بسم الله الرحمن الرحيم

آشنایی با نرم افزار آماری

مهدی جباری نوقابی، دانشگاه فردوسی مشهد، گروه آمار

کارگاه آموزشی، دانشکده ادبیات و علوم انسانی.....۹۴/۱۱/۲۷

۱ مقدمه ای بر R

زبان R، یک زبان برنامه‌نویسی و محیط نرم‌افزاری برای محاسبات آماری و تحلیل داده است. امروزه این زبان به عنوان یک استاندارد غیر رسمی برای کارهای آماری و داده‌کاوی مطرح می‌باشد[۴]. این زبان در حقیقت نسخه متن‌باز نرم‌افزار S می‌باشد [۵]. زبان R توسط نرم‌افزاری به همین نام که شامل مفسر زبان و محیط اسکریپت‌نویسی است پشتیبانی می‌گردد. این نرم‌افزار، بازمتن بوده و تحت اجازه‌نامه عمومی همگانی گنو عرضه می‌شود. نسخه‌های R بصورت رایگان و برای انواع سیستم‌های عامل (ویندوز، مک و انواع توزیع‌های لینوکس) ارائه شده‌است. هسته اصلی نرم‌افزار R به همراه بیش از ۴۳۰۰ کتابخانه مرتبط با آن، در شبکه CRAN (comprehensive R archive network) در دسترس کاربران می‌باشد. این شبکه که مخزن منابع مرتبط با R می‌باشد، از سرورهای دانشگاه‌های مختلف سراسر جهان برای نگهداری و گرفتن نسخه پشتیبان از این داده‌ها استفاده می‌کند[۱]. در میهن عزیز اسلامی ما نیز، دانشگاه فردوسی مشهد یکی از مراکز نگهداری نسخه‌های R می‌باشد[۲].

در حال حاضر اگر چه هیچ شرکت تجاری پشتیبان رسمی R را بر عهده ندارد ، اما شبکه پشتیبانی غیررسمی آن از طریق پیام‌های اینترنتی r-help قابل دسترس است. در حال حاضر R نرم افزاری کامل و بی‌نقص نیست و از آنجا که آزاد و رایگان هم می باشد، از پیشنهادات کاربران برای اصلاح و توسعه آن استقبال می‌کند.

۲-۱ مروری بر امکانات R

R یک مجموعه کامل از امکانات نرم افزاری برای کارکردن با داده ها و محاسبه و رسم نمودار می باشد. از جمله امکانات این مجموعه می‌توان به موارد زیر اشاره کرد:

- زبان برنامه‌نویسی ساده و پیشرفته شامل عبارت‌های شرطی، حلقه و توابع بازگشتی و ... .
- امکانات ذخیره، بازیابی و دستکاری داده‌ها
- مجموعه‌ای قوی از عملگرهای محاسباتی آرایه‌ها و ماتریس‌ها
- بسته‌های نرم‌افزاری قدرتمند برای تجزیه و تحلیل آماری
- کتابخانه‌های انجام عملیات داده‌کاوی و یادگیری ماشین مانند دسته‌بندی، خوشه‌بندی، تحلیل شبکه‌اجتماعی، یادگیری تقویتی و ...
- امکانات گرافیکی برای تجزیه و تحلیل داده‌ها و رسم نمودار
- کتابخانه‌های خاص منظوره برای انجام عملیات تحلیلی در زمینه‌های مختلف علمی
- دارای مستندات فرمت‌بندی شده و منظم برای استفاده از زبان و کتابخانه‌های مرتبط

طرز نصب نرم‌افزار

ابتدا فایل اجرایی R-2.7.0-win32.exe را از سایت R دانلود[5] نموده و با دوبارکلیک متوالی رؤی آن نصب می‌شود. پس از پایان عمل نصب رؤی desktop کامپیوتر شما یک آیکون به شکل حرف R قرار می‌گیرد. اگر رؤی حرف R دوبار کلیک کنید، صفحه‌ای باز می‌شود که R Console نام دارد. در این صفحه یک مقدار توضیحاتی وجود دارد و پس از آن علامت «>» ملاحظه می‌شود که در مقابل آن می‌توان عملیات مورد نظر را انجام داد.

۱-۱ تاریخچه پروژه R

پروژه R در سال ۱۹۹۱ در گروه آمار دانشگاه Auckland کشور نیوزیلند کلید خورد. بنیان‌گذاران این پروژه آقایان Ross Ihaka و Robert Gentelman بودند. وجه تسمیه این زبان نیز ابتدای نام این در نفر می‌باشد (در دپارتمان آمار این دانشگاه این پروژه به "R&R" نیز معروف می‌باشد.). در حال حاضر تیمی متشکل از ۱۹نفر در حال توسعه این طرح می‌باشند.

R برای اولین بار در سال ۱۹۹۳ به طور عمومی معرفی گردید. در ۱۹۹۵ مارتین ماچلر، راس و رابرت را متقاعد نمود تا تحت مجوز گنو، R را به یک نرم‌افزار رایگان تبدیل نمایند. یک سال بعد لیست رایانامه عمومی این پروژه ایجاد گردید (R-help و R-devel) که به کمک آن توسعه‌دهندگان و کاربران در سراسر جهان با یکدیگر مرتبط می‌شدند. در ۱۹۹۷ هسته اصلی توسعه‌دهنده R که شامل برخی از اعضای پروژه S-Plus نیز بودند ایجاد شد. این گروه، وظیفه کنترل و نظارت بر کد برنامه را به عهده داشت. با تلاش‌های توسعه‌دهندگان، در سال ۲۰۰۰ اولین نسخه R ارائه گردید. آخرین نسخه ارائه شده این نرم‌افزار 2.13.1 می‌باشد که در ۲۰۰۱/۷/۸ منتشر شده‌است.

داده‌ها در R

همانطورکه ملاحظه شد R با شی‌ها کار می‌کند که خود آن‌ها توسط نام و محتوی مشخص می‌شوند. هم‌چنین نوع داده که در شی قرار دارد با خصوصیت[8] معین می‌گردد. تمام شی‌ها دارای دو خصوصیت است:

- mode: نوع عناصر یک شی را مشخص می‌کند. چهار نوع اصلی mode وجود دارد: عددی، کاراکتر، مختلط و منطقی. البته modeهای دیگری نیز وجود دارد که در مورد data به‌کار نمی‌رود. برای مثال می‌توان از تابع یا عبارت نام برد.
- طول (length): تعداد عناصر یک شی را نشان می‌دهد.

اکنون به مثال‌های زیر توجه کنید.

```
> a<-"sahar";b<-TRUE;c<-1i
> mode(a);mode(b);mode(c)
[1] "character"
[1] "logical"
[1] "complex"
```

داده‌های آماده در زبان R

در زبان R تعدادی داده آماده برای استفاده در مثال‌ها قرار داده شده است که در این نوشتار نیز از آن استفاده می‌شود. مانند cars، iris، LakeHuron، Nile، trees و . . . البته می‌توان فهرست کامل آن‌ها را با استفاده از دستور زیر در R یافت.

```
> data()
```

```
> data(cars)
> cars
   speed dist
1      4    2
2      4   10
3      7    4
4      7   22
5      8   16
6      9   10
7     10   18
```

خواندن داده‌ها از فایل

زبان R داده‌های متنی (text) را از روی فایل می‌خواند. برای این کار می‌توان از دو روش استفاده نمود.

دستور خواندن read.table

اولین دستور مورد استفاده تابع read.table() است. اگر نام فایلی data.txt باشد و مثلاً در درایو c و در پوشه‌ای به نام test ذخیره شده باشد، آنگاه دستور خواندن به صورت زیر عمل می‌کند.

```
> mydata <- read.table("c:\\test\\data.txt")
```

با اجرای دستور فوق یک جدولی از داده‌ها به نام mydata تشکیل می‌شود که هر متغیر آن دارای نام است. به‌طور پیش‌فرض V1, V2, ... نامیده می‌شود و دسترسی به آن‌ها به صورت‌های mydata$V1, mydata$V2, ... و یا mydata["V1"], mydata["V2"], ... و mydata[, 1], mydata[, 2], ... می‌باشد. دستور read.table شامل خصوصیات اختیاری است که می‌توان حسب مورد از آن استفاده نمود.

اکنون به مثال زیر توجه کنید. فایلی به صورت زیر data.txt ذخیره شده است.

```
Author: John Davis
Date: 18-05-2007
Some comments...
Col1, Col2, Col3, Col4
23, 45, A, John
34, 41, B, Jimmy
12, 99, B, Patrick
```

حالا در زبان R فراخوانده می‌شود.

```
> mydata <- read.table("c: test data.txt",skip=3,sep=",",header=T)
> mydata
```

در تابع read.table() آرگومان skip از سه سطر اول فایل که مربوط توضیحات است عبور می‌کند و آرگومان sep ویرگول بین داده‌ها را حذف می‌نماید و بالاخره آرگومان header که از نوع منطقی است و در اینجا «T» که مخفف TRUE است ذکر شده، بنابراین اسامی داده‌ها را حفظ می‌کند. اکنون به خروجی توجه کنید.

	Col1	Col2	Col3	Col4
1	23	45	A	John
2	34	41	B	Jimmy
3	12	99	B	Patrick

در حال حاضر بسیاری از داده‌ها قبلاً در نرم‌افزار Excel ذخیره شده است. زبان R به تنهایی قادر به خواندن مستقیم فایل‌های Excel (یعنی پسوندهای *xls. و یا *xlsx.) نیست. برای خواندن چنین داده‌هایی سه راه حل وجود دارد.

۱) می‌توان داده‌ها را در Excel به فرمت *txt. ذخیره نمود، سپس با تابع read.table() آنها (فایل متنی *txt.) را خواند.

۲) می‌توان داده‌ها را در Excel به فرمت *csv. ذخیره نمود، سپس با تابع read.csv("نام و آدرس فایل") آنها را خواند.

۳) استفاده از package مربوطه که با استفاده از آن زبان R مستقیماً فایل‌های Excel را می‌خواند.

```
> read.csv("H:\\sahar\\book1.csv")
```

تولید داده‌ها

دنباله منظم از اعداد صحیح:

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

تابع ()seq می‌تواند دنباله‌ای از اعداد حقیقی را تولید کند. به عنوان مثال:

```
> seq(1,5,0.5)
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

در این تابع، اولین عدد شروع دنباله، دومین عدد خاتمه دنباله و سومین عدد میزان افزایش را نشان می‌دهد.

تابع دیگری تحت عنوان ()rep وجود دارد که اولین آرگومان آن بردار و دومین آرگومان آن تعداد تکرار عناصر آرگومان اول است. به مثال زیر توجه کنید.

```
> rep(1:5,2)
 [1] 1 2 3 4 5 1 2 3 4 5
```

ایجاد دنباله‌های تصادفی از داده‌ها

این نوع دنباله توسط توزیع‌های آماری قابل تولید است. بنابراین شرح مختصری در این باره ذکر می‌گردد. زبان R شامل تعداد قابل ملاحظه‌ای از توابع جرم و چگالی احتمال است. شکل عمومی این توابع به صورت rfunc(n, p1, p2, . . .) است که حرف r از واژه random به معنای تصادفی اخذ شده است. در آن‌ها n تعداد اعدادی است که قرار است تولید شود. حروف p1, p2, · مقادیر پارامترهای تابع را نشان می‌دهد. به مثال زیر توجه کنید.

```
> rnorm(4)
[1] -0.5484634 -0.3182905  1.8343405
0.1463850
```

ساختار داده‌ها

در زبان R می‌توان داده‌ها را به صورت‌های زیر نگهداری نمود.

- بردار (vector)
- ماتریس (matrix)
- آرایه (array)
- داده‌های چارچوب‌دار (data frame)
- داده‌های سری زمانی (time series)
- فهرست (list)

اکنون به شرح مجزای هر یک از آن‌ها پرداخته می‌شود.

بردارها

ساده‌ترین ساختار داده‌ها در زبان R، بردارها هستند. بردار موجودیتی است که شامل چند داده با نوع یکسان هستند، تماماً عدد و یا تماماً منطقی و . . . می‌باشند. همانطورکه قبلاً نیز ملاحظه شد می‌توان با تابع c() بردار را ساخت. به مقال زیر توجه کنید.

```
> x<-c(10,5,3,6)
```

محاسبات ریاضی روی بردارها

محاسبات روی بردارهای عددی معمولاً روی هر عنصرش انجام می‌شود. برای مثال x*x هر عنصر بردار x را مربع می‌کند.

```
> z<-x*x
> z
[1] 100  25   9  36
```

می‌توان توابع را روی عناصر یک بردار اثر داد. مثلا تابع لگارتیم را روی بردار x اعمال نمود.

```
> log(x)
[1] 2.302585 1.609438 1.098612
1.791759
```

ایجاد زیربردار

به دو صورت می‌توان یک زیربردار را ایجاد نمود.

- شماره عناصری که باید انتخاب شود را مشخص کنید. مثال

```
> x[c(2,3)]
[1] 5 3
```

- با استفاده از اعداد منفی (شماره عناصر) می‌توان عناصر غیر لازم را حذف نمود.

```
> x[-c(1,4)]
[1] 5 3
```

ماتریس‌ها

در واقع ماتریس بسط بردار است. مانند بردار تمام عناصر یک ماتریس دارای نوع داده‌های یکسان است. برای ساختن ماتریس کافی است برای عناصر تابع c() سطر و ستون تعریف نمود. به مثال زیر توجه کنید.

```
> a<-matrix(c(1,2,3,4),nr=2,nc=2)
> a
     [,1] [,2]
[1,]    1    3
[2,]    2    4
```

توجه داشته باشید که عناصر ماتریس در زبان R به‌صورت ستونی (پیش‌فرض) ذخیره می‌شوند.

اگر بخواهید که نحوه‌ی ذخیره شدن به شکل سطری باشد باید از عبارت byrow=T استفاده نمود. به مثال زیر توجه کنید.

```
> xx<-matrix(1:6, nc=3,byrow=T)
> xx
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
>
```

توابع rbind() و cbind() می‌توانند دو آرایه و یا دو ماتریس را بر حسب سطر و یا ستون به یکدیگر متصل نمایند. به عنوان مثال اولین ماتریس را در نظر بگیرید.

```
> m1<-matrix(1,nr=2,nc=2)
> m2<-matrix(2,nr=2, nc=2)
> rbind(m1,m2)
     [,1] [,2]
[1,]    1    1
[2,]    1    1
[3,]    2    2
[4,]    2    2
> cbind(m1,m2)
     [,1] [,2] [,3] [,4]
[1,]    1    1    2    2
[2,]    1    1    2    2
>
```

با تابع apply() می‌توان یک تابع را روی ستون، سطر و یا سطر و ستون اعمال نمود و نیازی به برنامه‌نویسی نیست. نحو کلی به صورت apply(X, MARGIN, FUN,...) است، که در آن X ماتریس، MARGIN نشان‌دهنده سطر (۱)، ستون (۲) و یا هر دو c(1,2)، FUN تابعی است که قرار است اعمال گردد و ... آرگومان‌های اختیاری تابع است. به مثال زیر توجه کنید.

```
> x<-rnorm(10,-5,0.1)
> y<-rnorm(10,5,2)
> X<-cbind(x,y)
> X
           x       y
 [1,] -5.004049 5.410928
 [2,] -4.949671 7.843164
 [3,] -4.887482 3.551941
 [4,] -4.859480 7.046951
 [5,] -5.012243 7.027064
 [6,] -5.097387 4.890546
 [7,] -5.021634 7.474694
 [8,] -4.924094 3.644602
 [9,] -5.075319 7.052260
[10,] -4.975315 8.312414
> apply(X,2,mean)
       x        y
-4.980667  6.225456
> apply(X,2,sd)
         x         y
0.0770022 1.7252354
```

در زبان R، داده‌ای وجود دارد که trees نامیده می‌شود و شامل سه ستون است. تابع ()apply به عنوان مثال روی آن اعمال می‌گردد.

```
> data(trees)
> apply(trees,2,sum)
 Girth Height Volume
 410.7 2356.0  935.3
```

اکنون اگر بخواهید که فقط مثلاً جمع یک ستون (Height) را محاسبه کنید، می‌توان به دو صورت زیر عمل نمود.

```
> sum(trees[,2])
[1] 2356
```

اما بالاخره در هر دو صورت فوق برای محاسبه جمع، میانگین و . . . فقط از نام ستون داده‌های trees نمی‌توان استفاده نمود. برای این کار از تابعی در زبان R به نام ()attach می‌توان استفاده نمود. ابتدا نام داده را با تابع مذکور فرا می‌خوانید و سپس می‌توان با نام ستون‌ها عملیات فوق را انجام داد.

```
> attach(trees)
> sum(Height)
[1] 2356
```

مشاهده ساختار یک شی

می‌توان از تابعی استفاده نمود که ()str نامیده می‌شود. نام تابع str از
کلمه structure به معنای ساختار ناشی می‌گردد. این تابع می‌تواند ساختار هر شی (اعم از داده‌ها، متغیرها، توابع و . . .)
را نشان دهد. به مثال زیر توجه کنید.

```
> str(trees)
'data.frame':   31 obs. of  3 variables:
 $ Girth : num  8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2
...
 $ Height: num  70 65 63 72 81 83 66 75 80 75 ...
 $ Volume: num  10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2
22.6 19.9 ...
```

در اینجا علاوه بر تعداد و اسامی متغیرها می‌توان نوع داده‌ها، تعداد داده‌ها و قسمتی از داده‌های هر متغیر را مشاهده
نمود.

ماتریس و عملیات آن

برای ضرب دو ماتریس از عملگر «%*%» استفاده می‌شود. برای مثال به حاصل‌ضرب دو ماتریس m1, m2 توجه کنید.

```
rbind(m1,m2)%*%cbind(m1,m2)
     [,1] [,2] [,3] [,4]
[1,]    2    2    4    4
[2,]    2    2    4    4
[3,]    4    4    8    8
[4,]    4    4    8    8
```

ترانهاده یک آرایه توسط تابع t() حاصل می‌گردد. تابع diag() برای استخراج و یا تغییر درآیه‌های قطری و یا ساختن یک ماتریس قطری به‌کار می‌رود.

```
> t(xx)
     [,1] [,2]
[1,]    1    4
[2,]    2    5
[3,]    3    6
> diag(m1)
[1] 1 1
> diag(m1)<-10
> m1
     [,1] [,2]
[1,]   10    1
[2,]    1   10
```

زبان R توابع خاصی برای محاسبات ماتریسی دارد. دستور det() برای محاسبه دترمینان، دستور solve() برای معکوس نمودن ماتریس و دستور eigen() برای بدست آوردن مقادیر و بردارهای ویژه به‌کار می‌رود.

```
> det(m1)
[1] 99
> solve(a)
     [,1] [,2]
[1,]   -2  1.5
[2,]    1 -0.5
> eigen(m1)
$values
[1] 11  9

$vectors
          [,1]       [,2]
[1,] 0.7071068 -0.7071068
[2,] 0.7071068  0.7071068
```

داده‌های چارچوب‌دار[۱۲]

در واقع این نوع داده‌ها بسط ماتریس است. داده‌های چارچوب‌دار دارای ستون‌های با نوع داده‌های مختلف است و مناسب‌ترین ساختار داده‌ها در تجزیه و تحلیل در R می‌باشد. در واقع، اکثر روال‌های آماری در زبان R نیازمند داده‌های ورودی از این دست است. اکنون به مثال زیر توجه کنید.

```
> mtcars
```

	mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb
Mazda RX4	21.0	6	160.0	110	3.90	2.620	16.46	0	1	4	4
Mazda RX4 Wag	21.0	6	160.0	110	3.90	2.875	17.02	0	1	4	4
Datsun 710	22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1
Hornet 4 Drive	21.4	6	258.0	110	3.08	3.215	19.44	1	0	3	1
Hornet Sportabout	18.7	8	360.0	175	3.15	3.440	17.02	0	0	3	2

این داده‌ها شامل اطلاعات خودروهای مختلف است. در جدول بالا هر سطر نماینده یک خودرو و ستون‌ها نمایش‌دهنده متغیرها است. در این مثال متغیر carb عدد کاربراتور را نشان می‌دهد.

ایجاد داده‌های چارچوب‌دار

برای ایجاد این نوع داده‌ها راه‌های مختلفی وجود دارد.

استفاده از تابع data.frame() می‌باشد.

```
> test<-matrix(rnorm(21),7,3)
> test<-data.frame(test)
> test
         X1         X2         X3
1  0.66277779 -0.3840960 -1.3595135
2 -0.04913578  0.8492189  2.0007778
3  0.34517674 -1.2079345  1.2743033
4 -0.15967494 -0.6256712  0.6585888
5  0.06225548 -0.3743871  2.7837282
6  0.39921940  0.5412310  0.1212971
7  0.41148286 -0.8607701 -0.0723364
```

زبان R به‌صورت خودکار اسامی ستون‌ها را X1, X2, X3 نامیده است. شما می‌توانید اسامی دلخواه خود را قرار دهید.

```
> names(test)<-c("Price","length","income")
> row.names(test)<-c("Ali","saeed","navid","babak","Majid","Amir","Hamid")
> test
         Price     length     income
Ali    0.66277779 -0.3840960 -1.3595135
saeed -0.04913578  0.8492189  2.0007778
navid  0.34517674 -1.2079345  1.2743033
babak -0.15967494 -0.6256712  0.6585888
Majid  0.06225548 -0.3743871  2.7837282
Amir   0.39921940  0.5412310  0.1212971
Hamid  0.41148286 -0.8607701 -0.0723364
```

```
> sum(test[,2])
[1] -2.062409
```

موجودیت‌های سری زمانی

در زبان R شیء سری‌های زمانی با تابع ts() ایجاد می‌گردد. دو مؤلفه در آن‌ها وجود دارد.

- داده‌ها، بردار یا ماتریسی از داده‌های عددی است که هر ستون یک سری زمانی مجزا را تشکیل می‌دهد.
- تاریخ داده‌ها، فواصل مساوی تاریخی است.

اکنون به مثال زیر توجه کنید.

```
> my.ts<-
ts(matrix(rnorm(36),ncol=3),start=c(1987),freq=12)
> my.ts
          Series 1   Series 2   Series 3
Jan 1987  1.9162493 -0.1661033 -1.9808024
Feb 1987 -0.5805433  1.8552607  0.7985123
Mar 1987  0.2521222  1.3434837  1.2004464
Apr 1987  0.3422510  2.5013974 -0.1498149
May 1987  0.2963408  0.2188443  0.1390155
Jun 1987  1.6654425 -0.7643525 -0.8574981
Jul 1987  0.1306895 -0.2848434  1.0944074
Aug 1987  0.4957888 -1.6637329 -0.1741922
Sep 1987 -0.2235258  0.5657969  0.6776202
Oct 1987 -0.9107982  1.0308249  0.3528864
Nov 1987  0.3451822 -0.7520748  0.4318638
Dec 1987  0.2092638 -0.5056582 -0.9681399
```

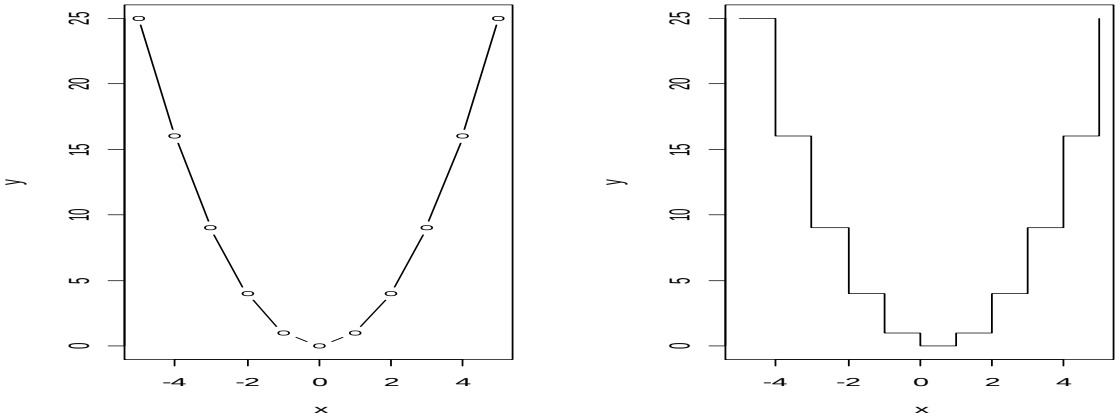
رسم نمودار

برای آشنایی با دستورات رسم نمودار در زبان R، تعدادی از آن‌ها ارایه خواهد شد.

توابع نموداری

در این قسمت تعدادی از توابع سطح بالای نموداری ملاحظه می‌گردد. تابعی که کاربرد فراوانی در ترسیم دارد، تابع plot() است.

```
> x<-seq(-5,5,1)
> y<-x^2
> par(mfrow=c(1,2))
> plot(x,y,type='b')
> plot(x,y,type='s')
```



تابع curve()

این تابع زبان R می‌تواند توابع پیوسته را روی یک فاصله معین رسم نماید. شکل کلی آن به صورت زیر است.

curve(expr, from, to, add = FALSE, ...)

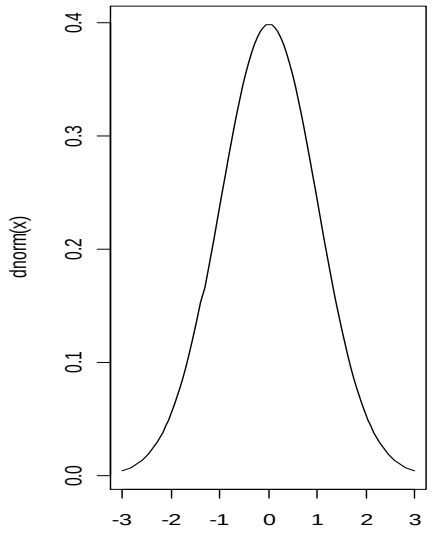
شرح آرگومان‌ها:

expr: عبارتی که بر حسب x نوشته شده است.

from, to: دامنه‌ای است که تابع باید روی آن رسم گردد.

add: منطقی است و اگر «TRUE» باشد شکل تابع به شکل حاضر اضافه می‌شود.

```
> curve(dnorm(x),from=-3,to=3)
```



توابع نموداری آماری

در زبان R می‌توان از توابع نمودار آماری استفاده نمود، که کاربرد فراوانی دارد.

hist(x): این تابع هیستوگرام ایجاد می‌کند.

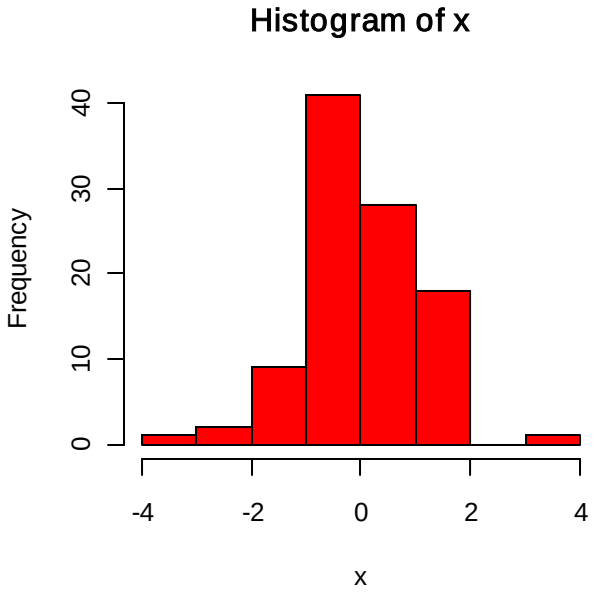
qqnorm(x): این تابع چندک‌ها را روی دو محور ایجاد می‌کند. که چندک‌های نرمال روی محور xها قرار دارد.

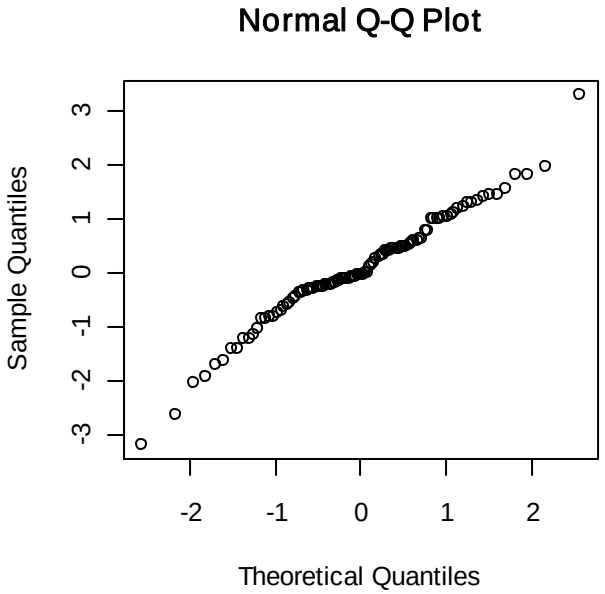
qqplot(x,y): این تابع چندک x را بر حسب y رسم می‌کند.

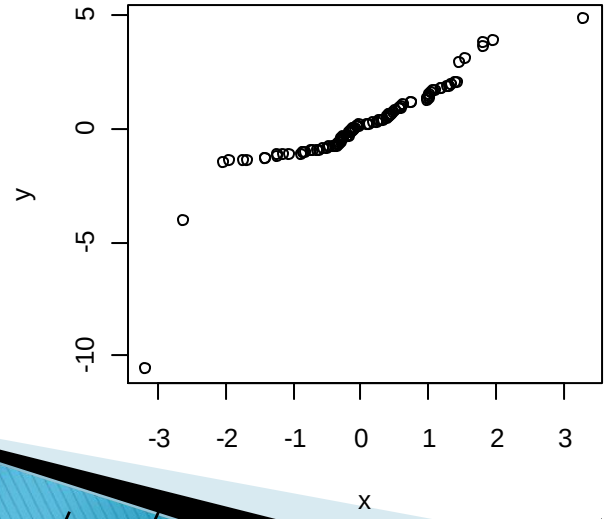
boxplot(x): این تابع نمودار box & whisker را ایجاد می‌نماید.

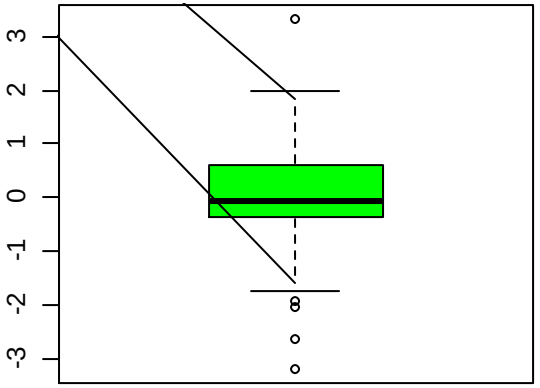
اکنون به مثال زیر توجه کنید.

```
> x<-rnorm(100)
> y<-rt(100,df=3)
> par(mfrow=c(2,2))
> hist(x,col=2)
> qqnorm(x)
> qqplot(x,y)
> boxplot(x,col="green")
```


Histogram of x
Frequency
0
10
20
30
40
-4
-2
0
2
4
x


Normal Q-Q Plot
Sample Quantiles
-3
-2
-1
0
1
2
3
-2
-1
0
1
2
Theoretical Quantiles


y
-10
-5
0
5
-3
-2
-1
0
1
2
3
x


-3
-2
-1
0
1
2
3

چند نمودار روی یک صفحه

با استفاده از پارامترهای mfrow و یا mfcol می‌توان روی یک صفحه چند نمودار را قرار داد. mf مخفف واژه multiframe است و row سطر و col ستون را بیان می‌کنند. هر دو پارامتر به شرح زیر تنظیم می‌شوند.

```
par(mfrow=c(r,k))
par(mfcol=c(r,k))
```

که در آن r تعداد سطرها و k تعداد ستون‌های صفحه‌ای که در آن نمودار رسم می‌گردد را نشان می‌دهد. پارامتر گرافیکی mfrow بیانگر چیدمان سطری و mfcol بیانگر چیدمان ستونی است.

تنظیم‌های خارجی

فهرست زیر، پارامترهای بیشتر و جزیی‌تر از نمودارها را نشان می‌دهد. برای مثال plot(x,y, col=2)

lwd: با این پارامتر می‌توان ضخامت خطوط نمودار را تعیین کرد.

lty: با این پارامتر می‌توان نوع خطوط مورد استفاده در نمودار را را تعیین کرد. مقدار این پارامتر می‌تواند عدد و یا کاراکتر باشد. برای مثال lty="dashed"

col: با این پارامتر می‌توان رنگ نمودار تعیین کرد. مقدار این پارامتر می‌تواند عدد و یا کاراکتر باشد. برای مثال col = "red". برای تغییر رنگ عنوان و زیرعنوان از col.main و col.sub، برای تغییر رنگ عناوین محورها از col.lab و بالاخره برای تغییر رنگ محورها از col.axis استفاده می‌شود.

font: مقدار عددی این پارامتر قلمِ نوشته‌های نمودار را معین می‌کند.

pch: با پارامتر pch[۳] می‌توان نشانه‌های نمودار را معین نمود. مثلاً نشانه دایره به مربع تبدیل شود.

xlab, ylab: با این پارامتر می‌توان اسامی محورهای مختصات را مشخص کرد. البته در پاره‌ای از توابع سطح بالا، این عمل خودبه‌خود انجام می‌شود.

xlim, ylim: با این پارامتر می‌توان مقادیر حداقل و حداکثر محورهای x و y را تعیین نمود. برای این کار می‌توان از دستورات xlim=c(low,high) و ylim=c(low,high) استفاده نمود، که در آن low مقدار حداقل و high مقدار حداکثر را نشان می‌دهد.

cex: با این پارامتر می‌توان نشانه‌ها و متون نمودار را بزرگتر نمود. اگر بخواهید اندازه عنوان را بزرگ کنید از cex.main و برای زیرعنوان از cex.sub استفاده کنید. برای بزرگتر نمودن اسامی محورها از cex.lab می‌توان کمک گرفت.

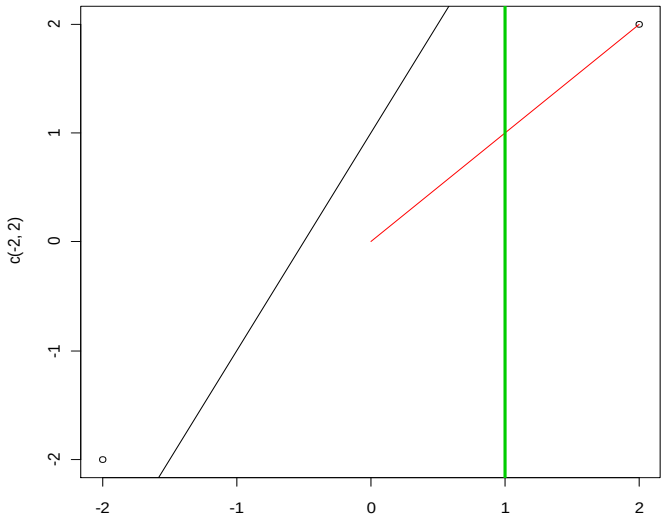
پاره‌ای از توابع سطح پایین

شما یک نمودار ایجاد می‌کنید، سپس می‌خواهید که بعضی از موارد را به آن اضافه کنید. این کار توسط توابع سطح پایین امکان‌پذیر است.

افزودن خطوط

توابع lines() و abline() برای افزودن خطوط به نمودار موجود استفاده می‌شود. تابع lines() نقاط بردار ورودی را به هم وصل می‌کند. تابع abline() خطوط راست با شیب و عرض از مبداء معین را ترسیم می‌کند. به مثال زیر توجه کنید.

```
plot(c(-2,2),c(-2,2))
> lines(c(0,2),c(0,2), col="red")
> abline(a=1, b=2)
> abline(v=1,col=3,lwd=3)
```



آمار و احتمال

زبان R شامل تعداد زیادی تابع برای محاسبات آماری، تحلیل داده‌ها و مدل‌سازی آماری است. علاوه بر آن‌ها تعداد قابل ملاحظه‌ای تابع در packageها وجود دارد. در اینجا روی تعداد محدودی از آن‌ها بحث خواهد شد.

توابع پایه آماری

توابع با کاربری زیاد

فهرست توابع در زبان R بسیار زیاد است. فهرست زیر دارای کابرد فراوان می‌باشد.

sum(x): مجموع عناصر x را می‌دهد.

cumsum(x): جمع تجمعی عناصر x را می‌دهد.

prod(x): حاصل‌ضرب عناصر x را می‌دهد.

max(x): حداکثر عناصر x را می‌دهد.

min(x): حداقل عناصر x را می‌دهد.

which.max(x): اندیس بزرگترین عنصر x را می‌دهد.

which.min(x): اندیس کوچکترین عنصر x را می‌دهد.

range(x): این تابع معادل c(min(x), max(x)) است.

length(x): تعداد عناصر x را می‌دهد.

mean(x): میانگین عناصر x را می‌دهد.

median(x): میانه عناصر x را می‌دهد.

var(x): واریانس عناصر x را می‌دهد.

sd(x): انحراف معیار عناصر x را می‌دهد.

cor(x): ماتریس همبستگی را محاسبه می‌کند، اگر x یک ماتریس باشد.

cor(x,y): ضریب همبستگی خطی را بین x و y بدست می‌دهد.

chisq.test(x): آزمون نکویی برازش به روش خی دو است.

ks.test(x): آزمون نکویی برازش به روش کلموگروف-اسمیرنف است.

t.test(x): آزمون tاستودنت برای یک یا دو نمونه است.

var.test(x,y): آزمون برابری واریانس x و y است.

توزیع‌های احتمالی و اعداد تصادفی

بیشتر توابع احتمالی در زبان R گذاشته شده است. هر تابع دارای چهار شکل متفاوت است، که به صورت زیر هستند.

- دستور dfunc(x, $\cdots$)، عرض تابع را در نقطه x نشان می‌دهد.
- دستور pfunc(x, $\cdots$)، مقدار احتمال تجمعی را تا نقطه x نشان می‌دهد.
- دستور qfunc(p, $\cdots$) مقدار چندک تابع را به ازای $0 < p < 1$ نشان می‌دهد.
- دستور rfunc(x, $\cdots$)، نمونه تصادفی از تابع را شبیه‌سازی می‌کند.

برای بدست آوردن احتمال تجمعی $F_X(x) = \Pr(X \leq x)$ یک توزیع به مثال زیر توجه کنید.

```
> pnorm(1.96)
[1] 0.9750021
```

اکنون می‌توان با یک امکان اضافی احتمال $1 - F_X(x) = \Pr(X > x)$ را نیز بدست آورد.

برای بدست آوردن مقادیر بحرانی و یا P-value در آزمون آماری مورد استفاده قرار می‌گیرد.

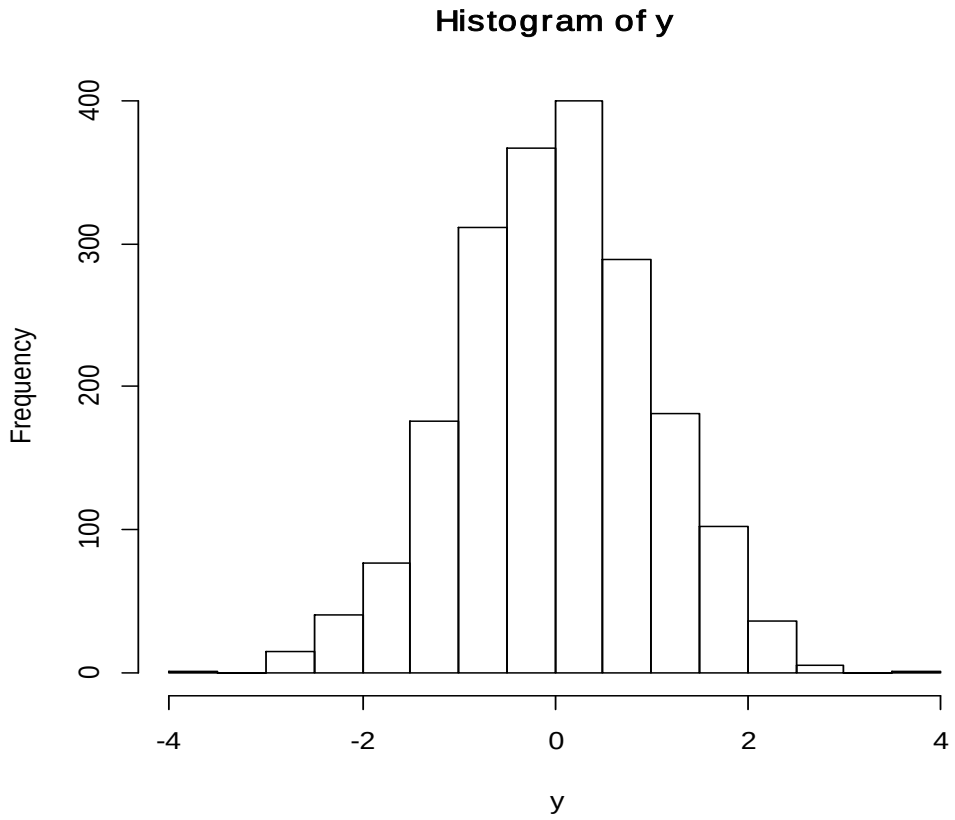
مقدار P-value برای آزمون $\chi^2 = 3.84$ با df=1 برابر است با:

```
> 1-pchisq(3.84,1)
[1] 0.05004352
```

هیستوگرام

در آمار و احتمال هیستوگرام، نمودار معروفی است

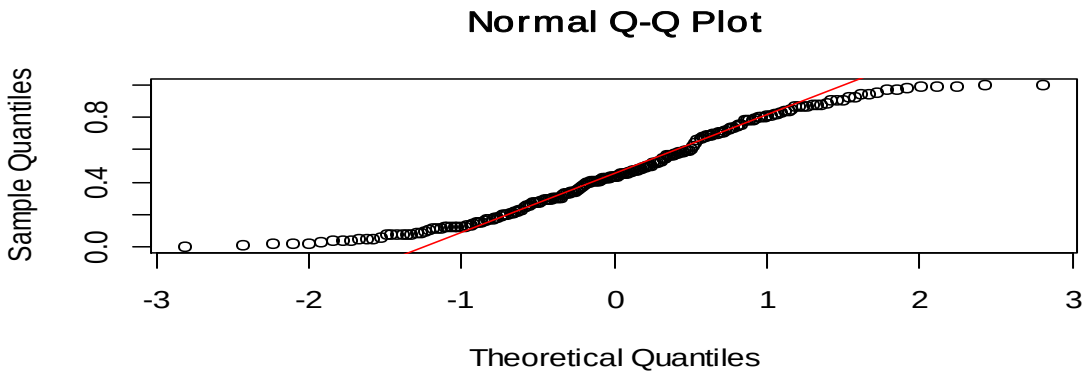
```
> y<-rnorm(2000)
> hist(y)
```

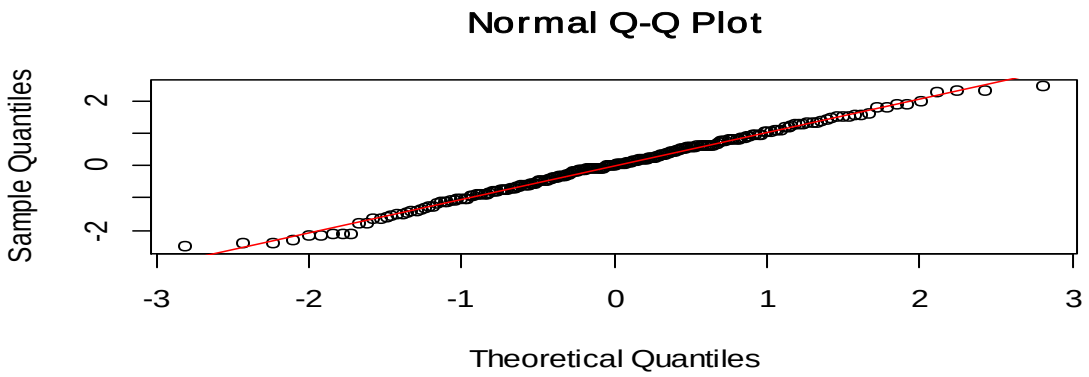


توابع گرافیکی qqnorm()، qqline() و qqplot()

می‌توان نرمال بودن یک سری داده را با تابع qqnorm() نشان داد. اگر داده‌ها نرمال باشند، آنگاه یک خط راست را نمایش می‌دهند. ترسیم خط راست از تابع qqline() حاصل می‌شود که برای مقایسه به‌کار می‌رود. به مثال زیر توجه کنید.

```
x<-runif(200)
> y<-rnorm(200)
> par(mfrow=c(3,1))
> qqnorm(x)
> qqline(x,col="red")
> qqnorm(y)
> qqline(y, col="red")
```

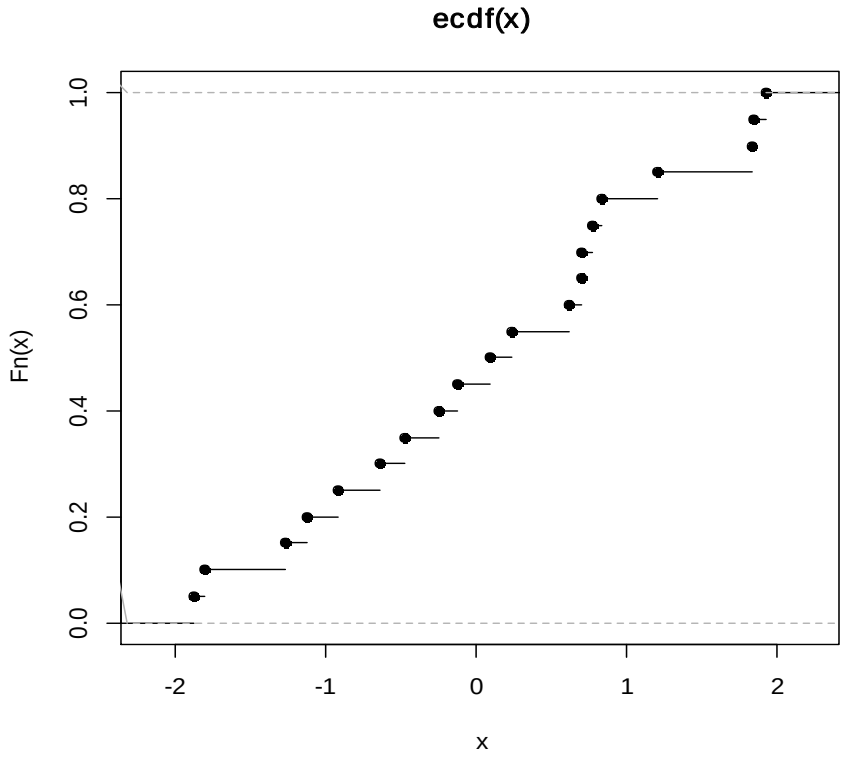




تابع ecdf()

همانطور که قبلاً ملاحظه شد. می‌توان تابع توزیع تجمعی را محاسبه و ترسیم کرد. اما تابعی وجود دارد که ecdf() نامیده می‌شود و توزیع تجمعی تجربی را بدست می‌دهد. به مثال زیر توجه کنید.

```
> x<-rnorm(20)
> plot(ecdf(x))
```



نمونه‌گیری تصادفی

تجربه‌های ساده احتمالی مانند انتخاب تصادفی اعداد از 1 تا 100 و کشیدن سه توپ از یک کیسه را می‌توان توسط زبان R شبیه‌سازی نمود. تابع ()sample به صورت کلی زیر است.

```
sample(x, size, replace = FALSE, prob = NULL)
```

که در آن،

x: بردار مورد نظر است که می‌تواند عدد و یا کاراکتر باشد.

size: تعدادی که باید انتخاب گردد.

replace: نمونگیری بدون جایگذاری و یا با جایگذاری انجام گردد.

prob: بردار اختیاری است که می‌تواند به انتخاب نمونه‌ها وزن بدهد.

مثال‌ها:

انتخاب یک عدد به تصادف از 1 تا 100

```
> sample(1:100, 1)
[1] 59
```

پرتاب 10 بار یک تاس

```
> sample(1:6,10, replace=T)
 [1] 5 1 5 2 4 2 6 6 6 2
```

درکیسه‌ای 8 مهره قرمز، 4 مهره آبی و 3 مهره زرد وجود دارد. اکنون 6 مهره به تصادف و بدون جایگذاری انتخاب کنید.

```
> ur=c(rep("red",8),rep("blue",4),rep("yellow",3))
> sample(ur,6,replace=F)
[1] "yellow" "yellow" "red"    "red"    "red"    "red"
```

برآورد پارامترهای یک تابع احتمال

همان‌طور که می‌دانید به روش‌های مختلف می‌توان پارامترهای یک تابع احتمال را برآورد نمود. از روش‌هایی مانند روش گشتاورها و روش حداکثر درستنمایی[1] می‌توان نام برد و به آنها اشاره نمود.

در زبان R بسته‌ای تحت عنوان MASS وجود دارد که پارامترهای تابع احتمال را به روش حداکثر درستنمایی برآورد می‌کند. در این بسته تابعی به نام fitdistr() موجود است که دارای دو آرگومان ضروری است: اول داده‌هایی که قرار است بر آنها تابعی برازش یابد و دوم تابع احتمال مورد نظر که باید در داخل کوتیشن قرار گیرد.

مثال: در اینجا ابتدا یک سری اعداد تصادفی با توزیع گاما (با پارامترهای مشخص) تولید می‌شود. اکنون با فرض تابع احتمال گاما مجدداً پارامترهای تابع احتمال گاما که بر داده‌های مذکور برازش یافته‌است، برآورد می‌گردد.

```
> x<-rgamma(500,8.5,2.5)
> fitdistr(x,"gamma")
     shape       rate
  8.5924296   2.5304502
 (0.5332122) (0.1617056)
```

روش های آماری

زبان R میزبان روش‌های آماری و آزمون فرض است. در اینجا روی مشهورترین آن‌ها تاکید می‌گردد.

آزمون یک و دو طرفه t

تابع اصلی برای این نوع آزمون t.test() است. فرض‌ها در اینجا با تابع چگالی احتمال t آزمون می‌شود.

```
t.test(x, y = NULL, alternative = c("two.sided", "less", "greater"), mu = 0,
paired = FALSE, var.equal = FALSE, conf.level = 0.95)
```

که در آن:

x,y: بردارهای عددی هستند. اگر y داده نشود، آنگاه یک آزمون ساده موجود است.

alternative: فرض جایگزین توسط رشته‌ای از کاراکترها بیان می‌گردد. پیش‌فرض آن "two.sided"، "greater" و "less". شما می‌توانید از حروف اول آن نیز برای به‌کارگیری استفاده کنید.

mu: عددی که مقدار درست میانگین را نشان می‌دهد (یا اختلاف میانگین‌ها را بیان می‌کند، اگر دو نمونه وجود داشته باشد). پیش‌فرض برابر صفر است.

paired: نشانه منطقی است اگر بخواهید آزمون دوتایی داشته باشید.

var.equal: متغیر منطقی است. اگر درست (T) باشد واریانس‌ها با هم برابرند و پیش‌فرض آن (F) است.

conf.level: سطح اطمینان (پیش‌فرض 95٪) برای برآورد فاصله‌ای میانگین بر حسب فرض جایگزین

مثال ۲: فرض کنید یک افزودنی به سوخت خودرو اضافه می‌گردد. اکنون سوال این است که آیا از مصرف آن‌ها کاسته می‌شود؟ برای این کار شش خودرو از آن استفاده می‌کند و شش خودروی دیگر از آن استفاده نمی‌کند. متغیر mpg (مایل برگالن) اندازه‌گیری شده است که به‌صورت زیر است.

Car	1	2	3	4	5	6
mpg w additive	24.6	18.9	27.3	25.2	22.0	30.9
mpg w o additive	23.8	17.7	26.6	25.1	21.6	29.6

```
> add<-
c(24.6,18.9,27.3,25.2,22.0,30.9)
> noadd<-
c(23.8,17.7,26.6,25.1,21.6,29.6)
> var.test(add,noadd)

        F test to compare two variances

data:  add and noadd
F = 1.0179, num df = 5, denom df = 5, p-value = 0.985
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 0.1424311 7.2740660
sample estimates:
ratio of variances
          1.017887
```

```
> t.test(add,noadd,paired=T,alt="greater")

        Paired t-test

data:  add and noadd
t = 3.9994, df = 5, p-value = 0.005165
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 0.3721225       Inf
sample estimates:
mean of the differences
                  0.75
```

اکنون به یک مثال از آزمون مربع خی توجه فرمایید. فرض کنید که یک تاس 300 بار پرتاب شود و نتایج زیر حاصل گردد.

وجه تاس	1	2	3	4	5	6
فراوانی	43	49	56	45	66	41

اکنون سوال این است که این تاس بی طرف است. یعنی احتمال آمدن هر وجه تاس $\frac{1}{6}$ است؟ برای این کار از آزمون مربع خی استفاده می شود. به گُدهای زیر توجه کنید.

```
> count<-c(43,49,56,45,66,41)
> pr=rep(1/6,6)
> chisq.test(count,p=pr)

        Chi-squared test for given probabilities

data:  count
X-squared = 8.96, df = 5, p-value = 0.1107
```

مدل‌های رگرسیون

زبان R دارای روتین‌های زیادی برای برازش مدل‌های آماری است. عموماً این مدل‌ها از طریق فراخواندن توابعی مثل lm, glm, ... عمل می‌کنند. شکل کلی یک مدل آماری برازش به صورت زیر است.

respose ~ expression

مدل‌های رگرسیون خطی

موجودیت‌های فرمول

زبان R یک مدل خطی را به صورت زیر برازش می‌دهد.

$$y = \beta_0 + \beta_1 x_1 + \cdots + \beta_p x_p + \varepsilon$$

که $\beta = (\beta_0, \beta_1, \cdots, \beta_p)$ عرض از مبداء و ضرایب رگرسیون را نشان می‌دهد. جمله خطا یعنی ε غالباً دارای توزیع نرمال با میانگین صفر و واریانس σ_ε^2 است.
برای رگرسیون با دو متغیر می‌توان با استفاده از تابع lm() و فرمول زیر استفاده نمود.

y ~ x1+x2

که عبارت فوق به منزله فرمول $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$ است.

به طور پیش‌فرض، زبان R شامل عرض از مبداء می‌شود. اما رابطه فوق را به صورت زیر بنویسید عرض از مبداء حذف می‌گردد.

```
y ~ -1+x1+x2
```

توجه داشته باشید که عملگرهای : ,\ ,^ ,- ,* دارای معانی خاصی در رگرسیون خطی هستند. به مثال‌های زیر توجه کنید.

```
y ~ x1+x2+x1:x2
```

که عبارت فوق به منزله فرمول $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \beta_{12} x_1 x_2 + \varepsilon$ است.

استفاده از عملگر ^ جمله‌های یگانه و دودویی را به ترتیب ایجاد می‌کند.

```
y ~ (x1+x2+x1:x2)^2
```

فرمول بالا معادل رابطه زیر است.

```
y ~ x1+x2+x3+x1:x2+x2:x3+x1:x3
```

عملگر - جمله و یا جملاتی را از معادله رگرسیون حذف می‌کند. همان‌طورکه قبلاً ملاحظه شد 1- عرض از مبداء را حذف نمود. به مثال زیر توجه کنید.

```
y ~ (x1+x2+x1:x2)^2 - x2:x3
```

فرمول بالا معادل رابطه زیر است.

```
y ~ x1+x2+x3+x1:x2+x1:x3
```

تابع I ضریبی را برای متغیر تکرار می‌کند. به مثال زیر توجه کنید.

```
y ~ I(x1+x2)
```

که عبارت فوق به منزله فرمول $y = \beta_0 + \beta(x_1 + x_2) + \varepsilon$ است. ضمناً اگر بخواهید متغیر x2 در مدل تغییر کند. مثلاً در دو ضرب شود. اگر به صورت زیر عمل کنید اشتباه کرده‌اید.

```
y ~ x1+2*x2
```

برای درست شدن مطلب به صورت زیر باید عمل نمود.

```
y ~ x1+I(2*x2)
```

اگر مدل مورد نظر به‌صورت $y = \beta_0 + \beta_1 x + \beta_2 x^2$ باشد، آنگاه فرمول آن در زبان R به شرح زیر است.

```
y ~ poly(x, 2)
```

و نه به‌صورت زیر:

```
y ~ x+x2^2
```

توابع مدل‌سازی

مدل‌های رگرسیون خطی کاربرد وسیعی برای بیان روابط خطی بین متغیرها دارد. توابع زیادی بلای برازش و تحلیل رگرسیون خطی وجود دارد. تابع اصلی برای این کار lm() است که پاره‌ای از آرآرگومان‌های آن‌ها به شرح زیر است.

```
lm(formula, data, weights, subset, na.action)
```

به داده‌های زیر توجه کنید.

```
> y<-read.table("G:\\Seminar
R\\reg.txt",header=T)
> y
    x1    x2
1  -1.06 -1.08
2  -0.81 -1.02
3  -0.48 -0.39
4  -0.42 -0.48
5  -0.30 -0.58
6  -0.35 -0.42
7  -0.31 -0.05
8  -0.18 -0.33
9  -0.20  0.51
10 -0.11 -0.53
11 -0.09 -0.47
12  0.16  0.01
13  0.45  0.39
14  0.53  0.11
15  0.67  0.52
16  0.80  0.34
17  0.87  1.08
18  0.92  1.21
```

ابتدا ضریب همبستگی و آزمون مربوط به آن را ملاحظه خواهید نمود. کدهای زیر آن را خوانده موارد مربوط به آن‌ها را بررسی می‌کنند.

```
> cor.test(y$x1,y$x2)

        Pearson's product-moment correlation

data:  y$x1 and y$x2
t = 7.6264, df = 16, p-value = 1.026e-06
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.7138577 0.9568435
sample estimates:
      cor
0.8855821

> lm(y$x1~y$x2)

Call:
lm(formula = y$x1 ~ y$x2)

Coefficients:
(Intercept)        y$x2
     0.0575       0.8008
```

تابع summary() برای بدست آوردن پاره‌ای از اطلاعات اضافی از مدل برازش یافته نظیر مقادیر t، خطای استاندارد و همبستگی بین پارامترها مفید است. به مثال زیر توجه کنید.

```
> summary(lm(y$x1~y$x2))

Call:
lm(formula = y$x1 ~ y$x2)

Residuals:
     Min       1Q   Median       3Q      Max
-0.66591 -0.10313 -0.01195  0.17380  0.47023

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.05750    0.06621   0.868    0.398
y$x2         0.80081    0.10500   7.626 1.03e-06 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2794 on 16 degrees of freedom
Multiple R-squared: 0.7843,     Adjusted R-squared: 0.7708
F-statistic: 58.16 on 1 and 16 DF,  p-value: 1.026e-06
```

با تابع confint() می‌توان برآورد فاصله‌ای ضرایب معادله رگرسیون را محاسبه نمود.

```
> confint(lm(y$x1~y$x2))
                2.5 %    97.5 %
(Intercept) -0.08285861 0.1978535
y$x2         0.57820813 1.0234092
```

```
>
predict(lm(y$x1~y$x2),new=data.frame(x2=c(0.1,0.2)),int="conf")
          fit          lwr         upr
1  -0.80737591 -1.0728560234 -0.54189579
2  -0.75932739 -1.0135439623 -0.50511081
3  -0.25481792 -0.4119892506 -0.09764660
4  -0.32689070 -0.4942166362 -0.15956477
5  -0.40697157 -0.5875282243 -0.22641492
6  -0.27884218 -0.4391918478 -0.11849252
7   0.01745702 -0.1221813294  0.15709538
8  -0.20676940 -0.3582686575 -0.05527015
9   0.46590988  0.2764333567  0.65538640
10 -0.36693114 -0.5406419858 -0.19322029
11 -0.31888262 -0.4849916175 -0.15277362
12  0.06550554 -0.0750993937  0.20611048
13  0.36981284  0.1972722873  0.54235339
14  0.14558641  0.0006242462  0.29054858
15  0.47391797  0.2829292365  0.66490669
16  0.32977240  0.1635292220  0.49601559
17  0.92237082  0.6316604203  1.21308122
18  1.02647595  0.7100767704  1.34287512
Warning message:
'newdata' had 2 rows but variable(s) found have 18 rows
```

منابع:

1-**“R language manual in Farsi”** by Saeed Mousavi (PDF, 2012-10-08, 179 pages).
2-**“SNA in R Tutorial in Farsi”** by Mohsen Raeesi (PDF, 2011-09-01).
3-**“Special topics with R in Farsi”** by Saeed Mousavi (PDF, 2012-09-26, 62 pages).
4-محاسبات آماری با R، مجید رضایی و همکاران، دانشگاه بیرجند.

باتشکر