Biostatistics 208 Lab #4 1/30/14

The purpose of this lab is to give you practice in using the `regress` and `medeff` commands in Stata to look at confounding and mediation, as well as interpreting regression results with log-transformed outcomes and predictors.

Mike Shlipak at the VA published a paper using HERS data to show that renal insufficiency is a risk factor for cardiovascular events among post-menopausal women with CHD. Motivated by these findings, we will look at the association of BMI with creatinine levels (log-transformed to deal with right skewness), first unadjusted, then adjusting for several demographic and lifestyle confounders, and finally adjusting for log-transformed triglyceride (TG) levels, a measure of dyslipidemia, and a possible mediator of the adverse effects of obesity on kidney function.

Download the dataset `lab4.dta` from the course website. The variables and values in the Stata dataset are already labeled. Some new variables will be needed; code to make them is given below. In addition, we will select observations with complete data on outcomes, predictors, mediators, and covariates; this is to ensure that nested models are run on the same sample. In HERS there isn't much missing data, so this is reasonable. If there were more, we might want to avoid this complete-case approach, which can be inefficient and induce bias. Dave Glidden covers multiple imputation methods for doing this in Biostat IV.

Note: Copying code from this handout and pasting it into the command line will often work (try it), but with some exceptions:

1. Any line of code involving leading single quotes, including `foreach` loops and the `mediate` program used in bootstrapping. You may have to type leading single quotes using the key at the upper left of your keyboard, just below escape; the following single quote is next to the return key.

2. A long command that takes more than one line (e.g., the `bootstrap` command on page 4); Stata will read each line as a separate command. The line separator `///` won't fix this, because it is only allowed in do-files and does not work on the command line.

One good solution is to copy all the code from the handout into a do-file, add the few commands that are missing (e.g., `use lab4, clear`), fix the leading quotes by hand, add a line separator in the bootstrap command, then run commands from the do-file, by highlighting lines of code and pushing the Do button.

In the code on the next page, the `gen` option in the `recode` command is a convenient way to make a new categorical variable from an existing one. It also correctly handles missing values, leaving them missing. The `tab` commands are for checking that the new variables are correct. The value label `noyes`, suitable for indicator variables, is already part of the dataset; you can see all of them using the command `label list`. Remember that the function `log()` computes the natural log; you have to use `log10()` to get the base-10 log.

```
* indicator for being relatively inactive
recode physact 2=1 3/5=0, gen(lessactive)
label variable lessactive "less active than peers"
label values lessactive noyes
tab physact lessactive

* three-level alcohol use variable
recode drnkspwk 0.2/4.99999 = 1 5/max = 2, gen(drinkamt)
label variable drinkamt "alcohol consumption"
label define drinklabel 0 "none" 1 "<5 drinks/week" 2 ">= 5 drinks/week"
label values drinkamt drinklabel
tab drnkspwk drinkamt

* natural log of triglyceride level
gen lntg = log(tgl)

* select for complete data so nested models run on same sample
foreach x in lncr bmi bmi age raceth educ drinkamt lessactive lntg {
     drop if missing(`x')
       }
```

1 Confounding

We will use two regression models to estimate the unadjusted and adjusted associations of BMI with log-transformed creatinine. Recall from last week that we can get the exponentiated coefficients using the option `eform("exp(beta)")`; the quoted string tells Stata how to label the column of coefficients in the `regress` output.

```
* unadjusted model for crude association of BMI with log-creatinine levels
reg lncreat bmi, eform("exp(beta)")
* Relative increase in creatinine associated with 5-kg/m^2 increase in BMI
lincom bmi*5, eform
* Percent increase in creatinine associated with 5-kg/m^2 increase in BMI
nlcom 100*(exp(_b[bmi]*5)-1)

* model adjusting for confounders
regress lncreat bmi age i.raceth educyrs i.drinkamt lessactive, eform("exp(beta)")
* Relative increase in creatinine associated with 5-kg/m^2 increase in BMI
lincom bmi*5, eform
* Percent increase in creatinine associated with 5-kg/m^2 increase in BMI
nlcom 100*(exp(_b[bmi]*5)-1)
```

Question 1. What is the interpretation of the unadjusted and adjusted exponentiated regression coefficients for BMI, also the `lincom` and `nlcom` results?

2 Mediation

To assess mediation of the effect of BMI on creatinine levels by TG, we fit three models

1. regress TG on BMI and confounders
2. regress log creatinine on BMI and confounders
3. regress log creatinine on BMI, TG, and confounders

In the `regress` commands below, the `eform("exp(beta)"` option is omitted so that we get results that are comparable to results calculated later using the `medeff` command.

```
* Model 1
regress lntg bmi age i.raceth educyrs i.drinkamt lessactive
* store coefficient needed to calculate indirect effect
scalar link1 = _b[bmi]
* Percent increase in TG for a 5-kg/m2 increase in BMI
nlcom 100*(exp(_b[bmi]*5)-1)
```

Question 2. Is BMI an independent predictor of TG levels? Or in other words, is there evidence that exposure affects the proposed mediator?

Now we will estimate Models 2 and 3, compute the percent increases in creatinine for a 25% increase in TG, and estimate the indirect effect of BMI via TG.

```
* Model 2
reg lncr bmi age i.raceth educ i.drinkamt lessact
* store coefficient needed to calculate PE
scalar b_overall = _b[bmi]
* Overall BMI effect: percent increase in creatinine for a 5-kg/m^2 increase in
BMI
nlcom 100*(exp(_b[bmi]*5)-1)

* Model 3
reg lncr bmi age i.raceth educ i.drinkamt lessact lntg
* needed to calculate indirect effect
scalar link2 = _b[lntg]
* needed to calculate PE
scalar b_direct = _b[bmi]
* Direct BMI effect: percent increase in creatinine for a 5-kg/m^2 increase in BMI
nlcom 100*(exp(_b[bmi]*5)-1)

* Relative increase in creatinine for a 25% increase in TG
local l125 = log(1.25)
lincom lntg*`l125', eform
* Percent increase in creatinine for a 25% increase in TG
nlcom 100*(exp(_b[lntg]*log(1.25))-1)

* Indirect effect of BMI on creatinine via TG
*    on log-creatinine scale, for comparison to medeff results below
display link1*link2
*    percent increase in creatinine for a 5-kg/m^2 increase in BMI
di 100*(exp(5*link1*link2)-1)
```

Question 3. Are TG levels an independent predictor of creatinine levels? What is the interpretation of the `lincom` and `nlcom` results?

Question 4. Is there convincing evidence for an indirect pathway from BMI to TG to creatinine? Interpret the indirect effect estimate. Does this pathway make biological sense?

Question 5. What is the interpretation of the BMI effect after adjustment for TG?

2.1 PE with bootstrap confidence intervals

Now we will estimate PE and calculate a 95% bootstrap confidence interval using a short user written program. Remember about typing the single quotes. To avoid this, download the program `mediate.do` from the website, and run it (or copy it into your do-file); then that part of the code will be in memory. Also, the long line that calls the bootstrap program has to be typed without a carriage return between `mediator(lntg)` and `covars(....`

```
* Percentage of adjusted BMI effect on log-creatinine explained by TG
scalar pe = (b_overall - b_direct)/b_overall*100
display round(pe, .1)
* Confidence interval for PE using the bootstrap
capture program drop mediate
program define mediate, rclass
      syntax varlist, outcome(varlist) mediator(varlist fv) covars(varlist fv)
      reg `outcome' `varlist' `covars'
      scalar b_overall = _b[`varlist']
      reg `outcome' `varlist' `covars' `mediator'
      scalar b_direct = _b[`varlist']
      return scalar pe = (b_overall - b_direct) / b_overall * 100
      end

* set seed for random number generator to make bootstrap results exactly
replicable
set seed 208
* next 2 lines must be typed without a carriage return
bootstrap pe=r(pe), reps(1000) notable: mediate bmi, outcome(lncr) mediator(lntg)
covars(age i.raceth educ i.drinkamt lessactive)
* get bias corrected bootstrap CI
estat bootstrap, all
```

Question 6. What percentage of the BMI effect on creatinine is explained by TG levels? What do you make of the upper bound of the bootstrap confidence interval?

2.2 Mediation analysis using `medeff`

Type `findit mediation` on the command line, scroll down in the viewer window, click on the link below that will appear somewhere in the returned results:

`mediation from http://fmwww.bc.edu/RePEc/bocode/m`

This link will download and install the `medeff` and `medsens` commands and help files for future use in Stata. Then use the following code to run the analysis.

The first few lines calculate indicator variables for the non-reference levels of race-ethnicity and drinking. These are needed because `medeff` does not allow factor variables (i.e., use of i. with multilevel categorical variables). In addition, `medeff` does not have an `eform` option, so we only get the indirect and direct effect estimates on the log-creatinine scale; however, we could calculate the back-transformed values using `display` commands.

```
foreach x in raceth drinkamt {
      qui tab `x', gen(`x'_)
      drop `x'_1
      }
medeff (regress lntg bmi age raceth_* educ drinkamt_* lessact) /// // model 1
       (regress lncr bmi age raceth_* educ drinkamt_* lessact lntg), /// // model 3
       mediate(lntg) treat(bmi) sims(2000)
```

Here's the command again, without comments or line breaks:

```
medeff (regress lntg bmi age raceth_* educ drinkamt_* lessact) (regress lncr bmi
age raceth_* educ drinkamt_* lessact lntg), mediate(lntg) treat(bmi) sims(2000)
```

Question 7. How do the `medeff` results compare to the earlier ones?