

OLC731  
R YAZILIMI ILE VERİ ANALİZİ

Kubra Atalay Kabasakal

2023-09-18

# Contents

# Ders Hakkında

Bu dersin amacı, R yazılımı kullanarak veri üzerinde istenilen çok değişkenli istatistiksel ve psikometrik işlemlerin yapılabilmesini sağlamaktır.

Tez ve makale çalışmalarında öğrencilerimizin analizlerini R yazılımı ile hiçbir paket programa ihtiyaç duymadan kendi başlarına yapmalarını sağlamaktır.

(zence) **zence**.

## Eğitmen

👤 Dr. Kübra Atalay Kabasakal

✉ katalay@gmail.com

✉ katalay@hacettepe.edu.tr

## Kitaplar

- Atar, B., Atalay Kabasakal, K., Unsal Ozberk, E. B., Ozberk, E. H. & Kibrislioglu Uysal, N. (2020). R ile Veri Analizi ve Psikometri Uygulamalar, Pegem Akademi, Ankara.
- Desjardins, C. D., & Bulut, O. (2018). Handbook of educational measurement and psychometrics using R. Boca Raton, FL: CRC Press.

## Ders Materyelleri

- Ders notları ve okumalar moodle.
- Dönem boyunca R ve RStudio kullanacağız.

şeyi

# Chapter 1

## R ve Rstudio Temeller

Bu bölümde, **R** ve **RStudio**'nın nasıl kullanılacağını yanı sıra bazı temel programlama kavramları ve terminolojisi, yaygın tuzaklar, faydalı ipuçları ve nereden yardım alınabileceği konularını ele alacağız. Programlama deneyimi olmayanlar bu bölümü özellikle yararlı bulacaklar, ancak daha önce R kullanmış olsanız bile bazı yararlı ipuçları ve püf noktaları bulabilirsiniz.

Bu bölüm kendi kodunuzu yazmaya başlayana kadar bir anlam ifade etmeyebilir:) Biraz sabretmenizi bekliyorum !

### 1.1 R Nedir?

- R istatistiksel hesaplamalar yapabilen bir programlama dilidir.
- 1996 yılında Auckland Üniversitesi'nde **Ross Ihaka** ve **Robert Gentleman** tarafından geliştirilmiştir.
- 1960 yılında Bell Laboratories'de John Chambers ve arkadaşları tarafından geliştirilen **S dilinin** açık kaynak kodlu halidir.
- R yazılımı Genel Kamu Lisansı (GNU\* General Public Licence) koşulları altında ücretsiz dağıtılmaktadır.
- R ve Temel Geliştirme Takimi (R core team) ile ilgili bilgilere R'nin internet sitesinden (<https://www.r-project.org/>) ulaşılabilir.
- R dilinin ilk sürümü 29 Şubat 2000 tarihinde yayınlanmıştır. Her iki\* üç ayda bir sürümler güncellenmektedir.
- The latest release (2022\* 06\* 23, Funny\* Looking Kid) R\* 4.2.1.tar.gz, read what's new in the latest version
- RStudio, R ile çalışmayı kolaylaştıran bir Entegre Geliştirme Ortamıdır (Integrated Development Environment [IDE]).
- Bunu en azından bilmek ve kitap yazmak için Notepad gibi düz bir metin editörü kullanmak ile Microsoft Word gibi bir kelime işlemci kullanmak gibi düşünün. Bunu yapabilirsiniz, ancak bu kadar iyi görünmez ve yazım denetimi ve biçimlendirme gibi işler olmadan çok daha zor olur. Benzer bir şekilde, **R Studio olmadan da R kullanabilirsiniz ancak bunu tavsiye etmiyorum.**
- Unutmamanız gereken en önemli şey, bu ders için tüm çalışmalarınızı RStudio kullanarak yapacak olsanız da, aslında iki yazılım parçası kullanıyorsunuz, bu da zaman zaman her ikisinin de **ayrı güncellemeleri** olabileceği anlamına geliyor.
- R'yi ölçme için kullanmanın iki yolu vardır. İlk olarak, web tarayıcınız aracılığıyla R ve R'nin çevrimiçi bir sürümünü kullanabilirsiniz ve bunu **R server/sunucusu** olarak adlandıracağız. İkincisi, R ve RStudio'yu dizüstü veya masaüstü bilgisayarınıza ücretsiz olarak indirip kurabilirsiniz.

## 1.2 Avantajlar

- R özgür istatistiksel bir programlama dilidir.
- R aynı zamanda bir yorumlayıcı (interpreter).
- R, bir veri tabanı **deildir** ama veri tabanlarına balanabilir.
- Önceki sürümleri kullanıcı dostu olmasa da, son zamanlarda kod editörlerine çok sayıda ilav eklenmiştir.
- Ayrıca Java gibi diller araçları ile arayüz desteğine sahip bir yazılım geliştirme ortamıdır.
- Tablolardan oluşan yazılım paketlerine (Excel, Minitab gibi) benzemekle birlikte, yeni geliştiren bazı paketler farklı arayüzler sağlamaktadır.
- Ücretsiz olması nedeniyle, ticari destek tabiri bir yazılım değildir. Ancak destek alınabilecek çok sayıda kaynağa erişilebilir. (stackoverflow, mail listeleri)

## 1.3 Neden R?

- R istatistiksel programlama, veri analizi ve grafiksel gösterim için kullanılan ve ticari bir amaç gütmeyen ücretsiz bir yazımdır.
- R, UNIX, Windows ve MacOS gibi çeitli platformlarda kodlar derlemekte ve çalışmaktadır.
- SPSS, SAS gibi veri analizi programları ücretlidir, ayrıca bazı özel psikometri analizleri için yeterli değildir.
- R, açık kaynak kodlu olduğu için program kodlarına istenildiği zaman erişilebilir.
- Diğer istatistiksel yazımlarla karşılaştırıldığında R **komut satırı** arayüzünü kullanmaktadır.
- **Basit kodlar, döngüler ve kii** tanımlı fonksiyonlar yazmaya uyumlu basit ve etkili bir yazım diline sahiptir.
- R'in ayrıca **grafiksel imkânlar** oldukça fazladır; bu nedenle yayınlanabilir/basma uygun grafikler oluşturmak kolaydır.
- R ekibi birçok alanda ayrıntılı dokümantasyonu yapılmış R paketleri geliştirmektedir.
- Klasik istatistik yazımlarında analiz sonuçları bir kez elde edilir. R yazılımında ise sonuçlar çalışma alanına kaydedilerek, ileriki analiz aşamalarında tekrar kullanılabilir.
- R, psikometri alanında sıklıkla kullanılan simülasyon çalışmaları için (tekrarlı işlemler için) de avantaj sağlamaktadır.
- R, diğer programlama dilleri ve istatistik paket programları ile uyumludur.

## 1.4 Dezavantajlar

- Basta öğrenilmesi kolay görünse de, R'da uzmanlaşmak oldukça zordur.
- Menü ile kullanılan programlara alışkın olan kişiler için başlangıçta korkutucu olabilir.
- R ile bir analizi yapabilmek için planlama yapılması gerekmektedir.
- R kullanıcıları çoğunlukla programlamacı **deildir**. Programlamaya hâkim olmayan kişiler tarafından hazırlanan, okunması ve sürdürülebilirliği zor kodlar oluşturulabilir.
- Başlangıçta kodlar yazmak yavaş olabilir; ancak çalışmaların tekrarlanabilirliği açısından avantaj sağlamaktadır.
- Bu duruma bir örnek vermek gerekirse, 20 adet regresyon denklemi kurulup regresyon katsayıları karşılaştırılmak istenirse, R yazılımı sadece regresyon katsayılarını gösterebilir ve tek bir satırda tüm regresyon sonuçlarını karşılaştırmaya olanak verir. Aynı işlem için diğer istatistiksel yazımlarda 20 ayrı pencereden elde edilen sonuçların elle yazılarak karşılaştırılması gerekecektir.

- R’da hata yapma olasılıklar diğer programlara göre daha fazladır. Hata kaynağı için varsayımların iyi bilinmesi gerekmektedir.
- Hz konusunda SPSS ve SAS’a göre avantajlı olsa da diğer dillere göre (Python, Matlab gibi) daha yavaştır.
- Gelştirilen çok fazla paket olduğu için, ihtiyaca uygun en iyi paketin seçimi zor olabilmektedir.
- Bu bir dezavantaj gibi görünse de istatistiksel işlemlerin arka planı anlamaya yardımcı olur.
- Bu tarz zayıf hazırlanmış kodlar farklı kullarlarda yavaş çalışabilmektedir.
- Çok kullanıcı bu eksiklikleri gidermek için kodları değiştiremez. Özellikle çok iyi yapılandırılmamış olan kodlar R’da yavaş çalışabilmektedir.

## 1.5 R ve Rstudio Yüklenmesi

- İnternet tarayıcısına R yazılımının internet sitesinin ana sayfasının adresi yazılır. <https://www.r-project.org/>
- Sol menüde yer alan “download CRAN” bölümüne tıkladıktan sonra ülke seçilir. Seçilen ülkenin yakınına sadece yükleme hızı değiştirilecektir.
- Çıkan sayfada “Download and Install R” başlıklarından iletim sistemine uygun olan bağlantı seçilir.
- R konsolda çalışmaya doğrudan bağlanabilir; ancak konsol kullanımı bir kod editörü olmadıkça için çok kullanışlı değildir.
- Rstudio hata ayıklama, görselleştirme araçları ile birlikte yüklenen modern bir kod editörüdür.
- <https://www.rstudio.com/> internet sitesinden kullanılan bilgisayar ve iletim sistemine uygun olarak seçilip indirilebilmektedir.
- Rstudio R ile daha üretken olmanıza yardımcı olacak bir dizi araç içerir, örneğin:
  - R kodlarının vurgulamak için bir sözdizimi vurgulama düzenleyicisi
  - R kodların yazılmasına yardımcı olacak işlevler (otomatik tamamlama)
  - Çeşitli grafikler oluşturmak ve kaydetmek için çeşitli araçlar (ör. histogramlar, dairesel grafik)
  - Verileri içe veya dışarı aktarmak için bir çalışma alanı yönetim aracı

## 1.6 Diğer Gerekli Yüklemelemler

- Benim açıklamalarım yetmediyse R’yi bilgisayarınızda kullanmak için, lütfen daha ayrıntılı talimatlar ve indirmeniz gereken dosyaların bağlantılarının yanı sıra R’yi farklı bilgisayar türlerine yüklemek için bir dizi klavyeye bağlantılar içeren Installing R adresine bakın!!
- Yüklemelemler konusunda daha fazla ihtiyacınız var mı R studio R !
- Verilen linkte yer alsa da ayrıca eklemeye ihtiyaç duyduğunuz bağlantılar:
- Java
- Rtools Rtools, kaynak koddan derleme yapmaya yarayan araçlar içeren bir R yardımcıdır. **Önemli:** Eğer Windows kullanıyorsanız, ayrıca Rtools yüklemeniz gerekir.
- devtools

```
install.packages("devtools")
```

## 1.7 R STUDIO

- Rstudio'da panellerin yerlerini deitirebiliriz.
- Bunun yan sra yaz tipi, büyüklüü gibi özellikleri de deitirebiliriz.
- Varsaylan olarak, R Studio'yu açtınızda, kodunuz ve oturduğunuz tüm nesneler dahil olmak üzere en son ne üzerinde çalıştığınız gösterir. Bu yararlı gibi görünebilir, ancak aslında deerinden daha fazla soruna neden olma eilimindedir, çünkü yanlışlıkla bir nesnenin eski bir sürümünü kullanma riskiniz oldu anlamına gelir. R Studio'yu her balattınızda yeni bir kopya açacak ekilde ayarlar deitirmenizi öneririz. Bunu **Araçlar - Global Seçenekler** seçeneine tıklayarak ve ardından aadaki gibi görünmesi için kutuların seçimini kaldırarak yapabilirsiniz.

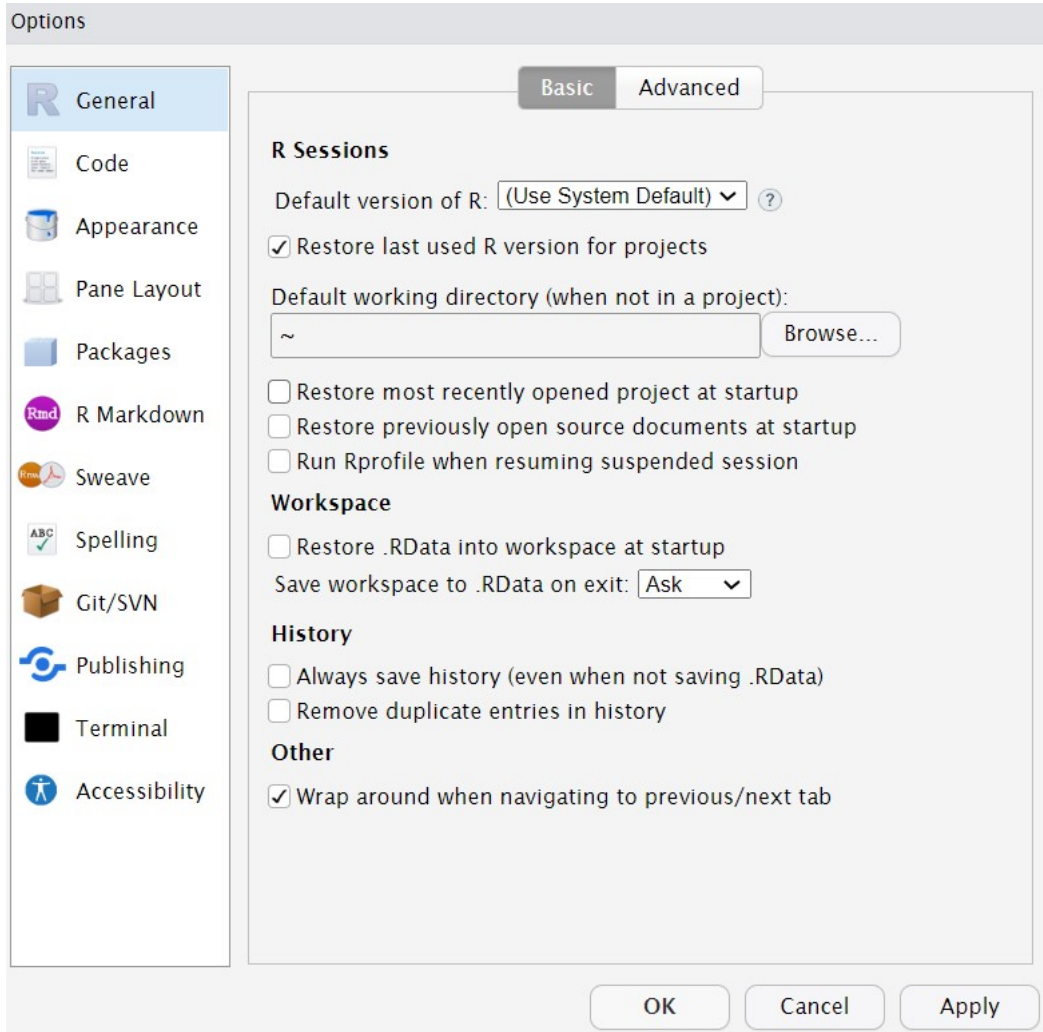


Figure 1.1: Global options

- Dönem boyunca Rstudio kullanıma aına olacaksınız. Bu süreci kolaylatırmak için balantlar verilen dökümanlara göz atabilirsiniz.
- Rstudio cheatsheet
- Oscar Torres\* Reyna tutorial

## 1.8 Hangi R sürümünü kullanmalısınız?

- R'yi bilgisayarınıza kurmanın avantajı, kullanmak için internete bağımlı olmanızın gerekmemesi, dosyalarınızın kaydetmenin ve yönetmenin daha kolay olması ve sunucunun çökmesi durumunda sorun yaşanmamasıdır (bu nadirdir, ancak olmaktadır).
- R sunucusunu kullanmanın avantajı, bilgisayarınıza herhangi bir şey yüklemenize gerek olmaması, sadece web tarayıcınız üzerinden erişebilirsiniz.
- R'yi yükleyemeyeceğiniz bir bilgisayarınız varsa (örneğin Chromebook) veya R'yi bilgisayarınıza yüklemeye ilgili ciddi sorunlarınız varsa sunucuyu kullanmanız önerilir.

## 1.9 R Studio Hakkında

- R Studio, kodu deneyebileceğiniz bir konsola sahiptir (ekil'de sol alt pencerede yer alır??).
- Ayrıca kod editörü (sol üst), "Ortam" sekmesinde oluşturduğunuz fonksiyonlar ve nesneleri gösteren bir pencere (sa üst pencere) ve grafikleri, dosya paketlerini ve yardım belgelerini gösteren bir pencere ise (sa alt) bulunur.

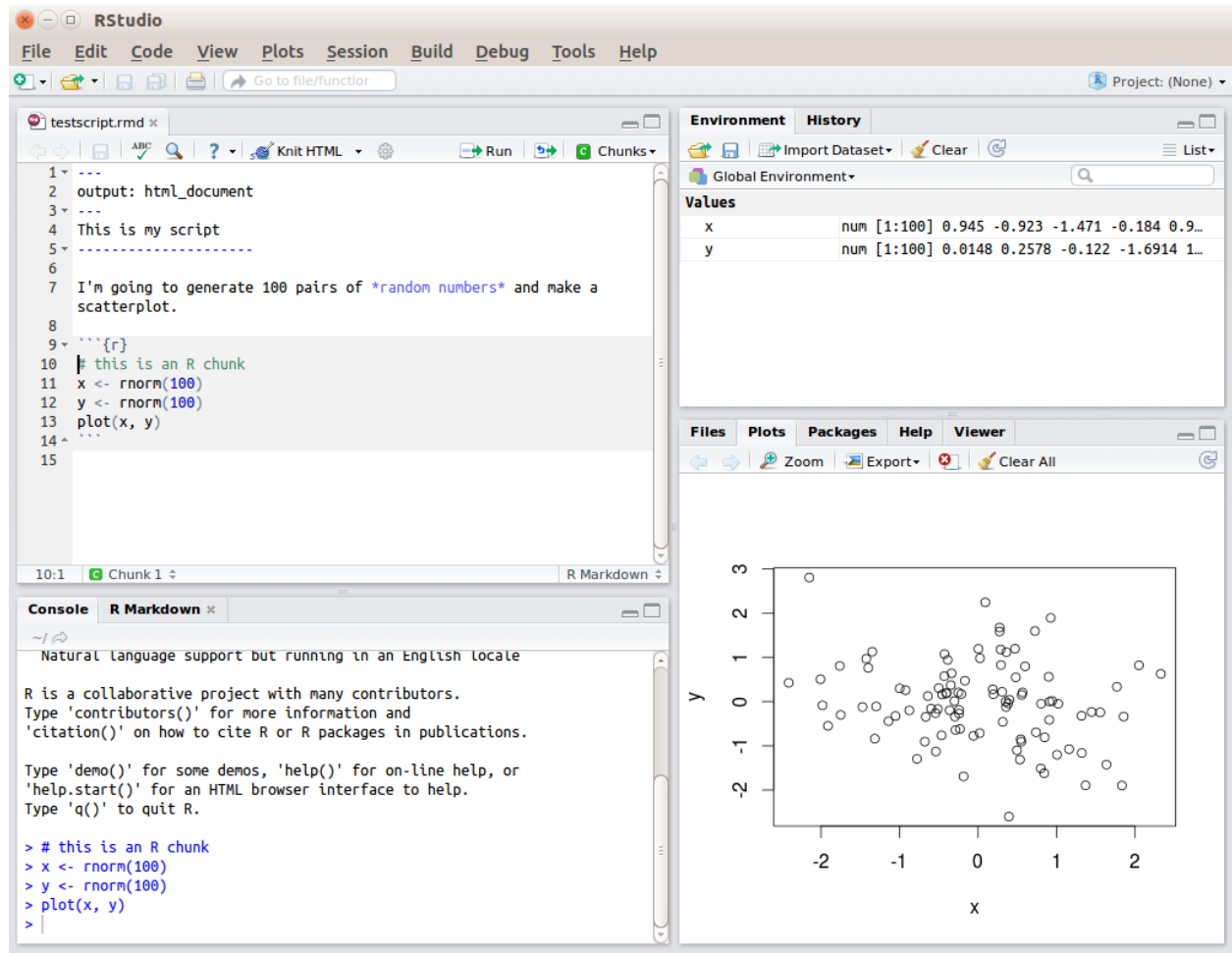


Figure 1.2: RStudio arayüzü

- Bu ders boyunca R Studio'da bulunan özelliklerin nasıl kullanılacağı hakkında daha fazla bilgi edineceksiniz, ancak R Studio ekibinden RStudio Essentials 1 izlemenizi ısrarla tavsiye ederim. Video



yaklaşık 30 dakika sürmekte ve R Studio'nun ana bölümlerini tanıtmaktadır.

## 1.10 R Temel Özellikler

- R konsolda görünen `>` isareti, R yazılımının sizden komut beklediğini belirtir. R'in hesap makinesi olarak kullanım örnekleri sunulmuştur.

```
2+2
```

```
[1] 4
```

- R bölümlere duyarlı değildir.

```
2 +      2
```

```
[1] 4
```

```
2+
  2
```

```
[1] 4
```

## 1.11 Atama Operatörü

- Atama operatörü olarak “küçüktür” simgesi ile “kasa çizgi” simgesi `<-` simgeleri kullanılabilir.
- `<-` yerine “eittir” = simgesi de atama operatörü olarak kullanılabilir.
- Ancak = operatörü programlama yaparken matematiksel eşlikle karabilmektedir.
- Atama yapılacak nesne isimlendirilirken harflerle (`A*Z` veya `a*z`) balamalıdır.
- simlendirmeye harfle balandıktan sonra rakamlar (`0*9`), nokta (`.`) ve alt çizgi (`_`) ile devam edilebilmektedir.
- R harflerin küçük ve ya büyük olmasına kar duyarlıdır.
- R fonksiyonlara benzer isimlerde nesne ismi kullanmamaya **dikkat edilmelidir**.
- Ayrıca **c,C,D,F,I,q,t,T** gibi tek harfli nesne ismi kullanmaktan kaçınılmalıdır; bunların R'da özel anlamlar bulunmaktadır.
- R yazılımında `#` isareti ile başlayan satır, yorum satırdır.
- Genellikle komutların anlamını açıklamak için kullanılmaktadır.
- R, bu satırları dikkate almaz, bunlar sadece kullanıcılar için bilgi ve hatırlatıcı açıklamalar içermektedir.

```
# Yorum satırları kodlarınızı anlamlı hale getirir.
a <- 2
y <- a * a
y
```

```
[1] 4
```

## 1.12 Basit İşlemler

- toplama işlemi için `+`,
- çıkarma işlemi için `-`,
- çarpma işlemi için `*`,

- bölme ilemi için /,
- üs alma ilemi için ^ veya \*
- mod alma için ise %% operatorleri kullanılmaktadır.
- Kodlamanın büyük bir kısmı nesne oluşturma ve nesneleri manipüle etmeyi içerecektir. Nesneler bir şeyler içerir. Bu şeyler sayılar, kelimeler veya işlemlerin ve analizlerin sonucu olabilir

### Altırma Nesneler oluşturma

- Aşağıdaki kodu kopyalayıp konsola yapıştırın, kodu kendi adınız ve yanz kullanacak şekilde değiştirin ve çalıştırın. Environment bölümünde `ad`, `yas`, `gun`, `yeniyil` ve `veri` nesnelerinin görüldüğünü göreceksiniz.

```
ad <- "ada"
yas <- 16 + 20
gun <- Sys.Date()
yeniyil <- as.Date("2024-01-01")
veri <- rnorm(n = 10, mean = 15, sd = 3)
```

The screenshot shows the RStudio Environment pane with the following objects:

Object	Value
ad	"ada"
gun	2023-09-06 UTC
veri	num [1:10] 11.5 12.3 17.5 13.7 14.5 ...
ya	35
yeniyil	2023-01-01 UTC

Figure 1.3: Çalışma alanındaki nesneler

- Bu örneklerde, `ad`, `yas` ve `yeniyil` her zaman `ada`, `36` yaşlarını ve `2024` Yeni Yıl Günü tarihini içerecektir, ancak `gun` tarihi işletim sisteminden alınacaktır ve `veri` rastgele oluşturulmuş bir veri kümesi olacaktır, bu nedenle bu nesnelerin değerleri statik olmayacaktır.
- Daha da önemlisi, nesneler hesaplamalara dahil olabilir ve birbirleriyle etkileime girebilir. Örneğin:

```
yas + 10
yeniyil - gun
mean(veri)
```

```
[1] 46
Time difference of 105 days
[1] 14.8447
```

- Son olarak, bu işlemlerin sonucunu yeni bir nesnele saklayabilirsiniz:

```
n1 <- yas + 10
```

<-ifadesini iğerireklinde okumak faydal olabilir, örneğin `adifadesiada` metnini içerir.

- Bu ders boyunca sürekli olarak nesneler yaratacağınız ve ilerledikçe onlar ve nasıl davrandıkları hakkında daha fazla bilgi edineceksiniz, ancak imdilik bunların değerleri kaydetmenin bir yolu olduğunu, bu değer-

lerin say, metin veya ilemlerin sonucu olabileceini ve yeni deikenler oluturmak için baka ilemlerde kullanlabileceini anlamak yeterlidir.

Nesnelerin ‘deikenler’ olarak adlandırıldn da görebilirsiniz. Programlama terimlerinde ikisi arasında fark vardır, ancak çok sk eanamlı olarak kullanırlar.

### Altrma Nesneler oluturma

- Aadaki kodu kopyalayp konsola yaptrn.
- Eni 4 cm, boyu 10 cm bir dikdörtgenin alan hesaplaynz.

```
# en nesnesi tanımlama  
  
# boy nesnesi tanımlama  
  
# alan nesnesi tanımlama  
  
# alan nesnesini yazdırma
```

```
[1] 40
```

- Eni 4 cm, boyu 10 cm bir dikdörtgenin köegen uzunluunu hesaplaynz.

```
# en nesnesi tanımlama  
  
# boy nesnesi tanımlama  
  
# köegen nesnesi tanımlama  
  
# köegen nesnesini yazdırma
```

```
[1] 10.77033
```

#### 1.12.1 Ödev

Datacamp hesapnzda yer alan 01\_Temeller workspaceni tamamlaynz.

## Chapter 2

# R Paketler

- R'yi yüklediğinizde, veri ileme ve istatistiksel analiz seçenekleri de dahil olmak üzere bir dizi fonksiyona erişebilirsiniz. Varsayılan kurulumda yer alan fonksiyonlar genellikle **Temel R/Base R** olarak adlandırılır ve birçok Temel R fonksiyonunu gösteren faydalı bir cheatsheet sayfası vardır: [cheatsheet](#)
- **Temel R** telefonunuzda gelen varsayılan uygulamalar, paketleri ise ayrıca indirmeniz gereken ek uygulamalar olarak düşünmek faydalı olabilir.
- R fonksiyonları **ayrı paketler** halinde düzenlenmiştir. Böylece gerekli paketlerle çalışarak daha az bellek kullanımı ve hızlı işlem gücü sağlanır.
- Bu paketlerin bir başka avantajı da yazılan fonksiyonlardan oluşan paketlerin CRAN'den temin edilerek yüklenebilmesidir.
- Her paketin bir yaratıcısı ve kendisine ait bir yardım dosyası bulunur.

```
# paket yükleme
install.packages("CTT")
# paket aktive etme
library(CTT)
```

- Paket yükleme işlemi Rstudio'da yer alan menüler aracılığıyla da yapılabilir.
- R paketleri R fonksiyonlarının, verilerinin ve iyi derlenmiş bir formatta kodların kombinasyonlarından oluşmaktadır. `library()` komutu ile kişisel kütüphanenizdeki yüklü paketleri görebilirsiniz.
- Sadece temel pakette 1000'den fazla fonksiyon bulunmaktadır.

```
# temel paket fonksiyonlarına ulaşmak için
fonksiyonlar <- builtins()
length(fonksiyonlar)
```

```
## [1] 1380
```

```
fonksiyonlar[910:920]
```

```
## [1] "Cstack_info"          "crossprod"
## [3] "cospi"                "cosh"
## [5] "cos"                  "contributors"
## [7] "Conj"                 "conflicts"
## [9] "conflictRules"        "conditionMessage.condition"
## [11] "conditionMessage"
```

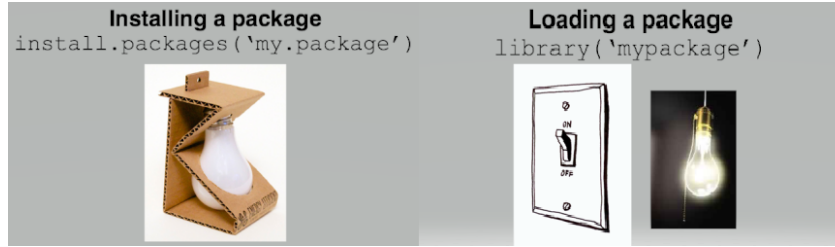


Figure 2.1: yükletme-aktifletme

### 2.0.1 Altrma : tidyverse yükletme

- Bir paketi kullanabilmek için önce onu yüklemeniz gerekir. Aadaki kod, bu derste çok sk kullanacamız bir paket olan **tidyverse** paketini yükler.

```
install.packages("tidyverse")
```

- Bir paketi yalnızca bir kez yüklemeniz gerekir, ancak R'yi her balattınızda kullanmak istediğiniz paketleri yüklemeniz gerekir, benzer ekilde telefonunuza bir uygulamayı bir kez yüklemeniz gerekir, ancak her kullanmak istediğinizde açmanız gerekir.

**UYARI: WARNING: Rtools is required to build R packages** gibi bir hata mesaj alırsanız, [Rtools] (<https://cran.r-project.org/bin/windows/Rtools/>) adlı ekstra bir yazılım indirmeniz ve yüklemeniz gerekebilir.

### 2.0.2 Altrma : tidyverse etkinletir

- Tidyverse'i etkinletirmek için aadaki kodu çalıştırın.

```
library(tidyverse)
```

- Bir hata mesajı gibi görünen bir şey alacaksınız - öyle değil. Bu sadece R'nin size ne yaptığını anlatmasıdır.
- İmdi **tidyverse** paketini etkinletirdiğimize göre, içerdii fonksiyonlardan herhangi birini kullanabiliriz, ancak unutmayın, R'yi her balattığınızda **library()** fonksiyonunu çalıştırmanız gerekir.

## 2.1 Yardım Sayfaları

- R'da temel ve diğer paketlerde yer alan fonksiyonların ilevleri görmek için yardım sayfalarını inceleyebilirsiniz. **? ve help()** fonksiyonları aynı işlevi sahiptir.

```
?is.na
```

```
help(sqrt)
```

- Örneğin CTT paketini hem yüklediniz hem de etkinletirdiniz. Paket fonksiyon ve veri içeriğini aadaki komutlarla görebilirsiniz.

```
# install.packages(CTT)
library(CTT)
ls("package:CTT")
data(package = "CTT") # yeni bir sekmede açılır.
?reliability
```

- Etkinletirdiğiniz paketlerde yer alan fonksiyonların yardım sayfalarına ulaşabilirsiniz.

## 2.2 Paket çakmalar

- Daha da fazla fonksiyona sahip binlerce farklı R paketi vardır. Ne yazık ki, bazen farklı paketler aynı fonksiyon isimlerine sahiptir. Örneğin, `dplyr` ve `MASS` paketlerinin her ikisi de `select()` adında bir fonksiyona sahiptir. Bu paketlerin her ikisini de yüklerseniz, R size bir çakma olduğunu söyleyen bir uyarı üretecektir.

```
library(dplyr)
library(MASS)
```

```
##
## Attaching package: 'MASS'
## The following object is masked from 'package:dplyr':
##
##      select
```

- Bu durumda, R size `dplyr` paketindeki `select()` fonksiyonunun aynı isimli başka bir fonksiyon tarafından gizlendiğini (veya ‘maskelendiğini’) söylüyor. Eğer `select()` fonksiyonunu kullanmay deneseydiniz, R en son yüklenen paketteki fonksiyonu kullanacaktı - bu durumda `MASS` fonksiyonunu kullanacaktı.
- Belirli bir fonksiyon için hangi paketi kullanmak istediğinizi belirtmek istiyorsanız, örneğin `package::function` biçiminde kod kullanabilirsiniz:

```
dplyr::select()
MASS::select()
```

## 2.3 Paket Güncelleme

- R ve R Studio güncellemelerine ek olarak, paketlerin yazarları da bazen kodların günceller. Bu, bir pakete fonksiyon eklemek için olabileceği gibi hataları düzeltmek için de olabilir. **Kaçınılması gereken bir şey, yüklü bir paketi istemeden güncellemektir.**
- `install.packages()` fonksiyonunu çalıştırdığınızda, her zaman paketin en son sürümü yüklenir ve yüklemiş olduğunuz eski sürümlerin üzerine yazılır. Bazen bu bir sorun değil, ancak bazen paket önemli ölçüde değişti için güncellenmenin kodunuzun artık çalışmadığını anlamına geldiğini görürsünüz. Bir paketin eski bir sürümüne geri dönmek mümkündür ancak yine de bundan kaçınmaya çalışın.

Bir paketin üzerine yanlışlıkla daha sonraki bir sürümün yazılması önlemek için, sizin veya bir başkasının kodu yanlışlıkla kaldırması ihtimaline karşı analiz komut dosyalarınıza `install.packages()` i asla dahil etmemelisiniz.

## 2.4 R ve RStudio’ya nasıl alıntı yapılır

- R’a atıfta bulunmanız ve referans vermeniz gereken bilimsel bir rapor yazmaktan biraz uzak olabilirsiniz, ancak zaman geldiğinde bunu onu getiren insanlara (çoğu ücretsiz!) kredi vermek için yapmak önemlidir. R, RStudio ve kullandığınız paketler için ayrı alıntılar salamalsınız.
- Kullandığınız R sürümü için atıf almak için, size her zaman en son atıf salayacak olan `citation()` fonksiyonunu çalıştırmanız yeterlidir.

```
citation()
```

```
##
## To cite R in publications use:
##
```

```
## R Core Team (2022). R: A language and environment for statistical
## computing. R Foundation for Statistical Computing, Vienna, Austria.
## URL https://www.R-project.org/.
##
## A BibTeX entry for LaTeX users is
##
## @Manual{,
##   title = {R: A Language and Environment for Statistical Computing},
##   author = {{R Core Team}},
##   organization = {R Foundation for Statistical Computing},
##   address = {Vienna, Austria},
##   year = {2022},
##   url = {https://www.R-project.org/},
## }
##
## We have invested a lot of time and effort in creating R, please cite it
## when using it for data analysis. See also 'citation("pkgname")' for
## citing R packages.
```

- Kullandnz herhangi bir paket için atf oluşturmak için, atf yapmak istediiniz paketin adyla birlikte `citation()` ilevini de kullanabilirsiniz.

```
citation("tidyverse")
```

```
##
## To cite package 'tidyverse' in publications use:
##
## Wickham H, Averick M, Bryan J, Chang W, McGowan LD, François R,
## Grolemond G, Hayes A, Henry L, Hester J, Kuhn M, Pedersen TL, Miller
## E, Bache SM, Müller K, Ooms J, Robinson D, Seidel DP, Spinu V,
## Takahashi K, Vaughan D, Wilke C, Woo K, Yutani H (2019). "Welcome to
## the tidyverse." _Journal of Open Source Software_, *4*(43), 1686.
## doi:10.21105/joss.01686 <https://doi.org/10.21105/joss.01686>.
##
## A BibTeX entry for LaTeX users is
##
## @Article{,
##   title = {Welcome to the {tidyverse}},
##   author = {Hadley Wickham and Mara Averick and Jennifer Bryan and Winston Chang and Lucy D'Agostini
##   year = {2019},
##   journal = {Journal of Open Source Software},
##   volume = {4},
##   number = {43},
##   pages = {1686},
##   doi = {10.21105/joss.01686},
## }
```

- Kullandnz RStudio sürümüne ait alnty oluşturmak için `RStudio.Vesion()` fonksiyonunu kullanabilirsiniz:

```
RStudio.Version()
```

- Son olarak, yöntem bölümünüzün yazmnda bunun nasıl görünebileceine dair bir örnek:  
Analiz R (R Core Team, 2020), RStudio (Rstudio Team, 2020) ve tidyverse paketi (Wickham, 2017) kullanılarak gerçekleştirilmiştir.

- Belirtildiği gibi, bunu bir süre yapmak zorunda kalmayabilirsiniz, ancak yaptınızda buna geri dönmek çünkü açık kaynak topluluuna çalmalar için kredi vermek önemlidir.



## Chapter 3

# Fonksiyonlar

- Fonksiyon belli bir görevi yerine getirmek için yazılan bir grup komuttur.
- Fonksiyonların çalışması için girdilerinin olması gerekmektedir. Fonksiyonlar girdileri ile yaptıkları işlem sonucunda bir çıktı olutururlar.
- Fonksiyonlar girdileri o fonksiyonun çalışması için önceden belirlenen **argümanlar** ve o argümanların değerlerinden oluşur. (dilbilimle ilgileniyorsanız, bunlar bir özne ve nesne gerektiren fiiller olarak düşünmek isteyebilirsiniz)
- Fonksiyonların kullanımında üç noktaya dikkat edilmelidir.
  1. argümanların sırası
  2. argümanların olaan (default) değerleri
  3. bazı argümanların zorunlu, bazı argümanların opsiyonel olmasıdır
- Bir fonksiyonun aldığı tüm argümanlara yardım dokümantasyonunu kullanarak `?function` formatını kullanarak bakabilirsiniz. Bazı argümanlar zorunlu, bazıları ise isteğe bağlıdır. İsteğe bağlı olanlar, herhangi bir değer girmezseniz genellikle varsayılan/olaan (normalde yardım belgelerinde belirtilen) bir değer kullanır.
- Örnek olarak, normal dağılıma sahip bir sayı kümesini rastgele üreten `rnorm()` fonksiyonunun yardım belgelerine bakalım.
- Bir fonksiyonun aldığı tüm argümanlara yardım dokümantasyonunu kullanarak `?function` formatını kullanarak bakabilirsiniz. Bazı argümanlar zorunlu, bazıları ise isteğe bağlıdır. İsteğe bağlı olanlar, herhangi bir değer girmezseniz genellikle varsayılan/olaan (normalde yardım belgelerinde belirtilen) bir değer kullanır.

### Alıştırma

- R Studio'yu açın ve konsola aşağıdaki kodu yazın:

```
?rnorm
```

- `rnorm()` için yardım belgeleri sağ alt yardım panelinde görünmelidir. Kullanım bölümünde, `rnorm()` un aşağıdaki formu alındığını görüyoruz:

```
rnorm(n, mean = 0, sd = 1)
```

- Argümanlar bölümünde, her bir argüman için açıklamalar bulunmaktadır. `n` oluşturmak istediğimiz gözlem sayısı, `mean` oluşturacağımız veri noktalarının ortalaması ve `sd` verinin standart sapmasıdır. Ayrıntılar bölümünde, `mean` ve `sd` için herhangi bir değer girilmezse, bu değerler için varsayılan olarak 0 ve 1 kullanılabileceği belirtilir. `n` için varsayılan bir değer olmadıysa, belirtilmesi gerekir, aksi takdirde kod çalışmaz.

- Bir örnek deneyelim ve R'den 5 rastgele say üretmesini istemek için gerekli `n` argümanın deitirelim.

### Altrma II

- Aadaki kodu kopyalayp konsola yaptrn.

```
set.seed(12042016)
rnorm(n = 5)
```

```
## [1] -0.2896163 -0.6428964  0.5829221 -0.3286728 -0.5110101
```

- Bu sayların ortalamas 0 ve SD'si 1'dir. imdi farkl bir say kümesi üretmek için ek argümanlar deitirebiliriz.

```
rnorm(n = 5, mean = 10, sd = 2)
```

```
## [1] 13.320853  9.377956 10.235461  9.811793 13.019102
```

- Bu kez R yine 5 rastgele say üretti, ancak imdi bu say kümesi belirtildi gibi 10 ortalama ve 2 sd deerine sahip. Bir fonksiyonun hangi argümanlar gerektirdiini anlamanza yardmc olmas için yardım belgelerini kullanmay her zaman unutmayn.

Eer internette kod örneklerine bakyorsanz, sk sk `set.seed()` fonksiyonu ile balayan kodlar görebilirsiniz. Bu fonksiyon rastgele say üreticini kontrol eder - rastgele say üreten herhangi bir fonksiyon kullanıyorsanız (`rnorm()` gibi), `set.seed()` fonksiyonunu çaltrmak ayn sonucu almanız salayacaktır (baz durumlarda yapmak istediiniz ey bu olmayabilir). Bu örnekte `set.seed()` diyoruz, bu ayn rastgele sayılar alacağınız anlamna geliyor.

## 3.1 Argüman isimleri

- Yukardaki örneklerde, kodumuzdaki bamsz deiken adların yazdk (örnein, `n`, `mean`, `sd`), ancak bu kesinlikle gerekli deildir. Aadaki iki kod satrının her ikisi de ayn sonucu üretecektir (`rnorm()` fonksiyonunu her çaltrdınızda rastgele oldu için biraz farkl bir say kümesi üretecektir, ancak yine de ayn ortalama ve SD'ye sahip olacaklardır):

```
rnorm(n = 6, mean = 3, sd = 1)
rnorm(6, 3, 1)
```

- Önemli olarak, eer argüman isimlerini yazmazsanz, R argümanların varsayılan sırasn kullanacaktır, yani `rnorm` için girdiğiniz ilk sayının `n` olduunu varsayacaktır. ikinci say `mean` ve üçüncü say `sd`dir.
- Eer argüman isimlerini yazarsanz, argümanlar istediiniz sırada yazabilirsiniz:

```
rnorm(sd = 1, n = 6, mean = 3)
```

- R'yi ilk öğrenirken, fonksiyonun her bir parçasının ne yaptn hatırlamanza ve anlamanza yardmc olabileceinden, argüman adların yazmay yararlı bulabilirsiniz. Ancak, becerileriniz ilerledikçe argüman adların atlamay daha hızlı bulabilirsiniz ve ayrıca argüman adların kullanmayan çevrimiçi kod örnekleri göreceksiniz, bu nedenle her bir kod parçasının hangi argümana atfta bulunduunu anlayabilmek önemlidir (veya kontrol etmek için yardım belgelerine bakın).
- Bu derste, her bir fonksiyonu ilk kez kullandımızda argüman adların her zaman yazacağız, ancak sonraki kullanımlarda bunlar atlanabilir.

## 3.2 TAB ile otomatik tamamlama

- R Studio'nun çok kullanlı bir özelli, fonksiyonlar için TAB otomatik tamamlama özelliidir (bkz. ekil ??). Fonksiyonun adn yazp tab tuuna basarsanz, R Studio size fonksiyonun ald argümanlar ksa bir

açıklama ile birlikte gösterecektir. Argüman adının üzerinde enter tuuna basarsanız, tıpkı telefonunuzdaki otomatik tamamlama gibi ad sizin için dolduracaktır. Bu, R'yi ilk öğrenirken inanılmaz derecede kullanılır ve bu özelliği sık sık kullanmayı **unutmamalı**sınız.

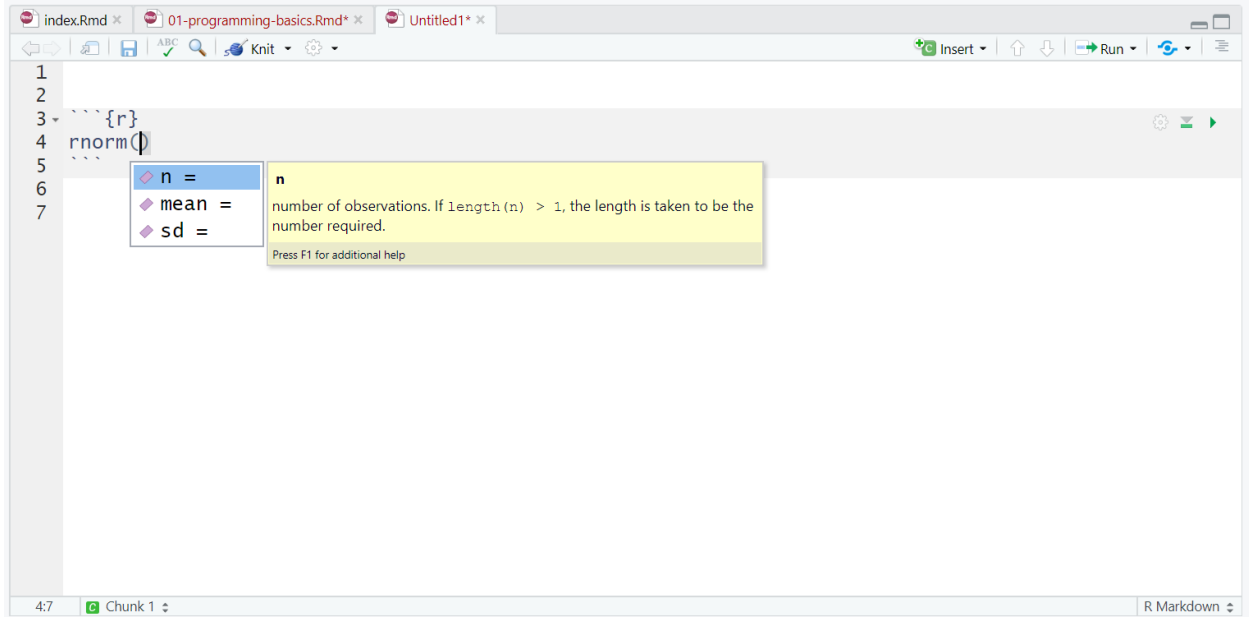


Figure 3.1: Tab ile otomatik durdurma

### 3.3 Kiisel tanıml fonksiyon

- Kiisel tanıml fonksiyon yazılmas aablonu aadaki gibidir.

```
fonksiyonadi<- function(argumanlar ve olagan degerleri){
  kodlar
  return()
}
```

- Oluturulan fonksiyon çalıtrırken ise aadaki eeklinde çalıtrılır.

```
fonksiyonadi(argumanlar ve degerleri)
```

- Kare alma ilemi aadaki eekilde yapılabilir.

```
sayi <- 4
sayi * sayi
sayi ^2
```

```
## [1] 16
## [1] 16
```

- Bu işlem sürekli yapılacaksa fonksiyon olarak yazılabilir.

```
# kare alma fonksiyonu
kare_al <- function(sayi){
  return(sayi*sayi)
}
kare_al(4)
```

```
## [1] 16
```

- Farklı dereceden üsler alabilen bir fonksiyon yazalım.

```
#üs alma
üs_al<- function(x,us){
  return(x^us)
}
üs_al(3,4)
```

```
## [1] 81
```

- Argümanlardan birine olaan değer girilmesi

```
#üs alma
üs_al<- function(x,us=2){
  return(x^us)
}
üs_al(3) # us argumanın olagan değeri olan
# 2 oldu için argumana
# değeri girilmediğinde kare alır.
```

```
## [1] 9
```

- Aadaki fonksiyona 3 ve 4 değerleri girilirse çıktı ne olur?

```
myfunc <- function(x,y)
{
  a <- x+y
  b <- x*y
  return(a*b)
}
myfunc(3,4)
```

- `mean()` fonksiyonu en sık kullandığımız fonksiyonlardan biridir.

```
x <- c(1,2,3)
mean(x)
```

```
## [1] 2
```

- R base pakette yer alan bu fonksiyonu kendiniz de yazabilirsiniz.
- R’da deneyim kazandıkça, yaptığımız işlemler karmaşıklaştıkça kendi fonksiyonlarımız yazma ihtiyacı duyacağız.
- `avg()` isminde vektör ortalaması hesaplayan fonksiyon yazalım.
- Yazdığımız fonksiyon ile aadaki işlemi yapalım.

```
x <- 1:1000
avg(x)
```

```
## [1] 500.5
```

- Yazdığımız fonksiyon temel pakette yer alan `mean()` fonksiyonu ile aynı sonucu verdi mi?

```
identical(avg(x),mean(x))
```

```
## [1] TRUE
```

- Fonksiyon içinde tanımlanan nesneler çalışma alanına kaydedilmezler.
- Fonksiyonlar da R nesnesidir.

```
ls()
```

```
## [1] "avg"           "backtick"      "h1"
## [4] "kare_al"       "path"          "pkg"
## [7] "psyteachr_colors" "psyteachr_colours" "sayi"
## [10] "üs_al"        "x"
```

### 3.4 R Çalma Alan

- çalma alan, nesnelerin ve bilgilerin kaydedildiği alandır.
  - `ls()` ve `objects()` fonksiyonlar çalma alanında kayıtlı nesneleri konsolda göstermektedir.
  - `ls()` fonksiyonu ile nesneleri çarma ilemi özelleştirilebilir.
  - `ls.str()` fonksiyonu ise hafızadaki nesneleri ayrıntılar ile göstermektedir.
  - Çok fazla kod yazıyorsanız, environment (veya çalma alanı) birçok nesne ile darmadan olduğunu fark edebilirsiniz. Bu, hangi nesneye ihtiyacınız olduğunu bulmanız zorlaşabilir ve bu nedenle yanlış veri seti kullanma riskiyle karşı karşıya kalabilirsiniz. Yeni bir veri kümesi üzerinde çalışıyorsanız veya son sürümünü elde etmeden önce çok sayıda farklı kod denediyseniz, yanlış nesneyi kullanmaktan kaçınmak için ortam/çalma alanı temizlemeyi unutmamak iyi bir uygulamadır. Bunu birkaç eklede yapabilirsiniz.
1. Nesneleri tek tek kaldırmak için konsola `rm(nesne_ad)` yazabilirsiniz. Önceki bölümde oluşturduğunuz nesnelerden birini kaldırmak için bunu şimdi deneyin.
  2. Ortamdaki tüm nesneleri temizlemek için konsolda `rm(liste = ls())` komutunu çalıştırın.
  3. Ortamdaki tüm nesneleri temizlemek için ortam bölmesindeki süpürge simgesine de tıklayabilirsiniz.
  4. • Konsolda yer alan işlemleri silmek için ise: CTRL + L (clear console) ya da süpürge iareti kullanılabilir.

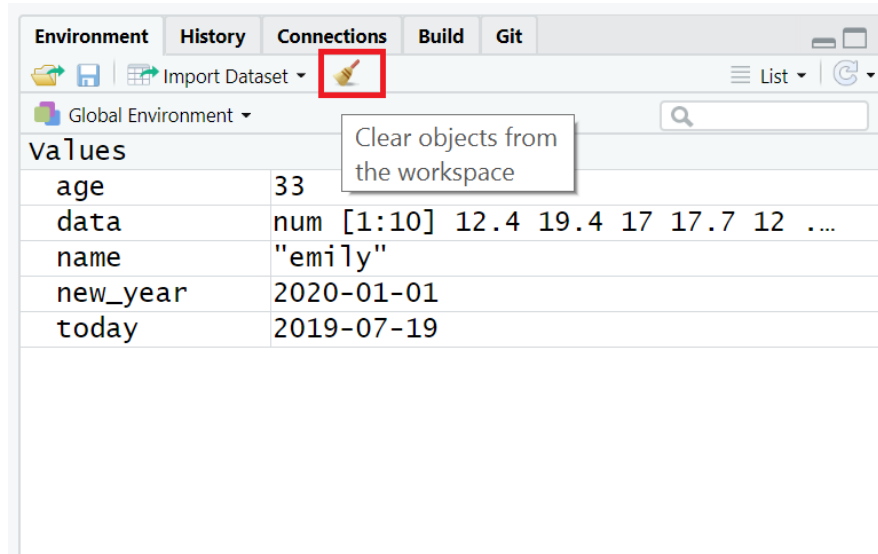


Figure 3.2: Clearing the workspace

### 3.5 R Çalma Dizini

- R yazılım Start/Başlangıç menüsü üzerinden çalıştırıldığında çalma dizini **C:/Users//Documents**
- Çalma dizinini sorgulamak için kullanılacak olan fonksiyon

- `getwd()` (get working directory)
- Çalma dizinini deitirmek için kullanılacak olan fonksiyon
  - `setwd()` (set working directory)
- Bu ilem Rstudio menusu “**Session**” sekmesinden ya da **CTRL + Shift + H** tular ile de yaplabilmektedir.

### 3.6 R’i Kapatma

- Kaydet (Save) ya da **CTRL + S** dosyadi.R uzantsyla kaydedilebilmektedir.
- Bu sayede tekrar kullanlabilmekte ya da bakalar ile kolaylıkla paylalabilmektedir.
- Tüm programlar gibi “**x**” iareti ile ya da `q()` fonksiyonunu ile sonlandrlabilir.
- R’dan çık yaparken, program çalma alanının kaydedilip kaydedilmeyeceini sormaktadr.
- Eger R’in çalma alanı kaydetmesini istenirse, R çalma dizinine ‘Rdata uzantlı bir dosya kaydeder.
- Çalma alan kayd için `save.image("dosyaadi")` komutu da kullanlabilmektedir.
- R’dan çık yapmadan yapılan ilem durdurulmak istenirse, konsol bölümündeki “**Stop**” iareti veya **Esc** tular kullanlabılır.

### 3.7 R Kaynaklar

- Alana özgü paketler
- Paket yardım sayfalar ve paket vignetteleri
- e\* posta gruplarındaki e\* postalara `RSiteSearch("sample.int")` ”
- `ltm_reliability` gibi fonksiyon isimler argumansız kullanırlırsa içeriği görünür. Karmaşık görünse de siz de yapabilirsiniz. Örenmek için iyi bir yoldur.
- <https://www.learnr4free.com/tr/index.html>
- Referans kartlar
- Cheat Sheets
- Hadley Wickham
- rforstats
- r is hot
- paralel programlama

### 3.8 Ödev

- Sadece temel pakette 1500’e yakın fonksiyon bulunduu için ders d alırmalar yapmanız gereklidir.
- R kurulumu ile ilgili learnr paketi hazırlanmış bir interaktif alıрма örneini inceleyiniz.
- Kitap Bölüm 1 alırmaların tamamlayınız.
- Datacamp da üzerine atanan bölüm alırmaların tamamlayınız.
- swirl package **learn R in R** paketi yükleyerek alıрма yapmay deneyiniz.
- Referans kart sayfasının çıktısını alarak duvarınıza asmanız öneririm.

### 3.9 R oturumlar

- R’yi açıp kod yazmaya, paketleri yüklemeye ve nesneler oluturmaya başladınızda, bunu yeni bir **oturumda** yaparsınız. Çalma alanı temizlemeye ek olarak, bazen yeni bir oturum balatmak yararlı olabilir.

Bu, bilgisayarınızda R'yi her balattınızda otomatik olarak gerçekleştirir, ancak oturumlar sunucuda kalabilir. Kodunuzun çalışmadı fark ederseniz ve nedenini bulamazsanız, yeni bir oturum balatmaya değer olabilir. Bu, ortamı temizleyecek ve yüklü tüm paketleri ayırarak - bunu telefonunuzu yeniden balatmak gibi düşünün.

### 3.10 Altrma

'Oturum - R'yi Yeniden Balat'a tıklayın.

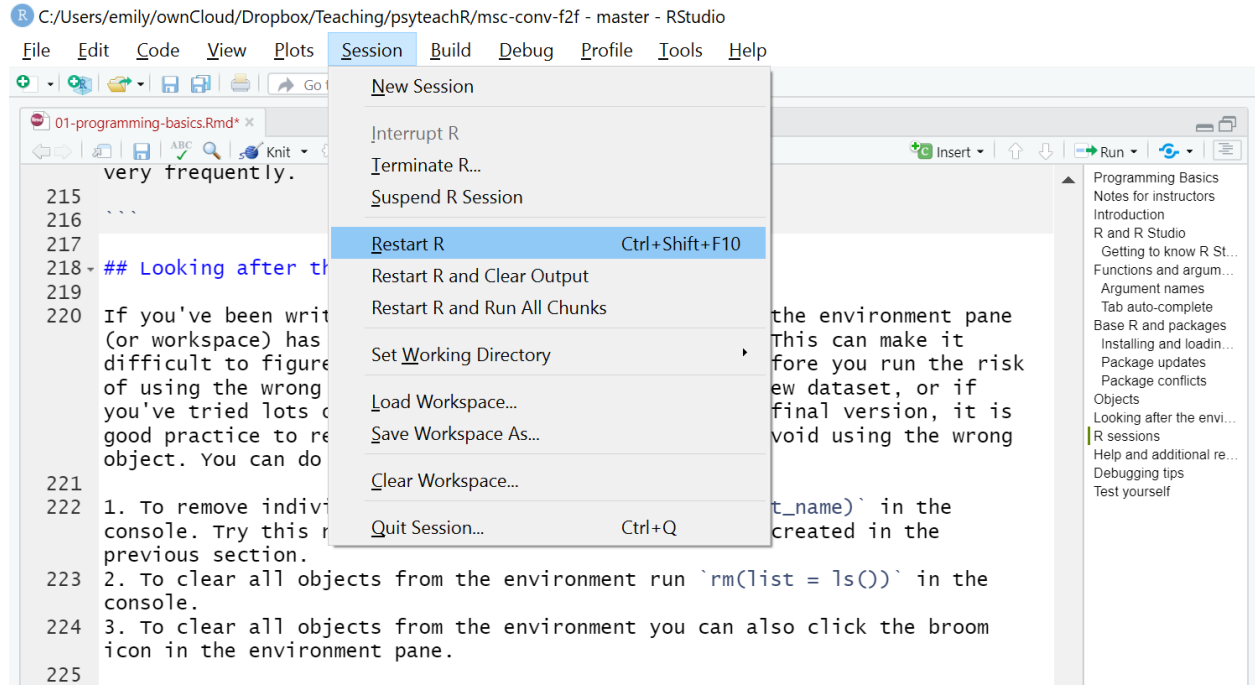


Figure 3.3: The truth about programming

### 3.11 Hata ayıklama ipuçları

-Kodlamanın büyük bir kısmı kodunuzun neden çalışmadığını anlamaya çalışmaktır ve bu acemi ya da uzman olmanız fark etmeksizin geçerlidir.

- Bu kurs boyunca ilerlerken yaptığınız hataların ve bunların nasıl düzelttiğinizin kayıtlarını tutmalısınız.
- Her bölümde dikkat etmeniz gereken bir dizi yaygın hata sunacağız, ancak ümitsiz kendiniz de yeni hatalar yapacaksınız (ve düzelteceksiniz!).
- Kullanmaya çalıştığınız fonksiyonlar için doğru paketleri yüklediniz mi? Çok yaygın bir hata, paketi yüklemek için kodu yazmaktır, örneğin `library(tidyverse)` ancak daha sonra çalıştırmayı unutmaktır.
- Bir yazım hatası mı yaptınız? Unutmayın `data` ile `DATA` aynı şey değildir ve `t.test` ile `t_test` aynı şey değildir.
- Bir paket çalışmıyor mu var? Paket ve fonksiyonu `package::function` ile belirtmeyi denediniz mi?
- Bu kesinlikle bir hata mı? R'deki tüm kırmızı metinler hata anlamına gelmez - bazen size sadece bilgi içeren bir mesaj verir.

### 3.12 Yardmc Kaynaklar

Programlamada iyi olmak demek, bir eyler denemek, internette yardm aramak ve kopyalanacak kod örnekleri bulmak demektir. B

- etkili bir ekilde problem çözmeyi öğrenmek, bu kurs boyunca gelitirmeniz gereken temel bir beceridir.
- Yardm belgelerini kullann. Bir fonksiyonun nasl çaltn anlamakta zorlanyorsanz, `?function` komutunu hatrlayn.
- Bir hata mesaj alırsanz, kopyalayp Google’a yaptrn - büyük olaslıkla baka biri de ayn sorunu yaamtr.
- Bu ders materyallerine ek olarak, R öğrenmek için bir dizi mükemmel kaynak vardır:
  - R Cookbook
  - StackOverflow
  - Veri Bilimi için R
  - Twitter’da `#rstats` hashtag’ini arayn veya kullann

### 3.13 Altrma : Kendini test et

**Soru 1.** Neden `install.packages()` kodunu analiz kodlarında asla dahil **etmemelisiniz**?

- (A) Bunun yerine `library()` kullanmalsnz
- (B) Paketler zaten temel R’n bir parçasdr
- (C) Siz (veya bir bakas) yanllkla kodunuzun çalması durduran bir paket güncellemesi yükleyebilirsiniz
- (D) Paketin en son sürümüne zaten sahipsiniz

Açklama

Unutmayın, `install.packages()` ilevini çaltrdınızda her zaman paketin en son sürümü yüklenir ve yüklemi olabileceiniz eski sürümlerin üzerine yazılır.

**Soru 2.** Aadaki kod ne üretecektir?

```
rnorm(6, 50, 10)
```

- (A) Ortalamas 6 ve SD’si 50 olan 10 saydan oluan bir veri seti
- (B) Ortalamas 50 ve SD’si 10 olan 6 saydan oluan bir veri seti
- (C) Ortalamas 10 ve SD’si 6 olan 50 saydan oluan bir veri seti
- (D) Ortalamas 10 ve SD’si 6 olan 50 saydan oluan bir veri seti

Açklama

`rnorm()` için varsayılan biçim `rnorm(n, mean, sd)` eklindedir. Bir fonksiyonun her bir argümanın ne ie yaradn hatrlamak için yardma ihtiyacnz varsa, `?rnorm` komutunu çaltrarak yardm belgelerine bakn

**Soru 3.** Ayn isimde fonksiyonlara sahip iki paketiniz varsa ve tam olarak hangi paketin kullanlacan belirtmek istiyorsanz, hangi kodu kullanırsnz?

- (A) `package::function`



- (B) `function::package`
- (C) `library(package)`
- (D) `install.packages(package)`

Açıklama

Örnein `dplyr::select` gibi `package::function` biçimini kullanmırsınız. Paketlerinizi ilk yüklediğinizde, herhangi bir fonksiyon aynı isme sahipse R'nin sizi uyaracağını unutmayın - buna dikkat etmeyi unutmayın!

**Soru 4.** Aşağıdakilerden hangisinin bir argüman olması en muhtemeldir?

- (A) 35
- (B) `read_csv()`
- (C) `<-`

**Soru 5.** Fonksiyonlar belirlemenin kolay bir yolu aşağıdakilerden hangisine bakmaktır

- (A) `()`
- (B) `[]`
- (C) `{}`

.

**Soru 6.** `<-`'nin görevi, fonksiyondan elde edilen çıktıyı bir/bir ..... atamaktır.

- (A) nesne
- (B) atama
- (C) argüman

.

## Chapter 4

# R Nesneler

- Örnek bir veri seti

```
library(tidyverse)
data(diamonds)
head(diamonds)
```

```
## # A tibble: 6 x 10
##   carat cut          color clarity depth table price      x      y      z
##   <dbl> <ord>        <ord> <ord>    <dbl> <dbl> <int> <dbl> <dbl> <dbl>
## 1  0.23 Ideal      E      SI2     61.5   55   326   3.95   3.98   2.43
## 2  0.21 Premium   E      SI1     59.8   61   326   3.89   3.84   2.31
## 3  0.23 Good      E      VS1     56.9   65   327   4.05   4.07   2.31
## 4  0.29 Premium   I      VS2     62.4   58   334   4.2    4.23   2.63
## 5  0.31 Good      J      SI2     63.3   58   335   4.34   4.35   2.75
## 6  0.24 Very Good J      VVS2     62.8   57   336   3.94   3.96   2.48
```

R nesne (object) yönelimli bir programlama dilidir.

- Karakter (character)
- Sayısal (numeric)
  - tam sayı (integer)
  - ondalıklı sayı (double)
  - karmaşık sayı (complex)
- Mantıksal (logical)
- Faktör (factor)
- Liste (list)
- Fonksiyon (function)

### 4.1 tam sayı

- tamsayı nesnesi oluşturulması

```
tamsayi <- 2L
```

- tamsayı nesnesinin türünün sorgulanması

```
typeof(tamsayi)
```

```
## [1] "integer"
```

- tamsayı nesnesinin yazdırılması

```
tamsayi
```

```
## [1] 2
```

## 4.2 ondalk say

- ondaliksayi nesnesinin oluturulmas

```
ondaliksayi <- 2.5
```

- ondaliksayi nesnesinin türünün sorgulanmas

```
typeof(ondaliksayi)
```

```
## [1] "double"
```

- ondaliksayi nesnesinin yazdırılması

```
ondaliksayi
```

```
## [1] 2.5
```

## 4.3 lemler

- tek elemanlı vektörler

```
x <- 1
```

```
y <- 1
```

```
x+y
```

```
## [1] 2
```

- çok elemanlı vektörler

```
x <- c(3,4,5)
```

```
y <- c(1,2,3)
```

```
# vektör eleman sayılar aynı mı?
```

```
length(x)==length(y)
```

```
## [1] TRUE
```

```
x+y
```

```
## [1] 4 6 8
```

```
x-y
```

```
## [1] 2 2 2
```

- çok elemanlı vektörler

```
x <- 1:9
```

```
y <- c(1,2,3)
```

```
# vektör eleman sayılar farklı mı?
```

```
length(x)/length(y)
```

```
## [1] 3
```

```
x+y
```

```
## [1] 2 4 6 5 7 9 8 10 12
```

```
x/y
```

```
## [1] 1.0 1.0 1.0 4.0 2.5 2.0 7.0 4.0 3.0
```

- çok elemanlı vektörler

```
x <- 1:5
y <- c(1,2)
# vektör eleman sayılar farklı olduğunda
length(x)/length(y)
```

```
## [1] 2.5
```

- x+y ileminin sonucu nedir? \_\_\_\_\_

Çözüm

```
x + y
```

```
## Warning in x + y: longer object length is not a multiple of shorter object
## length
```

```
## [1] 2 4 4 6 6
```

## 4.4 Karakter Nesneler

- karakter nesnesi oluşturulması

```
karakter <- "olcme"
```

- Oluşturulan nesnenin türünün sorgulanması

```
typeof(karakter)
```

```
## [1] "character"
```

- nesne yazdırılması

```
karakter
```

```
## [1] "olcme"
```

```
# karakter nesnesi oluşturulması
```

```
ad <- "Su"
```

```
soyad <- "Sevim"
```

- iki nesneyi arada boşluk bırakarak birleştirir.

```
paste(ad,soyad)
```

```
## [1] "Su Sevim"
```

- sep argümanı farklı eklerle özelleştirilebilir.

```
paste(ad,soyad, sep=" ")
```

```
## [1] "SuSevim"
```

```
paste(ad,soyad, sep="_")
```

```
## [1] "Su_Sevim"
```

- base pakette yer alan bazı karakter vektörleri bulunmaktadır.

```
letters
```

```
## [1] "a" "b" "c" "d" "e" "f" "g" "h" "i" "j" "k" "l" "m" "n" "o" "p" "q" "r" "s"
## [20] "t" "u" "v" "w" "x" "y" "z"
```

```
LETTERS
```

```
## [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J" "K" "L" "M" "N" "O" "P" "Q" "R" "S"
## [20] "T" "U" "V" "W" "X" "Y" "Z"
```

```
month.name
```

```
## [1] "January" "February" "March" "April" "May" "June"
## [7] "July" "August" "September" "October" "November" "December"
```

```
month.abb
```

```
## [1] "Jan" "Feb" "Mar" "Apr" "May" "Jun" "Jul" "Aug" "Sep" "Oct" "Nov" "Dec"
```

- Nesne birleştirme fonksiyonlarından en sık kullanan `paste()`
- `paste()` fonksiyonunun temel argümanları ise `sep` ve `collapse`'dir.

```
harf5 <- letters[1:5]
(harf51 <- paste(harf5, 1:5, sep="_"))
```

```
## [1] "a_1" "b_2" "c_3" "d_4" "e_5"
```

```
length(harf51)
```

```
## [1] 5
```

```
(harf52 <- paste(harf5, 1:5, sep="_",
collapse=" "))
```

```
## [1] "a_1 b_2 c_3 d_4 e_5"
```

```
length(harf52)
```

```
## [1] 1
```

- `paste()` fonksiyonunun yardım sayfasını inceleyiniz.

#### 4.4.1 Günün Sorusu

- Aadaki çıktıyı oluturacak olan kodu oluturunuz.

```
## [1] "1. maddenin guclugu: 0.52" "2. maddenin guclugu: 0.88"
## [3] "3. maddenin guclugu: 0.21" "4. maddenin guclugu: 0.67"
## [5] "5. maddenin guclugu: 0.69" "6. maddenin guclugu: 0.8"
## [7] "7. maddenin guclugu: 0.11" "8. maddenin guclugu: 0.35"
## [9] "9. maddenin guclugu: 0.16" "10. maddenin guclugu: 0.75"
```

Bunun birden fazla yolu olabilir, farklı ekollerde yapabilirsiniz.

**Büyük Küçük Harf Düzenleme Fonksiyonları** `toupper()` ve `tolower()`

```
toupper(harf5)
```

```
## [1] "A" "B" "C" "D" "E"
```

```
tolower(harf5)
```

```
## [1] "a" "b" "c" "d" "e"
```

casefold() fonksiyonu da upper argüman ile birlikte kullanılabilir.

```
casefold(harf5, upper = FALSE)
```

```
## [1] "a" "b" "c" "d" "e"
```

```
casefold(harf5, upper = TRUE)
```

```
## [1] "A" "B" "C" "D" "E"
```

- Karakter nesnelerin kaç harften olutuu nchar() fonksiyonu ile belirlenebilir.

```
nchar(month.name)
```

```
## [1] 7 8 5 5 3 4 4 6 9 7 8 8
```

- Karakter nesneleri belli bir yerden bölmek için substr() ve substring() fonksiyonlar kullanılabilir.

```
substr("YILMAZ", 1,3)
```

```
## [1] "YIL"
```

- substring("YILMAZ", 1:... , 1:6) kodunda “Y” “I” “L” “M” “A” “Z” çkts oluturacak kodu yaznz  
—
- ‘substring("YILMAZ", ... , 4:6) kodunda “ILM” “ILMA” “ILMAZ”‘ çktns oluturacak kodu yaznz  
—
- Karakter nesnelerde daha fazlas için aadaki fonksiyonlar inceleyebilirsiniz.
- strsplit()
- noquote()
- cat()
- grep()
- duplicated()
- agrep()

## 4.5 Mantksal Nesneler

```
mantiksal1 <-TRUE
```

```
typeof(mantiksal1)
```

```
## [1] "logical"
```

```
mantiksal1
```

```
## [1] TRUE
```

Mantksal operatörler programlamann temeli ve vazgeçilmezidir.

```
# eitlik snamas  
T==TRUE
```

```
## [1] TRUE
```

- 4==5 kodunun sonucu nedir? \_\_\_\_\_
- 4<5 kodunun sonucu nedir? \_\_\_\_\_
- 10>100 kodunun sonucu nedir? \_\_\_\_\_

- Mantksal operatörlerle yapılan snamalar ile mantksal nesneler oluturulur.

```
sonuc <- 4<5
typeof(sonuc)
```

```
## [1] "logical"
```

- Nesne türleri arasındaki deim uygunluk durumuna göre `as.*()` fonksiyonlar ile salanr.

```
# Karakter veri numerik veriye
as.numeric("3.14")
```

```
## [1] 3.14
```

```
# ondalk verin tam sayya
as.integer(pi)
```

```
## [1] 3
```

```
# karakter veri numerik veriye (NA)
as.numeric("olcme")
```

```
## Warning: NAs introduced by coercion
```

```
## [1] NA
```

```
# mantksal veri karakter veriye (NA)
as.character(TRUE)
```

```
## [1] "TRUE"
```

```
# numerik veri karakter veriye
as.character(10)
```

```
## [1] "10"
```

```
# mantksal veri numerik veriye
as.numeric(TRUE)
```

```
## [1] 1
```

## 4.6 Nesne Türleri Sorgulama

- Nesne türleri sorgulamak için ise `class()` ya da `mode()` fonksiyonlar kullanabilir. Ancak bir nesne türüne ait olup olmadn sorgulamak için ise `is.*()` fonksiyonlar kullanlr.

```
x<- 3.14; class(x)
```

```
## [1] "numeric"
```

```
is.numeric(x)
```

```
## [1] TRUE
```

```
is.logical(x)
```

```
## [1] FALSE
```

- Saysal nesnelerin türü için `typeof()` fonksiyonu da kullanlabılır.

```
y <- 2L; typeof(y); class(y) # satir içi kod ayırma
```

```
## [1] "integer"
```

```
## [1] "integer"
```

```
is.integer(y)
```

```
## [1] TRUE
```

```
is.double(y)
```

```
## [1] FALSE
```

#### 4.6.1 Günün Sorusu

- aada yer alan **ad\_soyad** nesnesini kullanarak

```
ad_soyad<- c("Ayse-Sel","Can-Yucel","Cem-Togay","Banu-Cift")
```

aadaki çıktıyı oluturmaya çalışınız.

```
## [1] "Ayse" "Can"  "Cem"  "Banu"
```

```
## [1] "Sel"   "Yucel" "Togay" "Cift"
```



## Chapter 5

# Vektörler

- R lineer cebir temelli bir programlama dilidir.
- Vektörler tek boyutludur.
- R'da vektörler birletirmek (combine/concatenate) anlamına gelen `c()` fonksiyonu ile oluşturulmaktadır.
- R'da veriler bir araya gelerek veri yapıları oluşturulur.
  - vektör (vector)
  - liste (list)
  - matris (matrix)
  - veri seti (data.frame)
  - dizi (array)

### 5.1 Vektör Oluşturma

```
(sayisal_vektor <- c(1,2,3))  
  
## [1] 1 2 3  
(karakter_vektor <- c("a","b","c")) ## çift tırnak  
  
## [1] "a" "b" "c"  
(mantıksal_vektor <- c(TRUE,TRUE,FALSE))  
  
## [1] TRUE TRUE FALSE
```

### 5.2 Vektör lemleri

- Vektör uzunluğu `length()` fonksiyonu ile vektör türleri ise `class()`, `mode()` ya da `typeof()` fonksiyonları ile tür belirlemek için kullanılmaktadır.
- Vektörler bir veya daha fazla elemandan oluşabilmektedir.

```
a <- 1 # tek elemandan oluşur.  
# Vektör uzunluğunu öğrenmek için length() fonksiyonu  
length(a)  
  
## [1] 1
```

```
x <- 1:10
```

- bir vektöründeki verilerin toplanması `sum(x)` 55
- bir vektöründeki verilerin çarpılması `prod(x)`  $3.6288 \times 10^6$
- bir vektöründeki verilerin küçükten büyüye sıralanması `sort(x)` 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
- bir vektörünün elemanların sıralarının tersine çevrilmesi `rev(x)` 10, 9, 8, 7, 6, 5, 4, 3, 2, 1
- bir vektöründeki verilerin standart sapmasının hesaplanması `sd(x)` 3.0276504
- bir vektöründeki en büyük verinin gösterilmesi `max(x)` 10
- bir vektöründeki en küçük verinin gösterilmesi `min(x)` 1
- En büyük verinin vektörün kaçın eleman olduğunu gösterilmesi `which.max(x)` 10
- En küçük verinin vektörün kaçın eleman olduğunu gösterilmesi `which.min(x)` 1
- Vektörlerden eleman sırası, isim ve mantıksal operatörler olmak üzere üç farklı yolla eleman seçilebilir.

```
ad <- c("Ali", "Elif", "Su", "Deniz",  
"Aras", "Berk", "Can", "Ece", "Efe", "Arda")
```

- `ad` vektörünün 1. elemanı `ad[1]` Ali
- `ad` vektörünün 5. elemanın yazdıracak kodu oluturunuz. \_\_\_\_\_
- `ad` vektörünün son elemanın yazdıracak kodu oluturunuz. \_\_\_\_\_
- `ad` vektörünün son elemanın yazdıracak kodu vektörün 10 elemanlı olduğunu bilmediğiniz de ne yaparsınız?  
\_\_\_\_\_
- Vektörün sadece 1., 5. 8 elemanın yazdıracak kodu oluturunuz. \_\_\_\_\_
- Vektörün sadece 1. elemanın hariç tutacak kodu oluturunuz \_\_\_\_\_
- Vektörün 1. ve 5. elemanın hariç tutacak kodu oluturunuz \_\_\_\_\_
- Vektörün son üç elemanın yazdıracak kodu oluturunuz `ad[8:10]`

### 5.3 Vektöre eleman eklenmesi

```
ad[11] <- "Asu"; ad
```

```
## [1] "Ali" "Elif" "Su" "Deniz" "Aras" "Berk" "Can" "Ece" "Efe"  
## [10] "Arda" "Asu"
```

- vektöre birden fazla eleman eklenmesi

```
ad[12:13] <- c("Ahu", "Han"); ad
```

```
## [1] "Ali" "Elif" "Su" "Deniz" "Aras" "Berk" "Can" "Ece" "Efe"  
## [10] "Arda" "Asu" "Ahu" "Han"
```

- Vektörün ortasına eleman eklenmesi `append()` fonksiyonu ile yapılabilir. Fonksiyon yardım sayfasını inceleyiniz.

```
(ad <- append(ad, "Taha", after = 3))
```

```
## [1] "Ali" "Elif" "Su" "Taha" "Deniz" "Aras" "Berk" "Can" "Ece"  
## [10] "Efe" "Arda" "Asu" "Ahu" "Han"
```

- ya da `c()` fonksiyonu ile yapılabilir.

```
ad <- c(ad[1:5], "Selim", ad[7:length(ad)]); ad
```

```
## [1] "Ali" "Elif" "Su" "Taha" "Deniz" "Selim" "Berk" "Can" "Ece"
## [10] "Efe" "Arda" "Asu" "Ahu" "Han"
```

## 5.4 Altrma

- 10 kiiden oluan bir gruptaki kiilerin boy ve kilo ölçümleri için ise aadaki vektör oluturulmutur.

```
ad <- c("Ali", "Elif", "Su", "Deniz",
"Aras", "Berk", "Can", "Ece", "Efe", "Arda")
boy <- c(160, 165, 170, 155, 167, 162, 169, 158, 160, 164)
kilo <- c(50, 55, 57, 50, 48, 65, 58, 62, 45, 47)
```

- Eer elimizdeki vektör isimlendirilmi bir vektör ise eleman seçimini isimle de yapabiliriz.

```
#isimsiz boy vektörü
names(boy) # names() fonksiyonu ile isimlendirme yapılabilir.
```

```
## NULL
```

- ad vektörünü boy vektörünü isimlendirirken nasl kullanabiliriz? \_\_\_\_\_
- Arda 'nn boyunu isimlendirilmi vektörü kullanarak nasl yazdrsrnz? `rftb("boy["Arda"]")`

## 5.5 Örüntülerle Vektör Oluturma

- Vektör oluturmann farkl yollar bulunmaktadr.
- En basit yolu iki nokta ":" operatörünü kullanmaktr.

```
rakamlar <- 0:9
rakamlar
```

```
## [1] 0 1 2 3 4 5 6 7 8 9
```

- büyükten küçüie rakamlardan vektör oluturulmas

```
rakamlar <- 9:0
rakamlar
```

```
## [1] 9 8 7 6 5 4 3 2 1 0
```

### 5.5.1 seq()

- Belirli bir kurala göre say dizileri oluturmak için ise `seq()`, `rep()` ve `paste()` fonksiyonlardan yararlanabilir. İlk olarak bu fonksiyonların yardım sayfalarını inceleyelim.
- 1'den 10'a kadar birer birer artan sayılardan dizi oluturulacak kodu oluturunuz. `seq(from=1, to=10, by=...)`

—

- Bir önceki ilemi argümanınız olarak oluturunuz. \_\_\_\_\_
- Aynı çıktıyı tek bir argümanla elde edebilir misiniz? \_\_\_\_\_
- `length` argümanını kullanarak aadaki çıktıyı oluturacak kodu oluturunuz. \_\_\_\_\_

```
## [1] 1.0 1.4 1.8 2.2 2.6 3.0
```

- `by` argümanını ile art miktarını kullanarak aadaki çıktıyı oluturacak kodu oluturunuz. \_\_\_\_\_

```
## [1] 1.0 1.5 2.0 2.5 3.0
```

- Belirli bir aralıkta kaç elemanın yer alacağını `length.out` argümanı kullanarak aadaki çıktıyı oluturacak kodu oluturunuz. \_\_\_\_\_

```
## [1] 1.0 1.5 2.0 2.5 3.0
```

### 5.5.2 rep()

`rep()` fonksiyonu için örnekler

```
# üç elemanlı bir vektörün üç kere tekrar ettirilmesi
rep(c(3,4,5), 3)
```

```
## [1] 3 4 5 3 4 5 3 4 5
```

```
# rakamların üç kere tekrar ettirilmesi
rep(0:9, times = 3)
```

```
## [1] 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9
```

- `a <- c(3,5,7)` vektörünü kullanarak aadaki çıktıyı elde edecek kodu hazırlaynz. \_\_\_\_\_

```
## [1] 3 3 3 5 5 5 7 7 7
```

- `a <- c(3,5,7)` vektörünü kullanarak aadaki çıktıyı elde edecek kodu hazırlaynz. \_\_\_\_\_

```
## [1] 3 3 3 5 5 5 7 7 7 3 3 3 5 5 5 7 7 7 3 3 3 5 5 5 7 7 7
```

- Çıktıyı elde edecek kodu hazırlaynz. \_\_\_\_\_

```
## [1] 1 1 2 2 3 3 4 4
```

- Çıktıyı elde edecek kodu hazırlaynz. \_\_\_\_\_

```
## [1] 1 2 2 3 3 3
```

### 5.5.3 paste()

- `paste()` fonksiyonu çkts her zaman için karakterdir.

```
paste(1:4) # çkts karakterdir
```

```
## [1] "1" "2" "3" "4"
```

```
class(paste(1:4))
```

```
## [1] "character"
```

- Çıktıyı elde edecek kodu tamamlaynz `paste("test",...)` \_\_\_\_\_

```
## [1] "test 1" "test 2" "test 3" "test 4" "test 5" "test 6" "test 7"
```

```
## [8] "test 8" "test 9" "test 10"
```

- Çıktıyı elde edecek kodu tamamlaynz `paste("test",1:10,"...",sep="_")` \_\_\_\_\_

```
## [1] "test_1_puan" "test_2_puan" "test_3_puan" "test_4_puan" "test_5_puan"
```

```
## [6] "test_6_puan" "test_7_puan" "test_8_puan" "test_9_puan" "test_10_puan"
```

- Çıktıyı elde edecek kodu tamamlaynz `paste("test",c("A","B","C","D",...))` \_\_\_\_\_

```
## [1] "test A" "test B" "test C" "test D" "test 1" "test 2" "test 3" "test 4"
```

## 5.6 Rasgele Veri Oluturma

- Farklı fonksiyonlarla rastgele veri üretilebilir. Örnek 0-100 arasında 20 farklı değer elde edilmek istenilsin. Bunu yapmak için `sample()`, `runif()` ya da `rnorm()` fonksiyonlarından yararlanılabilir.

```
sample(0:100,5)
```

```
## [1] 29 23 51 42 72
```

```
runif(10, 0, 5)
```

```
## [1] 1.40673901 0.57241018 2.17336066 3.96671890 2.68911204 4.51717755
```

```
## [7] 1.64190804 0.65066983 4.30389799 0.09875575
```

```
rnorm(10,50,5)
```

```
## [1] 49.08342 52.13135 50.98775 44.32830 49.60631 49.16930 42.16739 51.17894
```

```
## [9] 59.22586 54.84551
```

- Kullanılan üç fonksiyonun da yardım sayfaları ve kullanım amaçlarını inceleyiniz.

## 5.7 İlemler

BKI vücut ağırlığının metre cinsinden boy uzunluğunun karesine bölünmesi ile elde edilmektedir. Her bir bireye ait BKI değerini hesaplayınız. BKI değerlerinin ortalamasını kaçtır (iki ondalık yuvarlayınız)? \_\_\_\_\_

```
ad <- c("Ali","Elif","Su","Deniz","Aras","Berk","Can","Ece","Efe","Arda")
boy <- c(160,165,170,155,167,162,169,158,160,164)
kilo <- c(55,55,57,50,48,65,58,62,45,47)
```

Çözüm!, bakmadan yapmayın!

```
# BKI hesaplanması
boy_m <- boy/100
BKI <- kilo/( boy_m * boy_m)
BKI
round(mean(BKI),2)
```

```
## [1] 21.48437 20.20202 19.72318 20.81165 17.21109 24.76757 20.30741 24.83576
```

```
## [9] 17.57812 17.47472
```

```
## [1] 20.44
```

### 5.7.1 Kendinizi Test Edin

**S1.** Aadaki tabloda yer alan üç sütun için birer vektör oluturunuz. Öğrencilerin geçme notu her iki sınavın ortalaması olarak hesaplanacaktır. Bu öğrencilerin geçme notlarını hesaplayınız. Geçme notlarının betimsel istatistiklerini hesaplayınız.

Öğrenci	Vize	Final
Oğrenci1	50	45
Oğrenci2	55	65
Oğrenci3	60	85
Oğrenci4	70	90
Oğrenci5	80	85

Geçme notlarının minimum değeri: \_\_\_\_\_

Geçme notların ortalama değeri: \_\_\_\_\_

Geçme notların maksimum değeri: \_\_\_\_\_

**S2.** Birden  $n$ 'e kadar olan sayların toplamı hesaplayan fonksiyon yazın `toplam()` tek argümanlı fonksiyon oluturunuz. Argüman değeri 5 olduğunda  $1+2+3+4+5=15$  değerini versin.

- birden  $n$ 'e kadar olan sayların toplamı:  $(n*(n+1))/2$

**S3.** 1'den  $n$ 'e kadar olan sayların toplamı hesaplayan fonksiyonu argümanlı olarak aadaki şekilde yazmay deneyiniz. Fonksiyonu çalıştırdığınızda ekranda/konsolda kaç tane kadar olan sayların toplamı hesaplandı yazsın, kullanıcının girdi değeri göre aada çıktı çıksın.

```
toplam()
```

```
kaç tane kadar olan sayların toplamı hesaplandı: 10
```

```
[1] " 10 'e kadar olan sayların toplamı: 55"
```

## 5.8 ÖDEV

- Kitap Bölüm 2 1. Soruyu tamamlayınız.
- swirl package - learn R in R (Programming ilk 6 modül)
- datacamp ödevinizi yapmayı unutmayın

## Chapter 6

# Veri Setleri

- Veri setleri iki boyutludur.
- R'da bir çok fonksiyonun veri setleri ile çalışmaktadır.
- Veri setleri R ortamında `data.frame()` fonksiyonu ile oluşturulabilir.
- `data.frame()` fonksiyonu ile aynı uzunluktaki vektörlerden bir veri seti oluşturulabilir.

```
ad <- c("Ali","Elif","Su","Deniz","Aras", "Berk","Can","Ece","Efe","Arda")
boy <- c(160,165,170,155,167,162, 169,158,160,164)
kilo <- c(55,55,57,50,48,65, 58,62,45,47)
beden <- c("S","M","S","M","S", "L","M","L","S","S")
beden <- factor(beden)
```

```
(df <- data.frame(ad,boy, kilo, beden))
```

```
##      ad boy kilo beden
## 1   Ali 160   55     S
## 2  Elif 165   55     M
## 3    Su 170   57     S
## 4 Deniz 155   50     M
## 5  Aras 167   48     S
## 6  Berk 162   65     L
## 7   Can 169   58     M
## 8   Ece 158   62     L
## 9   Efe 160   45     S
## 10 Arda 164   47     S
```

- Eer uzunluklar farklı olan vektörlerle veri setleri oluşturulmaya çalışırsa kısa vektör, uzun vektör uzunluğunda tekrar eder.

```
# 4 farklı uzunlukta vektör oluşturulması
x <- 11:14; y <- 10; M <- c(10,35); N <- 2:4
```

```
data.frame(x, y) # (4,1)
```

```
##      x  y
## 1 11 10
## 2 12 10
## 3 13 10
```

```
## 4 14 10
data.frame(x, M) # (4,2)

##      x  M
## 1 11 10
## 2 12 35
## 3 13 10
## 4 14 35
data.frame(x,N) # (4,3) hata

## Error in data.frame(x, N): arguments imply differing number of rows: 4, 3
data.frame(y, M) # (1,2)

##      y  M
## 1 10 10
## 2 10 35
data.frame(y, N) # (1,3)

##      y N
## 1 10 2
## 2 10 3
## 3 10 4
data.frame(M, N) # (2,3)

## Error in data.frame(M, N): arguments imply differing number of rows: 2, 3
```

## 6.1 Hazır Veri Setleri

- Temel pakette yer alan veri setlerinin bir listesine aadaki komutla ulaşabilirsiniz.

```
data() # yeni bir pencerede açılır.
```

- Veri setlerinin yer ald paketlerde bulunmaktadır.

```
# install.packages("datasets")
library(datasets)
# install.packages("dslabs")
library(dslabs)
```

- Hazır veri setleri çalışma ortamına `data()` fonksiyonu ile aktarılabilir.

```
data(WorldPhones) # enviroanmete kontrol ediniz.
```

- hazır veri setlerini incelemek için aadaki komutlar kullanılabilir.

```
data(cars) # enviromente ekler
iris      # enviromente eklemez!
```

## 6.2 nceleme

- Boyut sorgulamamak için farklı fonksiyonlar kullanılabilir.

```
dim(cars) # satır Sütun
```

```
## [1] 50 2
```



```
nrow(cars)
```

```
## [1] 50
```

```
ncol(cars)
```

```
## [1] 2
```

- Veri setlerin ilk satrlar `head()`, son satrlar ise `tail()` fonksiyonu ile incelenebilir. `head()` fonksiyonu olaan olarak ilk 6 satr yazdr.

```
head(WorldPhones)
```

```
##      N.Amer Europe Asia S.Amer Oceania Africa Mid.Amer
## 1951  45939  21574 2876   1815   1646    89    555
## 1956  60423  29990 4708   2568   2366   1411   733
## 1957  64721  32510 5230   2695   2526   1546   773
## 1958  68484  35218 6662   2845   2691   1663   836
## 1959  71799  37598 6856   3000   2868   1769   911
## 1960  76036  40341 8220   3145   3054   1905  1008
```

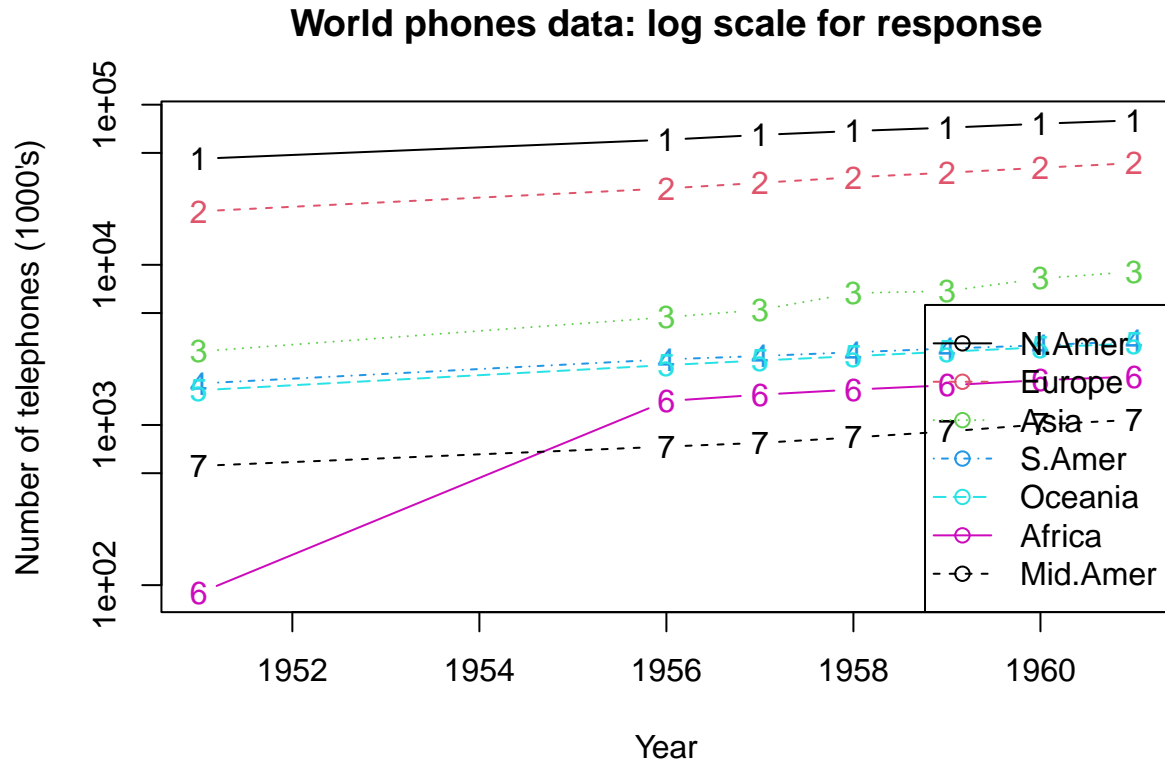
- Yazdrılacak satr says `n` argüman ile ayarlanr.

```
head(WorldPhones,n=2)
```

```
##      N.Amer Europe Asia S.Amer Oceania Africa Mid.Amer
## 1951  45939  21574 2876   1815   1646    89    555
## 1956  60423  29990 4708   2568   2366   1411   733
```

- `WorldPhones` veri setinin son 8 satrn yazdracak kodu yaznz. \_\_\_\_\_
- `datasets` paketinde yer alan veri setlerinde `examples()` bölümünde ğeitli örneklere yer verilmitir. Örnein `example(WorldPhones)`

```
matplot(rownames(WorldPhones), WorldPhones, type = "b", log = "y",
        xlab = "Year", ylab = "Number of telephones (1000's)")
legend("bottomright", colnames(WorldPhones), col = 1:6, lty = 1:5,
      pch = rep(21, 7))
title(main = "World phones data: log scale for response")
```



- Temel paket hariç diğer paketlerdeki veri setlerine `data(veriseti, package="packagename")` şeklinde ulaşılabilir.

```
data(CTTdata, package="CTT")
head(CTTdata)
```

- sk kullanılan veri setleri ile ilgili bir yaz:
- tüm veri setlerine ulaşabilmek için ise:
- Kullanıl olmasa da excel, spss gibi veri girişi sağlayan bir arayüz bulunmaktadır.
- Ali, Su ve Ece'nin boylarının ve kilolarının seçilmesi

```
df1<- data.frame()
df1 <- edit(df1)
# düzenlemek için
fix(df)
# gözetmek için
View(df)
```

### 6.3 Eleman Seçme

Veri setlerinde eleman seçme matrislerdeki gibidir.

```
df[satirindeks, sutunindeks]
```

- df'nin birinci satır elemanlarının seçilmesi \_\_\_\_\_

```
##      ad boy kilo beden
```

```
## 1 Ali 160 55 S
```

- df'nin birinci sütun elemanlarının seçilmesi \_\_\_\_\_

```
## [1] "Ali" "Elif" "Su" "Deniz" "Aras" "Berk" "Can" "Ece" "Efe"
## [10] "Arda"
```

- df'nin ikinci satır elemanlarının seçilmesi \_\_\_\_\_

```
##      ad boy kilo beden
## 2 Elif 165 55 M
```

- df'nin ikinci sütun elemanlarının seçilmesi \_\_\_\_\_

```
## [1] 160 165 170 155 167 162 169 158 160 164
```

- df'nin birinci satır üçüncü sütun elemanlarının seçilmesi \_\_\_\_\_

```
## [1] 55
```

- Veri setlerinde satır elemanlar yazdırıldığında veri seti (`data.frame`), sütun elemanlar yazdırıldığında ise vektör (`vector`) olmaktadır.

```
# satır secimi
is.data.frame(df[1,])
```

```
## [1] TRUE
```

```
# sütun secimi
is.data.frame(df[,1])
```

```
## [1] FALSE
```

- Sütun seçimi veri seti (`data.frame`) olarak yapılmak istenirse, `drop` argümanı `FALSE` değeri ile kullanılır.

```
df[,1,drop=FALSE]
```

```
##      ad
## 1   Ali
## 2  Elif
## 3    Su
## 4 Deniz
## 5  Aras
## 6  Berk
## 7   Can
## 8   Ece
## 9   Efe
## 10 Arda
```

- Veri seçim işlemi için `subset()` fonksiyonu kullanılabilir.
- ?subset bir fonksiyonu ilk daha kullanıyorsanız, mutlaka yardım sayfasını inceleyin.

```
subset(veriseti, kosul/Kosullar)
```

- Boyu 165cm den uzun öğrencilerin bilgilerinin seçilmesi

```
subset(df, boy >165)
```

```
##      ad boy kilo beden
## 3   Su 170 57 S
## 5 Aras 167 48 S
## 7  Can 169 58 M
```

- `subset()` Fonksiyonun yardım sayfasındaki örnekleri inceleyebilirsiniz.

```
subset(airquality, Temp > 90, select = c(Ozone, Temp))
```

```
##      Ozone Temp
## 42      NA  93
## 43      NA  92
## 69      97  92
## 70      97  92
## 75      NA  91
## 102     NA  92
## 120      76  97
## 121     118  94
## 122      84  96
## 123      85  94
## 124      96  91
## 125      78  92
## 126      73  93
## 127      91  93
```

```
subset(airquality, Day == 1, select = -Temp)
```

```
##      Ozone Solar.R Wind Month Day
## 1       41      190  7.4      5   1
## 32      NA      286  8.6      6   1
## 62     135      269  4.1      7   1
## 93       39       83  6.9      8   1
## 124      96      167  6.9      9   1
```

- df verisinde beden deikenini "S" olan satırların seçimi `subset(df, beden == "S")`
- df verisinde kilosu 50'in altında olan kişilerden oluşan veri seti oluşturma kodunu tamamlayınız  
`subset(df, .....)` \_\_\_\_\_

## 6.4 Eleman ekleme

- Veri setine yeni sütun ekleme işlemi `$` operatörü ile `[[ ]]` operatörü ile `cbind()` fonksiyonları ile yapılabilir.

```
df2 <- data.frame(
  S1 = sample(0:100, 20),
  S2 = runif(n=20, min= 50, max=70)
)
head(df2)
```

```
##      S1      S2
## 1 45 53.57809
## 2 71 53.87062
## 3 34 63.24698
## 4 75 60.10678
## 5 36 52.82080
## 6 69 60.51641
```

- `$` operatörü ile sütun ekleme

```
df2$S3 <- sample(60:80, 20, replace = TRUE)
head(df2)
```

```
##      S1      S2 S3
```

```
## 1 45 53.57809 76
## 2 71 53.87062 74
## 3 34 63.24698 80
## 4 75 60.10678 80
## 5 36 52.82080 65
## 6 69 60.51641 70
```

- `[[ ]]` operatörü ile sütun ekleme
- `df2` veri setinin ilk üç sütunun `rowMeans()` fonksiyonu ile ortalamasının alınarak `ort` isimi ile veri setine eklenmesi

```
df2[["ort"]] <- round(rowMeans(df2),2)
head(df2)
```

```
##   S1      S2 S3   ort
## 1 45 53.57809 76 58.19
## 2 71 53.87062 74 66.29
## 3 34 63.24698 80 59.08
## 4 75 60.10678 80 71.70
## 5 36 52.82080 65 51.27
## 6 69 60.51641 70 66.51
```

- `cbind()` fonksiyonu ile sütun ekleme

```
cbind( df2, S4 = 10)
```

```
##   S1      S2 S3   ort S4
## 1 45 53.57809 76 58.19 10
## 2 71 53.87062 74 66.29 10
## 3 34 63.24698 80 59.08 10
## 4 75 60.10678 80 71.70 10
## 5 36 52.82080 65 51.27 10
## 6 69 60.51641 70 66.51 10
## 7  5 69.58202 80 51.53 10
## 8 73 54.70502 68 65.24 10
## 9 43 51.10038 69 54.37 10
## 10 42 61.50939 67 56.84 10
## 11 11 53.09087 64 42.70 10
## 12 61 69.65540 60 63.55 10
## 13 19 60.52792 62 47.18 10
## 14 25 52.91681 75 50.97 10
## 15 89 50.05696 79 72.69 10
## 16 80 50.32710 70 66.78 10
## 17  2 50.00605 72 41.34 10
## 18 87 67.33245 64 72.78 10
## 19 44 66.10432 78 62.70 10
## 20 84 50.85064 70 68.28 10
```

## 6.5 Eleman çıkarma

- Veri setinden istenilen sütunun çıkarılabilir. Bu işlemi yapmak için iki farklı yol kullanılabilir.
- `-` operatörü

```
head(df2,3)
```

```
##   S1      S2 S3   ort
```

```
## 1 45 53.57809 76 58.19
## 2 71 53.87062 74 66.29
## 3 34 63.24698 80 59.08
```

```
df2 <- df2[,-4]
head(df2,3)
```

```
##      S1      S2 S3
## 1 45 53.57809 76
## 2 71 53.87062 74
## 3 34 63.24698 80
```

- NULL operatörü

```
df2$S3 <- NULL
head(df2,3)
```

```
##      S1      S2
## 1 45 53.57809
## 2 71 53.87062
## 3 34 63.24698
```

## 6.6 Satr ekleme

- Veri setlerine deiken ekleyip, çıkarabileceğiniz gibi gözlem de ekleyip, çıkarabilirsiniz. Veri setine iki satr ekleme

```
dim(df2)
```

```
## [1] 20 2
```

```
# eklenecek iki satırlık veri seti oluturma
df3 <- data.frame(S1=c(50,60),S2=c(55.3,65.5))
# yeni veri seti
df4 <- rbind(df2,df3)
dim(df4)
```

```
## [1] 22 2
```

## 6.7 Veri yapsı inceleme

- Veri setlerinin yapsını incelemek için `str()` fonksiyonundan yararlanılmaktadır.

```
str(df)
```

```
## 'data.frame': 10 obs. of 4 variables:
## $ ad : chr "Ali" "Elif" "Su" "Deniz" ...
## $ boy : num 160 165 170 155 167 162 169 158 160 164
## $ kilo : num 55 55 57 50 48 65 58 62 45 47
## $ beden: Factor w/ 3 levels "L","M","S": 3 2 3 2 3 1 2 1 3 3
```

- “df” veri seti 10 gözlemden, 4 deiken. Her bir deikenin önünde \$ operatörü olduuna dikkat ediniz.
- veri setinin incelenmek için kullanılabilecek dier fonksiyon ise `attributes()`

```
attributes(df)
```

```
## $names
## [1] "ad" "boy" "kilo" "beden"
```

```
##
## $class
## [1] "data.frame"
##
## $row.names
## [1] 1 2 3 4 5 6 7 8 9 10
```

## 6.8 Isimlendirme

- Veri setleri vektör birleştirme üzerinden yapılırsa, vektör adlar sütun ismi olarak kullanılır. Ancak bu isimler değiştirilebilir. Bu işlem `data.frame()` fonksiyonu içinde yapılabilir.

```
df <- data.frame(isim = ad,
                  boyolcum = boy,
                  kiloolcum= kilo,
                  bedenolcum=beden)
df
```

```
##      isim boyolcum kiloolcum bedenolcum
## 1    Ali      160      55          S
## 2   Elif      165      55          M
## 3     Su      170      57          S
## 4  Deniz      155      50          M
## 5   Aras      167      48          S
## 6   Berk      162      65          L
## 7    Can      169      58          M
## 8    Ece      158      62          L
## 9    Efe      160      45          S
## 10  Arda      164      47          S
```

- Veri seti isimlendirme de diğer bir yol ise `names()` ya da `colnames()` fonksiyonlardır.

```
df <- data.frame(ad,boy,kilo,beden)
names(df) <- c("isim","boyolcum","kiloolcum","bedenolcum")
df
```

```
##      isim boyolcum kiloolcum bedenolcum
## 1    Ali      160      55          S
## 2   Elif      165      55          M
## 3     Su      170      57          S
## 4  Deniz      155      50          M
## 5   Aras      167      48          S
## 6   Berk      162      65          L
## 7    Can      169      58          M
## 8    Ece      158      62          L
## 9    Efe      160      45          S
## 10  Arda      164      47          S
```

## 6.9 Betimsel istatistikler

- Veri setinin tümüne ilişkin betimsel istatistikler

```
summary(cars)
```

```
##      speed      dist
## Min.   : 4.0   Min.   : 2.00
```

```
## 1st Qu.:12.0    1st Qu.: 26.00
## Median :15.0    Median : 36.00
## Mean   :15.4    Mean   : 42.98
## 3rd Qu.:19.0    3rd Qu.: 56.00
## Max.   :25.0    Max.   :120.00
```

- Veri setinin tek deikenine ilikin betimsel istatistikler

```
summary(cars$speed)
```

```
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      4.0   12.0   15.0   15.4   19.0   25.0
```

`attach()` fonksiyonu ile bir veri setinin sütunlar sütun isimi ile enviromente eklenir. Ayn ilem `detach()` fonksiyonu ile tersine alınabilir.

```
summary(women$height)
attach(women)
summary(height)      # Aynı nesne isimi ile çarlr.
height <- height*2.54 # Bunu yapmamaya calisin!!
find("height")
```

```
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      58.0   61.5   65.0   65.0   68.5   72.0
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      58.0   61.5   65.0   65.0   68.5   72.0
## [1] ".GlobalEnv" "women"
```

```
summary(height)      # Yeni deiken
rm(height)
detach("women")
summary(women$height) # unchanged
```

```
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##     147.3  156.2  165.1  165.1  174.0  182.9
##      Min. 1st Qu.  Median      Mean 3rd Qu.      Max.
##      58.0   61.5   65.0   65.0   68.5   72.0
```

### 6.9.1 Kendinizi Test Edin

**S1.** Srayla deiken adlar `TamSayi`, `OndalikSayi`, `Karakter`, `Mantksal`, `Faktör` olan 5 deikenli hiçbir gözlemi olmayan bir `data.frame` oluturmanz ve bu `data.frame`in yapasn yazdrmanz bekliyorum. Beklenen çkt aadaki gibi olmaldr.

```
[1] "Bos data.framein yaps:"
'data.frame':  0 obs. of  5 variables:
 $ TamSayi      : int
 $ OndalikSayi : num
 $ Karakter     : chr
 $ Mantksal    : logi
 $ Faktör      : Factor w/ 0 levels:
NULL
```

**S2.** Aada size verilen dört vektörden bir veri seti oluturunuz. Oluturduunuz veri setinin deneme sütunundaki eksik veri sayasn hesaplayan komut yaznz.

```
ad = c('Su', 'Pera', 'Sule', 'Can', 'Cem', 'Name', 'Aras', 'Mete', 'Kaan', 'Pelin')
puan = c(12.5, 9, 16.5, 12, 9, 20, 14.5, 13.5, 8, 19)
```



```
deneme = c(1, NA, 2, NA, 2, NA, 1, NA, 2, 1)
bonus  = c(1,0,1, 0, 0, 1, 1, 0,0, 1)
```

“Deneme sütunundaki NA says:” [1] 4

## 6.10 Odev

Lütfen aadaki bölümleri haftaya kadar okuyunuz.

- <http://adv-r.had.co.nz/Data-structures.html>
- <http://adv-r.had.co.nz/Subsetting.html>
- Veri düzenleme konusunda **DataEditR** paketini inceleyiniz.

## Chapter 7

# Veri Okuma ve Yazma

- Veri girii istatistiksel analiz sürecinin ilk admdır.
- R’da veri girii dier yazlımlarla kıyaslandnda **çok kullanı deildir**.
- Bu nedenle aktarma/import yolu tercih edilir.
- Veri aktarm için çok sayıda fonksiyon ve paket bulunmaktadır.
- Ayrıca **menü ile de aktarma** yapılabilir.
- Bilgisayardan internetten farklı formattaki veriler okunabilir.
- Veri setleri genellikle Excel, SPSS veya metin dosyalar (.txt, .csv, .dat, vb.) gibi uygun veri biçimlerinde kaydedilir
- R, çeitli veri formatların içe aktarabilir (yani okuyabilir).

Bir veri setini R’ye aktarmann iki yolu vardır:

1. RStudio’da “Veri Kümesini Çe Aktar” menü seçeneini kullanarak
2. Belirli bir R komutunu kullanarak
  - Çe aktarmak istediiniz dosyaya göz atın.
  - Veri seti için bir isim verin.
  - Çe aktarılacak sayfayı seçin.
  - Deiken isimleri dosyanın ilk satırındaysa “First Row as Names”.

### 7.0.1 SPSS dosyaların içe aktarma

- Çe aktarmak istediiniz dosyaya göz atın.
- Veri seti için bir isim verin.

## 7.1 Veri Okuma

- En temel veri okuma/aktarma fonksiyonları
  - `scan()`
  - `read.*`
  - `read.table()`
  - `read.csv()`

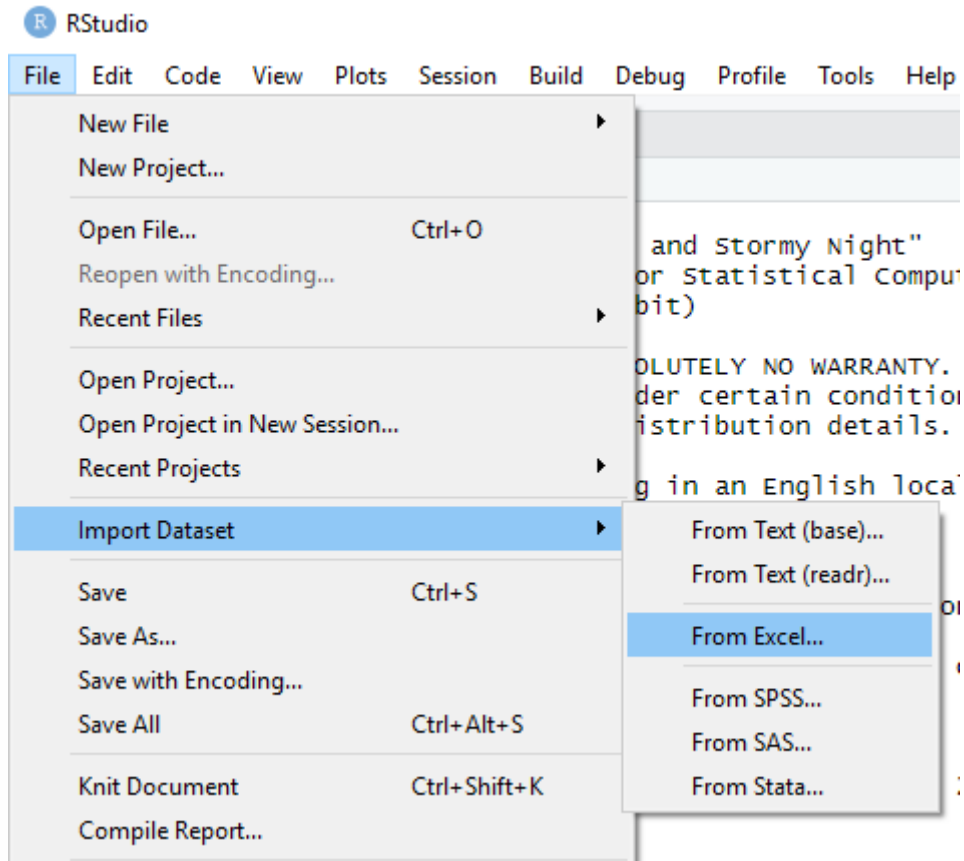


Figure 7.1: R studio

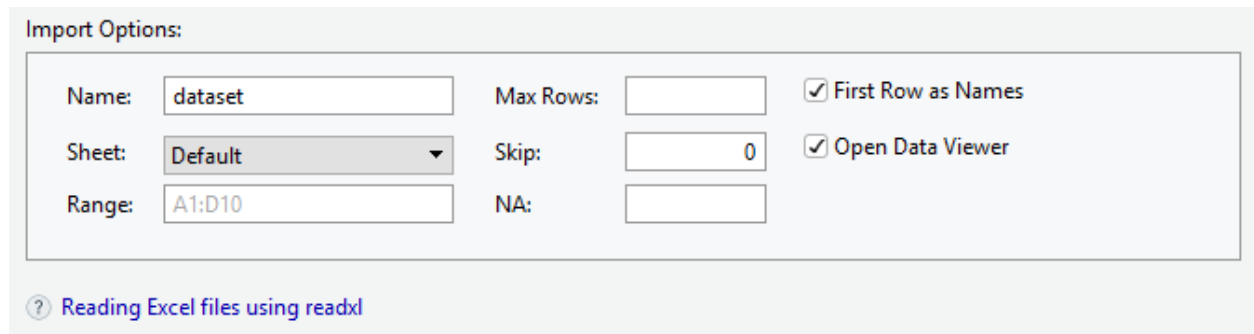


Figure 7.2: Excel dosylarn içe aktarma

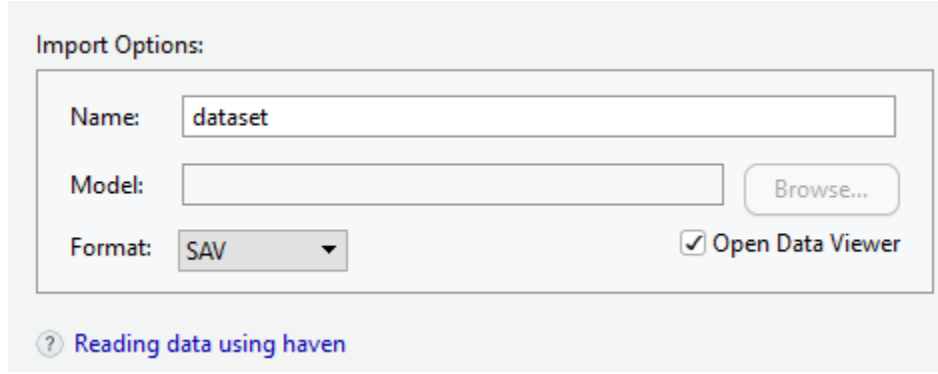


Figure 7.3: SPSS dosyaların içe aktarma

- `read.csv2()`
- `read.delim()`
- `read.delim2()`
- `readLines()`
- Verinin düzgün girilmi olmas okumayı kolaylatır.
- İlk satırda genellikle deiken adlarına (header), ilk sütunda ise kimlik veya sıra numarasına yer verilir.
- Gözlemlere ve deikenlere ilkin veri girilirken karakterler veya sayısal değerler arasında boşluk bırakılmaktan kaçınmak gerekmektedir. Deiken adı boşluklu yazılırsa ne olur?
- Eksik veri boyunca aynı şekilde girilmelidir.
- Deikenlerin birinden başka ayrılmış önemlidir. (, ; : / )
- Tercihimiz `.csv` uzantılı veriler ama büyük veri setleri az yer kalmaması için `.txt`, `.prn` formatında kaydedilebilir.
- Temel pakette `read.csv` ve `read.table` gibi bazı fonksiyonlar bulunmaktadır.
- Ayrıca, belirli formatların içe aktarmak için R paketleri bulunmaktadır. Örneğin, SPSS dosyaları için `foreign` ve Excel dosyaları için `xlsx` gibi

## 7.2 `read.*()` fonksiyonları

Argüman	Açıklama
<code>header</code>	Mantıksal değerler ile verinin ilk satırında deiken isimlerinin olup olmadığını test eder.
<code>sep</code>	Sütun ayracıdır.
<code>na.strings</code>	Kayıp değerleri belirtmek için kullanılır.
<code>dec</code>	Ondalık sayıların ne ile ayrıldığını gösteren argümandır.
<code>nrows</code>	Okunmak istenilen satır sayısını belirtmek için kullanılır.
<code>skip</code>	Bir dosya okunurken okunmadan atlanmak istenilen satır sayısı için kullanılır.

## 7.3 Excel dosyas aktarma

```
# yükle ve aktive et
install.packages("xlsx")
library("xlsx")

# read.xlsx fonksiyonunun kullanm
my_excel_file <- read.xlsx("dizin/dosyaadi.xlsx",sheetName = "sheetname")
```

### 7.3.1 SPSS dosyas aktarma

```
# yükle ve aktive et
install.packages("foreign")
library("foreign")

# read.spss fonksiyonunun kullanm
my_spss_file <- read.spss("dizin/dosyaadi.sav",
                          to.data.frame = TRUE)
```

### 7.3.2 text dosyas aktarma

- text dosyalar okumak için paket yüklemeye gerek yoktur.

```
# , ile ayrılm csv dosyalar
csv_dosya <- read.csv("dizin/dosyaadi.csv",header = TRUE)

# tab ile ayrılm txt dosyalar
txt_dosya <- read.table("dizin/dosyaadi.txt",header = TRUE, sep = "\t")
```

- Dikkat
- header = TRUE
- sep="\t"
- sep=", " for comma-separated files
- 

## 7.4 Uygulama

- Dosyalar buradan klasör halinde indirebilirsiniz. DOSYALAR
- veril.txt

```
##      no m_1  m_2 m_3  m_4 m_5
## 1 522  12 14.0  16 20.0  10
## 2 222   5  NA  20 10.0  10
## 3 454   5 10.2   6  4.0  10
## 4 567  10 20.0  NA 12.2  20
```

- veril.csv

```
##      no m_1  m_2 m_3  m_4 m_5
## 1 522  12 14.0  16 20.0  10
## 2 222   5  NA  20 10.0  10
## 3 454   5 10.2   6  4.0  10
```

```
## 4 567 10 20.0 NA 12.2 20
```

•

```
##      No M1  M2 M3  M4 M5
## 001 522 12  14 16  20 10
## 002 222  5 <NA> 20  10 10
## 003 454  5 10,2  6   4 10
## 004 567 10  20 NA 12,2 20
```

•

```
##      M1  M2 M3  M4 M5
## 001 12  14 16  20 10
## 002  5 <NA> 20  10 10
## 003  5 10,2  6   4 10
## 004 10  20 NA 12,2 20
```

• verifwf.txt

•

```
##  V1  V2 V3 V4 V5 V6 V7 V8 V9 V10 V11 V12 V13
## 1  1 689 A  1  0  1  0  1  0  1  0  1  0
## 2  2 654 B  1  1  1  1  0  1  0  1  0  1
## 3  3 436 A  1  0  1  0  1  1  1  1  1  1
```

•

```
##  sira  no kitapcik m1 m2 m3 m4 m5 m6 m7 m8 m9 m10
## 1    1 689          A  1  0  1  0  1  0  1  0  1  0
## 2    2 654          B  1  1  1  1  0  1  0  1  0  1
## 3    3 436          A  1  0  1  0  1  1  1  1  1  1
```

• factor.sav

```
## tibble [50 x 3] (S3: tbl_df/tbl/data.frame)
## $ id : num [1:50] 1 2 3 4 5 6 7 8 9 10 ...
## ..- attr(*, "format.spss")= chr "F6.2"
## $ bolge: num [1:50] 1 1 1 1 1 1 1 1 1 1 ...
## ..- attr(*, "format.spss")= chr "F6.2"
## $ puan : num [1:50] 9 8 6 8 10 4 6 5 7 7 ...
## ..- attr(*, "format.spss")= chr "F6.2"
```

• <https://www.statmodel.com/usersguide/chap3/ex3.1.dat>

```
## [1] -0.354517  0.561655  0.315551  3.347049 -0.122389 -0.251276 -0.517996
## [8]  1.888854  0.461254  2.237483
```

•

```
## [1] " -0.354517      0.573051     -0.175230"
## [2] "  0.561655     -0.368095      1.090042"
## [3] "  0.315551     -0.577052      0.425472"
## [4] "  3.347049      1.088520      1.149353"
## [5] " -0.122389     -0.694153     -0.766538"
## [6] " -0.251276     -0.017487     -1.367410"
## [7] " -0.517996     -0.817974     -1.559255"
## [8] "  1.888854     -0.658335      1.007614"
## [9] "  0.461254      0.463916     -0.898300"
## [10] "  2.237483      1.533398      0.180512"
```