210A Week 1 Notes

Goals of the class

One of the goals of the class is that you'll be prepared to take higher level statistics and quantitative methods classes so that if you choose, you can do quantitative research. However this is a *secondary* goal as realistically using quantitative methods is not for everybody nor should it be – lots of very solid research doesn't involve stats and lots of solid researchers never use stats. The real reason for the course is so that all of you can *read* stats. If you look in *ASR* or *AJS,* at least 2/3 of the articles use statistics. These articles can be difficult to read if you come to them with no methodological background, so if you don't have a basic grasp of stastics you're cut off from reading large parts of our discipline. For instance, you would be hopeless at 239 (stratification) without basic stats background because that literature is so thoroughly quantitative (though it has notable qualitative works, like Edin and Lein). In contrast, qualitative research is easier *to read* than stats, even though it's in some ways harder *to do* or to critique knowledgeably. I've tailored the class to meet both goals but to recognize the primacy of making you informed readers of quantitative research. As such, the use of Stata is optional but recommended.

Samples and Population

In lay usage "statistic" often means "figure." In technical usage these are "parameters" and "statistics" has to do with how much faith we can take that the parameter generalizes to the population. So when you generate a statistic, you get a parameter and a confidence interval (or a p-value, which is related to a confidence interval). Confidence intervals and p-values are *very* misunderstood in particular the totemic significance ascribed to the ".05" threshhold. The only thing a p-value means is how likely it is that your data on some cases (the sample) is to all like cases (the population). That is, it's about induction from samples to populations. This is fairly simple, but all of stats is built up from this so it's important to get it right. Population is the complete set of cases that meet your criteria and is often represented by capital "*N*." The sample is the data that you have and is often represented by lower case "*n*." The important thing is that the sample is representative of the population, an assumption that can be violated numerous ways (but mostly by bad sampling).

There are two surprising things about populations and samples. The first is that n/N (the ratio of sample size to population size) is an almost meaningless bit of information. To a first approximation *only* raw n of sample matters. This is the case regardless of whether you're generalizing to the population of UCLA (about 30,000), Belize (about 300,000), America (about 300 million), or China (about 1.2 billion). The second weird thing about populations is that if you assume that reality is structured by natural law but also subject to chance (and regression analysis implicitly assumes this) then the reality we have is only a sample of all possible realities. This implies that "population" is a much less meaningful concept that it seems to be since even the population we see is only a sample of the infinite number of universes that are subject to the same causal processes as ours.

All this is to say that samples are really important. In practice the only thing that population matters for is that you want the sample to be *representative* of the population. There are two ways to do this, good sampling and weighting. Weighting is when you know that your sample differs (either by design or by accident) from the population in some obvious way. For example if your opinion poll has a lower proportion of blacks than the US Census then you can "population" weight the blacks you do have so each counts more. This

can get sketchy though if you have a reason to suspect that the blacks you got are different from the blacks you missed sampling. (The other major type of weighting is "sample" weighting, which we'll discuss under stratified sampling).

Good sampling is much preferable to rying to fix a bad sample with population weights. The most basic sample is a simple random sample (SRS).[1] This only works occasionally in practice, but it's conceptually simple to think of having a complete list of the population (in our universe). You could then use Excel or Stata to generate random numbers, and base a sample on that. For instance, here's some basic Stata code for generating a sample of 1000 cases from a population list:

```
gen randomnumber=uniform()
sort randomnumber
keep in 1/1000
```

Even if you don't have access to a complete list of the population you can do the equivalent by creating a mechanism where any member of the population has an equal chance of being selected for the sample. So if we assume that every household has exactly one telephone and is as likely to answer the phone, then random digit dialing (ie, having a computer make up phone numbers) is basically an SRS.[2]

A more complicated strategy is stratified sampling. This is when you divide the population into strata (groups), then sample within the strata. This technique is good for "over-sampling" a minority. Remember that only sample size matters. If you are interested not just in estimating the "grand" population's traits or attitudes but also estimates specific to a small minority, then you want to over-sample the minority and get a sample that's half majority and half minority.

Clustered sampling is useful when there is some meaningful structure to the population: students within schools, soldiers within regiments, or households within neighborhoods. To cluster sample you first choose some of the aggregate units (eg, schools) and then within these aggregations randomly choose individuals (eg, students). There are two reasons to do a cluster sample rather than and SRS. The first is simply that sampling and data collection are much more convenient if you have the members of your sample bunched together. The second is that there are some substantive questions (eg, teacher level effects on student achievement) where it makes sense to have multiple observations for each sampled aggregate unit, something that is unlikely to happen much with an SRS.

Variables

The issue of sampling is all about cases, but once you've identified your cases you presumably would like to measure something about them in different variables. The convention is to show data in a spreadsheet where rows are cases identified as "i" and columns are variables identified as "X."[3] By convention there are three types of variables, but I think the third

[1] A note on the word "random" which is very important to statistics and probability. In the technical sense "random" does not mean "arbitrary" or "peculiar" (as in, "this is so random"). In the technical sense "random" means behaving in a way that may be ultimately caused, but which proximately can be treated as being drawn from a probability distributions. A good example from the hard sciences is that thermodynamics has a strong random element even though it is emergent from basically deterministic mechanics.

[2] The assumptions behind random digit dialing were basically valid in the 1980s but much less so now, just as they were not valid in the 1930s.

[3] If you have clustered data the clusters are usually identified as "j." Also note that there are alternative ways to arrange data such as the "field-tagged" format used by some databases, but don't worry about these unless you have to because statistical packages and spreadsheets generally demand row & column data.

category deserves to be unpacked.

Nominal variables are for differences in kind, such as gender, race, or occupation. When there are only two categories (eg, gender) we call nominal variables "binary" or "dummy" variables. When there are multiple categories (eg, race) we call nominal variables "categorical." Regression can't interpret categorical variables directly so we often have to break categorical variables into a dummy set (eg, race is often broken into a black dummy, Hispanic dummy, and Asian dummy, with white being the implicit omitted category). Another way to handle categorical variables is to find some continuous dimension on which you can scale the categories. For instance, occupation is often measured by occupational prestige or synthetic prestige indices like ISEI.

Ordinal variables are those where there is a clear rank order to the categories, but it is not clear how big the gap is between them. For instance, imagine a survey where respondents are allowed to choose one of three options:

1. President Obama was born in Hawaii
2. President Obama's Hawaiian birth certificate is suspicious
3. President Obama is definitely an Indonesian "manchurian candidate"

We can clearly rank these answers in order of how much trust they evidence in the legitimacy of the president's citizenship, with #1 > #2 > #3. However it is not at all clear that the gap between #1 and #2 is comparable in magnitude to the gap between #2 and #3. It seems like once you go from answer #1 to answer #2 you've made a categorical distinction whereas #3 is only slightly more crazy than #2. However it's hard to say exactly how much more crazy so the issue is irreducibly ordinal.

One of the most common forms of ordinal variables are Likert scales, for example:

Consider the statement that the Bush administration and/or Mossad staged the so called "terrorist" attacks on 9/11 with an elaborate conspiracy involving controlled demolitions, etc.

1. Strongly disagree
2. Somewhat disagree
3. Neither agree nor disagree
4. Somewhat agree
5. Strongly agree

Again we can see that it doesn't entirely make sense to treat the distance between each response category as comparable since intuitively there is a categorical distinction between the sane answer (#1) and several crazy answers that differ only marginally in the extent of embracing the lunacy (#2-#5).

Likert scales are extremely common in opinion polls and social science surveys. The most conservative approach is to treat them as ordinal, and some methodologists are adamant about doing this. However this is a hassle analytically so if possible it's nice to recode them. You can recode ordinal variables as continuous if you assume that the gaps between each category are comparable. Likewise you can recode them as dummies if you can draw a "cutpoint" categorical distinction to either side of which answers are comparable, as I did in saying that there is a qualitative distinction between full-fledged embrace of reality and

any consideration of a conspiracy theory. Such a decision can be based on both theoretical and technical issues.

Continuous variables are ones where the rank order is clear and a given interval is consistently meaningful across the scale. For instance, it is meaningful in some ways to say that the difference between $4 and $5 is comparable to that between $5 and $6. The standard textbook definition is to leave it at that, but this elides a distinction that is very important in practice.

Continuous *normal* variables (and roughly normal variables) are ones that are basically symmetrical – aka bell curves. For instance IQ and (within gender) height are distributed normally. If the typical IQ is 100, then there will be similar numbers of people with IQ of 85 and IQ of 115. Normal variables tend to result from causal processes where numerous things are going on and in the aggregate their effects are basically additive. So if you flip 10 quarters and count the number of heads, then repeat this little procedure several hundred times, you'll find that the most common outcome is 5 or 6 heads, it's a little rarer to get 4 or 7 heads, rarer still to get 3 or 8 heads, ..., and only in $\frac{2}{2^{10}}$ cases will you get wither straight tails or straight heads. However note that it's symmetrical so 5 and 6 are equally common whereas 0 and 10 are equally rare. Many basic statistics assume continuous normal data and to a certain extent all other types of data are treated as special cases.

Continuous *count* variables are those with an extreme peak at zero (and no negative values), a few moderate values, and a small number of extremely high values. Things like citation counts for academic articles and box office for motion pictures are distributed as counts. Count variables tend to result from causal processes that involve cumulative advantage, or alternately, where a single failure is catastrophic. So if you flip a single quarter an infinite number of times and count how long are the streaks of heads, the result would be a count, with half the streaks being exactly zero heads (ie, you immediately flip tails), one quarter being exactly one head (ie, you flip heads then tails), one eighth being exactly two heads (ie, flip two heads then a tail), etc, with there being a very small number of extremely long streaks of heads. Although count distributions seem continuous in some deep ways they are more analogous to binary variables.

One way to contrast the two distributions is if you imagine a sports league with a bunch of equally talented teams, the number of wins during the regular season would be distributed as a normal whereas during the sudden-death playoffs the number of wins would follow a count. Also note that there are lots of things that are pure counts or pure normal, but there are also many things that blur the line. The Poisson distribution is technically a count distribution but it's behaves somewhat like a normal (a Poisson looks like a normal that is scrunched together on the left and stretched out to the right).

Using Stata

Stata is a very powerful statistics and database program. Stata is currently the most popular language among sociologists and economists, because it's more flexible and scriptable than SPSS but easier than SAS. [4] The Stata interface consists of:

Menu bar. This is a pretty standard menu bar and provides access to most of Stata's standard features (but not the user-written "ado" files). Most Stata users only use the

[4]Lately R has been increasingly popular. It is more difficult than Stata but more flexible, it is mostly used by people who need to write really cutting edge techniques or people with a principled commitment to open source. For awhile it had better graphs than Stata but Stata graphs are comparable to those of R since version 10.

menu bar interface for graphs and do everything else from the command line.

Tool bar. This bar of icons provides ready access to several commands, mostly having to do with basic file operations. The most useful button is the stop sign, which let's you interrupt a slow program. On the Mac version the menu bar also shows you the present working directory (i.e., where Stata will look for stuff and put stuff if you're not specific).

Review. This window shows you the command log, or all the commands you've enterred in this session. It includes both commands enterred from the command window and those using the menu bar. Incorrectly written commands are in red. You can also access the command history from the "command" window by using the "page up" key.

Results. This window shows the command log and what was returned by the commands. (Note this is different from programs like SAS which tend to keep the command log and results in separate places). The same thing will appear in a "log" file if you are using one, which you should. Also note that unless you enter "set more off," Stata will pause the output every time a single bit of output exceeds the size of the results window.

Variables. This window shows the variables for the data in memory, including some basic facts about these variables.

Command. This is the command-line interface for Stata.

Most Stata commands can be executed through either "command" or the menu bar (aka GUI, or graphic user interface). We are all really used to doing point-and-click but you'll learn pretty quickly that for Stata the command line is *much* easier to use. The GUI is useful when you are learning how a command works, especially for "graph" commands, but I suggest that you read the command-line version and try to learn the command line. The command line version is displayed to your screen and written to your log file even if you use the GUI so looking at old log files (either your own or other people's) is a good way to understand syntax. It's often faster just to type the command than to find the menu.

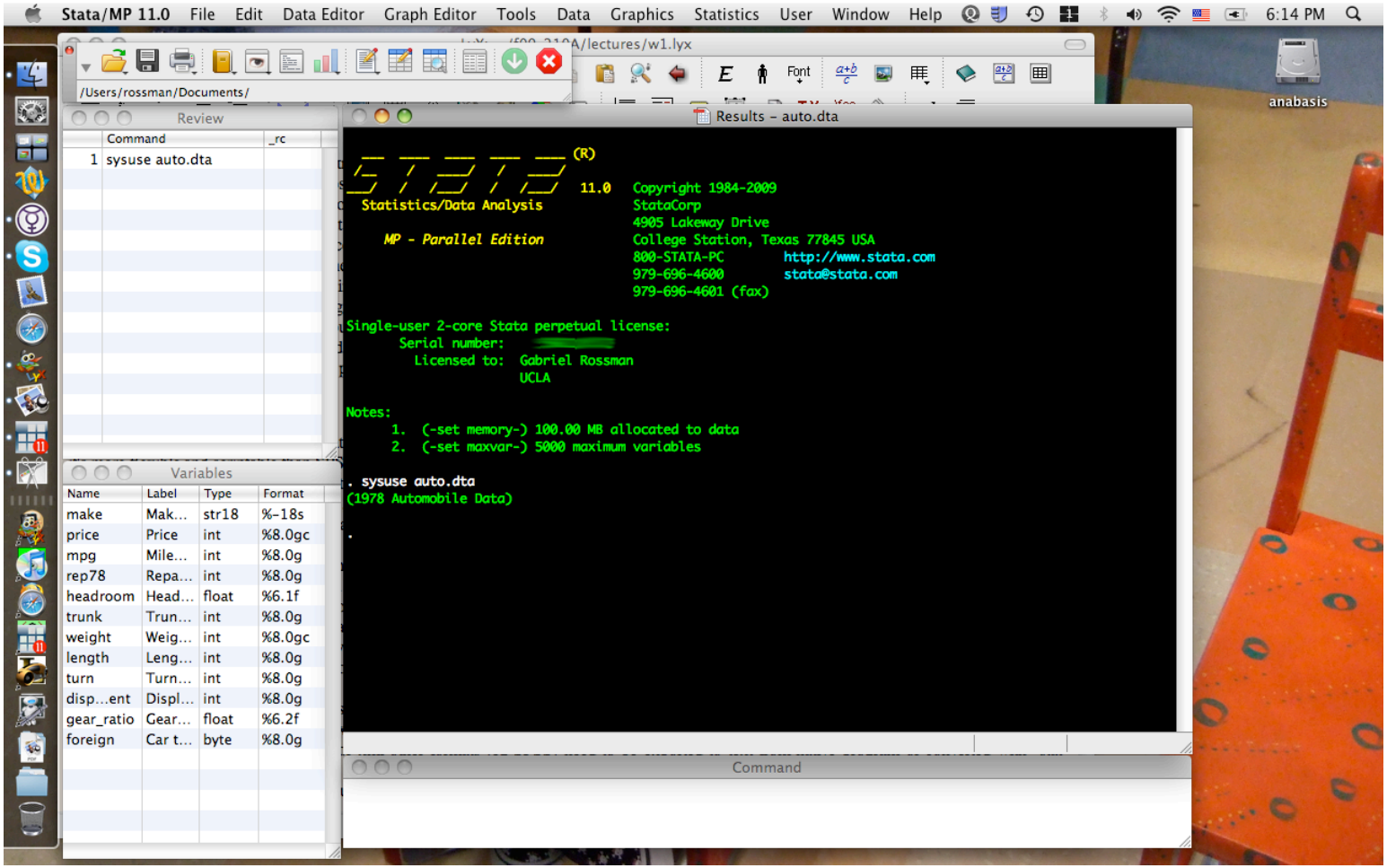
The syntax for most commands is "command objects, options" or if you prefer you can think of it as "verb nouns, adverbs". For example to load the dataset "mydata" you would write:

```
use mydata, clear
```

In this example "use" is the command, "mydata" is the object, and "clear" is an option telling Stata to get rid of any dataset already in memory. Likewise to make a table showing how gender is associated with political party you could write:

```
tab republican female, col nofreq
```

The command is "tab" (an abbreviation of "tabulate"), the objects are the dummy variables "female" and "republican," and the options "nofreq col" say to not show the raw numbers of these combinations (eg, 50 people in the sample are both female and Republican) but to show the column percentages (ie, what percentage of males and of females are Republicans). In many cases you can just use "command" and Stata will make a very good guess as to what objects and options are implied. For instance, the variable description commands


Stata/MP 11.0 File Edit Data Editor Graph Editor Tools Data Graphics Statistics User Window Help 6:14 PM
/Users/rossman/Documents/
Review
Command _rc
1 sysuse auto.dta
Variables
Name Label Type Format
make Mak... str18 %-18s
price Price int %8.0gc
mpg Mile... int %8.0g
rep78 Repa... int %8.0g
headroom Head... float %6.1f
trunk Trun... int %8.0g
weight Weig... int %8.0gc
length Leng... int %8.0g
turn Turn... int %8.0g
disp...ent Displ... int %8.0g
gear_ratio Gear... float %6.2f
foreign Car t... byte %8.0g
Results - auto.dta
(R)
11.0
Statistics/Data Analysis
MP - Parallel Edition
Copyright 1984-2009
StataCorp
4905 Lakeway Drive
College Station, Texas 77845 USA
800-STATA-PC http://www.stata.com
979-696-4600 stata@stata.com
979-696-4601 (fax)
Single-user 2-core Stata perpetual license:
Serial number:
Licensed to: Gabriel Rossman
UCLA
Notes:
1. (-set memory-) 100.00 MB allocated to data
2. (-set maxvar-) 5000 maximum variables
. sysuse auto.dta
(1978 Automobile Data)
.
Command
anabasis

(sum, desc, codebook) will just assume you want every variable. One useful command is "help," which takes as its object another command, eg "help use". This opens up a window briefly explaining the command. (It's similar to the "man" command in Unix). The most useful part of a help file is the list of syntax examples, which are always towards the end. The help files are summaries of the manuals, which beginning with Stata 11 are linked from the help file as PDFs.

The real advantage of the command line is that you can write "do files," which are scripts or text files containing a bunch of commands. Working through do-files means that your work is replicable and you can change a few things without having to actually laboriously redo them one at a time. It also means that if you're doing really processor intensive stuff you can leave your computer running while you make dinner. It's OK to do very exploratory stuff (and your homework) just interactively, but serious research should always be done with do-files so you have a record of what you did. Stata has an integrated do-file editor, which you can invoke from the toolbar (the one that looks like a paper and pencil) or from the menu bar "Window/Do-File Editor/New Do-File." As of Stata 11, Stata for Windows has a really good do-file editor, but if you're using an older version or a different operating system you'll want to write do-files in a good Stata-friendly text editor like Smultron or Notepad++. The main thing you want in your editor is syntax highlighting, which is automatic color-coding to help you understand your code. Here's an example of syntax highlighting in Smultron. Note that there are different colors for key aspects of the syntax: quotes, comments, commands, and macros. This makes it easy to read and spot mistakes (for instance a hanging quote would be obvious because there'd be red text in the wrong places).

```
45  global noisiness          `1'
46  * how bad is our measure of Y, should range 0 (good measure) to
47  *  1 (1 signal: 1 noise), though theoretically it could be even higher
48  global beta_treatment     `2'
49  * how effective is the treatment. should range from -.5 (counter-productive)
50  * to .5 (pretty good), where 0 means no effect
51  disp "noise " float(`1') " -- efficacy " float(`2')
52  clear
53  quietly set obs $nagents
54  gen y_0true=rnormal()
55  gen y_0observed=y_0true + (rnormal()*$noisiness)
56  gen treatment=0
```

I'll walk you through a simple Stata session. Lines beginning with asterix are comments and Stata ignores them

```
*first let's make sure we know where Stata is going to keep and look for files
pwd
*if you don't like the pwd, use the ''cd'' command to choose another location
*  for instance in the lab you might want Stata to do everything off your
*  thumb drive
*now let's keep a log, which is basically the ''results'' window written to disc
log using practicesession.log, replace
*note that stata can log in either plain text (.log) or it's own language (.smcl)
*smcl has syntax highlighting but only Stata can read it
*
*it's often a good idea to change some of the settings, for instance,
*  if you want to just let output scroll by
```

```
set more off
*you should always include ‘‘set more off’’ in a do-file
*
*now let’s load a dataset
*you’d usually use the command ‘‘use’’ for Stata data
*  or ‘‘insheet’’ for text data
*Stata can also read Excel 07 and SAS Transport. For anything else you
*  need Stat Transfer
*however for practice we use ‘‘sysuse’’ for the sample datasets included with Stata.
*also, ’’webuse’’ is useful for loading datasets directly from the internet
sysuse nlsw88, clear
*note that ‘‘Variables’’ finally has stuff in it
*note that unlike R or perl, Stata can only have one dataset in memory at a time
*now that we’ve done our housekeeping and have a dataset in memory,
*  it gets a little more open ended
*we can use ‘‘desc’’ to print the ‘‘variables’’ window into the results window, which
*  is both easier to read and keeps a copy in the ‘‘log’’
desc
*if we want more detail on the variables we can use
sum
*for yet more detail try
sum, detail
*or
codebook
*to do this for just a few variables, include them in the command
sum age race married, detailed
*to see a few cases try ‘‘list’’
list in 1/5
*or open a spreadsheet view with
browse
*we can see the relationship between two nominal variables
tab race married
*or a nominal vs continuous variable
table married, c(m age)
*or two continuous variables
corr age wage
*can also graph a continuous variable
histogram wage
*note that wage follows a Poisson
*
*you can also create new variables either as random noise
gen randomnumber=runiform()
*or based on other variables
gen marriedcollege=0
replace marriedcollege=1 if married==1 & collgrad==1
*note that there are usually several ways to accomplish something
gen marriedcollege2=married*collgrad
tab marriedcollege2
```