A redefined Variance Inflation Factor: overcoming the limitations of the Variance Inflation Factor

Román Salmerón Gómez and Catalina García García

2024-02-12

RVIF function:

The function used in this paper to calculate the RVIF is the following:

```
RVIF = function(X, l_u = F, intercept = T){
  n = dim(X)[1]
  p = dim(X)[2]
  if (l_u == T){X = lu(X)} # lu is a multiColl R package function
  RVIFs = integer()
  a = integer()
  if (intercept == T) row_names = c("Intercept") else row_names = paste("Variable", 1)
  for (i in 1:p){
    if (det(crossprod(X[,-i],X[,-i])) != 0)
      a_i = crossprod(X[,i],X[,-i])%*%solve(crossprod(X[,-i],X[,-i]))%*%crossprod(X[,-i],X[,i])
    else a_i = NaN
    d_i = crossprod(X[,i])
    RVIFs[i] = 1/(d_i-a_i)
    a[i] = round(a_i*100/d_i, digits=4)
    if (i>1){row_names =c(row_names, paste("Variable", i))}
  }
  salida = data.frame(RVIFs, a)
  rownames(salida) = row_names
  colnames(salida) = c("RVIF", "%")
  return(salida)
}
```

Section 2.3 example

Taking into account the previous function to calculate the RVIF and the following matrix **X**:

```
X = matrix(c(1,1,1,1,1,1,1,3,2,1,3,2,2,6,4,2,6,4,-0.5,-0.5,1,-0.5,-0.5,1),6,4)
X
```

```
##      [,1] [,2] [,3] [,4]
## [1,]    1    1    2 -0.5
## [2,]    1    3    6 -0.5
## [3,]    1    2    4  1.0
## [4,]    1    1    2 -0.5
## [5,]    1    3    6 -0.5
## [6,]    1    2    4  1.0
```

The following results are obtained:

```
RVIF(X)
```

```
##             RVIF   %
## Intercept    NaN NaN
## Variable 2   Inf 100
## Variable 3   Inf 100
## Variable 4   NaN NaN
```

For illustrative purpose we have eliminated the third column of matrix **X**:

```
X = X[,-3]
X
```

```
##      [,1] [,2] [,3]
## [1,]    1    1 -0.5
## [2,]    1    3 -0.5
## [3,]    1    2  1.0
## [4,]    1    1 -0.5
## [5,]    1    3 -0.5
## [6,]    1    2  1.0
```

```
RVIF(X)
```

```
##                 RVIF       %
## Intercept  1.1666667 85.7143
## Variable 2 0.2500000 85.7143
## Variable 3 0.3333333  0.0000
```

Section 4 example

Calculating the RVIF after removing the third column of **X** and considering that the data have been transformed to unit length:

```
RVIF(X, l_u = T)
```

```
##             RVIF       %
## Intercept      7 85.7143
## Variable 2     7 85.7143
## Variable 3     1  0.0000
```

Section 4.1 example 1

Data of financial model in which euribor (E, %) is analyzed from the harmonized index of consumer prices (HICP, %), the balance of payments to net current account (BC, million of euros) and the Goverment deficit to net nonfinancial accounts (GD, millions of euros):

```
E = c(3.63,3.90,3.45,3.01,2.54,2.23,2.20,2.36,2.14,2.29,2.35,2.32,2.32,2.19,2.20,2.63,
      2.95,3.31,3.62,3.60,4.09,4.38,4.65,4.68,4.48,5.05,5.37,4.34,2.22,1.67,1.34,1.24,
      1.22,1.25,1.40,1.52,1.74,2.13,2.11,2.06,1.67,1.28,0.90,0.60,0.57,0.51,0.54)
HIPC = c(92.92,  93.85,  93.93,  94.41,  95.08,  95.73,  95.90,  96.40,  96.77,
         97.97,  98.06,  98.67, 98.76,  99.96, 100.30, 100.97, 101.07, 102.44,
         102.52, 102.79, 102.97, 104.38, 104.45, 105.77, 106.43, 108.18, 108.49,
         108.21, 107.46, 108.37, 108.08, 108.67, 108.67, 110.12, 109.95, 110.87,
```

```r
         111.36, 113.15, 112.91, 114.12, 114.35, 115.93, 115.78, 116.75, 116.47,
         117.55, 117.34)
BC = c(17211,   2724,  17232,   9577,   4117,  -2134,   6117,  10949,  18360,  13646,
     8424,  14319, 3885,   4493,   -320,  -2736,  -6909,  -4848,  -4255,   1347,
     8781,   8723,   3662, -17548, -37041, -27624, -37723, -43584, -16070, -5029,
     7294,     85,  -4399,  -2431,   2137,  -4345, -12643,  -2272,  -3592,   8071,
     12202,  35619,  42161,  43880,  52483,  56376,  48981)
GD = c(-51384.0,  -49567.1,  -52128.4,  -53593.3,  -65480.0,  -50343.8,  -75646.4,
     -59120.8,  -69246.3,  -60313.8,  -56782.9,  -55313.1,  -67034.4,  -61942.8,
     -46258.4,  -43761.4,  -37562.6,  -35609.6,  -27064.0,  -32497.2,  -18389.0,
     -9923.5,   -9727.0,  -23729.9,  -28909.3,  -46527.0,  -49654.0,  -81729.7,
     -121227.5, -142580.9, -164699.2, -152269.2, -162477.4, -128366.4, -169848.0,
     -129290.2, -104646.7, -103143.8, -102621.8, -104240.4, -82309.3,  -91620.9,
     -85054.4,  -99998.2,  -81287.1,  -77738.8,  -73003.3)
cte1 = rep(1, length(E))
data1 = cbind(cte1, HIPC, BC, GD)
```

The variation inflation factors, the condition number and RVIF are the following:

```r
VIF(data1)
```

```
##     HIPC       BC       GD 
## 1.349666 1.058593 1.283815
```

```r
CN(data1)
```

```
## [1] 39.35375
```

```r
RVIF(data1, l_u = T)
```

```
##                   RVIF       %
## Intercept  250.294157 99.6005
## Variable 2 280.136873 99.6430
## Variable 3   1.114787 10.2967
## Variable 4   5.525440 81.9019
```

Centering the variable **HIPC** the problem is mitigated:

```r
data1[,2] = data1[,2] - mean(data1[,2])
CN(data1)
```

```
## [1] 4.648044
```

```r
RVIF(data1, l_u = T)
```

```
##                 RVIF       %
## Intercept  5.341031 81.2770
## Variable 2 1.349666 25.9076
## Variable 3 1.114787 10.2967
## Variable 4 5.525440 81.9019
```

However, if the VIF is calculated for the modified data, then it is observed that it is invariant to the transformation performed:

```r
VIF(data1)
```

```
##     HIPC       BC       GD 
## 1.349666 1.058593 1.283815
```

Section 4.2 example 2

Data to analyze the Cobb-Douglas production function in Mexico:

```
P = c(37641114, 42620804, 37989413, 40464915, 41002031, 45135601, 39748030, 40708136,
      37171183, 36728430, 41901143, 41503828, 44329866, 43924072, 46334199, 46697539,
      48888741, 48141320, 50883107, 50084943, 51429861, 55248539, 55048964, 57174488,
      58391982, 55159985, 54484149, 61912943)
K = c(61671750, 66444000, 64876500, 66121000, 67176000, 69239250, 68410000, 69174250,
      66720000, 65730250, 70208500, 71918250, 70438750, 72629000, 73222250, 74101750,
      76706500, 76821750, 77334000, 80031500, 80529500, 83348750, 83581250, 86221750,
      89120500, 87714250, 91848500, 93730000)
W = c(5700000, 5989000, 5637000, 5874000, 5931000, 6096000, 5890000, 5930000, 5911000,
      6058000, 5779000, 6214512, 6157699, 6244883, 6318703, 6193512, 6309359, 6116378,
      6345504, 6392005, 6286195, 6356448, 6281631, 6394984, 6547140, 6279000, 5995000,
      5843000)
cte2 = rep(1, length(P))
data2 = cbind(cte2, log(K), log(W))
```

The RVIF, VIF, CN and CV for these variables are:

```
RVIF(data2, l_u = T)
```

```
##                   RVIF       %
## Intercept  178888.82 99.9994
## Variable 2  38071.36 99.9974
## Variable 3 255219.62 99.9996
```

```
VIF(data2)
```

```
##
## 1.507716 1.507716
```

```
CNs(data2)
```

```
## $`Condition Number without intercept`
## [1] 379.6053
##
## $`Condition Number with intercept`
## [1] 1131.663
##
## $`Increase (in percentage)`
## [1] 66.45597
```

```
CVs(data2)
```

```
## [1] 0.006293163 0.002430547
```

If the RVIF is calculated ignoring the intercept (first column), then the aforementioned relationship between both variables is also detected:

```
RVIF(data2[,-1], intercept = F, l_u = T)
```

```
##                 RVIF       %
## Variable 1 36025.54 99.9972
## Variable 2 36025.54 99.9972
```

If capital and work are centered:

```r
data2[,2] = data2[,2] - mean(data2[,2])
data2[,3] = data2[,3] - mean(data2[,3])
RVIF(data2, l_u = T)
```

```
##                 RVIF       %
## Intercept  1.000000  0.0000
## Variable 2 1.507716 33.6745
## Variable 3 1.507716 33.6745
```

```r
VIF(data2)
```

```
##
## 1.507716 1.507716
```

```r
CNs(data2)
```

```
## $`Condition Number without intercept`
## [1] 1.940433
##
## $`Condition Number with intercept`
## [1] 1.940433
##
## $`Increase (in percentage)`
## [1] 3.432913e-14
```

```r
CVs(data2)
```

```
## [1] 1.498622e+14 4.988415e+13
```

Section 4.3 example 3

Data of Spanish companies:

```r
NE<- c(2637,15954,162503,162450,28389,132120,63387,26422,19661,8448,9819,7964,169766,
       15122,13881)
FA<- c(44153,9389509,17374000,9723088,95980120,103667000,17588000,48848000,38901000,
       29109985,25529000,14502621,12639483,26787667,6681800)
OI<- c(38903,4293386,23703000,23310532,29827663,52574000,22567000,35679000,23449000,
       14475554,18979000,3717342,32436917,4916125,4472900)
S<- c(38867,4231043,23649000,23310532,29215382,52036000,22567000,34689000,23184000,
      14207980,18979000,3709581,31975212,4758244,4472900)
data3 = cbind(rep(1, length(NE)), FA, OI, S)
```

The RVIF, matrix of simple correlations, its determinant and VIF for these variables are:

```r
RVIF(data3, l_u = T)
```

```
##                     RVIF       %
## Intercept       2.984146 66.4896
## Variable 2      5.011397 80.0455
## Variable 3 15186.744870 99.9934
## Variable 4 15052.679178 99.9934
```

```r
RdetR(data3)
```

```
## $`Correlation matrix`
##           FA        OI         S
```

```
## FA 1.0000000 0.7264656 0.7225473
## OI 0.7264656 1.0000000 0.9998871
## S  0.7225473 0.9998871 1.0000000
##
## $`Correlation matrix's determinant`
## [1] 9.190317e-05
```

```
VIF(data3)
```

```
##         FA         OI          S
##    2.45664 5200.31530 5138.53548
```

Section 4.4 example 4

Two simulations will be performed: one in which there is not an approximate troubling degree of multicollinearity and another one in which there is.

Thus, 50 observations are simulated (using the command *set.seed(2022)*) for a variable **V** distribuited by following a normal with mean equal to 10 and variance equal to 100 and other variable **Z** distribuited by a normal with mean equal to 10 and variance equal to 0.1:

```
set.seed(2022)
obs = 50
cte4 = rep(1, obs)
V = rnorm(obs, 10, 10)
Z = rnorm(obs, 10, 0.1)
data4.1 = cbind(cte4, V)
data4.2 = cbind(cte4, Z)
```

In the first case:

```
RVIF(data4.1, l_u = T)
```

```
##                RVIF       %
## Intercept  2.015249 50.3783
## Variable 2 2.015249 50.3783
```

while in the second case:

```
RVIF(data4.2, l_u = T)
```

```
##                RVIF       %
## Intercept  8620.076 99.9884
## Variable 2 8620.076 99.9884
```

Section 4.4.1 example

```
y = rnorm(obs, 100, 5)
reg1 = lm(y~V)
# vif(reg1) # if '#' is removed, compilation fails
reg2 = lm(y~Z)
# vif(reg2) # if '#' is removed, compilation fails
```

```
reg3 = lm(y~cte4+V+0)
vif(reg3)
```

```
##      cte4        V
## 2.015249 2.015249
```

```
reg4 = lm(y~cte4+Z+0)
vif(reg4)
```

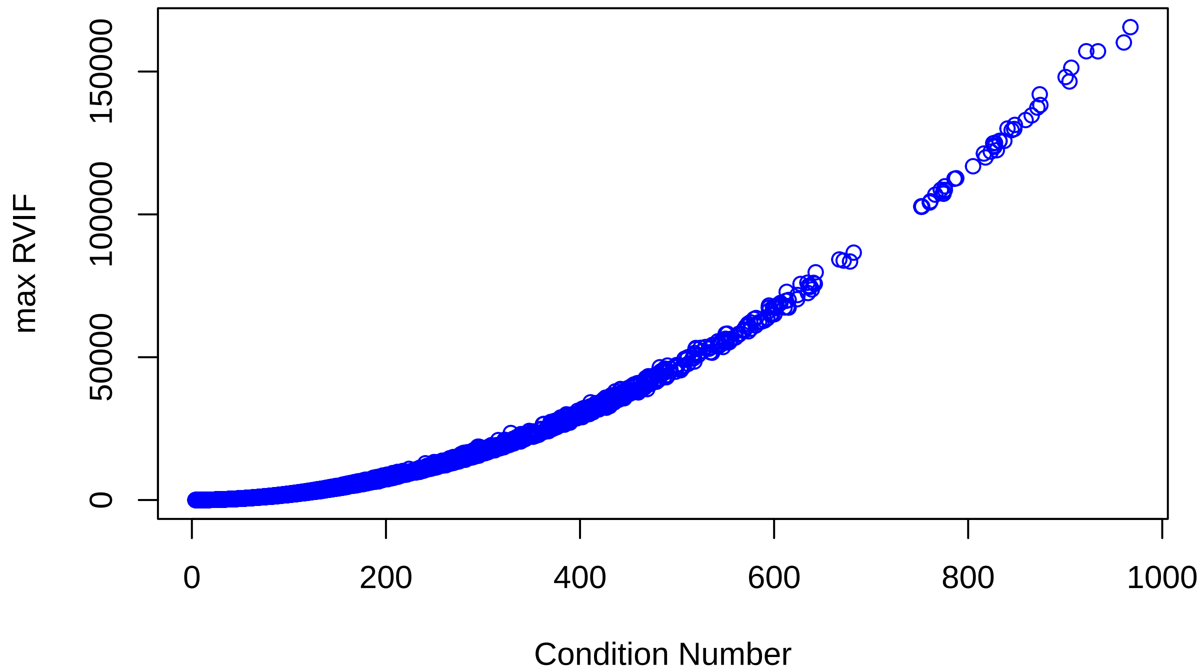
```
##      cte4        Z
## 8620.076 8620.076
```

Monte Carlo simulation

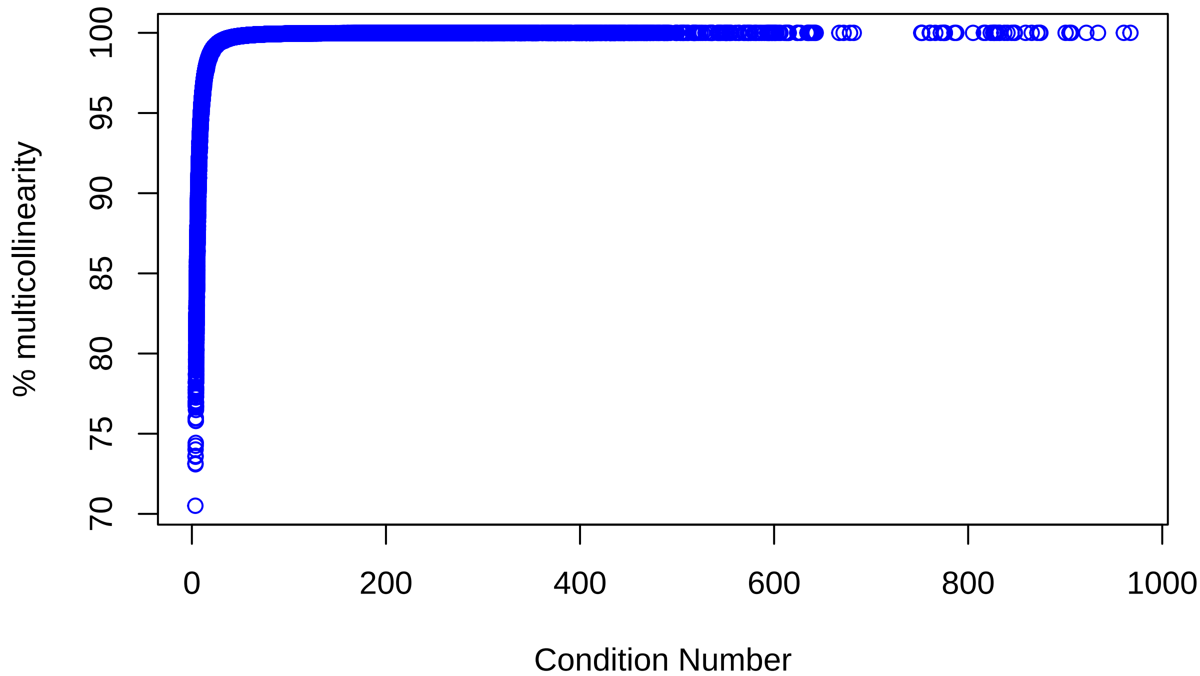
Monte Carlo simulation is performed to generate data for a multiple linear regression with 3 independent variables:

```
set.seed(1234)
observaciones = seq(15,200,5)
n1 = length(observaciones)
sigmas = c(0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.02,
           0.03, 0.04, 0.05, 0.06, 0.07, 0.08, 0.09, 0.1, 0.2, 0.3, 0.4, 0.5)
n2 = length(sigmas)
gammas = c(0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9,0.95,0.96,0.97,0.98,0.99)
n3 = length(gammas)

n = n1*n2*n3

res = array(,c(n,6))
colnames(res) = c("Observaciones","Sigma","Réplica","RFIV","%", "NC")

k=1
for (sigma in sigmas){
  for (obs in observaciones) {
    for (gamma in gammas) {
      Wsim1 = rnorm(obs,1,sqrt(sigma))
      Wsim2 = rnorm(obs,1,sqrt(sigma))
      cte = matrix(1, obs, 1)
      Xsim1 = sqrt(1-gamma)*Wsim1 + gamma*Wsim2
      Xsim2 = sqrt(1-gamma)*Wsim2 + gamma*Wsim2
      Xsim = cbind(cte, Xsim1, Xsim2)

        res[k,1] = obs
        res[k,2] = sigma
        res[k,3] = gamma
        rvif = RVIF(Xsim, l_u = T)
        res[k,4] = max(rvif[,1])
        res[k,5] = max(rvif[,2])
        res[k,6] = CN(Xsim)

      k = k + 1
    }
  }
}

maxRVIF = res[,4]
porc = res[,5]
```

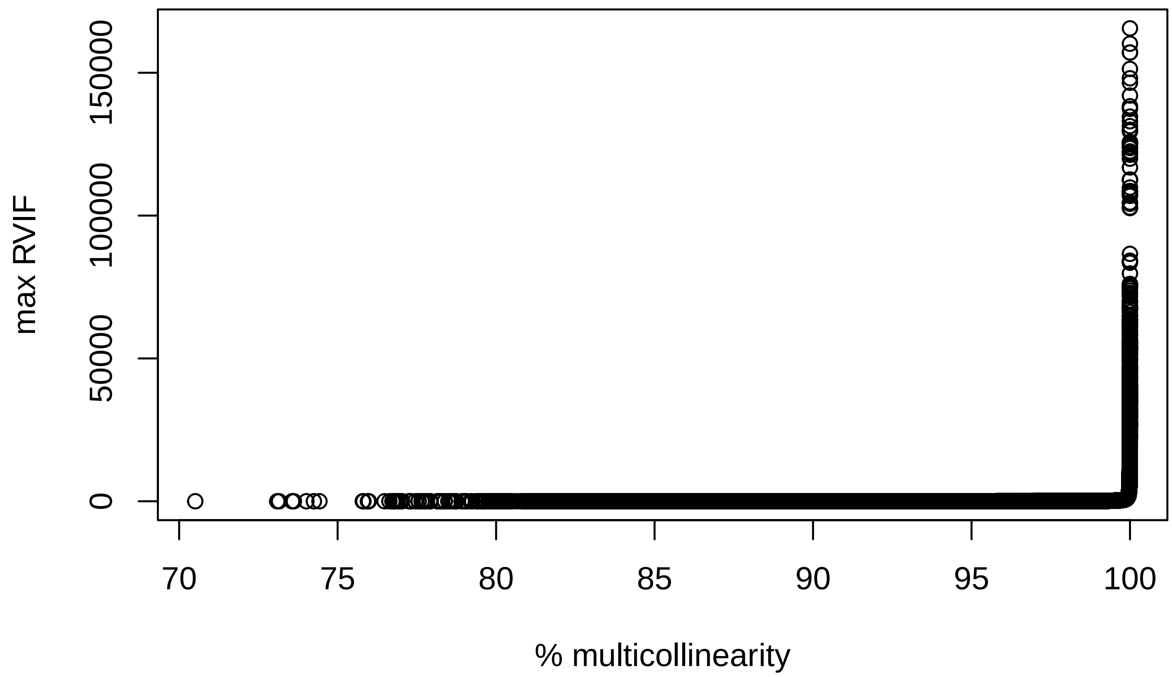
```
CN = res[,6]

plot(CN, maxRVIF, ylab = "max RVIF", xlab = "Condition Number", col="blue")
```




```
plot(CN, porc, ylab = "% multicollinearity", xlab = "Condition Number", col="blue")
```




```
plot(porc, maxRVIF, ylab = "max RVIF", xlab = "% multicollinearity")
```




```r
maxRVIF_minmax = c(min(maxRVIF), max(maxRVIF))
porc_minmax = c(min(porc), max(porc))
CN_minmax = c(min(CN), max(CN))
```

According to the estimates of each of the above models:

```
##
## Call:
## lm(formula = porc ~ sqrt(CN) + 0)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -199.277    8.931   36.963   58.125   66.250
##
## Coefficients:
##          Estimate Std. Error t value Pr(>|t|)
## sqrt(CN)  9.62282    0.04844   198.6   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 49.11 on 13109 degrees of freedom
## Multiple R-squared:  0.7506, Adjusted R-squared:  0.7506
## F-statistic: 3.946e+04 on 1 and 13109 DF,  p-value: < 2.2e-16
```

```
##
## Call:
## lm(formula = porc ~ log(CN) + 0)
```

```
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -67.035  -9.249   8.599  29.016  45.341
##
## Coefficients:
##         Estimate Std. Error t value Pr(>|t|)
## log(CN) 24.29787    0.05836   416.4   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 26.07 on 13109 degrees of freedom
## Multiple R-squared:  0.9297, Adjusted R-squared:  0.9297
## F-statistic: 1.734e+05 on 1 and 13109 DF,  p-value: < 2.2e-16
```

Section 6.1 example 5

Henri Theil's textile consumption data are the following:

```r
data(theil)
consume = theil[,2]
income = theil[,3]
relprice = theil[,4]
twentys = theil[,5]
cte5 = rep(1, length(consume))
data5 = cbind(cte5, income, relprice)
```

Excluding the binary variable, the VIFs, CN, CVs and RVIF (with and without intercept) are:

```r
VIF(data5)
```

```
##   income relprice
## 1.033043 1.033043
```

```r
CNs(data5)
```

```
## $`Condition Number without intercept`
## [1] 9.576912
##
## $`Condition Number with intercept`
## [1] 48.95347
##
## $`Increase (in percentage)`
## [1] 80.43671
```

```r
CVs(data5)
```

```
## [1] 0.04993766 0.21441845
```

```r
RVIF(data5, l_u = T)
```

```
##                 RVIF       %
## Intercept  403.20963 99.7520
## Variable 2 415.28266 99.7592
## Variable 3  23.50258 95.7451
```

```r
RVIF(data5[,-1], l_u = T, intercept = F)
```

```
##                 RVIF       %
## Variable 1 23.43204 95.7323
## Variable 2 23.43204 95.7323
```

If the dummy variable **twentys** is considered, the RVIF and the percentage of multicollinearity are:

```r
data5 = cbind(cte5, income, relprice, twentys)
RVIF(data5, l_u = T, intercept = F)
```

```
##                   RVIF       %
## Variable 1 427.445968 99.7661
## Variable 2 427.228985 99.7659
## Variable 3 136.668316 99.2683
## Variable 4   9.972766 89.9727
```

Finally, if the VIF is calculated by using the package *car*, it is obtained:

```r
reg.theil = lm(consume~income+relprice+twentys)
vif(reg.theil)
```

```
##   income relprice  twentys
## 1.062760 6.007181 5.866333
```