***Downloading a Pew Dataset and Using SPSS for Analysis: Part One***

If you are in the department’s research methods class, this is the first in a series of handouts designed to teach you the basics of SPSS that you will need to complete class activities and do well on exams.

If you are taking senior seminar, a 4000-level PSC class, or working on an independent project, these handouts should collectively cover almost everything that you will need to know about SPSS to do the statistical elements of your project. Keep in mind that these handouts are meant to reinforce lessons covered in class. For almost every procedure, I have created screencast videos that will walk you through the same process and probably be easier to absorb if you are trying something for the first time.

**Where will you get a dataset that you can manipulate in SPSS?**   
Many students doing independent or thesis projects will want to download datasets from one of the various Pew Research Center websites; there are hundreds—if not thousands—of datasets available for free download there. If you cannot find what you are looking for at Pew, there are lots of other research units that have free, easily downloadable survey dataset on specific issues, countries/regions, and populations. Links to many different sources of data are available in the Unit 1 assignment schedules for PSC 2019 and PSC 4099 on Dr. Setzler’s website.

Once you have found a survey that looks like what you need, you will need to **download your survey**. The process below describes what you do for Pew datasets. Most of the other major sources of data have a similar process even if the interface looks quite different.

1. Before downloading any dataset, **think carefully about how you want to organize the folders where you will be putting the data**. You should **create a separate folder for your dataset**, and **put that folder in your computer’s Documents folder** so that you can automatically and continuously sync all of your work to the cloud using Google Drive or a similar service. One big advantage to Google Drive is that it saves multiple versions of your files, allowing you to recover work if you accidentally save over a file you needed to keep.   
     
   It is important to think through in advance the location of the folder where your dataset is going. It will create issues if you have to move things around later on. For most SPSS projects, you will be creating syntax (i.e., writing out a list of commands telling SPSS what to do, aka “coding”). Saved syntax often includes instructions telling SPSS where to get and save files (i.e., your syntax will list a specific pathway); if you move your dataset after you have written syntax telling SPSS where to get or save data, the syntax either won’t work because the data won’t be where its supposed to be or it may put your changed dataset in the wrong place.
2. When try to download datasets from a reputable research thinktank, you may need to sign-up for a use id (email address) and password the first time you download data. Pick something easy to remember or write it down so you can reuse your log-in information.
3. When you click on the download button on the Pew website, you will be saving a **compressed** (“zipped) file that has at least three different, important files (the one with the \*.sav extension is an SPSS-ready dataset that has all of the survey’s respondents compiled in an excel-like spreadsheet, a second file explains the survey’s methodology, and a third--the “codebook” (it will be labeled “questionnaire”--gives you the exact wording used for every question in the survey and details how specific questions are named in the dataset). If the dataset you download is compressed, extract it to the same folder.   
     
   For other websites, you may need to download the codebook and dataset separately. In many cases, the methodology of a survey will be explained in the codebook. If not, you may have to locate it elsewhere.

Make sure to download all three items before you begin to work with the data. To write a research report, you will need to have a copy of the exact phrasing for the questions people were asked (often, this requires the questionnaire) and the survey’s methodology.

1. Pew makes data available in SPSS format (i.e., \*.sav files). However, many websites instead make files available in Stata (\*.dta), CSV, and sometimes R files. **If there is no SPSS version of the dataset, download the Stata (\*dta) version and open it within SPSS**. You won’t be able to open a Stata file directly, but you will be able to open SPSS and then import the Stata file into SPSS format (File -> Import Data -> Stata -> browse until you find the folder where you have save the data file. After the import, you can save a copy of the dataset in SPSS format.
2. Downloading datasets takes time and sometimes data won’t be available forever. **As soon as you download a dataset that seems like it will work for your project, make a copy of it and always work with the copy.** One easy way to keep things straight in your ahead is to add these to the end of the two files: original\_YourDataset and copy\_YourDataset.

**WHAT ARE YOU SUPPOSED TO DO WITH A DATASET ONCE IT IS DOWNLOADED?**   
Once you have downloaded, extracted from a zipped file (if necessary), and saved each of the files you will be able to **review three important types of files**. (Again, if you aren’t getting your data from Pew, you may have to download these three files separately rather than as one compressed folder).

1. First, ***review the methodology text (or pdf) file*** in order to see how many respondents were interviewed for the survey, who was surveyed (is the sample representative of all Americans? Of young people? Of Latinos), what the margin of error is, and how different demographic groups have been weighted to make the sample more representative of the population it is supposed to mirror. You will need to have access to this file when you write up the “methods” section of your research paper.
2. Next, ***carefully review the codebook, aka the questionnaire*** (for Pew, this file usually has “que” in the title for “questionnaire”) in order to find your variables of interest and appropriate control variables. This will allow you to answer important questions like: What is the structure of the variable that you are trying to explain (e.g., whether someone supports torturing suspected terrorists)? What other variables might be used as indicators for the factors that you think are likely to impact your primary variable of interest (e.g., religiosity or level of education). Looking through the codebook for your variables is much easier and less confusing than trying to look through the dataset (in variable view) in SPSS’s data editor window, so take your time and read through the questions and responses. Think and take notes about how these variables may need to be changed to make them more useful in your analysis and visual presentation of data. You may be tempted to skip reading the codebook because the dataset’s variable labels will list the survey question—Don’t do this.   
     
   **One really important piece of information in the codebook is whether every respondent was asked each question.** You may want to study women in a survey that was given to 1000 people. However, sometimes, Pew will only ask a subsample of respondents a given question (i.e., some will be given “form A,” others “form B,” etc. This is done so that more people can be asked more questions, but it also means that sometimes your sample size is not as big as you think it is (i.e. only 250 women may have been asked some questions, and that sample will be too small for some types of studies). Also, you sometimes can find that you are interested in the cause of variable x on variable y, but the half of the sample who were asked question x was not asked question y. In this situation, you will have zero respondents who answered both x and y.
3. Once you have isolated some variables of interest and noted their variable names, it will be time to open the main dataset so that you can ***use SPSS*** to:   
   a) recode all of the variables that need to be “cleaned” up before we can analyze them,  
   b) compute any new variables we may want to make by combining data from multiple questions, and  
   c) begin analyzing the data once our dataset has been “cleaned” (i.e. recoded to deal with missing data and other issues) and saved.

**SPSS BASICS 1: What do you need to know about the THREE main work areas within SPSS?**   
When using SPSS, you will be toggling back and forth between three work areas (each of which can be saved and opened separately): The data editor, the syntax window, and the output screen

1. When you open a dataset or a new SPSS file, you will see***the data editor window*, which has two views** (you access them toggling between the tabs at the bottom of the window).

**The “Data View” tab** displays all of the values for your dataset in a spreadsheet format that looks like an Excel worksheet. The variables (i.e., the questions) are listed in columns across the top of the screen in the columns, while information about each individual respondent is listed in the rows. So, if you have 1756 rows, that’s how many folks answered the survey. You may use SPSS’s point-and-click interface from this screen. Even if you are working with a dataset that already has variable labels, this window is a good resource if you want to look at patterns across specific respondents (every row contains the set of survey answers provided by a different respondent).

If you click on **the “Variable View” tab** at the bottom of the main window, you can review all of the basic information about your variables. The variable-view tab is handy if you want to quickly see how a variable has been labeled as well as how its various response categories are labeled.

1. ***The output file*** is where SPSS puts the results of most commands, including those related to changes you make to your dataset and any statistical analysis you run.

It is also where SPSS will let you know if something went wrong with your analysis, so it is important to take a look at this screen every time you run an SPSS command.

While you can look at Variable View in the data editor to get information about your variables, another way quickly to find information about variables is to **run a Codebook command** (to do so: Go to the SPSS command labeled Analyze and then -> Reports -> Codebook. Double-click on each variable you want information on, shifting that variable from the left to the right window. Once you are done, hit OK.

When you run a Codebook command, the output window will list information about the coding numbers and labels for each variable you have identified as well as the number and percentage of respondents in each variable’s response categories.

1. ***The syntax file*** is where you keep a written record of commands you are using to make changes to your dataset (e.g., to open the dataset, to re-label variables, to compute new variables that combine several variables in the original dataset, to remove variables that don’t need to be in the dataset, and to save the final version of the dataset. **THIS IS THE MOST IMPORTANT FILE YOU WILL BE USING, SO YOU WANT TO MAKE SURE TO SAVE FREQUENTLY AS YOU MANIPULATE THE DATASET SO THAT YOU CAN USE IT!**    
     
   When you are working with a dataset for the first time, you will can **create a new syntax file in the same folder where the dataset is**: SPSS File -> New -> Syntax.   
     
   If you want **to make any notes in your syntax to remember what different commands are about, you can type a sentence that begins with an asterisk (the asterisk tells SPSS not to run that line of code).** If you begin a syntax sentence with an asterisk, SPSS will know that it is not supposed to read that sentence up to the next period or line break, whichever comes first.   
     
   In many cases, you can make changes to your dataset using SPSS’s point-and-click options, but you want to use syntax to tell SPSS what to do step-by-step because this is the easiest and most effective way to ensure that (1) your work is always saved, (2) you have a clear record of every decision you made to change some aspect of the dataset, (3) you will be able to pick up right where you left off during the last session, and (4) you will be able to quickly fix any mistake you make in coding without   
     
   So, does this mean that in order to use SPSS you are going to have to learn a bunch of complicated coding? Nope. Looking at the menu at the top of the syntax page, you will see that you can point-and-click your way through different SPSS program commands (e.g., you can re-label a variable or generate various statistics), much in the same way you would if you were using drop-down menus to change formatting in a Word or Excel document. **Once you have used the point-and-click menus and windows to enter all of the specifics of what you want to do, you will just need to hit the “paste” button, which adds the code you have just created to your syntax file**, where you will be able to select the new code and run it, making changes if necessary and saving you code in case you need to see later on exactly what you did.

**SPSS BASICS 2: What are THE main TYPES OF COMMANDS in SPSS?**

There are several drop-down menus on the main menu bar that can be accessed at the top of the data editor or syntax screen. The most important options on those menus are the following:

1. **Data:** The Data menu provides commands for defining variables, inserting variables or cases, sorting files, splitting files, merging data sets, aggregating data, or using a select command to look at a subgroup within the data file. For our students, the two most common choices here will be:  
    **Split file -> compare groups-> Select a variable.**  You can do this if you want to compare statistics across different values of a single variable. For example, you might spilt on the variable “women” so that you would get results for women and men separately. This step is very helpful when you are first starting a project because you can split each of your potential independent variables (one at a time, not all at the same time!) and run a frequency of your dependent variable/s), quickly seeing whether that IV variable appears to have different DV distributions for each value of the IV**. After you are done with analyses using the split option, you need to go back and reset the options so everyone is analyzed together.**   
     
   **Select cases -> filter off** is very helpful if your study is only looking at people from a certain place or a certain party because it allows you keep only observations (e.g., telling SPSS only keep cases if the variable country is equal to 14, 18, or 23 could be used with a Global Attitudes survey where you wanted to only look at people from the countries corresponding to these three numbers). If you work with this command, make sure that you are not using your original data. **When you want to permanently drop observations, you should select the option that tells SPSS to delete all observations that do not meet the keep criteria.** Important: When you are using this command is permanently alter your dataset, make sure you are working with the correct copy of your dataset and run the command from a syntax line so that you document every change you are making to the dataset.
2. **Transform:** The *Transform* menu allows you to transform your data set on the basis of existing variables. Among other things, you can recode your variables and compute new variables from existing ones.
3. **Analyze:** The *Analyze* menu helps you to perform statistical operations on your data set, the output of which will be displayed in the Output Viewer.
4. **Graphs:** The *Graphs* menu contains a number of graph options that allow you to visually display data in the Output Viewer; however, you also can generate the most common graphs by checking the appropriate options when using procedures from the *Analyze* menu. For your finished projects, you usually will want to use Excel rather than SPSS graphs because the former are much more attractive.

**SPSS BASICS 3: How hard is it to actually use SPSS?**

Using SPSS is pretty straightforward if you move slowly and pay attention to the logic of your coding. SPSS is used throughout the business world and government. It is designed to be easy to use, intuitive, and powerful. We have made your thesis work and homework in PSC 2019 even more straightforward by asking you to use Pew’s or similar organizations’ datasets, which all come with pre-labeled variables, saving you the tens of hours that researchers must commit to entering and labeling data they have collected themselves.

**Almost all of your work with SPSS will use the same basic six-step strategy** to either prepare the dataset for analysis (e.g., creating or labeling variables or making it so you can compare women respondents to men) or actually “run” statistics (e.g., figure out what the average education level is for the survey’s respondents).

1. First, you **go to the syntax window and use the point-and-click options to tell SPSS what you want to do to the dataset**. This frequently will involve checking a bunch of options and supplying information in a separate dialog window that will pop up after you’ve told SPSS what command you want to use.
2. After you use the dialog boxes for a specific command tell SPSS exactly what you want it to do, you **select the “paste”** button so that these commands are added to your syntax file (we’ll walk you through this in more detail in the example below). If you don’t see a paste button—this should only happen with Mac users—make sure that you have configured SPSS to run as though it were on a PC computer (see: <https://marksetzler.org/generalissues/SPSS%20for%20Mac%20configuration/SPSSforMacConfiguration.html>)
3. Next, in your syntax, you will need to **select and run** the command code you just pasted into your syntax file (again, specifics are below). To do this, highlight the code you want to run and hit the green button at the top of the screen.
4. Next, **review your output** screen to make sure that everything worked okay and/or to review and/or print your statistical output.
5. **VERY IMPORTANT: Once you are done with each session, you need to make sure that you save your syntax file so that the next time you work with the dataset, you can begin by selecting and running the entire syntax file, which will recreate your modified dataset**. By simply saving your command code as you go, you will not have to keep saving and re-saving altered SPSS datasets. More importantly, you can correct mistakes (maybe it turns out that you should have coded “seniors” as people who are older than 65 rather than 55) with just a few keystrokes. Even if you don’t need to make future changes to the various variables you have recoded, your syntax file will provide you with a record of every coding decision you made in your project, which is something you will need to know when you write the methods section of your project.
6. **JUST AS IMPORTANT: Remember not to save changes to the copy of your original dataset; in other words, do not save changes to the *Data Editor.***  If you save these changes, you will alter your copy of the original dataset permanently. Remember, you will have saved your syntax file, so next time you open things up, you will want to be working with a copy of the original version of your Pew dataset.

**SPSS BASICS 4 (PREPPING A DATASET, STEP 1): keeping just the variables you will be using in your study (i.e. do this before recoding any variables)**

Now that we have covered the basic features of SPSS, let’s look at what you do with a dataset once you have downloaded one that will work for your project. If you are going to be using a big dataset with a lot of variables for a long time, you want to consider a few minutes of initial prep work that will save you a lot of time over the length of the project.

Most students will find it easier if they remove many or even most of the variables in their dataset. If you are the rare lucky ducky that is working with a dataset that has just a couple of dozen variables that all have distinct names and labels (so you don’t mix them up as you are using dropdown menus listing all variable labels), you can skip this step.

For everyone else, perhaps this block of work will strike you as a tedious exercise. However, as you begin to recode and analyze your dataset, you will much appreciate not having to deal with a full list of variables every time you want to run a statistic with SPSS’s point-and-click menus.

We want to drop variables using syntax for a couple of reasons. First, we want a record of every way we alter our original dataset. Also, you will be creating a smaller “working” dataset with syntax so that you can go back and add and drop additional variables as necessary in the future in just a few keystrokes.

* + - 1. This section assumes that you already have:   
         (1) downloaded a dataset that has a survey dataset you are going to analyze,   
         (2) that that dataset is in SPSS format (i.e., it will be a \*.sav file); if not, you will need to import it into SPSS and save it as SPSS,   
         (3) the sav file is not compressed (i.e., it is not in a zipped file); because these files are big, they often are compressed when they are first downloaded, but SPSS will not analyze data when it is compressed, and   
         (4) loaded SPSS onto your computer and have created folders/subfolders for your project/data file in the Documents folder on your computer, where you will consistently access and save your syntax and any smaller versions of your dataset. Having previously created a subfolder with your original dataset and a copy of it is important. **The SPSS syntax you are going to create below will attempt to get and save data for the specific file path you give it.**
      2. Again, before you do anything, make sure you are working with a copy rather than the only version of your original dataset. You need to do all of your work with a copy of the original file to make sure that you don’t have to download your original dataset again if something goes wrong and you save changes to the wrong dataset.
      3. Get ready, by opening up copy\_MyData.sav (i.e., the copy of your dataset” and deciding which variables and observations (more on that in the next section) you want to keep in a smaller version of the dataset**.**
      4. Next, you will need a list with the names of all of the variables you want in the small version of the dataset
         1. The fastest way to generate a list of variable names with no typing mistakes (in which case your syntax won’t run in a few steps) is to use SPSS’s point and click the Descriptives command:  
            Analyze -> Descriptives -> Descriptives.
         2. After running the command, hover over the variables listed on the left side of the window. Using the right-hand mouse button, select the option to display all of the variables by their names (rather than labels), and then use the right-hand button again and sort the variable names alphabetically, so you can quickly find the variables you are looking for.
         3. Move the variables that will go into the small dataset over to the right-hand window.
         4. Hit the paste command, and all of these variables will be listed in your syntax. You don’t need to run the Descriptives command in your syntax. This step was a way to get a list of variables into your syntax without typing or copy-pasting them in individually, which is error-prone.
      5. Now you are ready to save a small version of the dataset   
         1. In the dataset window (not the syntax window) of copy.MyData.sav, select File -> Save As
         2. Browse to the folder where your dataset is and enter a name for the dataset. I recommend keeping the dataset the same, but adding small\_ to the front of it: small\_MyData.sav
         3. Hit the paste button. And go to your syntax file where, at the bottom of the syntax, you will have something that looks like:   
            SAVE OUTFILE='C:\Users\MyName\Documents\SPSSpracticework\small\_MyFile.sav'

/COMPRESSED.

* + - * 1. Now, you need to add a line of code to tell SPSS to keep just the variables you want. That subcommand is Keep =
        2. So, your final code will look something like:  
           SAVE OUTFILE='C:\Users\MyName\Documents\SPSSpracticework\small\_MyFile.sav'

/KEEP= DVvar sex age educ race   
 /COMPRESSED.

* + - * 1. Finally, select only this syntax and run it by pressing the green arrow button at the top of the page. If you look in your data folder, there should now be a small version of the dataset in your data subfolder. You can open and use this smaller dataset for the rest of your project.
        2. Make sure to save your syntax. If you ever need to remove or add additional variables, you can just delete the small dataset and rerun your syntax after modifying the /KEEP= subcommand, adding or removing any variables as necessary. To save syntax, go to the syntax window and File ->Save As. And name this file something that explains what have done with it. For example: Syntax for small dataset.
        3. Finally, save the smaller version of the dataset. Consider labeling it something like MyData.small. If you did that and have followed the suggestions above, you would now have three versions of your dataset: MyData.original, MyData.copy, and MyDataSmall.

**SPSS BASICS 4 (PREPPING A DATASET, STEP 2): removing ANY respondents that you know will not be used in your analyses).**

1. Some students do not need to do this step, but many do. **The most common reason to drop large numbers of observations is when we have a key variable in our study and many respondents have not been asked the relevant question. If so, we want to delete all observations in the dataset that have "missing" data for that variable.**

If we are dealing with a variable that is essential to the study but is not available for a whole bunch of respondents, our study’s descriptive statistics for all of the other variables will analyze lots of respondents who will not be included in any of the bivariate and multivariate analyses. You don’t want your descriptives to say you are analyzing 2,000 people while most of the statistical tables only include 1000 respondents.

In this example, let’s say, I want to remove all respondents who did not answer a question about whether they support making public colleges free to attend because the relevant question was presented to only half of the people taking the survey:  
  
\*This code drops any respondent who has missing data on the variable "free\_college\_good"  
FILTER OFF.  
USE ALL.   
SELECT IF (NOT MISSING(free\_college\_good)).  
EXECUTE.

2. And some student projects involve looking at just a subset of observations. If the point is to compare some group to others—say, you want to know why women do something versus men—you should keep all of the observations. However, if you are using a global attitudes survey and only want to look at people from a few countries, you will want to drop respondents who are not from that country. You can do that in syntax by simply adding a line of code like this:

* \*Here, we’ll keep everyone who isn’t from, say, the US, France, or the UK  
  FILTER OFF.   
  USE ALL.   
  SELECT IF (country = 17) or (country = 2) or (country = 4).   
  EXECUTE.
* You could write these short examples of code directly into your syntax file. Alternatively, you can use the point-and-click interface in SPSS to create this same code for you (which you then will paste into a syntax file):  
  (Step 1) Go to the data editor  
  (Step 2) Then go Data -> Select cases -> If condition is satisfied

(Step 3) Then, code... to get the above, you would click on variables and logic keys or just type what you need to get:  
 (country = 17) or (country = 2) or (country = 4)

(Step 4)  Select the "continue" box, and then on the next screen select  
"Delete unselected cases"

(Step 5) Paste the code into syntax, select the code, and run it by pressing the green arrow button at the top of the screen.

3. Once you have dropped any necessary observations, save your syntax and the small version of the dataset as explained in the last section.

**SPSS BASICS 5: recoding data so that you can use your dataset**

Once you are done making a smaller dataset with a subset of the variables you will want to examine in your study, you can begin the work necessary to make your dataset useable by recoding or creating the variables you will need.   
  
Before you start recoding, it is a good idea to make a copy of the smaller version of your dataset so that any changes you make in coding won’t accidentally save over the work you have done to this point. Consider labeling the small version of your dataset that will have all of your variable coding, your working dataset. If you do this, you will now have four copies of your dataset, with each acting as a backup in case you need to go back and do part of your work again:  
(1) MyData.original, (2) MyData.copy, (3)MyData.small, and (4)MyData.working.

Generally, it also makes sense to have one syntax file documenting all of the changes you made to the original dataset to get to a small one and then a second syntax file where you keep track of all of the work you are doing to create, recode, and label variables and their response categories.

If you are following the strategy just suggested, open a new syntax file, make a note of what this syntax file will do:  
\*This is the syntax where I am recoding and creating variables.

Save the new syntax file with a name that makes sense:

File ->Save As -> name it something like recoding syntax.

To begin recoding, keep this syntax file open and open the working.sav version of your dataset.

1. **It is almost always necessary to recode *all* of the variables that you will be analyzing**. On the one hand, the response categories may be too specific when you would rather just focus on one or two responses (e.g., there are 14 different religious affiliations, but you are only interested in the differences between Catholics and Protestants). On the other hand, you also need to make sure that you recode answers from people who didn’t answer a question in which you are interested so that SPSS doesn’t analyze these answers when generating statistics.
2. In order to recode a variable, **click on “Transform”** and **then select “Recode into a different variable.”** It usually is helpful to create different variables rather than changing the original variables in case you decide that you want to change how you are coding that variable later on.
3. Once the dialogue screen appears, you will see all of the variables in the dataset on the left-hand side. Scroll down to **find the variable you wish to recode** (consult your codebook for help to make sure you get the right variable), **click on it and then click the 🡪 button** in the middle of the screen.
4. At this point you will give the variable a new name and description on the right-hand side of the box and then **click on “Old and New Values.”**
5. You will be taken to a separate screen where you will **enter the old response number/s** (which can be found in the codebook) on the left and **list the new response number/s** on the right. Save yourself time by using the “range,” “else,” and “copy old values” options when you can. **Each time you recode a response value, remember to hit the add button** before moving on to the next old coding value/s to be given a new value. If a coding number (usually 9, 99, or 999 in Pew data) corresponds to an answer of “**don’t know” or “no response” in the codebook, you need to transform that value into “user or system missing”** so that non-answers are not treated like valid answers statistical analyses. **Make sure to double-check to ensure that your recoding encompasses all of the answer values identified in the codebook.**   
     
   Once you are back **on the main recoding page, you need to hit that page’s change button**. Then, **click on the “Paste” button** in order to paste the code for the transform function directly into your syntax file. Once you click paste you will see the command code pop-up on your syntax screen.
6. **There are several reasons to consider recoding your variables beyond what is necessary to deal with “don’t know” or “didn’t answer” responses  
     
   First, recoding is pretty easy.** While it may at first appear cumbersome, it is anything but. From this point forward all you need to do is copy and paste this command into your syntax in order to change the dataset. Furthermore, you will be able see the command language and use it to change future variables:  
     
    RECODE variable name (old value = new value) INTO new variable name  
    EXECUTE.  
      
   Second, **recoding is the easiest way to create a “dummy” (yes/no) variable**s, which are used to identify respondents who are members of groups or to create yes/no dependent variables that are necessary for certain kinds of statistical analyses. If the original coding scheme for the variable *icecream* is “1=loves it, 2=likes, 3=neutral, 4=doesn’t like, 5 =hates it, 9=don’t know/refused,” we may want to create dummy variable that identifies just people who like or love icecream so that we can distinguish them in our analyses from everyone else:

RECODE *icecream* (1=1)(2=1) (9=SYSMIS) (ELSE = 0) INTO *likesIC2*  
EXECUTE.  
(as a side note, I named this variable with a 2 at the end to remind myself that it is a dummy variable with only two values)

Another common type of recoding is to reverse the way an index runs so that our graphs, tables, and measures of association (we’ll talk about these below) make more sense. Let’s reverse the *icecream* variable’s coding so that people who love the stuff are coded with the highest values:

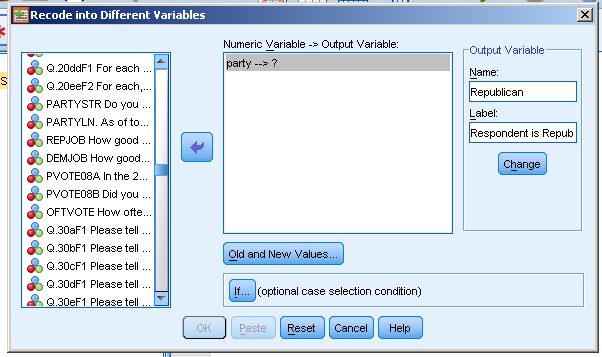
RECODE *icecream* (5=1)(4 = 2)(3=2) (2=3)(1=5)(9=SYSMIS) INTO *likeIC5*  
EXECUTE.

And then label our new variable so that it has a name that makes sense:

VARIABLE LABELS *likeIC5* 'Likes icecream; 1 = hates, 5=loves '.  
 EXECUTE.

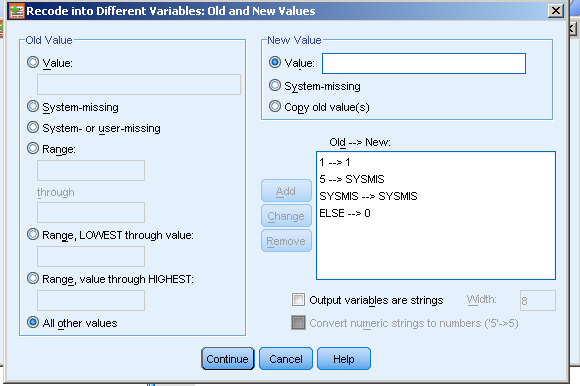
1. When you have completed these steps, highlight the relevant command code and click “Run” and then “Selection” and SPSS will automatically recode all the data for you.

**1. Select the input variable.** Let’s say you want to use the variable “party” to make a new variable that identifies just Republicans. Select “party” from the left-hand list and then hit the arrow button to bring it into the right hand window

1. Let’s give you a visual example:   
   

**3. Enter the coding changes.** Here’s where you put in the coding value/s for your variable as listed in the original codebook. In this case, Republicans in the original survey were coded 1, members of various other parties were coded 2-4 and 7-10, respondents who didn’t belong to any party were coded 5, and those who “didn’t know” or who “refused to answer” were coded 9.   
  
Notice that you can use ranges , “all other values” and “copy old values” options so that you don’t have to enter each coding change by number. These can be a handy options if, say, you want to code a long list of religions into a single variable that separates people who belong to any religion from those who don’t. **Each time you change an old value, you need click on the “add” button** before entering additional old🡪 new values.

2. **Name your new variable and give it a label.** Don’t hit “change: yet. Then click on the “Old and New Values” option.



1. As one last final step in the recoding, you should **label the output values for the variables** that you have just created so that it will be easier to read data output and charts later on. Although there are other ways to do this, you can most quickly label the response values for a new variable by typing them in using syntax command. In the example we will work with below, you would type the following:

VALUE LABELS Republican 1 'Republican' 0 'NonRepublican'.

EXECUTE.

Incidentally, you can use this the *label values* command for a bunch of variables simultaneously:

VALUE LABELS Republican Democrat Senior BornAgain 1 'Yes' 0 'No'.

EXECUTE.

1. **What do you do if you want to recode a variable that has a large number of values with separate labels, and you don’t want to go through the hassle of relabeling them all as described in the previous step?** If you are certain that you aren’t going to want go back and recode the original variable during your current session, you may want to recode “into the same variable” rather than “into different variables.” (If you do this, you need to either make double sure that you remember not to save the changes you are making to original variables at the end of the session or, better yet, you need to make sure that you have saved a backup of your original data. The kind of recoding is very straight-forward as we can see in this example where we are going to recode a 9-level education variable, keeping all of the original labels but telling SPSS to ignore “don’t know” and “refused” answers when calculating statistics:

RECODE educ (10=SYSMIS) (11=SYSMIS).  
EXECUTE.

**CONCLUDING COMMENTS:**

If all you want to do is to figure out some basic univariate statistics or to make sure that nothing has gone wrong , you will be ready to do so at this point by just using the point-and-click interface. The commands that will give you the information you need:

Analyze🡪Descriptive statistics🡪frequencies   
Analyze🡪Descriptive statistics🡪descriptives

To learn more about how we create new variables from a downloaded dataset, proceed to Part 2 of this handout.

To learn more about actually analyzing your data and testing relationships between variables, please proceed to Part 3 of this handout.