Regression Analysis

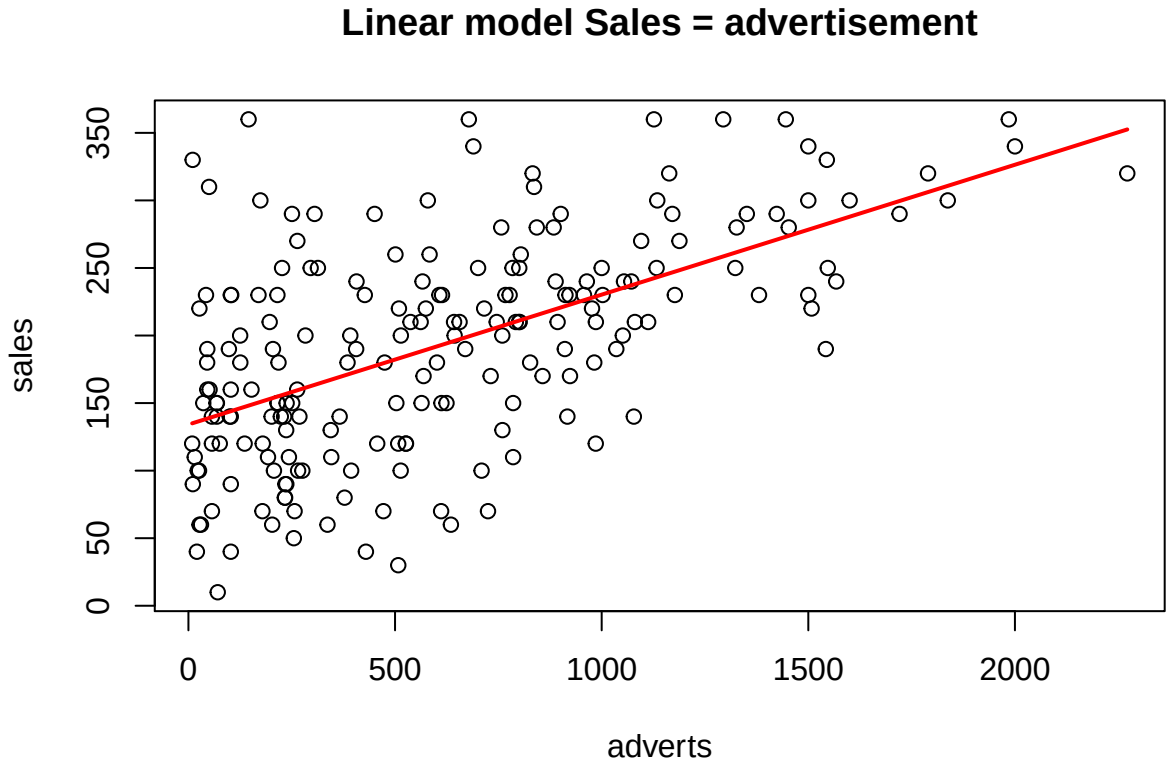
```
## Setting the directory
setwd("~/R-stuff")
## Package required for analysis
library("car")
```

Simple linear regression model

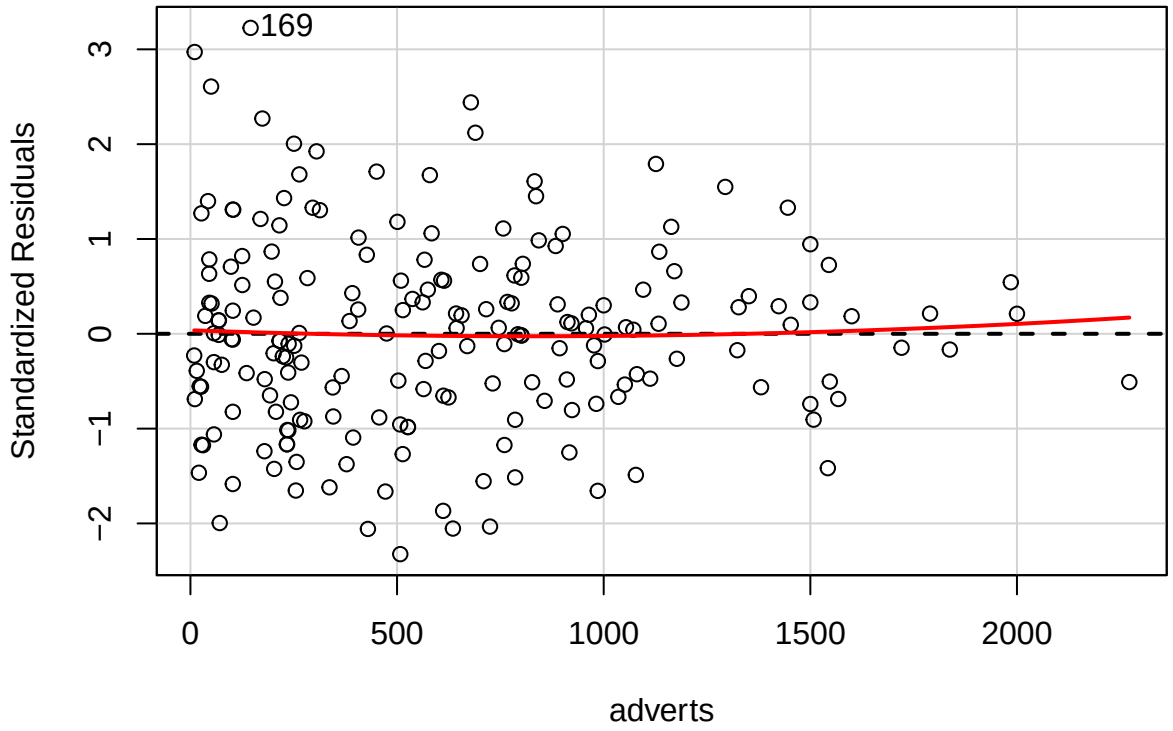
```
#--- Loading and reading dat file
album1<-read.delim("Album Sales 1.dat", header = TRUE)
#--- Simple linear regression
albumSales1 <- lm(sales ~ adverts, data = album1)
summary(albumSales1)
```

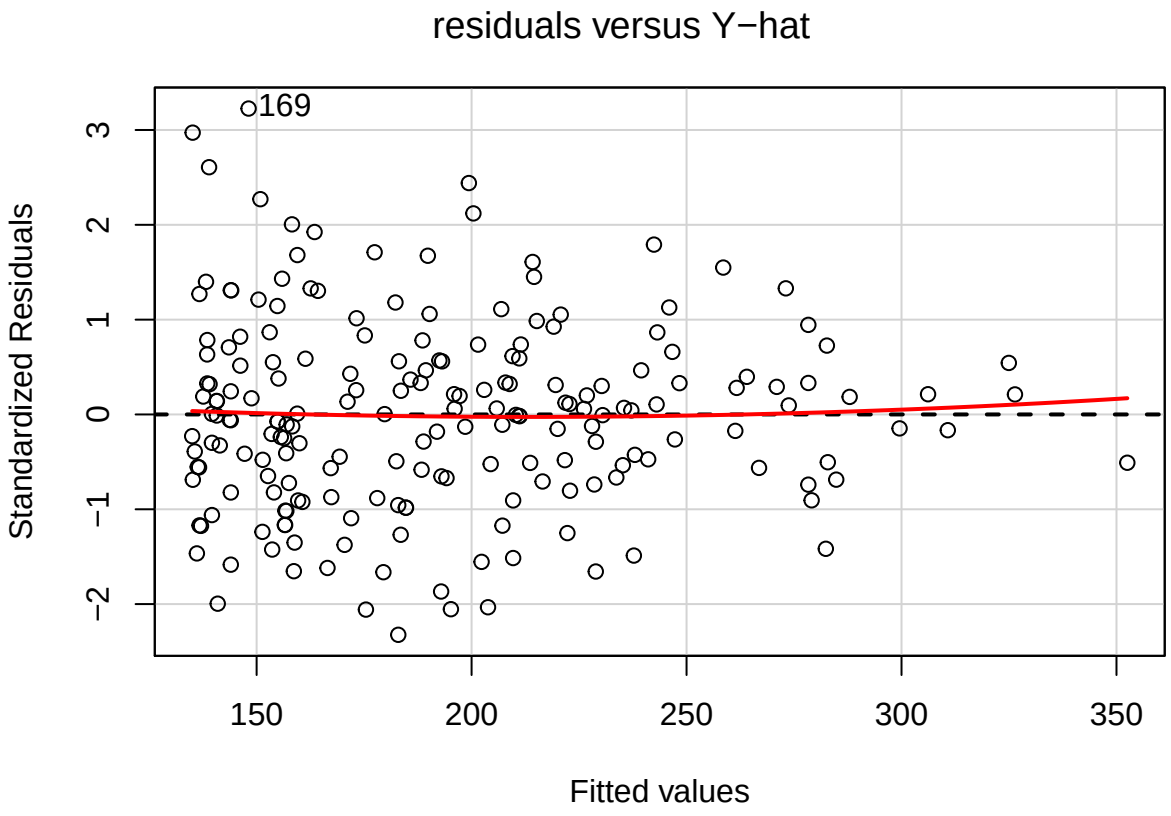
```
##
## Call:
## lm(formula = sales ~ adverts, data = album1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -152.949  -43.796   -0.393   37.040  211.866
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 1.341e+02  7.537e+00  17.799   <2e-16 ***
## adverts     9.612e-02  9.632e-03   9.979   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 65.99 on 198 degrees of freedom
## Multiple R-squared:  0.3346, Adjusted R-squared:  0.3313
## F-statistic: 99.59 on 1 and 198 DF,  p-value: < 2.2e-16
```

```
#--- Plots of the model, and some diagnostics
plot(sales ~ adverts, data=album1, main = "Linear model Sales = advertisement")
regLine(albumSales1)
```




```
#--- Residual plots
residualPlots(albumSales1, type= "rstandard", layout = c(1, 1), ask=FALSE, id.n=1, main="residuals versu
```






```
##            Test stat Pr(>|t|)
## adverts        0.379    0.705
## Tukey test     0.379    0.704
```

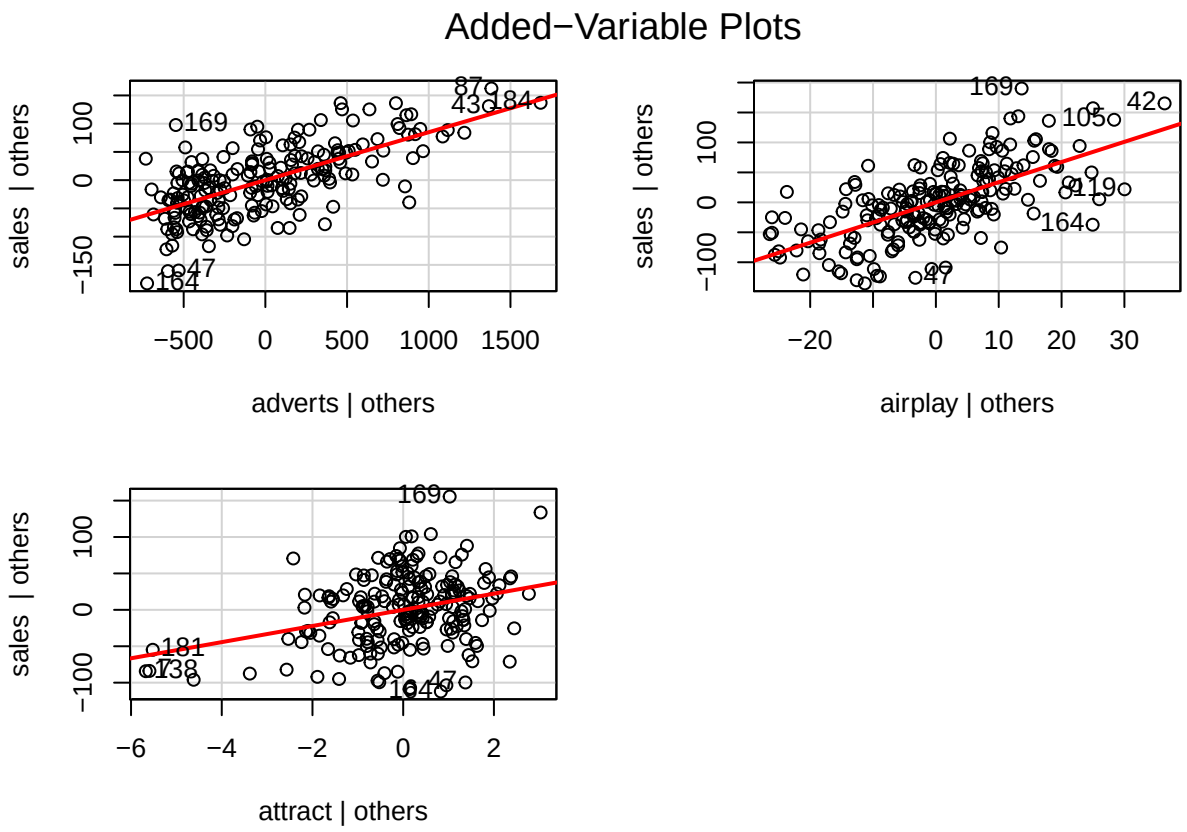
Multiple regression model

```
#--- Loading and reading dat file
album2<-read.delim("Album Sales 2.dat", header = TRUE)
#---Run the multiple regression model----------------------------
albumSales<-lm(sales ~ adverts + airplay + attract, data = album2)
summary(albumSales)
```

```
##
## Call:
## lm(formula = sales ~ adverts + airplay + attract, data = album2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -121.324  -28.336   -0.451   28.967  144.132
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -26.612958  17.350001  -1.534    0.127
## adverts       0.084885   0.006923  12.261  < 2e-16 ***
## airplay       3.367425   0.277771  12.123  < 2e-16 ***
```

```
## attract      11.086335   2.437849   4.548 9.49e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 47.09 on 196 degrees of freedom
## Multiple R-squared:  0.6647, Adjusted R-squared:  0.6595
## F-statistic: 129.5 on 3 and 196 DF,  p-value: < 2.2e-16
```

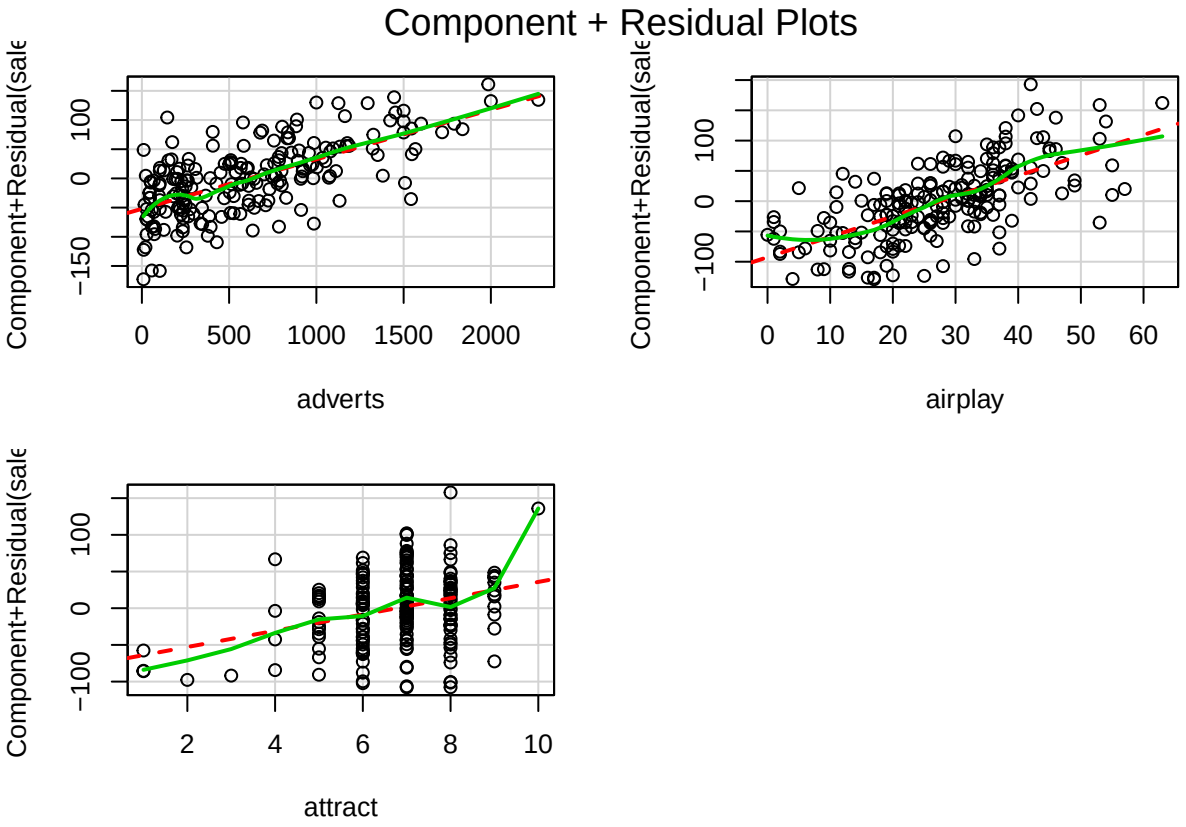
```
#--- Added-Value Plots (DV vs. IV while holding other variables constant). like partial regression plot
avPlots(albumSales, id.n=3)
```



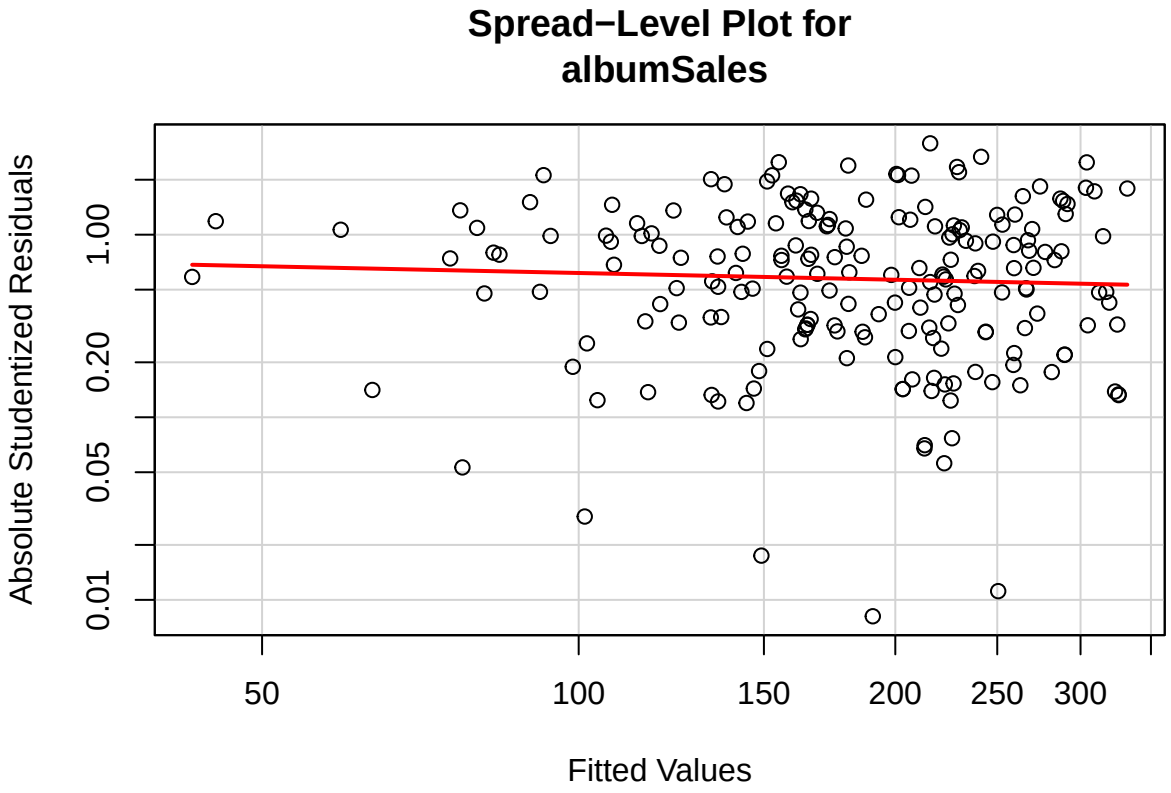
Assumptions of multiple regression

1. Relationship between the DV and IV is linear

```
#--- Independent variable & residual plot
crPlots(albumSales)
```




```
#--- Predicted Dependent variable & standardized residual
#--- plot studentized residuals vs. fitted values. Helps identify potential issues of non-linearity
spreadLevelPlot(albumSales)
```



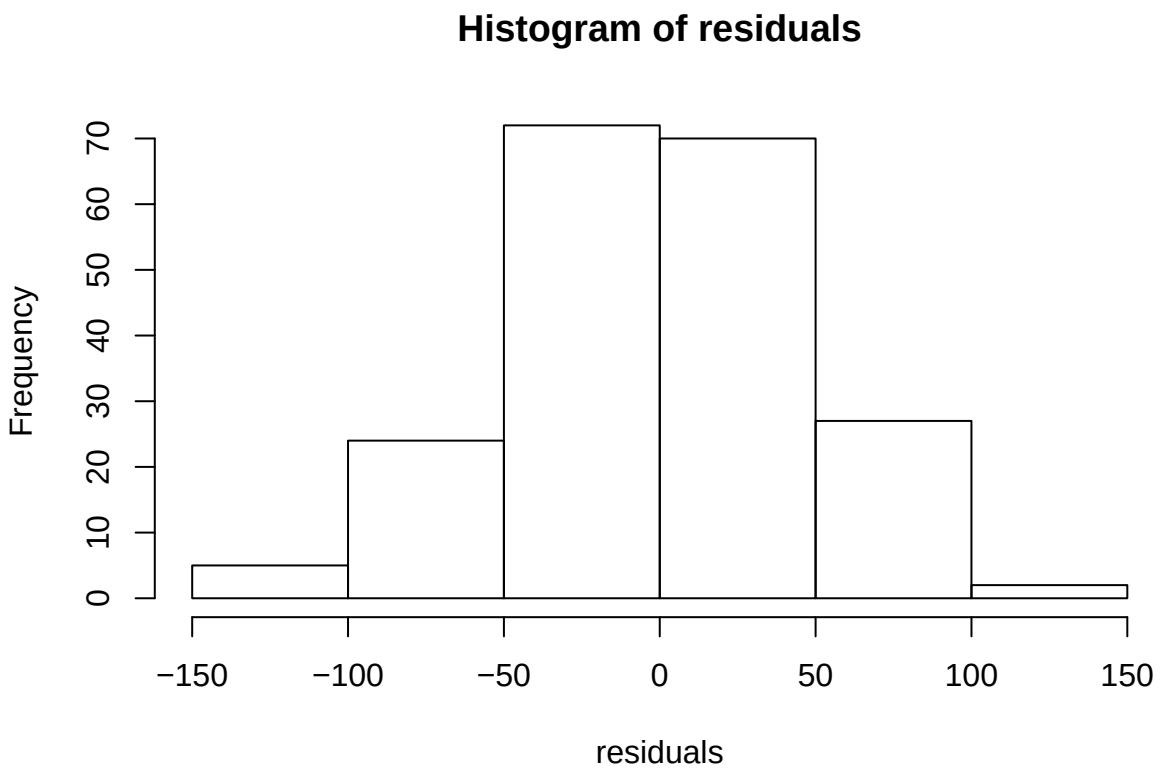

```
##
## Suggested power transformation:  1.122613
```

2. Error terms has a mean of zero

```
#--- Examine the mean of the residuals
mean(albumSales$residuals, na.rm=TRUE)
```

```
## [1] -1.258056e-15
```

```
hist(albumSales$residuals, main="Histogram of residuals", xlab="residuals")
```

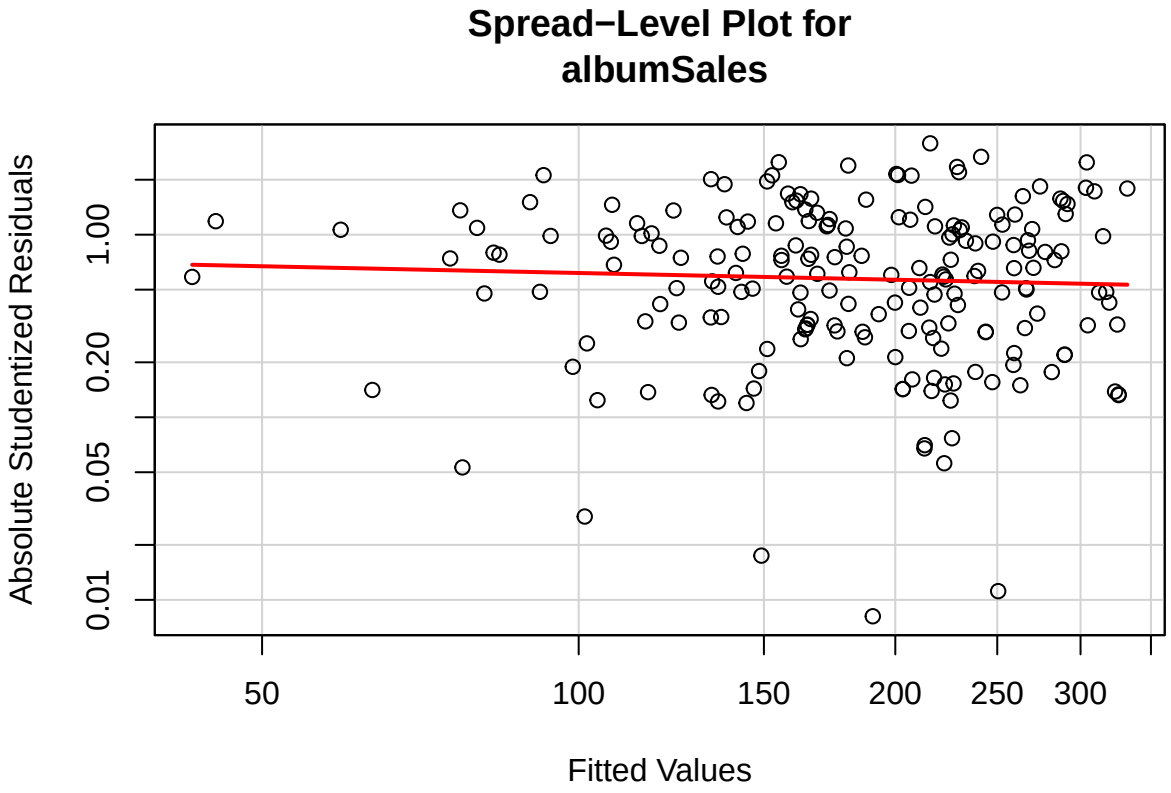


3. Error term has a constant variance

```
#--- Examining for presences of homoscedasticity
#--- non-constant error variance test
ncvTest(albumSales)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.3030143    Df = 1     p = 0.5819989
```

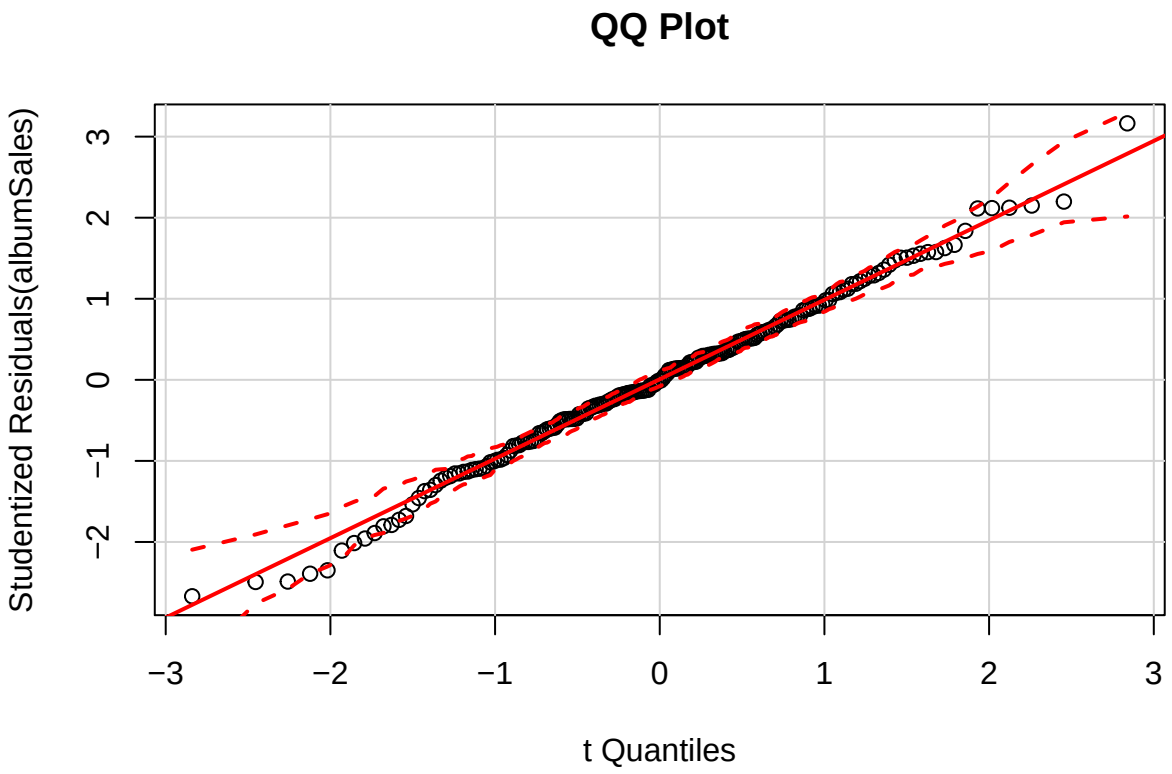
```
#--- plot studentized residuals vs. fitted values (same as before). Helps identify issues of homoscedas
spreadLevelPlot(albumSales)
```



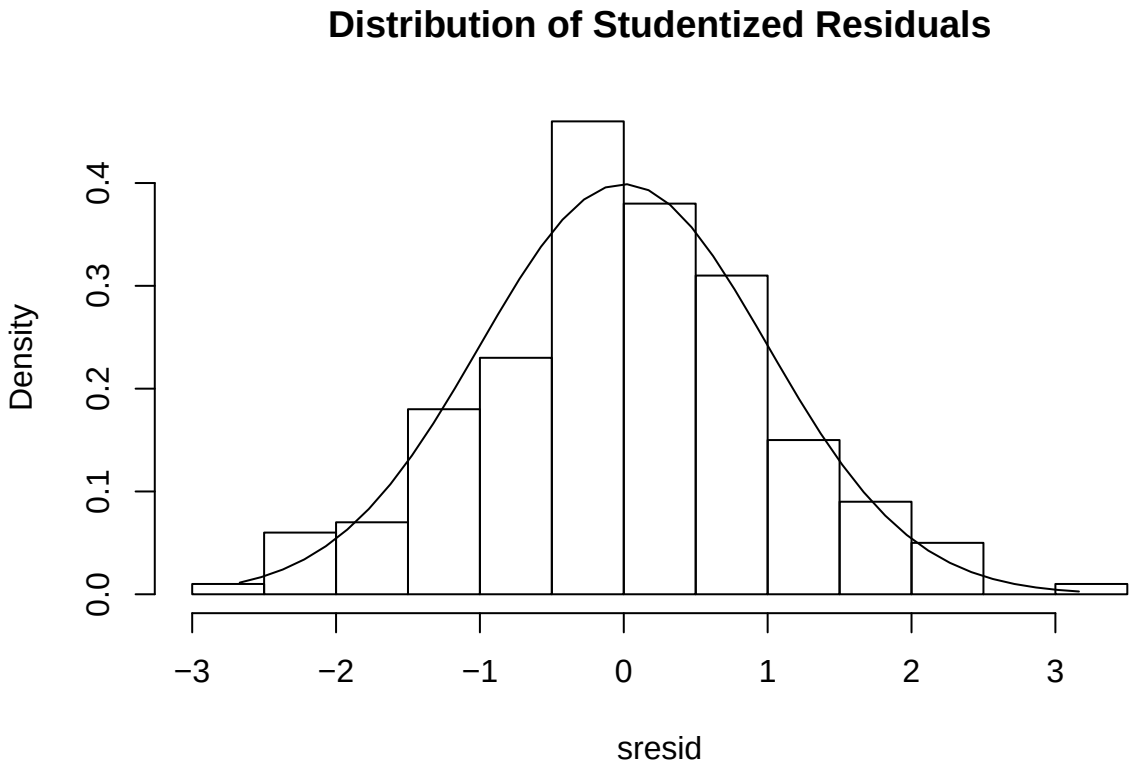

```
##
## Suggested power transformation:  1.122613
```

4.Errors are normally distributed

```
#--- Examining Normality of Residuals
#--- qq plot for studentized resid
qqPlot(albumSales, main="QQ Plot")
```




```
#--- distribution of studentized residuals
library("MASS")
sresid <- studres(albumSales)
hist(sresid, freq=FALSE,
   main="Distribution of Studentized Residuals")
xfit<-seq(min(sresid),max(sresid),length=40)
yfit<-dnorm(xfit)
lines(xfit, yfit)
```



5. Errors are uncorrelated

```
#--- Examining the correlation between erros in a longitudinal studies
#--- Test for Autocorrelated Errors
durbinWatsonTest(albumSales)
```

```
##  lag Autocorrelation D-W Statistic p-value
##    1       0.0026951      1.949819   0.728
##  Alternative hypothesis: rho != 0
```

Diagnostics in Multiple regression

1. Multicollinearity (Variance inflation factor)

```
#--- Examining for relationship between the independent variables
vif(albumSales) # variance inflation factors
```

```
##  adverts  airplay  attract
## 1.014593 1.042504 1.038455
```

```
sqrt(vif(albumSales)) > 2 # problem?
```

```
## adverts airplay attract
##   FALSE   FALSE   FALSE
```

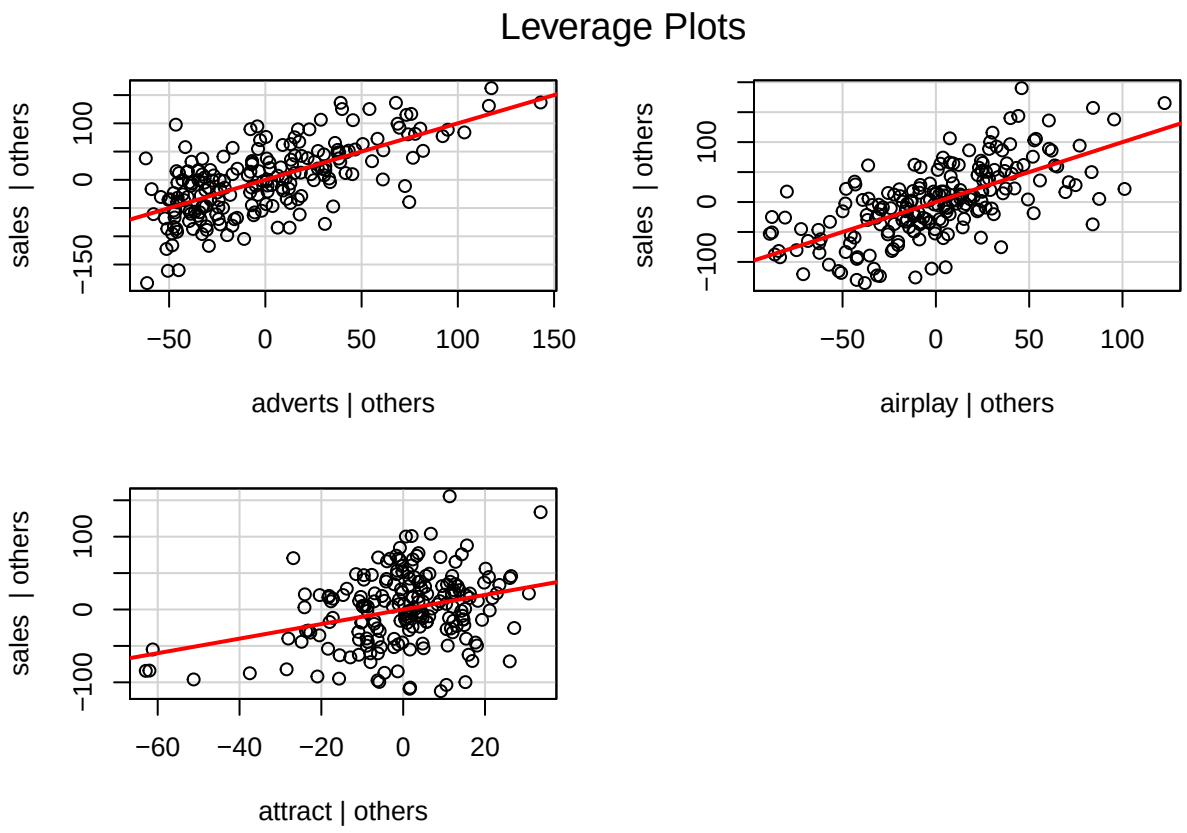
Estimating outliers and influential observation

2. Outliers

```
#--- Assessing the Outliers in the data
outlierTest(albumSales) # Bonferonni p-value for most extreme obs
```

```
##
## No Studentized residuals with Bonferonni p < 0.05
## Largest |rstudent|:
##     rstudent unadjusted p-value Bonferonni p
## 169 3.163622          0.0018077      0.36154
```

```
leveragePlots(albumSales) # leverage plots
```

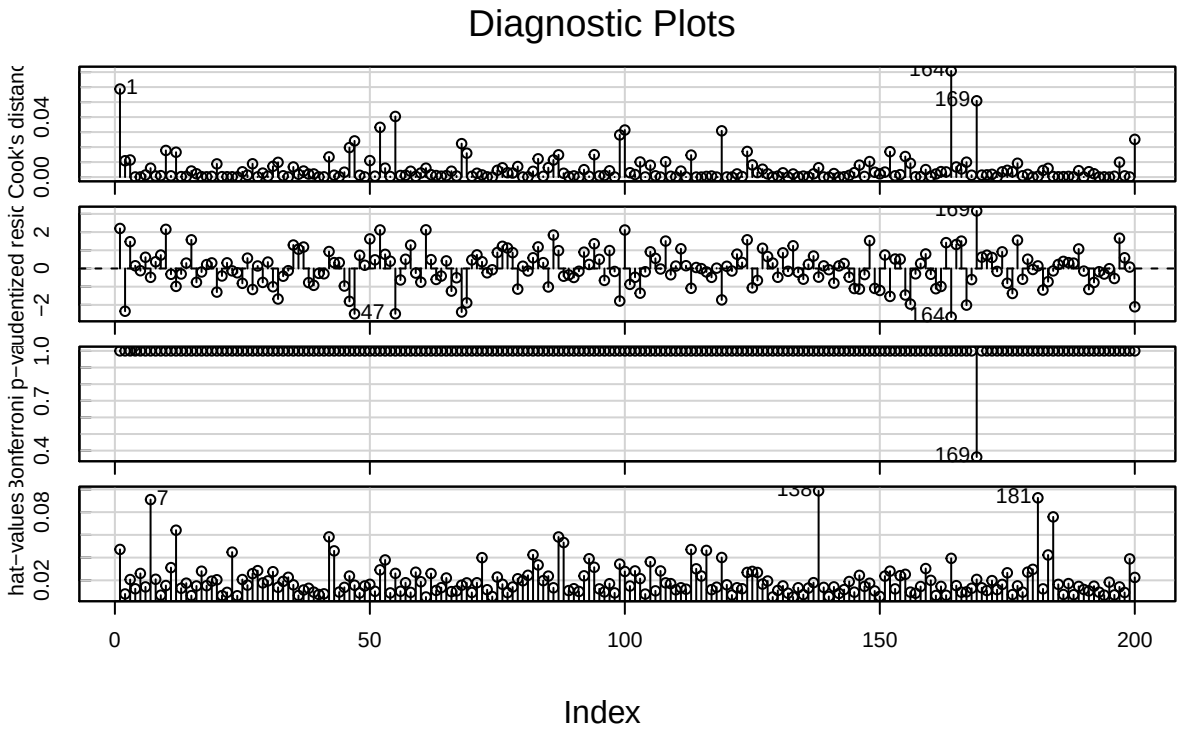



```
album2$residuals<-resid(albumSales)
album2$studentized.residuals <- rstudent(albumSales)
album2$standardized.residuals <- rstandard(albumSales)
```

3. Influential observations

```
#--- Assessing the influential observation
album2$cooks.distance<-cooks.distance(albumSales)
album2$dffit <- dffits(albumSales)
album2$leverage <- hatvalues(albumSales)
album2$covariance.ratios <- covratio(albumSales)
```

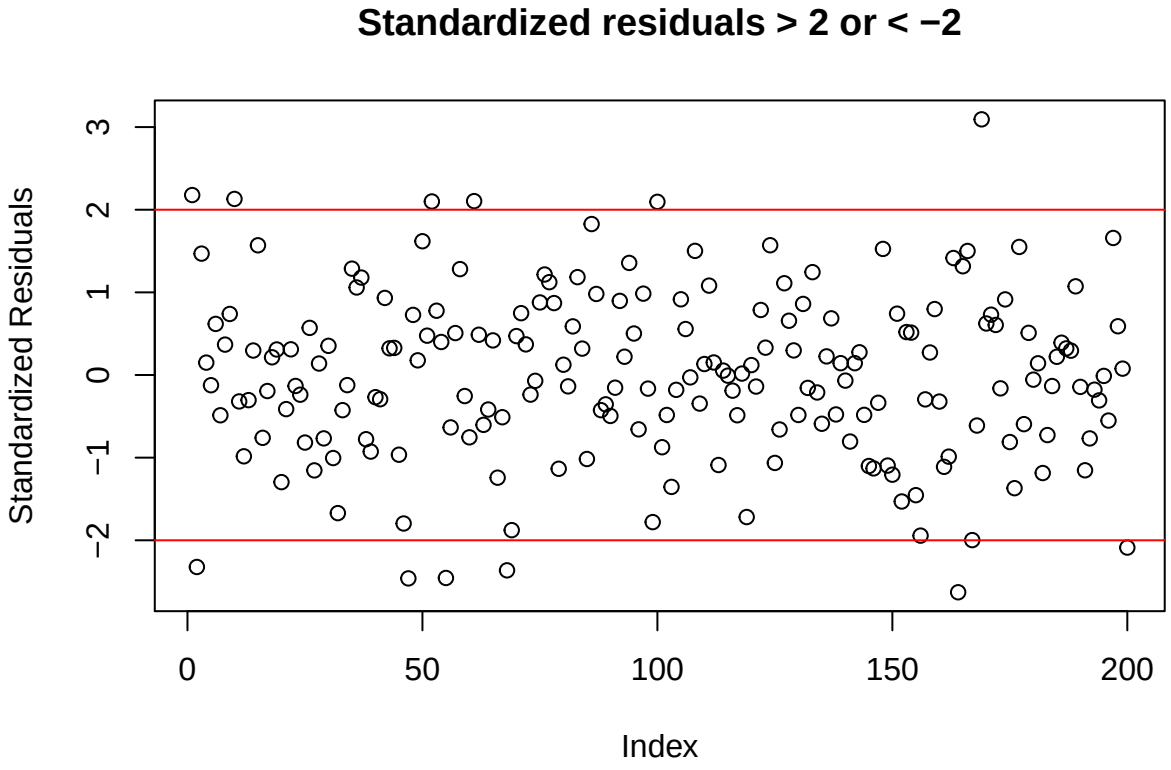
```
dfbs.album2 <-dfbetas(albumSales)
dfbs <- data.frame (dfbs.album2)
names(dfbs) <- c("dfb.intercept", "dfb.adverts", "dfb.airplay", "dfb.attract")
album2.total <- cbind(album2, dfbs)
#---------- Influence index plots: Cook's distance; studentized residuals; Bonferroni values and hat-va
influenceIndexPlot(albumSales, id.n=3)
```



Analyzing outliers and influential observation

```
#Save file
# write.table(album2, "Album Sales With Diagnostics.dat", sep = "\t", row.names = FALSE)
#look at the data (and round the values)
# round(album2, digits = 3)

#--- List of standardized residuals greater than 2
plot(album2.total$standardized.residuals, main = "Standardized residuals > 2 or < -2", ylab = "Standard
abline(h = 2, col=2)
abline(h= -2, col=2)
```




```
#--- Create a variable called large.residual, which is TRUE (or 1) if the residual is greater than 2, o
album2.total$large.residual <- album2$standardized.residuals > 2 | album2$standardized.residuals < -2
#--- Count the number of large residuals-------------
sum(album2.total$large.residual)
```

```
## [1] 12
```

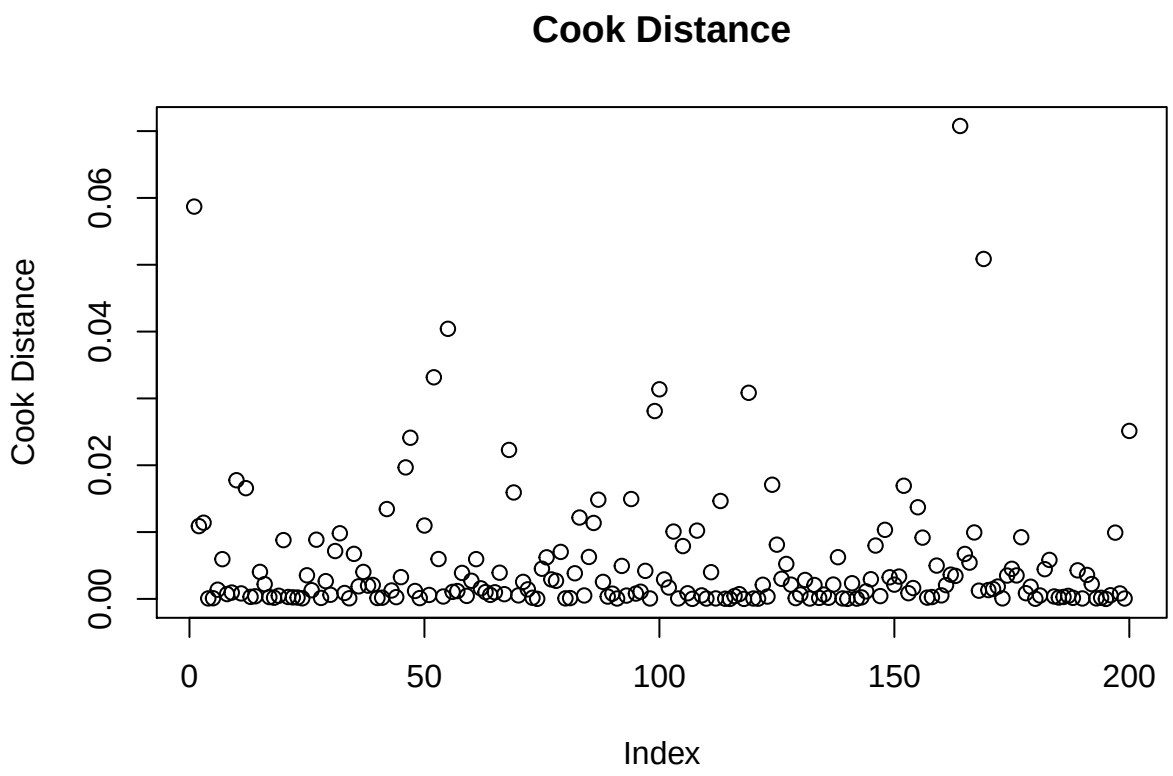
```
#--- Display the value of sales, airplay, attract, adverts, and the standardized residual, for those ca
album2.total[album2.total$large.residual,c("sales", "airplay", "attract", "adverts", "standardized.resi
```

```
##     sales airplay attract  adverts standardized.residuals
## 1     330      43      10   10.256               2.177404
## 2     120      28       7  985.685              -2.323083
## 10    300      40       7  174.093               2.130289
## 47     40      25       8  102.568              -2.460996
## 52    190      12       4  405.913               2.099446
## 55    190      33       8 1542.329              -2.455913
## 61    300      30       7  579.321               2.104079
## 68     70      37       7   56.895              -2.363549
## 100   250       5       7 1000.000               2.095399
## 164   120      53       8    9.104              -2.628814
## 169   360      42       8  145.585               3.093333
## 200   110      20       9  785.694              -2.088044
```

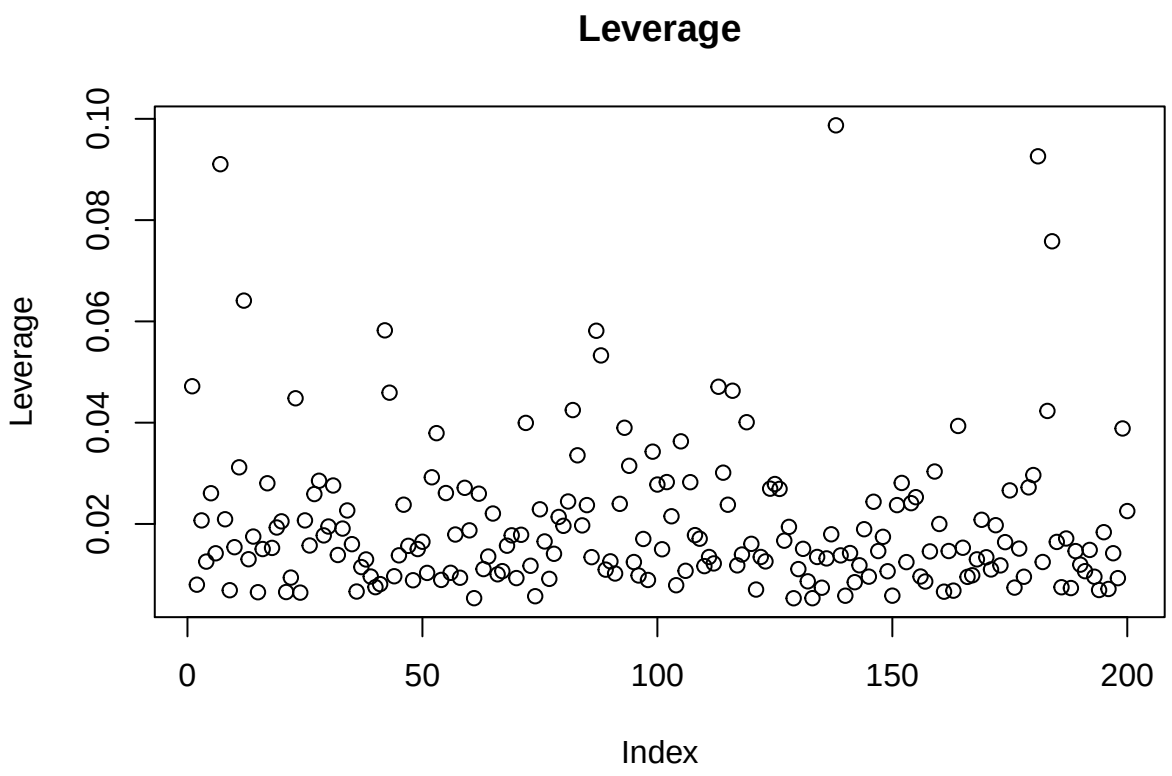
```
#--- Cook's distance, leverage and covariance ratio for cases with large residuals.---------
```

Diagnostic Plots for influential observations

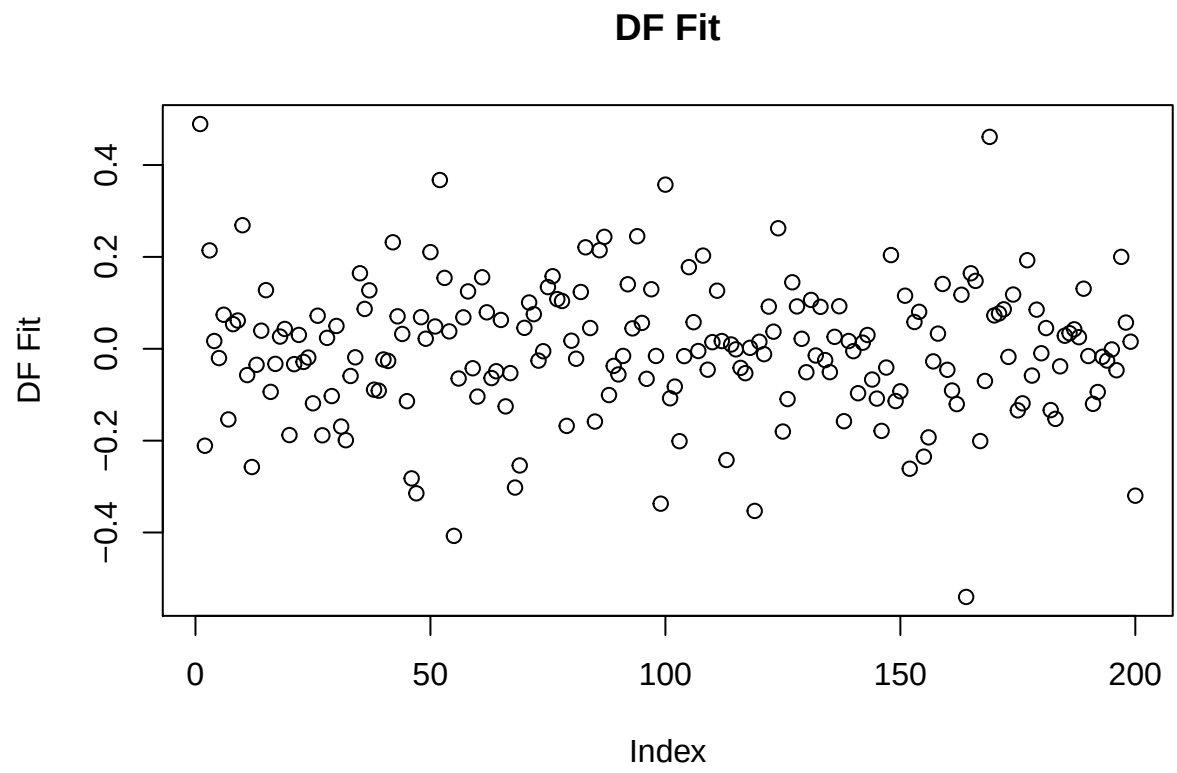
```r
#--- Cook's distance
plot(album2.total$cooks.distance, main = "Cook Distance", ylab = "Cook Distance")
```



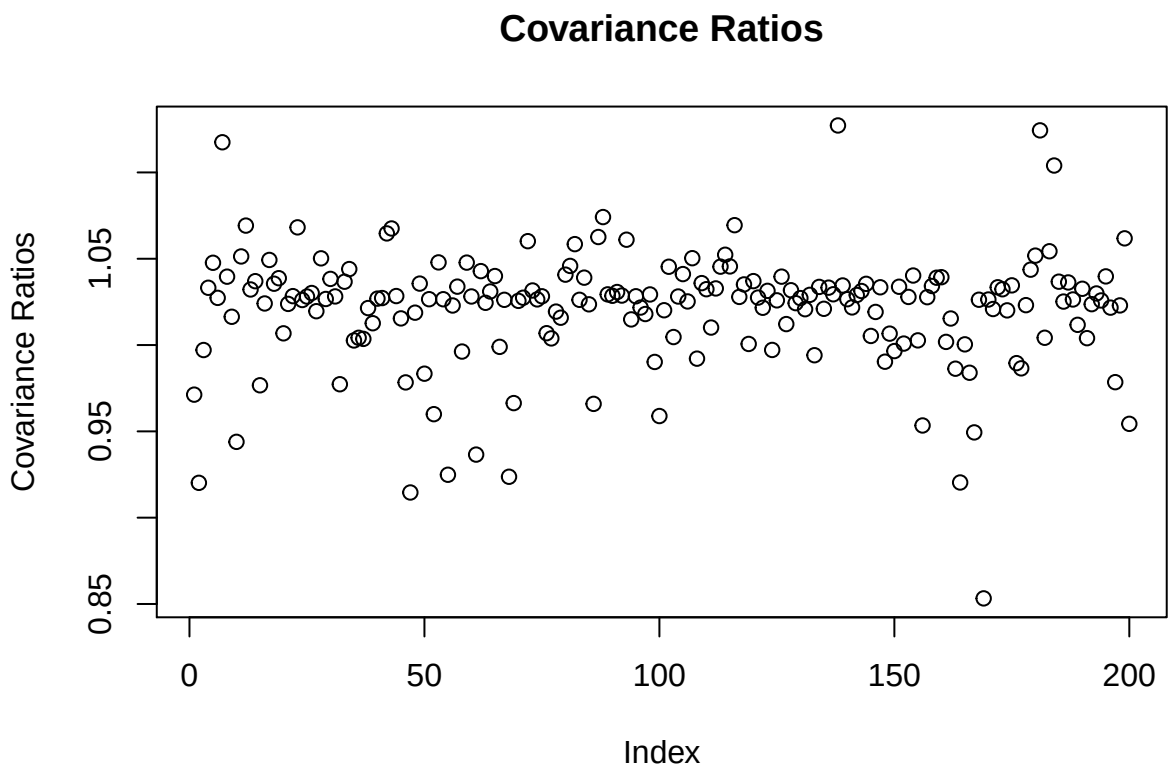

```r
#--- Leverage
plot(album2.total$leverage, main = "Leverage", ylab = "Leverage")
```



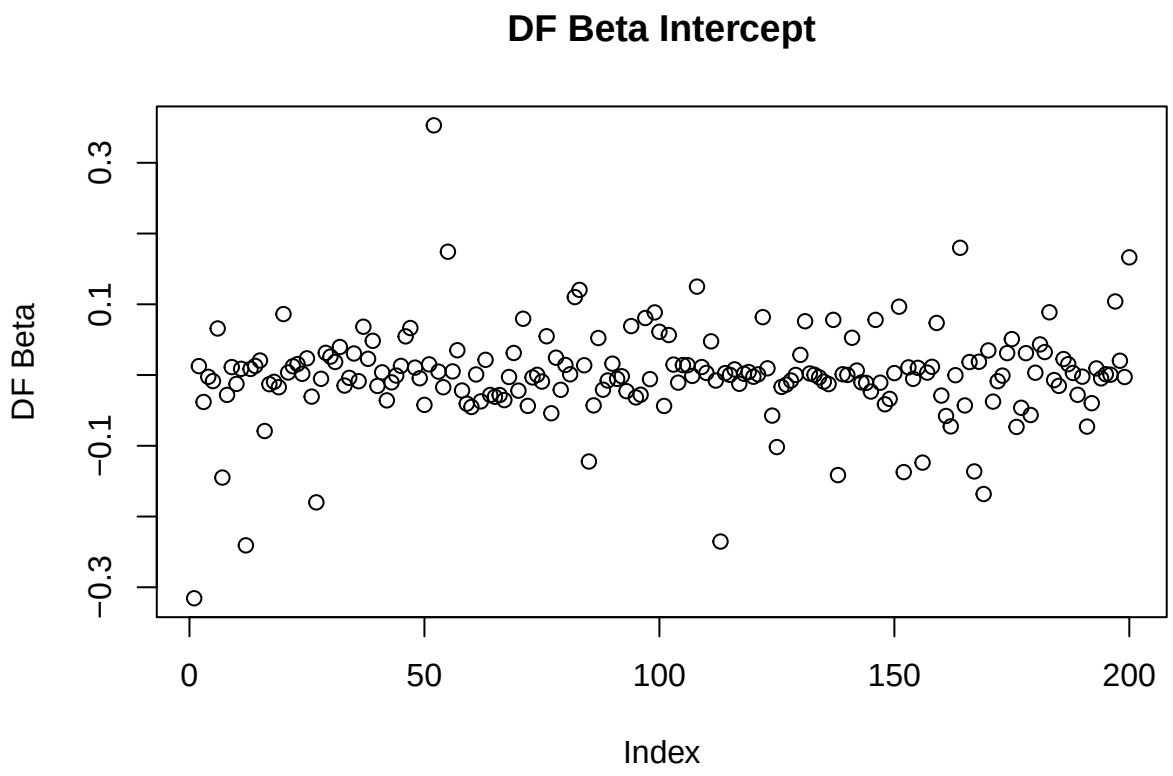

```
#--- DF Fit
plot(album2.total$dffit, main = "DF Fit", ylab = "DF Fit")
```



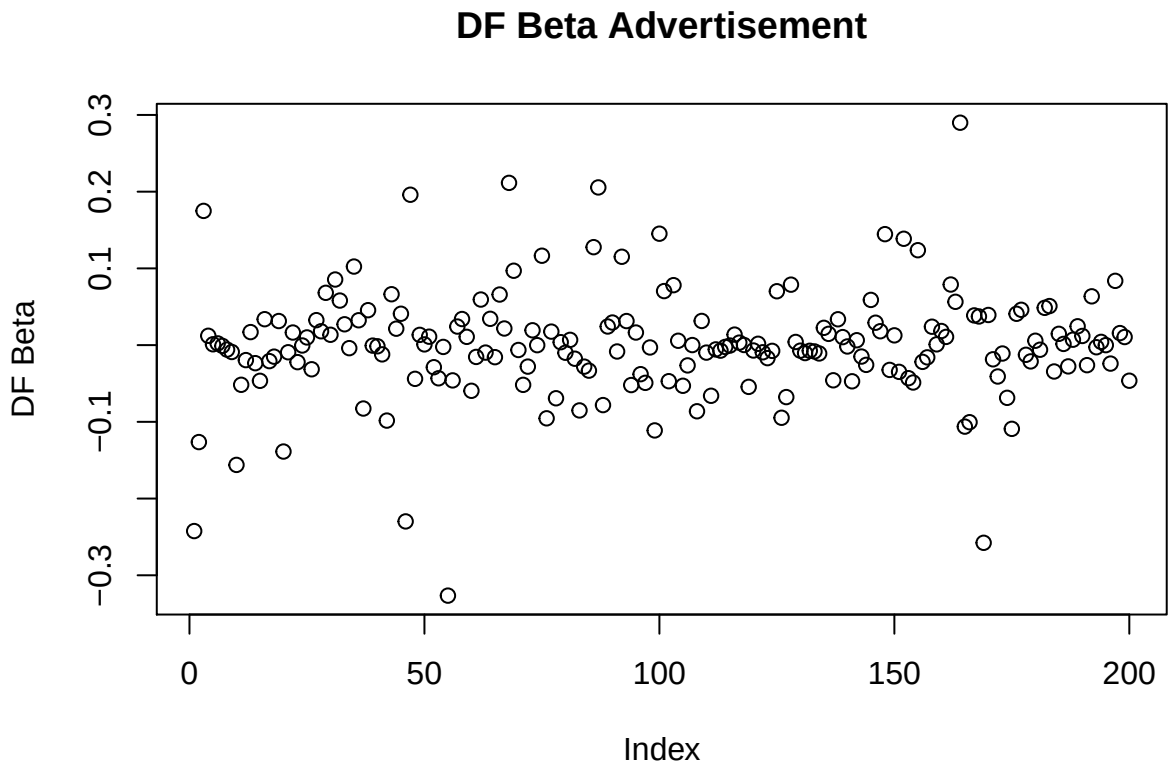
```r
#--- Cov Ratio
plot(album2.total$covariance.ratios, main = "Covariance Ratios", ylab = "Covariance Ratios")
```



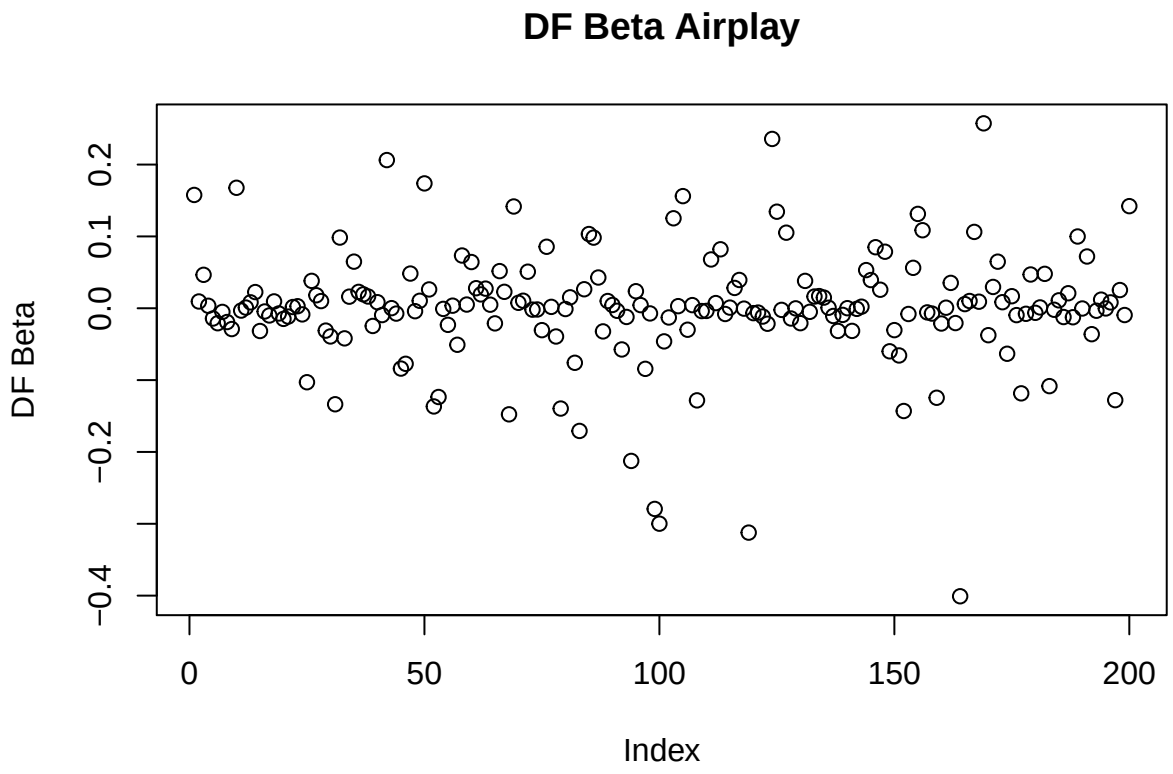

```r
#--- Df Beta intercept and each IV
plot(album2.total$dfb.intercept, main = "DF Beta Intercept", ylab = "DF Beta")
```



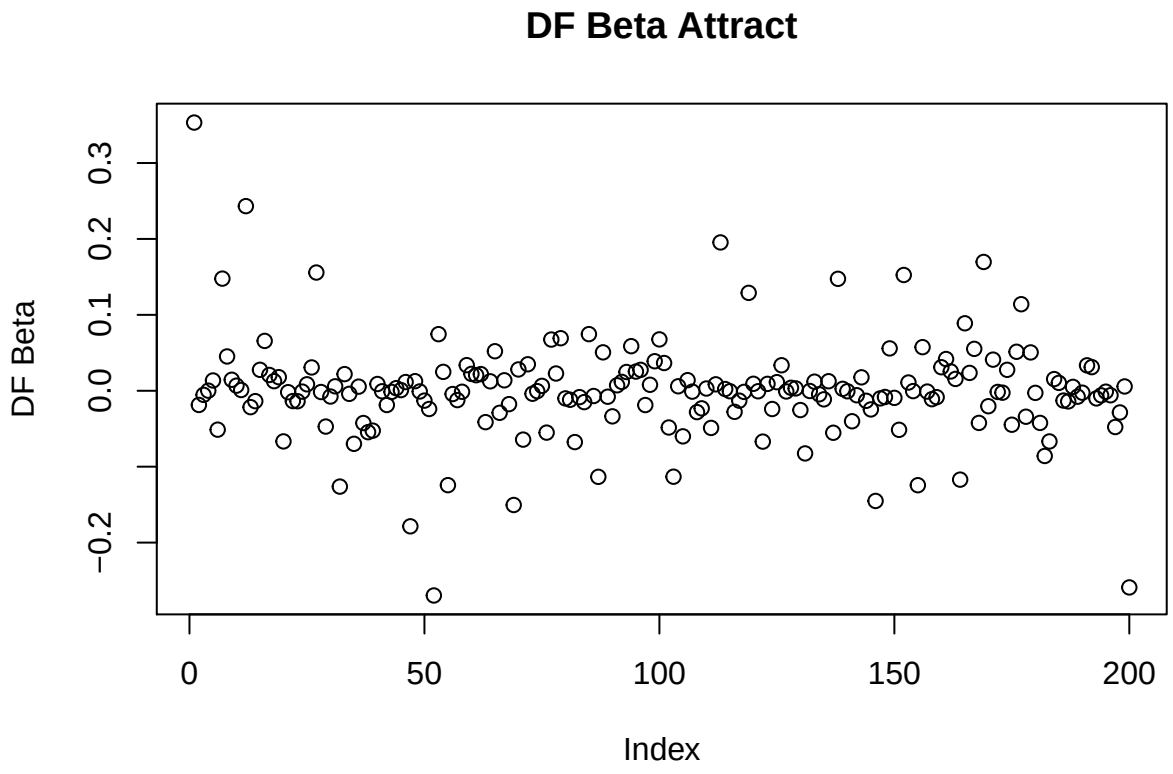

```
plot(album2.total$dfb.adverts, main = "DF Beta Advertisement", ylab = "DF Beta")
```




```
plot(album2.total$dfb.airplay, main = "DF Beta Airplay", ylab = "DF Beta")
```




```
plot(album2.total$dfb.attract, main = "DF Beta Attract", ylab = "DF Beta")
```




```
#---- list of Cook distance, Leverage and Covariance Ratio for large residuals
album2.total[album2.total$large.residual , c("cooks.distance", "leverage", "covariance.ratios")]
```

```
##     cooks.distance    leverage covariance.ratios
## 1      0.058703882 0.047190526         0.9712750
## 2      0.010889432 0.008006536         0.9201832
## 10     0.017756472 0.015409738         0.9439200
## 47     0.024115188 0.015677123         0.9145800
## 52     0.033159177 0.029213132         0.9599533
## 55     0.040415897 0.026103520         0.9248580
## 61     0.005948358 0.005345708         0.9365377
## 68     0.022288983 0.015708852         0.9236983
## 100    0.031364021 0.027779409         0.9588774
## 164    0.070765882 0.039348661         0.9203731
## 169    0.050867000 0.020821154         0.8532470
## 200    0.025134553 0.022539842         0.9543502
```