Statistical Consulting Group

San Diego State University

Linear Regression in R: Abalone Dataset

This tutorial will perform linear regression on a deceptively simple dataset. The abalone dataset from UCI Machine Learning Arvhives comes with the goal of attempting to predict abalone age (through the number of rings on the shell) given various descriptive attributes of the abalone (Shell sizes, weights of whole abalone and parts of shucked abalone). In following this goal, we will attempt to predict rings using the the shell sizes (height, length, width), and weights (shell weight, shucked weight, viscera weight, whole weight). The problem associated with this dataset is that these descriptive attributes are all heavily correlated.

First we need to load the data.

```
aburl = 'http://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.d
abnames = c('sex','length','diameter','height','weight.w','weight.s','weight.v','we
abalone = read.table(aburl, header = F , sep = ',', col.names = abnames)
```

Then we load the libraries we'll be using.

```
library(MASS)
library(rms)
```

Step 1: Verify Data Integrity

The first step of analysis is to examine the data integrity. We can't draw conclusions on heavily suspect data.

```
summary(abalone)
```

We can tell right off that there exists a problem with the abalone height, where some values are registered as 0's. This is not possible. We will need to investigate, so we take a look at those values.

```
abalone[abalone$height==0,]
```

We see two abalone for which the height is obviously not 0, but is not properly recorded. We will have

to set these to null values, and exclude these observations from the analysis (if height turns out to be significant)

```
abalone$height[abalone$height==0] = NA
```

The minimum weights are also a bit low compared to other measurements, so We should take a look at them.

```
abalone[abalone$weight.w < .01,]
```

It seems these abalone are legitimately really small, so this is probably not a data entry error. Thus we cannot exclude it. Now let's examine the structure of the data. Specifically, we're looking at pairwise correlation.

```
as.matrix(cor(na.omit(abalone[,-1])))
```

As we can see, the data is heavily correlated. This is a problem if we attempt analysis off these raw numbers. We'll use various methods later on to try and reduce the apparent correlation.

This is only a preliminary approach to verifying data integrity. We'll be getting into more detail later after we do some fits.

Step 2: Initial Fit

```
abfit1 = lm(rings ~ sex + length + diameter + height + weight.w
               + weight.s + weight.v + weight.sh, data = abalone)
abfit2 = stepAIC(abfit1)
summary(abfit2)
```

The sex being used as the baseline in this case is female. Since male is not significantly different, and infant is the only difference, we can redefine this feature to define whether gender has been expressed (infant vs non-infant)

```
abalone$sex = as.character(abalone$sex)
abalone$sex[abalone$sex != 'I'] = 'K'
abalone$sex = as.factor(abalone$sex)
```

A bit worrying is that the AIC is picking as significant all four of the weight measures, despite that they should be linear functions of eachother.

Whole Weight = Shucked Weight + Viscera Weight + Shell Weight + Unknown mass of water/blood lost from shucking process

We should investigate this abalone weight problem, and ensure that the variables in the model are not functions of eachother.

```
abalone$weight.diff = abalone$weight.w -
      (abalone$weight.v + abalone$weight.s + abalone$weight.sh)
par(mfrow=c(1,1))
hist(abalone$weight.diff,breaks=50)
```

It would appear that there are some instances where the whole weight is less than the sum of the components, which does not stand up to logic. We will examine the worst offenders of these.

```
abalone[abalone$weight.diff < -.1,]
```

This is somewhat frustrating, as we know these are wrong. In cases where the shucked weight is greater than the whole weight, we might postulate that the person entering the data got them backwards. Because of this lack of faith in the measurements, and realizing that they should be a linear function of eachother, we should stick to only using one or two of the weight measurements. We'll pick which weight measure to use.

```
abfit2.1 = lm(rings ~ sex + diameter + height + weight.w, data = abalone)
abfit2.2 = lm(rings ~ sex + diameter + height + weight.s, data = abalone) # !
abfit2.3 = lm(rings ~ sex + diameter + height + weight.v, data = abalone)
abfit2.4 = lm(rings ~ sex + diameter + height + weight.sh, data = abalone)
```

Now let's examine the best fit model to try and identify any potential outliers. The best model uses the shucked weight as a predictor.

```
par(mfrow=c(2,2))
plot(abfit2.2)
```

We see observation 2052 as a potential outlier. We shall investigate to see why that is.

```
abalone[2052,]
summary(abalone[abalone$sex=='K',])
```

For comparison, we also print out summary information for known genders. length, diameter, and weight say this abalone is slightly small, but otherwise unremarkable. Height on the other hand would seem to indicate that this abalone is exceptional. When we look at the height, it would seem that there was a data entry error. 0.130 height seems believable, 1.130 does not for an abalone that is otherwise small. We will change the entered height.

```
abalone$height[2052] = 0.130
```

Also, we'll try picking some functions of the various weights to see if we can get a more compact result. First, the geometric mean of the sub-weights.

```
abalone$weight.mean1 = (abalone$weight.s*abalone$weight.v*abalone$weight.sh)^(1/3)
abalone$weight.mean2 = (abalone$weight.w*abalone$weight.s*abalone$weight.sh*abalone
abalone$weight.norm1 = sqrt(abalone$weight.s^2 + abalone$weight.v^2 + abalone$weigh
abalone$weight.norm2 = sqrt(abalone$weight.w^2 + abalone$weight.s^2 + abalone$weigh
```

We'll also define some other measures of size that incorporate all 3 dimensions. This allows us to

describe 3 dimensions with 1 (hopefully). We pick the euclidean norm of its size, along with the geometric mean of size. We'll pick one of these measures to use as well.

```
abalone$size.norm = sqrt(abalone$length^2 + abalone$diameter^2 + abalone$height^2)
abalone$size.mean = (abalone$length*abalone$diameter*abalone$height)^(1/3)
```

Because we're looking at many transformations of the variables, and we won't accept certain variables in the model at the same time, we have to make comparisons manually.

```
abfit3 = lm(rings ~ sex + size.norm + weight.w + weight.s + weight.v + weight.sh, d

abfit3.1 = lm(rings ~ sex + size.norm + weight.w, data = abalone)
abfit3.2 = lm(rings ~ sex + size.norm + weight.s, data = abalone)
abfit3.3 = lm(rings ~ sex + size.norm + weight.v, data = abalone)
abfit3.4 = lm(rings ~ sex + size.norm + weight.sh, data = abalone) # best

abfit4.1 = lm(rings ~ sex + size.norm + weight.norm1, data = abalone)
abfit4.2 = lm(rings ~ sex + size.norm + weight.norm2, data = abalone)
abfit4.3 = lm(rings ~ sex + size.norm + weight.mean1, data = abalone)
abfit4.4 = lm(rings ~ sex + size.norm + weight.mean2, data = abalone)

abfit5.1 = lm(rings ~ sex + size.mean + weight.norm1, data = abalone)
abfit5.2 = lm(rings ~ sex + size.mean + weight.norm2, data = abalone)
abfit5.3 = lm(rings ~ sex + size.mean + weight.mean1, data = abalone)
abfit5.4 = lm(rings ~ sex + size.mean + weight.mean2, data = abalone)

abfit6 = lm(rings ~ sex + size.mean + weight.w + weight.s + weight.v + weight.sh, d
abfit6.1 = lm(rings ~ sex + size.mean + weight.s + weight.v + weight.sh, data = aba
abfit6.2 = lm(rings ~ sex + size.mean + weight.s + weight.sh, data = abalone)
abfit6.3 = lm(rings ~ sex + size.mean + I(size.mean^2) + weight.s + weight.sh, data
```

We will want to visually check the fit and validity of our model.

```
par(mfrow=c(2,2))
plot(abfit6.3)
```

Thus far, we've been approaching the dataset with a very linear method in mind. We have pretty much exhausted our options dealing with this problem without changing the dependent variable. Thus far we have been unsucessful in getting the QQ plot line to merge to the diagonal.

When we look at the residuals vs fitted values plot, we see a fan shape indicating that as the fitted values increase, so to do the scale of the residuals. One way to account for that is to use a log transformation of the dependent variable. Subsequent analysis will take log(rings) as the dependent variable to account for this.

```
abfit7 = lm(log(rings) ~ sex + size.mean + weight.s + weight.sh, data = abalone)
plot(abfit7)
```

We're closing in on a final model. Here at least we have two separate measures of weight that don't measure the same thing.

```
vif(abfit7)
```

The VIF is a bit worrying, because a VIF higher than 5 is cause for concern. + This is a highly correllated data set. There are methods of dealing with multicollinearity within a data set which involve declaring new variables as linear combinations of existing variables. One of these methods is called “Principle Components Analysis”. We won’t be using it here, though. We’re going to do the best we can by reasoning through the model.

```
abfit7.1 = lm(log(rings) ~ sex + size.mean + I(size.mean^2) + weight.s + weight.sh,
abfit7.2 = lm(log(rings) ~ sex + log(size.mean) + weight.s + weight.sh, data = abal
abfit7.3 = lm(log(rings) ~ sex + log(size.mean) + weight.s*weight.sh, data = abalon
abfit7.4 = lm(log(rings) ~ sex + log(size.mean) + weight.w, data = abalone)
```

In going through these models, it becomes clear that 7.2 and 7.3 offer the best visually fitting model. Performing an anova on 7.2 and 7.3 results in the additional interaction term in 7.3 not being significant, so we don’t include it. We check the VIF for 7.2, and find all elements with a VIF under 6. While that is high, it’s the best we can do with this dataset without bringing in more advanced methods.

As such, our resulting model is 7.2

$$\log(\text{rings}_i) = \beta_0 + \beta_1(\text{Sex}_i) + \beta_2 \log(\sqrt[3]{\text{l*w*h}_i}) + \beta_3(\text{Shell}_i) + \beta_4(\text{Shucked}_i)$$

This entry was posted in R, Tutorials and tagged MLR, Model Selection, R, Regression on April 16, 2014 [http://scg.sdsu.edu/linear-regression-in-r-abalone-dataset/] .