Multicollinearity

As a data-scientist most of the time we have to work on some unknown problems or data. I am cricket fan and what if someone gave me data of baseball for analysis. It is very important that My inability to understand different terms in the data due to my lack of knowledge about the sport shouldn't impact the model /analysis. How can we make sure that we can avoid some of the common mistakes while building or planning for our model? Few things that makes a model a parsimonious.

> *A parsimonious model is a model that accomplishes a desired level of explanation or prediction with as few predictor variables as possible. They explain data with a minimum number of parameters, or predictor variables.*

We know the four assumption of Linear Regression lm(X~y) :

Linearity: The relationship between X and the mean of Y is linear.

Homoscedasticity: The variance of residual is the same for any value of X.

Independence: Observations are independent of each other.

Normality: For any fixed value of X, Y is normally distributed.

> *Multicollinearity: Multicollinearity (also collinearity) is a statistical phenomenon in which two or more predictor variables in a multiple regression model are highly correlated, meaning that one can be linearly predicted from the others with a non-trivial degree of accuracy.*

Multicollinearity is a problem because it undermines the statistical significance of an independent variable as other variables are correlated, higher degree of correlation between variables can't guarantee that our model can better explain all the variations of the data.

Here I have data about Kids score depending upon moms age and if mom were working(mom_work) and Mom_iq. Let's try to fit the model based on this data.

Response: - Kid_score

Predicotr:- mom_hs , mom_iq , mom_work, mom_age

```
cognitive <- read.csv("http://bit.ly/dasi_cognitive")
head(cognitive)
```

```
##   kid_score mom_hs    mom_iq mom_work mom_age
## 1        65    yes 121.11753      yes      27
## 2        98    yes  89.36188      yes      25
## 3        85    yes 115.44316      yes      27
## 4        83    yes  99.44964      yes      25
```

```
## 5         115     yes  92.74571       yes       27
## 6          98      no 107.90184        no       18
```

Here I am going to create some colinear predictors then we will see impact of those eon the model.

Model1

```
cog_full = lm(kid_score~.,data = cognitive)
summary(cog_full)
```

Adding new Predictors

```
cognitive$c_ageiq <- cognitive$mom_age*cognitive$mom_iq
cognitive$c_iq  <- cognitive$mom_iq^2
head((cognitive))
```

```
##   kid_score mom_hs    mom_iq mom_work mom_age  c_ageiq      c_iq
## 1        65    yes 121.11753      yes      27 3270.173 14669.456
## 2        98    yes  89.36188      yes      25 2234.047  7985.546
## 3        85    yes 115.44316      yes      27 3116.965 13327.124
## 4        83    yes  99.44964      yes      25 2486.241  9890.231
## 5       115    yes  92.74571      yes      27 2504.134  8601.767
## 6        98     no 107.90184       no      18 1942.233 11642.807
```

Side by Side Mode 1 and Model 2

```
Call:
lm(formula = kid_score ~ ., data = cognitive)

Residuals:
    Min      1Q  Median      3Q     Max 
-54.045 -12.918   1.992  11.563  49.267 

Coefficients:
            Estimate Std. Error t value Pr(>|t|)    
(Intercept) 19.59241    9.21906   2.125   0.0341 *  
mom_hsyes    5.09482    2.31450   2.201   0.0282 *  
mom_iq       0.56147    0.06064   9.259   <2e-16 ***
mom_workyes  2.53718    2.35067   1.079   0.2810    
mom_age      0.21802    0.33074   0.659   0.5101    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 18.14 on 429 degrees of freedom
Multiple R-squared:  0.2171,	Adjusted R-squared:  0.2098 
F-statistic: 29.74 on 4 and 429 DF,  p-value: < 2.2e-16
```

```
Call:
lm(formula = kid_score ~ ., data = cognitive)

Residuals:
    Min      1Q  Median      3Q     Max 
-55.286 -11.102   2.476  11.594  48.622 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept) -1.624e+02  6.687e+01  -2.428 0.015591 *  
mom_hsyes    4.018e+00  2.313e+00   1.737 0.083067 .  
mom_iq       3.543e+00  9.368e-01   3.782 0.000178 ***
mom_workyes  1.630e+00  2.342e+00   0.696 0.486653    
mom_age      2.818e+00  2.200e+00   1.281 0.200937    
c_ageiq     -2.378e-02  2.132e-02  -1.115 0.265487    
c_iq        -1.170e-02  3.608e-03  -3.242 0.001282 ** 
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 17.95 on 427 degrees of freedom
Multiple R-squared:  0.2371,	Adjusted R-squared:  0.2264 
F-statistic: 22.12 on 6 and 427 DF,  p-value: < 2.2e-16
```

As we can see the two model1(cog_full) and Model2 (cog_full2),

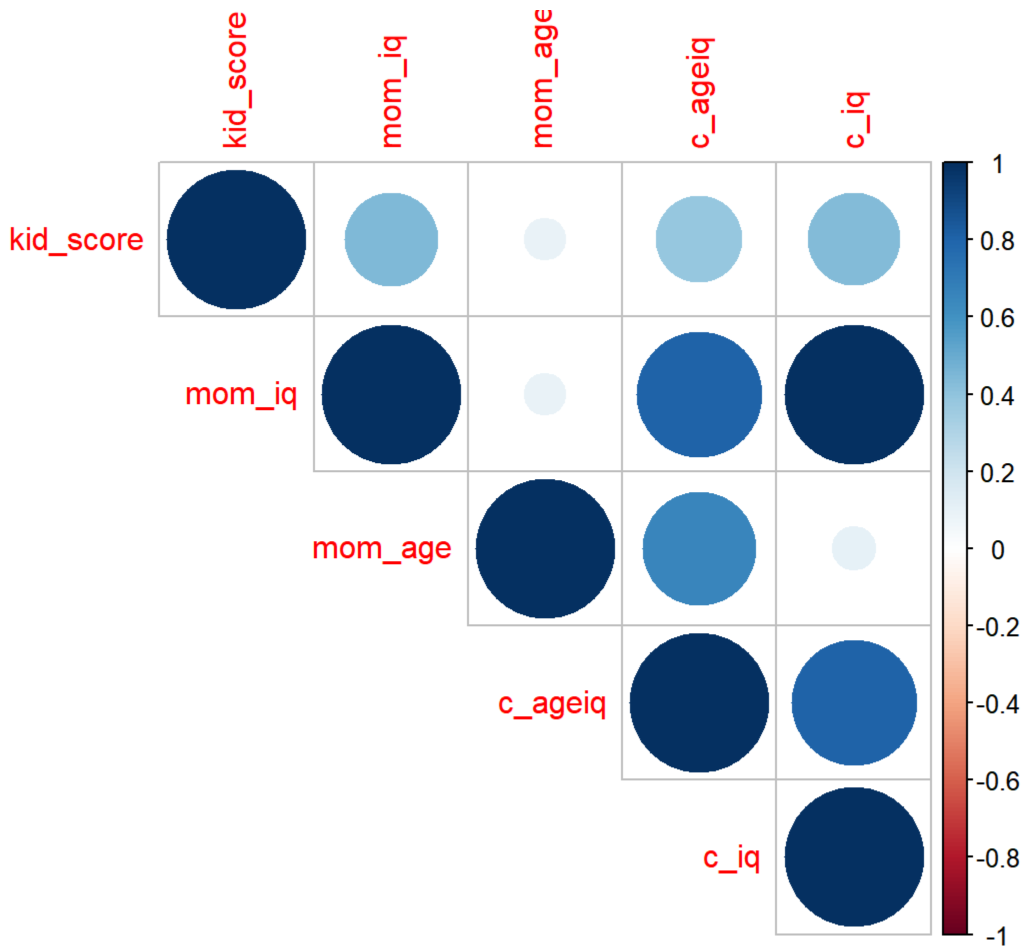
Model Name	**R-squared**	**Adjusted R-squared**	**Significant**	**standard error**
Model1	0.2171	0.2098	mom_hsyes, mom_iq	18.14
Model2 (c)	0.2371	0.2264	mom_iq,c_iq	17.95

From the above two model we see the coefficients of mom_iq is showing very high std error and its p-value has also gone down. Why we see this problem, it’s because of existing correlated predictor `c_ageiq` and `c_iq`.

How to catch such collinearity which is present in data and not easy to locate, We can use two ways to test theses:

a. Using the Correlation among the predictors.
b. VIF (variance inflation factor), which measures how much the variance of a regression coefficient is inflated due to multicollinearity in the model. The smallest possible value of VIF is one (absence of multicollinearity). Here we will look for VIF value, if that exceeds 5 or 10 indicates a problematic amount of collinearity. Read More.

```
library(corrplot)
corrplot(cor(cognitive[,-c(2,4)],use="pairwise.complete.obs"),type = 'upper')
```



Correlation among variables:

```
cor(cognitive[,-c(2,4)])[,c(2,5)]
##              mom_iq      c_iq
## kid_score 0.4482758 0.4354671
## mom_iq    1.0000000 0.9968604
## mom_age   0.0916084 0.1020082
## c_ageiq   0.8045191 0.8077851
## c_iq      0.9968604 1.0000000
```

Above plots and correlation matrix, very clearly shows that `Mom_iq` ia colinear with `c_iq` and `c_ageiq`

Now we will see Variance Inflation Factors for each predictor in the model.

```
library(car)
vif(cog_full2)
```

```
##     mom_hs     mom_iq   mom_work    mom_age    c_ageiq       c_iq
##   1.212658 265.287084   1.077794  47.457234 132.212711 169.667177
```

Here again the value from Correlation matrix and Variance Inflation Factors, is giving same result with some correlation that exist between predictors: `mom_iq` , `mom_iq` , `c_ageiq` , `c_iq`

In contrast if we check the inflation factor for the `Model1` we see that none of the predictors are correlated. Which is good sign for further improving this model.

```
vif(cog_full)
```

```
##   mom_hs   mom_iq mom_work  mom_age
## 1.189102 1.088349 1.063187 1.049762
```

High Variance Inflation Factor (VIF) and Low Tolerance: These two useful statistics are reciprocals of each other. So either a high VIF or a low tolerance is indicative of multicollinearity. VIF is a direct measure of how much the variance of the coefficient (ie. its standard error) is being inflated due to multicollinearity.

Ref:

https://www.theanalysisfactor.com/eight-ways-to-detect-multicollinearity/

http://www.sthda.com/english/articles/39-regression-model-diagnostics/160-multicollinearity-essentials-and-vif-in-r/