محاسبات آماری با R

بر اساس سرفصل درس محاسبات آماری

مولفان:

دکتر حسین زمانی

دانشگاه هرمزگان

عباس افتخاریان

دانشگاه هرمزگان

فصل اول

آشنایی با نرم افزار R

۱.مقدمه

امروزه برای انجام محاسبات آماری، نرم افزارهای متعددی وجود دارد که هر یک از آنها دارای قابلیتهای منحصر بفردی می باشند. در این میان و از سالها قبل، نرم افزار S-Plus که یک نرم افزار تجاری می باشد، از محبوبیت بسیاری برخوردار بوده است. دلایل این محبوبیت مربوط به توانایی بسیار بالای این نرم افزار در پردازش و تجزیه و تحلیل داده ها می باشد. نرم افزار S-Plus شامل دو بخش دستور نویسی و ویژوال(visual) می باشد و همان طور که اشاره شد این نرم افزار یک نرم افزار تجاری می باشد و لذا دسترسی به نسخه کامل (full version) آن برای همه امکان پذیر نمی باشد.

به دلیل قابلیت بالای نرم افزار S-Plus و به دلیل محدودیت دسترسی به آن، نرم افزار جدیدی عرضه شد که فقط قسمت دستور نویسی برنامه S-Plus را در بر داشت و مهمترین مزیت آن رایگان بودن آن است. این نرم افزار R نام دارد و پروژه آن از سال ۱۹۹۵ در گروه آمار دانشگاه اوکلند (Auckland) توسط آقایان Robert Gentleman و Ross Ihaka شروع شد و خیلی سریع مخاطبان زیادی جذب کرد. اسم این نرم افزار، از حرف اول اسم دو سازنده آن گرفته شده است. در واقع R یک نرم افزار رایگان از نسخه تجاری S-Plus است تمامی قابلیتهای موجود در بخش دستورنویسی برنامه S-Plus را دارد و به همین دلیل در یک دهه اخیر نرم افزار R در همه سطوح فراگیر شده است. برخی دیگر از دلایل محبوبیت و مطلوبیت این نرم افزار، به اختصار عبارتند از:

۱- به روز بودن بسته های نرم افزاری (Packages) نرم افزار R
۲- قابلیت انجام و بررسی مسائل کاملا تخصصی علم آمار و امکان دسترسی به داده های واقعی
۳- پذیرش بسته های نرم افزاری جدید از طرف سایت R که این بسته ها توسط افراد متخصص مختلف عرضه می شوند.
۴- امکان ساخت و عرضه بسته های نرم افزاری و توابع جدید متناسب با نیازهای موجود و براساس پیشرفتهای علمی بدست آمده اخیر.
۵- نرم افزار R دارای راهنمای داخلی خوبی است.
۶- نرم افزار R دارای قابلیتهای قابل ملاحظه گرافیکی است.

با وجود تمامی مزایای اشاره شده در بالا، نرم افزار R داری محدودیتهایی نیز می باشد که برخی از آنها به قرار زیر می باشند:

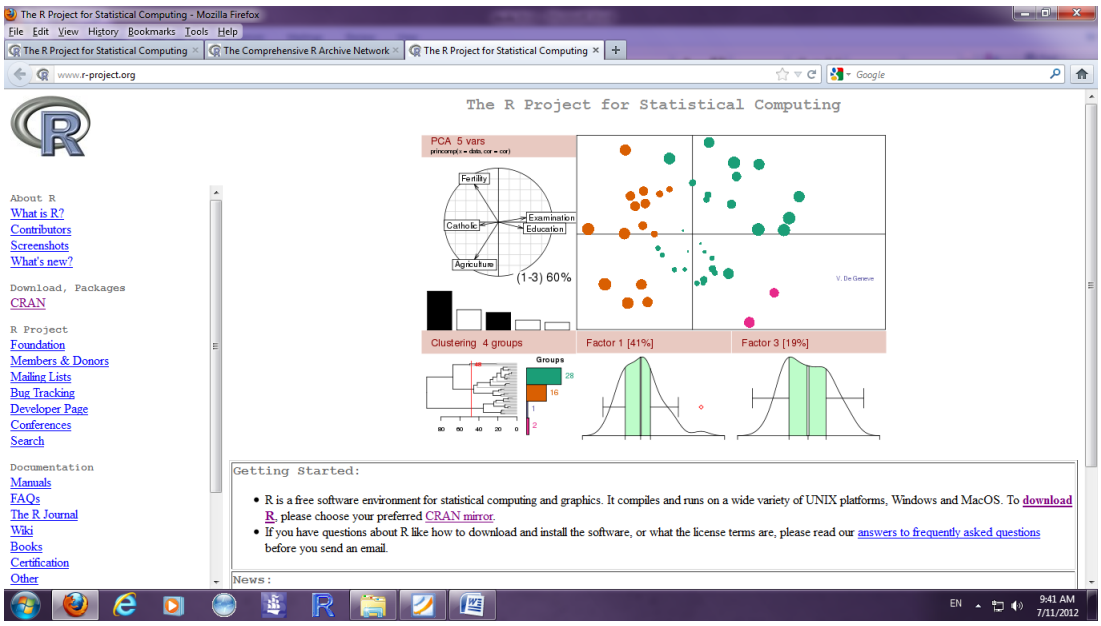
۱- دارای امکان ایجاد Gui (Graphic user interface) نیست، در حالیکه نرم افزار S-Plus در این باره امکانات خوبی دارد.
۲- از طرف یک شرکت تجاری پشتیبانی نمی شود.
۳- برای استفاده از آن، باید فرامین آن را آموخت.
۴- خروجی آن به صورت یک فایل اجرایی (*.exe) در نمی آید.

در اینجا ما ابتدا چگونگی دسترسی به فایل نصبی آن (source) و نصب این نرم افزار را شرح خواهیم داد و سپس به بررسی درباره خود نرم افزار و چگونگی کار کردن با آن خواهیم پرداخت.

۲.دسترسی به نرم افزار R و نصب آن

این نرم افزار را می توان از مسیر زیر دانلود کرد. ابتدا وارد سایت نرم افزار R به آدرس زیر شوید.

http://www.r-project.org

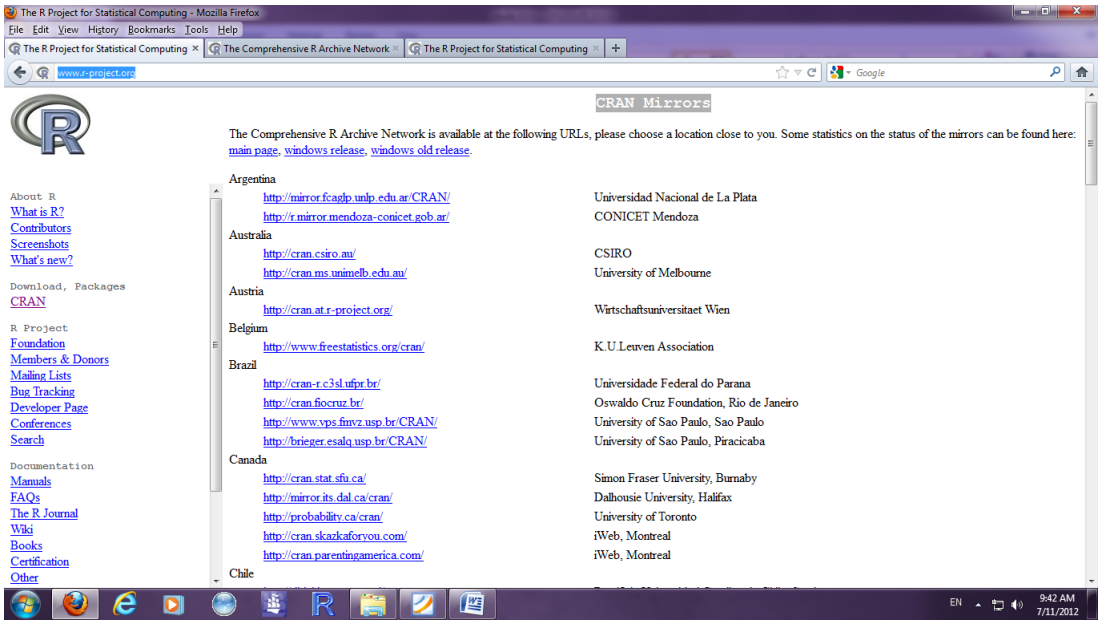


شکل ۱-۱

سپس در این صفحه (شکل ۱-۱) و در قسمت Getting started بر روی لینک download R کلیک کنید.

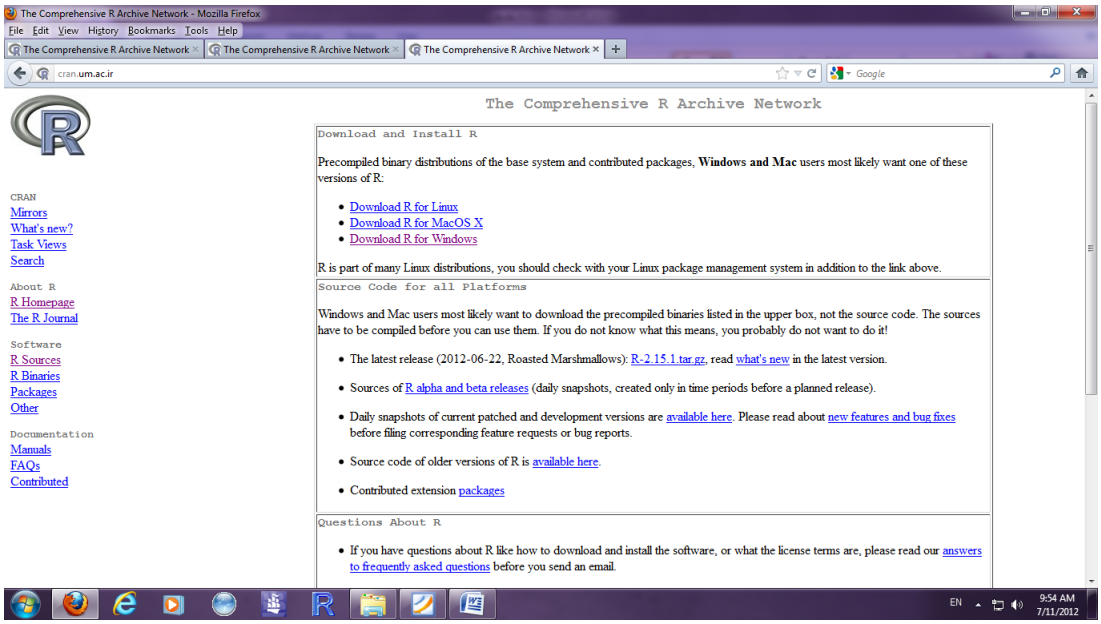
در صفحه بعد (شکل ۱-۲) که به CRAN Mirrors معروف است نام سرورهای موجود در کشورهای مختلف برای عرضه فایلهای این نرم افزار وجود دارد. مثلا در ایران، سرور مربوط به دانشگاه فردوسی مشهد با لینک زیر

http://cran.um.ac.ir/



شکل ۱-۲

این کار را انجام می دهد. اگر بر روی لینک فوق که در صفحه CRAN Mirrors مقابل نام Iran نیز موجود است کلیک کنید صفحه دیگری (شکل ۱-۳) به نام The Comprehensive R Archive Network باز می شود که به صورت زیر می باشد.

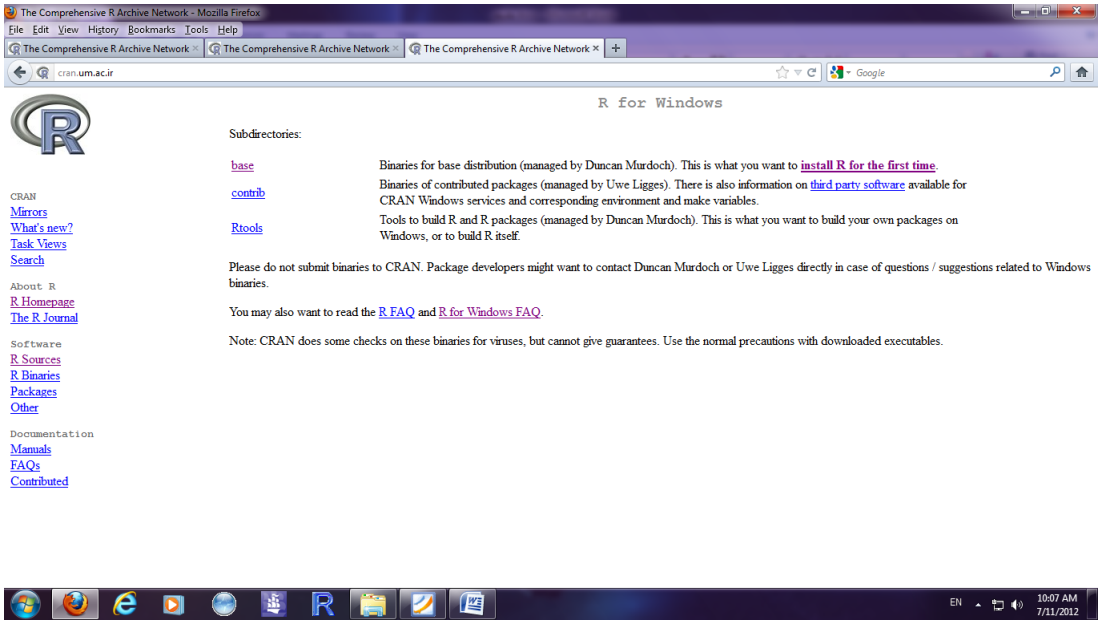


شکل ۱-۳

در این صفحه، با توجه به سیستم عاملی که قرار است نرم افزار R روی آن نصب شود یکی از سه لینک

- Download R for Linux
- Download R for MacOS X
- Download R for Windows

را انتخاب و برروی آن کلیک کنید. مثلا فرض کنید که قرار است نرم افزار R برروی سیستم عامل ویندوز نصب شود. پس کافیست بر روی لینک Download R for Windows کلیک کنید. بعد از این صفحه جدیدی به نام R for Windows باز می شود (شکل ۱-۴) که به شکل زیر است:



شکل ۱-۴

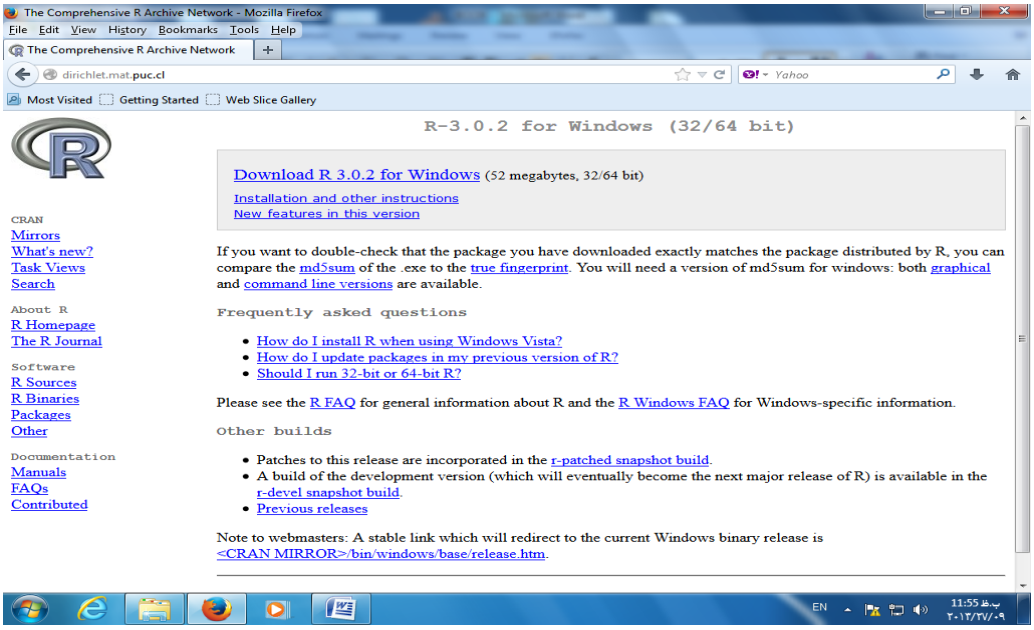
در این صفحه بر روی لینک install R for the first time کلیک می کنید تا صفحه دیگری (شکل ۱-۵) به نام R-3.0.2 for Windows (32/64 bit) باز شود . شایان ذکر است که نام این صفحه همیشه به این صورت نخواهد بود و دائما تغییر می کند. زیرا نام این صفحه همواره متناسب با جدیدترین نسخه نرم افزار R به روز می شود. مثلا در حال حاضر جدیدترین نسخه این نرم افزار 3.0.2 می باشد.

در آخرین مرحله برای دستیابی به فایل نصبی این نرم افزار، کافی است برروی لینک

Download R 2.15.1 for Windows

کلیک نمایید تا بتوانید فایل نصب کننده این نرم افزار را بدست آورید.

پس از اتمام دانلود فایل نصب کننده R ، برای نصب این نرم افزار به طریق زیر عمل کنید.



شکل ۱-۵

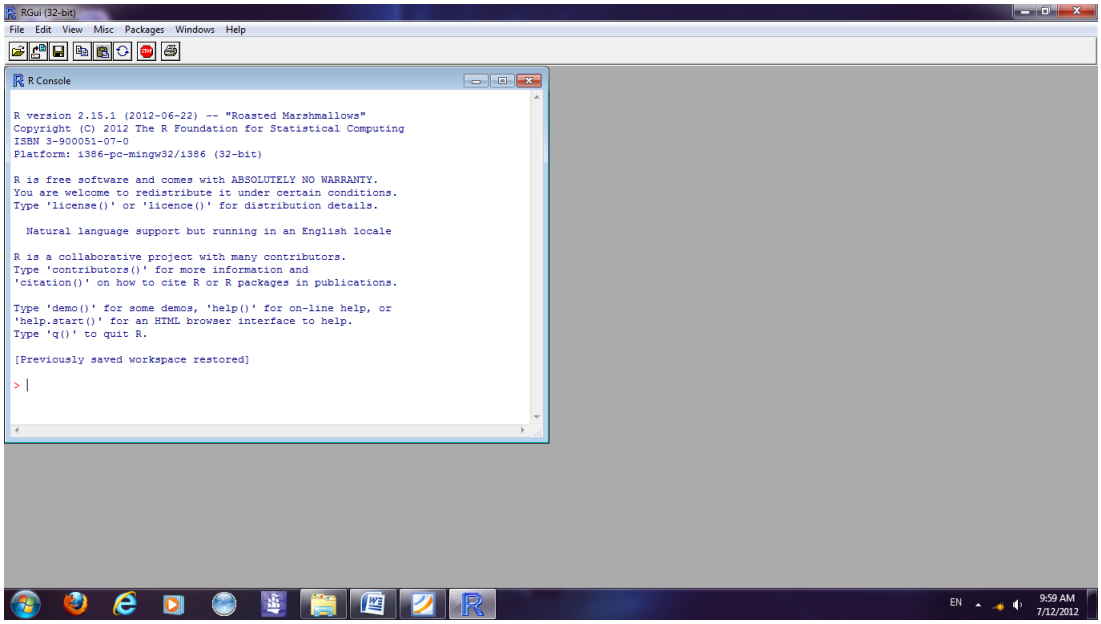
ابتدا روی فایل اجرایی R-2.15.1-win32.exe که دانلود شده دوبار کلیک کنید. سپس همانند نصب هر نرم افزار دیگری و بدون تغییر پیش فرضهای در نظر گرفته شده، با کلیک روی گزینه Next به طور متوالی، می توانید این نرم افزار را نصب کنید و نکته خاصی در نصب این نرم افزار وجود ندارد. پس از پایان عمل نصب، روی desktop کامپیوتر شما یک آیکون به شکل حرف R ظاهر می شود. این آیکون در واقع، آیکون اجرایی نرم افزار R می باشد و با دو بار کلیک کردن روی آن، نرم افزار شروع به کار کرده و آماده پذیرش دستورات می باشد.

۳.آشنایی با شکل ظاهری نرم افزار R

هنگامی که فایل اجرایی موجود بر روی Desktop را اجرا کنیم پنجره ای به صورت زیر باز می شود (شکل ۱-۶) که شکل ظاهری نرم افزار R می باشد.

پنجره اولیه که مطابق شکل فوق است به پنجره RGui/R Console موسوم است که در این پنجره برخی از مشخصات این نرم افزار از قبیل نوع نسخه، زمان به روز شدن آن و... نوشته شده است.

در کل، نرم افزار R دارای سه نوع پنجره می باشد که عبارتند از:



شکل ۱-۶

۱- پنجره RGui/R Console

در این پنجره هر دستور که به نرم افزار می دهیم خط به خط اجرا می کند.

۲- پنجره R-Editor

اگر نخواهیم دستورها خط به خط اجرا شوند می توان از مسیر File > New script پنجره R-Editor را باز کرد که پنجره‌ای شبیه پنجره Notepad می باشد. در این پنجره می توان دستورهای مورد نظر را نوشته و سپس با انتخاب آنها و سپس فشردن کلیدهای Ctrl + R برنامه نوشته شده را اجرا کرد.

۳- پنجره R Graphics

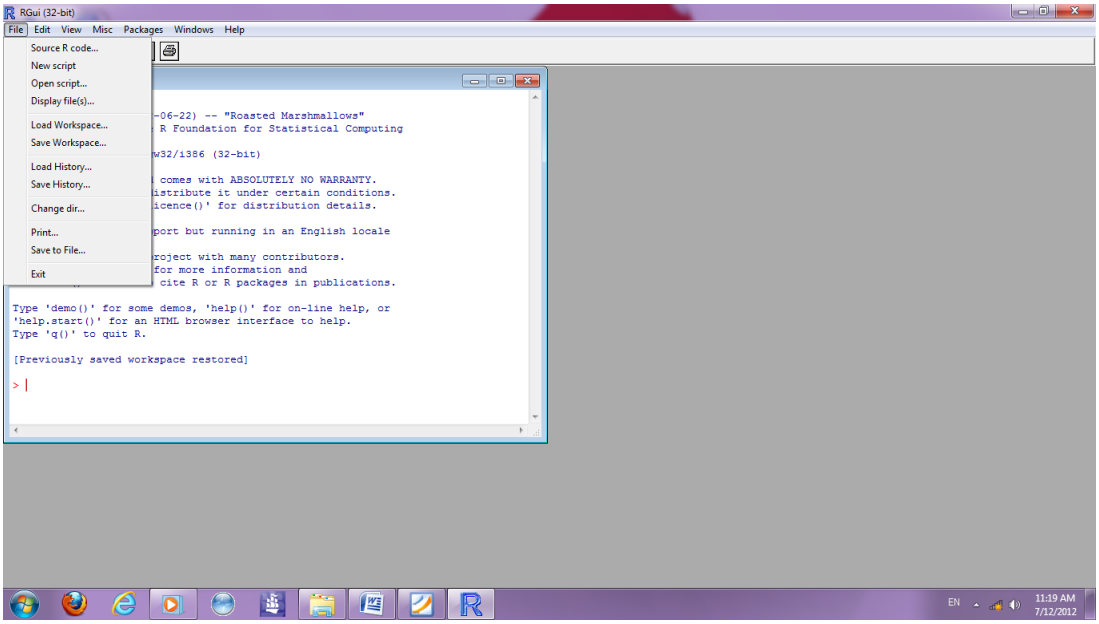
در این پنجره، خروجی نمودارهایی که برنامه آنها را در پنجره مثلا R Consol نوشته ایم قابل مشاهده می باشد. در فصلهای آینده راجع به این پنجره و نمودارهای ترسیم شده در آن بیشتر صحبت خواهیم کرد.

حال به بررسی مختصر منوهای موجود در این نرم افزار می پردازیم. منوهای نرم افزار R عبارتند از:

File, Edit, View, Misc, Packages, Windows, Help

۱- منوی File

شکل ظاهری این منو به صورت زیر است (شکل ۱-۷)



شکل ۱-۷

منوی File دارای زیر منوهای زیر است:

i) Source R code

با انتخاب این گزینه می توان برنامه ای که قبلا نوشته شده را باز و اجرا کرد.

ii) New script

همانطور که قبلا اشاره شد از طریق این زیر منو می توان پنجره جدید R-Editor باز کرد.

iii) Open script

با این زیر منو می توان اسناد ذخیره شده قبل را فرا خوانی کرد.

iv) Display file(s)

از طریق این گزینه می توان فایلهای قابل شناسایی در این نرم افزار را نمایش داد.

v) Load Workspace

صفحه ای که با آن کار می کنیم Workspace (صفحه کاری) نام دارد (مثلا R-console). با گزینه Load Workspace می توان یک صفحه کاری را فراخوانی کرد.

vi) Save Workspace

با این زیر منو می توان صفحه کاری را ذخیره کرد.

vii) Load History

به وسیله این گزینه می توان تاریخچه را فراخوانی کرد.

viii) Save History

از طریق این زیر منو تاریخچه ذخیره می شود.

ix) Change dir

این زیر منو، دایرکتوری کاری را تغییر می دهد.

x) Save to File

با این گزینه می توان دستورات و خروجی ها را به صورت فایل متنی ذخیره کرد.

۲- منوی Edit

در این منو موارد استفاده از گزینه هایی مانند Copy و Paste و نیز Select all کاملا واضح است و نیازی به توضیح بیشتر احساس نمی شود. پس به بررسی سایر گزینه ها می پردازیم.

i) Paste commands only

با این زیر منو می توان فقط دستورات را از جایی به جای دیگر منتقل کرد.

ii) Clear console

این گزینه می تواند صفحه دستورات را کاملا پاک کند. لازم به ذکر است که این عمل با کلیدهای Ctrl + L نیز انجام می شود.

iii) Data Editor

از طریق این زیر منو می توان داده ها را ویرایش کرد.

iv) GUI Preferences

این گزینه می تواند پیش فرضهای نرم افزار را تغییر دهد.

۳- منوی View

این منو شامل گزینه های Toolbar و Statusbar است که به ترتیب نشانگر نوار ابزار در بالای صفحه نرم افزار و یک نوار روشن در پایین صفحه نرم افزار است که برروی آن نوع نسخه نرم افزار روی آن نوشته شده است.

۴- منوی Misc

این منو شامل زیر منوهای ذیل است:

i) Stop current computation & Stop all computations

این گزینه ها به ترتیب محاسبات فعلی و تمام محاسبات را متوقف می کند.

ii) Buffered output

iii) Word completion

iv) Filename completion

v) List objects

این زیر منو تمام اشیاء R را نمایش می دهد.

vi) Remove all objects

این گزینه تمام اشیاء در R را حذف می کند.

vii) List search path

این زیر منو فهرست تمام بسته های نرم افزاری در حال استفاده در R را نمایش می دهد.

۵- منوی Packages

این منو دارای گزینه ها زیر است:

i) Load package

به وسیله این گزینه می توان بسته های نرم افزاری را فراخوانی کرد.

ii) Set CRAN mirror

با این گزینه می توان سرور پیش فرض را تنظیم کرد. برای مثال سرور ایران در دانشگاه فردوسی مشهد واقع است.

iii) Set repositories

این گزینه مخزن بسته های نرم افزاری را مشخص می کند که مثلا به صورت پیش فرض CRAN می باشد.

iv) Install package(s)

نصب آنلاین بسته های نرم افزاری توسط این گزینه انجام می شود.

v) Update packages

بروز رسانی بسته های نرم افزاری با این گزینه انجام می شود.

vi) Install package(s) from local zip files

نصب بسته های نرم افزاری از طریق فایلهای دانلود شده و نه به صورت آنلاین، از طریق این زیر منو انجام می پذیرد.

۶- منوی Windows

گزینه های این منو فقط مربوط به تنظیمات ظاهری (مانند کوچک یا بزرگ کردن) پنجره نرم افزار R می شوند و به همین دلیل از دادن توضیحات اضافی خودداری می کنیم.

۷- منوی Help

این شامل گزینه های زیر است:

i) Console

این گزینه راهنمای کلیدهای میانبر در صفحه Console می باشد.

ii) FAQ on R

در این گزینه بیشترین سوالاتی که از طرف کاربران درباره R مطرح شده است ارائه می شود.

iii) FAQ on R for Windows

بیشترین سوالاتی که راجع به R در سیستم عامل ویندوز از قبیل آخرین نسخه این نرم افزار یا طریقه نصب یا حذف این نرم افزار روی سیستم عامل ویندوز، که از طرف کاربران مطرح شده است در این زیر منو وجود دارد.

iv) Manuals (in PDF)

این گزینه حاوی چند فایل راهنما به صورت PDF درباره نرم افزار R می باشد.

v) R functions (text)

این گزینه شامل راهنمای توابع R می باشد.

vi) Html help

به وسیله این زیر منو می توان به راهنمای کامل نرم افزار R که به صورت فایل های html است دست یافت.

vii) Search help

جستجوی موضوعی در نام یا دستورات و یا توضیحات توابع به وسیله این گزینه صورت می پذیرد.

viii) search.r-project.org

جستجو در سایت نرم افزار R توسط این زیر منو امکان پذیر می باشد.

ix) Apropos

این گزینه جستجوی تطبیقی در نام توابع را انجام می دهد و تمامی توابعی را که شامل آن نام مورد نظر می باشد را نمایش می دهد.

x) R Project home page

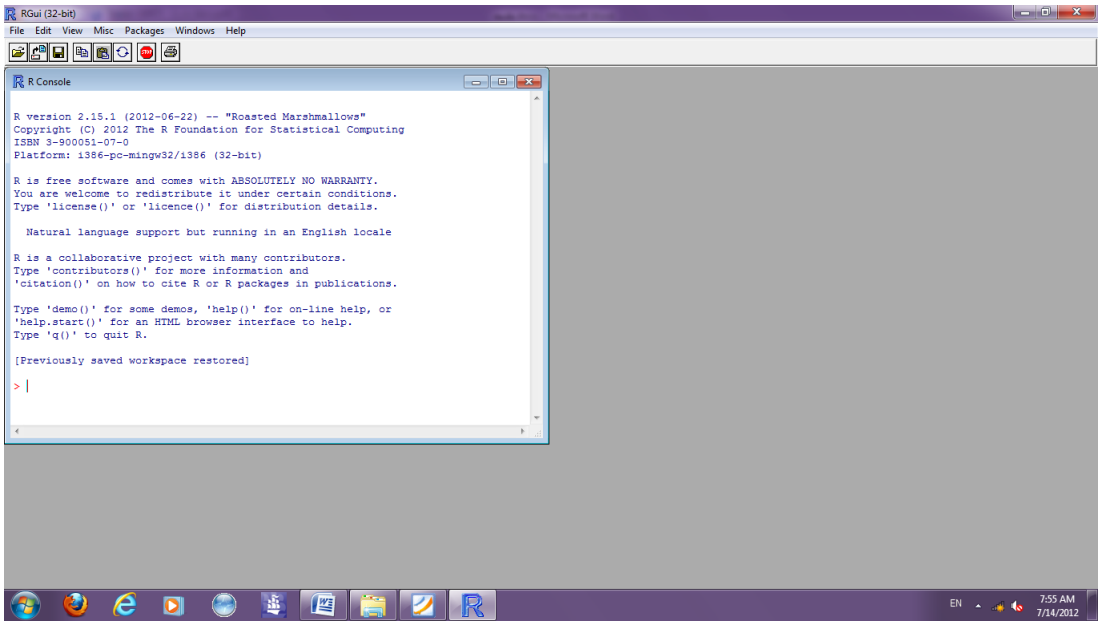
این گزینه در واقع یک لینک است که توسط آن می توان به سایت اصلی R دسترسی پیدا کرد.

xi) CRAN home page

به وسیله این زیر منو می توان به سایت آرشیو نرم افزار R دسترسی پیدا کرد و بسته های نرم افزار را از آنجا دانلود کرد.

۴.آشنایی با دستورنویسی در R

زمانی که نرم افزار R را باز می کنیم شکل ظاهری آن (شکل ۱-۸) به صورت زیر است و مکان نما در جلوی علامت ">" قرار دارد و آماده گرفتن دستور می باشد. البته این علامت می تواند به صورت "$" نیز ظاهر شود.



شکل ۱-۸

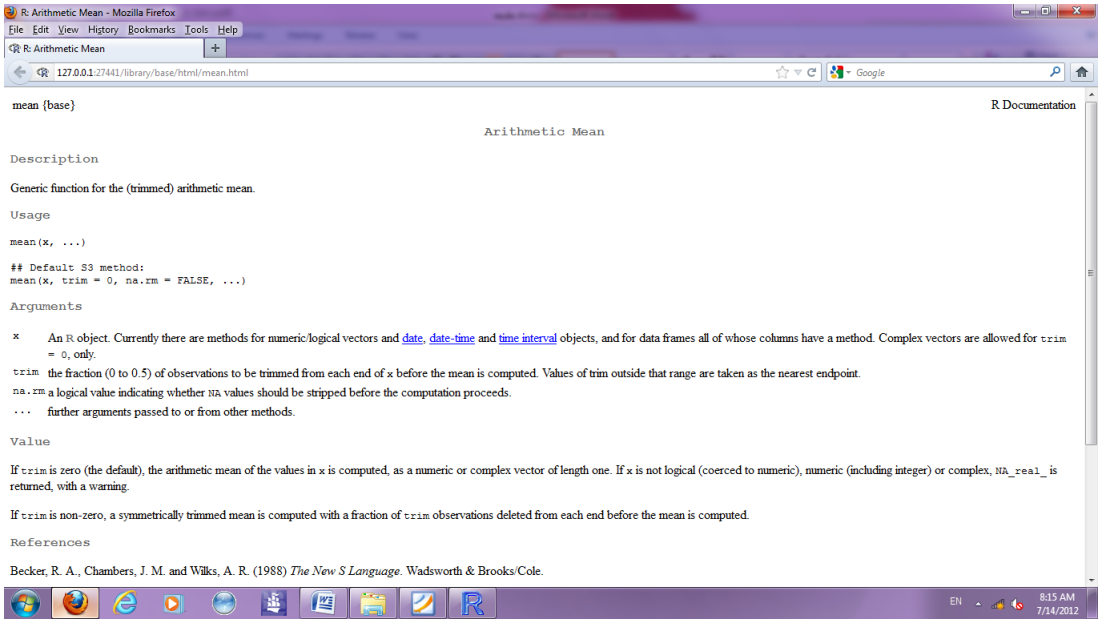
در این حالت اگر از دستور

```
>help()
```

استفاده کنیم می توانیم به راهنمای نرم افزار جهت دستیابی به دستورات و اطلاعات جهت یک موضوع خاص دسترسی پیدا کنیم. مثلا اگر بخواهیم دستور محاسبه میانگین را بدست آوریم کافی است بنویسیم

```
>help(mean)
```

که در این صورت یک صفحه html به صورت زیر(شکل ۱-۹) باز می شود که در آن اطلاعات جامعی درباره دستور میانگین، نوع داده های ورودی، چند مثال درباره میانگین همراه با منابع وجود دارد.



شکل ۱-۹

باید توجه داشت که نرم افزار R مانند یک ماشین حساب ساده نیز عمل می کند، یعنی چهار عمل اصلی جمع، تفریق، ضرب و تقسیم را بدون اینکه نیاز به نوشتن دستور خاصی باشد انجام می دهد.

فصل ۲

متغیرها و داده ها در R

۱. مقدمه

اساس کار در نرم افزار R منطبق بر متغیرها (اشیاء) و داده های ورودی است، لذا آشنایی با چگونگی تعریف و نیز انواع متغیرها و همچنین چگونگی وارد کردن داده ها، از اهمیت بسیاری برخوردار است و اولین گام و در عین حال یکی از مهمترین گامها جهت انجام تجزیه و تحلیل درست داده ها محسوب می شود. در این بخش ابتدا به چگونگی تعریف متغیرها پرداخته و سپس انواع متغیرها را معرفی می کنیم و در پایان طریقه وارد کردن داده ها را در حالت کلی مورد بحث قرار می دهیم.

۲. تعریف متغیرها در R

تعریف یک متغیر در R عبارتست از نسبت دادن یکی از حروف انگلیسی (مثلا x) یا یک کلمه (مثلا variable) و یا ترکیبی از حروف و اعداد (مثلا x1) به یک عدد، یا بردار یا ماتریس و یا یک تابع. برای این کار می توان از علامت های منها و کوچکتر به صورت (<-) استفاده کرد که در این حالت نام متغیر حتما باید در طرف علامت کوچکتر و مقدار عددی یا تابع مورد نظر باید در طرف منها قرار داشته باشد. برای درک این موضوع به مثالهای زیر توجه نمایید:

```
> x<-5
> 5->x
> variable<-1000
> 200->x1
```

حال اگر بخواهیم مقدار مثلا x یا variable و یا x1 را مشاهده کنیم کافی است اسم آنها را فقط بنویسیم.

```
> x
[1] 5
> variable
[1] 1000
> x1
[1] 200
```

البته برای نسبت دادن نام متغیرها می توان مانند سایر زبان های برنامه نویسی از عملگر "=" نیز استفاده کرد. اما باید توجه داشت که علامت "<-" کاربرد وسیع تری دارد، بدین معنی که می تواند به عنوان آرگومان یک تابع مورد استفاده قرار گیرد، در صورتی که علامت تساوی این خاصیت را ندارد. در بخش های بعد به یک نمونه از کاربردهای وسیع تر "<-" اشاره خواهیم کرد که در آن علامت تساوی کارساز نمی باشد.

تذکر: می توان یک عبارت را نیز به یک متغیر نسبت داد. برای مثال می توان نوشت:

```
> n<-(100*12)/8
> n
[1] 150
```

در اینجا لازم است برخی نکات را درباره نام متغیرها متذکر شویم:

۱- حروف کوچک و بزرگ در محیط R با هم فرق می کنند. یعنی محیط R نسبت به حروف کوچک و بزرگ حساس است.
۲- عدد نمی تواند در اول اسم متغیر قرار گیرد، یعنی نمی توانیم بنویسیم `1x` ولی می توانیم بنویسیم `x1`.
۳- برخی از حروف بزرگ در R نمایانگر علامت خاص یا دستورات خاصی می باشند و از آن ها برای نام دهی متغیرها نمی توان استفاده کرد مثل T (True) و F (False).
۴- اگر بخواهیم عبارتی را به یک متغیر نسبت دهیم و آن عبارت در بیش از یک سطر نوشته شود، آنگاه در سطر یا سطرهای بعدی، نرم افزار R ادامه کار را با علامت"+" نشان می دهد. برای درک بهتر این موضوع به مثال زیر توجه نمایید.

```
n<-(121*12)/
+ 8-
+ 42
> n
[1] 139.5
```

۳. انواع متغیرها و داده ها در R

همانطور که ملاحظه شد نرم افزار R با متغیرها کار می کند که خود آن ها توسط نام و محتوی مشخص می شوند. در ادامه خواهیم دید که هدف از تعریف این متغیرها، نسبت دادن داده ها به این متغیرها می باشد. در واقع برای سادگی کار با داده ها، ما به داده ها اسمی نسبت می دهیم که به آن ها متغیر می گوییم. هریک این متغیرها که خود معرف داده هایی می باشند دارای دو خصوصیت (attribute) می باشند که این دو خصوصیت عبارتند از : نوع (mode) و طول (length).

۱- نوع متغیرها

نوع متغیرها در R، با دستور زیر مشخص می شوند.

```
> mode()
```

در محیط R ما با چهار نوع متغیر سروکار خواهیم داشت که عبارتند از:

الف) متغیرهای عددی (Numeric)

اگر متغیر مورد نظر، مقادیر عددی را اختیار کند در این صورت نوع متغیر عددی خواهد بود.

مثال

```
n<-1000
> mode(n)
[1] "numeric"
```

ب) متغیرهای رشته ای یا کاراکتر (Character)

این نوع متغیرها، داده هایی که اختیار می کنند به صورت حروف یا کلمه می باشد و در واقع جزء داده های کیفی به حساب می آیند. هر یک از داده ها در چنین مواردی حتما باید مابین علامت "" بگیرند.

مثال

```
n<-"a"
> n
[1] "a"
> mode(n)
[1] "character"
```

باید توجه داشت که چهار عمل اصلی ریاضی را نمی توان بر روی چنین متغیرهایی اجرا کرد.

ج) متغیرهای مختلط(Complex)

اگر متغیر مورد نظر، داده هایی را که اختیار می کند به صورت اعداد مختلط باشند، در این صورت نوع متغیر مختلط خواهد بود.

مثال

```
c<-1-2i
> mode(c)
[1] "complex"
```

د) متغیرهای منطقی (Logical)

چنین متغیرهایی معمولا به صورت TRUE یا FALSE ظاهر می شوند و از نوع متغیرهای منطقی به حساب می آیند. باید توجه داشت که تمام حروف هر دو کلمه بزرگ نوشته شوند در غیر این صورت نرم افزار پیغام خطا به صورت " Error: object 'True' not found" می دهد. همچنین اگر این دو کلمه داخل گیومه نوشته شوند (حتی با حروف بزرگ)، آنگاه R نوع آن ها را به صورت رشته ای در نظر می گیرد. پس برای اینکه نوع آن ها را به صورت منطقی در نظر بگیرد نباید آن ها را داخل گیومه نوشت.

مثال

```
l<-"TRUE"
> mode(l)
[1] "character"
> l<-TRUE
> mode(l)
[1] "logical"
```

در پایان این بخش لازم است چند نکته را متذکر شویم. نخست اینکه نرم افزار R داده های گمشده (missing data or missing value) را با نماد NA(Not Available) نمایش می دهد (مثال درباره داده های گمشده را به قسمتهای بعد موکول می کنیم). همچنین این نرم افزار مقادیر عددی نامعین مانند $\pm\infty$ را به ترتیب با نمادهای Inf و Inf– نشان می دهد. اگر مقدار عدد نباشد، آن را با نماد NaN (Not a Number) نمایش می دهد. برای درک بهتر این موضوع به مثال زیر توجه کنید.

```
n<-5/0
> n
[1] Inf
> m<-0/0
> m
[1] NaN
```

۲- طول متغیرها

برای دانستن اینکه هر متغیر دارای چه تعداد عضو است می توانیم از دستور زیر استفاده کنیم. اهمیت این دستور در بخشهای بعد و به خصوص در بخش مربوط به بردارها و ماتریس ها بیشتر مشخص خواهد شد.

```
length()
```

```
n<-1000
> length(n)
[1] 1
```

۴. بردارها، ماتریسها و آرایه ها و اعمال روی آن ها

در این قسمت ما به معرفی بردار، ماتریس و آرایه در R می پردازیم و خصوصیات واعمال روی آن ها را به تفصیل مورد بحث قرار می دهیم. ذکر این نکته لازم است که اساس وارد کردن داده ها، که در فصل آینده به آن خواهیم پرداخت مبتنی بر بردارها و ماتریسها می باشد. لذا این فصل از اهمیت بسزایی برخوردار است.

بردارها در R

در ابتدا فرض کنید که بردار مد نظر به صورت x باشد. برای وارد کردن این بردار در R از دستور زیر می توانیم استفاده کنیم.

```
> assign("x",c())
```

دستور فوق شکل کلی ساختن یک بردار را نشان می دهد که در آن اسم بردار یعنی x بین دو گیومه قرار گرفته و پس از آن نماد c() محلی است که باید اعداد مربوط به این بردار را در آن وارد کنیم. به عنوان مثال فرض کنید بردار x به صورت x=[1,2,3,4,5] باشد. پس دستور آن در R به صورت زیر خواهد بود:

```
> assign("x",c(1,2,3,4,5))
> x
[1] 1 2 3 4 5
```

البته به صورت ساده تر نیز می توان بردار x را نوشت:

```
> x=c(1,2,3,4,5);x
[1] 1 2 3 4 5
```

اگر اعضای یک بردار همگی اعداد صحیح بوده که به صورت صعودی یا نزولی مرتب شده باشند آنگاه می توان با استفاده از دستور زیر آن ها را نوشت:

```
> a:b
```

که در آن a عدد شروع و b عدد پایان خواهد بود. به عنوان مثال فرض کنید بخواهیم اعداد ۱ تا ۱۰ را به صورت صعودی در بردار x و اعداد ۲۰ تا ۱۱ را به صورت نزولی در بردار y بریزیم. در این صورت داریم:

```
> x=c(1:10);x
 [1]  1  2  3  4  5  6  7  8  9 10
> y=c(20:11);y
 [1] 20 19 18 17 16 15 14 13 12 11
```

پس هر تعداد داده از یک متغیر را که داشته باشیم می توانیم در قالب یک بردار وارد R کنیم.

طول هر بردار را نیز می توانیم مشخص کنیم.

```
> length(x)
[1] 10
```

یک بردار از داده ها می تواند به صورت دنباله از اعداد نیز در نظر گرفته شود. شکل کلی دستور تولید یک دنباله از اعداد به صورت زیر است:

```
> seq(a,b,by=z)
Or
> seq(a,b,length=l)
```

که در آن a عدد شروع دنباله، b عدد پایان دنباله، z فاصله بین دو عدد متوالی و l طول دنباله می باشد.

مثال:

```
> seq(-2,2,length=10)
 [1] -2.0000000 -1.5555556 -1.1111111 -0.6666667 -0.2222222  0.2222222
     0.6666667  1.1111111  1.5555556  2.0000000
> x=seq(-2,2,by=.4);x
 [1] -2.0 -1.6 -1.2 -0.8 -0.4  0.0  0.4  0.8  1.2  1.6  2.0
> length(x)
[1] 11
```

در مثال فوق، فرض کنید که دنباله x را به عنوان یک بردار در نظر بگیریم و بخواهیم بدانیم که عضو i ام ($i=1,2,\ldots,11$) این بردار چه عددی است. شکل کلی این دستور برای این منظور، به صورت زیر است:

```
> x[i]
```

برای مثال، فرض کنید هدف این است که بدانیم عضو پنجم این بردار چه عددی است، یا اینکه در بردار مورد نظر هز عضو دوم تا عضو ششم شامل چه اعدادی می شوند.

```
> x[5]
[1] -0.4
> x[2:6]
[1] -1.5555556 -1.1111111 -0.6666667 -0.2222222  0.2222222
> x[x>1.2]
[1] 1.555556 2.000000
```

اگر بخواهیم از کل داده ها آن تعداد از داده ها را که بزرگتر یا کوچکتر از مقداری خاص باشند را انتخاب کنیم مانند آخرین دستور بالا عمل می کنیم. همچنین اگر بخواهیم از کل بردار داده ها قسمتی در نظر گرفته نشود از دستور x[-c()] استفاده می کنیم که در آن اعضای c() شماره آن اعضا از بردار x است که نمی خواهیم ظاهر شوند.

```
 > x=seq(-2,2,length=10)
> x[-c(1,3,5,7)]
[1] -1.5555556 -0.6666667  0.2222222  1.1111111  1.5555556  2.0000000
```

با استفاده از دستور rep() می توان یک داده یا بردای از داده ها را به هر تعداد دلخواه تکرار کرد. شکل کلی این دستور و مثال آن به صورت زیر می باشد:

```
> rep(x,each=n,times=m)
> x=c(1:4)
> rep(x,each=2,times=3)
 [1] 1 1 2 2 3 3 4 4 1 1 2 2 3 3 4 4 1 1 2 2 3 3 4 4
```

۵. اعمال روی بردارها

فرض کنید که A و B دو بردار با طول یکسان باشند. در این صورت چهار عمل اصلی روی این دو بردار، با دستورهای زیر امکان پذیر است.

A±B , A*B , A/B

در اینجا دو نکته وجود دارد. اول اینکه در عمل تقسیم، هیچیک از درایه های ماتریس B نباید صفر باشند و دوم اینکه این چهار عمل اصلی، به صورت درایه به درایه انجام می پذیرد، یعنی اینکه مثلا در عمل ضرب منظور، ضرب داخلی یا خارجی نمی باشد بلکه ضرب دو ماتریس در این حالت بدان معناست که هر درایه ماتریس A با درایه متناظر با خودش در ماتریس B ضرب می شود.

ضرب داخلی دو بردار A و B با دستور A%*%B و نیز با دستور crossprod(A,B) انجام پذیر است.

```
> A=c(1:10)
> B=c(-10:-1)
> crossprod(A,B)
     [,1]
[1,] -220
```

۶. ماتریس ها در R

ماتریس ها در واقع شکل کلی بردارها می باشند، یعنی حالت خاص یک ماتریس که دارای یک سطر یا ستون باشد یک بردار خواهد بود. شکل عمومی دستور یک ماتریس m*n در R به صورت زیر می باشد:

```
> matrix(data,nrow=m,ncol=n,byrow=T)
```

که در آن data داده ها یا برداری از داده ها، m تعداد سطرهای ماتریس، n تعداد ستونهای ماتریس می باشد. اگر گزینه byrow=T برقرار باشد، داده ها به صورت سطری به طور متوالی پست سر هم قرار می گیرند و اگر گزینه byrow=F برقرار باشد، داده ها به صورت ستونی به طور متوالی پشت سر هم قرار می گیرند. اگر این گزینه اصلا نوشته نشود، نرم افزار به صورت پیش فرض داده ها را به صورت ستونی به طور متوالی پشت سر هم قرار می دهد.

مثال

```
> matrix(1:9,3,3,byrow=T)
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
[3,]    7    8    9
```

نکته ای که درباره ماتریس ها باید متذکر شویم این است که بردار داده ها و یا داده ها بایستی از نظر تعداد به گونه ای باشند که تعداد آن ها برابر m*n باشد، زیرا در غیر این صورت نرم افزار R با دادن یک پیغام هشدار به این صورت عمل می کند که اگر تعداد داده ها کمتر از m*n باشد آنگاه کمبود داده خود را با تکرار داده ها از اول تا جایی که کمبود جبران شود ادامه می دهد. برای روشن شدن این مسئله به مثال زیر توجه نمایید:

```
> matrix(1:5,3,3)

     [,1] [,2] [,3]

[1,]    1    4    2

[2,]    2    5    3

[3,]    3    1    4

Warning message:

In matrix(1:5, 3, 3) :

  data length [5] is not a sub-multiple or multiple of the number of
rows [3]
```

حال اگر تعداد داده ها بیشتر اندازه (طول) ماتریس باشد، آنگاه بازهم یک پیغام هشدار داده و داده ها را از اول تا جایی که برای تشکیل یک ماتریس m*n لازم باشد، مورد استفاده قرار می دهد و مابقی را بدون استفاده می گذارد. برای درک بهتر این موضوع، به مثال زیر توجه کنید:

```
> matrix(1:16,3,4)

     [,1] [,2] [,3] [,4]

[1,]    1    4    7   10

[2,]    2    5    8   11

[3,]    3    6    9   12

Warning message:

In matrix(1:16, 3, 4) :

  data length [16] is not a sub-multiple or multiple of the number of
rows [3]
```

تذکر: با استفاده از دستور length می توان طول یک ماتریس (یعنی تعداد اعضای آن ماتریس) را بدست آورد.

۷. اعمال روی ماتریس ها

اعمالی که در R می توان روی ماتریس ها پیاده کرد بسیار می باشد که در این بخش ما بدان ها می پردازیم. در ابتدا با چهار عمل اصلی آغاز می کنیم. فرض کنید که $A_{m*n}=[a_{ij}]$ و $B_{m*n}=[b_{ij}]$ دو ماتریس m*n باشند. در این صورت داریم

$A\pm B=[a_{ij}\pm b_{ij}]$, $A*B=[a_{ij}*b_{ij}]$, $A/B=[a_{ij}/b_{ij}]; b_{ij}\neq 0$

سایر اعمال روی ماتریس ها و دستورهای آن ها در R ، با فرض اینکه $A_{m*n}=[a_{ij}]$ و $B_{m*n}=[b_{ij}]$ دو ماتریس m*n باشند عبارتند از:

ترانهاده ماتریس: t(A)

ضرب ماتریس ها (نه به صورت درایه نظیر به نظیر): crossprod(A,B) یا A%*%B

ماتریس قطری: برای ساختن یک ماتریس قطری، ابتدا لازم است اعضای روی قطر اصلی را مشخص کنید. مثلا فرض کنید بردار x بیانگر اعضای روی قطر اصلی ماتریس قطری D باشد. در این صورت با استفاده از دستور D=diag(x) می توان ماتریس قطری D را تشکیل داد.

مثال

```
> x=c(1:3)
> D=diag(x);D
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    2    0
[3,]    0    0    3
```

ماتریس همانی: برای ساختن یک ماتریس قطری، کافی است در دستور diag(x) به جای x یک عدد صحیح مثبت قرار دهیم مثلا 3. در این صورت دستور diag(3) یک ماتریس همانی 3*3 را خواهد ساخت.

```
> D=diag(3);D
     [,1] [,2] [,3]
[1,]    1    0    0
[2,]    0    1    0
```

```
[3,]    0    0    1
```

اثر(trace) ماتریس: اگر ماتریس A یک ماتریس مربعی باشد در آن صورت، اثر ماتریس A برابر است با مجموع اعضای روی قطر اصلی ماتریس A یعنی: sum(diag(A))

دترمینان ماتریس: اگر A یک ماتریس مربعی باشد آنگاه دترمینان آن عبارتست از: det(A)

معکوس ماتریس: اگر A یک ماتریس مربعی باشد که دترمینان آن مخالف صفر باشد در آن صورت معکوس ماتریس A از دستور solve(A) بدست می آید.

مقادیر ویژه و بردارهای ویژه: فرض کنید ماتریس A یک ماتریس مربعی باشد، در این صورت مقادیر ویژه و بردارهای ویژه آن با استفاده از دستور eigen(A) بدست می آید.

مثال

```
> A=matrix(c(-1,2,4,0,3,-3,5,6,1),3,3)
> sum(diag(A))
[1] 3
> det(A)
[1] -111
> solve(A)
           [,1]       [,2]        [,3]
[1,] -0.1891892 0.13513514  0.13513514
[2,] -0.1981982 0.18918919 -0.14414414
[3,]  0.1621622 0.02702703  0.02702703
> eigen(A)
$values
[1]  3.577099+3.73153i  3.577099-3.73153i -4.154198+0.00000i
$vectors
                      [,1]                  [,2]          [,3]
[1,] -0.2294043-0.276754i -0.2294043+0.276754i -0.8275454+0i
[2,] -0.8309781+0.000000i -0.8309781+0.000000i -0.2064801+0i
```

```
[3,] -0.0034580-0.424552i -0.0034580+0.424552i  0.5220484+0i
```

حال اگر بخواهیم که فقط مقادیر ویژه (یا بردارهای ویژه) را بدست آوریم، کافی است به صورت زیر عمل کنیم:

```
> eigen(A)$values
[1]  3.577099+3.73153i  3.577099-3.73153i -4.154198+0.00000i
> eigen(A)$vectors
                      [,1]                  [,2]          [,3]
[1,] -0.2294043-0.276754i -0.2294043+0.276754i -0.8275454+0i
[2,] -0.8309781+0.000000i -0.8309781+0.000000i -0.2064801+0i
[3,] -0.0034580-0.424552i -0.0034580+0.424552i  0.5220484+0i
```

همه توابع ریاضی و آماری را می توان بر روی ماتریس ها و بردار پیاده نمود که این موضوع را به بخش مربوط به توابع موکول می کنیم.

۸. ویژگی های ماتریس ها در R

ماتریس ها در R دارای ویژگی های متعددی هستند که موجب افزایش کارایی آن ها و سادگی استفاده از آن ها می شود. در این بخش به این ویژگی ها خواهیم پرداخت.

۱- برای ساختن یک ماتریس علاوه بر استفاده از دستور matrix()، می توان از دستور dim() میز استفاده کرد.

```
> m=1:12
> dim(m)=c(3,4)
> m
     [,1] [,2] [,3] [,4]
[1,]    1    4    7   10
[2,]    2    5    8   11
[3,]    3    6    9   12
```

البته می توان از دستور dim() برای مشخص نمودن ابعاد یک ماتریس نیز استفاده کرد. به عنوان مثال اگر در مثال فوق بنویسیم dim(m) آنگاه جواب به صورت 4 3 ظاهر می شود که عدد اول از سمت چپ، تعداد سطرها و عدد بعد تعداد ستونهای ماتریس مورد نظر را نشان می دهد.

۲- انتخاب یک زیر مجموعه از یک ماتریس مشابه انتخاب یک زیر مجموعه از یک بردار است و تنها تفاوت آن این است که ماتریس ها دارای دو بعد بوده و لذا برای انتخاب یک عضو باید سطر و ستون آن را مشخص نمود. در حالت کلی برای انتخاب عضو سطر i ام از ستون j ام از دستور زیر استفاده می کنیم.

```
> x=matrix(1:15,3,5);x
     [,1] [,2] [,3] [,4] [,5]
[1,]    1    4    7   10   13
[2,]    2    5    8   11   14
[3,]    3    6    9   12   15
> x[1,3]
[1] 7
```

برای انتخاب چند سطر یا ستون از یک ماتریس می توان از تابع c() و برای حذف چند سطر یا ستون از تابع c()– استفاده کرد.

```
> x[c(2,3),]
     [,1] [,2] [,3] [,4] [,5]
[1,]    2    5    8   11   14
[2,]    3    6    9   12   15
> x[c(2,3),1]
[1] 2 3
> x[2,c(1,3)]
[1] 2 8
> x[-c(1,2),]
[1]  3  6  9 12 15
```

۳- ویژگی بعد درباره ترکیب ماتریس ها می باشد. اگر بخواهیم ماتریس ها را به صورت سطری با هم ترکیب کنیم از تابع rbind() و اگر بخواهیم آن ها را به صورت ستونی ترکیب کنیم از تابع cbind() استفاده می کنیم.

مثال:

```
> x1=matrix(1:6,2,3)
> x2=matrix(10:18,3,3)
> rbind(x1,x2)
     [,1] [,2] [,3]
[1,]    1    3    5
[2,]    2    4    6
[3,]   10   13   16
[4,]   11   14   17
[5,]   12   15   18
> y1=matrix(101:106,2,3)
> y2=matrix(-6:1,2,4)
> cbind(y1,y2)
     [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,]  101  103  105   -6   -4   -2    0
[2,]  102  104  106   -5   -3   -1    1
```

نکته ای که در اینجا باید به آن دقت نمود این است که هنگام استفاده از دستور rbind() بایستی تعداد ستونهای ماتریس ها و هنگام استفاده از دستور cbind() باید تعداد سطرهای ماتریس ها با هم برابر باشند.

۴- ماتریس ها نیز مانند بردارها دارای خصوصیاتی مانند طول و نوع می باشند که مشابه قبل به ترتیب با دستورهای length() و mode() قابل بررسی می باشند. علاوه بر این، ماتریس ها دارای دو خصوصیت دیگر می باشند که عبارتند از: بعد و نام بعد. همانطور که قبلا اشاره شد برای بدست آوردن بعد یک ماتریس از دستور dim() استفاده می کنیم. ویژگی نام بعد یک ماتریس برای نام گذاری سطر و ستون یک ماتریس است که با دستور dimnames() انجام می شود. استفاده از این دستور برای نام گذاری سطر و ستون یک ماتریس، به دو طریق انجام می شود که در قالب دو مثال هر دو روش را شرح می دهیم.

روش اول

```
> x=matrix(1:12,3,4)
>dimnames(x)=list(paste("row",letters[1:3]),paste("col",LETTERS[1
:4]))
> x
      col A col B col C col D
row a     1     4     7    10
row b     2     5     8    11
row c     3     6     9    12
```

در مثال بالا letters و LETTERS بردارهای کارکتری (رشته ای) با مقادیر حروف الفبای کوچک و بزرگ هستند . اگر بخواهیم فقط سطرها (ستونها) را نام گذاری کنیم، کافی است در تابع list() به جای مولفه مربوط به سطر (ستون) کلمه NULL را بنویسیم. برای مثال اگر در ماتریس x بخواهیم فقط ستون ها را نام گذاری کنیم خواهیم داشت:

```
> dimnames(x)=list(NULL,paste("col",LETTERS[1:4]))
> x
     col A col B col C col D
[1,]     1     4     7    10
[2,]     2     5     8    11
[3,]     3     6     9    12
```

روش دوم

```
> x=matrix(1:12,3,4,dimnames=list(NULL,paste("col",LETTERS[1:4])))
> x
     col A col B col C col D
[1,]     1     4     7    10
[2,]     2     5     8    11
[3,]     3     6     9    12
```

تذکر مهم:

در بخش تعریف متغیرها در R ،درباره تفاوت استفاده از علامت "=" و علامت "<-" اشاره ای شد، اما تفاوت آن ذکر نشد. در اینجا به تفاوت استفاده از دو علا مت می پردازیم. مثال اول کاربرد علامت تساوی در دستور (آرگومان) ماتریس است که با خطا مواجه می شود.

```
> matrix(a=10,5,5)
Error in matrix(a = 10, 5, 5) : unused argument(s) (a = 10)
```

مثال دوم کاربرد علامت "<-" را در آرگومان ماتریس نشان می دهد.

```
> matrix(a<-10,5,5)
     [,1] [,2] [,3] [,4] [,5]
[1,]   10   10   10   10   10
[2,]   10   10   10   10   10
[3,]   10   10   10   10   10
[4,]   10   10   10   10   10
[5,]   10   10   10   10   10
```

۹. آرایه ها در R

آرایه شکل تعمیم یافته ماتریس می باشد. در آرایه ها بعد بیش از دو می باشد. برای تشکیل آرایه در R از دستور array() به صورت زیر استفاده می کنیم که در آن شناسه اول بیانگر بردار داده ها و شناسه دوم، ابعاد آرایه را مشخص می کند.

```
> array(c(),dim=)
> array(1:24,dim=c(4,3,2))
, , 1

     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12

, , 2

     [,1] [,2] [,3]
```

```
[1,]   13   17   21
[2,]   14   18   22
[3,]   15   19   23
[4,]   16   20   24
```

با استفاده از دستور dim() نیز می توان یک آرایه ساخت. ما این روش را با ذکر مثال زیر توضیح می دهیم.

مثال

```
> x=1:24
> dim(x)=c(4,3,2)
> x
, , 1

     [,1] [,2] [,3]
[1,]    1    5    9
[2,]    2    6   10
[3,]    3    7   11
[4,]    4    8   12

, , 2

     [,1] [,2] [,3]
[1,]   13   17   21
[2,]   14   18   22
[3,]   15   19   23
[4,]   16   20   24
```

تذکر: تمام ویژگی های ذکر شده در بخش قبل در مورد ماتریس ها، درباره آرایه ها نیز برقرار می باشد و لذا ما از تکرار مجدد آن ها خودداری می کنیم.

۱۰. وارد کردن داده ها در R

یکی از مهمترین مراحل کار کردن با نرم افزار R چگونگی وارد کردن داده ها در آن می باشد. برای ورود داده ها در R روشهای متفاوتی وجود دارد که ما در اینجا به اختصار به آن ها می پردازیم. شایان ذکر است که نوع داده هایی که در این کتاب استفاده می شود اغلب از نوع عددی و رشته ای است. در حالت کلی می توانیم ورود داده ها در R را به دو دسته کلی تقسیم بندی کنیم: دسته اول عبارتست از اینکه داده های موجود را به طور مستقیم در R وارد کنیم که در این حالت نیز چند روش وجود دارد. دسته دوم عبارتست از اینکه داده ها را از فایلی دیگری فراخوانی کنیم. ما در اینجا هر دو دسته را مورد بررسی قرار داده و در طول این کتاب از هر دو روش استفاده می کنیم.

الف) وارد کردن داده ها به طور مستقیم در R

برای ورود داده ها به صورت مستقیم، اساس کار بر پایه بردارها می باشد که در قسمتهای گذشته به تشریح آنها پرداختیم. همانطور که قبلا اشاره شد برای تشکیل یک بردار از تابع c() می توانیم استفاده کنیم تا بتوان داده را در قالب یک بردار در R تعریف کرد. این بردار وارد شده را می توان به عنوان یک متغیر مثلا x1 در نظر گرفت. حال فرض کنید که به جای یک متغیر، سه متغیر داشته باشیم که دو تا از آن ها دارای داده های کمی (عددی) و دیگری دارای داده ها کیفی (کاراکتری) باشد. حال می خواهیم این سه متغیر را که مثلا x1، x2 و x3 نام دارند را وارد R کنیم.

```
> x1=c(10,12,14,16,18,20);x1
[1] 10 12 14 16 18 20
> x2=c("male","female","male","female","male","female");x2
[1] "male"   "female" "male"   "female" "male"   "female"
> x3=c(1.2,2.3,1.5,2.1,1.1,2.5);x3
[1] 1.2 2.3 1.5 2.1 1.1 2.5
```

نکته ای که در اینجا وجود دارد این است که هر یک از متغیرها به طور جداگانه وارد شده اند و نیز به طور جداگانه قرار گرفته اند. حال اگر داده های مربوط به این سه متغیر، از یک نمونه ۶تایی بدست آمده باشند و بخواهیم آن ها به صورت سه ستون به عنوان سه متغیر نمایش دهیم (مانند ستونهای متغیرها در نرم افزارهایی مانند SPSS یا MiniTab) در این صورت می توانیم از روش داده های چارچوب دار (Data frame) استفاده کنیم.

داده های چارچوب دار (Data frame)

داده های چارچوب دار دارای ستون های متعدد با نوع داده های مختلف است و در حقیقت مناسب ترین ساختار داده ها در تجزیه و تحلیل در R می باشد. در واقع، برای اجرای اکثر روشهای تجزیه و تحلیل آماری در نرم افزار R، نیازمند داده هایی با ساختار داده های چارچوب دار هستیم.

برای ایجاد چنین ساختاری از داده ها، روشهای مختلفی از جمله فراخوانی داده ها از طریق فایل وجود دارد که آن را در بخش بعدی توضیح می دهیم. روش دیگر برای ایجاد داده های چارچوب دار، استفاده از دستور زیر است:

```
> data.frame()
```

حال فرض کنید که در مثال قبل بخواهیم متغیرهای x1، x2 و x3 را در قالب داده های چارچوب دار وارد R کنیم. برای این منظور، در ابتدا آن ها را به صورت سه بردار همانند قبل وارد کرده و سپس این سه بردار را به صورت زیر در قالب داده های چارچوب دار به R معرفی می کنیم.

```
> data=data.frame(x1,x2,x3);data
  x1     x2  x3
1 10   male 1.2
2 12 female 2.3
3 14   male 1.5
4 16 female 2.1
5 18   male 1.1
6 20 female 2.5
```

داده های چارچوب دار را می توان به منزله یک صفحه گسترده (Spreadsheet) تصور کرد. هر ستون داده های چارچوب دار، یک بردار از داده ها می باشد. داخل هر ستون (بردار)، تمام عناصر از یک نوع (mode) هستند، اما ستونهای مختلف می توانند

دارای داده هایی با انواع متفاوت باشند. با این وجود، تمامی ستونها در این ساختار دارای طول یکسان می باشند. به عنوان مثال اگر طول هر یک از ستونها (بردار) را در مثال اخیر اندازه بگیریم مشخص خواهد شد که هر سه دارای طول ۶ می باشند.

```
> length(x1);length(x2);length(x3)
[1] 6
[1] 6
[1] 6
```

از دیگر خواص مهم داده های چارچوب دار می توان به این نکته اشاره کرد که در چنین ساختاری، داده ها می توانند دارای خواص اسامی و نام در سطرها و ستونها باشند. برای مثال، نام سطرها و ستونها در داده های چارچوب دار data را می توان به صورت زیر مشخص نمود.

```
> names(data)
[1] "x1" "x2" "x3"
> rownames(data)
[1] "1" "2" "3" "4" "5" "6"
```

داده های چارچوب دار حتی این خاصیت را دارند که می توان نام سطرها و ستونها آن را تغییر داد. به عنوان مثال فرض کنید در داده های data بخواهیم نام ستونها را از x1، x2 و x3 بترتیب به var1، var2 و var3 و نیز نام سطرها را از 6,5,4,3,2,1 بترتیب به sample2, sample1, sample6, sample5, sample4, sample3 تغییر دهیم، برای این منظور از دستورهای زیر استفاده می کنیم:

```
> names(data)=c("var1","var2","var3")
>rownames(data)=c("sample1","sample2","sample3","sample4","sample5","sample6")
> data
        var1   var2 var3
sample1   10   male  1.2
sample2   12 female  2.3
sample3   14   male  1.5
sample4   16 female  2.1
sample5   18   male  1.1
```

```
sample6   20 female  2.5
```

درباره داده های چارچوب دار، مطلب بسیار است که در جای خود به آن ها خواهیم پرداخت.

در اینجا به روش دیگری که برای وارد کردن داده ها استفاده می شود می پردازیم.

ب) فراخوانی داده ها از فایلی دیگر

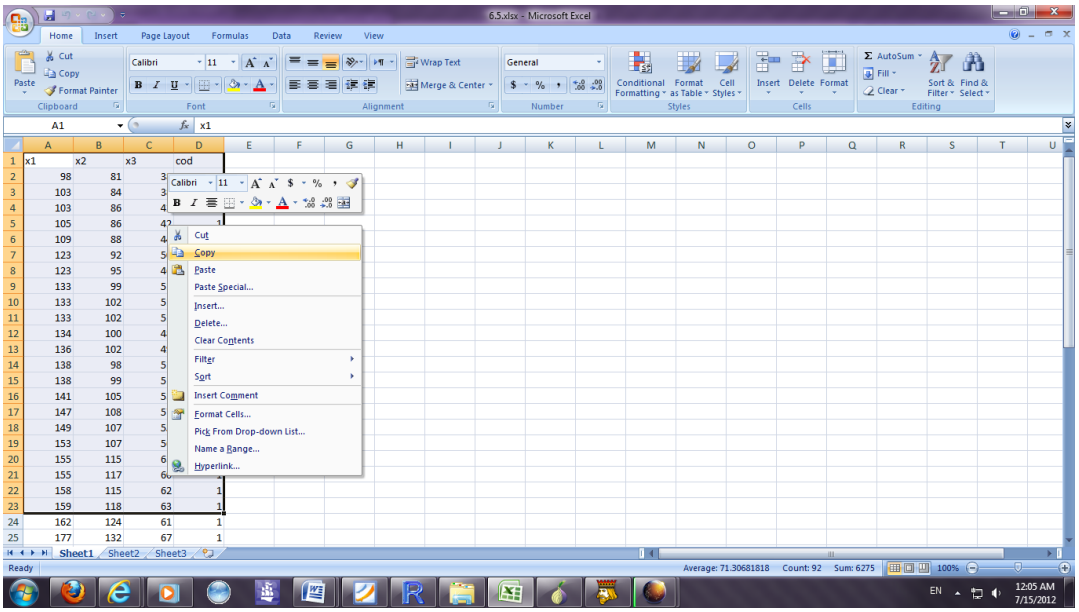
نرم افزار R این قابلیت را دارد که فایل داده ها را با پسوندهای مختلفی بخواند، بدین معنا که این نرم افزار قادر است فایل داده هایی را که به صورت Excel یا Notepad می باشند را فراخوانی کند. برای فراخوانی داده ها از فایلهای دیگر، دستورهای متعددی وجود دارد که ما به برخی از آن ها اشاره خواهیم کرد.

اولین دستور که برای فراخوانی داده ها از آن استفاده می شود دستور read.table است که شکلی عمومی این دستور به قرار زیر می باشد

```
> read.table(x,file=””)
```

این دستور خود به طرق مختلف می تواند داده ها را از فایلهایی مانند Excel و Notepad فراخوانی کند. ساده ترین و شاید در عین حال بهترین روش فراخوانی داده ها از طریق این دستور، شیوه ای است که در زیر آن را توضیح می دهیم:

فرض کنید داده ها در یک فایل Excel (یا Notepad) قرار داشته باشند. ابتدا کل داده ها یا آن بخشی از داده ها را که قرار است وارد R کنیم را انتخاب کرده و آن ها را مطابق شکل زیر کپی می کنیم:



سپس با دستور read.table به صورت زیر آن را فراخوانی می کنیم:

```
> mydata=read.table("clipboard",header=T)

> mydata

    x1   x2 x3 cod

1    98   81 38   1

2   103   84 38   1

3   103   86 42   1

4   105   86 42   1

5   109   88 44   1

6   123   92 50   1

7   123   95 46   1

8   133   99 51   1

9   133 102 51   1

10 133 102 51   1


11 134 100 48   1
```

باید دقت نمود که اگر header=T باشد آنگاه اسم متغیرها هم کپی می شود ولی اگر header=F باشد آنگاه اسم متغیرها کپی نمی شود.

روش دیگر برای استفاده از دستور read.table، استفاده از آدرس فایل است. اگر یک فایل Notepad (یا Excel) را داشته باشیم که در محلی مثلا درایو E، فولدر Data و به نام data1.txt ذخیره شده باشد و بخواهیم آن را فراخوانی کنیم می توانیم با استفاده از دستور read.table و به یکی از دو شیوه زیر آدرس فایل را در این دستور وارد کرده و داده ها را فراخوانی کنیم.

```
> data1=read.table("E:/Data/data1.txt",header=T)
```

Or

```
>data1=read.table("E:\\Data\\data1.txt",header=T)
```

علاوه بر شناسه های file و header شناسه های دیگری هم در دستور ()read.table وجود دارد که در برخی موارد استفاده از آن ها لازم می باشد. این شناسه ها عبارتند از:

- شناسه sep=""

در این شناسه کاراکتر جداکننده تعیین می شود. مقدار قراردادی این شناسه فاصله خالی است. این شناسه معمولا زمانی مورد استفاده قرار می گیرد که داده ها به صورت فایل متنی باشند.

- شناسه row.names=NULL

مقدار این شناسه برداری کاراکتری شامل نام سطرها است. زمانیکه به صورت NULL باشد یعنی سطرها هیچ نامی نمی گیرند.

- شناسه col.names=NULL

مقدار این شناسه برداری کاراکتری شامل نام ستون هاست.

یکی دیگری از روشهای فراخوانی داده ها، استفاده از دستور file.choose است. شکل کلی این دستور به صورت زیر است:

```
> file.choose()
```

با اجرای این دستور، پنجره ای باز می شود که تنها کافی است مسیر محلی که فایل مورد نظر ذخیره است را به آن بدهیم.

تذکر: دستور file.choose با دستور read.table به صورت زیر نیز کار می کند:

```
> data=read.table(file.choose())
```

فصل ۳

توابع

۱. مقدمه

در این فصل به بررسی توابع مختلف در R می پردازیم. به طور کلی توابعی که در R مورد استفاده قرار می گیرند را می توان به دو گروه تقسیم کرد: گروه اول شامل توابعی است که صرفا مخصوص خود نرم افزار R است و برای چگونگی کار با این نرم افزار، ورود، خواندن، شناسایی داده ها و ... مورد استفاده قرار می گیرند و گروه دوم شامل توابع محاسباتی می باشند که این گروه، تمام توابع ریاضی و آماری را در بر می گیرد. ما در این فصل هر دو گروه از توابع را معرفی کرده و درباره آن ها بحث خواهیم کرد. البته توابع محاسباتی را در فصل های بعد و هنگام استفاده از روشهای مختلف آماری برای تجزیه و تحلیل داده ها بیشتر مورد استفاده قرار می دهیم و کاربرد اصلی آنها را در آینده مشاهده خواهید کرد.

۲. توابع خاص نرم افزار R

این گروه از توابع، خود شامل انواع مختلفی از توابع می باشند که عبارتند از: توابع مقدماتی R، توابع مربوط به داده ها و توابع آزمون و تبدیل ساختارها. برخی از توابع در این بخش، مشترک با قسمت های قبلی بوده و لذا ما از توضیح اضافی پیرامون آن ها خودداری خواهیم کرد.

۲-۱ :توابع مقدماتی R

توابع مقدماتی R شامل توابع بسیار ساده و ابتدایی می باشند که در برخی موارد بسیار کارا می باشند. این توابع عبارتند از:

- تابع date()

این تابع تاریخ و زمان را نشان می دهد.

- تابع help()

این تابع همانطور که قبلا نیز اشاره شد راهنمای نرم افزار جهت دستیابی به دستورات و اطلاعات درباره یک موضوع خاص می باشد.

- تابع ls() و تابع objects()

هر دوی این توابع، فهرست تمام داده های استفاده و وارد شده توسط کاربر در نرم افزار را نشان می دهند.

- تابع data()

فهرست تمام داده های موجود در نرم افزار R که هنگام نصب بر روی سیستم ایجاد می شود و قابل دسترسی می باشد را نشان می دهد.

- تابع page()

این تابع متن فایل، برنامه یا ساختار داده ای را که به عنوان شناسه به آن می دهید را به صورت صفحه به صفحه نمایش می دهد.

- تابع q()

برای خروج از نرم افزار R از این تابع می توانید استفاده کنید.

- تابع rm()

این تابع، ساختار داده ای را که به آن معرفی شده را حذف می کند.

- تابع search()

این تابع فهرستی از بسته های اضافه شده به نرم افزار را به ما می دهد.

توابع مقدماتی دیگری نیز وجود دارد که بیان آن ها از حوصله این کتاب خارج است.

۲-۲ : توابع مربوط به داده ها

- تابع attach()

این تابع یکی از مهمترین و پرکاربرد ترین توابع در R برای کار کردن با داده ها می باشد. این تابع کاربردهای فراوانی دارد و ما در اینجا مهمترین کاربرد این تابع را معرفی کرده و بقیه کاربردها را به خواننده واگذار می کنیم که وی می تواند برای اطلاع بیشتر به راهنمای R مراجعه نماید.

کاربرد اساسی این دستور آن است که امکان دستیابی کاربر به تک تک اجزاء و متغیرهای موجود در داده ها(داده های چاچوب دار) را باز می کند. برای تشریح کاربرد دستور attach() با طرح این مسئله، بحث را شروع می کنیم. فرض کنید که یک فایل ذخیره شده از داده ها (یا داده های چارچوب دار) را داشته باشیم و بخواهیم از طریق R به یک یا چند متغیر از این داده ها دسترسی داشته و آن ها را مورد بررسی قرار دهیم. در نگاه اول به نظر می رسد که با دستور read.table() و یا file.choose() این عمل امکان پذیر باشد ولی این دستورها کل داده ها را فراخوانی کرده و ما نمی توانیم یک یا چند متغیر آن دسترسی داشته و بتوانیم روی آن تجزیه و تحلیل انجام دهیم. اما دستور attach() به همراه دستور read.table() این کار را برای ما انجام می دهد. روند کار بدین گونه است که ما ابتدا از طریق دستور read.table() داده ها را فراخوانی کرده و سپس با استفاده از دستور attach()، امکان دسترسی به هر یک از متغیرهای موجود فایل داده ها را فراهم می کنیم. برای فهم بهتر این موضوع به مثال زیر توجه کنید.

```
>data1=read.table("E:\\My    document\\Data\\Data
for R\\data1.txt",header=T)
> data1
   X1  X2 X3 Code
1 109  88 44    1
2 123  92 50    1
3 123  95 46    1
4 133  99 51    1
5 133 102 51    1
6 133 102 51    1
7 134 100 48    1
8 136 102 49    1
> X1
Error: object 'X1' not found
> attach(data1)
> X1
[1] 109 123 123 133 133 133 134 136
```

- تابع detach()

این تابع عکس تابع attach() را انجام می دهد. یعنی زمانیکه کاربر این تابع را روی داده هایی (داده های چاچوب داری) اجرا کند آنگاه دیگر دسترسی به اجزاء آن داده ها (داده های چاچوبدار) امکان پذیر نمی باشد. (به عنوان مثال روند مثال قبل را با تابع detach() اجرا کرده و سپس متغیر X1 را از نرم افزار بخواهید).

- تابع list()

برای ایجاد یک فهرست از اشیاء و داده ها با ساختارهای مختلف می توان از تابع list() استفاده کرد. این تابع مشابه یک بردار عمل می کند، با این تفاوت که هر عنصر در این تابع می تواند شامل هر نوع داده و هر ساختاری باشد. لذا هر فهرست براساس این تابع خود می تواند شامل فهرست دیگری باشد، بنابراین می توان برای ساختارهای مختلف داده ها از آن استفاده نمود. این تابع اغلب برای خروجی آماری در R مورد استفاده قرار می گیرد. موجودیت خروجی اغلب شامل مجموعه ای از برآوردها پارامترها، باقیمانده ها و ... می باشد. اسامی مولفه های فهرست و محتوای مولفه های فهرست آرگومان های این تابع هستند.

```
> x=1:7
> y=c(T,F,F,T,T,F,T)
> z=list(numbers=x,logical=y)
> z
$numbers
[1] 1 2 3 4 5 6 7

$logical
[1]  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE
```

مشاهده می شود که که در سمت چپ عملگر "=" نام مولفه و طرف راست آن یک شیء از زبان R وجود دارد. مرتبه مولفه به ترتیب قرار گرفتن آن ها از چپ به راست است. در مثال فوق شیء منطقی logical مولفه دوم فهرست z می باشد. حال اگر فقط بخواهیم یکی از مولفه های فهرست چاپ شود کافی است به صورت زیر عمل کنیم.

```
> z$numbers

[1] 1 2 3 4 5 6 7
```

تابع names() می تواند اسامی مولفه های موجود در فهرست را ارائه دهد. با این تابع می توان اسامی مولفه ها را نیز تغییر داد. همچنین می توان مولفه دیگری را به فهرست اضافه کرد.

مثال

```
> z[[3]]=-10:1

> z

$numbers

[1] 1 2 3 4 5 6 7
```

```
$logical
[1]  TRUE FALSE FALSE  TRUE  TRUE FALSE  TRUE


[[3]]
 [1] -10  -9  -8  -7  -6  -5  -4  -3  -2  -1   0   1
```

نکته ای که درباره فهرستها باید بدان توجه کرد این است که استفاده از یک کروشه و یا دو کروشه دارای مفاهیم متفاوتی می باشد. بدین معنا که اگر از یک کروشه استفاده کنیم مولفه ای که حاصل می شود خود یک فهرست است، اما زمانی که از دو کروشه استفاده می کنیم فقط خود مولفه بدست می آید.

مثال

```
> z[1]
$numbers
[1] 1 2 3 4 5 6 7


> z[[1]]
[1] 1 2 3 4 5 6 7
```

- تابع factor()

این تابع برای نمایش داده های کیفی در R مورد استفاده قرار می گیرد. داده هایی از قبیل جنسیت، وضعیت تاهل، وضعیت اسکان و ... جزء داده های کیفی به حساب می آیند. فرض کنید بخواهیم افراد را بر حسب وضعیت تاهل آن ها دسته بندی کنیم و نتیجه را در قالب یک بردار عددی بیان کنیم. توجه داشته باشید که این بردار عددی بدست آمده قابلیت پیاده کردن اعمال ریاضی روی آن وجود ندارد. گیریم عدد ۱ نشاندهنده افراد مجرد و عدد ۲ نشاندهنده افراد متاهل باشد.

```
> class=c(1,2,2,1,1,2,1)
> class
[1] 1 2 2 1 1 2 1
> factor(class)
```

```
[1] 1 2 2 1 1 2 1

Levels: 1 2
```

با توجه به مثال فوق مشاهده می شود که استفاده از تابع factor() باعث ایجاد سطوح مختلف بین داده ها می شود. حال اگر بخواهید یک عملیات ریاضی را روی بردار class پیاده نمایید این امر امکان پذیر است ولی این موضوع درباره بردار factor(class) صادق نیست، چون تابع factor() علاوه بر سطح بندی داده ها، داده ها را به عنوان داده های کیفی به نرم افزار R معرفی می کند.

- تابع scan()

این تابع یک بردار می سازد. البته تابع scan() کاربردهای دیگری نیز دارد که برای فراخوانی داده ها مورد استفاده قرار می گیرد.

مثال

```
> a=scan()

1: 5

2: 6

3: 2

4: 3

5: 8

6:

Read 5 items

> a

[1] 5 6 2 3 8
```

- تابع save()

این تابع تمام محیط کاری را که با آن کار می کنیم را می تواند ذخیره کند. از این تابع به دو طریق می توان استفاده کرد: یک روش این است که تمام محیط کار را ذخیره کنیم که آن با دستور زیر انجام می شود:

```
> save.image("D:/name of file.Rdata")
```

که در آن و در داخل گیومه مسیر جایی که قرار است محیط کار ذخیره شود و نیز نام فایل نوشته می شود و با پسوند Rdata. ذخیره خواهد شد.

در روش دیگر ما به جای ذخیره کردن تمام میحط کار، فقط آن بخشی از متغیرها را که لازم است ذخیره می کنیم. در این حالت ذخیره کردن با دستور زیر انجام می شود:

```
> save(var1,var2,…, file="D:/name of file.Rdata")
```

که در آن var1، var2 و... نام متغیرهایی است که قرار است ذخیره شوند.

- تابع cat()

از این تابع برای ذخیره کردن یک بردار می توان استفاده کرد.

```
> cat(name of vector,file="D:/name of file.txt")
```

- تابع write.table()

این تابع یک داده چارچوب دار را ذخیره می کند. در واقع این تابع متناظر با تابع read.table() می باشد. شکل عمومی این دستور به صورت زیر می باشد:

```
> write.table(name of data.frame,"D:/name of file.txt")
```

۳-۲:توابع آزمون و تبدیل ساختارها

می دانیم که در R هر داده دارای ساختاری مشخص است مثلا: بردار، آرایه، داده چارچوب دار و ... بر این اساس برای هر نوع ساختار داده، تابعی برای آزمون کردن نوع ساختار آن و تابعی برای تبدیل نوع ساختار به نوعی دیگر وجود دارد که ما در اینجا بدان خواهیم پرداخت. البته باید توجه داشت که نوع برخی از ساختارها منحصر بفرد نیست. مثلا بردار یک ماتریس و ماتریس نیز به نوبه خود یک آرایه می باشد. اگر ساختاری از نوع ساختار نام برده شده در تابع آزمون باشد، نتیجه تابع آزمون مقدار منطقی T و در غیر این صورت مقدار منطقی F است. جدول توابع آزمون و تبدیل ساختارها به صورت زیر می باشد و براساس آن ها مثال هایی را در ادامه ارائه خواهیم کرد.

جدول توابع آزمون و تبدیل ساختارها

ساختار	تابع آزمون	تابع تبدیل
بردار	is.vector()	as.vector()
ماتریس	is.matrix()	as.matrix()
آرایه	is.array()	as.array()
فهرست	is.list()	as.list()
عامل	is.factor()	as.factor
سری زمانی	is.ts()	as.ts()

داده چارچوب دار	is.data.frame()	as.data.frame()

تذکر: تابع ts() مربوط به تولید داده های سری زمانی است که در بخش تولید داده ها و شبیه سازی توزیع ها درباره آن صحبت خواهیم کرد.

مثال:

```
> x=seq(-2,2,.5)
> is.vector(x)
[1] TRUE
> is.matrix(x)
[1] FALSE
> as.matrix(x)
      [,1]
 [1,] -2.0
 [2,] -1.5
 [3,] -1.0
 [4,] -0.5
 [5,]  0.0
 [6,]  0.5
 [7,]  1.0
 [8,]  1.5
[9,]  2.0
> as.array(x)
[1] -2.0 -1.5 -1.0 -0.5  0.0  0.5  1.0  1.5  2.0
```

۳. توابع محاسباتی

در این بخش توابع محاسباتی در R را معرفی می کنیم. گاهی برای استفاده از توابع محاسباتی نیاز است که از برخی عملگرهای حسابی و منطقی مثل "کوچکتر" یا " نامساوی" را هم بکار ببریم. برای این منظور و قبل از معرفی توابع ریاضی و آماری در R، عملگرهای حسابی و منطقی را معرفی کرده و پس از آن توابع ریاضی و آماری مورد نیاز را ارائه خواهیم کرد. برای راحتی کار عملگرهای حسابی و منطقی را در یک جدول و توابع ریاضی و آماری را نیز در جدول دیگری معرفی می کنیم.

جدول عملگرهای حسابی و منطقی

عملگر	توضیح درباره عملگر	مثال
+	جمع	> 2.2+7.9 [1] 10.1
-	تفریق	> 3-2 [1] 1
*	ضرب	> 10*9 [1] 90
/	تقسیم	> 15/2 [1] 7.5
^	توان و ریشه	> 8^(1/3) [1] 2
%/%	خارج قسمت تقسیم	> 8%/%2 [1] 4
%%%	باقیمانده تقسیم	> 8%%2 [1] 0
عملگر	توضیح درباره عملگر	مثال
<	کوچکتر	-
>	بزرگتر	-
<=	کوچکتر یا مساوی	-
>=	بزرگتر یا مساوی	-
==	مساوی	-
!=	نامساوی	-
&	"و" منطقی	> T&F [1] FALSE
|	"یا" منطقی	> T|F [1] TRUE
!	نقیض	> x<-!T;x [1] FALSE

جدول توابع ریاضی و آماری

تابع	توضیح درباره تابع	مثال
abs()	تابع قدر مطلق	> abs(-3.76) [1] 3.76
ceiling()	نزدیکترین عدد صحیح بزرگتر به عدد	> ceiling(2.5) [1] 3
cumsum()	مجموع تجمعی اعداد یک بردار	> cumsum(1:10) [1] 1 3 6 10 15 21 28 36 45 55
cumprod()	حاصلضرب تجمعی اعداد یک بردار	> cumprod(1:5) [1] 1 2 6 24 120
sum()	مجموع اعداد	> sum(1:10) [1] 55
prod()	حاصلضرب اعداد	> prod(2,3,4) [1] 24
min()	کوچکترین عدد	> min(2,4,3,9,0,1) [1] 0
max()	بزرگترین عدد	> max(1,0,-1,3,6) [1] 6
floor()	جزء صحیح یک عدد	> floor(-2.45) [1] -3
rank()	رتبه اعداد یک بردار	> rank(c(1,-2,4,7)) [1] 2 1 3 4
sort()	مرتب کردن اعداد یک بردار به صورت صعودی	> sort(c(-1,2,0,3,5)) [1] -1 0 2 3 5
sqrt()	ریشه دوم عدد	> sqrt(245.8) [1] 15.67801
exp()	تابع نمایی	> exp(2.3) [1] 9.974182
log()	تابع لگاریتم طبیعی	> log(9.974182) [1] 2.3
log10()	لگاریتم در مبنای ۱۰	> log10(10) [1] 1
gamma()	تابع گاما	> gamma(4) [1] 6
lgamma()	لگاریتم تابع گاما	> lgamma(4) [1] 1.791759
beta(a,b)	تابع بتا	> beta(2,4) [1] 0.05
trunk()	نزدیکترین عدد صحیح کوچکتر به عدد	> trunc(2.84) [1] 2
factorial()	تابع فاکتوریل	> factorial(10) [1] 3628800
choose(n,r)	تابع ترکیب $\binom{n}{r}$	> choose(10,6) [1] 210
sin(), cos(), tan()	سینوس، کسینوس، تانژانت	> sin(pi/4) [1] 0.7071068

asin(), acos(), atan()	توابع معکوس مثلثاتی	> atan(1) [1] 0.7853982
sinh(), cosh(), tanh()	توابع مثلثاتی هیپربولیک	> cosh(1) [1] 1.543081
asinh(), acosh(), atanh()	معکوس توابع مثلثاتی هیپربولیک	> asinh(1.5) [1] 1.194763
mean()	میانگین اعداد	> mean(1,5,-7,0,3,4) [1] 1
median()	میانه برداری از اعداد	> median(c(-1,3,5,8,0,2)) [1] 2.5
var()	واریانس برداری از اعداد	> var(c(-1,3,2,0,6,4,5)) [1] 6.571429

تذکر: برای گرد کردن اعداد می توان از تابعround() استفاده کرد که روش بکار بردن آن بقرار زیر می باشد. شکل کلی این تابع به صورت زیر می باشد:

```
> round(x,r)
```

که در آن x عدد مورد نظر بوده که قرار است گرد شود و r تعداد رقمهای اعشاری است که می خواهیم به آن گرد شود (مثلا اگر r=2 باشد در این صورت عدد مورد نظر تا دو رقم اعشار گرد می شود).

```
> round(2.844,1)
[1] 2.8
> round(2.844,2)
[1] 2.84
```

۴. تولید اعداد تصادفی از توزیع های آماری

در نرم افزار R این امکان وجود دارد که اعداد تصادفی را از توزیع های مشخص آماری مانند نرمال ، دوجمله ای ، گاما و بسیاری از توزیع های دیگر تولید نمود. این امکانات در بسته نرم افزاری {stats} وجود دارد که دستورات مربوط به تولید داد های تصادفی از هر توزیع و آرگومانهای مربط به آن در قسمت توضیحات مربوط به هر توزیع وجود دارد. برای مشاهده ی این بخش و قسمتهای مربوط به آن می توان از دستور زیر استفاده نمود.

```
> library(stats)
> help.search("distribution",package="stats")
```

در جدول زیر دستور تولید اعداد تصادفی از برخی توزیع های آماری داده شده است.

```
> rbeta(n, shape1, shape2, ncp = 0)            Beta
> rbinom(n, size, prob)                        Binomial
> rgamma(n, shape, rate = 1, scale = 1/rate)   Gamma
> rnbinom(n, size, prob, mu)                   Negative binomial
> rgeom(n, prob)                               Geometric
> rpois(n, lambda)                             Poisson
> rnorm(n, mean , sd )                         Normal
> runif(n, min , max )                         Uniform
```

به عنوان مثال دستور تولید اعداد تصادفی از توزیع نرمال به صورت زیر می باشد.

```
>rnorm(n, mean , sd )
```

دستور فوق شامل سه آرگومان می باشد که به ترتیب تعداد نمونه مورد نظر ، میانگین و انحراف معیار می باشد. به عنوان مثال جهت تولید ۱۵۰ عدد تصادفی از این توزیع با میانگین ۱۰ و واریانس ۲۵ می توان از دستور زیر استفاده نمود.

```
>rnorm(150, 10, 5)
```

فصل ۴

آمار توصیفی و رسم نمودار در R

۱.مقدمه

برای تجزیه و تحلیل داده ها با روشهای مختلف و پیشرفته آماری با نرم افزار R، نیاز است که با چگونگی محاسبه مفاهیم پایه آماری و رسم نمودارها در R آشنا شویم. به همین منظور ما در این فصل با آمار توصیفی و رسم نمودار در R آشنا خواهیم شد. مطالب ارائه شده در این فصل، در عین سادگی از اهمیت بسیاری برخوردار می باشند، زیرا که در فصل های آتی از آن ها به دفعات استفاده می کنیم. ما این فصل را به دو بخش کلی تقسیم بندی کرده ایم که عبارتند از: آمار توصیفی و رسم نمودارهای مختلف و قابلیت های رسم نمودارها در R. در بخش اول آمار توصیفی را تشریح می کنیم.

۲.آمار توصیفی

در فصل های قبل با چگونگی وارد کردن داده ها آشنا شدیم. حال فرض کنید یک مجموعه داده داشته باشیم و بخواهیم شاخص-های مربوط به آمار توصیفی درباره این داده ها را بدست آوریم. ما این کار را ابتدا درباره شاخص‌های مرکزی و سپس در مورد شاخص‌های پراکندگی و در نهایت براساس نمودارهای آمار توصیفی انجام می دهیم.

۱) شاخص های مرکزی

همانطور که می دانیم شاخص های مرکزی شامل میانگین، میانه و چندک ها می باشد. فرض کنید که x بردار (مجموعه) داده مورد نظر باشد. در این صورت همانطور که قبلا در بخش توابع محاسباتی نیز بیان شد دستور mean(x) و median(x) به ترتیب برای یافتن میانگین و میانه داده ها مورد استفاده قرار می گیرد.

```
> x=c(1:10)
> mean(x)
[1] 5.5
> median(x)
[1] 5.5
```

برای محاسبه چندک ها از دستور quantile() استفاده می کنیم. شکل عمومی این دستور به صورت زیر می باشد:

```
> quantile(x,prob=c())
```

که در آن آرگومان prob=c() برداری از چندک های مد نظر کاربر است را به نرم افزار معرفی می کند.

```
> quantile(x)
   0%   25%   50%   75%  100%
 1.00  3.25  5.50  7.75 10.00
> quantile(x,probs=c(.15,.2,.37,.76))
 15%  20%  37%  76%
2.35 2.80 4.33 7.84
```

تذکر: دستور summary() شاخص های تمرکز میانگین، میانه، چارک اول تا سوم را به همراه کوچکترین و بزرگترین مقدار داده ها را به ما می دهد.

```
> summary(x)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   1.00    3.25    5.50    5.50    7.75   10.00
```

۲) شاخص‌های پراکندگی

این شاخص‌ها شامل واریانس، انحراف معیار و دامنه تغییرات داده ها می باشد. همانطور که در بخش توابع محاسباتی اشاره شد از دستور var() برای محاسبه واریانس داده ها می توان استفاده نمود.

```
> var(x)
[1] 9.166667
```

دستورهای sd() و range() به ترتیب انحراف معیار و دامنه تغییرات داده ها را می هد.

```
> sd(x)
[1] 3.02765
> range(x)
[1]  1 10
```

۳.رسم نمودار

نرم افزار R دارای قابلیت های زیادی از نظر گرافیکی و رسم نمودار می باشد و از آنجا که در بسیاری موارد، رسم نمودار برای فهم مسئله و نیز تجزیه و تحلیل داده ها بسیار کارساز می باشد، در این قسمت به تفصیل درباره رسم برخی نمودارها در آمار توصیفی صحبت خواهیم کرد و سپس به بحث پیرامون امکانات این نرم افزار برای رسم نمودارهای مختلف می پردازیم.

- **هیستوگرام**

برای رسم نمودار هیستوگرام، جنس داده های ورودی باید پیوسته باشد. برای رسم نمودار هیستوگرام در R از دستور زیر می‌توان استفاده کرد

```
> hist(x,breaks="Sturges",prob=F)
```

که در آن آرگومان breaks تعداد طبقه ها را نشان می دهد و آرگومان freq=F و prob=T فراوانی غیر فعال شده و شکل طوری تنظیم می شود که مجموع مساحت مستطیل ها در هیستوگرام برابر واحد شود.

در مورد آرگومان breaks، نرم افزار به صورت پیش فرض از فرمول Sturges استفاده می کند. برای اطلاع بیشتر راجع به این روش به کتاب آمار و احتمال مهندسی نوشته دکتر نعمت اللهی مراجعه نمایید. روش های دیگری نیز وجود دارد که طول دسته را معین می کنند. به عنوان مثال می توان در آرگومان breaks به جای "Sturges" از "FD" استفاده کرد که براساس روش Freedman-Diaconis عمل می کند و اساس آن مبتنی بر محدوده بین چارکی (iqr) بوده و فرمول آن به صورت زیر می باشد:

$$2 * iqr * n^{-\frac{1}{3}}$$

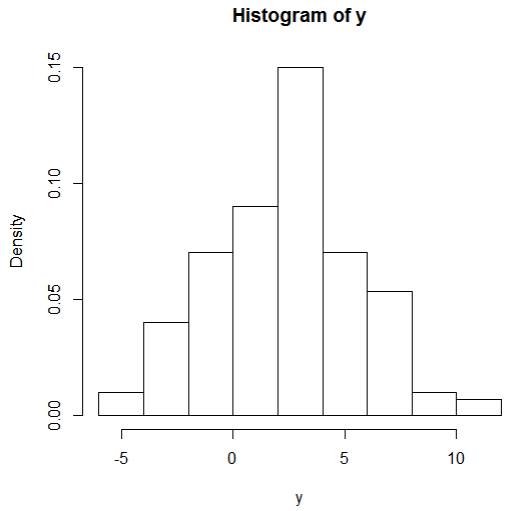
روش دیگری نیز وجود دارد که Scott آن را معرفی کرده است و رابطه آن به صورت زیر می باشد. حال اگر در آرگومان breaks قرار دهیم "Scott" در آن صورت براساس این روش طول دسته ها مشخص می شود.

$$3.5 * s * n^{-\frac{1}{3}}$$

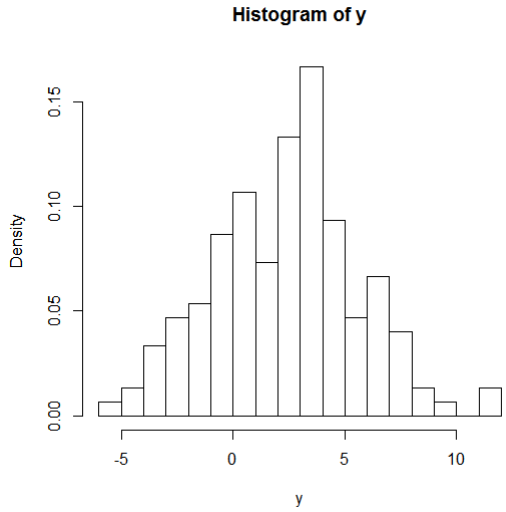
که در آن s انحراف معیار می باشد.

مثال:

```
>y=
> hist(y,breaks="Sturges",prob=T)
> hist(y,breaks="Scott",prob=T)
> hist(y,breaks="FD",prob=T)
```



نمودار هیستوگرام براساس روش Sturges و Scott



نمودار هیستوگرام براساس روش FD

نکته: در آرگومان breaks می توانیم به جای معرفی روشی خاص، تعداد طبقه ها را خودمان برای نرم افزار تعیین کنیم مثلا ۵. (breaks=5)

- **نمودار میله ای (bar plot)**

برای رسم نمودار میله ای، جنس متغیر ورودی بایستی به صورت جدول فراوانی براساس داده های گسسته یا کیفی باشد. به منظور تشکیل جدول فراوانی برای داده های گسسته از دستور table() استفاده می کنیم. فرض کنید که x یک متغیر گسسته به صورت زیر باشد

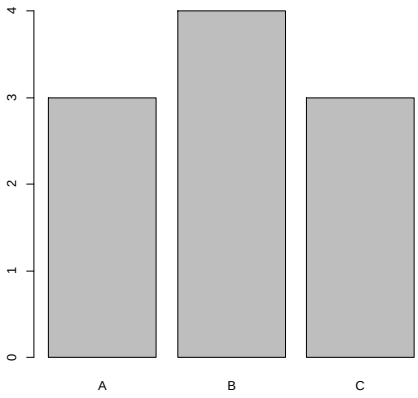
```
x=c(rep("A",3),rep("B",4),rep("C",3))
```

حال با استفاده از دستور table() جدول فراوانی را برای این متغیر تشکیل می دهیم.

```
> table(x)

x

A B C

3 4 3
```

مشاهده می شود که دستور table() داده های مربوط به متغیر x را به صورت یک جدول فراوانی در می آورد. حال می توانیم با استفاده از دستور barplot() نمودار میله ای را برای table(x) و نه خود x رسم کنیم.

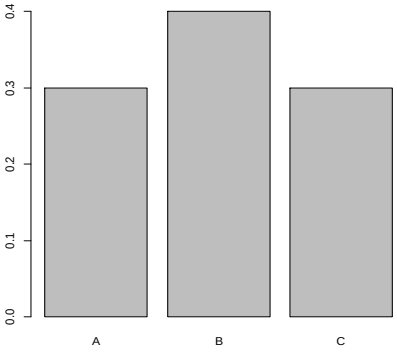
```
> barplot(table(x))
```



نمودار میله ای برای متغیر x بر حسب فراوانی

اگر بخواهیم نمودار میله ای بر حسب فراوانی نسبی به جای فراوانی باشد کافی است نمودار میله ای را به جای table(x) برای table(x)/length(x) رسم کنیم.

```
> barplot(table(x)/length(x))
```



نمودار میله ای برای متغیر x بر حسب فراوانی نسبی

- **نمودار جعبه ای**

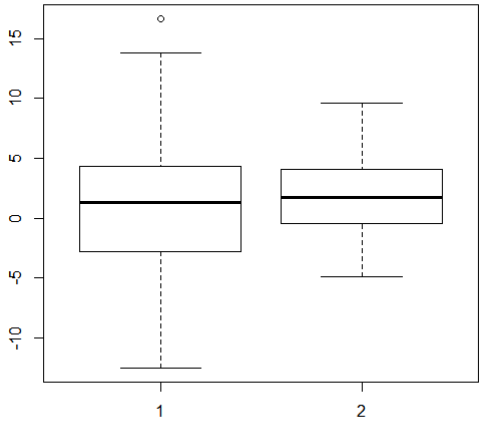
برای رسم نمودار جعبه ای، متغیر ورودی باید از نوع عددی باشد. به منظور رسم نمودار جعبه ای از دستور boxplot() استفاده می کنیم. نمودار جعبه ای برای دو متغیر نیز رسم می شود. شکل کلی این دستور به صورت زیر است:

```
> boxplot(x,y,horizontal=T)
```

اگر آرگومان horizontal=T باشد آنگاه نمودارهای جعبه ای به صورت افقی رسم می شوند.

مثال

```
> y=rnorm(150,2,3)

> x=rnorm(200,1,5)
> boxplot(x,y,horizontal=F)
```



نمودار جعبه ای متغیرهای x و y

۴.رسم نمودار با استفاده از امکانات و قابلیتهای R

نرم افزار R قابلیت رسم نمودارهای متنوعی را دارد. با نوشتن دستورهای demo(graphics) و demo(persp) در R، می توان بخشی از نمودارهایی که این نرم افزار قادر به رسم آن ها می باشد را مشاهده نمود. در کل برای رسم نمودار در R دو نوع تابع وجود دارد که عبارتند از:

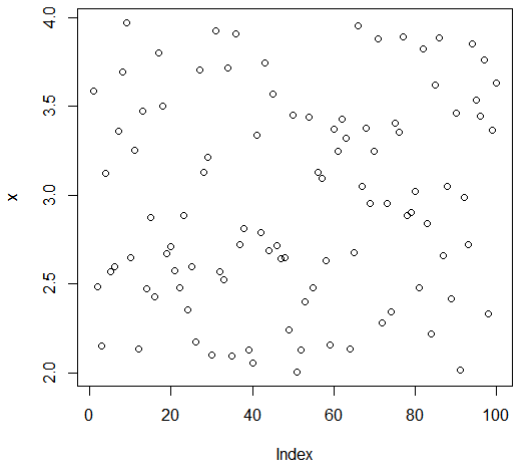
۱. توابع رسم سطح بالا (high-level plotting functions) که این قابلیت را دارند که نمودارهای جدیدی را رسم نمایند.
۲. توابع رسم سطح پایین (low-level plotting functions) که می توانند اجزاء دیگری را به نمودار رسم شده بیافزایند.

برای شروع ما با ساده ترین دستور برای رسم نمودار آغاز کرده و تمامی امکاناتی را که می توان در این دستور پیاده کرده را توضیح می دهیم (این امکانات در واقع همان توابع سطح پایین برای این دستور محسوب می شوند).

برای یک بردار از داده ها، ساده ترین نموداری که می توان رسم کرد نمودار پراکنش آن ها می باشد. این نمودار با دستور plot() رسم می شود. این دستور برای دو بردار از داده ها نیز کاربرد دارد و می تواند نمودار پراکنش دو بردار از داده ها را در مقابل هم رسم کند.

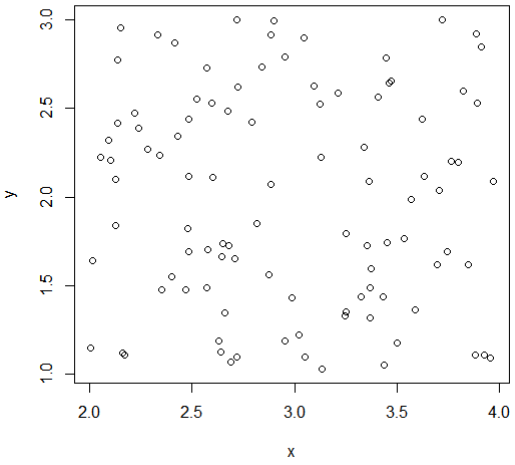
مثال

```
> x=runif(100,2,4)
> plot(x)
```



نمودار پراکنش x

```
> y=runif(100,1,3)

> plot(x,y)
```



نمودار پراکنش x در مقابل y

تنظیمات نمودارها

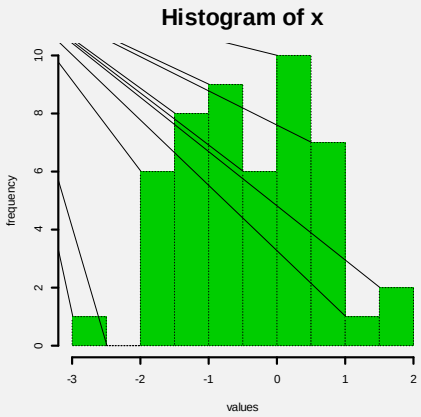
نمودارهایی که در R رسم می شوند دارای آرگومانهایی هستند که به کاربر این اجازه را می دهد که نمودار را به شکل دلخواه در آورد. این آرگومانها عبارتند از:

- آرگومان lwd: این آرگومان مقادیر 0,1,2,… را اختیار می کند. این پارامتر برای تعیین ضخامت خطوط و نقاط نمودار بکار می رود و هر چه مقدار عددی که با آن اختصاص می دهیم بزرگتر باشد، ضخامت خطوط و نقاط بیشتر می شود.
- آرگومان lty: اگر نمودار رسم شده براساس خطوط باشد و نه نقاط (مثل هیستوگرام یا منحنی تابع چگالی)، می توان نوع خطوط را با این آرگومان تغییر داد. این آرگومان هم مقدار عددی و هم کاراکتر اختیار می کند. مثلا اگر بخواهیم در نمودار هیستوگرام به جای خطوط به هم پیوسته از خط چین استفاده کنیم کافی است این آرگومان را به صورت lty=2 و یا lty=”dashed” بنویسیم.
- آرگومان col: با این آرگومان می توان رنگ نمودار را تنظیم کرد. مقدار این آرگومان نیز می تواند عدد یا کاراکتر باشد. به عنوان مثال برای رنگ قرمز می توان نوشت: col=2 یا col=”red”. برای تغییر رنگ عنوان و زیر عنوان به ترتیب از col.main و col.sub، برای تعیین رنگ عناوین محورها از col.lab و بالاخره برای تغییر رنگ محورها از col.axis می توان استفاده نمود.
- آرگومان font: مقدار عددی این آرگومان نوع قلم نوشته های نمودار را تغییر می دهد.
- آرگومان pch: به وسیله این آرگومان می توان شکل نشانه های نمودار (مثلا نقاط) را تغییر داد. این آرگومان نیز مقدار عددی اختیار می کند. مثلا اگر قرار دهیم pch=2 آنگاه شکل نقاط به صورت مثلث در می آید.
- آرگومانهای xlab و ylab: با این آرگومانها می توان اسم محورهای مختصات را تعیین کرد.

- آرگومان cex: به وسیله این آرگومان می توان نشانه ها و متون نمودار را بزرگتر کرد. برای بزرگ کردن اندازه عنوان و زیر عنوان به ترتیب از cex.main و cex.sub می توان استفاده کرد. برای بزرگتر کردن اسامی محورها از cex.lab می توان استفاده کرد. این آرگومان مقدار عددی می گیرد.

مثال

```
>hist(x,xlab="values",ylab="frequency",
main=”Histogramof x”,cex.main=2,lty=3,lwd=3,col=3)
```



چند نمودار در یک صفحه

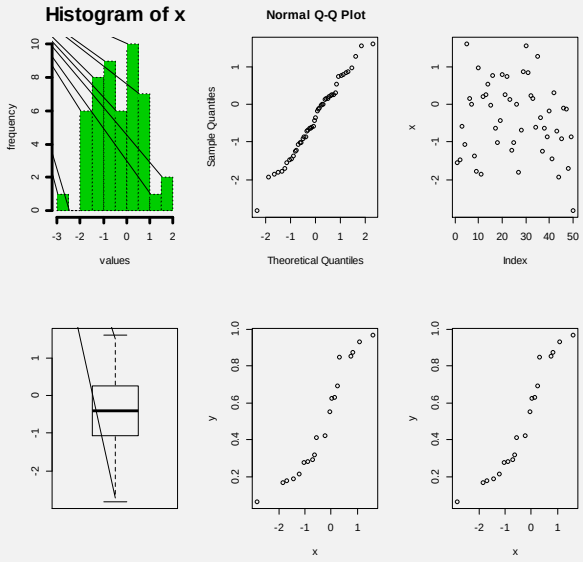
نرم افزار R این قابلیت را دارد که چند نمودار مختلف را به طور همزمان در یک صفحه نمایش دهد. برای این منظور از دستور par() می توان استفاده کرد. این دستور برای تنظیم صفحه گرافیکی مورد استفاده قرار می گیرد و دارای آرگومانهای مختلفی است که دوتا از مهمترین آرگومانهای آن که برای تنظیم چند نمودار در یک صفحه بکار می روند عبارتند از : mfrow و mfcol . شکل کلی این دو آرگومان در دستور par() به صورت زیر می باشد:

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

که در آن پارامتر گرافیکی mfrow و mfcol به ترتیب بیانگر چیدمان ردیفی و ستونی می باشد. همچنین پارامتر r نشاندهنده تعداد سطرها و پارامتر k معرف تعداد ستونهای صفحه ای است که نمودارها در آن رسم می شوند.

به عنوان مثال فرض کنید بخواهیم ۶ نمودار مختلف را در یک صفحه به طور همزمان به گونه ای نمایش دهیم که به صورت دو ردیف و سه ستون ظاهر شده و ترتیب رسم آن ها به صورت ردیفی باشد. در این صورت خواهیم داشت:

```
> par(mfrow=c(2,3))
> hist(x,xlab="values",ylab="frequency",cex.main=2,lty=3,lwd=3,col=3)
> qqnorm(x)
> plot(x)
> boxplot(x)
> qqplot(x,y)
> qqplot(x,y)
```



راجع به نمودارهای بالا مانند Normal Q-Q Plot در آینده صحبت خواهیم کرد.

برای نمایش چندین نمودار در یک صفحه روش دیگری نیز وجود دارد. در این روش از دستور layout() می توان استفاده کرد. شکل کلی این دستور به صورت زیر است:

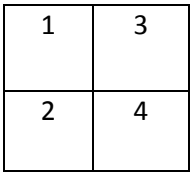
```
> layout(mat,widths=rep(1,ncol(mat)),heights=rep(1,nrow(mat)))
```

که در آن mat ماتریس مختصات نمودارها در صفحه، widths برای پهنا و heights برای بلندی نمودارها در صفحه می باشد. پیش از اینکه این روش را با رسم نمودار توضیح دهیم، طریقه تقسیم بندی صفحه نمایش توسط این روش را تشریح می کنیم. فرض کنید دستور layout() به صورت زیر باشد:

```
> layout(matrix(1:4,2,2))
```

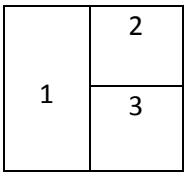
با توجه با ساختار ماتریس، صفحه نمایش به صورت یک مربع ۲*۲ در می آید. برای دیدن چگونگی تقسیم بندی صفحه نمایش از دستور layout.show() استفاده می کنیم.

```
> layout.show(4)
```



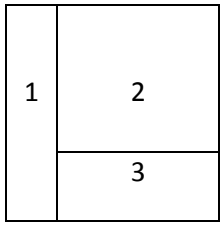
مثال دیگر برای این قسمت به صورت زیر می باشد.

```
> layout(matrix(c(1,1,2,3),2,2))
> layout.show(3)
```



حال می توانیم از آرگومانهای widths و heights نیز استفاده کنیم.

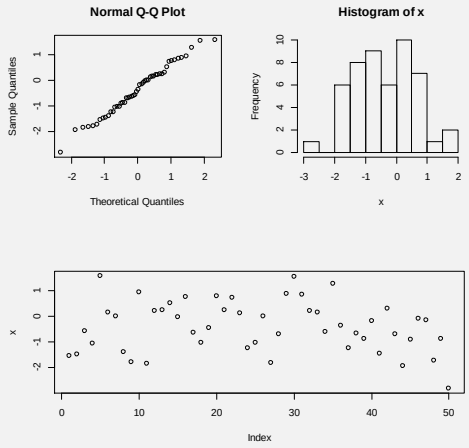
```
> layout(matrix(c(1,1,2,3),2,2),widths=c(1,4),heights=c(4,1))
> layout.show(3)
```



با توجه به مثال فوق در می یابیم که می توان صفحه نمایش را به هر تعداد دلخواه و به هر اندازه دلخواه تقسیم بندی کرد، در واقع دو آرگومان widths و heights برای تنظیم اندازه هر یک از بخشهای صفحه نمایش بکار می روند.

برای فهم این روش جهت رسم چند نمودار در یک صفحه به مثال زیر توجه نمایید. می‌خواهیم سه نمودار را در یک صفحه رسم کنیم بطوریکه اولین و دومین نمودار به صورت مشترک در نیمه بالای صفحه قرار گرفته و سومین نمودار در نیمه پایین صفحه قرار می گیرد.

```
> layout(matrix(c(1,2,3,2),2,2))
> qqnorm(x)
> hist(x)
> plot(x)
```



تذکر: اگر بخواهیم چند نمودار را به طور همزمان در چند پنجره گرافیکی جداگانه رسم کنیم از دستور windows() استفاده می کنیم. نرم افزار R برای رسم هر نمودار، نمودار قبلی را پاک کرده و به جای آن نمودار جدید را رسم می کند. اما با استفاده از دستور windows()، نمودار قبلی باقی مانده و نمودار جدید در یک صفحه گرامیکی جدید رسم می شود. مثلا فرض کنید بخواهیم سه نمودار مختلف را در سه صفحه جداگانه رسم کنیم. در این حالت، به صورت زیر دستورات را می نویسیم:

```
> hist(x)
> windows()
> plot(x)
> windows()
```

```
> qqnorm(x)
```

۵. توابع رسم سطح پایین

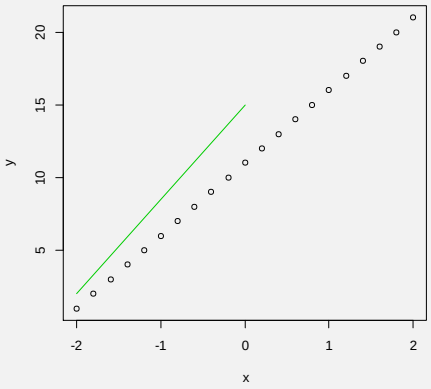
همان طور که قبلا اشاره شد توابع رسم سطح پایین این امکان را ایجاد می کنند که به توان به نمودار رسم شده برخی موارد را اضافه کرد. این گونه توابع می توانند برای افزدون خطوط، نقاط و برچسب‌ها مورد استفاده قرار گیرند. در این قسمت ما به بحث پیرامون این توابع و چگونگی استفاده از آن ها در نمودارها می پردازیم.

توابع رسم سطح پایین برای افزدون خطوط به نمودارها

برای اینکه بتوان خطوطی را به یک نمودار رسم شده اضافه کرد از توابع lines() و abline() می توان استفاده کرد. تابع lines() این قابلیت را دارد که نقاط بردار ورودی را به هم وصل کند و تابع abline() امکان رسم خطوط راست با شیب و عرض از مبدا معین را ایجاد می کند.

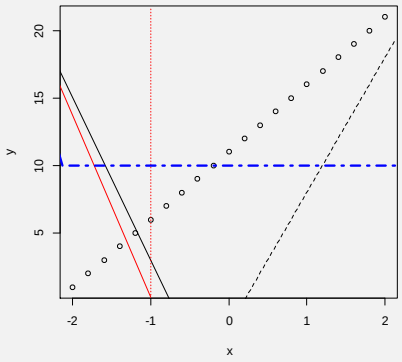
مثال: تابع lines()

```
> x=seq(-2,2,.2)
> y=c(1:21)
> plot(x,y)
> lines(c(-2,0),c(2,15),col=3)
```



مثال: تابع abline()

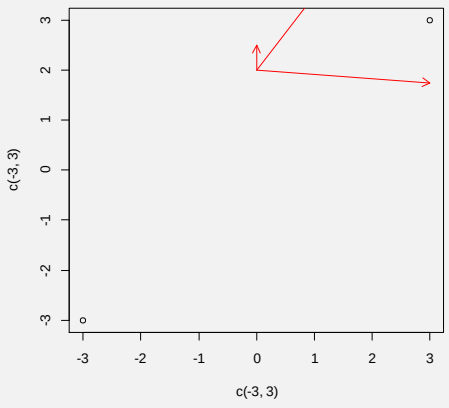
```
> plot(x,y)
> abline(a=-2,b=10,lty=2)
> abline(v=-1,lty=3,col=2)
> abline(h=10,lty=4,col=4,lwd=3)
```



با توجه به مثال فوق براحتی می توان دریافت که در تابع abline() از پارامتر v برای رسم خط عمودی و از پارامتر h برای ترسیم خط افقی استفاده می شود. نکته دیگری که از مثالهای بالا می توان به آن اشاره کرد این است که آرگومانهای تنظیم نمودارها، در توابع سطح پایین نیز قابل استفاده می باشد.

همچنین توابع arrows() و segments() را می توان به ترتیب برای رسم پیکان و قطعه خط بکار برد.

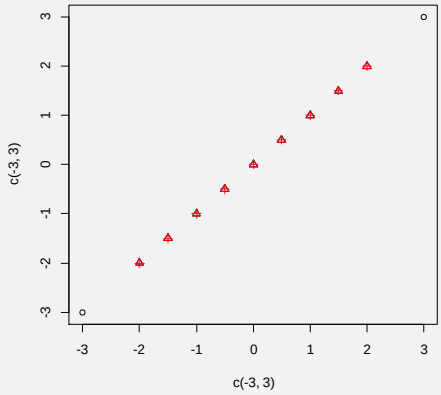
```
> plot(c(-3,3),c(-3,3))
> arrows(c(0,0,0),c(2,2,2),c(0,3,1),c(2.5,1.75,3.5),length=.1,col=2)
```



توابع رسم سطح پایین برای افزدون نقاط و نشانه‌ها به نمودار

برای افزودن نقاط و نشانه ها به نمودار رسم شده از تابع points() می توان استفاده کرد. برای درک بهتر این دستور به مثال زیر توجه نمایید.

```
> plot(c(-3,3),c(-3,3))
>  points(seq(-2,2,.5),seq(-2,2,.5),pch=3,col=2)
```



توابع رسم سطح پایین برای افزدون عنوان و متن به نمودار

برای افزودن متن یا عنوان به نمودار می توان از دستورهای title()، legend()، mtext() و text() استفاده کرد. شکل کلی این دستورات بقرار زیر می باشد:

```
> title(main="NULL",sub="NULL",xlab="NULL",ylab="NULL")
```

در دستور فوق آرگومان‌های main، sub، xlab و ylab به ترتیب برای عنوان اصلی نمودار، عنوان زیر نویس نمودار، عنوان محور افقی و عنوان محور عمودی به کار می روند.

```
> legend(x,y,c())
```

در دستور legend() ، آرگومان x، y و c() به ترتیب مختصات x، مختصات y و بردار مربوط به عنوان می باشد. به عبارت دیگر با این دستور، مختصات بخشی از نمودار را که قرار است در آن قسمت عنوان یا نشانه ای را اضافه کنیم مشخص می نماییم.

```
> mtext(text,side=number,line=0,at=NA,)
```

این تابع، متن مورد نظر را به یکی از حاشیه های نمودار اضافه می‌کند. آرگومانهای این دستور عبارتند از : text، side، line و at. آرگومان text در واقع همان متنی است که قرار است به نمودار اضافه شود. پارامتر side محور مورد نظر را مشخص می کند. عدد ۱ در این پارامتر برای محور پایین، عدد ۲ برای محور سمت چپ، عدد ۳ جهت محور بالا و در نهایت عدد ۴ برای محور سمت راست بکار می رود. پارامتر line فاصله نوشته را (از محور مورد نظر) معین می کند و این مقدار به صورت پیش فرض ۰ می باشد. آرگومان at مختصات نوشته را روی خط فوق نشان می دهد.

```
> text(x,y,"text")
```

در دستور ()text، آرگومانهای x، y و text بترتیب بیانگر مختصات طول، عرض و متنی می باشد که می خواهیم روی نمودار اضافه شود.

فصل ۵

آزمون فرض های آماری

۱. مقدمه

یکی از مهم ترین و پرکاربردترین مباحث آماری آزمون فرضها می باشد که به دو گروه آزمونهای پارامتری و ناپارامتری تقسیم می شوند. آزمونهای پارامتری که مبتنی بر فرض نرمال بودن توزیع جامعه می باشد شامل آزمون های t یک نمونه ای ، t دو نمونه مستقل و t زوجی می باشد.

۲. آزمون t یک نمونه ای

این آزمون مبتنی بر فرض نرمال بودن توزیع جامعه می باشد و در آن فرض صفر $H_0 : \mu = \mu_0$ در برابر فرض مقابل آزمون خواهد شد. آماره ی آزمون بر اساس یک نمونه ی تصادفی n تایی که دارای توزیع نرمال $n(\mu, \sigma^2)$ می باشد به صورت زیر می باشد.

$$t = \frac{\overline{x} - \mu_0}{S_{\overline{x}}}$$

که آماره ی فوق دارای توزیع t استودنت با n-1 درجه آزادی می باشد.

دستور R برای آزمون فرض فوق به صورت زیر می باشد.

```
>t.test(data, mu, alternative)
```

مثال ۱ . داده های زیر میزان انرژی غذایی اخذ شده توسط ۱۰ نفر در طول یک روز می باشد.

۵۲۶۰ ، ۵۴۷۰ ، ۵۶۴۰ ، ۶۱۸۰ ، ۶۳۹۰ ، ۶۵۱۵ ، ۶۸۰۵ ، ۷۵۱۵ ، ۸۲۳۰ ، ۸۷۷۰

مطلوبست آزمون فرض $H_0 : \mu = 7725$ در مقابل

الف) $H_1 : \mu \neq 7725$

ب) $H_1 : \mu > 7725$

ج) $H_1 : \mu < 7725$

برای حل قسمت الف ، با استفاده از دستور R خواهیم داشت

```
>dailyEn=c(8770,8230,7515,6805,6515,6390,6180,5640,5470,5260)

>t.test(dailyEn,mu=7725,alternative="two.sided"
                                        ,conf.level=0.95 )

        One Sample t-test

data:  dailyEn
t = -2.8213, df = 9, p-value = 0.02001
alternative hypothesis: true mean is not equal to 7725
95 percent confidence interval:
 5837.592 7517.408
sample estimates:
mean of x
   6677.5
```

در خروجی فوق ، با توجه به اینکه p-value = 0.02001<0.05 می باشد بنابراین فرض صفر رد می شود.

همچنین دستور R جهت حل قسمت ب و قسمت ج به ترتیب به صورت زیر می باشد.

```
>t.test(dailyEn,mu=7725,alternative="greater")

>t.test(dailyEn,mu=7725,alternative="less")
```

۳. آزمون t برای مقایسه میانگین های دو نمونه مستقل

از این آزمون جهت مقایسه ی میانگین های دو جامعه مستقل که هر دو نرمال باشند استفاده می شود. فرض کنید جامعه ی اول دارای توزیع نرمال $n(\mu_1, \sigma_1^2)$ و جامعه ی دوم نیز دارای توزیع نرمال $n(\mu_2, \sigma_2^2)$ باشد. فرض صفر جهت آزمون مقایسه ی میانگین های دو جامعه فوق به صورت $H_0 : \mu_1 = \mu_2$ می باشد که در برابر فرض مقابل بایستی آزمون شود.

آماره ی مناسب جهت انجام آزمون فوق به صورت زیر می باشد

$$t=\frac{\overline{x}_1-\overline{x}_2}{S_{\overline{x}_1-\overline{x}_2}}$$

که دارای توزیع t می باشد. در رابطه ی فوق $S_{\overline{x}_1-\overline{x}_2}$ انحراف معیار $\overline{X}_1-\overline{X}_2$ که مقدار آن و همچنین مقدار درجه آزادی توزیع t به برابری یا عدم برابری واریانس دو جامعه یعنی دارد.

جهت انجام آزمون فوق با کمک دستور R می توان به دو صورت عمل نمود. در روش اول دستور R به صورت زیر می باشد

```
>t.test(y~x)
```

که در آن y بردار مقادیر عددی مشاهده شده از دو جامعه که به ترتیب قرار دارند می باشدو همچنین x نیز یک بردار عددی از متغیرهای نشانگر می باشد که در آن می توان از عدد 1 به عنوان نشانگر جامعه اول و از عدد 2 به عنوان مقدار نشانگر جامعه دوم استفاده نمود.

در روش دوم دستور R به صورت زیر می باشد

```
>t.test(y1,y2)
```

که در آن y1 و y2 به ترتیب بردار مقادیر مشاهده شده از جامعه ی اول و دوم می باشد.

مثال ۲. جهت مقایسه ی استحکام دو نوع سپر اتومبیل ، شش سپر از هر کدام انتخاب شده و بر روی یک خودروی بخصوص نصب شده است.سپس خودرو با سرعت ۵ مایل در ساعت به یک دیوار بتونی کوبیده می شود که هزینه های تعمیربر حسب دلار به صورت زیر می باشد.

سپر ۱ : ۱۲۷ ، ۱۶۸ ، ۱۴۳ ، ۱۶۵ ، ۱۲۲ ، ۱۳۹

سپر ۲ : ۱۵۴ ، ۱۳۵ ، ۱۳۲ ، ۱۷۱ ، ۱۵۳ ، ۱۴۹

در سطح خطای ۰/۰۵ آزمون کنید آیا میانگین هزینه تعمیر (استحکام) دو نوع سپر یکسان است یا خیر.

جهت حل مثال فوق با استفاده از دستور R و با روش اول خواهیم داشت

```
> y=c(127,168,143,165,122,139,154,135,132,171,153,149)
> x=rep(c(1,2),c(6,6))
> t.test(y~x,conf.level=0.95)

        Welch Two Sample t-test
```

```
data:  y by x
t = -0.5152, df = 9.248, p-value = 0.6185
alternative hypothesis: true difference in means is not equal to
0
95 percent confidence interval:
 -26.86641  16.86641
sample estimates:
mean in group 1 mean in group 2
            144             149
```

در خروجی فوق فرض بر این است که واریانس دو جامعه با هم برابر نیست. اما برای اطمینان از برابری واریانس دو جامعه می توان ابتدا آزمون برابری واریانس دو جامعه را انجام داد. این آزمون در R با دستور زیر انجام می شود.

```
>var.test(y~x, conf.level=0.95)

        F test to compare two variances

data:  y by x
F = 1.798, num df = 5, denom df = 5, p-value = 0.5353
alternative hypothesis: true ratio of variances is not equal to
1
95 percent confidence interval:
  0.2515986 12.8493360
sample estimates:
ratio of variances
           1.79802
```

همانطور که در خروجی آزمون فوق مشاهده می شود با توجه به اینکه p-value = 0.5353>0.05 می باشد بنابراین فرض صفر مبنی بر برابری واریانس دو جامعه پذیرفته می شود. بنابراین آزمون فرض برابری میانگین دو جامعه با فرض برابری واریانس دو جامعه بایستی انجام شود که در R با دستور زیر نجام می شود.

```
> t.test(y~x,var.equal=T,conf.level=0.95)

        Two Sample t-test

data:  y by x
t = -0.5152, df = 10, p-value = 0.6176
alternative hypothesis: true difference in means is not equal to
0
95 percent confidence interval:
 -26.62558  16.62558
```

```
sample estimates:
mean in group 1 mean in group 2
            144             149
```

۴. آزمون t زوجی

آزمون t زوجی برای مقایسه مقادیری از یک آزمودنی که در دو مرحله اندازه گیری شده باشد استفاده می شود بطوریکه مقادیر اندازه گیری شده در این دو مرحله از هم مستقل نمی باشند. فرض صفر و مقابل در این آزمون به صورت زیر می باشد.

$$\begin{cases} H_0 : \mu_d = 0 \\ H_1 : \mu_d \neq 0 \end{cases}$$

که در آن μ_d میانگین تفاضل مقادیر مشاهده شده در دو مرحله می باشد. آماره ی مناسب برای آزمون فوق به صورت زیر می باشد

$$t = \frac{\bar{d}}{S_{\bar{d}}}$$

که دارای توزیع t با $n-1$ درجه آزادی می باشد.

دستور R برای انجام آزمون فوق به صورت زیر است.

```
>t.test(y1,y2,paired=T,conf.level)
```

مثال ۳. جهت بررسی تاثیر یک تمرین ورزشی در کاهش وزن، تحقیقی صورت گرفته است. در این تحقیق گروهی مرکب از ۱۰ نفر به مدت یک ماه در این تمرین ورزشی سرکت نموده اند که نتایج زیر بدست آمده است.

۲۱۱ ، ۱۸۰ ، ۱۷۱ ، ۲۱۴ ، ۱۸۲ ، ۱۹۴ ، ۱۶۰ ، ۱۸۲ ، ۱۷۲ ، ۱۵۵ : وزن قبل از ورزش

۱۹۸ ، ۱۷۳ ، ۱۷۲ ، ۲۰۹ ، ۱۷۹ ، ۱۹۲ ، ۱۶۱ ، ۱۸۲ ، ۱۶۶ ، ۱۵۴ : وزن بعد از ورزش

در سطح خطای ۰/۰۵ آزمون کنید آیا تمرین ورزشی در کاهش وزن موثر بوده است یا خیر.

جهت حل مثال فوق با دستور R خواهیم داشت

```
> pre=c(211,180,171,214,182,194,160,182,172,155)
> post=c(198,173,172,209,179,192,161,182,166,154)

> t.test(pre,post,paired=T,conf.level=0.95)
```

```
        Paired t-test

data:  pre and post
t = 2.5281, df = 9, p-value = 0.03234
alternative hypothesis: true difference in means is not equal to
0
95 percent confidence interval:
 0.3681852 6.6318148
sample estimates:
mean of the differences
                   3.5
```

در خروجی فوق با توجه به اینکه p-value = 0.03234< 0.05 می باشد بنابراین فرض صفر مبنی بر عدم تاثیر تمرین ورزشی در سطح خطای ۰/۰۵ رد می شود به عبارت دیگر تمرین ورزشی موثر بوده است.

فصل ۶

تجزیه و تحلیل واریانس

۱. مقدمه

هر گاه مقایسه ی میانگین های بیش از دو جامعه مستقل مد نظر باشد از تحلیل واریانس استفاده می شود. به عبارتی جهت مقایسه ی میانگین های دو جامعه مستقل نرمال از آزمون t استفاده می شود اما برای مقایسه ی میانگین های $k > 2$ جامعه مستقل که همگی نرمال باشند از تحلیل واریانس استفاده می شود. در این فصل تحلیل واریانس یکطرفه ، دو طرفه و تحلیل کواریانس با استفاده از دستورات R توضیح داده خواهد شد.

۲. تحلیل واریانس یکطرفه

در تحلیل واریانس یکطرفه یک متغیر وابسته و یک متغیر مستقل وجود دارد که متغیر مستقل ، یک متغیر اسمی با بیش از دو سطح می باشد. بنابراین تحلیل واریانس یکطرفه بر اساس یک مدل رگرسیونی نیز قابل بیان است. به طور کلی مدل تحلیل واریانس یکطرفه به صورت زیر می توان نوشت

$$y_{ij} = \mu + \alpha_i + \varepsilon_{ij}, \quad \varepsilon_{ij} \sim n(0, \sigma^2)$$

که در آن $i = 1, 2, ..., k$ و $j = 1, 2, ..., n$ می باشد. بر این اساس آزمون فرض برابری میانگین های k جامعه به صورت زیر می باشد.

$$\begin{cases} H_0 : \alpha_i = 0, \ i = 1, 2, ..., k \\ H_1 : \alpha_i \neq 0, \ \exists i \end{cases}$$

آماره ی آزمون جهت آزمون فرض فوق به صورت زیر می باشد

$$F=\frac{MST}{MSE}$$

که دارای توزیع F با $k-1$ و $k(n-1)$ درجه آزادی می باشد.

دستور R برای تحلیل واریانس به صورت زیر می باشد.

```
>fit=aov(y~A,data=mydata)
```

در اینجا توجه داشته باشید که data بایستی به صورت data.frame باشد.

مثال ۱ . یک محقق درصدد است تا بداند آیا شدت شوک الکتریکی در حل مساله های مشکل موثر است یا خیر. به همین منظور تعداد ۱۸ نفر را به طور تصادفی در سه گروه قرار داده است که به ترتیب تحت شوکهای الکتریکی سطح کم ، متوسط و بالا قرار گرفته اند و سپس مدت زمان حل مساله ها توسط آنان بر حسب دقیقه اندازه گیری شده است. نتایج آزمایش در جدول زیر می باشد .

۱۵ ، ۱۰ ، ۲۵ ، ۱۵ ، ۲۰ ، ۱۸ : سطح کم

۳۰ ، ۱۵ ، ۲۰ ، ۲۵ ، ۲۳ ، ۲۰ : سطح متوسط

۴۰، ۳۵ ، ۵۰ ، ۴۳ ، ۴۵ ، ۴۰ : سطح بالا

جهت حل مثال فوق با R خواهیم داشت

```
> y=c(15,10,25,15,20,18,30,15,20,25,23,20,40,35,50,43,45,40)
> Level=factor(rep(c("1","2","3"),c(6,6,6)))
> SHockdata=data.frame(y,Level)
> fit=aov(y~Level,data=SHockdata)
> summary(fit)
            Df Sum Sq Mean Sq F value   Pr(>F)
Level        2 2100.0  1050.0   40.13 9.53e-07 ***
Residuals   15  392.5    26.2
```

همانگونه که در خروجی فوق مشخص می باشد فرض صفر یعنی برابری میانگین سه سطح در سطح خطای ۰/۰۵ رد می شود.

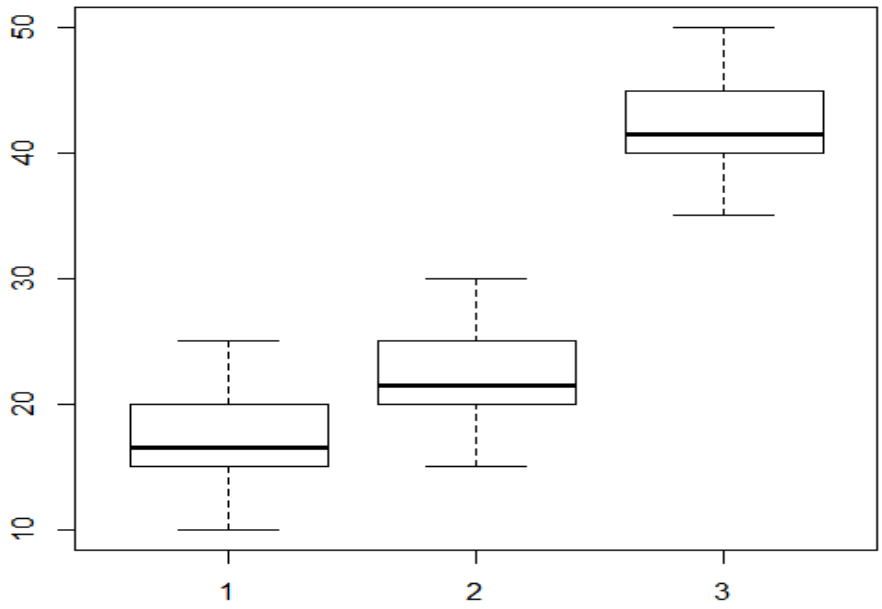
همانطور که می دانیم در تحلیل واریانس یکطرفه هرگاه فرض صفر رد شود برای بررسی دلیل رد این فرض می توان از مقایسات دوتایی یا همان پس آزمون استفاده نمود که از آن جمله می توان آزمون های شفه ، دانکن ، توکی و LSD را نام برد. در این مثال از آزمون توکی برای نمونه استفاده شده است.

```
> tuk=TukeyHSD(fit,"Level")
> tuk
  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = y ~ Level, data = SHockdata)

$Level
      diff       lwr       upr      p adj
2-1    5     -2.671215   12.67121  0.2400638
3-1   25      17.328785  32.67121  0.0000012
3-2   20      12.328785  27.67121  0.0000178
```

با توجه به مقایسات زوجی فوق ، بین سطح کم و متوسط اختلافی وجود ندارد ولی بین سطح بالا با دو سطح دیگر اختلاف معنی دار است.

همچنین می توان نمودار boxplot برای هر کدام از سطوح رسم نمود که دستور و خروجی آن به صورت زیر می باشد.

```
boxplot(y~Level)
```



همانطور که در نمودار فوق نیز مشخص می باشد ، سطح شوک الکتریکی بالا با دو سطح دیگر اختلاف زیادی دارد.

۳. تحلیل واریانس دوطرفه

تحلیل واریانس دوطرفه ، تعمیم تحلیل واریانس یکطرفه می باشد . به عبارت دیگر در تحلیل واریانس دوطرفه یک متغیر وابسته و دو متغیر مستقل که هر دو اسمی می باشند وجود دارد که هدف بررسی اثر هر کدام از متغیرهای مستقل می باشد.

مدل آماری تحلیل واریانس دوطرفه به صورت زیر می باشد

$$y_{ij} = \mu + \alpha_i + \beta_j + \gamma_{ij} + \varepsilon_{ij}\,,\ \ \varepsilon_{ij} \sim n(0, \sigma^2)$$

که در آن α_i, β_j اثر متغیرهای مستقل و γ_{ij} اثر متقابل آنها می باشد که همانند تحلیل واریانس یکطرفه می توان برای بررسی اثر هر کدام ، آزمون فرض مربوط به آن را انجام داد.

مثال ۲. محققی در صدد است تا تاثیر دو نوع روش تدریس A و B را بر حفظ کرن متون آسان (Easy) و مشکل (Hard) را بررسی نماید. از ترکیب سطوح فوق ، ۴ حالت مختلف به وجود می آید. این محقق در هر سطح از آزمایش ۶ نفر جهت تکرار قرار داده و تعداد اشتباهات هر فرد را اندازه گیری نموده است که داده ها در جدول زیر داده شده است.

	A	B
Easy	۱۳	۲۰
	۱۱	۱۸
	۸	۱۴
	۹	۱۴
	۵	۱۲
	۶	۱۶
Hard	۱۵	۱۶
	۱۷	۱۳
	۲۳	۱۵
	۲۱	۲۰
	۲۲	۱۱
	۲۰	۱۲

جهت حل مثال فوق با کمک R خواهیم داشت

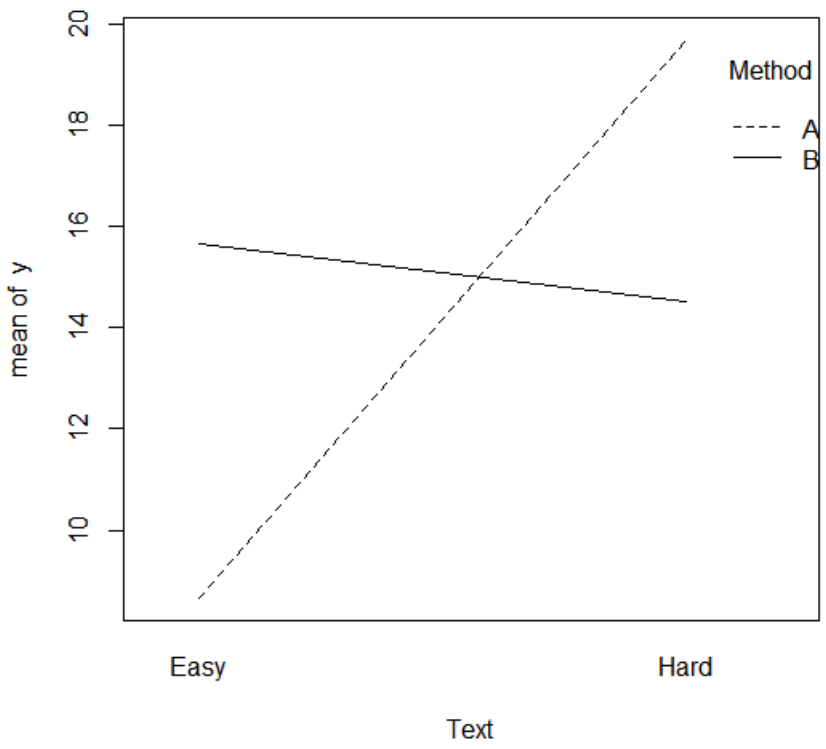
```
> y=c(13,11,8,9,5,6,15,17,23,21,22,20,20,18,14,14,12,16,16,13,
     15,20,11,12)
> Metod=factor(rep(c("A","B"),c(12,12)))
> Text=factor(rep(c("Easy","Hard","Easy","Hard"),c(6,6,6,6)))
> Misdata=data.frame(y,Method,Text)
> Misdata
    y Method Text
1  13  A     Easy
2  11  A     Easy
3   8  A     Easy
4   9  A     Easy
5   5  A     Easy
6   6  A     Easy
7  15  A     Hard
```

```
8  17  A     Hard
9  23  A     Hard
10 21  A     Hard
11 22  A     Hard
12 20  A     Hard
13 20  B     Easy
14 18  B     Easy
15 14  B     Easy
16 14  B     Easy
17 12  B     Easy
18 16  B     Easy
19 16  B     Hard
20 13  B     Hard
21 15  B     Hard
22 20  B     Hard
23 11  B     Hard
24 12  B     Hard
```

پس از ورود داده ها ، ابتدا وجود اثر متقابل بین دو متغیر مستقل (روش تدریس و حفظ کردن متون) را با استفاده از نمودار بررسی می نماییم. این نودار در R با دستور زیر بدست می آید که خروجی آن نیز در زیر داده شده است.

```
> interaction.plot(Text,Method,y)
```

همانطور که در نمودار زیر مشخص می باشد ، با توجه به اینکه دو خط همدیگر را قطع نموده اند بنابراین اثر متقابل وجود دارد.



وجود اثر متقابل در اینجا بدین معنی است که تاثیر روش تدریس بر روی تعداد اشتباهات بستگی به سختی یا آسان بودن متن دارد. مثلا با توجه به نمودار فوق ، تعداداشتباهات هر فرد در روش تدریس B برای متون آسان و سخت تقریبا یکسان است اما تعداد اشتباهات هر فرد در روش تدریس A کاملا بستگی به سخت یا آسان بودن متن دارد.

حال جهت بررسی اثر هر کدام از متغیرها ی مستقل و همچنین اثر متقابل آنها با استفاده از جدول تحلیل واریانس با استفاده از R خواهیم داشت

```
> fit=lm(y~Text+Method+Text:Method)
> summary(aov(fit))
             Df Sum Sq Mean Sq F value   Pr(>F)
Text          1 145.04  145.04  15.308 0.000863 ***
Method        1   5.04    5.04   0.532 0.474182
Text:Method   1 222.04  222.04  23.434 9.92e-05 ***
Residuals    20 189.50    9.47
```

با توجه به خروجی فوق ، سختی یا آسانی متن در تعداد اشتباهات موثر است ولی روش تدریس تاثیری نداشته است. همچنین بین متغیر روش تدریس و نوع متن اثر متقابل وجود دارد.

۴. تحلیل واریانس کوواریانس

تحلیل کوواریانس بسط تحلیل واریانس می باشد و این امکان را به وجود می آورد تا اختلاف بین گروهها را در حضور یک متغیر مستقل پیوسته بررسی نماید.به عبارت دیگر در تحلیل کوواریانس ، دو متغیر مستقل وجود دارد که یکی اسمی و دیگری یک متغیر پیوسته می باشد. دلیل حضور متغیر مستقل پیوسته در اینجا اینست که فرض می شود ممکن است این متغیر مستقل بر روی متغیر وابسته تاثیر داشته باشد. آنالیز کوواریانس این امکان را می دهد تا تغییرات متغیر وابسته که مربوط به متغیر مستقل پیوسته می باشد را حذف نماید. بنابراین ، بین متغیر وابسته و متغیر مستقل پیوسته یک رابطه ی معنا داری وجود دارد و می توان جهت تجزیه و تحلیل از مدل رگرسیونی استفاده نمود.

تحلیل کوواریانس را می توان به عنوان بخشی از تحلیل واریانس یکطرفه ، دوطرفه و یا تحلیل واریانس چند متغیره در نظر گرفت.

یکی از حالتهایی که معمولا جهت تجزیه و تحلیل آن از تحلیل کوواریانس استفاده می شود آزمونهایی هستند که به صورت پیش آزمون/ پس آزمون طراحی می شوند. در این حالت ، مقادیر متغیر وابسته در پیش آزمون به عنوان متغیر مستقل پیوسته در نظر گرفته می شود زیرا به نظر می رسد که بر روی مقادیر متغیر وابسته تاثیر داشته باشد.

جهت انجام تحلیل کوواریانس ، بایستی یک سری از پیش فرضها برقرار باشد باشد که از مهمترین آنها وجود رابطه ی خطی میان متغیر وابسته و متغیر مستقل پیوسته و همچنین همگنی ضریب رگرسیونی می باشد. همگن بودن ضریب رگرسیونی به این مفهوم است که ضریب متغیر مستقل پیوسته برای همه ی سطوح متغیر مستقل اسمی یکسان باشد و این موضوع را می توان با استفاده از بررسی اثر متقابل بین دو متغیر مستقل بررسی نمود. به عبارت دیگر اگر اثر متقابل میان دو متغیر معنی دار باشد ، آنگاه ضریب رگرسیونی همگن نمی باشد و روش تحلیل کواریانس برای تحلیل این داده ها مناسب نمی باشد و در صورتی که اثر متقابل معنی دار نباشد بدین معنی است که ضریب رگرسیونی همگن بوده و مدل تحلیل کواریانس برای تحلیل داده ها بدون اثر متقابل مناسب می باشد.

مثال ۳ . هدف یک تحقیق بررسی آنست که معلوم شود آیا شرکت در یک دوره ی آموزشی تجاری خاص در افزایش درآمد افراد موثر است یا خیر. به همین منظور درآمد افرادی که در این دوره شرکت نموده اند با درآمد افرادی که در این دوره شرکت ننموده اند مقایسه خواهد شد. اما به نظر می رسد درآمد افراد قبل از برگزاری این دوره ها به عنوان یک متغیر پیش آزمون بر روی درآمد افراد بعد از برگزاری دوره موثر باشد. داده ها در جدول زیر داده شده است. در این جدول ،

Incbef= (Income Before) درآمد قبل از شرکت در دوره

Incaft= (Income After) درآمد بعد از شرکت در دوره

Prog= (Program Status) (Yes=1, No=0) وضعیت شرکت در دوره

Incbef	Incaft	Prog	Incbef	Incaft	Prog	Incbef	Incaft	Prog
8	12	0	7	9	0	9	15	0
8	10	0	8	10	0	8	17	1
8	11	0	11	26	1	11	16	0
9	18	1	7	10	0	8	12	0
7	12	0	7	8	0	10	17	1
8	15	1	8	12	0	13	27	1
8	13	0	8	15	0	7	15	0
9	22	1	9	13	0	9	18	0
7	18	1	11	14	0	10	17	0
7	9	0	6	14	1	7	16	1
6	8	0	8	18	1	10	23	1
10	20	1	11	20	0	8	14	1
6	14	1	9	17	1	9	19	1
8	16	1	10	16	0	8	11	0
12	25	0	9	13	0	9	9	0
9	20	1	9	12	0	7	13	0
10	18	1	9	21	0	10	9	0
9	21	1	8	13	0	7	15	1
11	20	1	8	12	1	11	18	1
8	17	1	12	24	1	9	19	1

بعد از ورود داده ها به R ، جهت انجام تحلیل کوواریانس خواهیم داشت.

```
> fit=lm(Incaft~Incbef+Prog+Incbef:Prog)
> summary(fit)

Call:
lm(formula = Incaft ~ Incbef + Prog + Incbef:Prog)

Residuals:
    Min      1Q  Median      3Q     Max
-6.8964 -1.5758 -0.1468  1.2557  6.9692

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -2.7601     2.8576  -0.966    0.338
Incbef        1.8657     0.3305   5.646 5.69e-07 ***
Prog          6.0400     3.9456   1.531    0.131
Incbef:Prog  -0.1730     0.4460  -0.388    0.700
```

با توجه به خروجی فوق ، اثر متقابل معنی دار نیست بنابراین تحلیل کوواریانس برای این داده ها مناسب می باشد. حال اثر متقابل را از مدل حذف نموده و یک بار دیگر مدل را برازش می دهیم.

```
>fit=lm(Incaft~Incbef+Prog)
> summary(fit)

Call:
lm(formula = Incaft ~ Incbef + Prog)

Residuals:
    Min      1Q  Median      3Q     Max
-6.7569 -1.5214 -0.0968  1.2880  7.0137

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.9499     1.9355  -1.007    0.318
Incbef        1.7707     0.2203   8.039 5.93e-11 ***
Prog          4.5331     0.6844   6.623 1.35e-08 ***
```

با توجه به خروجی فوق ، اثر شرکت در برنامه آموزشی موثر است بنابراین میانگین درآمد افرادی که در دوره شرکت نموده اند با میانگین درآمد افرادی که در این دوره شرکتت ننموده اند تفاوت معنی داری دارد. همچنین معنی دار بودن ضریب درآمد پیشین ، و مثبت بودن آن ، بدین معنی است که درآمد افرادی که در دوره شرکت نموده اند افزایش یافته است.

فصل ۷

مدلهای خطی و تحلیل واریانس چند متغیره

۱. مقدمه

در آزمایشهایی که شامل یک متغیر وابسته و چند متغیر مستقل می باشد ، معمولا از مدلهای خطی برای بررسی تاثیر متغیرهای مستقل بر متغیر وابسته استفاده می شود. حال اگر متغیر وابسته خود شامل چند متغیر که بین آنها همبستگی وجود دارد باشد ، حالت مذکور را چند متغیره گویند گویند و تحلیل مربوط به آن را تحلیل چند متغیره گویند. در ای بخش انواع مختلفی از جمله آنالیز اندازه های مکرر، تحلیل واریانس چند متغیره یکطرفه و دو طرفه معرفی و در نحوه تجزیه و تحلیل آنها با استفاده از R ارائه خواهد شد.

۲. آنالیز اندازه های مکرر

آنالیز اندازه های مکرر تعمیم آزمون t مقادیر زوج شده می باشد. در آزمون مقادیر زوج شده ، همانطور که گفته شد از آزمودنی های یکسان در دو مرحله که این مراحل با توجه به موقعیت زمانی یا مکانی در نظر گرفته می شود مقادیر اندازه گیری می شوند و سپس این مقادیر به هم مقایسه خواهند شد. حال اگر مراحل اندازه گیری ، بیش از دو مرحله باشد ، آنگاه حالت مورد نظر را اندازه های مکرر گویند. تفاوت آن با تحلیل واریانس اینست که در تحلیل واریانس گروهها از هم مستقل ی باشند ولی در آنالیز اندازه های مکرر به دلیل یکسان بودن آزمودنی ها در هر مرحله ، از هم مستقل نمی باشند.

مثال۱. محققی در نظر دارد تاثیر دمای هوا را بر حل مساله های ریاضی بررسی نماید. بدین منظور تعداد ۵ نفر را انتخاب نموده و از آنها خواسته شده تا به مجموعه ای از سوالات ریاضی پاسخ دهند. دمای اتاقی که برای حل سوالات در نظر گرفته شده است ۳۵ درجه سانتیگراد می باشد و سپس هر ۵ دقیقه ، به میزان ۵ درجه از دمای هوا کاسته شده که به ترتیب دمای هوای اتاق به۳۰ ، ۲۵ و ۲۰ درجه رسیده است. سپس در هر کدام از این دماها تعداد اشتباهات هر فرد در پاسخ به سوالات اندازه گیری شده است. داده ها در جدول زیر آمده است. بررسی کنید آیا دمای هوا بر روی تعداد اشتباهات موثر بوده است یا خیر.

	دمای هوا			
آزمودنی	35	30	25	20
S1	۷	۵	۲	۳
S2	۸	۷	۵	۴
S3	۶	۵	۳	۳
S4	۸	۸	۴	۲
S5	۵	۴	۳	۲

جهت حل مثال فوق با استفاده از R به صورت زیر عمل می کنیم.

```
> y=c(7,8,6,8,5,5,7,5,8,4,2,5,3,4,3,3,4,3,2,2)
> Temp=factor(rep(c("35","30","25","20"),c(5,5,5,5)))
> Subject=factor(rep(c("s1","s2","s3","s4","s5"),times=4))

> Mathdata=data.frame(y,Temp,Subject)

> fit=aov(y~Temp+Error(Subject/Temp),data=Mathdata)

> summary(fit)

Error: Subject
          Df Sum Sq Mean Sq F value Pr(>F)
Residuals  4   16.7   4.175

Error: Subject:Temp
          Df Sum Sq Mean Sq F value   Pr(>F)
Temp       3   54.6  18.200   24.54 2.09e-05 ***
Residuals 12    8.9   0.742
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

در این مثال هدف بررسی اثر متغیر دمای اتاق بر روی تعداد اشتباهات می باشد. همانطور که در خروجی فوق مشخص می باشد ، با توجه به اینکه p-value=2.09e-05>0.05 می باشد بنابراین دمای اتاق بر روی تعداد اشتباهات موثر می باشد.

در مثال فوق تاثیر دمای اتاق (Temp) با توجه به هر فرد (Subject) مورد ارزیابی قرار گرفت (تاثیر دما بر فرد) بنابراین خطا به صورت Subject/Temp که خوانده می شود دما بر فرد، در نظر گرفته شده است. اکنون با توجه به رد فرض صفر و پذیرش فرض تاثیر دمای اتاق ، می توان از مقایسات زوجی (پس آزمون) استفاده نمود. جهت انجام مقایسات زوجی از دستور زیر استفاده می شود که خروجی داده شده ، مقادیر p-value در مقایسات دوبدو می باشد.

```
> with(Mathdata,pairwise.t.test(y, Temp,p.adjust.method="holm",
                                                    paired=T))

        Pairwise comparisons using paired t tests

data:  y and Temp

   20    25    30
25 0.305 -     -
30 0.054 0.037 -
35 0.011 0.013 0.068

P value adjustment method: holm
```

۳. تحلیل واریانس چند متغیره ی یکطرفه

تحلیل واریانس چند متغیره تعمیم تحلیل واریانس یک متغیره می باشد. به عبارت دیگر در تحلیل واریانس یک متغیره ، یک متغیر وابسته و یک یا چند متغیر مستقل (با توجه به یکطرفه یا دو طرفه بودن تحلیل واریانس) وجود دارد که میانگین متغیر وابسته با توجه به سطوح متغیر مستقل با هم مقایسه می شود. اما در تحلیل واریانس چند متغیره ، همزمان چند متغیر وابسته وجود دارد که میان این متغیرهای وابسته نیز همبستگی وجود دارد و همچنین یک یا چند متغیر مستقل (با توجه به یکطرفه یا دو طرفه بودن تحلیل واریانس چند متغیره)که میانگین متغیرهای وابسته با توجه به سطوح متغیرهای مستقل به هم مقایسه می شوند.

دستور کلی R جهت انجام تحلیل واریانس چند متغیره به صورت زیر می باشد.

```
> fit=manova(A~Factor, data=data)
> summary(fit,test=c("Pillai","Wilks","Hotelling-Lawley","Roy"))
```

مثال ۲. محققی در نظر دارد مدت زمان دویدن چند دونده را در مسافتهای ۵۰ ، ۱۰۰ ، ۲۰۰ و ۳۰۰ متر را برای دوندگان زن و مرد با هم مقایسه نماید. به همین منظور تعداد ۵ دونده را از هر جنسیت انتخاب نموده و مدت زمان دویدن آنها در مسافتهای فوق را اندازه گیری نموده است که اطلاعات آن در جدول زیر آمده است.

		Running Events		
Men	**50m**	**100m**	**200m**	**300m**
1	9	20	42	67
2	9	19	45	66
3	8	17	38	62
4	9	18	46	64
5	8	22	39	59
Women				
1	15	30	41	77
2	14	29	44	76
3	13	27	39	72
4	14	28	56	74
5	13	32	49	69

جهت مقایسه میانگین مدت زمان دوندگان زن و مرد با استفاده از روش Manova ، به صورت زیر عمل می کنیم.

```
>fifty=c(9,9,8,9,8,15,14,13,14,13)
>oneH=c(20,19,17,18,22,30,29,27,28,32)
>twoH=c(42,45,38,46,39,41,44,39,56,49)
>threeH=c(67,66,62,64,59,77,76,72,74,69)
>gender=rep(c("Men","Women"),c(5,5))
>running=data.frame(fifty,oneH,twoH,threeH,gender)
> fit=manova(cbind(fifty,oneH,twoH,threeH)~gender,data=running)
>Summary(fit,test="Pillai")

          Df  Pillai  approx F   num Df   den Df    Pr(>F)
gender     1  0.97092   41.742      4      5      0.000494 ***
Residuals  8
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

همانطور که در خروجی فوق مشخص می باشد مقدار P-value=0.000494<0.05 می باشد بنابراین می توان نتیجه گرفت اختلاف بین میانگین های زمان دویدن ، برای دوندگان زن و مرد در مسافتهای مختلف معنی دار است.

۴. تحلیل واریانس چند متغیره ی دوطرفه

روش کار در تحلیل واریانس چند متغیره ی دوطرفه همانند یکطرفه می باشد با این تفاوت که در این حالت به جای یک متغیر مستقل دو متغیر مستقل وجود دارد که میانگین متغیرهای وابسته نیز با توجه به سطوح دو متغیر مستقل مقایسه می شوند.

مثال ۳. داده های مثال ۲ را در نظر بگیرید که در آن علاوه بر متغیر مستقل جنسیت ، متغیر مستقل نژاد دوندگان نیز در مدل وجود داشته باشد و هدف مقایسه میانگین زمان دویدن در مسافتهای مختلف بر اساس جنسیت و نژاد باشد. داده ها در جدول زیر داده شده است.

	Running Events			
Men	**50m**	**100m**	**200m**	**300m**
White				
1	9	20	42	67
2	9	19	45	66
3	8	17	38	62
4	9	18	46	64
5	8	22	39	59
Nonwhite				
1	6	17	39	64
2	6	16	42	63
3	5	14	35	59
4	6	15	43	61
5	5	19	36	56
Women				
White				
1	15	30	41	77
2	14	29	44	76
3	13	27	39	72
4	14	28	56	74
5	13	32	49	69
Nonwhite				
1	20	35	46	82
2	19	34	49	81
3	18	32	44	77
4	19	32	61	79
5	18	37	54	74

پس از ورود داده ها ، برای اجرای دستور manova ، خواهیم داشت.

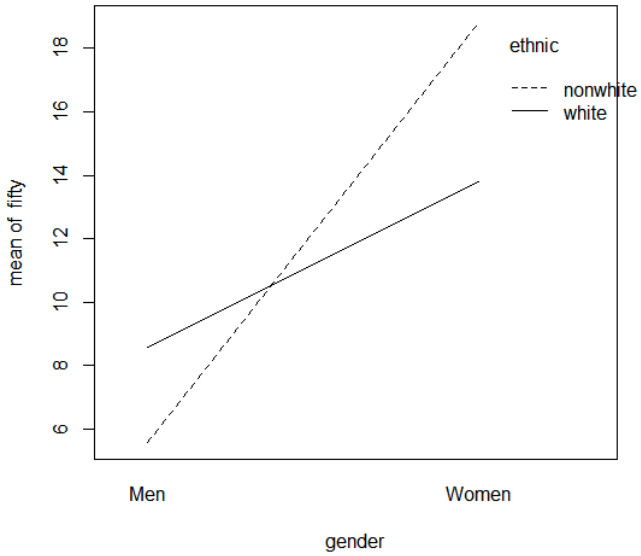
```
>fifty=c(9,9,8,9,8,6,6,5,6,5,15,14,13,14,13,20,19,18,19,18)

>oneH=c(20,19,17,18,22,17,16,14,15,19,30,29,27,28,32,35,34,32,32
     ,37)
>twoH=c(42,45,38,46,39,39,42,35,43,36,41,44,39,56,49,46,49,44,61
     ,54)
>threeH=c(67,66,62,64,59,64,63,59,61,56,77,76,72,74,69,82,81,77,
     79,74)
> gender=rep(c("Men","Women"),c(10,10))

>ethnic=rep(c("white","nonwhite","white","nonwhite"),c(5,5,5,5))

> running=data.frame(fifty,oneH,twoH,threeH,gender,ethnic)

 >fit=manova(cbind(fifty,oneH,twoH,threeH)~gender*ethnic,
     data=running)
> summary(fit,test="Pillai")

              Df  Pillai  approx F  num Df den Df    Pr(>F)
gender         1 0.99179   392.43      4     13  2.072e-13 ***
ethnic         1 0.65804     6.25      4     13  0.004934 **
gender:ethnic  1 0.96825    99.12      4     13  1.330e-09 ***
Residuals     16
---
```

با توجه به خروجی فوق ، اثر gender (جنسیت) ، ethnic (نژاد) و اثر متقابل آنها معنی دار است. به عبارت دیگر متوسط زمان دویدن در مسافتهای داده شده ، بر اساس جنسیت و همچنین با توجه به نژاد متفاوت است.

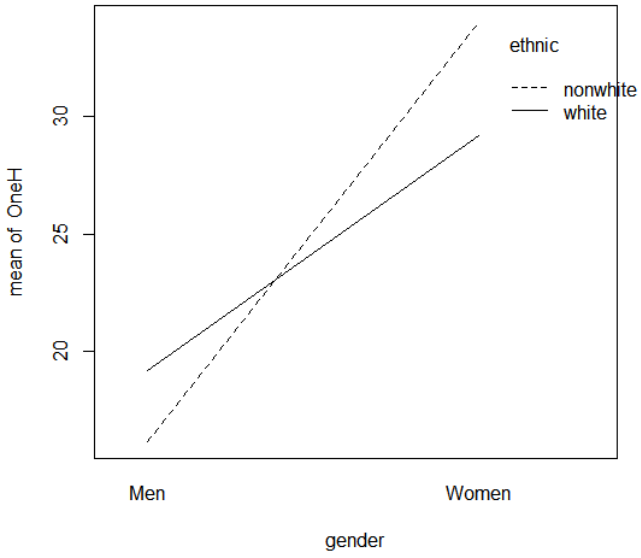
جهت تفسیر و بررسی اثر متقابل ، ابتدا نمودار اثرهای متقابل را رسم می کنیم.

با استفاده از دستور R خواهیم داشت.

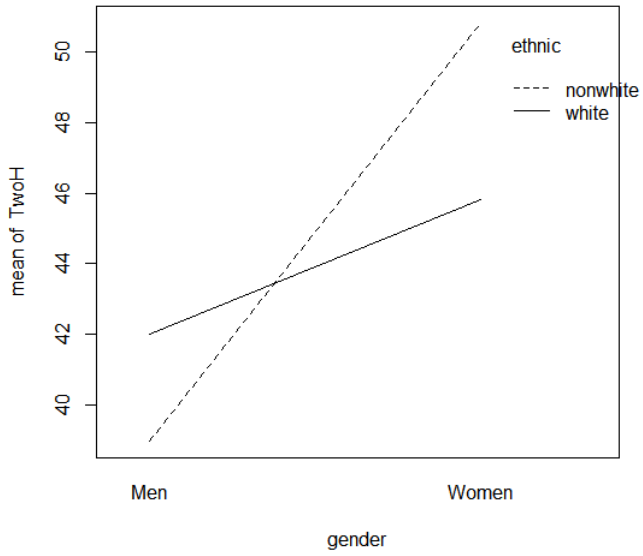
```
> interaction.plot(gender,ethnic,fifty)
>interaction.plot(gender,ethnic,OneH)
> interaction.plot(gender,ethnic,TwoH)
> interaction.plot(gender,ethnic,ThreeH)
```



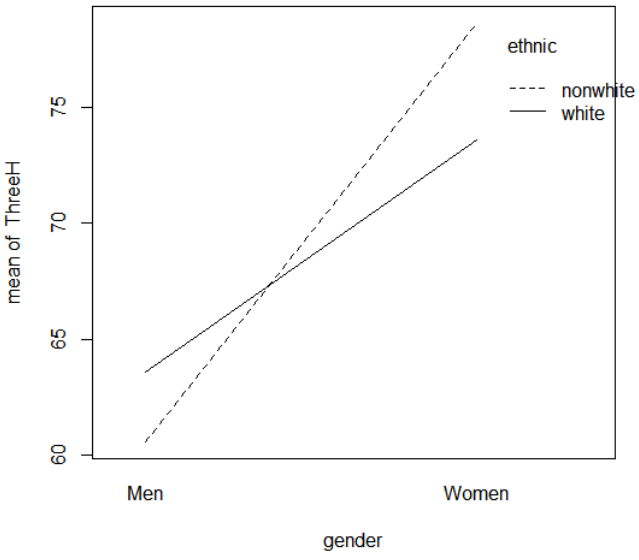
نمودار ۲-۵



نمودار۳-۵



نمودار ۴-۵



نمودار ۵-۵

با توجه به متقاطع بودن خطوط در همه ی نمودارهای فوق ، نتیجه می گیزیم که بین متغیر های مستقل جنسیت و نژاد اثر متقابل وجود دارد. به عنوان مثال تفسیر اثر متقابل بر اساس نمودار ۲-۵ به اینصورت می باشد که متوسط زمان دویدن در ۵۰ متر ، برای متغیر جنسیت ، به سطوح متغیر نژاد بستگی دارد. به عبارت دیگر در دو ۵۰ متر ، سرعت دویدن مردان رنگین پوست ، از مردان سفید پوست بیشتر است ، ولی این نسبت در زنان برعکس می باشد.

فصل ۸

ناپارامتری

۱.مقدمه

آزمونهای ناپارامتری در مواردی که توزیع های نمونه گیری کاملا مشخص نیستند و یا تحت تاثیر مقادیر نمونه میباشند می توانند به عنوان یک جایگزین مناسب در انجام آزمون فرضها مورد استفاده قرار گیرند. آزمونهای ناپارامتری معمولا به پیش فرضهای کمتر و گاهی به هیچ پیش فرضی نیاز ندارند و معلوم یا مجهول بودن پارامترهای جامعه بر روی آنها تاثیری ندارند لذا از این جهت که به پارامترهای جامعه وابسته نیستند ، آن را ناپارامتری گویند.

آزمونهای ناپارامتری می توانند به عنوان بدیلی برای آزمونهای t ی یک نمونه ای ، t ی زوج شده ، آزمون t با گروههای مستقل ، آنالیز واریانس و آزمون بلوکهای تصادفی مورد استفاده قرار گیرند.

۲. آزمون رتبه علامت دار ویلکاکسون

آزمون رتبه علامت دار ویلکاکسون می تواند به عنوان یک جایگزین برای آزمونهای t ی یک نمونه ای و t ی زوج شده بکار رود.در حالتی که به عنوان بدیل برای t ی یک نمونه ای مورد استفاده قرار گیرد در واقع آزمون فرض $\mu = \mu_0$ در برابر فرض مقابل مربوطه مد نظر می باشد .

در این حالت ابتدا تفاضل بین مشاهدات و مقدار μ_0 را بدست آورده و بدون در نظر گرفتن علامت آنها و صرفا با توجه به مقدار قدر مطلق ، کوچکترین مقدار را رتبه ی ۱ ، دومین مقدار را رتبه ی ۲ و به بزرگترین مقدار از لحاظ قدر مطلق رتبه ی n را اختصاص می دهیم.

البته در این رتبه گذاری ، رتبه های صفر کنار گذاشته می شود و اگر دو یا چند تفاضل مقداری مساوی داشته باشند به هر یک از آنها میانگین رتبه هایی که کسب نموده اند اختصاص داده می شود.

در این حالت T^+ عبارتست ازمجموع رتبه هایی که به تفاضل های مثبت اختصاص یافته و T^- مجموع رتبه هایی که به تفاضل های منفی اختصاص یافته و همچنین $T = \min(T^+, T^-)$ می باشد.

ناحیه ی بحرانی هر یک از آزمونها در سطح معنی دار بودن α به صورت زیر می باشد

فرض مقابل	ناحیه ی رد فرض صفر
$\mu \neq \mu_0$	$T \leq T_\alpha$
$\mu > \mu_0$	$T^- \leq T_{2\alpha}$
$\mu < \mu_0$	$T^+ \leq T_{2\alpha}$

البته در نرم افزار R مقدار P-value محاسبه می شود و برای تصمیم گیری در مورد پذیرش و یا عدم پذیرش فرض صفر نیاز به مراجعه به جداول T نمی باشد.

در حالتی که از این آزمون به عنوان یک بدیل برای t ی زوج شده استفاده شود ، همان مراحل آزمون t ی یک نمونه ای دنبال می شود با این تفاوت که به جای تفاضل نمونه ها و مقدار μ_0 ، از تفاضل زوجها جهت رتبه گذاری استفاده می شود.

دستور کلی R برای این آزمون فرض به صورت زیر می باشد.

```
> Wilcox.test(x, y=NULL, alternative=c("two.sided",
"less", "greater"), mu=a, paired=F, conf.int=F,
conf.level=0.95)
```

با توجه به نوع آزمون فرض ، هر کدام از قسمت های دستور فوق می تواند از دستور حذف و یا تغییر کند که مثالهای زیر این نکات راتشریح خواهند کرد.

مثال۱ . اندازه های حاصل از ۱۵ بار اندازه گیری درجه اکتان نوع خاصی بنزین شامل ۹۷/۵ ، ۹۵/۲، ۹۷/۳، ۹۶، ۹۶/۸، ۱۰۰/۳، ۹۷/۴، ۹۵/۳، ۹۳/۲، ۹۹/۱، ۹۶/۱، ۹۷/۶، ۹۸/۲، ۹۸/۵، ۹۴/۹. با استفاده از رتبه ی آزمون علامت دار ویلکاکسون در سطح معنی داری ۵٪ آزمون فرض $H_0: \mu = 98.5$ را در مقابل $H_1: \mu < 98.5$ انجام دهید.

حل:

```
> x=c(97.5,95.2,97.3,96, 96.8, 100.3, 97.4, 95.3, 93.2, 99.1,
     96.1,97.6,98.2,98.5, 94.9)

>wilcox.test(x,alternative="less",mu=98.5,,conf.int=T,conf.level
     =.95)
```

```
  Wilcoxon signed rank test with continuity correction

data:  x
V = 10, p-value = 0.004187
alternative hypothesis: true location is less than 98.5
95 percent confidence interval:
 -Inf 97.6
sample estimates:
(pseudo)median
         96.75
```

با توجه به مقدار p-value در خروجی فوق فرض صفر رد می شود.

مثال ۲ . در زیر وزن ۱۰ نفر بر حسب پوند که به مدت چهار هفته تحت رژیم لاغری خاصی قرار داشته اند، به صورت زیر داده شده است .

قبل	۲۰۸/۱	۱۴۷/۷	۲۱۵/۴	۱۷۷	۲۰۶/۳	۱۹۷/۵	۱۶۱/۶	۲۳۲/۱	۱۸۳/۵	۱۴۷
بعد	۱۹۵/۴	۱۴۹	۲۰۳/۲	۱۸۰/۶	۲۰۱/۴	۱۹۳/۵	۱۶۳/۸	۲۱۹	۱۷۶/۲	۱۳۷/۹

آزمون کنید آیا در سطح معنی داری ۰/۱ رژیم لاغری موثر بوده است یا نه؟

حل: آزمون فرض مربوط به مساله فوق به صورت $H_0: \mu_d = 0$ در مقابل $H_1: \mu_d < 0$ میباشد که در آن d عبارست از وزنهای بعد(x) منهای وزنهای قبل (y)از رژیم لاغری. پس از ورود داده ها همانند مثال ۱ دستور زیر را اجرا می نماییم.

```
> Wilcox.test(x,y, paired=T,alternative="less", conf.level=0.95)
Wilcoxon signed rank test

data:  x and y
V = 49, p-value = 0.9902
alternative hypothesis: true location shift is less than 0
```

با توجه به مقدار p-value فرض صفر پذیرفته می شود.

۳. آزمون مجموع رتبه علامت دار ویلکاکسون (من-ویتنی)

این آزمون ، که به آن آزمون **U** نیز گفته می شود یک آزمون جایگزین برای آزمون t با گروههای مستقل می باشد ،بدون اینکه فرض نرمال بودن دو جامعه مورد نیاز باشد. جهت انجام آزمون ابتدا دو نمونه را با هم ادغام و سپس به صورت غیر نزولی مرتب می نماییم. پس از آن به داده ای مرتب شده رتبه ی ۱ تا n اختصاص داده می شود و اگر مقادیر مساوی وجود داشته باشند به هر یک از آنها ، میانگین رتبه های اختصاص یافته تعلق خواهد گرفت.

اساس تحلیل بر این ایده استوار است که اگر اختلاف قابل توجهی بین میانگینهای دو جامعه وجود نداشته باشد، نبایستی بین رتبه های اختصاص یافته به مشاهدات دو گروه اختلاف قابل توجهی وجود داشته باشد.

برای این آزمون ، اگر مجموع مقادیر رتبه های نمونه ی اول و دوم به ترتیب W_1 و W_2 و حجم نمونه ی اول و دوم برابر n_1 و n_2 باشند در این صورت ازآماره های

$$U_1 = W_1 - \frac{n_1(n_1+1)}{2}$$
$$U_2 = W_2 - \frac{n_2(n_2+1)}{2}$$

و همچنین $U = \min(U_1, U_2)$ جهت رد یا قبول فرض های آزمون به صورتی که در جدول زیر آمده است مورد استفاده میشود.

ناحیه ی رد فرض صفر	فرض مقابل
$U \le U_\alpha$	$\mu_1 \neq \mu_2$
$U_2 \le U_{2\alpha}$	$\mu_1 > \mu_2$
$U_1 \le U_{2\alpha}$	$\mu_1 < \mu_2$

بدیهی است نرم افزار R مقدار P-value را که از این آماره ها بدست می آید ارائه می نماید و مبنای تحلیل نهایی نیز همین مقدار خواهد بود.

مثال ۳ . یک امتحان سنجش میزان آگاهی از تاریخ ملی ،بر روی نمونه های تصادفی از دانشجویان سال اول و دوم در دو دانشگاه بزرگ به عمل آمده است که نمرات آنها به صورت زیر ثبت شده است.

دانشگاه A : ۷۷، ۷۲، ۵۸، ۹۲، ۸۷، ۹۳، ۹۷، ۹۱، ۷۰، ۹۸، ۷۶، ۹۰، ۶۲، ۹۰

دانشگاه B : ۸۹، ۷۴، ۴۵، ۵۶، ۷۱، ۷۴، ۹۴، ۸۸، ۶۶، ۶۲، ۸۸، ۶۳، ۸۸، ۳۷

با استفاده از آزمون **U** و سطح معنی داری ۰/۰۵ این فرض را آزمون کنید که آیا تفاوتی در سطح آگاهی دانشجویان دو دانشگاه وجود دارد یا خیر.

حل : در این مساله هدف انجام آزمون فرض $H_0: \mu_1 = \mu_2$ در مقابل $H_1: \mu_1 \neq \mu_2$ میباشد که برای حل آن پس از ورود داده ها از دستور زیر استفاده میشود.

```
> wilcox.test (x, y, alternative="two.sided", conf.level=0.95)
  Wilcoxon rank sum test with continuity correction

data:  x and y
W = 139.5, p-value = 0.05934
alternative hypothesis: true location shift is not equal to 0
```

با توجه به مقدارp-value در خروجی فوق فرض برابری میانگینها پذیرفته می شود.

۴. آزمون کروسکال – والیس

آزمون کروسکال-والیس در مواردی استفاده می شود که هدف مقایسه ی میانگینهای k جامعه ی مستقل پیوسته باشد و در واقع به عنوان یک جایگزین برای آنالیز واریانس یکطرفه می تواند مورد استفاده قرار گیرد.

این آزمون نیز همانند آزمون ویلکاکسون مبتنی بر مجموع رتبه ها می باشد. به عبارت دیگر پس از ادغام مقادیر همه گروهها ، آنها را به صورت غیر نزولی مرتب نموده و مجموع رتبه های متعلق به گروه i –ام که با R_i نشان داده می شودرا محاسبه می نماییم. بدیهی است به هنگام رتبه گذاری ، به مقادیر مساوی میانگین مجموع رتبه های اختصاص یافته تعلق می گیرد.

در این حالت آماره ی آزمون $H_0: \mu_1 = \mu_2 = ... = \mu_k$ در برابر فرض مقابل ،از رابطه ی زیر محاسبه می شود

$$H = \frac{12}{n(n+1)} \sum_{i=1}^{k} \frac{R_i^2}{n_i} - 3(n+1)$$

که در آن $n = n_1 + n_2 + ... + n_k$ و k تعداد جامعه های مورد مقاسه می باشد. آماره ی H دارای توزیع کای-دو با k-1 درجه آزادی می باشد. دستور آزمون فوق در نرم افزار R به صورت زیر می باشد ،

```
>Kruskal.test (y~A, data=mydata)
```

که در آن y مقادیر عددی و A فاکتور گروهی می باشد.

مثال ۴.اعداد زیر مقدار مسافت پیموده شده بر حسب کیلومتر به ازای هر لیتر بنزین می باشد که بر روی سه نوع بنزین مختلف آزمایش شده است.

بنزین A : ۲۰،۳۱،۲۴،۳۳،۲۳،۲۴،۲۸،۱۶،۱۹،۲۶

بنزین B : ۲۹،۱۸،۲۹،۱۹،۲۰،۲۱،۳۴،۳۳،۳۰،۲۳

بنزین C : ۱۹،۳۱،۱۶،۲۶،۳۱،۳۳،۲۸،۲۸،۲۵،۳۰

در سطح معنی داری ۰۵/. آیا میتوان پذیرفت که کارایی هر سه بنزین یکسان است؟

حل : ابتدا مقادیر متغیرها به عنوان y و فاکتور گروهی به عنوان A را وارد می کنیم و سپس دستورات مربوطه را به صورت زیر اجرا می نماییم.

```
>y=c(20,31,24,33,23,24,28,16,19,26,29,18,29,19,20,21,34,33,30,23
     ,19,31,16,26,31,33,28,28,25,30)
> A=c(rep(c(1,2,3),c(10,10,10)))
> mydata=data.frame(y,A)
> kruskal.test(y~A, data=mydata)


Kruskal-Wallis rank sum test

data:  y by A
Kruskal-Wallis chi-squared = 0.8683, df = 2, p-value = 0.6478
```

با توجه به مقدار p-value در خروجی فوق فرض برابری میانگینها پذیرفته می شود.

۵. آزمون فریدمن

آزمون فریدمن همانند آزمون کروسکال والیس برای مقایسه ی اختلاف بین چند نمونه بکار می رود با این تفاوت که در آزمون فریدمن ، نمونه ها از هم مستقل نمی باشند. در واقع آزمون فریدمن ، تعمیم آزمون رتبه علامت دار ویلکاکسون می باشد که به عنوان جایگزین آزمون t ی زوج شده بکار می رود. بنابراین ، آزمون فریدمن به عنوان جایگزین آنالیز واریانس با اندازه های مکرر (repeated measure ANOVA)می باشد که در آنها بین نمونه ها نوعی همبستگی وجود دارد.

آماره ی آزمون فرید من به صورت زیر است ،

$$Q = \frac{12}{nk(k+1)} \sum_{j=1}^{k} R_j^2 - 3n(k+1)$$

که در آن k و n به ترتیب عبارتست از تعداد ستونها (آزمودنی ها) و تعداد سطرها (بلوکها) و همچنین R_j عبارتست از مجموع رتبه های سطر **j** ام. در این حالت آماره ی Q دارای توزیع کای دو با k−1 درجه آزادی می باشد.

دستور R برای اجرای آزمون فوق هرگاه داده ها به صورت ماتریسی (بلوکی) وارد شده باشند به صورت

```
>friedman.test(mydata)
```

میباشدکه این دستور در library(stats) موجود می باشد.

مثال ۵. یک شرکت فعال در زمینه زیست محیطی در نظر دارد میزان تمیزی آب یک رودخانه را با توجه به معیار میزان اکسیژن موجود در آن بررسی نماید. به همین منظور ، میزان اکسیژن موجود در آب آن رودخانه در ۱۲ نقطه ، قبل ، یک ماه بعد و یک سال بعد از تصفیه ای که بر روی آب رودخانه صورت گرفته اندازه گیری شده که مقادیر آن در جدول زیرآمده است. در سطح معنی داری ۵./. آزمون کنیدآیا تصفیه ی رودخانه بر میزان اکسیژن موجود در آب موثر بوده است یا خیر؟

	میزان اکسیژن		
یک سال بعد	یک ماه بعد	قبل	نقاط مورد آزمایش
۱۳/۲	۱۳/۶	۱۷/۴	۱
۹/۸	۱۰/۱	۱۵/۷	۲
۹/۷	۱۰/۳	۱۲/۹	۳
۹	۹/۲	۹/۸	۴
۱۰/۷	۱۱/۱	۱۳/۴	۵
۱۹/۶	۲۰/۴	۱۸/۷	۶

۱۰/۲	۱۰/۴	۱۳/۹	۷
۱۱/۵	۱۱/۴	۱۱	۸
۵/۲	۴/۹	۵/۴	۹
۹/۲	۸/۹	۱۰/۴	۱۰
۱۱	۱۱/۲	۱۶/۴	۱۱
۴/۶	۴/۸	۵/۶	۱۲

حل :

فرض صفرمساله $H_0: \theta_1 = \theta_2 = \theta_3$ می باشد که بیانگر عدم تاثیر روش تصفیه ی آب رودخانه می باشد.

ابتدا داده ها را به صورت ماتریسی وارد R می کنیم و سپس دستور مربوطه را به صورت زیر اجرا می کنیم.

```
>waterdata=matrix(c(17.4,15.7,12.9,9.8,13.4,18.7,13.9,11,5.4,10.

     4,16.4,5.6,13.6,10.1,10.3,9.2,11.1,20.4,10.4,11.4,4.9,8.9

     ,11.2,4.8,13.2,9.8,9.7,9,10.7,19.6,10.2,11.5,5.2,
      9.2,11,4.6),nrow=12,byrow=F,
      dimnames=list(1:12,c("before","after 1m","after 1 y")))

>friedman.test(waterdata)
Friedman rank sum test

data:  waterdata
Friedman chi-squared = 9.5, df = 2, p-value = 0.008652
```

با توجه به مقدار p-value در خروجی فوق فرض صفر رد وی شود.

۶. آزمون های دو جمله ای و نسبتها

این آزمون ها مبتنی بر توزیع دوجمله ای و داده های شمارشی می باشند. در آزمون دو جمله ای نسبت در یک جامعه مورد آزمون قرار می گیرد و در آزمون نسبت ، نسبتهای دو و یا چند جامعه با هم مقایسه می شوند.

در آزمون دوجمله ای فرض صفر $H_0 : p = p_0$ در برابر فرض مقابل مناسب آزمون خواهد شد و همچنین در آزمون نسبت ، فرض صفر به صورت $H_0 : \pi_1 = \pi_2 = ... = \pi_k$ خواهد بود که در آن برابری نسبتهای چند جامعه آزمون خواهد شد.

نرم افزار R ، در همه ی آزمون های فوق ، آماره ی آزمون را بر اساس توزیع کای دو محاسبه می نماید و مقدار p-value را جهت تحلیل آزمون ارائه می نماید.

دستور کلی R برای حل آزمونهای فوق به صورت زیر است که جزئیات آن متناسب با نوع آزمون تغییر خواهد کرد.

```
>prop.test(x,n,p=NULL,
alternative=c(”two.sided”,”less”,”greater”)
           ,conf.level=0.95, correct=T)
```

در دستور فوق ، correct=T، این امکان را به R می دهد که به هنگام محاسبه ی آماره ی آزمون از تصحیح پیوستگی یتس استفاده نماید .

مثال ۶. یک شرکت تولید بنزین مدعی است درصد افرادی که از بنزین آن شرکت استفاده نمی کنند کمتر از ۲۰ درصد می باشد. برای این منظور یک بررسی نشان داده که از میان ۲۰۰ دارنده اتوموبیل ، ۲۲ نفر از بنزین تولیدی این شرکت استفاده نکرده اند. آزمون فوق را در سطح معنی داری ۰/۰۱ انجام دهید.

حل:

فرض صفر و فرض مقابل در آزمون فوق به صورت $H_0 : p = 0.2$ در مقابل $H_0 : p < 0.2$ می باشد. دستور R برای حل آزمون فوق عبارتست از

```
>prop.test(x=22,n=200,p=0.2,alternative=”less”,conf.level=0.99)

1-sample proportions test with continuity correction

data:  22 out of 200, null probability 0.2
X-squared = 9.5703, df = 1, p-value = 0.0009887
alternative hypothesis: true p is less than 0.2
99 percent confidence interval:
 0.0000000 0.1750043
sample estimates:
   p
0.11
```

با توجه به مقدار p-value در خروجی فوق فرض صفر رد وی شود.

مثال ۷. در نمونه های تصادفی متشکل از ۲۵۰ نفر از افراد پردرآمد ، ۲۰۰ نفر از افراد با درآمد متوسط و ۱۵۰ نفر از افراد با درآمد کم ، به ترتیب تعداد ۱۵۵، ۱۸۸ و ۸۷ نفر موافق تصویب یک قانون خاص هستند. در سطح معنی داری ۰/۰۵ آزمون کنید

الف: آیا نسبت افراد موافق در همه ی گروهها یکسان است ؟

ب: آیا نسبت افراد موافق در همه ی گروهها یکسان و برابر ۰/۷ است؟

حل:

دستور R برای حل قسمت الف به صورت زیر است،

```
>prop.test(x=c(155,188,87),n=c(250,200,150),p=NULL,conf.level=0.
     95)

3-sample test for equality of proportions without continuity
        correction

data:  c(155, 188, 87) out of c(250, 200, 150)
X-squared = 74.4295, df = 2, p-value < 2.2e-16
alternative hypothesis: two.sided
sample estimates:
prop 1 prop 2 prop 3
  0.62   0.94   0.58
```

و برای حل قسمت ب به صورت زیر می باشد.

```
>prop.test(x=c(155,188,87),n=c(250,200,150),p=c(0.7,0.7,0.7),
                    conf.level=0.95)

3-sample test for given proportions without continuity
correction

data:  c(155, 188, 87) out of c(250, 200, 150), null
probabilities c(0.7, 0.7, 0.7)

X-squared = 72.7619, df = 3, p-value = 1.093e-15
alternative hypothesis: two.sided
null values:
prop 1 prop 2 prop 3
   0.7    0.7    0.7
sample estimates:
```

```
prop 1 prop 2 prop 3
  0.62   0.94   0.58
```

همانطور که در خروجی فوق مشخص می باشد فرض صفر رد می شود و همه نسبتها برابر ۰/۷ نمی باشند.

۷. جدولهای توافقی

جدولهای توافقی ، جداولی هستند که از آنها برای آزمونهای استقلال و یا آزمون همگنی معیارهای مورد سنجش استفاده می شوند که بسته به سطوح معیارها ، دارای r سطر و c ستون می باشند. در آزمون استقلال ، فرضیه ی استقلال دو متغیر که دست کم یکی از آنها کیفی است مورد بررسی قرار می گیرد در حالی که آزمون همگنی این فرضیه را می آزماید که آیا نمونه ها از جمعیتهایی همگن انتخاب شده اند یا خیر؟

ساختار نظری هر دو حالت فوق یکسان است و آماره ی آزمون در هر دو عبارتست از

$$\chi^2 = \sum_{i=1}^{k} \frac{(O_i - e_i)^2}{e_i}$$

که در آن O_i و e_i به ترتیب فراوانیهای مشاهده شده و مورد انتظار می باشند. آماره ی فوق دارای توزیع کای دو با $df = (r-1)(c-1)$ درجه آزادی می باشد.

نحوه ورود داده ها و اجرای دستور R جهت حل جدولهای توافقی در مثال زیر داده شده است.

مثال ۸ . جهت بررسی نگرش دانشجویان نسبت به برنامه های آموزشی دانشگاه ، پرسشنامه ای طراحی شده که در آن از دانشجویان تازه وارد، دانشجویان سال دوم و دانشجویان سال آخر تحصیلی، سوالاتی در این خصوص پرسیده شده است. دراین سوالات از دانشجویان پرسیده شده که آیا برنامه های آموزشی دانشگاه پایین تر از توانایی شما ، متناسب با توانایی شما و یا بالاتر از توانایی شماست. داده های این تحقیق در جدول زیر داده شده است. بر اساس این داده ها آزمون کنید آیا بین مقطع تحصیلی و سطح نگرش آنها به برنامه های آموزشی رابطه ای وجود دارد یا از هم مستقلند.

	سال اول	سال دوم	سال آخر
پایین تر از سطح توانایی	۵	۸	۱۱
متناسب با سطح توانایی	۳۰	۳۵	۴۰
بالاتر از سطح توانایی	۲۵	۱۷	۹

حل:

فرض صفر در آزمون فوق عبارت از مستقل بودن مقطع تحصیلی و سطح نگرش دانشجویان به برنامه آموزشی می باشد و فرض مقابل بیانگر وابستگی این دو متغیر می باشد. برای حل مساله فوق به کمک نرم افزار R دستورات زیر را به ترتیب اجرا می کنیم.

```
> Studata=matrix(c(5,8,11,30,35,40,25,17,9),nrow=3,byrow=T)
> colnames(Studata)<-c("year1","year2","year3")
> rownames(Studata)<-c("lower","medium","higher")
> Studata
       year1 year2 year3
lower      5     8    11
medium    30    35    40
higher    25    17     9
```

جهت محاسبه ی مقادیر آماره ی توزیع کای-دو و انجام آزمون فرض از دستور زیر استفاده می شود.

```
> chisq.test(Studata)

        Pearson's Chi-squared test

data:  Studata
X-squared = 11.208, df = 4, p-value = 0.02432
```

همانطور که در خروجی فوق مشخص شده با توجه به مقدارp-value فرض صفر رد می شود و لذا مقطع تحصیلی و سطح نگرش دانشجویان به برنامه های آموزشی از هم مستقل نمی باشند.

برای بدست آوردن مقادیر فوق ، همچنین برای محاسبه مقادیر مورد انتظار می توان از روش زیر نیز استفاده نمود.

```
>E=chisq.test(Studata)$expected
> O=chisq.test(Studata)$observed
> chi=(O-E)^2/E
           year1 year2     year3
lower  1.1250000     0 1.1250000
medium 0.7142857     0 0.7142857
higher 3.7647059     0 3.7647059
sum(chi)
[1] 11.20798
```

۸. ضرایب همبستگی پیرسن و اسپیرمن

ضریب همبستگی ابزاری آماری برای تعیین نوع و درجه رابطه ی دو متغیر می باشد که این متغیرها ممکن است کمی (فاصله ای – رتبه ای) و یا اسمی باشند. در حالتی که هر دو متغیر کمی فاصله ای باشند از ضریب همبستگی پیرسن جهت محاسبه ی میزان و نوع همبستگی آنها استفاده می شود. ضریب همبستگی پیرسن بین دو متغیر x و y از رابطه ی زیر محاسبه می شود،

$$r=\frac{\sum xy-n\bar{x}\bar{y}}{\sqrt{\sum x^2-n\bar{x}^2}\sqrt{\sum y^2-n\bar{y}^2}}$$

در صورتی که داده ها هر دو از نوع رتبه ای باشند ، از ضریب همبستگی رتبه ای اسپیرمن جهت محاسبه ی میزان و نوع همبستگی میان آنها استفاده می شود. همچنین در حالتی که داده ها کمی فاصله ای باشند و تعداد آنها کم و توزیع نرمال برای آنها معقول نباشد می توان از رتبه ی داده ها به جای خود داده ها استفاده نمود و در این حالت به جای ضریب همبستگی پیرسن ، از ضریب همبستگی رتبه ای اسپیرمن برای محاسبه ی میزان همبستگی دو متغیر استفاده نمود. ضریب همبستگی رتبه ای اسپیرمن از رابطه ی زیر محاسبه می شود،

$$r_s=1-\frac{6\sum_{i=1}^{k}d_i^2}{n(n^2-1)}$$

دستور کلی R برای محاسبه ی ضرایب همبستگی فوق به صورت زیر می باشد

```
>cor(x,y=NULL, method=c(”pearson”,”spearman”))
```

که با توجه به نوع داده ها و همچنین نوع ضریب همبستگی جزئیات فوق تغییر می کند.
مثال ۹. فرض کنید X میزان بارندگی روزانه (واحد ۰/۰۱ سانتیمتر) و Y میزان آلودگی هوا (واحد میکروگرم در هر متر مکعب هوا) باشد. با استفاده از داده های زیر ضریب همبستگی پیرسن را بدست آورید.

میزان بارندگی	۴/۳	۴/۵	۵/۹	۵/۶	۶/۱	۵/۲	۳/۸	۲/۱	۷/۵
آلودگی هوا	۱۲۶	۱۲۱	۱۱۶	۱۱۸	۱۱۴	۱۱۸	۱۳۲	۱۴۱	۱۰۸

```
>rain=c(4.3,4.5,5.9,5.6,6.1,5.2,3.8,2.1,7.5)
>pollution=c(126,121,116,118,114,118,132,141,108)
>cor(rain,pollution,method=”pearson”)

> -0.9786584
```

مثال ۱۰ : از دو نفر خواسته شده که ۹ نوع چای مختلف را از نظر رنگ و طعم رتبه بندی نمایند. نتایج آن در جدول زیر داده شده است. ضریب همبستگی اسپیرمن را بدست آورید.

نوع چای	A	B	C	D	E	F	G	H	I
نفر اول	۱	۵	۹	۷	۴	۶	۸	۲	۳
نفر دوم	۴	۳	۶	۸	۲	۷	۹	۱	۵

```
>judge1=c(1,5,9,7,4,6,8,2,3)
>judge2=c(4,3,6,8,2,7,9,1,5)
>mydata=data.frame(judge1,judge2)
>cor(mydata,method="spearman")

  judge1    judge2
judge1 1.0000000 0.7166667
judge2 0.7166667 1.0000000
```

فصل۹

رگرسیون خطی

۱.مقدمه

در این فصل رگرسیون خطی چندگانه شامل برازش مدل ، تشخیص های رگرسیونی (هم خطی چندگانه ، ناهمسانی واریانس، استقلال باقی مانده ها ، غیر خطی بودن رابطه ی متغیرهای مستقل و وابسته) مورد بررسی قرار می گیرند. همچنین روشهای برازش مدل شامل (Stepwise , Forward, Backward) و تجزیه و تحلیل باقی مانده ها بر اساس مقادیر (استاندارد، پرس، استیودنت) بررسی خواهند شد.

برای اجرای مراحل فوق به کمک نرم افزار R ، لازم است ابتدا بسته ی نرم افزاری library(car) نصب شود زیرا بسته ی فوق شامل بسیاری از دستورات جهت برازش مدل رگرسیون خطی و همچنین اعتبار سنجی مدل می باشد. البته برخی از مراحل اعتبارسنجی مدل رگرسیونی در این بسته وجود ندارد و بایستی بسته های دیگری برای این منظور نصب شود که در ادامه هر جا لازم باشد به آن اشاره خواهد شد.

در این فصل ابتدا مفاهیم فوق به صورت اختصار توضیح داده خواهند شد و سپس برای اجرای مراحل فوق به کمک نرم افزار R ، سعی شده که دستورات در قالب مثالهای عددی بیان شود که این امر به درک بهتر موضوع کمک خواهد کرد.

فرض کنید متغیرهای مستقل $x_1, x_2, ..., x_k$ و متغیر وابسته ی y وجود داشته باشد. در این صورت رگرسیون خطی عبارتست از یک رابطه ی خطی که میان متغیر وابسته (پاسخ) و متغیرهای مستقل وجود دارد که این رابطه بر اساس یک نمونه ی n تایی به صورت زیر بیان می شود.

$$y_i = \beta_0 + \beta_1 x_{1i} + ... + \beta_k x_{ki} + \varepsilon_i \qquad i = 1,2,...,n$$

که در آن β_j ها ضرایب رگرسیونی و ε_i مقدار باقی مانده یا خطای مدل می باشد. در یک مدل رگرسیونی فرض بر این است که باقی ماندها ، متغیرهای تصادفی مستقل ، دارای توزیع نرمال با میانگین $E(\varepsilon_i) = 0$ و واریانس ثابت $Var(\varepsilon_i) = \sigma^2$ می باشند . بررسی هر کدام از این شرایط (مستقل بودن ، نرمال بودن، ثابت بودن واریانس)، خود بخشی از تشخیص رگرسیونی (اعتبار سنجی) مدل می باشد.

۲.برازش مدل رگرسیون خطی

برازش مدل رگرسیونی عبارتست از برآورد ضرایب رگرسیونی به کمک روشهای آماری که مدل برازش شده به صورت زیر بیان می شود.

$$\hat{y}_i = b_0 + b_1 x_{1i} + \ldots + b_k x_{ki}$$

جهت برازش مدل و برآورد پارامترها به کمک نرم افزار R ، از داده های مثال زیر استفاده می نماییم.

مثال: داده های زیر مربوط به اطلاعات بیمارستانی(Hospital Data) است که در آن متغیر وابسته ی y و متغیرهای مستقل x_1 ، x_2 ،...، x_5 به صورت زیر تعریف شده اند.(Raymond H. Myers 1990)

y: Monthly man-hours

x_1: Average daily patient load

x_2: Monthly X-ray exposure

x_3: Monthly occupied bed days

x_4: Eligible population in the area÷ 1000

x_5: Average length of patients' stay in days

Site	y	x_1	x_2	x_3	x_4	x_5
1	556.52	15.57	2463	472.92	18.0	4.45
2	696.82	44.02	2048	1339.75	9.5	6.92
3	1033.15	20.42	3940	620.25	12.8	4.28
4	1603.62	18.74	6505	568.33	36.7	3.90
5	1611.37	49.20	5723	1497.60	35.7	5.50
6	1613.27	44.92	11520	1365.83	24.0	4.60
7	1854.17	55.48	5779	1687.00	43.3	5.62
8	2160.55	59.28	5969	1639.92	46.7	5.15
9	2305.58	94.39	8461	2872.33	78.7	6.18
10	3503.93	128.02	20106	3655.08	180.5	6.15
11	3571.89	96.00	13313	2912.00	60.9	5.88
12	3741.40	131.42	10771	3921.00	103.7	4.88
13	4026.52	127.21	15543	3865.67	126.8	5.50
14	10343.81	252.90	36194	7684.10	157.7	7.00
15	11732.17	409.20	34703	12446.33	169.4	10.78
16	15414.94	463.70	39204	14098.40	331.4	7.05
17	18854.45	510.22	86553	15524.00	371.6	6.35

پس از وارد نمودن داده ها ، دستور زیر را اجرا می کنیم.

```
>library(car)

>Mydata=data.frame(y,x1,x2,x3,x4,x5)

>fit=lm(y~., data=mydata)
```

دستور زیر خلاصه ای از خروجی شامل آزمونهای معنی داری مدل و پارامترهای مدل ، ضریب تعیین و اطلاعاتی در مورد باقی مانده ها را ارائه می کند.

```
> summary(fit)

               lm(formula = y ~ ., data = mydata)

                        Residuals:
             Min       1Q   Median       3Q      Max
         -611.08  -430.95   -69.25   333.41  1576.53

                        Coefficients:

              Estimate Std. Error t value Pr(>|t|)
     (Intercept) 1959.41680 1071.99366   1.828   0.0948 .
      x1           -15.77586   97.71013  -0.161   0.8747
     x2             0.05592    0.02126   2.630   0.0234 *
      x3             1.58698    3.09389   0.513   0.6181
      x4            -4.21005    7.18072  -0.586   0.5695
     x5          -393.83771  209.76501  -1.878   0.0872 .
                          ---
```

```
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 642.5 on 11 degrees of freedom
Multiple R-squared: 0.9908,     Adjusted R-squared: 0.9867
F-statistic: 237.6 on 5 and 11 DF,  p-value: 8.112e-11
```

دستورات زیر به ترتیب فاصله اطمینان ۹۵ درصد برای پارامترهای مدل :

```
> confint(fit,level=.95)
```

مقادیر پیش بینی (برازش شده) :

```
> fitted(fit)
```

مقادیر باقی مانده ها :

```
> residuals(fit)
```

و ماتریس واریانس-کواریانس پارامترهای مدل را ارائه می نمایند.

```
> vcov(fit)
```

۳.بررسی فرضیات مدل

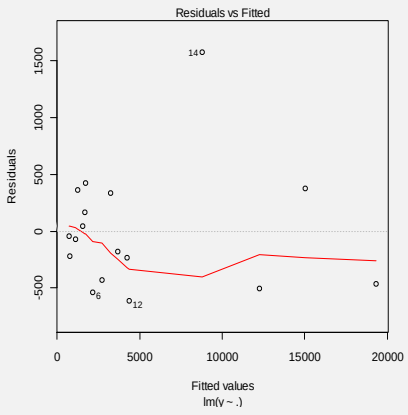
بعد از برازش مدل رگرسیونی ، بایستی مدل برازش شده اعتبار سنجی شود که شامل قسمتهای مختلفی می باشد . یکی از مهمترین قسمتها در اعتبار سنجی مدل رگرسیونی بررسی فرضیات مربوط به باقیمانده ها شامل ثابت بودن واریانس ، نرمال بودن و مستقل بودن باقیمانده ها می باشد. بررسی مشاهدات غیر معمول و پرنفوذ از دیگر اجزای اعتبار سنجی مدل رگرسیونی می

باشد. همچنین فرضیه ی خطی بودن رابطه ی متغیرهای مستقل و متغیر وابسته ، عدم همبستگی درونی میان متغیرهای مستقل (هم خطی) و در نهایت نحوه انتخاب و ورود متغیرهای مستقل به مدل در اعتبار سنجی مدل رگرسیونی باید در نظر گرفته شوند که در زیر به اختصار به آنها اشاره خواهد شد.

الف) ثابت(همگن) بودن واریانس مانده ها (Heteroscedasticity)

جهت بررسی ثابت بودن واریانس مدل معمولا از نمودار باقیمانده ها استفاده می شود که مهمترین آنها رسم نمودار مقادیر برازش شده ، $\hat{Y}_i$ در مقابل مقدار باقیمانده ها یعنی e_i می باشد. جهت رسم نمودار فوق به کمک R از دستور زیر استفاده می کنیم.

```
>plot(fit, which=1)
```



پس از مشاهده نمودار ، در صورتی که مقادیر باقیمانده ها حول خط صفر پراکنده شده باشند و نشان دهنده روند خاصی نباشند می توان نتیجه گرفت که واریانس مدل مقداری ثابت می باشد.

یک روش ابتکاری نیز در این زمینه وجود دارد که توسط Faraway(2005) ارائه شد . این روش بر مبنای آزمون معناداری مدل رگرسیونی $|e_i| = \beta_0 + \beta_1 \hat{Y}_i$ می باشد . معنی دار بودن مدل رگرسیونی فوق بدین معنی است که بین دو مقدار رابطه وجود دارد و لذا واریانس مدل ثابت نمی باشد. جهت بررسی این موضوع از دستور زیر استفاده میشود.

```
>summary(lm(abs(residuals(fit)~fitted(fit))))
```

```
F-statistic:  2.33 on 1 and 15 DF,  p-value: 0.1477
```

همانطور که مشاهده می شود با توجه به آزمون معنی داری مدل رگرسیونی، برای داده های مثال فوق این مدل معنادار نیست و لذا می توان نتیجه گرفت که واریانس ثابت است.

البته چندین آزمون نیز در این زمینه وجود دارد که مهمترین آنها آزمون Breusch-Pagan می باشد که در آن فرض صفر بیانگر ثابت بودن واریانس مدل می باشد.

جهت انجام این آزمون خواهیم داشت،

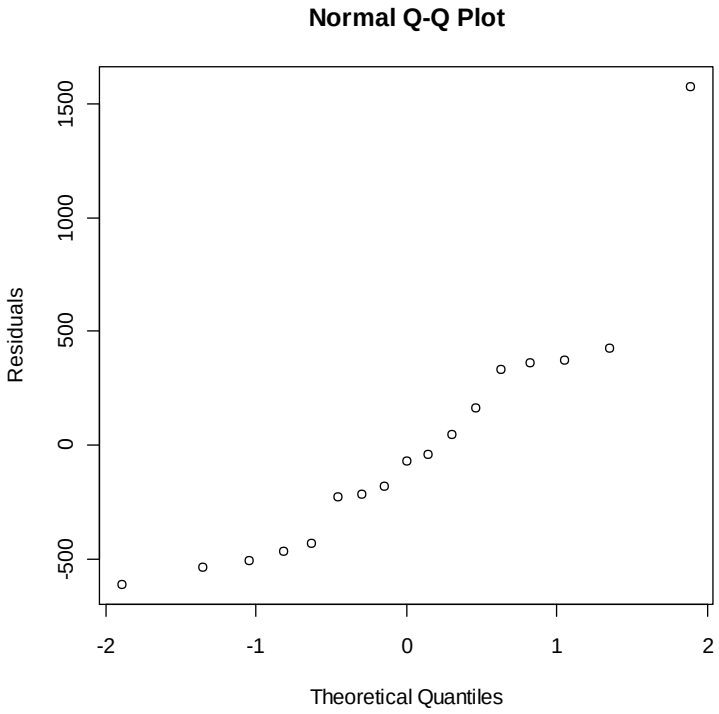
```
>library(zoo)
> library(lmtest)
> bptest(fit)

        studentized Breusch-Pagan test

data:  fit
BP = 2.2085, df = 5, p-value = 0.8196
```

همانگونه که از نتیجه ی آزمون مشخص می باشد ، فرض صفر پذیرفته می شود و لذا می توان نتیجه گرفت که واریانس مانده ها ثابت می باشد.

ب) نرمال بودن باقیمانده ها (Normality)

جهت بررسی نرمال بودن باقیمانده ها می توان از نمودار qq-plot ، نمودار هیستوگرام و همچنین آزمون Shapiro استفاده نمود.جهت رسم نمودار Q-Q plotدر R از دستور زیر استفاده می شود.

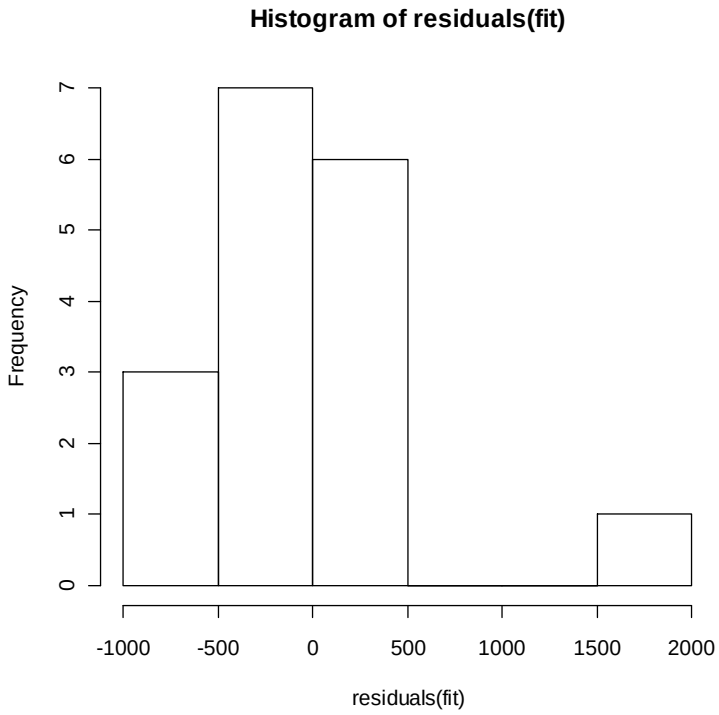
```
>qqnorm(residuals(fit),ylab=”Residuals”)
```



در صورتی که نقاط روی خط نیمساز واقع شده باشند ، توزیع باقیمانده ها نرمال است و در صورتی که اکثر نقاط بالای خط نیمساز باشند توزیع چوله به راست و در صورتی که پایین خط باشند توزیع چوله به چپ می باشد.

رسم نمودار هیستوگرام نیز می توان در این زمینه مفید باشد که از دستور زیر بدست می آید.

```
>hist(residuals(fit))
```


Histogram of residuals(fit)
Frequency
0
1
2
3
4
5
6
7
-1000
-500
0
500
1000
1500
2000
residuals(fit)

.

جهت بررسی فرض نرمال بودن باقی مانده ها از آزمون Shapiro نیز می توان استفاده نمودکه فرض صفر در آن معادل نرمال بودن باقیمانده ها می باشد. جهت انجام این آزمون از دستور زیر استفاده می شود.

```
>library(stats)
>res=residuals(fit)
>Shapiro.test(res)

                  Shapiro-Wilk normality test

                          data:  res
                  W = 0.8591, p-value = 0.01481
```

ج) همبستگی باقیمانده ها (Autocorrelation)

جهت بررسی همبستگی میان باقیمانده های مدل (Autocorrelation) می توان از آزمون Durbin-Watson استفاده نمود که به صورت زیر می باشد

$$\begin{cases} H_0 : \rho = 0 \\ H_1 : \rho > 0 \end{cases}$$

که در آن فرض صفر بیانگر مستقل بودن باقیمانده ها می باشد. جهت انجام آزمون فوق در R ابتدا باید library(zoo) و library(lmtest) نصب شود و سپس دستور مربوطه به صورت زیر اجرا شود.

```
>library(zoo)
>library(lmtest)
>dwtest(fit)

                      Durbin-Watson test

                          data:  fit
                  DW = 2.7319, p-value = 0.927
  alternative hypothesis: true autocorrelation is greater than 0
```

د) هم خطی چندگانه (Multicollinearity)

همانطور که بیان شد ،هدف رگرسیون ،کشف ارتباط خطی میان یک متغیر پاسخ و چند متغیر مستقل می باشد. متغیرهای مستقل در رگرسیون ، همانگونه که از اسم آنها پیداست از هم مستقل می باشند و در صورتی که این فرض برقرار نباشد و میان برخی از متغیرهای مستقل ، همبستگی وجود داشته باشد آنگاه برآورد پارامترهای مدل رگرسیونی را تحت تاثیر قرار می گیرد و نتایج نمی تواند قابل استناد باشد.

به همبستگی خطی میان متغیرهای مستقل ، هم خطی چندگانه (multicollinearity) گویند که برای تشخیص آن از آماره ی VIF (Variance Inflation factor) گویند. آماره ی VIF از رابطه ی زیر محاسبه می شود

$$VIF = \frac{1}{1-R_j^2}$$

که در آن R_j^2 عبارتست از ضریب تعیین یک مدل رگرسیونی که در آن متغیر مستقل j ام به عنوان متغیر پاسخ در مقابل بقیه ی متغیرهای مستقل بدست آمده باشد. مقادیر بزرگ VIF نشان دهنده وجود همبستگی میان متغیرهای مستقل می باشند . بر اساس یک قاعده مقادیر بزرگتر از ۵ ، مقادیری بزرگ می باشند که باید مورد توجه قرار گیرند و مقدار عددی ۱۰ تعیین کننده وجود هم خطی می باشد. جهت محاسبه ی میزان هم خطی در R از دستور زیر استفاده میشود.

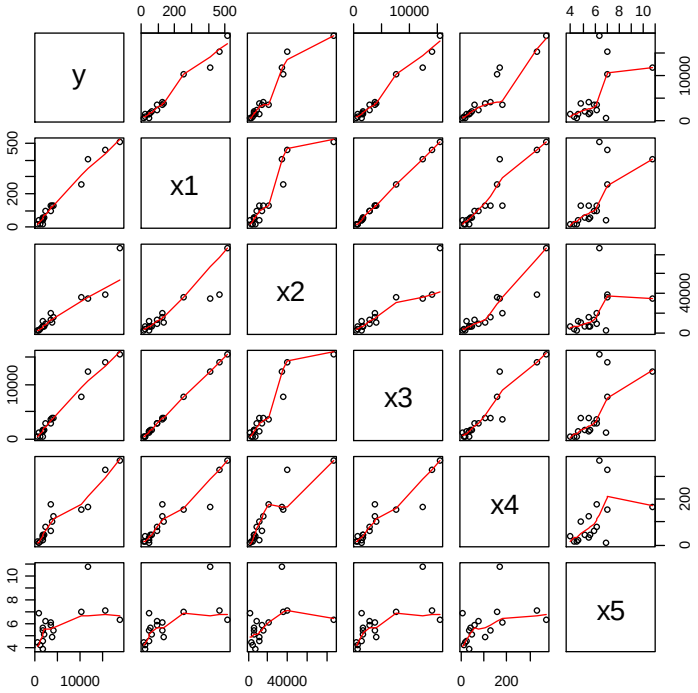
```
>library(car)
>vif(fit)

vif(fit)
         x1          x2          x3          x4          x5
9597.589140    7.937602 8933.109934   23.293651    4.279958
```

۴.بررسی خطی بودن ارتباط میان متغیرهای مستقل و متغیر پاسخ

در بسیاری از مواقع ، غیر خطی بودن ارتباط میان متغیر وابسته و متغیرهای مستقل سبب می شود که مدل رگرسیونی از اعتبار لازم برخوردار نباشد و باعث نقض برخی از فرضیات پایه در مدل رگرسیونی شود. در این حالت با اعمال برخی تبدیلات می توان رابطه ی میان متغیر مذکور و متغیر پاسخ را از غیر خطی به خطی تبدیل نمود . جهت بررسی ارتباط میان متغیرهای مستقل و متغیر وابسته می توان از نمودار پراکنش استفاده نمود. همچنین با توجه به اینکه نمودار پراکنش ،ارتباط میان متغیرهای مستقل

را نیز به صورت نمودار نشان می دهد ، می توان از آن جهت کشف هم خطی استفاده نمود. جهت رسم نمودار فوق به کمک R از دستور زیر استفاده می شود.

```
>pairs(mydata, panel=panel.smooth)
```



۵.انتخاب متغیرهای مدل

در یک رگرسیون خطی چندگانه این امکان وجود دارد که همه ی متغیرهای مستقل که وارد مدل می شوند موثر نباشند و یا حتی وجود آنها در مدل باعث شود که از اعتبار مدل کاسته شود. به همین جهت بایستی آنها را از مدل حذف نمود که در نتیجه مدل رگرسیونی بدون حضور آنها باعث می شود که مدلی کاراتر باشد. به همین منظور، روشهایی جهت حذف متغیرهای غیر ضروری از مدل وجود دارد که شامل Forward selection و Backward Elimination و همچنین Stepwise می باشد.

انتخاب متغیرها با استفاده از روشهای فوق ، می تواند بر اساس دو معیار ضریب تعیین R^2 و یا آماره ی AIC (Akaike Information Criterion) باشد که روشهای موجود در نرم افزار R بر غالبا بر اساس معیار AIC می باشند.

مقدار آماره ی AIC از رابطه ی زیر محاسبه می شود.

$$AIC=-2\text{loglikelihood}+2p$$

که در رگرسیون خطی p برابر تعداد پارامترهای مدل می باشد و

$$-2\text{loglikelihood}=n\log\left(\frac{SS_{\text{Res}}}{n}\right)$$

هر مدلی که دارای AIC کمتری باشد می تواند مدل بهتری باشد.

در روش Forward ابتدا فرض بر این است که هیچ متغیری در مدل وجود ندارد . بر اساس این معیار اولین متغیری که وارد مدل می شود متغیری است که بیشترین میزان را در کاهش مقدار AIC مدل ایجاد می نماید. جهت ورود دومین متغیر به مدل ، باز هم از همین ملاک استفاده می شود و دومین متغیری که بیشترین کاهش را در مقدار AIC ایجاد می کند وارد مدل می شود و این عمل تا جایی ادامه می یابد که متغیرهای باقیمانده نتوانند کاهشی را در مقدار AIC مدل ایجاد نمایند.

روش Backward ساختاری مانند روش فوق دارد با این تفاوت که ابتدا همه ی متغیرها وارد مدل می شوند و اولین متغیری که از مدل حذف می شود متغیری است که با حذف آن بیشترین کاهش در مقدار AIC مدل ایجاد می شود و به همین ترتیب بقیه ی متغیرها حذف می شوند.

روش Stepwise تلفیقی از دو روش فوق می باشد و ممکن است متغیری در یک مرحله وارد مدل شود ولی در مرحله ی بعد حذف شود. دلیل آن هم ممکن است مربوط به وجود هم خطی (همبستگی) میان متغیر موجود و متغیری که وارد مدل می شود باشد.

جهت اجرای روشهای فوق به کمک نرم افزار R ، می توان مراحل زیر را به ترتیب انجام داد.

ابتدا یک مدل رگرسیونی را بدون حضور متغیرها به صورت زیر بر داده ها برازش می دهیم.

```
>null=lm(y~1,data=mydata)
```

سپس مدل رگرسیونی دیگری را با حضور همه ی متغیرها ایجاد می نماییم.

```
>full=lm(y~., data=mydata)
```

جهت روش Forward از دستور زیر استفاده می کنیم.

```
>step(null,scope=list(lower=null,upper=full),method="forward")


Start:  AIC=294.17
y ~ 1

       Df Sum of Sq       RSS    AIC
+ x3    1 481039799  13761075 235.27
+ x1    1 480703027  14097847 235.68
+ x2    1 442000282  52800592 258.13
+ x4    1 437544920  57255954 259.51
+ x5    1 165665800 329135074 289.24
<none>              494800874 294.17

Step:  AIC=235.27
y ~ x3

       Df Sum of Sq       RSS    AIC
+ x2    1   7188838   6572236 224.71
+ x5    1   6214116   7546958 227.06
+ x4    1   1572631  12188443 235.21
<none>               13761075 235.27
+ x1    1    161659  13599416 237.07
- x3    1 481039799 494800874 294.17
```

```
Step:  AIC=224.71
y ~ x3 + x2

       Df Sum of Sq      RSS    AIC
+ x5    1   1655450  4916786 221.77
<none>               6572236 224.71
+ x1    1    301038  6271198 225.91
+ x4    1     17706  6554530 226.66
- x2    1   7188838 13761075 235.27
- x3    1  46228356 52800592 258.13

Step:  AIC=221.77
y ~ x3 + x2 + x5

       Df Sum of Sq      RSS    AIC
<none>               4916786 221.77
+ x4    1    365674  4551112 222.46
+ x1    1    234549  4682237 222.94
- x5    1   1655450  6572236 224.71
- x2    1   2630172  7546958 227.06
- x3    1  32736768 37653554 254.38

Call:
lm(formula = y ~ x3 + x2 + x5, data = mydata)

Coefficients:

(Intercept)           x3           x2           x5

 1520.75917      0.97850      0.05298   -320.61526
```

جهت اجرای روش Backward از دستور زیر استفاده می شود

```
>step(full,data=mydata,method="backward")


Start:  AIC=224.42
y ~ x1 + x2 + x3 + x4 + x5

       Df Sum of Sq     RSS    AIC
- x1    1     10760 4551112 222.46
- x3    1    108600 4648952 222.82
- x4    1    141884 4682237 222.94
```

```
<none>              4540352 224.42
- x5    1   1455008 5995360 227.15
- x2    1   2854572 7394924 230.71

Step:  AIC=222.46
y ~ x2 + x3 + x4 + x5

       Df Sum of Sq      RSS    AIC
- x4    1    365674  4916786 221.77
<none>               4551112 222.46
- x5    1   2003418  6554530 226.66
- x2    1   2875093  7426205 228.78
- x3    1  19062869 23613981 248.45

Step:  AIC=221.77
y ~ x2 + x3 + x5

       Df Sum of Sq      RSS    AIC
<none>               4916786 221.77
- x5    1   1655450  6572236 224.71
- x2    1   2630172  7546958 227.06
- x3    1  32736768 37653554 254.38

Call:
lm(formula = y ~ x2 + x3 + x5, data = mydata)

Coefficients:
(Intercept)           x2           x3           x5
 1520.75917      0.05298      0.97850   -320.61526
```

و جهت اجرای روش Stepwise از دستور زیر استفاده می شود

```
>step(null,scope=list(upper=full),method="both")
```

در صورت عدم اجرای دستورات فوق می توانید ابتدا `library(leaps)` را نصب و اجرا کنید.

۶. شناسایی مشاهدات موثر، پرنفوذ و پرت

مقادیر موثر در مدل رگرسیونی مقادیری هستند که معمولا دارای مقادیری غیر معمول (بزرگتر یا کوچکتر) نسبت به سایر مقادیر مشابه باشند به طوری که حضور آنها و یا حذف آنها در برازش مدل رگرسیونی موثر باشند. این مقادیر معمولا به دو گروه مشاهدات پرنفوذ و داده های پرت تقسیم بندی می شوند . البته ذکر این نکته ضروری است که یک مشاهده ی پرنفوذ و یا یک داده ی پرت الزاما نمی تواند مقداری موثر در مدل باشد. هدف از این قسمت شناسایی مشاهدات پرنفوذ موثر و داده های پرت موثر می باشند.

الف) مشاهدات پرنفوذ (High leverage)

مشاهده ی پرنفوذ x_i ، مقداری از متغیر مستقل X می باشد که به صورتی غیر معمول از بقیه ی مشاهدات بزرگتر یا کوچکتر باشد.

مشاهدات پرنفوذ می توانند بالقوه (و نه الزاما) بر روی برآورد پارامترهای مدل موثر باشند. جهت شناسایی مشاهدات پرنفوذ موثر چندین روش وجود دارد که ابتدا آنها را به اختصار توضیح داده و سپس روش محاسبه ی آنه با استفاده از نرم افزار R ارائه خواهد شد.

در رگرسیون خطی چندگانه ماتریس $H = X(X'X)^{-1}X'$ را اصلاحا hat-matrix گویند که کاربردهای متعددی دارد. به عنوان مثال در محاسبه ی ماتریس واریانس باقیمانده های مدل داریم

$$Var(\hat{\varepsilon}) = (I - H)\sigma^2$$

بنابراین $Var(\hat{\varepsilon}_i) = (1-h_i)\sigma^2$ که در آن $h_i = H_{ii}$ را میزان نفوذ(leverage) گویند و برابر i –امین عنصر قطر اصلی ماتریس H می باشد. با توجه به رابطه ی فوق هر چه میزان نفوذ یک متغیر بیشتر باشد مقدار واریانس خطای مدل به ازای آن متغیر کمتر خواهد بود.

اگر P تعداد پارامترهای مدل باشد آنگاه $\sum_i h_i = P$ می باشد و لذا مقدار متوسط برای هر h_i برابر $\frac{P}{n}$ می باشد که در آن n تعداد مشاهدات می باشد. بر اساس یک قاعده ی سرانگشتی هر مشاهده با مقدار h_i بزرگتر از $\frac{2P}{n}$ ، را باید یک مشاهده پرنفوذ موثردر نظر گرفت.

جهت شناسایی مشاهدات پرنفوذ مدل رگرسیونی که در مثالهای قبل مورد استفاده قرار گرفت ، با استفاده از مطالب فوق به صورت زیر عمل می کنیم.

ابتدا hat-matrix را بدست می آوریم ،

```
>inf=influence(fit)
>lev=inf$hat
```

```
>lev
1           2           3           4           5           6
7
0.14114786 0.26598478 0.18814680 0.17438235 0.10823131
0.19004671 0.11632854
          8           9          10          11          12
13          14
0.53350328 0.20465203 0.83079237 0.08672868 0.26228952
0.23922654 0.15427172
         15          16          17
0.79892875 0.83210885 0.87322990
```

در خروجی فوق میزان نفوذ هر متغیر y ، در زیر آن نوشته شده است. اکنون بایستی متوسط میزان نفوذ و مقادیر بزرگتر از آن را بدست آوریم. بنابراین خواهیم داشت

```
 > 2*sum(lev)/length(y)
 0.7058824
```

که در آن P=sum(lev) و n=length(y) . بنابراین هر مشاهده با مقدار lev>0.7058824 می تواند یک مشاهده ی پرنفوذ باشد. جهت شناسایی این مشاهدات بر اساس متغیر وابسته خواهیم داشت

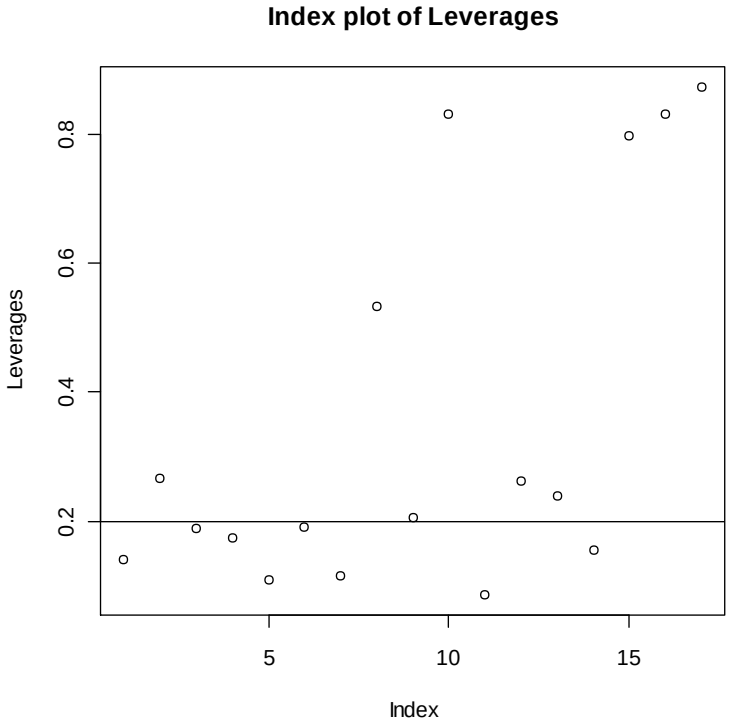
```
>names(lev)=y
> lev[lev>.7]

   10        15        16        17
0.8307924 0.7989287 0.8321088 0.8732299
```

جهت شناسایی مشاهدات پرنفوذ می توان از نمودار نیز کمک گرفت . جهت انجام این کار نمودار مقادیرنفوذ را به صورت زیر رسم می کنیم.

```
> plot(lev,ylab="Leverages",main="Index plot of Leverages")

> abline(h=2*5/50)
```

همانطور که در نمودار فوق مشخص می باشد مشاهدات دهم ، پانزدهم ، شانزدهم و هفدهم مقادیری پرنفوذ می باشند. جهت شناسایی این نقاط بر روی نمودار می توان پس از اجرای دستور زیر، بر روی نقاط درون نمودار با استفاده از موس کلیک نمایید.

```
> identify(1:17,lev,y)
```



ب) داده های پرت (Outliers)

داده های پرت ، مقادیری از متغیر پاسخ (وابسته) می باشند که توسط مدل به خوبی پیش بینی نمی شوند به عبارت دیگر مقدار باقیمانده (خطای) مدل به ازای آنها مقداری بزرگ باشد. وجود یک مشاهده ی پرت در مدل می تواند باعث افزایش مقدار واریانس باقیمانده های مدل شود و مقدار ضریب تعیین مدل را تحت تاثیر قرار دهد. جهت شناسایی این مقادیر چندین روش وجود دارد که در زیر به برخی از آنها اشاره می شود و روش محاسبه ی آنها با استفاده از نرم افزار R نیز ارائه خواهد شد.

یکی از روشهای شناسایی مشاهدات پرت استفاده از باقیمانده های استیودنت شده (Studentized residuals) می باشد که از رابطه ی زیر محاسبه می شود.

$$r_i = \frac{e_i}{s\sqrt{1-h_{ii}}}$$

که در آن e_i برابر i –امین مقدار باقیمانده و s انحراف استاندارد باقیمانده ها می باشد. مشاهده ای که مقدار باقیمانده ی استیودنت شده ی آن به طور غیر معمول بزرگتر از بقیه ی مقادیر باشد می تواند مشاهده ای پرت باشد.

در برخی از متون جهت شناسایی مشاهده ی پرت از باقیمانده ی پرس (PRESS residuals) نیز استفاده می شود که با توجه به اینکه بین هر دو باقیمانده رابطه ای یک به یک برقراراست لذا نتایج تحلیل با هر دو روش یکسان می باشد.

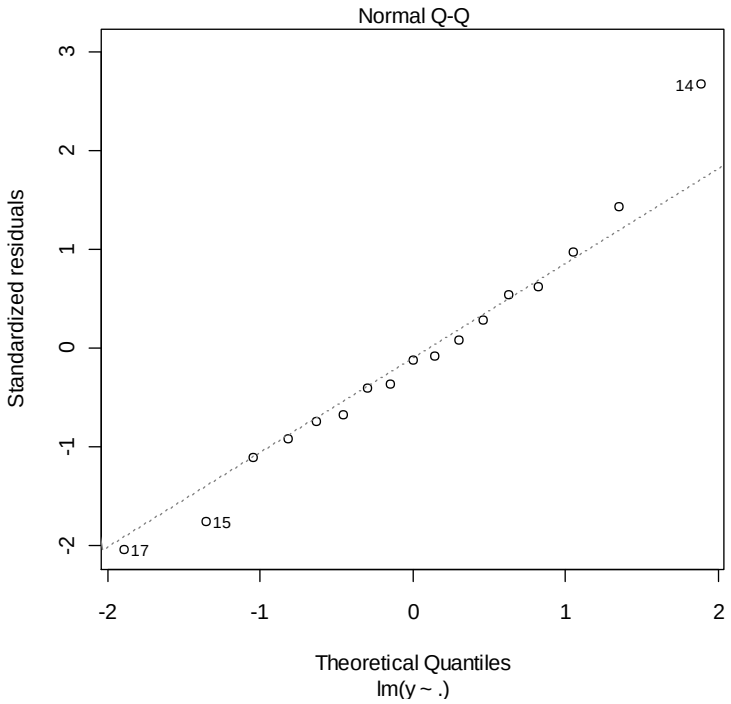
جهت شناسایی مشاهده ی پرت با استفاده از این روش با استفاده از R داریم،

```
>rstudent(fit)

          1           2           3           4           5
6
-0.34985603 -0.07531590 -0.11412902  0.60643916  0.07523142 -
0.92150669
          7           8           9          10          11
12
 0.26182776  0.96532898 -0.73632547 -0.65835546  0.52484185 -
1.12014794
         13          14          15          16          17
-0.39431558  4.28367512 -1.97199429  1.51139781 -2.46596748
```

با توجه به مشاهدات فوق ، مقدار باقیمانده ی مشاهده ی ۱۴، ۱۵ و ۱۷ مقادیری بزرگتر از بقیه را دارد که می توانند مشاهداتی پرت باشند.

دستور زیر نمودار qq-plot باقیمانده های استیودنت شده می باشد که می تواند به عنوان روش دیگری جهت شناسایی مشاهدات پرت مورد استفاده قرار گیرد.

```
>plot(fit,which=2)
```



همانطور که در نمودار فوق نشان داده شده ، مشاهدات ۱۴، ۱۵ و ۱۷ می توانند مشاهدات پرت باشند.

دستور زیر آزمونی جهت شناسایی داده های پرت ارائه می نماید که می تواند از دو روش فوق دقیق تر باشد.

```
>library(car)
> outlier.test(fit)

max|rstudent| = 4.283675, degrees of freedom = 10,
unadjusted p = 0.001602088, Bonferroni p = 0.0272355

Observation: 14
```

همانطور که مشاهده می شود از میان متغیرهای ۱۴، ۱۵ و ۱۷ که بالقوه مشاهداتی پرت بودند ، فقط مشاهده ی ۱۴ با استفاده از آزمون به عنوان مشاهده ای پرت در نظر گرفته شده است.

ج) مشاهدات موثر

همانطور که قبلا بیان شد مشاهده ی موثر می تواند یک مشاهده ی پرنفوذ و یا پرت باشد که به طور کلی روند برازش مدل رگرسیونی را تحت تاثیر قرار می دهد. جهت شناسایی این مشاهدات روشهای متعددی شامل DFFITS, DFBETAS و Cook's D وجود دارند که به صورت زیر تعریف می شوند.

DFBETAS : عبارتست از میزان تاثیر مشاهده ی i-ام بر روی برآورد ضرایب رگرسیونی و مقدار آن برای مشاهده ی i –ام عبارتست از ضریب مدل رگرسیونی هر گاه در برآورد ضریب مذکور مشاهده ی i –ام حضور نداشته باشد . به عبارت دیگر مشاهده ی مذکور به هنگام برازش مدل از داده ها حذف شده باشد و به ازای ضرایب مختلف قابل محاسبه می باشد. معمولا مشاهده ای با مقدار $DFBETAS > \frac{2}{\sqrt{n}}$ ، مشاهده ای موثر در نظر گرفته می شود.

DFFITS : عبارتست از میزان تاثیر مشاهده ی i –ام بر روی مقادیر برازش شده و تابعی است از $\hat{y}_i - \hat{y}_{i,-i}$ که در آن $\hat{y}_i$ عبارتست از i-امین مقدار برازش شده مدل رگرسیونی با حضور مشاهده ی i-ام و $\hat{y}_{i,-i}$ عبارتست از امین مقدار برازش شده مدلی که مشاهده ی i-ام در برآورد پارامترهای آن حضور نداشته باشد و هر مشاهده با $DFFITS > 2\sqrt{p/n}$ ، یک مشاهده ی موثر محسوب می شود.

Cook's Distance : عبارتست از فاصله ی استاندارد میان بردار ضرایب رگرسیونی b و بردار b_{-i} که در آن b_{-i} عبارتست از بردار ضرایب رگرسیونی که در برآورد آنها مشاهده ی i-ام حضور نداشته باشد. مشاهداتی با $Cook's\ D > 1$ می توانند بالقوه مشاهداتی موثر(داده ی پرت یا پرنفوذ) باشند .

جهت محاسبه ی DFBETAS با استفاده از R به صورت زیر عمل می کنیم.

```
> dfbetas(fit)

     (Intercept)     x1            x2           x3           x4
x5
1  -0.082063609  0.017908927  0.0198893806 -0.020314583
0.025491191   0.055577990
2   0.025560979  0.017544486 -0.0077283426 -0.016774372 -
0.012040057  -0.032514750
3  -0.042021801 -0.006162631  0.0037596763  0.004802863
0.026315684   0.033506240
4   0.163978230 -0.050789105 -0.0094264900  0.053937557 -
0.014626129  -0.119417653
5  -0.001766098 -0.011810163  0.0007033375  0.011709777
0.005808435   0.007261281
6  -0.268602044 -0.068750497 -0.1500072711  0.062068597
0.251424002   0.202923211
7  -0.017999957 -0.049954560 -0.0032564695  0.049207081
0.035561726   0.036836262
8   0.524404622  0.948011567 -0.0233320402 -0.944950526 -
```

```
0.664536694  -0.507835911
9   0.188915653  0.255793967  0.0536295953 -0.249596944 -
0.267067873  -0.239794817
10  0.617708115 -0.471382777 -0.1826295291  0.532229053 -
0.615796996  -0.591011838
11 -0.013425473 -0.038049991  0.0503902355  0.037045809 -
0.001850034   0.044342335
12 -0.524049385 -0.378553118  0.3444534587  0.362425526
0.285563768   0.502968971
13  0.075007795  0.155955170  0.0411580311 -0.152411628 -
0.183958714  -0.096760287
14 -0.460338271 -0.107721213  1.1174815197  0.091808336 -
0.222511581   0.562798553
15  0.858968538 -0.805859921  0.0916084540  0.755756249
1.482362573  -0.853747929
16  0.799124200 -0.454415310 -2.5562895626  0.535541669
0.789242026  -0.948730591
17 -0.204231960  0.231932001 -3.9453317814 -0.197585738
0.322400825   0.496369405
```

مقادیر DFFITS به صورت زیر محاسبه می شود،

```
>library(stats)
> dffits(fit)

   1        2         3           4        5           6
  -0.141  -0.0453  -0.0549   0.278         0.0262    -0.446

  7         8          9        10       11       12
 0.0949  1.032      -0.373  -1.458      0.161   -0.667

 13          14          15          16          17
      -0.221       1.829      -3.930        3.364     -6.472
```

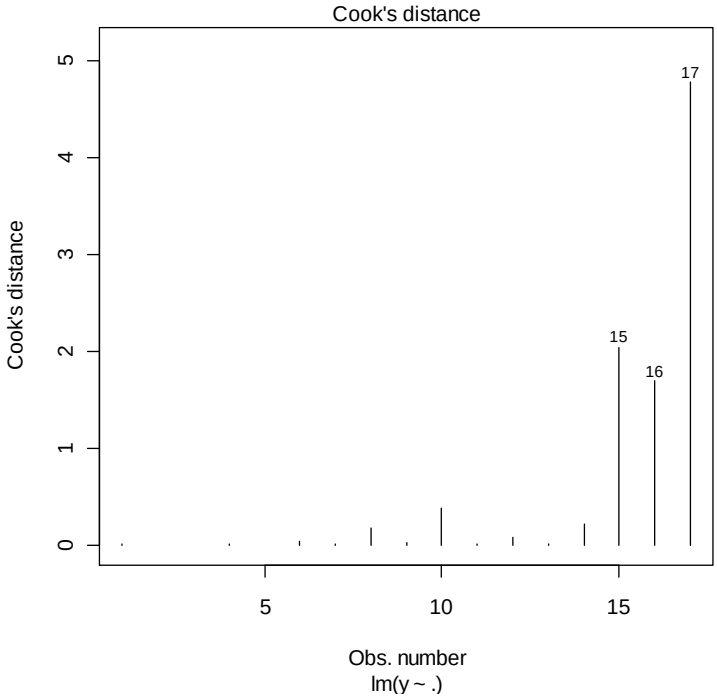
و همچنین مقادیر Cook’s D نیز با دستور زیر قابل محاسبه می باشد.

```
>cook=cooks.distance(fit)
> cook

     1       2          3         4         5       6
 0.0036  0.000376    0.00055   0.01373   0.000125   0.0336
```

```
      7         8         9        10        11        12
0.00164   0.178    0.0242    0.3739   0.00466    0.0726

     13        14        15        16        17
0.00882    0.2164   2.03961    1.6896    4.775
```

جهت شناسایی مشاهدات موثر با استفاده از روش Cook's D می توان از نمودار آن نیز کمک گرفت که از دستور زیر بدست می آید.

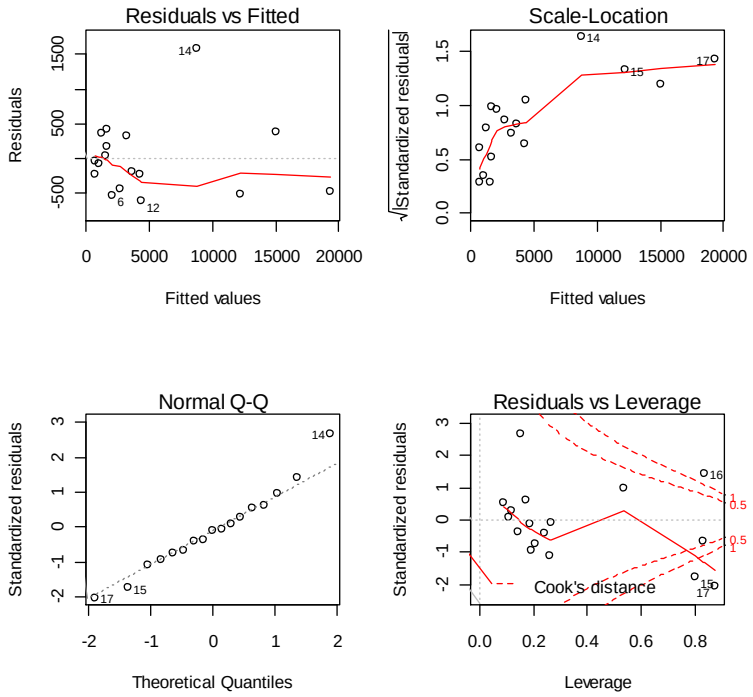
```
>plot(fit,which=4)
```



۷ .چهار نمودار مفید جهت ارزیابی مدل

جهت ارزیابی مدل رگرسیونی به کمک نمودار، می توان از نمودارهای مختلفی کمک گرفت اما اهمیت برخی از آنها از بقیه بیشتر است. در نرم افزارR امکان ارائه همزمان چهار نمودار که در ارزیابی مدل رگرسیونی از اهمیت بیشتری برخوردارند وجود دارد که با دستور زیر بدست می آید.

```
>layout(matrix(1:4),2,2)
>plot(fit)
```



نمودار بالا سمت چپ ، نمودار مقادیر برازش شده در مقابل باقیمانده ها می باشد که از آن می توان جهت بررسی ثابت (همگن) بودن واریانس و خطی بودن مدل رگرسیونی (در رگرسیونی خطی ساده) کمک گرفت. همچنین داده های پرت نیز با کمک این نمودار قابل تشخیص می باشند. مثلا در نمودار فوق چون نقاط روند خاصی را نشان نمی دهند لذا واریانس ثابت است اما همانطور که نشان داده شده مشاهده ی ۱۴، یک داده ی پرت می باشد.

نمودار بالا سمت راست مربوط به مقادیر برازش شده در مقابل جذر مانده های استاندار شده می باشد که همانند نمودار قبل بایستی روند خاصی در آن مشاهده نشود. همچنین این نمودارمی تواند جهت شناسایی مشاهدات موثر ، همانطور که در شکل مشخص شده، مورد استفاده قرار گیرد.

نمودار پایین سمت چپ ، نمودار qq-plot مانده های استاندارد شده می باشد که در مباحث قبل به آن اشاره شد.

نمودار پایین سمت راست ، نموداری است که از آن جهت شناسایی مشاهدات موثر و پرنفوذ استفاده می شود. همانطور که در شکل مشاهده شده مشاهدات ۱۵، ۱۶ و ۱۷ می توانند به عنوان مشاهدات موثر تشخیص داده شده اند که ممکن است مشاهداتی پرنفوذ یا داده های پرت باشند.

فصل ۱۰

برازش توزیع های آماری بر داده ها

۱.مقدمه

هدف این فصل برازش توزیعهای آماری مشخص بر داده های موجود می باشد. جهت برازش دلهای آماری بر داد ها، ابتدا می بایست نوع داده ها را مشخص نمود. مثلا برای داده های شمارشی توزیع هایی مانند پواسن ، دوجمله ای ، دوجمله ای منفی و یا هندسی می تواند مفید باشد. یا اگر داده ها از نوع متغیرهای پیوسته هستند توزیع هایی مانند نرمال ، گاما ، نمایی و یا هر توزیع پیوسته دیگری ممکن است مفید باشد. البته به هنگام برازش یک توزیع بر داده های موجود بایستی به این نکته نیز توجه نمود که ماهیت داده ها با دامنه توزیع همسان باشد. مثلا اگر در داده ها ، اعدا منفی نیز وجود داشته باشند ، نمی توان از توزیع هایی مانند گاما یا نمایی استفاده نمود چون دامنه متغیرهای این توزیع ها فقط اعداد مثبت را شامل می شوند.

از طرف دیگر برای یک گروه از اعداد، ممکن است چندین توزیع مناسب جهت برازش وجود داشته باشد. مثلا متغیرهای توزیع هایی مانند پواسن ، دوجمله ای منفی و هندسی دارای دامنه یکسان هستند و بنابراین جهت برازش داده های شمارشی که دامنه صفر تا بینهایت هستند می توان از این سه توزیع استفاده نمود.

پس از برازش مدلهای مناسب بر داده ها ، بایستی توجه نمود که کدام یک از این توزیع ها در برازش عملکرد مناسب تری داشته است. با توجه به اینکه ، شیوه برازش مدل بر اساس برآورد پارامترها از روش درستنمایی ماکزیمم می باشد، بنابراین مدلی عملکرد مناسب تری دارد که مقدار عددی لگاریتم تابع درستنمایی آن ، یعنی $LogLikelihood$ آن از بقیه بیشتر باشد. اما همانطور که می دانیم یکی از نکات مهم در برازش یک مدل تعداد پارامترهای می باشد که بایستی به عنوان یک عامل مهم در نظر گرفت. به عبارت دیگر در مقایسه دو مدل برازش شده ، ممکن است مقدار $LogLikelihood$ یک تابع بیشتر باشد ، اما تعداد پارامترهای آن نیز به حدی زیاد باشد که عملا ، در برازش مدل عملکرد قابل قبولی نداشته باشد.

جهت در نظر گرفتن این عامل در مقایسه مدلها ، می توان از آماره ی AIC (Akaike Information Criterion) استفاده نمود. مقدار آماره ی AIC از رابطه ی زیر محاسبه می شود.

$$AIC=-2\text{loglikelihood}+2p$$

که در رابطه ی فوق p برابر تعداد پارامترهای مدل می باشد. بنابراین ، با توجه به رابطه ی فوق ، به ازای افزایش هر پارامتر ، تعداد دو واحد از AIC کاسته می شود و اینکه هر مدل با AIC کمتر ، مدل بهتری می باشد.

۲. برازش توزیع های پیوسته و گسسته بر داده ها

جهت برازش مدلهای مختلف ، می توان از بسته نرم افزاری {MASS} استفاده نمود. جهت کسب اطلاعات بیشتر ابتدا دستور زیر را اجرا نمایید.

```
> library(MASS)
> ?fitdistr
```

پس از اجرای دستور فوق ، صفحه راهنما ی باز خواهد شد که در آن توزیع هایی مختلف آماری که در این بسته نرم افزاری در نظر گرفته شده است وجود دارد. این توزیع ها شامل "beta", "cauchy", "chi-squared", "exponential", "f", "gamma", "geometric", "log-normal", "lognormal", "logistic", "negative binomial", "normal", "Poisson", "t" and "weibull" می باشد.

در این صفحه همچنین توضیحاتی در باره برازش مدلهای مختلف داده شده است. دستور برازش مدل ها و آرگومانهای مربوطه به صورت زیر است.

```
>fitdistr(x, densfun, start, ...)
```

در دستور فوق ، بردار x یک بردار عددی از داده هاست که مدل مذکور بایستی بر آن برازش داده شود. ارگومان بعدی ، نام توزیع مورد نظر که بایستی بر داده ها برازش داده شود می باشد که در بالا داده شده است. آرگومان start ، برای برخی از توزیع هایی که برآورد پارامترها در آن با استفاده از روش درستنمایی ماکزیمم دارای یک فرم بسته نمی باشد. در این حالت برازش مدل با استفاده از روشهای عددی انجام می شود ، بنابراین جهت برازش مدل بایستی یک مقدار اولیه مناسب بر اساس داده ها (مثلا برآورد گشتاوری پارامترها) را پیشنهاد داد.

مثال ۱. داده های جدول زیر مربوط به تعداد تصادفات رانندگی در کشور بلژیک بین سالهای ۱۹۵۸-۱۹۵۶ گرفته شده است که در کتاب "Loss Modells From Data to Decision" توسط Klugmaan(2008) داده شده است.

فراوانی	تعداد تصادفات هر خودرو
۸۱۷۱۴	۰
۱۱۳۰۶	۱
۱۶۱۸	۲
۲۵۰	۳
۴۰	۴
۷	۵

همانطور که در جدول فوق مشاهده می شود ، با توجه به شمارشی بودن داده ها ، توزیع هایی مانند پواسن ، هندسی و دوجمله ای منفی مناسب می باشند هر کدام از این توزیع را بر داده ها برازش می دهیم و مقدار AIC را بدست می آوریم..

```
>x=rep(c(0,1,2,3,4,5),c(81714,11306,1618,250,40,7))

####fitting Poisson Distribution#####

>poi=fitdistr(x,"poisson")
> poi$estimate
    lambda
    0.163
> poiAIC=-2*logLik(poi)+2
> poiAIC
  'log Lik.' 90598 (df=1)


####fitting Geometric Distribution####

> geo=fitdistr(x,"geometric")
> geo$estimate
   prob
   0.86
> geoAIC=-2*logLik(geo)+2
> geoAIC
  'log Lik.' 55324 (df=1)

####fitting Negative Binomial Distribution####

> nbinom=fitdistr(x,"negative binomial")
> nbinom$estimate
   size    mu
```

```
  0.893 0.163
> nbinomAIC=-2*logLik(nbinom)+4
> nbinomAIC
  'log Lik.' 89533 (df=2)
```

با توجه به خروجی فوق و مقایسه ی مقادیر AIC ، توزیع هندسی در برازش مدل عملکرد بهتری نسبت به پواسن و دوجمله ای منفی داشته است.

مثال ۲. جهت مدل بندی داده های بقا معمولا از توزیع هایی مانند گاما ، وایبل و لوگ نرمال استفاده می شود. داده های زیر را در نظر بگیرید. برای این داده ها ، توزیع های زیر را برازش می دهیم.

```
x=c(620,470,260,89,388,242,
+     103,100,39,460,284,1285,
+     218,393,106,158,152,477,403,
+     103,69,158,818,947,399,1274,32,12,134,660,
+     548,381,203,871,193,531,317,85,1410,250,41,1101,
+     32,421,32,343,376,1512,1792,47,95,76,515,72,1585,
+     253,6,860,89,1055,537,101,385,176,11,565,164,16,
+     1267,352,160,195,1279,356,751,500,803,560,151,24,
+     689,1119,1733,2194,763,555,14,776,1)
```

```
#### fitting gamma distribution ####

> gam=fitdistr(x,"gamma")
> gam
      shape          rate
  0.860677111   0.001817632
 (0.099178410) (0.000232688)
> gamAIC=-2*logLik(gam)+4
```

```
> gamAIC
'log Lik.' 1277.125 (df=2)

#### fitting Weibull distribution ####

> weib=fitdistr(x,"weibull")
> weib
      shape           scale
    0.92200225   464.68631144
 ( 0.07792558) ( 56.76754277)
> weibAIC=-2*logLik(weib)+4
> weibAIC
'log Lik.' 1277.481 (df=2)



#### fitting lognormal distribution ####

> logn=fitdistr(x,"lognormal")
> logn
    meanlog      sdlog
  5.4770930   1.4333311
 (0.1519328) (0.1074327)
> lognAIC=-2*logLik(logn)+4
> lognAIC
'log Lik.' 1295.574 (df=2)
```

با توجه به خروجی فوق و مقایسه ی مقادیر مقادیر AIC ، عملکرد توزیع گاما در برازش از بقیه بهتر می باشد.

۳. برازش توزیع نرمال یک متغیره و چند متغیره بر داده ها

جهت برازش توزیع نرمال چند متغیره که نرمال یک تغیره نیز می تواند به عنوان حالت خاصی از آن در نظر گرفته شود ، می توان از بسته نرم افزاری که برای همین منظور در R وجود دارد استفاده نمود. این بسته نرم افزاری {mclust} نام دارد که دستور مربوط به برازش توزیع نرمال {mvn} نام دارد.

آرگومان مربوط به این دستور به صورتهای زیر می باشد.

```
>mvnX(data, prior = NULL, warn = NULL, ...)
>mvnXII(data, prior = NULL, warn = NULL, ...)
>mvnXXI(data, prior = NULL, warn = NULL, ...)
```

```
>mvnXXX(data, prior = NULL, warn = NULL, ...)
```

در دستورات فوق ، mvnX جهت برازش نرمال یک متغیره ، mvnXII جهت برازش نرمال چند متغیره هر گاه ماتریس کوواریانس به عوان ضریبی از ماتریس همانی در نظر گرفته شود. mvnXXI برای برازش نرمال چند متغیره هر گاه ماتریس کواریانس به عنوان یک ماتریس قطری در نظر گرفته شود و از دستور mnvXXX جهت برازش نرمال چند متغیره هر گاه محدودیتی روی ماتریس کوواریانس نباشد استفاده می شود.

مثال ۳. در این مثال ابتدا مقادیر عددی از توزیع نرمال یک متغیره ، به صورت تصادفی تولید می کنیم و سپس مدل نرمال یک متغیره را بر آن برازش خواهیم داد.

```
>library(mclust)
>x=rnorm(1000,mean=-5,sd=2)
> mvnX(x)
$modelName
[1] "X"

$parameters$mean
[1] -5.024984


$parameters$variance$sigmasq
[1] 4.184613



$loglik
[1] -2134.646
```

در مثال فوق ابتدا تعداد ۱۰۰۰ عدد تصادفی از توزیع نرمال با میانگین ۵- و انحراف معیار ۲ تولید شد و سپس توزیع نرمال یک متغیره بر آن برازش داد شد که برآورد پارامترها برای میانگین و واریانس به ترتیب ، ۵/۰۲- و ۴/۱۸ می باشد.

مثال۴. در این مثال ابتدا یک بردار تصادفی دیگر از توزیع نرمال متشکل از ۱۰۰۰ عدد با میانگین ۲ و انحراف معیار ۲ تولید می کنیم و سپس توزیع نرمال دو متغیره را برای داده های این مثال و داده های تولید شده در مثال قبل برازش خواهیم داد.

خروجی زیر را خواهیم داشت ،

```
> library(mclust)
```

```
> x=rnorm(1000,mean=-5,sd=2)
> y=rnorm(1000,mean=2,sd=2)
> z=data.frame(x,y)

> mvnXXX(z)
$modelName
[1] "XXX"
$parameters$mean
          [,1]
[1,] -5.024984
[2,]  1.979237



$parameters$variance$Sigma
            [,1]        [,2]
[1,]  4.18461320 -0.02380429
[2,] -0.02380429  4.06708623



$loglik
[1] -4255.031

attr(,"returnCode")
[1] 0
>
```

در خروجی فوق برآورد میانگین و واریانس متغیر X به ترتیب برابر ۵/۰۲- و ۴/۱۸ و این مقادیر برای متغیر Y برابر ۱/۹۷ و ۴/۰۶ می باشد. همچنین مقدار کوواریانس میان دو متغیر برابر ۰/۰۲۳- برآورد شده است.