NAG Library Function Document

nag_tsa_dickey_fuller_unit (g13awc)

1 Purpose

nag_tsa_dickey_fuller_unit (g13awc) returns the (augmented) Dickey–Fuller unit root test.

2 Specification

```
#include <nag.h>
#include <nagg13.h>
```

```
double nag_tsa_dickey_fuller_unit (Nag_TS_URTestType type, Integer p,
       Integer n, const double y[], NagError *fail)
```

3 Description

If the root of the characteristic equation for a time series is one then that series is said to have a unit root. Such series are nonstationary. nag_tsa_dickey_fuller_unit (g13awc) returns one of three types of (augmented) Dickey–Fuller test statistic: τ, τ_μ or τ_τ, used to test for a unit root, a unit root with drift or a unit root with drift and a deterministic time trend, respectively.

To test whether a time series, y_t, for $t = 1, 2, \dots, n$, has a unit root the regression model

$$\nabla y_t = \beta_1 y_{t-1} + \sum_{i=1}^{p-1} \delta_i \nabla y_{t-i} + \epsilon_t$$

is fit and the test statistic τ constructed as

$$\tau = \frac{\hat{\beta}_1}{\sigma_{11}}$$

where ∇ is the difference operator, with $\nabla y_t = y_t - y_{t-1}$, and where $\hat{\beta}_1$ and σ_{11} are the least squares estimate and associated standard error for β_1 respectively.

To test for a unit root with drift the regression model

$$\nabla y_t = \beta_1 y_{t-1} + \sum_{i=1}^{p-1} \delta_i \nabla y_{t-i} + \alpha + \epsilon_t$$

is fit and the test statistic τ_μ constructed as

$$\tau_\mu = \frac{\hat{\beta}_1}{\sigma_{11}}$$

To test for a unit root with drift and deterministic time trend the regression model

$$\nabla y_t = \beta_1 y_{t-1} + \sum_{i=1}^{p-1} \delta_i \nabla y_{t-i} + \alpha + \beta_2 t + \epsilon_t$$

is fit and the test statistic τ_τ constructed as

$$\tau_\tau = \frac{\hat{\beta}_1}{\sigma_{11}}$$

The distributions of the three test statistics; τ, τ_μ and τ_τ, are nonstandard. An associated probability can be obtained from nag_prob_dickey_fuller_unit (g01ewc).

4 References

Dickey A D (1976) Estimation and hypothesis testing in nonstationary time series *PhD Thesis* Iowa State University, Ames, Iowa

Dickey A D and Fuller W A (1979) Distribution of the estimators for autoregressive time series with a unit root *J. Am. Stat. Assoc.* **74 366** 427–431

5 Arguments

1: **type** – Nag_TS_URTestType *Input*

On entry: the type of unit test for which the probability is required.

type = Nag_UnitRoot
A unit root test will be performed and τ returned.

type = Nag_UnitRootWithDrift
A unit root test with drift will be performed and τ_μ returned.

type = Nag_UnitRootWithDriftAndTrend
A unit root test with drift and deterministic time trend will be performed and τ_τ returned.

Constraint: **type** = Nag_UnitRoot, Nag_UnitRootWithDrift or Nag_UnitRootWithDriftAndTrend.

2: **p** – Integer *Input*

On entry: p, the degree of the autoregressive (AR) component of the Dickey–Fuller test statistic. When $p > 1$ the test is usually referred to as the augmented Dickey–Fuller test.

Constraint: **p** > 0.

3: **n** – Integer *Input*

On entry: n, the length of the time series.

Constraints:

if **type** = Nag_UnitRoot, **n** > 2**p**;
if **type** = Nag_UnitRootWithDrift, **n** > 2**p** $+ 1$;
if **type** = Nag_UnitRootWithDriftAndTrend, **n** > 2**p** $+ 2$.

4: **y**[**n**] – const double *Input*

On entry: y, the time series.

5: **fail** – NagError * *Input/Output*

The NAG error argument (see Section 3.6 in the Essential Introduction).

6 Error Indicators and Warnings

NE_ALLOC_FAIL

Dynamic memory allocation failed.
See Section 3.2.1.2 in the Essential Introduction for further information.

NE_BAD_PARAM

On entry, argument ⟨*value*⟩ had an illegal value.

NE_INT

On entry, **n** = ⟨*value*⟩.
Constraint: **n** > ⟨*value*⟩.

NE_INTERNAL_ERROR

An internal error has occurred in this function. Check the function call and any array sizes. If the call is correct then please contact NAG for assistance.

An unexpected error has been triggered by this function. Please contact NAG.
See Section 3.6.6 in the Essential Introduction for further information.

NE_NO_LICENCE

Your licence key may have expired or may not have been installed correctly.
See Section 3.6.5 in the Essential Introduction for further information.

NE_ORDERS_ARIMA

On entry, $\mathbf{p} = \langle value \rangle$.
Constraint: $\mathbf{p} > 0$.

NW_SOLN_NOT_UNIQUE

On entry, the design matrix used in the estimation of β_1 is not of full rank, this is usually due to all elements of the series being virtually identical. The returned statistic is therefore not unique and likely to be meaningless.

NW_TRUNCATED

$\sigma_{11} = 0$, therefore depending on the sign of $\hat{\beta}_1$, a large positive or negative value has been returned.

7 Accuracy

None.

8 Parallelism and Performance

Not applicable.

9 Further Comments

None.

10 Example

In this example a Dickey–Fuller unit root test is applied to a time series related to the rate of the earth's rotation about its polar axis.

10.1 Program Text

```
/* nag_tsa_dickey_fuller_unit (g13awc) Example Program.
 *
 * Copyright 2014 Numerical Algorithms Group.
 *
 * Mark 25, 2014.
 */

#include <stdio.h>
#include <nag.h>
#include <nag_stdlib.h>
#include <nagg01.h>
#include <nagg13.h>

int main(void)
{
```

```
  /* Integer scalar and array declarations */
  Integer n, nsamp=0, p, i;
  Integer exit_status = 0;
  Integer *state = 0;

  /* NAG structures and types */
  NagError fail;
  Nag_TS_URProbMethod method;
  Nag_TS_URTestType type;

  /* Double scalar and array declarations */
  double   pvalue, ts;
  double   *y = 0;

  /* Character scalar and array declarations */
  char ctype[30];

  /* Initialise the error structure */
  INIT_FAIL(fail);

  printf("nag_tsa_dickey_fuller_unit (g13awc) Example Program Results\n\n");

  /* Skip heading in data file */
#ifdef _WIN32
  scanf_s("%*[^\n] ");
#else
  scanf("%*[^\n] ");
#endif

  /* Read in the problem size, test type, order of the AR process */
#ifdef _WIN32
  scanf_s("%"NAG_IFMT"%29s%"NAG_IFMT"%*[^\n] ", &n, ctype, _countof(ctype), &p);
#else
  scanf("%"NAG_IFMT"%29s%"NAG_IFMT"%*[^\n] ", &n, ctype, &p);
#endif
  type = (Nag_TS_URTestType) nag_enum_name_to_value(ctype);

  /* Allocate memory */
  if (!(y = NAG_ALLOC(n, double)))
    {
      printf("Allocation failure\n");
      exit_status = -1;
      goto END;
    }

  /* Read in the time series */
  for (i = 0; i < n; i++)
#ifdef _WIN32
    scanf_s("%lf", &y[i]);
#else
    scanf("%lf", &y[i]);
#endif
#ifdef _WIN32
  scanf_s("%*[^\n] ");
#else
  scanf("%*[^\n] ");
#endif

  /* nag_tsa_dickey_fuller_unit (g13awc):
     Calculate the Dickey-Fuller test statistic */
  ts = nag_tsa_dickey_fuller_unit(type,p,n,y,&fail);
  if (fail.code != NE_NOERROR)
    {
      printf("Error from nag_tsa_dickey_fuller_unit (g13awc).\n%s\n",
             fail.message);
      exit_status = 1;
      goto END;
    }

  /* g01ewc: Get the associated p-value using simulation */
  method = Nag_ViaLookUp;
```

```
  pvalue = nag_prob_dickey_fuller_unit(method,type,n,ts,nsamp,state,&fail);
  if (fail.code != NE_NOERROR && fail.code != NW_EXTRAPOLATION)
    {
      printf("Error from nag_prob_dickey_fuller_unit (g01ewc).\n%s\n",
             fail.message);
      exit_status = 1;
      goto END;
    }

  /* Display the results */
  printf("Dickey-Fuller test statistic     = %6.3f\n",ts);
  printf("associated p-value               = %6.3f\n",pvalue);
  if (fail.code == NW_EXTRAPOLATION)
    {
      printf("NB: p-value obtained via extrapolation\n");
    }

 END:
  NAG_FREE(y);

  return exit_status;
}
```

10.2 Program Data

```
nag_tsa_dickey_fuller_unit (g13awc) Example Program Data
 30 Nag_UnitRoot 1                                 :: n,type,p
 -217 -177 -166 -136 -110  -95  -64  -37  -14  -25
  -51  -62  -73  -88 -113 -120  -83  -33  -19   21
   17   44   44   78   88  122  126  114   85   64 :: End of y
```

10.3 Program Results

```
nag_tsa_dickey_fuller_unit (g13awc) Example Program Results

Dickey-Fuller test statistic     = -2.540
associated p-value               =  0.013
```