October 13, 2010
PLS 300 Fall 2010

Recoding Variables and Doing Cross Tabulations in R

1. These commands are applied with the NES 2004 dataset. Commands for loading the data, and examples in R are on the R lab script file. In addition, you may copy and paste the R commands from this handout directly into R.

   For reference, download the introductory and main codebooks for the 2004 NES, `http://faculty.gvsu.edu/kilburnw/nes04int.txt` , and `http://faculty.gvsu.edu/kilburnw/nes04var.txt`. After these two text documents download, you'll want to save them for reference to the variable contents. And then open each one up in a separate text file window on your computer. To use each one, search for the variable name by pressing 'CTRL+F' to find variable names such as "v045088'. .

2. Next, you will find it useful to have the introductory codebook open in a text file along side your R window. Once you have these items ready, we'll begin:

3. using these commands require two additional packages, `gmodels` and `car`. These packages should be added to the list of packages in your R syntax file.

After loading the NES dataset, you'll see with `ls()` that it is saved as a data frame in the workspace. We can refer to it with the prefix `nes2004$`, which is used below.

Creation of new variables Most variables in the NES dataset need to be recoded prior to use in quantitative analysis — we need to figure out what to do with the respondents who said they 'didn't know' or perhaps we want to recode the existing response categories into others. Take as an example Bush's overall job handling (his 'presidential approval rating'). This variable is `v043025`. Try `table(nes2004$v043025)`.

While we could directly recode this variable, it is better coding practice to create a new variable that we subsequently recode. For `v043025`, we want to recode as missing values those persons who chose 'don't know' or who refused to answer the question. We do this with the 'gets' function, $<-$.

We use the symbol `$` combined with the data object to instruct R to create a new variable within that data object, in this case, 'nes2004':

`nes2004$bushjob<-nes2004$v043025` tells R to create the variable bushjob in nes2004

recoding variables for missing values Let's first create a new variable for one of the Bush feeling thermometer variables. `nes2004$bushfeel<-nes2004$v043038`.

With > `table(nes2004$bushfeel)` you'll see that the variable contains an 888 code for a don't know response.

So to drop these respondents from the analysis, we'll use
>`nes2004$bushfeel[nes2004$bushfeel>100]<-NA`. The condition inside the brackets `[]` tells you what values of the variable should be recoded, in this case, all responses greater

than 100 should be given a missing value, indicated by the <`-NA`. Any other numeric value would valid as a replacement for NA.

For preparing your cross-tabs, this command is very useful. For any variable for which you would like to have 'don't know' or 'refusal' responses coded to missing, you can just tell R to make values such as those greater than 7 or 100, or whatever it is, to be assigned missing values.

Transforming a numerical, continuous (interval or ratio) scale variable into a factor This is accomplished with the `cut()` function and the `recode()` function. The `cut()` function as two arguments.

converting a numeric, continuous variable into discrete groups You may find it useful to convert variables that are continuous, such as feeling thermometers, into variables that contain only a few levels, such as 'cool', 'lukewarm', or 'hot'. Or 'low', 'medium', or 'hot'. To do so, we use the `cut` function to create a new variable that is stored as a factor with the specified labels.

The option `labels=c(Low,Medium,High)` adds specific labels to each of the factor levels. Try first creating a new variable for any of the continuous variables, such as a feeling thermometer score for Bush, and then convert it to a factor with three levels of equal length along the feeling thermometer scale.

```
> nes2004$bushhot=cut(nes2004$bushfeel, 3, labels=c('cool', 'lukewarm', 'hot'))
```

Then check `table(bushhot)`. Or after you learn to do a Cross Tabulation, cross-tabulate it with the original variable to see which numeric points on the thermometer scale were included with which of the three categories.

While not required for right now, an alternative way to recode variables is via the `car` package. If you wish to try it, you need to load it via `library(car)`. The function we will use is called `recode()`. Bring up the help file to see example commands. You could use the car package to set your own cut-offs.

removing a variable Our NES dataset is stored as a data frame in R; to remove a variable created within it, we use the following command:

`nes2004$varname<-NULL` , such as >`nes2004$kerryft<-NULL`

Tabulations and Cross-Tabulations

To tabulate variables, or to do cross-tabulations like in class, we first need to install the `gmodels` package. Enter the commands: <`library(gmodels)`. We will use the function `CrossTable`. (After the library is loaded, you can type `>help(CrossTable)` to bring up the various options for this function.)

The function is useful for producing frequency tabulations. Try this command:

>`CrossTable(v043025, max.width=1)` or max.width=2 for double column display

The `max.width` command is not always necessary, but without it CrossTable will make some assumptions you may not like about how to display the results. Note that CrossTable provided proportions for each category, but if you prefer percentages, try the 'SPSS' formatting option: `format=c("SPSS")`

```
>CrossTable(v043025, max.width=2, format=c("SPSS"))
```

Or to produce a cross–tabulation of variables named 'bushjob' and 'pid':

```
>CrossTable(bushjob, pid, digits=2, format=c("SPSS")
```

A few tips on using CrossTable: If one variable has longer value labels than the other, list it first in the command so that it appears as the row variable. You'll see that in addition to frequency counts and row and column percentages, the table also includes a measure of each cell's contribution to a Chi-Square test of independence. To turn off this option, add `prop.chisq=F` to the `CrossTable()` function. The `digits=2` limits the display to two decimal places.

So with this brief overview of recoding and doing cross-tabulations, you should be prepared to do some of your own cross-tabs for the homework assignment. In addition, you'll want to look through the codebooks for the NES to identify any potential variables you might want to use in your research paper.