GETTING DATA INTO THE PROGRAM

1. Have a Stata "dta" dataset.

Go to File→ then Open.
OR
Type "use *pathname*" in the command line.

2. Using a SAS or SPSS dataset.

Use Stat Transfer. (Note: do not become dependent on Stat Transfer, you may not always have access to the program and it does not always work).

Use the Transport System

To do this, you must transfer your SAS/SPSS dataset into a *transport* file ("xpt).
In SPPS, simply choose "xpt" from the drop down menu when you save your data file.
In SAS, use the following example:

You have a SAS data file called "thesis.sas7bdat" and it is stored on your D drive. You want to turn it into a transport file, call it "xthesis" and store it in the same place.

```
1. libname mywork 'd:\';
2. libname out xport 'd:\xthesis.xpt';

3. data out.xthesis;
4. set mywork.thesis; run;
```

Line 1 points to the location of your current SAS file you want to convert.
Line 2 points to the location of your new converted file and names if "xthesis.xpt".
Line 3 creates your new transport file by setting it equal to your "thesis" file.

Getting an "xpt" file into Stata.

Use "fdause" import option in Stata
File→import→fda data
Browse to your xpt file.

TIP: Once you think your data is in the system; look at it to make sure it is there!

FAQ'S ABOUT THE WINDOWS

1. How do I make the font bigger?
Click on the little box in the upper left corner of the results window.
Select *Font.*

2. How do I change the color of my screen?
Click on the little box in the upper left corner of the results window.
Select *Preferences.*

3. What are those little push pins?
They let you hide the window (pin to the side), or show (pin straight down).

LOGS, OUTPUTS AND PROGRAMS

"Log" file. It is not like other program, the log file must be created at the beginning of your session. This is how you also keep your output. There are two options—

File → Begin → choose 'log' rather than 'formatted log'. (This will save you a step later).

Your output is generated into a text document that you can then use in excel or word. As you get used to the idea of using the log file this way, you can begin to turn it on and off. For instance, perhaps you want to run a quick model but are not interested in keeping the output. In the command window, you would type:

log off	This suspends the log until you type	*log on*
log close	This turns it completely off.	

"Do" file.
The Stata program is written in a .do file. File definition statements, commands, recodes, etc all can be executed from this document.

"Command" window. This is the smaller window in which you can type single commands. This is handy for quick coding. However, remember that *unless you have a log file open,* there is no record of the coding written in the command window so there is no 'trail' to follow.

"Output" window. The output window only holds so much information, which is why the log file is so important. However, if you want to pull results from the window you can. Just highlight, copy and paste. If you want to make the output window hold the maximum amount of information before it starts clearing then type: *set scrollbufsize 300000*

BASIC COMMANDS

des provides a list of variables (use *ds* and you will get a more manageable list of variables).
cd provides the location of the permanent stata directory
codebook lists information about each variable

nmissing and *npresent* USEFUL shows the number of missing and nonmissing values for the variables. *tabmiss* is actually even better. To use it type *findit tabmiss* into the command line, then click *install.* This takes about 10 seconds.

tab or *table* to create frequency distributions.

A single distribution looks like this:	*tab var1*
A cross tab like this:	*tab var1 var2*
With row, col, cell freq	*tab var1 var2, row col cell*
More than two	*table var1 var2 var3*
Frequencies of more than 2	*tab1 var1 var2 var3 var5-var10*

su displays means

Mean of all variables in the dataset	*su*
Mean of the variable *gpa*	*su gpa*
Median of *gpa*	*su gpa, detail*
Mean of gpa only for girls	*su gpa, if sex==2*

generate or *gen* and *replace* is one way to create a new variable.

An arithmetic expression	*gen rate=time*cost*
	gen loginc=log(inc)
A dummy variable with male=1	*gen male = (sex==1)*

This is the same as above	*gen male=0* *replace male=1 if sex==1*
A series of dummy variables	*tab(race), gen(racedum)*

egen is similar to *gen* in an indescribable way.

Standardizing a variable	*egen zgpa=std(gpa)*
Creating a sum across groups *(think of this as attaching summary statistics to a dataset as a variable)*	*bysort sex: egen grouptotal =total (income)*

dups finds DUPLICATE id numbers. *dups id, key(id)*

This will list how many duplicate ID's are in the dataset, as well as which one's they are. **(This is one of those things you may really want some day, especially if you collect your own data, or are responsible for the input of data).** Again, if not installed already type *findit dups* in the command line.

!= is never equal to
== is always equal to
| is OR
>, < greater than, less than
>=, <= greater than or equal to, less than or equal to

Macro variables

Often in a program, you will want to reference the same set of variables repeatedly. Rather than risk a typo that might leave an important variable out, try using a macro variable.

The code can be run once at the beginning of your program:

global ivars "sex black hisp baby year80 year81 year82"

To reference it later in a regression model for example:

reg dvar $ivars

It is a GOOD IDEA to use these macros when you are running models, especially nested models.

PROGRAMMING FAQ'S

1. Why do I have to keep hitting return to clear my output screen?

set more off Stops Stata from requiring a tap on a key to forward the output screen.

2. Why does Stata tell me I have "no more room…" or "not enough memory to open" the data?

Type this in the command line:

memory this tells you how much is currently allocated
set mem 5m this tells Stata to increase the memory to 5 megabytes.

3. Why is Stata telling me that my variable is already defined?

gen newvar=1 if sex==1	newvar = 1 if sex is equal to 1
gen newvar=0 if sex==0	won't work
replace newvar=0 if sex==0	MUST use 'replace'

SAVING, SORTING, MERGING YOUR DATA

You can save your data as a new file, or replace the existing file by using the *save* and *save as* commands located from the file menu. Alternatively, the code to save a dataset is as follows:

save wkshop (Saves the data to it's current location).
save c:\data\mywork\wkshop.dta (Saves the data to a new location listed in the path).

save wkshop, replace Only use this if you are SURE you want to write over your file.

Data sorting is simple.

sort var1 var2	this sorts the data by var1 and var2

bysort used to sort the variables for a command, similar to a 'by' statement.

Mean gpa by gender	*bysort gender: su gpa*

Adding more observations is just like stacking two datasets together.

use one	this is the first dataset called 'one'
append using two	this will add 'two' to 'one'

Merging data—just like in SAS there must be a common variable(s) on which the observations can be matched, and both datasets must be sorted on this common variable. Stata has a system of a *master* data set and a *using* dataset. (Master is what you start with, using is what you are merging on).

use newdata	this is the Using dataset
sort id	sorted by id number
save tempa	save as a new file called 'tempa'
clear	clears file
use maindata	opens the Master file
sort id	sorted by id number
merge id using tempa	merges the 'tempa' data onto the 'maindata' by id
tab _merge	stata creates a variable called '_merge' which tells you the status of the matches. 1= obs in the master dataset that didn't match using 2= obs in the using dataset that didn't match master 3= obs that match in both datasets
keep if _merge==3	this keeps only those cases in both datasets
save newfile	saves it as a file called 'newfile'

Where are the datasets?
In stata, the data are stored in directories: that is simply a location on your computer. To see where your current directory is located, type in

cd	this lists the path name
pwd	also lists the path name
cd new path here	to change the directory

When you create a temporary file, those files are stored in whichever directory is current at the time.

USING THE SVY COMMANDS

The survey weights are set at the beginning of your program.
In version 9:
Svyset psu_variable [*pweight*=weightvar], *strata (*stratumvar)

Once you assign a series of survey weights to a dataset, they remain and do not have to be reassigned if you save the dataset. **Check the user guide for the dataset you are using for the appropriate weight, psu and strata variable.**

NOTE: in Stata 9, the former 'by' option has been replaced with the 'over' option. For example, to run a mean on the variables *age gpa income* separately for boys and girls the following command is used:

svy:mean age gpa income, over(sex)

This slight difference in coding from version 8 comes into play with the 'subpop' command as well.

svy, subpop (if gender==0): mean age gpa income

OR, if you have already created a 'subpop' variable then you can use it as well. In the following example, the subpop variable that is created refers to those respondents who are ages 16+.

```
gen mysample=0
replace mysample=1 if age >= 16        *creation of subpopulation indicator
svy, subpop(mysample): mean gpa
```

Here is an example of linear regression with survey commands.

svy: reg depvar var1 var2 var3

Be sure to check your output—notice whether the PSU, weight and stratum are correct.

There is a special Stata manual on survey commands that you may check out from the CFDR!

Interaction Expansion
xi is SUPERB for the creation of interaction and dummy terms
Creation of a set of interactions *xi i.race*
(here, race has 3 values—the *xi* command creates an omitted category, and two dummies.)