

# TOPIC 4: Regression Model Diagnostics

## Multiple Linear Regression

### Part IV: MULTIPLE REGRESSION DIAGNOSTICS

© Thuntida Ngamkham 2022 (edited by Paul Galpern)

#### Residual Analysis: Checking the Regression Assumptions

Most statistical tests rely upon certain assumptions about the variables used in the analysis. **When these assumptions are not met the results may not be trustworthy.** The assumptions and conditions for the multiple regression model sound nearly the same as for simple linear regression, but with more variables in the model.

#### 1. Linearity Assumption

The linear regression model assumes that there is a straight-line relationship between the predictors and the response. If the true relationship is far from linear, then virtually all of the conclusions that we draw from the fit are suspect. In addition, the prediction accuracy of the model can be significantly reduced.

**Residual plots** are a useful graphical tool for identifying non-linearity. In the case of multiple regression model since there are multiple predictors, we instead plot the residuals versus predicted (or fitted) values  $\hat{y}_i$ . Ideally, the residual plot will show no discernible pattern. The presence of a pattern may indicate a problem with some aspect of the linear model.

If the residual plot indicates that there are non-linear associations in the data, then a simple approach is to use non-linear transformations of the predictors, such as  $\log(X)$ ,  $\sqrt{X}$ , and  $X^2$ , in the regression model.

```
library(ggplot2)
Advertising=read.table("Advertising.txt", header = TRUE, sep = "\t" )
model<-lm(sale~tv+radio+tv*radio, data=Advertising)
summary(model)
```

```
##
## Call:
## lm(formula = sale ~ tv + radio + tv * radio, data = Advertising)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -6.3366  -0.4028   0.1831   0.5948   1.5246
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.750e+00  2.479e-01  27.233  <2e-16 ***
## tv           1.910e-02  1.504e-03  12.699  <2e-16 ***
```

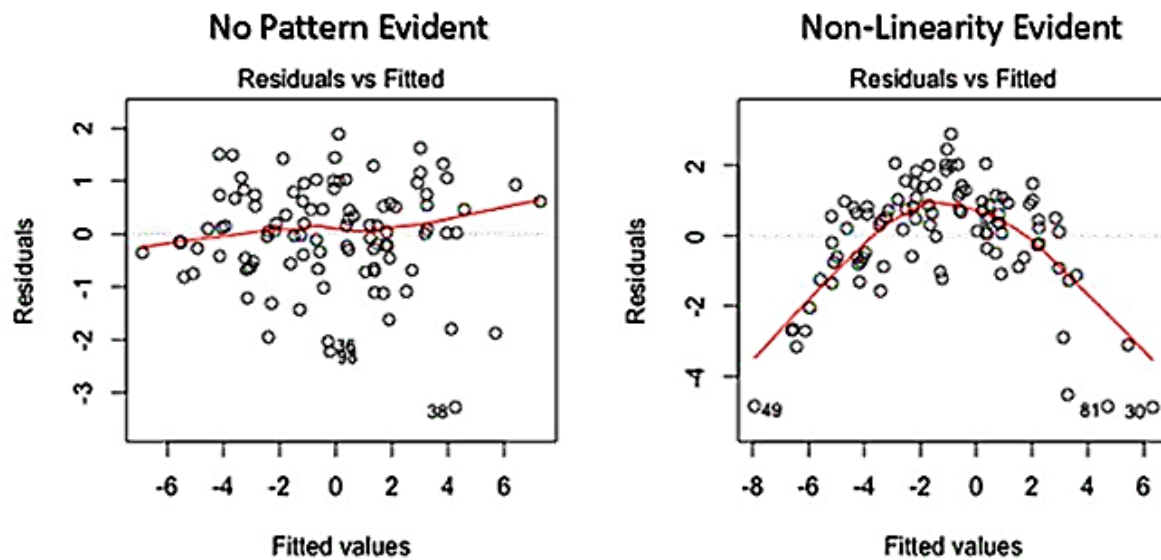
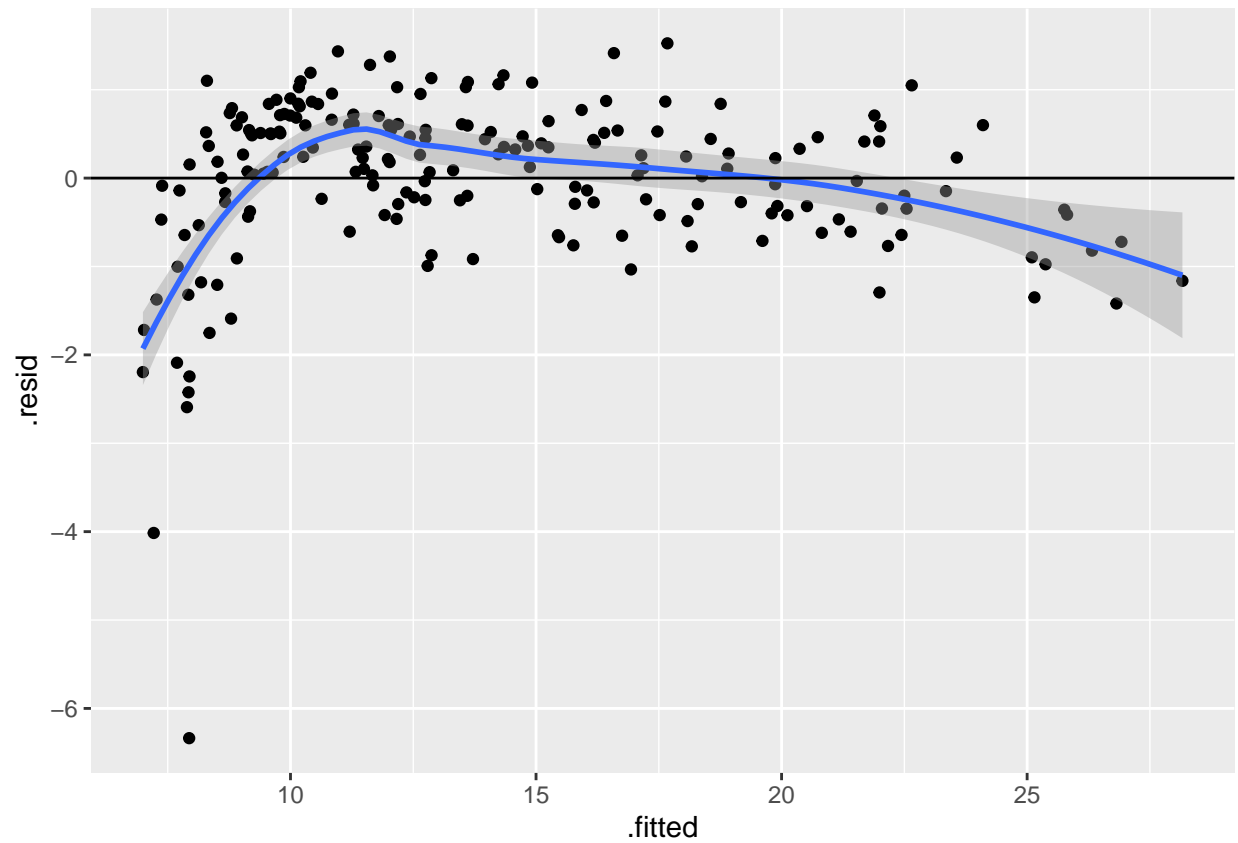


Figure 1: This scatter plot shows the distribution of residuals (errors) vs fitted values (predicted values)

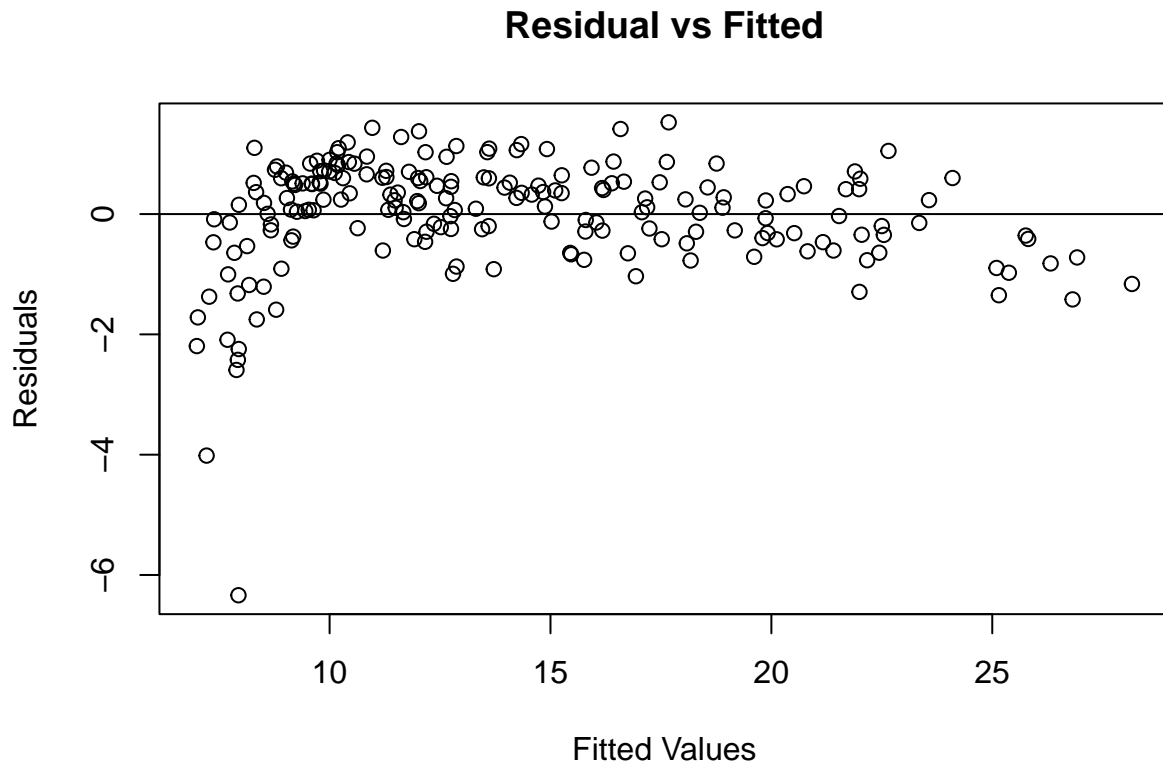
```
## radio      2.886e-02  8.905e-03  3.241  0.0014 **
## tv:radio    1.086e-03  5.242e-05 20.727  <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9435 on 196 degrees of freedom
## Multiple R-squared:  0.9678, Adjusted R-squared:  0.9673
## F-statistic: 1963 on 3 and 196 DF, p-value: < 2.2e-16
```

```
ggplot(model, aes(x=.fitted, y=.resid)) +
  geom_point() + geom_smooth()+
  geom_hline(yintercept = 0)
```

```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```



```
#optional graph from plot()  
plot(fitted(model), residuals(model), xlab="Fitted Values", ylab="Residuals")  
abline(h=0, lty=1)  
title("Residual vs Fitted")
```



*R functions*

`ggplot(model, aes(x=..., y=...))`: mapping 2 variables on a and y axis

`.fitted` : Fitted values of model

`.resid` : Residuals

`geom_hline(yintercept = 0)`:add a horizontal line

`geom_point()` : add a layer of points to the plot

From the Advertising example, the output displays the residual plot that results from the model  $\hat{Sale} = 6.750 + 0.01910tv + 0.02886radio + 0.001086tv * radio$ . There appears to be a little pattern in the residuals, suggesting that the quadratic term or logarithmic might improve the fit to the data.

```
library(ggplot2)
Advertising=read.table("Advertising.txt", header = TRUE, sep = "\t" )
library(GGally)
```

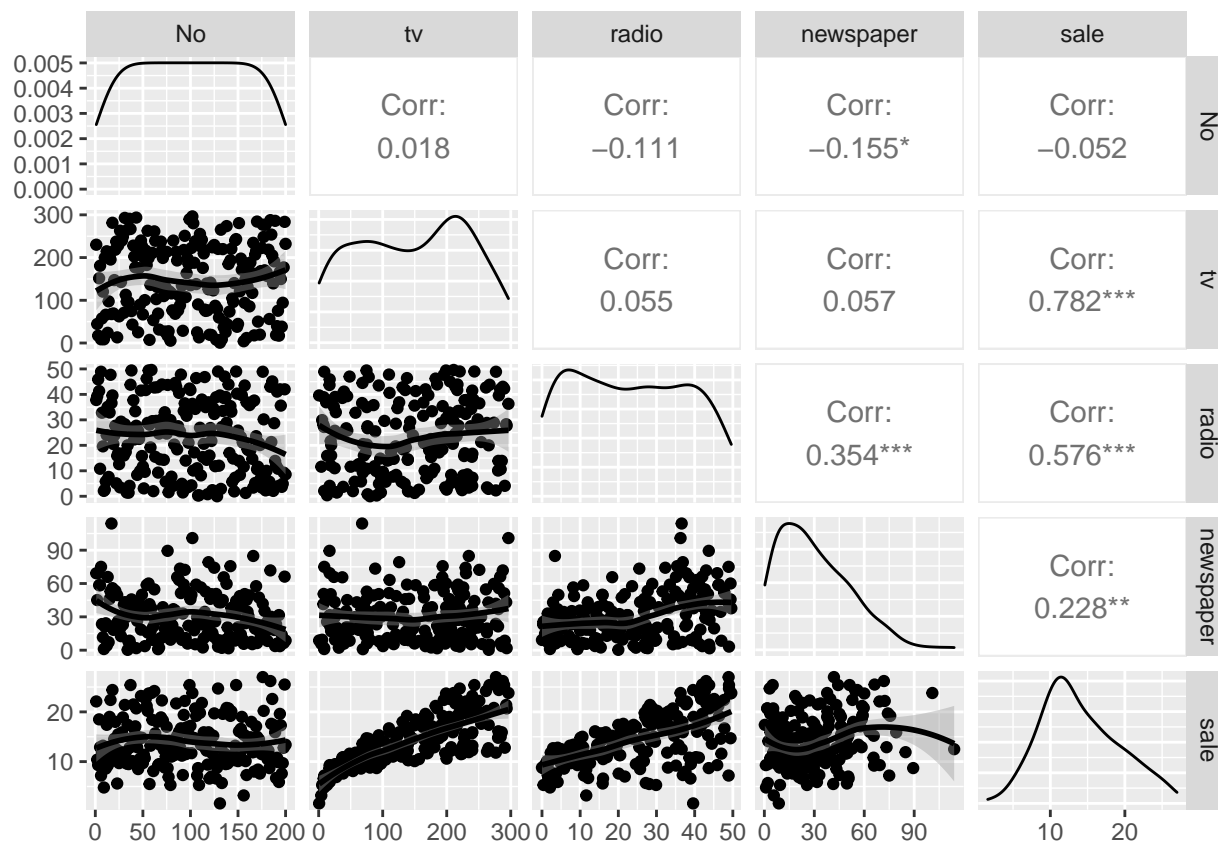
```
## Warning: package 'GGally' was built under R version 4.2.2
```

```
## Registered S3 method overwritten by 'GGally':
```

```
##   method from
```

```
##   +.gg   ggplot2
```

```
ggpairs(Advertising, lower = list(continuous = "smooth_loess", combo =
  "facethist", discrete = "facetbar", na = "na"))
```



```
model<-lm(sale~tv+radio+tv*radio, data=Advertising)
quadmodel<-lm(sale~tv+I(tv^2)+radio+tv*radio, data=Advertising)
cubic<-lm(sale~tv+I(tv^2)+I(tv^3)+radio+tv*radio, data=Advertising)
### OR
# cubic<-lm(sale~poly(tv,3,raw=T)+radio+tv*radio, data=Advertising)

summary(model)$adj.r.squared
```

```
## [1] 0.9672975
```

```
summary(quadmodel)$adj.r.squared
```

```
## [1] 0.985707
```

```
summary(cubic)$adj.r.squared
```

```
## [1] 0.99072
```

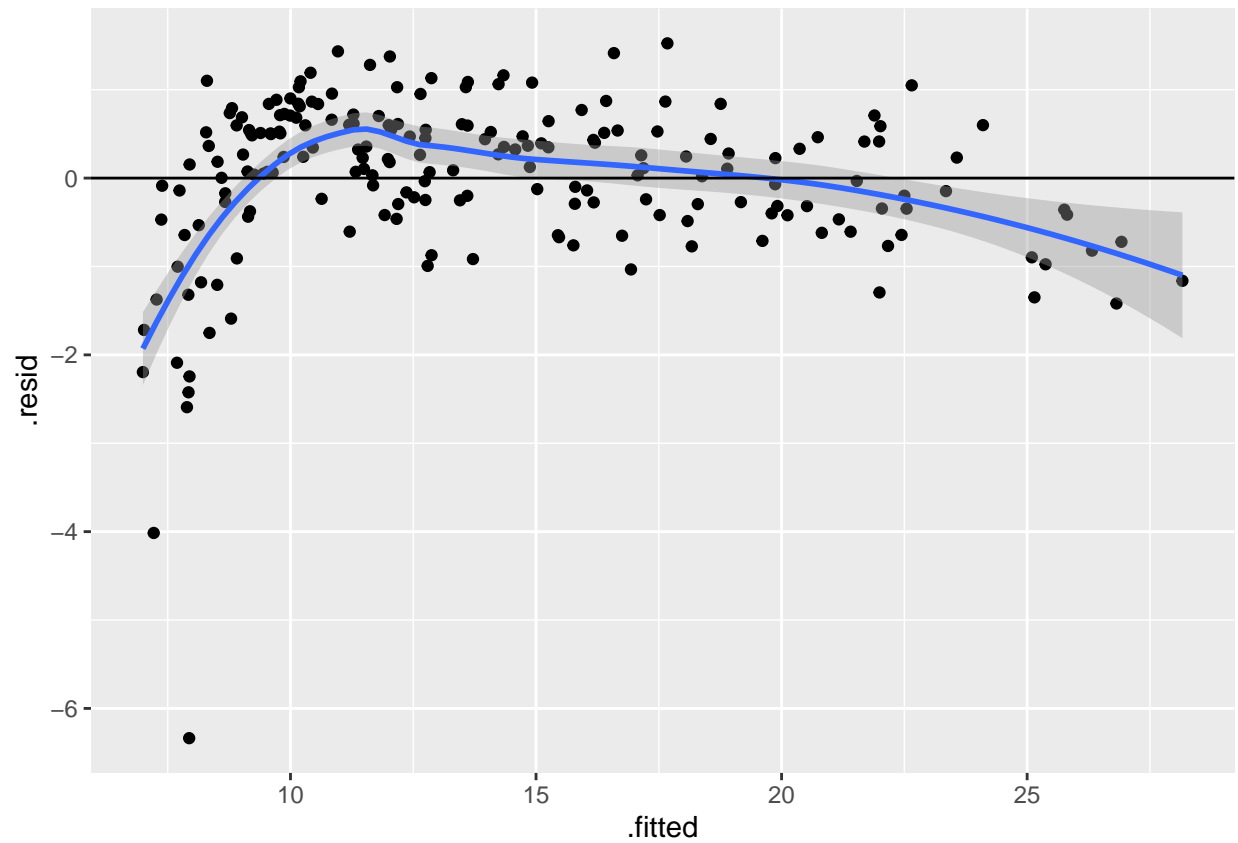
```
summary(cubic)
```

```
##
## Call:
## lm(formula = sale ~ tv + I(tv^2) + I(tv^3) + radio + tv * radio,
```

```
##      data = Advertising)
##
## Residuals:
##      Min        1Q      Median        3Q        Max
## -4.2184 -0.2106  0.0223   0.2454   1.1677
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.061e+00  1.871e-01  21.709 < 2e-16 ***
## tv           8.998e-02  4.193e-03  21.458 < 2e-16 ***
## I(tv^2)      -4.327e-04  3.180e-05 -13.604 < 2e-16 ***
## I(tv^3)       7.278e-07  7.058e-08  10.312 < 2e-16 ***
## radio        4.206e-02  4.801e-03   8.761 9.63e-16 ***
## tv:radio      1.044e-03  2.811e-05   37.129 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5026 on 194 degrees of freedom
## Multiple R-squared:  0.991, Adjusted R-squared:  0.9907
## F-statistic: 4250 on 5 and 194 DF, p-value: < 2.2e-16
```

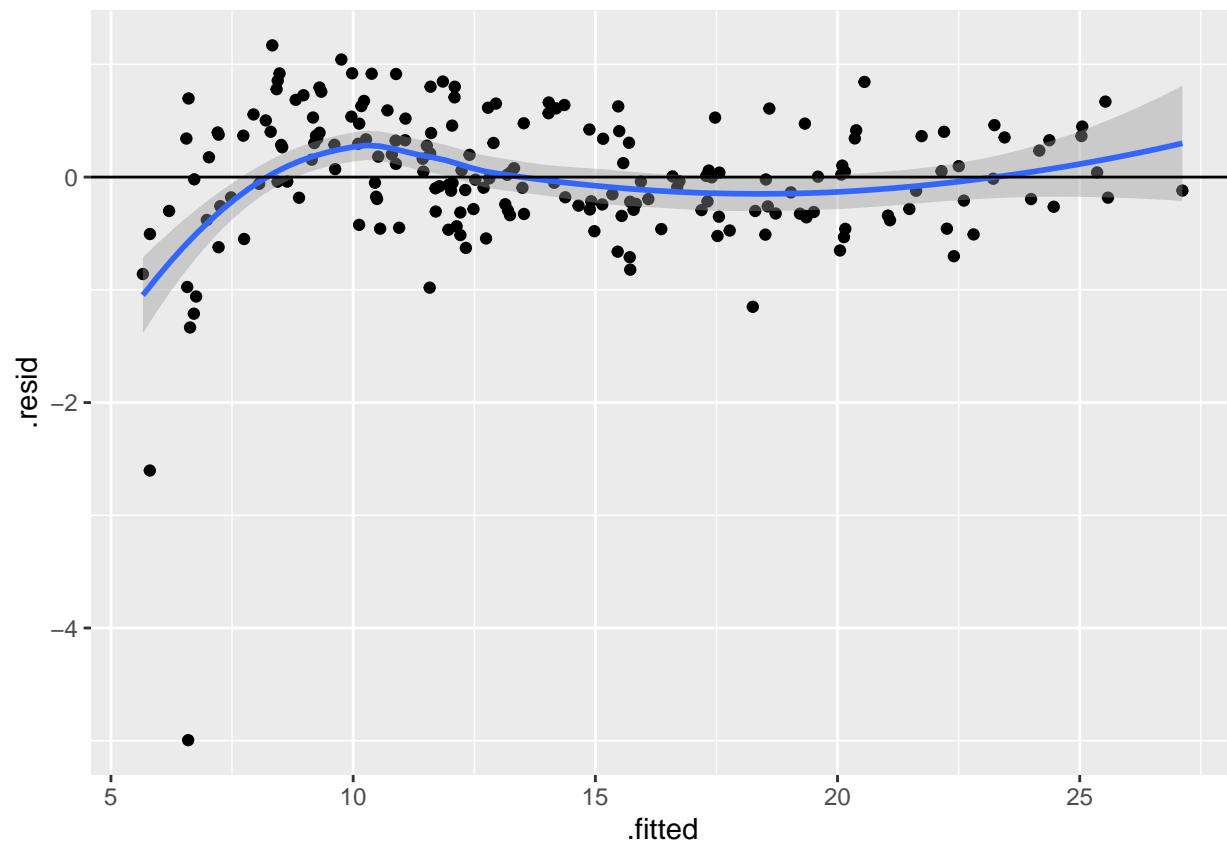
```
#residual vs fitted data plot for the simple model
ggplot(model, aes(x=.fitted, y=.resid)) +
  geom_point() +geom_smooth()+
  geom_hline(yintercept = 0)
```

```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```



```
#residual vs fitted data plot for the quadratic model  
ggplot(quadmodel, aes(x=.fitted, y=.resid)) +  
  geom_point() +geom_smooth()+  
  geom_hline(yintercept = 0)
```

```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```

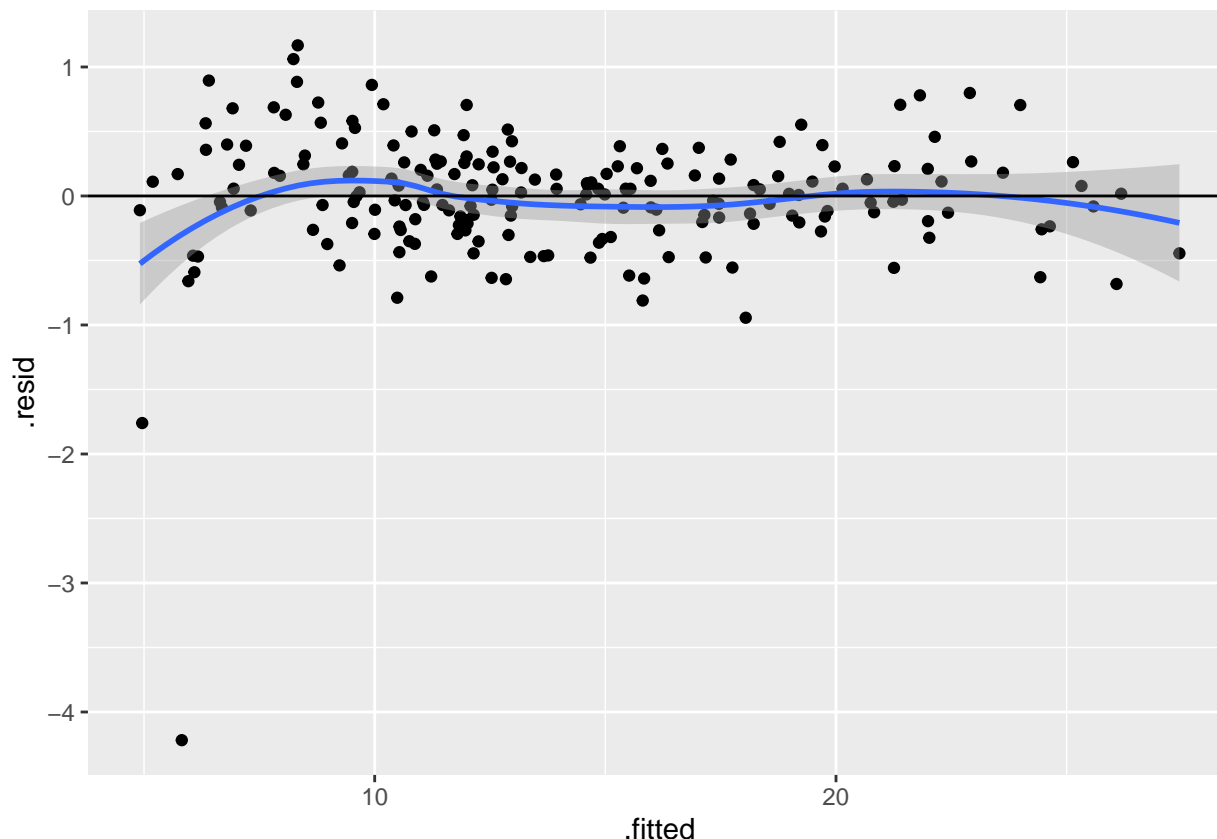


```
#residual vs fitted data plot for the cubic model
```

```
ggplot(cubic, aes(x=.fitted, y=.resid)) +  
  geom_point() +geom_smooth()+  
  geom_hline(yintercept = 0)
```

```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```





From the output, there appears to be a little pattern for the quadratic regression model while the cubic model shows no pattern of the residuals at all. Moreover, the  $R^2 - adj$  of the cubic model is 0.9907 indicates the variation in  $y$  that can be explained by this model is 99.02% with RMSE= 0.5026. Therefore, we can conclude that the cubic model is the best fit model to predict  $Y$  among the models we considered.

---

## Inclass Practice Problem 16

From the clerical staff work hours, use residual plots to conduct a residual analysis of the data. Check Linearity Assumption. If a trend is detected, how would you like to transform the predictors in the model?

## 2. Independence Assumption

An important assumption of the linear regression model is that the error terms,  $\epsilon_1, \epsilon_2, \epsilon_3, \dots, \epsilon_n$  are uncorrelated (must be mutually independent). What does this mean? For instance, if the errors are uncorrelated, then the fact that  $\epsilon_i$  is positive provides little or no information about the sign of  $\epsilon_{i+1}$ .

The assumption of independent errors is violated when successive errors are correlated. This typically occurs when the data for both dependent and independent variables are observed sequentially over a period of time-called **time-series data**

We can check displays of the regression residuals for evidence of patterns, trends or clumping, any of which would suggest a failure of independence. In the special case when response  $Y$  is related to time (time series data), a common violation of the Independence Assumption is for the errors to be correlated. This violation

can be checked by plotting the residuals against the order of occurrence (time plot of the residuals and looking for pattern).

In the Advertising example, the subjects were not related to time, so we can pretty sure that their measurement are independent.

### 3. Equal Variance Assumption

Another important assumption of the linear regression model is that the error terms have a constant variance (homoscedasticity),  $Var(\epsilon_i) = \sigma^2$ . Unfortunately, it is often the case that the variances of the error terms are non-constant. For instance, the variances of the error terms may increase with the value of the response. One can identify non-constant variances in the errors, or **heteroscedasticity**

Heteroscedasticity means unequal scatter. In regression analysis, heteroscedasticity is a systematic change in the spread of the residuals over the range of measured values. An example is shown in the left-hand panel of the figure below, in which the magnitude of the residuals tends to increase with the fitted values. When faced with this problem, one possible solution is to transform the response  $Y$  using a concave function such as  $\log(Y)$  or  $\sqrt{X}$ . Such a transformation results in a greater amount of shrinkage of the larger responses, leading to a reduction in heteroscedasticity. The right-hand panel of the figure below displays the residual plot after transforming the response.

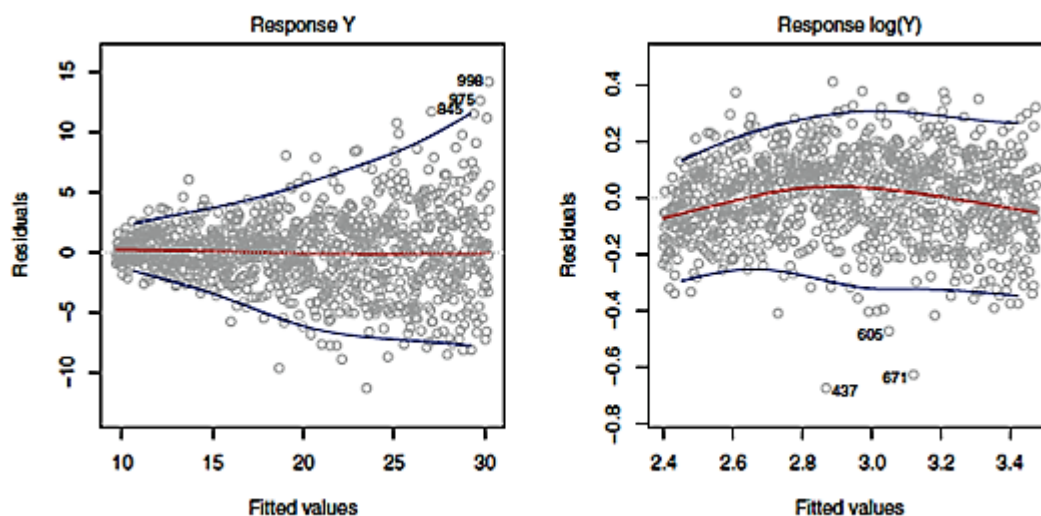


Figure 2: Residual plots.

*In each plot, the red line is a smooth fit to the residuals, intended to make it easier to identify a trend. The blue lines track the outer quantiles of the residuals, and emphasize patterns.*

*Left: The funnel shape indicates heteroscedasticity.*

*Right: The predictor has been log-transformed, and there is now no evidence of heteroscedasticity.*

A scale-location plot between fitted value and standardized residuals can also be checked for heteroscedasticity. It's also called Spread-Location plot. This plot shows if residuals are spread equally along the ranges of predictors. You can check the assumption of equal variance (homoscedasticity). It's good if you see a

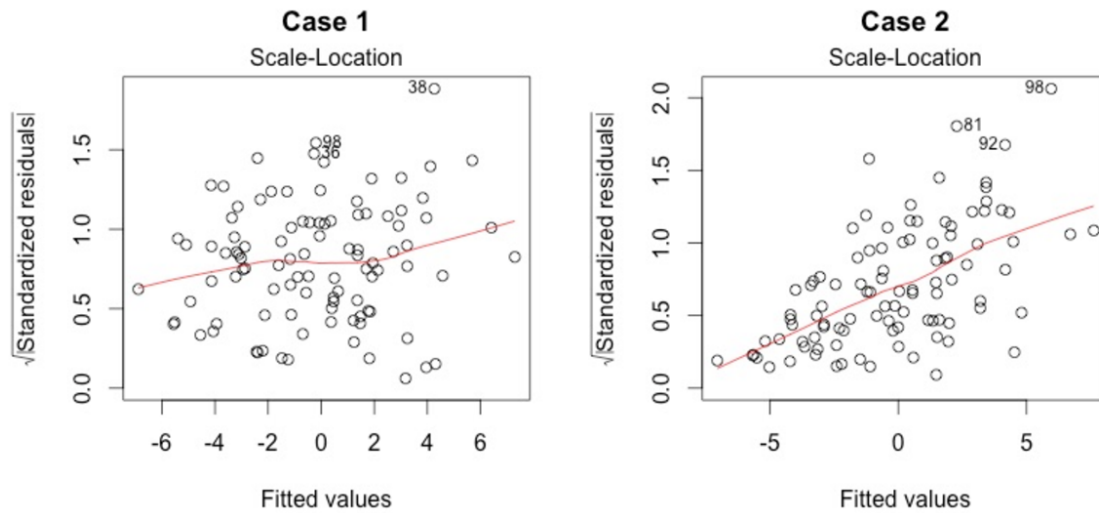


Figure 3: Residual plots.

horizontal line with equally (randomly) spread points. From the figure above in Case 1, the residuals appear randomly spread. Whereas, in Case 2, the residuals begin to spread wider along the x-axis as it passes around 5. Because the residuals spread wider and wider, the red smooth line is not horizontal and shows a steep angle in Case 2.

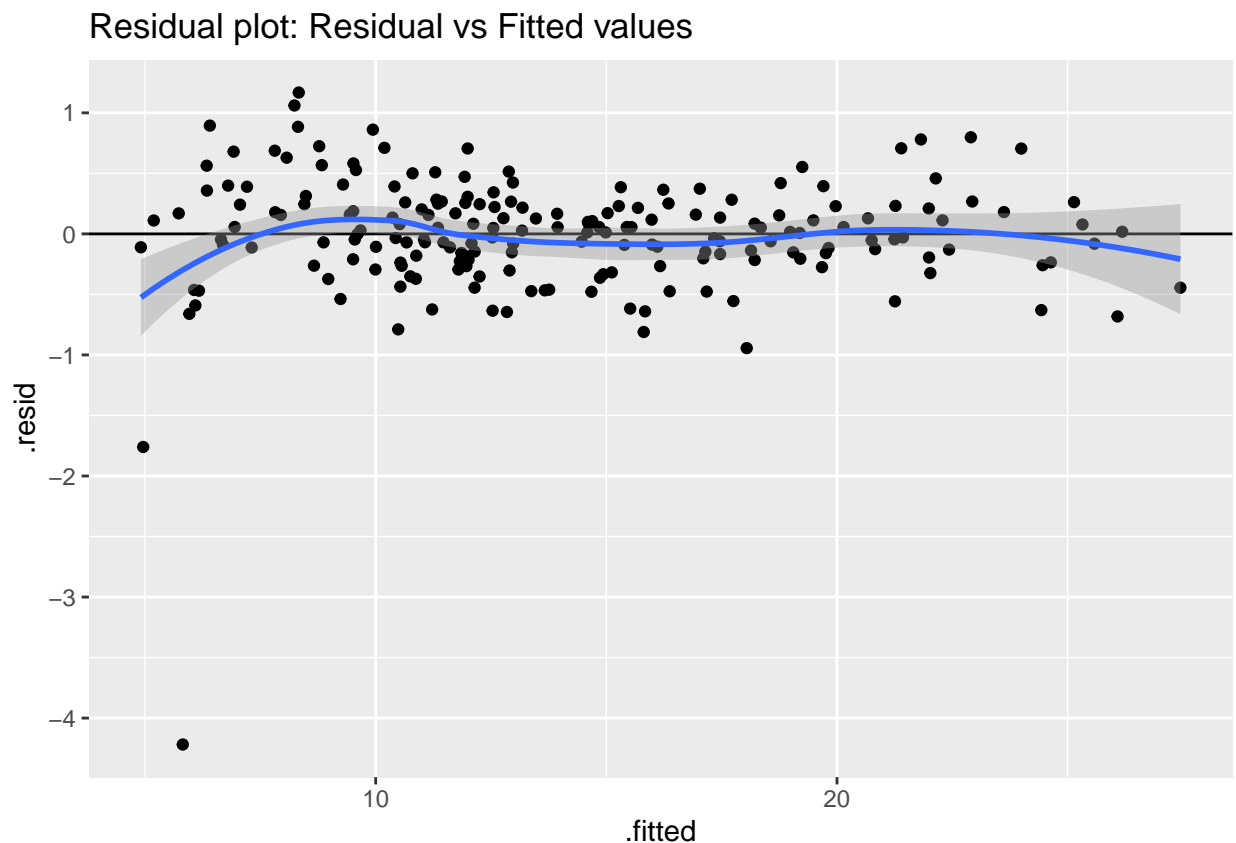
```
library(ggplot2)
Advertising=read.table("Advertising.txt", header = TRUE, sep = "\t" )
cubic<-lm(sale~tv+I(tv^2)+I(tv^3)+radio+tv*radio, data=Advertising)
summary(cubic)
```

```
##
## Call:
## lm(formula = sale ~ tv + I(tv^2) + I(tv^3) + radio + tv * radio,
##     data = Advertising)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.2184 -0.2106  0.0223  0.2454  1.1677
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  4.061e+00  1.871e-01  21.709  < 2e-16 ***
## tv           8.998e-02  4.193e-03  21.458  < 2e-16 ***
## I(tv^2)      -4.327e-04  3.180e-05 -13.604  < 2e-16 ***
## I(tv^3)       7.278e-07  7.058e-08  10.312  < 2e-16 ***
## radio        4.206e-02  4.801e-03   8.761  9.63e-16 ***
## tv:radio      1.044e-03  2.811e-05  37.129  < 2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.5026 on 194 degrees of freedom
## Multiple R-squared:  0.991, Adjusted R-squared:  0.9907
## F-statistic: 4250 on 5 and 194 DF, p-value: < 2.2e-16
```

```
#residuals plot
ggplot(cubic, aes(x=.fitted, y=.resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  geom_smooth()+
  ggtitle("Residual plot: Residual vs Fitted values")
```

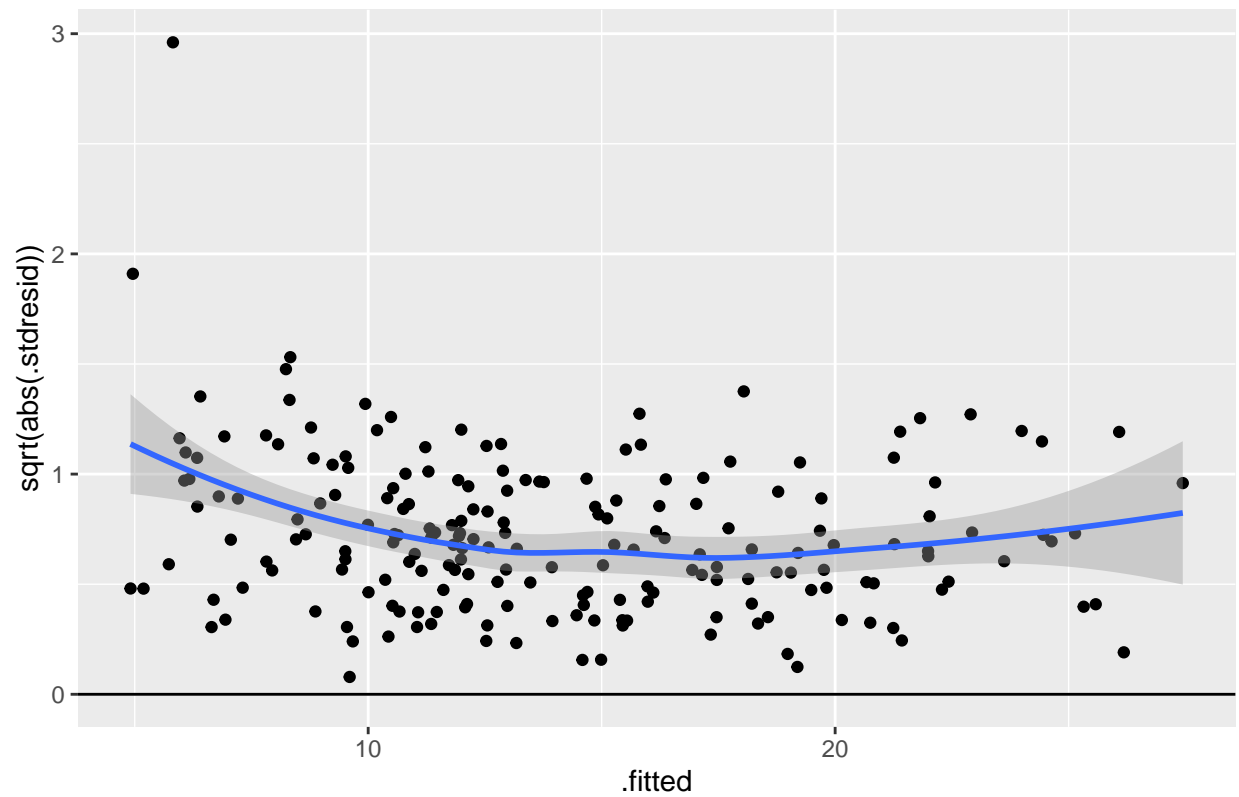
```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```



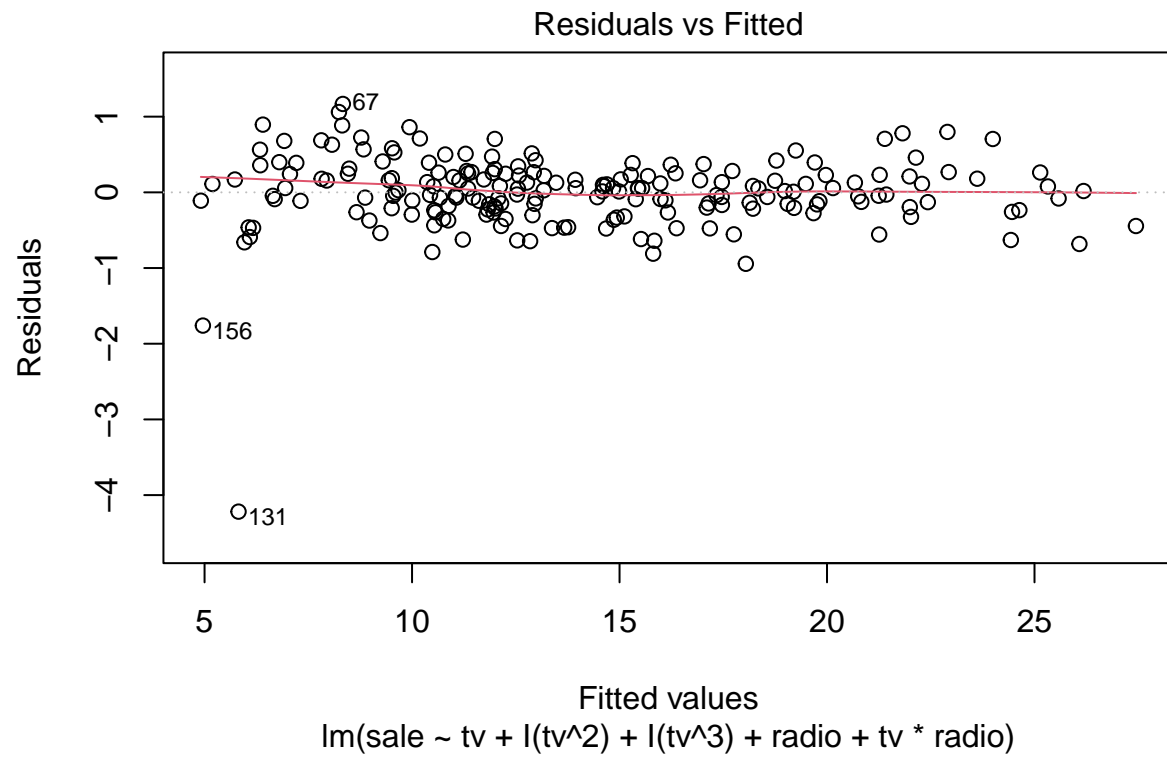
```
#a scale location plot
ggplot(cubic, aes(x=.fitted, y=sqrt(abs(.stdresid)))) +
  geom_point() +
  geom_hline(yintercept = 0) +
  geom_smooth()+
  ggtitle("Scale-Location plot : Standardized Residual vs Fitted values")
```

```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```

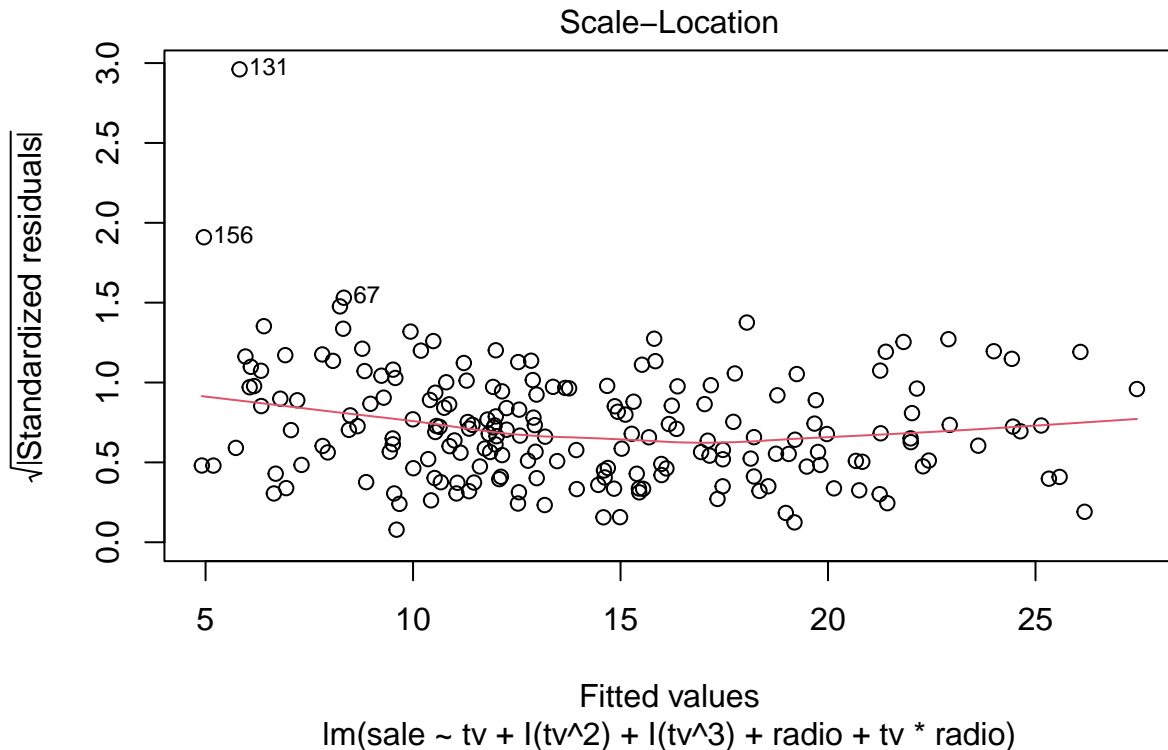
Scale–Location plot : Standardized Residual vs Fitted values



```
#optional graphs for residual plots and a scale location plot  
plot(cubic, which=1) #residuals plot
```



```
plot(cubic, which=3) #a scale location plot
```



R functions `ggplot(model, aes(x=..., y=...))`: mapping 2 variables on a and y axis `.fitted` : Fitted values of model `.resid` : Residuals `.stdresid` : Standardised Residuals `geom_hline(yintercept = 0)`:add a horizontal line `geom_point()` : add a layer of points to the plot `ggtitle()` ; add a title to the plot

**From the Advertising example**, the output displays the residual plot and Scale-Location plot that result from the cubic model. In our case, the residuals tend to form a horizontal band-indicates that the plot does not provide evidence to suggest that heteroscedasticity exists.

A more formal, mathematical way of detecting heteroscedasticity is what is known as **the Breusch-Pagan test**. It involves using a variance function and using a  $\chi^2$  test to test

$H_0$  : heteroscedasticity is not present (homoscedasticity)

$H_a$  : heteroscedasticity is present

or

$H_0 : \sigma_1^2 = \sigma_2^2 = \dots = \sigma_n^2$

$H_0$  : at least  $\sigma_i^2$  is different from the others  $i = 1, 2, \dots, n$

$$\chi^2 = nR^2 \sim \chi_{p-1}^2$$

where

$n$  = sample size

$R^2$  = coefficient determination

$p$  = number of regression coefficients

```
library(lmtest)
```

```
## Warning: package 'lmtest' was built under R version 4.2.2
```

```
## Loading required package: zoo
```

```
##
```

```
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
```

```
##
```

```
## as.Date, as.Date.numeric
```

```
Advertising=read.table("Advertising.txt", header = TRUE, sep = "\t" )
cubic<-lm(sale~tv+I(tv^2)+I(tv^3)+radio+tv*radio, data=Advertising)
bptest(cubic)
```

```
##
```

```
## studentized Breusch-Pagan test
```

```
##
```

```
## data: cubic
```

```
## BP = 22.934, df = 5, p-value = 0.0003476
```

```
morepower<-lm(sale~tv+I(tv^2)+I(tv^3)+I(tv^4)+I(tv^5)+I(tv^6)+I(tv^7)+
               I(tv^8)+I(tv^9)+I(tv^10)+I(tv^11)+radio+tv*radio,
               data=Advertising)
summary(morepower)
```

```
##
```

```
## Call:
```

```
## lm(formula = sale ~ tv + I(tv^2) + I(tv^3) + I(tv^4) + I(tv^5) +
##      I(tv^6) + I(tv^7) + I(tv^8) + I(tv^9) + I(tv^10) + I(tv^11) +
##      radio + tv * radio, data = Advertising)
```

```
##
```

```
## Residuals:
```

```
##      Min       1Q   Median       3Q      Max
## -0.80872 -0.19715  0.02148  0.16548  0.74259
```

```
##
```

```
## Coefficients:
```

```
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.334e-01  3.325e-01  -0.702 0.483684
## tv           8.353e-01  7.962e-02  10.491 < 2e-16 ***
## I(tv^2)      -4.639e-02  7.054e-03  -6.577 4.73e-10 ***
## I(tv^3)       1.504e-03  2.985e-04   5.039 1.10e-06 ***
## I(tv^4)      -2.974e-05  7.074e-06  -4.205 4.06e-05 ***
## I(tv^5)       3.780e-07  1.025e-07   3.687 0.000298 ***
## I(tv^6)      -3.180e-09  9.533e-10  -3.335 0.001028 **
## I(tv^7)       1.786e-11  5.799e-12   3.081 0.002378 **
## I(tv^8)      -6.623e-14  2.295e-14  -2.886 0.004366 **
## I(tv^9)       1.554e-16  5.692e-17   2.730 0.006950 **
## I(tv^10)     -2.088e-19  8.032e-20  -2.600 0.010075 *
```



```
## I(tv^11)      1.224e-22  4.919e-23   2.489 0.013691 *
## radio         4.585e-02  3.066e-03  14.955 < 2e-16 ***
## tv:radio      1.023e-03  1.794e-05  57.055 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3179 on 186 degrees of freedom
## Multiple R-squared:  0.9965, Adjusted R-squared:  0.9963
## F-statistic: 4110 on 13 and 186 DF,  p-value: < 2.2e-16
```

```
bptest(morepower)
```

```
##
## studentized Breusch-Pagan test
##
## data: morepower
## BP = 15.005, df = 13, p-value = 0.307
```

```
morepower1<-lm(sale~poly(tv,11,raw = T)+radio+tv*radio, data=Advertising)
summary(morepower1)
```

```
##
## Call:
## lm(formula = sale ~ poly(tv, 11, raw = T) + radio + tv * radio,
##     data = Advertising)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.80872 -0.19715  0.02148  0.16548  0.74259
##
## Coefficients: (1 not defined because of singularities)
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -2.334e-01  3.325e-01  -0.702 0.483684
## poly(tv, 11, raw = T)1  8.353e-01  7.962e-02  10.491 < 2e-16 ***
## poly(tv, 11, raw = T)2 -4.639e-02  7.054e-03  -6.577 4.73e-10 ***
## poly(tv, 11, raw = T)3  1.504e-03  2.985e-04   5.039 1.10e-06 ***
## poly(tv, 11, raw = T)4 -2.974e-05  7.074e-06  -4.205 4.06e-05 ***
## poly(tv, 11, raw = T)5  3.780e-07  1.025e-07   3.687 0.000298 ***
## poly(tv, 11, raw = T)6 -3.180e-09  9.533e-10  -3.335 0.001028 **
## poly(tv, 11, raw = T)7  1.786e-11  5.799e-12   3.081 0.002378 **
## poly(tv, 11, raw = T)8 -6.623e-14  2.295e-14  -2.886 0.004366 **
## poly(tv, 11, raw = T)9  1.554e-16  5.692e-17   2.730 0.006950 **
## poly(tv, 11, raw = T)10 -2.088e-19  8.032e-20  -2.600 0.010075 *
## poly(tv, 11, raw = T)11 1.224e-22  4.919e-23   2.489 0.013691 *
## radio          4.585e-02  3.066e-03  14.955 < 2e-16 ***
## tv              NA          NA        NA        NA
## radio:tv       1.023e-03  1.794e-05  57.055 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.3179 on 186 degrees of freedom
## Multiple R-squared:  0.9965, Adjusted R-squared:  0.9963
## F-statistic: 4110 on 13 and 186 DF,  p-value: < 2.2e-16
```

*R function bptest(): to perform the Breusch-Pagan test*

**From the Advertising example**, the output displays the Breusch-Pagan test that results from the cubic model. The p-value = 0.00034 < 0.05, indicating that we do reject the null hypothesis. Therefore, the test provides evidence to suggest that heteroscedasticity does exist. However, a model with more power on tv (power of 11) shows evidence to suggest that heteroscedasticity does not exist.

## Inclass Practice Problem 17

From the clerical staff work hours, use residual plots to conduct a residual analysis of the data. Check Equal Variance Assumption by graphs and the Breusch-Pagan test. If you detect a trend, how would you like to transform the predictors in the model?

*R function*

*geom\_smooth() : add the regression slope*

## 4. Normality Assumption

The multiple linear regression analysis requires that the errors between observed and predicted values (i.e., the residuals of the regression) should be normally distributed. This assumption may be checked by looking at a histogram, a normal probability plot or a Q-Q-Plot.

If the distribution is normal, the points on such a plot (Probability Plot or Q-Q-Plot) should fall close to the diagonal reference line. A bow-shaped pattern of deviations from the diagonal indicates that the residuals have excessive skewness. An S-shaped pattern of deviations indicates that the residuals have excessive kurtosis, i.e., there are either too many or too few large errors in both directions. Sometimes the problem is revealed to be that there are a few data points on one or both ends that deviate significantly from the reference line ("outliers"), in which case they should get close attention.

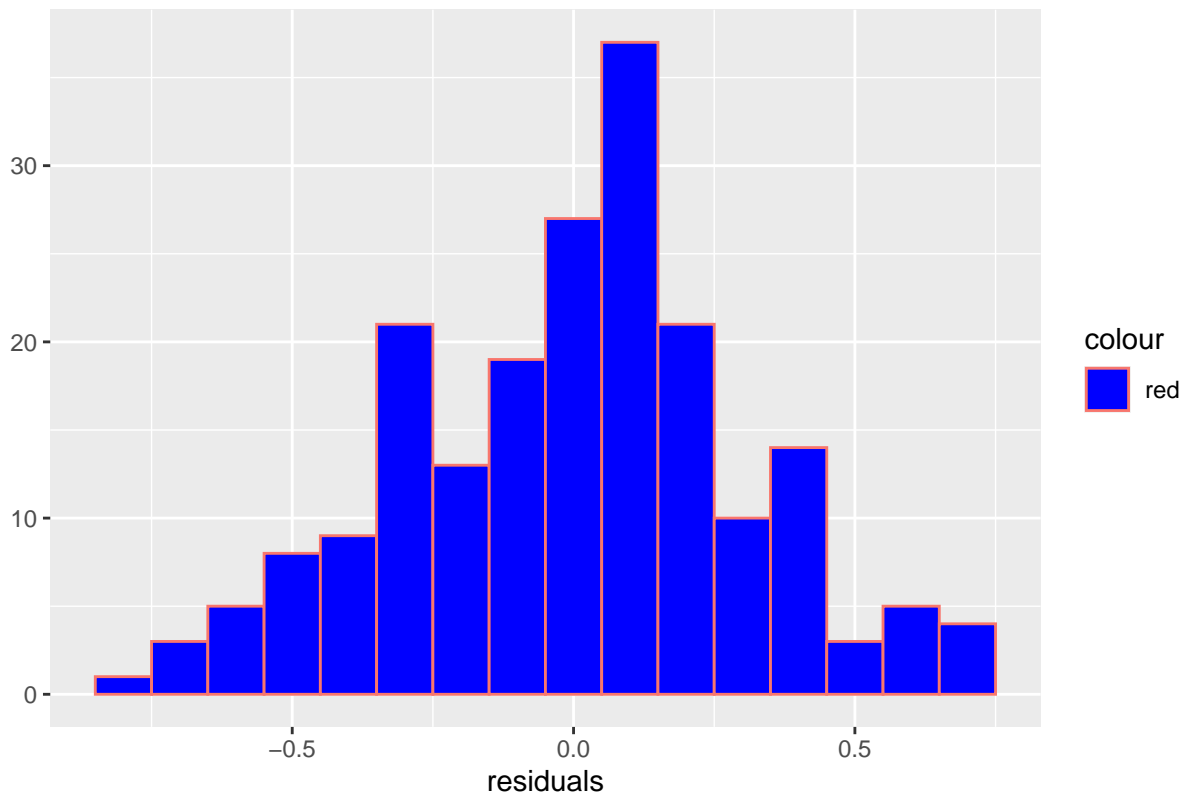
There are also a variety of statistical tests for normality, including the Kolmogorov-Smirnov test and the Shapiro-Wilk test.

$H_0$  : the sample data are significantly normally distributed

$H_a$  : the sample data are not significantly normally distributed

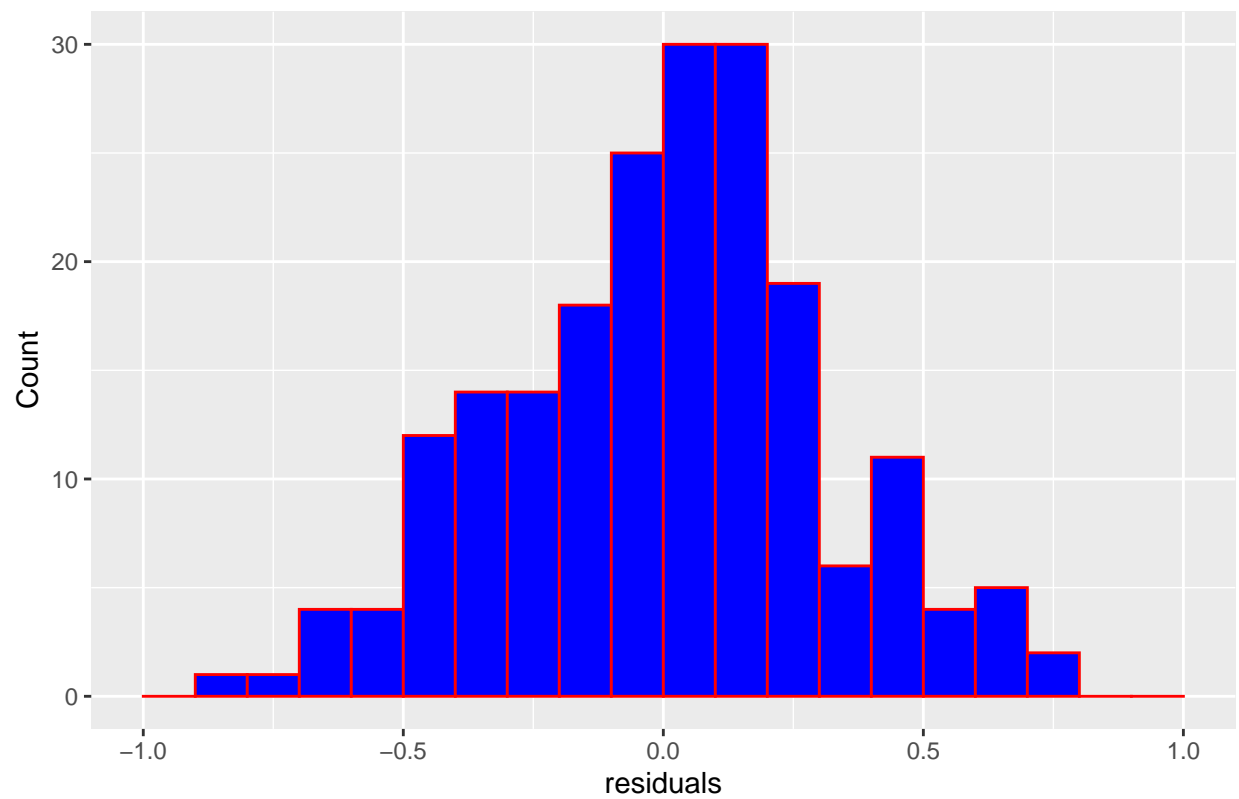
```
library(ggplot2)
Advertising=read.table("Advertising.txt", header = TRUE, sep = "\t" )
#logmodel<-lm(sale~log(tv)+radio+tv*radio, data=Advertising)
morepower<-lm(sale~poly(tv,11,row = T)+radio+tv*radio, data=Advertising)
#option 1 (histogram)
qplot(residuals(morepower),
      geom="histogram",
      binwidth = 0.1,
      main = "Histogram of residuals",
      xlab = "residuals", color="red",
      fill=I("blue"))
```

Histogram of residuals

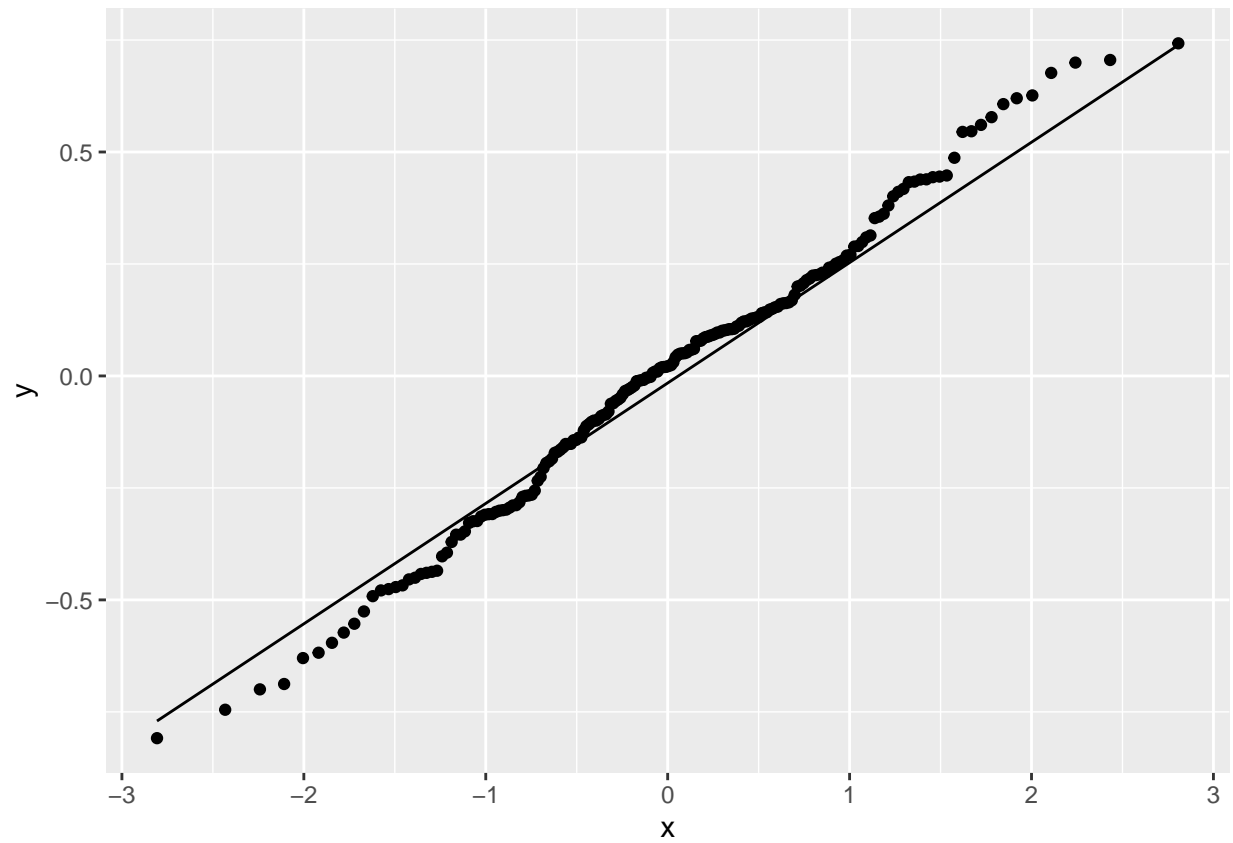


```
#option 2 (histogram)
ggplot(data=Advertising, aes(residuals(morepower))) +
  geom_histogram(breaks = seq(-1,1,by=0.1), col="red", fill="blue") +
  labs(title="Histogram for residuals") +
  labs(x="residuals", y="Count")
```

Histogram for residuals

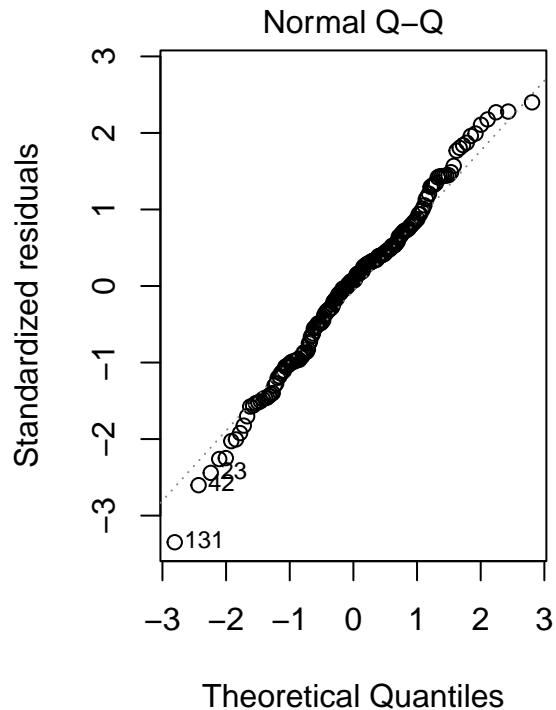
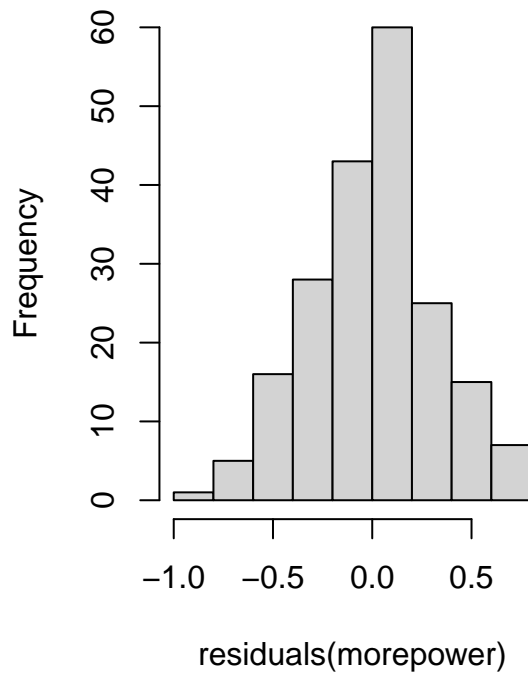


```
#normal QQ plot  
ggplot(Advertising, aes(sample=morepower$residuals)) +  
  stat_qq() +  
  stat_qq_line()
```



```
#optional histogram  
par(mfrow=c(1,2))  
hist(residuals(morepower))  
plot(morepower, which=2) #a Normal plot
```

## Histogram of residuals(morepower)



```
#Testing for Normality
shapiro.test(residuals(morepower))
```

```
##
## Shapiro-Wilk normality test
##
## data: residuals(morepower)
## W = 0.99171, p-value = 0.3129
```

*R functions `stat_qq()` : produce quantile-quantile plots `stat_qq_line()` : compute the slope and intercept of the line connecting the points at specified quartiles of the theoretical and sample distributions*

`geom_qq_line` and `stat_qq_line` compute the slope and intercept of the line connecting the points at specified quartiles of the theoretical and sample distributions. The outputs show that the residual data have normal distribution (from histogram and Q-Q plot). Moreover, Shapiro-Wilk normality test also confirms that the residuals are normally distributed as the  $p\text{-value} = 0.3129 > 0.05$ .

## Inclass Practice Problem 18

From the clerical staff work hours, use residual plots to conduct a residual analysis of the data. **Check Normality Assumption by graphs and the Shapiro-Wilk normality test.** If you detect a trend, how would you like to transform the predictors in the model?

## 5. Multicollinearity

Often, two or more of the independent variables used in the model for  $E(Y)$  provide redundant information. That is, the independent variables will be correlated with each other. For example, suppose we want to construct a model to predict the gasoline mileage rating,  $Y$ , of a truck as a function of its load,  $X_1$ , and the horsepower,  $X_2$ , of its engine. In general, you would expect heavier loads to require greater horsepower and to result in lower mileage ratings. Thus, although both  $X_1$  and  $X_2$  contribute information for the prediction of mileage rating, some of the information is overlapping, because  $X_1$  and  $X_2$  are (linearly) correlated. When the independent variables are (linearly) correlated, we say that multicollinearity exists. In practice, it is not uncommon to observe correlations among the independent variables. However, a few problems arise when serious multicollinearity is present in the regression analysis.

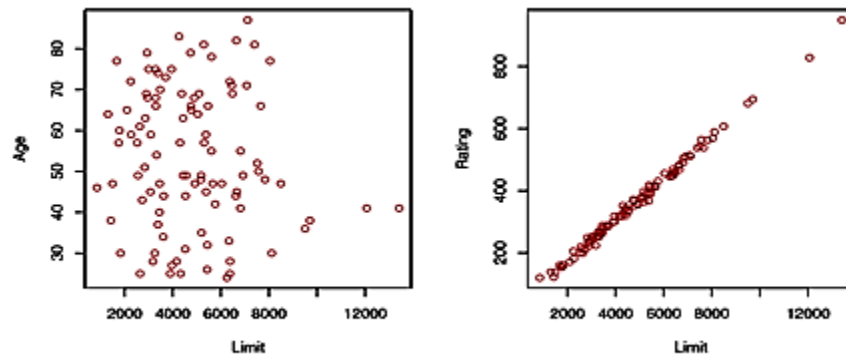


Figure 4: The scatter plot shows multicollinearity between Rating and Limit

In the left-hand panel of Figure 3, the two predictors limit and age appear to have no obvious relationship. In contrast, in the right-hand panel of Figure 3, the predictors limit and rating are very highly linearly correlated with each other, and we say that they are collinear.

### What Problems Do Multicollinearity Cause?

Multicollinearity causes the following two basic types of problems:

1. 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.
2. Multicollinearity reduces the precision of the estimate 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.

### Testing for Multicollinearity with Variance Inflation Factors (VIF)

If you can identify which variables are affected by multicollinearity and the strength of the correlation, you're well on your way to determining whether you need to fix it. Fortunately, there is a very simple test to assess multicollinearity in your regression model which is called **"The variance inflation factor (VIF)"**

The variance inflation factor (VIF)

VIF identifies correlation between independent variables and the strength of that correlation. It can be computed using the formula

$$VIF(\hat{\beta}_j) = \frac{1}{1 - R_{X_j|X_{-j}}^2}$$

where  $R_{X_j|X_{-j}}^2$  is the  $R^2$  from a regression of  $X_j$  onto all of the other predictors. If  $R_{X_j|X_{-j}}^2$  is close to one, then collinearity is present, and so the VIF will be large.

Statistical software calculates a VIF for each independent variable. Value of VIFs start at 1 and have no upper limit and can be interpreted as following;

\*VIFs=1 indicates that there is no collinearity between this independent variable and any others.

\*1<= VIFs <=5 suggest that there is a moderate collinearity, but it is not severe enough to warrant corrective measures.

VIFs > 5 or 10 represent critical levels of multicollinearity where the coefficients are poorly estimated, and the p-values are questionable.

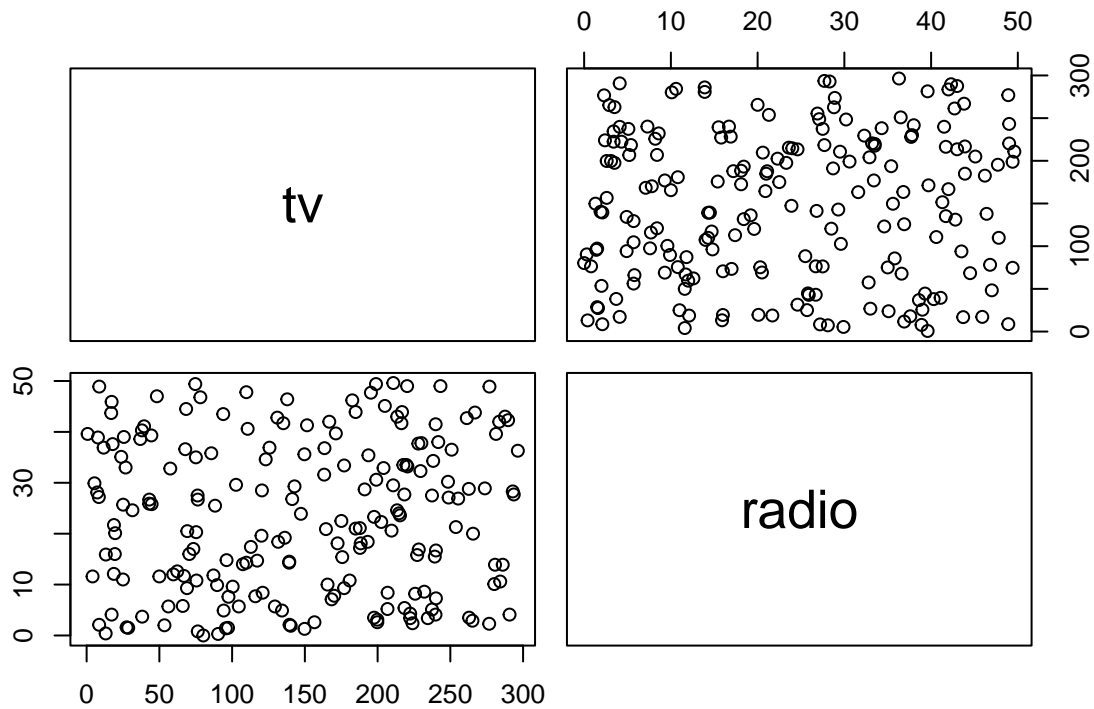
We use VIFs to identify correlations between variables and determine the strength of the relationships.

**Attention:** *When the high VIFs are caused by the inclusion of powers or products of other variables, you can Safely Ignore Multicollinearity*

From Advertising data, check Multicollinearity Assumption by using scatter plots and VIF

```
library(mctest) #for VIF
Advertising=read.table("Advertising.txt", header = TRUE, sep = "\t" )
#logmodel<-lm(sale~log(tv)+radio+tv*radio, data=Advertising)
firstordermodel<-lm(sale~tv+radio, data=Advertising)
pairs(~tv+radio, data=Advertising)
```





```
#Calculate VIF for multicollinearity model
#option 1
```

```
imcdiag(firstordermodel, method="VIF")
```

```
##
## Call:
## imcdiag(mod = firstordermodel, method = "VIF")
##
##
## VIF Multicollinearity Diagnostics
##
##      VIF detection
## tv      1.003      0
## radio 1.003      0
##
## NOTE: VIF Method Failed to detect multicollinearity
##
##
## 0 --> COLLINEARITY is not detected by the test
##
## =====
```

```
#option 2
library(car)
```

```
## Warning: package 'car' was built under R version 4.2.1
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 4.2.1
```

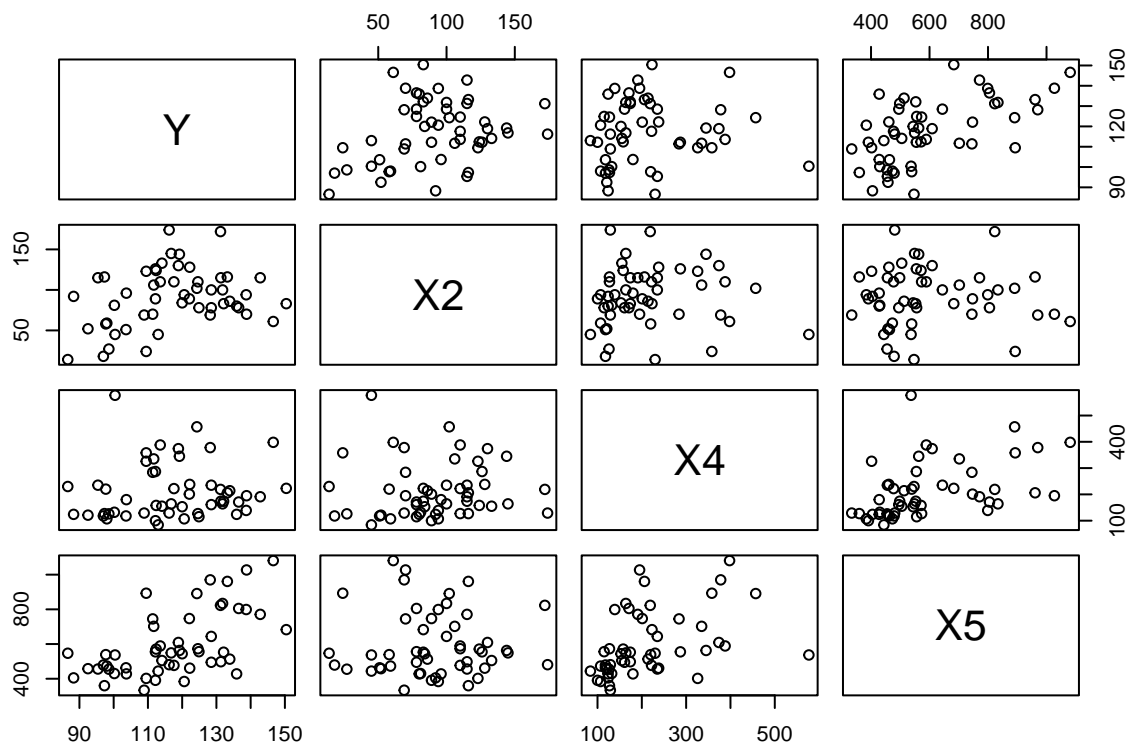
```
vif(firstordermodel)
```

```
##          tv      radio  
## 1.003013 1.003013
```

From the output, you can see that the  $VIF_{TV} = VIF_{Radio} = 1.003013$ , which suggests that there is no correlation between these predictors.

From the clerical staff work hours, checking for Multicollinearity by scatter plots between independent predictors and VIF test. Note! consider only main effect predictors

```
library(mctest) #for VIF  
workhours=read.csv("CLERICAL.csv",header = TRUE)  
#improvemodel<-lm(Y~X2+I(X2^2)+X4+X5,data=workhours)  
  
firstordermodel<-lm(Y~X2+X4+X5,data=workhours)  
pairs(~Y+X2+X4+X5,data=workhours)
```



```

#Calculate VIF for multicollinearity model
#option 1

imcdiag(firstordermodel, method="VIF")

##
## Call:
## imcdiag(mod = firstordermodel, method = "VIF")
##
##
## VIF Multicollinearity Diagnostics
##
## VIF detection
## X2 1.0027      0
## X4 1.2469      0
## X5 1.2455      0
##
## NOTE: VIF Method Failed to detect multicollinearity
##
##
## 0 --> COLLINEARITY is not detected by the test
##
## =====

#option 2
library(car)
vif(firstordermodel)

## X2 X4 X5
## 1.002657 1.246889 1.245509

#vif(improvemodel)

```

*R functions imcdiag(x=y,method="VIF") : detects the existence of multicollinearity due to independent variables*

## Inclass practice Problem 19

From the credit card example, check for Multicollinearity by scatter plots between independent predictors and VIF test . Note! consider only main effect predictors

In the credit data example, a regression of balance on Age, Rating, and Limit indicates that the predictors have VIF values of 2.776906 , 230.869514, 1.039696, 229.238479, 1.439007, and 1.009064. As we suspected, there is considerable multicollinearity in the data! When faced with the problem of multicollinearity.

**There are two simple solutions.**

**The first solution** is to drop one of the problematic variables from the regression model. This can usually be done without much compromise to the regression model, since the presence of multicollinearity implies that the information that this variable provides about the response is redundant in the presence of the other variables.

**The second solution** is to combine the collinear variables together into a single predictor. For instance, we might take the average of standardized versions of Limit and Rating in order to create a new variable that measures credit worthiness.

## Inclass Practice Problem 20

From the credit card example, after dropping the Limit variable, find the best model and check for Multicollinearity by scatter plots between independent predictors and VIF test. Note! consider only main effect predictors

### 6. Outlier (The Effect on Individual Cases)

An outlying case is defined as a particular observation ( $Y, X_1, X_2, \dots, X_p$ ) that differs from the majority of the cases in the data set. There are several ways we can find and evaluate outlier or influential points.

#### 1. Residuals vs Leverage plot

This plot helps us to find influential cases (i.e., subjects) if any. Not all outliers are influential in linear regression analysis (whatever outliers mean). Even though data have extreme values, they might not be influential to determine a regression line. That means, the results wouldn't be much different if we either include or exclude them from analysis. They follow the trend in the majority of cases and they don't really matter; they are not influential. On the other hand, some cases could be very influential even if they look to be within a reasonable range of the values. They could be extreme cases against a regression line and can alter the results if we exclude them from analysis. Another way to put it is that they don't get along with the trend in the majority of the cases.

Unlike the other plots, this time patterns are not relevant. We watch out for outlying values at the upper right corner or at the lower right corner. Those spots are the places where cases can be influential against a regression line. Look for cases outside of a dashed line, Cook's distance. When cases are outside of the Cook's distance (meaning they have high Cook's distance scores), the cases are influential to the regression results. The regression results will be altered if we exclude those cases.

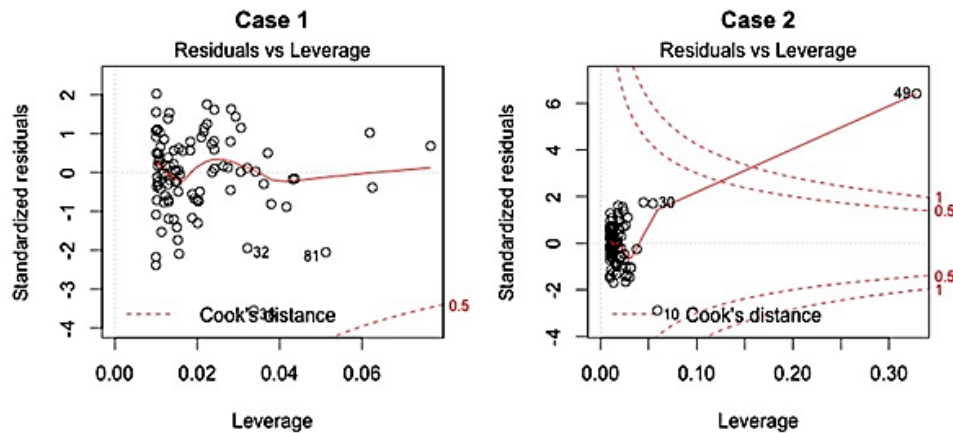
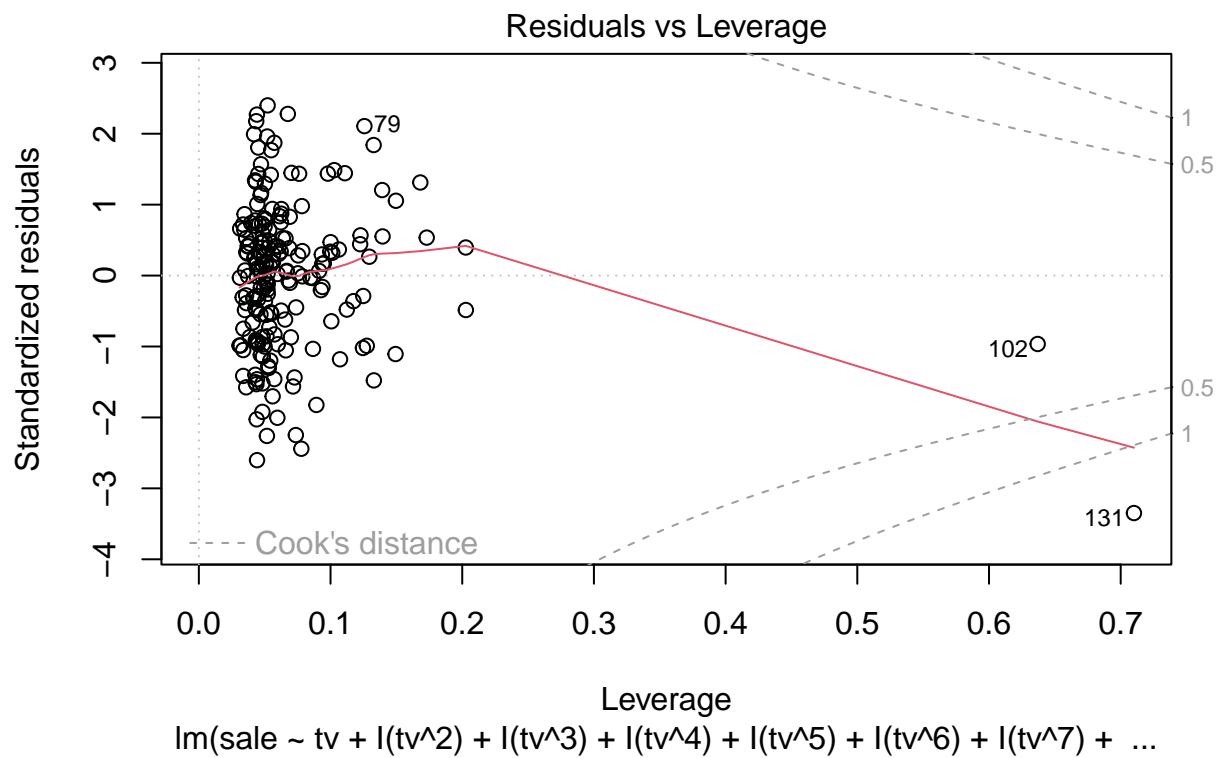


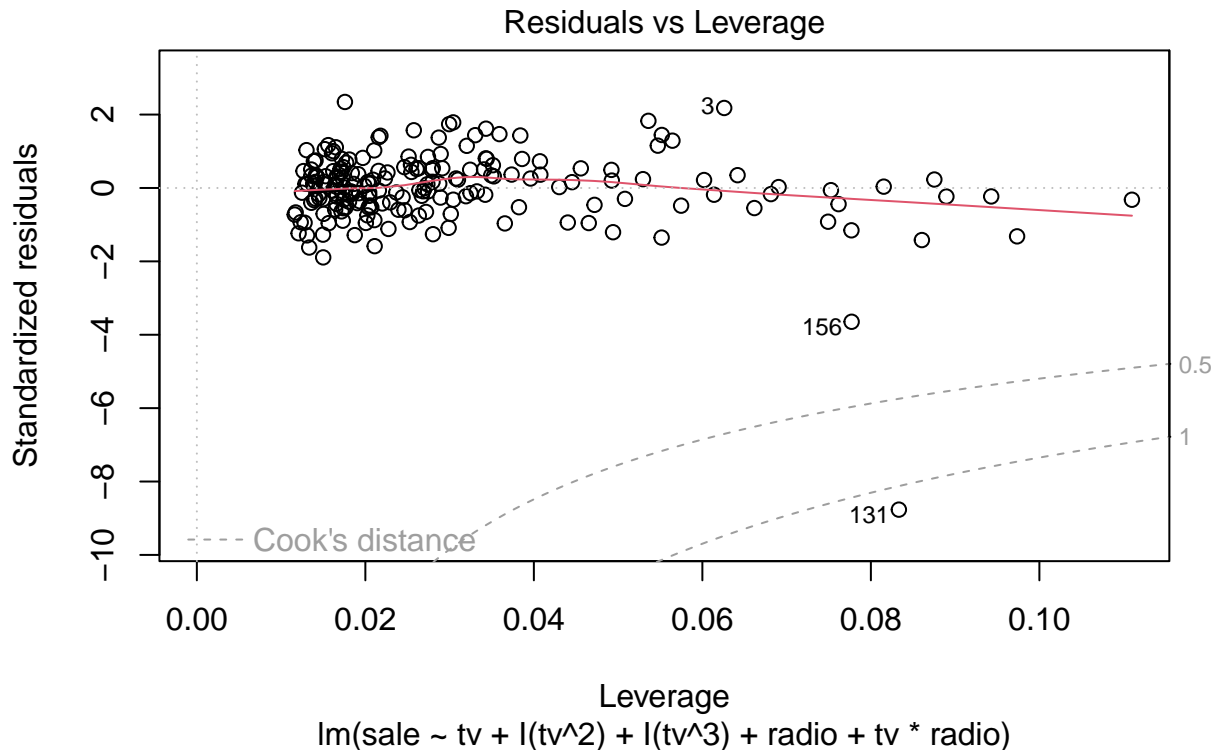
Figure 5: Residuals vs Leverage plot for detecting outliers or influential points

Case 1 is the typical look when there is no influential case, or cases. You can barely see Cook's distance lines (a red dashed line) because all cases are well inside of the Cook's distance lines. In Case 2, a case is far beyond the Cook's distance lines (the other residuals appear clustered on the left because the second plot is scaled to show larger area than the first plot). The plot identified the influential observation as #49.

```
Advertising=read.table("Advertising.txt", header = TRUE, sep = "\t" )
morepower<-lm(sale~tv+I(tv^2)+I(tv^3)+I(tv^4)+I(tv^5)+I(tv^6)+I(tv^7)+I(tv^8)+
              I(tv^9)+I(tv^10)+I(tv^11)+radio+tv*radio, data=Advertising)
plot(morepower,which=5)
```



```
cubic<-lm(sale~tv+I(tv^2)+I(tv^3)+radio+tv*radio, data=Advertising)
plot(cubic,which=5)
```



From the Advertising example (both cubic and morepower models), you can see that data point 131 is an influential case.

## 2. Cook's Distance

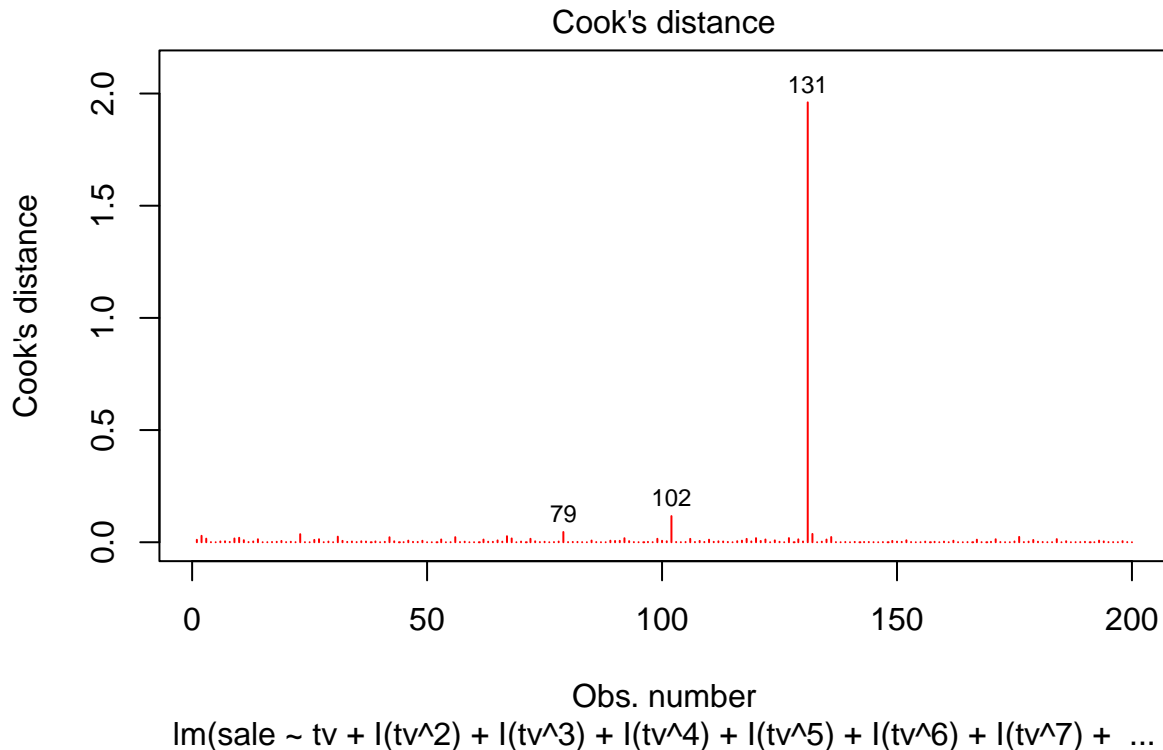
A measure of the overall influence an outlying observation has on the estimated coefficients was proposed by R. D. Cook (1979). The **Cook's distance**  $D_i$  measures the effect of deleting a given observation, and is interpreted for the  $i$ th observation as follows:

A large value of  $D_i$  indicates that the observed  $Y_i$  value has strong influence on the estimated coefficients (since the residual, the leverage, or both will be large). A general rule of thumb is that observations with a Cook's  $D$  of more than 3 times the mean,  $\mu$ , is a possible outlier. An alternative interpretation is to investigate any point over  $4/n$ , where  $n$  is the number of observations. Other authors suggest that any "large"  $D_i$  should be investigated. How large is "too large"? The consensus seems to be that a  $D_i$  value of more than 1 indicates an influential value, but you may want to look at values above 0.5. Like the other numerical measures of influence, options for calculating Cook's distance are available in most statistical software packages.

```
Advertising=read.table("Advertising.txt", header = TRUE, sep = "\t" )
morepower<-lm(sale~tv+I(tv^2)+I(tv^3)+I(tv^4)+I(tv^5)+I(tv^6)+I(tv^7)+I(tv^8)+I(tv^9)+I(tv^10)+I(tv^11))
Advertising[cooks.distance(morepower)>0.5,] #have Cook statistics larger than 0.5

##      No  tv radio newspaper sale
## 131 131 0.7  39.6      8.7  1.6

plot(morepower,pch=18,col="red",which=c(4))
```



### 3. Leverage points

Points that fall horizontally far from the line are points of high leverage; these points can strongly influence the slope of the least squares line. If one of these high leverage points does appear to actually invoke its influence on the slope of the line then we call it an influential point. Usually we can say a point is influential if, had we plotted the line without it, the influential point would have been unusually far from the least squares line.

Leverage values for multiple regression models are extremely difficult to calculate without the aid of a computer. Fortunately, most of the statistical software packages have options that give the leverage associated with each observation.

A good rule of thumb to identify an observation  $y_i$  as influential if its leverage value  $h_i$  is

$$h_i > \frac{2p}{n} \text{ or } \frac{3p}{n}$$

where  $h_i$  is the leverage for the  $i$ th observation

$p$  = the number of predictors

$n$  = the number of the sample size

```
morepower<-lm(sale~tv+I(tv^2)+I(tv^3)+I(tv^4)+I(tv^5)+I(tv^6)+I(tv^7)+I(tv^8)+I(tv^9)+I(tv^10)+I(tv^11))
lev=hatvalues(morepower)
p = length(coef(morepower))
n = nrow(Advertising)
outlier2p = lev[lev>(2*p/n)]
```

```

outlier3p = lev[lev>(3*p/n)]
print("h_I>2p/n, outliers are")

## [1] "h_I>2p/n, outliers are"

print(outlier2p)

##          31          36          43          99          102          131          156          184
## 0.1681832 0.1730329 0.2027613 0.1493568 0.6369562 0.7100702 0.2026995 0.1496396

print("h_I>3p/n, outliers are")

## [1] "h_I>3p/n, outliers are"

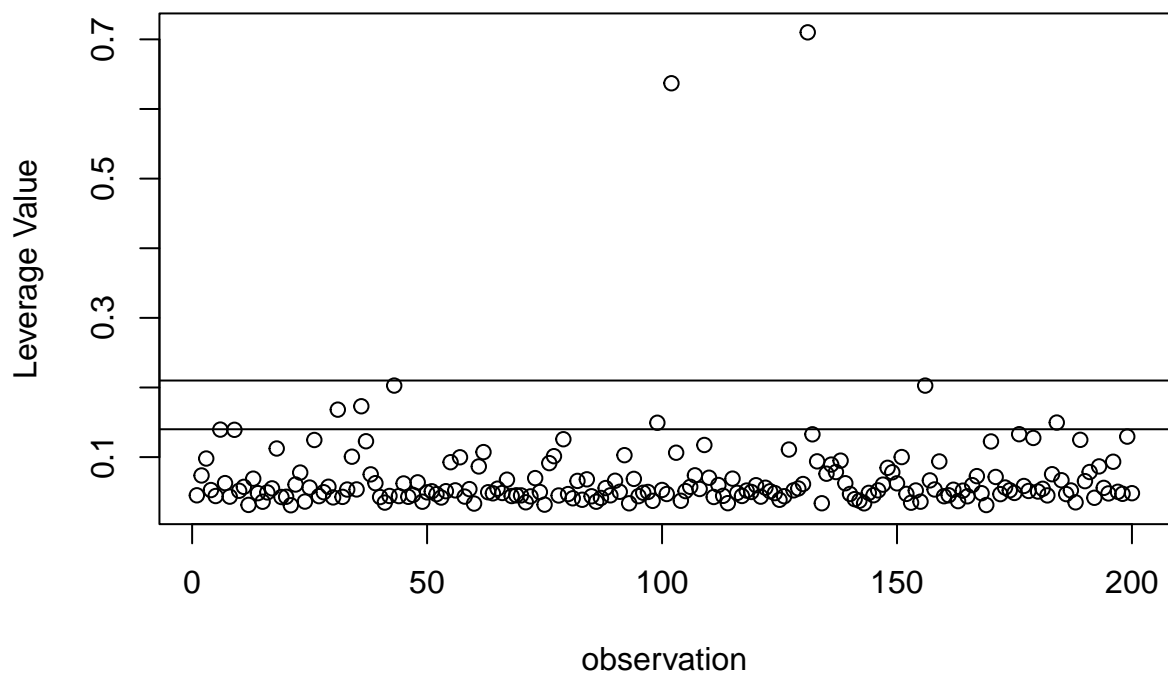
print(outlier3p)

##          102          131
## 0.6369562 0.7100702

plot(rownames(Advertising),lev, main = "Leverage in Advertising Dataset", xlab="observation",
      ylab = "Leverage Value")
abline(h = 2 *p/n, lty = 1)
abline(h = 3 *p/n, lty = 1)

```

## Leverage in Advertising Dataset





Big question now is, once we identify an outlier, or influential observation, **what do we do with it?**

For a good understanding of the regression model, if we have some outliers or influential points, we may want to...

-See what happens when we exclude these from the model as an outlier has occurred due to an error in data collection or recoding, then the solution is to simply remove the observation.

-Investigate these cases separately as it may happen that we mistyped.

## Inclass Practice Problem 21

From the clerical staff work hours, using residual plots to conduct a residual analysis of the data. Check any potential outliers.

## Inclass Practice Problem 22

Check assumptions for the model below to predict executive salary (Y)

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_2 + \beta_4 X_3 + \beta_5 X_4 + \beta_6 X_5 + \beta_7 X_3 * X_4 + \epsilon$$

## How to deal with Heteroscedasticity?

### 1. Log-transformation

As the simple solver, log-transformation can be one of the candidates. When  $\log()$  takes the numbers, the difference between big and small numbers relatively becomes small.

### 2. Box-Cox transformations.

This will help the 'transformed' data to have equal variance, and, as usually happens, will also make the transformed data to follow a normal distribution.

### 3. Weight Least Squares Regression

If one wants to correct for heteroskedasticity by using a fully efficient estimator rather than accepting inefficient OLS and correcting the standard errors, the appropriate estimator is weight least squares, which is an application of the more general concept of generalized least squares.

## Box-Cox Transformations (Transformations for Nonnormality and Heteroscedasticity)

Unequal variances and nonnormality of the error terms frequently appear together. To remedy these departures from a linear regression model, **we need a transformation on Y**, since the shapes and spreads of the distributions of Y need to be changed. Such a transformation on Y may also at the same time help to linearize a curvilinear regression relation. At other times, a simultaneous transformation on X may be needed to obtain or maintain a linear regression relation.

**The Box Cox transformation** is named after statistician George Box and Sir David Roxbee Cox who collaborated on a 1964 paper and developed the technique.

It is often difficult to determine which transformation on response variable (Y) to use, the Box-Cox method considers a family of power transformations on **strictly positive response variables (all data is positive and greater than 0)** such that  $Y^{(\lambda)} = Y^\lambda$ , where  $\lambda$  is a parameter to be determined using the data, varies from -5 to 5. Note that this family encompasses the following simple transformations:

Common Box-Cox Transformations:

$$\lambda = 2, Y' = Y^2$$

$$\lambda = 0.5, Y' = \sqrt{Y}$$

$$\lambda = 0, Y' = \log_e Y \text{ (by definition)}$$

$$\lambda = -0.5, Y' = 1/\sqrt{Y}$$

$$\lambda = -1.0, Y' = 1/Y$$

Note! the transformation for 0 is  $\log_e(Y)$ , otherwise all data would transform to  $Y^0 = 1$

A regression model with the response variable a member of the family of power transformations becomes:

$$Y_i^{(\lambda)} = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \epsilon_i$$

where

$$Y_i^{(\lambda)} = \begin{cases} \frac{Y^\lambda - 1}{\lambda}, & \lambda \neq 0 \\ \log_e Y, & \lambda = 0 \end{cases}$$

Note that the regression model includes an additional parameter,  $\lambda$ , which needs to be estimated. The Box-Cox procedure uses **the method of maximum likelihood** to estimate  $\lambda$ , as well as the other parameters  $\beta_0, \beta_1, \beta_2, \dots$ . In this way, the Box-Cox procedure identifies  $\hat{\lambda}$ , the maximum likelihood estimate of  $\lambda$  to use in the power transformation.

**Example 1** A practical question for the head of a university could be how study fees (stfees) raise the universities net assets (nassets). A simple linear regression could help to explain the relation between these two variables.

```
library(Ecdat) #for dataset University
```

```
## Warning: package 'Ecdat' was built under R version 4.2.2
```

```
## Loading required package: Ecfun
```

```
## Warning: package 'Ecfun' was built under R version 4.2.2
```

```
##
```

```
## Attaching package: 'Ecfun'
```

```
## The following object is masked from 'package:base':
```

```
##
```

```
##      sign
```

```
##
```

```
## Attaching package: 'Ecdat'
```

```
## The following object is masked from 'package:carData':  
##  
##      Mroz
```

```
## The following object is masked from 'package:datasets':  
##  
##      Orange
```

```
reg=lm(nassets~stfees,data=University)  
summary(reg)
```

```
##  
## Call:  
## lm(formula = nassets ~ stfees, data = University)  
##  
## Residuals:  
##      Min       1Q   Median       3Q      Max   
## -70529 -20410  -5350   11948 244354   
##  
## Coefficients:  
##              Estimate Std. Error t value Pr(>|t|)      
## (Intercept) -36446.662  12078.934  -3.017  0.00374 **    
## stfees        12.957      1.472    8.802 2.13e-12 ***  
## ---  
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1  
##  
## Residual standard error: 48450 on 60 degrees of freedom  
## Multiple R-squared:  0.5636, Adjusted R-squared:  0.5563   
## F-statistic: 77.48 on 1 and 60 DF,  p-value: 2.126e-12
```

```
#Testing for Homoscedasticity  
library(lmtest)  
bptest(reg)
```

```
##  
## studentized Breusch-Pagan test  
##  
## data:  reg  
## BP = 9.8244, df = 1, p-value = 0.001722
```

```
#Testing for Normality  
shapiro.test(residuals(reg))
```

```
##  
## Shapiro-Wilk normality test  
##  
## data:  residuals(reg)  
## W = 0.79217, p-value = 6.024e-08
```

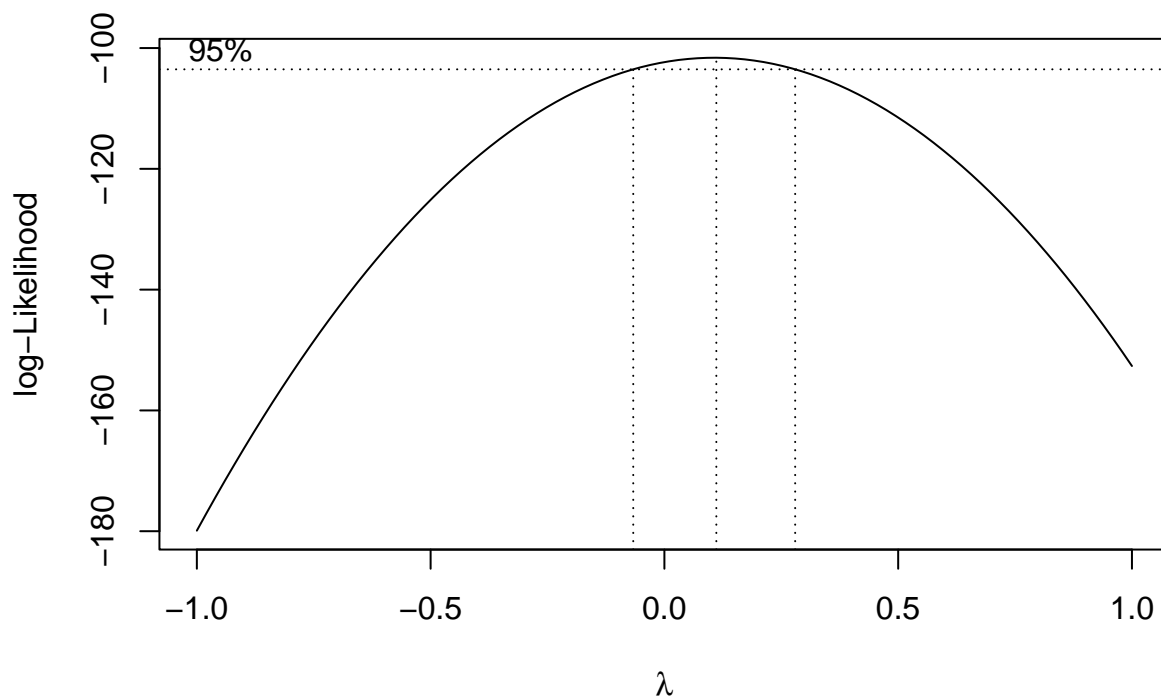
```
library(MASS) #for the boxcox()function
```

```
##
## Attaching package: 'MASS'

## The following object is masked from 'package:Ecdat':
##
##      SP500

## The following object is masked from 'package:olsrr':
##
##      cement
```

```
bc=boxcox(reg,lambda=seq(-1,1))
```



```
#extract best lambda
bestlambda=bc$x[which(bc$y==max(bc$y))]
bestlambda
```

```
## [1] 0.1111111
```

From the output we found that **bestlambda** is approximately between **-0.2 to 0.2**. Below are the outputs when we chose  $\lambda=0$  and  $\lambda=0.1111$ .

```
library(Ecdat)#for dataset University
bcmodel1=lm(log(nassets)~stfees,data=University)
summary(bcmodel1)
```

```
##
## Call:
## lm(formula = log(nassets) ~ stfees, data = University)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.1331 -0.2437  0.1189  0.4059  1.3533
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 9.014e+00  1.677e-01  53.765  < 2e-16 ***
## stfees      1.925e-04  2.043e-05   9.423 1.93e-13 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6725 on 60 degrees of freedom
## Multiple R-squared:  0.5967, Adjusted R-squared:  0.59
## F-statistic: 88.78 on 1 and 60 DF,  p-value: 1.934e-13
```

#### *#Testing for Homoscedasticity*

```
library(lmtest)
bptest(bcmodel1)
```

```
##
## studentized Breusch-Pagan test
##
## data:  bcmodel1
## BP = 0.71582, df = 1, p-value = 0.3975
```

#### *#Testing for Normality*

```
shapiro.test(residuals(bcmodel1))
```

```
##
## Shapiro-Wilk normality test
##
## data:  residuals(bcmodel1)
## W = 0.91401, p-value = 0.0003535
```

```
bcmodel2=lm((((nassets^0.1111)-1)/0.1111)~stfees,data=University)
summary(bcmodel2)
```

```
##
## Call:
## lm(formula = (((nassets^0.1111) - 1)/0.1111) ~ stfees, data = University)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -6.3143 -0.9367 0.2243 1.2150 4.4613
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.521e+01 5.085e-01 29.91 < 2e-16 ***
## stfees      6.340e-04 6.197e-05 10.23 8.99e-15 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.04 on 60 degrees of freedom
## Multiple R-squared: 0.6356, Adjusted R-squared: 0.6295
## F-statistic: 104.6 on 1 and 60 DF, p-value: 8.988e-15
```

#### *#Testing for Homoscedasticity*

```
library(lmtest)
bptest(bcmodel2)
```

```
##
## studentized Breusch-Pagan test
##
## data: bcmodel2
## BP = 0.066869, df = 1, p-value = 0.796
```

#### *#Testing for Normality*

```
shapiro.test(residuals(bcmodel2))
```

```
##
## Shapiro-Wilk normality test
##
## data: residuals(bcmodel2)
## W = 0.94981, p-value = 0.01307
```

### *Example 2*

Now we will use **the gala dataset** as an example of using the Box-Cox method to justify a transformation for Multiple Linear Regression. There are 30 Galapagos islands and 7 variables in the dataset. The relationship between the number of plant species and several geographic variables is of interest. We fit an additive multiple regression model with Species as the response and most of the other variables as predictors.

*The dataset gala contains the following variables:*

Species: the number of plant species found on the island

Endemics: the number of endemic species

Area: the area of the island ( $km^2$ )

Elevation: the highest elevation of the island (m)

Nearest: the distance from the nearest island (km)

Scruz: the distance from Santa Cruz island (km)

Adjacent: the area of the adjacent island (square km)

*Source: M. P. Johnson and P. H. Raven (1973) "Species number and endemism: The Galapagos Archipelago revisited" Science, 179, 893-895*

```
library(faraway)#for the dataset gala
```

```
## Warning: package 'faraway' was built under R version 4.2.2
```

```
##
```

```
## Attaching package: 'faraway'
```

```
## The following object is masked from 'package:olsrr':
```

```
##
```

```
##      hsb
```

```
## The following objects are masked from 'package:car':
```

```
##
```

```
##      logit, vif
```

```
## The following object is masked from 'package:GGally':
```

```
##
```

```
##      happy
```

```
reg1=lm(Species ~ Area + Elevation + Nearest + Scrutz + Adjacent, data = gala)
summary(reg1)
```

```
##
```

```
## Call:
```

```
## lm(formula = Species ~ Area + Elevation + Nearest + Scrutz + 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
## Scrutz       -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
```

```
reg2=lm(Species ~ Elevation + Adjacent, data = gala)
summary(reg2)
```

```
##
```

```
## Call:
## lm(formula = Species ~ Elevation + Adjacent, data = gala)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -103.41  -34.33  -11.43   22.57  203.65
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.43287    15.02469   0.095 0.924727
## Elevation     0.27657     0.03176   8.707 2.53e-09 ***
## Adjacent     -0.06889     0.01549  -4.447 0.000134 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.86 on 27 degrees of freedom
## Multiple R-squared:  0.7376, Adjusted R-squared:  0.7181
## F-statistic: 37.94 on 2 and 27 DF,  p-value: 1.434e-08
```

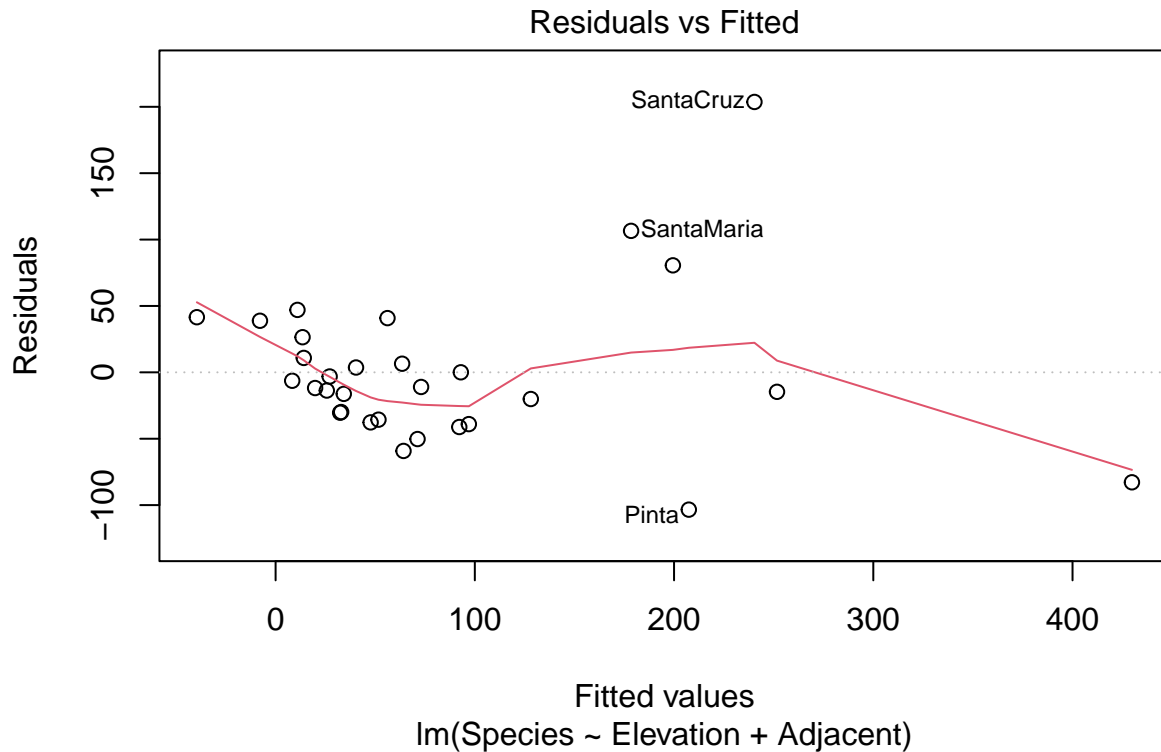
#### *#Testing for Homoscedasticity*

```
library(lmtest)
bptest(reg2)
```

```
##
## studentized Breusch-Pagan test
##
## data:  reg2
## BP = 7.6832, df = 2, p-value = 0.02146
```

```
plot(reg2,which=1)
```





```
#Testing for Normality
shapiro.test(residuals(reg2))
```

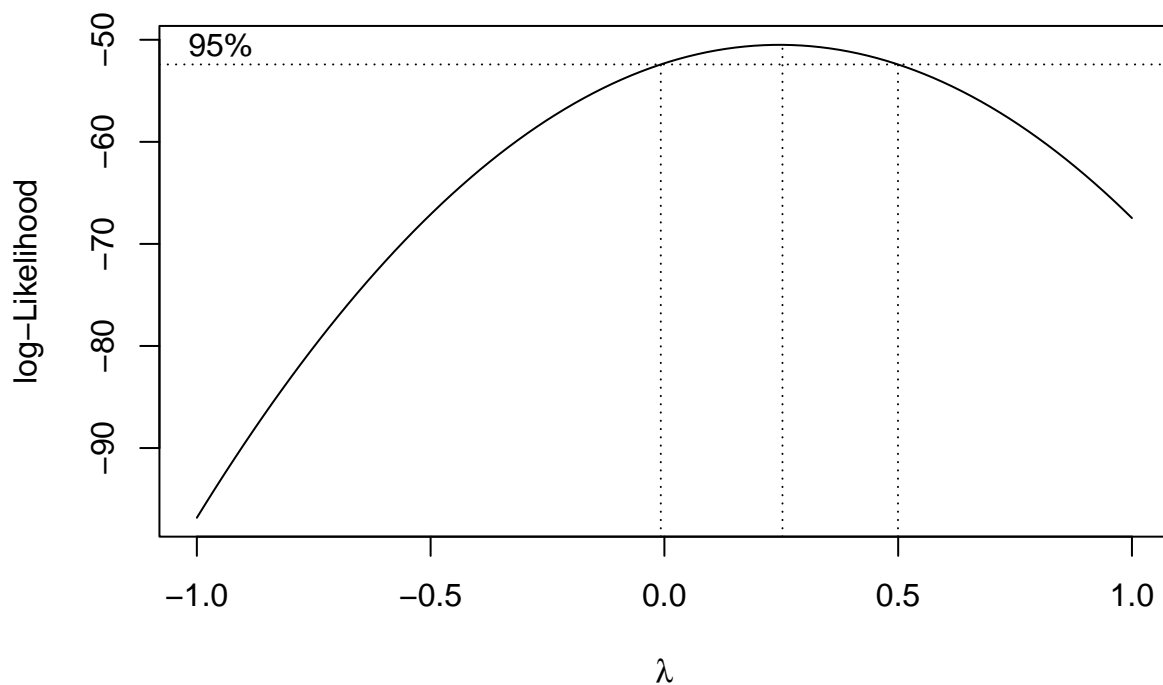
```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(reg2)
## W = 0.89306, p-value = 0.005706
```

```
library(MASS) #for the boxcox()function
reg2=lm(Species ~ Elevation + Adjacent, data = gala)
summary(reg2)
```

```
##
## Call:
## lm(formula = Species ~ Elevation + Adjacent, data = gala)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -103.41  -34.33  -11.43   22.57   203.65
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   1.43287    15.02469   0.095 0.924727
## Elevation     0.27657     0.03176   8.707 2.53e-09 ***
```

```
## Adjacent    -0.06889    0.01549   -4.447 0.000134 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 60.86 on 27 degrees of freedom
## Multiple R-squared:  0.7376, Adjusted R-squared:  0.7181
## F-statistic: 37.94 on 2 and 27 DF,  p-value: 1.434e-08
```

```
bc=boxcox(reg2,lambda=seq(-1,1))
```



```
#extract best lambda
bestlambda=bc$x[which(bc$y==max(bc$y))]
bestlambda
```

```
## [1] 0.2525253
```

Using the Box-Cox method, we see that  $\lambda = 0.2525$  falls both in the confidence interval, and is extremely close to the maximum, which suggests a transformation of the form

$$Y_i^{(0.2525)} = \frac{Y_i^{0.2525} - 1}{0.2525}.$$

$$Y_i^{(0.2525)} = \beta_0 + \beta_1 \text{Elevation} + \beta_2 \text{Adjacent} + \epsilon_i$$

Will be used. We then fit a model with this transformation applied to the response.

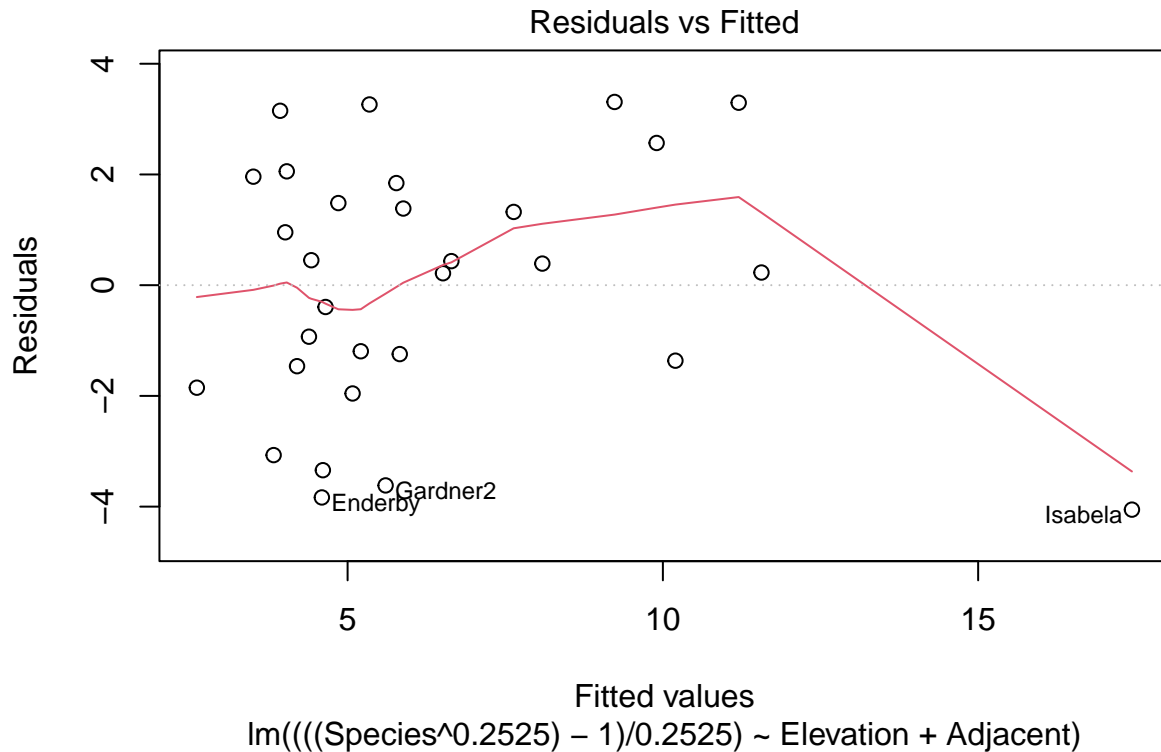
```
#We found that bestlambda=0.2525
bcmodel=lm((((Species^0.2525)-1)/0.2525) ~ Elevation + Adjacent, data = gala)
summary(bcmodel)
```

```
##
## Call:
## lm(formula = (((Species^0.2525) - 1)/0.2525) ~ Elevation + Adjacent,
##     data = gala)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.0544 -1.4386  0.3095  1.7540  3.3096
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  3.6073542   0.5779241    6.242 1.12e-06 ***
## Elevation    0.0087917   0.0012217    7.196 9.71e-08 ***
## Adjacent    -0.0018533   0.0005958   -3.111 0.00437 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.341 on 27 degrees of freedom
## Multiple R-squared:  0.6607, Adjusted R-squared:  0.6356
## F-statistic: 26.29 on 2 and 27 DF,  p-value: 4.604e-07
```

```
#Testing for Homoscedasticity
library(lmtest)
bptest(bcmodel)
```

```
##
## studentized Breusch-Pagan test
##
## data:  bcmodel
## BP = 3.3897, df = 2, p-value = 0.1836
```

```
plot(bcmodel,which=1)
```



```
#Testing for Normality
shapiro.test(residuals(bcmodel))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(bcmodel)
## W = 0.95031, p-value = 0.1724
```

From the output, the resulting fitted versus residuals plot looks much better! Moreover, the output displays the Breusch-Pagan test that result from the Box-Cox model. The  $p\text{-value} = 0.1836 > 0.05$ , indicating that we do not reject the null hypothesis. Therefore, the test provide evidence to suggest that homoscedasticity does exist. Additinally, Shapiro-Wilk normality test also confirms that the residuals are normally distributed as the  $p\text{-value} = 0.1724 > 0.05$ . So Box-Cox Tranformation is helpful for these cases.

## Important points about OLS Regression

1. There must be linear relationship between independent and dependent variables.
2. Multiple regression suffers from multicollinearity and heteroskedasticity.
3. Linear Regression is very sensitive to Outliers. It can terribly affect the regression line and eventually the forecasted values.
4. Multicollinearity can increase the variance of the coefficient estimates and make the estimates very sensitive to minor changes in the model. The result is that the coefficient estimates are unstable.

```
credit=read.csv("credit.csv", header = TRUE)
```

```
nomultimodel2<-lm(Balance~Income+Limit+Age+Cards+factor(Student),data=credit)
summary(nomultimodel2)
```

```
##
## Call:
## lm(formula = Balance ~ Income + Limit + Age + Cards + factor(Student),
##     data = credit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -187.05  -79.57  -12.59   56.06  322.56
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -4.673e+02  2.199e+01 -21.250  < 2e-16 ***
## Income        -7.760e+00  2.341e-01 -33.149  < 2e-16 ***
## Limit          2.661e-01  3.535e-03  75.296  < 2e-16 ***
## Age           -6.220e-01  2.933e-01  -2.120   0.0346 *
## Cards          2.355e+01  3.628e+00   6.492  2.55e-10 ***
## factor(Student)Yes 4.284e+02  1.655e+01  25.886  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 99.12 on 394 degrees of freedom
## Multiple R-squared:  0.9541, Adjusted R-squared:  0.9535
## F-statistic: 1638 on 5 and 394 DF,  p-value: < 2.2e-16
```

```
intmodel<-lm(Balance~(Income+Limit+Age+Cards+factor(Student))^2,data=credit)
summary(intmodel)
```

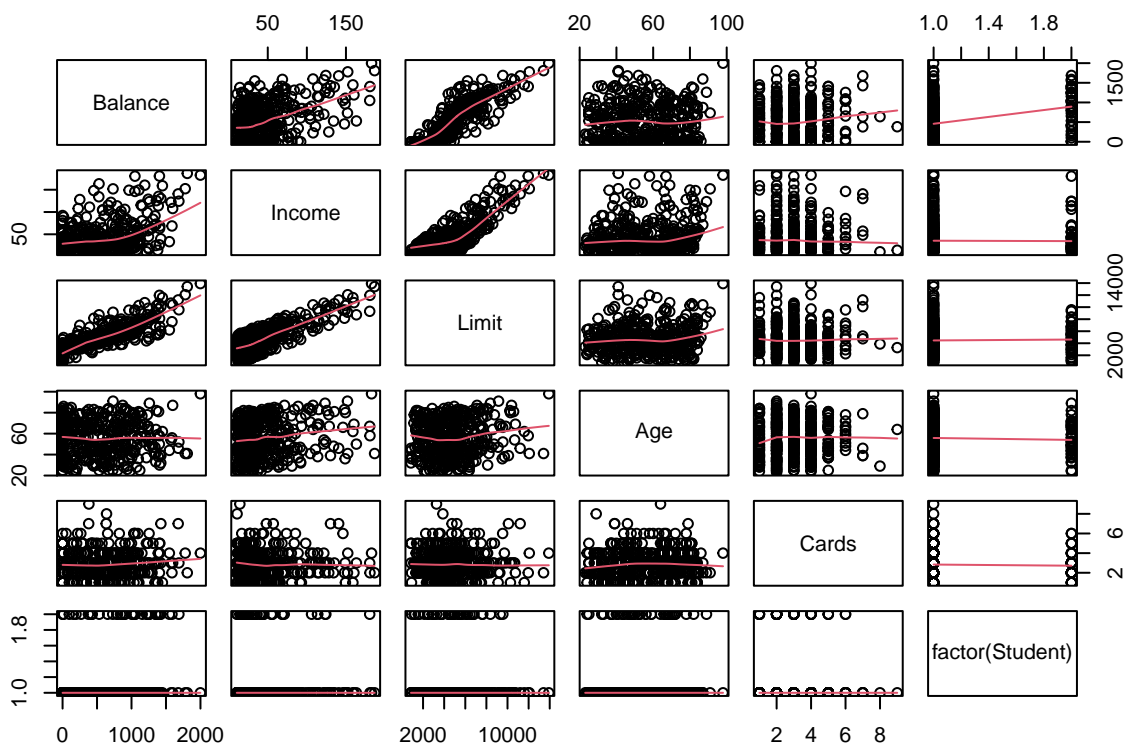
```
##
## Call:
## lm(formula = Balance ~ (Income + Limit + Age + Cards + factor(Student))^2,
##     data = credit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -168.973  -70.407   -3.792   53.021  264.913
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -3.827e+02  5.380e+01  -7.114  5.57e-12 ***
## Income        -9.631e+00  9.637e-01  -9.994  < 2e-16 ***
## Limit          2.433e-01  1.338e-02  18.189  < 2e-16 ***
## Age           6.358e-01  8.280e-01   0.768  0.443032
## Cards          1.522e+01  1.444e+01   1.054  0.292436
## factor(Student)Yes 2.669e+02  8.031e+01   3.324  0.000974 ***
## Income:Limit    3.260e-04  4.551e-05   7.164  4.04e-12 ***
## Income:Age     -5.152e-03  1.237e-02  -0.416  0.677413
## Income:Cards   -4.481e-02  1.705e-01  -0.263  0.792879
```

```
## Income:factor(Student)Yes -2.307e+00 6.648e-01 -3.470 0.000581 ***
## Limit:Age -1.217e-04 1.880e-04 -0.647 0.517923
## Limit:Cards 2.905e-03 2.643e-03 1.099 0.272432
## Limit:factor(Student)Yes 6.991e-02 1.102e-02 6.341 6.41e-10 ***
## Age:Cards -1.017e-01 2.045e-01 -0.497 0.619272
## Age:factor(Student)Yes -1.004e+00 1.068e+00 -0.940 0.347629
## Cards:factor(Student)Yes -6.897e-01 1.203e+01 -0.057 0.954327
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 89.9 on 384 degrees of freedom
## Multiple R-squared: 0.9632, Adjusted R-squared: 0.9618
## F-statistic: 670.2 on 15 and 384 DF, p-value: < 2.2e-16
```

```
bestintmodel<-lm(Balance~Income+Limit+Age+Cards+factor(Student)+Income*Limit+Income*factor(Student)+Lim
summary(bestintmodel)
```

```
##
## Call:
## lm(formula = Balance ~ Income + Limit + Age + Cards + factor(Student) +
##     Income * Limit + Income * factor(Student) + Limit * factor(Student),
##     data = credit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -167.549  -66.191   -6.375   55.591  265.763
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -3.550e+02  2.441e+01 -14.544 < 2e-16 ***
## Income          -1.000e+01  4.228e-01 -23.656 < 2e-16 ***
## Limit           2.454e-01  3.956e-03  62.046 < 2e-16 ***
## Age            -5.926e-01  2.662e-01  -2.226  0.0266 *
## Cards           2.260e+01  3.292e+00   6.866 2.60e-11 ***
## factor(Student)Yes 2.163e+02  3.697e+01   5.851 1.04e-08 ***
## Income:Limit     3.165e-04  4.477e-05   7.069 7.21e-12 ***
## Income:factor(Student)Yes -2.593e+00  6.272e-01  -4.134 4.36e-05 ***
## Limit:factor(Student)Yes  7.164e-02  1.082e-02   6.620 1.19e-10 ***
## ---
## Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 89.88 on 391 degrees of freedom
## Multiple R-squared: 0.9625, Adjusted R-squared: 0.9618
## F-statistic: 1256 on 8 and 391 DF, p-value: < 2.2e-16
```

```
library(ggplot2)
library(GGally)
pairs(~Balance+Income+Limit+Age+Cards+factor(Student),data=credit,panel=panel.smooth)
```



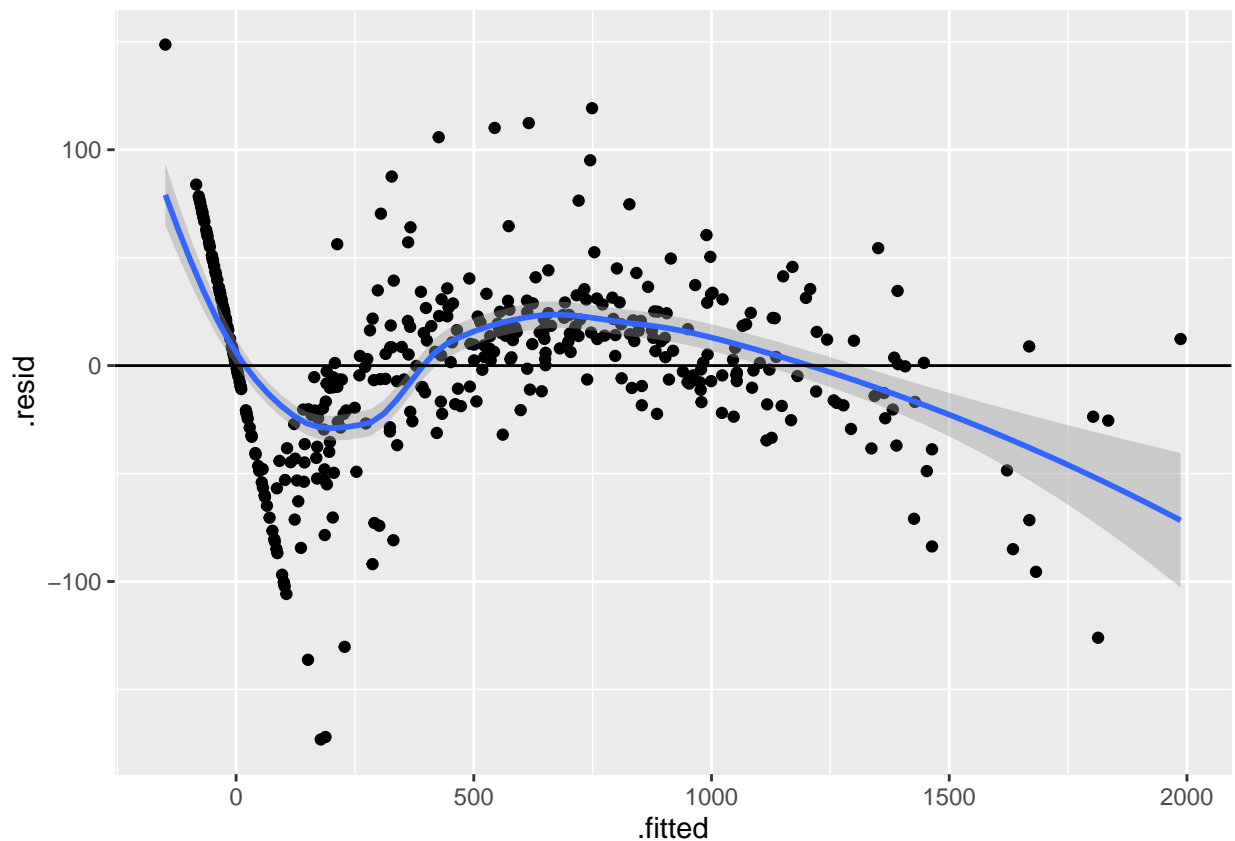
```
bestintmodel1<-lm(Balance~Income+I(Income^2)+Limit+I(Limit^2)+I(Limit^3)+I(Limit^4)+I(Limit^5)+I(Limit^6)+I(Limit^7)+I(Limit^8)+Age+Cards+factor(Student)+Income*Limit+Income*factor(Student)+Limit*factor(Student), data = credit)
summary(bestintmodel1)
```

```
##
## Call:
## lm(formula = Balance ~ Income + I(Income^2) + Limit + I(Limit^2) +
##     I(Limit^3) + I(Limit^4) + I(Limit^5) + I(Limit^6) + I(Limit^7) +
##     I(Limit^8) + Age + Cards + factor(Student) + Income * Limit +
##     Income * factor(Student) + Limit * factor(Student), data = credit)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -173.154  -20.716    3.865   23.408  148.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -7.894e+02  1.547e+02  -5.103 5.28e-07 ***
## Income         -3.485e+00  4.421e-01  -7.884 3.33e-14 ***
## I(Income^2)     3.493e-02  4.271e-03   8.178 4.28e-15 ***
## Limit          1.709e+00  3.137e-01   5.448 9.14e-08 ***
## I(Limit^2)     -1.350e-03  2.471e-04  -5.463 8.42e-08 ***
## I(Limit^3)      5.498e-07  1.004e-07   5.477 7.85e-08 ***
## I(Limit^4)     -1.202e-10  2.331e-11  -5.157 4.03e-07 ***
## I(Limit^5)      1.530e-14  3.203e-15   4.777 2.54e-06 ***
## I(Limit^6)     -1.132e-18  2.566e-19  -4.411 1.34e-05 ***
```

```
## I(Limit^7)          4.513e-23  1.105e-23   4.084 5.39e-05 ***
## I(Limit^8)          -7.502e-28  1.973e-28  -3.803 0.000166 ***
## Age                 -8.369e-01  1.299e-01  -6.443 3.51e-10 ***
## Cards               2.085e+01  1.595e+00  13.068 < 2e-16 ***
## factor(Student)Yes  1.592e+02  1.801e+01   8.840 < 2e-16 ***
## Income:Limit        -1.724e-03  1.457e-04 -11.833 < 2e-16 ***
## Income:factor(Student)Yes -2.482e+00  3.088e-01  -8.039 1.14e-14 ***
## Limit:factor(Student)Yes  8.231e-02  5.238e-03  15.714 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 43.13 on 383 degrees of freedom
## Multiple R-squared:  0.9916, Adjusted R-squared:  0.9912
## F-statistic: 2810 on 16 and 383 DF, p-value: < 2.2e-16
```

```
#check linearity
#residual vs fitted data plot for the model
ggplot(bestintmodel1, aes(x=.fitted, y=.resid)) +
  geom_point() +geom_smooth()+
  geom_hline(yintercept = 0)
```

```
## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'
```



Improving model by using  $\log(X)$ , but not recommended



```

library(ggplot2)
Advertising=read.table("Advertising.txt", header = TRUE,sep ="\t" )
modellog<-lm(sale~log(tv)+radio+tv*radio, data=Advertising)
summary(modellog)

##
## Call:
## lm(formula = sale ~ log(tv) + radio + tv * radio, data = Advertising)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.85606 -0.20548  0.01615  0.19011  0.77385
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.310e-01  1.774e-01   1.302   0.194
## log(tv)      1.943e+00  4.723e-02  41.148 <2e-16 ***
## radio        4.668e-02  2.902e-03  16.087 <2e-16 ***
## tv           4.888e-04  6.629e-04   0.737   0.462
## radio:tv     1.018e-03  1.697e-05  60.015 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.304 on 195 degrees of freedom
## Multiple R-squared:  0.9967, Adjusted R-squared:  0.9966
## F-statistic: 1.461e+04 on 4 and 195 DF,  p-value: < 2.2e-16

#residual vs fitted data plot for the simple model
ggplot(modellog, aes(x=.fitted, y=.resid)) +
  geom_point() +geom_smooth()+
  geom_hline(yintercept = 0)

## 'geom_smooth()' using method = 'loess' and formula 'y ~ x'

```

