Lesson 12: Multicollinearity & Other Regression Pitfalls

bas, magí

2023-05-25

Uncorrelated predictors

Effect of perfectly uncorrelated predictor variables

This exercise reviews the benefits of having perfectly uncorrelated predictor variables. The results of this exercise demonstrate a strong argument for conducting "*designed experiments*" in which the researcher sets the levels of the predictor variables in advance, as opposed to conducting an "*observational study*" in which the researcher merely observes the levels of the predictor variables as they happen. Unfortunately, many regression analyses are conducted on observational data rather than experimental data, limiting the strength of the conclusions that can be drawn from the data. As this exercise demonstrates, you should conduct an experiment, whenever possible, not an observational study. Use the (contrived) data stored in the Uncorrelated Predictor data set to complete this lab exercise.

1. Using the Stat » Basic Statistics » Correlation... command in Minitab, calculate the correlation coefficient between x1 and x2. Are the two variables perfectly uncorrelated?

```
data <- data.frame(
  x1 = c(-1, -1, -1, -1, -1, 0, 0, 0, 0, 1, 1, 1, 1, 1),
  x2 = c(1, 2, 3, 4, 5, 7, 8, 9, 10, 1, 2, 3, 4, 5),
  y = c(91, 107, 101, 121, 95, 84, 108, 102, 98, 73, 75, 102, 94, 113)
)

data
```

```
##    x1 x2   y
## 1  -1  1  91
## 2  -1  2 107
## 3  -1  3 101
## 4  -1  4 121
## 5  -1  5  95
## 6   0  7  84
## 7   0  8 108
## 8   0  9 102
## 9   0 10  98
## 10  1  1  73
## 11  1  2  75
## 12  1  3 102
## 13  1  4  94
## 14  1  5 113
```

```
correlacion= cor(data$x1, data$x2)

print(correlacion)
```

```
## [1] 0
```

2. **Fit the simple linear regression model with y as the response and as the single predictor:**

```
model1= lm(y ~ x1, data= data)
summary(model1)
```

```
##
## Call:
## lm(formula = y ~ x1, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.629 -11.229   1.471   8.921  21.371
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   97.429      3.501   27.83 2.86e-12 ***
## x1            -5.800      4.142   -1.40    0.187
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.1 on 12 degrees of freedom
## Multiple R-squared:  0.1404, Adjusted R-squared:  0.0688
## F-statistic: 1.961 on 1 and 12 DF,  p-value: 0.1868
```

- **What is the value of the estimated slope coefficient b1?**

  b1= -5.800

- **What is the regression sum of squares, SSR (x1), when x1 is the only predictor in the model?**

  To calculate SSR(x1), we need to calculate the predicted values ŷi using the formula:

  ŷi = b0 + b1 * xi

```
# Calculate mean of y (ȳ)
y_mean= mean(data$y)

# Calculate predicted values (ŷi)
b0= 97.429
b1= -5.800
predicted= b0 + b1 * data$x1

# Calculate SSR(x1)
SSR_x1= sum((predicted - y_mean)^2)

# Print SSR(x1)
print(SSR_x1)
```

```
## [1] 336.4
```

3. **Now, fit the simple linear regression model with y as the response and x2 as the single predictor:**

- **What is the value of the estimated slope coefficient b2?**
- **What is the regression sum of squares, SSR (X2), when x2 is the only predictor in the model?**

```r
model2= lm(y ~ x2, data)
summary(model2)
```

```
##
## Call:
## lm(formula = y ~ x2, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -19.570  -5.862  -1.512   6.509  24.349
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   91.210      6.874  13.270 1.57e-08 ***
## x2             1.360      1.280   1.063    0.309
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 13.51 on 12 degrees of freedom
## Multiple R-squared:  0.08607,    Adjusted R-squared:  0.00991
## F-statistic:  1.13 on 1 and 12 DF,  p-value: 0.3087
```

b2= 1.360

```r
# Calculate predicted values (ŷi)
b0= 91.21
b2= 1.36
predicted= b0 + b2 * data$x2

# Calculate SSR(x2)
SSR_x2= sum((predicted - y_mean)^2)

# Print SSR(x1)
print(SSR_x2)
```

```
## [1] 206.0983
```

4. **Now, fit the multiple linear regression model with y as the response and x1 as the first predictor and x2 as the second predictor:**

- **What is the value of the estimated slope coefficient b1? Is the estimate b1 different than that obtained when x1 was the only predictor in the model?**
- **What is the value of the estimated slope coefficient b2? Is the estimate different than that obtained when x2 was the only predictor in the model?**
- **What is the sequential sum of squares, SSR (x1|x2)? Does the reduction in the error sum of squares when x_2 is added to the model depend on whether x1 is already in the model?**

```r
model3= lm(y ~ x1 + x2, data)
summary(model3)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -16.7321  -8.4513  -0.7718   6.9288  20.7885
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   91.210      6.605  13.810 2.71e-08 ***
## x1            -5.800      4.104  -1.413    0.185
## x2             1.360      1.229   1.106    0.292
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.98 on 11 degrees of freedom
## Multiple R-squared:  0.2265, Adjusted R-squared:  0.08587
## F-statistic: 1.611 on 2 and 11 DF,  p-value: 0.2435
```

b1= -5.800; it's the same has model 1

b2= 1.360: also it's the same

SSE (Suma de Cuadrados del Error) se refiere a la suma de los cuadrados de los residuos en un modelo de regresión. Representa la cantidad total de variabilidad no explicada por el modelo y se calcula como la suma de los residuos al cuadrado.

SSR (Suma de Cuadrados de la Regresión) se refiere a la suma de los cuadrados de las diferencias entre los valores ajustados por el modelo y la media de la variable dependiente. Representa la cantidad total de variabilidad explicada por el modelo y se calcula como la diferencia entre la suma total de cuadrados (SST) y la suma de cuadrados del error (SSE).

SST (Sum of Squeares Total) = SSE + SSR

Para calcular SSR(x2|x1), necesitamos comparar el SSE (Suma de Cuadrados del Error) del modelo que incluye solo x1 con el SSE del modelo completo que incluye tanto x1 como x2.

```
# SSR(x2|x1)
# Modelo con solo x1
SSE_x1= sum(model1$residuals^2)

# Modelo completo con x1 y x2
SSE_x1_x2= sum(model3$residuals^2)

# SSR(x2|x1)
SSR_x2_given_x1= SSE_x1 - SSE_x1_x2
SSR_x2_given_x1
```

```
## [1] 206.176
```

SSR(x2|x1) = 206.176 = SSR_x2, so this doesn't depend on whether x1 is already in the model.

5. **Now, fit the multiple linear regression model with y as the response and x2 as the first predictor, and x1 as the second predictor:**

- **What is the sequential sum of squares, SSR (x1|x2)? Does the reduction in the error sum of squares when is added to the model depend on whether is already in the model?**

```
model4= lm(y ~ x2 + x1, data)
summary(model4)
```

```
##
```

```
## Call:
## lm(formula = y ~ x2 + x1, data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -16.7321  -8.4513  -0.7718   6.9288  20.7885
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   91.210      6.605  13.810 2.71e-08 ***
## x2             1.360      1.229   1.106    0.292
## x1            -5.800      4.104  -1.413    0.185
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 12.98 on 11 degrees of freedom
## Multiple R-squared:  0.2265, Adjusted R-squared:  0.08587
## F-statistic: 1.611 on 2 and 11 DF,  p-value: 0.2435
```

```
# SSR(x1|x2)
# Modelo con solo x1
SSE_x2= sum(model2$residuals^2)

# Modelo completo con x1 y x2
SSE_x2_x1= sum(model4$residuals^2)

# SSR(x2|x1)
SSR_x1_given_x2= SSE_x2 - SSE_x2_x1
SSR_x1_given_x2
```

```
## [1] 336.4
```

SSR(x1|x2) = 206.176 = SSR_x1, so this doesn't depend on whether x2 is already in the model.

6. **When the predictor variables are perfectly uncorrelated, is it possible to quantify the effect a predictor has on the response without regard to the other predictors?**

Yes, when the predictor variables are perfectly uncorrelated, it is possible to quantify the effect of a predictor on the response without regard to the other predictors. In this case, the predictors are orthogonal to each other, meaning they do not share any linear relationship.

When the predictors are orthogonal, the regression coefficients can be interpreted as the individual effects of each predictor on the response variable, holding all other predictors constant. The coefficient for a predictor represents the change in the response variable for a one-unit change in that predictor, assuming all other predictors remain constant.

This is possible because when the predictors are uncorrelated, there is no shared variance or collinearity among them. Each predictor provides unique information and contributes independently to the prediction of the response variable.

However, it's important to note that if the predictors are correlated or have a multicollinearity issue, the interpretation of the individual effects becomes more challenging. In such cases, the coefficients can be influenced by the presence of other predictors, and their individual effects may not be accurately estimated or interpreted without considering the other predictors.

7. **In what way does this exercise demonstrate the benefits of conducting a designed experiment rather than an observational study?**

This exercise show us the efect of a predictor agains the response. Control over predictor variables: In a designed experiment, the researcher has control over the values of the predictor variables. This allows for systematic manipulation and control of the variables of interest. In contrast, in an observational study, the researcher does not have control over the predictor variables as they occur naturally. In this exercise, by manipulating the values of x1 and x2, the researcher can assess their individual effects on the response variable y.

Correlated predictors

Effects of correlated predictor variables

This exercise reviews the impacts of multicollinearity on various aspects of regression analyses. The Allen Cognitive Level (ACL) test is designed to quantify one's cognitive abilities. David and Riley (1990) investigated the relationship of the ACL test to the level of psychopathology in a set of 69 patients from a general hospital psychiatry unit. The Allen Test data set contains the response y = ACL and three potential predictors:

- **x1 = *Vocab*, scores on the vocabulary component of the Shipley Institute of Living Scale**
- **x2 = *Abstract*, scores on the abstraction component of the Shipley Institute of Living Scale**
- **x3 = *SDMT*, scores on the Symbol-Digit Modalities Test**

1. **Determine the pairwise correlations among the predictor variables to get an idea of the extent to which the predictor variables are (pairwise) correlated.**

```
allen= read.csv('allentest.csv')
allen= subset(allen, select = -X)
head(allen)
```

```
##   Subj ACL SDMT Vocab Abstract
## 1    1 6.0   70    28       36
## 2    2 5.4   49    34       32
## 3    3 4.7   28    19        8
## 4    4 4.8   47    32       28
## 5    5 4.9   29    22        4
## 6    6 4.5   23    24       24
```

```
# Creating a correlation matrix
allen_correlation= cor(allen[c('Vocab' , 'Abstract', 'SDMT')])
print(allen_correlation)
```

```
##              Vocab  Abstract      SDMT
## Vocab    1.0000000 0.6978405 0.5560707
## Abstract 0.6978405 1.0000000 0.5769238
## SDMT     0.5560707 0.5769238 1.0000000
```

- The correlation between "Vocab" and "Abstract" is 0.698, indicating a moderate positive correlation.
- The correlation between "Vocab" and "SDMT" is 0.556, indicating a moderate positive correlation.
- The correlation between "Abstract" and "SDMT" is 0.577, indicating a moderate positive correlation.

We can see positive correlation between the predictor variables, but not very strong correlations.

2. **Fit the simple linear regression model with y = *ACL* as the response and = *Vocab* as the predictor. After fitting your model, request that Minitab predict the response y = *ACL* when x1 = 25.**

```r
allen_model= lm(ACL ~ Vocab, allen)
summary(allen_model)
```

```
##
## Call:
## lm(formula = ACL ~ Vocab, data = allen)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4490 -0.6021 -0.1723  0.7106  1.5999
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   4.2253     0.3524  11.989   <2e-16 ***
## Vocab         0.0298     0.0141   2.113   0.0383 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7761 on 67 degrees of freedom
## Multiple R-squared:  0.0625, Adjusted R-squared:  0.04851
## F-statistic: 4.467 on 1 and 67 DF,  p-value: 0.03829
```

```r
# Predict the response y = ACL when x1 = 25
prediction_25= predict(allen_model, newdata= data.frame(Vocab = 25))
print(prediction_25)
```

```
##        1
## 4.970253
```

- **What is the value of the estimated slope coefficient b1?**

  $\text{b1} = 0.0298$

- **What is the value of the standard error of b1?**

  $\text{SE(b1)} = 0.0141$

- **What is the regression sum of squares, SSR (x1), when x1 is the only predictor in the model?**

  ```r
  # SSR = R² * SST
  mean_y= mean(allen$ACL)
  SST= sum((allen$ACL - mean_y)^2)
  SSR_x1= 0.0625^2 * SST
  print(SSR_x1)
  ```

  ```
  ## [1] 0.1681624
  ```

- **What is the predicted response of $y = ACL$ when x1 = 25?**

```r
# SSR = R² * SST
mean_y= mean(allen$ACL)
SST= sum((allen$ACL - mean_y)^2)
SSR_x1= 0.0625^2 * SST
print(SSR_x1)
```

```
## [1] 0.1681624
```