Multicollinearity

Monday, April 13, 2020 1:27 PM

multicollin...

Multiple Regression

Multicollinearity: Sleuth3 Chapter 12

Background

- Problems arise when too many explanatory variables are included in a model, particularly when some of these explanatory variables are correlated with each other
- In particular:
  - *Precision* in estimating important regression coefficients can be lost (meaning the variance is too high for those estimates to learn enough about the coefficients);
  - *Prediction* for a future response (value) may be negatively impacted
- The *variance inflation factor (VIF)* can help us measure the amount of *multicollinearity* in a set of candidate explanatory variables
  - Suppose you have an explanatory variable, X_j, in your regression model, and $R^2_{X_j}$ is the proportion of the variation in X_j that is explained by its relationship to other explanatory variables (this is like the coefficient of determination, R^2). *Multicollinearity* arises when X_j can be explained well by other explanatory variables. In other words, X_j is highly correlated with other explanatory variables in the model.
  - To determine the degree of multicollinearity between each X_j variable and the other explanatory variables in the model, we calculate:

$$VIF_j = \frac{1}{1 - R^2_{X_j}}$$

If large, X_j is explained well by X_{-j}, so $R^2_{X_j}$ is large (between 0 and 1), and VIF_j will be large.

VIF Rules of Thumb:

- VIF < 4: no multicollinearity between X_j and other explanatory variables
- $4 \leq$ VIF ≤ 10: moderate multicollinearity - warrents further investigation
- VIF > 10: serious multicollinearity - requires correction

Simulation with multicollinearity

Here I am going to simulate some data where some variables are correlated, and others are not to illustrate the multicollinearity issue and discuss cutoffs for VIF.

```
x1 <- rnorm(100, 2, 1)
x2 <- rnorm(100, 1, 5)
x3 <- x1 + x2 + rnorm(100, 0, 1)
x4 <- x2 + rnorm(100, 0, 1)
y <- rnorm(100, mean=x1+x2, sd=sd(x1+x2))
sim_df <- data.frame(y=y,
                     x1=x1,
                     x2=x2,
                     x3=x3,
                     x4=x4)

lm_sim <- lm(y ~ x1 + x2 + x3 + x4, data=sim_df)
vif(lm_sim) # function vif from car package
```

```
##        x1        x2        x3        x4
##  2.369171 51.518287 24.420758 23.252615
```

```
confint(lm_sim)
```

```
##                  2.5 %    97.5 %
## (Intercept) -1.9147608 2.6721559
## x1          -0.5941507 2.2416647
## x2          -0.2448766 2.8679931
## x3          -0.7484275 1.3445816
## x4          -1.6954309 0.3448929
```

Removing multicollinearity

```
lm_sim_red <- lm(y ~ x1 + x2, data=sim_df)

vif(lm_sim_red) # function vif from car package
```

```
##       x1       x2
## 1.027656 1.027656
```

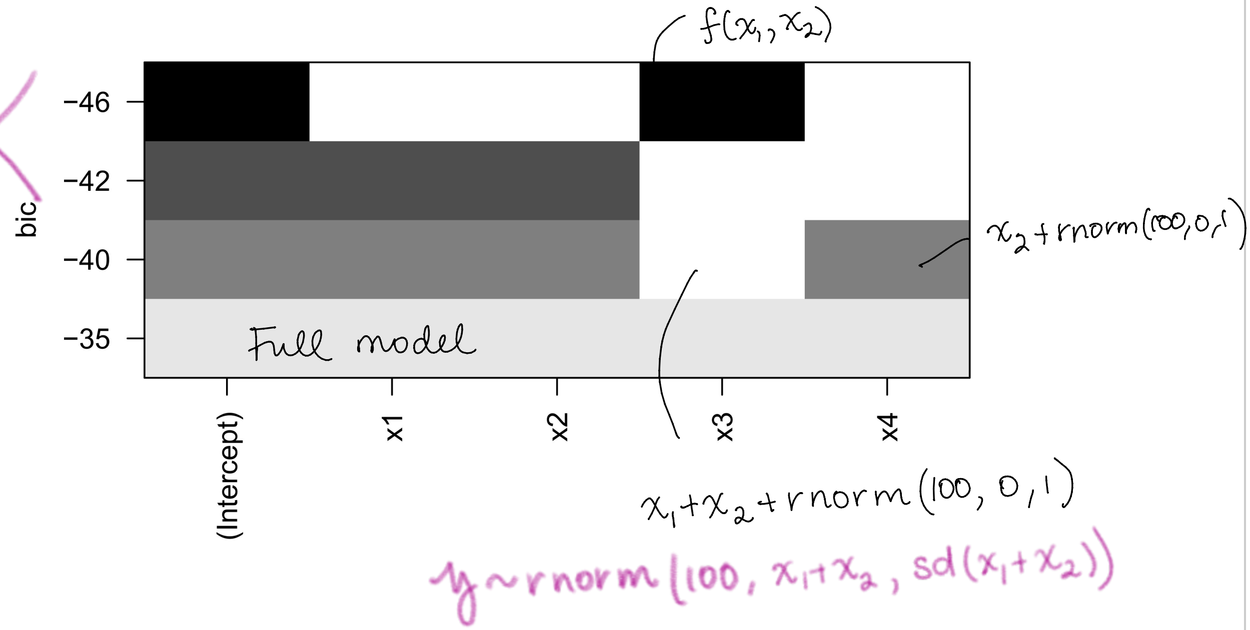
```
confint(lm_sim_red)
```

```
##                  2.5 %   97.5 %
## (Intercept) -2.0657199 2.503585
## x1           0.2617680 2.131011
## x2           0.7194877 1.159501
```

Relationship to all subsets regression

```
candidate_models <- regsubsets(y ~ x1 + x2 + x3 + x4, data=sim_df)
plot(candidate_models)
```

reasonable




```r
summary(candidate_models)
```

```
## Subset selection object
## Call: regsubsets.formula(y ~ x1 + x2 + x3 + x4, data = sim_df)
## 4 Variables  (and intercept)
##    Forced in Forced out
## x1     FALSE      FALSE
## x2     FALSE      FALSE
## x3     FALSE      FALSE
## x4     FALSE      FALSE
## 1 subsets of each size up to 4
## Selection Algorithm: exhaustive
##          x1  x2  x3  x4
## 1  ( 1 ) " " " " "*" " "
## 2  ( 1 ) "*" "*" " " " "
## 3  ( 1 ) "*" "*" " " "*"
## 4  ( 1 ) "*" "*" "*" "*"
```

```r
summary(candidate_models)$bic
```

```
## [1] -46.08163 -42.40569 -39.76786 -35.49870
```

does better because functionally equivalent, but only one variable