Dummy Coding

Background

When you have a categorical (i.e., nominal) variable, such as gender (male/female), relationship status (single/dating/married), or race (Asian/Black/Hispanic/White), you will need to assign numbers to each category in order to be able to analyze your data.

One common way of assigning numbers to categories is to use dummy variables. Dummy variables are variables that are either 0 or 1. For example, if we wanted to dummy code gender, we might create a variable called *male*. We would set the *male* variable to 0 for women and we would set it to 1 for men. Thus, dummy variables can also be thought of as "binary flag variables."

Dummy Coding Variables with Two Levels

If your categorical variable has only two levels (such as gender: male or female), you can create a single dummy variable. For example, you could create a variable called *male*. You should set this variable to 0 for women and 1 for men. For example, consider the hypothetical data below:

participant_id	gender	male	height
1	female	0	60
2	male	1	70
3	male	1	68
4	female	0	64

The "male" variable is dummy coded. It is either 0 or 1 and captures whether a person is male or not.

You get to decide which direction you want to code each of your variables. For example, you could create a "female" variable instead of a "male" one, if you would like:

participant_id	gender	female	height
1	female	1	60
2	male	0	70
3	male	0	68
4	female	1	64

Always give your dummy variables meaningful names. It's a good idea to name your dummy variables such that 1 means "true" and 0 means "false." Thus, in the example above, if it is true that the participant is male, the "male" variable is 1. If it is false that the participant is male, the "male" variable is 0. There is no ambiguity! Contrast that with a dummy-coded variable simply called "gender." If you simply call your dummy variable "gender," it's not clear whether 0s and 1s are males or females.

Dummy Coding Variables with Three or More Levels

If your categorical variable has three or more levels, you will need to create multiple dummy variables. Specifically, if your variable has *L* levels, you will need to create *L*-1 dummy variables. For example, imagine you measure race as Asian, Black, Hispanic, or White. There are four racial groups, so you will have to create at least three dummy variables.

participant_id	race	asian	black	hispanic	height
1	Asian	1	0	0	67
2	Black	0	1	0	69
3	Hispanic	0	0	1	66
4	White	0	0	0	68

Why do you need only 3 dummy variables?

In this example, if all three dummy variables (asian, black, hispanic) are zero, we know the person is White. Creating an additional dummy code for "white" would give no additional information. (In this case "Other" was not a racial option. If it were, we should create a dummy variable for "race_other".)

How do you decide which category to leave out?

The category you leave out while dummy coding is called the *reference group*. All of the other groups will be compared to the reference group. For example, if you were to analyze the above data, your results would tell you that the Asian participant is 1 inch *shorter than the White participant*, whereas the Black participant is 1 inch *taller than the White participant*.

So, when deciding which group you want to be the reference group, you need to decide which group you want to compare all of the other groups to.

Can I have multiple reference groups?

No. You can only have one reference group for each individual analysis you run. But you can run multiple analyses and switch the reference group across different analyses. Using the above example, you might run one regression with Whites as the reference group, and a second regression with Asians as the reference group.

Can I include dummy variables for *all* levels of the categorical variable in my regression?

No. You *cannot* include dummy codes for all levels of the categorical variable in a single regression. The reason why is easiest to see with a two-level variable:

participant_id	gender	male	female	height
1	female	0	1	60
2	male	1	0	70
3	male	1	0	68
4	female	0	1	64

In the above example, we have dummy codes for *both* "male" and "female." When using regression, you have to decide whether you want women to be the reference group…

```
mixed height with male
/fixed=male
/print=solution
```

…OR whether you want men to be the reference group:

```
mixed height with female
/fixed=female
/print=solution
```

The first regression tells you how much taller men are than women, whereas the second regression tells you how much shorter women are than men.

Why can't you include both?

```
mixed height with male female
/fixed=male female
/print=solution
```

This regression asks the question, "How much taller are men than women, holding gender constant?" In other words, this regression asks, "If men and women were the same height, how much taller would men be?" This question is logically nonsensical. Attempting to run this analysis will also literally break the math regression uses. SPSS will either return an error or simply ignore one of the redundant variables.

Similar logic can be applied to variables with three or more levels. Thus, irrespective of how many levels of your categorical variable has, you must always include dummy codes in your model for all levels *except one*. The category you leave out is your "reference group." All other groups included in the model are compared to the reference group.

Tips for Dummy Coding

You can use SPSS's syntax to easily dummy code variables. Imagine you want to dummy code race and your data is currently in this format.

participant_id	race
1	Asian
2	Black
3	Hispanic
4	White

First, you will need to create dummy variables for the racial groups and set them all to zero.

```
compute asian = 0.
compute black = 0.
compute hispanic = 0.
execute.
```

This code will create variables that are set to zero for everybody:

participant_id	race	asian	black	hispanic
1	Asian	0	0	0
2	Black	0	0	0
3	Hispanic	0	0	0
4	White	0	0	0

Next, we can use "if" statements to change our dummy variables to 1 for the appropriate people:

```
if (race="Asian") asian=1.
if (race="Black") black=1.
If (race="Hispanic") hispanic=1.
execute.
```

The end result is properly dummy-coded variables:

participant_id	race	asian	black	Hispanic
1	Asian	1	0	0
2	Black	0	1	0
3	Hispanic	0	0	1
4	White	0	0	0