

Lab 8: Regression analysis

Due with your final writeup: 1 Question, 1 Bonus

In this lab, we will examine the data on mercury contamination considered in lecture. You can think of it as a prelude to your final, which will largely concern regression analysis. Perhaps it is more precise to say that the final, with its focus on regression, will let us explore the big themes addressed during the quarter: Graphical and numerical summaries, formulating and testing hypotheses, sampling variability and confidence intervals. To get us started, we will spend this Lab session learning about how R implements the simple linear model.

Getting Started

As usual, we begin this lab by loading data into your R session. This time, however, our data are not stored in R's special format, but rather as comma-separated values (CSV). Programs like Excel, for example, can read and write data in this format. After weeks of the usual routine, we've decided to publish lab materials this way so that you see how to bring your own data into R.

The data on fish contamination is linked to the course home page, or is available directly at

```
http://www.stat.ucla.edu/~cocteau/stat13/data/fish.csv
```

In either case, use a browser to download the file, and save it in the “working directory” for your R session. The command

```
getwd()
```

should return the folder where R will look for files to load. Save `fish.csv` there.

(We have not talked about this much, but when you start using R outside of this class, you will inevitably want to save your work in different folders, giving each new project its own folder. In general, you can specify a new working directory with the command `setwd`, which takes the location of your project's folder as an argument.)

Once you have stored `fish.csv` in your current working directory, you can read the data into R with the following command.

```
fish = read.csv("fish.csv")
```

R has a number of functions for handling data with different formats. The function `read.csv` is specially designed for files of comma separated values. After reading the contents of the file, the command above saves the data in `fish`.

You can also read in the file without downloading it. Instead you could create a `url` from the address of the file.

```
fish = read.csv(  
  url("http://www.stat.ucla.edu/~cocteau/stat13/data/fish.csv"))
```

Whether you go the `url` route or you download the file to your computer, the following command

```
ls()
```

will give you a list of objects that should include the dataset `fish`. Recall from lecture that these data were collected to study the extent of mercury concentration in two rivers in North Carolina. Researchers considered only largemouth bass; and each row of this dataset refers to a fish taken from one of two rivers, the Lumber or the Waccamaw.

```
dim(fish)
```

You will see that the data consists of 171 rows (`fish`) and 5 columns. Data for the first five fish can be shown with the command

```
fish[1:5,]
```

which should produce the following output.

	river	station	length	weight	mercury
1	0	0	47.0	1616	1.60
2	0	0	48.7	1862	1.50
3	0	0	55.7	2855	1.70
4	0	0	45.2	1199	0.73
5	0	0	44.7	1320	0.56

As we have noted several times, the first column in this output is just a row index, printed by R and not part of the dataset. The remaining five columns in this printout refer to data contained in `fish`. The data are described as follows:

`river` The river in which the fish was caught;
0 for the Lumber River, 1 for the Waccamaw

`station` The station along the river at which the fish was caught;
stations are labeled 0, 1, 2, ..., 15

`length` The length of the fish in cm

`weight` The weight of the fish in g

`mercury` The mercury content of a “filet” extracted from
the fish, given in parts-per-million

Introduction to regression

For the moment, we will focus primarily on the last three variables, `length`, `weight` and `mercury`. A scatterplot of two variables can be made with the command

```
plot(fish$length, fish$weight)
```

or, by using column numbers instead,

```
plot(fish[,3], fish[,4])
```

All possible pairs of scatterplots of these three variables can be created in one go

```
pairs(fish[,3:5])
```

where we have asked for the third through the fifth columns. Let’s create a new dataset that consists of just the 98 fish taken from the Waccamaw river. (We will return to the Lumber shortly.)

```
waccamaw = subset(fish, fish$river==1)
dim(waccamaw)
```

You should now have 98 rows. Recall the relationship between length and mercury content for these fish.

```
plot(waccamaw$length, waccamaw$mercury)
```

We observed in class that the data have a positive trend in the sense that longer fish tend to have higher mercury content. The relationship is somewhat noisy, however. Still, we decided to posit a population model of the form

$$(\text{mercury}) = \beta_0 + \beta_1(\text{length}) + (\text{error})$$

that is, the mercury content of a fish is described by linear model. Given a fish's length, its mercury level is determined by a point on the line plus some error; a fish's length is multiplied by the slope (β_1), we add the intercept β_0 to the result, and finally an error term is introduced.

In class we examined the use of least squares as a way of estimating the slope and intercept of the population model from a sample of fish. In R, this is done with the command

```
fit = lm(mercury~length,data=waccamaw)
summary(fit)
```

Here, the command `lm` refers to a "linear model." The expression inside the parentheses, `mercury~length`, is a formula in R. It specifies a statistical model, in this case the one given above: that mercury content depends linearly on a fish's length. Notice that this expression is missing any reference to the population parameters or the error. To make this specification clearer, if we wanted a model of the form

$$(\text{mercury}) = \beta_0 + \beta_1(\text{length}) + \beta_2(\text{weight}) + (\text{error})$$

our formula would be `mercury~length+weight`. And with this, you get a sense that the linear model is not just about one input and one response; but more on that later.

In the summary output, we find a table of the **Coefficients** and their standard errors. Recall that these come from an analytical expression for the sampling distribution that is not dissimilar in spirit to what we discussed for the t -distribution (consult yesterday's lecture for a list of connections).

The object `fit` represents your fitted linear model. It contains information about

```
coefficients(fit)
```

and

```
residuals(fit)
```

and a number of other derived quantities that we'll cover in our last lecture. You can look at the residuals with

```
r = residuals(fit)
hist(r)
qqnorm(r)
```

You can add the regression line to your scatterplot of mercury content versus weight with the following.

```
plot(waccamaw$length,waccamaw$mercury)
abline(coefficients(fit))
```

At the risk of beating a dead horse, we can bootstrap the regression coefficients in a fairly straightforward way. Again, the bootstrap gives us a quick assessment of the variability in our estimates, this time $\hat{\beta}_0$ and $\hat{\beta}_1$. Below we bootstrap the slope.

```
nsim = 5000
bootbeta1 = rep(0,5000)

for(i in 1:nsim)
{
  brows = sample(1:98,replace=T)
  boots = waccamaw[brows,]

  bfit = lm(mercury~length,data=boots)
  bcoeff = coefficients(bfit)

  bootbeta1[i] = bcoeff[2]
}

sd(bootbeta1)
```

Here, we are sampling row numbers (1 to 98) with replacement and then using those to create a new data set `boots`. We then fit the linear model using the bootstrap data set, pull out the coefficients and keep the second one, the bootstrap replicates

of the slope. The standard deviation of the bootstrap replicates should be close to the standard error in the summary table `summary(fit)`.

Your turn

Finally, let's have a brief look at the data from the Lumber river. As we did for the Waccamaw, we will create a special data set containing just these 73 fish.

```
lumber = subset(fish, fish$river==0)
dim(lumber)
```

Question 1: (a) Make a plot of mercury versus length for these 73 fish. (b) Fit a regression to these samples. What is the coefficient on `length`? What are its units? How does it compare to its (analytical) standard error? (c) Make plots of the residuals; do they look as you expected them to? (Mean zero, normally distributed, etc.)

Bonus: Add the regression line to your scatterplot in (a). Using the coefficient of determination (the Multiple R-squared in `summary(fit)`), how does this fit compare to the one for fish from the Waccamaw river?