Lsn 29

Clark

Admin

Today we're going to have another 'one-off' lesson. The ideas we'll talk about you may have seen in previous MA206 classes or not, either way, I'm going to give *one* way to think about it, it may or may not be the *best* way to think about variable selection.

In either case, the first step is going to be the same. We have to start with a reserach question. For instance, we might want to know whether taking Omega-3 supplements impact fatty acid levels in the blood.

Here we have $n = 1812$ observations where we have age, sex, race, BMI, Type of supplement (3 levels), dosage, duration, blood level before supplement and blood level after supplement.

Let's write out the full statistical model

We can examine the pairs plot:

```r
omega.dat<-read.csv("Walkerdisc.csv")
omega.dat <- omega.dat %>% dplyr::select(-"pre_O3I",-"post_O3I")
omega.dat %>% ggpairs()
```

Anything jump out?

Let's say we fit this model

```r
omega.lm<-lm(chg_O3I~.,data=omega.dat)
summary(omega.lm)
```

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

```
## lm(formula = chg_O3I ~ ., data = omega.dat)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.054239 -0.009169 -0.000154  0.009772  0.079188
##
## Coefficients: (1 not defined because of singularities)
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)           1.797e-03  4.988e-03   0.360 0.718706
## StudyCarney 2009      3.175e-03  2.746e-03   1.156 0.247914
## StudyDewell 2011     -1.570e-03  3.227e-03  -0.487 0.626701
## StudyFlock 2012       3.414e-03  2.725e-03   1.253 0.210504
## StudyGidding 2014     2.529e-03  4.252e-03   0.595 0.552048
## StudyGrenon 2013     -4.739e-03  3.236e-03  -1.465 0.143315
## StudyHarris 2007     -3.106e-03  6.170e-03  -0.503 0.614718
## StudyHarris 2016     -1.127e-04  2.202e-03  -0.051 0.959188
## StudyHendengren 2015 -1.410e-03  2.392e-03  -0.590 0.555573
## StudyKwong 2016      -2.359e-03  2.259e-03  -1.044 0.296602
## StudyLarson 2008      1.328e-02  7.058e-03   1.881 0.060157 .
## StudyNewman 2014      2.788e-05  6.164e-03   0.005 0.996392
## StudySarter 2015     -9.153e-03  3.790e-03  -2.415 0.015893 *
## StudyShearer 2012     8.191e-03  3.081e-03   2.658 0.007955 **
## StudySkulas Ray 2010  3.376e-03  2.883e-03   1.171 0.241853
## Age                   7.881e-05  4.254e-05   1.853 0.064173 .
## SexM                 -2.119e-03  1.114e-03  -1.902 0.057423 .
## RaceBlack             5.439e-03  4.258e-03   1.277 0.201678
## RaceHispanic          3.388e-03  4.729e-03   0.716 0.473865
## RaceOther             1.280e-02  6.602e-03   1.940 0.052661 .
## RaceWhite             6.391e-03  3.229e-03   1.980 0.047981 *
## BMI                  -3.740e-04  1.042e-04  -3.590 0.000344 ***
## TypeS_EE              1.023e-02  2.035e-03   5.027 5.76e-07 ***
## TypeS_TG              2.029e-02  2.548e-03   7.962 3.95e-15 ***
## Dose                  9.129e-06  7.902e-07  11.552  < 2e-16 ***
## Duration                     NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01624 on 1183 degrees of freedom
## Multiple R-squared:  0.5791, Adjusted R-squared:  0.5706
## F-statistic: 67.82 on 24 and 1183 DF,  p-value: < 2.2e-16
```

What's going on here?

```
omega.dat %>% ggplot(aes(x=Study,y=Duration))+geom_boxplot()
```

Here we've got an issue…

Let's remove study and try again

```
omega.mod<-omega.dat %>% dplyr::select(-Study)
omega.mod.lm<-lm(chg_O3I~.,data=omega.mod)
summary(omega.mod.lm)
```

```
##
## Call:
## lm(formula = chg_O3I ~ ., data = omega.mod)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.054452 -0.009649 -0.000663  0.009797  0.076773
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.795e-03  4.571e-03   1.268   0.2051
## Age           3.286e-06  3.019e-05   0.109   0.9134
## SexM         -2.437e-03  1.041e-03  -2.342   0.0193 *
## RaceBlack     4.044e-03  4.147e-03   0.975   0.3296
## RaceHispanic  1.749e-03  4.722e-03   0.370   0.7111
## RaceOther     1.061e-02  6.576e-03   1.614   0.1068
## RaceWhite     6.413e-03  3.153e-03   2.034   0.0422 *
## BMI          -3.061e-04  9.675e-05  -3.164   0.0016 **
## TypeS_EE      1.076e-02  1.609e-03   6.687 3.48e-11 ***
## TypeS_TG      1.907e-02  1.675e-03  11.383  < 2e-16 ***
## Dose          9.020e-06  5.344e-07  16.878  < 2e-16 ***
## Duration     -8.440e-05  8.658e-05  -0.975   0.3299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01638 on 1196 degrees of freedom
## Multiple R-squared:  0.5675, Adjusted R-squared:  0.5635
## F-statistic: 142.6 on 11 and 1196 DF,  p-value: < 2.2e-16
```

Now we can fit the model, but is it the *best* model? In general we want the most **parsimonious** statistical model. So how do we fix this? Well, one way is through a step-wise procedure. Our book says one way is to either do forward or backward stepwise selection eliminating the variable with the largest P value first, then refitting.

So here we would eliminating `Age` and try again.

```
omega.again<-omega.mod %>% dplyr::select(-Age)
another.lm<-lm(chg_O3I~.,data=omega.again)
summary(another.lm)
```

```
##
## Call:
## lm(formula = chg_O3I ~ ., data = omega.again)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.054425 -0.009618 -0.000687  0.009833  0.076755
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   5.976e-03  4.256e-03   1.404  0.16050
## SexM         -2.427e-03  1.036e-03  -2.342  0.01933 *
## RaceBlack     4.091e-03  4.123e-03   0.992  0.32136
## RaceHispanic  1.781e-03  4.712e-03   0.378  0.70552
## RaceOther     1.062e-02  6.573e-03   1.616  0.10635
## RaceWhite     6.464e-03  3.117e-03   2.074  0.03830 *
## BMI          -3.071e-04  9.629e-05  -3.189  0.00146 **
## TypeS_EE      1.076e-02  1.609e-03   6.689 3.43e-11 ***
## TypeS_TG      1.904e-02  1.650e-03  11.536  < 2e-16 ***
## Dose          9.018e-06  5.338e-07  16.892  < 2e-16 ***
## Duration     -8.557e-05  8.587e-05  -0.996  0.31922
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01637 on 1197 degrees of freedom
## Multiple R-squared:  0.5675, Adjusted R-squared:  0.5638
## F-statistic:   157 on 10 and 1197 DF,  p-value: < 2.2e-16
```

Why do the P values change here?

Another common technique is to eliminate based on *Akaike's Information Criterion.* The Akaike information criterion (AIC) is an estimator of out-of-sample prediction error and is calculated from:

In R it is implimented through the `MASS` library and allows you to do forward, backward, or both direction stepping

```
step.mod<-stepAIC(omega.mod.lm,direction="both")
```

So, according to AIC, the final model is:

To see how well our model fits the data we want to test it on some data that we **did not use to build our model**. This is called cross-validation

```
validation.data<-read.csv("Walkervalid.csv")
our.predictions<-predict.lm(step.mod,validation.data)
```

To see how well our data fit we can calculate the Mean Square Predicition Error, or MSPE

```
sum((our.predictions-validation.data$chg_O3I)^2)
```

```
## [1] 0.1611435
```

To see what would have happened had we not done any variable selection we could calculate, from the full model

```
full.predictions<-predict.lm(omega.mod.lm,validation.data)
sum((full.predictions-validation.data$chg_O3I)^2)
```

```
## [1] 0.1630276
```

I always like to compare to the naive predictor or just predicting the mean for everything

```
naive.pred<-mean(omega.dat$chg_O3I)
sum((naive.pred-validation.data$chg_O3I)^2)
```

```
## [1] 0.3976803
```

So both models improve over the null model but the reduced model is better than the full model for prediction.

Note that this is not the only way of doing this. If prediction is our only goal, then why stop here? We can consider all sorts of different modesl that make no intuitive sense.

For instance, there's a type of regression called lasso that works in the following way:

In R we do:

```
library(glmnet)
x_vars <- model.matrix(omega.mod.lm)[,-1]
y_var <- omega.mod$chg_O3I
lambda_seq <- 10^seq(1, -10, by = -.01)

cv_output <- cv.glmnet(x_vars, y_var,
            alpha = 1, lambda = lambda_seq)

# identifying best lamda
best_lam <- cv_output$lambda.min
```

Then we do:

```
lasso_best <- glmnet(x_vars, y_var, alpha = 1, lambda = best_lam)
test.data<-validation.data %>% dplyr::select(-"pre_O3I",-"post_O3I",-"Study")
x.test<-model.matrix(lm(chg_O3I~.,data=test.data))[,-1]
pred <- predict(lasso_best, s = best_lam, newx = x.test)
sum((pred-validation.data$chg_O3I)^2)
```

```
## [1] 0.1619458
```

Which isn't as good as the stepwise, but there's also Neural Networks, Random Forest, Bayesian methods, etc. Once we are no longer exploring a mechanism of interest, but rather building a predictive model, we can really start to be creative.