Inference for Regression

Math 145

Fish and mercury

Does the amount of mercury found in fish depend on the pH level of the lake? Fish were captured and pH measured in a number of Florida lakes. We can use this data to explore this question.

1. Type the following command and use the results to determine what a case is in this study.

   ```
   library(Lock5withR)
   head(FloridaLakes)
   ```

2. The following commands will (a) plot the data and the least squares regression line, and (b) show us the coefficients (slope and intercept) of the regression line. Does a line look like a reasonable fit for this data?

   ```
   gf_point(AvgMercury ~ pH, data = FloridaLakes) %>%
     gf_lm()
   lm(AvgMercury ~ pH, data = FloridaLakes )
   # terser output -- just the coefficients
   coef(lm( AvgMercury ~ pH, data = FloridaLakes))
   lm( AvgMercury ~ pH, data = FloridaLakes) %>% coef()  # alternative
   ```

3. Write an equation for the least squares regression line.

4. If the pH is 6, what does the least squares regression line predict for the average amount of mercury in fish?

5. R can automate this for you. Give this a try. Did you get the same answer when you did it by hand?

   ```
   fish_model <- lm(AvgMercury ~ pH, data=FloridaLakes)
   coef(fish_model)
   est_AvgMercury <- makeFun(fish_model)
   est_AvgMercury(6)
   est_AvgMercury(8)
   ```

Confidence Intervals

So how good are the estimates for the slope and intercept? How precise is our estimated average mercury when the pH of a lake is 6? We would like have interval estimates rather than just point estimates. As with our other intervals, we have two ways to do this: bootstrap and using SE-formulas.

Bootstrap

Big Idea: Simply resample the data and compute the least squares fit over and over to see how much the slope varies.

6. Create a bootstrap distribution using `do()`, `lm()`, and `resample()`.

   R is clever: When we use `do()` with `lm()` it automatically keeps track of the coefficients for us. You will have to look at the bootstrap data to see what things are called. (Use `head()` to look at just the first few rows.)

7. Now use your bootstrap distribution to compute 95% confidence intervals for the slope and intercept. Which of this is more scientifically interesting? Why?

8. Here is another way to look at the bootstrap distribution. This will plot a line for each combination of slope and intercept in our bootstrap distribution.

```
gf_point(AvgMercury ~ pH, data = FloridaLakes) %>%
  gf_abline(slope = ~pH, intercept = ~ Intercept, data = Lakes_boot, alpha = 0.05)
```

   Things to notice:

   - Most of the slopes (95%) should lie within our 95% confidence interval for the slope.
   - We can be more certain about predictions from our linear regression model near the center of the data than near the edges.

Using SE formulas

The `msummary()` function summarizes a linear model in a way that includes standard errors for the slope and intercept. (The formulas for these standard errors are a bit complicated, but simple for a computer to calculate with.) The degrees of freedom we need are listed as well. (It should be two less than the number of observations for a simple linear model.)

9. Try this.

```
msummary(fish_model)    # or summary(fish_model)
```

10. Now compute a confidence interval for the slope use the standard error and degrees of freedom reported. (How will you compute t_*?) How does this interval compare to the bootstrap interval from before?

11. Are you feeling really lazy? Try this:

```
confint(fish_model)
```

P-values

We can compute p-values as well, either using the SE formulas or by creating a randomization distribution.

12. Usually the only hypothesis of interest is that the slope is 0. What does it mean for the slope to be 0?

13. The p-value for this test is listed in the output of `msummary()`. Do you see it?

14. You can calculate it by hand using SE. Do it and check that it matches.

Or we could create a randomization distribution. If we shuffle one of the two variables (I like to shuffle the explanatory variable), then we will simulate a world where the slope is 0 because the relationship between predictor and response will just be due the randomness of how our shuffling happens to match them up.

15. Create a randomization distribution and use it to test the null hypothesis that the slope is 0.

ANOVA for regression and the correlation coefficient

We can think about regression as a way to analyze the variability in the response. Some of the variability is explained by the predictor (average mercury is different for high pH lakes vs. low pH lakes). Some of the variability is not explained by the predictor (even among lakes with similar pH, some have higher average mercury levels and other lower levels.)

The ANOVA table for regression is similar to the ANOVA table for comparing multiple means.

16. To get the ANOVA table, type one of these:

```r
anova(fish_model)
anova(lm(AvgMercury ~ pH, data = FloridaLakes))
```

This is a lot like the ANOVA tables we have seen before. This time:

$$
\begin{aligned}
SST &= \sum (y - \overline{y})^2 \\
SSE &= \sum (y - \hat{y})^2 \\
SSM &= \sum (\hat{y} - \overline{y})^2 \\
SST &= SSM + SSE
\end{aligned}
$$

- y is the observed response.
- $\hat{y}$ is the predicted response from the linear regression. (For comparing multiple means, $\hat{y}$ was the group mean.)
- $\overline{y}$ is the mean response.
- The degrees of freedom for the simple linear regression model is 1, leaving $n - 1$ degrees of freedom for the residuals.

As before, when SSM is large and SSE is small, then the model ($\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$) explains a lot of the variability and little is left unexplained (SSE). On the other hand, if SSM is small and SSE is large, then the model explains only a little of the variability and most of it is due to things not explained by the model.

By some fancy algebra, it can be shown that the percentage of variability that is explained by the regression model is the square of the correlation coefficient: denoted r^2 or R^2:

$$
R^2 = \frac{SSM}{SST} = \frac{SSM}{SSM + SSE}
$$

For our the Florida lakes study, we see that

- $SSM = 2.00$
- $SSE = 4.05$
- $SST = 2.00 + 4.05 = 6.05$
- $R^2 = \frac{SSM}{SST} = \frac{2}{6.05} = 0.3305785$

  This number is listed as “Multiple R-squared" on the summary output.

So pH explains roughly 1/3 of the variability in average mercury levels. The other two thirds of the variability in mercury levels is due to other things. (We can think of many things that might matter: size of the lake, depth of the lake, types of fish in the lake, types of plants in the lake, proximity to industrialization – highways, streets, manufacturing plants, etc.) More complex studies might investigate the effects of several of these simultaneously.

17. Now create a randomization distribution to test the null hypothesis that the correlation coefficient is 0. (Use `cor()` to calculate the correlation coefficient. These two tests mean the same thing because the slope is 0 exactly when the correlation coefficient is 0. Recall that slope $= r\frac{s_y}{s_x}$.)

Anscombe’s famous examples

18. Compute the regression summary tables for Anscombe’s four pairs of variables. What do you notice?

```r
msummary(lm( y1 ~ x1, anscombe))
msummary(lm( y2 ~ x2, anscombe))
msummary(lm( y3 ~ x3, anscombe))
msummary(lm( y4 ~ x4, anscombe))
```

19. Now make scatter plots for each and add the regression line (with `gf_lm()`). Now what do you notice?
20. Why do you think Anscombe create these data sets? What was his main point?

Residuals

Recall that a residual is the difference between the observed response and the value predicted by the regression line.

$$\text{residual} = y - \hat{y} = \text{observed} - \text{predicted}$$

Looking at residuals can be a good way to see if the model is working well. Residuals should

- be approximately normally distributed.
- demonstrate roughly the same amount of variability across the data set.
- subject to those to things, be basically "random". (Patterns in residuals are generally an indication of a problem.)

21. Create these plots. How do things look for our fish model?

```r
gf_histogram(~ resid(fish_model))
gf_qq(~ resid(fish_model)) %>%
  gf_qqline()
gf_point(resid(fish_model) ~ pH, data = FloridaLakes)
mplot(fish_model, which = 1:2)     # or plot(fish_model, which = 1:2)
```

Remember: A normal-quantile plot is a plot where the dots will fall nearly along a line when the data come from a normal distribution. It is basically a comparison (via a scatter plot) of the values we have and the values we would have expected from a normal distribution. This plot works better than a histogram for detecting normality issues because our eyes are good at detecting lines and because histograms don't work very well with small data sets.

22. Now do the same thing for the Anscombe data sets. Here is the code for the first pair.

```r
model1 <- lm(y1 ~ x1, data = anscombe)
gf_histogram( ~ resid(model1))
gf_point(resid(model1) ~ x1, data = anscombe)
mplot(model1, which = 1:2)
```