EPIB 613 Thursday November 1, 2012 Stata Session 5

Generate and Replace

```
sysuse auto
gen kpl = mpg*0.425144
label var kpl "km per liter"
replace mpg = 20 in 4
replace mpg = 20 if make == "Buick Century"
replace rep78 = . if make == "AMC Spirit"
gen guzzler = .
replace guzzler = 1 if kpl >= 8.5 & kpl < .
replace guzzler = 0 if kpl <  8.5
gen guzzler2 = (kpl >= 8.5 & kpl < .)
compare guzzler guzzler2
gen guzzler2 = (kpl >= 8.5 & !missing(kpl))
gen guzzler2 = (kpl >= 8.5) if !missing(kpl))    * better, I think
```

Functions

```
gen y = round(x)
gen y = round(x, 0.25)
gen y = int(x)
gen y = floor(x)            * rounding down
gen y = ceil(x)             * rounding up
gen y = mod(x1, x2)         * remainder after division
gen y = abs(x)
gen y = sign(x)
```

```
gen y = exp(x)
gen y = ln(x)
gen y = log10(x)
gen y = logit(x)
gen p = invlogit(x)
gen y = sqrt(x)
gen y = max(x1, x2, x3…xn)
gen y = min(x1, x2, x3…xn)
gen y = sum(x)                    * sum from first to index observation

display chi2tail(1, 3.84)
display invchi2tail(1, 0.05)
display normal(-1.96)
display invnormal(0.025)
display ttail(20,2.09)
display invttail(20,0.025)

gen y = _n
gen y = _N
gen y = runiform()
help sin()
```

Generating Constants

```
egen meankpl = mean(kpl)
by foreign: egen meankpl = mean(kpl)
```

```
egen medkpl = med(kpl)
egen sumkpl = total(kpl)
egen maxkpl = max(kpl)
egen minkpl = min(kpl)
egen validkpl = count(kpl)
```

Summarizing Across Rows

```
egen qmin = rowmin(q1-q17)
egen qmax = rowmax(q1-q17)
egen qmean = rowmean(q1-q17)
egen qmed = rowmedian(q1-q17)
egen qsum = rowtotal(q1-q17)
```

Categorizing Variables

```
egen kpl_cat = cut(kpl), at(5 7(2)17 20)
tab kpl_cat
egen kpl_cat = cut(kpl), at(5 7(2)17 20) label
tab kpl_cat
tab kpl_cat, nolab
drop kpl_cat
egen kpl_cat = cut(kpl), group(4) label
tab kpl_cat
tab kpl_cat, nolab
```

Recoding Variables

```
recode foreign (1=1)(0=2), gen(new_for)
tab for new_for, nolab
drop new_for
recode foreign (1=1 “Foreign”)(0=2 “Domestic”), gen(new_for)
tab for new_for

recode kpl (15/max=3 “15+”)(10/15=2 “10-15”)(min/10=1 “5-10”), gen(kplcat)
tab kplcat
numlabel, add
tab kplcat

tabstat kpl, by(kplcat) stat(min max)
```

Explicit Subscripting (only use on RHS of an expression)

kpl	* current observation
kpl[_n]	* current observation
kpl[1]	* first observation
kpl[_N]	* last observation
kpl[_n-1]	* previous observation
kpl[_n+1]	* next observation
kpl[27]	* 27th observation

```
gen x = kpl[_n-1]
list kpl x in 1/10
```