Chapter 14: Regression Diagnostics and Structure

Ram Gopal, Dan Philps, and Tillman Weyde

2022

Contents

Load packages

```
library(car)
library(caret)
library(ggplot2)
library(leaps)
library(MASS)
library(corrgram)
set.seed(987654321)
```

We will conduct a variety of diagnostic tests to ensure that all the key assumptions invoked in developing the model are met and these use the diagnostic outcomes to aid in developing a "good" structure for the regression model. In our context, "good" refers to satisfying the assumptions and enhancing the fit of the model. This process also enables us to move the final regression structure we employ closer to the "true" data generation process that creates the data we study.

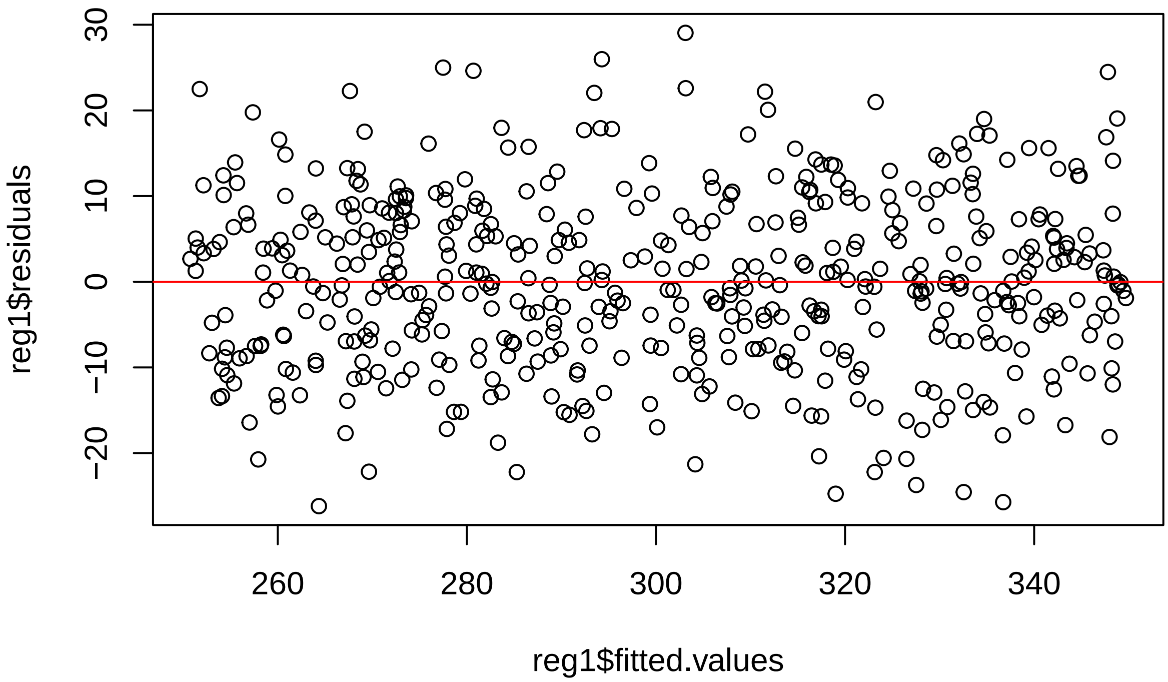
Diagnostics

Perfect model

```
x = runif(500,1,100)
y = 250 + x + rnorm(500,0,10)
reg1 = lm(y~x)
coef(reg1)
```

```
## (Intercept)           x
##     249.607       1.001
```

```
plot(reg1$fitted.values,reg1$residuals)
abline(h=0,col="red")
```

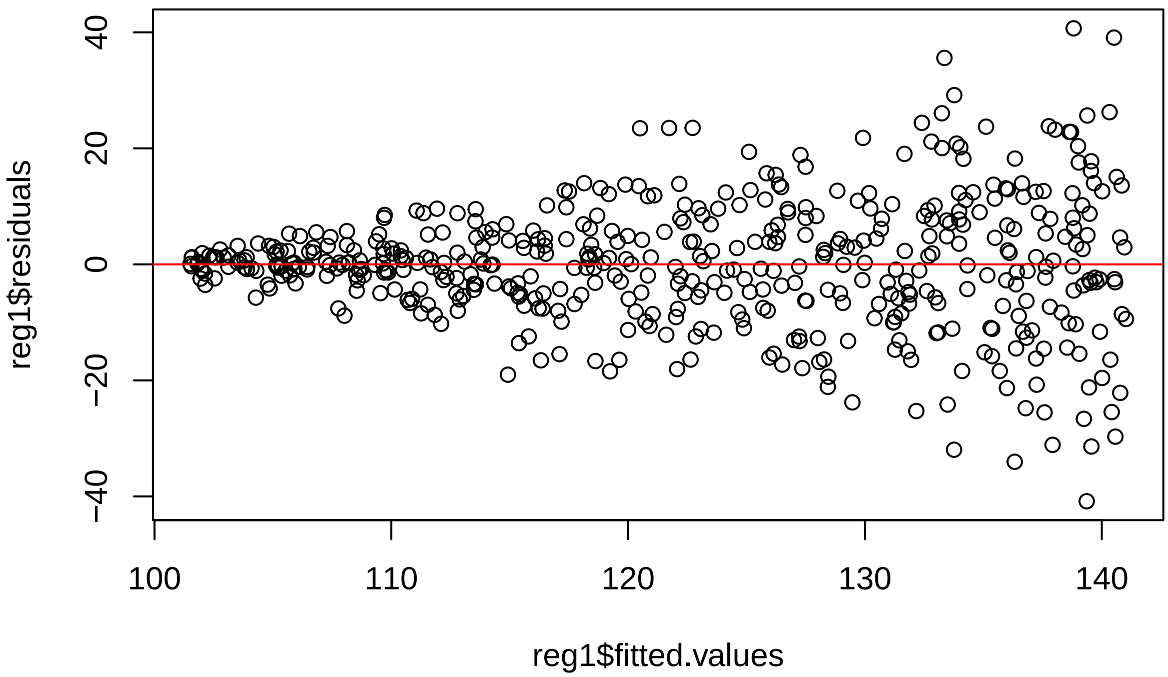


Heteroskedasticity

```
x = runif(500,1,20)
y = 100+2*x + x*rnorm(500)
reg1 = lm(y~x)
coef(reg1)
```

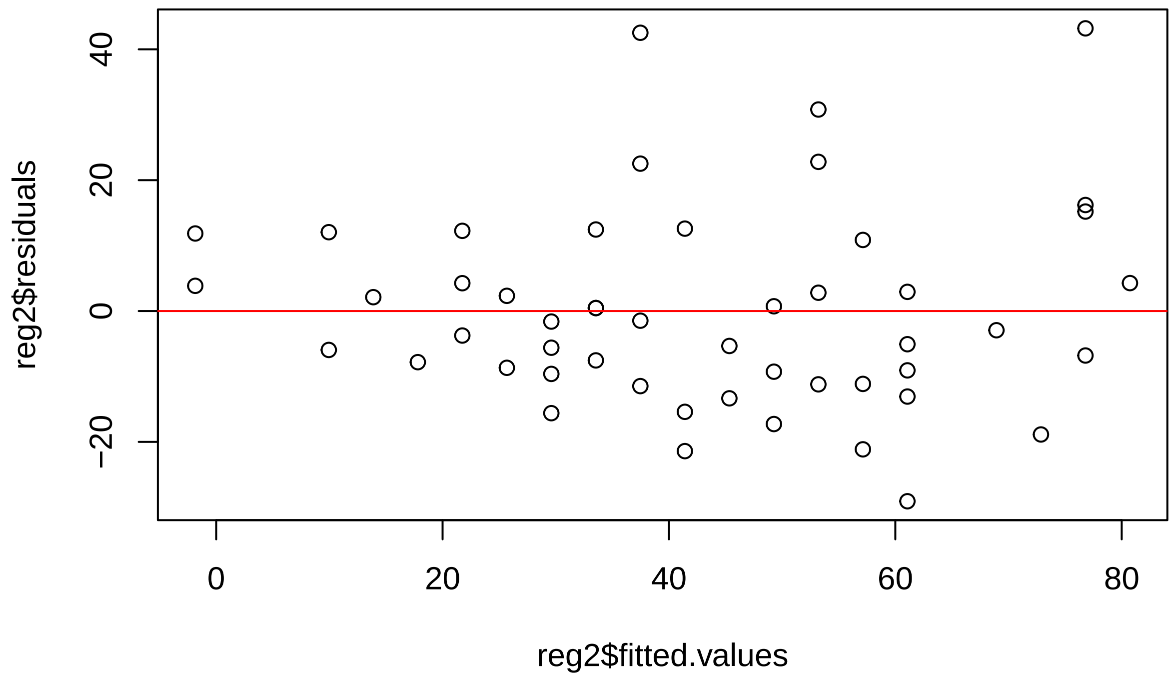
```
## (Intercept)           x
##      99.408       2.083
```

```
plot(reg1$fitted.values,reg1$residuals)
abline(h=0,col="red")
```



Examine cars data

```
cars = read.csv("../../data/cars.csv")
reg2 = lm(dist ~ speed, data=cars)
plot(reg2$fitted.values,reg2$residuals)
abline(h=0,col="red")
```




```
summary(reg2)$r.squared
```

```
## [1] 0.6511
```

```
coef(reg2)
```

```
## (Intercept)       speed
##     -17.579       3.932
```

Detecting heteroskedasticity

```
ncvTest(reg1)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 147.1, Df = 1, p = <0.0000000000000002
```

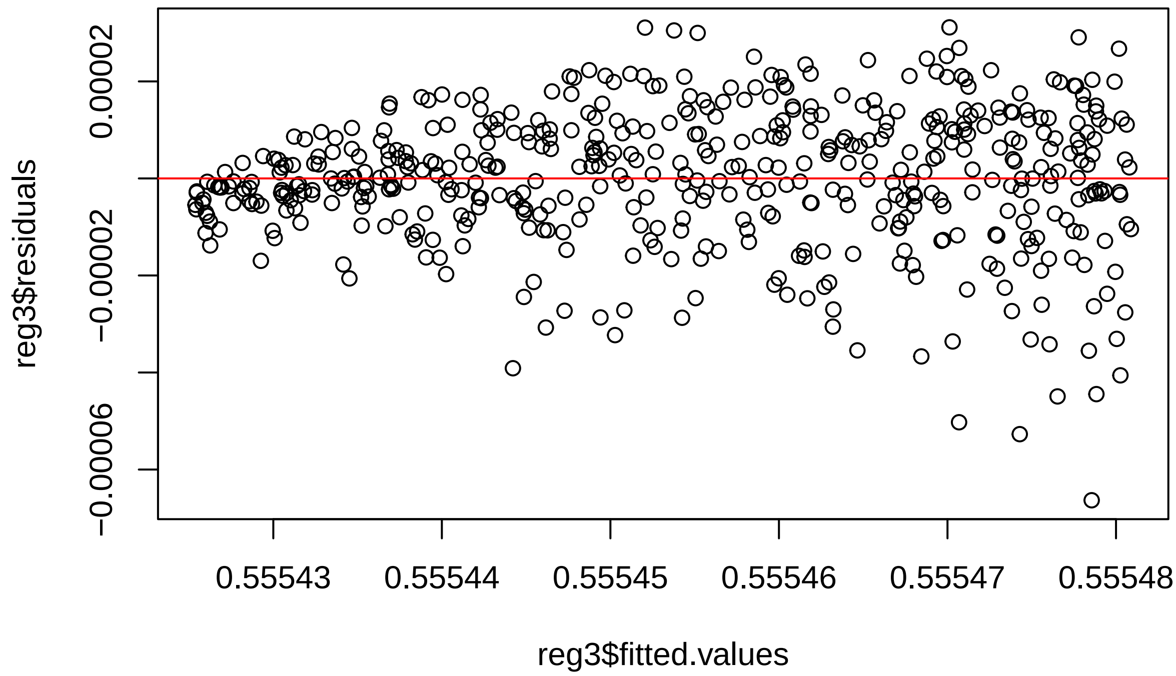
```
ncvTest(reg2)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 4.65, Df = 1, p = 0.031
```

Box-Cox transformation

- Box-Cox tranform the y variable in the first regression, rerun the model with the new y variable, and assess if the problem of heteroskedasticity is alleviated.

```
y1 = predict(BoxCoxTrans(y),y)
reg3 = lm(y1~x)
plot(reg3$fitted.values,reg3$residuals)
abline(h=0,col="red")
```




```
ncvTest(reg3)
```

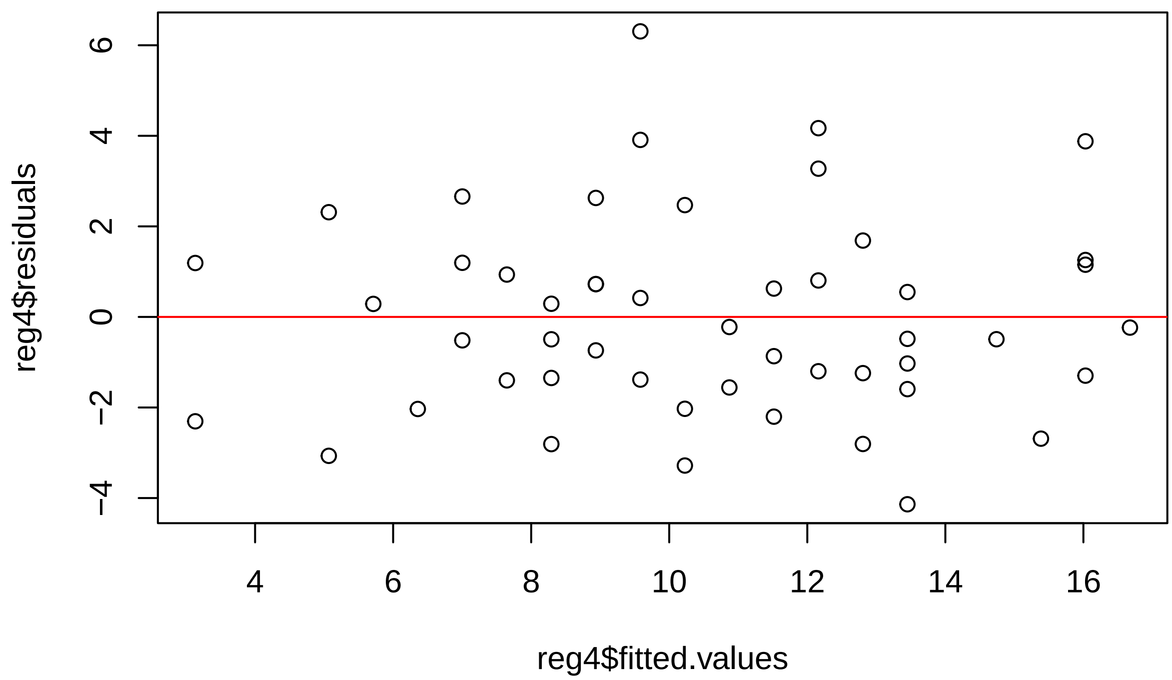
```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 63.18, Df = 1, p = 0.0000000000000019
```

```
summary(reg3)$r.squared
```

```
## [1] 0.5787
```

- Repeat for the cars data.

```r
cars$dist1 = predict(BoxCoxTrans(cars$dist),cars$dist)
reg4 = lm(dist1~speed,data=cars)
plot(reg4$fitted.values,reg4$residuals)
abline(h=0,col="red")
```




```r
ncvTest(reg4)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 0.01205, Df = 1, p = 0.91
```

```r
summary(reg4)$r.squared
```

```
## [1] 0.7094
```

```r
coef(reg4)
```

```
## (Intercept)       speed
##      0.5541      0.6448
```

Extreme Values

Let us assess the impact of an extreme value on the coefficient estimates with an example. We will run two regressions and in the second one we introduce an extreme value for a single x observation.

```r
x = runif(500,1,100)
y = 250 + x + rnorm(500,0,10)
reg1 = lm(y~x)
reg1$coefficients
```

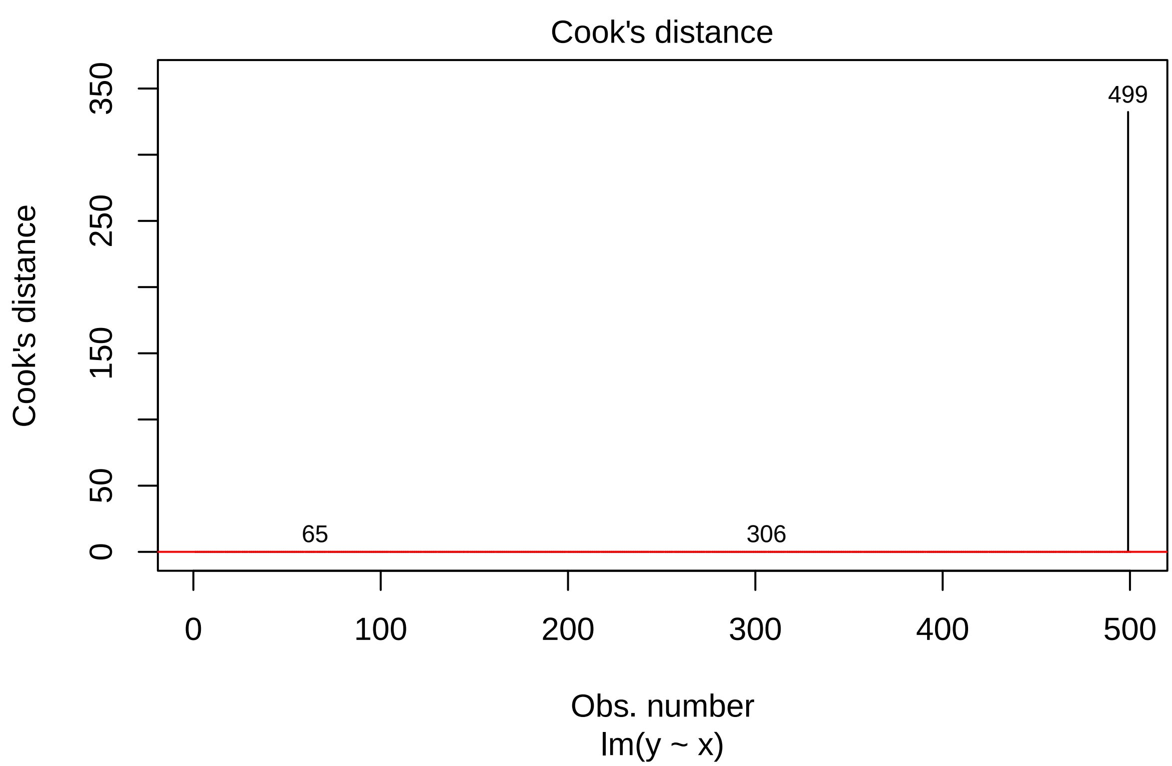
```
## (Intercept)           x
##     249.735       1.002
```

```r
x[499] = 860
reg1 = lm(y~x)
reg1$coefficients
```

```
## (Intercept)           x
##    281.0817      0.3682
```

Detecting extreme values with Cook's Distance

```r
cd = cooks.distance(reg1)
cutoff = 4/500
plot(reg1,which=4,cook.levels = cutoff)
abline(h=cutoff,col="red")
```



* Rerun the model by dropping the extreme values.

```r
reg2 = lm(y[-c(159,309,499)]~x[-c(159,309,499)])
reg2$coefficients
```

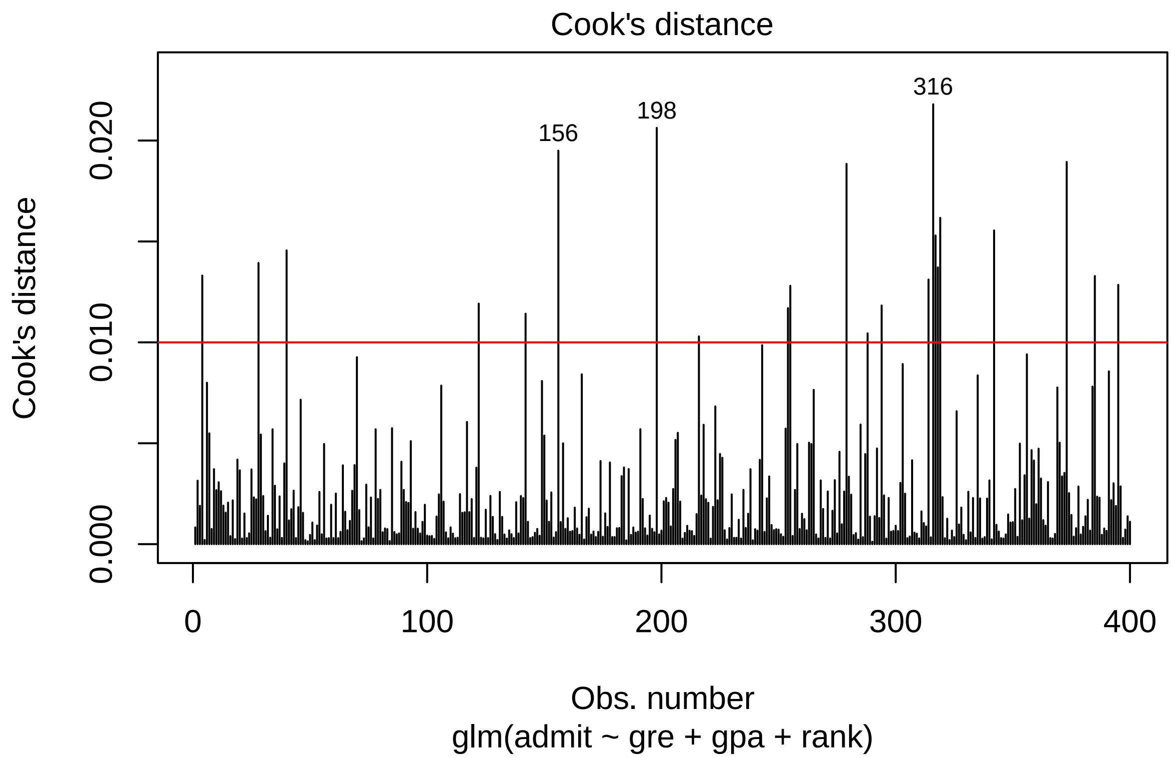
```
##         (Intercept) x[-c(159, 309, 499)]
##             249.703                1.002
```

```r
admit <- read.csv("../../data/admit.csv")
breg1 = glm(admit~gre+gpa+rank,data=admit,family = "binomial")
round(breg1$coefficients,3)
```

Logistic regression example

```
## (Intercept)         gre         gpa        rank
##      -3.450       0.002       0.777      -0.560
```

```r
z = cooks.distance(breg1)
cutoff = 4/nrow(admit)
plot(breg1,which=4,cook.levels = cutoff)
abline(h=cutoff,col="red")
```



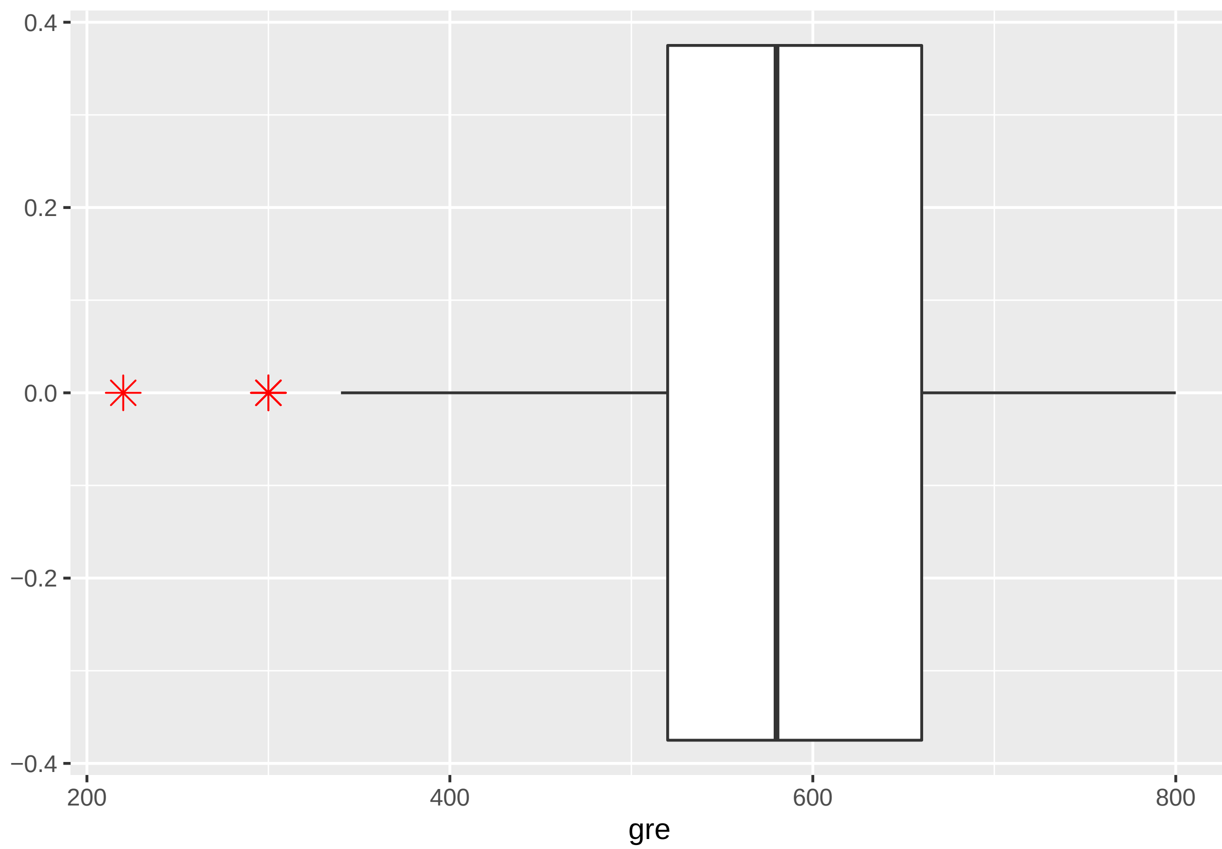
* Model without the extreme values

```
breg1 = glm(admit~gre+gpa+rank,data=admit[-c(156,198,316),],family = "binomial")
round(breg1$coefficients,3)
```

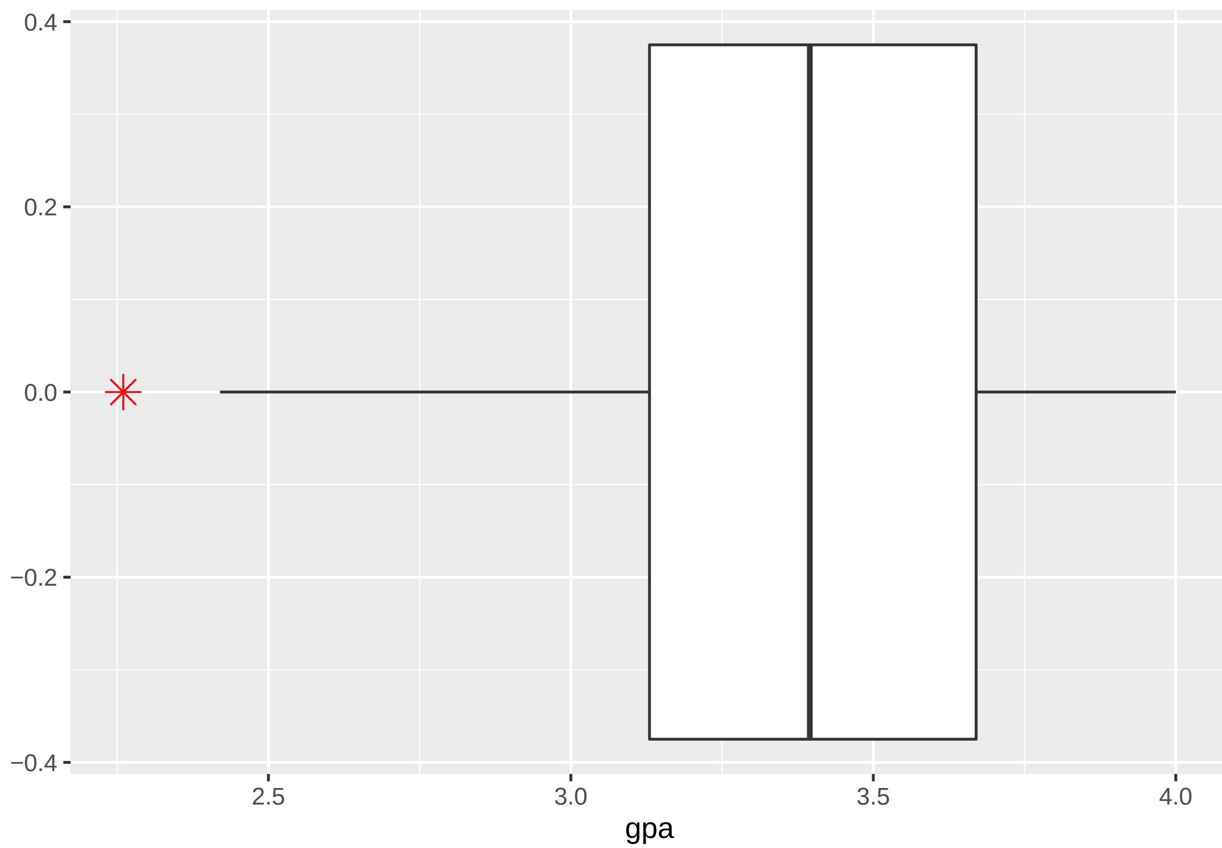
```
## (Intercept)         gre         gpa        rank
##      -3.913       0.003       0.861      -0.607
```

Boxplot for extreme value detection

```
ggplot(admit, aes(x=gre)) +
  geom_boxplot(outlier.colour="red", outlier.shape=8,
                outlier.size=4)
```




```
ggplot(admit, aes(x=gpa)) +
  geom_boxplot(outlier.colour="red", outlier.shape=8,
                outlier.size=4)
```



Multicollinearity

We will introduce correlation between the two input variables through the variable lambda in the code below. You can experiment with the effects of multicollinearity by changing the values of lambda:

```
x1 = runif(500,1,10)
lambda = 0.7
x2 = (lambda*x1) + (1-lambda)*runif(500,1,10)
cor(x1,x2)
```

```
## [1] 0.9196
```

VIF

```
y = 2*x1 + x2 + rnorm(500,0,10)
reg1 = lm(y~x1+x2)
round(reg1$coefficients,3)
```

```
## (Intercept)          x1          x2
##       1.747       2.288       0.478
```

```
vif(reg1)
```

```
##    x1    x2
## 6.479 6.479
```

The following code illustrates the computation of VIF in our example.

```
reg2 = lm(x1~x2)
r2_1 = summary(reg2)$r.squared
r2_1
```

```
## [1] 0.8457
```

```
vif_x1 = 1/(1-r2_1)
vif_x1
```

```
## [1] 6.479
```

Low values of VIF below indicate that we do not have to worry about multicollinearity in the logistic regression example.

```
round(cor(admit[,-1]),3)
```

```
##         gre    gpa   rank
## gre   1.000  0.384 -0.123
## gpa   0.384  1.000 -0.057
## rank -0.123 -0.057  1.000
```

```
vif(breg1)
```

```
##   gre   gpa  rank
## 1.121 1.124 1.004
```

Regression Structure

Illustrative example

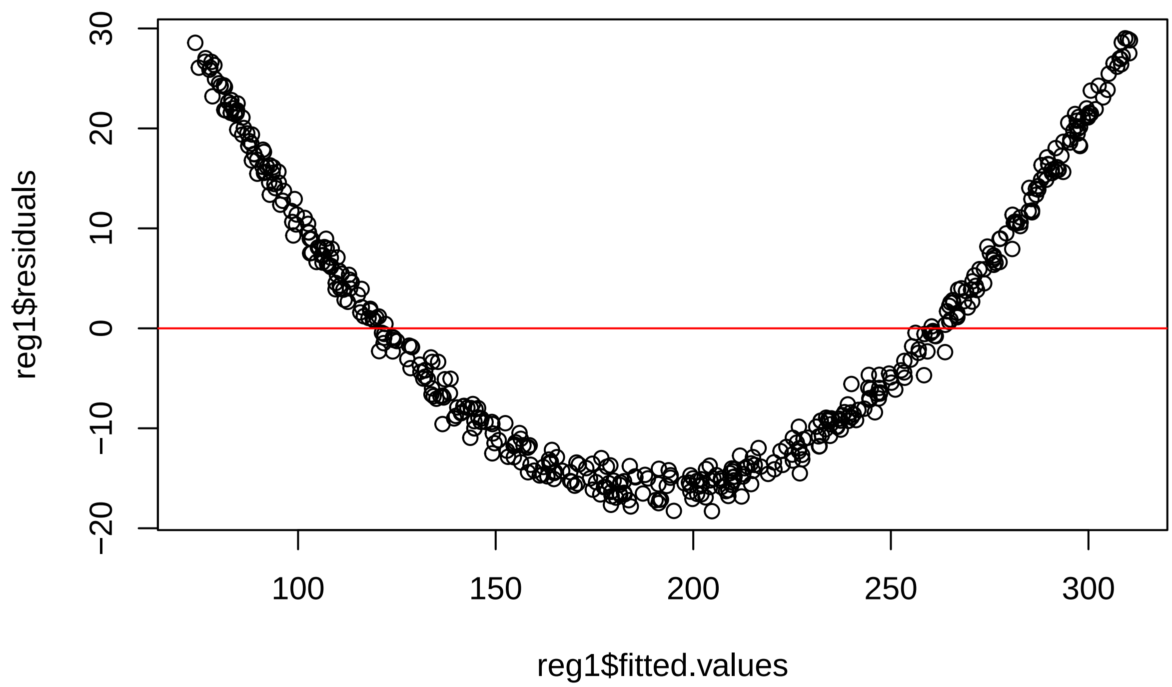
```
x = runif(500,1,20)
y = 100+2*x +0.5*x^2 + rnorm(500)
reg1 = lm(y~x)
summary(reg1)$r.squared
```

```
## [1] 0.9641
```

```
reg1$coefficients
```

```
## (Intercept)           x
##       61.48       12.45
```

```r
plot(reg1$fitted.values,reg1$residuals)
abline(h=0,col="red")
```



Box-Tidwell tranformation

```r
boxTidwell(y~x)
```

```
##  MLE of lambda Score Statistic (z)            Pr(>|z|)
##            1.8                  108 <0.0000000000000002 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## iterations =  3
```

A more comprehensive analysis with the MASchools.csv:

```r
MASchools <- read.csv("../../data/MASchools.csv")
df = MASchools[,c(13,7,8,9,11,15)]
df1 = df[complete.cases(df),]
reg1 = lm(score4~exptot+scratio+special+stratio+salary,data=df1)
summary(reg1)$r.squared
```

```
## [1] 0.2755
```

```
ncvTest(reg1)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 14.36, Df = 1, p = 0.00015
```

- Box-Tidwell test

```
boxTidwell(score4~exptot+scratio+special+stratio+salary,data=df1)
```

```
##          MLE of lambda Score Statistic (z) Pr(>|z|)
## exptot            -1.7                 1.5    0.132
## scratio           -2.2                 0.4    0.708
## special           -1.8                 1.0    0.318
## stratio            4.6                -3.0    0.003 **
## salary             6.5                 3.3   0.0009 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## iterations =  26
```

- Assess the non-linear model based on the test.

```
reg2 = lm(score4~exptot+scratio+special+stratio+salary+I(stratio^4)+I(salary^6),data=df1)
summary(reg2)$r.squared
```

```
## [1] 0.3352
```

```
ncvTest(reg2)
```

```
## Non-constant Variance Score Test
## Variance formula: ~ fitted.values
## Chisquare = 5.04, Df = 1, p = 0.025
```

- Logistic regression example

```
breg1 = glm(admit~gre+gpa+rank,data=admit,family = "binomial")
logodds = breg1$linear.predictors
boxTidwell(logodds~gre+gpa+rank,data=admit)
```

```
##      MLE of lambda Score Statistic (z) Pr(>|z|)
## gre              1                 0.4      0.7
## gpa              1                -0.7      0.5
## rank             1                -4.5 0.000006 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## iterations =  0
```

Interaction terms

```
x1 = runif(500,1,20)
x2 = runif(500,1,20)
y = x1+4*x2+0.5*x1*x2 + rnorm(500)
reg1 = lm(y~x1+x2)
reg1$coefficients
```

```
## (Intercept)          x1          x2
##     -54.052       6.227       9.215
```

ANOVA to detect important interactions

```
res = step(reg1,~.^2)
```

```
## Start:  AIC=2757
## y ~ x1 + x2
##
##         Df Sum of Sq     RSS  AIC
## + x1:x2  1    122040     528   35
## <none>                122568 2757
## - x1     1    598581  721149 3641
## - x2     1   1309022 1431590 3984
##
## Step:  AIC=35.47
## y ~ x1 + x2 + x1:x2
##
##         Df Sum of Sq    RSS  AIC
## <none>                  528   35
## - x1:x2  1    122040 122568 2757
```

```
res$anova
```

```
##      Step Df Deviance Resid. Df Resid. Dev     AIC
## 1         NA       NA       497   122568.1 2756.91
## 2 + x1:x2 -1   122040       496      528.2   35.47
```

- MASchools data

```
reg2 = lm(score4~exptot+scratio+special+stratio+salary+I(stratio^4)+I(salary^6),data=df1)
res = step(reg2,~.^2)
```

```
## Start:  AIC=957.1
## score4 ~ exptot + scratio + special + stratio + salary + I(stratio^4) +
##     I(salary^6)
##
##                              Df Sum of Sq   RSS AIC
## + exptot:scratio              1      1256 28051 951
```

```
## + special:stratio           1      1127 28180 952
## + special:I(stratio^4)      1      1094 28214 952
## + scratio:special           1       645 28662 955
## - stratio                   1        49 29356 955
## - scratio                   1       137 29444 956
## + scratio:stratio           1       475 28832 956
## + scratio:I(stratio^4)      1       469 28838 956
## - salary                    1       220 29527 957
## + exptot:salary             1       408 28899 957
## + salary:I(salary^6)        1       400 28907 957
## <none>                                29307 957
## + special:salary            1       279 29028 957
## + exptot:I(salary^6)        1       275 29032 957
## + exptot:I(stratio^4)       1       256 29051 957
## + exptot:stratio            1       181 29126 958
## + scratio:salary            1       131 29176 958
## + scratio:I(salary^6)       1       128 29179 958
## + stratio:I(stratio^4)      1        94 29213 959
## + exptot:special            1        73 29234 959
## + stratio:salary            1        63 29244 959
## + salary:I(stratio^4)       1        45 29262 959
## + special:I(salary^6)       1        40 29267 959
## + stratio:I(salary^6)       1         2 29305 959
## + I(stratio^4):I(salary^6)  1         1 29306 959
## - I(stratio^4)              1       907 30214 961
## - special                   1       940 30247 961
## - exptot                    1      1600 30907 965
## - I(salary^6)               1      1972 31279 967
##
## Step:  AIC=951
## score4 ~ exptot + scratio + special + stratio + salary + I(stratio^4) +
##     I(salary^6) + exptot:scratio
##
##                            Df Sum of Sq   RSS AIC
## + special:I(stratio^4)      1      1412 26638 943
## + special:stratio           1      1221 26830 945
## + scratio:special           1       599 27451 949
## - stratio                   1       105 28156 950
## - salary                    1       156 28207 950
## + salary:I(salary^6)        1       424 27627 950
## + stratio:I(stratio^4)      1       343 27708 951
## <none>                                28051 951
## + scratio:salary            1       273 27778 951
## + scratio:I(salary^6)       1       230 27821 951
## + special:salary            1       216 27835 952
## + stratio:I(salary^6)       1       126 27924 952
## + I(stratio^4):I(salary^6)  1        96 27955 952
## + exptot:special            1        49 28002 953
## + salary:I(stratio^4)       1        32 28019 953
## + exptot:stratio            1        31 28020 953
## + special:I(salary^6)       1        27 28023 953
## + scratio:I(stratio^4)      1        27 28024 953
## + stratio:salary            1        18 28032 953
## + scratio:stratio           1        15 28036 953
```

```
## + exptot:salary              1        11 28040 953
## + exptot:I(salary^6)         1         8 28043 953
## + exptot:I(stratio^4)        1         0 28050 953
## - special                    1       831 28882 954
## - I(stratio^4)               1      1145 29196 956
## - exptot:scratio             1      1256 29307 957
## - I(salary^6)                1      1935 29986 961
##
## Step:  AIC=943.4
## score4 ~ exptot + scratio + special + stratio + salary + I(stratio^4) +
##     I(salary^6) + exptot:scratio + special:I(stratio^4)
##
##                             Df Sum of Sq   RSS AIC
## + scratio:special            1       956 25682 939
## + scratio:salary             1       429 26209 942
## + scratio:I(salary^6)        1       401 26238 943
## + exptot:I(stratio^4)        1       377 26261 943
## - salary                     1       226 26865 943
## + salary:I(salary^6)         1       337 26301 943
## <none>                                   26638 943
## + special:salary             1        91 26548 945
## + exptot:stratio             1        79 26560 945
## + salary:I(stratio^4)        1        77 26562 945
## + I(stratio^4):I(salary^6)   1        54 26585 945
## + scratio:I(stratio^4)       1        26 26613 945
## + stratio:salary             1        20 26619 945
## + exptot:I(salary^6)         1        16 26622 945
## + exptot:salary              1        15 26623 945
## + scratio:stratio            1        14 26625 945
## + special:stratio            1        13 26626 945
## + special:I(salary^6)        1         4 26634 945
## + exptot:special             1         1 26638 945
## + stratio:I(salary^6)        1         1 26638 945
## + stratio:I(stratio^4)       1         0 26638 945
## - stratio                    1       616 27255 946
## - special:I(stratio^4)       1      1412 28051 951
## - exptot:scratio             1      1575 28214 952
## - I(salary^6)                1      2114 28753 956
##
## Step:  AIC=938.6
## score4 ~ exptot + scratio + special + stratio + salary + I(stratio^4) +
##     I(salary^6) + exptot:scratio + special:I(stratio^4) + scratio:special
##
##                             Df Sum of Sq   RSS AIC
## + exptot:I(stratio^4)        1       509 25174 937
## - salary                     1       190 25873 938
## + salary:I(salary^6)         1       311 25371 938
## + scratio:salary             1       281 25401 939
## <none>                                   25682 939
## + scratio:I(salary^6)        1       274 25408 939
## + salary:I(stratio^4)        1       150 25532 939
## + exptot:stratio             1       145 25537 940
## + I(stratio^4):I(salary^6)   1       139 25543 940
## + special:salary             1       126 25556 940
```

```
## + stratio:salary           1        47 25635 940
## + exptot:special           1        39 25644 940
## + special:stratio          1        38 25644 940
## + stratio:I(stratio^4)     1        25 25658 940
## + stratio:I(salary^6)      1        22 25660 940
## + exptot:I(salary^6)       1        15 25667 940
## + special:I(salary^6)      1        14 25669 940
## + exptot:salary            1         6 25676 941
## + scratio:stratio          1         1 25681 941
## + scratio:I(stratio^4)     1         0 25682 941
## - stratio                  1       714 26396 942
## - scratio:special          1       956 26638 943
## - exptot:scratio           1      1559 27241 948
## - special:I(stratio^4)     1      1769 27451 949
## - I(salary^6)              1      1981 27664 950
##
## Step:  AIC=936.9
## score4 ~ exptot + scratio + special + stratio + salary + I(stratio^4) +
##     I(salary^6) + exptot:scratio + special:I(stratio^4) + scratio:special +
##     exptot:I(stratio^4)
##
##                           Df Sum of Sq   RSS AIC
## + exptot:stratio           1       754 24420 933
## + salary:I(salary^6)       1       369 24805 936
## - salary                   1       203 25376 936
## <none>                                 25174 937
## + stratio:I(stratio^4)     1       261 24912 937
## + scratio:salary           1       179 24995 938
## + stratio:I(salary^6)      1       172 25002 938
## + scratio:I(salary^6)      1       163 25010 938
## + special:stratio          1       159 25014 938
## + I(stratio^4):I(salary^6) 1       123 25051 938
## + special:salary           1       110 25063 938
## - stratio                  1       472 25646 938
## + stratio:salary           1        63 25111 938
## + salary:I(stratio^4)      1        53 25121 938
## + scratio:stratio          1        41 25132 939
## - exptot:I(stratio^4)      1       509 25682 939
## + scratio:I(stratio^4)     1        33 25140 939
## + special:I(salary^6)      1         9 25165 939
## + exptot:I(salary^6)       1         5 25168 939
## + exptot:special           1         4 25169 939
## + exptot:salary            1         1 25172 939
## - exptot:scratio           1       671 25844 940
## - scratio:special          1      1088 26261 943
## - I(salary^6)              1      1991 27164 949
## - special:I(stratio^4)     1      2277 27450 951
##
## Step:  AIC=933.2
## score4 ~ exptot + scratio + special + stratio + salary + I(stratio^4) +
##     I(salary^6) + exptot:scratio + special:I(stratio^4) + scratio:special +
##     exptot:I(stratio^4) + exptot:stratio
##
##                           Df Sum of Sq   RSS AIC
```

```
## - salary                    1       140 24559 932
## + scratio:I(salary^6)       1       377 24042 932
## + scratio:salary            1       361 24059 932
## + special:stratio           1       293 24127 933
## <none>                                 24420 933
## + salary:I(salary^6)        1       257 24162 933
## + I(stratio^4):I(salary^6)  1       182 24238 934
## + special:salary            1       143 24277 934
## + stratio:I(stratio^4)      1       126 24294 934
## + salary:I(stratio^4)       1        92 24328 934
## + stratio:I(salary^6)       1        66 24354 935
## + scratio:I(stratio^4)      1        65 24354 935
## + special:I(salary^6)       1        38 24382 935
## + scratio:stratio           1        36 24384 935
## + exptot:I(salary^6)        1        34 24386 935
## + stratio:salary            1        17 24403 935
## + exptot:salary             1        17 24403 935
## + exptot:special            1         6 24414 935
## - exptot:stratio            1       754 25174 937
## - scratio:special           1      1043 25463 939
## - exptot:I(stratio^4)       1      1117 25537 940
## - exptot:scratio            1      1166 25586 940
## - I(salary^6)               1      1828 26247 945
## - special:I(stratio^4)      1      2699 27119 951
##
## Step:  AIC=932.3
## score4 ~ exptot + scratio + special + stratio + I(stratio^4) +
##     I(salary^6) + exptot:scratio + special:I(stratio^4) + scratio:special +
##     exptot:I(stratio^4) + exptot:stratio
##
##                            Df Sum of Sq   RSS AIC
## + special:stratio           1       381 24178 931
## + scratio:I(salary^6)       1       296 24263 932
## <none>                                 24559 932
## + stratio:I(stratio^4)      1       143 24417 933
## + salary                    1       140 24420 933
## + I(stratio^4):I(salary^6)  1       131 24428 933
## + exptot:I(salary^6)        1       130 24430 933
## + scratio:I(stratio^4)      1        74 24485 934
## + scratio:stratio           1        53 24507 934
## + special:I(salary^6)       1        40 24519 934
## + stratio:I(salary^6)       1        39 24521 934
## + exptot:special            1         9 24550 934
## - exptot:stratio            1       817 25376 936
## - scratio:special           1      1072 25631 938
## - exptot:I(stratio^4)       1      1184 25743 939
## - exptot:scratio            1      1245 25805 939
## - special:I(stratio^4)      1      2648 27207 949
## - I(salary^6)               1      7802 32361 982
##
## Step:  AIC=931.4
## score4 ~ exptot + scratio + special + stratio + I(stratio^4) +
##     I(salary^6) + exptot:scratio + special:I(stratio^4) + scratio:special +
##     exptot:I(stratio^4) + exptot:stratio + special:stratio
```

```
##
##                            Df Sum of Sq   RSS AIC
## + scratio:I(salary^6)       1       358 23820 931
## <none>                                  24178 931
## - special:stratio           1       381 24559 932
## + scratio:I(stratio^4)      1        97 24081 933
## + I(stratio^4):I(salary^6)  1        79 24099 933
## + exptot:special            1        78 24100 933
## + salary                    1        51 24127 933
## + exptot:I(salary^6)        1        49 24129 933
## + scratio:stratio           1        44 24134 933
## + special:I(salary^6)       1        43 24135 933
## + stratio:I(stratio^4)      1        37 24141 933
## + stratio:I(salary^6)       1        30 24148 933
## - scratio:special           1       598 24776 934
## - special:I(stratio^4)      1       925 25103 936
## - exptot:stratio            1       954 25132 937
## - exptot:scratio            1      1223 25401 939
## - exptot:I(stratio^4)       1      1547 25725 941
## - I(salary^6)               1      6462 30640 973
##
## Step:  AIC=930.6
## score4 ~ exptot + scratio + special + stratio + I(stratio^4) +
##     I(salary^6) + exptot:scratio + special:I(stratio^4) + scratio:special +
##     exptot:I(stratio^4) + exptot:stratio + special:stratio +
##     scratio:I(salary^6)
##
##                            Df Sum of Sq   RSS AIC
## <none>                                  23820 931
## - scratio:I(salary^6)       1       358 24178 931
## + scratio:I(stratio^4)      1       127 23693 932
## + salary                    1       102 23717 932
## + special:I(salary^6)       1        76 23743 932
## + scratio:stratio           1        75 23745 932
## - special:stratio           1       444 24263 932
## - scratio:special           1       450 24270 932
## + exptot:I(salary^6)        1        60 23759 932
## + exptot:special            1        40 23780 932
## + I(stratio^4):I(salary^6)  1        28 23791 932
## + stratio:I(stratio^4)      1        14 23805 932
## + stratio:I(salary^6)       1         3 23817 933
## - special:I(stratio^4)      1      1027 24847 936
## - exptot:stratio            1      1196 25015 938
## - exptot:scratio            1      1485 25304 940
## - exptot:I(stratio^4)       1      1776 25595 942
```

```
res$anova
```

```
##                      Step Df Deviance Resid. Df Resid. Dev   AIC
## 1                         NA       NA       178      29307 957.1
## 2       + exptot:scratio -1   1256.5       177      28051 951.0
## 3 + special:I(stratio^4) -1   1412.1       176      26638 943.4
## 4      + scratio:special -1    956.1       175      25682 938.6
## 5  + exptot:I(stratio^4) -1    508.8       174      25174 936.9
```

```
## 6       + exptot:stratio -1    753.7       173      24420 933.2
## 7               - salary  1    139.5       174      24559 932.3
## 8      + special:stratio -1    381.3       173      24178 931.3
## 9  + scratio:I(salary^6) -1    358.5       172      23820 930.6
```

- Logistic regression example

```
breg1 = glm(admit~gre+gpa+rank,data=admit,family = "binomial")
res = step(breg1,~.^2)
```

```
## Start:  AIC=467.4
## admit ~ gre + gpa + rank
##
##            Df Deviance AIC
## + gre:gpa   1      457 467
## <none>              459 467
## + gpa:rank  1      459 469
## + gre:rank  1      459 469
## - gre       1      464 470
## - gpa       1      465 471
## - rank      1      480 486
##
## Step:  AIC=466.6
## admit ~ gre + gpa + rank + gre:gpa
##
##            Df Deviance AIC
## <none>              457 467
## - gre:gpa   1      459 467
## + gpa:rank  1      456 468
## + gre:rank  1      457 469
## - rank      1      478 486
```

```
res$anova
```

```
##        Step Df Deviance Resid. Df Resid. Dev   AIC
## 1           NA       NA       396      459.4 467.4
## 2 + gre:gpa -1    2.844       395      456.6 466.6
```

Variable Selection

Stepwise Regression

```
x1 = runif(500,1,10)
x2 = runif(500,1,10)
y = 2*x1 + x2 + rnorm(500,0,10)
reg1 = lm(y~x1+x2+x1:x2+I(x1^2)+I(x^3))
step(reg1,direction="backward")$anova
```

```
## Start:  AIC=2312
## y ~ x1 + x2 + x1:x2 + I(x1^2) + I(x^3)
```

```
##
##            Df Sum of Sq   RSS  AIC
## - I(x1^2)  1       0.1 49717 2310
## - x1:x2    1      68.6 49786 2310
## <none>                 49717 2312
## - I(x^3)   1     312.2 50029 2313
##
## Step:  AIC=2310
## y ~ x1 + x2 + I(x^3) + x1:x2
##
##          Df Sum of Sq   RSS  AIC
## - x1:x2   1      68.7 49786 2308
## <none>                49717 2310
## - I(x^3)  1     314.8 50032 2311
##
## Step:  AIC=2308
## y ~ x1 + x2 + I(x^3)
##
##          Df Sum of Sq   RSS  AIC
## <none>                49786 2308
## - I(x^3)  1       319 50105 2310
## - x2      1      3446 53232 2340
## - x1      1     15046 64832 2438
```

```
##        Step Df Deviance Resid. Df Resid. Dev  AIC
## 1            NA       NA       494      49717 2312
## 2 - I(x1^2)  1  0.06523       495      49717 2310
## 3   - x1:x2  1 68.72130       496      49786 2308
```

- MASchools data

```
reg1 = lm(score4 ~ exptot + scratio + special + stratio + I(stratio^4) +
    I(salary^6) + exptot:scratio + special:I(stratio^4) + scratio:special +
    exptot:I(stratio^4) + exptot:stratio + special:stratio +
    scratio:I(salary^6),data=df1)
step(reg1,direction="both")$anova
```

```
## Start:  AIC=930.6
## score4 ~ exptot + scratio + special + stratio + I(stratio^4) +
##     I(salary^6) + exptot:scratio + special:I(stratio^4) + scratio:special +
##     exptot:I(stratio^4) + exptot:stratio + special:stratio +
##     scratio:I(salary^6)
##
##                        Df Sum of Sq   RSS AIC
## <none>                              23820 931
## - scratio:I(salary^6)   1       358 24178 931
## - special:stratio       1       444 24263 932
## - scratio:special       1       450 24270 932
## - special:I(stratio^4)  1      1027 24847 936
## - exptot:stratio        1      1196 25015 938
## - exptot:scratio        1      1485 25304 940
## - exptot:I(stratio^4)   1      1776 25595 942
```

```
##   Step Df Deviance Resid. Df Resid. Dev   AIC
## 1      NA       NA       172      23820 930.6
```

- Logistic regression example

```
breg1 = glm(admit~gre+gpa+rank+gre:gpa,data=admit,family = "binomial")
step(breg1,direction="both")$anova
```

```
## Start:  AIC=466.6
## admit ~ gre + gpa + rank + gre:gpa
##
##           Df Deviance AIC
## <none>            457 467
## - gre:gpa  1      459 467
## - rank     1      478 486
```

```
##   Step Df Deviance Resid. Df Resid. Dev   AIC
## 1      NA       NA       395      456.6 466.6
```

- Boston data

```
Boston = read.csv("../../data/Boston.csv")
reg1=lm(medv~.,data=Boston)
step(reg1,direction="both")$anova
```

```
## Start:  AIC=1590
## medv ~ crim + zn + indus + chas + nox + rm + age + dis + rad +
##     tax + ptratio + black + lstat
##
##           Df Sum of Sq   RSS  AIC
## - age      1         0 11079 1588
## - indus    1         3 11081 1588
## <none>                 11079 1590
## - chas     1       219 11298 1598
## - tax      1       242 11321 1599
## - crim     1       243 11322 1599
## - zn       1       257 11336 1599
## - black    1       271 11349 1600
## - rad      1       479 11558 1609
## - nox      1       487 11566 1609
## - ptratio  1      1194 12273 1639
## - dis      1      1232 12311 1641
## - rm       1      1871 12950 1667
## - lstat    1      2411 13490 1687
##
## Step:  AIC=1588
## medv ~ crim + zn + indus + chas + nox + rm + dis + rad + tax +
##     ptratio + black + lstat
##
##           Df Sum of Sq   RSS  AIC
## - indus    1         3 11081 1586
## <none>                 11079 1588
```

```
## + age       1         0 11079 1590
## - chas      1       220 11299 1596
## - tax       1       242 11321 1597
## - crim      1       243 11322 1597
## - zn        1       260 11339 1597
## - black     1       272 11351 1598
## - rad       1       481 11560 1607
## - nox       1       521 11600 1609
## - ptratio   1      1200 12279 1638
## - dis       1      1352 12431 1644
## - rm        1      1960 13038 1668
## - lstat     1      2719 13798 1697
##
## Step:  AIC=1586
## medv ~ crim + zn + chas + nox + rm + dis + rad + tax + ptratio +
##     black + lstat
##
##           Df Sum of Sq   RSS  AIC
## <none>                 11081 1586
## + indus    1         3 11079 1588
## + age      1         0 11081 1588
## - chas     1       227 11309 1594
## - crim     1       245 11327 1595
## - zn       1       258 11339 1595
## - black    1       271 11352 1596
## - tax      1       274 11355 1596
## - rad      1       501 11582 1606
## - nox      1       542 11623 1608
## - ptratio  1      1206 12288 1636
## - dis      1      1449 12530 1646
## - rm       1      1964 13045 1666
## - lstat    1      2723 13805 1695
```

```
##      Step Df Deviance Resid. Df Resid. Dev  AIC
## 1         NA       NA       492      11079 1590
## 2    - age  1  0.06183       493      11079 1588
## 3 - indus  1  2.51754       494      11081 1586
```

Subsets regression

- MASchools data

```
bestsub1 =  regsubsets(score4 ~exptot + scratio + special+ I(stratio^4) +
    I(salary^6) + exptot:scratio + special:I(stratio^4) + scratio:special +
    exptot:I(stratio^4) + exptot:stratio + special:stratio +
    scratio:I(salary^6),data=df1,nvmax = 12)
summary(bestsub1)
```

```
## Subset selection object
## Call: regsubsets.formula(score4 ~ exptot + scratio + special + I(stratio^4) +
##     I(salary^6) + exptot:scratio + special:I(stratio^4) + scratio:special +
##     exptot:I(stratio^4) + exptot:stratio + special:stratio +
```

```
##     scratio:I(salary^6), data = df1, nvmax = 12)
## 12 Variables  (and intercept)
##                      Forced in Forced out
## exptot                   FALSE      FALSE
## scratio                  FALSE      FALSE
## special                  FALSE      FALSE
## I(stratio^4)             FALSE      FALSE
## I(salary^6)              FALSE      FALSE
## exptot:scratio           FALSE      FALSE
## special:I(stratio^4)     FALSE      FALSE
## scratio:special          FALSE      FALSE
## exptot:I(stratio^4)      FALSE      FALSE
## exptot:stratio           FALSE      FALSE
## special:stratio          FALSE      FALSE
## scratio:I(salary^6)      FALSE      FALSE
## 1 subsets of each size up to 12
## Selection Algorithm: exhaustive
##           exptot scratio special I(stratio^4) I(salary^6) exptot:scratio
## 1  ( 1 )  " "    " "     " "     " "          "*"         " "
## 2  ( 1 )  " "    " "     " "     " "          "*"         " "
## 3  ( 1 )  "*"    " "     " "     " "          "*"         " "
## 4  ( 1 )  " "    " "     "*"     " "          "*"         " "
## 5  ( 1 )  " "    "*"     " "     " "          "*"         " "
## 6  ( 1 )  " "    "*"     " "     "*"          "*"         "*"
## 7  ( 1 )  " "    "*"     " "     "*"          "*"         "*"
## 8  ( 1 )  " "    "*"     " "     "*"          "*"         "*"
## 9  ( 1 )  " "    "*"     " "     "*"          " "         "*"
## 10  ( 1 ) "*"    "*"     "*"     "*"          " "         "*"
## 11  ( 1 ) "*"    "*"     "*"     "*"          " "         "*"
## 12  ( 1 ) "*"    "*"     "*"     "*"          "*"         "*"
##           special:I(stratio^4) scratio:special exptot:I(stratio^4)
## 1  ( 1 )  " "                  " "             " "
## 2  ( 1 )  " "                  " "             "*"
## 3  ( 1 )  " "                  " "             "*"
## 4  ( 1 )  "*"                  " "             "*"
## 5  ( 1 )  "*"                  "*"             "*"
## 6  ( 1 )  "*"                  "*"             " "
## 7  ( 1 )  "*"                  "*"             "*"
## 8  ( 1 )  "*"                  "*"             "*"
## 9  ( 1 )  "*"                  "*"             "*"
## 10  ( 1 ) "*"                  " "             "*"
## 11  ( 1 ) "*"                  "*"             "*"
## 12  ( 1 ) "*"                  "*"             "*"
##           exptot:stratio special:stratio scratio:I(salary^6)
## 1  ( 1 )  " "            " "             " "
## 2  ( 1 )  " "            " "             " "
## 3  ( 1 )  " "            " "             " "
## 4  ( 1 )  " "            " "             " "
## 5  ( 1 )  " "            " "             " "
## 6  ( 1 )  " "            " "             " "
## 7  ( 1 )  " "            " "             " "
## 8  ( 1 )  "*"            " "             " "
## 9  ( 1 )  "*"            "*"             "*"
## 10  ( 1 ) "*"            "*"             "*"
```

```
## 11  ( 1 ) "*"            "*"             "*"
## 12  ( 1 ) "*"            "*"             "*"
```

```
names(summary(bestsub1))
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
round(cbind(
    Cp     = summary(bestsub1)$cp,
    r2     = summary(bestsub1)$rsq,
    Adj_r2 = summary(bestsub1)$adjr2,
    BIC    = summary(bestsub1)$bic
),3)
```

```
##          Cp    r2 Adj_r2    BIC
##  [1,] 87.54 0.158  0.154 -21.57
##  [2,] 46.86 0.291  0.284 -48.41
##  [3,] 41.02 0.316  0.305 -49.73
##  [4,] 29.60 0.358  0.344 -56.26
##  [5,] 23.55 0.383  0.366 -58.46
##  [6,] 20.87 0.398  0.377 -57.69
##  [7,] 18.35 0.412  0.389 -56.88
##  [8,] 16.08 0.425  0.399 -55.92
##  [9,] 15.81 0.432  0.403 -53.00
## [10,] 12.40 0.449  0.418 -53.40
## [11,] 11.29 0.459  0.425 -51.48
## [12,] 13.00 0.460  0.422 -46.56
```

- Boston data

```
bestsub1 =  regsubsets(medv~.,data=Boston,nvmax = 14)
summary(bestsub1)
```

```
## Subset selection object
## Call: regsubsets.formula(medv ~ ., data = Boston, nvmax = 14)
## 13 Variables  (and intercept)
##         Forced in Forced out
## crim        FALSE      FALSE
## zn          FALSE      FALSE
## indus       FALSE      FALSE
## chas        FALSE      FALSE
## nox         FALSE      FALSE
## rm          FALSE      FALSE
## age         FALSE      FALSE
## dis         FALSE      FALSE
## rad         FALSE      FALSE
## tax         FALSE      FALSE
## ptratio     FALSE      FALSE
## black       FALSE      FALSE
## lstat       FALSE      FALSE
## 1 subsets of each size up to 13
```

```
## Selection Algorithm: exhaustive
##           crim zn  indus chas nox rm  age dis rad tax ptratio black lstat
## 1  ( 1 )  " "  " " " "   " "  " " " " " " " " " " " " " "     " "   "*"
## 2  ( 1 )  " "  " " " "   " "  " " "*" " " " " " " " " " "     " "   "*"
## 3  ( 1 )  " "  " " " "   " "  " " "*" " " " " " " " " "*"     " "   "*"
## 4  ( 1 )  " "  " " " "   " "  " " "*" " " "*" " " " " "*"     " "   "*"
## 5  ( 1 )  " "  " " " "   " "  "*" "*" " " "*" " " " " "*"     " "   "*"
## 6  ( 1 )  " "  " " " "   "*"  "*" "*" " " "*" " " " " "*"     " "   "*"
## 7  ( 1 )  " "  " " " "   "*"  "*" "*" " " "*" " " " " "*"     "*"   "*"
## 8  ( 1 )  " "  "*" " "   "*"  "*" "*" " " "*" " " " " "*"     "*"   "*"
## 9  ( 1 )  "*"  " " " "   "*"  "*" "*" " " "*" "*" " " "*"     "*"   "*"
## 10  ( 1 ) "*"  "*" " "   " "  "*" "*" " " "*" "*" "*" "*"     "*"   "*"
## 11  ( 1 ) "*"  "*" " "   "*"  "*" "*" " " "*" "*" "*" "*"     "*"   "*"
## 12  ( 1 ) "*"  "*" "*"   "*"  "*" "*" " " "*" "*" "*" "*"     "*"   "*"
## 13  ( 1 ) "*"  "*" "*"   "*"  "*" "*" "*" "*" "*" "*" "*"     "*"   "*"
```

```r
round(cbind(
    Cp     = summary(bestsub1)$cp,
    r2     = summary(bestsub1)$rsq,
    Adj_r2 = summary(bestsub1)$adjr2,
    BIC    = summary(bestsub1)$bic
),3)
```

```
##           Cp    r2 Adj_r2    BIC
##  [1,] 362.75 0.544  0.543 -385.1
##  [2,] 185.65 0.639  0.637 -496.3
##  [3,] 111.65 0.679  0.677 -549.5
##  [4,]  91.48 0.690  0.688 -562.0
##  [5,]  59.75 0.708  0.705 -585.7
##  [6,]  47.17 0.716  0.712 -593.0
##  [7,]  37.06 0.722  0.718 -598.2
##  [8,]  30.62 0.727  0.722 -600.2
##  [9,]  25.87 0.730  0.725 -600.6
## [10,]  18.20 0.735  0.730 -604.0
## [11,]  10.12 0.741  0.735 -608.0
## [12,]  12.00 0.741  0.734 -601.9
## [13,]  14.00 0.741  0.734 -595.7
```

Use Case: Profit Forecasting, Steps for a Safety-first Linear Regression

We have examined profit forecasting using R&D and marketing spend in a parametric context, where we had a good idea what the population distribution of profits was, and a non-parametric approach when we were not sure. We now tackle the same challenge but using a 4-point safety first process:

1. Check the data
2. Check for collinearities
3. Check model fit
4. Check residuals

Check1: Check the data

Check1 is simply data exploration, examining distributions and relationships as we have seen in previous chapters. We also need to check for imbalances in the dataset, particularly in classification problems, where we might be forecasting credit card loan defaults from a dataset where only 5% of the rows represent defaults (we will address this later in the book). Can we take a view on what the population distribution is? If so, our model will always be more accurate if we use tests that assume distributions that most resemble the true population distribution of our data.

```
df_train = read.csv("../../data/50_Startups.csv")
```

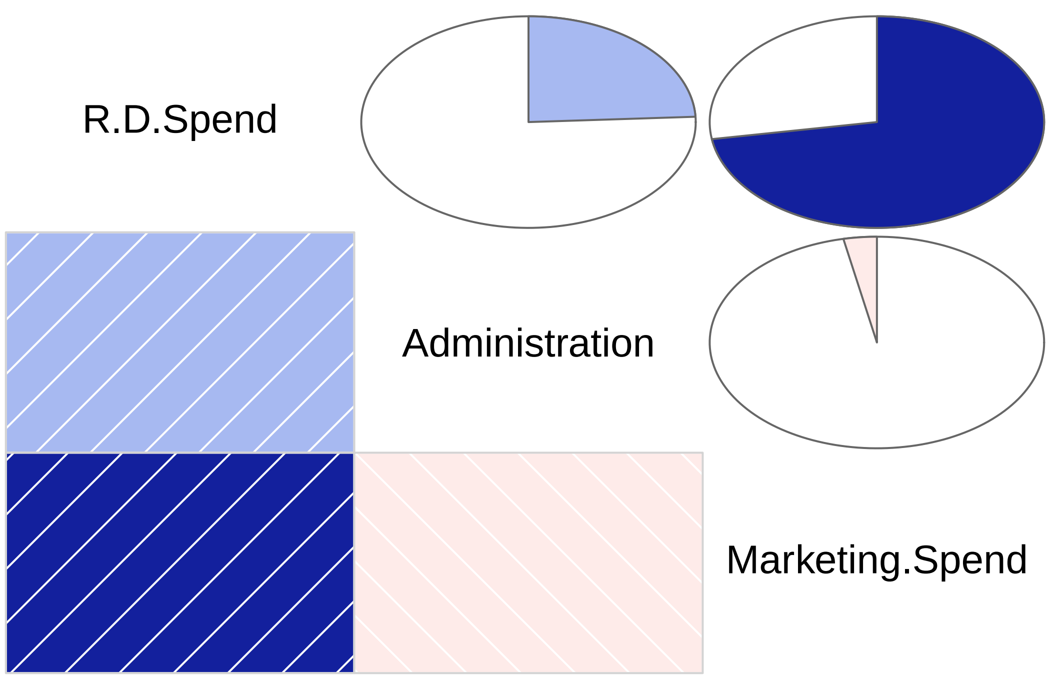
Check2: Check for collinearities

- corrgram package provides a nice visual

```
library(corrgram)
cor(df_train[1:3])
```

```
##                 R.D.Spend Administration Marketing.Spend
## R.D.Spend          1.0000        0.24196         0.72425
## Administration     0.2420        1.00000        -0.03215
## Marketing.Spend    0.7242       -0.03215         1.00000
```

```
corrgram(df_train[1:3],upper.panel = panel.pie)
```



We will use a rule of thumb that no 2 input variables should have a correlation coefficient of >0.5. You can see that R&D Spend and Marketing Spend have a correlation coefficient of 0.72 and so breach our rule of thumb. We will use *differencing* to see if correlation is reduced.

```
df_train_dif = df_train
df_train_dif$Marketing.Spend = df_train_dif$Marketing.Spend - df_train_dif$R.D.Spend
cor(df_train_dif[1:3])
```

```
##                 R.D.Spend Administration Marketing.Spend
## R.D.Spend          1.0000         0.2420          0.4515
## Administration     0.2420         1.0000         -0.1591
## Marketing.Spend    0.4515        -0.1591          1.0000
```

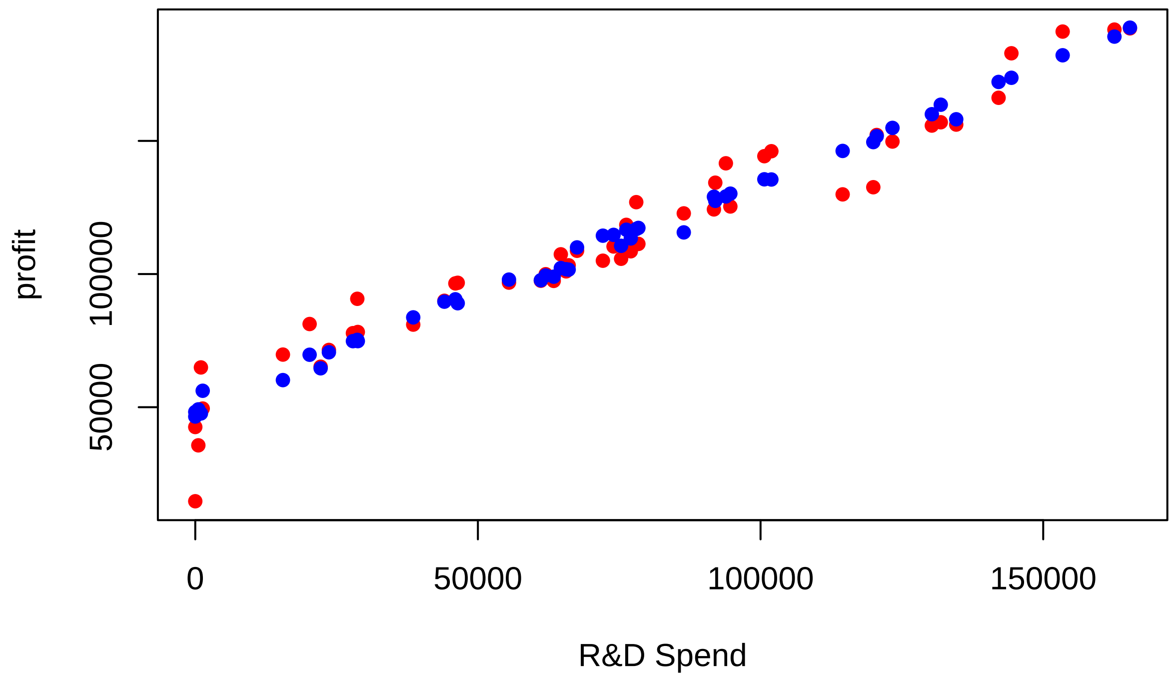
Check3: Check for Model Fit

We can now run the regression and assess the goodness of the model fit:

```
reg1 = lm(Profit ~ .-State, data=df_train_dif)
summary(reg1)
```

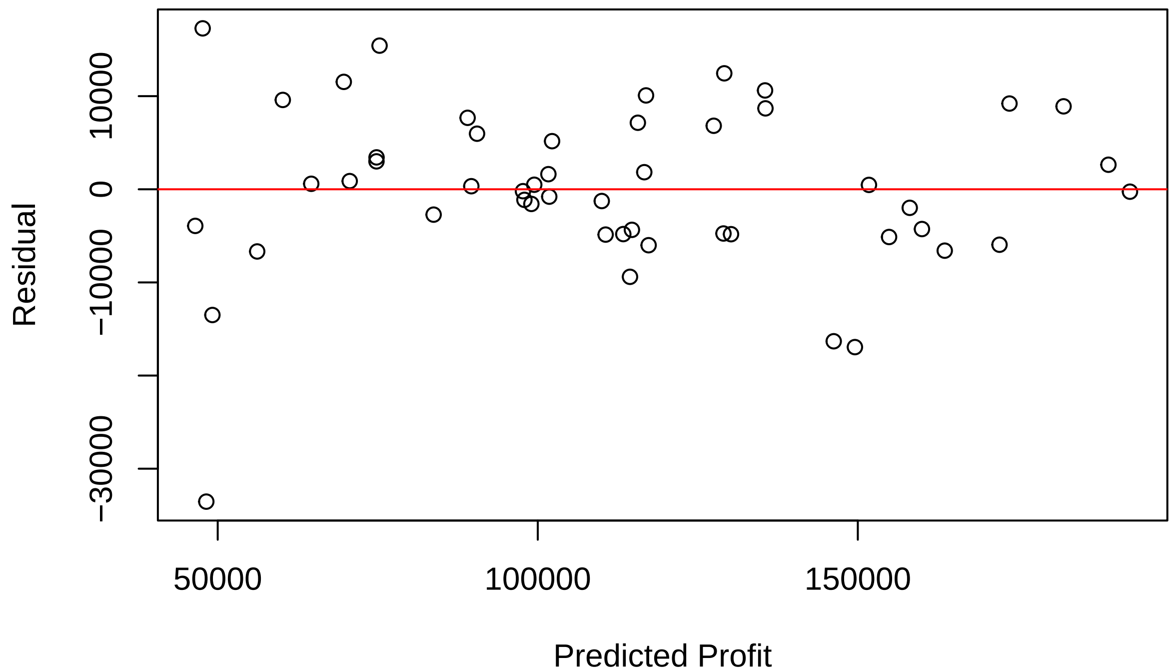
```
##
## Call:
## lm(formula = Profit ~ . - State, data = df_train_dif)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -33534  -4795     63   6606  17275
##
## Coefficients:
##                   Estimate Std. Error t value             Pr(>|t|)
## (Intercept)     50122.1930  6572.3526    7.63         0.0000000011 ***
## R.D.Spend           0.8329     0.0345   24.17 < 0.0000000000000002 ***
## Administration     -0.0268     0.0510   -0.53                  0.6
## Marketing.Spend     0.0272     0.0165    1.66                  0.1
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9230 on 46 degrees of freedom
## Multiple R-squared:  0.951,  Adjusted R-squared:  0.948
## F-statistic:  296 on 3 and 46 DF,  p-value: <0.0000000000000002
```

```
plot(df_train_dif$R.D.Spend,df_train_dif$Profit,col="red",pch=16,xlab="R&D Spend",ylab="profit")
points(df_train_dif$R.D.Spend,reg1$fitted.values,col="blue",pch=16)
```

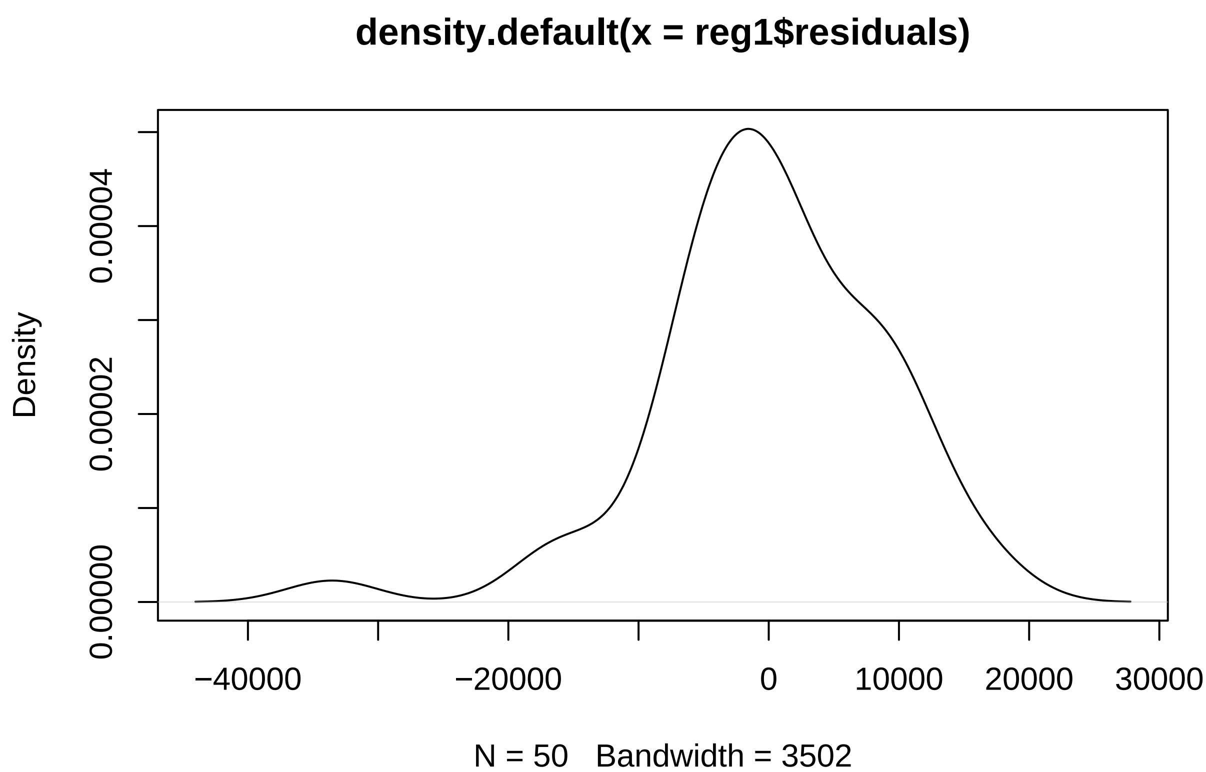


Check4: Check Residuals

```
plot(reg1$fitted.values,reg1$residuals,
     xlab="Predicted Profit",
     ylab = "Residual")
abline(h=0,col="red")
```




```
plot(density(reg1$residuals))
```




```
shapiro.test(reg1$residuals)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  reg1$residuals
## W = 0.94, p-value = 0.01
```

The plot and the normality test indicate that the normality assumption is a bit weak.

Automating Model Construction

In the R code below, we will run the subsets regression to select the right input variables:

```
library(leaps)
bestsub1 =  regsubsets(Profit ~ . - State, data = df_train_dif,nvmax = 12)
summary(bestsub1)
```

```
## Subset selection object
## Call: regsubsets.formula(Profit ~ . - State, data = df_train_dif, nvmax = 12)
## 3 Variables  (and intercept)
##                  Forced in Forced out
## R.D.Spend            FALSE      FALSE
```

```
## Administration      FALSE      FALSE
## Marketing.Spend     FALSE      FALSE
## 1 subsets of each size up to 3
## Selection Algorithm: exhaustive
##          R.D.Spend Administration Marketing.Spend
## 1  ( 1 ) "*"       " "            " "
## 2  ( 1 ) "*"       " "            "*"
## 3  ( 1 ) "*"       "*"            "*"
```

```
names(summary(bestsub1))
```

```
## [1] "which"  "rsq"    "rss"    "adjr2"  "cp"     "bic"    "outmat" "obj"
```

```
round(cbind(
    Cp     = summary(bestsub1)$cp,
    r2     = summary(bestsub1)$rsq,
    Adj_r2 = summary(bestsub1)$adjr2,
    BIC    = summary(bestsub1)$bic
),3)
```

```
##         Cp    r2 Adj_r2    BIC
## [1,] 3.932 0.947  0.945 -138.6
## [2,] 2.276 0.950  0.948 -138.5
## [3,] 4.000 0.951  0.948 -134.9
```

The results recommend using only “R&D Spend” and “Marketing Spend” as the input variables. Rerunning the regression model yields the following outcome which is an improvement over the initial regression model (for example, based on Adjusted R^2 values):

```
reg2 = lm(Profit ~ R.D.Spend+Marketing.Spend, data=df_train_dif)
summary(reg2)
```

```
##
## Call:
## lm(formula = Profit ~ R.D.Spend + Marketing.Spend, data = df_train_dif)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -33645  -4632   -414   6484  17097
##
## Coefficients:
##                   Estimate Std. Error t value            Pr(>|t|)
## (Intercept)     46975.8642  2689.9329   17.46 <0.0000000000000002 ***
## R.D.Spend           0.8265     0.0320   25.87 <0.0000000000000002 ***
## Marketing.Spend     0.0299     0.0155    1.93                0.06 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 9160 on 47 degrees of freedom
## Multiple R-squared:  0.95,   Adjusted R-squared:  0.948
## F-statistic:  451 on 2 and 47 DF,  p-value: <0.0000000000000002
```