Stata Recitation - Week 5 - Modifying Data II

McCourt School of Public Policy, Georgetown University

Key Ideas:

- Use if-statements
- Create indicator variables
- Verify results

If statements

- We have seen if statements in passing - now we will cover them thoroughly
- If statements restrict commands, making them act on a portion of the data set.

```
clear
sysuse auto
```

Basic usage

- Summary stats for foreign cars

```
sum weight length mpg if foreign==1
```

- Can use several different logical operators

  ```
  help operators
  ```

- Summary stats for domestic cars

  ```
  sum weight length mpg if foreign!=1
  ```

- If statements work with string variables, but require quotes

  ```
  list make weight length mpg if make=="Buick Riviera"
  ```

- Can make complex conditions with and/or
- Summary stats for heavy domestic cars

  ```
  sum weight length mpg if foreign==0 & weight>=3317
  ```

- Summary stats for light or short domestic cars

  ```
  sum weight length mpg if foreign==0 & (weight<=3317 | length<=196)
  ```

Ways to go wrong with if statements

1. Missing values

```
tab rep78
list make rep78 if rep78>4
list make rep78 if rep78>999999
```

- Missing values are the biggest numbers Stata can hold.
- If you don't want to include them:

  ```
  list make rep78 if rep78>4 & rep78<.
  ```

2. Complex conditions without parentheses

- domestic cars that are light or short

  ```
  tab foreign if foreign==0 & (weight<=3317 | length<=196)
  ```

- domestic cars that are light plus all short cars

  ```
  tab foreign if foreign==0 & weight<=3317 | length<=196
  ```

  *** Always use parentheses when mixing if/and conditions ***

3. Equal statements with non-integers

- Find the car with the biggest gear ratio:

  ```
  sum gear_ratio
  list make gear_ratio if gear_ratio==3.89
  list make gear_ratio if gear_ratio>3.88999 & gear_ratio<3.89001
  describe
  ```

- Any variable that has decimal values may have a hidden `.00000000001`,
- or some similar very small deviation that will make it not `==`
- Don't use `==` with decimal valued variables

Generating variables with if statments

Most common usage is indicator variables

- Create an indicator for lowprice cars “‘ sum price gen lowprice = 0 replace lowprice = 1 if price <= 6000

browse make price lowprice sum price if lowprice==1 sum price if lowprice==0

```
* To Check: Look at max and min for both summarize results
```

```
* Create an indicator for low rep78
```

tab rep78 gen lowrep78 = 0 replace lowrep78 = 1 if rep78<=3

```
* Use two-way tab to verify results
```

tab rep78 lowrep78

```
* That looks good, but what about missing values?
```

tab rep78 lowrep78 , missing

```
* Missing values were set to zero in initial statement, and never changed
* We need one more case:
```

replace lowrep = . if rep78==.

```
#### Whenever you create an indicator, you need to consider three cases:
1. When should the indicator equal 0
2. When should the indicator equal 1
3. When should the indicator equal .

#### Always verify results:
 - Use summarize for continuous variables
 - Use twoway tab with missing option for categorical/discrete variables

* - When (not if) you find mistakes, fix them where the variable was created,
* - not where you found the mistake.
```

```
## Many ways to construct indicator variables ...
* Create an indicator that equals 1 for all cars that have mpg between 20-29


### Specify each possible value
```

sysuse auto.dta, clear

gen midmpg = 0 replace midmpg = . if mpg==. replace midmpg = 1 if mpg==20 replace midmpg = 1 if mpg==21 replace midmpg = 1 if mpg==22 replace midmpg = 1 if mpg==23 replace midmpg = 1 if mpg==24 replace midmpg = 1 if mpg==25 replace midmpg = 1 if mpg==26 replace midmpg = 1 if mpg==27 replace midmpg = 1 if mpg==28 replace midmpg = 1 if mpg==29

tab mpg midmpg, missing

```
### Specify each possible value using `inlist()` function
```

sysuse auto.dta, clear

gen midmpg = 0 replace midmpg = . if mpg==. replace midmpg = 1 if inlist(mpg,20,21,22,23,24,25,26,27,28,29)

tab mpg midmpg, missing

```
### Specify a range
```

sysuse auto.dta, clear

gen midmpg = 0 replace midmpg = . if mpg==. replace midmpg = 1 if mpg>=20 & mpg<30

tab mpg midmpg, missing

```
### Specify a range using `inrange()` function
```

sysuse auto.dta, clear

gen midmpg = 0 replace midmpg = . if mpg==. replace midmpg = 1 if inrange(mpg,20,29)

tab mpg midmpg, missing

```
* Use recode command
```

sysuse auto.dta, clear

recode mpg (0/19 =0) (20/29 =1) (30/max =0) (.=.), gen(midmpg)

tab mpg midmpg, missing “‘