Module 5 Lab

This lab revisits the husband and wife dataset from *OpenIntro Statistics*, which we used for the Module 2 Lab. This lab covers the following topics.

- Manually perform a Sum of Squares F-test
- Use `anova()` to perform a Sum of Squares F-test
- Use model selection criteria functions `AIC()` and `BIC()` to compare models
- Explore the usefulness of R^2 for model comparison
- Create confidence and prediction intervals

Multiple Linear Regression

This time, after loading the data, we remove the rows containing `NA` values. This will simplify model comparison later in the lab.

```
HWData <- read.csv("Age.csv", row.name = 1) # Husband and Wife Data
HWData <- na.omit(HWData) # Remove rows with NAs
head(HWData)
```

```
##   HAge HHght WAge WHght HAgeMar
## 1   49  1809   43  1590      25
## 2   25  1841   28  1560      19
## 3   40  1659   30  1620      38
## 4   52  1779   57  1540      26
## 5   58  1616   52  1420      30
## 6   32  1695   27  1660      23
```

In Lab 3 we used a husband's age to predict his wife's age. This time, let's add `HAgeMar`, the age of the husband when the husband and wife were married, as a predictor. Then our model is:

$$WAge_i = \beta_0 + \beta_1 HAge_i + \beta_2 HAgeMar_i + \epsilon_i,$$

where it is assumed that $\epsilon_i \overset{ind}{\sim} N(0, \sigma^2)$.

As usual, plot `WAge` against each of the predictors to get a sense of the relationship between the predictors and the response. Also take a moment to look at the `HAgeMar` against `HAge` plot, to consider multicollinearity.

```
library(ggplot2)
qplot(HAge, WAge, data = HWData)
qplot(HAgeMar, WAge, data = HWData)
qplot(HAgeMar, HAge, data = HWData)
```

The relationship between the `HAgeMar` and `WAge` is not as strong as the relationship `HAge` and `WAge`, but they appear related. Notice that `HAgeMar` and `HAge` show a moderately positive correlation, which could slightly inflate the variance our estimators.

Now we fit both models–with and without `HAgeMar` as a predictor, and take a look at the output for the new model.

```
fit1 <- lm(WAge ~ HAge, data = HWData) # Regress WAge against HAge
fit2 <- lm(WAge ~ HAge + HAgeMar, data = HWData)  # Add HAgeMar
```

Before proceeding to inference, it is a good idea to look at the residuals. To review, we are looking for signs that our model assumptions–linearity, Normality, independence, constant variance–have been violated. If they have, it could show up as non-random patterns in the residual plots. Therefore, we will plot the residuals against each of the explanatory to check for indications of non-linearity; and residuals against fitted values to check for indications of non-constant variance. You should keep an eye out for non-independence and non-Normality in all of the residual plots.

```
qplot(fitted(fit2), residuals(fit2))
qplot(HAge, residuals(fit2), data = HWData)
qplot(HAgeMar, residuals(fit2), data = HWData)
```

The plots do not raise any red flags, so we proceed to inference. Let's start with the summary of our two model fits.

```
summary(fit1)
summary(fit2)
```

The output includes a t-test result indicating the `HAgeMar` coefficient is non-zero. What about the new model, `fit2`, as a whole? Suppose we want to compare the new *model* to the old *model*. This is a very useful capability because often models differ beyond the addition of a single predictor, in which case a t-test on one predictor is not sufficient. A Sum of Squares F-test allows us to compare any two models, call them the simple model and a more complex model, where the simple model is "nested" in the complex model. Nested means that the complex model has all predictors the simple model has, plus additional predictors. Put another way, the predictors in the simple model are a subset of the predictors in the complex model. Note that this is the only setting where we can use an F-test.

Sum of Squares F-test

Basically, we want to compare the proportion of total variability explained by each model, and ask if the increase in explained variability is sufficient to justify adding the additional term/s into the model. Equivalently, is the *decrease* in *unexplained* variability sufficient to justify the more complex model? The F-statistic contains the information we need, and its distribution has an answer.

$$F = \frac{(RSS_1 - RSS_2)/(df_1 - df_2)}{RSS_2/df_2} \sim F_{df_1 - df_2, df_2}$$

"RSS" stands for Residual Sum of Squares, "df" stands for degrees of freedom, and the subscripts denote model 1 and model 2. The following code calculates the necessary pieces, and then performs a Sum of Squares F-test.

```
rss1 <- deviance(fit1) # Model 1 RSS
rss2 <- deviance(fit2) # Model 2 RSS
df1 <- df.residual(fit1) # Model 1 Residual Degrees of Freedom
df2 <- df.residual(fit2) # Model 2 Residual Degrees of Freedom
fstat <- ( (rss1 - rss2)/(df1 - df2) )/(rss2/df2) # F-statistic
1 - pf(fstat, df1 - df2, df2) # p-value
```

```
## [1] 2.161366e-05
```

The first two lines extract the residuals sum of squares from each model fit. The next two lines extract the residual degrees of freedom from each model. The next line calculates the F-statistic, and the final line calculates the p-value. To review, the p-value is the probability of seeing an F-statstic this unlikely, or more unlikely, if the null hypothesis (simple model) is true. What do we conclude?

The extra term is worth it! We confidently draw this conclusion based on a very small p-value. You can probably guess, there is an easier way to peform the same test.

```
anova(fit1, fit2) # Sum of Squares F-test
```

```
## Analysis of Variance Table
##
## Model 1: WAge ~ HAge
## Model 2: WAge ~ HAge + HAgeMar
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    167 2617.6
## 2    166 2347.2  1    270.34 19.119 2.161e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The analysis of variance table contains all the information we just calculated. Conveniently, the two models being compared are provided, followed by the table. The columns give the residual degrees of freedom, residual sum of squares, difference in degrees of freedom (numerator degrees of freedom), difference in sum of squares between the two models, F-statistic, and p-value.

What other ways can we compare two or more models?

Model Comparison Criteria

An analyst may want additional model comparison criteria. Two tools in the toolbox are Akaike information criterion (AIC), and Bayesian information criterion (BIC). The details of these measures are beyond the scope of this lab, but suffice it to say that the lower the score, the better the fit (**discussion of "better" below). A strength of AIC and BIC, in contrast to the F-test, is that not only can we compare more than two models simultaneously, but we can compare models that are not nested. With this in mind, let's add a third model with `HAgeMar` as the only predictor.

```
fit3 <- lm(WAge ~ HAgeMar, data = HWData)  # Only HAgeMar
AIC(fit1, fit2, fit3)
BIC(fit1, fit2, fit3)
```

Both criteria agree with the F-test: based on the information gained about the response variable, `HAgeMar` is worth adding to a model with only `HAge` as a predictor. We know this because `fit2` has a lower AIC and BIC score than `fit1`. The newest model, `fit3`, is inferior to both `fit2` and `fit1`, as evidenced by a much higher AIC and BIC. Keep in mind that lower indicates a **better fit, but only within each criteria; AIC values should not be compared to BIC values (and vice versa).

**What does "better" mean to AIC and BIC? There is not a simple answer, but there are a few ideas worth mentioning. AIC and BIC place value on model "parsimony," in the Occam's Razor sense of the word. If two models explain variation in the response equally well, then AIC and BIC tend to prefer the simpler model (with fewer terms). AIC and BIC implement this preference by penalizing a model for extra predictors. This preference aids interpretation; the greater the number of terms in a model, typically the more difficult it is to understand and interpret the meaning of those terms. For example, imagine interpreting a marginally useful squared three-way interaction term! This makes AIC and BIC somewhat ill suited to model selection for prediction. If prediction is your ultimate goal, and that squared three-way interaction term improves prediction, then so be it!

R^2 - Coefficient of Determination

What about R^2 as a model comparison metric? R^2 is sometimes called the "Coefficient of Determination," and described as the proportion of total variation in the response explained by the model.

Revisiting the two models we fit, R^2 did increase with the additional predictor.

```
summary(fit1) # Multiple R-squared:  0.8803
summary(fit2) # Multiple R-squared:  0.8927
```

Does this mean `fit2` is superior to `fit1`? To explore this answer, consider adding another explanatory variable, but this time a completely meaningless one. We will generate Normal random variables with the same mean as the response variable `WAge`, and a standard deviation of one. What will happen to R^2 if we include this predictor in the model?

```
set.seed(101513)
X4 <- rnorm(169, mean(HWData$WAge), 1)
fit4 <- lm(WAge ~ HAge + HAgeMar + X4, data = HWData)
summary(fit4) # Multiple R-squared:  0.894
```

R^2 increased! Did we really *explain* more of the variation in our data? Of course not. In fact, R^2 will never decrease when you add more terms; it can only increase. AIC and BIC, on the other hand, take into consideration the number of terms included in a model.

```
AIC(fit2, fit4)
```

```
##      df      AIC
## fit2  4 932.2572
## fit4  5 932.1094
```

```
BIC(fit2, fit4)
```

```
##      df      BIC
## fit2  4 944.7768
## fit4  5 947.7589
```

Analysts generally consider AIC or BIC scores within four units of one another equivalent. So, although technically the AIC decreased and BIC increased, they both essentially stayed the same. Other things being equal then, the simpler model is preferred, so we should stick with `fit2`. This is an example of why selecting the "best" model for a dataset can be tricky, and goal specific, business.

Prediction and Confidence Intervals

Confidence intervals for the model coefficients, and for the response variable at given values of the predictors, are straightforward to obtain.

```
confint(fit2)
predict(fit2, interval = "confidence")
```

What if we want prediction and/or confidence intervals at new explanatory variable values? The procedure is similar to the one we used for SLR.

```
new2 <- data.frame(HAge = c(30, 35, 40), HAgeMar = c(22, 24, 26))
predict(fit2, newdata = new2, interval = "confidence")
predict(fit2, newdata = new2, interval = "prediction")
```

The first line creates a data frame with new values of `HAge` and `HAgeMar`, and the next two calculate confidence and prediction intervals. Do you remember why prediction intervals are so much wider?

What about model three, with the randomly generated predictor `X3`? Will the intervals be affected by the meaningless predictor?

```
confint(fit3)
predict(fit3, interval = "confidence")

new3 <- data.frame(HAge = c(30, 35, 40),
                   HAgeMar = c(22, 24, 26),
                   X3 <- rnorm(3, mean(HWData$WAge), 1))
predict(fit3, newdata = new3, interval = "confidence")
predict(fit3, newdata = new3, interval = "prediction")
```

Notice that `new3` required an extra column compared to `new2`, because `fit3` has an extra predictor compared to `fit2`. What effect did the randomly generated predictor have on our intervals?