Saving Data and Merging Datasets

Saving Data

- Stata makes it easy to save data using the `save` command:

```
save filename
```

- Since Stata always has only one dataset in memory at a time, it is always clear what data we are saving

- Similar to the generate vs. replace commands for variables, Stata only lets you overwrite a file on disk if you use the replace option:

```
save filename, replace
```

Saving Files

- Always important to think about your project organization when saving .dta files
  - What is my current working directory? Use `pwd` command to check you are where you think you are
  - Should I be saving this file in a new folder, or under a new name?

Prepare for Merging

- We will be using some toy Stata datasets that are available via the `webuse` command
  - To start out, we'll load the `autosize` and `autoexpense` datasets

```
webuse autosize, clear

save autosize.dta
```

Overwriting Saved Files

- When overwriting a saved file with a standard `save` command we get a warning, so we use

```
save autosize.dta, replace
```

- Stata users often get so used to the `replace` option that they use it everywhere in their scrips

- This can lead to heartbreak!

Merging Datasets

- In Stata, combining the columns, or variables, of two or more datasets is called “merging” them (another common term for this process is “joining” datasets)

- In order to **`merge`** datasets, they must share at least one common variable (there must be a way to link them together)

- Stata refers to the dataset in memory as `master`, and to the additional dataset(s) as `using`

- The type of link is described using `master` : `using`, for example —

    - 1:1 (one observation in `master` is linked to one observation in `using`)

    - 1:many (one observation in `master` is linked to many observations in `using`)

    - many:1 (many observations in `master` are linked to one observation in `using`)

    - many:many (many observations in `master` are linked to many observations in `using`)

`merge 1:1`

- We will start out with a merge between datasets with a 1:1 variable link

- Let’s create a new .do file and move through this process together:

```
merge 1:1 linking_variable using file_to_merge

merge 1:1 make using autoexpense
```

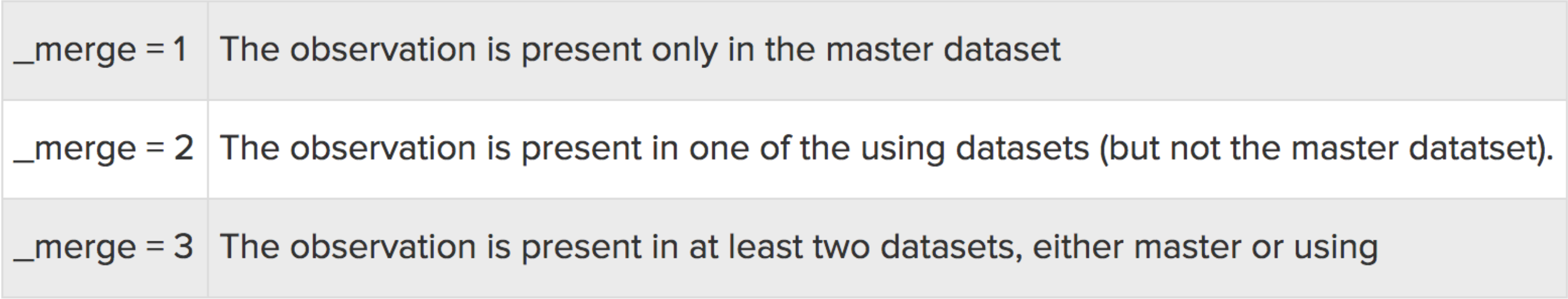
- This will result in a dataset loaded to memory with all the variables in `autosize` and all the variables in `autoexpense`

`merge 1:1`

- In the Stata Results window we will see a merge summary, showing the results of the merge

- Notice the `_merge` variable column:

  - A new variable `_merge` has been added to our dataset. This variable indicates the status of each observation (row) after the merge.



_merge = 1	The observation is present only in the master dataset
_merge = 2	The observation is present in one of the using datasets (but not the master datatset).
_merge = 3	The observation is present in at least two datasets, either master or using

merge 1:1

- In this particular example we see that there is one observation from the `master` dataset that is not matched in the `using` dataset (`Plym. Arrow`)

- We see that the values from the `price` and `mpg` variables from the `autoexpense` dataset are left blank (missing) for this observation

merge, assert

- We can use the assert option with an argument to automatically check on the status of our merge

```
merge 1:1 linking_variable using
file_to_merge, assert(match)

merge 1:1 linking_variable using
file_to_merge, assert(match master)
```

merge 1:m

- The syntax for these merges is largely the same, except for `1:m` replacing `1:1`

```
merge 1:m linking_variable using
file_to_merge
```

- The Stata toy datasets `overlap2` and `overlap1` can be used for this purpose

`merge` options

- To keep only observations with that have matched, use the option `merge, keep(match)`

- To perform a merge without producing the _merge variable, use the option `merge, nogen`

Exercises (1)

1. Auto Data

   A. Save your auto data as auto.dta.

   B. Now split your auto data into two files: save one as *auto_cont.dta* that contains **make** and all **continuous variables,** save the other as *auto_other.dta* that contains **make** and all the **non-continuous variables.**

   C. Open auto_cont and perform a 1:1 merge using auto_other. Save this dataset as auto_merged. How can you make it identical to your original auto.dta file? Do it and save.

   D. Load your original auto.dta file. Replace the values of the headroom variable with the weight variable.

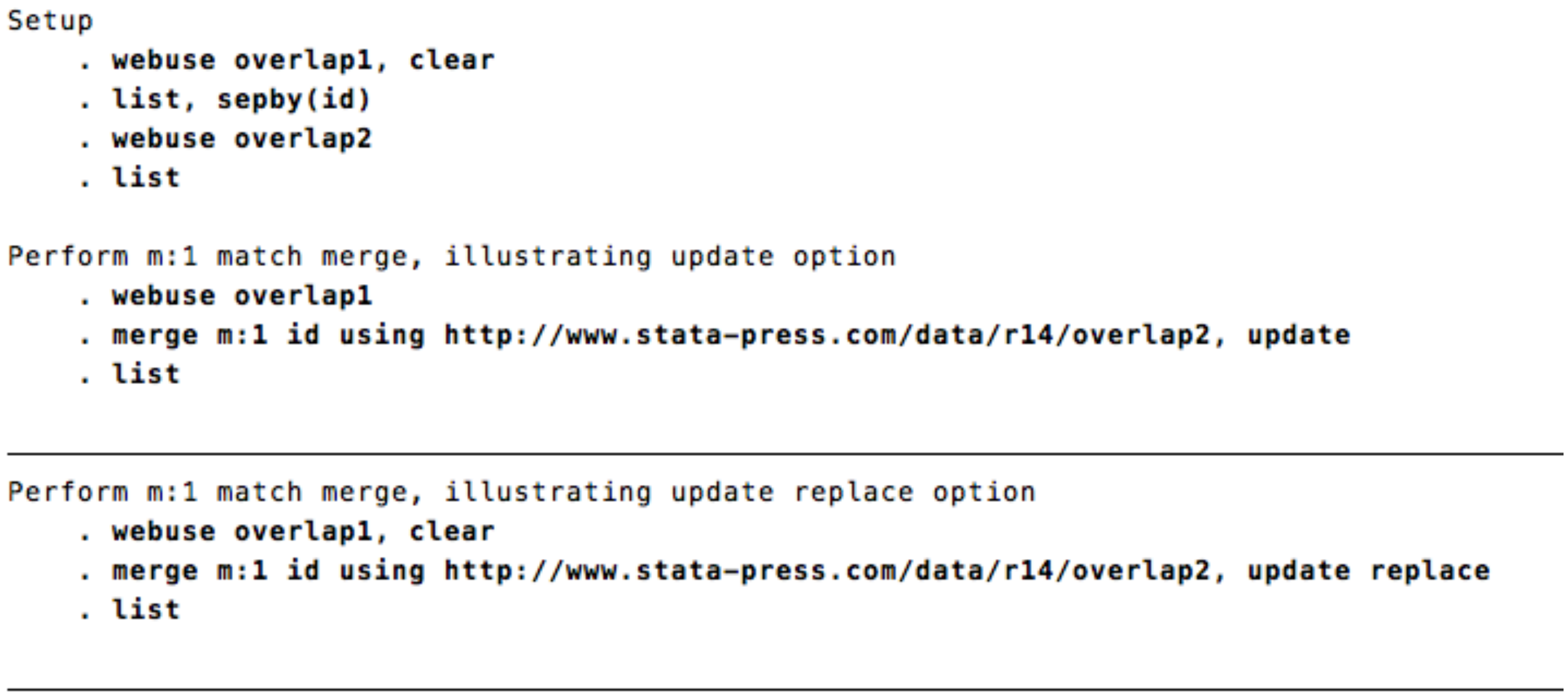
   E. Perform a 1:1 merge with the auto data you currently have in memory with the auto.dta file saved on disk. What happened to headroom? What does this tell you about merging datasets who share variable names?

Exercises (2)

1. m:1 Merging Example

   A. Create a do-file named merging.do

   B. Copy the following series of commands into your do-file. Use comments in your file to explain step-by-step what is happening. Don’t be afraid to explore **help merge!**



```
Setup
    . webuse overlap1, clear
    . list, sepby(id)
    . webuse overlap2
    . list

Perform m:1 match merge, illustrating update option
    . webuse overlap1
    . merge m:1 id using http://www.stata-press.com/data/r14/overlap2, update
    . list
```

```
Perform m:1 match merge, illustrating update replace option
    . webuse overlap1, clear
    . merge m:1 id using http://www.stata-press.com/data/r14/overlap2, update replace
    . list
```