به نام خداوند جان و خرد

انتشارات
۸۲۵

کاربرد نرم‌افزار R در آنالیز داده‌های ژنتیکی

دکتر محمد تیموریان
دکتر محمدمهدی شریعتی
عضو هیئت علمی دانشگاه فردوسی مشهد

سرشناسه: تیموریان، محمد، ۱۳۶۱-
عنوان و نام پدیدآور: کاربرد نرم‌افزار R در آنالیز داده‌های ژنتیکی/ محمد تیموریان، محمدمهدی شریعتی؛ ویراستار ادبی هانیه اسدپورفعال مشهد.
مشخصات نشر: مشهد: دانشگاه فردوسی مشهد، انتشارات، ۱۴۰۱.
مشخصات ظاهری: ۱۹۲ ص.: جدول، نمودار.
فروست: انتشارات دانشگاه فردوسی مشهد؛ ۸۲۵.
شابک: ISBN: 978-964-386-517-7
وضعیت فهرست‌نویسی: فیپا.
یادداشت: کتابنامه: ص. [۱۸۷] - ۱۹۰. نمایه.
موضوع: آر (زبان برنامه‌نویسی کامپیوتر)
R (Computer program language)
ژنتیک -- روش‌های آماری
Genetics -- Statistical methods
شناسه افزوده: شریعتی، محمدمهدی، ۱۳۵۲ -
شناسه افزوده: دانشگاه فردوسی مشهد، انتشارات.
رده‌بندی کنگره: QA۲۷۶/۴۵
رده‌بندی دیویی: ۵۱۹/۵۰۲۸۵۵۱۳۳
شمارهٔ کتابشناسی ملّی: ۸۷۴۶۷۹۳

کاربرد نرم‌افزار R در آنالیز داده‌های ژنتیکی

پدیدآورندگان: دکتر محمد تیموریان؛ دکتر محمدمهدی شریعتی
ویراستار ادبی: هانیه اسدپور فعال مشهد
مشخصات: وزیری، ۱۰۰ نسخه، چاپ دوم، پاییز ۱۴۰۴ (اوّل، ۱۴۰۱)
چاپ و صحافی: همیار
بها : ۲/۳۰۰/۰۰۰ ریال

انتشارات
۸۲۵

مراکز پخش:

فروشگاه و نمایشگاه کتاب پردیس: مشهد، میدان آزادی، دانشگاه فردوسی مشهد، جنب سلف یاس
تلفن: ۳۸۸۰۲۶۶۶- ۳۸۸۳۳۷۲۷ (۰۵۱)
مؤسّسهٔ کتابیران: تهران، میدان انقلاب، خیابان کارگر جنوبی، بین روانمهر و وحید نظری، بن‌بست گشتاسب، پلاک ۸ تلفن: ۶۶۴۸۴۷۱۵ (۰۲۱)
مؤسّسهٔ دانشیران: تهران، خیابان انقلاب، خیابان منیری جاوید (اردیبهشت) نبش خیابان نظری، شمارهٔ ۱۴۲
تلفکس: ۶۶۴۰۰۲۲۰ – ۶۶۴۰۰۱۴۴ (۰۲۱)

http://press.um.ac.ir Email: press@um.ac.ir

تقدیم به:

همسر و فرزندانم، سنا و سایدا
محمد تیموریان

تقدیم به:

شادروان دکتر فریدون افتخار شاهرودی
محمدمهدی شریعتی

فهرست مطالب

پیشگفتار

ستایش خدای را که باران رحمت بی‌حسابش همه را رسیده و خوان نعمت بی‌دریغش همه‌جا کشیده و درود بر رسول گرامی‌اش که تاریکی جهل و نادانی را با نور جمال معرفت خود زائل کرد.

در دهه‌های اخیر، مطالعات زیستی و ژنتیکی با سرعت بسیار زیادی درحال انجام بوده است. با پیشرفت‌های زیست‌شناسی مولکولی و ژنتیک و همچنین گسترش توالی‌یابی ژنوم‌های مختلف، تجزیه‌وتحلیل داده‌های ژنتیکی و فهم مسائل مرتبط با آن چالش بزرگی را برای زیست شناسان ایجاد کرده است. بررسی و آنالیز توالی‌های ژنومی گونه‌های مختلف و داده‌های نسل سوم، نیازمند نرم‌افزارهای قدرتمند آماری و زبان‌های مختلف برنامه‌نویسی است. نرم‌افزار R محیط بسیار مناسبی برای محاسبات و آنالیزهای آماری در بسیاری از رشته‌هاست و به دلیل رایگان بودن و نصب بر روی اکثر سیستم‌ها در سال‌های اخیر توجّه کاربران زیادی را به خود جلب کرده است. همچنین امکان نصب بسته‌ها یا کتابخانه‌های متنوع در رشته‌های مختلف، قدرت زیادی به این نرم‌افزار داده است. با توجه به پیشرفت‌های صورت‌گرفته در علم ژنتیک و افزایش داده‌های مختلف این رشته در کشور عزیزمان، ایران، بر آن شدیم که آنالیز داده‌های ژنتیکی را به‌صورت مفید و مختصر با استفاده از نرم‌افزار پرکاربرد R در قالب یک کتاب بررسی کنیم. در این کتاب ابتدا نرم‌افزار آماری R به‌صورت کلی و مقدّماتی و با تکیه بر داده‌های ژنتیکی بررسی شده است. انواع داده‌ها و کار با توالی‌های ژنتیکی به‌عنوان مهم‌ترین بخش آنالیزهای ژنتیکی، توابع ریاضی و آماری مرتبط با ژنتیک، آزمون‌های آماری و استنباطی، انواع نمودارهای پایه و کاربردی در رشتهٔ ژنتیکی موردبحث قرار گرفته و برآورد اثرات ثابت و پیش‌بینی اثرات تصادفی با روش‌های مختلف مطرح شده‌اند. آنالیزهای ارتباط ژنی، پویش کل ژنوم و پیش‌بینی ژنومی با مثال‌های متعدد تشریح شده‌اند و در پایان مهم‌ترین بخش از آنالیزهای نوین ژنتیکی و ژنومی در سالیان اخیر به‌عنوان آنالیز داده‌های نسل سوم بیان ژن به‌صورت مفصّل و در قالب چند فصل به‌طور کامل و جامع بررسی شده است و انواع کتابخانه‌های مهم و کاربردی در این مباحث به‌صورت عملی بررسی شده‌اند. در پایان، از همهٔ عزیزان تقاضا داریم ما را درجهت بهبود هرچه بهتر این کتاب یاری کنند.

محمد تیموریان- محمدمهدی شریعتی

۱

کلّیات نرم‌افزار R

۱-۱ نصب نرم‌افزار در ویندوز

بعد از دانلود نسخهٔ جدید فایل اجرایی از سایت[1] و نصب آن بر روی کامپیوتر، آیکون نرم‌افزار به‌شکل حرف R ظاهر می‌شود و با کلیک کردن بر روی آن صفحهٔ موسوم به Console باز می‌شود. جهت کار کردن با کدهای طولانی می‌توان از ویرایشگر داخلی یا خارجی نرم‌افزار استفاده کرد تا دستورات به‌صورت گروهی اجرا شود. برای استفاده از ویرایشگر داخلی، از منوی File گزینهٔ New script انتخاب می‌شود که علاوه‌بر اجرای گروهی کدهای موردنظر، امکان ذخیره و بازیابی کدها را نیز میسّر می‌کند. برای اجرای دستور یا دستورات سند[2] موردنظر از کلید Ctrl به‌همراه کلید R یا Enter استفاده می‌شود. یک سند درواقع یک فایل متنی شامل کدهای R برای یک آنالیز کامل است. از مهم‌ترین ویرایشگرهای خارجی، RStudio است که پس از نصب R باید جداگانه از سایت نرم‌افزار[3] دانلود و نصب شود. با توجه به ابعاد بالای داده‌های زیستی، آنالیز این داده‌ها در نرم‌افزار R با سرعت پایینی انجام می‌شود. یکی از گزینه‌های مناسب برای افزایش سرعت آنالیزها، استفاده از نسخهٔ ارتقایافته، یعنی Revolution R می‌باشد.

۱-۲ نصب نرم‌افزار در لینوکس

برای نصب نرم‌افزار R در محیط اوبونتوی لینوکس می‌توان از قسمت Ubuntu Software عبارت r-base را جست‌وجو و نصب یا دستورات زیر را در ترمینال اجرا کرد. برای بستن R در ترمینال از دستور ()q استفاده می‌شود.

1. https://cran.r-project.org/
2. Script
3. https://www.rstudio.com/products/rstudio/download/

```
$ sudo apt-get update
$ sudo apt-get install r-base
$ R
```

برای نصب RStudio می‌توان با انتخاب نسخهٔ اوبونتو از قسمت Download RStudio Desktop در سایت نرم‌افزار، ویرایشگر خارجی را دانلود و نصب کرد. جست‌وجو و نصب RKWard ازطریق Ubuntu Software گزینهٔ جایگزین دیگری برای R می‌باشد.

۱-۳ نصب کتابخانه

در محیط R می‌توان از منوی packages گزینه‌های مختلف نصب کتابخانه را انتخاب کرد. از دستور زیر برای دانلود و نصب کتابخانه نیز استفاده می‌شود. در این دستور گزینهٔ lib="" برای تعیین محل ذخیرهٔ کتابخانه و گزینهٔ dependencies=TRUE برای دریافت کتابخانه‌های مرتبط استفاده می‌شوند.

```
> install.packages("MASS")
```

در RStudio نیز می‌توان از منوی tools گزینه install packages را انتخاب کرد. در قسمتinstsll from یک مخزن[1] یا یک فایل فشرده[2] انتخاب می‌شود. درصورت انتخاب مخزن در دو قسمت بعدی نام کتابخانه و محل ذخیره مشخص می‌شود. درصورت انتخاب فایل فشرده در قسمت بعدی، آدرس فایل فشردهٔ ذخیره‌شده مشخص می‌شود.

۱-۳-۱ مخزن Bioconductor

مخزن[3] اصلی نرم‌افزار R و کتابخانه‌های مرتبط با آن CRAN می‌باشد. علاوه‌بر این مخزن اصلی، بیوکانداکتور[4] یکی از مخزن‌هایی است که حاوی تعداد زیادی کتابخانهٔ مرتبط با داده‌های ژنتیکی می‌باشد. برای نصب کتابخانه‌های این مخزن یا مخازن دیگر R غیر از مخزن اصلی می‌توان با تابع setRepositories لیست مخازن را مشاهده کرد و با وارد کردن شمارهٔ مخازن، آن‌ها را نیز به حالت پیش‌فرض نرم‌افزارْ اضافه و سپس با دستور ()install.packages کتابخانه را نصب کرد.

```
> setRepositories()
> install.packages("GeneticsPed")
```

همچنین برای نصب کتابخانه‌های مخزن بیوکانداکتور از دستورات زیر استفاده می‌شود.

```
> if (!requireNamespace("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
> BiocManager::install("GeneticsPed")
```

1. Repository
2. Package Archive File
3. Repository
4. Bioconductor

۱-۳-۲ مخزن گیت‌هاب

با توجّه به ذخیرهٔ مجموعه‌داده‌های بسیاری از پروژه‌ها در مخزن گیت‌هاب[1]، این مخزن در آنالیزهای ژنتیکی کاربرد زیادی دارد. برای دانلود فایل داده‌های خاص از این مخزن از دستور download کتابخانهٔ downloader استفاده می‌شود. برای نصب کتابخانه‌های گیت‌هاب نیاز به نصب کتابخانهٔ devtools می‌باشد. درصورتی‌که در محیط ویندوز اجرا شود، نیاز به نصب نرم‌افزار Rtools نیز می‌باشد.

```
> setwd(choose.dir()); library(downloader)
> url <-
https://raw.githubusercontent.com/genomicsclass/dagdata/master/inst/
extdata/femaleMiceWeights.csv"
> filename <-
"femaleMiceWeights.csv";download(url,destfile=filename)
> dat <- read.csv(filename)
```

۱-۳-۳ کتابخانهٔ Rcmdr

بستهٔ نرم‌افزاری Rcmdr با ارائهٔ Commander از کتابخانه‌هایی است که گزینه‌های متنوعی برای فراخوانی داده‌ها با فرمت‌های مختلف، مشاهده، ویرایش و آنالیز آن‌ها را دارا می‌باشد. مشابه سایر کتابخانه‌ها با دستور install.packages("Rcmdr") نصب و با دستور library("Rcmdr") فراخوانی می‌شود. درصورت بسته شدن کتابخانه، مجدّداً با دستور Commander() فراخوانی می‌شود. منوی Data در این کتابخانه، گزینه‌هایی برای ورود و ویرایش داده‌ها، منوی Statistics گزینه‌هایی برای آنالیزهای آماری و منوی Graphs گزینه‌هایی برای نمودارها ارائه می‌دهند.

۱-۴ نکات کلی

- نرم‌افزار R بین حروف کوچک و بزرگ تفکیک قائل می‌شود و اغلب به فاصله حساس نیست.
- دستور Sys.time() تاریخ و ساعت را مشخص می‌کند.
- دستور options(digits=...) تعداد ارقام اعشاری را تغییر می‌دهد.
- برای تغییر علامت نشانگر از حالت پیش‌فرض > به R> از دستور زیر استفاده می‌شود.

```
options(prompt = "R>"
```

- تابع sessionInf() نسخهٔ R و کتابخانه‌های مورداستفاده را نشان می‌دهد.

```
> sessionInfo()
R version 4.0.2 (2020-06-22)
Platform: x86_64-w64-mingw32/x64 (64-bit)
...
```

1. Github

- برای به‌دست‌آوردن راهنمای یک تابع از دستور ()help یا ? استفاده می‌شود. درصورتی‌که یک تابع بدون آرگومان اجرا شود، کدهای تابع نمایش داده می‌شود.

```
> Help(lm)
> ?lm
> var
function (x, y = NULL, na.rm = FALSE, use)
{
...
```

- دستور apropos توابع دارای یک عبارت خاص را جست‌وجو می‌کند.

```
> apropos("boxplot")
[1] "boxplot"    "boxplot.default" "boxplot.matrix"  "boxplot.stats"
```

- برای نسبت دادن از عملگرهای =، ->، <- یا تابع assign استفاده می‌شود.

```
> x=c(1,2,3)
> x
[1] 1 2 3
> x <- c(1,2,3)
> x
[1] 1 2 3
> c(1,2,3) -> x
> x
[1] 1 2 3
> assign("y",c(1,2,3))
> y
[1] 1 2 3
```

- ساختار داده‌ها در R شامل بردار[1]، آرایه[2]، ماتریس[3]، فاکتور[4]، لیست[5] و چهارچوب داده[6] با دستور ()class و نوع داده‌ها شامل پنج نوع اصلی عدد حقیقی[7]، عدد صحیح[8]، کاراکتر[9]، منطقی[10] و مختلط[11] با دستور ()mode مشخص می‌شود. از توابع ()length، ()dim، ()dimnames، ()nrow و ()ncol به‌ترتیب برای تعیین طول یا تعداد، ابعاد، نام ابعاد، تعداد ردیف و تعداد ستون استفاده می‌شود.
- علامت NA برای داده‌های گم‌شده، NaN برای موارد تعریف‌نشده، INf برای بی‌نهایت و NULL برای ساختار داده‌ای که وجود ندارد، به‌کار می‌رود.
- در محیط R می‌توان چند دستور را در یک سطر نوشت. دستورات با علامت ; از هم جدا می‌شوند.

```
> a=1:3 ; b=2:4 ;a*b
[1]  2  6 12
```

1. Vector
2. Array
3. Matrix
4. Factor
5. List
6. Data.Frame
7. Numeric
8. Integer
9. Character
10. Logical
11. Complex

- با دستور ls() می‌توان لیست اشیای محیط کاری[1] را مشخص کرد. از دستور ls(pat=) برای مشاهدهٔ متغیرهای دارای الگوی خاص در لیست متغیرهای تعریف‌شده استفاده می‌شود. با دستور rm() و نوشتن نام شیء یا اشیای موردنظر می‌توان آن‌ها را از محیط کاری حذف کرد. برای شروع یک آنالیز، بهتر است محیط کاری با دستور rm(list=ls()) از آنالیزهای قبلی پاک شود تا تداخلی با دستورات قبلی نداشته باشد.

```
> ls(pat="data")
 [1] "data"      "data2"     "data3"
```

- تابع grep رشته یا الگوی خاصی را در بردار کاراکترها جست‌وجو و مشخص می‌کند. گزینهٔ ignor.case=T جست‌وجو را به حروف بزرگ و کوچک بسط می‌دهد.

```
> words=c("Foo", "Bar", "Baz", "One", "Two", "bar")
> grep("Ba", words, value=FALSE)
[1] 2 3
> grep("Ba", words, value=TRUE)
[1] "Bar" "Baz"
> grep("Ba", words, value=TRUE,ignore.case=TRUE)
[1] "Bar" "Baz" "bar"
```

- از تابع par(mfrow=c(n1,n2)) جهت تقسیم‌بندی پنجرهٔ ترسیم نمودارها استفاده می‌شود.
- از دستور dir() برای نمایش فایل‌های موجود در دایرکتوری فعال استفاده می‌شود.
- برای توقف یک دستور درحال اجرا در محیط R ویندوز از کلید Esc و در ترمینال محیط لینوکس از Ctrl+C استفاده می‌شود.
- دستور getwd() برای تعیین دایرکتوری فعال و setwd برای تنظیم دایرکتوری جدید استفاده می‌شود. در ویندوز، آدرس موردنظر به یکی از شیوه‌های D:/folder1 یا D:\\folder1 نوشته می‌شود.

```
> setwd("D:/folder1")
```

- با دستور زیر ازطریق باز شدن پنجرهٔ انتخاب پوشه، دایرکتوری انتخاب می‌شود.

```
> setwd(choose.dir())
```

- در Rstudio از قسمت New project در منوی File یک پروژه جدید در آدرس خاص ایجاد و تمامی داده‌ها و مثال‌ها در پوشه مربوط به این پروژه ذخیره می‌شوند.

۱-۵ فراخوانی داده‌ها

بعد از انتخاب پوشهٔ موردنظر می‌توان فایل داده را فراخوانی کرد. برای فراخوانی داده‌ها در نرم‌افزار R می‌توان از روش‌های مختلفی استفاده کرد که ساختار داده‌ها در به‌کار بردن روش موردنظر نقش مهمی ایفا

1. Workspace

می‌کند. تابع read.table با دارا بودن قابلیت‌های تبدیل فاکتور و عدد، تنظیم داده‌های گم‌شده، بررسی عناوین و... بیشترین کاربرد را در فراخوانی داده‌ها دارد.

```
> read.table(file="data1.txt")
```

با دستورات زیر می‌توان مستقیماً فایل دادهٔ دلخواه را انتخاب و فراخوانی کرد. A=choose.files() می‌تواند جایگزین سطر اوّل شود.

```
> A<- file.choose()
> read.table(A)
```

همچنین می‌توان هم‌زمان با فراخوانی، فایل آدرس موردنظر را نیز مشخص کرد.

```
> read.table(file="D:\\folder1\\data1.txt")
```

در دستور read.table گزینه‌های مختلفی وجود دارد. گزینهٔ header=TRUE زمانی به‌کار می‌رود که در فایل داده‌ها به هر ستون یک نام اختصاص یافته است و درصورتی‌که ستون‌ها فاقد نام باشد، باید از گزینهٔ header=FALSE استفاده کرد که حالت پیش‌فرض نرم‌افزار هم می‌باشد و به‌صورت خودکار به ستون‌ها نام ستون را اختصاص می‌دهد. درصورتی‌که بخواهیم از عناوین خاص استفاده کنیم، از گزینهٔ col.names=c("x","y") استفاده می‌کنیم. درصورتی‌که داده‌ها دارای عنوان باشند، ولی از header=FALSE استفاده شود، نام ستون‌ها نیز به‌عنوان داده درنظر گرفته می‌شود و برعکس اگر برای داده‌های فاقد عنوان از گزینهٔ header=TRUE استفاده شود، ردیف اوّل داده‌ها به‌عنوان نام ستون‌ها از داده‌ها حذف می‌گردد. گزینهٔ sep="" برای مشخص کردن جداکنندهٔ داده‌ها استفاده می‌شود و "\t" برای فاصلهٔ ۸ کاراکتری و "," برای ایجاد فاصله با کاما انتخاب‌های رایج می‌باشند. هنگام کار با داده‌های ژنتیکی با ابعاد بالا؛ مواردی مانند تعریف نوع کلاس ستون‌ها، تعیین تعداد ردیف‌ها و خالی درنظرگرفتن توضیحات درصورت موجود نبودن در بهبود عملکرد تابع read.table مؤثر است. تابع read.table زمانی‌که تعداد ردیف‌ها از تعداد ستون‌ها بیشتر باشد، عملکرد بهتری دارد و می‌توان با برعکس کردن ردیف‌ها و ستون‌ها در داده‌ها مدت‌زمان فراخوانی را کاهش داد. گزینهٔ na.strings در تابع read.table برای مشخص کردن مقادیر گم‌شده استفاده می‌شود و درصورتی‌که به‌درستی استفاده نشود، مشکلاتی ایجاد می‌کند. در مثال زیر در داده‌های ذخیره‌شده از علامت * به‌عنوان داده گم‌شده استفاده شده است و در هنگام فراخوانی درصورتی‌که مقادیر گم‌شده مشخص نشود، متغیر عددی این ستون به‌صورت فاکتور کلاس‌بندی می‌شود. با استفاده از دستور as.numeric(as.character(phenowrong$weight)) و تبدیل متغیر موردنظر از فاکتور به کاراکتر و سپس به عدد نیز می‌توان این مشکل را برطرف کرد. درصورت استفاده از علامت NA، نرم‌افزار به‌صورت پیش‌فرض آن را به‌عنوان دادهٔ گم‌شده درنظر می‌گیرد.

```
> phenowrong=read.table("animals2.txt", header=T,sep="\t")
> print(phenowrong)
         id weight
1   animal1    300
...
4   animal4      *
> class(phenowrong$weight)# [1] "factor"
> phenoright=read.table("animals2.txt", header=T, sep="\t",
na.strings="*")
> print(phenoright)
         id weight
1   animal1    300
...
4   animal4     NA
> class(phenoright$weight)# [1] "integer
> phenowrong$weight= as.numeric(as.character(phenowrong$weight))
Warning message:
NAs introduced by coercion
> class(phenowrong$weight)#[1] "numeric"
```

با استفاده از دستور na.omit می‌توان ردیف‌های دارای دادهٔ گم‌شده را در هنگام فراخوانی یا پس از فراخوانی حذف کرد.

```
> dat=na.omit(read.csv(ACS.csv))
> dat2=ead.csv(ACS.csv); dat2=na.omit(dat)
```

برای فایل‌هایی که با فرمت فاصله ویرگول ذخیره شده‌اند، از دستور read.csv() استفاده می‌شود.

```
> dat <- read.csv("http://mgimond.github.io/ES218/Data/ACS.csv",
header=TRUE)
```

نرم‌افزار R فرمت مختص خودش با پسوند rds را نیز دارد که علاوه‌بر حفظ نوع و کلاس داده‌ها، حجم کمتری اشغال می‌کند و برای ذخیرهٔ داده‌های ژنومی مناسب می‌باشد و به‌صورت زیر فراخوانی می‌شود. در هنگام فراخوانی از سایت، ابتدا تابع gzcon فایل را از حالت فشرده خارج می‌کند.

```
> dat <- readRDS("ACS.rds")
> dat <-
  readRDS(gzcon(url("http://mgimond.github.io/ES218/Data/ACS.rds")))
```

برای فراخوانی داده‌های اکسل، از کتابخانهٔ readxl استفاده می‌شود.

```
> library(readxl)
> dat <- read_excel("a.xlsx", sheet = "a1")
```

برای دانلود فایل‌های اکسل از سایت، ابتدا فایل در یک پوشهٔ موقّت ذخیره می‌شود.

```
> web.file <-
"http://mgimond.github.io/ES218/Data/Discharge_2004_2014.xlsx"
> tmp <- tempfile(fileext=".xlsx")
> download.file(web.file,destfile=tmp, mode="wb")
> xl <-  read_excel(tmp, sheet = "Discharge")
```

با استفاده از تابع download کتابخانهٔ downloader می‌توان فایل داده را از سایت دانلود و سپس آن را فراخوانی کرد.

```
> library(downloader)
> url <-
"https://raw.githubusercontent.com/genomicsclass/dagdata/master/inst
/extdata/femaleMiceWeights.csv"
> filename <- "femaleMiceWeights.csv"
> download(url, destfile=filename)
> dat <- read.csv(filename);head(dat)
  Diet Bodyweight
1 chow      21.51
...
```

۱-۶ ذخیرهٔ داده‌ها

برای ذخیرهٔ داده‌ها یا ذخیرهٔ بخشی از R می‌توان با دستور save آن‌ها را به‌صورت یک فایل مخصوص برای R ذخیره کرد. از list=ls() برای انتخاب تمام اشیای جاری استفاده می‌شود. برای فراخوانی این فایل از دستور load استفاده می‌شود.

```
> setwd(choose.dir()) ;
> save(xl, file = "xl.Rdata")
> save(list=ls(), file = "objects.Rdata")
> Save(list=c(a,b),file="ab.Rdata")
> setwd(choose.dir())
> load("xl.RData")
```

برای ذخیرهٔ داده‌ها به‌صورت فایل متنی از دستور write.table استفاده می‌شود. دستور اوّل، داده‌ها را به حالت csv و دستور دوم به‌صورت txt ذخیره می‌کند. برای ذخیرهٔ داده‌ها با فرمت rds از دستور saveRDS استفاده می‌شود.

```
> write.table(data1, file=" a1.csv", sep=",", quote=FALSE,
row.names=FALSE)
> write.table(data1, file=" a1.txt",
quote=FALSE,row.names=FALSE,sep="\t")
> saveRDS(data1, "a1.rds")
```

برای ذخیرهٔ نتایج می‌توان از دستور sink استفاده کرد. با این دستور، ابتدا یک فایل خاص ایجاد و خروجی دستورات بعدی در این فایل ذخیره می‌شود. در مثال زیر، خروجی دستور anova در فایل متنی anova.txt ذخیره می‌شود.

```
> weight=sample(10:20,20,replace=T);
treat=sample(c("A","B"),20,replace=T)
> setwd(choose.dir)
> sink("anova.txt")
> anova(lm(weight~treat)); sink()
```

۱-۷ مشاهده و بررسی داده‌ها

برای مشـــاهـدهٔ داده‌ها در محیط R علاوه‌بر نوشـــتن نام دادهٔ موردنظر به‌تنهایی، می‌توان از توابع edit()، data.entry() و View() به‌همراه نام داده استفاده کرد که امکان بررسـی داده‌ها را راحت‌تر می‌سـازند. این توابع داده‌ها را به‌صورت جدول نمایش داده و قابلیت ویرایش داده‌ها امکان‌پذیر می‌کند که با بستن پنجره، تغییرات به‌صـورت خودکار در فایل ذخیره‌شـده اعمال خواهد شد. می‌توان با توابع head() و tail() قسمتی از داده‌ها (به‌صـورت پیش‌فرض ۶ سـطر اوّل یا ۶ سـطر آخر) را مشـاهده کرد. برای مشاهدهٔ تعداد کمتر یا بیشتر از ۶ سطر، تعداد سطر نیز نوشته می‌شود.

```
> data("Orange"); Orange
Tree  age circumference
1     1  118            30
...
> edit(Orange)
> View(Orange)
> head(Orange)
  Tree  age circumference
1    1  118            30
...
> head(Orange,8)
  Tree  age circumference
1    1  118            30
...
```

۲

انواع داده‌ها در R

۲-۱ بردار

ساده‌ترین ساختار داده در محیط R، بردار است که دارای سه ویژگی طول، حالت و نام است. طول، بیانگر تعداد عناصر بردار است و حالت بردار، بیانگر نوع عناصر آن است که یکی از انواع عددی، کاراکتری، مختلط یا منطقی است. عناصر یک بردار همه باید از یک نوع باشند. بردار اغلب با دستور c() ساخته می‌شود. دنباله‌ها بردارهای خاصی هستند که با عبارت ":" یا توابع seq و rep ایجاد می‌شوند. برای نوشتن یک دنباله با تابع seq از گزینه‌های by و length برای تعیین قدر نسبت و تعداد جمله‌های دنباله استفاده می‌شود. برای یک دنبالهٔ حسابی از a تا b با قدر نسبت یک می‌توان از a:b استفاده کرد. طول بردار با تابع length و حالت آن با تابع mode مشخص می‌شود.

```
> c(2,3,5,7,3,8,1,5)# [1] 2 3 5 7 3 8 1 5
> 1:5; seq(from=5,to=20,by=2);seq(from=5,to=20,length=4);
seq(4,by=3,length=5)
[1] 1 2 3 4 5       [1]  5  7  9 11 13 15 17 19
[1]  5 10 15 20     [1]  4  7 10 13 16
> rep(0,10);
rep(1:3,5);rep(c(1,2,3),c(3,4,5));rep(1:4,time=3);rep(1:4,each=3)
 [1] 0 0 0 0 0 0 0 0 0 0    [1] 1 2 3 1 2 3 1 2 3 1 2 3 1 2 3
 [1] 1 1 1 2 2 2 2 3 3 3 3 3       [1] 1 2 3 4 1 2 3 4 1 2 3 4
 [1] 1 1 1 2 2 2 3 3 3 4 4 4
```

در آنالیزهای ژنتیکی و بیوانفورماتیک، چسباندن کاراکترها به دنباله‌ها کاربرد زیادی دارد که با استفاده از تابع paste انجام می‌شود.

```
> paste("A",1:5,sep="")
[1] "A1" "A2" "A3" "A4" "A5"
> paste(c("B","C"),rep(1:5,each=2),sep="_")
 [1] "B_1" "C_1" "B_2" "C_2" "B_3" "C_3" "B_4" "C_4" "B_5" "C_5"
```

با استفاده از علامت [] می‌توان عناصر مشخصی از بردار را انتخاب یا حذف کرد. برای تغییر عناصر بردار نیز از این علامت می‌توان استفاده کرد. تابع which برای دسترسی به موقعیت عناصر خاصی از بردار استفاده می‌شود.

```
> a=c(2,3,5,7,3,8,1,5)
> a[4]; a[-c(1,2)]; a[2:5]; a[a>5]
[1] 7  [1] 5 7 3 8 1 5     [1] 3 5 7 3  [1] 7 8
> a[a>3 & a<7];a[a<2 | a>7]; a[which(a<5)]
 [1] 5 5     [1] 8 1      [1] 2 3 3 1
> which(a==3)
 [1] 2 5
> a[1]=4; a
 [1] 4 3 5 7 3 8 1 5
```

برای نام‌گذاری عناصر بردارها می‌توان از دو روش استفاده کرد.

```
> grades=c(a=20,b=15,c=18); grades["a"]  # a   20
>grades2=c(12,10,15);names(grades2)=c("a","b","c");which.max(grades2
)
# c  3
> sort(grades2)
 b  a  c
10 12 15
```

۲-۲ ماتریس

ماتریس‌ها آرایه‌های دوطرفه هستند که کار کردن با آن‌ها در ژنتیک کمّی و بیوانفورماتیک از اهمیت بالایی برخوردار است. متغیرهای موردنظر به‌صورت ستون و واحدهای آزمایشی به‌صورت سطر معرفی می‌شوند و می‌توان هر نوع داده‌ای را در قالب ماتریس ذخیره کرد. برای مشاهدهٔ ابعاد، حالت و طول ماتریس (کل درایه‌ها) از توابع dim، mode و length استفاده می‌شود.

برای ساخت ماتریس از توابع matrix، array یا dim استفاده می‌شود. dim درواقع ابعاد ماتریس را مشخص می‌کند که در ساخت ماتریس نیز استفاده می‌شود. تعداد ردیف‌ها با دستور dim()[1] و تعداد ستون‌ها با دستور dim()[2] تعیین می‌شود. تعداد سطر و ستون در تابع matrix با nrow و ncol مشخص می‌شود و به‌صورت پیش‌فرض مقادیر به‌صورت ستونی در ماتریس قرار می‌گیرند و با گزینهٔ byrow=TRUE مقادیر به‌صورت سطری قرار خواهند گرفت. از دو تابع rbind و cbind جهت ترکیب سطری و ستونی بردارها و ماتریس‌ها و ایجاد ماتریس جدید استفاده می‌شود.

```
> A=matrix(c(5,4,7,9,8,10),nrow=3,ncol=2); dim(A); mode(A);length(A)
[1] 3 2   , [1] "numeric"  , [1] 6
> array(c(5,4,7,9,8,10),dim=c(3,2))
    [,1] [,2]
[1,]    5    9
[2,]    4    8
[3,]    7   10
> matrix(c(5,4,7,9,8,10),3,2,byrow=T)
     [,1] [,2]
[1,]    5    4
[2,]    7    9
[3,]    8   10
> array(x,c(3,6))
     [,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
> matrix(x,3,6,byrow = T)
     [,1] [,2] [,3] [,4] [,5] [,6]
[1,]    1    2    3    4    5    6
[2,]    7    8    9   10   11   12
[3,]   13   14   15   16   17   18
> cbind(A=1:4,B=5:8,C=9:12)
     A B  C
[1,] 1 5  9
[2,] 2 6 10
[3,] 3 7 11
[4,] 4 8 12
```

از توابع rownames، colnames و dimnames برای نام‌گذاری سطرها و ستون‌ها در ماتریس استفاده می‌شود و از ترکیب این دو تابع با توابع LETTERS یا letters که شامل حروف بزرگ یا کوچک هستند، نام‌گذاری راحت‌تر انجام می‌شود.

```
> x<-matrix(1:9,nrow=3); rownames(x)<-LETTERS[1:3];x
  [,1] [,2] [,3]
A    1    4    7
B    2    5    8
C    3    6    9
> dimnames(x)<-
list(paste("row",letters[1:3]),paste("col",LETTERS[1:3])); x
      col A col B col C
row a     1     4     7
row b     2     5     8
row c     3     6     9
```

برای به‌دست‌آوردن حاصل‌جمع ردیف‌ها و ستون‌های یک ماتریس از دستورات rowSums و colSums و میانگین آن‌ها به‌ترتیب از rowMeans و colMeans استفاده می‌شود.

```
> x <- cbind(x1=3,x2=c(4:1,2:5))
> rowSums(x); colSums(x);rowMeans(x);colMeans(x)
[1] 7 6 5 4 5 6 7 8
x1 x2
24 24
[1] 3.5 3.0 2.5 2.0 2.5 3.0 3.5 4.0
x1 x2
 3  3
```

ماتریس‌های همانی، قطری و یک در ژنتیک کاربرد زیادی دارند و به‌صورت زیر ایجاد می‌شوند.

```
> diag(2)
     [,1] [,2]
[1,]    1    0
[2,]    0    1
> diag(x=2,3);diag(c(2,3,4))
      [,1] [,2] [,3]
[1,]    2    0    0
[2,]    0    2    0
[3,]    0    0    2
     [,1] [,2] [,3]
[1,]    2    0    0
[2,]    0    3    0
[3,]    0    0    4
> diag(2,nrow=3,ncol=4)
     [,1] [,2] [,3] [,4]
[1,]    2    0    0    0
[2,]    0    2    0    0
[3,]    0    0    2    0
> matrix(1,3,3)
     [,1] [,2] [,3]]
[1,]    1    1    1
[2,]    1    1    1
[3,]    1    1    1
```

اثر[1] یک ماتریس به‌صورت جمع عناصر قطری محاسبه می‌شود و در تعیین درجهٔ آزادی جدول تجزیهٔ واریانس و برآورد مؤلّفه‌های واریانس در روش حداکثر درست‌نمایی محدودشده کاربرد دارد. برای انتخاب زیرمجموعه‌ای از یک ماتریس همانند بردارها از [] و تعیین شمارهٔ سطر و ستون درایهٔ موردنظر استفاده می‌شود. در ماتریس‌ها می‌توان با دستورات زیر به درایهٔ خاص، چند درایه، سطر و ستون خاصی اشاره کرد یا آن‌ها را تغییر داد. به‌عنوان مثال، درایهٔ واقع در سطر سوم و ستون اوّل با A[3,1]، تمام درایه‌های سطر سوم با A[3,]، تمام درایه‌های ستون دوم با A[,2]، درایه‌های سطر دوم تا سوم و ستون‌های اوّل و سوم با A[2:3,c(1,3)]، تمام درایه‌های غیر از ستون دوم و سوم با A[,-c(2,3)] و درایه‌های بزرگ‌تر از ۵ با A[A>5] مشخص می‌شود. از تابع eign برای به‌دست‌آوردن مقادیر و بردارهای ویژهٔ یک ماتریس استفاده می‌شود. ضرب داخلی، خارجی و کرونکر (حاصل‌ضرب تک‌تک درایه‌های ماتریس اوّل در تمام دارایه‌های ماتریس دوم) ماتریس‌ها با دستورات A*B ، A%*%B و %X% محاسبه می‌شود. از تابع crossprod نیز می‌توان برای حاصل‌ضرب خارجی دو ماتریس استفاده کرد.

1. Trace