Economics Senior Project Colloquium

Data Management and Methodology

Thomas Masterson

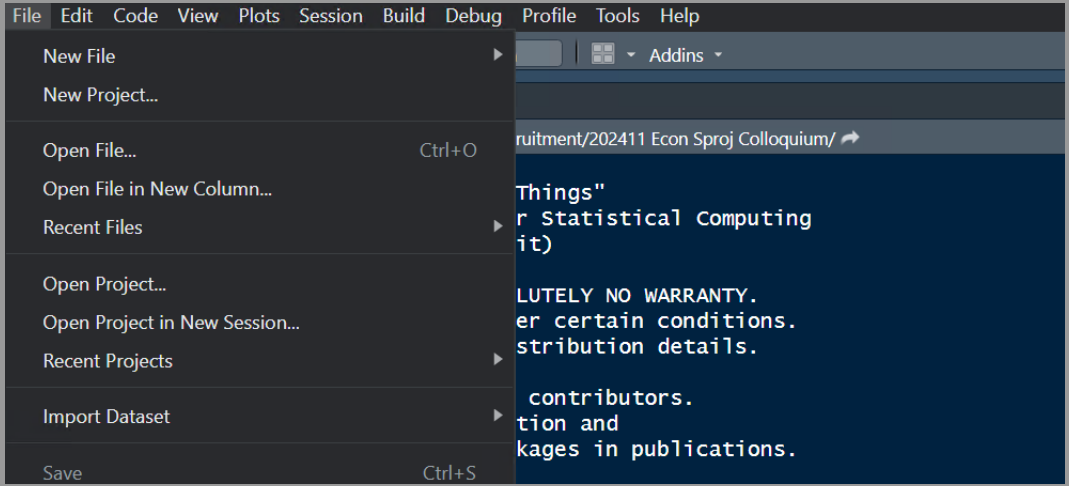
November 1, 2024

Agenda

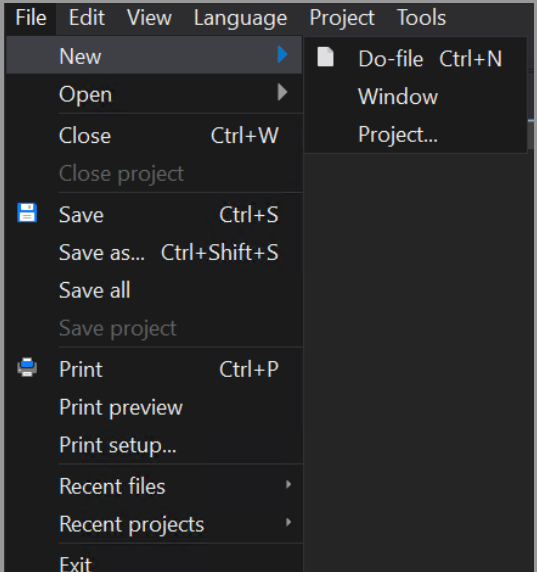
Data Management

- Folders
  - project
  - sub-folders: raw data, results, publication, etc
- Projects
- First, a little setup:
  - First create a new folder you're going to use for organizing today's work
  - In that folder, create a new folder called 'Data'
  - Next, navigate to: https://github.com/masterson-levy/econ_sproj_colloquium
  - From the data folder, download the three data sets to the *Data* folder you just created

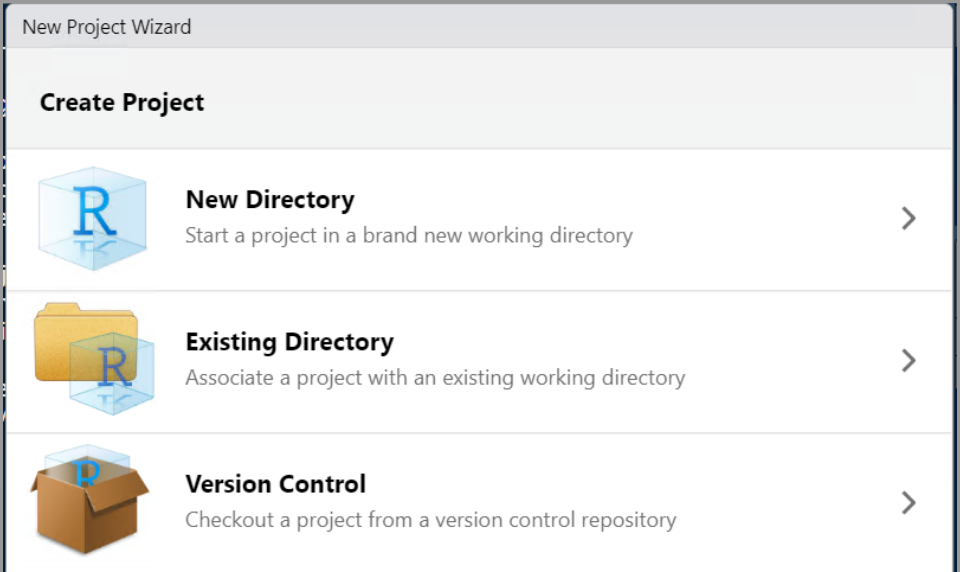
New Project - R



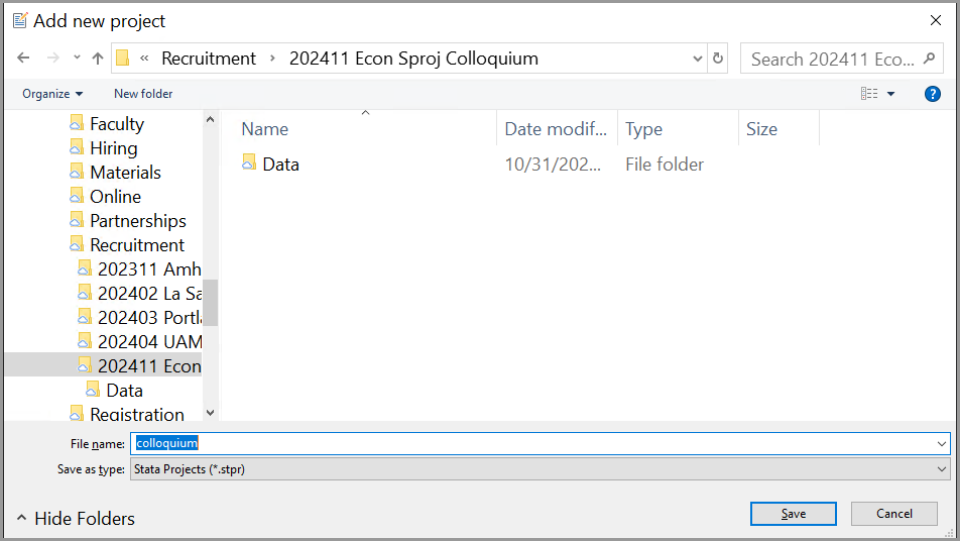
New Project - STATA



Project Directory - R



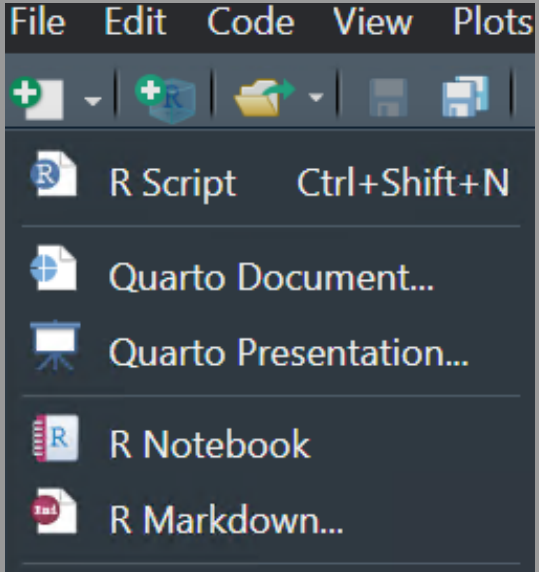
Project File - STATA



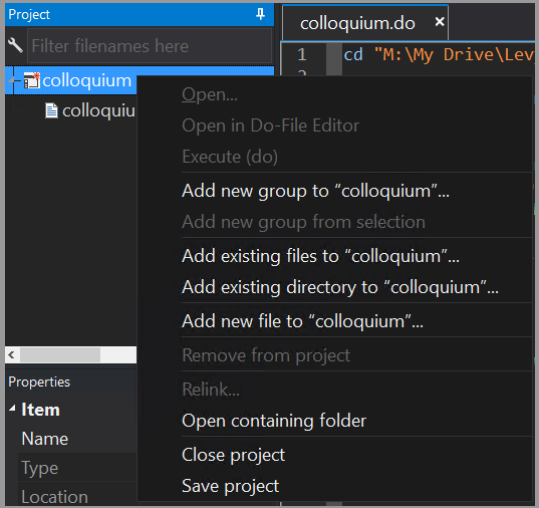
Organizing your code

- Always, **always**, use a saved program file to do your analysis
  - In R, program files are called R scripts, and have a *.R extension.
  - In STATA, program files are called do files, and have a *.do extension

Create a New R Script



Add a New do File to Your Project



Comments, Whitespace, and Variable Names

- You want your programs to be easily understood, both by you and others
- Three important practices to make your programs understandable
  1. Comments
     - In R, the # character at the beginning of the line
     - In Stata, the * character at the beginning of the line, all text between /* and */, or all text after ///
  2. Whitespace and indentation
     - When combined with good comments, spacing your code makes it much more readable, and it's free!
     - Indentation is another good way to organize your code, especially within structures such as loops, if/then/else blocks etc.
  3. Variable Names
     - $X1$ is not very obvious, but *logWage* is
- Bookmarks: STATA only, in do file editor

Data in Excel Sheets

- For those of you doing macro projects, or doing your own surveys, you might gather the data you need in an excel sheet before working on the analysis
- Both STATA and R have the capability to read from and write to excel sheets (and other kinds of files, as well)
  - In R, you need to install a package

```
in console:
install.packages("openxlsx")

in code:
library(openxlsx)
penn <- read.xlsx("Data/pwt1001.xlsx", "Data")
```

  - In STATA, the command is built in

```
import excel Data/pwt1001.xlsx, sheet("Data") first clear
```

Summary Statistics

- Straightforward in STATA: summary command

```
su
means
tabstat rgdpe if country=="Ghana", by(year)
```

- Less so in R:

```
# built-in base R:
summary(penn)

# skimr package
library(skimr)
skim(penn)

# vtable package
library(vtable)
st(penn)
```

Frequency Tables

- Again, straightforward in STATA: tab

```
tab year
```

- R, not quite as nice

```
table1 <- table(penn$year)
table1

library(questionr)
ftable1 <- freq(table1)
```

Missing Values

- STATA and R treat them differently
  - In R, they are automatically excluded from most calculations and functions

```
penn$high <- penn$rgdpo>150000
table(penn$high)
```

  - In STATA, missing values are treated as positive infinity for comparisons

```
gen high = rgdpo>150000
tab high

replace high = . if mi(rgdpo)
tab high
```

Regressions

- There are user-contributed packages for both R and STATA
  - For R: stargazer
  - For STATA: estout

R and stargazer

```
library(readstata13)
mroz <- read.dta13("Data/mroz.dta")

wage1 <- lm(lwage ~ exper + expersq + age + educ, data = mroz)

library(stargazer)
stargazer(wage1, type = "text", digits = 2,
    covariate.labels = c("Experience",
"Experience Squared",
"Age", "Education"),
    title = "Wage determination using the Mroz data")
```

STATA and estout

```
use Data/mroz, clear
su

reg lwage exper expersq age educ
est store wage1

estout wage1, ///
     cells("b(label(Coef.) fmt(%6.3f)) se(label(Std. Err.))") ///
     stats(r2_a N, fmt(%6.3f %6.0f) labels("Adj. R^2" "No. of cases")) ///
     varlabels(_cons Constant exper Experience expersq "Experience Squared" ///
     age Age educ Education ) varwidth(30) ///
     prehead("Wage determination using the Mroz data")
```