Requisites

- Matlab or Octave
- Matlab Econometrics Toolbox (for charting only)
- Project path must be added or the main method must be called from command line

How to run

- Save to someDir, call someDir\main.m, which saves results in .csv format in someDir\res

Structure

- **p.dgp.CDgp:** Abstract data generating process, implements orchestration, that is running simulations, storing results
- **p.dgp.CArx:** Concrete data generating process, implements the
  - actual data generating process, for example AR1
  - Iterate functions, which generate simulated data under various parameters
- **p.s:** statistics,
  - **CUnitRootTests:** Test-suite which collects various unit root tests together
  - **unitRoot:** various unit root tests
  - **binomTest:** Binomial test
- **p.bcc.CBcc:** implements counting representation of some data as well as unit root tests based on this counting representation
- **Other**
  - analytical functions (a.)
  - data structures (d.)
  - test (t.), helps creation of tests
  - utilities (u.)

How to explore

- Each functionality has a static method starting with mTest. These methods return an instance of the object. For example
  - cArx = p.dgp.CArx.mTestUnitRootIndependent() can be used to explore how the simulation of unit root data is implemented
  - cCount =p.bcc.CCount.mTestUnitRoot() can be used to explore how counting-based unit root test is implemented
  - cArx = p.dgp.CArx.mTestIterateWithDistDependence() can be used to explore the parameter space for dependent panels.

How to expand

- To add a new test, put the test somewhere in the project and add the test to the CUnitRootTests suite. Also specify the header in the main.m
- To add a new type of data generating process, for example ARMA(1,1) implement it in CArx and create new or modify existing iterator. Finally call these from the main.m
- To alter the orchestration, for example to modify the way files are saved can be done in CDgp.
- Finally, reusable code can be added to the packages.