Multiple Regression

Jan Rovny

In most situations, we are interested in estimating the effect of one or a few predictors on the dependent variable, while controlling for a set of other characterstics. In such as case, we need to run multiple regression.

The multiple regression model is the following:

$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_K x_{iK} + \epsilon_i$

Let's look at a specific example, predicting happiness with religious attendence and income:

```
library(foreign)
ESS<-read.dta("https://jan-rovny.squarespace.com/s/ESS_CZ.dta")
```

```
#religious attendence is coded backwards (low values mean higher attendence), so we turn it around:
ESS$relig<--1*ESS$rlgatnd+7
summary(ESS$relig)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   0.000   1.041   2.000   6.000
```

Let's first specify a simple regression model, testing whether happiness might be a function of religiosity:

```
m1<-lm(ESS$happy~ESS$relig)
summary(m1)
```

```
##
## Call:
## lm(formula = ESS$happy ~ ESS$relig)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.8792 -1.0247  0.1208  1.1208  3.1208
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.87919    0.08126  84.655  < 2e-16 ***
## ESS$relig    0.14554    0.04625   3.147  0.00171 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.831 on 780 degrees of freedom
## Multiple R-squared:  0.01254,    Adjusted R-squared:  0.01127
## F-statistic: 9.902 on 1 and 780 DF,  p-value: 0.001714
```

We can see that there is a non-trivial effect of religiosity on happiness. However, one may question whether income isn't a more relevant predictor of happiness. What if wealthy people were happier, and wealthier people were also more religious? The relationship between religisity and happiness would be spurrious!

To test for this, we include income into our regression equation:

```
m2<-lm(ESS$happy~ESS$relig + ESS$hinctnt)
summary(m2)
```

```
##
## Call:
```

```
## lm(formula = ESS$happy ~ ESS$relig + ESS$hinctnt)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -6.9993 -1.1509  0.1637  1.1749  3.3322
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.18463    0.17714  34.914  < 2e-16 ***
## ESS$relig    0.15729    0.04579   3.435 0.000624 ***
## ESS$hinctnt  0.16292    0.03704   4.399 1.24e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.81 on 779 degrees of freedom
## Multiple R-squared:  0.03647,    Adjusted R-squared:  0.034
## F-statistic: 14.74 on 2 and 779 DF,  p-value: 5.188e-07
```

Note that income is very significant, but religiosity remains significant as well. Also, note that the coefficient on religiosity becomes stronger and more significant! Why is that?

To see what is going on, we may wish to look at the pair-wise relationships between the variables in our model:

```r
cor(ESS$happy,ESS$hinctnt)
```

```
## [1] 0.1479205
```

```r
cor(ESS$happy,ESS$relig)
```

```
## [1] 0.1119641
```

```r
cor(ESS$hinctnt,ESS$relig)
```

```
## [1] -0.05833911
```

- We can conclude the following from the results above:
  - happy is positively related to religiosty and income
  - income is negatively related to religiosity
- This means that our simple regression suffered from omitted variable bias:
  - when we omitted income, religiosity captured both, the effect of religiosity, and the effect of income
  - since the correlation between income and religiosity is negative, the omitted variable biased the coefficient on religiosity downwards
  - thus, without including income, we *underestimate* the effect of religiosity!

We were able to overcome this bias through multiple regression, and the inclusion of both religiosity and income as predictors of happiness.

Comparing coefficients

The natural question to ask at this point is: what matters more for happines, religiosity or income? This means that we want to compare the effect of religiosity on happiness with the effect of income on happiness. We, however, cannot just directly compare the size of the coefficients, because they are a function of the scaling of the predictors. When these are on a diffirent scales, direct comparison is impossible.

One way to overcome this problem is to *standardize* the predictors. We create a standardized predictor by subtracting its mean from it, and dividing by its standard deviation: $x_{std} = \frac{x_i - \bar{x}}{s_x}$

We can do this manually:

```
x<-c(1,2,3,4,5,6,7,8,9,10) #produce a madeup variable x
x_std<-(x-mean(x))/sd(x) #standardize x
summary(x_std)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
## -1.4860 -0.7432  0.0000  0.0000  0.7432  1.4860
```

This centers the predictor around 0, and its standard deviation is 1.

As a shortcut, we can use a package that allows us to produce standardized coefficients – that is regression coefficients based on standardized predictors – directly:

```
library(lm.beta)
lm.beta(m2) #produces standardized coefficients of m2
```

```
##
## Call:
## lm(formula = ESS$happy ~ ESS$relig + ESS$hinctnt)
##
## Standardized Coefficients::
## (Intercept)   ESS$relig ESS$hinctnt
##   0.0000000   0.1210055   0.1549798
```

We can see from the model results that income has a slightly stronger effect on happiness than religiosity.

Alternatively, we can recode all predictors to be on a 0-1 scale. This option allows to directly compare unstandardized coefficients. The problem is that we lose intrinsically meaningful scales.

```
x_c<-(x-min(x))/(abs(min(x))+abs(max(x)))
summary(x_c)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.0000  0.2045  0.4091  0.4091  0.6136  0.8182
```

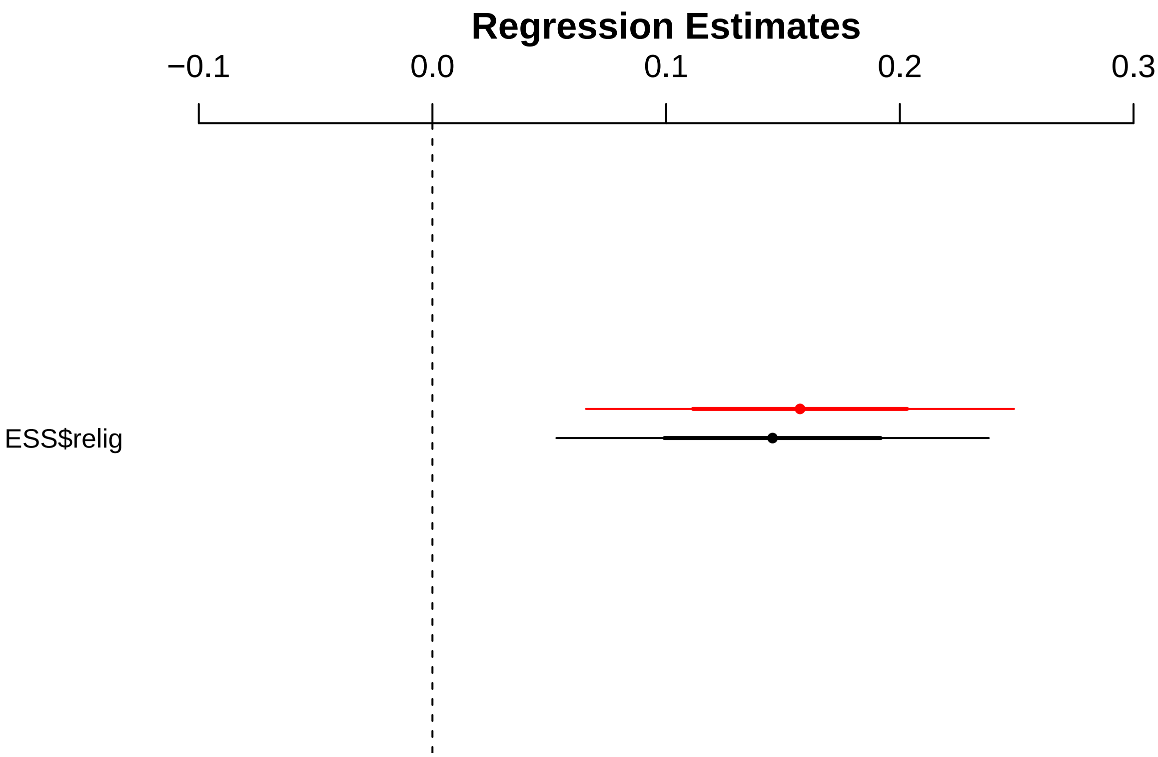
We can then use these recoded predictors in our regression model.

Comparing models

We may be interested in seeing how going from model *m1* to model *m2* above changed the coefficient on religiosity. While we can see the value change, it may be usefulto graph both coefficients in one plot to see the difference:

```
library(arm) #load the necessary library
```

```
coefplot(m1, xlim=c(-0.1,0.3)) #plots the coef of relig from m1
coefplot(m2, add=TRUE, col.pts="red") #this adds the coef of relig from m2 in red, so we can see the (s
```



Comparing diverse models

How do we go about comparing the effectiveness of different models?

When we compare different regression models we differentiate between Nested and Non-Nested models:\

Nested Models

A nested model M2 is nested in another model M1, if M2 is a special case of M1: $M2: y_i = \beta_0 + \beta_1 x_i + \beta_2 z_i + \epsilon_i$ and $M1: y_i = \beta_0 + \beta_1 x_i + \beta_2 z_i + \beta_3 w_i + \epsilon_i$ Here it is clear that M2 is a special case of M1, since M1 arises when $\beta_3 = 0$ Thus M2 is nested in M1.

Non-nested models are ones that specify different relationships:

$M2: y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 z_i + \epsilon_i$ and $M1: y_i = \gamma_0 + \gamma_1 x_{i1} + \gamma_2 z_i + \epsilon_i$ Here y_i is seen as a function of different predictors. These models are thus non-nested.

Let's first create some madeup data:

```
x<-rnorm(1000,0,1)  #generating random variables x, z, w, r, and e that will serve as random error)
z<-rnorm(1000,2,0.4)
w<-rnorm(1000,-3,1.1)
r<-rnorm(1000,1,1)
e<-rnorm(1000,0,1)
y<-2*x + 0.7*z - 0.3*w + 2*r + e #specifying y as a function of the above variables and error
```

Let's run a small model:

```r
model.small<-lm(y~x+z)
summary(model.small)
```

```
##
## Call:
## lm(formula = y ~ x + z)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.879 -1.418  0.037  1.467  7.471
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.89108    0.36591   7.901 7.28e-15 ***
## x            1.96037    0.06952  28.199  < 2e-16 ***
## z            0.70423    0.18067   3.898 0.000104 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.277 on 997 degrees of freedom
## Multiple R-squared:  0.4475, Adjusted R-squared:  0.4464
## F-statistic: 403.8 on 2 and 997 DF,  p-value: < 2.2e-16
```

The general F-test, printed in the model summary tests whether all predictors (simultaneously) have no effect. Given the size of the p-value, we reject this Ho, and conclude that the predictors have non-zero effect.

Next, we run a larger model

```r
model.larger<-lm(y~x+z+w)
summary(model.larger)
```

```
##
## Call:
## lm(formula = y ~ x + z + w)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9400 -1.4311  0.0077  1.4864  7.3449
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.02651    0.41192   4.920 1.01e-06 ***
## x            1.94737    0.06894  28.246  < 2e-16 ***
## z            0.69719    0.17902   3.894 0.000105 ***
## w           -0.29099    0.06581  -4.421 1.09e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.256 on 996 degrees of freedom
## Multiple R-squared:  0.4581, Adjusted R-squared:  0.4565
## F-statistic: 280.7 on 3 and 996 DF,  p-value: < 2.2e-16
```

And we teste the Ho: that β_z and β_w are simultaneously 0. Formally, Ho: $\beta_z = \beta_w = 0$

```r
library(car) #load the necessary library
```

```
##
```

```
## Attaching package: 'car'

## The following object is masked from 'package:arm':
##
##     logit
```

```r
linearHypothesis(model.larger, c("z = 0", "w = 0")) #run the test
```

```
## Linear hypothesis test
##
## Hypothesis:
## z = 0
## w = 0
##
## Model 1: restricted model
## Model 2: y ~ x + z + w
##
##   Res.Df    RSS Df Sum of Sq      F   Pr(>F)
## 1    998 5248.5
## 2    996 5070.2  2     178.3 17.512 3.35e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Given the very small size of the p-value on the F-test is super small, we reject Ho. (This is of course obvous to use, since we know that β_z and β_w are NOT 0, given our model specification above.)

Comparing Nested Models

Let's go back to our small model

```r
summary(model.small)
```

```
##
## Call:
## lm(formula = y ~ x + z)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.879 -1.418  0.037  1.467  7.471
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.89108    0.36591   7.901 7.28e-15 ***
## x            1.96037    0.06952  28.199  < 2e-16 ***
## z            0.70423    0.18067   3.898 0.000104 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.277 on 997 degrees of freedom
## Multiple R-squared:  0.4475, Adjusted R-squared:  0.4464
## F-statistic: 403.8 on 2 and 997 DF,  p-value: < 2.2e-16
```

Let's ask whether running a full model, which also includes w and r as predictors significantly improves our explanation of y. We do this by testing whether the effects of w and r are simultaneously zero:

```r
model.full<-lm(y~x+z+w+r)
summary(model.full)
```

```
##
## Call:
## lm(formula = y ~ x + z + w + r)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.7989 -0.7053 -0.0273  0.6956  3.8127
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.15821    0.18692  -0.846    0.398
## x            2.04119    0.03078  66.307   <2e-16 ***
## z            0.74753    0.07985   9.362   <2e-16 ***
## w           -0.31694    0.02936 -10.797   <2e-16 ***
## r            2.03871    0.03219  63.343   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.006 on 995 degrees of freedom
## Multiple R-squared:  0.8923, Adjusted R-squared:  0.8919
## F-statistic:  2062 on 4 and 995 DF,  p-value: < 2.2e-16
```

```r
linear.hypothesis(model.full, c("w = 0", "r = 0")) #hypothesis test
```

```
## Warning: 'linear.hypothesis' is deprecated.
## Use 'linearHypothesis' instead.
## See help("Deprecated") and help("car-deprecated").
```

```
## Linear hypothesis test
##
## Hypothesis:
## w = 0
## r = 0
##
## Model 1: restricted model
## Model 2: y ~ x + z + w + r
##
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    997 5169.7
## 2    995 1007.5  2    4162.2 2055.3 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

We can see that the p-value on the hypothesis test is miniscule. The effect of w and r is not zero. We therefore conclude that adding w and r significantly improves our explanation of y.

Comparing non-nested models

Let's first setup some additional variables:

```r
k<-y*2+rnorm(1000,0,1)
l<-y*-3+rnorm(1000,0,0.2)
```

Now let's produce a new model that is not nested in our original "model.full":

```
model.nonnest<-lm(y~k+l)
summary(model.nonnest)
```

```
##
## Call:
## lm(formula = y ~ k + l)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.199896 -0.047720  0.001341  0.045701  0.252709
##
## Coefficients:
##              Estimate Std. Error  t value Pr(>|t|)
## (Intercept) -0.001399   0.003675   -0.381    0.704
## k            0.011204   0.002111    5.307 1.37e-07 ***
## l           -0.325698   0.001430 -227.755  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.06721 on 997 degrees of freedom
## Multiple R-squared:  0.9995, Adjusted R-squared:  0.9995
## F-statistic: 1.035e+06 on 2 and 997 DF,  p-value: < 2.2e-16
```

Let us test whether this new non-nested model predicts y better than our full model from above. We do this by considering the Akaike's or Baysian Information Criteria (AIC, BIC). These criteria are defined in the following way: $AIC_i = -2l_i + 2K_i$ BIC is very similar to AIC, but it penalizes more severely models with 'too many' predictors. $BIC_i = -2l_i + 2K_i \ln n$

Here i denotes a particular model, l_i is the log-likelihood function of a particular model and K_i is the number of estimated parameters of a given model.

AIC and BIC are in a sense a combined measure of fit and $parsimony$. The model with the *smallest* AIC or BIC is preferred.

In R:

```
AIC(model.nonnest)
```

```
## [1] -2556.954
```

```
AIC(model.full)
```

```
## [1] 2857.341
```

```
library(stats4) #load requisite library for BIC
```

```
BIC(model.nonnest)
```

```
## [1] -2537.323
```

```
BIC(model.full)
```

```
## [1] 2886.788
```

Since the non-nested model has smaller AIC and BIC, it provides a better prediction for y.