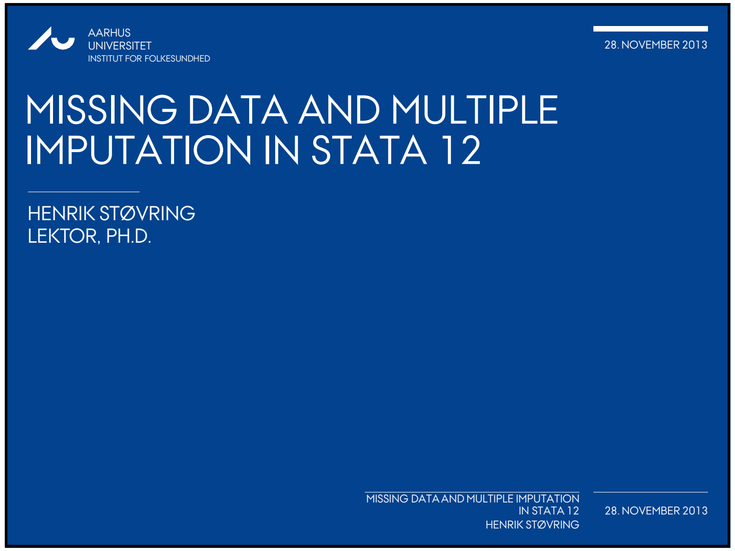
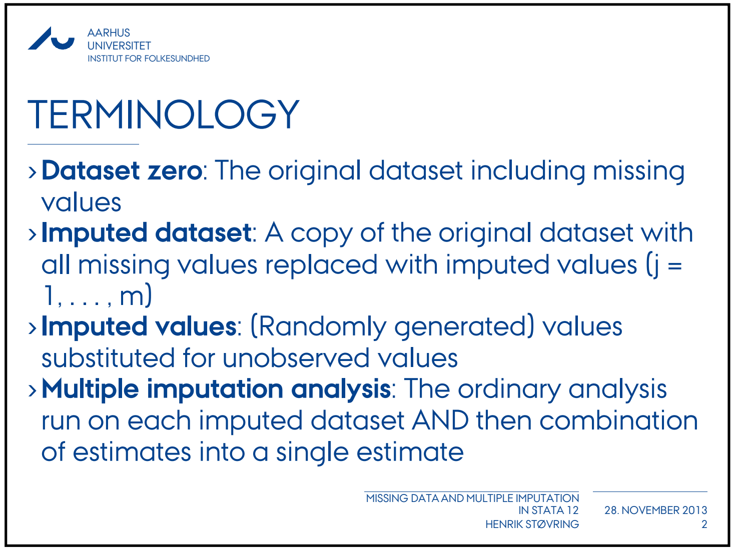


28. NOVEMBER 2013

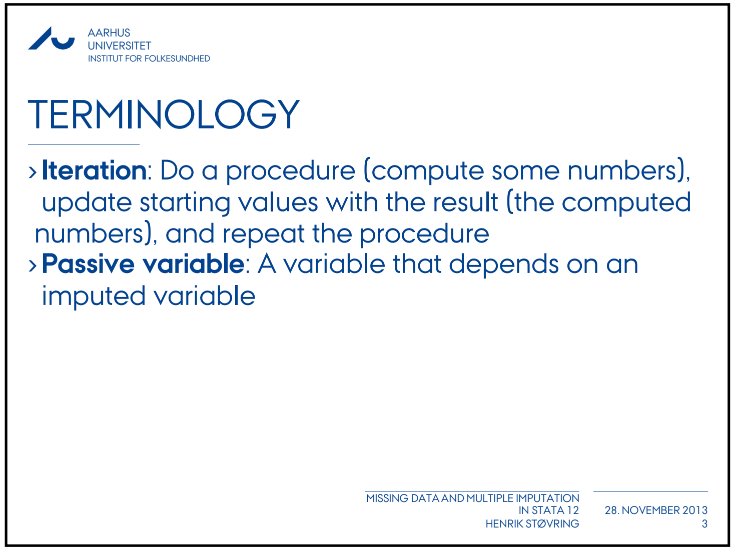
MISSING DATA AND MULTIPLE IMPUTATION IN STATA 12

HENRIK STØVRING
LEKTOR, PH.D.



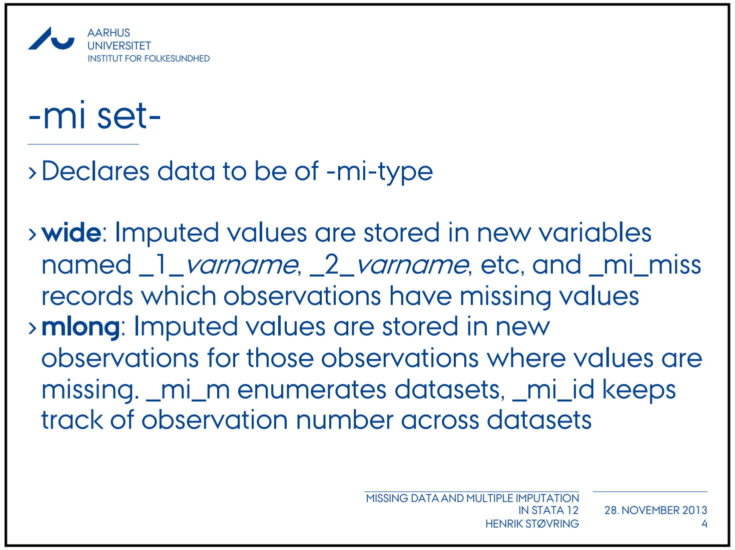
TERMINOLOGY

- **Dataset zero**: The original dataset including missing values
- **Imputed dataset**: A copy of the original dataset with all missing values replaced with imputed values (j = 1, . . . , m)
- **Imputed values**: (Randomly generated) values substituted for unobserved values
- **Multiple imputation analysis**: The ordinary analysis run on each imputed dataset AND then combination of estimates into a single estimate



TERMINOLOGY

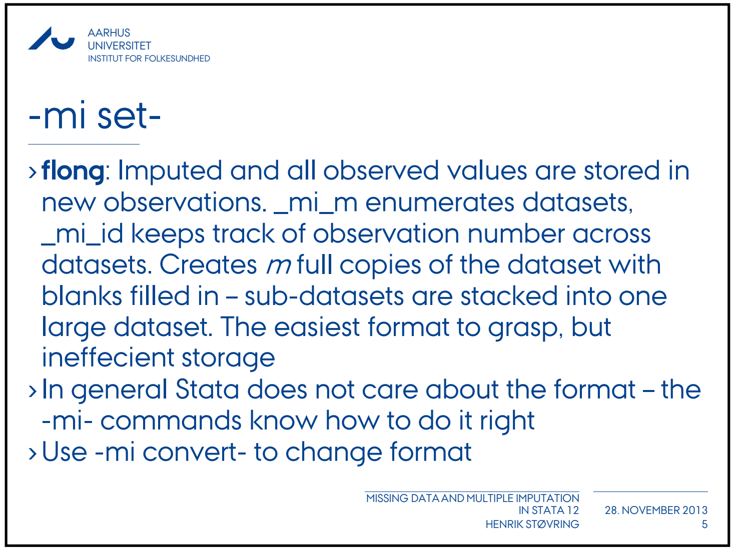
- **Iteration**: Do a procedure (compute some numbers), update starting values with the result (the computed numbers), and repeat the procedure
- **Passive variable**: A variable that depends on an imputed variable



-mi set-

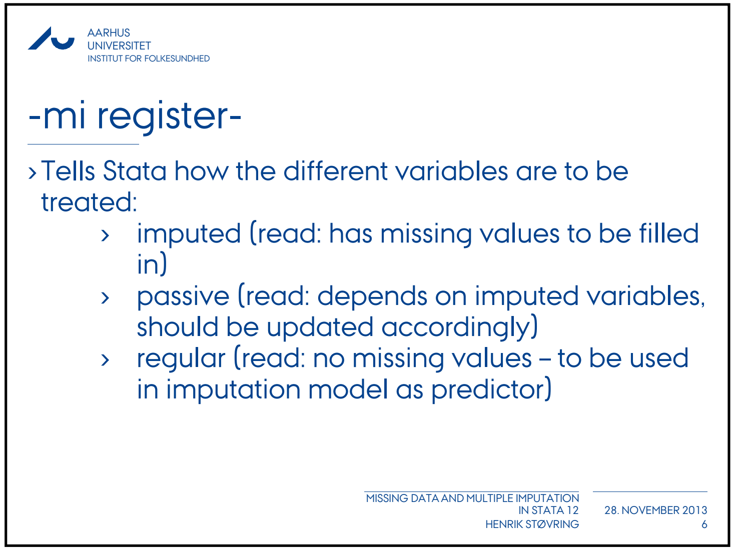
- Declares data to be of -mi-type

- **wide**: Imputed values are stored in new variables named _1_*varname*, _2_*varname*, etc, and _mi_miss records which observations have missing values
- **mlong**: Imputed values are stored in new observations for those observations where values are missing. _mi_m enumerates datasets, _mi_id keeps track of observation number across datasets



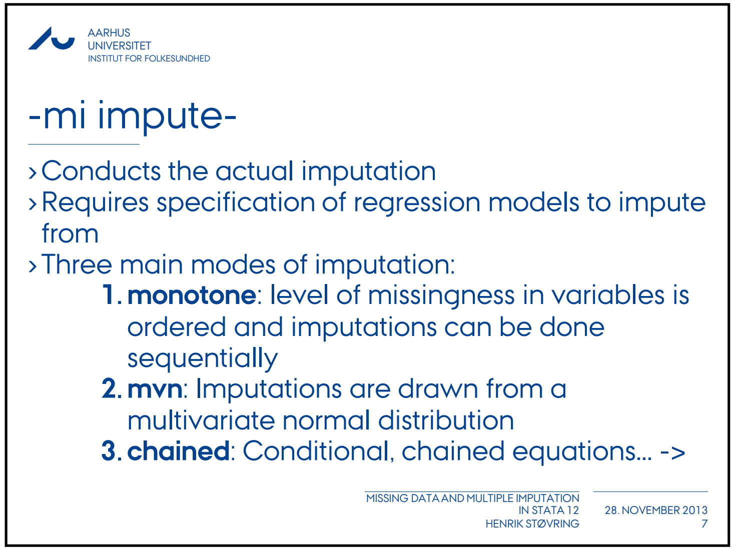
-mi set-

- **flong**: Imputed and all observed values are stored in new observations. _mi_m enumerates datasets, _mi_id keeps track of observation number across datasets. Creates *m* full copies of the dataset with blanks filled in – sub-datasets are stacked into one large dataset. The easiest format to grasp, but ineffecient storage
- In general Stata does not care about the format – the -mi- commands know how to do it right
- Use -mi convert- to change format



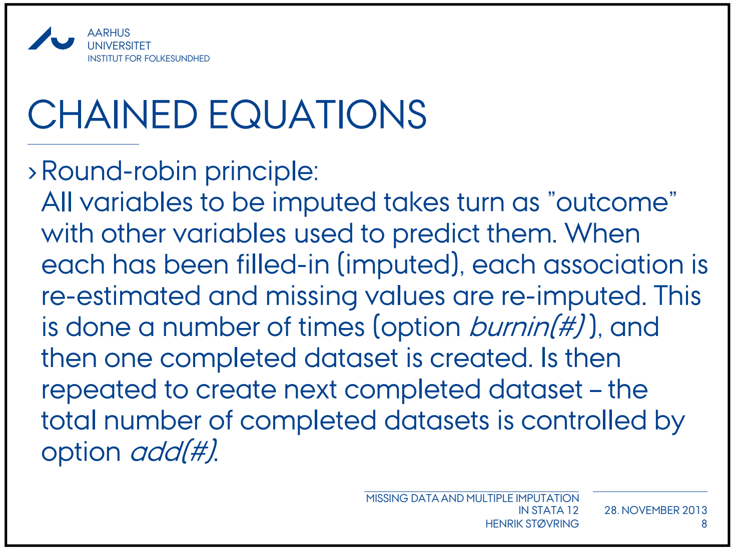
-mi register-

- Tells Stata how the different variables are to be treated:
  - imputed (read: has missing values to be filled in)
  - passive (read: depends on imputed variables, should be updated accordingly)
  - regular (read: no missing values – to be used in imputation model as predictor)



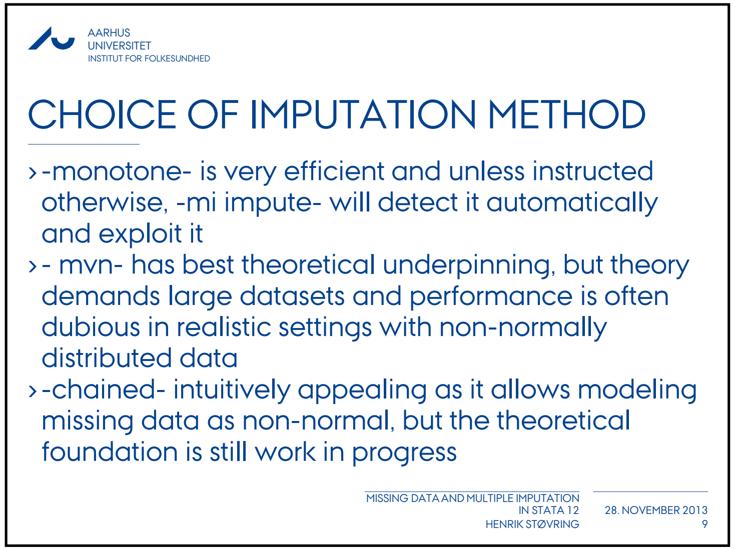
-mi impute-

- Conducts the actual imputation
- Requires specification of regression models to impute from
- Three main modes of imputation:
  1. **monotone**: level of missingness in variables is ordered and imputations can be done sequentially
  2. **mvn**: Imputations are drawn from a multivariate normal distribution
  3. **chained**: Conditional, chained equations… ->



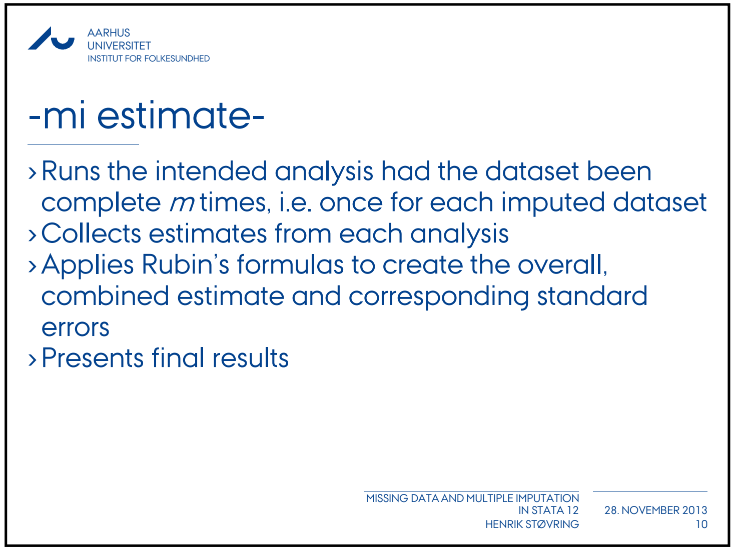
CHAINED EQUATIONS

- Round-robin principle:
  All variables to be imputed takes turn as ”outcome” with other variables used to predict them. When each has been filled-in (imputed), each association is re-estimated and missing values are re-imputed. This is done a number of times (option *burnin(#)*), and then one completed dataset is created. Is then repeated to create next completed dataset – the total number of completed datasets is controlled by option *add(#)*.



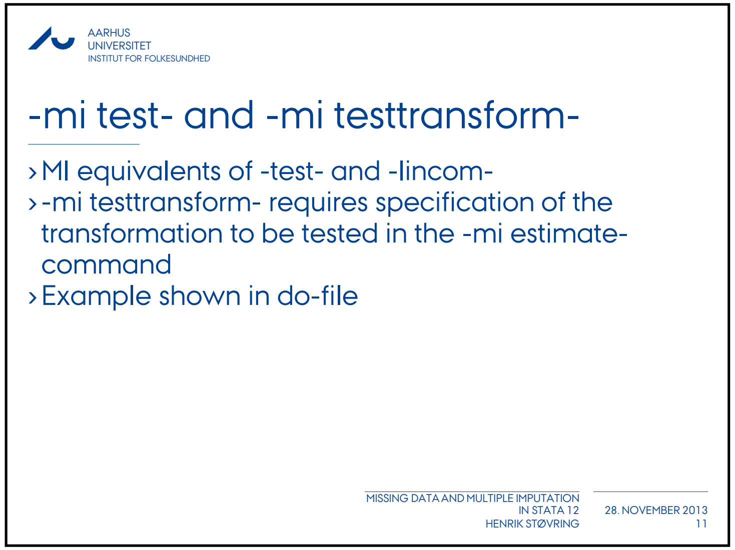
CHOICE OF IMPUTATION METHOD

- -monotone- is very efficient and unless instructed otherwise, -mi impute- will detect it automatically and exploit it
- - mvn- has best theoretical underpinning, but theory demands large datasets and performance is often dubious in realistic settings with non-normally distributed data
- -chained- intuitively appealing as it allows modeling missing data as non-normal, but the theoretical foundation is still work in progress



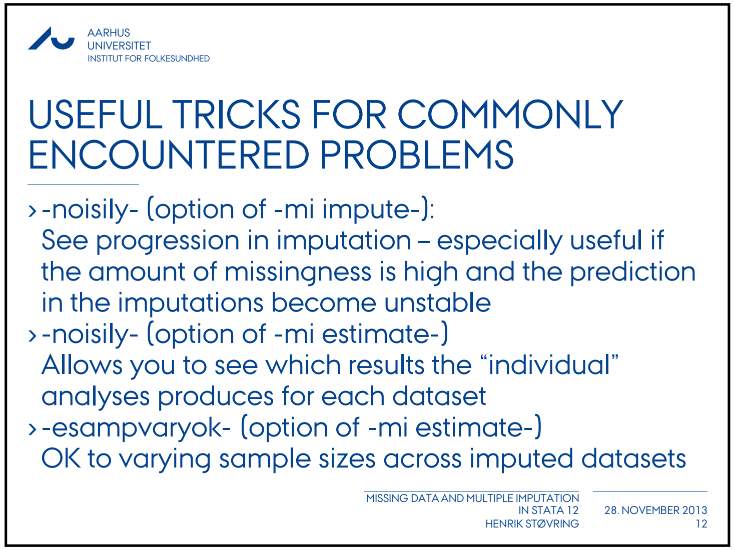
-mi estimate-

- Runs the intended analysis had the dataset been complete *m* times, i.e. once for each imputed dataset
- Collects estimates from each analysis
- Applies Rubin's formulas to create the overall, combined estimate and corresponding standard errors
- Presents final results



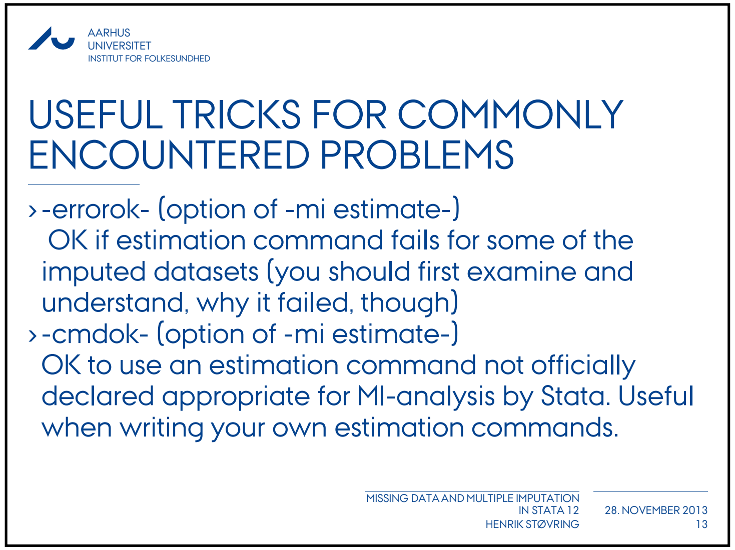
-mi test- and -mi testtransform-

- MI equivalents of -test- and -lincom-
- -mi testtransform- requires specification of the transformation to be tested in the -mi estimate- command
- Example shown in do-file



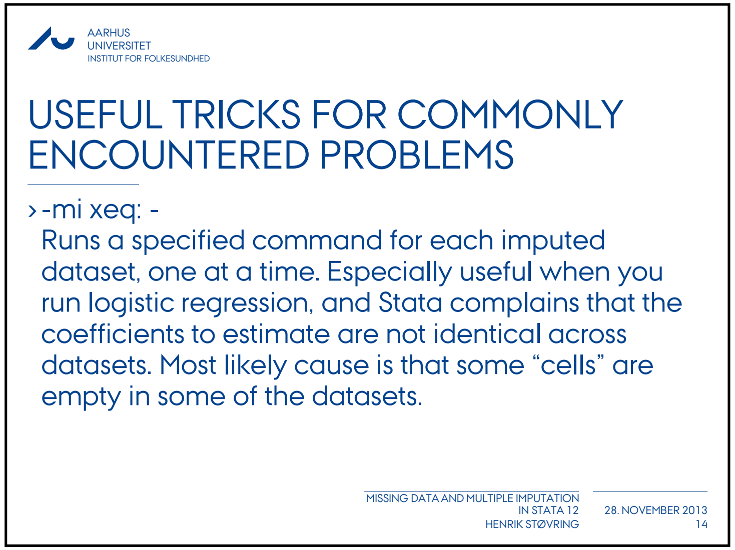
USEFUL TRICKS FOR COMMONLY ENCOUNTERED PROBLEMS

- -noisily- (option of -mi impute-):
  See progression in imputation – especially useful if the amount of missingness is high and the prediction in the imputations become unstable
- -noisily- (option of -mi estimate-)
  Allows you to see which results the "individual" analyses produces for each dataset
- -esampvaryok- (option of -mi estimate-)
  OK to varying sample sizes across imputed datasets



USEFUL TRICKS FOR COMMONLY ENCOUNTERED PROBLEMS

- -errorok- (option of -mi estimate-)
  OK if estimation command fails for some of the imputed datasets (you should first examine and understand, why it failed, though)
- -cmdok- (option of -mi estimate-)
  OK to use an estimation command not officially declared appropriate for MI-analysis by Stata. Useful when writing your own estimation commands.



USEFUL TRICKS FOR COMMONLY ENCOUNTERED PROBLEMS

- -mi xeq: -
  Runs a specified command for each imputed dataset, one at a time. Especially useful when you run logistic regression, and Stata complains that the coefficients to estimate are not identical across datasets. Most likely cause is that some “cells” are empty in some of the datasets.