

Notes

Week 1

Introduction to gretl

Data edits:

- Dataset used: Country data.gdt
- Load with `ctrl+o`
- Rename your Descriptive label for easy interpretation each Variable name to remember what data a variable contains.
 - Left click on variable -> Variable -> Edit attributes
 - Edit the gdp description to GDP per capita USD from GDP per capital
- To see what commands the GUI introduces, go to Tools -> Command log
- Can highlight multiple variable with `ctrl+left` click or `shift+up` and `down` cursor
- Highlight country and gdp -> right click -> Display values
- View -> Icon view -> double click on Data set to see the spreadsheet view
- File -> Session files -> Save session to save results output.

gretl scripting:

- Scripting not required for the course, but is cool
- Commenting is good: starts with `#`
- File -> Script files -> New script -> gretl script
- `ctrl+r` / gears button to Run script
- Can also highlight and run just the highlighted script
- Example first command:

```
eval 5+5
```

- Doesn't care about spaces:

```
eval 5      +    5
```

- But is case-sensitive:

```
- Eval 5+5
```

- Other examples

```
eval 5^2
```

```
eval log(1) # Natural log
```

```
eval sqrt(25)
```

```
eval exp(1) # e^1, the inverse of the natural log
```

```
# I want a new variable that is equal to the natural log of gdp
```

```
# Generate a variable named lndp; lower case letters for
```

```
# convention, but don't use periods; underscores are ok; don't start with
```

```
# numeric;
```

```
# Creates a new variable in the dataset
```

```
genr lngdp = ln(gdp) # ln is same as log, which is natural log
```

```
genr urban_prop = urban / 100
```

```
genr constant10 = 10 # Doesn't generate a new variable
```

```
list todelete = lngdp urban_prop constant10 # New list
```

```
delete todelete # deletes the list
```

```

# More specific than genr:
series lngdp = ln(gdp)
scalar constant10 = 1

rename lngdp ln_gdpd # Renames a variable

print ln_gdp country # Display the series values

print ln_gdp country --byobs # Display the series values, by observation

# We are interested in the relationship between female life expectancy and
# doctor availability (note the UNITS OF MEASUREMENT: Doctors per 10000 people)

```

- ^ Be more specific than genr if can be
- ctrl+left click lifeexpf and docs -> View -> Summary statistics -> Ok -> Show full statistics:

	Mean	Median	Minimum	Maximum
lifeexpf	66.311	68.000	41.000	83.000
docs	10.521	6.3052	0.18800	42.918

	Std. Dev.	C.V.	Skewness	Ex. kurtosis
lifeexpf	11.285	0.17019	-0.34994	-1.1100
docs	11.108	1.0557	0.94567	-0.33877

	5% perc.	95% perc.	IQ range	Missing obs.
lifeexpf	47.150	81.000	20.000	0
docs	0.33624	31.550	15.654	1

- kurtosis - thickness of tales; less or more likely to generate outliers compared to the Normal Distribution
- With code:

```

summary lifeexpf docs
summary lifeexpf docs --simple

```

- Scatter plot:
 - View->Graph specified variables -> X-Y Scatter , docs on X, lifeexpf on Y
 - Via code:

```

gnuplot lifeexpf docs --fit=none --output=display # No line fit; show graph
graph1 <- gnuplot lifeexpf docs --fit=none # Save into View -> Icon View, graph1
# Can now Save session to save results

```

- Correlation coefficient between two variables:
 - View-> Correlation Matrix -> choose variables, Ok
 - Via code:

```

corr lifeexpf docs

```

- Distribution:
 - Highlight region -> Variable -> Frequency distribution
 - Via code:

```

# Frequency distribution for qualitative variable region:
freq region # W/o graph
freq region --plot=display # W/graph

```

- Cross-tabulation: for a relationship between two categorical/qualitative variables, two-way table:
 - Can point and click
 - Via code (develop is 0 for “developed”, 1 for “developing”):

```

xtab region develop

```

- Summary stats by categories of a categorical variables:

```
summary lifeexpf --by=develop
```

- Create a binary variable for when a country is in Africa:

```
# We are going to create a new variable that indicates if country is in Africa:
```

```
print country region --byobs # Print country and regions:
```

```
# Discover that regions 1 through 5 are African:
```

```
dataset sortby region # Sort by region, because regions are grouped by number
```

```
print country region --byobs # Print country and regions:
```

```
# Create the new binary variable:
```

```
series africa = region<=5
```

```
# Frequency table, b/c categorical variable:
```

```
freq africa
```

```
summary lifeexpf docs --simple --by=africa
```

Output:

africa = 0 (n = 80):

	Mean	Median	S.D.	Min	Max
lifeexpf	71.38	74.00	9.334	43.00	83.00
docs	14.76	12.02	11.27	0.3704	42.92

africa = 1 (n = 42):

	Mean	Median	S.D.	Min	Max
lifeexpf	56.67	55.50	7.916	41.00	74.00
docs	2.242	0.7747	3.624	0.1880	16.23

```
# Restrict sample to only African countries:
```

```
smpl africa==1 --restrict
```

```
summary lifeexpf docs --simple
```

Or for full:

```
smpl --full
```