

# 実習 8-1 ADF 検定の関数を作る

Scilab で ADF 検定のプログラム

## Contents

<b>1</b>	<b>目的</b>	<b>2</b>
<b>2</b>	<b>プログラム 1/3</b>	<b>3</b>
2.1	プログラム 2/3 .....	4
2.2	プログラム 3/3 .....	5
<b>3</b>	<b>例 1</b>	<b>6</b>

## 1 目的

- Scilab のユーザー定義関数 `function ... endfunction` を使う
- 時系列データ (Y) とラグ (p) を与え、
- 3 種類のタオ値を返す関数 ADF をつくる。
- OLS の t 値を返す関数 `OLStvalue` と共に利用する。
- この関数を使っているいろいろなデータで実験してみる。
- 動かし方は実習 4-2 を参考に

プログラムは、テキストエディタ (大東ならば `xyzzzy`) に書くこと。

## 2 プログラム 1/3

以下3ページ全て同じファイルに書いて、ADF.sci という名前で保存する。

```
function F=ADF(yvec,p)
    n=size(yvec,1);
    dyvec=yvec(2:n)-yvec(1:n-1);
    yt_1=yvec(p:n-1);

    n=n-1;
    dymat=zeros(n-p+1,p);
    for i=0:p-1
        dymat(:,1+i)=dyvec(p-i:n-i);
    end
```

## 2.1 プログラム 2/3

```
X1=[yt_1,dymat(:,2:p)];
```

```
t1=OLStvalue(dymat(:,1),X1);
```

```
tao1=t1(1);
```

```
X2=[ones(n-p+1,1), yt_1 , dymat(:,2:p)];
```

```
t2=OLStvalue(dymat(:,1),X2);
```

```
tao2=t2(2);
```

```
X3=[ones(n-p+1,1), [1:n-p+1]', yt_1, dymat(:,2:p)];
```

```
t3=OLStvalue(dymat(:,1),X3);
```

```
tao3=t3(3);
```

```
F=[tao1 tao2 tao3];
```

```
endfunction
```

## 2.2 プログラム 3/3

```
function F=OLStvalue(y,X)

    b=inv(X'*X)*X'*y;
    e=y-X*b;
    s2=e'*e/(size(X,1)-size(X,2));
    covb=s2*inv(X'*X);
    stdb=sqrt(diag(covb));
    F=b./stdb;
endfunction
```

### 3 例 1

Scilab 上で次のように入力する。

```
u=[1:5 4:-1:1]'
```

```
ADF(u,1)
```

```
ADF(u,2)
```

結果として

```
ADF(u,1)=[-.4131969 - 1.0954451 - .4879500]
```

```
ADF(u,2)=[-1.3591507 -1.7320508 -1.1338934]
```

が得られる。

(各  $\tau$  値は、定数項なし・定数項のみ・定数項とトレンドあり の順)

End  
Push Esc Key

**(C)KADODA Tamotsu (角田 保)**  
**@ Daito Bunka Univ. (大東文化大学)**  
**Last Modified: July 1, 2003**