

Ekonometriye Yeni Başlayanlar için Kısa bir **R** Kılavuzu

Yazan: Mahmood ARAI

Stockholm Üniversitesi, İktisat Bölümü

Türkçeleştiren: Emrah ER*

Ankara Üniversitesi, SBF İktisat Bölümü

İlk Versiyon: 11/06/2010

1 Giriş

1.1 R Hakkında

R GPL (GNU Kamu Lisansı) altında yayınlanan ve tüm ana platformlarda çalıştırılabilen bir programdır. **R**, **R Anasayfa**'sında şu şekilde tanımlanmaktadır:

"R istatistiki hesaplama ve grafik için bir özgür program ortamdır. Çok çeşitli Unix platformunda, Windows ve MacOS altında harmanlama yapabilmekte ve çalışmaktadır. R'i indirmek için lütfen tercih ettiğiniz CRAN-mirror'ı seçiniz."

Kılavuzlar ve farklı dökümanlar için **R Anasayfa**'sını ziyaret ediniz. **R** üzerine yazılmış çok sayıda kitap mevcuttur. R ile ilişkili yayınların bir listesi için R-Project¹ web sayfasını ziyaret ediniz. Dalgaard [2008] ve Fox [2002] başlangıç düzeyi için ideal kitaplardır. İleri düzey bir kitap için ise klasik bir referans olan Venables and Ripley [2002]'yi inceleyiniz. Kleiber and Zeileis [2008] özel olarak ekonometri konularını ele almaktadır. Ayrıca **CRAN Konu Görüntüleme: Sayısal Ekonometri** sayfasını inceleyiniz.

R ve L^AT_EX'i dokumak için Sweave'e² bakınız. **R** kullanarak yeniden türetilen araştırmalar için Koenker and Zeileis [2007]'ye bakınız. **R**'a atıf yapmak için ise **R Development Core Team** [2008]'e başvurulabilir.

*Bu çeviri benim **R** ile ilgili olarak gerçekleştirdiğim ilk çalışma olduğundan bazı çeviri hataları içerebilir. Hataları bana bildirirseniz sevinirim.

¹ <http://www.R-project.org/doc/bib/R.bib>

² <http://www.stat.uni-muenchen.de/~leisch/Sweave/>

1.2 Bu sayfalar hakkında

Bu ekonometriye yeni başlayanlar için bir el kılavuzudur. Bu kılavuz Mahmood Arai tarafından hazırlanan kılavuz kullanılarak oluşturulmuştur³. # sembolü yorumlar için kullanılmıştır. Dolayısıyla bir satırda # ardından gelen tüm yazılar yorumdur. > işaretini takip eden yazılar ise **R**-komutlarıdır. Örneğin:

```
> ornegim <- "ornek"
> ornegim

[1] "ornek"
```

Kod yorumlarını da içeren ve çalıştırılmayan R-kodları aşağıdaki biçimde paragraflandırılmıştır.

```
ornegim <- "ornek" # <ornegim> adlı bir nesne yaratır.
ornegim
```

< > arasında yer alan karakterler dosyaların, fonksiyonların, vs. isimleridir. <verim>, <nesnem> gibi <benimbirseyim> şeklindeki isimler genel bir veriye, nesneye, vs. ye atıfta bulunmak için kullanılmaktadır.

1.3 Nesnelere ve dosyalar

R nesnelere gibi şeylere önem vermektedir. Bir veri seti, vektör, matris, regresyon sonucu, çizim, vs. hepsi nesnelere aittir. Bir dosya içerisine bir veya daha fazla nesne kaydedilebilir. **R**-verisi içeren bir dosya bir nesne değildir fakat bir nesnelere kümesidir.

Temel olarak kullanılan tüm komutlar *fonksiyonlardır*. birsey(nesne) şeklindeki bir komut, nesne üzerinde işlem gerçekleştirir. Bu ise birçok parantez yazacağınızı anlamına gelmektedir. Parantezlerin olup olmadığını, ayrıca doğru yerde olup olmadıklarını kontrol edin.

2 İlk birkaç şey

2.1 Kurulum

R birçok farklı platform altında çalışabilmektedir ve [CRAN-mirror]'dan indirilebilir.

³ http://people.su.se/~ma/R_intro/R_intro.pdf

2.2 R ile çalışma

Bir proje için bir klasör oluşturma ve **R**'ı bu klasör altında çalıştırmak iyi bir fikirdir. Bu, çalışmanızı kaydetmeniz ve daha sonraki oturumlarda tekrar bulmanızda kolaylık sağlar.

R'ın *MS-Windows*'ta belirli bir dizinde başlamasını istiyorsanız `<start in directory>` dizinini çalışma dizini olarak tanımlamanız gerekmektedir. Bu işlem R-ikonu üzerinde farenin sağ tuşuna tıklanıp, `<properties>` kısmına gidilerek değiştirilebilir.

R içerisinde çalışma dizinini görüntüleme:

```
> getwd()
[1] "G:/r_giris"
```

Çalışma dizinini var olan başka bir dizin olan `</ee/r_ekon>` ile değiştirmek için:

```
setwd("/ee/r_ekon")
```

2.3 R'da isimlendirme

Bir nesneyi `<benim_nesnem>` veya `<benim-nesnem>` şeklinde isimlendirmeyin bunun yerine `<benim.nesnem>` kullanın.

R'da `<benim.nesnem>` ve `<Benim.nesnem>` isimlerinin iki farklı isim olduğuna dikkat edin. Sayı ile başlayan adlandırmalar (`<1a>` gibi) **R**'da kabul edilmemektedir. Bunun yerine `<a1>` kullanabilirsiniz.

Bir veri setinde değişken isimlerini nesne isimleri olarak kullanmamanız gerekir. Eğer kullanırsanız; nesne, değişkeni benzer isimdeki başka bir nesne ile gölgelendirecektir. Bu durumda değişkeni çağırdığımızda nesnenin görüntülenmesi problemi ortaya çıkacaktır.

Bu sorundan kaçınmak için:

1. Bir nesneye veri setinizdeki bir değişken ile aynı ismi vermeyin.
2. Eğer bu kuralı takip edemiyorsanız, değişkenlere, değişkenleri içeren veri seti ve değişken adıyla başvurun. Örneğin, `<df1>` veri setindeki `<ucret>` değişkenine:

```
df1$ucret.
```

komutu ile başvurulabilir.

“Gölgelendirme” problemi aynı zamanda **R** fonksiyonları ile de ilişkilidir. **R** fonksiyonları ile benzer isimlere sahip nesne isimleri kullanmayın.

`<conflicts(detail=TRUE)>` komutu yarattığınız bir nesnenin **R** paketlerinde yer alan başka bir nesne ile çakışıp çakışmadığını kontrol eder ve listeler. Yalnızca `<.GlobalEnv>` –çalışma alanınız– altında listelenen çakışmalara dikkat etmelisiniz `<.GlobalEnv>` altında listelenen tüm nesneler, **R** paketlerindeki nesneleri gölgelendirmektedir. Nesnelerin **R** paketlerinde kullanılması için gölgelendirmeye neden olan nesnelerin kaldırılması gerekmektedir.

Aşağıdaki örnek `<T>` nesnesini yaratmakta –bundan kaçınılmalıdır çünkü `<T>`, `<TRUE>` anlamına gelmektedir– ve çakışmaları kontrol edip, `<T>`'yi kaldırarak çakışma sorununu çözmektedir.

```
T <- "zaman"
conflicts(detail=TRUE)
rm(T)
conflicts(detail=TRUE)
```

`<c,C,D,F,I,q,t,T>` gibi tek-harflik kelimelerin de isim olarak kullanımından da kaçınılmalıdır. Bunların hepsinin **R**'da özel anlamları vardır.

Dosya uzantıları

R-kodu içeren dosyalarınızın uzantısında `<R>` kullanmak iyidir. `<lab1.R>` şeklindeki bir dosya R-kodlarını içeren bir metin dosyası olacaktır.

`<rda>` uzantısı çalışma görüntüleri (yani `<save()>` ile yaratılan dosyalar) için idealdir. `<lab1.rda>` şeklindeki bir dosya R-nesnelerini içerecektir.

Kaydedilmiş çalışma görüntüsü için varsayılan isim `<.RData>`'dir. `<RData>`'yı dosya uzantısı olarak kullandığınızda bir dosyayı `<.RData>` olarak adlandırmadığınıza dikkat edin.

2.4 Çalışma alanı nesnelerini ve görüntülerini kaydetme

`<DataWageMacro.rda>` dosyasını indirin⁴.

`<lnu>` ve `<macro>` veri setlerini içeren `<DataWageMacro.rda>` dosyasını

```
load("DataWageMacro.rda")
ls() # nesneleri listeler
```

komutu ile okuyabilirsiniz.

Aşağıdaki komut `<lnu>` nesnesini `<verim.rda>` dosyasına kaydeder.

```
save(lnu, file="verim.rda")
```

⁴http://people.su.se/~ma/R_intro/data/

R'ı tekrar aynı dizinde başlattığımızda otomatik olarak yüklenecek olan çalışma alanının bir görüntüsünü kaydetmek için,

```
save.image()
```

komutunu kullanın.

Çalışma alanınızın görüntüsünü **R**'dan çıkarken karşınıza gelen `<Save workspace image? [y/n/c] :>` sorusuna `<yes>` cevabını vererek de kaydedebilirsiniz.

Bu şekilde çalışma alanınızın görüntüsü `<.RData>` gizli dosyasına kaydedilecektir.

Cari çalışma alanınızın görüntüsünü kaydedebilir ve `<goruntum.rda>` adını verebilirsiniz.

```
save.image("goruntum.rda")
```

2.5 Genel seçenekler

`<options(>` komutu hesaplamaların ve sonuçları görüntülemenin farklı özelliklerini ele alan bir dizi seçeneğin düzenlenmesinde kullanılabilir.

Şimdi birkaç faydalı seçeneğe bakalım. Satır uzunluğunu 60 karaktere sınırlayarak başlıyoruz.

```
> options(width=60)

options(prompt=" R> ") # promptu R> olarak değiştirir.
options(scipen=3)      # R version 1.8'den. Bu seçenek
                      # R'a sayıları üstel format
                      # yerine sabit formatta
                      # göstermesini söyler. Örneğin,
                      # <exp(28)>'in sonucu olarak
                      # <1.446257e+12> yerine
                      # <1446257064291>.

options()              # seçenekleri görüntüler.
```

2.6 Yardım alma

```
help.start()          # yardım sayfalarını açar.
help(lm)              # <lm> doğrusal model
                      # (linear model)
                      # ile ilgili yardım.
?lm                   # yukarıdaki ile aynı.
```

3 Temel komutlar

```
ls()                # tüm nesnelere listeler.
ls.str()           # tüm nesnelere detaylarını listeler.
str(nesnem)        # <nesnem>'in detaylarını listeler.
list.files()       # dizindeki tüm dosyaları listeler.
dir()              # dizindeki tüm dosyaları listeler.
nesnem             # basit olarak nesneyi görüntüler.
rm(nesnem)         # <nesnem> nesnesini kaldırır.
rm(list=ls())      # çalışma alanındaki tüm nesnelere
                  # kaldırır.

save(nesnem,
file="nesnem.rda") # <nesnem> nesnesini <nesnem.rda>
                  # dosyasına kaydeder.

load("calismam.rda") # "calismam.rda"yı hafızaya yükler.
summary(verim)       # <verim> için basit istatistikleri
                  # görüntüler.

hist(x,freq=TRUE)   # <x> nesnesine ait histogramı
                  # görüntüler.
                  # <freq=TRUE> frekansı gösterir ve
                  # <freq=FALSE> olasılıkları gösterir.

q()                 # R'dan çıkar.
```

Bir komutun çıktısı doğrudan `< <- >` kullanılarak bir nesneye atanabilir ki bu durumda nesneye bir değer atanmış olur. Aşağıda yer alan kodun ilk satırı 1,2 ve 3 değerlerini içeren `<VV>` isimli bir nesne yaratır. İkinci satır ise `<VV>` isimli bir nesne yaratır ve `<VV>` nesnesinin içeriğini görüntüler.

```
> VV <- c(1,2,3)
> (VV <- 1:2)

[1] 1 2
```

4 Veri düzenleme

4.1 Düz metin formatında veriyi okuma:

Sütunlarda veri

Bu örnekteki veri seti bir metin dosyasındandır. `<tmp.txt>` dosyası ilk satırında

değişken isimlerini içermektedir (isimler boşluk ile ayrılmıştır) ve bu değişkenlerin değerleri diğer satırlarda yer almaktadır (değerler boşluk ile ayrılmıştır).

Aşağıdaki kod <tmp.txt> dosyasının içeriğini okur ve bunu <dat> isimli bir nesneye atar.

```
> FILE <- "http://people.su.se/~ma/R_intro/data/tmp.txt"
> dat <- read.table(
+ file= FILE,
+ header = TRUE)
> dat
```

	wage	school	public	female
1	94	8	1	0
2	75	7	0	0
3	80	11	1	0
4	70	16	0	0
5	75	8	1	0
6	78	11	1	0
7	103	11	0	0
8	53	8	0	0
9	99	8	1	0

<header = TRUE> argümanı ilk satırda değişken isimlerinin olduğunu belirtir. <dat> nesnesi **R**'da adlandırıldığı şekliyle bir veri setidir.

Eğer <tmp.txt> veri dosyasındaki sütunlar <,> ile ayrılmışsa, kod yazımı,

```
read.table("tmp.txt", header = TRUE, sep=",")
```

şeklinde olmalıdır.

Eğer ondalık karakteri <.> şeklinde ise bunun tanımlanması gereklidir. Eğer ondalık karakteri <,> şeklinde ise <read.csv> komutunu kullanabilir ve ardından gelen argümanı <dec=","> fonksiyonu içerisinde tanımlayabilirsiniz.

4.2 Tablo verisinde mevcut olmayan gözlem (NA) ve sınırlayıcılar

<data1.txt> dosyası aşağıdaki veriyi içermektedir:

```
1 . 9
6 3 2
```

Burada ikinci sütünde (değişkende) yer alan ilk gözlem eksiktir ve <.> olarak kodlanmıştır.

R'a `<.>`'nin eksik değer olduğunu bildirmek için `<na.strings=".>` argümanını kullanırız.

```
> FILE <- "http://people.su.se/~ma/R_intro/data/data1.txt"
> read.table(file=FILE ,na.strings=". " )

  V1 V2 V3
1  1 NA  9
2  6  3  2
```

Bazen sütunlar boşluklardan farklı ayraçlar ile ayrılırlar. Ayraç, örneğin, `<,>` olabilir ki bu durumda `<sep=",>` argümanını kullanmamız gerekir.

Eğer sütunlar `<,>` ile ayrılmışsa ve bazı sütunlarda aşağıdakine benzer şekilde boşluklar varsa bu durumda `<na.strings=".>` argümanının çalışmayacağına dikkat edin. NA aslında iki boşluk şeklinde kodlanır, bir nokta ve iki boşluk, ve `<na.strings=".>` şeklinde ifade edilmesi gerekir.

```
1, . ,9
6, 3 ,2
```

Bazen eksik değerler basitçe `<boşluk>` şeklindedir.

```
1 9
6 3 2
```

İlk satırda 1 ile 9 arasında iki boşluk olduğuna dikkat edin. Bu ikinci sütundaki değer boşluk olduğu anlamına gelir. Bu ise eksik değer anlamındadır. Burada önemli olan şey `<na.strings="">` ile birlikte `<sep=" ">` tanımlanması gerektiğidir.

4.3 Diğer formatlarda veri okuma ve yazma

`<foreign>` kitaplığımı çeşitli standart paketlerin veri formatında verileri okumak için ekleyin. Örnekler arasında SAS, SPSS, STATA, vs. yer almaktadır.

```
library(foreign)
# <wage.dta> verisini okur ve <lnu> nesnesine atar.
lnu <- read.dta(file="wage.dta")
```

`<read.ssd()>`, `<read.spss()>`, vs. SAS ve SPSS formatlarında verileri okumak için `foreign` paketinin içerisinde yer alan komutlardır.

Yabancı formatta veri yazmak da kolaydır. Aşağıdaki kodlar `<lnu>` nesnesini `<lnu.dta>` adındaki Stata dosyasına yazdırır.

```
library(foreign)
write.dta(lnu,"lnunew.dta")
```

4.4 Veri seti nesnesinin içeriğini inceleme

Burada **İsveç Yaşam Düzeyi Anketi** LNU 1991'den elde edilen veri seti kullanılmaktadır.

```
> FILE <- "http://people.su.se/~ma/R_intro/data/lnu91.txt"
> lnu <- read.table(file=FILE, header = TRUE)
```

<lnu> verisini <attach(lnu)> ile iliştiirmek <lnu> veri setine ve veri setindeki değişkenlere isimleriyle erişimi sağlar. Eğer <lnu> veri setini iliştiirmediyse, <lnu> veri setinde yer alan <female> değişkenine atıfta bulunmak için <lnu\$female> komutunu kullanabilirsiniz. Veri setine iliştiirilmiş olarak ihtiyacınız kalmadığında iliştiirme işlemini <detach(> komutuyla geri alabilirsiniz.

lnu veri setinin içeriğinin tanımlanması.

```
> str(lnu)          # Veri yapısının tanımı

'data.frame':      2249 obs. of  6 variables:
 $ wage   : int  81 77 63 84 110 151 59 109 159 71 ...
 $ school : int  15 12 10 15 16 18 11 12 10 11 ...
 $ expr   : int  17 10 18 16 13 15 19 20 21 20 ...
 $ public : int   0 1 0 1 0 0 1 0 0 0 ...
 $ female : int   1 1 1 1 0 0 1 0 1 0 ...
 $ industry: int  63 93 71 34 83 38 82 50 71 37 ...

> summary(lnu)     # Veri tanımlamasının özeti

      wage          school          expr
Min.   : 17.00    Min.   : 4.00    Min.   : 0.00
1st Qu.: 64.00    1st Qu.: 9.00    1st Qu.: 8.00
Median : 73.00    Median :11.00    Median :18.00
Mean   : 80.25    Mean   :11.57    Mean   :18.59
3rd Qu.: 88.00    3rd Qu.:13.00    3rd Qu.:27.00
Max.   :289.00    Max.   :24.00    Max.   :50.00

      public        female        industry
Min.   :0.0000    Min.   :0.0000    Min.   :11.00
1st Qu.:0.0000    1st Qu.:0.0000    1st Qu.:50.00
Median :0.0000    Median :0.0000    Median :81.00
Mean   :0.4535    Mean   :0.4851    Mean   :69.74
3rd Qu.:1.0000    3rd Qu.:1.0000    3rd Qu.:93.00
Max.   :1.0000    Max.   :1.0000    Max.   :95.00
```

4.5 Veri setinde değişken yaratma ve değişken çıkartma

Burada `<logwage>` değişkenini `<wage>` değişkeninin logaritması olarak tanımlıyoruz. Sonrasında ise değişkeni çıkartıyoruz.

```
> lnu$logwage <- log(lnu$wage)
> lnu$logwage <- NULL
```

Orjinal değişkenlerin basit dönüştürmeleri olan değişkenleri yaratmaya gerek olmadığına dikkat edin. Dönüştürmeyi doğrudan hesaplamaların ve tahminlerin içerisinde gerçekleştirebilirsiniz.

4.6 Veri setinde yer alan değişkenlerin alt kümesini seçme

```
# lnu veri setinde (wage,female)
# değişkenlerinin altkümesini okuma.
lnu.female <- subset(lnu, select=c(wage,female))

# Veri setinde iki nesneyi (değişkeni)
# bir araya getirme.
attach(lnu)
lnu.female <- data.frame(wage,female)

# lnu'da female hariç tüm değişkenleri okuma.
lnux <- subset(lnu, select=-female)

# wage'den public'e kadar değişkenleri okuma.
lnuxx <- subset(lnu, select=wage:public)
```

4.7 Veri setinde gözlemlerin altkümesini seçme

```
attach(lnu)

# Bir değişkende var olan eksik değer
# içeren gözlemleri silme.
lnu <- na.omit(lnu)

# Sadece female değişkenine ait veriyi saklama.
fem.data <- subset(lnu, female==1)

# Sadece female ve public employees
# değişkenlerine ait veriyi saklama.
fem.public.data <- subset(lnu, female==1 & public==1)

# wage > 90 olan tüm gözlemleri seçme.
```

```
highwage <- subset(lnu, wage > 90)
```

4.8 Değişkenlerin değerlerini değiştirme

Okul değişkeni içerisinde yer alan değerleri değiştirerek bireyin üniversite eğitimi alıp almadığını gösteren bir değişken yaratıyoruz.

Okul değişkenini kopyalayın.

```
> lnu$university <- lnu$school
```

Üniversite değerini eğer okula gidilen yıl sayısı 13 yıldan az ise 0 değeri ile değiştirin.

```
> lnu$university <- replace(lnu$university,  
+ lnu$university<13, 0)
```

Üniversite değerini eğer okula gidilen yıl sayısı 12 yıldan fazla ise 1 değeri ile değiştirin.

```
> lnu$university <- replace(lnu$university,  
+ lnu$university>12, 1)
```

<lnu\$university> değişkeni şimdi üniversite eğitimi için bir kukla değişken haline gelmiştir. Kaydettikten sonra veri setini tekrar iliştiirmeyi unutmayın. Kategori değişkenleri yaratmak için <cut> komutunu kullanabilirsiniz.

```
> attach(lnu, warn.conflicts=FALSE)  
> table(university)
```

```
university  
  0    1  
1516 733
```

Kukla değişken yaratmak için şu şekilde ilerliyoruz:

```
> university <- school > 12  
> table(university)
```

```
university  
FALSE TRUE  
1516  733
```

Aslında genellikle kukla değişken yaratmamıza gerek yoktur. `<school > 12>` üzerinden doğrudan işlemleri gerçekleştirebiliriz.

```
> table(school > 12)

FALSE TRUE
1516   733
```

4.9 Eksik değerleri değiştirmek

Bir vektör yaratıyoruz. Bir değeri eksik değer olarak yeniden kodluyoruz ve ardından eksik değeri orijinal değeri ile değiştiriyoruz.

```
a <- c(1,2,3,4) # vektör yaratır
is.na(a) <- a ==2 # a==2 'yi NA olarak kodlar
a <- replace(a, is.na(a), 2)# NA'yı 2 ile değiştirir
# or
a[is.na(a)] <- 2
```

4.10 Faktörler

Bazen değişkenlerimizin çeşitli aralıklara karşılık gelen uygun düzeydeki kategori değişkenleri biçiminde kullanılması için yeniden tanımlanması gerekir. 9 yıla kadar, 10 ila 12 yıl arası ve 12 yıl üstü için okul kategorileri oluşturmak isteyebiliriz. Bu `<cut(>` kullanılarak kodlanabilir. En düşük kategoriyi dahil etmek için `<include.lowest=TRUE>` argümanını kullanırız.

```
> SchoolLevel <- cut(school,
+ c(9,12, max(school), include.lowest=TRUE))
> table(SchoolLevel)

SchoolLevel
(1,9] (9,12] (12,24]
 608   908   733
```

Her düzey için etiket ayarlanabilir. Bir önceki bölümde yaratılan üniversite değişkenini ele alalım.

```
> SchoolLevel <- factor(SchoolLevel,
+ labels=c("basic", "gymnasium", "university"))
> table(SchoolLevel)
```

```

SchoolLevel
  basic  gymnasium university
    608    908    733

```

Yukarıdaki gibi tanımlanan bir faktör, örneğin regresyon modelinde kullanılabilir. Referans kategori en düşük değere sahip düzeydir. <Basic>'e karşılık gelen en düşük değer 1 'dir ve <Basic> sütunu karşılaştırma matrisinde yer almamaktadır. Taban kategoriyi değiştirmek bu sütun yerine başka bir sütunu dışlayacaktır. Bu aşağıda yer alan örnekte gösterilmiştir.

```

> contrasts(SchoolLevel)

          gymnasium university
basic           0           0
gymnasium       1           0
university       0           1

> contrasts(SchoolLevel) <-
+ contr.treatment(levels(SchoolLevel),base=3)
> contrasts(SchoolLevel)

          basic gymnasium
basic           1           0
gymnasium       0           1
university       0           0

```

Aşağıdaki komut <school> değişkenini sayısal bir değişken olarak tanımlar.

```

> lnu$school <- as.numeric(lnu$school)

```

4.11 Veriyi grup bazında toplulaştırmak

V1, V2 ve V3 gibi 3 değişkenden oluşan basit bir veri seti yaratalım. V1 grup birimi ve V2 ile V3 iki sayısal değişken olsunlar.

```

> (df1 <- data.frame(V1=1:3, V2=1:9, V3=11:19))

  V1 V2 V3
1  1  1 11
2  2  2 12
3  3  3 13
4  1  4 14
5  2  5 15
6  3  6 16

```

```
7 1 7 17
8 2 8 18
9 3 9 19
```

<aggregate> komutunu kullanarak <sum>, <mean>, vb. grup özelliklerini içeren yeni bir data.frame oluşturalım. Burada toplam (sum) fonksiyonu <V1> grup birimi ile <df1[,2:3]> üzerine yani <df1>'in ikinci ve üçüncü sütunlarına uygulanır.

```
> (aggregate.sum.df1 <-
+ aggregate(df1[,2:3],list(df1$V1),sum))
```

```
  Group.1 V2 V3
1         1 12 42
2         2 15 45
3         3 18 48
```

```
> (aggregate.mean.df1 <-
+ aggregate(df1[,2:3],list(df1$V1),mean))
```

```
  Group.1 V2 V3
1         1  4 14
2         2  5 15
3         3  6 16
```

<Group.1> değişkeni, grupları tanımlayan bir faktördür.

Aşağıda toplulaştırma (aggregate) fonksiyonuna ilişkin bir örnek yer almaktadır. Birim tanımlayıcı <dat\$id>'yi içeren <dat> şeklinde bir veri setimiz olduğunu varsayalım. Zaman içerisinde tekrar tekrar gözlemlenen birimler <dat\$Time> değişkeni ile gösterilsin.

```
> (dat <- data.frame(id=rep(11:12,each=2),
+ Time=1:2, x=2:3, y =5:6))
```

```
  id Time x y
1 11    1 2 5
2 11    2 3 6
3 12    1 2 5
4 12    2 3 6
```

Bu veri setinde yer alan tüm değişkenler için grup ortalamalarını hesaplar ve <Time> değişkeni ile otomatik olarak oluşturulmuş olan grup tanımlayıcı değişken <Group.1>'i dışlar.

```
> (Bdat <- subset(aggregate(dat,list(dat$id),FUN=mean),
+ select=-c(Time,Group.1)))
```

```

      id  x  y
1 11 2.5 5.5
2 12 2.5 5.5

```

<id> uzunluğunda ve <id> içerisinde yer alan her bir gözlem için tekrarlayan grup ortalamalarını içeren bir veri seti yaratmak için <Bdat> ve <dat\$id>'yi birleştirin.

```

> (dat2 <- subset(merge(data.frame(id=dat$id),Bdat),
+ select=-id))

      x  y
1 2.5 5.5
2 2.5 5.5
3 2.5 5.5
4 2.5 5.5

```

Şimdi <id> ve <Time> tanımlayıcılarını ve diğer tüm değişkenlerin ortalama değerlerden sapmalarını içeren bir veri seti yaratabiliriz.

```

> (within.data <- cbind(id=dat$id, Time=dat$Time,
+ subset(dat,select=-c(Time,id)) - dat2))

      id Time  x  y
1 11     1 -0.5 -0.5
2 11     2  0.5  0.5
3 12     1 -0.5 -0.5
4 12     2  0.5  0.5

```

4.12 Farklı veri setlerini kullanma

Genellikle farklı veri setlerinden verileri kullanmamız gerekir. **R**'da bu verileri, diğer birçok istatistik paket programında olduğu üzere tüm verinin aynı anda ve bir tablo şeklinde var olduğu biçimde, aynı veri seti içerisine koymamıza gerek yoktur.

Örneğin, değişkenler aynı uzunluğa sahip olduğu sürece (aynı gözlem sayısı) ve aynı sırada (tüm değişkenlerdeki i'inci gözlem aynı birime karşılık gelir) olduğu sürece bir veri setinden bir değişken, diğer bir veri setinden başka bir değişken kullanarak bir regresyon kurmak mümkündür. Aşağıdaki iki veri setini ele alalım:

```

> data1 <- data.frame(wage =
+ c(81,77,63,84,110,151,59,109,159,71),

```

```

+ female = c(1,1,1,1,0,0,1,0,1,0),
+ id      = c(1,3,5,6,7,8,9,10,11,12))
> data2 <- data.frame(experience =
+ c(17,10,18,16,13,15,19,20,21,20),
+ id      = c(1,3,5,6,7,8,9,10,11,12))

```

Şimdi veri setlerini birleştirmeden her ikisinden de değişkenleri kullanabiliriz. <data1\$wage>'in <data1\$female> ve <data2\$experience> üzerine regresyonunu gerçekleştirelim.

```

> lm(log(data1$wage) ~ data1$female + data2$experience)

Call:
lm(formula = log(data1$wage) ~ data1$female + data2$experience)

Coefficients:
(Intercept)      data1$female  data2$experience
    4.641120         -0.257909          0.001578

```

Farklı veri setlerinde yer alan değişkenleri tek bir veri setine yerleştirerek analizimizi bu yeni veri seti üzerinden de gerçekleştirebiliriz.

```

> (data3 <- data.frame(data1$wage,
+ data1$female,data2$experience))

  data1.wage data1.female data2.experience
1          81           1             17
2          77           1             10
3          63           1             18
4          84           1             16
5         110           0             13
6         151           0             15
7          59           1             19
8         109           0             20
9         159           1             21
10         71           0             20

```

Veri setlerini birleştirebiliriz. Eğer her iki veri setinde de aynı değişken yer alıyorsa, veri bu değişken ile tutarlı biçimde birleştirilir.

```

> (data4 <- merge(data1,data2))

  id wage female experience
1  1  81      1          17
2  3  77      1          10

```

3	5	63	1	18
4	6	84	1	16
5	7	110	0	13
6	8	151	0	15
7	9	59	1	19
8	10	109	0	20
9	11	159	1	21
10	12	71	0	20

Diğer paket programlarda olduğu üzere gözlemlerin <id>'de tanımlandığı şekliyle aynı sırada olmasına gerek olmadığına dikkat ediniz.

Eğer iki veri setini ortak bir değişken kullanarak eşleştirmek istiyorsak ve bu değişken veri setlerinde farklı isimlere sahipse, isimleri aynı olacak biçimde değiştirebilir veya veri olduğu şekliyle kullanıp verileri eşleştirirken kullanılacak değişkenleri tanımlayabiliriz. Eğer <data2> ve <data1> veri setlerinde eşleştirilecek değişken <id2> ve <id> ise aşağıdaki yazım kullanılabilir:

```
merge(data1,data2, by.x="id", by.y="id2")
```

<by.x="id", by.y="id2"> argümanları data1'da eşleştirme değişkeninin id ve data2'de id2 olduğunu ifade eder.

Aynı zamanda veri setlerini <data.frame> veya <cbind> yardımıyla varolan sırasını birleştirebilirsiniz. Bu şekilde veri, veri setindeki sırasını gözlem gözlem eşleştirilir. Bu aşağıdaki örnekte gösterilmiştir.

```
> data1.noid <- data.frame(wage = c(81,77,63),
+ female = c(1,0,1))
> data2.noid <- data.frame(experience = c(17,10,18))
> cbind(data1.noid,data2.noid)

  wage female experience
1   81      1         17
2   77      0         10
3   63      1         18
```

Eğer bir veri setinin sonuna gözlem eklemek isterseniz, <rbind> kullanabilirsiniz. Aşağıdaki örnek <data4>'teki 2.,3. ve 4. sütunları iki parçaya ayırarak <rbind> ile tekrar bir araya getirmektedir.

```
> data.one.to.five <- data4[1:5,2:4]
> data.six.to.ten <- data4[6:10,2:4]
> rbind(data.one.to.five,data.six.to.ten)
```

	wage	female	experience
1	81	1	17
2	77	1	10
3	63	1	18
4	84	1	16
5	110	0	13
6	151	0	15
7	59	1	19
8	109	0	20
9	159	1	21
10	71	0	20

5 Temel istatistikler

Bir veri setindeki tüm değişkenler için özet istatistikler:

```
summary(mydata)
```

Bir değişkene ait Ortalama (Mean), Ortanca (Median), Standart Sapma (Standard deviation), Maksimum ve Minimum:

```
mean (myvariable)
median (myvariable)
sd (myvariable)
max (myvariable)
min (myvariable)
# 10, 20, ..., 90 yüzdeleri hesapla
quantile(myvariable, 1:9/10)
```

R `<sum>`, `<mean>`, vs.'yi `<NA>` içeren bir nesne üzerine hesapladığında, `<NA>` sonucunu verir. Bu fonksiyonları verinin var olduğu yerdeki gözlemlere uygulamak için `<na.rm=TRUE>` argümanını kullanmalıyız. Diğer bir alternatif ise `<NA>` içeren tüm veri satırlarını `<na.omit>` ile dışlamaktır.

```
> a <- c(1,NA, 3,4)
> sum(a)

[1] NA

> sum(a,na.rm=TRUE)

[1] 8
```

```
> table(a, exclude=c())
```

```
a
  1   3   4 <NA>
  1   1   1   1
```

Ayrıca, NA'ları dışlayan ve toplamı hesaplayan

```
<sum(na.omit(a))>
```

veya <a> içerisinde NA olmayan (lis.na) elemanların toplamını hesaplayan

```
<sum(a[!is.na(a)])>
```

kullanabilirsiniz.

5.1 Çizelgeleme

İlk olarak veri setini okutalım.

Çapraz Çizelgeleme

```
> attach(lnu, warn.conflicts=FALSE)
> table(female,public) # frekansları verir

      public
female  0   1
  0 815 343
  1 414 677

> (ftable.row <- cbind(table(female,public),
+ total=table(female)))

      0   1 total
0 815 343 1158
1 414 677 1091

> (ftable.col <- rbind(table(female,public),
+ total=table(public)))

      0   1
0      815 343
1      414 677
total 1229 1020
```

```

# Bunu deneyin:
# satırlara göre göreli frekans: female
ftable.row/c(table(female))
# sütunlara göre göreli frekans: public
ftable.col/rep(table(public),each=3)
# rep(table(public),each=3)
# table(public)'deki her değeri 3 kez tekrar eder

```

Kategoriler ile çeşitli istatistikler oluşturma. Aşağıdaki komut erkekler (males) ve kadınlar (female) için ortalama ücretleri (wage) verir.

```

> tapply(wage, female, mean)
      0      1
88.75302 71.23190

```

<length>, <min>, <max>, vs. kullanımı erkekler ve kadınlar için farklı gözlem sayısı, minimum, maximum, vs. oluşturur.

```

> tapply(wage, female, length)
      0      1
1158 1091

```

Aşağıda yer alan örnek özel (private) sektör ve kamu (public) sektöründeki erkekler ve kadınlar için ortalama ücretleri verir.

```

> tapply(wage, list(female,public), mean)
      0      1
0 89.52883 86.90962
1 71.54589 71.03988

```

Aşağıdaki komut ise eş uzunlukta bir vektör yaratarak grup bazında ortalamaları hesaplar. Eş uzunluk her grubun tüm elemanları için grup istatistiklerinin elde edildiği anlamına gelmektedir. Erkekler ve kadınlar için ortalama ücret:

```

> attach(lnu, warn.conflicts=FALSE)
> lnu$wage.by.sex<- ave(wage,female,FUN=mean)

```

<mean> fonksiyonu <min>, <max>, <length>, vs. ile değiştirilebilir.

6 Matrisler

R'da matrisi şu şekilde tanımlarız (**R**'da bakınız ?matrix):

3 satır ve 4 sütundan oluşan ve sütunlarında 1'den 12'ye kadar elemanları içeren bir matris:

```
> matrix(1:12,3,4)
      [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

3 satır ve 4 sütundan oluşan ve satırlarında 1'den 12'ye kadar elemanları içeren bir matris:

```
> (A <- matrix(1:12,3,4,byrow=TRUE))
      [,1] [,2] [,3] [,4]
[1,]    1    2    3    4
[2,]    5    6    7    8
[3,]    9   10   11   12

> dim(A)          # Matrisin boyutu
[1] 3 4

> nrow(A)         # Satır sayısı, dim(A)[1] ile aynı
[1] 3

> ncol(A)        # Sütun sayısı, dim(A)[2] ile aynı
[1] 4
```

6.1 Endeksleme

Bir matrisin elemanları, matris adından sonra köşeli parantez kullanılarak ve aralarına virgül olacak şekilde satır ve sütünlara başvurularak listelenebilir. Benzer bir endeklemeyi diğer nesnelere elemanlarını listelemek için de kullanabilirsiniz.

```
A[3,]          # 3. satırı listeleme
A[,3]          # 3. sütunu listeleme
A[3,3]         # 3. satır ve üçüncü sütun
A[-1,]         # 1. satır hariç matrisin tamamı
A[,-2]         # 2. sütun hariç matrisin tamamı
```

Matris elemanları üzerinde bazı koşulların değerlendirilmesi

```
> A>3 # 3'ten büyük elemanlar
      [,1] [,2] [,3] [,4]
[1,] FALSE FALSE FALSE TRUE
[2,] TRUE TRUE TRUE TRUE
[3,] TRUE TRUE TRUE TRUE

> A==3 # 3'e eşit elemanlar
      [,1] [,2] [,3] [,4]
[1,] FALSE FALSE TRUE FALSE
[2,] FALSE FALSE FALSE FALSE
[3,] FALSE FALSE FALSE FALSE
```

Bazı koşulları yerine getiren elemanları listeleme

```
> A[A>6] # 6'dan büyük tüm elemanlar
[1] 9 10 7 11 8 12
```

6.2 Skalar Matris

Skalar matris matrisin özel bir biçimidir. Köşegen-dışı elemanları sıfır, köşegen üzerindeki elemanları ise aynıdır ve eşit sayıda satır ile sütuna sahip olan kare bir matristir. Aşağıdaki örnek matrislerin köşegenleri ile ilişkili bazı matris özelliklerini göstermektedir. Ayrıca `?upper.tri` ve `?lower.tri`'ye de bakınız.

```
> diag(2,3,3)
      [,1] [,2] [,3]
[1,] 2 0 0
[2,] 0 2 0
[3,] 0 0 2

> diag(diag(2,3,3))
[1] 2 2 2
```

6.3 Matris işlemleri

Matrisin devriği

Bir matrisin satırlarını sütun, sütunlarını da satır yapmak matrisin devriğini verir.

```
> t(matrix(1:6,2,3)) #
      [,1] [,2]
[1,]    1    2
[2,]    3    4
[3,]    5    6
```

(1:6,2,3) matrisi ve (1:6,3,2, byrow=T) matrisini deneyin.

Toplama ve çıkarma

Toplama ve çıkarma işlemleri boyutları aynı olan matrislere veya bir skalar ve bir matrise uygulanabilir.

```
# Bunu deneyin
A <- matrix(1:12,3,4)
B <- matrix(-1:-12,3,4)
C1 <- A+B
D1 <- A-B
```

Skalar çarpımı

```
# Bunu deneyin
A <- matrix(1:12,3,4); TwoTimesA = 2*A
c(2,2,2)*A
c(1,2,3)*A
c(1,10)*A
```

Matris çarpımı

Matris çarpımı için **R** <% * %> kullanır ve bu sadece matrisler uyumlu ise çalışır.

```
E <- matrix(1:9,3,3)
crossproduct.of.E <- t(E)%*%E
# veya çapraz çarpımları elde etmenin daha etkin diğer bir yolu:
crossproduct.of.E <- crossprod(E)
```

Matris tersi

Kare matris **A**'nın tersi \mathbf{A}^{-1} şeklinde gösterilir ve **A** ile çarpıldığında birim matris (ana köşegen üzerindeki tüm elemanları 1, köşegen-dışı elemanları 0 olan matris) oluşturan matris olarak tanımlanır.

$$\mathbf{A}\mathbf{A}^{-1} = \mathbf{A}^{-1}\mathbf{A} = \mathbf{I}$$

```
FF <- matrix((1:9),3,3)
detFF<- det(FF)          # determinanı kontrol eder

B <- matrix((1:9)^2,3,3) # tersi alınabilir bir
                          # matris oluřturur

Binverse <- solve(B)
Identity.matrix <- B%*%Binverse
```

7 En küçük kareler

EKK kullanarak doğrusal bir regresyon modeli oluşturmak için kullanılan fonksiyon `<lm()>`'dir. Aşağıdaki örnekte bağımlı değişken `<log(wage)>` ve açıklayıcı değişkenler `<school>` ve `<female>`'dir. Sabit terim varsayılan olarak dahil edilmektedir. `<lm>` veri setini tanımlamamıza gerek yoktur çünkü bu değişkenleri içeren veri seti zaten iliştilmiş durumdadır. Regresyon sonucu `<reg.model>` isimli nesneye atanmaktadır. Bu nesne ileride `<lm>` kullanımını gösteren birkaç örnekten sonra gösterilecek olan birçok ilginç regresyon sonucunu içermektedir.

Veri setini oku first.

```
> reg.model <- lm (log(wage) ~ school + female)
> summary (reg.model)

Call:
lm(formula = log(wage) ~ school + female)

Residuals:
    Min       1Q   Median       3Q      Max
-1.46436 -0.15308 -0.01852  0.13542  1.10402

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  4.087730    0.022203   184.10 <2e-16 ***
school        0.029667    0.001783    16.64 <2e-16 ***
female       -0.191109    0.011066   -17.27 <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2621 on 2246 degrees of freedom
Multiple R-squared:  0.2101,    Adjusted R-squared:  0.2094
F-statistic: 298.8 on 2 and 2246 DF,  p-value: < 2.2e-16
```

Bazen regresyonu veri setinin bir alt kümesi üzerinde kurmak isteyebiliriz.

```
lm (log(wage) ~ school + female, subset=wage>100)
```

Bazen de modelde değişkenlerin dönüştürülmüş değerlerini kullanmak isteyebiliriz. Dönüştürme `<I()>` fonksiyonu içerisinde verilmelidir. `I()` birim fonksiyonu anlamına gelmektedir. `<expr^2>`, `<expr>`'nin karesidir.

```
lm (log(wage) ~ school + female + expr + I(expr^2))
```

Etkileşimli değişkenler: <female>, <school>

```
lm (log(wage) ~ female*school, data=lnu)
```

benzer biçimde:

```
lm (log(wage) ~ female + school + female:school, data= lnu)
```

Sabit terimsiz bir model.

```
reg.no.intercept <- lm (log(wage) ~ female - 1)
```

Sadece sabit terim içeren bir model.

```
reg.only.intercept <- lm (log(wage) ~ 1 )
```

Aşağıdaki örnek <female> ve <public> değişkenleri ile tanımlandığı şekliyle özel ve kamu sektöründeki erkekler ve kadınlar için doğrusal model (<lm>) tahmin eder. <lnu> verisi 4 farklı hücreye ayrılmıştır: (0,0) özel sektördeki erkekler, (1,0) özel sektördeki kadınlar, (0,1) kamu sektöründeki erkekler ve (1,1) kamu sektöründeki kadınlar. Elde edilen <by.reg> nesnesi bir listedir ve her elemanın özetini (<summary(>) görüntülemek için <lapply> (liste uygula) kullanılır.

```
by.reg <- by(lnu, list(female,public),
             function(x) lm(log(wage) ~ school, data=x))
# ayrı regresyonların özeti
lapply(by.reg, summary)
# listedeki ikinci eleman için özet
# yani özel sektördeki kadınlar için.
summary(by.reg[[2]])
```

Aşağıdaki komut erkek ve kadın işçiler için değişkenlerin ortalamalarını listeler (ilk satır), iki veri setinden by.female.lnu adında bir liste yaratır (ikinci satır) ve erkek ve kadın işçiler için bir regresyon tahmini gerçekleştirir (üçüncü vfe dördüncü satır).

```
by(lnu, list(female), mean)
by.female.lnu <- by(lnu, list(female),
                   function(x) x); str(by.female.lnu)
summary(lm(log(wage) ~ school, data=by.female.lnu[[1]]))
summary(lm(log(wage) ~ school, data=by.female.lnu[[2]]))
```

7.1 Model formülü ve sonuçlarını görüntüleme

Model formülü

```
(equation1 <- formula(reg.model))  
log(wage) ~ school + female
```

Tahmin edilen katsayılar

```
> coefficients(reg.model)      # <coefficients> (katsayılar),  
(Intercept)      school      female  
4.08772967  0.02966711 -0.19110920
```

Standart hatalar

```
> coef(summary(reg.model))[,2]  
(Intercept)      school      female  
0.022203334 0.001782725 0.011066176
```

<coef(summary(reg.model))[,1:2]> komutu,
<Estimate> (tahminleri) ve <Std.Error> (standart hataları) görüntüler.

t-değerleri

```
> coef(summary(reg.model))[,3]  
(Intercept)      school      female  
184.10432      16.64144     -17.26967
```

Ayrıca <coef(summary(reg.model))> komutunu da deneyin. Benzer biçimde
lm-nesnesinin diğer elemanlarını:

Varyans-Kovaryans matrisi: <vcov(reg.model)>:

Artıklar serbestlik derecesi:

<df.residual(reg.model)>

Hata kareler toplamı:

<deviance(reg.model)>

ve diğer bileşenler:

<residuals(reg.model)>

<fitted.values(reg.model)>

<summary(reg.model)\$r.squared>

<summary(reg.model)\$adj.r.squared>

```
<summary(reg.model)$sigma>
<summary(reg.model)$fstatistic>
```

komutları ile görüntüleyebilirsiniz.

7.2 White değişen varyans düzeltilmiş standart hatalar

<car>, <sandwich> ve <Design> paketleri *robust* standart hataları hesaplamak için önceden tanımlanmış fonksiyonları içerir. İki farklı ağırlıklandırma seçeneği mevcuttur.

White düzeltmesi

```
> library(car)
> f1 <- formula(log(wage) ~ female +school)
> sqrt(diag(hccm(lm(f1),type="hc0")))

(Intercept)      female      school
0.022356311 0.010920889 0.001929391
```

<sandwich> kitaplığını kullanarak.

```
> library(sandwich)
> library(lmtest)
> coefTest(lm(f1), vcov=(vcovHC(lm(f1), "HC0")))
```

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t)
(Intercept)	4.0877297	0.0223563	182.845	< 2.2e-16 ***
female	-0.1911092	0.0109209	-17.499	< 2.2e-16 ***
school	0.0296671	0.0019294	15.376	< 2.2e-16 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

<car> kitaplığındaki <hc0> ve <sandwich> kitaplığındaki <HC0> orjinal White formülünü kullanmaktadır. <hc1>, <HC1> varyansları $\frac{N}{N-k}$ ile çarpılmaktadır. <Design> kitaplığını kullanarak:

```
> library(Design, warn.conflicts = FALSE)

Design library by Frank E Harrell Jr

Type library(help='Design'), ?DesignOverview, or ?Design.Overview')
to see overall documentation.
```

```
> f1 <- formula(log(wage) ~ female +school)
> fm1 <- robcov(ols(f1, x=TRUE, y=TRUE))
```

7.3 Grup-içi sabit olmayan hata varyansı

Bu <Design> kitaplığını kullanmaktadır ve verinin <industry> değişkeni tarafından da işaret edildiği gibi sektörler arası kümelenme gösterdiği durumlara açıklama getirmektedir.

```
> robcov(ols(f1, x=TRUE, y=TRUE), cluster=industry)
```

Linear Regression Model

```
ols(formula = f1, x = TRUE, y = TRUE)
```

n	Model L.R.	d.f.	R2	Sigma
2249	530.5	2	0.2101	0.2621

Residuals:

Min	1Q	Median	3Q	Max
-1.46436	-0.15308	-0.01852	0.13542	1.10402

Coefficients:

	Value	Std. Error	t	Pr(> t)
Intercept	4.08773	0.036285	112.66	0
female	-0.19111	0.016626	-11.49	0
school	0.02967	0.002806	10.57	0

Residual standard error: 0.2621 on 2246 degrees of freedom

Adjusted R-Squared: 0.2094

Grup sayısı M küçük olduğunda standart hataları şu şekilde de düzeltebilirsiniz:

```
> library(Design)
> f1 <- formula(log(wage) ~ female +school)
> M <- length(unique(industry))
> N <- length(industry)
> K <- lm(f1)$rank
> c1 <- (M/(M-1))*((N-1)/(N-K))
> fm1 <- robcov(ols(f1, x=TRUE, y=TRUE), cluster=industry)
> fm1$var <- fm1$var*c1
> coeftest(fm1)
```

t test of coefficients:

	Estimate	Std. Error	t value	Pr(> t)
Intercept	4.0877297	0.0222033	184.104	< 2.2e-16 ***
female	-0.1911092	0.0110662	-17.270	< 2.2e-16 ***
school	0.0296671	0.0017827	16.641	< 2.2e-16 ***

Signif. codes: 0 *** 0.001 ** 0.01 * 0.05 . 0.1 1

7.4 F-testi

Kısıtlı (bazı (veya tüm) eğim katsayıları üzerine sıfır kısıtının bulunduğu) ve kısıtsız (sıfır olan ve sıfır olmayan katsayılarla izin veren) modelleri tahmin edin. Kısıtlı modelde tanımlandığı şekliyle `anova()` komutunu kullanarak ortak hipotezleri test edebilirsiniz.

```
> mod.restricted <- lm(log(wage) ~ 1)
> mod.unrestricted <- lm(log(wage) ~ female + school)
> anova(mod.restricted,mod.unrestricted)

Analysis of Variance Table

Model 1: log(wage) ~ 1
Model 2: log(wage) ~ female + school
  Res.Df  RSS Df Sum of Sq    F    Pr(>F)
1    2248 195.34
2    2246 154.29  2    41.047 298.76 < 2.2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1  1
```

Sabit olmayan hata varyansı altında White varyans-kovaryans matrisini kullanırız. Modele ait F-değeri aşağıdaki şekildedir. Aşağıdaki kodlarda yer alan `<-1>` sabit terim ile ilişkili satır/sütunu dışlar.

```
> library(car)
> COV <- hccm(mod.unrestricted, "hc1")[-1,-1]
> beta <- matrix(coef(mod.unrestricted, ,1))[-1,]
> t(beta)%*%solve(COV)%*%beta/(lm(f1)$rank -1)

      [,1]
[1,] 253.1234
```

8 Zaman serileri

Satırlarda veri

Zaman serileri genellikle verinin satırlarda olduğu biçimde olurlar. Örnek olarak **National Institute of Economic Research** tarafından gerçekleştirilen İsveç Tüketici Anketi verisi kullanılmıştır. Veri setinde üç seri yer almaktadır: tüketici güven endeksi, bir makro endeks ve bir de mikro endeks.

İlk olarak orjinal veri metin formatında bir dosyaya kaydedilmiştir. Veri **R**'da girdi olarak kullanılmadan önce bir metin düzenleyicisi kullanılarak boşluk ile ayrılmış ilk üç serinin değerleri dosyanın içerisinde bırakılmıştır. Seriler satırlardadır. Serilere ait değerler değişken isimleri olmaksızın ayrı satırlarda listelenmiştir.

`<macro.txt>` dosyasını okumak için, aşağıda yer alan kod 1993 yılında başlayan ve 12 frekansa (aylık) sahip zaman serisi nesnesini tanımlamadan önce `<scan>` ve `<matrix>` yardımıyla veriyi 3 sütun ve 129 satıra yerleştirir. Seriler `<cci>`, `<macro.index>` ve `<micro.index>` olarak adlandırılmıştır. `<matrix>`'in varsayılan olarak veriyi sütunlara doldurduğuna dikkat edin.

```
> FILE <- "http://people.su.se/~ma/R_intro/macro.txt"
> macro <-ts(
+ matrix(
+ scan(FILE),129,3),
+ start=1993,frequency=12,
+ names=c("cci","macro.index", "micro.index"))
```

Burada bir değişkenin gecikmeli değerlerinin yaratılması ve veri setine eklenmesi örneği verilmektedir. Hesaplama farklılıkları için ayrıca `<diff>`'e bakınız.

`<macro>` veri setindeki veriye gecikmeli `<cci>`'yi (1 ay geciktirilmiş) ekleyerek yeni bir veri seti yaratalım. `<ts.union>` fonksiyonu tüm gözlemleri saklayarak serileri biraraya getirirken, `<ts.intersect>` yalnızca seilerin çakışan kısımlarını saklar.

```
> macro2 <- ts.union(
+ macro, 1.cci = lag(macro[,1],-1))
```

aggregate fonksiyonu zaman serisi verisinin frekansını değiştirmek için kullanılabilir. Aşağıdaki örnek verinin frekansını değiştirmektedir. `nfrequency=1` yıllık veri oluşturmaktadır. `FUN=mean` zaman içerisinde değişkenlerin ortalamasını hesaplamaktadır. Varsayılan ise `<sum>`'dir.

```
> aggregate(macro,nfrequency=1,FUN=mean)
```

```

Time Series:
Start = 1993
End = 2002
Frequency = 1
      cci macro.index micro.index
1993 -19.75833333 -33.38333333 -13.241667
1994 -0.29166667  14.16666667 -5.1583333
1995 -10.85833333 -3.72500000 -10.250000
1996 -7.91666667 -20.55000000 -3.8666667
1997  3.53333333  0.78333333  2.141667
1998 11.90833333 16.35000000  7.250000
1999 17.60833333 20.60833333 11.891667
2000 26.26666667 40.40833333 14.925000
2001  3.45833333 -13.70833333  9.6083333
2002  7.21666667 -7.32500000 10.125000

```

8.1 Durbin Watson

<lmtest> paketindeki <dwtest> ve <car> paketindeki <durbin.watson> kullanılabilir. Ayrıca Breusch-Godfrey yüksek sıradan ardışık bağımlılık testi için <lmtest> paketindeki <lmtest>'e bakınız.

```

> # Fitting the model.
> mod1 <- lm(cci ~ macro.index, data=macro)
> library(lmtest)
> dwtest(mod1)

      Durbin-Watson test

data:  mod1
DW = 0.063, p-value < 2.2e-16
alternative hypothesis: true autocorrelation is greater than 0

```

9 Grafikler

9.1 Grafikleri postscript formatında kaydetme

```

postscript("myfile.ps")
hist(1:10)
dev.off()

```

9.2 Grafikleri pdf formatında kaydetme

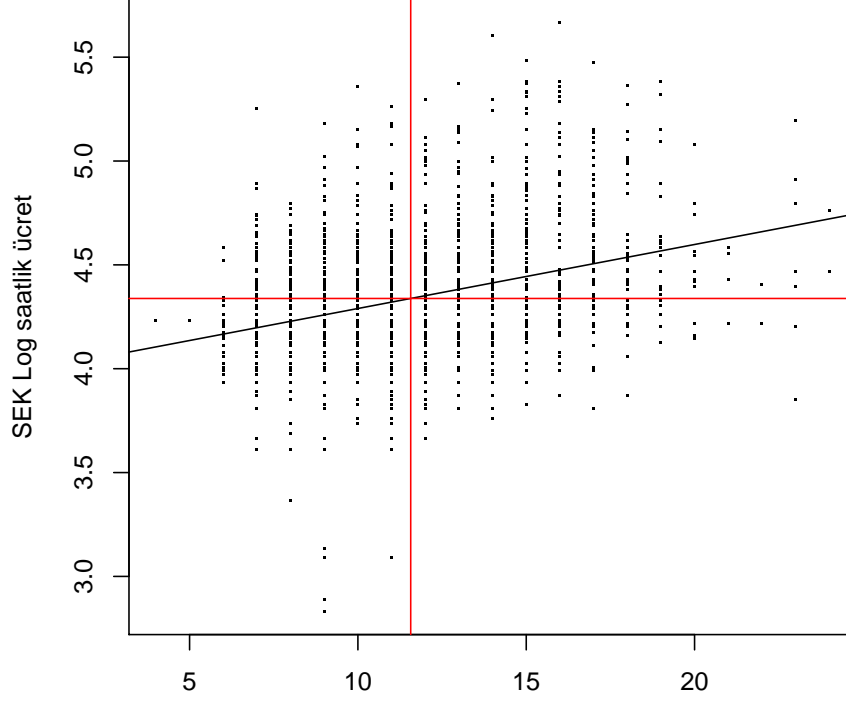
```
pdf("myfile.pdf")
hist(1:10)
dev.off()
```

9.3 Gözlemlerin ve regresyon doğrusunun grafiğini çizme

Verinin ve regresyon doğrusunun grafiğini çiz. <school>'u <log(wage)>'e karşı çiz.

```
> X.LABEL= "Okul süresi"
> Y.LABEL= "SEK Log saatlik ücret"
> TITLE <- "Grafik 1: Serpilim ve Regresyon"
> SubTitle <- "Kaynak: Yasam Düzeyi Anketi, LNU, 1991"
> plot(school,log(wage), pch=".",
+ main = TITLE, sub = SubTitle, xlab=X.LABEL, ylab=Y.LABEL)
> abline(lm(log(wage) ~ school ))
> abline(v = mean(school), col="red")
> abline(h = mean(log(wage)), col="red")
```

Grafik 1: Serpilim ve Regresyon



Kaynak: Yasam Düzeyi Anketi, LNU, 1991

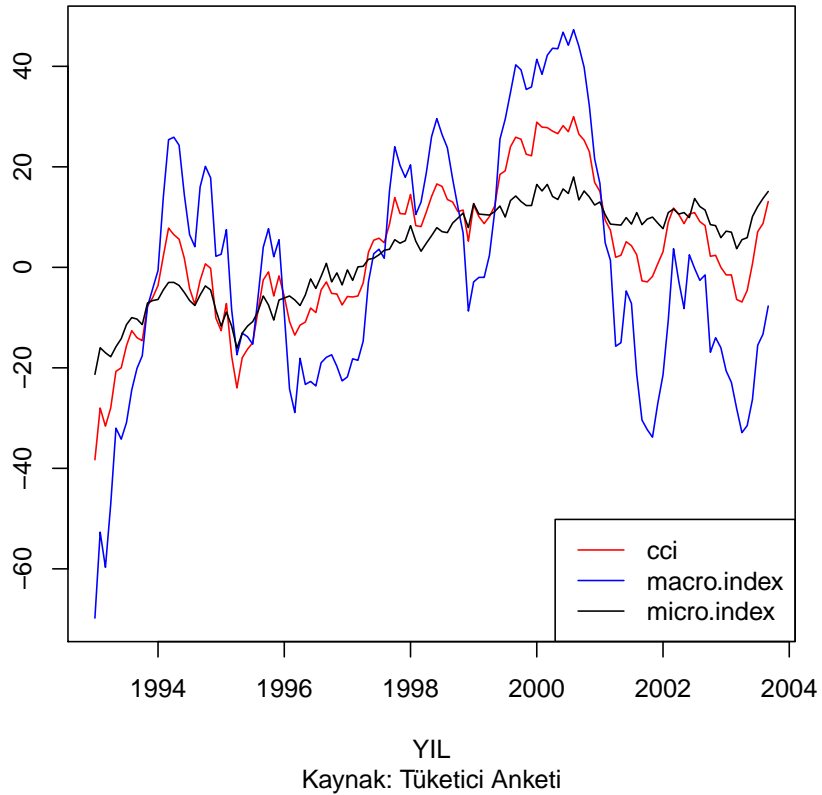
9.4 Zaman serileri grafiđi

Zaman serisi veri setini okutarak bařlayın. Ayrıntılar için 8. kısma bakınız.

Serileri bir grafikte çiz

```
> TITLE <- "Grafik 2: Tüketici güven endeksi"
> SubTitle <- "Kaynak: Tüketici Anketi"
> X.LABEL <- "YIL"
> COLORS = c("red","blue","black")
> ts.plot(macro, col=COLORS,
+ main = TITLE, sub = SubTitle, xlab=X.LABEL)
> legend("bottomright",legend=colnames(macro),
+ lty=1,col=COLORS)
```

Grafik 2: Tüketici güven endeksi



<plot.ts(macro)> serileri ayrı ayrı çizdirir.

10 Fonksiyon yazma

Yazım: `myfunction <- function(x, a, ...) \{...\}` şeklindedir. Fonksiyon için argümanlar fonksiyon gövdesinde, yani `{ }` içerisinde yer alan kodlar, tanımlanmış işlemlerde kullanılan değişkenlerdir. Bir kez bir fonksiyon yazıp onu kaydettiğinizde, bu fonksiyonu `{...}` içerisinde tanımlanmış işlemler için fonksiyonuza atıf yaparak ve gerçekleştirilen işlem için gerekli olan argümanları kullanarak gerçekleştirebilirsiniz.

Aşağıdaki fonksiyon bir değişkenin ortalamasının karesini hesaplamaktadır. `<ms>` fonksiyonunu tanımlayarak `<x>` değişkeninin ortalamasının karesini hesaplamak istediğinizde her seferinde `<(mean(x))^2>` yazmak yerine `<ms(x)>` yazabilirsiniz.

```
> ms <- function(x) {(mean(x))^2}
> a <- 1:100
> ms(a)

[1] 2550.25
```

Fonksiyonun argümanları:

Aşağıdaki fonksiyon argüman içermemektedir ve `<Hosgeldiniz>` metin dizisini görüntülemektedir.

```
> welc <- function() {print("Hosgeldiniz")}
> welc()

[1] "Hosgeldiniz"
```

Bu fonksiyon `x` argümanını almaktadır. Fonksiyonun argümanları yazılmalıdır.

```
> myprog.no.default <- function(x)
+ print(paste("I use", x , "for statistical computation."))
```

Eğer varsayılan değer belirlenmişse, herhangi bir argüman yazılmadığında varsayılan değer kabul edilir.

```
> myprog <- function(x="R")
+ {print(paste("I use", x , "for statistical computation."))}
> myprog()

[1] "I use R for statistical computation."

> myprog("R and sometimes something else")

[1] "I use R and sometimes something else for statistical computation."
```

10.1 Kümelenmiş Standart Hataları hesaplamak için bir fonksiyon

Burada kümelenmiş standart hataları hesaplamak için bir fonksiyon yer almaktadır (Yukarıda anlatılan *Design* kitabındaki *robcov* fonksiyonuna da bakınız). Argümanlar veri seti `<dat>`, model formülü `<f1>` ve kümelenme değişkenidir `<cluster>`.

```
clustered.standard.errors <- function(dat,f1, cluster){
  attach(dat, warn.conflicts = FALSE)
  M <- length(unique(cluster))
  N <- length(cluster)
  K <- lm(f1)$rank
  cl <- (M/(M-1))*((N-1)/(N-K))
  X <- model.matrix(f1)
  invXpX <- solve(t(X) %*% X)
  ei <- resid(lm(f1))
  uj <- as.matrix(aggregate(ei*X,list(cluster),FUN=sum)[-1])
  sqrt(cl*diag(invXpX%*%t(uj)%*%uj%*%invXpX)) }
```

Son satırı `sqrt(diag(invXpX %*%t(ei*X)%*(X*ei)%*%invXpX))` ile değiştirmenin White standart hataları vereceğine dikkat ediniz.

11 Çeşitli ipuçları

Gelir Dağılımı	Bakınız ineq .
Logit	<code><glm(formula, family=binomial(link=logit))></code> . Bakınız <code><?glm></code> & <code><?family></code> .
Negatif binom	<code><?negative.binomial or ?glm.nb></code> in MASS , VR .
Poisson regresyonu	<code><glm(formula, family=poisson(link=log))></code> . Bakınız <code><?glm></code> & <code><?family></code> .
Probit	<code><glm(formula,family=binomial(link=probit))></code> . Bakınız <code><?glm></code> & <code><?family></code> .
Eşanlı denklemler	Bakınız sem , systemfit .
Zaman Serileri	Bakınız <code><?ts></code> tseries , urca ve strucchange .
Tobit	Bakınız <code><?tobin></code> , survival içerisinde.

12 Teşekkür

Michael Lundholm , Lena Nekby ve Achim Zeileis'e değerli yorumları için minnettarım.

Kaynaklar

- Peter Dalgaard. *Introductory Statistics with R*. Springer, 2nd edition, 2008. URL <http://www.biostat.ku.dk/~pd/ISwR.html>. ISBN 978-0-387-79053-4.
- John Fox. *An R and S-Plus Companion to Applied Regression*. Sage Publications, Thousand Oaks, CA, USA, 2002. URL <http://socserv.socsci.mcmaster.ca/jfox/Books/Companion/index.html>. ISBN 0-761-92279-2.
- Christian Kleiber and Achim Zeileis. *Applied Econometrics with R*. Springer, New York, 2008. URL <http://www.springer.com/978-0-387-77316-2>. ISBN 978-0-387-77316-2.
- Roger Koenker and Achim Zeileis. Reproducible econometric research (a critical review of the state of the art). Report 60, Department of Statistics and Mathematics, Wirtschaftsuniversität Wien, Research Report Series, 2007. URL <http://www.econ.uiuc.edu/~roger/research/repro/>.
- R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. URL <http://www.R-project.org>. ISBN 3-900051-07-0.
- William N. Venables and Brian D. Ripley. *Modern Applied Statistics with S. Fourth Edition*. Springer, New York, 2002. URL <http://www.stats.ox.ac.uk/pub/MASS4/>. ISBN 0-387-95457-0.