

Final Exam, Question 4 Script

Stats 506, Fall 2019

12/16/2019

```
* Final Exam, Question 4
* Stats 506, Fall 2019
*
* This Stata script reads in NHANES demographic data (DEMO_D) and
* audiometry data (AUX_D) and merges them together. It then uses the merged
* data to compute t-statistics comparing the audible volume at which
* younger people (25 years old or younger) and older people can hear at various
* sounds frequencies or pitches.
*
* Date: December 15, 2019
* Author: James Henderson
* 80: -----

// Import and save AUX_D data: -----
fdause AUX_D.XPT, clear
quietly compress
gsort +seqn
save AUX_D.dta, replace

//Import DEMO_D and merge AUX_D: -----
fdause DEMO_D.XPT, clear
quietly compress
gsort +seqn
merge 1:1 seqn using AUX_D.dta

// Reduce to matched data
keep if _merge==3
save AUX_DEMO_merge.dta, replace

// Reduce to the variables of interest: -----
keep seqn riagendr ridageyr auxu*

// We will ignore the '2nd read'
drop *1k2*

// Decode missing values: -----
mvdecode _all, mv(888=.\666=.)

// Create age groups: -----
generate byte young = 0
replace young = 1 if ridageyr <= 25

generate byte old = 0
replace old = 1 if ridageyr >= 50 & ridageyr != .

// Save for repeated use: -----
save aud_clean.dta, replace
```

```

*****
** compare old and young by ear **
*****

// reshape to long after dropping cases with all missing values: -----
egen nmiss = rowmiss(auxu*)
drop if nmiss==14

// we have to specify that j is a string
reshape long auxu, i(seqn) j(thresh, string)

// Use a regular expression to extract left ear and frequency from "thresh": --

// left
generate left = 0
replace left = 1 if regexm(thresh,"l$")

// frequency
generate freq = .5 if regexm(thresh, "^500")
replace freq = 1 if regexm(thresh, "^1k")
replace freq = 2 if regexm(thresh, "^2k")
replace freq = 3 if regexm(thresh, "^3k")
replace freq = 4 if regexm(thresh, "^4k")
replace freq = 6 if regexm(thresh, "^6k")
replace freq = 8 if regexm(thresh, "^8k")

// compute t-statistics comparing young to old for each ear/frequency: -----
// compute mean, sd, and count within each group
preserve
collapse (mean) mean=auxu (sd) sd=auxu (count) N=auxu, by(freq old left)

// reshape to compute diffs after grouping
generate group = 8*left + freq
reshape wide mean sd N, i(group) j(old)
gsort freq left

// compute difference and t
generate difference = mean1 - mean0
generate se = sqrt(sd1^2/N1 + sd0^2/N0) // unpooled as more variance for older
generate lower = difference - 1.96*se
generate upper = difference + 1.96*se

// produce a nicely formatted table: -----
drop group sd0 sd1
generate Avg_Young = mean0, after(freq) // after indicates column position
generate Avg_Older = mean1, after(Avg_Young)
generate Ear = "left" if left==1, after(freq)
replace Ear = "right" if left==0
rename freq Frequency
drop mean0 N0 mean1 N1 left se

foreach var in Avg_Young-upper {
  format `var' %3.1f
}

// print the table
list

```