Detecting Multicollinearity in R

Grady Keene, Yuka Chen

2022-04-10

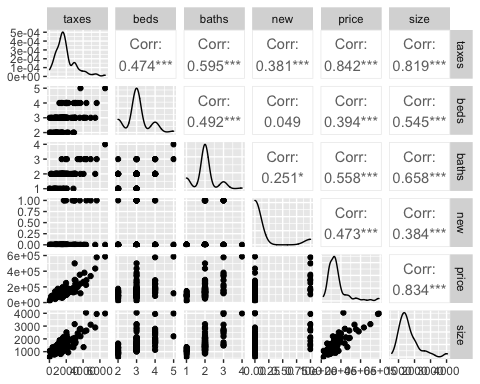
library(tidyverse)  
library(broom)  
library(GGally)  
library(fastDummies)  
library(car)

Houses <- read\_table("https://users.stat.ufl.edu/~aa/smss/data/Houses.dat",   
 col\_types = cols(X7 = col\_skip()))  
names(Houses)[2] <- 'beds'  
glimpse(Houses)

## Rows: 100  
## Columns: 6  
## $ taxes <dbl> 3104, 1173, 3076, 1608, 1454, 2997, 4054, 3002, 6627, 320, 630, …  
## $ beds <dbl> 4, 2, 4, 3, 3, 3, 3, 3, 5, 3, 3, 3, 3, 3, 3, 2, 3, 3, 2, 3, 2, 4…  
## $ baths <dbl> 2, 1, 2, 2, 3, 2, 2, 2, 4, 2, 2, 2, 2, 2, 1, 1, 2, 1, 2, 2, 1, 3…  
## $ new <dbl> 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1…  
## $ price <dbl> 279900, 146500, 237700, 200000, 159900, 499900, 265500, 289900, …  
## $ size <dbl> 2048, 912, 1654, 2068, 1477, 3153, 1355, 2075, 3990, 1160, 1220,…

# Correlation Pair Matrix

ggpairs(Houses)



# Creating the Model With All Variables

model <- lm(price ~ taxes + beds + baths + new + size, Houses)  
glance(model)

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.793 0.782 47238. 72.2 1.17e-30 5 -1215. 2444. 2462.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

tidy(model)

## # A tibble: 6 × 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 4526. 24474. 0.185 0.854   
## 2 taxes 38.1 6.82 5.60 0.000000216  
## 3 beds -11259. 9115. -1.24 0.220   
## 4 baths -2114. 11465. -0.184 0.854   
## 5 new 41711. 16887. 2.47 0.0153   
## 6 size 68.4 13.9 4.90 0.00000392

new\_model <- dummy\_cols(Houses, select\_columns = "new", remove\_selected\_columns = TRUE)  
dummy\_model <- lm(price ~ taxes + beds + baths + new\_1 + size, data = new\_model)  
tidy(dummy\_model)

## # A tibble: 6 × 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) 4526. 24474. 0.185 0.854   
## 2 taxes 38.1 6.82 5.60 0.000000216  
## 3 beds -11259. 9115. -1.24 0.220   
## 4 baths -2114. 11465. -0.184 0.854   
## 5 new\_1 41711. 16887. 2.47 0.0153   
## 6 size 68.4 13.9 4.90 0.00000392

glancedummy <- glance(dummy\_model)  
glancedummy

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.793 0.782 47238. 72.2 1.17e-30 5 -1215. 2444. 2462.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

### Finding the VIF Values

#### Calculating Taxes VIF

taxes <- lm(taxes ~ 1 + beds + baths + new\_1 + size, data = new\_model)  
taxes\_g <- glance(taxes)  
taxes\_g

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.682 0.669 711. 51.0 7.46e-23 4 -796. 1604. 1620.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

taxesvif <- 1/(1 - taxes\_g[[1]])  
taxesvif

## [1] 3.147119

#### Calculating beds VIF

beds <- lm(beds ~ taxes + 1 + baths + new\_1 + size, data = new\_model)  
beds\_g <- glance(beds)  
beds\_g

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.361 0.334 0.532 13.4 0.0000000108 4 -76.2 164. 180.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

bedsvif <- 1/(1 - beds\_g[[1]])  
bedsvif

## [1] 1.563795

#### Calculating baths VIF

baths <- lm(baths ~ taxes + beds + 1 + new\_1 + size, data = new\_model)  
baths\_g <- glance(baths)  
baths\_g

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.467 0.444 0.423 20.8 2.46e-12 4 -53.2 118. 134.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

bathsvif <- 1/(1 - baths\_g[[1]])  
bathsvif

## [1] 1.875628

#### Calculating new\_1 VIF

new\_1 <- lm(new\_1 ~ taxes + beds + baths + 1 + size, data = new\_model)  
new\_1\_g <- glance(new\_1)  
new\_1\_g

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.201 0.167 0.287 5.97 0.000251 4 -14.5 41.0 56.6  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

new\_1vif <- 1/(1 - new\_1\_g[[1]])  
new\_1vif

## [1] 1.251166

#### Calculating size VIF

size <- lm(size ~ taxes + beds + baths + new\_1 + 1, data = new\_model)  
size\_g <- glance(size)  
size\_g

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.739 0.728 348. 67.3 6.91e-27 4 -724. 1461. 1477.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

sizevif <- 1/(1 - size\_g[[1]])  
sizevif

## [1] 3.832948

#### We can use vif() function from package {car} to see all variables’ VIF

vif(dummy\_model)

## taxes beds baths new\_1 size   
## 3.147119 1.563795 1.875628 1.251166 3.832948

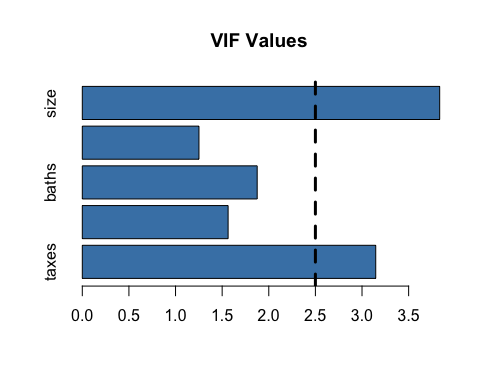
## Correlation Matrix

x\_vari <- new\_model[,c("taxes", "beds", "new\_1", "baths", "size")]  
cor(x\_vari)

## taxes beds new\_1 baths size  
## taxes 1.0000000 0.47392873 0.38087410 0.5948543 0.8187958  
## beds 0.4739287 1.00000000 0.04931556 0.4922224 0.5447831  
## new\_1 0.3808741 0.04931556 1.00000000 0.2514810 0.3843277  
## baths 0.5948543 0.49222235 0.25148095 1.0000000 0.6582247  
## size 0.8187958 0.54478311 0.38432773 0.6582247 1.0000000

## Visualize Predictor VIFs

vif\_vals <- vif(dummy\_model)  
barplot(vif\_vals, main = "VIF Values", horiz = TRUE, col = "steelblue")  
abline(v = 2.5, lwd = 3, lty = 2)



### Without taxes, as taxes and size are highly correlated

model\_2 <- lm(price ~ beds + baths + new\_1 + size, new\_model)  
tidy(model\_2)

## # A tibble: 5 × 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -28849. 27261. -1.06 2.93e- 1  
## 2 beds -8202. 10450. -0.785 4.34e- 1  
## 3 baths 5274. 13080. 0.403 6.88e- 1  
## 4 new\_1 54562. 19215. 2.84 5.53e- 3  
## 5 size 118. 12.3 9.59 1.27e-15

## Original Model

## Model Without “taxes”

### Detecting Multicollinearity in R - Second Data Set

The variables for this data set are violent crime rate (number of violent crimes per 100,000 population), murder rate, percent in metropolitan areas, percent white, percent high school graduates,percent below the poverty level, and percent of families headed by a single parent. The data are from StatisticalAbstract of the United States for 2005.

Crime <- read\_table("https://users.stat.ufl.edu/~aa/smss/data/Crime2.dat",   
 col\_types = cols(X9 = col\_skip()))

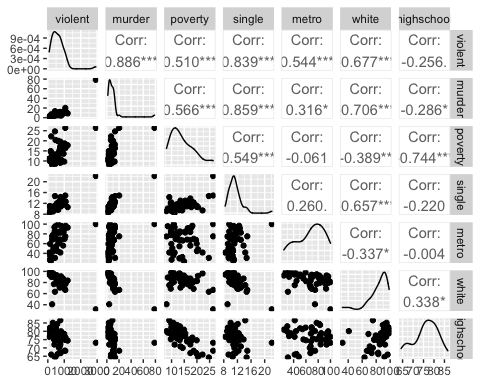
## Warning: Missing column names filled in: 'X9' [9]

glimpse(Crime)

## Rows: 51  
## Columns: 8  
## $ State <chr> "AK", "AL", "AR", "AZ", "CA", "CO", "CT", "DE", "FL", "GA",…  
## $ violent <dbl> 761, 780, 593, 715, 1078, 567, 456, 686, 1206, 723, 261, 32…  
## $ murder <dbl> 9.0, 11.6, 10.2, 8.6, 13.1, 5.8, 6.3, 5.0, 8.9, 11.4, 3.8, …  
## $ poverty <dbl> 9.1, 17.4, 20.0, 15.4, 18.2, 9.9, 8.5, 10.2, 17.8, 13.5, 8.…  
## $ single <dbl> 14.3, 11.5, 10.7, 12.1, 12.5, 12.1, 10.1, 11.4, 10.6, 13.0,…  
## $ metro <dbl> 41.8, 67.4, 44.7, 84.7, 96.7, 81.8, 95.7, 82.7, 93.0, 67.7,…  
## $ white <dbl> 75.2, 73.5, 82.9, 88.6, 79.3, 92.5, 89.0, 79.4, 83.5, 70.8,…  
## $ highschool <dbl> 86.6, 66.9, 66.3, 78.7, 76.2, 84.4, 79.2, 77.5, 74.4, 70.9,…

# Correlation Pair Matrix

ggpairs(Crime[2:8])



# Creating the Model With All Variables

model2 <- lm(violent ~ murder + poverty + single + metro + white + highschool, Crime)  
glance(model2)

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.895 0.881 152. 62.5 6.52e-20 6 -325. 666. 681.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

tidy(model2)

## # A tibble: 7 × 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -1144. 585. -1.95 0.0570   
## 2 murder 19.3 4.44 4.35 0.0000794   
## 3 poverty 15.0 9.72 1.54 0.130   
## 4 single 54.9 21.3 2.57 0.0135   
## 5 metro 6.62 1.12 5.92 0.000000442  
## 6 white -0.696 2.51 -0.278 0.783   
## 7 highschool 4.79 6.68 0.717 0.477

### Finding the Individual VIF Values

#### Calculating murder VIF

murder <- lm(murder ~ 1 + poverty + single + metro + white + highschool, data = Crime)  
murder\_g <- glance(murder)  
murder\_g

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.795 0.773 5.11 35.0 1.98e-14 5 -152. 319. 332.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

murdervif <- 1/(1 - murder\_g[[1]])  
murdervif

## [1] 4.885397

#### Calculating poverty VIF

poverty <- lm(poverty ~ murder + 1 + single + metro + white + highschool, data = Crime)  
poverty\_g <- glance(poverty)  
  
povertyvif <- 1/(1 - poverty\_g[[1]])  
povertyvif

## [1] 4.278128

#### Calculating single VIF

single <- lm(single ~ murder + poverty + 1 + metro + white + highschool, data = Crime)  
single\_g <- glance(single)  
  
singlevif <- 1/(1 - single\_g[[1]])  
singlevif

## [1] 4.400805

#### Calculating metro VIF

metro <- lm(metro ~ murder + poverty + single + 1 + white + highschool, data = Crime)  
metro\_g <- glance(metro)  
  
metrovif <- 1/(1 - metro\_g[[1]])  
metrovif

## [1] 1.299233

#### Calculating white VIF

white <- lm(white ~ murder + poverty + single + metro + 1 + highschool, data = Crime)  
white\_g <- glance(white)  
  
whitevif <- 1/(1 - white\_g[[1]])  
whitevif

## [1] 2.375882

#### Calculating highschool VIF

highschool <- lm(highschool ~ murder + poverty + single + metro + white + 1, data = Crime)  
highschool\_g <- glance(highschool)  
  
highschoolvif <- 1/(1 - highschool\_g[[1]])  
highschoolvif

## [1] 3.002861

#### We can use vif() function from package {car} to see all variables’ VIF

vif(model2)

## murder poverty single metro white highschool   
## 4.885397 4.278128 4.400805 1.299233 2.375882 3.002861

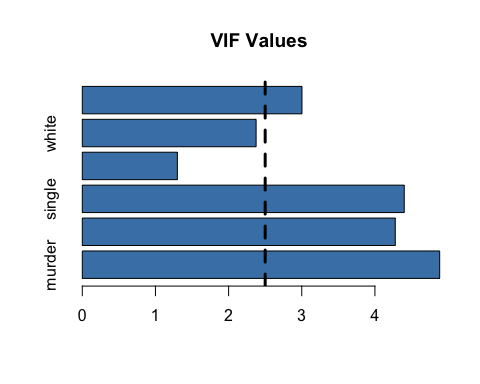
## Correlation Matrix

x\_vari2 <- Crime[ , c("murder", "poverty", "single", "metro", "white", "highschool")]  
cor(x\_vari2)

## murder poverty single metro white  
## murder 1.0000000 0.5658711 0.8589106 0.316114166 -0.7062589  
## poverty 0.5658711 1.0000000 0.5485890 -0.060538499 -0.3891346  
## single 0.8589106 0.5485890 1.0000000 0.259810085 -0.6567078  
## metro 0.3161142 -0.0605385 0.2598101 1.000000000 -0.3374351  
## white -0.7062589 -0.3891346 -0.6567078 -0.337435120 1.0000000  
## highschool -0.2860708 -0.7439382 -0.2197829 -0.003977358 0.3381212  
## highschool  
## murder -0.286070828  
## poverty -0.743938249  
## single -0.219782892  
## metro -0.003977358  
## white 0.338121236  
## highschool 1.000000000

## Visualize Predictor VIFs

vif\_vals2 <- vif(model2)  
barplot(vif\_vals2, main = "VIF Values", horiz = TRUE, col = "steelblue")  
abline(v = 2.5, lwd = 3, lty = 2)



### Without murder, as murder, single, and white are highly correlated

model2\_1 <- lm(violent ~ poverty + single + metro + white + highschool, Crime)  
glance(model2\_1)

## # A tibble: 1 × 12  
## r.squared adj.r.squared sigma statistic p.value df logLik AIC BIC  
## <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>  
## 1 0.850 0.833 180. 50.9 2.05e-17 5 -334. 682. 696.  
## # … with 3 more variables: deviance <dbl>, df.residual <int>, nobs <int>

tidy(model2\_1)

## # A tibble: 6 × 5  
## term estimate std.error statistic p.value  
## <chr> <dbl> <dbl> <dbl> <dbl>  
## 1 (Intercept) -1796. 669. -2.69 0.0101   
## 2 poverty 26.2 11.1 2.37 0.0222   
## 3 single 109. 20.4 5.38 0.00000260   
## 4 metro 7.61 1.30 5.87 0.000000480  
## 5 white -4.48 2.78 -1.61 0.114   
## 6 highschool 8.65 7.83 1.10 0.275

## With variable “murder”

## Without Variable “murder”

## VIF Values of Model Without “murder”

vif(model2\_1)

## poverty single metro white highschool   
## 3.975997 2.873489 1.245801 2.089245 2.949878