

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

آموزش مقدماتی برنامه نویسی
به زبان R

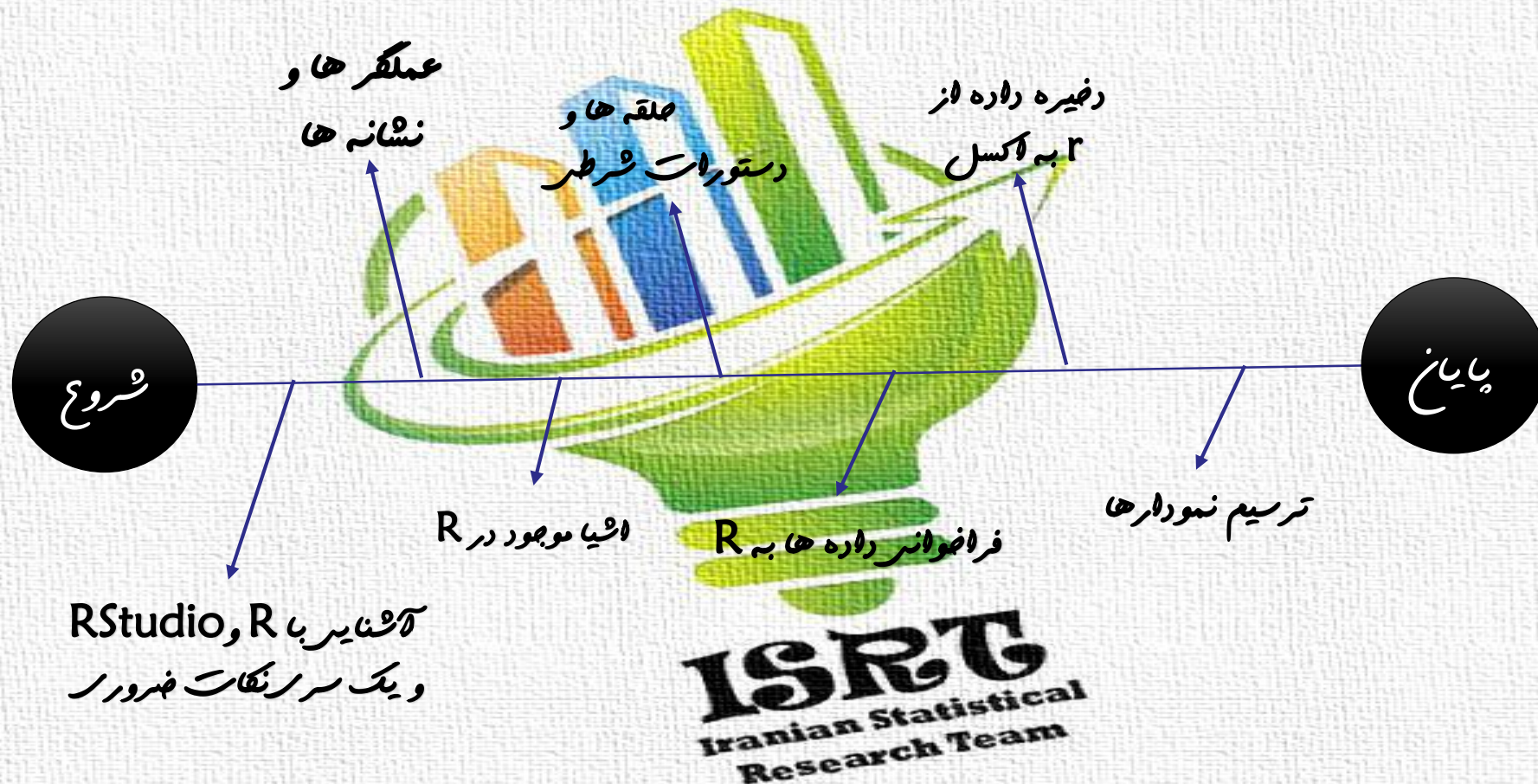
احمد وکیل بصیر

دانشجوی ارشد آمار زیستی

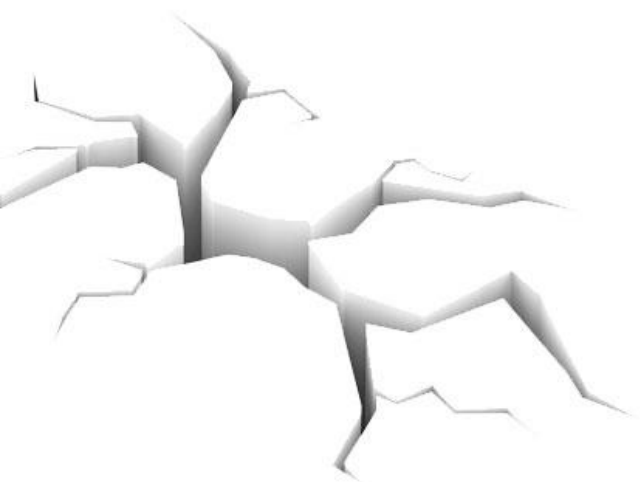
دانشگاه تربیت مدرس

5 dec 2018

Road Map



آشنایی با R و R Studio



معرفی R

- R یک زبان و محیط برنامه نویسی برای محاسبات آماری و رسم نمودارهای متفاوت می باشد که با استفاده از خط فرمان دستورات را اجرا می کند.
- زبان R بر اساس زبان های اس و اسکیم طراحی شده است.
- R یک زبان برنامه نویسی شی گراست. بدین معنی که هر متغیر قبل از اینکه دستوری روی آن اجرا شود یا فراخوانی شود باید در ساختار R تعریف شود.
- این زبان برنامه نویسی یک پروژه متن باز است، به این معنی که رشد و تکامل آن متکی به یک جامعه جهانی از توسعه دهندگان فعال است.
- از این منظر، R را می توان پدیده فرهنگی دانست که در توسعه، نگهداری و پشتیبانی آن هزاران فردی که از آن استفاده می کنند، می توانند سهمیم باشند.

تاریخچه R

پروژه R در سال ۱۹۹۵ در گروه آمار دانشگاه Auckland توسط آقایان



Ross Ihaka



Robert Gentleman

شروع شد و سپس مخاطبین زیادی پیدا کرد (نام آن برگرفته از اول حرف این دو است)

چرا R ؟

- ✓ رایگان و متن باز
- ✓ دارای راهنمای داخلی بسیارخوب
- ✓ زبانی قوی با قابلیت یادگیری ساده
- ✓ قابلیت اجرا بر روی انواع سیستم عامل ها و حتی اندروید
- ✓ قابلیت های قابل ملاحظه گرافیکی
- ✓ قابلیت ساختن پکیج و استفاده از آن در محیط **R**
- ✓ قابلیت نوشتن برنامه برای روش های جدید آماری با الگوریتم موردنظر
- ✓ وجود تعداد زیادی پکیج در زمینه های مختلف آماری
- ✓ شباهت بسیار زیاد آن با نرم افزار تجاری **s-plus**

نحوه دانلود R

The image shows a Google search for "Download R" and the corresponding CRAN website. A large green diagonal watermark reads "http://cran.r-project.org/".

Google Search Results:

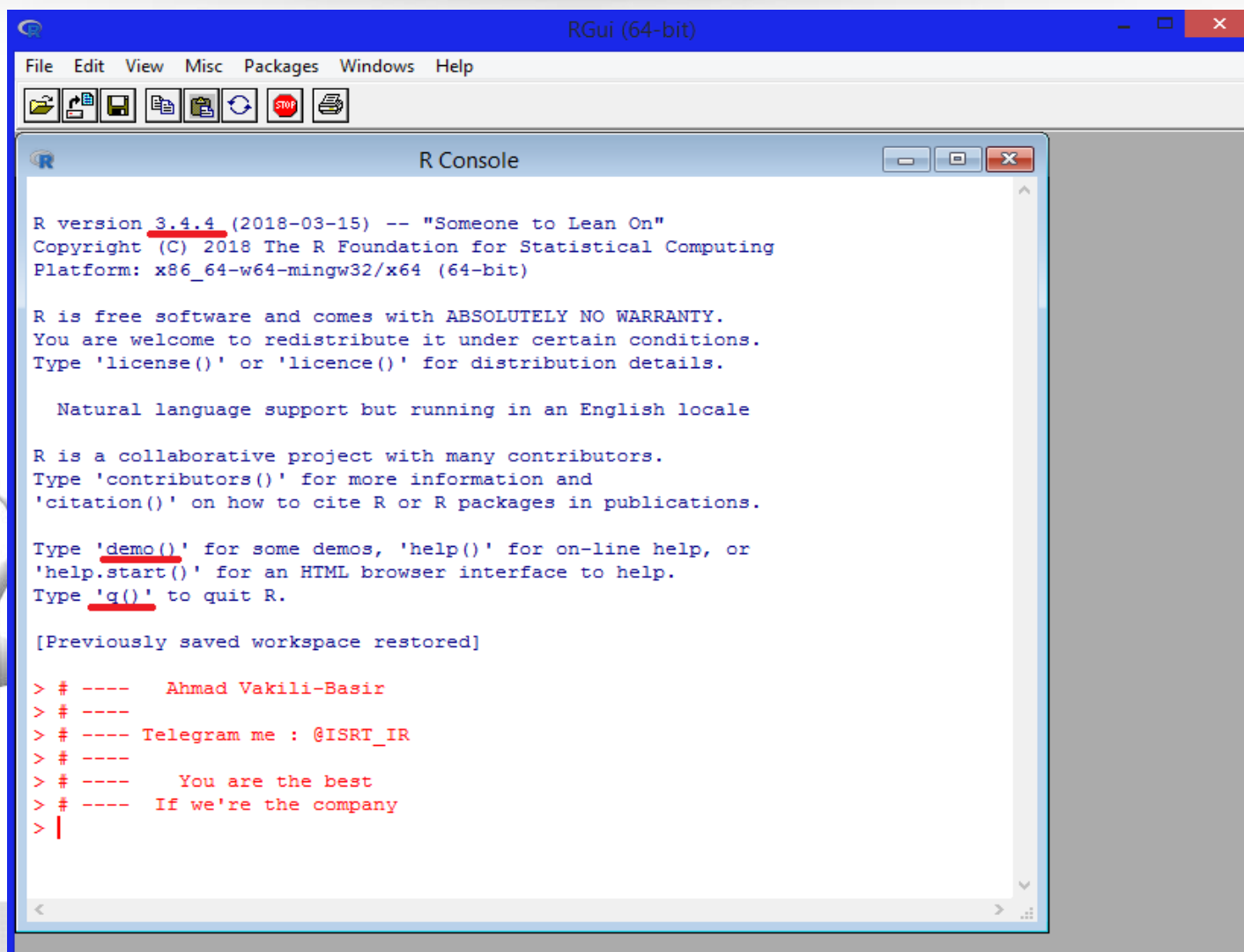
- Search query: Download R
- Results: About 6,830,000,000 results (0.50 seconds)
- Top result: Download R-3.5.1 for Windows. The R-project for statistical computing.
URL: <https://cran.r-project.org/bin/windows/base/>
- Snippet: If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the md5sum of the .exe to the ...
- Links: [R-patched snapshot build](#) · [R-devel snapshot build](#) · [Previous release](#)
- Footer: You've visited this page many times. Last visit: 10/27/18

CRAN Website (https://cran.r-project.org/bin/windows/base/):

- Page title: R-3.5.1 for Windows (32/64 bit)
- Download link: [Download R 3.5.1 for Windows](#) (62 megabytes, 32/64 bit)
- Other links: [Installation and other instructions](#), [New features in this version](#)

If you want to double-check that the package you have downloaded matches the package distributed by CRAN, you can compare the [md5sum](#) of the .exe to the [fingerprint](#) on the master server. You will need a version of md5sum for windows: both [graphical](#) and [command line versions](#) are available.

با دوبار کلیک بر روی آیکون نرم افزار، نرم افزار اجرا می شود :



The screenshot shows the RGui (64-bit) window. The title bar is blue with the R logo and the text 'RGui (64-bit)'. The menu bar includes File, Edit, View, Misc, Packages, Windows, and Help. Below the menu bar is a toolbar with icons for file operations and execution. The main window is the R Console, which displays the following text:

```
R version 3.4.4 (2018-03-15) -- "Someone to Lean On"
Copyright (C) 2018 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (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.

Natural language support but running in an English locale

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.

[Previously saved workspace restored]

> # ---- Ahmad Vakili-Basir
> # ----
> # ---- Telegram me : @ISRT_IR
> # ----
> # ---- You are the best
> # ---- If we're the company
> |
```


نکات ضروری در مورد R

✓ سه نوع نماد تخصیص یک نام به شیء در R وجود دارد:

«<-»، «=» و «>-»

✓ اولین نام یک شیء حتما باید با حروف A-Z یا a-z شروع گردد.

✓ سایر کاراکترهای نام می تواند شامل اعداد (۰-۹)، نقطه (.) و زیرخط (_) نیز باشد.

✓ زبان R بین حروف بزرگ و کوچک تفاوت قائل می شود.

✓ اشیاء شامل متغیرها، توابع، نتایج و ... در R با یک نام ذخیره می شوند.

✓ R بردار بیس است بدین معنی که یک متغیر تک عضوی به صورت یک بردار تک

عضو در نظر گرفته می شود.

آشنایی با R Studio

شرکتی است که تهیه نرم افزار، آموزش و خدماتی را برای محیط محاسباتی R تخصیص داده است.

برنامه ساخته شده توسط این شرکت، محیطی توسعه یافته و یکپارچه برای نرم افزار آر به حساب می آید.

از قابلیت های این محیط می توان به:

۱- ابزارهای سودمند قدرتمند (متمایز کردن، کامل شدن دستورات، دندانه های هوشمند...)

۲- محیط برنامه نویسی ساخته شده برای نرم افزار آر (جستجو در فضای کاری، نمایشگر داده ها خروجی پی دی اف و ...)

۳- سازگار (قابلیت کار کردن با هر نسخه ای از نرم افزار آر (نسخه های ۲.۱۱.۱ به بالا)، رایگان و منبع آزاد و...)

R Studio نحوه داندود و نصب

← → ↺ <https://www.rstudio.com/products/rstudio/download/#download> ☆ 🔒 🌐 📁 📄 📷

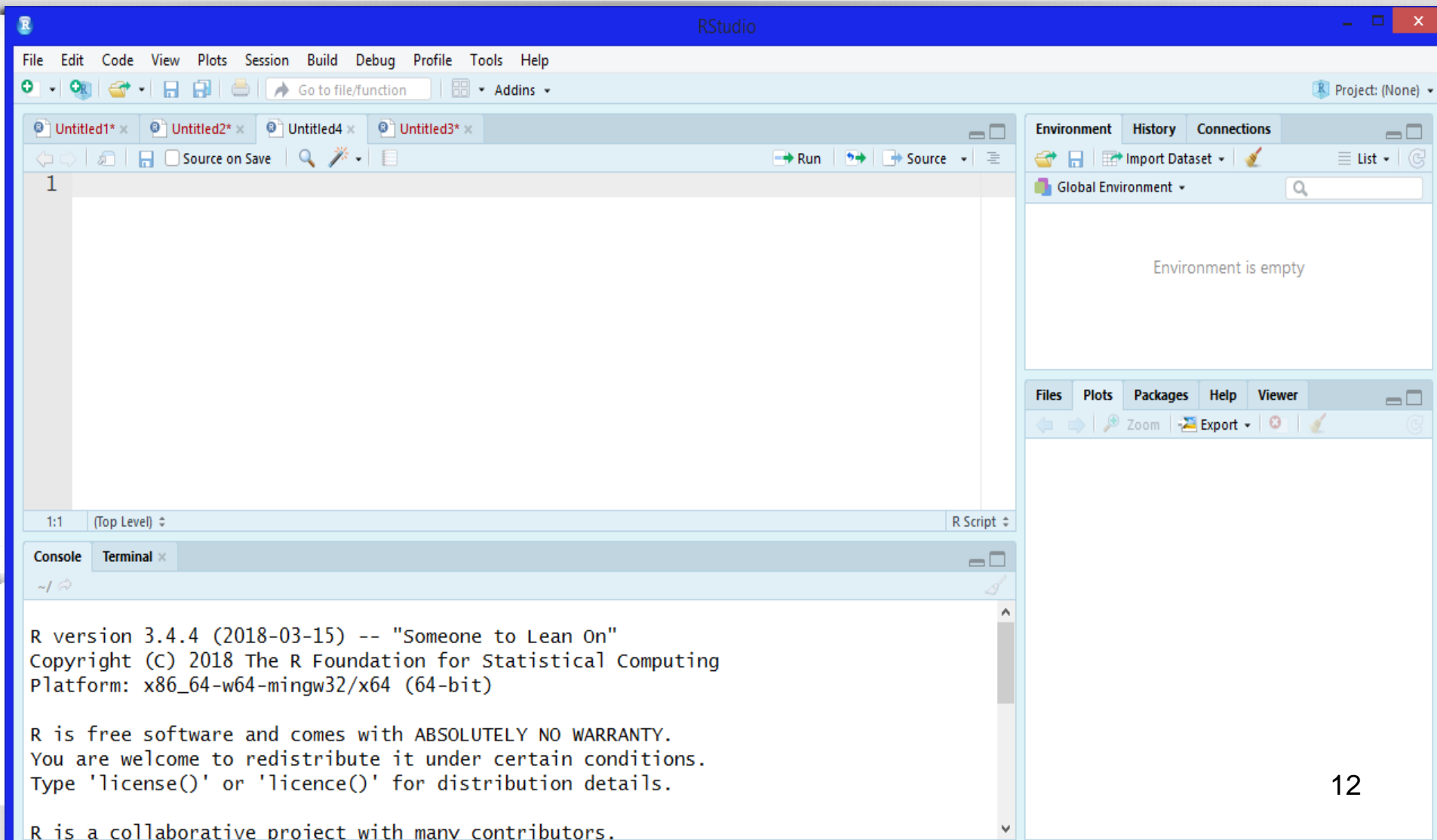


| Installers | Size | Date | MD5 |
|--|---------|------------|----------------------------------|
| RStudio 1.1.463 - Windows Vista/7/8/10 | 85.8 MB | 2018-10-29 | 58b3d796d8cf96fb8580c62f46ab64d4 |
| RStudio 1.1.463 - Mac OS X 10.6+ (64-bit) | 74.5 MB | 2018-10-29 | a79032ba4d7daaa86a8da01948278d94 |
| RStudio 1.1.463 - Ubuntu 12.04-15.10/Debian 8 (32-bit) | 89.3 MB | 2018-10-29 | 8a6755fa9fae2bafce289df3358aaf63 |
| RStudio 1.1.463 - Ubuntu 12.04-15.10/Debian 8 (64-bit) | 97.4 MB | 2018-10-29 | bc50d6bd34926c1cc3ae4a209d67d649 |
| RStudio 1.1.463 - Ubuntu 16.04+/Debian 9+ (64-bit) | 65 MB | 2018-10-29 | cf659db18619cc78d1592fefaa7c753 |
| RStudio 1.1.463 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit) | 88.1 MB | 2018-10-29 | 742f0bad60dfeaa3281576e14ad6699e |
| RStudio 1.1.463 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit) | 90.6 MB | 2018-10-29 | c7303067a0ca99deea7e427b856952d1 |

Zip/Tarballs

| Zip/tar archives | Size | Date | MD5 |
|--|----------|------------|----------------------------------|
| RStudio 1.1.463 - Windows Vista/7/8/10 | 122.9 MB | 2018-10-29 | 1eb1d7758bd4bf4bb68d4a7c3fe8d894 |
| RStudio 1.1.463 - Ubuntu 12.04-15.10/Debian 8 (32-bit) | 90 MB | 2018-10-29 | ef9242b621d36c30de9d86b808840b41 |
| RStudio 1.1.463 - Ubuntu 12.04-15.10/Debian 8 (64-bit) | 98.3 MB | 2018-10-29 | 5944b3dd118cfba46f2a6c484d768324 |
| RStudio 1.1.463 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (32-bit) | 88.8 MB | 2018-10-29 | 6bee446eeb4cc0e32967845e1f7ffe2b |
| RStudio 1.1.463 - Fedora 19+/RedHat 7+/openSUSE 13.1+ (64-bit) | 91.4 MB | 2018-10-29 | 3f63725769b7b976fedb754eb2a19108 |

محیط R Studio



نکاتی در مورد R Studio

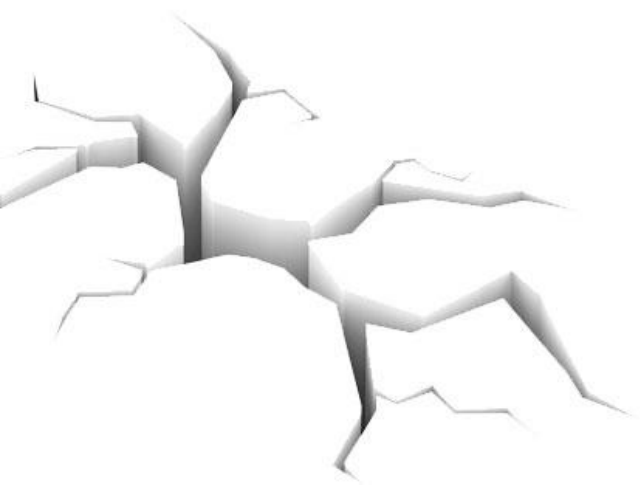
- ✓ RStudio یک ویرایشگر می باشد بدین معنی که حتما حتما باید در سیستم شما R نصب شده باشد تا بتوانید از RStudio استفاده نمایید.
- ✓ RStudio به صورت خودکار به آخرین ورژنی R ی که رو سیستمتان نصب میکنید وصل می شود. ولی کاربر می تواند آن را به دلخواه تغییر دهد.
- ✓ می توان رنگ و اندازه فونت ها را به راحتی و به دلخواه در RStudio تغییر داد.
- ✓ با تایپ سه حرف اول هر دوستور، Rstudio دستوراتی را که با این سه حرف شروع می شوند پیشنهاد و لیست می کند.
- ✓ با زدن کلید Tab آرگومان های هر دستور ظاهر می شوند.

نصب بسته یا پکیج در R

برای نصب بسته ها در R به دو روش می توان عمل کرد:

✓ روش آفلاین

✓ روش آنلاین

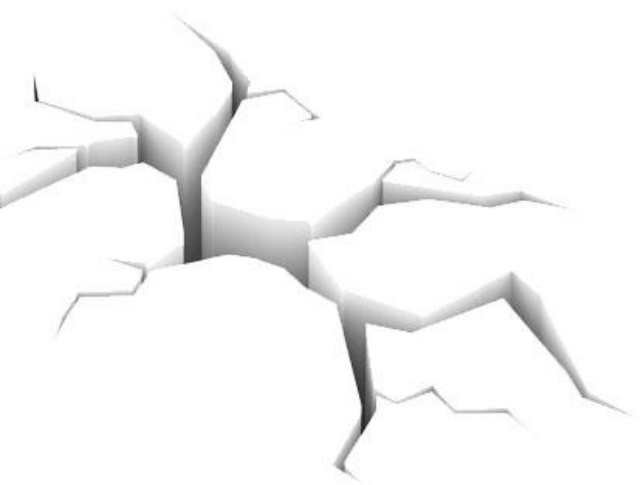


نصب بسته یا پکیج در R Studio

مانند R ، برای نصب بسته در R Studio به دو روش زیر می توان عمل کرد:

✓ روش آفلاین

✓ روش آنلاین



✓ آسانترین راه کلیک برروی گزینه Help در بالای صفحه نرم افزار می باشد

✓ در صورت دسترسی به اینترنت کمک ازسایت نرم افزار (CRAN)

✓ در صورتی که نام تابع را بدانیم با آوردن علامت ؟ ابتدای تابع به help نرم افزار

هدایت می شویم `?lm`

✓ وقتی نام تابع را به یاد نمی آوریم ولی می دانیم موضوع تابع چیست می توان از

دستور مقابل استفاده کرد: `help.search("data input")`

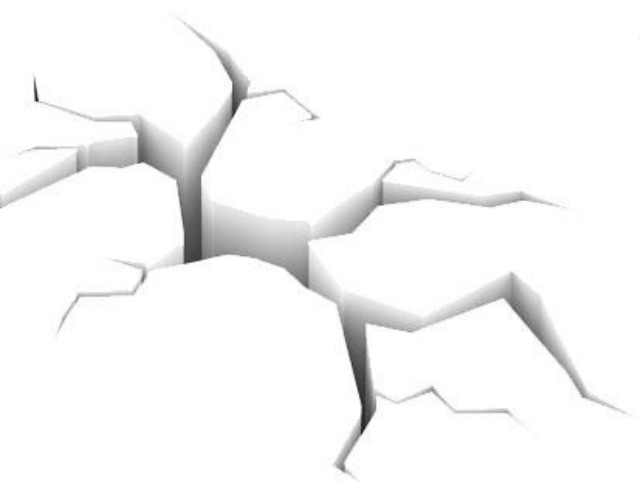
✓ دستور مقابل می گوید تابع در کدام پکیج است `find("lowess")`

✓ وقتی چند حرف از دستور را می دانیم `aproos("lm")`

✓ برای دسترسی به مثال های کار شده دستور `example(lm)`

✓ دستور ویرایش داده های موجود در نرم افزار `fix(data.frame.name)`

آشنایی با علائم و نشانه های موجود در R



علايم و نشانه ها

عملگرهای حسابی

| عملگر | توضیحات |
|---------|-----------------|
| + | جمع |
| - | تفریق |
| * | ضرب |
| / | تقسیم |
| ^ or ** | توان |
| x %% y | باقی مانده صحیح |
| x %/% y | تقسیم صحیح |

$$5 \bmod 2 = 1$$

$$5 = 2 * 2 + 1$$

عملگر های منطقی

| توضیحات | عملگر |
|-----------------------|-------|
| کو کتر | < |
| کوچکتر مساوی | <= |
| بزرگتر | > |
| بزرگتر مساوی | >= |
| دقیقا برابر | == |
| نا برابری | != |
| خود X نیست | !x |
| x و y | x y |
| x یا y | x & y |
| فراخوانی عضوی از لیست | \$ |

all() , any()

سور عمومی: به ازای همه مقادیر یک مجموعه یک گزاره منطقی درست باشد.

all()

سور وجودی: وجود داشته باشد عضوی از یک مجموعه که یک گزاره منطقی به ازای آن برقرار باشد.

any()



`floor(5.7)`

`ceiling(5.7)`

`trunc(5.7)`

`round(5.587,2)`

`signif(12345678,4)`

`z <- 3-5i`

`Re(z)`

`Im(z)`

`Mod(z)`

`Arg(z)`

`Conj(z)`

`is.complex(z)`

`as.complex(z)`

نمایش اعداد خیلی بزرگ یا خیلی کوچک (نمایش نمایی)

نرم افزار R برای نمایش اعداد خیلی بزرگ و خیلی کوچک طرح زیر را دارد:

$1.2e3$ means 1200 because the $e3$ means 'move the decimal point 3 places to the right'

$1.2e-2$ means 0.012 because the $e-2$ means 'move the decimal point 2 places to the left'

$3.9+4.5i$ is a complex number with real (3.9) and imaginary (4.5) parts, and i is the square root of -1 .

در زبان R داده‌های گم شده با علامت **NA (Not Available)** نشان داده می‌شوند.

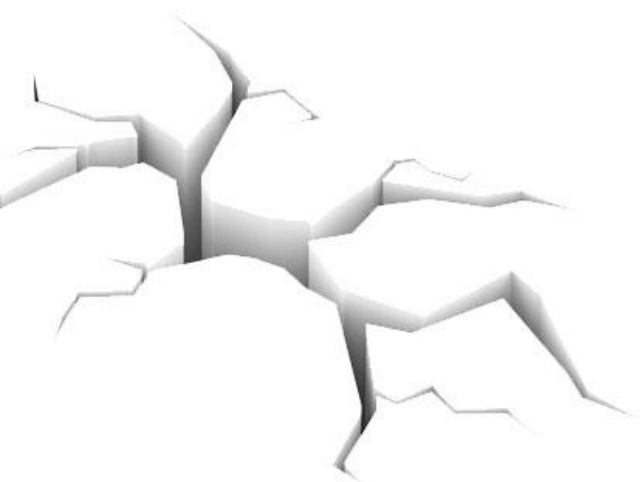
نحوه عمل توابع R در مورد NA ها متفاوت است:

➤ تابع **order()** داده‌های NA را به آخر سری منتقل می‌کند.

➤ تابع **sort()** داده‌های NA را خودبه‌خود حذف می‌کند.

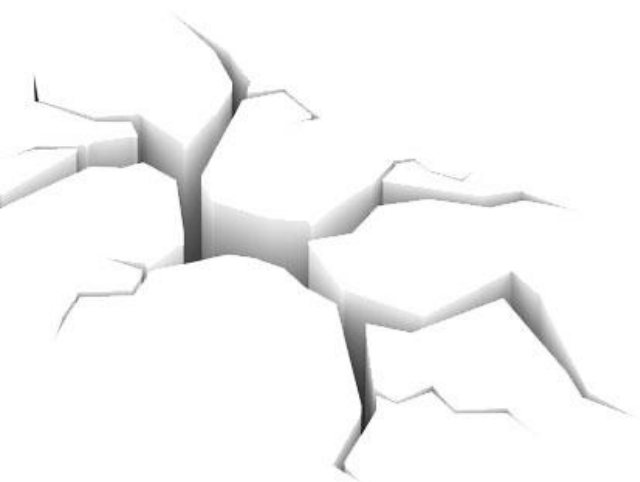
➤ در تابع **mean()** باید از آرگومان **na.rm** استفاده نمود.

اشیاء موجود در R



انواع مختلف اشیاء در R عبارتند از :

- ✓ بردارها
- ✓ فاکتورها
- ✓ ماتریسها
- ✓ آرایهها
- ✓ لیستها
- ✓ دیتا فریمها
- ✓ توابع.



برای ساخت بردارهایی توان دستورات زیر را به کاربرد که معادل هم هستند

```
y <- c(8,3,5)
```

```
assign("x",c(8,3,5))
```

برای تولید اعداد متوالی از عملگر ":" میتوان استفاده کرد

```
n <- 1:10
```

```
Seq(from=a, to=z, by=b, length=c)
```

برای تکرار یک بردار از تابع `rep()` به صورت زیر استفاده می کنیم
`rep(x, each = a, time = b)`

برای تولید نمونه تصادفی از تابع `sample()` به صورت زیر استفاده می کنیم
`sample(x , size , replace , prop)`

برای مرتب کردن یک بردار دستورات زیر را داریم



`sort(y)`
`rev(sort(y))`

بردارها

برای فراخوانی و یا کار بر روی عضو یا عضوهای از یک بردار از عملگر "[]" استفاده می شود

`X[a]`

`X[-a]`

بردار کارکترها نیز همانند اعداد با استفاده از تابع `c()` ساخته می شود. برای ایجاد یک بردار از کاراکترها از دستور زیر استفاده می شود که در آن `sep` مشخص می کند که کاراکترها و اعداد با چه کاراکتری از هم جدا شوند:

`paste("x", c(), sep = " ")`

برای تولید بردارهایی از حرف از بردارهای `LETTERS` و `letters` می توان استفاده کرد که هر کدام شامل ۲۶ حرف بزرگ و کوچک لاتین است

فاکتورها

فاکتورها بسیار شبیه بردارها هستند. از فاکتورها برای گروه بندی و ایجاد تمایز بین عناصر درون بردارها استفاده میشود. فاکتورها در R را می توان بطور مستقیم با دستور `gl()` و یا با تبدیل یک بردار به یک فاکتور با دستور `factor()` ، ساخت.

`factor(x)`

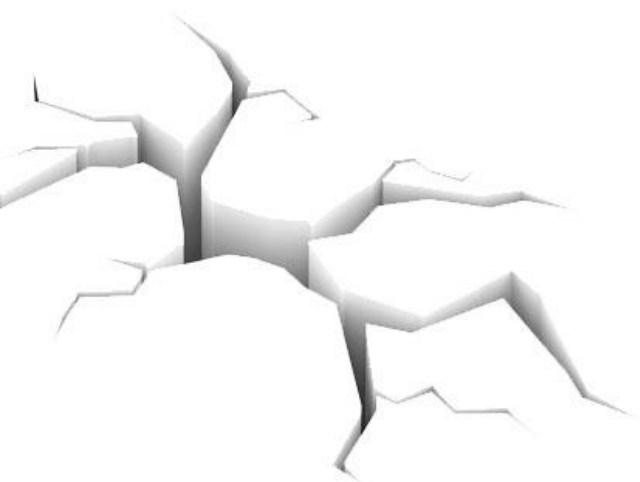
`gl(n, k, length=n*k, labels=1:n, ordered=F)`

که n مشخص کننده تعداد سطوح، k تعداد تکرارها، $length$ طول فاکتور و $labels$ برچسب سطح می باشد و `ordered` مشخص میکند سطوح مرتب باشند یا خیر.

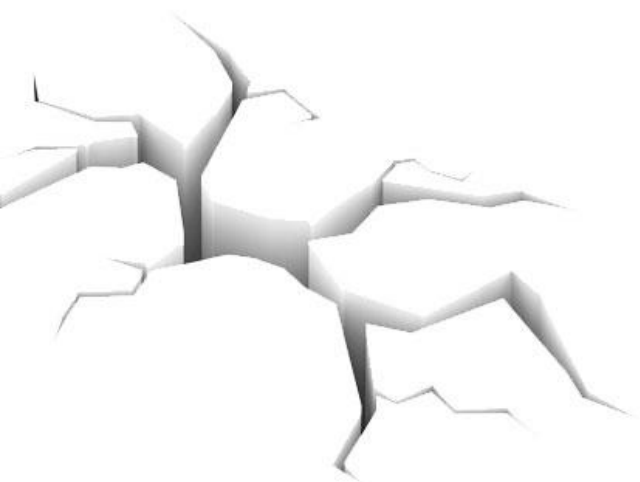
به آرایشی مستطیلی شکل از اعداد یا عبارات ریاضی که بصورت سطر و ستون شکل یافته، ماتریس گفته می شود.

یک طریقه ساختن یک ماتریس در R به صورت زیر است:

matrix(data = , nrow = , ncol = , byrow = , dimnames =)



در صورتی که بخواهیم با وصل کردن چند بردار یا چند ماتریس به هم ماتریس یک ماتریس جدید بسازیم از توابع `rbind()` و `cbind()` استفاده میکنیم. تابع `rbind()` بردار ها یا ماتریس ها را به صورت سطری کنار هم قرار می دهد و تابع `cbind()` این کار را به صورت ستونی انجام می دهد.



عملیات بر روی ماتریس ها

`rownames()`

برای نام گذاری سطر ها

`colnames()`

برای نام گذاری ستون ها

`t()`

ترانواده یک ماتریس

`A %*% B`

ضرب ماتریس ها

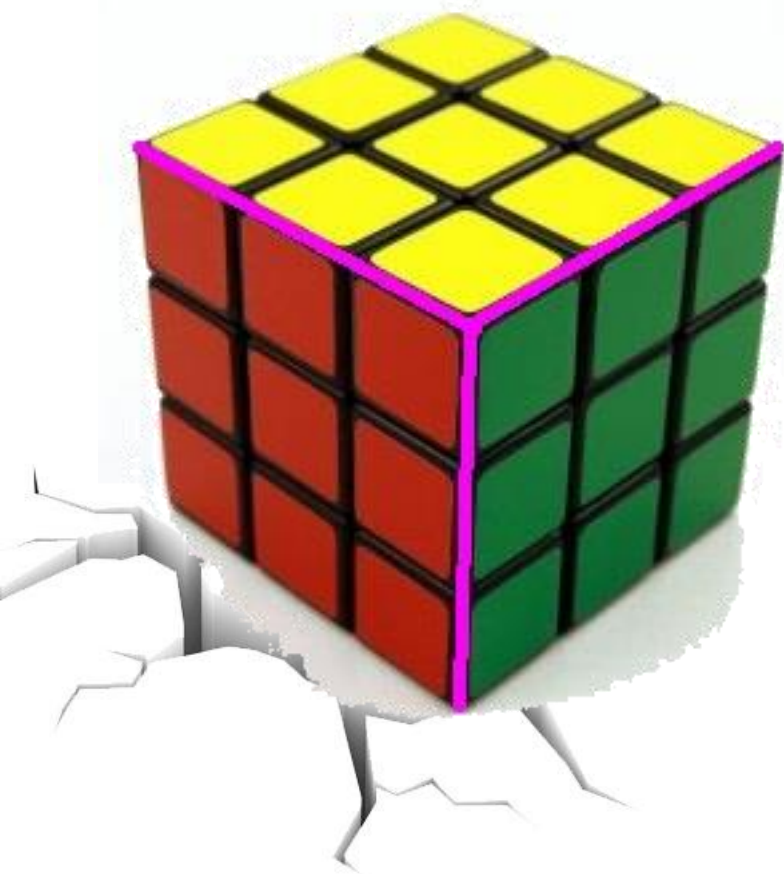
برای محاسبه معکوس یک ماتریس و یا حل یک معادله ی ماتریس از تابع

`solve()` استفاده میشود. اگر در تابع فقط ماتریس را وارد کنیم معکوس ماتریس

را به ما می دهد ولی اگر علاوه بر ماتریس بردار جواب را نیز وارد کنیم جواب

معادله $AX=b$ را می دهد.

آرایه ها



آرایه ها یکی از اشیا موجود در R برای
کار با داده ها هستند که ما را قادر
می سازند تا داده ها را به صورت چند
بعدی نگه داری کرده و با آنها کار کنیم.
در حالت خاص **بردارها** آرایه های یک بعدی
و **ماتریس ها** آرایه های دو بعدی هستند.

ساختن آرایه ها

ساده ترین راه ساختن یک آرایه استفاده از تابع `dim()` به صورت زیر است:

```
z <- 1:18
```

```
dim(z) <- c( 2 , 3 , 3 )
```

روش دیگر ساختن آرایه و ماتریس استفاده از تابع `array()` است. برای ساخت یک آرایه از این روش بردار داده ها و بردار بعد را در تابع مشخص میکنیم.

```
x <- 1:27
```

```
array( x , c(3 , 3 , 3) )
```


لیست ها

یک لیست مجموعه منظم از اشیا است که اشیا را مولفه های آن لیست می نامیم
مولفه های یک لیست لزومی ندارد هم نوع و یا هم اندازه باشند به عبارتی برای
یک لیست می توانیم مولفه هایی از قبیل ماتریس بردارهایی از اعداد یا کارکترها
و ... را داشته باشیم.

برای ساختن یک لیست از تابع `list()` استفاده میکنیم در این تابع نام هر مولفه
را وارد کرده و بعد از علامت مساوی شی مورد نظر را وارد می کنیم

`list(name = , score = , ...)`

دیتا فریم ها

دیتا فریم ها از لحاظ ساختار شباهت زیادی به ماتریس ها دارند. یک دیتا فریم را می توان ماتریسی در نظر گرفت که ستون های آن بردار هایی با ساختار های مختلف هستند.

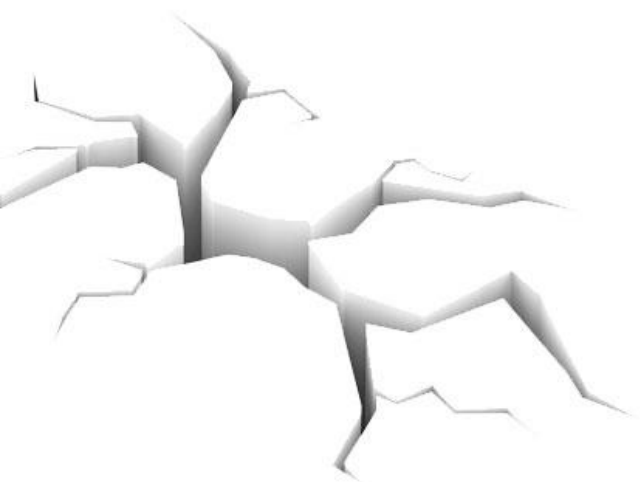
یک دیتا فریم در واقع لیستی از بردارهای هم اندازه و با ساختار های متفاوت است برای ساختن یک دیتا فریم از تابع `data.frame()` استفاده میشود. در این تابع می بایست نام مورد نظر برای هر ستون را به همراه بردار شامل داده های آن ستون وارد کنیم

توابع برداری موجود در R

| Operation | Meaning |
|--------------------------|---|
| <code>max(x)</code> | maximum value in x |
| <code>min(x)</code> | minimum value in x |
| <code>sum(x)</code> | total of all the values in x |
| <code>mean(x)</code> | arithmetic average of the values in x |
| <code>median(x)</code> | median value in x |
| <code>range(x)</code> | vector of <code>min(x)</code> and <code>max(x)</code> |
| <code>var(x)</code> | sample variance of x |
| <code>cor(x,y)</code> | correlation between vectors x and y |
| <code>sort(x)</code> | a sorted version of x |
| <code>rank(x)</code> | vector of the ranks of the values in x |
| <code>order(x)</code> | an integer vector containing the permutation to sort x into ascending order |
| <code>quantile(x)</code> | vector containing the minimum, lower quartile, median, upper quartile, and maximum of x |
| <code>cumsum(x)</code> | vector containing the sum of all of the elements up to that point |
| <code>cumprod(x)</code> | vector containing the product of all of the elements up to that point |
| <code>cummax(x)</code> | vector of non-decreasing numbers which are the cumulative maxima of the values in x up to that point |
| <code>cummin(x)</code> | vector of non-increasing numbers which are the cumulative minima of the values in x up to that point |
| <code>pmax(x,y,z)</code> | vector, of length equal to the longest of x , y or z , containing the maximum of x , y or z for the i th position in each |

توابع برداری موجود در R

| | |
|--------------------------|---|
| <code>pmin(x,y,z)</code> | vector, of length equal to the longest of x , y or z , containing the minimum of x , y or z for the i th position in each |
| <code>colMeans(x)</code> | column means of dataframe or matrix x |
| <code>colSums(x)</code> | column totals of dataframe or matrix x |
| <code>rowMeans(x)</code> | row means of dataframe or matrix x |
| <code>rowSums(x)</code> | row totals of dataframe or matrix x |



| Function | Meaning |
|------------------------------|---|
| log(x) | log to base e of x |
| exp(x) | antilog of x (e^x) |
| log(x,n) | log to base n of x |
| log10(x) | log to base 10 of x |
| sqrt(x) | square root of x |
| factorial(x) | $x!$ |
| choose(n,x) | binomial coefficients $n!/(x! (n-x)!)$ |
| gamma(x) | $\Gamma(x)$, for real x $(x-1)!$, for integer x |
| lgamma(x) | natural log of $\Gamma(x)$ |
| floor(x) | greatest integer $< x$ |
| ceiling(x) | smallest integer $> x$ |
| trunc(x) | closest integer to x between x and 0 $\text{trunc}(1.5) = 1$, $\text{trunc}(-1.5) = -1$ trunc is like floor for positive values and like ceiling for negative values |
| round(x, digits=0) | round the value of x to an integer |
| signif(x, digits=6) | give x to 6 digits in scientific notation |
| runif(n) | generates n random numbers between 0 and 1 from a uniform distribution |
| cos(x) | cosine of x in radians |
| sin(x) | sine of x in radians |
| tan(x) | tangent of x in radians |
| acos(x), asin(x), atan(x) | inverse trigonometric transformations of real or complex numbers |
| acosh(x), asinh(x), atanh(x) | inverse hyperbolic trigonometric transformations of real or complex numbers |
| abs(x) | the absolute value of x , ignoring the minus sign if there is one |

توابع و توزیع های معروف آماری در R

| نام توزیع | کد دستوری |
|------------|-----------|
| دو جمله ای | binom |
| پوآسن | pois |
| یکنواخت | unif |
| نرمال | norm |
| گاما | gamma |
| بتا | beta |
| خی دو | chisq |
| نمایی | exp |
| f | f |
| t | t |

d: تابع چگالی / جرم احتمال

p: تابع توزیع

q: چندک

r: تولید اعداد تصادفی

فرمت کلی جهت نوشتن تابع بدین صورت است:

```
name <- function( variables ) {
```

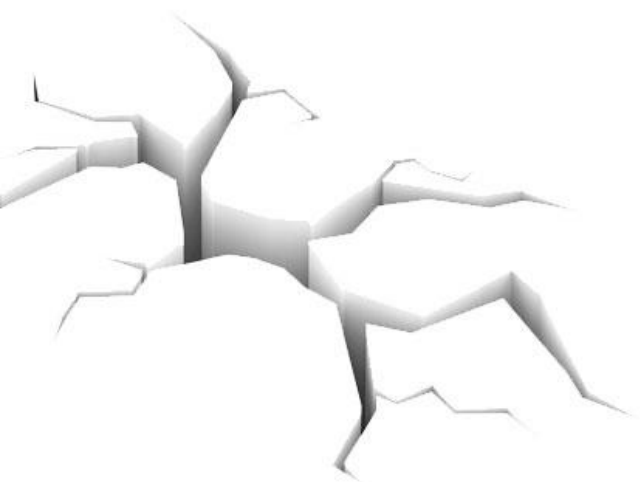
```
  statements
```

```
  return( )
```

```
}
```



دستورات شرطی و حلقه ها



دستورات شرطی

دستور شرطی **if**

بدین مفهوم که اگر شرط

برقرار شد این کار در غیر این

صورت برو کار دیگر را انجام

بده

```
if (شرط) {  
    در صورت برقراری شرط  
    این کار را انجام بده  
}  
else {  
    در صورت عدم برقراری شرط  
    این کار را انجام بده
```

، در صورت برقراری شرط این کار را انجام بده ، شرط) **ifelse**

(در صورت عدم برقراری شرط این کار را انجام بده


```
for (i in 1: n) {
```

دستورات

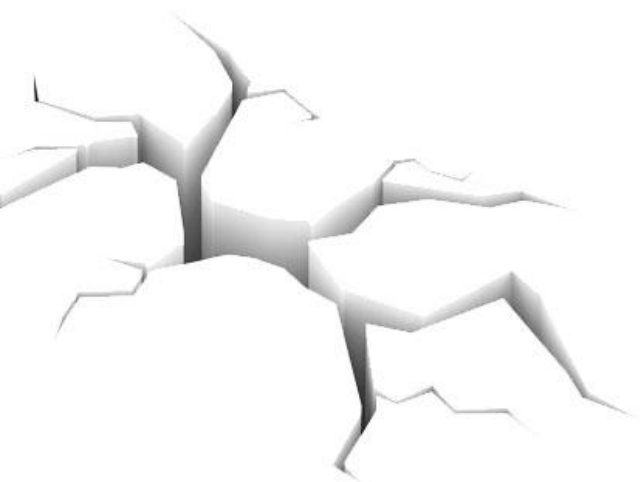
```
}
```

حلقه for()

حلقه دستوراتی که برایش

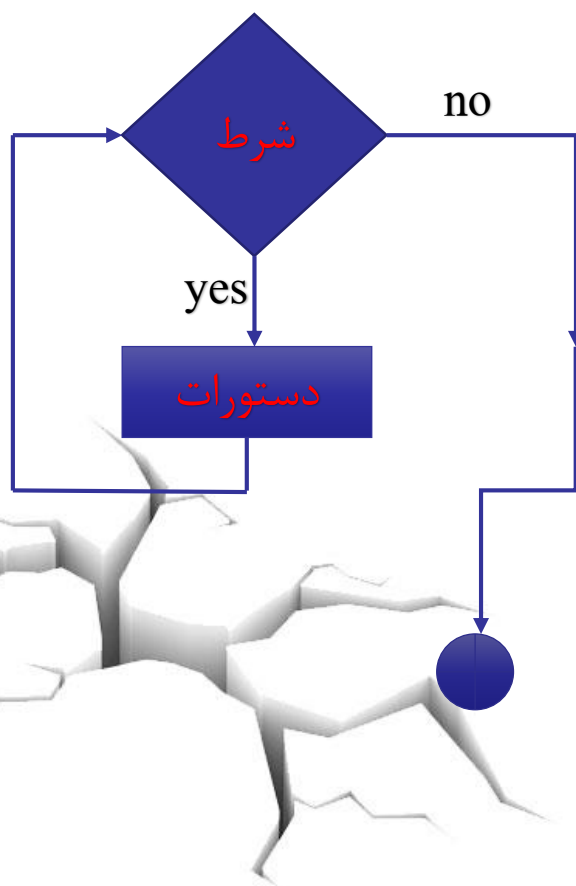
نوشته شده را به تعداد تعریف

شده تکرار می کند



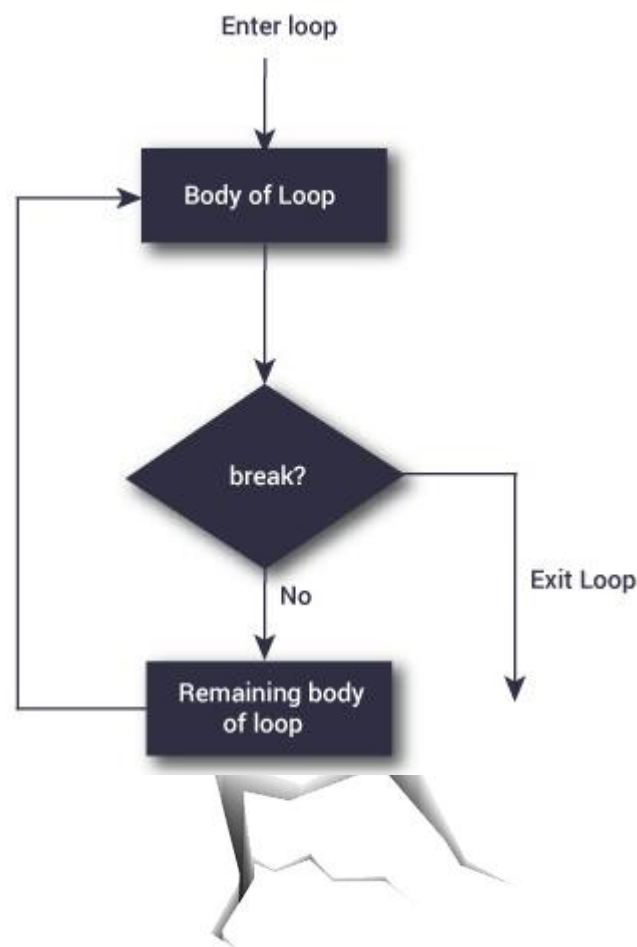
حلقه `while()`

تا زمانی که شرط داخلی برقرار باشد حلقه اجرا می شود. به طور کلی در این حلقه تعداد تکرار ها از ابتدای کار معلوم نیست.

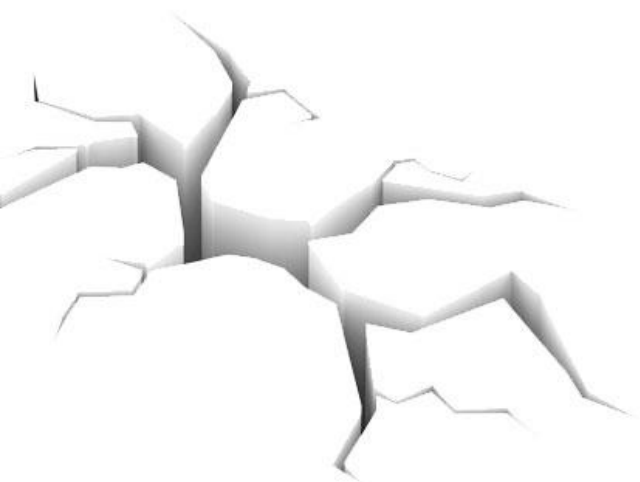


حلقه repeat()

تا زمانی که شرط داخلی برقرار باشد حلقه اجرا می شود. به طور کلی در این حلقه تعداد تکرار ها از ابتدای کار معلوم نیست.

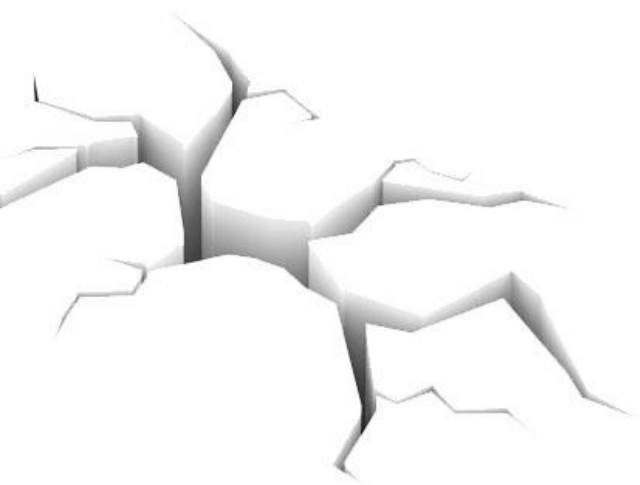


فراخوانی داده ها به R



فرمت کلی ورود دیتا به R

(....، "مسیری که داده ها ذخیره شده اند") پسوند داده ها.read <- Data.name



فایل با پسوند .txt

این فرمت ساده ترین نوع و جر پر کاربردترین فرمت های ذخیره داده به حساب می آید.

```
Text <- read.table(  
  "D:\\Chanel\\WorkShop\\01Introduction\\ISRT_IR.txt",  
  header = T , sep = "\\t")
```

Text

فایل با پسوند .dat

برای فراخوانی داده هایی با پسوند dat طبق فرمت کلی و از دستور

```
Dat <- read.delim (  
"D:\\Chanel\\WorkShop\\01Introduction\\ISRT_IR.dat",  
header = T , sep="\t")
```

Dat

فایل با پسوند CSV

فرمت CSV نیز فرمت پرکاربردی هست چرا که توسط notepad ویندوز قابل باز شدن است.

```
CSV <- read.csv(  
"D:\\Chanel\\Work Shop\\01 Introduction\\ISRT_IR.csv",  
header = T)
```

CSV



جهت فراخوانی فایل های اکسل به R روش ها و بسته های زیادی وجود دارد که در اینجا به یک مورد اشاره می کنیم. ابتدا بسته xlsx را که بسته های وابسته (xlsxjars و rJava) نیز دارد باید نصب شود.

```
install.packages("xlsx")
```

```
install.packages("xlsxjars")
```

```
install.packages("rJava")
```

```
library("xlsx")
```

```
library("xlsxjars")
```

```
library("rJava")
```

توجه! حتما برای استفاده از بسته xlsx باید برنامه java run time نیز را روی سیستم خود نصب کرده باشید.

فراخوانی فایل های اکسل

پس از نصب بسته های لازم، کافی طبق فرمت کلی و به شکل زیر فایل اکسل مورد نظر را به R فراخوانی کنید

```
Excel <- read.xlsx (  
"D:\\Chanel\\Work Shop\\01Introduction\\ISRT_IR.xlsx",  
sheetName = "Ahmad" )
```

Excel

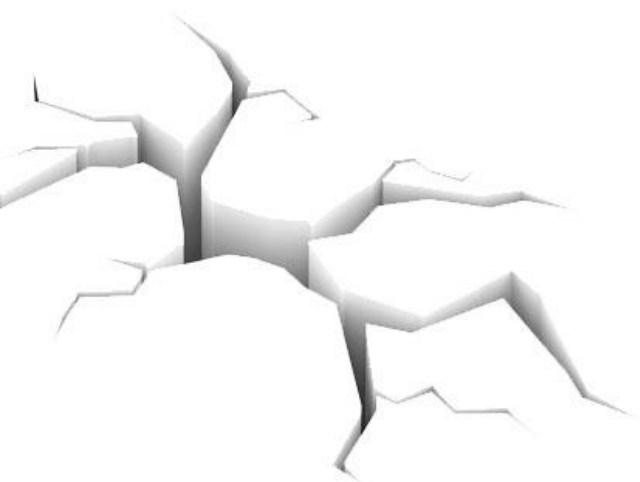


فایل های SPSS

جهت فراخوانی فایل های SPSS به R روش ها و بسته های زیادی وجود دارد که در اینجا به دو مورد اکتفا می کنیم.

✓ بسته foreign

✓ بسته Hmisc



فراخوانی فایل های SPSS

با استفاده از بسته foreign

ابتدا بسته را نصب و فراخوانی کرده سپس مانند فرمت کلی، فایل را فراخوانی

می کنیم

```
install.packages("foreign")
```

```
library("foreign")
```

```
Spss <- read.spss(
```

```
"D:\\Chanel\\Work Shop\\01 Introduction\\ISRT_IR.sav",
```

```
use.value.labels = F,
```

```
to.data.frame = T,
```

```
use.missings = T)
```

```
Spss
```

فراخوانی فایل های SPSS

با استفاده از بسته Hmisc

ابتدا بسته را نصب و فراخوانی کرده سپس با استفاده از دستور زیر، فایل را

فراخوانی می کنیم

```
install.packages("Hmisc")
```

```
library("Hmisc")
```

```
Spss1 <- spss.get (
```

```
"D:\\Chanel\\Work Shop\\01 Introduction\\ISRT_IR.sav",
```

```
use.value.labels = F,
```

```
to.data.frame = T,
```

```
use.missings = T)
```

```
Spss1
```

خواندن دیتا با استفاده از clipboard

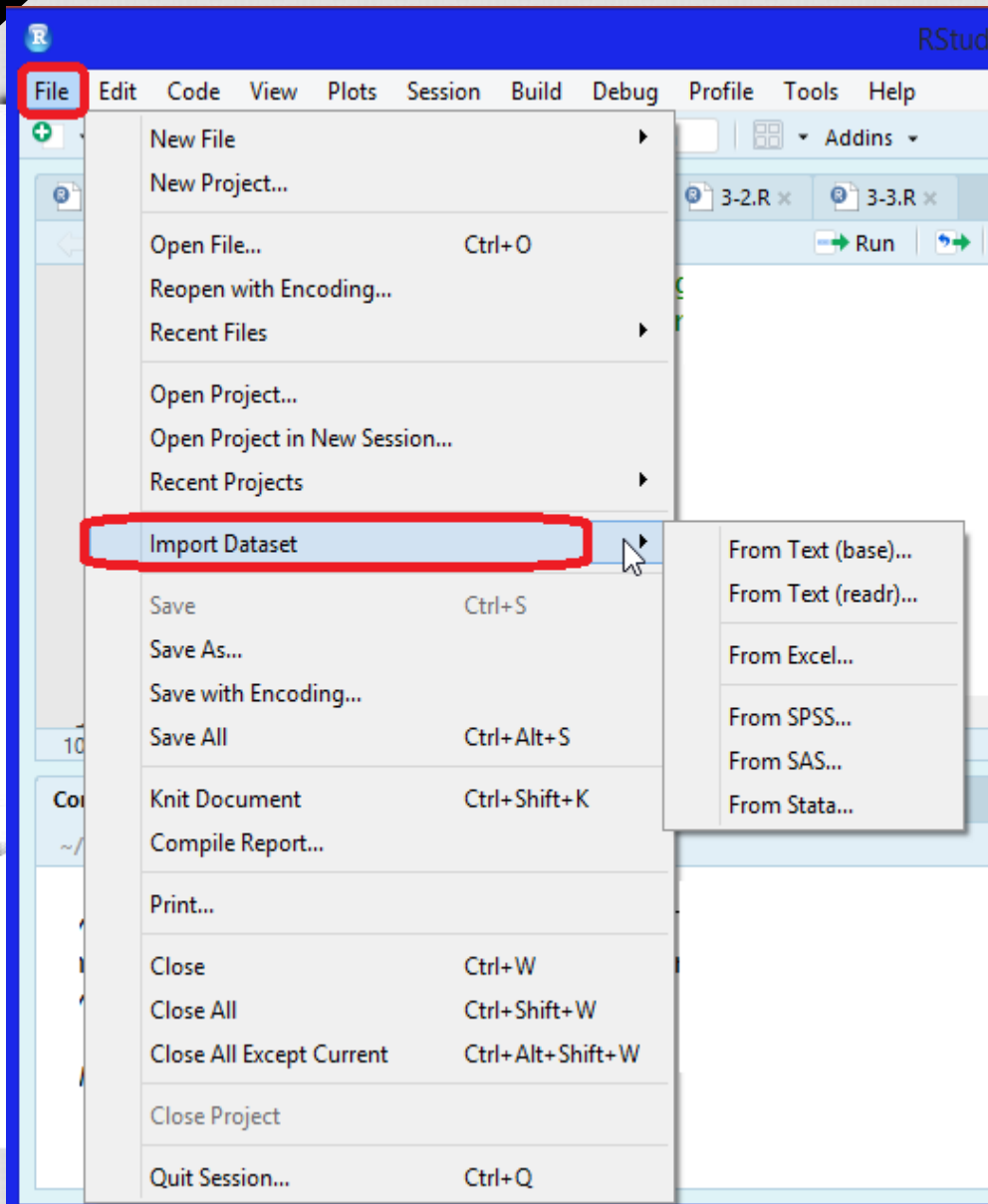
فایل اکسل ، SPSS و یا ... رو باز کنید. داده ها رو انتخاب و بعد کپی کنید و بعد در R این دستور رو اجرا کنید

```
Clipboard <- read.table("clipboard",  
sep="\t", header = T)
```

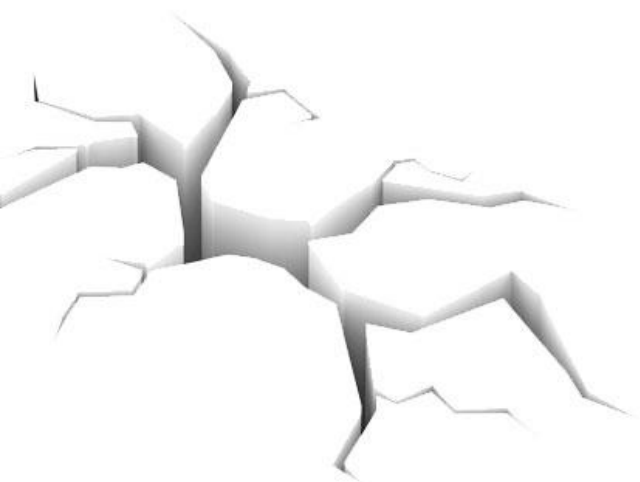
Clipboard



در صورتی که از Rstudio استفاده
می کنید کافی است مسیر
رو به رو را طی کنید:



ذخیره داده های موجود R

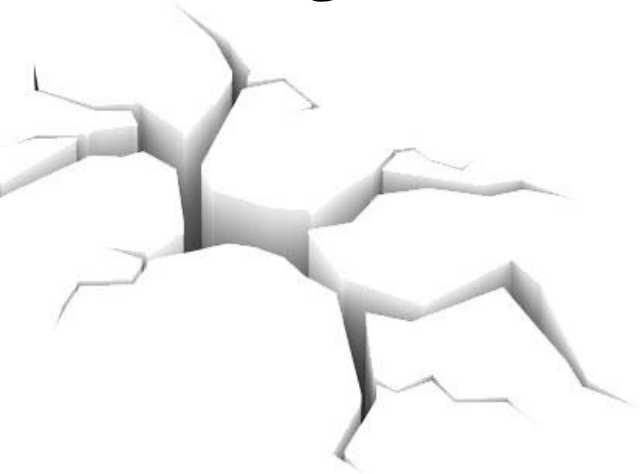


ذخیره داده های R به فرمت CSV

جهت ذخیره داده ها در یک فایل CSV از دستور زیر استفاده می شود:

```
write.csv( Data.name , " N_Data.name .csv")
```

با زدن این دستور یک فایل با نام **N_Data.name** در پوشه کاری ایجاد می شود.



ذخیره داده های R در فایل اکسل

جهت ذخیره داده ها در یک فایل اکسل، باید بسته های مربوطه فراخوانی شوند و سپس از دستور زیر استفاده می شود:

```
library("xlsx")
```

```
library("xlsxjars")
```

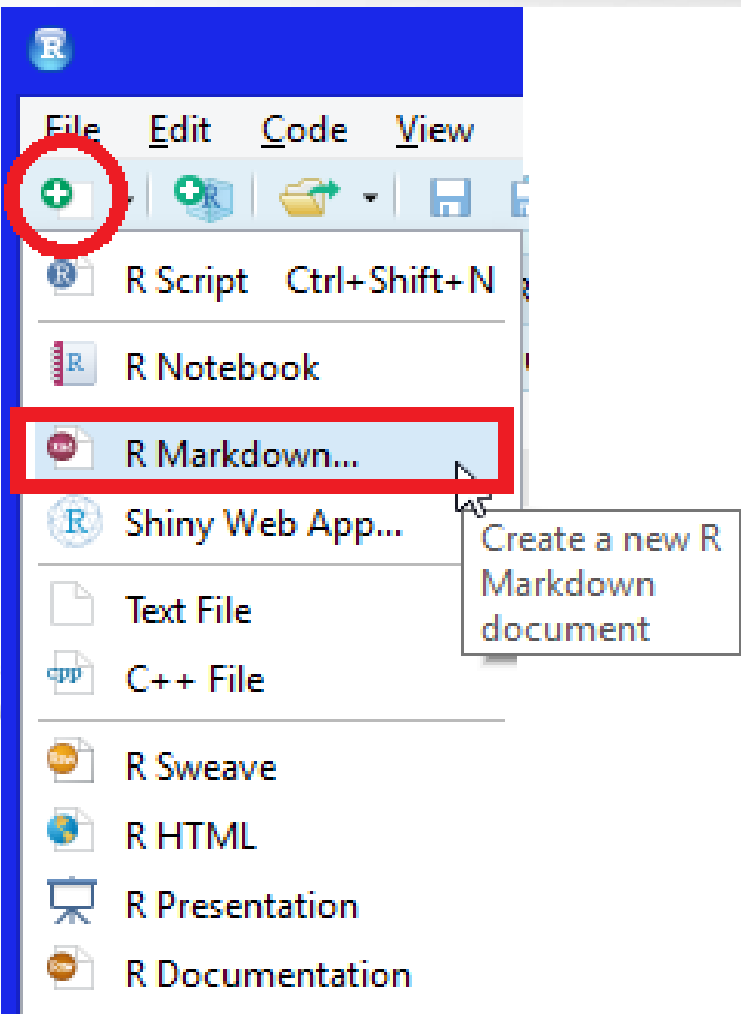
```
library("rJava")
```

```
write.xlsx( Data.name , " N_Data.name .xlsx")
```

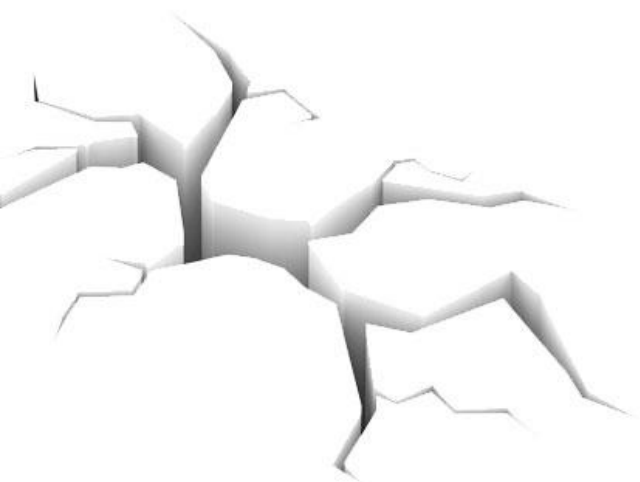
با زدن این دستور یک فایل با نام **N_Data.name** در پوشه کاری ایجاد می شود.

گرفتن خروجی به صورت pdf , word

با استفاده از بسته Rmarkdown
در Rstudio به راحتی می توان
خروجی ها منظم و مرتبی را در قالب
pdf, word و html گرفت.



ترسیم نمودارها در R

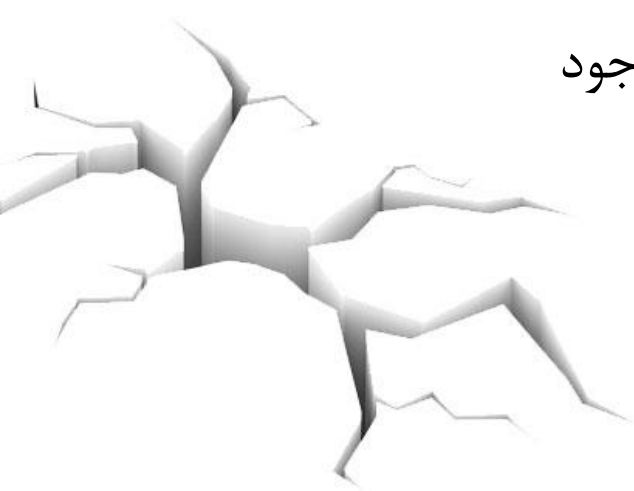


✓ توابع ترسیم در **R** را می‌توان به سه دسته تقسیم کرد:

➤ توابع ترسیم سطح بالا: رسم نمودارهایی شامل محورها، برچسب‌ها، عنوان و مواردی از این دست.

➤ توابع ترسیم سطح پایین: اعمال تغییرات روی نمودارهای موجود

➤ توابع ترسیم تعاملی: تبادل اطلاعات با نمودارهای موجود



✓ یکی از پرکاربردترین توابع رسم نمودار تابع **plot()** است.

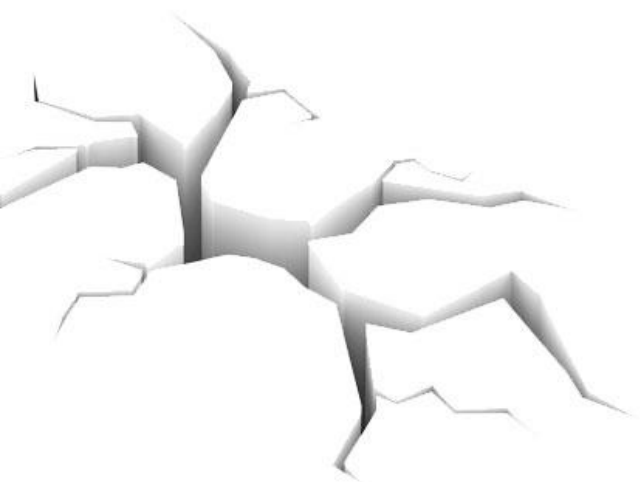
➤ **plot(x, y, ...)**

برخی از پارمترهای تابع **plot()**:

□ **type**: نوع نمودار

□ **lwd**: ضخامت خطوط

□ **lty**: نوع خطوط



□ pch: نوع نقاط

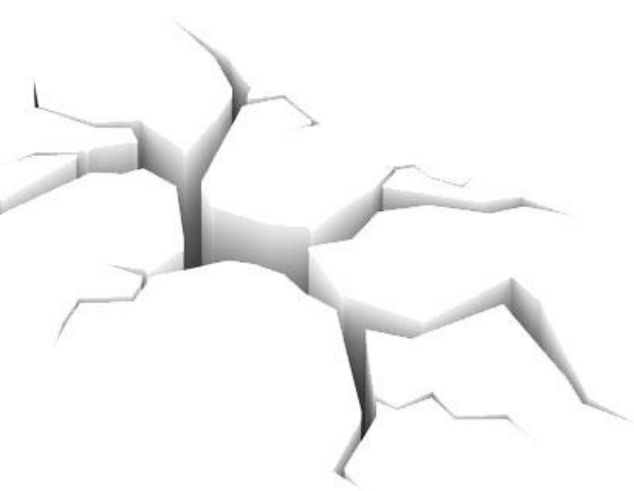
□ main: عنوان اصلی نمودار

□ sub: عنوان فرعی نمودار

□ xlab , ylab: اسامی محورهای مختصات

□ xlim , ylim: محدوده محورهای مختصات

□ col: رنگ نمودار



برخی دیگر از توابع ترسیم سطح بالا عبارتند از:

✓ **curve()**: رسم منحنی

✓ **pairs()**: رسم ستون‌های ماتریس در مقابل هم

✓ **coplot()**: رسم یک بردار در مقابل بردار دیگر به شرط یک بردار یا فاکتور دیگر

✓ **hist()**: رسم نمودار هیستوگرام

✓ **barplot()**: رسم نمودار میله‌ای



رسم نمودار ها

✓ **boxplot()**: رسم نمودار جعبه‌ای

✓ **dotchart()**: رسم نمودار نقطه‌ای

✓ **image()** , **contour()** , **persp()**: رسم نمودارهای سه‌بعدی

✓ **qqnorm()** , **qqline()** , **qqplot()**: رسم نمودارهای چندک-چندک

✓ **acf()** , **pacf()**: رسم نمودارهای ACF و PACF در سری‌های زمانی



✓ **text()**: اضافه کردن برداری از اعداد یا کاراکترها به نمودار موجود

✓ **lines()**: رسم نمودار جدید در نمودار موجود

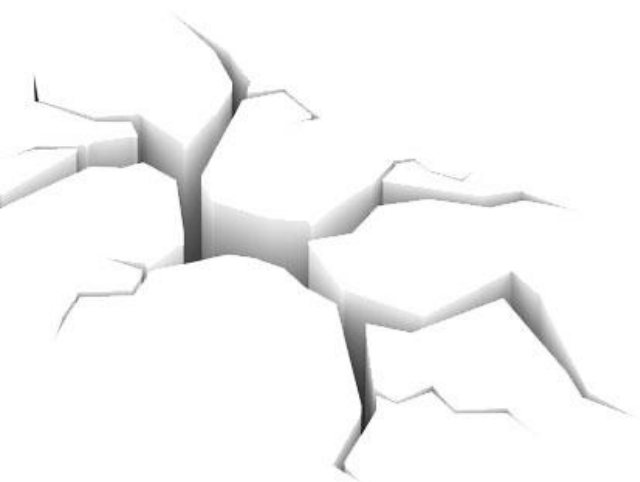
✓ **abline()**: رسم یک خط با شیب و عرض از مبدا دلخواه روی نمودار موجود

✓ **legend()**: وارد کردن یک نوشته در قسمتی از نمودار



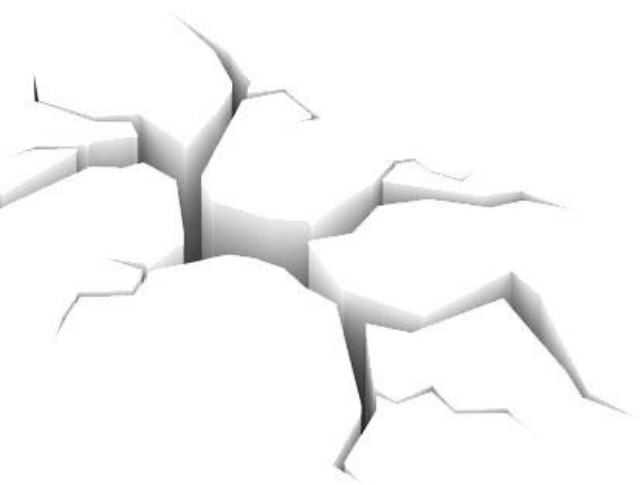
✓ **locator()**: تعیین مختصات نقاط با استفاده از ماوس

✓ **identify()**: تعیین ترتیب نقطه با استفاده از ماوس



❖ برای رسم چند نمودار جداگانه در صفحات مجزا، از تابع **windows()** استفاده می شود.

❖ برای رسم چند نمودار جداگانه در یک صفحه از دستورات **par()** استفاده می شود.



با تسکین از

توجه تان

