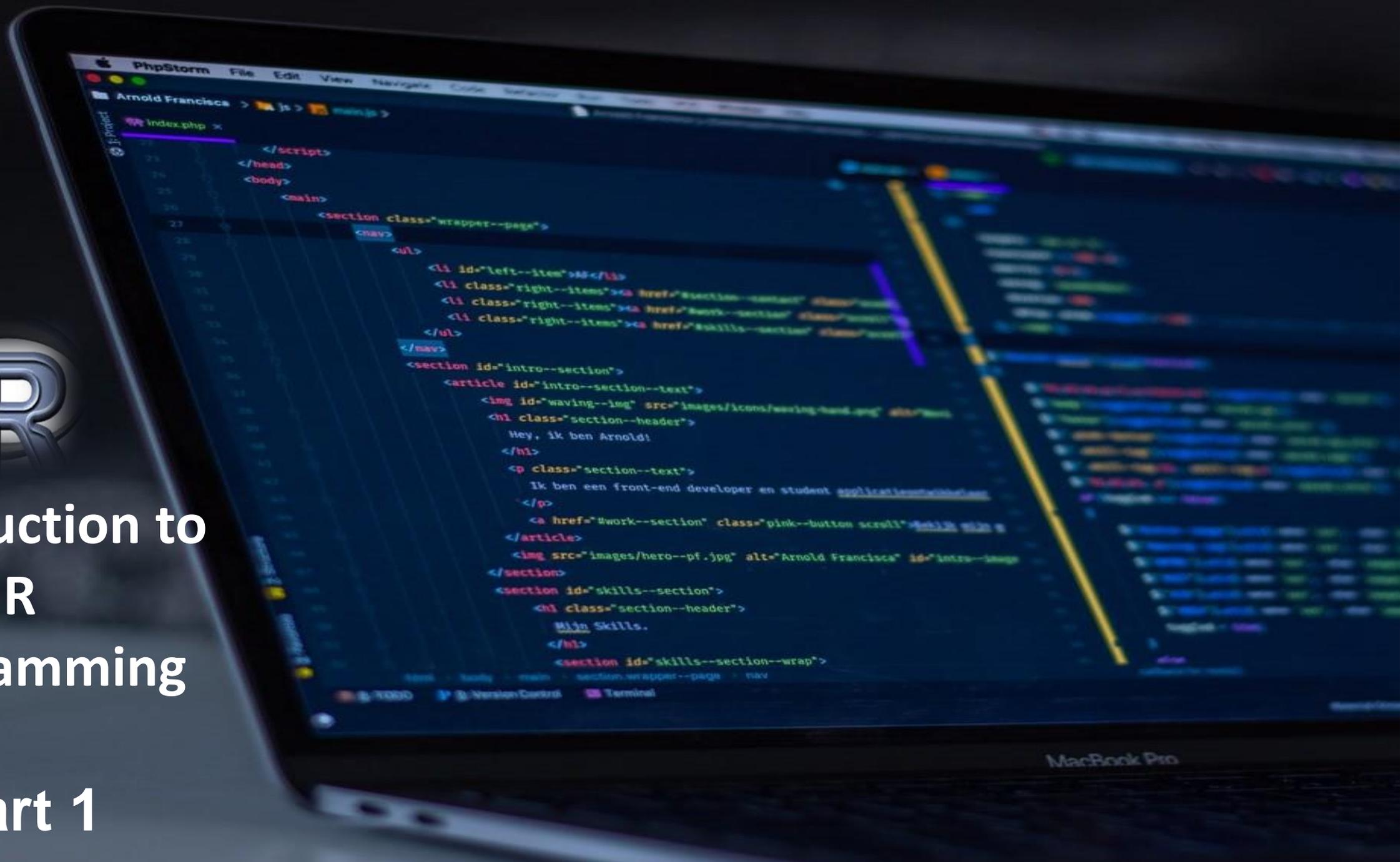




Introduction to R Programming

Part 1





چرا برنامه نویسی در محیط آماری وجود دارد؟

- روی خروجی های خود بیشتر کنترل داشته باشیم.
- زبان R رایگان است و دارای متن باز است و روی سیستم عامل مکینتاش، لینوکس و ویندوز اجرا می شود.
- زبان R دارای راهنما داخلی خوبی است.
- زبان R دارای قابلیت های قابل ملاحظه گرافیکی است.
- زبان R زبانی قوی است. یادگیری آن ساده است و دارای قابلیت های پیش ساخته آماری فراوان است.
- در این زبان به راحتی می توان توابع مورد نظر را ساخت.

زبان R چه محدودیت هایی دارد؟

- دارای امکان ایجاد Gui نیست.
- یک سیستم تجاری آن را پشتیبانی نمی کند.
- برای استفاده و برنامه نویسی باید فرامین آن را آموخت.



چرا RStudio را استفاده کنیم؟

- ویرایشگر متن کامل

یکی از اشکالات عمده نسخه R معمولی این است که از ویرایشگر متن کامل برخوردار نیست. اگر می خواهید از نحو برجسته و غیره استفاده کنید ، متأسفانه امکان پذیر نیست. داشتن یک ویرایشگر متن خوب هنگام تلاش برای یادگیری یک زبان برنامه نویسی جدید تفاوت ایجاد می کند.

- پنجره ها را گروه بندی کنید

بعضی اوقات ما نمی دانیم که فایل متنی تمام کدهای بزرگ ما به کجا رفته است. دلیل این است که R پنجره ها را گروه بندی نمی کند ، اما RStudio این کار را انجام می دهد. تمام ویندوزهای مرتبط به هم گره خورده اند ، که پیدا کردن آنها هنگام مقایسه با استفاده از نسخه R معمولی ، آسان تر می شود.

- ویژگی کامل

RStudio این گزینه را دارد که متن را برای کاربران تکمیل کند. فقط کلید Taband را فشار داده و ابزار برای رفع خطای شما سعی خواهد کرد. اما توجه داشته باشید که همیشه تصمیم درستی نمی گیرد.

- رابط کاربری بسیار بهتر است

نسخه معمولی R دارای رابط کاربری قدیمی است ، اما این برای کاربران پیشرفته که از قبل می دانند چه کاری انجام می دهند خوب است. لازم به ذکر نیست که رابط مینیمالیستی عملکرد انعطاف پذیر را تضمین می کند ، بنابراین نیازی به سرمایه گذاری روی یک کامپیوتر قدرتمند نیست.



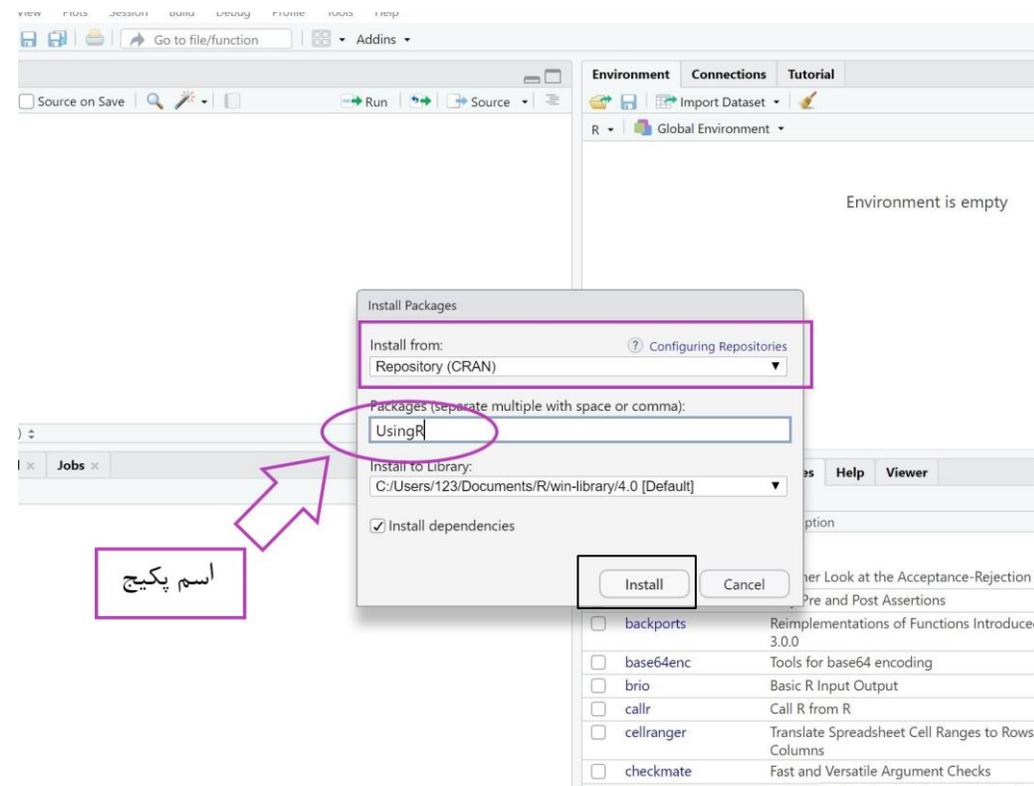
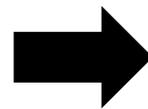
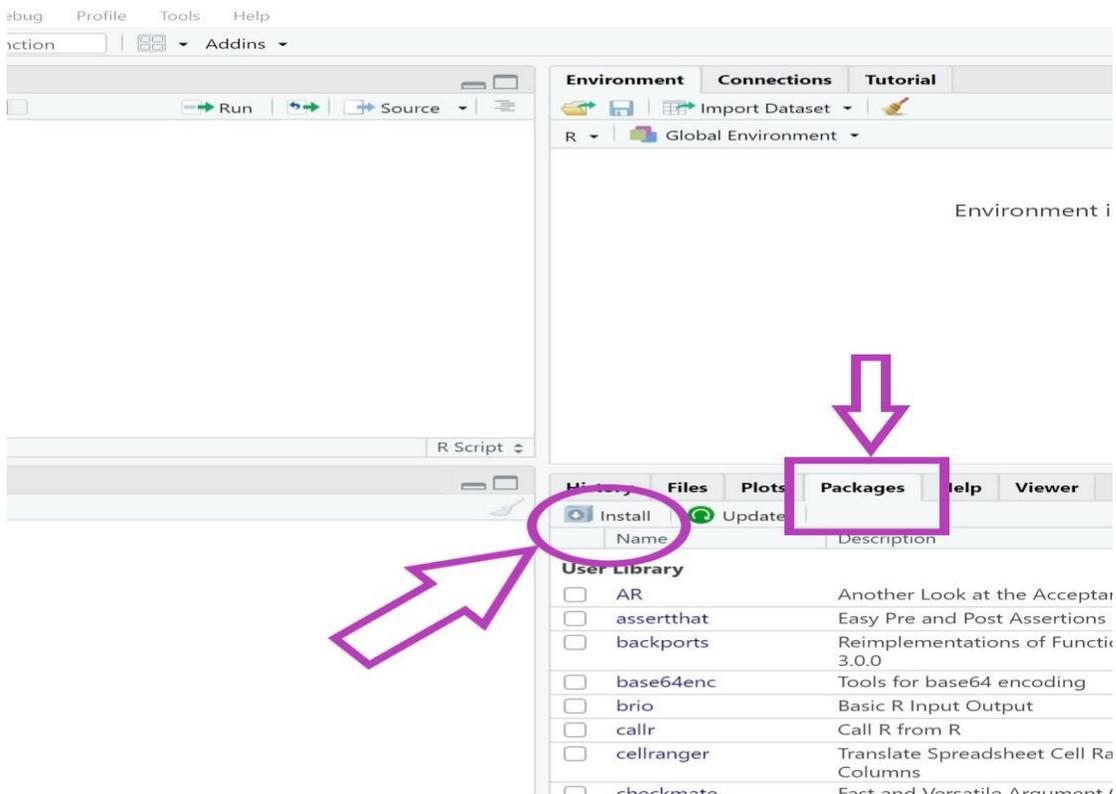
معرفی سریع RStudio

The image shows the RStudio interface with several key components highlighted and annotated in Persian:

- Code Editor (Top Left):** Labeled "1. اسکریپت (script)". It contains a "Run" button circled in red. A text box explains: "اینجا، جایی است که ما کد های مان را خواهیم نوشت دقت کنید کد ها تا زمانی که آن ها را اجرا نکنید ارزیابی نمی شوند." Another text box says: "برای ارزیابی کد ها 'Run' را بزنید (بعد کد ها و نتایج به کنسول ارسال می شوند)".
- Environment Pane (Top Right):** Labeled "4. تاریخچه". It shows "Environment is empty". A text box says: "مورد استفاده برای وارد کردن داده".
- Console (Bottom Left):** Labeled "2. کنسول (console)". It shows a prompt ">". A text box explains: "اینجا، جایی است که کد های اسکریپت شما توسط R ارزیابی و محاسبه می شوند. هم چنین میتوان از کنسول برای محاسبات سریع که نیاز به ذخیره آنها ندارید هم استفاده کنید."
- History Pane (Bottom Right):** Labeled "3. فایل ها / نمودار ها / بسته ها / کمک نرم افزار".

Other visible elements include the menu bar (File, Edit, Code, View, Plots, Session, Build, Debug, Profile, Tools, Help), the toolbar with "Source on Save" and "Go to file/function", and the "Connections" tab in the Environment pane.

نصب پکیج - راه اول



نصب پکیج - راه دوم

CRAN - Contributed Packages

cran.r-project.org/web/packages/

آدرس سایت

Contributed Packages

Available Packages

Currently, the CRAN package repository features 17194 available packages.

[Table of available packages, sorted by date of publication](#)

[Table of available packages, sorted by name](#)

Installation of Packages

Please type `help("INSTALL")` or `help("install.packages")` in R for information on how to install packages from this repository in detail.

[CRAN Task Views](#) allow you to browse packages by topic and provide tools to automatically install all packages for special topics.

Package Check Results

All packages are tested regularly on machines running [Debian GNU/Linux](#), [Fedora](#), macOS (formerly OS X), Solaris and Windows.

The results are summarized in the [check summary](#) (some [timings](#) are also available). Additional details for Windows check results are available in the [check summary](#).

Writing Your Own Packages

The manual [Writing R Extensions](#) (also contained in the R base sources) explains how to write new packages and how to contribute them to CRAN.

[usdarnass](#)

[usdata](#)

[usdm](#)

[usedist](#)

[useful](#)

[usefun](#)

[usemodels](#)

[usethis](#)

[usfertilizer](#)

[USgas](#)

[USgrid](#)

[UsingR](#)

[uskewFactors](#)

[usl](#)

[usmap](#)

[USP](#)

[UStatBookABSC](#)

[ustyc](#)

[utc](#)

[utf8](#)

USDA NASS Quick Stats API

Data on the States and Counties of the United States

Uncertainty Analysis for Species Distribution

Distance Matrix Utilities

A Collection of Handy, Useful Functions

A Collection of Useful Functions by John Fox

Boilerplate Code for 'Tidymodels' Analysis

Automate Package and Project Setup

County-Level Estimates of Fertilizer Application

The Demand for Natural Gas in the United States

The Demand and Supply for Electricity in the United States

Understanding Suppression of HIV Infection

Data Sets, Etc. for the Text "Using R for Data Science"

Model-Based Clustering via Mixtures of Experts

Analyze System Scalability with the U.S. Census

US Maps Including Alaska and Hawaii

U-Statistic Permutation Tests of Independence

A Companion Package to the Book "Using R for Data Science"

Fetch US Treasury yield curve data

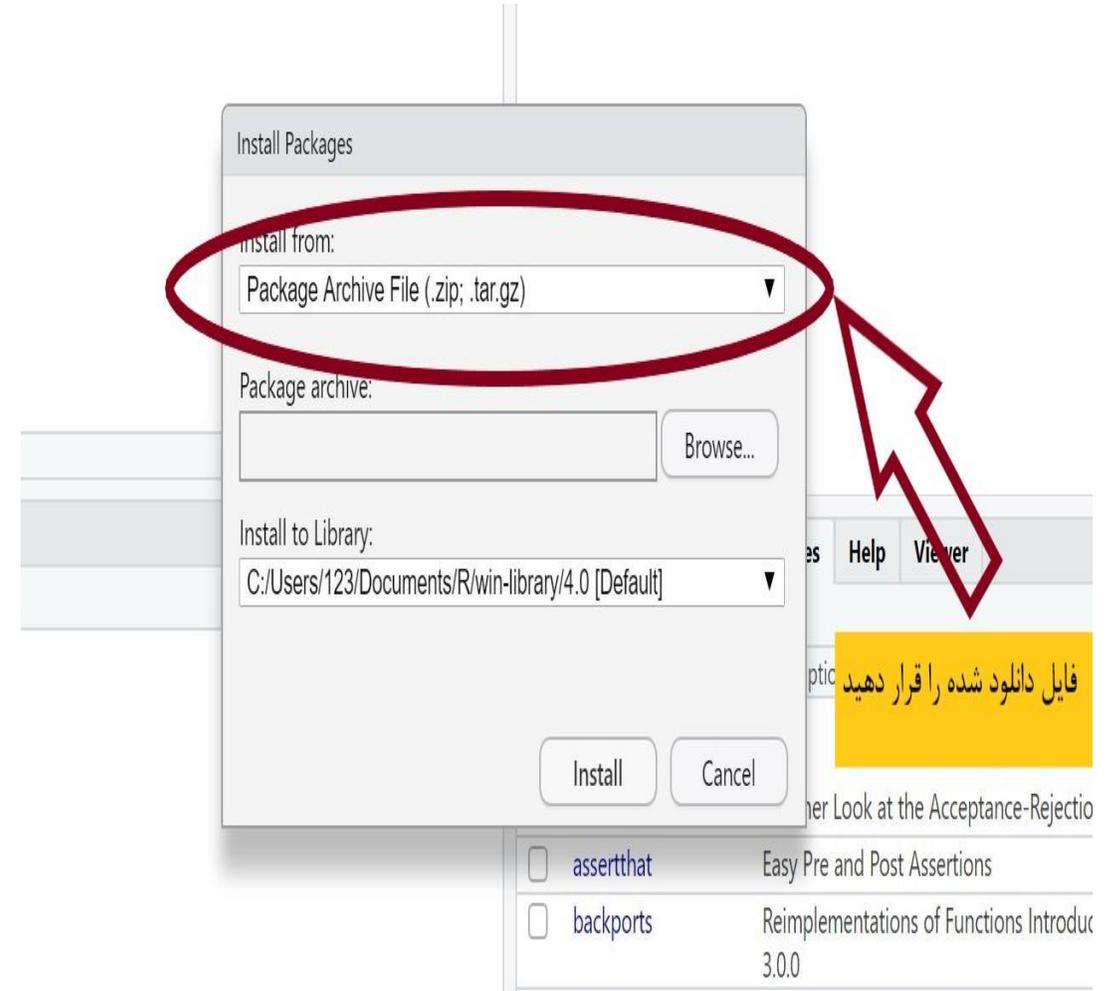
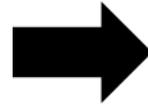
Coordinated Universal Time Transformation

Unicode Text Processing

Version: 2.0-6
Depends: R (\geq 2.15.0), [MASS](#), [HistData](#), [Hmisc](#)
Suggests: [zoo](#), [coin](#), [ggplot2](#), [vcd](#), [lubridate](#), [aplpack](#)
Published: 2018-05-01
Author: John Verzani
Maintainer: John Verzani <verzani at math.csi.cuny.edu>
License: [GPL-2](#) | [GPL-3](#) [expanded from: GPL (\geq 2)]
NeedsCompilation: no
Materials: [README](#) [NEWS](#) [ChangeLog](#)
CRAN checks: [UsingR results](#)

Downloads:

Reference manual: [UsingR.pdf](#)
Package source: [UsingR_2.0-6.tar.gz](#)
Windows binaries: r-devel: [UsingR_2.0-6.zip](#), r-release: [UsingR_2.0-6](#)
macOS binaries: r-release: [UsingR_2.0-6.tgz](#), r-oldrel: [UsingR_2.0-6](#)
Old sources: [UsingR archive](#)



نصب پکیج - راه سوم

The screenshot shows the RStudio interface with a script editor containing the following code:

```
1 ###install code
2
3 install.packages("UsingR")
```

Annotations in Persian:

- A red box highlights the code `install.packages("UsingR")`.
- A purple oval highlights the package name `"UsingR"`.
- A purple box with an arrow pointing to the package name contains the text: **اسم پکیج مورد نظر** (Package name).
- A yellow box with an arrow pointing to the package name contains the text: **نوشتن این خط کد در اسکریپت و سپس ران کردن** (Writing this code line in the script and then running it).

The console at the bottom shows the output of the command:

```
Natural language support but running in an English locale
R is a collaborative project with many contributors.
```

The right-hand side of the interface shows the Environment pane with 'Global Environment' selected and the Package pane with 'Install' and 'Update' buttons.

استفاده از R به عنوان ماشین حساب (Using R as a calculator)

R از قراردادهای استاندارد برای عملیات ریاضی و ماشین حسابی استفاده می‌کند.

در شکل دستور "۲ + ۲" را تایپ کردیم و سپس کلید enter را فشار دادیم.

برای ارسال دستور به مفسر R این پاسخ با پاسخ ۴ وپیشوند [۱] به ما برگشت داده شد. برای تفریق نیز از علامت - استفاده می‌کنیم.

```
Console ~/ |
> 2+2
[1] 4
> 2-2
[1] 0
> |
```



عملیات تقسیم در R (Divided in R)

```
Console ~/   
> (1+3+4+12)/4  
[1] 5  
> |
```

با استفاده از علامت «/» در R می‌توان اعداد را تقسیم کرد. همانند شکل روبه‌رو که با استفاده از این علامت میانگین حسابی اعداد به دست آمده است.

عملیات ضرب در R (Times in R)

```
Console ~/   
> 2*2  
[1] 4  
> |
```

با نوشتن علامت * (shift+8) می‌توان در R از ضرب استفاده کرد. همانند شکل روبه‌رو که با استفاده از این علامت ضرب اعداد به دست آمده است و حاصل برابر ۴ شده است.

عملیات توان در R (power in R)

با نوشتن علامت $^$ (shift+6) یا با استفاده از دو ستاره پشت هم (***) می‌توان توان را به دست آورد. همانند شکل روبه رو که با استفاده از این علائم حاصل هر دو روش عدد ۸ شده است.

همان‌طور که مشاهده می‌شود می‌توان از R مانند یک ماشین حساب استفاده کرد. اما این برنامه واقعاً محیطی برای محاسبات آماری و گرافیکی با قدرت بسیار فراتر از آن است.

عملیات جذر در R (square root in R)

با استفاده از تابع `sqrt` در زبان برنامه نویسی R، می‌توانیم جذریک عدد را به دست آوریم. یعنی اگر $x=9$ باشد، آنگاه برای به دست آوردن جذر x ، باید کد را به صورت روبه رو اجرا کنیم.

Console ~/ ↗

```
> 2^3  
[1] 8  
> 2**3  
[1] 8  
> |
```

Console ~/ ↗

```
> x=9  
> x  
[1] 9  
> sqrt(x)  
[1] 3  
> |
```

عملیات لگاریتم در R (Logarithm in R)

Console ~/ ↗

```
> semnan<-log(8)
> semnan
[1] 2.079442
> semnan<-log(exp(3))
> semnan
[1] 3
> |
```

با استفاده از تابع `log` در زبان برنامه نویسی R، می توانیم لگاریتم طبیعی (در مبنا ۱۰) یک عدد را محاسبه کنیم.

تابع `exp` در زبان برنامه نویسی R، برای محاسبه e^x (عدد نپر) به کار می رود.

محاسبه لگاریتم در مبناهای مختلف

Console ~/ ↗

```
> semnan <- log2(8)
> semnan
[1] 3
> semnan <- log10(8)
> semnan
[1] 0.90309
> |
```

با استفاده از تابع `log2` در زبان برنامه نویسی R، می توانیم لگاریتم بر مبنای ۲ یک عدد را محاسبه کنیم. همچنین می توان بر هر مبنایی از این دستوار استفاده کرد. همانند شکل روبه رو عمل می کنیم.

عملگرهای مقایسه‌ای

عملیات	عملگر
مساوی	==
نامساوی	!=
بزرگتر	>
کوچکتر	<
بزرگتر و مساوی	>=
کوچکتر و مساوی	<=

```
Console -/ ↗
> A = 2
> B = 3
>
> A == B
[1] FALSE
> A != B
[1] TRUE
> A > B
[1] FALSE
> A < B
[1] TRUE
> A >= B
[1] FALSE
> A <= B
[1] TRUE
> |
```

عملگرهای مقایسه‌ای در زبان برنامه‌نویسی R برای مقایسه بین دو مقدار به کار می‌روند و براساس نتیجه مقایسه، مقادیر TRUE یا FALSE را برمی‌گردانند. جدول رو به رو نحوه صحیح کد برای این عملگرها را به تصویر کشیده است.

استفاده از دستور C() برای وارد کردن داده ها (Using c () to enter data)

```
Console ~/ ↵  
> x<-c(10,4,5.6)  
> x  
[1] 10.0 4.0 5.6  
> |
```

R عملیات خود را بر روی ساختارهای نامگذاری شده‌ای از داده‌ها انجام می‌دهد. ساده‌ترین این ساختارها یک بردار عددی است که از مجموعه‌ای مرتب شده از اعداد تشکیل می‌شود. برای ایجاد یک بردار با نام x که از ۴ عدد تشکیل شده است، از دستور روبه رو استفاده می‌شود.

```
Console Terminal x Jobs x  
~/ ↵  
> x = c(74, 122, 235, 111, 292)  
> y = c(111, 211, 133, 156, 79)  
> c(x,y)  
[1] 74 122 235 111 292 111 211 133 156 79  
> |
```

دستور عکس پایین نشان دهنده این است که اگر ما دو بردار همانند x و y داشته باشیم با دستور C می‌توانیم آنها را در یک بردار بریزیم.

نسبت دهی

Console ~/ ↗

```
> stat<-c(1,2,3,4,5,6,7)
> x<-1+1
> y<-2+2
> z<-12/2
> total<-x+y+z
> total
[1] 12
>
```

باید توجه کرد که عملگر نسبت دهی ($<-$) از دو کاراکتر ($<$) و ($-$) در کنار هم تشکیل شده.

در تصویر خط اول ما یک بردار را به کلمه `stat` نسبت داده ایم یعنی هر کجا پس از اجرا این دستور از کلمه `stat` استفاده کنیم نرم افزار آن را به صورت بردار تعریف شده در نظر می گیرد. در ادامه دستور ها ما یک عملیات ریاضی را به یک حرف نسبت داده ایم یعنی هر جا نرم افزار `x` را مشاهده کرد آن را مجموع `۱` و `۱` در نظر می گیرد.

داده ها

Console ~/ ↗

```
> #numeric
> a<-1
> mode(a)
[1] "numeric"
> #character
> b<-"Hi"
> mode(b)
[1] "character"
> #logical
> c<-TRUE
> mode(c)
[1] "logical"
> #complex
> d<-1+2i
> mode(d)
[1] "complex"
> |
```

انواع mode ها : ما چهار نوع اصلی mode داریم که شامل : عددی (Numeric) و کاراکتر (character) و مختلط (complex) و منطقی (logical)

نکات کاربردی

دستور `length()`: تعداد عناصر یک شی را نشان می دهد.

دستور `mode()`: جنس داده ما را مشخص می کند.

دستور `x<- " "`: برای ایجاد یک رشته استفاده می شود.

دستور `install.packages("MASS")`: برای نصب یک بسته استفاده می شود.

دستور `library("MASS")`: اگر بسته را قبلاً نصب کرده باشید با این دستور آن را فراخوانی می کنید.

دستور `data()`: یک داده را فراخوانی می کند.

دستور `view()`: می توان مجموعه داده مورد نظر را مشاهده کرد.

دستور `head()`: نشان دهنده شش سطر اولیه.

دستور `tail()`: نشان دهنده شش سطر آخر

دستور `?survey`: در اینجا `survey` مثالی از مجموعه داده است اگر قبل کدی علامت سوال بزاریم از `help` نرم افزار می توانیم اطلاعات و راهنمایی هایی دریافت کنیم.

```
1.R* x
Source on Save
Run
Source
46
47
48 one<-c(11,12,13,14,15,16,17,18,19)
49 length(one)
50 mode(one)
51 x<-"we are learning R"
52 x
53 install.packages("MASS")
54 library("MASS")
55 data(survey)
56 view(survey)
57 head(survey)
58 tail(survey)
59
```

Console ~/ ↗

```
> Two<-c(110,112,112,115,98,88,105,113,123,101,122,99,22)
> sum(Two)
[1] 1320
> cumsum(Two)
[1] 110 222 334 449 547 635 740 853 976 1077 1199 1298 1320
> prod(Two)
[1] 5.359784e+25
> which.max(Two)
[1] 9
> which.min(Two)
[1] 13
> |
```

Console ~/ ↗

```
> A <- matrix(1:18 , nrow = 3)
> A
      [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    4    7   10   13   16
[2,]    2    5    8   11   14   17
[3,]    3    6    9   12   15   18
> dim(A)
[1] 3 6
> nrow(A)
[1] 3
> ncol(A)
[1] 6
> |
```

دستور `sum()` : برای محاسبه مجموع عناصر
دستور `cumsum()` : برای محاسبه جمع تجمعی
دستور `prod()` : برای محاسبه حاصل ضرب عناصر
دستور `which.max()` : برای محاسبه اندیس بزرگترین عنصر
دستور `which.min()` : برای محاسبه اندیس کوچکترین عنصر

وارد کردن ماتریس

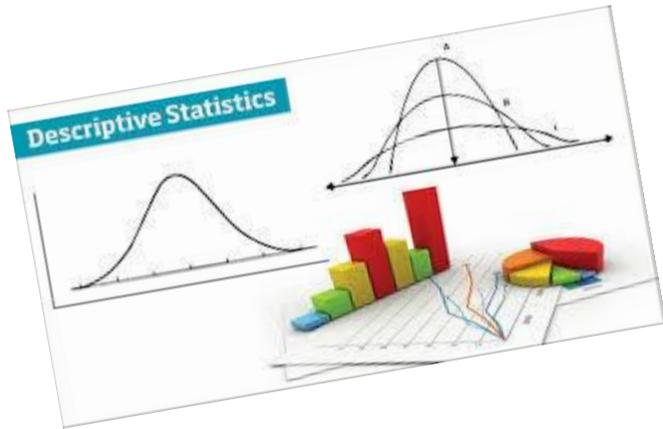
دستور مورد استفاده `matrix()` است که ابتدا مقادیر مورد نظر را وارد می کنیم (`1:12` به معنی اعداد از یک تا دوازده است) و دستور `nrow()` برای مشخص کردن تعداد سطر ها است.

دستور `dim()` : برای نشان دادن بعد ماتریس

دستور `nrow()` : تعداد سطر ها ماتریس

دستور `ncol()` : تعداد ستون های ماتریس

آمار توصیفی در R



گاهی هدف از انجام یک تحقیق، بررسی و شناخت ساختار کلی و نحوه توزیع کمیت هایی در یک جامعه آماری است. در این صورت، کمیت های مربوط به این جامعه آماری به طور کامل اندازه گیری شده یا به اصطلاح سرشماری شده است که در این حالت ابزارهای آمار توصیفی کارائی زیادی در خلاصه سازی اطلاعات و توصیف آن دارد. گاهی نیز جامعه آماری به طور کامل اندازه گیری نشده و به اصلاح نمونه گیری صورت گرفته است و برای بررسی ویژگی های کلی این نمونه تصادفی از ابزارهای موجود در آمار توصیفی استفاده می شود. در آمار توصیفی از دو ابزار برای ترسیم یا گزارش ویژگی های موجود در داده ها استفاده می شود. یکی از این ابزارها، جداول آماری است که متشکل از شاخص های آماری است که غالباً در دو دسته شاخص های پراکندگی و مرکزی قرار می گیرند. ابزار دیگر شامل نمودارهای آماری است که انواع مختلفی دارد و بنا به نوع داده ها و اهداف محقق، مورد استفاده قرار می گیرد.

Console ~/ ↵

```
> Two<-c(110,112,112,115,98,88,105,113,123,101,122,99,22)
> mean(Two)
[1] 101.5385
> median(Two)
[1] 110
> var(Two)
[1] 668.6026
> sd(Two)
[1] 25.85735
> std<-sqrt(var(Two))
> range(Two)
[1] 22 123
> max(Two)
[1] 123
> min(Two)
[1] 22
> summary(Two)
  Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  22.0   99.0   110.0   101.5   113.0   123.0
> |
```

شاخص‌های مرکزی

میانگین: برای به دست آوردن میانگین از دستور `mean()` استفاده می شود

میانه : برای محاسبه ی میانه از دستور `median()` استفاده می شود.

شاخص‌های پراکندگی

واریانس : برای محاسبه واریانس از دستور `var()` استفاده می کنیم.

انحراف معیار: برای محاسبه انحراف استاندارد از دستور `sd()` استفاده می کنیم.

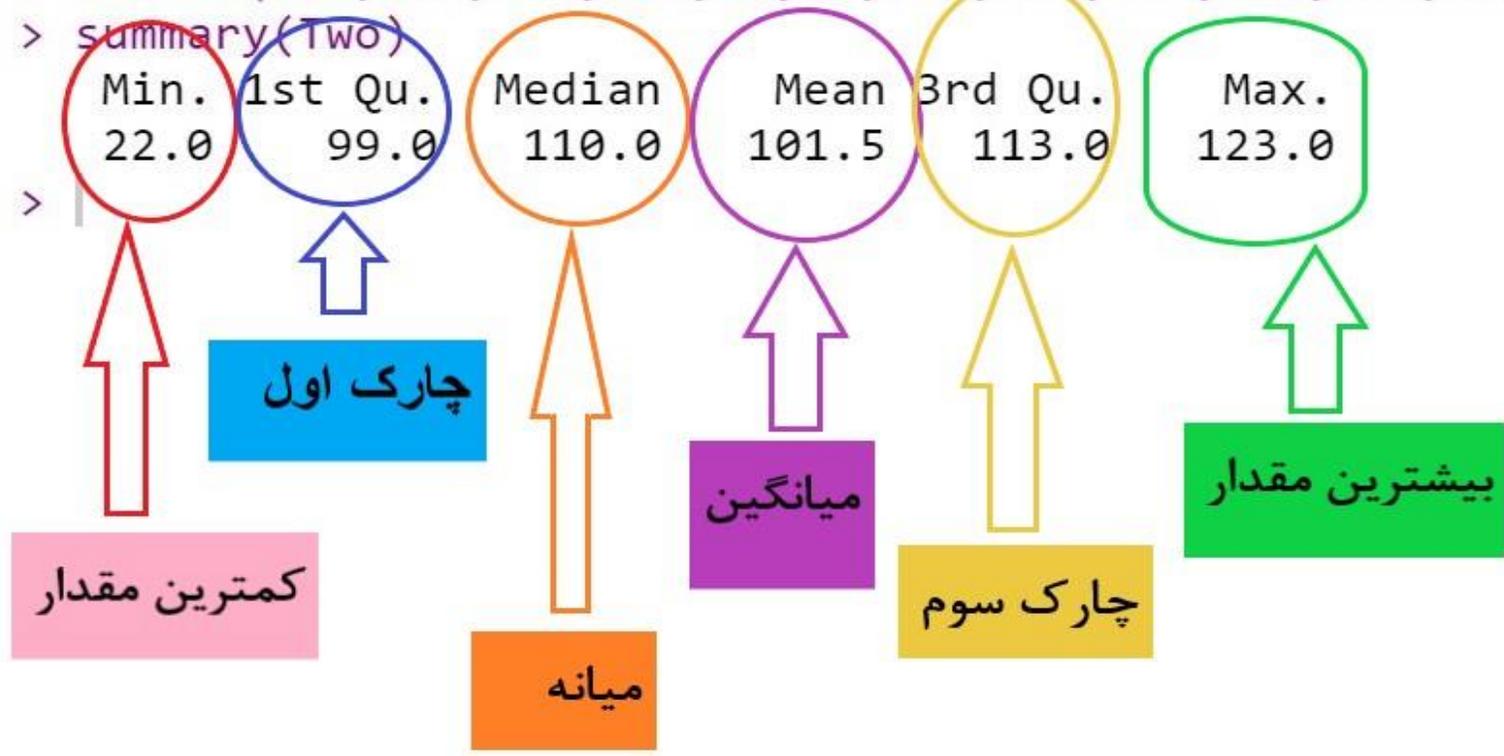
دامنه: برای محاسبه دامنه از دستور `range()` استفاده می کنیم.

ماکسیمم: برای محاسبه بیشترین مقدار از دستور `max()` استفاده می کنیم.

مینیمم: برای محاسبه کمترین مقدار از دستور `min()` استفاده می کنیم.

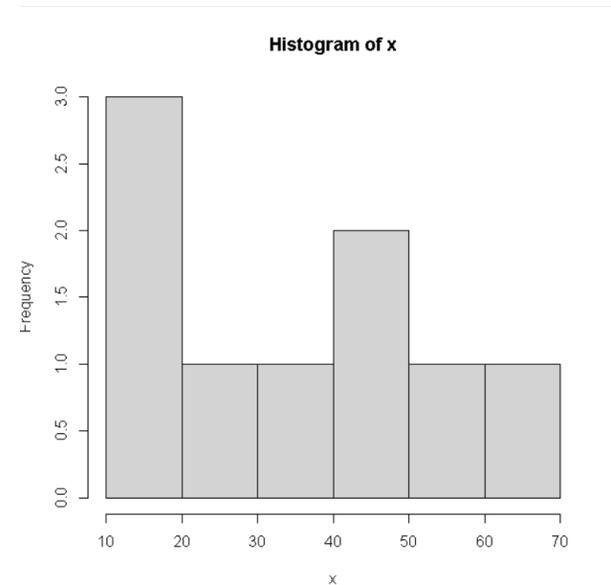
خلاصه‌ای از آمار توصیفی در R با دستور summary()

```
Console ~/   
> Two<-c(110,112,112,115,98,88,105,113,123,101,122,99,22)  
> summary(Two)  
Min. 1st Qu. Median Mean 3rd Qu. Max.  
22.0 99.0 110.0 101.5 113.0 123.0  
> |
```

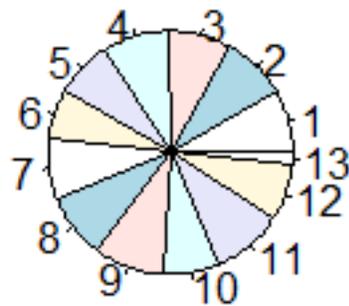


Statistic	Value	Persian Name
Min.	22.0	کمترین مقدار
1st Qu.	99.0	چارک اول
Median	110.0	میانه
Mean	101.5	میانگین
3rd Qu.	113.0	چارک سوم
Max.	123.0	بیشترین مقدار

نمودارهای آماری در R



```
0 | 238
2 | 34
4 | 356
6 | 7
```



hist(x): هیستوگرام

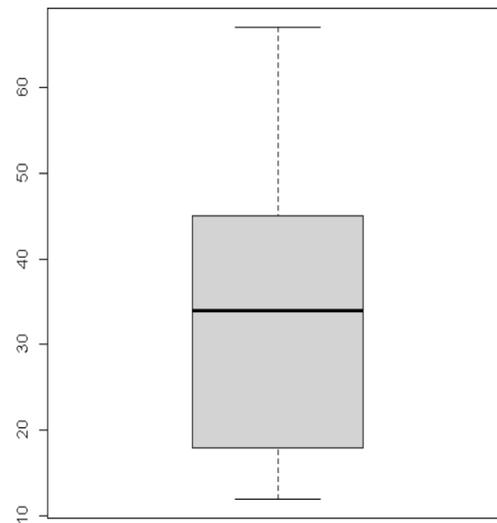
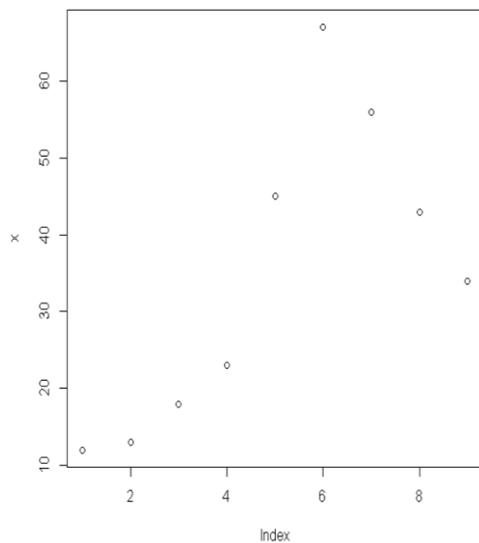
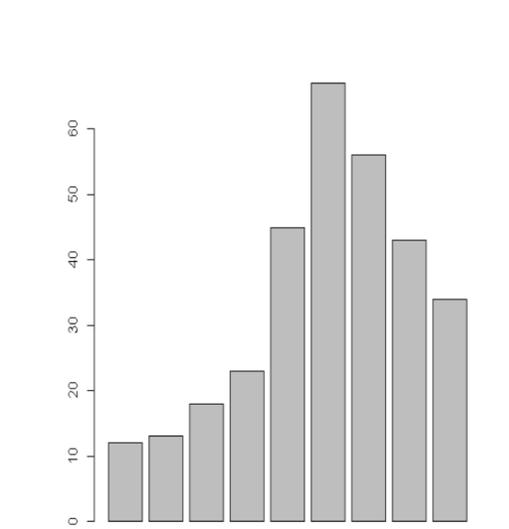
stem(x): نمودار شاخه و برگ

barplot(x): نمودار میله ای

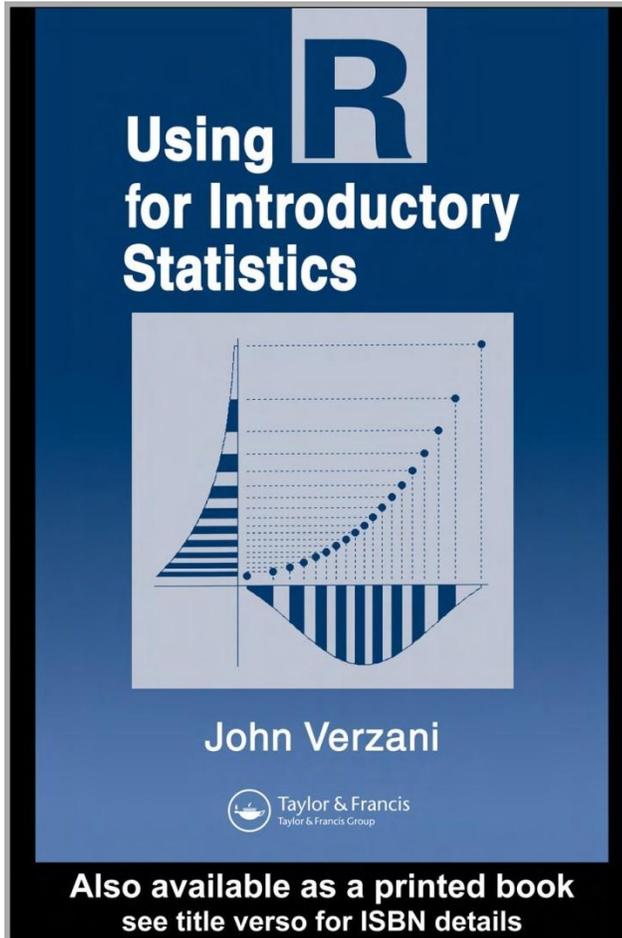
plot(x): نمودار پراکنش

boxplot(x): نمودار جعبه ای

pie(x): نمودار دایره ای



کتاب و سایت مفید برای یادگیری



آدرس جهت دانلود کتاب: <https://cran.r-project.org/doc/contrib/Verzani-SimpleR.pdf>

سایت مفید رفع اشکال: <https://stackoverflow.com/>

سایت برای ویدیو آموزشی: www.datacamp.com

سایت برای یافتن داده: <https://datasetsearch.research.google.com/>

سایت برای یافتن مقاله: <https://scholar.google.com/>

دیکشنری آنلاین تخصصی: <http://www.barsadic.com/>

جزوه فارسی مفید: جزوه آشنایی با زبان محاسبات آماری R / سیدسعید موسوی ندوشنی