Mercury levels in fish tissue for large mouth bass in the Wacamaw and Lumber Rivers

Rivers in North Carolina contain small concentrations of mercury which can accumulate in fish over their lifetimes.

Directly measuring the mercury concentration in the water is impossible since it is almost always below detectable limits.

The concentration of mercury in fish tissue can be obtained at considerable expense by catching fish and sending samples to a lab for analysis.

A study was recently conducted in the Wacamaw and Lumber Rivers to investigate mercury levels in tissues of large mouth bass. At several stations along each river, a group of fish were caught, weighed, and measured. In addition a tissue sample from each fish caught was sent to the lab so that the tissue concentration of mercury could be determined for each fish.

Questions:

1. Is there a relationship between mercury concentration and size (weight and/or length) of a fish?

2. A concentration over 1 part per million (0 on the log scale) is considered unsafe for human consumption. In light of this, what recommendations can you make for fish caught from these rivers?

```
> fishHG = read.table("fishHG.txt", header=TRUE)
> attach(fishHG)
> head(fishHG)
   river station length_cm weight_g HG_conc_ppm
1 lumber      l1      47.0     1616        1.60
2 lumber      l1      48.7     1862        1.50
3 lumber      l1      55.7     2855        1.70
4 lumber      l1      45.2     1199        0.73
5 lumber      l1      44.7     1320        0.56
6 lumber      l1      43.8     1225        0.51
```

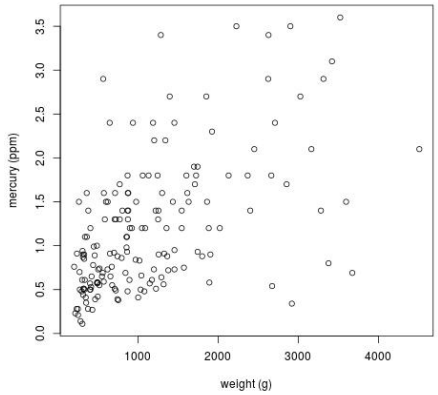
RELATIONSHIPS BETWEEN VARIABLES....

```
> cor(HG_conc_ppm, weight_g)
[1] 0.553838
```

Correlation tells us whether two variables are **linearly related**, and if they are what direction their relationship is in. Positive correlation = positive relationship (one variable goes up, the other tends to go up as well; one variable goes down, the other variable tends to go down as well). Negative correlation = negative relationship (the variables tend to move in the opposite directions).

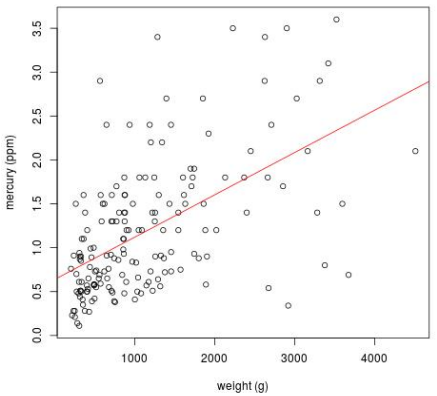
Similar information can be obtained from a scatterplot diagram:

```
> plot(weight_g,HG_conc_ppm,ylab="mercury", xlab="weight")
```



To get the quantification of the relationship we employ MODELS. The first simple model we have is a simple linear regression. This fits a line through the scatterplot of two variables. The line is estimated using the least squares procedure that estimates the intercept and the slope.

```
> Hg.w.lm = lm(HG_conc_ppm ~ weight_g)

> plot(weight_g, HG_conc_ppm, ylab="mercury (ppm)", xlab="weight (g)")

> abline(Hg.w.lm)
```



> summary(Hg.w.lm)

```
Call:
lm(formula = HG_conc_ppm ~ weight_g)
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 6.387e-01  8.035e-02   7.948 2.56e-13 ***
weight_g    4.818e-04  5.572e-05   8.647 3.93e-15 ***

Residual standard error: 0.6361 on 169 degrees of freedom
Multiple R-squared:  0.3067,  Adjusted R-squared:  0.3026
F-statistic: 74.77 on 1 and 169 DF,  p-value: 3.929e-15
```

The estimated intercept is 0.64 and the estimated slope is 0.0005.

Intercept denotes the average mercury level for a fish that has weight of 0. It is meaningless in many situations as you can see.

Slope denotes the average change in mercury when weight goes up by 1 unit (in this case gram). So if you catch two fishes, one weighing x and the other weighing (x+1) grams you'd expect their mercury levels to differ by 0.0005 ppm.

So the estimated model (line) is:

Expected Mercury = 0.64 + 0.0005 weight

Is that relationship STATISTICALLY SIGNIFICANT?

We can answer that question by looking at the p-value of the weight in the above table: it is basically 0.000. This pretty much means that the relationship will be significant at almost any significance level (recall that we usually use 0.05 or 0.1 level).

```
weight_g    4.818e-04  5.572e-05   8.647      3.93e-15 ***
```

Alternatively – look at the 95% confidence interval: it **does not** include 0, which tells us that at 5% level we would reject the null hypothesis of no relationship between mercury and weight. So the mercury-weight relationship IS statistically significant.

```
> confint(Hg.w.lm, level = 0.95)

                    2.5 %       97.5 %
(Intercept) 0.4800552234 0.7973072814
weight_g    0.0003718145 0.0005918011
```

This model will allow us predict how much mercury a fish of any given weight will have on average. (Out of all fishes of that given weight, the average observed mercury level will be equal to the fitted value from the model above).

For example, we can predict (and give CI for) the average mercury level for a fish of 1kg.

```
Hg.w.1kg.fit = predict(Hg.w.lm, newdata = data.frame(weight_g = 1000), interval =
"conf", level = 0.95)
```

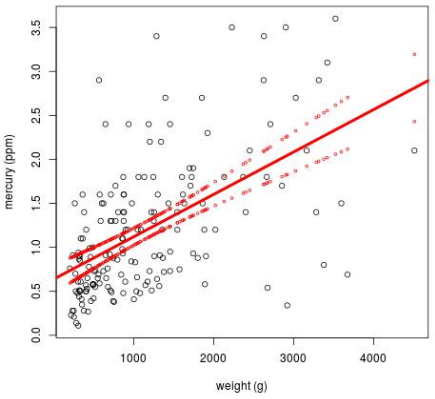
```
> Hg.w.1kg.fit
       fit    lwr      upr
1 1.120489 1.0231 1.217878
```

So the estimated average mercury concentration for a fish of 1 kg is `1.12 ppm`.

The 95% CI for the true average mercury concentration for a fish of 1 kg is `(1.02, 1.22)`

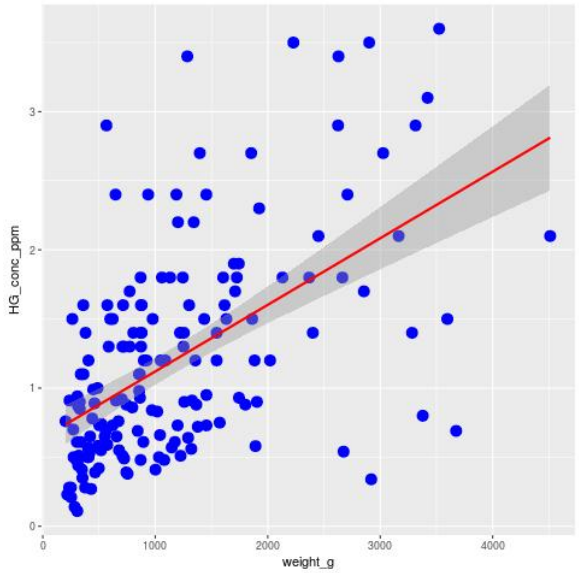
We can do this for every point on the line: we'll find the 95% CI for the true average mercury concentration for every fish we have in the sample:

```
Hg.w.fit = predict(Hg.w.lm, interval = "conf", level = 0.95)

Hg.w.fit.lb = Hg.w.fit[,2]
Hg.w.fit.ub = Hg.w.fit[,3]

plot(weight_g, HG_conc_ppm, ylab="mercury
(ppm)", xlab="weight (g)")

abline(Hg.w.lm, col="red", lw = 4)

points(weight_g, Hg.w.fit.lb, col="red",
cex=0.5)

points(weight_g, Hg.w.fit.ub, col="red",
cex=0.5)
```



There is an easier way to make this graph, using package ggplot2

```
require(ggplot2)
ggplot(fishHG, aes(x=weight_g, y=HG_conc_ppm)) +
  geom_point(color='blue', size = 4) +
  geom_smooth(method=lm, color='red')
```



We can also say something about the actual mercury level of the fish of 1kg based on our model – we can find not only its mean (and the CI for it based on the above), but also its forecast interval: the middle 95% of all mercury weights for all fishes of 1kg. This prediction interval takes into account the uncertainty in the estimation of the mean as well as the actual randomness of mercury weight around the mean.

```
> Hg.w.1kg.forecast = predict(Hg.w.lm, newdata = data.frame(weight_g =
1000), interval = "pred", level = 0.95)

> Hg.w.1kg.forecast
       fit         lwr      upr
1 1.120489 -0.1389147 2.379893
```

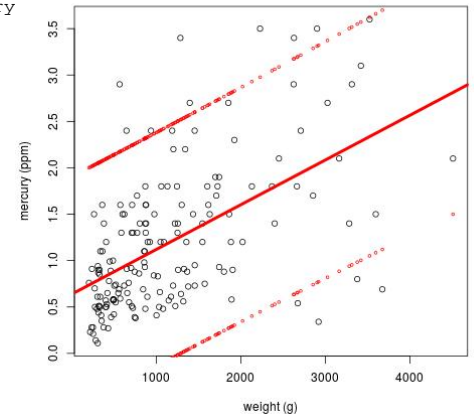
So the forecasted mercury concentration for a fish of 1 kg is still `1.12 ppm`.

The 95% prediction interval for the mercury concentration for a fish of 1 kg is `(-0.14, 2.38)`

Note that this interval includes 0, which is nonsensical – but can happen when using a linear model that doesn't know about natural constraints. You can simply state that this interval is

We can do this for every point on the line: we'll find the 95% prediction interval for the mercury concentration for every fish we have in the sample:

```
Hg.w.forecast = predict(Hg.w.lm, interval = "pred", level = 0.95)

Hg.w.forecast.lb = Hg.w.forecast[,2]
Hg.w.forecast.ub = Hg.w.forecast[,3]

plot(weight_g, HG_conc_ppm, ylab="mercury
(ppm)", xlab="weight (g)")

abline(Hg.w.lm, col="red", lw = 4)

points(weight_g,Hg.w.forecast.lb,
col="red", cex=0.5)

points(weight_g,Hg.w.forecast.ub,
col="red", cex=0.5)
```



These bands are much wider than the CI bands. These bands are also bow-tie shaped, with the skinniest interval at the average weight. However, it is hard to see the bow-tie shape as the variance is so large.

Q. A concentration over 2.5 part per million is certainly considered unsafe for human consumption. In light of this, what recommendations can you make for fish caught from these rivers?

The basic recommendation would be not to eat large fish.

A more specific recommendation could be based on the following reasoning: Limit the probability that a fish will have more than 2.5ppm to 2.5%.

So, we need:

P(Y > 2.5ppm) < 0.025

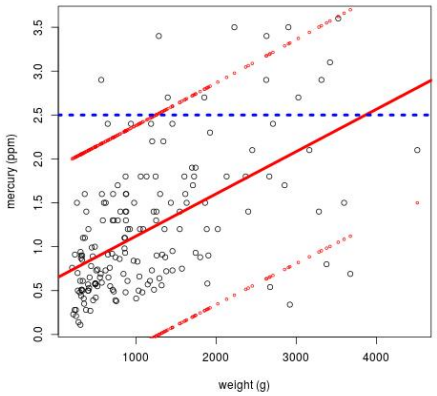
where Y ~ N (0.64 + 0.0005 w*, 0.6361 sqrt(1+1/171+(w*-mean(w))^2/ SSX))

So we'd need to solve for w*

Visually, we'd be looking at the weight that corresponds to the intersection of the blue and top red line

> min(weight_g[which(Hg.w.forecast.ub>2.5)])
[1] 1251

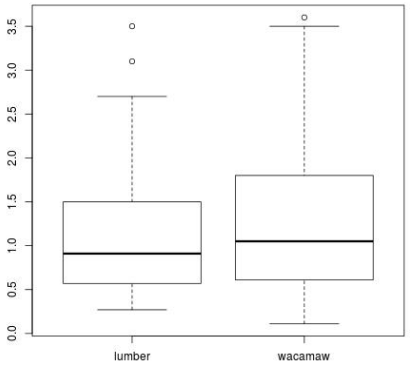
Fish larger than 1251 grams should not be consumed.



Multiple regression
– Is this relationship the same for the two rivers?

Note that if the two rivers are very similar, and have the same fish distribution, we could simply test the average Hg concentration between the rivers:

```
> plot(river,HG_conc_ppm)

> t.test(HG_conc_ppm ~ river, var.equal=TRUE)

    Two Sample t-test

data:  HG_conc_ppm by river

t = -1.6936, df = 169,
p-value = 0.09218

alternative hypothesis: true difference
in means is not equal to 0

95 percent confidence interval:
 -0.42954353  0.03285077

sample estimates:
lumber          wacamaw
1.078082        1.276429
```



This is equivalent to:

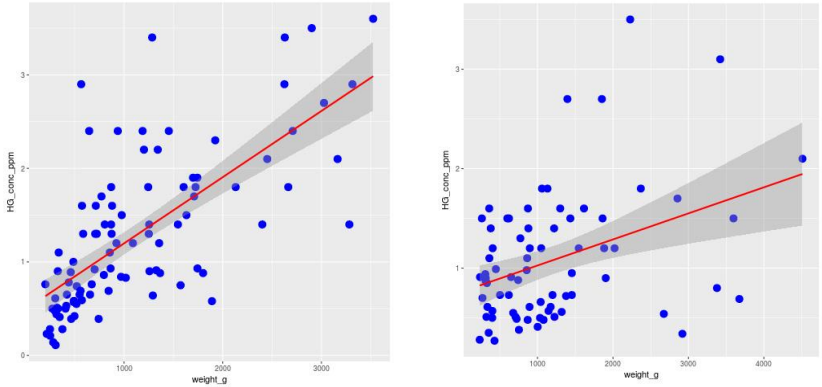
```
> Hg.river.lm = lm(HG_conc_ppm ~ river, data=fishHG)

> summary(Hg.river.lm)

Call:
lm(formula = HG_conc_ppm ~ river, data = fishHG)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept)   1.07808    0.08866  12.160   <2e-16 ***
riverwacamaw  0.19835    0.11712   1.694   0.0922 .
---
Residual standard error: 0.7575 on 169 degrees of freedom
Multiple R-squared:  0.01669, Adjusted R-squared:  0.01087
F-statistic: 2.868 on 1 and 169 DF,  p-value: 0.09218
```

This model implies:

```
Wacamaw (1):  Average Hg = 1.078 + 0.198 = 1.276

Lumber (0): Average Hg = 1.078
```

But we do not know that the fish population is the same in two rivers. Maybe one river is faster and has more eddies, implying that there are more skinny long fish there?

In order to "adjust" our t-test for different sizes of fish, let's first run the regression separately:




```
ggplot(fishHG[river=='lumber',], aes(x=weight_g, y=HG_conc_ppm)) +
geom_point(color='blue', size = 4) +   geom_smooth(method=lm,
color='red')

ggplot(fishHG[river=='wacamaw',], aes(x=weight_g, y=HG_conc_ppm)) +
geom_point(color='blue', size = 4) +   geom_smooth(method=lm,
color='red')
```

```
Hg.lumber.lm = lm(HG_conc_ppm ~ weight_g, fishHG[river=='lumber',])

summary(Hg.lumber.lm)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 7.644e-01  1.147e-01   6.663 4.84e-09 ***
weight_g    2.620e-04  7.547e-05   3.472 0.000884 ***
```

```
Hg.wac.lm = lm(HG_conc_ppm ~ weight_g, fishHG[river=='wacamaw',])

summary(Hg.wac.lm)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 4.895e-01  1.006e-01   4.866 4.46e-06 ***
weight_g    7.081e-04  7.283e-05   9.723 5.86e-16 ***
```

These effects of weight on Hg in fish in two rivers do not appear the same.

How would the graphs above look like if this effect of weight were the same in two rivers?

The lines would have been parallel.

This is what multiple regression would do: it forces the same effect within two rivers:

```
Hg.river.w.lm = lm(HG_conc_ppm ~ river+weight_g, data=fishHG)
summary(Hg.river.w.lm)

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.933e-01  9.852e-02   5.007 1.38e-06 ***
riverwacamaw 2.403e-01  9.699e-02   2.478   0.0142 *
weight_g     4.884e-04  5.495e-05   8.888 9.35e-16 ***
---
Residual standard error: 0.6266 on 168 degrees of freedom
Multiple R-squared:  0.3312,  Adjusted R-squared:  0.3232
F-statistic: 41.59 on 2 and 168 DF,  p-value: 2.119e-15
```

Note the binary river variable gets interpreted as 0/1 variable, with 0 being Lumber, and 1 Wacamaw.

So the model is really as follows:

```
Lumber: Average Hg = 0.5 + 0.0005 weight_g
Wacamaw: Average Hg = 0.5 - 0.24 + 0.0005 weight_g =
                    = 0.26 + 0.0005 weight_g
```

So there are two lines – one for Lumber, one for Wacamaw. They are parallel – they have different intercepts but the same slope.

That common slope is the effect of weight on average Hg. It is assumed to be the same in the two rivers: for any given river, the effect of weight on Hg is the same (0.0005)

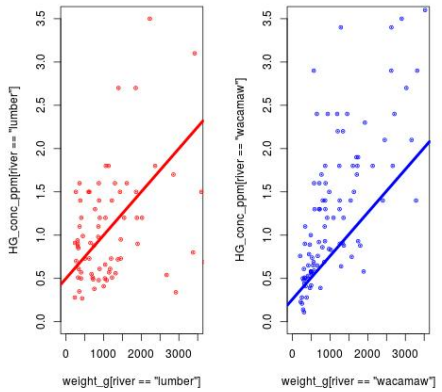
As a result, the estimated effect of weight here (`0.0005`) is a compromise between individual weight coefficients from the separate regressions – `0.00026` and `0.00071`

The remaining river effect then modifies the intercept to pull these lines closer to each river's data.

So now, average Hg for a population of fishes of the same weight is significantly different between two rivers.

Are you happy with this model?

```
plot(weight_g[river=="lumber"],HG_conc_ppm[river=="lumber"],
cex=0.1,col="red",xlim=c(0,3500),ylim=c(0,3.5))
points(weight_g[river=="lumber"],HG_conc_ppm[river=="lumber"],cex=.7,
col="red")
abline(0.5, 0.0005, col="red",lw=4)
```




```
plot(weight_g[river=="wacamaw"],HG_conc_ppm[river=="wacamaw"],
cex=0.1,col="blue",xlim=c(0,3500),ylim=c(0,3.5))
points(weight_g[river=="wacamaw"],HG_conc_ppm[river=="wacamaw"],cex=.7,
col="blue")
abline(0.26, 0.0005, col="blue",lw=4)
```

Maybe a better model would be to have the same intercept but different slopes for each river?

Note that the model with different intercepts and slopes would be two separate regressions.

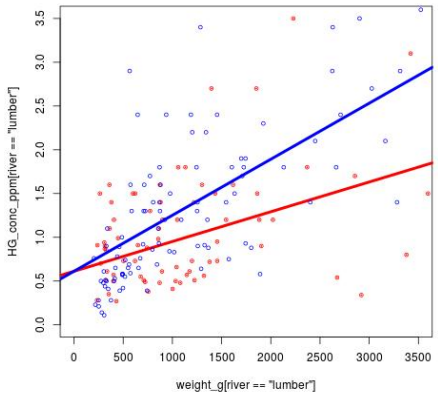
```
> Hg.river.w.int.lm = lm(HG_conc_ppm ~ river:weight_g, data=fishHG)
> summary(Hg.river.w.int.lm)

Call:
lm(formula = HG_conc_ppm ~ river:weight_g, data = fishHG)

Coefficients:
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)            6.118e-01  7.613e-02   8.036 1.56e-13 ***
riverlumber:weight_g   3.411e-04  6.080e-05   5.610 8.19e-08 ***
riverwacamaw:weight_g 6.369e-04  6.242e-05  10.203  < 2e-16 ***

Residual standard error: 0.6009 on 168 degrees of freedom
Multiple R-squared:  0.385,    Adjusted R-squared:  0.3776
F-statistic: 52.58 on 2 and 168 DF,  p-value: < 2.2e-16
```

Now the two models are:

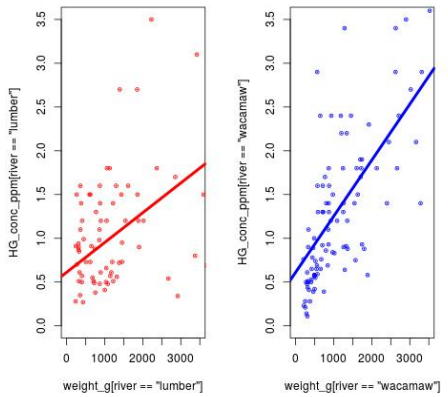
```
Lumber: Average Hg = 0.612 + 0.00034 weight_g
Wacamaw: Average Hg = 0.612 + 0.00064 weight_g
```

Visualy, this model looks like it fits better. Also according to R2

```
plot(weight_g[river=="lumber"],HG_conc_ppm[river=="lumber"],
cex=.1,col="red",xlim=c(0,3500),ylim=c(0,3.5))
points(weight_g[river=="lumber"],HG_conc_ppm[river=="lumber"],cex=.7,
col="red")
abline(0.612, 0.00034, col="red",lw=4)
points(weight_g[river=="wacamaw"],HG_conc_ppm[river=="wacamaw"],cex=.7,
col="blue")
abline(0.612, 0.00064, col="blue",lw=4)
```




```
par(mfrow=c(1,2))
plot(weight_g[river=="lumber"],HG_conc_ppm[river=="lumber"],
cex=.1,col="red",xlim=c(0,3500),ylim=c(0,3.5))
points(weight_g[river=="lumber"],HG_conc_ppm[river=="lumber"],cex=.7,
col="red")
abline(0.612, 0.00034, col="red",lw=4)
plot(weight_g[river=="wacamaw"],HG_conc_ppm[river=="wacamaw"],
cex=.1,col="blue",xlim=c(0,3500),ylim=c(0,3.5))
points(weight_g[river=="wacamaw"],HG_conc_ppm[river=="wacamaw"],cex=.7,
col="blue")
abline(0.612, 0.00064, col="blue",lw=4)
```



Additional models

```
> Hg.river.l.w.lm = lm(HG_conc_ppm ~ river+length_cm+weight_g)
> summary(Hg.river.l.w.lm)


Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.4568077  0.3661228  -3.979 0.000103 ***
riverwacamaw  0.1232408  0.0920110   1.339 0.182256
length_cm     0.0675456  0.0122838   5.499 1.41e-07 ***
weight_g     -0.0001062  0.0001194  -0.889 0.375193


Residual standard error: 0.5783 on 167 degrees of freedom
Multiple R-squared:  0.4337,  Adjusted R-squared:  0.4235
F-statistic: 42.63 on 3 and 167 DF,  p-value: < 2.2e-16
```

We interpret this model as follows:

For a given river, and a given weight of fish, an additional cm in length results in 0.068 ppm increase in average Hg.

Similarly, for a given river, and length of fish, an additional g in weight results in 0.0001 ppm decrease in average Hg.

Note that these effects are the same for every combination of other predictors.

But this model seems to imply that weight is no longer significant, once we account for river and length. Let's look at the model without weight:

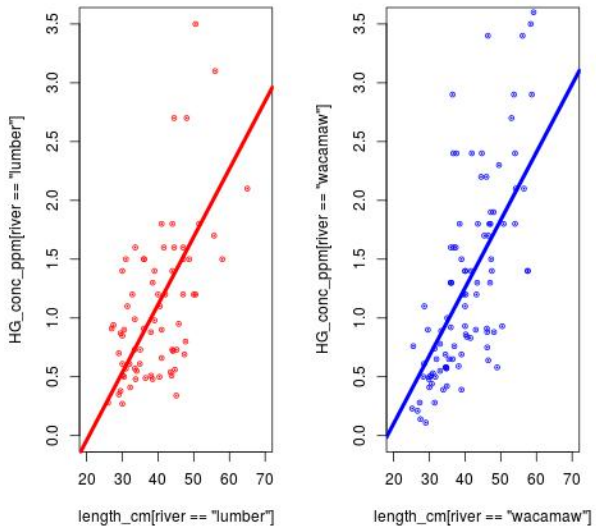
```
> Hg.river.l.lm = lm(HG_conc_ppm ~ river+length_cm, data=fishHG)
> summary(Hg.river.l.lm)


Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.194229   0.216287  -5.521 1.26e-07 ***
riverwacamaw  0.142027   0.089496   1.587    0.114
length_cm     0.057657   0.005213  11.061  < 2e-16 ***

Residual standard error: 0.5779 on 168 degrees of freedom
Multiple R-squared:  0.431,   Adjusted R-squared:  0.4243
F-statistic: 63.63 on 2 and 168 DF,  p-value: < 2.2e-16
```

Now the two models are:

Lumber: Average Hg = -1.194 + 0.0577 weight_g

Wacamaw: Average Hg = -1.052 + 0.0577 length_cm =

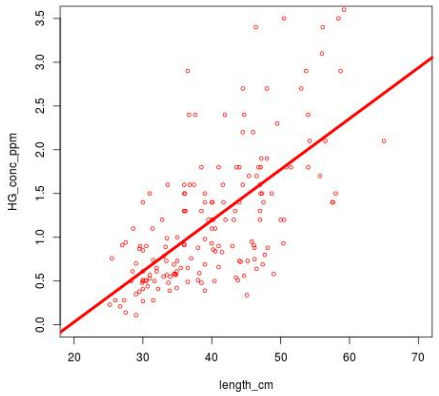
But…is the relationship linear in Wacamaw?



Also, river is no longer significant if length is in the model!

So let's look at the model with length alone:

```
> Hg.l.lm = lm(HG_conc_ppm ~ length_cm, data=fishHG)
> summary(Hg.l.lm)

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.131645   0.213615  -5.298 3.62e-07 ***
length_cm    0.058127   0.005228
11.119  < 2e-16 ***

Residual standard error: 0.5805 on
169 degrees of freedom
Multiple R-squared:  0.4225,
    Adjusted R-squared:  0.4191
F-statistic: 123.6 on 1 and 169 DF,
p-value: < 2.2e-16
```



Ok, so which model is the best? We can simply look at the adjusted R2: highest for the model with river and length, but close second is the model with length only.

Or, we can do this more formally using your favorite model selection criteria, like AIC or BIC – pick the lowest one, or the simplest one among the cluster of similarly low scores:

```
> AIC(Hg.river.w.lm, Hg.river.w.int.lm, Hg.river.l.w.lm, Hg.river.l.lm,
Hg.l.lm)
                  df      AIC
Hg.river.w.lm      4 330.3816
Hg.river.w.int.lm  4 316.0457
Hg.river.l.w.lm    5 303.9256
Hg.river.l.lm      4 302.7332
Hg.l.lm            3 303.2776
```

```
> BIC(Hg.river.w.lm, Hg.river.w.int.lm, Hg.river.l.w.lm, Hg.river.l.lm,
Hg.l.lm)
                  df      BIC
Hg.river.w.lm      4 342.9483
Hg.river.w.int.lm  4 328.6124
Hg.river.l.w.lm    5 319.6339
Hg.river.l.lm      4 315.2999
Hg.l.lm            3 312.7026
```

So given that the model with river+length, and length alone are so similar, and given that for policy purposes we'd like to have the simplest model possible, we'll choose the model with length only.

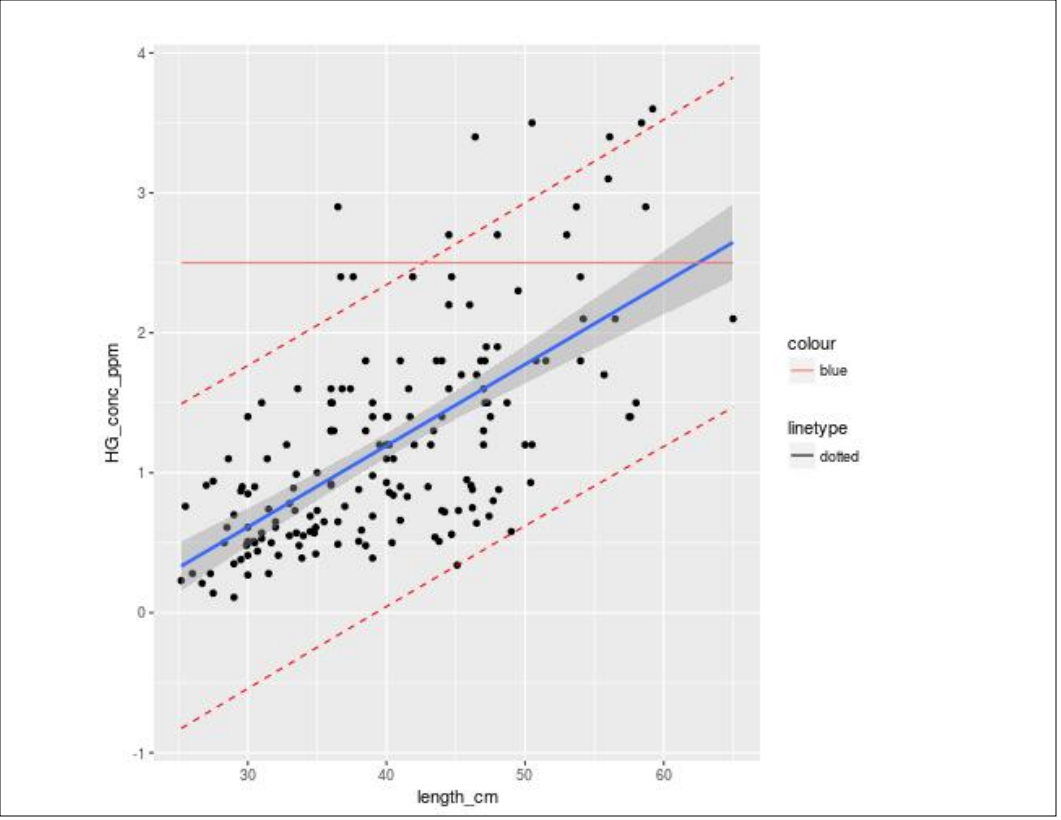
```
Average Hg = -1.13 + 0.058 length_cm
```

So, if 2.5ppm is bad, what should the policy be?

```
Hg.l.forecast = predict(Hg.l.lm, interval = "pred", level = 0.95)
Hg.l.forecast.lb = Hg.l.forecast[,2]
Hg.l.forecast.ub = Hg.l.forecast[,3]

ggplot(fishHG, aes(x=length_cm, y=HG_conc_ppm))+
    geom_point() +
    geom_line(aes(x=length_cm,y=Hg.l.forecast.lb), color = "red",
linetype = "dashed")+
    geom_line(aes(x=length_cm,y=Hg.l.forecast.ub), color = "red",
linetype = "dashed")+
    geom_smooth(method=lm, se=TRUE)+
geom_line(aes(x=length_cm,y=2.5, color="blue", linetype="dotted"))
```

```
min(length_cm[which(Hg.l.forecast.ub>2.5)])
```

Answer: 43cm
We should avoid eating fish that is longer than 43 cm.


HG_conc_ppm
length_cm
4
3
2
1
0
-1
30
40
50
60
colour
blue
linetype
dotted