

Ödev 5, Stata kullanıcıları için ipuçları

Raymond P. Guiteras
MIT 14.382, İlkbahar 2006

Bu ipuçları Stata 7.0, 8.0 ve 8.2’de çalışmaktadır. Daha ilerki versiyonlarda çalışmazsa do-dosyasına “version 8.0”ı ilk satır olarak ekleyerek çalıştırabilirsiniz (**Önemli not:** Bu dosya aslından tercüme edilmiş ama komutlar yazılımda denenmemiştir; bu sebeple, orijinal dosyada kullanılan programlara dikkat ediniz. Stata komutları koyu renkle ifade edilmiştir. Stata yazılımında kullanılan tırnak işaretlerine de dikkat etmeniz gerekmektedir).

1. Loop’lar

-forvalues- bir komutun iterasyonlarını loop etmenizi sağlar.

Örnek kod:

```
forvalues i=1/10 {  
display “ i Makrosunun değeri ‘i’ ” dir  
}
```

-forvalues- loop’lar şu şekilde yuvalanabilir:

```
forvalues i=1/5 {  
    forvalues j=1/5 {  
        display “ i Makrosunun değeri ‘i’ ” dir  
        display “ j Makrosunun değeri ‘j’ ” dir  
        scalar k=‘i’+‘j’  
        display “ Skalar k’nin degeri sudur ” k  
    }  
}
```

Dikkat ediniz: Makrolar sol-tek-tırnak işareti ve sağ-tek-tırnak işareti ile ifade edilmektedirler, skalarlar ise değişken gibi ifade edilirler. Başka komutlar da, **-while-** gibi, aynı işlevleri görürler.

2. Regresyon çıktılarına erişim

Regresyona yönelik **-reg-** ve **-qreg-** gibi komutlar kullanıldıktan sonra Stata oldukça fazla faydalı sonucu saklar. Nelerin saklandığını **-reg-** komutu sonrası **ereturn list** yazarak görebilirsiniz. Örneğin:

```
Use “C:\Stata8\auto.dta”, clear  
reg price mpg  
ereturn list
```

Burada katsayılar **e(b)** şeklinde isimlendirilen 2x1 matrisi içinde ve varyans-kovaryans matrisi de **e(V)** içerisinde saklanır. Başka faydalı sonuçlar da, R^2 veya ayarlanmış R^2 gibi, mevcuttur. Dikkat ederseniz, iki parametreyi (kesme ve mpg’nin katsayısı) tahmin ettiğimiz için **e(b)**’nin boyutu 2x1’dir. **-qreg-** komutu da aynı işlevleri görmektedir.

e(b) matrisinin içinde ne yer aldığını görmek için **matrix list e(b)** yazınız.

3. Matrisleri oluşturmak

10x2 boyutunda eksik değerlerden oluşan bir matris oluşturmak için, şunu yazınız:

matrix MyTestMat = J(10,2,.)

MyTestMat'ın nasıl görüldüğünü görmek için, şunu yazınız: **matrix list MyTestMat**

Hafızada tutulan matrisleri görmek için şunu yazınız: **matrix dir**

2x10 boyutunda 0'lardan oluşan bir matris oluşturmak için şunu yazınız: **matrix**

MyTestMat = J(2,10,0)

Varsayalım önceki kısımda yer alan regresyonu yaptınız. MyTestMat'ın ilk sırasını katsayı vektörü **e(b)** ile değiştirmek için, şunu yazınız: **matrix MyTestMat[1,1]=e(b)** (Tüm birinci sırayı değiştirmek için MyTestMat'ın (1,1) hücresine yönelmek garip gelebilir ama doğru komut budur).

Varsayalım bir forvalues loop'u kullanarak bir regresyonu çeşitli defalar tekrar etmek istiyorsunuz ve **i**'yi sayaç makrosu olarak kullanıyorsunuz. MyTestMat'ın **i**'inci satırını **i**'inci tekrarda tahmin edilen **e(b)** katsayı vektörü ile değiştirmek isterseniz, şunu yazınız: **matrix MyTestMat['i',1]=e(b)**. Makro **i**'yi çağırmak için kullanılan sol-tek-tırnak ve sağ-tek-tırnak işaretlerine dikkat ediniz.

4. 20 gözlemlili sahte bir veriseti nasıl oluşturulur?

```
clear
set more off
scalar T=20
matrix FakeData=J(T,1,.)
svmat FakeData, names(placeholder)
rename placeholder1 placeholder
replace placeholder=_n
gen x1=uniform() /*so x1~U[0,1]*/
replace x1=(4*x1)+1/*now x1 ~U[1,5]*/
gen e=invnorm(uniform())/*e ~N(0,1) uncorrelated*/
scalar beta=1
gen y=(beta*x1)+e
save FakeData.dta, replace
```

5. Belirli bir dağılıma göre rastsal sayı üretme

İlk strateji: bir tekdüze dağılımdan dönüştürelim. Casella and Berger Bölüm 5.6 ve Teorem 2.1.10'a bakınız

scalar u=uniform () tekdüze dağılımdan U[0,1] çekilen bir skalar oluşturur

generate scalar u=uniform () (verisetinizde yer alan gözlem sayısı kadar) herbiri

tekdüze dağılımdan U[0,1] çekilen ve bağımsız olan bir değişken oluşturur

gen e1=invnorm(uniform()) herbir gözlemi N(0,1) dağılımından çekilen ve bağımsız olan bir değişken oluşturur

e2=invttail(5,uniform()) herbir gözlemi t(5) dağılımından çekilen ve bağımsız olan bir değişken oluşturur

e3=invttail(1,uniform()) herbir gözlemi t(1) dağılımından çekilen ve bağımsız olan bir değişken oluşturur. Neden bununla ilgilenelim?

e4=invchi2(2,uniform()) herbir gözlemi ki-kare(2) dağılımından çekilen ve bağımsız olan bir değişken oluşturur

İkinci strateji: istediğiniz dağılımı veren bir komut bulun