Introduction to Stata Workshop

Learning objectives of this Session

Finding your way around Stata, basic introduction including different types of data, and different things to save. Creating new variables and labelling them. Tables and descriptive statistics. Dealing with missing data, including warning regarding new variables.

This gives example code, and exercises (with solutions available) for you to see how these commands are used in practice.

Learning objectives of this series of workshops

Reading data into Stata, preparing data for analysis, further data manipulations, merging datasets in many different ways, reshaping datasets, looping in Stata, and extracting saved results into files to produce tables of output.

Further resources complementary to this series

This series teaches most of the material contained in Stata Data Management.doc, referenced SDM. The accompanying Stata commands crib sheet.xls, SCCS, acts as a quick reference guide (and also summarises some data analysis commands).Stata manuals (accessed online and via help) and Stata help itself, are both excellent resources. The manuals teach statistics, as well as Stata, and provide statistics references.

Learning Statistics commands in Stata

Introduction to Statistical thinking and data analysis MPH/ MSc module Imperial College module teaches Stata analysis – t-test, non-parametric equivalents, correlation, calculating odds ratios, chi-squared tests, linear regression, logistic regression, Poisson regression and survival analysis, using linear and categorical predictors. Material can be accessed via blackboard, Imperial students and staff and Imperial honorary staff can request access from Jo Tite. This Statistics course material is not available external to Imperial.

Contents

1 Stata – pros and cons of using Stata

Stata is good at data management, so can cope much better than SPSS with large data sets and many variables. It is much easier to learn how to do data management than R, (in R you might have to programme up from basic commands, rather than having more sophisticated commands to do a lot of the work for you). SAS is somewhat superior if you want to do sophisticated data management, but is a bit harder to learn than Stata and more expensive.

In SPSS, you might lean on the keyboard and accidentally change the data in the data editor. In Stata, you are advised to view the data in a mode that makes this impossible. Even if you do accidentally over-right the data in Stata, commands will come up listing the changes made to the dataset.

There is a menu system for the commands in Stata, so this is sometimes helpful to explore what commands are available, or to see what options are available on specific commands. If you use this system, then Stata still prints out the written command, so you can still save the syntax. This gives you another way of exploring Stata and finding out what it does, and an option for those who want to learn to use Stata quickly.

Commands are relatively simple compared to some packages and pretty flexible. You do nevertheless need to pay a lot of attention to detail in order to learn Stata syntax.

Saving your Stata commands means that you have a record of what you have done. This is crucially important, so this feature is available in all the main software packages. You might need to rerun many of your commands, but with a small change, and then saved commands will save you lots of time. This is also crucial in responding to referees comments, when you might need to do a similar analysis to what you have already done.

The manuals can be used to learn Statistics, so this one help system is a very valuable resource. They are also very good at teaching Stata, which is a benefit of Stata over some other software.

Help is great, good at teaching Stata for specific commands, however, it does still need some practice to use it well. There is also an option to search internet resources for Stata material. This can be useful when you do not know the name of the command that you are looking for.

There are also many Youtube videos from the Stata channel to help you learn. UCLA web page and other resources are great resources.

Graphs are of publication quality. It is simple to produce basic graphs. If motivated to learn, you can do graphs of pretty much any level of complexity, adding text to graphs, combining graphs and adding many features to them.

Some analysis techniques where Stata is useful: meta-analysis, survey data (you can use svyset and then use survey commands), where you have longitudinal data, remember also the survival analysis commands (where you can use stset and st commands – see Stata help and also see practical 9 session/ Stata notes on introductory stats course).

You may become one of the many people who love Stata. I've just heard of a man who loves Stata so much that he named his cat Stata.

2 Detailed Learning Objectives with references to Stata Data Management teaching document

2.1 Navigating the Stata interface.

See SDM chapter 1

2.2 Opening and saving Stata data files.

See SDM chapter 2.1: Changing the Stata working directory and SDM 2.2: Opening a Stata dataset

2.3 Types of data:

numeric (type: int, byte, float, double), and string (e.g. str2, str24) and numeric with value labels (use codebook command to see label values – type typically int or byte). Variables also have labels for a fuller description of what they are.

See SDM chapter 3: viewing your data in Stata, data type and data display formats.

2.4 Basic descriptive commands and graphs:

Summarize, tab, list, browse, describe, count, histogram, scatter. See Stata help for more information.

2.5 If statement at end of commands.

See SDM 5.2 IF statements. The accompanying Stata commands crib sheet.xls (SCCS) contains many examples too.

2.6 Stata help.

See SDM 1.1

2.7 Running commands from do files of Stata commands and saving these files.

See SDM 4.1 do files

2.8 Opening and saving log files of results.

See SDM 4.2 Saving Stata output.

2.9 Saving Stata datasets

See SDM 4.4 saving data in Stata format.

3 Beginners Stata Exercise 1

Absolute beginners are referred to SDM chapter 1 for the Stata layout, and for links to a video tour of Stata, to find you way around the different windows in Stata. Then you may try the following exercises. The references to SDM above give further information if you want on the different commands.

Always save Stata datasets into your own directory, open Stata and then open the file within Stata (do not click on the Stata dataset to open it up, this creates confusion, since you readily end up with several copies of Stata open when using this approach). Please use google chrome to download files from Imperial College blackboard, if you have access to this.

Stata instructions

Change Stata working directory by using the following command in the command window:

cd "H:\introstats\statapracticals" (replace with your own name for the directory that you want to use)

Open the "census4.dta" data set, by clicking on the relevant **"open file" icon** (top left in Stata on toolbar), and selecting the file (or type ***use census4***).

Look at the variables window, to see the list of variables in the dataset.
Look at the review window, to see the commands that you have typed in (including anywhere you used icons to perform the commands).
Click on the "data browse" icon (towards the right of the toolbar) (or type ***browse*** in the command window), and look at your data. Scroll down to see how many states you have in your dataset (there is one line of data per state). ***From the colour and look of the variables, determine the data type of main format of each variable.***
Type ***describe*** or ***descr*** in the command window to see a description of your variables.
(byte/ float/ double/ int are all numeric, str2, str24, str54 are all string variables – no strings here – this is the main distinction that is important). ***Determine the correspondence between this description and the data types determined from the data editor.***
Look also at the variable labels (notice these are also in the variables window).
Notice the ***value label*** against region.
Type ***codebook region*** and see that this gives the numerical codes for the values of this variable and also shows how the variable is displayed.

Type ***tab region*** to give tabulation of region, and compare to frequencies from codebook command. Note output shows value labels rather than numerical coding (unless you type ***tab region, nolabel***.

Explore any other value labels, against other variables.

Type ***tab state*** to point out that it is also possible to use the tab command for string variables, but note that most analysis commands can only be done using numeric data. Type ***Tab state*** and ***tab STATE*** and get error messages to point out that Stata is case sensitive.

Type ***summarize pop*** and see that this gives results for numeric variables.

Type ***summarize region*** to show that this uses the numeric values and then ***summ region*** to demonstrate that this command can be abbreviated, and then ***summarize pop, detail*** noticing what extra information this gives you.
Type ***summ region state*** and see that since observations are not numeric, that no results are shown for state (obs=0, indicates no numeric values).
Type ***summ*** to summarize all variables at the same time.
replace pop=. if pop<0 /* recode impossible population value to missing, coded . (dot) in Stata for numeric variables */
replace divorce=. if divorce==999999 /* 999999 looks like a missing data code for number of divorces, rather than real data, so recode to missing */
** find ***histograms of the data also for more thorough checking of data*** to find anything that looks completely wrong.

Now ***click on the do file icon, to open a do file of commands***. Cut and paste (from review window) the commands that you want to put into your do file (all that did not give error messages). Then click on tools - do menu items within do file, to "do" these commands, i.e. to repeat what you have already done. Type all the following directly into this do file, and run each command in turn (by selecting each command then clicking on the far right icon within the do file). You can also "do" a few at a time – though graphs will only be viewable until you create a further graph.

list pop poplt5 pop5_17 pop18p pop65p popurban // lists variables in the results window
list pop poplt5 pop5_17 pop18p pop65p popurban in 5/10 // lists in the results window – restricted to rows 5 to 10, where each row represents a state
gen poptotal= poplt5+ pop5_17+ pop18p // generate command abbreviated to gen creates a new variable – see it has been added to the bottom of the variables window – we are checking population variable in the data set
label var poptotal "Population total calculated to check current total" // for checking pop variable
list poptotal pop // list in the results window
browse poptotal pop // see variables in the data browser window, can select just limited numbers of variables like this if you want
list if poptotal!=pop // list if these variables are not equal – note not equals symbol in Stata – this checks the population total variable that we created agrees to the one already there – always get in the habit of looking at any newly created variables and checking them as far as possible
list if pop65p<pop18p // list if the population aged over 65 is less than the popn aged over 18 – must always be true
list if pop65p>pop18p // list if the population aged over 65 is greater than the popn aged over 18 – not possible
count if pop65p<pop18p // count the number of observations satisfying this condition – all of them
count // counts all observations in the data set – i.e. counts number of rows in your data set
count if popurban>pop // zero since urban population is the subset of the total population, pop
histogram pop // histogram of pop variable
histogram pop, normal // histogram of pop variable with overlaid Normal curve

help histogram // read options listed (near the start of the help file), and try one out - examples part towards the end of the help file is also worth reading
scatter death medage // scatter plot
gen death_pop=death/pop // creating new variable called death_pop

```
scatter death_pop medage
label var death_pop "Deaths per head of the Population" // labelling newly created variable
scatter death_pop medage  // notice label become axis label on graph
gen marriage_pop = marriage/pop
label var marriage_pop "Marriage per head of the Population"
scatter marriage_pop medage
list if marriage_pop>0.1   // to see characteristics of outlying value
scatter marriage_pop medage if marriage_pop<0.1  // to plot scatter plot excluding the outlying
value
```

Now check all commands are in your do file. ***Save this do file*** with a suitable file name (by clicking on save icon in the do file editor).

Save dataset with new variables selected, as say census4b.dta, so as not to overwrite the original data file: **save census4b**

Close the do file (by clicking on ***X*** on top right of do file window).

Now minimise the do file window, and ***click on the log file icon***. This opens a log file – you need to ***give it a name***, check you are happy with ***directory where it will be saved***, and change to ***.log format***, so that you can edit it as needed. ***Now rerun all the commands from the do file***, so that they are now saved into this log file. [(click on "new do file" icon), then open file icon within the do file window. Select the do file that you have just saved. Click on tools menu in do file, then select "***execute (do) to bottom***" to run all commands from the cursor onwards.

Click on log file icon again, and view the log file. Click on the icon again, and close the log file. Now it can be opened in word or wordpad or other programmes. Now ***cut and paste that command that opens and closes the log file*** into the start and end of this do file. You might want to add "***, replace***" to the open log file to overwrite the previous log file (e.g. full command ***log using session1.log, replace***). There are pros and cons of this approach of automatically overwriting what went before – will you choose this approach?

Optional extra:
If you want to save graphs along with the log, then open the log file in word. Word can be changed to landscape, font changed to courier new, font size 8, then Stata should appear in a good format here. ***Rerun any graph commands*** in Stata. In the ***graph window*** in Stata, click on ***edit- copy graph***. Then go into word, and paste each graph in turn into your word document.
Save the word file, so this now contains results and graphs together.

3.1 Student exercise:

Write all these commands into a do file and run them from the do file as you go along.
Open the version of the census4b.dta data set that you saved earlier, which has a couple of new variables added (if you open using the icon, remember to copy command into do file)

1. Create a new variable, divorces per head of population, and give this an appropriate variable label. Find its mean and its median.
2. Plot graphs, or a few graphs, using this variable (e.g. histogram, scatter plots).
3. Save also the amended data set, with this variable added.

4. Now create a log file of these results – so rerun all the commands from the do file, checking that all commands are included in the do file.
5. help operator to see what operations (+, - and similar) can be used in generate command.
6. help function then click on ***maths functions***, to see what options are available to you

3.2 Optional extra exercise:

7. help scatter – briefly look at the many options available
8. Using Stata menus, click on ***graphics – two way graphs*** – and recreate a scatter plot, adding titles to the graph and to legends, and amending the appearance in other ways as you see fit. Copy the ***command into your do file***. Copy the command within your do file to ***create a similar graph*** of a different variable.
9. Use the help command to look at some of the above commands. Which parts of the help files do you find particularly helpful? Options can be useful and also examples (near the end). Note also links to Stata manuals e.g. [U], [GSM], [D]. Click on one of these to access the manuals. Note what additional explanations, information and references you find here.

3.3 Exercise to help you access a quick reminder of many different commands

10. Look at the Stata Commands Crib sheet.xls, SCCS, and find the commands that you have used.

4 Labels in Stata – labelling datasets, variables and values

See SDM 5.4 Labelling variable names, creating value labels and labelling data sets. Find the relevant commands also on SCCS.

Also see help labels

You may label a dataset, so that you can describe what it contains. Labelling variables and values (for categorical variables) can save loads of time, so that you know what data you have. Labelling categorical variables shows what the different numerical values stand for, and then means that these category names appear in Stata output. Variable labels often appear in Stata output, including as graph legends. This also helps you at the time to interpret your results, and helps you to return to work some months or years later.

Introducing the replace command: Once a variable has been recreated, it can be amended with replace command – generate (gen) is required to create the variable in the first place.

Example code:
** using the census4 data set

```
gen pop_grp=1 if pop<1000000
replace pop_grp=2 if pop>=1000000 & pop<5000000
replace pop_grp=3 if pop>=5000000 /// these 3 lines create a new variable
tab pop_grp, summ(pop)  /// these give some indication of what is created
replace pop_grp=3 if pop>=5000000 & pop<10000000000000  /// these 3 lines create a new variable
tab pop_grp, summ(pop) missing  // these give some indication of what is created
```

```
tabstat pop, stat(n min max mean sd) by(pop_grp) missing /* these allow us to check that the
variable that we have created is correct */
summ pop pop_grp // this shows that we have more non-missing data for pop_grp than for pop
list if pop==. // this shows all data when pop is missing
replace pop_grp=. if pop==. // replaces pop_grp to missing when pop is missing
label var pop_grp "Population by state in categories"  // this labels the newly created variable
label define pop_lbl 1 "Popn <1 million" 2 "Popn 1 million to < 5 million" 3 "Popn >=5 million" /* this
line creates a new label which is called pop_lbl, which gives a correspondence between category
values and category names */
label values pop_grp pop_lbl   // this attaches the appropriate value name to the variable pop_grp
tab pop_grp   // this now shows category names
tab pop_grp, nolabel   // this shows the original numerical category values
label list   // this shows all labels which are in the current Stata memory
codebook pop_grp  // this shows the relevant variable label for the specified variable
label data "Census data with population groups added"
```

Note: use summarize command to keep track of missing data – make sure derived variables have at least as much missing data as original variables.

4.1 Exercises to practice these label commands

1. Create a new categorical variable for number of deaths, with groups of your choosing, from the number of deaths variable.
2. Tabulate this new variable to check that you have correctly created the new variable.
3. Summarize the new and old variables, to check the quantity/ correct treatment of missing data.
4. Label the variable and label the numeric categories appropriately (use label var, label define, and label values statements).
5. Tabulate again, and use appropriate commands to check what you have done.
6. Now repeat for death rate.
7. Create and label a variable for percentage of the population living in urban areas
8. Label the data set and save with the new label attached (label data).

5 Learning Tables and Summary Statistics commands in Stata

This is a fairly comprehensive exploration of the possibilities in Stata. In practice, when you are new to Stata, you don't need all the different versions. SCCS summarises the basic table commands near the start and the more advanced one after the end of the analysis commands.

Using census4 dataset above, demonstration of table commands:

```
summ region
summ region, detail      // options after comma
bys region: summ pop    /* bys can come before several commands to perform them on subgroups
defined by specified variable (by region here) – then command follows in usual syntax  */
tab region, summ(pop) miss   //  options after comma, then specifications for option in brackets

tab region pop_grp, miss    // 2 way frequency table (including row/ col for missing values if present)

tab region pop_grp, miss col    // 2 way frequency table with %ages by column (& missing as above)
tab region pop_grp, miss row cell    /* 2 way frequency table with % by row and % of grand total (&
missing as above) */
```

```
tabstat pop, statistics( mean count min max ) by(region) miss   /* Table of mean, count, min and
max of variable pop, by region, and including counts of missing values */
tab pop_grp region, miss   // produces 2 way table
tab1 pop_grp region    // produces 1 way table for each variable specified
tab1 pop_grp region death   // produces 1 way table for each variable specified
tab2 pop_grp region death    // produces 2 way table for each pair of variables specified
tab pop_grp region, summ(medage)    // summarised medage by values of both pop_grp and region
```

5.1 Stata Channel Video Instructions

I suggest you choose one of these to watch and see if you find it helpful. The first is the simplest, good if you are new to Stata. The last is the most complex. In Stata 13, links to these appear at the bottom of Stata help.

Tables and cross tabulations in Stata
https://www.youtube.com/watch?v=3WpMRtTNZsw

Descriptive statistics in Stata (including tabstat)
https://www.youtube.com/watch?v=kKFbnEWwa2s

Combining cross tabulations and descriptive statistics in Stata:
https://www.youtube.com/watch?v=Dzg6AMSt10w

See help misstable for tables on missing data

5.2 Exercises to practice these table commands

Using dataset nlsw.dta:

1. Find two by two tables of each of race and marital status by union membership.
2. Find tables of each of union membership, race and marital status in turn.
3. Find percentage by race and marital status of union membership categories
4. Find a 3 way tables of union membership, race and marital status.
5. Look at help file for tabstat, and see what statistics are available.
6. Find median, mean and SD of age by marital status
7. Find the 5th and 95th centiles of wage by race
8. Find mean and SD of age by each of race and marital status separately
9. Find mean and SD of age by each combination of race and marital status
10. Find median age by marital status
11. You might find it to explore the Stata menus – tables commands can be found under Statistics – summaries, tables and tests – then both (i) frequency tables and (ii) other tables are useful.

6 Exercises to practice if statements,

Mastering these allows creation of new variables from existing ones and allows you to analyse and produce graphs of subsets of the data

The Stata Commands crib sheet.xls, SCCS, gives several examples of if statement, which you can look at to help you with these exercises. Remember to look out for missing data and deal with them appropriately.

For dates, use the function mdy, which stands for month, day, year. So count if birthdate<mdy(6,1,1960) will count if birthdate is before 1 June 1960 – arguments represent months, then days, then year ***(Note American ordering of dates in this function!)***.

Exercises on if statements:

Read in nlsw Stata data set (download from blackboard first, using google chrome, saving firstly to your chosen directory). Then open Stata, and read your own file in from there.

1. Count if age is 40: Answer: count if age==40 (note double equal signs used in if statements)
2. Count if age is less than 36
3. Count if age is 41 or above
4. Count if age is missing
5. Count if age is 40 and they live in the south
6. Count if usual hours worked is greater than or equal to 50
7. Count if usual hours worked is less than 20
8. Create a histogram of age, in those who have union membership, and change the presentation by using at least one option (e.g. make sure plotted bars for age have sensible starting values).
9. Summarise hours worked amongst union members
10. Count if questionnaire date is before 30 June 2011
11. Count if questionnaire date is after 30 June 2011
12. Count if ethnic group is White British
13. Count if ethnic group is White British or White European
14. Count if ethnic group is White British or White European and if age is less than 40
15. Count if ethnic group is Black African and hours worked are 20 or more
16. Count if ethnic group is Black Caribbean and if the person's industry is mining or construction
17. Tabulate race amongst union members
18. Tabulate age amongst those in the south, with questionnaire dates after 31 July 2011.
19. Tabulate occupation amongst young (<35 years) union members
20. Tabulate race against union membership amongst those over 40 years, with questionnaire date after 30 September 2011.

7 Further Stata resources

Look at the Stata command crib sheet.xls. Many students find it useful to have a crib sheet regularly available to them. You can add commands as you learn them.

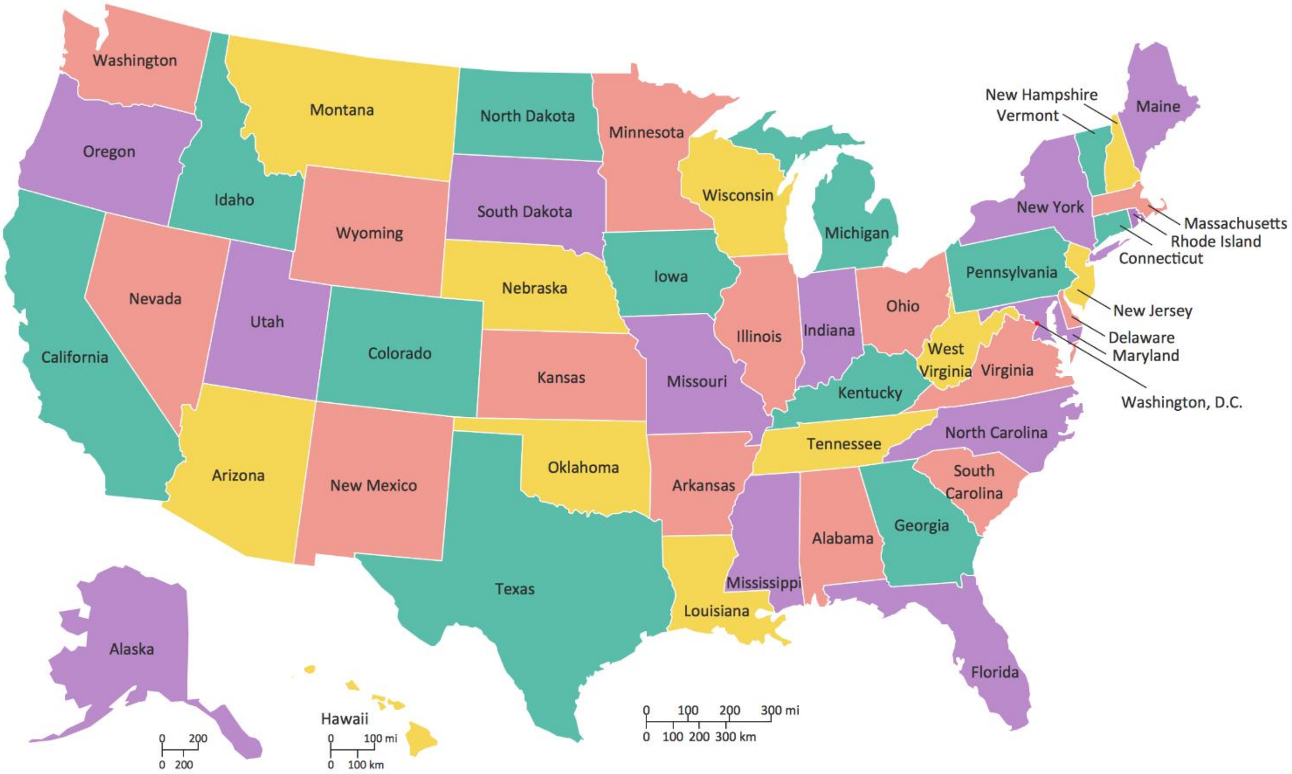
Stata resources - What are pros and cons of each?

- Stata help
- Stata manuals
- Stata youtube videos
- Stata Commands Crib sheet.xls – value of adding regularly to this
- Introduction to Stata Data Management
- Menus in Stata to learn new commands/ find out what is available

8 Extra practice and consolidation

This mainly consolidates the initial exercise, an introduction to Stata. It also encourages use of stata help to learn about a few new commands.

1. Start a new do file and keep all commands in it.
2. Change the directory, and use/ keep a command to open the dataset census4b.dta (you were given the dataset census4.dta and asked to save a version with additional variables added above, called census4b.dta. You can do most of this exercise using census4.dta if you wish).
3. Look at the data and the variable names.
4. Tabulate region on its own, and region against state.
5. Create a new variable, "south", to indicate whether each state is in the North or South of America (use the map below to help you here).
6. See whether the population differs by North/ South divide, using description statistics.
7. Optionally also perform an independent samples t-test to evaluate this further (help ttest should describe how to do this).
8. Compare median age between Northern and Southern states.
9. Optionally use a one sample t-test to compare the median age to 32.
10. Create a new variable, to denote whether the median state age is greater than or less than its average value across states.
11. Create a new variable, divorces per head of population and label the variable.
12. Compare divorces per head according to whether median State age is higher or lower than the average.
13. Plot divorces per head of population against median age.
14. Plot histograms of divorce per head, for each of North and South, on one plot (i.e. beside each other or in one column).
15. Plot a box plot of divorces per head against North/ South. Use help graph box – scroll down to description or to examples.
16. Create a scatter plot of divorces per head against marriages per head.
17. Remove the outlying value from the above plot and recreate the plot.
18. Create a log file of the Stata outcome, by rerunning all the commands in your do file.
19. Word can be changed to landscape, font changed to courier new, font size 8, then Stata should appear in a good format here. You might want to rerun the graphs, and then click on copy graph in the graph window, then paste into your word document.


Washington
Montana
North Dakota
Minnesota
New Hampshire
Vermont
Maine
Oregon
Idaho
Wyoming
South Dakota
Wisconsin
Michigan
New York
Massachusetts
Rhode Island
Connecticut
Nevada
Utah
Colorado
Nebraska
Iowa
Illinois
Indiana
Ohio
Pennsylvania
New Jersey
Delaware
Maryland
Washington, D.C.
California
Kansas
Missouri
Kentucky
West Virginia
Virginia
Arizona
New Mexico
Oklahoma
Arkansas
Tennessee
North Carolina
South Carolina
Texas
Mississippi
Alabama
Georgia
Louisiana
Florida
Alaska
Hawaii
0 200
0 200
0 100 mi
0 100 km
0 100 200 300 mi
0 100 200 300 km