WEB APPENDIX CHAPTER 5

Mooi, E., Sarstedt, M., & Mooi-Reci, I. (2018). *Market Research. The Process, Data, and Methods using Stata.* Heidelberg: Springer.

In Chapter 5, we discussed dummy variables (often referred to as "dummies" but occasionally as binary indicators or fixed effects). A dummy variable represents the presence, or absence, of a characteristic. Consequently, dummies take on two values, 0 and 1.[1] If, for example, advertising is used, a dummy variable takes on the value 1. If no advertising is used, the dummy variable takes on the value 0. Dummy variables are often used in regression analysis (discussed in Chapter 7), but also in other statistical techniques, such as *t*-tests (discussed in Chapter 6) and as part of cluster analysis (discussed in Chapter 9).

There are several types of dummy variables that serve several purposes:

1. Dummy variables that are naturally coded (or recoded) as 1 or 0. For example, variables such as gender are often coded as 1 (e.g., for females) and 0 (for males). There is no need to explicitly create dummies because these variables already only take on two values.

2. Dummy variables are sometimes required for interpretation of categorical or nominal variables. One such variable could be, for example, the variable *shop type* distinguishing between three different levels: 1) department stores, 2) supermarkets, and 3) discount stores. Assigning values 1-3 and interpreting this as meaning that department stores are three times more profitable than discount stores makes no sense as the measurement scale is nominal. In this case, dummies offer a solution by facilitating meaningful interpretations. We can indicate these three different levels by creating two (not three, as we'll discuss later) dummies. A first dummy can be created to indicate department stores (1 for department stores, 0 for both supermarkets and department stores). A second dummy can indicate supermarkets (1 for supermarkets, 0 for department stores and discount stores). The third dummy is not required, or even possible to use statistically because if we already know a store is not a department store or supermarket, it is a discount store. Therefore, always create one dummy less than the number of categories or levels! Thus for 4 levels, you only need to create 3 dummies. An issue to consider is the level for which we do *not* create a dummy. The level in which you are the least interested in research-wise or one that is a common comparison base would be best for this.

3. Dummy variables can also be used to split up ordinal, interval, or ratio scaled variables. For example, if we have a ratio scaled variable called *age* measuring the age of customers, we can define customers as underage (younger than 18 years) or mature (18 or older). In doing so, we create two categories. We could, of course, also split age into multiple categories. Again, create one dummy less than the number of categories. If you have five categories, only four dummies are needed. Keep in mind that when we do this we lose information. That is, if we only have access to dummies measuring age as below or above 18 years we cannot construct the variable age but we can construct the dummies from this variable. The same applies for interval scaled variables. For

[1] And, you'd be well advised to use 0 and 1 and not, say, 1 and 2 as the estimated coefficient directly indicates the effect when the dummy takes on the value 1.

ordinal variables, we lack information on the precise differences between scale categories and dummies are more useful.

Creating Dummy Variables in Stata

We use the Oddjob dataset (🖱 Web Appendix → Chapter 5) for the subsequent examples and illustrations. Relevant variables for these examples include customer's *gender*, *age* and *traveler status*.

1. *Dummies for nominal variables*
   In the Oddjob dataset, the variable *Gender* is coded as 1 for female and 2 for male customers. Let's assume that you want to create a new dummy variable for gender that you want to call *female* where 1 indicates female customers and 0 those otherwise. The easiest way to create this directly is by typing the follow in the **Command** window:

   ```
   generate female = gender==1
   ```

   Stata will create a new variable called "female" with two possible categories: one that equals 1 if customer's gender is female and another category equaling 0 if otherwise. You have now created a new variable *female* located at the bottom of the variables list.

2. *Dummies for interval variables*
   Let's assume, that you like to contrast the flight behavior of young and middle-aged customers (<40 years) with that of older customers. One way to construct a dummy variable *based on an interval variable* (i.e., age) would be to type the following command:

   ```
   generate age_dummy = 0 if age<40 & age ~=.
    replace age_dummy = 1 if age>=40 & age ~=.
   ```

   The first line of the command indicates that we generate a new dummy variable that we call *age_dummy* and that takes on the value of 0 when age is less than 40 and `age ~=.` indicates that age should not be missing. The following `replace` command indicates that the dummy should take on the value of 1 when age is equal to 40 or higher. Again, we specify that age should not be missing.

   If you wish to create this variable using the Menu bar options, we follow the same steps as discussed in Chapter 5. Go to ► Data ► Create or change data ► Other variable-transformation commands ► Recode categorical variables. This will open a dialog box similar to the **Main** Tab left in Fig. 1.

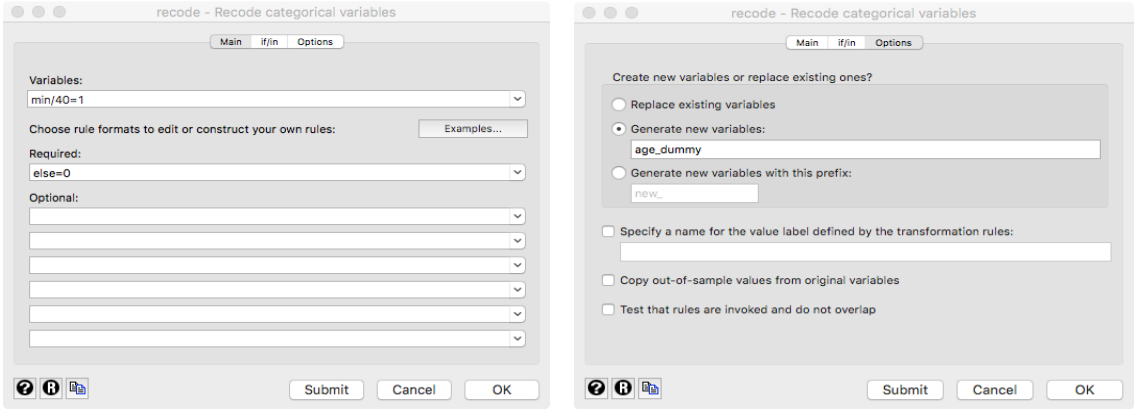

Fig. 1 Recode into different variables (Main and Options tabs)

Specify the name of the variable that you want to recode (i.e., *age*) under **Variables** in the **Main** tab. Next, specify the values of the new variable under **Required**. These are based on the values of the original variable (*age*). The option (min/40=1) indicates that all values ranging from the youngest age observations to the age of 40 should be coded as 1. Under **Optional** all other age observations are coded as 0 (else=0).

Next, click on the **Options** tab (right in Fig. 1). In the dialog box that follows, you need to indicate whether you want to: (1) **Replace existing variables**, (2) **Generate new variables**, or (3) **Generate new variables with this prefix**. We always recommend using either the second option or the third. If you were to use the first option, any changes you make to the variable will result in the overwriting of the original variable. Consequently, if you thereafter wish to return to the original data, you will either need to revert to a saved previous version, or need to enter all the data again, because Stata cannot undo these actions! Select the second option (i.e., generate new variables), enter the name of the new variable (i.e., *age_dummy*), and then click on **OK**. Alternatively, the recoding of this variable can be obtained through the following command:

```
recode age(min/40=1)(else=0),generate(age_dummy)
```

3. *Dummies for splitting nominal variables*
We can create dummies for nominal variables from the original variables provided. Let's assume that we want to compare the behavior of customers with different travelers' status (*status*). Given that status includes 3 categories: 1) Blue; 2) Silver; 3) Gold, this means that we need to create only two dummies. As the Blue states is the lowest tier, this seems to be a good base category against which to compare. We could follow the same procedure as in example 1 and type the following in the **Command** window:

```
generate silver_dummy = status==2
generate gold_dummy = status==3
```

You have now created two new dummy variables that are listed in the bottom of the variable list.

Alternatively, Stata can create a dummy variable out of frequency table. This is useful if you wish to generate different sets of dummy variables out of nominal or categorical

variables with more than 2 groups. In the case of *status*, you type the following in the **Command** window:

```
tabulate status, generate(status)
```

The first half of the command (before the comma), asks Stata to display a frequency table of the original nominal variable *status*. The second half of the command (after the comma) asks Stata to create automatically a set of dummy variables for each category in the *status* variable. That is, one dummy that equals 1 for Blue status travelers and 0 if otherwise, one second dummy that equals 1 for Silver status travelers and 0 if otherwise, and another third dummy that equals 1 for Gold status travelers and 0 if otherwise. Consequently, Stata will automatically produce three new dummy variables labeled as `status1, status2` and `status3` that are listed at the bottom of the variable list. Note in a regression analysis, you should leave out the reference or base category which in this case was *status1* (i.e., the Blue status travelers).