Multicollinearity

Pranav Gopalkrishna, 1940223

2021-10-28

# DATA SET

myData = read.csv("~/Documents/Study/computerScience/programming/r/data/multicollinearityData.csv")  
head(myData)

## y x1 x2 x3 x4 x5 x6 x7 x8 x9 x10  
## 1 2.13 0.789 39.8 66.9 23.4 33.4 77.3 79.2 15.3 92.1 90.2  
## 2 2.69 0.644 41.7 63.4 41.4 30.4 60.4 42.1 27.7 95.6 83.4  
## 3 1.94 0.681 36.1 72.6 14.4 29.9 79.5 66.0 10.1 88.4 94.0  
## 4 2.38 0.601 44.7 52.6 16.1 32.0 53.9 77.2 14.4 80.8 78.1  
## 5 1.99 0.679 41.7 63.3 21.6 29.7 68.7 56.1 25.3 99.5 82.9  
## 6 3.21 0.537 65.3 47.2 58.4 30.2 36.2 33.9 49.2 81.8 73.9

# LINEAR REGRESSION MODEL

model = lm(y~., data = myData)  
summary(model)

##   
## Call:  
## lm(formula = y ~ ., data = myData)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -0.43837 -0.14250 -0.04833 0.19676 0.35463   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 4.686737 1.970421 2.379 0.028660 \*   
## x1 0.613751 1.488450 0.412 0.684958   
## x2 -0.001258 0.009745 -0.129 0.898732   
## x3 -0.033710 0.007148 -4.716 0.000172 \*\*\*  
## x4 0.019574 0.008449 2.317 0.032514 \*   
## x5 -0.024824 0.024090 -1.030 0.316427   
## x6 0.005273 0.011015 0.479 0.637931   
## x7 -0.017476 0.006283 -2.782 0.012310 \*   
## x8 -0.007767 0.011157 -0.696 0.495198   
## x9 -0.006798 0.006318 -1.076 0.296125   
## x10 0.012862 0.018141 0.709 0.487409   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 0.2751 on 18 degrees of freedom  
## Multiple R-squared: 0.8994, Adjusted R-squared: 0.8434   
## F-statistic: 16.08 on 10 and 18 DF, p-value: 5.185e-07

## Interpretations

By the adjusted R-squared value, the above model seems to explain around 84.34% of the variation on the response, and judging by the p-value (which is less than 0.05), the coefficients of the regressors are significant in the model i.e. they contribute significantly to the explanatory power of the model.

# FINDING CORRELATION BETWEEN REGRESSORS

## Necessary package(s)

# install.packages("MASS")  
# install.packages("ppcor")  
library(ppcor)

## Loading required package: MASS

## Warning: package 'MASS' was built under R version 3.6.2

Official documentation: <https://rdrr.io/cran/ppcor/man/pcor.html>

**ppcor** => partial and semi-partial (part) correlation  
This library helps calculate partial and semi-partial (part) correlations in a linear regression model. It also calculates the p-values of each correlation coefficient (discussed later).

## Choosing columns of independent variables

# All rows from all except the 1st column...  
xs = myData[, -1]  
head(xs)

## x1 x2 x3 x4 x5 x6 x7 x8 x9 x10  
## 1 0.789 39.8 66.9 23.4 33.4 77.3 79.2 15.3 92.1 90.2  
## 2 0.644 41.7 63.4 41.4 30.4 60.4 42.1 27.7 95.6 83.4  
## 3 0.681 36.1 72.6 14.4 29.9 79.5 66.0 10.1 88.4 94.0  
## 4 0.601 44.7 52.6 16.1 32.0 53.9 77.2 14.4 80.8 78.1  
## 5 0.679 41.7 63.3 21.6 29.7 68.7 56.1 25.3 99.5 82.9  
## 6 0.537 65.3 47.2 58.4 30.2 36.2 33.9 49.2 81.8 73.9

## Finding correlation matrix for independent variables

The p-value of a correlation coefficient for two variables (obtained from a sample) is the probability that you would have found the current result if the correlation coefficient were in fact zero (i.e. if you found correlation in the sample while there is none in the population i.e. wrongly rejected null hypothesis). If this probability is lower than the significance level, the correlation coefficient is said to be statistically significant.

### NOTE

Significance level is the proportion of the lowest probability values of a statistic that, if the statistic actually takes such a value, you would consider it as significantly different from the population. Here, the population distribution contains probabilities of getting values assuming that the null hypothesis is true (in this case, null hypothesis is that there is zero correlation between the two variables).

## Correlation matrix

correlMatrix = pcor(xs, method = "pearson")  
correlMatrix

## $estimate  
## x1 x2 x3 x4 x5 x6  
## x1 1.00000000 -0.40045413 -0.12610881 0.2540673 0.010496507 0.52771043  
## x2 -0.40045413 1.00000000 0.29347828 0.1161464 0.040295079 -0.09797283  
## x3 -0.12610881 0.29347828 1.00000000 0.2829794 0.181736978 0.22637593  
## x4 0.25406730 0.11614637 0.28297937 1.0000000 0.302517350 -0.33711836  
## x5 0.01049651 0.04029508 0.18173698 0.3025173 1.000000000 0.12557469  
## x6 0.52771043 -0.09797283 0.22637593 -0.3371184 0.125574690 1.00000000  
## x7 0.29290290 -0.39022525 0.47896446 -0.1423483 0.131670151 -0.36859543  
## x8 0.36437311 0.08076675 0.09207697 0.2570188 -0.260828974 -0.40818080  
## x9 -0.09131298 -0.27409977 0.57618346 0.1903150 -0.444159270 -0.05841553  
## x10 0.42036466 -0.12968435 0.42637473 -0.4479665 -0.008057368 0.02020569  
## x7 x8 x9 x10  
## x1 0.2929029 0.36437311 -0.091312979 0.420364657  
## x2 -0.3902253 0.08076675 -0.274099766 -0.129684350  
## x3 0.4789645 0.09207697 0.576183463 0.426374730  
## x4 -0.1423483 0.25701878 0.190314974 -0.447966480  
## x5 0.1316702 -0.26082897 -0.444159270 -0.008057368  
## x6 -0.3685954 -0.40818080 -0.058415535 0.020205689  
## x7 1.0000000 -0.52486478 -0.188163101 -0.353217378  
## x8 -0.5248648 1.00000000 -0.118674267 -0.030567938  
## x9 -0.1881631 -0.11867427 1.000000000 -0.005521763  
## x10 -0.3532174 -0.03056794 -0.005521763 1.000000000  
##   
## $p.value  
## x1 x2 x3 x4 x5 x6  
## x1 0.00000000 0.07203655 0.585960597 0.26641760 0.96398244 0.01394657  
## x2 0.07203655 0.00000000 0.196637544 0.61611732 0.86232285 0.67266363  
## x3 0.58596060 0.19663754 0.000000000 0.21387990 0.43044741 0.32376478  
## x4 0.26641760 0.61611732 0.213879905 0.00000000 0.18256259 0.13506543  
## x5 0.96398244 0.86232285 0.430447413 0.18256259 0.00000000 0.58756132  
## x6 0.01394657 0.67266363 0.323764782 0.13506543 0.58756132 0.00000000  
## x7 0.19755749 0.08031841 0.028040348 0.53820753 0.56940529 0.10014038  
## x8 0.10439384 0.72782105 0.691398835 0.26071136 0.25346062 0.06621925  
## x9 0.69383958 0.22922192 0.006262173 0.40862160 0.04368247 0.80141499  
## x10 0.05777718 0.57529315 0.053928205 0.04169988 0.97234852 0.93072500  
## x7 x8 x9 x10  
## x1 0.19755749 0.10439384 0.693839577 0.05777718  
## x2 0.08031841 0.72782105 0.229221921 0.57529315  
## x3 0.02804035 0.69139883 0.006262173 0.05392821  
## x4 0.53820753 0.26071136 0.408621600 0.04169988  
## x5 0.56940529 0.25346062 0.043682467 0.97234852  
## x6 0.10014038 0.06621925 0.801414988 0.93072500  
## x7 0.00000000 0.01456685 0.414039087 0.11626349  
## x8 0.01456685 0.00000000 0.608406139 0.89535407  
## x9 0.41403909 0.60840614 0.000000000 0.98104842  
## x10 0.11626349 0.89535407 0.981048424 0.00000000  
##   
## $statistic  
## x1 x2 x3 x4 x5 x6  
## x1 0.00000000 -1.9049515 -0.5541194 1.1450260 0.04575573 2.70799451  
## x2 -1.90495150 0.0000000 1.3381674 0.5097200 0.17578495 -0.42911811  
## x3 -0.55411942 1.3381674 0.0000000 1.2860443 0.80558844 1.01304853  
## x4 1.14502596 0.5097200 1.2860443 0.0000000 1.38346643 -1.56083245  
## x5 0.04575573 0.1757849 0.8055884 1.3834664 0.00000000 0.55173482  
## x6 2.70799451 -0.4291181 1.0130485 -1.5608324 0.55173482 0.00000000  
## x7 1.33529737 -1.8474167 2.3783050 -0.6268656 0.57897770 -1.72836454  
## x8 1.70551451 0.3532080 0.4030665 1.1592627 -1.17769295 -1.94897218  
## x9 -0.39969387 -1.2423538 3.0728776 0.8450079 -2.16089071 -0.25506297  
## x10 2.01941528 -0.5700952 2.0546474 -2.1840388 -0.03512240 0.08809254  
## x7 x8 x9 x10  
## x1 1.3352974 1.7055145 -0.39969387 2.01941528  
## x2 -1.8474167 0.3532080 -1.24235376 -0.57009524  
## x3 2.3783050 0.4030665 3.07287757 2.05464740  
## x4 -0.6268656 1.1592627 0.84500788 -2.18403875  
## x5 0.5789777 -1.1776930 -2.16089071 -0.03512240  
## x6 -1.7283645 -1.9489722 -0.25506297 0.08809254  
## x7 0.0000000 -2.6878186 -0.83510068 -1.64571985  
## x8 -2.6878186 0.0000000 -0.52097071 -0.13330485  
## x9 -0.8351007 -0.5209707 0.00000000 -0.02406917  
## x10 -1.6457199 -0.1333048 -0.02406917 0.00000000  
##   
## $n  
## [1] 29  
##   
## $gp  
## [1] 8  
##   
## $method  
## [1] "pearson"

### Fields in the result

**estimate** => matrix of the partial correlation coefficient between two variables.

**p.value** => a matrix of the p values of the correlation coefficients.

**statistic** => a matrix of the value of the test statistics (not sure what this means).

**n** => sample size.

## Conclusions

We see moderate to strong partial correlation estimates[[1]](#footnote-1) between variables in the "estimate" field which also have significant p-values (given a 0.05 significance level), such as the correlation coefficient between x6 and x1 being around 0.5277 and having a p-value of around 0.01395. Hence, we can conclude that there is moderate to strong multicollinearity in the model.

# SEARCHING FOR CAUSE OF MULTICOLLINEARITY

## Approach

We need the VIF (variance inflation factor i.e. the factor by which a variable’s regression coefficient has inflated from the constant variance of the error term) of each regressor. As a thumb rule, if the VIF of a regressor is greater than 10, it indicates that the particular regressor is the cause of the multicollinearity in the model.

## Library needed

# install.packages("car")  
library(car) # Contains VIF calculation function.

## Loading required package: carData

## Warning: package 'carData' was built under R version 3.6.2

## Applying the VIF function on the linear regression model

vif(model)

## x1 x2 x3 x4 x5 x6 x7 x8   
## 10.342818 8.857510 3.097699 6.585016 2.115003 11.024608 6.006303 6.787283   
## x9 x10   
## 2.254842 7.401606

## Conclusions

We see that x1 and x6 have VIF values above 10, suggesting that the model is multicollinear and that the presence of both of these variables together is the cause of the multicollinearity in the model (we may remove one of them as a remedial measure).

1. Since we are checking correlation based on a sample, the correlation coefficients are estimates. [↑](#footnote-ref-1)