Multicollinearity

The formulas are primarily derived from *Linear Regression Analysis, Fifth Edition* by Montgomery et al. and/or the materials from the STAT 6021 course taught by Dr. Woo at the University of Virginia. Except where specially cited, it is believed that the materials are well-known equations and concepts in the public domain. If you believe otherwise, please reach out to me through my Github account so that I can correct the material.

Variance inflation factor

We define the variance inflation factor as

$$\text{VIF}_j = \frac{1}{1 - R_j^2}$$

where R_j^2 is the coefficient of multiple determination (i.e., the non-adjusted R^2) when fitting x_j as the response of the other predictors:

$$x_j = \zeta_0 + \zeta_1 x_1 + \cdots + \zeta_{j-1} x_{j-1} + \zeta_{j+1} x_{j+1} + \cdots + \zeta_n x_n.$$

Interpretation

Here are some rules for interpreting VIFs:

- The book notes that values higher than five may imply a problem: "practical experience indicates that if any of the VIF exceed 5 or 10, it is an indication that the associated regression coefficient are poorly etimated becaue of multicollinearity" (p. 296).
- VIFs also affect the length of the confidence intervals for coefficients. According to the book, the j confidence interval will be longer by a factor of $\sqrt{\text{VIF}_j}$.

Example calculation

For example:

```r
library(faraway)

# produce some hypothetical data
t = seq(1,10, by=0.1)           # just some linear values
x1 = sin(t)                     # x1
x2 = t + x1                     # x2
e = rnorm(length(t), sd = 10)   # error term for y
y = x1 + 10*x2  + e             # hypothetical model
#data.frame(y, x1, x2)           # (uncomment to) print the outcome

# ask R to fit the model
model = lm(y ~ x1 + x2)         # fit the model
# summary(model)                  # (uncomment to) print the summary

# calculate the VIFs with faraway
```

```r
faraway_vifs =  vif(cbind(x1,x2))

# now calculate them directly using 1 / (1 - R^2)
x1_r2 = summary(lm(x1 ~ x2))$r.squared
x2_r2 = summary(lm(x2 ~ x1))$r.squared
our_vifs = 1 / (1 - c(x1_r2, x2_r2))

results = as.matrix(rbind(faraway_vifs, our_vifs))
results
```

```
##                    x1       x2
## faraway_vifs 1.058237 1.058237
## our_vifs     1.058237 1.058237
```

Matrix calculation

The values of VIF_j is also the value of C_{jj} when fitting the linear model. This leads to an alternative formula:

$$\text{VIF}_j = ((\mathbf{W}'\mathbf{W})^{-1})_{jj}$$

where $\mathbf{W}$ arises from the scaled version of $\mathbf{X}$ obtained when using unit length scaling.

We show this with R:

```r
s1 = sum((x1 - mean(x1))^2)
s2 = sum((x2 - mean(x2))^2)
w1 = (x1 - mean(x1)) / sqrt(s1)
w2 = (x2 - mean(x2)) / sqrt(s2)
w = as.matrix(data.frame(w1, w2))
result = diag(solve(t(w) %*% w))
c("vif" = result)
```

```
##   vif.w1   vif.w2
## 1.058237 1.058237
```