EViews: Introductory User Guide

Data Functions

Random Numbers Generators | Basic Data Functions | Advanced Data Functions

Learning support material for the courses:

- ✓ **NMST537 Time Series Analysis**
- ✓ **NEKN432 Econometrics**

Based on official EViews Tutorials & EViews Illustrated.

Data Functions in EViews

- EViews has a large number of built-in functions to manipulate data.
- EViews functions are typically denoted by the symbol @.
- This tutorial reviews some typical functions used for data manipulation:
  - ✓ Random numbers
  - ✓ Statistical Functions
  - ✓ Commonly Used Time Series Functions

EViews: Introductory User Guide

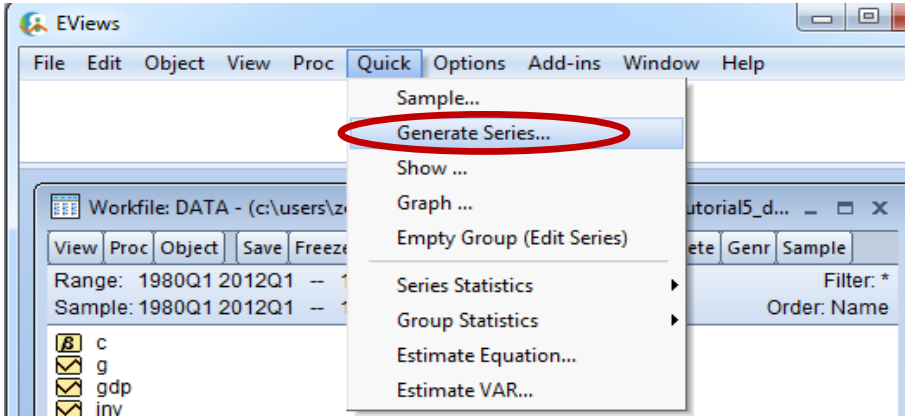
RANDOM NUMBERS GENERATORS

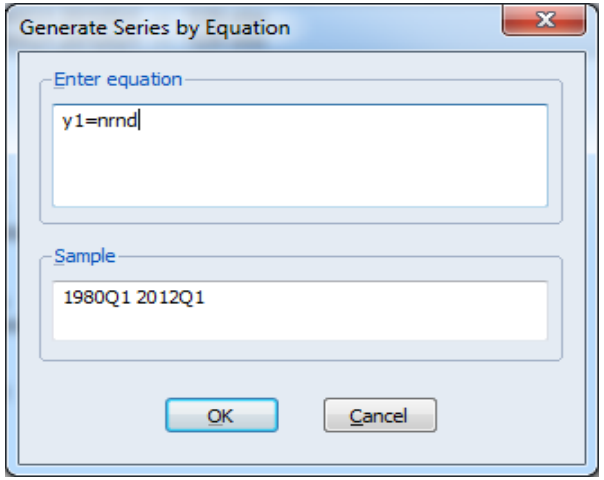
Generating Random Numbers (Part I)

- One can generate a series of (pseudo) random numbers drawn from a variety of distributions. There are a number of ways to generate a random series.

Generating a random series (Example 1):

1. Open an EViews workfile.
2. Select **Quick** → **Generate Series** from the main menu.
3. Type, e.g. ***y1 = nrnd*** in the dialog box and press **Enter.**
4. This generates a series ***y1***, which represents a N(0,1) random sample.

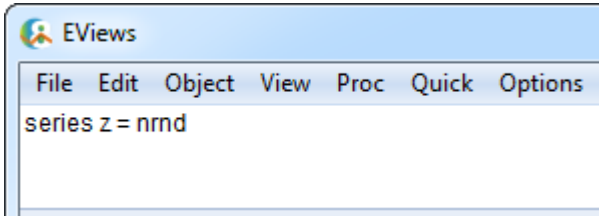


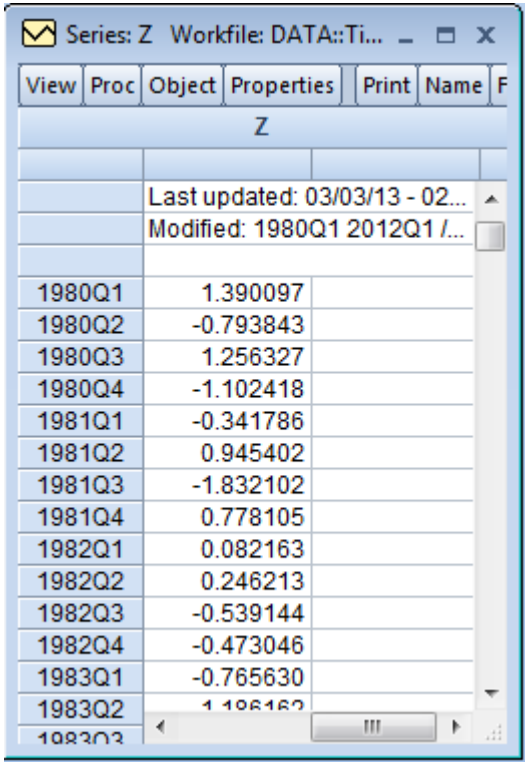


Generating Random Numbers (Part II)

Generating a random series (Example 2):

1. Open an EViews workfile.
2. Type in the command window: ***series z=nrnd***.
3. Press **Enter**.
4. This creates a new series ***z***, which represents a N(0,1) random sample.





Generating Random Numbers (Part III)

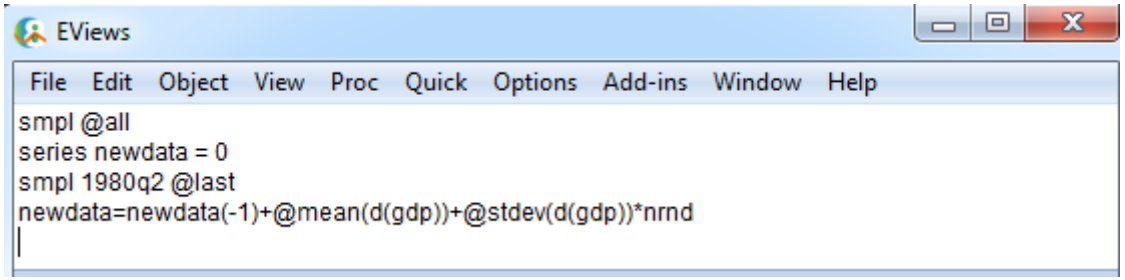
- Suppose you want to simulate a random walk process with distribution properties similar to the observed distribution of an existing series.

Generating a random series (Example 3):

1. Type in the command window (*gdp* is a time series object):

```
smpl @all
series newdata = 0
smpl 1980q2 @last
newdata=newdata(-1)+@mean(d(gdp))+@stdev(d(gdp))*nrnd
```

2. Press **Enter** after each command line.



*Note: the functions **@mean** and **@stdev** calculate the sample mean and standard deviance of its argument. The operator **d(*)** creates the first differences of its argument.*

Note: the commands instruct EViews to first create a new series (newdata) consisting of zeros.

Generating Random Numbers: Functions

Common Functions/Commands	Description
series y=nrnd or (y=@rnorm)	Normal distribution (mean 0, st. dev. 1)
series y=3+@sqr(4)*nrnd	Normal distribution (mean 3, variance 4)
series y=@rlognorm(1,4)	Lognormal distribution (mean=1, st. dev 4)
series y=@runif(1,3)	Uniform distribution on (1,3)
series y=rnd	Uniform distribution on (0,1)
series y=1+(3-1)* rnd	Uniform distribution on (1,3) (same as @runif(1,3))
series y=rndint(0,100)	Fills series y with random integers drawn randomly from [0, 100]
series d1 = @runif(0,2)	Creates d1 as uniform distribution on (0,2)
@dunif(d1,0,2)	Creates pdf of d1
@cunif(d1,0,2)	Creates cdf of d1

EViews: Introductory User Guide

DESCRIPTIVE STATISTICS

Descriptive Statistics: Introduction

- EViews has extensive built-in descriptive statistical functions.
- These descriptive statistical functions take an optional sample as an argument.
- The default sample is the current workfile range.

Descriptive Statistics: Common Functions

Common Function/Command	Description
series y=@gmean(x)	Computes the geometric average of x
series y=@mean(x)	Creates a series where each observation is equal to the mean of x
series y=@mean(x,"1980m01 1990m12")	Creates a series where each observation is equal to the mean of x for the defined sample (1980m01 to 1990m12)
series y=@median(x)	Creates a series where each observation is equal to the median of x
series y=@vars(x)	Computes the sample variance of x (adj. by n-1)
series y=@varp(x) OR y=@var(x)	Computes the population variance of x (adj. by n)
series y=@stdev(x) OR y=@stdevs(x)	Computes the sample st. dev. of x (adj. by n-1)
series y=@stdevp(x)	Computes the population st. dev. of x (adj. by n)

Descriptive Statistics: Examples (Part I)

Descriptive Statistics Functions (Example 1):

1. Type in the command window (*gdp* is a time series object):

```
smpl 1980q1 1989q4
series x2=@mean(gdp, "1980q1 1989q4")
smpl 1990q1 1999q4
series x2=@mean(gdp, "1990q1 1999q4")
smpl 2000q1 @last
series x2=@mean(gdp, "2000q1 @last")
smpl @all
```

2. Press **Enter** after each command line.
3. This series of commands creates the series ***x2*** according to the used specification.

Descriptive Statistics: Examples (Part II)

- Suppose you want to create a variable which is the average (or sum) of multiple series, e.g. ***x***, ***y*** and ***z***.

Descriptive Statistics Functions (Example 2):

1. Type one of the following commands in the command window:

   a. ***series new=(x+y+z)/3***
   *[this creates the series **new** containing an average of three time series]*

   b. ***group groupdata x y z***
   *[this creates the group **groupdata** containing all three time series]*
   series new=@rmean(groupdata)
   [this creates a series which is computed by taking the mean of all the three series for each row*]*

2. Press **Enter** after each command line.

Descriptive Statistics: Examples (Part III)

Function	Description
vector(3) v	Creates a vector v with 3 elements
v(1)=@mean(x)	Assigns the first element of the vector v to equal the mean of x over the defined sample
v(2)=@varp(x)	Assigns the second element of the vector v to equal the population variance of x (adj. by n)
v(3)=@covs(x,y)	Assigns the third element of the vector **v** to equal the sample covariance between x and y (adj. by n-1)

EViews: Introductory User Guide

COMMON TIME SERIES FUNCTIONS

Lags & Leads

- It is easy to work with lags/leads and other time series functions in Eviews.
- You do not need to generate series of lags/leads in many places in Eviews - simply write the command for them when needed.

Function/Command	Description
x(-4)	Denotes the 4th lag of the x series
x(2)	Denotes the 2nd lead of the x series
x(-1 to -4)	Specifies all x lags from 1 to 4
x(to -5) OR x(0 to -5)	Specifies all x lags from 0 to -5
series y = @lag(x,3)	Generates series y, as the 3rd lag of x
series y =@lag(x-@mean(x),4)	Generates series y, as the 4th lag of the transformation.

Differences

Function/Command	Description
d(x)	Takes the first difference of the x series
dlog(x)	Takes the first difference of log(x) series
d(x,3)	3rd order difference of x series (i.e. $(1-L)^3x$))
dlog(x,4)	4th order difference of log(x) series (i.e. $\underline{(1-L)^3 log(x}$)).
d(x,1,4)	1st order differences with seasonal difference in lag 4
d(x,0,4)	Captures only seasonal difference (at lag 4)

Percent Changes

Function/Command	Description
@pc(x)	Calculates the one-period percent change in x (in percent)
@pch(x)	Calculates the one-period percent change in x (in decimal)
@pca(x)	Calculates the one-period ***annualized*** percent change in x (in percent) $(1+@pch(x))^{n}-1$ (where n=4 for quarterly data, n=12 for monthly, etc.)
@pcha(x)	Calculates the one-period ***annualized*** percent change in x (in decimal)
@pcy(x)	Calculates the one-year percent change (in percent)
@pchy(x)	Calculates the one-year percent change (in decimal)

Cumulative Statistic Functions

- Cumulative functions perform “running-total”-type calculations.
- For these functions the length of the window changes with each observation.

Function/Command	Description
@cumsum(x,s)	Cumulative Sum of the values in x over sample s
@cumprod(x,s)	Cumulative Product of the values in x over sample s
@cummean(x,s)	Mean of the values in x over sample s up to and including current observation
@cumobs(x,s)	The number of non-missing observations in x from over sample s
@cumbsum(x,s)	Backwards cumulative sum of the values of x over sample s beginning with the end of the sample
@cumbmean(x,s)	Backwards cumulative mean of the values of x over sample s beginning with the end of the sample up to and including current observation
@cumbstdev(x,s)	Backwards cumulative standard deviation of the values of x over sample s beginning with the end of the sample up to and including current observation

Moving Statistic Functions (Part I)

- These types of functions have shorter, fixed, user-specified window lengths.
- They provide information on *n* observations (including the current observation).
- The window length *n* is chosen by the user.
- If the original data has missing values (**NA**), results may or may not propagate NA.

Generic Function	Description
@***mov***[*specified statistic*]	This command generates missing values
@***m***[*specified statistic*]	This command skips **NA** observations and does not generate **NA** values

Moving Statistic Functions (Part II)

Function/Command	Description
@movav(x,n)	n-period backwards moving average (if n=3, (x+x(-1)+x(-2))/3) **NA**s are generated
@movav(x(-1),n)	n-period backwards moving average lagged by one period **NA**s are generated
@movsum(x,n)	n-period backwards moving sum (if n=3, x+x(-1)+x(-2)) **NA**s are generated
@movvar(x,n)	n-period backwards moving variances (population variance for the current and previous n-1 observations) **NA**s are generated
@mav(x,n)	n-period backwards moving average (if n=3, (x+x(-1)+x(-2))/3). **NA**s are NOT generated
@msum(x,n)	n-period backwards moving sum (if n=3, x+x(-1)+x(-2)). **NA**s are NOT generated
@mvars(x,n)	n-period backwards moving sample variance (sample variance for the current and previous n-1 observations) **NA**s are NOT generated

Trends

Function/Command	Description
@trend	Time trend increasing with each observation
@trendc	Calendar-based time trend
@trend^2	Quadratic time trend increasing with each observation

Radek Hendrych
Charles University
Faculty of Mathematics and Physics
Department of Probability and Mathematical Statistics
Sokolovská 83, 186 75 Prague, Czech Republic

E-mail: hendrych@karlin.mff.cuni.cz