



ile Veri Analizi

Dr. Bülent Çobanoğlu

cobanoglubulent[at]gmail.com

- **Sunu içeriği;**

- Veri ve Analizi
- R ve Python Veri Analizi Yazılımları Karşılaştırması
- R Kurulumu ve Araçlar
- R Dili ile İlgili Faydalanılacak Kaynaklar
- Paket Kurulumu
- Veri Giriş-Çıkış Komutları
- Değişkenler, Sabitler
- Yorumlar
- Operatörler ve işlem önceliği
- Veri Türleri (double, integer, complex, ...)
- Tip dönüşümleri
- Karar Yapıları(if, else, switch, ,...)

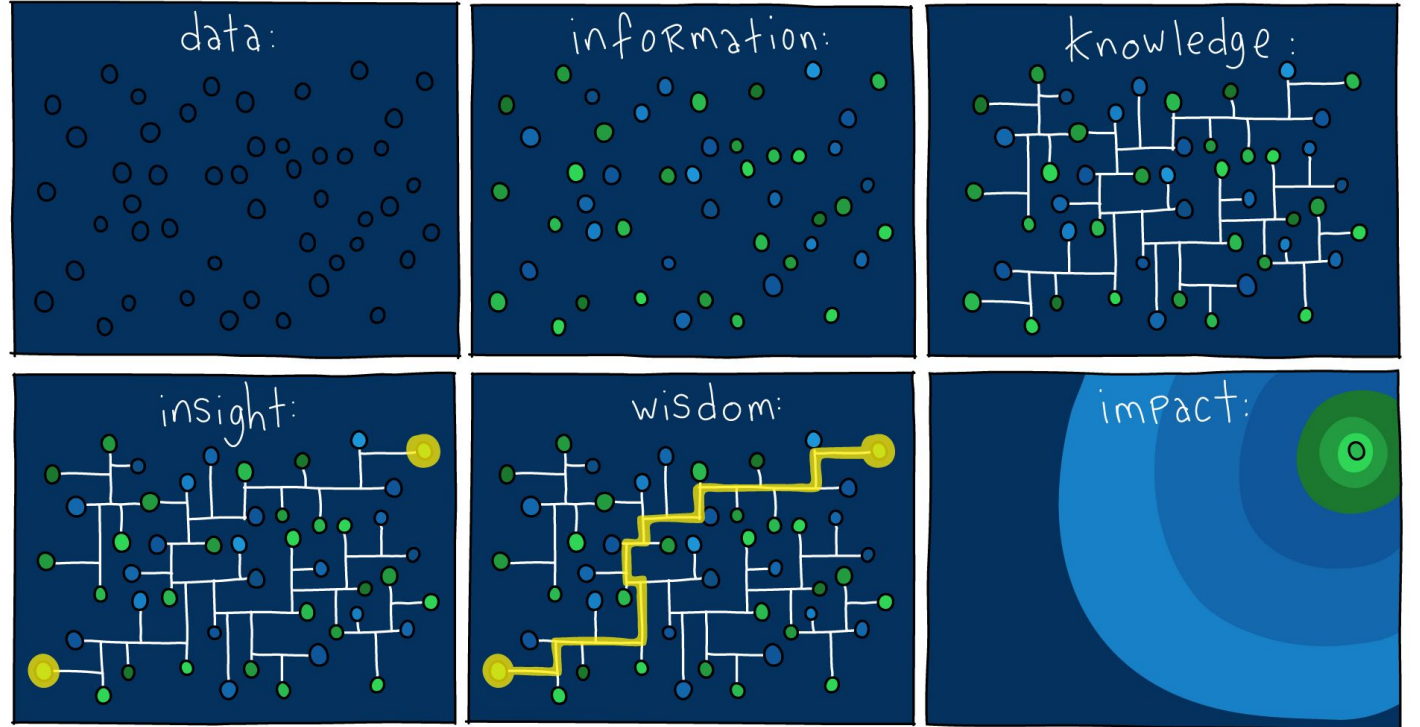
Kaynaklar



- <https://www.r-project.org/>
- <https://r4ds.had.co.nz/>
- <https://rstudio-education.github.io/hopr/starting.html>
- <https://intro2r.com/>
- <https://info201.github.io/r-intro.html>
- ...

Data ve evrimsel yolculuđu?

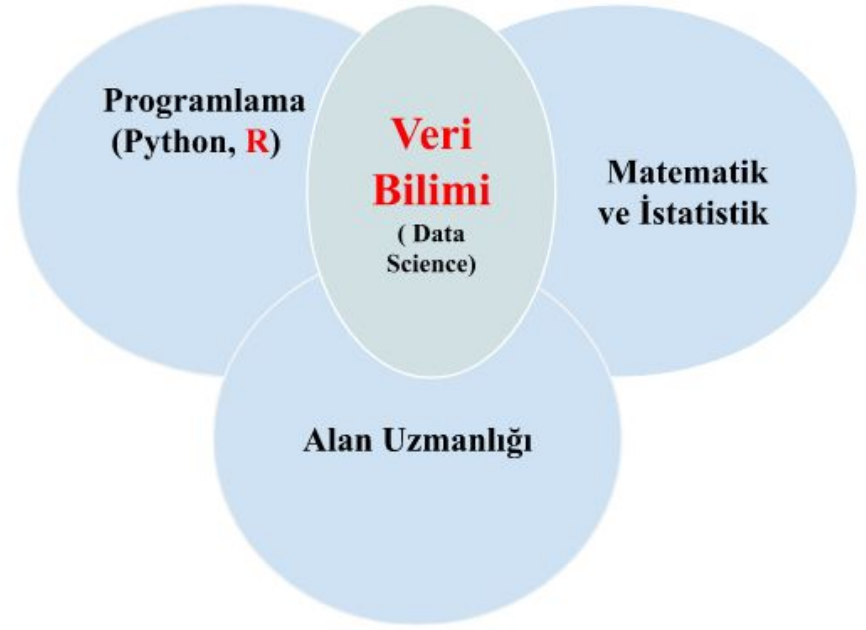
"The art of being wise is knowing what to overlook." - William James



@gapingvoid

03112023 -> 03.11.2023 -> 03.11.2023 10^{10} -> -500.000->?

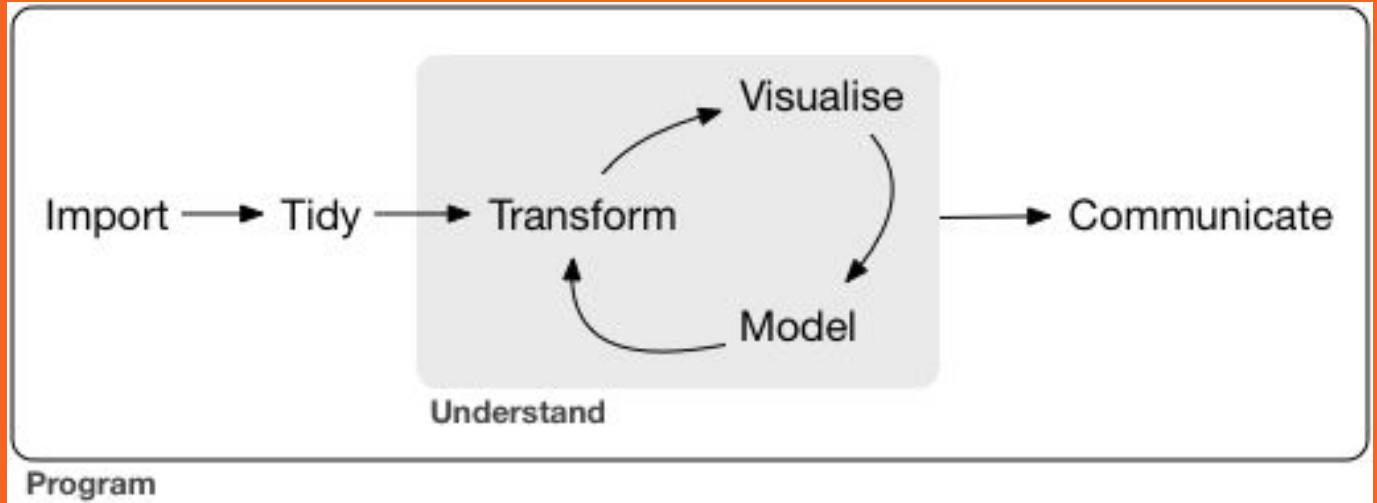
Data Science?



Veri bilimi (data science); *verilerden bir değer elde etmek ve anlamlı çıkarımlar yapabilmek için alan bilgisi (uzmanlığı), istatistik, programlama dili ve araçları* gibi birçok alanı bir araya getiren çok disiplinli bir bilim dalıdır.

Veri Analizi

Tipik bir veri bilimi projesinin modellemesi (%80 şuna benzer):



Program

İlgili veri setini al(import) -> Düzenle(tidy) -> Analiz et ve Dönüştür(transform)-> Görselleştir(Visualise) -> Modelle(Model) -> İletişim ve pazarlama (Communicate)

Q: Veri bilimi için tercih edilen diller neler? Avantaj ve dezavantajları hakkında neler söylersiniz?



Students, write your response!

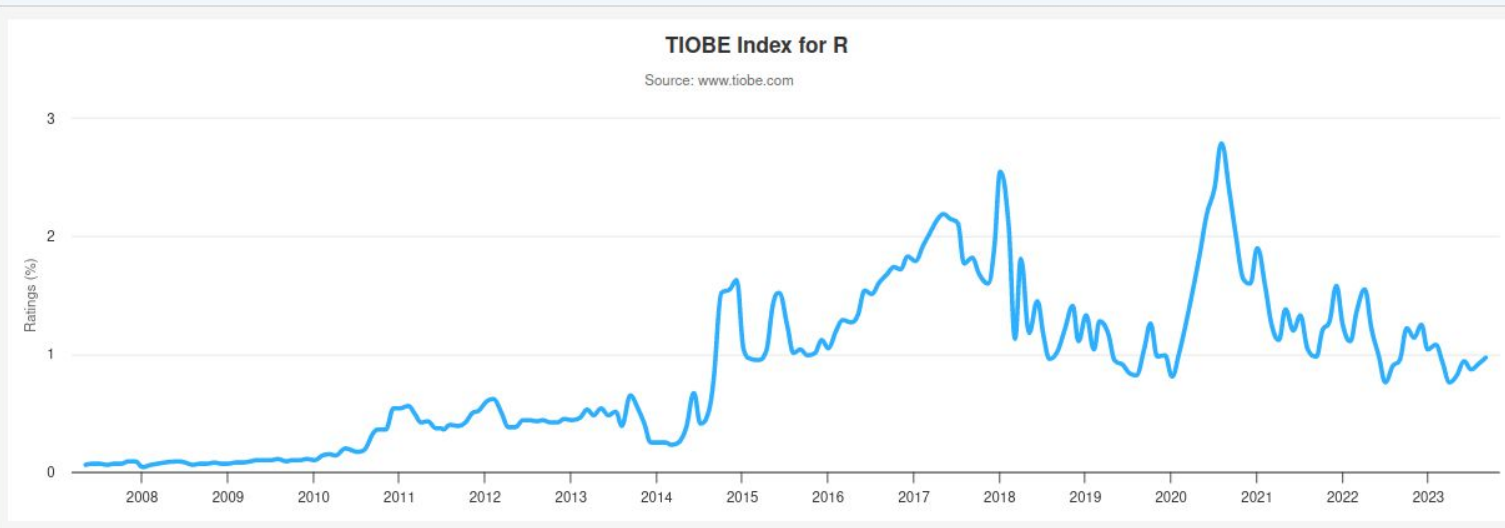
Niçin R?

The R Programming Language

Some information about R:

📈 Highest Position (since 2007): #8 in Aug 2020

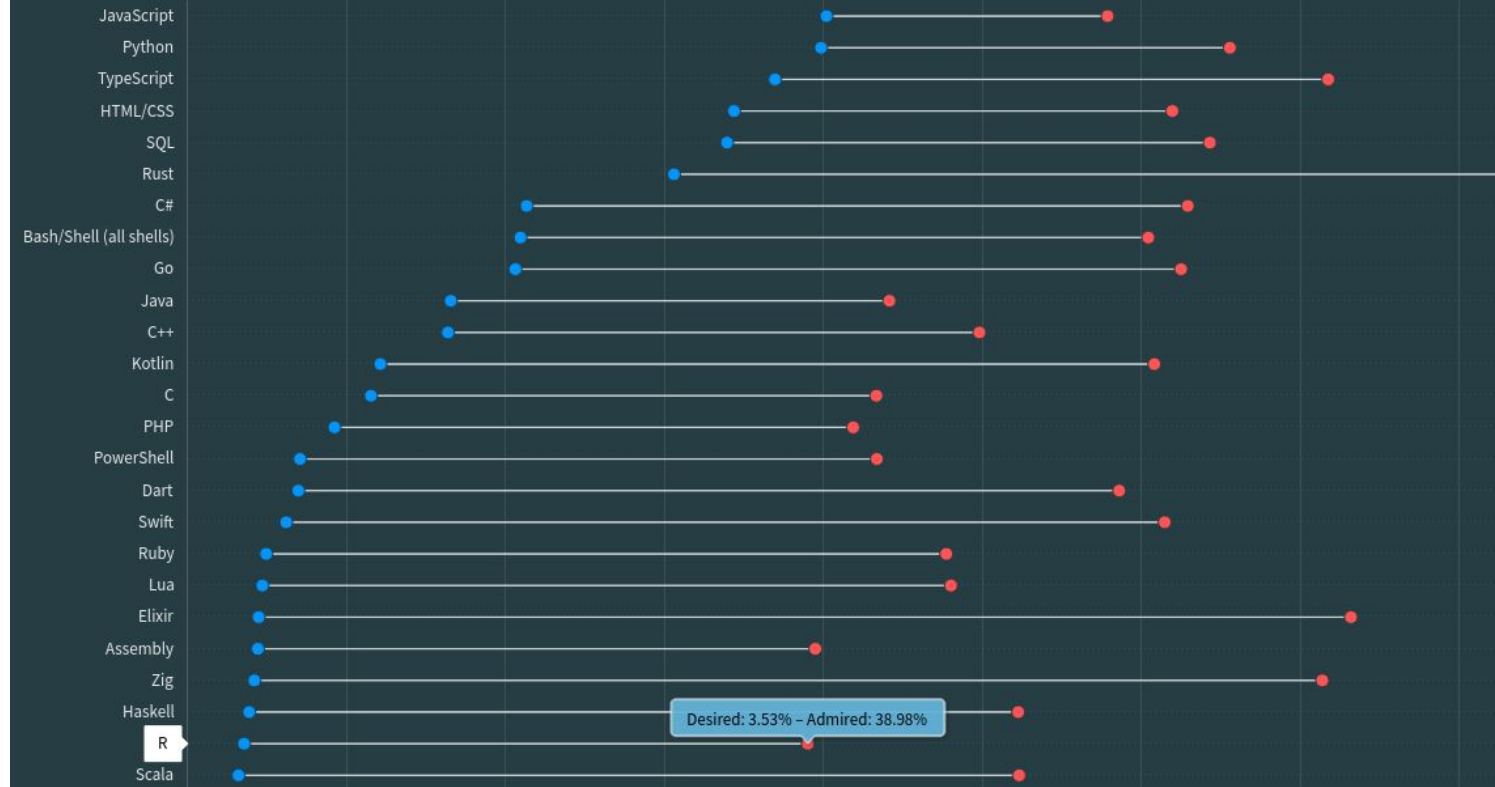
📉 Lowest Position (since 2007): #73 in Dec 2008



<https://www.tiobe.com/tiobe-index/>

Niçin R?

Kullanıcıları
tarafından ne kadar
seviliyor?



R Nasıl bir DİL?



- İstatiksel analizler ve grafiksel çizimler için geniş paketler içeren,
- Sınırlı programlama becerisine sahip araştırmacılar için kolay bir dil 😊 .

R Nasıl bir DİL?



- **Açık kaynaklı** (Open source),
 - **Yorumlayıcı tabanlı** (interpreted),
 - **Her yerde çalışabilen** (platform independent)
 - **Yüksek seviyeli** (High level),
 - **Nesne yönelimli** (OOP),
 - Diğer yazılım dilleri (C, C++, Java, Python) ile entegre çalışabilen **vektör tabanlı** bir dildir. 😊
-



<https://www.stat.auckland.ac.nz/~ihaka/>

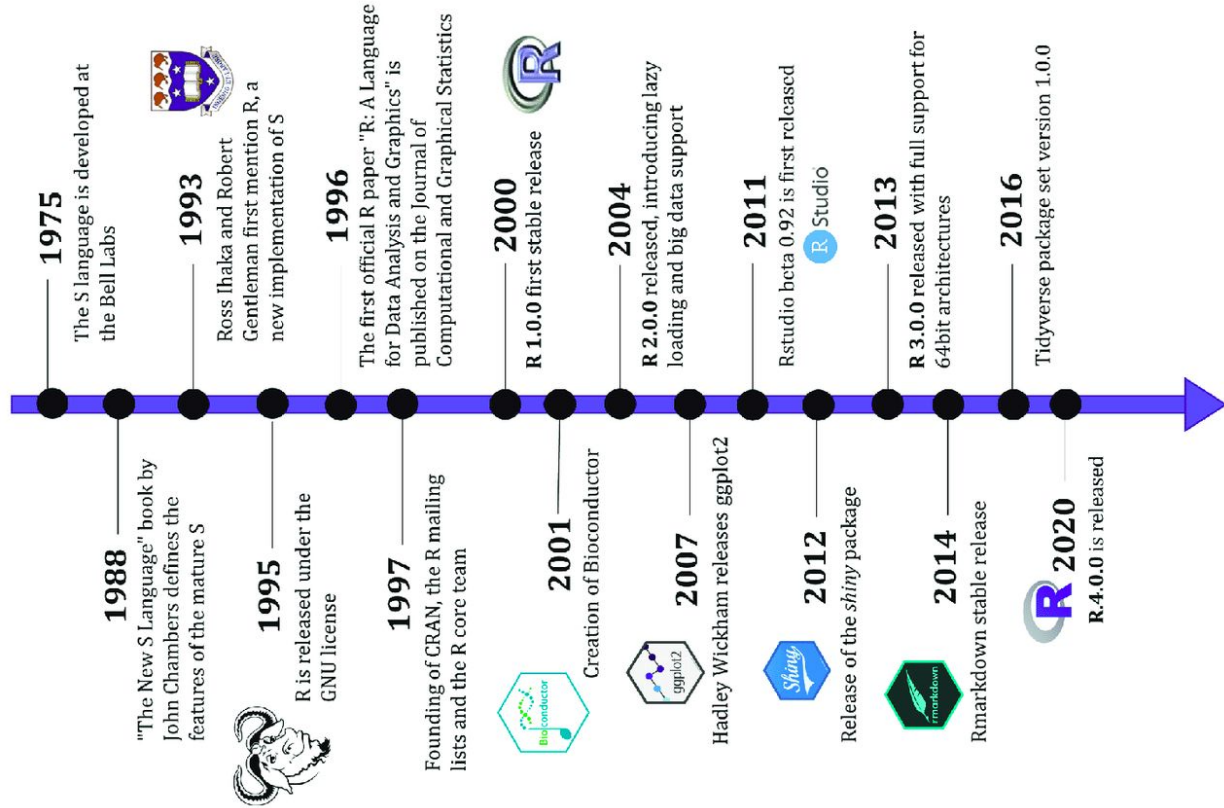


<https://dbmi.hms.harvard.edu/people/robert-gentleman>

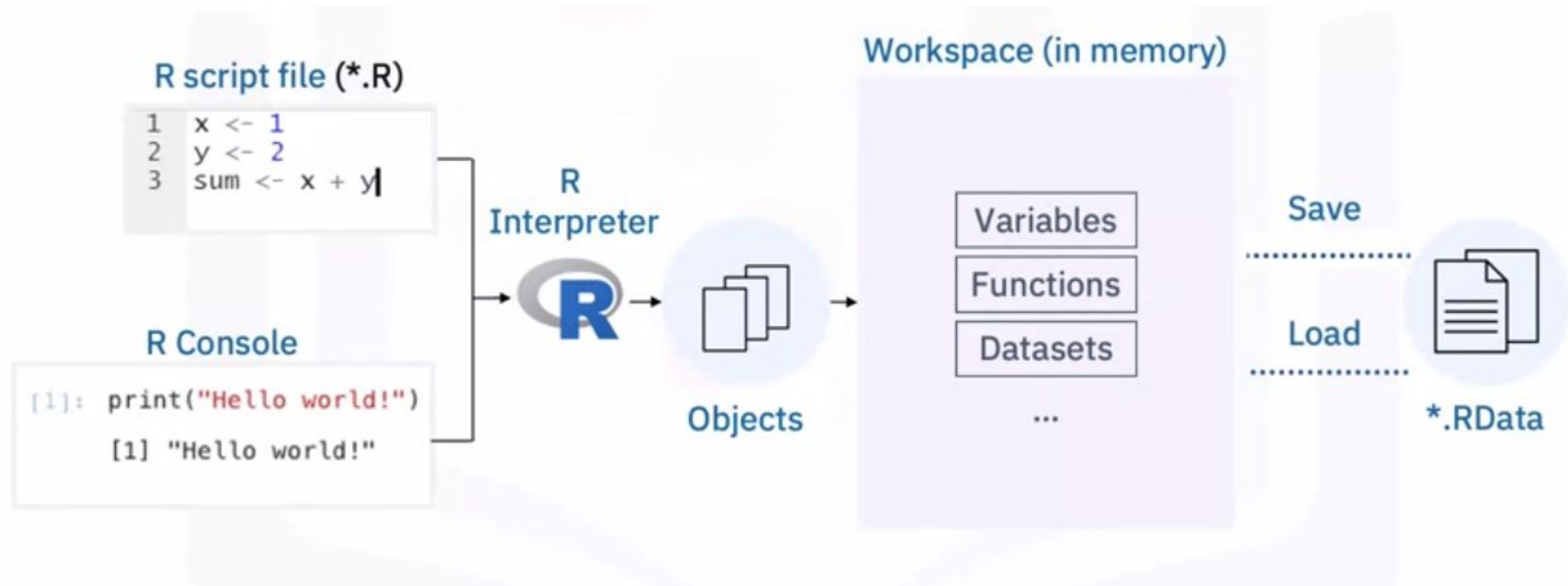
- R ilk olarak Auckland Üniversitesi İstatistik Bölümü'nden "R&R" olarak da bilinen **Robert Gentleman** ve **Ross Ihaka** tarafından yazılmıştır.
- 1997 ortalarından bu yana da 21 kişilik R çekirdek ekibi tarafından güncellenmektedir.
<https://www.r-project.org/contributors.html>
- En güncel R sürümünü <https://www.r-project.org/> adresinden indirebilir, tüm dokümanlara bu adresten ulaşabilirsiniz.

R Tarihçesi

Sizce mevcut R logosu hangi yıl çıkmıştır?



Students, write your response!



R Kodu Nasıl Çalıştırılır?



Interactive mode

The screenshot shows the RStudio interface in Interactive mode. The menu bar includes File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. The toolbar has icons for adding files, saving, running, and other functions. The source editor shows a file named 'Untitled1' with a single line of code: `1`. The console shows the R prompt `>` and the output of the command `2+5`, which is `[1] 7`. The terminal window is open, showing the R version 4.3.1 (2023-06-16) and the Beagle Scouts logo. It also displays the copyright notice for the R Foundation and the license information.

Script mode

The screenshot shows the RStudio interface in Script mode. The menu bar is the same as in the Interactive mode. The source editor shows a file named 'test.r' with the following code: `1 print("hello")`, `2 print(2+5)`, `3`, and `4`. The console shows the R prompt `>` and the output of the command `source("~/r_projects/secim/test.r")`, which is `[1] "hello"` and `[1] 7`. The terminal window is open, showing the R version 4.3.1 (2023-06-16) and the Beagle Scouts logo. It also displays the copyright notice for the R Foundation and the license information.

R Studio

The screenshot displays the R Studio interface with four main panes:

- File Editor:** Contains a script named `test.R` with the following code:

```
1= new.function <- function(a,b,c) {  
2   result <- a * b + c  
3   print(result)  
4 }  
5  
6 a<-1  
7 b<-2  
8 c<-3  
9  
10 new.function(a, b, c)
```
- Environment:** Shows the Global Environment with variables `a`, `b`, `c`, and `sum`, and a function `new.function`.

Values	Environment
a	1
b	2
c	3
sum	3
- Console:** Shows the execution of the script, resulting in the output `[1] 5`.

```
> /resources/rstudio/  
>   
> a<-1  
> b<-2  
> c<-3  
>   
> new.function(a, b, c)  
[1] 5  
> # Simple Scatterplot  
> attach(mtcars)  
> plot(wt, mpg, main="Scatterplot Example",  
+      xlab="Car Weight ", ylab="Miles Per Gallon ", pch=19)  
>
```
- Plots:** Displays a scatterplot titled "Scatterplot Example" showing Miles Per Gallon (Y-axis) versus Car Weight (X-axis). The plot includes data points from the `mtcars` dataset.

R Studio Cloud

The screenshot displays the R Studio Cloud web interface. At the top, the browser address bar shows the URL `https://posit.cloud/spaces/437099/content/6872186`. The main workspace is titled "Ders1 / Untitled Project" and includes a menu bar with options like File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, and Help. Below the menu is a toolbar with icons for saving, running, and other functions. The central editor area contains two R code chunks. The first chunk is empty, and the second chunk contains the code `12%/5` followed by a new line with `[1] 2`. Below the code, a text block explains how to add a new chunk and how to preview the HTML output. The right sidebar shows the "Environment" pane, which is currently empty, and the "Files" pane, which lists the project structure including `..`, `.Rhistory`, and `project.Rproj`. The bottom pane is a terminal window showing the R console output, including the error message `Error in 12 %7% 5 : could not find function "%7%"` and the final output `[1] 2`.

https://posit.cloud/spaces/437099/content/6872186

Ders1 / Untitled Project *Click to name your project*

File Edit Code View Plots Session Build Debug Profile Tools Help

Go to file/function Addins

Untitled1* x

Source Visual B I Normal Format Insert Table Outline

```
{r}
```

```
{r}
12%/5

[1] 2
```

No outline available

Add a new chunk by clicking the *Insert Chunk* button on the toolbar or by pressing *Ctrl+Alt+I*.

When you save the notebook, an HTML file containing the code and output will be saved alongside it (click the *Preview* button or press *Ctrl+Shift+K* to preview the HTML file).

The preview shows you a rendered HTML copy of the contents of the editor. Consequently, unlike *Knit*, *Preview* does not run any R code chunks. Instead, the output of the chunk when it was last run in the editor is displayed.

Chunk 2

R Markdown

Console Terminal Background Jobs

R 4.3.1 - /cloud/project/

Copyright (c) 2023 The R Foundation for Statistical Computing

Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

```
> 12%7%5
Error in 12 %7% 5 : could not find function "%7%"
> 12%/5
[1] 2
>
```

Environment History Connections Tutorial

Import Dataset 234 MiB

R - Global Environment

Environment is empty

Files Plots Packages Help Viewer Presentation

New Folder New Blank File Upload Delete Rename More

Cloud project

Name	Size	Modified
..		
.Rhistory	0 B	Oct 24, 2023, 5:37 PM
project.Rproj	205 B	Oct 24, 2023, 5:37 PM

The diagram illustrates the process of changing the runtime type in Google Colab. On the left, the 'Runtime' menu is open, showing options like 'Run all', 'Run before', 'Run the focused cell', 'Run selection', 'Run after', 'Interrupt execution', 'Restart runtime', 'Restart and run all', 'Disconnect and delete runtime', and 'Change runtime type'. A blue arrow points from the 'Change runtime type' option to a modal window on the right. The modal window is titled 'Change runtime type' and shows the current runtime type as 'Python 3'. Below this, there is a section for 'Hardware accelerator' with a dropdown menu showing 'Python 3' and 'T4 GPU'.

Runtime Tools Help All c

- Run all
- Run before
- Run the focused cell
- Run selection
- Run after
- Interrupt execution
- Restart runtime
- Restart and run all
- Disconnect and delete runtime
- Change runtime type

3.ipynb ☆

Change runtime type

Runtime type

Python 3 ▲

Hardware accelerator

Python 3 ?

T4 GPU

R

Jupyter/Colab Notebook

Visualizing COVID-19 New Cases

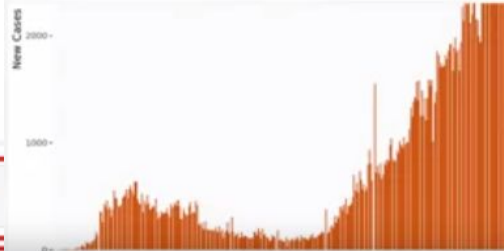
Narration

```
[3]: total_active <- cases[c("date", "numactive")]
      total_active$type <- as.factor("Active")
      total_active[is.na(total_active)] <- 0
      colnames(total_active) <- c("Date", "Value", "Type")
```

Code

653	2021-01-15	28825	Active
654	2021-01-15	197194	Recovered
655	2021-01-16	28618	Active

Data



Plots



```
[1]: x<-1
```

```
[2]: print("Hello Jupyter")
```

```
[1] "Hello Jupyter"
```

```
[3]: y <- x + 1
```

```
y
```

```
2
```

```
[ ]:
```

R sürümünü (version) nasıl öğrenirim?

```
> R.version
```

```
platform      x86_64-pc-linux-gnu
arch           x86_64
os             linux-gnu
system         x86_64, linux-gnu
status
major          4
minor          3.1
year           2023
month          06
day            16
svn rev        84548
language       R
version.string  R version 4.3.1 (2023-06-16)
nickname       Beagle Scouts
```

```
> R.version.string
```

```
[1] "R version 4.3.1 (2023-06-16)"
```

Q : Bir R paketini
yüklemek için hangi
deyim kullanılır?



Students choose an option

Pear Deck Interactive Slide
Do not remove this bar

– Paketleri (packages) Kurmak ?

```
# Bir paket kurmak için
```

```
install.packages("dplyr")
```



*belirtilen bir depodan
(varsayılan olarak **CRAN**)
paketi alıp R programına
yükler*

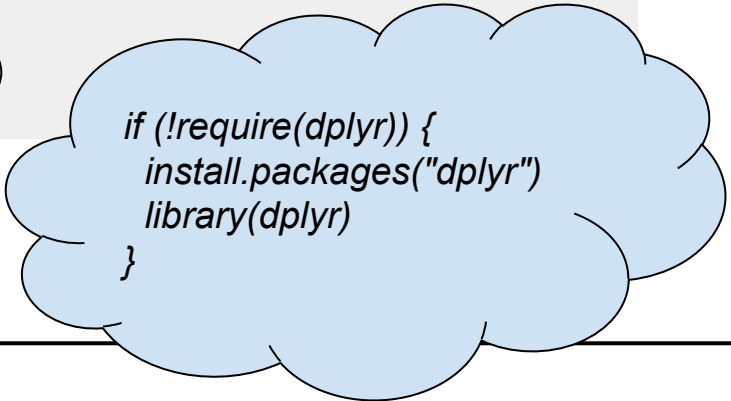
– Paketi Programınıza Yüklemek ?

```
# Bir paketi programa yüklemek için
```

```
library("dplyr")
```

```
# veya
```

```
require("dplyr")
```



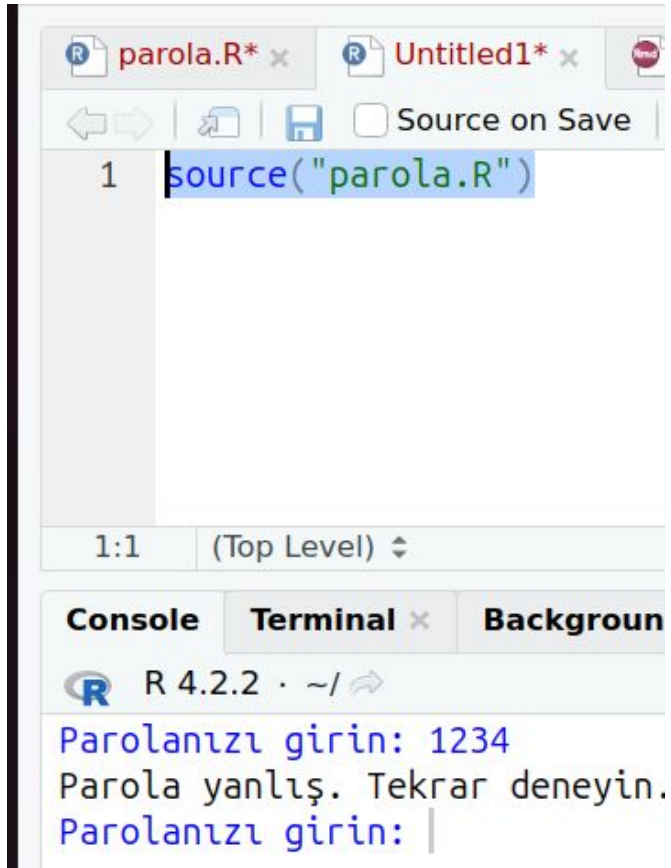
```
if (!require(dplyr)) {  
  install.packages("dplyr")  
  library(dplyr)  
}
```

Q : Bir .R programını veya kaynak kodunu çalıştırmak için hangi deyim kullanılır?



Students choose an option

Q : Bir .R programını veya kaynak kodunu çalıştırmak için hangi deyim kullanılır?



The screenshot shows the RStudio environment. The top pane displays a script editor with a file named 'parola.R*' open. The code in the editor is a single line: `source("parola.R")`. The bottom pane is split into two sections: 'Console' and 'Terminal'. The 'Console' section shows the R prompt 'R 4.2.2 · ~/ ' followed by the output of the `source()` function: 'Parolanızı girin: 1234', 'Parola yanlış. Tekrar deneyin.', and 'Parolanızı girin: |'. The 'Terminal' section is currently empty.



Students choose an option

– Paketler hakkında yardım almak

```
library(help="dplyr") # detaylı bilgi
```

```
vignette(package="dplyr")
```

```
?dplyr
```

```
help("dplyr")
```

Yardım Alma: **help()**

Bir R komutu hakkında bilgi almak için `help()` fonksiyonu veya `?` simgesi kullanılabilir;

Örneğin `print()` hakkında yardım almak için;

```
1 help(print)
```

```
1 ?print
```

Konsolundan R Sitesinde doğrudan arama yapmak için;

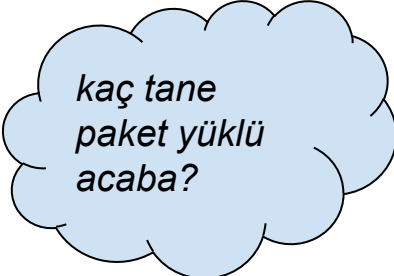
```
> RSiteSearch("random number generators")  
A search query has been submitted to https://search.r-project.org  
The results page should open in your browser shortly
```

– Yüklü paketlerin bir listesini almak ?

```
1 # Yüklü paketlerin bir listesi
2 installed_packages <- installed.packages()
3 installed_packages[,c("Package", "Version")]
```

A matrix: 164 × 2 of type chr

	Package	Version
dplyr	dplyr	1.1.3
IRdisplay	IRdisplay	1.1
IRkernel	IRkernel	1.3.2
pbdZMQ	pbdZMQ	0.3-10



*kaç tane
paket yüklü
acaba?*

– find() / example

Kullanacağınız fonksiyonun ismini biliyor
fakat hangi pakette yer aldığını bilmiyorsanız
bu noktada **find** komutu kullanılabilir;

```
find("sum")
```

```
## [1] "package:base"
```

Bir fonksiyonun
kullanımı ile ilgili
örnekleri görmek
için **example** komutu
kullanılabilir;



```
1 example(sum)
```



```
sum> ## Pass a vector to sum, and it will add the elements together.
```

```
sum> sum(1:5)
```

```
[1] 15
```

```
sum> ## Pass several numbers to sum, and it also adds the elements.
```

```
sum> sum(1, 2, 3, 4, 5)
```

```
[1] 15
```

```
sum> ## In fact, you can pass vectors into several arguments, and everything gets added.
```

```
sum> sum(1:2, 3:5)
```

```
[1] 15
```

Yüklenebilir durumdaki paketlerin bir listesini almak ?

```
1 # yüklenebilir durumdaki paketlerin listesini almak için
2 a<-available.packages()
3 print(paste(row.names(a)))
```

```
1 length(row.names(a)) # Toplam paket sayısı
```

19938

https://cran.r-project.org/web/packages/available_packages_by_name.html

kaç tane
paket
yüklenebilir
durumda?

Yeni bir paket (package) oluşturmak

```
library(devtools)  
usethis::create_package("mypackage")
```



– Paketleri güncellemek/kaldırmak

```
update.packages() # güncelle
```

```
remove.packages("mypackage") # kaldır
```

Çakışan paketlerde fonksiyon seçimi

```
4] 1 library("tidyverse")
```

```
— Attaching core tidyverse packages — tidyverse 2.0.0 —
✓ dplyr      1.1.3    ✓ readr      2.1.4
✓ forcats    1.0.0    ✓ stringr    1.5.0
✓ ggplot2     3.4.4    ✓ tibble     3.2.1
✓ lubridate   1.9.3    ✓ tidyr      1.3.0
✓ purrr       1.0.2

— Conflicts — tidyverse_conflicts() —
✗ dplyr::filter() masks stats::filter()
✗ dplyr::lag()     masks stats::lag()
```

Çakışan paketlerde fonksiyon seçimi

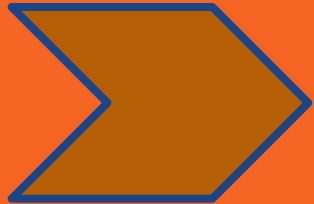


```
1 # install package
2 install.packages("conflicted")
3
4 # çakışan fonksiyonlar için paket tercihi
5 library(dplyr)
6
7 library(conflicted)
8 conflict_prefer("select", "dplyr")
9 conflict_prefer("filter", "dplyr")
```

Installing package into '/usr/local/lib/R/site-library'
(as 'lib' is unspecified)

```
[conflicted] Will prefer dplyr::select over any other package.
[conflicted] Will prefer dplyr::filter over any other package.
```

ilk Program:



> 'hello world'

→ R dilinde bir **string (text) mesajı yazdırmak** çok kolaydır.

```
> print("Hello World!")
```

```
[1] "Hello World!"
```

→ R dilinde bir **string (text) mesajı yazdırmak** çok kolaydır.

```
> msg = "hello"
```

```
> print(msg)
```

```
[1] "hello"
```

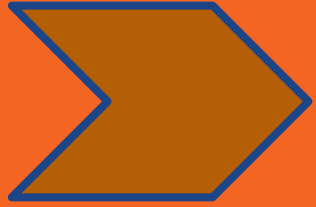
→ R dilinde bir **sayısal işlem sonucunu yazdırmak da** çok kolaydır.

```
> print(x = 22/7, digits=4)
```

```
[1] 3.143
```

```
> print(digits=6, x = 22/7)
```

```
[1] 3.14286
```



Tirnaklar (Quotes)

Tırnak içerisine alınan her şey bir string sabittir, mesajdır.

Tek tırnak



```
print('Hello World!')
```

Çift tırnak



```
print("Hello World!")
```

Hangi tırnak ile başlandı ise öyle kapatılmalıdır.



```
> print("Hello World")  
[1] "Hello World"  
> print('Hello world!')  
[1] "Hello world!"  
> print("Ali, 'gel dedi.'")  
[1] "Ali, 'gel dedi.'"
```



```
print('Hello World')  
print("Hello World")  
print(Hello World!)
```

Q : Arasındaki fark nedir?



```
print('3.14')
```

```
print(3.14)
```



Students, write your response!

Pear Deck Interactive Slide
Do not remove this bar

Q : Arasındaki fark nedir?

✓
0s

```
[1] 1 print('3.14')
```

```
[1] "3.14"
```

'character'

✓
0s

```
[2] 1 print(3.14)
```

```
[1] 3.14
```

double



typeof() / class(): Verinin tipi hakkında bilgi verir.

```
> typeof(3.14)
[1] "double"
> typeof('3.14')
[1] "character"
```

```
> class(3.14)
[1] "numeric"
> class('3.14')
[1] "character"
```

Metin ve değerini yazdırma da `print()` kullanılabilir mi?

```
1 x <- 3
2 y <- 7
3 print("x ..:", x)
4 print("y ..:", y)
```

```
[1] "x ..:"
```

```
[1] "y ..:"
```

Metin ve değerini yazdırma da `print()` yanında `paste()` kullanılabilir...

```
1 x <- 3
2 y <- 7
3 print(paste("x değeri:", x))
4 print(paste("y değeri:", y))
5
```

```
[1] "x değeri: 3"
[1] "y değeri: 7"
```

Metin ve değerini yazdırma da **print()** eşdeğerleri

```
1 x <- 3
2 y <- 7
3 cat("x ...", x, "\ny ...", y)
```

```
x ...: 3
y ...: 7
```

```
1 x <- 3
2 y <- 7
3 result <- paste("x ...", x, "\ny ...", y)
4 cat(result)
```


```
x ...: 3
y ...: 7
```

```
1 x <- 3
2 y <- 7
3 message("x ...", x, "\ny ...", y)
4
```

```
x ...:3
y ...:7
```

paste() : Metin birleştirme, karakter dizisi oluşturma ve metinleri biçimlendirme de kullanılır.

Usage



```
paste (... , sep = " ", collapse = NULL)
paste0 (... , collapse = NULL)
```

 No sep = " "

```
x <- 3
```

```
y <- 5
```

```
print(paste0("Merhaba", x+y)) # "Merhaba8"
```

```
print(paste("Merhaba", x+y)) # "Merhaba 8"
```

sep - collapse parametrelerinin farkı

Usage



```
paste (... , sep = " ", collapse = NULL)
paste0 (... , collapse = NULL)
```

 No sep = " "

```
# collapse (default) = NULL; sep(default) = " "
```

```
1 labs <- paste(c("X", "Y"), 1:10, sep="-")
2 print(labs)
```

```
[1] "X-1" "Y-2" "X-3" "Y-4" "X-5" "Y-6" "X-7" "Y-8" "X-9" "Y-10"
```

```
1 ad <- c("Bulent", "Zehra", "İsmail", "Ekrem", "Can")
2 soyad <- "Çoban"
3 ad_soyad <- paste(ad, soyad, sep="-", collapse=",")
4 print(ad_soyad)
```

```
[1] "Bulent-Çoban,Zehra-Çoban,İsmail-Çoban,Ekrem-Çoban,Can-Çoban"
```

sep - collapse parametrelerinin farkı



```
vek1 <- c("A", "B", "C")
```

```
vek2 <- c("1", "2", "3")
```

```
print(paste(vek1, vek2, sep="-"))
```

```
print(paste(vek1, vek2, collapse = " - "))
```

```
print(paste(vek1, vek2, collapse = "-", sep="*"))
```



Students, write your response!

Pear Deck Interactive Slide
Do not remove this bar

sep - collapse parametrelerinin farkı



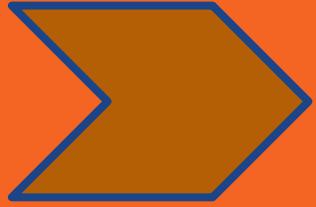
```
vek1 <- c("A", "B", "C")
```

```
vek2 <- c("1", "2", "3")
```

```
print(paste(vek1, vek2, sep="-")) # "A-1" "B-2" "C-3"
```

```
print(paste(vek1, vek2, collapse = " - ")) # "A 1 - B 2 - C 3"
```

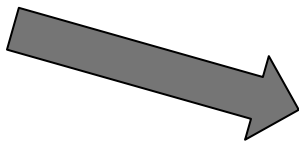
```
print(paste(vek1, vek2, collapse = "-", sep="*")) # "A*1-B*2-C*3"
```



Çıkış Karakterleri (Escape Characters)

Escape Characters

Çıktı formatını düzenleme de 'cat()',
writeLines()' ile kullanılabilir karakterler;



\'	Single quote
\"	Double quote
\\	backslash
\n	New line
\r	Carriage Return
\t	Horizontal tab
\b	Backspace

Q : Çıktısı nedir?



```
cat('Ali\tgel\'dedi ben olmaz\'\ndedim.')
```



Students, write your response!

Pear Deck Interactive Slide
Do not remove this bar

Q : Çıktısı nedir?



```
cat('Ali\tgel\'dedi ben olmaz\'\ndedim.')
```

```
Ali\tgel\'dedi ben olmaz\'  
dedim.
```

Q : Çıktısı nedir?



```
x <- "a\\b"
```

```
cat(x)
```

```
x <- "a\\nb"
```

```
cat(x)
```

```
x <- "\\a"
```

```
cat(x)
```

```
x <- "a\b"
```

```
cat(x)
```



Students, write your response!

Q : Çıktısı nedir?



```
1 x <- "a\\b"  
2 cat(x)  
3  
4 x <- "a\\nb"  
5 cat(x)  
6  
7 x <- "\\a"  
8 cat(x)  
9  
10 x <- "a\b"  
11 cat(x)
```

a\ba
b👁

ASCII/Uni Kod Tablosundaki karakterlerin gösterimi

Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
48	30	110000	60	0	96	60	1100000	140	`
49	31	110001	61	1	97	61	1100001	141	a
50	32	110010	62	2	98	62	1100010	142	b
51	33	110011	63	3	99	63	1100011	143	c
52	34	110100	64	4	100	64	1100100	144	d
53	35	110101	65	5	101	65	1100101	145	e
54	36	110110	66	6	102	66	1100110	146	f
55	37	110111	67	7	103	67	1100111	147	g
56	38	111000	70	8	104	68	1101000	150	h
57	39	111001	71	9	105	69	1101001	151	i
58	3A	111010	72	:	106	6A	1101010	152	j
59	3B	111011	73	;	107	6B	1101011	153	k
60	3C	111100	74	<	108	6C	1101100	154	l
61	3D	111101	75	=	109	6D	1101101	155	m
62	3E	111110	76	>	110	6E	1101110	156	n
63	3F	111111	77	?	111	6F	1101111	157	o
64	40	1000000	100	@	112	70	1110000	160	p
65	41	1000001	101	A	113	71	1110001	161	q
66	42	1000010	102	B	114	72	1110010	162	r
67	43	1000011	103	C	115	73	1110011	163	s
68	44	1000100	104	D	116	74	1110100	164	t
69	45	1000101	105	E	117	75	1110101	165	u
70	46	1000110	106	F	118	76	1110110	166	v
71	47	1000111	107	G	119	77	1110111	167	w
72	48	1001000	110	H	120	78	1111000	170	x
73	49	1001001	111	I	121	79	1111001	171	y
74	4A	1001010	112	J	122	7A	1111010	172	z

○ `\xhh`: 2 hex digits.

○ `\uhhhh`: 4 hex digits.

```
1 x <- "\u0048\u0069"  
2 cat(x)
```

Hi

```
1 txt = "\x48\x69"  
2 cat(txt)
```

Hi

ASCII Kod -> Karakter Dönüşümleri

intToUtf8 -> utf8ToInt

Decimal	Hexadecimal	Binary	Octal	Char	Decimal	Hexadecimal	Binary	Octal	Char
48	30	110000	60	0	96	60	1100000	140	`
49	31	110001	61	1	97	61	1100001	141	a
50	32	110010	62	2	98	62	1100010	142	b
51	33	110011	63	3	99	63	1100011	143	c
52	34	110100	64	4	100	64	1100100	144	d
53	35	110101	65	5	101	65	1100101	145	e
54	36	110110	66	6	102	66	1100110	146	f
55	37	110111	67	7	103	67	1100111	147	g
56	38	111000	70	8	104	68	1101000	150	h
57	39	111001	71	9	105	69	1101001	151	i
58	3A	111010	72	:	106	6A	1101010	152	j
59	3B	111011	73	;	107	6B	1101011	153	k
60	3C	111100	74	<	108	6C	1101100	154	l
61	3D	111101	75	=	109	6D	1101101	155	m
62	3E	111110	76	>	110	6E	1101110	156	n
63	3F	111111	77	?	111	6F	1101111	157	o
64	40	1000000	100	@	112	70	1110000	160	p
65	41	1000001	101	A	113	71	1110001	161	q
66	42	1000010	102	B	114	72	1110010	162	r
67	43	1000011	103	C	115	73	1110011	163	s
68	44	1000100	104	D	116	74	1110100	164	t
69	45	1000101	105	E	117	75	1110101	165	u
70	46	1000110	106	F	118	76	1110110	166	v
71	47	1000111	107	G	119	77	1110111	167	w
72	48	1001000	110	H	120	78	1111000	170	x
73	49	1001001	111	I	121	79	1111001	171	y
74	4A	1001010	112	J	122	7A	1111010	172	z

```
1 # ASCII kodunu kullanarak karakteri alma
2 karakter <- intToUtf8(97)
3 print(karakter)
```

```
[1] "a"
```

```
1 # Karakterin ASCII kodunu alma
2 ascii_kodu <- utf8ToInt("a")
3 print(ascii_kodu)
```

```
[1] 97
```

Comments (Yorum Satırları)

1- Tek satırlı yorumlar için;

```
# Bu bir tek satırlı yorumdur.
```

2- Satır içi (inline) yorumlar için;

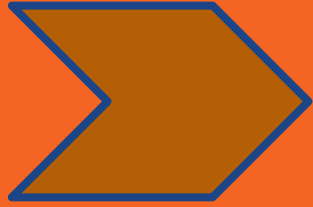
```
print('hello') # Bu bir satır içi yorumdur.
```

Comments (Yorum Satırları)

3- Çok satırlı yorumlar için;

```
# Bu bir çok satırlı yorum örneğidir.  
print ("hello")  
  
# Bu yorum, çok satırlı bir açıklama içerebilir.  
# Ayrıca kodunuzun anlaşılmasını kolaylaştırır.  
# RStudio için kısayolu:          Ctrl+Shift+C  
# Colab notebok için kısayolu:    Ctrl+/  

```



readline()
ile Veri Girişi

readline()

R’ da klavyeden veri girişi için **input()** kullanılır.

```
1 ad <- readline("Lütfen adınızı girin: ")
2 cat("Girilen ad:", ad)
```

```
Lütfen adınızı girin: Ali
Girilen ad: Ali
```

Task : Bu hata nasıl giderilir?



```
1 num1 <- readline("Sayı-1.: ")
2 num2 <- readline("Sayı-2.: ")
3 toplam <- num1 + num2
4 cat("Toplam:", toplam)
```

Sayı-1.: 3

Sayı-2.: 5

Error in num1 + num2: non-numeric argument to binary operator



Students, write your response!

Task : Bu hata nasıl giderilir?



```
1 num1 <- readline("Sayı-1.: ")
2 num2 <- readline("Sayı-2.: ")
3 toplam <- as.numeric(num1) + as.numeric(num2)
4 cat("Toplam:", toplam)
```

```
Sayı-1.: 3
Sayı-2.: 5
Toplam: 8
```

Tip dönüşümü yapmalıyız

Farklı tiplerde veri almak!

```
1 user_input <- as.numeric(readline("Lütfen bir ondalıklı sayı girin: "))
2 cat("Kullanıcının girdisi:", user_input)
```

Lütfen bir ondalıklı sayı girin: 34.5
Kullanıcının girdisi: 34.5

```
1 user_input <- as.integer(readline("Lütfen bir tam sayı girin: "))
2 cat("Kullanıcının girdisi:", user_input)
```

Lütfen bir tam sayı girin: 34.5
Kullanıcının girdisi: 34

```
1 user_input <- as.logical(readline("Lütfen bir mantıksal değer girin (TRUE veya FALSE): "))
2 cat("Kullanıcının girdisi:", user_input, "\n")
3
```

Lütfen bir mantıksal değer girin (TRUE veya FALSE): false
Kullanıcının girdisi: FALSE

Ders nasıl gidiyor?



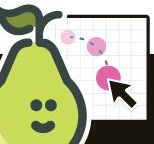
Sıkıcı



Eh işte!



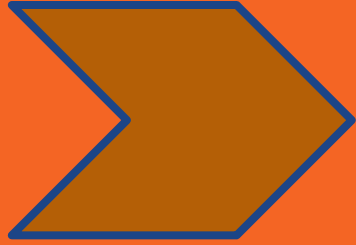
İyi, akıcı



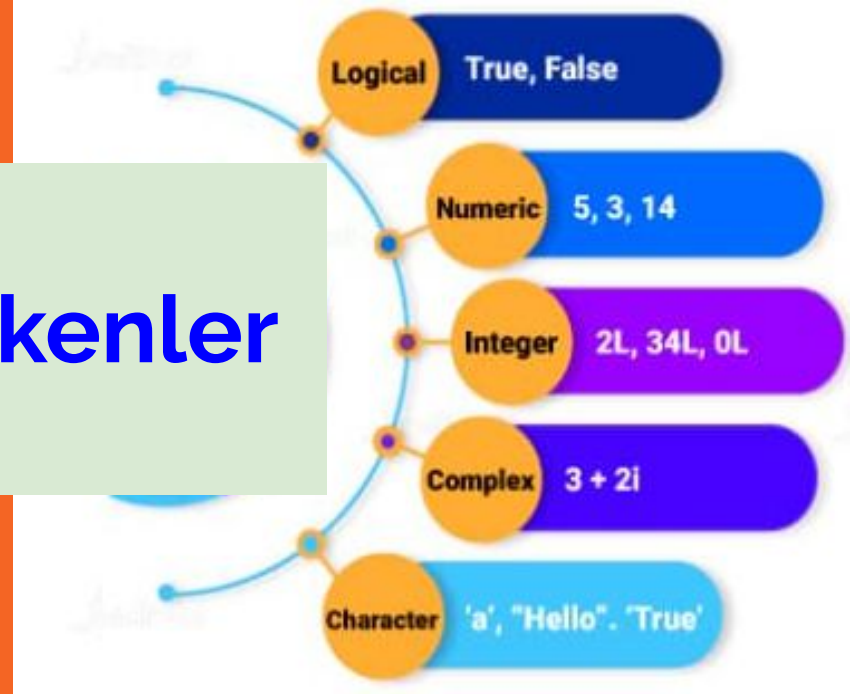
Students, drag the icon!



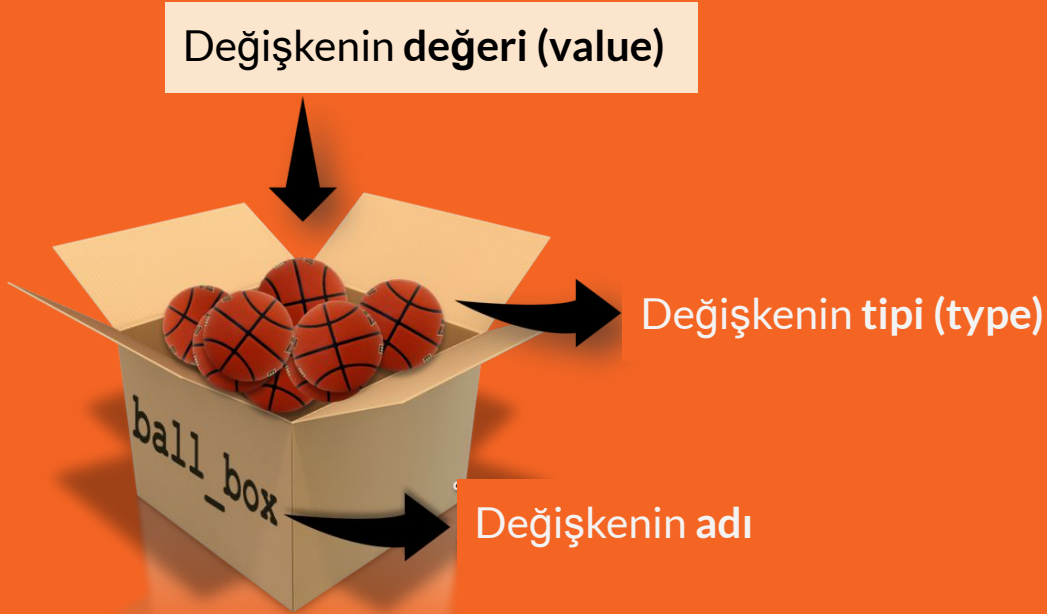
Pear Deck Interactive Slide
Do not remove this bar



Değişkenler



Değişkenler (Variables)



Matematiksel ya da sözel bir değeri bir değişkene aktarmak/atamak için '**<-**' veya '**=**' eşittir operatörü kullanılır;

```
sayi <- 3
```

```
int_num <- 5
```

```
ad = "Ali"
```

```
PI = 3.1415
```

Değişkenler (Variables)

👉 Q: Aşağıdaki programın çıktısı nedir?

```
first_number = 400
second_number = 200
second_number -> first_number
print(first_number)
print(second_number)
```



Students, write your response!

Pear Deck Interactive Slide
Do not remove this bar

Değişkenlere Değer Atama

```
1 first_number = 400
2 second_number = 200
3 second_number -> first_number
4 print(first_number)
5 print(second_number)
```

```
[1] 200
[1] 200
```

Okun
yönü
önemli...

Değişkenler (Variables)

👉 Q: Aşağıdaki programın çıktısı nedir?

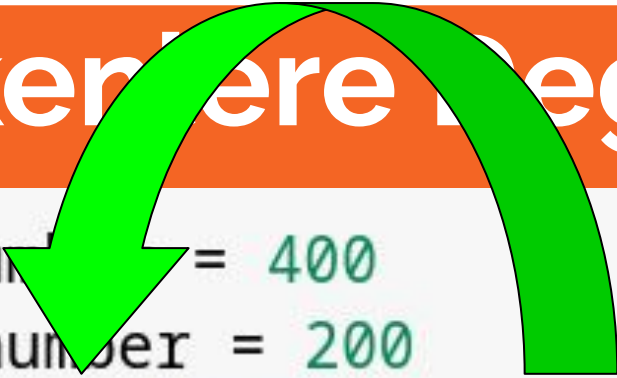
```
first_number = 400  
second_number = 200  
second_number <- first_number  
print(first_number)  
print(second_number)
```



Students, write your response!

Pear Deck Interactive Slide
Do not remove this bar

Değişkenlere Değer Atama



```
1 first_number = 400
2 second_number = 200
3 second_number <- first_number
4 print(first_number)
5 print(second_number)
```

```
[1] 400
[1] 400
```

Okun
yönü
önemli...

Değişkenlere tek satırda çoklu atama var mı?

👉 Q: Aşağıdaki programın çıktısı nedir?

```
a, b, c = 5, 3.2, "Hello"
```

```
cat (a, b, c)
```



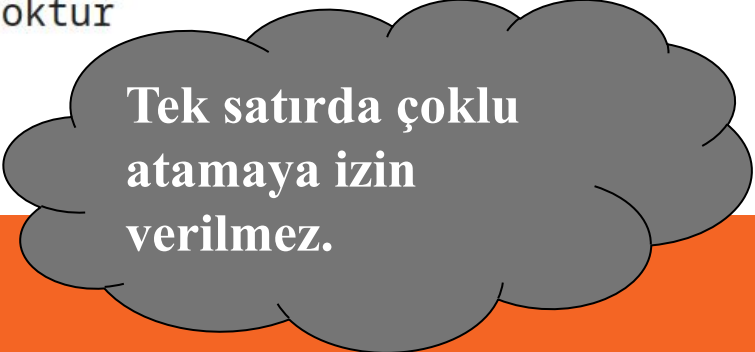
Students, write your response!

Pear Deck Interactive Slide
Do not remove this bar

Değişkenlere Değer Atama

```
# Tek satırda çoklu atama yoktur  
a, b, c = 5, 3.2, "Hello"  
cat (a, b, c)
```

```
Error in parse(text = x, srcfile = src): <text>:2:2: unexpected ' , '  
1: # Tek satırda çoklu atama yoktur  
2: a,      ^  
   Traceback:
```



Tek satırda çoklu
atamaya izin
verilmez.



















R rezerve kelimeleri (?reserved)



Değişken ismi olarak kullanılamayacak deyimler;



```
`if' `else' `repeat' `while' `function'  
`for' `in' `next' `break' `TRUE' `FALSE'  
`NULL'
```



```
`Inf' `NaN' `NA' `NA_integer_' `NA_real_'  
`NA_complex_' `NA_character_'
```



Değişken Tanımlama Kuralları



- 2me  
- Big boss  
- for  
- 1453Fatih  
- last name  
- _first  
- \$price  
- 1_2_3  
- big- boss  



data4me  



big_boss  



not  



fatih1453  

last_name  

.first  

price_dolar  

.1_2_3  

big.boss  

Değişken Tanımlama Kuralları

- İyi İsimlendirme 😊

`weekly_pay`

`bigData`

`numara`

`asalSayilar`

`avgIncomeFeb`

Kötü İsimlendirme 😞

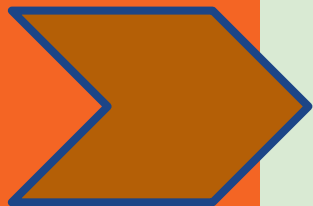
`w_p`

`b_d`

`n`

`asal_sayilar_listesi`

`# Average income of February`
`average_income_of_february`



R Style Guide (Temiz Kod)

<https://jef.works/R-style-guide/>
<https://style.tidyverse.org/>

by Hadley Wickham

R Stil Kılavuzu Önerileri

Good

getData.R

Acceptable

get_data.R

Bad

foo.bar.r

89317240934735.r

Good

x <- 23

y <- 12

Bad

x = 23; y = 12

23 -> x

Good

day_one

day_1

Bad

DayOne

dayone

Good

x[, 1]

Bad

x[,1]

x[,1]

x[, 1]

R Stil Kılavuzu Önerileri

Good

```
function(x) {}
```

Bad

```
function (x) {}
```

```
function(x){}
```

Good

```
sqrt(x^2 + y^2)
```

```
df$z
```

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

Bad

```
sqrt(x ^ 2 + y ^ 2)
```

```
df $ z
```

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

Good

```
if (a < d) {
```

```
  a <- (b + c) * d
```

```
} else {
```

```
  a <- d
```

```
}
```

Bad

```
if (a < d)
```

```
{
```

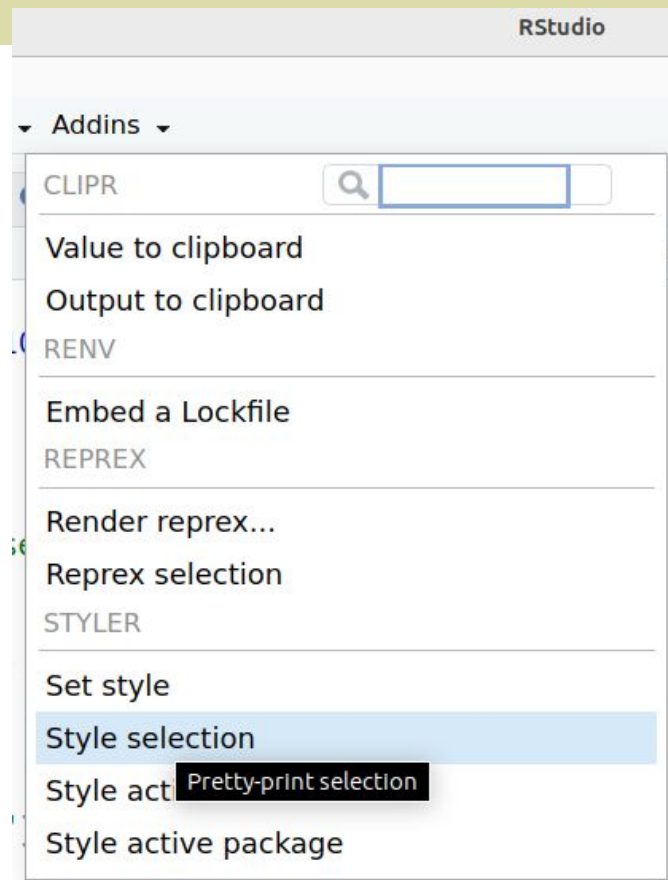
```
  a <- (b + c) * d
```

```
} else a <- d
```

R Stil Kılavuzu Önerileri

[install.packages\("styler"\)](#)

RStudio' da "**styler**" paketi ile seçilen kod satırları, otomatik olarak stil klavuzuna göre ayarlanır.



SWAP(Yer deęiřtirme)



Q: 

- İki deęiřkenin **x = '5'; y = '10'** içerięini yer deęiřtiren uygulama nasıl kodlanabilir?



Students, write your response!

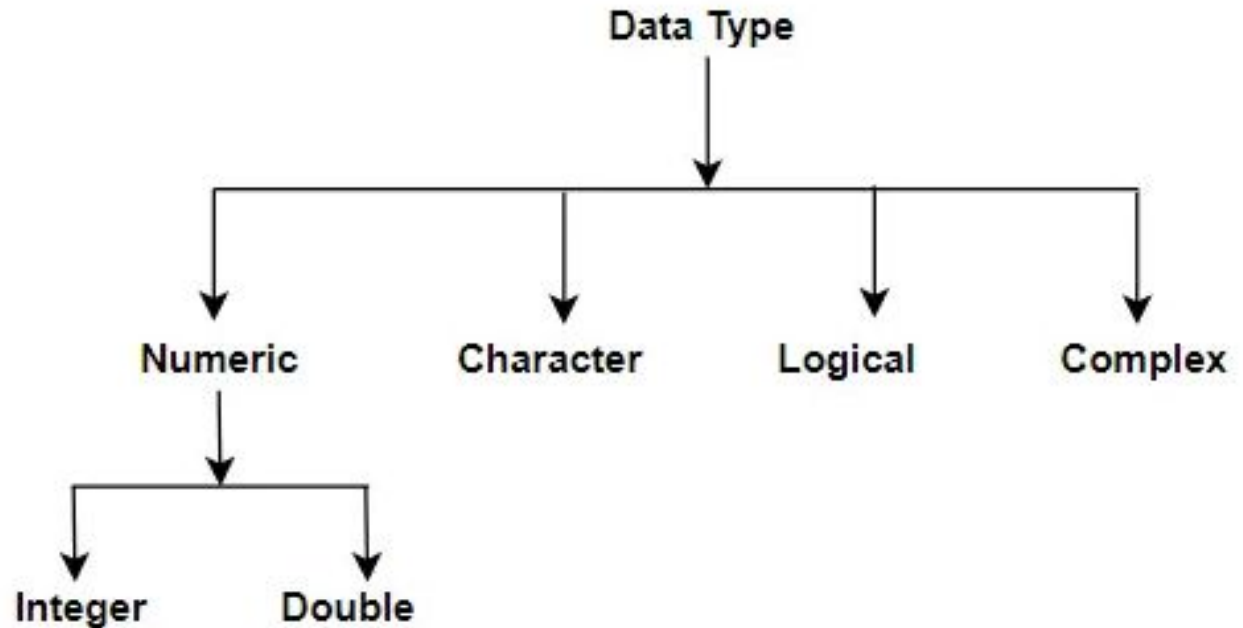
SWAP(Yer deęiřtirme)

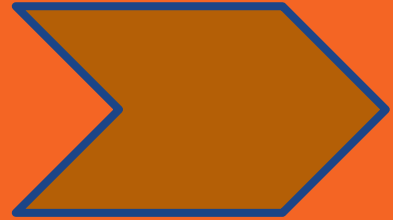
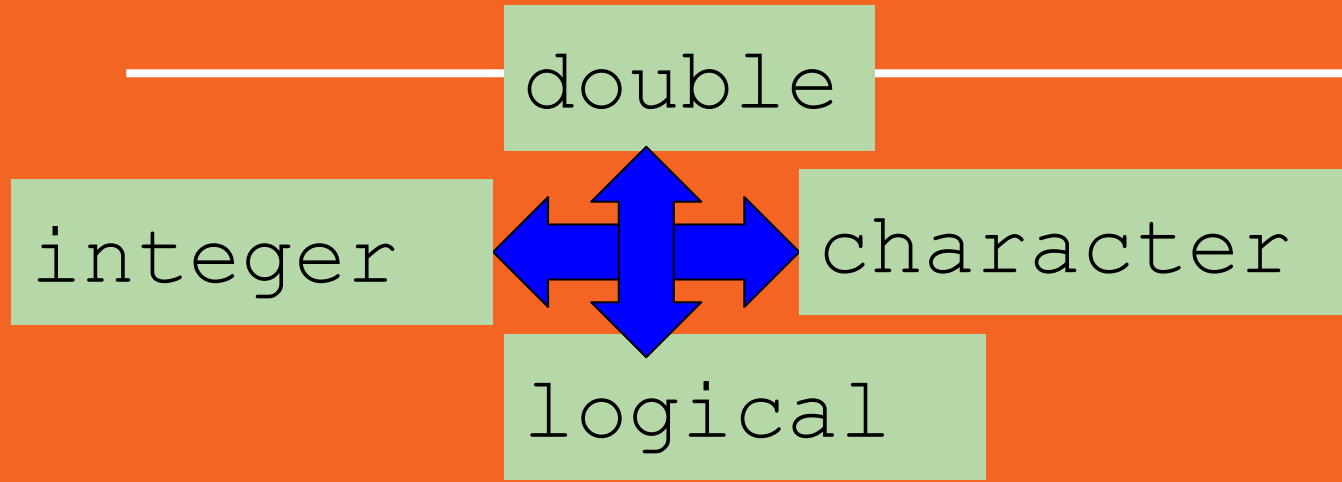


```
1 # İki deęiřkenin ierięini deęiřtirmek
2 x <- 5
3 y <- 10
4
5 # Deęiřim iin geici bir deęiřken kullanma
6 temp <- x
7 x <- y
8 y <- temp
9
10 cat("x:", x, "\ny:", y, "\n")
```

```
x: 10
y: 5
```

R Veri Tipleri





Tip Dönüşümü (Type Conversion)

Tip Dönüşüm Fonksiyonları

Tip dönüşümünde; veri, dönüştürülecek tip sarmalına alınır;

- as.integer()** : Integer tipine dönüştürür.
- as.character()** : String tipine dönüştürür.
- as.double()** : Double tipine dönüştürür.
- as.numeric()** : Double tipine dönüştürür.
- as.logical()** : Boolean(True/False) tipine dönüştürür.

Q: 🙋 Bu programın çıktısı ne olur?

```
x <- "42"
```

```
y <- as.numeric(x)
```

```
q <- as.double(y)
```

```
z <- as.character(q)
```

```
r <- as.complex(y)
```

```
print(paste(y, typeof(y), sep=":"))
```

```
print(paste(q, typeof(q), sep=":"))
```

```
print(paste(z, typeof(z), sep=":"))
```

```
print(paste(r, typeof(r), sep=":"))
```



Students, write your response!

Task: 🙌 Bu programın çıktısı ne olur?

```
x <- "42"
```

```
y <- as.numeric(x)
```

```
q <- as.double(y)
```

```
z <- as.character(q)
```

```
r <- as.complex(y)
```

```
print(paste(y, typeof(y), sep=":"))
```

```
print(paste(q, typeof(q), sep=":"))
```

```
print(paste(z, typeof(z), sep=":"))
```

```
print(paste(r, typeof(r), sep=":"))
```

Çıktısı:

```
[1] "42:double"  
[1] "42:double"  
[1] "42:character"  
[1] "42+0i:complex"
```

Q: 🙋 Bu programların çıktısı ne olur?

```
dt = "2023-11-02"
```

```
dt1 = as.Date(dt)
```

```
print(paste(dt1, typeof(dt1)))
```

```
tarih = Sys.Date() # "YYYY-MM-DD"
```

```
print(paste(tarih, class(tarih)))
```



Students, write your response!

Q:  Bu programların çıktısı ne olur?

```
dt = "2023-11-02"  
dt1 = as.Date(dt)  
print(paste(dt1, typeof(dt1)))
```

```
"2023-11-02 double"
```

```
tarih = Sys.Date() # "YYYY-MM-DD"  
print(paste(tarih, class(tarih)))
```

```
"2023-11-02 Date"
```



Students, write your response!

bool (logical) tipinde 0/0.0 hariç tüm sayısal değerler 'True' kabul edilir.

Q:  Bu programın çıktısı ne olur?

```
print (as.logical(3)) # ?  
print (as.logical(-2)) # ?  
print (as.logical(0)) # ?  
print (as.logical(NaN)) # ?  
print (as.logical(0.0)) # ?  
print (as.logical(5) + as.logical(-5)) #?
```



Students, write your response!

Q:  Bu programın çıktısı ne olur?

```
print (as.logical(3)) # ?  
print (as.logical(-2)) # ?  
print (as.logical(0)) # ?  
print (as.logical(NaN)) # ?  
print (as.logical(0.0)) # ?  
print (as.logical(5) + as.logical(-5)) #?
```

```
[1] TRUE  
[1] TRUE  
[1] FALSE  
[1] NA  
[1] FALSE  
[1] 2
```

Negatif sayıların karekökü nasıl alınır?



```
1 sqrt(-4)
```

```
Warning message in sqrt(-4):  
"NaNs produced"  
NaN
```



Students, write your response!

Negatif sayıların karekökü nasıl alınır?



```
1 sqrt(-4+0i)
```

```
0+2i
```

```
1 karekok <- sqrt(as.complex(-4))  
2 Im(karekok) #Re: Gerçel, Im: Sanal
```

```
2
```

complex formda
yazılabilir,
Karekök alma sanal
kartezyende
gerçekleşir.

OPERATÖRLER

Arithmetic Operators

+ - * / %% %/% ^

Relational Operators

< > == <= >= !=

Logical Operators

& | ! && ||

Assignment Operators

= <- -> <<- ->>

Misc. Operators

: %in% %*%

OPERATÖRLER

' $x = a + b - 5$ ' işleminde ;
' $+$, ' $-$, ' $=$ ' simgelerine **operator** yani **işleç**,
 a , b , x , 5 gibi değişken ve değerlere ise
operand, yani **işlenen** denir.

' $a + b - 5$ ' ise **expression**/**statement**
yani **deyim**/**ifade** denir.

Operatörlerin Önceliği

12.10 - 30.50 < 24.90 - 21.15
TRUE / FALSE?

1. Parantez
2. Üs
3. Bölme veya Çarpma
4. Toplama veya Çıkarma
5. Karşılaştırma
<, >, ==, !=, >=, <=
6. Mantıksal operatörler
|, &, !, ...



Students choose an option

Operatörlerin Önceliği

```
1 # sayı dizisi operatörü ":"  
2 print(1:10)
```

```
[1]  1  2  3  4  5  6  7  8  9 10
```

^

exponentiation (right to left)

- +

unary minus and plus

:

sequence operator

%any% |>

special operators (including %% and %/%)

* /

multiply, divide

+ -

(binary) add, subtract

< > <= >= == !=

ordering and comparison

!

negation

& &&

and

| ||

or

~

as in formulae

-> ->>

rightwards assignment

<- <<-

assignment (right to left)

=

assignment (right to left)

?

help (unary and binary)

Aritmetiksel Operatörler

Arithmetic Operators

Operator	Description
+	addition
-	subtraction
*	multiplication
/	division
^ or **	exponentiation
x %% y	modulus (x mod y) 5%%2 is 1
x %/% y	integer division 5%/2 is 2

```
1 2^3
```

8

```
1 2**3
```

8

```
1 5%%2 # 5 mod 2
```

1

```
1 5%/2
```

2

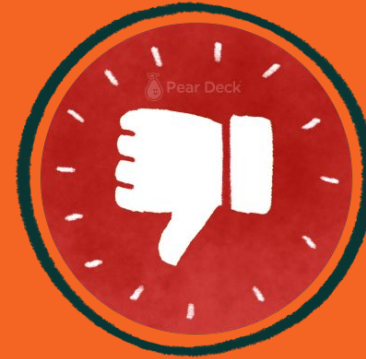
$x == (x \% y) + y * (x \% / y)$

işleminin sonucu her daim TRUE' dur.

👉 Bu ifade doğru mu ? yanlış mı?



TRUE



FALSE



Students, write your response!

Pear Deck Interactive Slide
Do not remove this bar

Aritmetiksel Operatörler



Task: 📌 Bir eşkenar üçgenin alanını hesaplayalım.

s eşkenar üçgenin bir kenarının uzunluğuna eşittir.

$$A = s^2 \frac{\sqrt{3}}{4}$$

Aritmetiksel Operatörler



Task: 📌 Bir eşkenar üçgenin alanını hesaplayalım



```
1 s = 5
2 area = s**2*(3**.5/4)
3 print(area)
```

```
[1] 10.82532
```

Mantıksal Operatörler

```
1 x <- c(1:5)
2 cat(x, sep="\t", "\n")
3 print(x >= 3) # F F T T T
```

```
1           2           3           4           5
[1] FALSE FALSE TRUE TRUE TRUE
```

Operation	Operator	Example Input	Answer
Less Than	<	4 < 10	TRUE
Less Than or Equal To	<=	4 <= 4	TRUE
Greater Than	>	11 > 12	FALSE
Greater Than or Equal To	>=	4 >= 4	TRUE
Equal To	==	3 == 2	FALSE
Not Equal To	!=	3 != 2	TRUE
Not	!	!(3==3)	FALSE
Or		(3==3) (4==7)	TRUE
And	&	(3==3) & (4==7)	FALSE

isTRUE(x) test if X is TRUE

Mantıksal Operatörler

NOT

`! TRUE` yields `FALSE`

`! FALSE` yields `TRUE`

AND

`TRUE && TRUE` yields `TRUE`

`TRUE && FALSE` yields `FALSE`

`FALSE && TRUE` yields `FALSE`

`FALSE && FALSE` yields `FALSE`

OR

`TRUE || TRUE` yields `TRUE`

`TRUE || FALSE` yields `TRUE`

`FALSE || TRUE` yields `TRUE`

`FALSE || FALSE` yields `FALSE`

&, && veya |, || farkı

&& ve **||** operatörleri, yalnızca bir elemanı değerlendirirken

& ve **|** operatörleri, birden fazla elemanı değerlendirir

```
1 x <- c(TRUE, FALSE, TRUE)
2 y <- c(FALSE, TRUE, FALSE)
3 result1 <- x & y
4 result2 <- x | y
5 print(result1)
6 print(result2)
```

```
[1] FALSE FALSE FALSE
[1] TRUE TRUE TRUE
```

```
2 result3 <- x[1] && y[1] # FALSE
3 result4 <- x[1] || y[1] # TRUE
4 print(result3)
5 print(result4)
```

```
[1] FALSE
[1] TRUE
```

%in% match operatörü : ? "%in%"

```
1 # 3,4,5 listede varmı?  
2 c(3:5) %in% c(1,2,3,5,6,7,8,9)
```

TRUE · FALSE · TRUE

```
1 10 %in% c(1,3,5,9)
```

FALSE

```
1 # listede sesli harf var mı?  
2 sstr <- c("c", "ab", "B", "bbae", "e", "c", NA, "@", "bla", "a", "%", "in")  
3 sstr[sstr %in% c("a", "e", "i", "o", "u", "ü", "ı", "ü")]
```

'e' · 'a'

Özel operatör Oluşturma: **%any%**

```
# %any% kullanımı
```

```
`%my_op%` <- function(x, y) {  
  return(x * y)  
}
```

```
# Özelleştirilmiş operatörü kullanma
```

```
a <- 5
```

```
b <- 3
```

```
sonuc <- a %my_op% b
```

```
print(sonuc) # 15
```

infix Operatörler

```
1 # infix operators
2 '+'(3, 2) # 3 + 2
3 '*'(5, 9) # 5 * 9
4 '-'(5, 9) # 5 - 9
5 '/'(9, 2) # 9 / 2
6 '^'(3, 2) # 3 ^ 2
7
8 '>`(6, 1) # 6 > 1
9 '<`(6, 1) # 6 < 1
10 '==`(6, 6) # 6 == 6
11 '!=`(6, 6) # 6 != 6
12 '>='(6, 6) # 6 >= 6
13 '<='(6, 6) # 6 <= 6
```



```
5
45
-4
4.5
9
TRUE
FALSE
TRUE
FALSE
TRUE
TRUE
```

Bitsel Operatörler

`bitwAnd()`: İkili "ve" işlemi yapar.

`bitwOr()`: İkili "veya" işlemi yapar.

`bitwXor()`: İkili "ya da" işlemi yapar.

`bitwNot()`: İkili "değili" işlemi yapar. Sayının negatifini almada kullanılır.

`bitwShiftL()`: İkili sola kaydırma işlemi yapar.

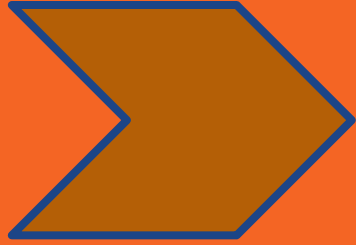
`bitwShiftR()`: İkili sağa kaydırma işlemi yapar.

```
1 x <- 5 # İkili temsil: 0101
2 y <- 3 # İkili temsil: 0011
3
4 # Bitwise AND işlemi
5 bitwAnd(x, y) # Sonuç: 0001 (1)
6
7 # Bitwise OR işlemi
8 bitwOr(x, y) # Sonuç: 0111 (7)
9
10 # Bitwise XOR işlemi
11 bitwXor(x, y) # Sonuç: 0110 (6)
12
13 # x'i (5), '-x' (-5) yapalım
14 bitwNot(x) + 1 # Sonuç: -5
```

Aritmetiksel İşlemler



```
1 # Matematiksel işlemler:
2 factorial(5)           # 5! = 120
3 abs(-5)                # |-5| = 5
4 sqrt(9)                # 9^.5 = 3
5 round(123.456, digits = 2) # 123.46
6 signif(123.456, digits=4) # 123.5
7 floor(123.45)          # 123
8 ceiling(123.45)        # 124
9 pi                     # 3.14159265358979
10 max(3,4,5,66,1,7)     # 66
11 min(3,4,5,66,1,7)     # 1
12 sum(1,2,3,4,5)        # 15
13 1000^2                # 1e+06
14 1.2e6/2e3             # 600
15 '+'(2,3)              # 5
```

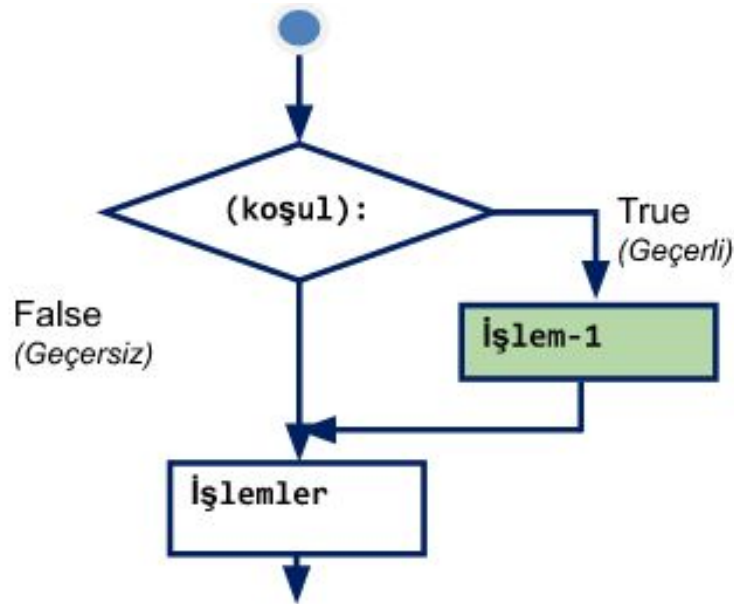


Koşullu Yapılar

if, if else, switch

if

- Tek bir koşula bağlı işlemler için;

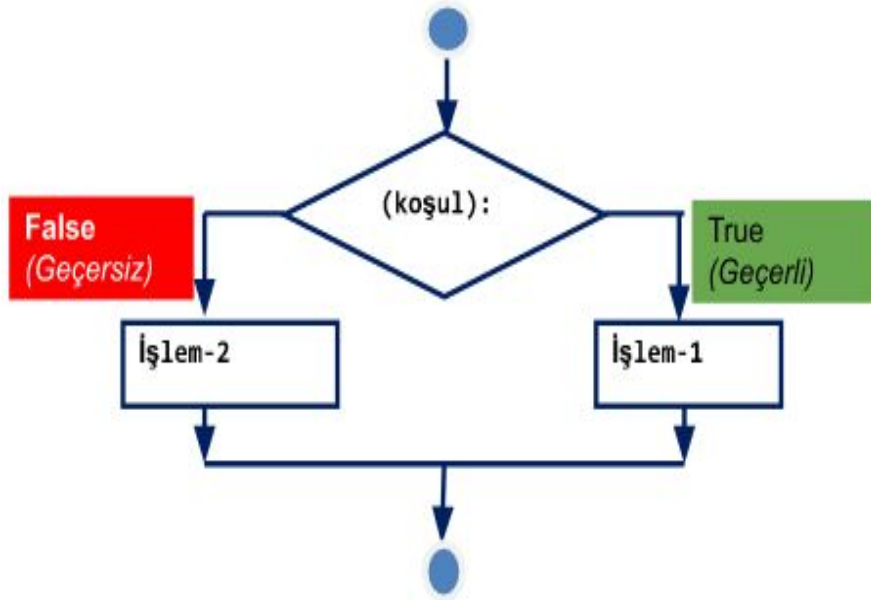


```
1 sayi <- 6.0
2 if (sayi>0) {
3   ...print("sayı pozitif")
4 }
5 if (sayi<0){
6   ...print("sayı negatif")
7 }
8 if(sayi==0){
9   ...print("sayı sıfırdır.")
10 }
```

```
[1] "sayı pozitif"
```

if ... else

- Çift koşullu işlemler için;



```
1 num = as.integer(readline ('Notunuz..:'))
2 if (num >= 50) {
3     print ("Geçtiniz")
4 } else {
5     print ("Kaldınız") }
```

```
Notunuz..:88
[1] "Geçtiniz"
```

tek satıra indirgeme

```
if (num >= 50) print ("Geçtiniz") else print ("Kaldınız")
```

Zincirleme if ... else

```
1 sayi <- 6
2 if (sayi>0) {
3 |   print("sayı pozitif")
4 } else if (sayi<0){
5 |   print("sayı negatif")
6 } else {
7 |   print("sayı sıfırdır.")
8 }
```

```
[1] "sayı pozitif"
```

ifelse

```
1 sayi <- 6
2 print(
3   ifelse (sayi>0, "sayı pozitif",
4   ifelse (sayi<0, "sayı negatif", "sayı sıfırdır.")
5   )
6 )
```

```
[1] "sayı pozitif"
```

switch : switch(expression, case1, case2, case3....)

```
1 AA = 'foo'
2 switch(AA,
3 foo={ # case 'foo' ...
4 | print('foom')
5 },
6 bar={ # case 'bar'...
7 | print('barım')
8 },
9 { # case else ...
10 | print('default')
11 }
12 )
```

[1] "foom"

```
1 indis = 2
2 switch(indis, foo={"one"}, bar={"two"})
```

'two'

—

Bu programı switch() yapısı ile nasıl kodlarız?



```
1 sayi <- 6
2 if (sayi>0) {
3 |   print("sayı pozitif")
4 } else if (sayi<0){
5 |   print("sayı negatif")
6 } else {
7 |   print("sayı sıfırdır.")
8 }
```

```
[1] "sayı pozitif"
```



Students, write your response!

Bu programı switch() yapısı ile nasıl kodlarız?



```
1 sayi = 6
2 switch(as.character(sign(sayi)),
3 "1" = {
4 |   print('sayı pozitif')
5 },
6 "-1" = {
7 |   print('sayı negatif')
8 },
9 {
10 |   print('sayı sıfır')
11 }
12 )
```

```
[1] "sayı pozitif"
```



Students, write your response!

Bugünkü dersin işleyiş hızı nasıldı?



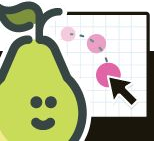
Çok yavaş



İyi



Çok hızlı

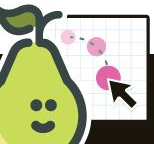


Students, drag the icon!



Pear Deck Interactive Slide
Do not remove this bar

Eğitmenin performansından memnun kaldık mı?



Students, drag the icon!



Pear Deck Interactive Slide
Do not remove this bar