

Week 3: Linear models SOLUTIONS

Exercise 1: Fox food

(a) We fit the model with:

```
m1 <- lm( d$WEIGHT ~ d$AVFOOD )
```

The estimates are:

```
> summary(m1)
```

Call:

```
lm(formula = d$WEIGHT ~ d$AVFOOD)
```

Residuals:

```
      Min       1Q   Median       3Q      Max
-1.24454 -0.40467 -0.05149  0.38736  1.25156
```

Coefficients:

```
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    4.4103      0.4882   9.035 8.62e-10 ***
d$AVFOOD        0.2543      0.6790   0.374  0.711
```

```
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Residual standard error: 0.6836 on 28 degrees of freedom

Multiple R-squared: 0.004982, Adjusted R-squared: -0.03055

F-statistic: 0.1402 on 1 and 28 DF, p-value: 0.7109

So for each unit increase in AVFOOD, the WEIGHT is expected to increase by 0.2543. That is, body weight is expected to increase slightly as food increases.

(b) The confidence intervals are:

```
> confint(m1)
```

```
              2.5 %   97.5 %
(Intercept)  3.410340 5.410244
d$AVFOOD     -1.136710 1.645222
```

So the estimate for AVFOOD is quite unreliable—it spans a lot of distance on both sides of zero, so it's hard to be confident about its magnitude or direction.

(c) Estimating the model and generating the confidence intervals:

```
> m2 <- lm( d$WEIGHT ~ d$GFSIZE )
```

```
> coef(m2)
```

```
(Intercept)      d$GFSIZE
 5.0500240    -0.1197476
```

```

> confint(m2)
                2.5 %      97.5 %
(Intercept)  4.3012164  5.79883172
d$GFSIZE     -0.3024380  0.06294277

```

The effect of GSIZE seems to be to reduce average weight, by 0.12 for each unit change in group size. This makes sense, as more foxes, holding the amount of food constant, means each fox gets to eat less. The confidence interval is mostly on the negative side of zero, so we can be reasonably confident this is a negative effect, but it doesn't seem very large.

(d) Now:

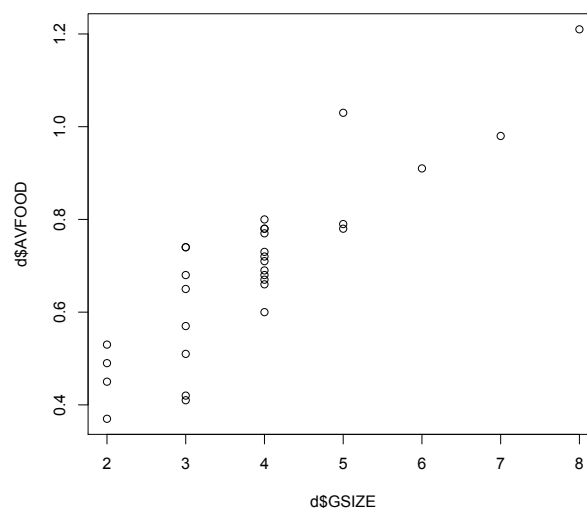
```

> m3 <- lm( d$WEIGHT ~ d$AVFOOD + d$GFSIZE )
> coef(m3)
(Intercept)      d$AVFOOD      d$GFSIZE
  3.9885681    4.4920211   -0.6526352
> confint(m3)
                2.5 %      97.5 %
(Intercept)  3.1835854  4.7935508
d$AVFOOD     2.2272747  6.7567675
d$GFSIZE     -0.9587493 -0.3465211

```

Now the estimate for AVFOOD is larger and reliably on the positive side of zero. In fact, the lower bound of the interval is much higher than the previous upper bound. The estimate for GSIZE is more negative (greater in absolute value) than before, and also consistently on one side of zero. By adding both variables to the same model, suddenly both of them are much stronger effects and precisely estimated than before.

What is going on here is that these two variables have opposite effects, but are correlated with one another. You can see this plainly from the scatter of AVFOOD and GSIZE:

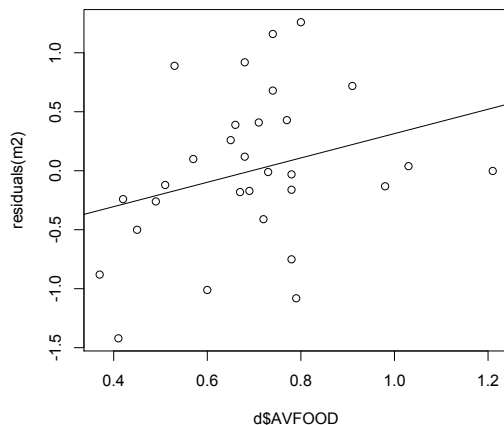


That's quite a tight correlation—use `cor(d)` to get a bivariate correlation matrix for the data frame. When two variables are correlated in the natural sample but have opposite causal effects, you need to include both or it will look like neither has much of an effect. To see why, consider why these two variables are correlated—AVFOOD is the average amount of food in a territory. It is a measure of territory quality. Foxes are attracted to good territories, and as a result, groups are larger in territories with larger values of AVFOOD. This explains why AVFOOD and GSIZE are correlated so much in this observational sample.

So every fox you add to a group saturates the food available to each fox, reducing average weight in the group. But a better territory with more food increases average weight. You showed that with the estimates above. If you include only one of these variables, it is hard to tell what is going on, because in many groups, the increase in group size almost exactly cancels out the increase in food. (Foxes are ideal free?) Luckily, there are some groups in which GSIZE and AVFOOD are not so tightly correlated—particularly at small group sizes—and so when we include both variables in a model, we can measure the effect of each, controlling for the other.

Another way to think about the combined effects of GSIZE and AVFOOD is that at any particular GSIZE, variation in AVFOOD is positively correlated with WEIGHT. You can in fact make plots that demonstrate this, using sequential regressions. Take our fit model from part (c), WEIGHT regressed on GSIZE, and plot the *residuals* against AVFOOD. The residuals are the distances from each actual WEIGHT value to the regression line. They are the remaining variation in the data, after accounting for GSIZE (in this case).

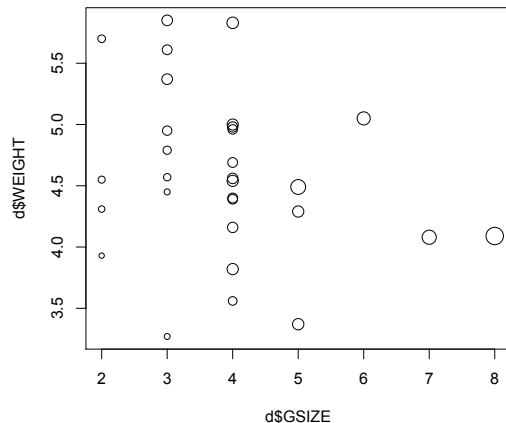
```
plot( residuals(m2) ~ d$AVFOOD )
abline( lm(residuals(m2) ~ d$AVFOOD) )
```



The reason the slope through those points is positive is that there is additional correlation between residual WEIGHT and AVFOOD, after regressing out GSIZE.

And here's yet another way to visualize the effect, plot AVFOOD on the scatter of WEIGHT and GSIZE, using point size:

```
plot( d$WEIGHT ~ d$GSIZE , cex=d$AVFOOD*2 )
```



Point size above is proportional to AVFOOD. So you can see that in each column of GSIZE values, larger points (more AVFOOD) are associated with larger WEIGHT values.

(e) Adding AREA to the model:

```
> m4 <- lm( d$WEIGHT ~ d$AVFOOD + d$GFSIZE + d$AREA )
> coef(m4)
(Intercept)      d$AVFOOD      d$GFSIZE      d$AREA
  4.0024008    3.1888929   -0.6833843    0.3468955
> confint(m4)
              2.5 %      97.5 %
(Intercept)  3.21805946  4.7867421
d$AVFOOD     0.41891097  5.9588749
d$GFSIZE     -0.98418054 -0.3825880
d$AREA       -0.09901441  0.7928054
```

The estimates for AVFOOD and GSIZE haven't changed much. AREA doesn't have much an effect, according to this model. This might be because AREA is only an imperfect measure of territory quality, while AVFOOD is a direct measure of territory quality. So once we have controlled for AVFOOD, there isn't much additional correlation between AREA and WEIGHT to explain.

(f)

Model	Mathematical form	With estimates
m1	$w_i \sim \mathcal{N}(\alpha + \beta f_i, \sigma)$	$w_i \sim \mathcal{N}(4.41 + 0.25 f_i, 0.68)$
m2	$w_i \sim \mathcal{N}(\alpha + \beta g_i, \sigma)$	$w_i \sim \mathcal{N}(5.05 - 0.12 g_i, 0.66)$
m3	$w_i \sim \mathcal{N}(\alpha + \beta f_i + \beta_2 g_i, \sigma)$	$w_i \sim \mathcal{N}(3.99 + 4.49 f_i - 0.65 g_i, 0.53)$
m4	$w_i \sim \mathcal{N}(\alpha + \beta f_i + \beta_2 g_i + \beta_3 a_i, \sigma)$	$w_i \sim \mathcal{N}(4.00 + 3.19 f_i - 0.68 g_i + 0.35 a_i, 0.52)$

The point of having you construct this table is to show you how to produce predictions from your fit models. Once you have the estimates, you can plug them back into the model itself and run the model forwards to predict outcomes for new data (or old data).

Exercise 2: Predicting primates

(a) To estimate the model:

```
m <- lm( d$Body.mass.kg ~ d$Genus + d$Type )
```

This model uses only categorical variables. So what has happened is that R constructs a series of 0/1 dummy variables for Genus. Since Type has only two categories, it needs just one dummy variable for it. So we get a complicated looking table of coefficients:

```
> summary(m)
```

Call:

```
lm(formula = d$Body.mass.kg ~ d$Genus + d$Type)
```

Residuals:

Min	1Q	Median	3Q	Max
-2.842e+00	-7.742e-01	-5.551e-16	8.808e-01	1.638e+00

Coefficients:

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	5.0233	1.0748	4.674	0.000538	***
d\$GenusMacaca	0.9050	1.1400	0.794	0.442679	
d\$GenusNasalis	7.9133	1.7984	4.400	0.000865	***
d\$GenusPiliocolobus	1.3800	1.4420	0.957	0.357434	
d\$GenusPresbytis	-0.0750	1.2488	-0.060	0.953097	
d\$GenusSimias	2.9567	1.7984	1.644	0.126095	
d\$GenusTrachypithecus	3.2100	1.4420	2.226	0.045928	*
d\$TypeMainland	2.1733	0.6797	3.197	0.007671	**

Notice that a couple of Genus categories are missing from this list. “Mandrillus” is missing, but that’s because there is no body mass measurement for the only Mandrillus row. “Hylobates” is also missing—so it must be the (Intercept) category. As for Type, R decided that Mainland would be the 1 value, so the (Intercept) must contain the Island estimate. This means the (Intercept) corresponds to a member of Genus Hylobates living on an island, and this primate has an average body mass of 5.02kg.

Going by the sizes of the coefficients alone, Type has an effect size comparable to several of the Genus categories. Living on Mainland tends to increase body mass by 2.17kg. The Genus Nasalis is much larger than the others, and this effect is larger than Mainland/Island Type. But many of the other Genus categories are small than Type, like that of Macaca. So one answer to the question “Is Type or Genus more important a predictor of body mass?” would be “it depends upon which Genus.”

What if we approach the question a different way. Suppose we operationalize “important in prediction” as proportion of variance in body mass explained. Then let’s compare two models:

```
m1 <- lm( d$Body.mass.kg ~ d$Genus )
```

```
m2 <- lm( d$Body.mass.kg ~ d$Type )
```

From the summaries of each, we can see that the residual variation for `m1` is 1.885 and the residual variation for `m2` is 2.232. This implies that Genus leaves less variation to be explained, so it is more important in prediction.

The complication here is that Genus is actually 6 parameters, while Type is only 1. So there are model complexity issues to consider here, as you take up in week 4.

(b) First print those rows with missing body mass values:

```
d[ is.na(d$Body.mass.kg) , ]
```

The line above needs some explanation. The comma inside the `[]` separates rows from columns. Data frames have two dimensions, so when you index them, you must index them in row and column dimensions. Rows are first, then columns. So the code above says “give me only the rows of `d` where body mass is not missing (see `?is.na`), and all columns.” By leaving the space after the comma blank, it means “all.” So if you type `d[,]`, you get the original data frame back, because it returns all rows and all columns, but `d[,1]` returns all rows but only the first column.

	Type	Genus	Species	Range	Body.mass.kg
1	Mainland	Mandrillus	Mandrillus leucophaeus leucophaeus	Cameroon, Gabon	NA
5	Mainland	Macaca	Macaca fascicularis fascicularis	Malaysia, Indonesia	NA
6	Mainland	Macaca	Macaca ochreata ochreata	Sulawesi	NA
14	Mainland	Hylobates	Hylobates concolor	Vietnam, Laos	NA
15	Island	Macaca	M. l. poensis	Bioko	NA
19	Island	Macaca	M. f. fusca	Simeulue	NA
20	Island	Macaca	M. o. brunescens	Buton	NA
28	Island	Hylobates	H. hainanus	Hainan	NA

All we need to do now is look up the parameter estimates for each Genus and the appropriate Type, for each row. We add these together with the (Intercept) and we get the expected body mass. In order, these are:

Row	Type	Genus	Formula	Body.mass.kg
1	Mainland	Mandrillus	$5.02 + 2.17$	7.19 kg
5	Mainland	Macaca	$5.02 + 0.91 + 2.17$	8.10 kg
6	Mainland	Macaca	$5.02 + 0.91 + 2.17$	8.10 kg
14	Mainland	Hylobates	$5.02 + 2.17$	7.19 kg
15	Island	Macaca	$5.02 + 0.91$	5.93 kg
19	Island	Macaca	$5.02 + 0.91$	5.93 kg
20	Island	Macaca	$5.02 + 0.91$	5.93 kg
28	Island	Hylobates	5.02	5.02 kg

Since Hylobates is the (Intercept) Genus, and Island is the (Intercept) Type, the island Hylobates on row 28 gets its prediction purely from the (Intercept). There is an awkward case here, on row 1, because Mandrillus was not in our original model at all. Technically, the prediction above is correct. But a reasonable response is also to say the model doesn’t make a clear prediction about Mandrillus, because the model is ignorant of them.