# Stata Recitation - Week 7 - ttests and Strings

McCourt School of Public Policy, Georgetown University

## Key Ideas:

- Calculate confidence intervals from data or from components (mean, sd, n).
- Calculate t-tests for single, paired, pooled, and non-pooled samples.
- Calculate tests for population proportions.
- Perform one and two mean t-tests
- Use string functions
- Destring variables

## Confidence Intervals

```
help ci
```

- Syntax for ci: `ci var1name var2name`
- `level()` option, sets confidence level for ci: ci var1name var2name, level(99)
- Note where defaults are given (i.e. default for level is 95%)

**Example: Compute 90% confidence intervals**

```
clear
sysuse auto
ci weight length mpg , level(90)
```

## Imediate command for confidence intervals

- Syntax for cii, immediate command: `cii #obs mean sd`
- Kind of like a calculator for CI, can use to confirm your results of computing the formula
- Immediate commands are based on the inputs you type into the command.
- They are not dependent on the data that is in memory.
- You can clear current data set and get the same result

**Example: Recompute 90% confidence intervals from components**

- Careful about standard error vs. standard deviation
- – **Don't use st. err. reported by ci, instead summarize to get st. dev.**

```
* weight
sum weight
cii 74 3019.459 777.1936 , level(90)

* length
sum length
cii 74 187.9324 22.26634 , level(90)
```

## T-Test

```
help ttest
```

- Types of Tests:

1. One sample test: One variable.
2. Paired test: Two variables, no options.
3. Unpaired, Pooled test:
   - One variable with `by()` option,
   - or two variables with unpaired option.
4. Unpaired, Nonpooled test:
   - One variable with `by()` option and unequal option,
   - or two variables with unpaired and unequal options

- For 3. and 4. data will usually be set up for the `by()` option version, not the two variable version.

**Example 1: One-sample mean-comparison test**

```
clear
sysuse auto
ttest mpg==20
```

**Review output:**

- Null Hypothesis
- Test statistic
- Degrees of freedom
- Alternative hypotheses for one and two-sided tests
- P-values for each alternative hypothesis

**Example 2: Unpaired, Pooled test**

- Different units sampled for each group, - not paired
- But, groups are from the same population (theoretical) - same standard deviation

```
clear
webuse fuel3
ttest mpg, by(treated)
```

**Example 3: Unpaired, unpooled test**

- Different units were sampled for each group - not paired
- Groups may not be from the same population (theoretical) - standard deviations may be different.

```
ttest mpg, by(treated) unequal
```

**Example 4: Paired test (Rarely Used)**

- Two outcomes from each unit of analysis, e.g. person, state, car.

```
clear
webuse fuel
```

- Assumption is that each line is a single car, tested with two types of fuel additive.
- That is why we use the paired test.

```
ttest mpg1==mpg2
```

### Proportion Tests

- Test population proportions: prtest - used for binary outcomes
- Examples:

```
clear
sysuse nlsw88.dta
tab married
tab union
tab collgrad
```

- Is the proportion of college graduates different for married and non-married populations?

```
prtest collgrad, by(married)
```

- Is it different for union and non-union populations?

```
prtest collgrad, by(union)
```

## In Class Activity 1

PRACTICE using the `nlsw88.dta` example dataset

1. Test the null hypothesis that the true average value of wage is 7.60. What is the probability of seeing the data in this sample given that the average wage is actually 7.60, with a two-tailed test?
2. Report the 90 percent confidence interval for the average value of hours.
3. Test the hypothesis that wages are equal for married and non-married respondents. Do not assume that the two groups have equal variances.
4. Test the hypothesis that wages are equal for union and non-union workers. Assume that the two groups have equal variances.

```
* Load Data
sysuse nlsw88.dta, clear


*1
ttest wage == 7.60
* Ans: p-value for two sided test is 0.1694

*2
```

```
ci hours , level(90)
* Ans: 90% confidence interval for the average value of hours is :  36.85289 to 37.58333

*3
ttest wage , by(married) unequal
* Ans: P-value for the two-sided test is 0.0652

*4
ttest wage , by(union)
*Ans: P-value for the two-sided test less than 0.001
```

## Strings

- We have seen strings, but we haven't really worked with them.

**String values always go in quotes**

**Example: Look at key variables for a single vehicle**

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

- Strings are case sensitive

  ```
  list make mpg weight length if make=="buick century"
  ```

**Example 2: Create an indicator for all Buick vehicles**

- Remember to account for missing values
- Missing value for string variables is an empty string, ""

```
gen buick=0
replace buick=1 if inlist(make , "Buick Century" , "Buick Electra" , "Buick LeSabre" ,///
"Buick Opel" , "Buick Regal" , "Buick Riviera" , "Buick Skylark")

* Or use regular expression (for reference)
generate Buick=regexm(make, "^Buick")

* Don't forget missing values!
replace buick=. if make==""

list make mpg weight length if buick==1
```

**String Functions**

```
help string functions
```

- String functions request string inputs (s, s1, s2, etc.)
- These can be actual strings or the names of string variables.
- Strings should be in quotes, string variables should not be in quotes.

**Example: length(s)**

```
clear
sysuse auto

* actual string, use quotes
gen len_1 = length("test")
browse make len_*

* variable, no quotes
gen len_2 = length(make)
browse make len_*

* actual string, use quotes
gen len_3 = length("make")
browse make len_*

* variable that doesn't exist -> error
gen len_4 = length(test)
```

**Example 2: Create an indicator for all Buick vehicles**

```
gen make1 = word(make,1)
browse make make1

gen buick=0
replace buick=. if make1==""
replace buick=1 if make1=="Buick"
```

# In Class Activity 2

PRACTICE using the `auto.dta` example dataset

1. What is the longest car(s)? What is the car(s) with the longest name?

2. What percentage of domestic car observations are Chevrolets?
3. Create a new string variable Excellent_Cars that adds the word EXCELLENT to the end of cars you think are excellent.

```
* Load Data
sysuse auto.dta, clear
*1
*Lincoln Continental
//here's how i do the longest car
*sort method - will only return one value
sort length
list make in -1

*sum method - will return all values matching the largest value
sum length
list make if length == r(max)
// and here are the cars with the longest name
 Chev. Monte Carlo and  Linc. Continental
gen lengthName = length(make)
sum lengthName
list make if length(make) ==17
*note that I don't have to worry about hidden decimal places as the length() funciton return

*2
*so I want the number of Chevrolet cars (Chev)
count if regexm(make, "^Chev")
display 6/74

*3
gen Excellent_Cars = ""
sum mpg, detail
replace Excellent_Cars= make + " EXCELLENT" if mpg > r(p95)
```

**Converting between strings and numbers**

- You probably won't need these commands until thesis, so we won't cover them in depth at this time. Just know that they exist, and how to find help on them.

**'Destring/tostring'**

- Sometimes a variable is stored as a string when it should be a number.
- This is a frequent problem after importing data from Excel.
- You can change the data in Excel or

- You can convert the string to a number using destring.

```
help destring
```

- See examples in the help file and manual.