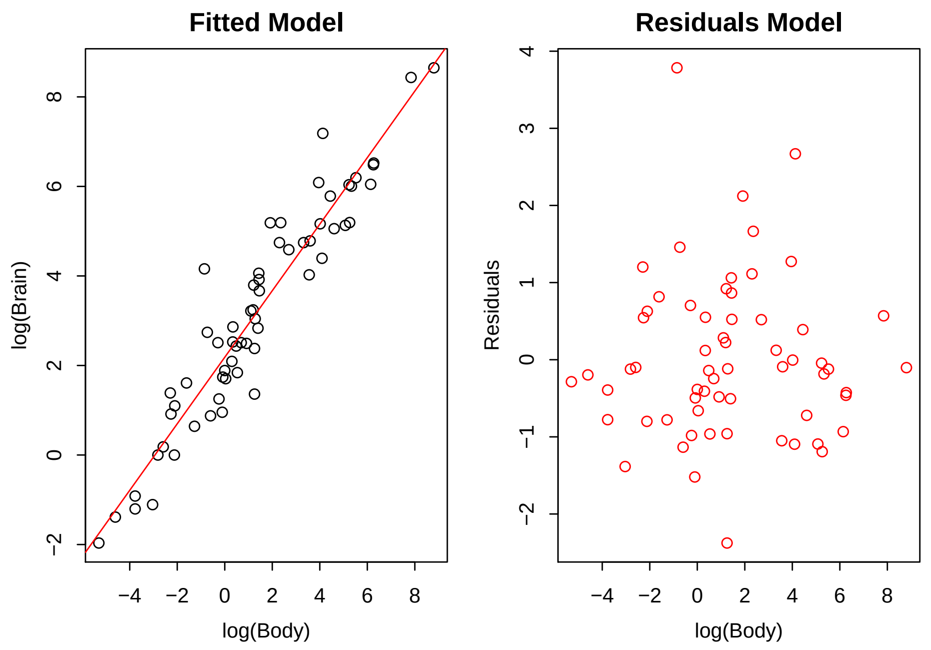
```
> par(mfrow = c(1, 2))
> hist(Body); hist(log(Body))
```



Transformation for predictors

```
> boxTidwell(Body ~ Brain)

##  Score Statistic p-value MLE of lambda
##        -6.875299       0     0.6920404
##
## iterations =  5
```

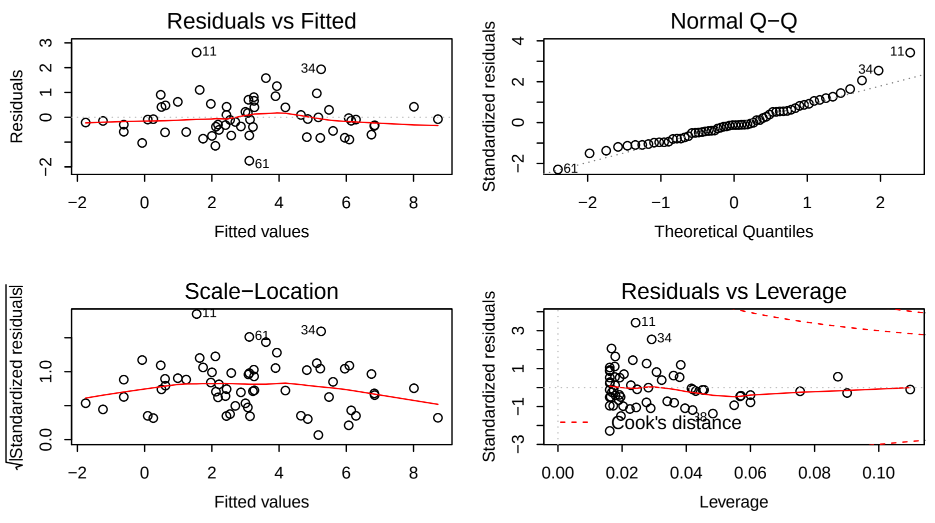
Log-Log Model



- *Much better!*

Transformed Log-Log Model

```
> par(mfrow=c(2,2))
> fitmamln<-lm(logbody~logbrain)
> plot(fitmamln)
```



Log-Log Model

```
> summary(fitmamln)

##
## Call:
## lm(formula = logbody ~ logbrain)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.75335 -0.53911 -0.09239  0.42068  2.61153
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.18328    0.10682   20.44   <2e-16 ***
## logbrain     0.74320    0.03166   23.48   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7722 on 60 degrees of freedom
## Multiple R-squared:  0.9018,Adjusted R-squared:  0.9002
## F-statistic: 551.2 on 1 and 60 DF,  p-value: < 2.2e-16
```

Outliers

```
> rstudent(fitmamln)[rstudent(fitmamln)>2]

##       11       34       50
## 3.784865 2.669789 2.122100
```

There is a residual value of 3.7848652, extreme outlier.

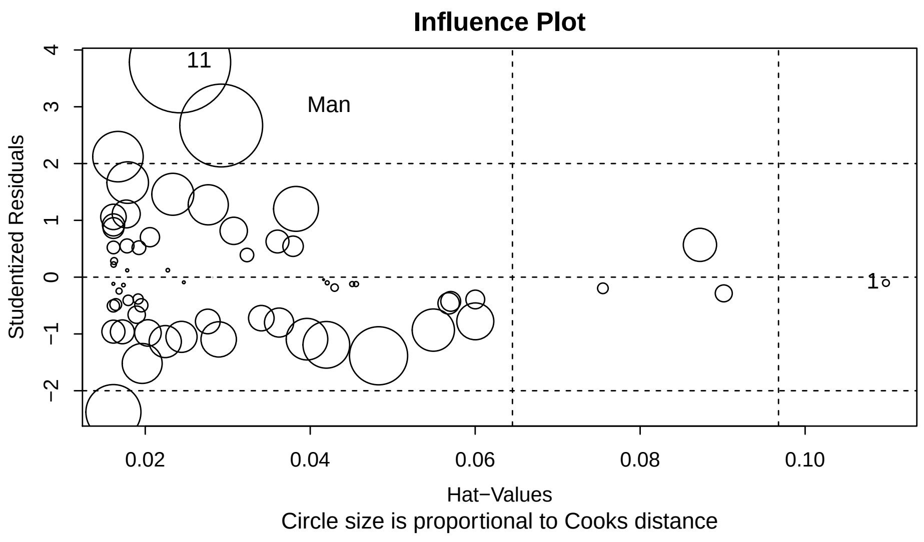
How to dealing with the outlier?

It corresponds to the *Chinchilla*. This suggests that the Chinchilla is a master race of supreme intelligence!

- This is clearly not true, it should be a data entry error. the brain weight is given as 64 grams and should only be 6.4 grams.
- The next largest residual corresponds to mankind.

Influence Points

```
> influencePlot(fitmamln,main="Influence Plot",
+ sub="Circle size is proportional to Cooks distance")
> text(0.038,3,"Man",pos=4)
```



```
> detach(mam)
```

Summary

Model Assumptions:

- Normality
- Linearity
- Homoscedasticity
- Independence
- Outliers , high leverage points and influential observations in your dataset

Corrective Measures

- Transforming variables
- Deleting observations

Comment: *It is crucial to perform appropriate model diagnostics*