

Reference categories for discrete predictors

A couple of questions and ideas were raised in class that needed a better answer than I could give in class. Here are a few notes about them.

Picking a reference category for discrete predictors

The problem can be illustrated by this simple regression model, using the kidiq data frame:

```
> library(arm)

> kidiq <- read.dta("child-iq/kidiq.dta", convert.underscore=TRUE)

> attach(kidiq)

> # mom.work takes the values
> #
> # 1 = did not work first 3 years of child's life
> # 2 = worked in 2nd or 3rd year of child's life
> # 3 = worked part time in first year of child's life
> # 4 = worked full time in first year of child's life

> work.factor <- as.factor(mom.work)
> # work.factor is a categorical version of mom.work, suitable
> # for one-way analysis of variance, for example...

> summary(fit1 <- lm(kid.score ~ work.factor))
> # ...
```

	Estimate	Std. Error	t value	Pr(> t)	
(Intercept)	82.000	2.305	35.568	<2e-16	***
work.factor2	3.854	3.095	1.245	0.2137	
work.factor3	11.500	3.553	3.237	0.0013	**
work.factor4	5.210	2.704	1.927	0.0547	.

```
> # ...
```

As you can see from the above R illustration, when you give R a factor it will drop the *first category in a factor* if necessary to make the model full-rank (estimable).

On the other hand, if there are no factors (e.g. you code dummies for the categories and enter them into the model by hand), then R will drop the *last regressor in the model* if necessary to make the model full-rank:

```
> summary(fit2 <- lm(kid.score ~ work.none +
+                   work.23 + work.1p + work.1f))
> # ...
```

Coefficients: (1 not defined because of singularities)

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	87.210	1.413	61.723	<2e-16 ***
work.none	-5.210	2.704	-1.927	0.0547 .
work.23	-1.356	2.502	-0.542	0.5882
work.1p	6.290	3.050	2.062	0.0398 *
work.1f	NA	NA	NA	NA

> # ...

If you like working with factor variables directly, there is a command in R called `relevel()`, that can change which category R will drop (by making it be the first category in the factor:

```
> summary(fit3 <- lm(kid.score ~ relevel(work.factor,ref="4")))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	87.210	1.413	61.723	<2e-16 ***
relevel(work.factor, ref = "4")1	-5.210	2.704	-1.927	0.0547 .
relevel(work.factor, ref = "4")2	-1.356	2.502	-0.542	0.5882
relevel(work.factor, ref = "4")3	6.290	3.050	2.062	0.0398 *

```
> # ...
```

```
> summary(fit3 <- lm(kid.score ~ relevel(work.factor,ref="2")))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	85.854	2.065	41.582	<2e-16 ***
relevel(work.factor, ref = "2")1	-3.854	3.095	-1.245	0.2137
relevel(work.factor, ref = "2")3	7.646	3.402	2.248	0.0251 *
relevel(work.factor, ref = "2")4	1.356	2.502	0.542	0.5882

```
> # ...
```

Alternatively, if you have coded dummies for the categories yourself, you can just drop whichever level you don't want in the model:

```
> summary(fit4 <- lm(kid.score ~ work.none + work.1p + work.1f))
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	85.854	2.065	41.582	<2e-16 ***
work.none	-3.854	3.095	-1.245	0.2137
work.1p	7.646	3.402	2.248	0.0251 *
work.1f	1.356	2.502	0.542	0.5882

```
> # ...
```

The reference category

The category that is dropped becomes the “reference category” for interpreting the coefficients. This means that the intercept becomes the estimated effect for the *dropped* category, and the coefficients on the other

categories become deviations from this effect. For example, in the `fit4` model above:

- The estimated effect for `work.23` is the intercept, 85.854.
- The estimated effect for `work.none` is the sum of the intercept and this coefficient estimate: $85.854 + (-3.854) = 82.000$.
- etc.

A difficulty with “reference categories” is that standard errors cannot always be read off the coefficient summary table. For example we do not know the SE for the effect of `work.none`, because it is a combination of the two SE’s listed (2.065 and 3.095) together with some covariance between $\hat{\beta}_{intercept}$ and $\hat{\beta}_{work.none}$ in this model; that is, it depends on more entries of the variance-covariance matrix $(X^T X)^{-1} \hat{\sigma}^2$.

I generally prefer to omit the intercept, so that I get direct estimates of the effects of each category. In this case, the SE’s listed in the coefficient table are exactly the SE’s for the category effects, so there is no offstage magic with $(X^T X)^{-1} \hat{\sigma}^2$ to worry about:

```
> summary(fit5 <- lm(kid.score ~ work.none + work.23 + work.1p + work.1f - 1))
> # ...
```

	Estimate	Std. Error	t value	Pr(> t)
work.none	82.000	2.305	35.57	<2e-16 ***
work.23	85.854	2.065	41.58	<2e-16 ***
work.1p	93.500	2.703	34.59	<2e-16 ***
work.1f	87.210	1.413	61.72	<2e-16 ***

```
> # ...
```

In some situations it does make sense to choose a reference category. In my experience there is always a *substantive, scientific* reason for choosing the reference category. For example, in the `fit5` model above, as a matter of social science or education policy we may be directly interested in the difference in expected kid’s score between moms who didn’t work at all and moms who worked at various levels during the first three years of the child’s life. In that case we should just drop `work.none`, because now the other estimated coefficients (and their SE’s) directly estimate these differences:

```
> summary(fit6 <- lm(kid.score ~ work.23 + work.1p + work.1f))
> # ...
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	82.000	2.305	35.568	<2e-16 ***
work.23	3.854	3.095	1.245	0.2137
work.1p	11.500	3.553	3.237	0.0013 **
work.1f	5.210	2.704	1.927	0.0547 .

```
> # ...
```

For example, the difference in expected child’s score between mothers who work part time in the first year, vs mothers who don’t work at all, is an additional 11.5 points, with an SE of 3.095; this is a statistically significant difference. On the other hand the difference in expected child’s score between mothers who

The most frequent category

[illegible]

```
> summary(no.X2 <- lm(Y ~ X1 + X3))
> # ...
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.7645	0.2776	17.165	< 2e-16 ***
X1	-0.2677	0.3926	-0.682	0.497
X3	1.2832	0.2944	4.358	3.26e-05 ***

```
> # ...
```

```
> summary(no.X3 <- lm(Y ~ X1 + X2))
> # ...
```

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	6.04773	0.09814	61.624	< 2e-16 ***
X1	-1.55086	0.29442	-5.268	8.33e-07 ***
X2	-1.28320	0.29442	-4.358	3.26e-05 ***

```
> # ...
```

It does appear that the SEs for model no.X3 are smaller than (or equal to) the SEs for models no.X1 and no.X2, but *one must keep in mind that different quantities are being estimated*. For example in no.X3 the coefficient for X1 -1.55086, and SE 0.29442, says that the effect of X1 is significantly different from the effect of X3. In model no.X2 the coefficient for X1 -0.2677, and SE 0.3926, says that the effect of X1 is *not* significantly different from the effect of X2.

Thus, we haven't actually improved the same SEs for the same estimated effects by dropping X3, rather we have generated SEs for different estimated effects. The SEs are not at all comparable.

Moreover, predicted values for a new data point have the same SEs under all three models (as the matrix algebra would tell us they must):

```
> unlist(predict(orig.model, data.frame(X1=1, X2=0, X3=0), se=T))
      fit.1      se.fit      df residual.scale
4.4968655    0.2775785    97.0000000    0.8777804
> unlist(predict(no.X1, data.frame(X2=0, X3=0), se=T))
      fit.1      se.fit      df residual.scale
4.4968655    0.2775785    97.0000000    0.8777804
> unlist(predict(no.X2, data.frame(X1=1, X3=0), se=T))
      fit.1      se.fit      df residual.scale
4.4968655    0.2775785    97.0000000    0.8777804
> unlist(predict(no.X3, data.frame(X1=1, X2=0), se=T))
      fit.1      se.fit      df residual.scale
4.4968655    0.2775785    97.0000000    0.8777804
```

My conclusion is that dropping a category to reduce SEs doesn't actually make sense, because the SEs are for estimating different things, and therefore the SEs cannot be compared.

It is far better, as suggested above, to choose a reference category to match the scientific questions you are asking. If there is no good substantive scientific reason for choosing a reference category, I suggest dropping the intercept instead, where possible.