```r
## Coffee example (Coffee Quality Institute, 2018) continued
coffee <- read.csv("csv/coffee_arabica.csv")

# cor(coffee) doesn't work as there's a categorical variable.
cor(coffee[, -1]) # e.g., remove first column
```

```
##                  Aroma     Flavor Aftertaste        Body     Acidity    Balance
## Aroma       1.00000000  0.7339782  0.6892744  0.56699932  0.60115765  0.6156508
## Flavor      0.73397820  1.0000000  0.8582783  0.67694834  0.73845546  0.7324530
## Aftertaste  0.68927440  0.8582783  1.0000000  0.67407704  0.69408861  0.7657979
## Body        0.56699932  0.6769483  0.6740770  1.00000000  0.60795391  0.6924568
## Acidity     0.60115765  0.7384555  0.6940886  0.60795391  1.00000000  0.6417994
## Balance     0.61565084  0.7324530  0.7657979  0.69245676  0.64179938  1.0000000
## Sweetness   0.06955938  0.1345364  0.1185760  0.03977892  0.06906093  0.1016718
## Uniformity  0.14785498  0.2132347  0.2143116  0.07195778  0.14876428  0.2180726
## Moisture   -0.11567549 -0.1327342 -0.1745366 -0.21009097 -0.10391684 -0.2161964
##              Sweetness Uniformity    Moisture
## Aroma      0.06955938 0.14785498 -0.11567549
## Flavor     0.13453644 0.21323472 -0.13273418
## Aftertaste 0.11857600 0.21431157 -0.17453658
## Body       0.03977892 0.07195778 -0.21009097
## Acidity    0.06906093 0.14876428 -0.10391684
## Balance    0.10167183 0.21807265 -0.21619640
## Sweetness  1.00000000 0.34756414  0.08049300
## Uniformity 0.34756414 1.00000000  0.02105693
## Moisture   0.08049300 0.02105693  1.00000000
```

Plot the pairs (disabled due to loading time on PDF).

```r
pairs(
  ~ Flavor + Aroma + Aftertaste + Body +
    Acidity + Balance + Sweetness + Uniformity + Moisture,
  data = coffee
)
```

```r
# Code our own indicators, so that we can more easily interpret VIFs.
# 1 = wet, 0 otherwise
coffee$wet <-
  ifelse(coffee$Processing.Method == 'Washed / Wet', 1, 0)
# 1 = semi/dry, 0 otherwise
coffee$semi <-
  ifelse(coffee$Processing.Method == 'Semi-washed / Semi-pulped',
         1, 0)
```

Model:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \beta_3 x_{i3} + \beta_4 x_{i4} + \beta_5 x_{i5} + \beta_6 x_{i6} + \beta_7 x_{i7} + \beta_8 x_{i8} + \beta_9 x_{i9} + \beta_{10} x_{i(10)} + \varepsilon_i$$

where

- y = flavour
- $x_1 = 1$ if wet, 0 otherwise
- $x_2 = 1$ if semi, 0 otherwise
- x_3 = Aroma
- x_4 = Aftertaste

- x_5 = Body
- x_6 = Acidity
- x_7 = Balance
- x_8 = Sweetness
- x_9 = Uniformity
- x_{10} = Moisture

```
# Full MLR with our manually coded indicators.
mfull <- lm(
  Flavor ~ wet + semi + Aroma + Aftertaste +
    Body + Acidity + Balance + Sweetness + Uniformity + Moisture,
  dat = coffee
)
summary(mfull)
```

```
##
## Call:
## lm(formula = Flavor ~ wet + semi + Aroma + Aftertaste + Body +
##     Acidity + Balance + Sweetness + Uniformity + Moisture, data = coffee)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.68587 -0.08465  0.00079  0.08910  0.63633
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.728757   0.168516  -4.325 1.67e-05 ***
## wet         -0.033061   0.011024  -2.999  0.00277 **
## semi        -0.001396   0.022021  -0.063  0.94947
## Aroma        0.220302   0.020447  10.774  < 2e-16 ***
## Aftertaste   0.468759   0.023912  19.603  < 2e-16 ***
## Body         0.096140   0.024334   3.951 8.28e-05 ***
## Acidity      0.216751   0.021194  10.227  < 2e-16 ***
## Balance      0.046806   0.022558   2.075  0.03823 *
## Sweetness    0.025507   0.010150   2.513  0.01211 *
## Uniformity   0.016297   0.009803   1.663  0.09669 .
## Moisture     0.169012   0.102480   1.649  0.09938 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.148 on 1108 degrees of freedom
## Multiple R-squared:  0.8091, Adjusted R-squared:  0.8073
## F-statistic: 469.5 on 10 and 1108 DF,  p-value: < 2.2e-16
```

```
# Full MLR alternative, using the factor command.
mfull_alternative <-
  lm(
    Flavor ~ factor(Processing.Method) + Aroma + Aftertaste +
      Body + Acidity + Balance + Sweetness + Uniformity + Moisture,
    dat = coffee
  )
```

Suppose we want to check the VIF for $j = 1$; that is, x_1. Now, we fit:

$$x_{i1} = \alpha_0 + \alpha_2 x_{i2} + \alpha_3 x_{i3} + \alpha_4 x_{i4} + \alpha_5 x_{i5} + \alpha_6 x_{i6} + \alpha_7 x_{i7} + \alpha_8 x_{i8} + \alpha_9 x_{i9} + \alpha_{10} x_{i(10)} + \varepsilon_i$$

```
wet_reg <-
  lm(
    wet ~ semi + Aroma + Aftertaste + Body + Acidity + Balance +
      Sweetness + Uniformity + Moisture,
    dat = coffee
  )
summary(wet_reg)
```

```
##
## Call:
## lm(formula = wet ~ semi + Aroma + Aftertaste + Body + Acidity +
##     Balance + Sweetness + Uniformity + Moisture, data = coffee)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.0015 -0.0283  0.1770  0.2522  0.7704
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.81748    0.45838   1.783 0.074794 .
## semi        -0.75675    0.05551 -13.632  < 2e-16 ***
## Aroma        0.09690    0.05562   1.742 0.081774 .
## Aftertaste  -0.13169    0.06502  -2.026 0.043054 *
## Body        -0.21885    0.06596  -3.318 0.000936 ***
## Acidity      0.18696    0.05746   3.254 0.001173 **
## Balance     -0.10804    0.06136  -1.761 0.078563 .
## Sweetness    0.08373    0.02753   3.041 0.002413 **
## Uniformity   0.03547    0.02668   1.329 0.184053
## Moisture     0.59486    0.27858   2.135 0.032956 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4031 on 1109 degrees of freedom
## Multiple R-squared:  0.1911, Adjusted R-squared:  0.1845
## F-statistic: 29.11 on 9 and 1109 DF,  p-value: < 2.2e-16
```

```
r2_wet <- summary(wet_reg)$r.squared
r2_wet
```

```
## [1] 0.191077
```

R_j: In our case, $R_1 = 0.191077$.

```
VIF_wet <- 1 / (1 - r2_wet)
VIF_wet
```

```
## [1] 1.236212
```

VIF_j: $\text{VIF}_1 = 1.236212$. Interpretation: in a regression with all the variables compared to a regression with just this one, the estimated variance has increased by a factor of 1.24, which is not a very large inflation. The variable wet is not very linearly correlated or dependent on the other predictors that we have in the model.

```
Aroma_reg <- lm(
  Aroma ~ wet + semi + Aftertaste +
    Body + Acidity + Balance + Sweetness + Uniformity + Moisture,
  dat = coffee
)
r2_Aroma <- summary(Aroma_reg)$r.squared
r2_Aroma
```

```
## [1] 0.5204716
```

```
VIF_Aroma <- 1 / (1 - r2_Aroma)
VIF_Aroma
```

```
## [1] 2.085382
```

$R_3 = 0.5204716$, $\text{VIF}_3 = 2.085382$.

```
Aftertaste_reg <- lm(
  Aftertaste ~ wet + semi + Aroma +
    Body + Acidity + Balance + Sweetness + Uniformity + Moisture,
  dat = coffee
)
r2_Aftertaste <- summary(Aftertaste_reg)$r.squared
r2_Aftertaste
```

```
## [1] 0.7101012
```

```
VIF_Aftertaste <- 1 / (1 - r2_Aftertaste)
VIF_Aftertaste
```

```
## [1] 3.449479
```

```
# Load car library for automatic VIF calculation using vif()
library(car)
```

```
vif(mfull)
```

```
##        wet       semi      Aroma Aftertaste       Body    Acidity    Balance
##   1.236212   1.178004   2.085382   3.449479   2.317728   2.232210   3.002813
##  Sweetness Uniformity   Moisture
##   1.159602   1.209901   1.086101
```

No serious signs of inflation, all VIFs are less than 10.

```
## Python in FL everglades example (2017)
## Sex, length, total mass, fat mass, and specimen condition data for
## 248 Burmese pythons (Python bivittatus) collected in the Florida Everglades

python <- read.csv("csv/FLpython.csv")
head(python)
```

```
##   sex  svl mass length    fat
## 1   F 70.0  186   77.5  6.000
## 2   M 76.0  310   83.8 11.000
## 3   M 77.0  260   86.1  6.000
## 4   M 78.0  262   87.1  8.000
## 5   M 81.0  306   91.1  4.000
## 6   M 93.5  605  104.6 18.959
```

```
python$male <- ifelse(python$sex == 'M', 1, 0) # 1 = M, 0 =F

cor(python[, -1])
```

```
##               svl       mass     length        fat       male
## svl     1.0000000  0.8843022  0.9994935  0.8098652 -0.1602418
## mass    0.8843022  1.0000000  0.8858256  0.9419114 -0.2190993
## length  0.9994935  0.8858256  1.0000000  0.8114658 -0.1593512
## fat     0.8098652  0.9419114  0.8114658  1.0000000 -0.2933111
## male   -0.1602418 -0.2190993 -0.1593512 -0.2933111  1.0000000
```

```
pairs(python[, -1])
```

```
mpf <- lm(fat ~ male + svl + mass + length, data = python)
summary(mpf)
```

```
##
## Call:
## lm(formula = fat ~ male + svl + mass + length, data = python)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2445.77  -137.41    -5.29   110.00  1527.27
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.021e+02  1.331e+02   1.518    0.130
## male        -1.971e+02  4.732e+01  -4.165 4.32e-05 ***
## svl         -3.370e+00  1.125e+01  -0.300    0.765
## mass         1.178e-01  5.302e-03  22.210  < 2e-16 ***
## length       1.594e+00  1.010e+01   0.158    0.875
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 360.9 on 243 degrees of freedom
## Multiple R-squared:  0.897,  Adjusted R-squared:  0.8953
## F-statistic:   529 on 4 and 243 DF,  p-value: < 2.2e-16
```

```
vif(mpf)
```

```
##        male         svl        mass      length
##    1.058699  994.546545    4.813078 1007.484200
```

```
mpf_l <- lm(length ~ male + svl + mass, data = python)
1 / (1 - summary(mpf_l)$r.squared)
```

```
## [1] 1007.484
```

Misleading conclusion: `svl` and `length` are both irrelevant (this is not the case). Also, the standard errors are very large.

```
# remove "length" based on VIF
mpf2 <- lm(fat ~ male + mass + svl, data = python)
summary(mpf2)$adj
```

```
## [1] 0.8957164
```

```r
vif(mpf2)
```

```
##     male     mass      svl
## 1.056139 4.720065 4.611903
```

```r
anova(mpf2)
```

```
## Analysis of Variance Table
##
## Response: fat
##            Df    Sum Sq   Mean Sq   F value  Pr(>F)
## male        1  26435988  26435988  203.7689 < 2e-16 ***
## mass        1 248624259 248624259 1916.3988 < 2e-16 ***
## svl         1    567377    567377    4.3734 0.03754 *
## Residuals 244  31655372    129735
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
AIC(mpf2)
```

```
## [1] 3629.527
```

`svl` now has a significant t-statistic.