



آشنایی با نرم افزار R
درس آمار حیاتی و احتمالات
مدرس بخش نرم افزار : پویا تقی پور لنگرودی

در این ارائه خواهیم دید :

مدیریت کاراکتر

Pattern matching

مدیریت تاریخی

ساختارهای کنترلی

ساختارهای تکرار



مدیریت data frame

در این بخش بر اساس چند روش مرتب سازی و مدیریت آنها انجام می شود. برای مثال می خواهیم بر اساس ستون رنگ، داده های خود را مرتب کنیم. بنابراین از تابع order استفاده می کنیم.

```
# Sort data frame based on color
ord <- order(painters$Colour)
ord
painters.color <- painters[ord, ]
View(painters.color)
painters.color <- painters[rev(ord), ]
View(painters.color)
painters.color[c(1, 2), ]
```

تابع order پس از sort کردن ستون رنگ، بجای اینکه مقدار برگرداند index مربوط به سطر را نمایش می دهد.

بنابراین میتوان با توجه به خروجی تابع order کل data frame را بر اساس ستون رنگ مرتب نمود چنانچه دو data frame داشته باشیم، میتوانیم به کمک تابع merge آنها را باهم ادغام کنیم. این ادغام بر اساس يك خصوصیت مشترك در هر دو data frame صورت میگیرد.

مدیریت
کاراکتر

R

Pattern
matching

مدیریت
تاریخی

ساختارهای
کنترلی

ساختارهای
تکرار

توابع پایه مدیریت کاراکتر

R عمدتاً یک نرم افزار برای انجام محاسبات عددی بشمار می رود. با این وجود R حاوی توابعی برای مدیریت داده هایی از جنس کاراکتر هست.

توابع `tolower`, `toupper`, `nchar` و `noquote` توابع پایه برای کار با نوع کاراکتر در R است. یکی از قابلیت های این توابع، `vectorized operations` است.

تابع `toupper` جهت تبدیل حروف کوچک به حروف بزرگ و تابع `tolower` برعکس آن یعنی حروف بزرگ را به کوچک تبدیل می کند.

جهت ترکیب مقادیر از راه های مختلفی استفاده می شود:

- جهت ترکیب مقادیر از تابع `cat` استفاده می شود.
- تابع بعدی جهت ترکیب مقادیر تابع `paste` است.
- تابع دیگر با همین کاربرد تابع `sprintf` است.

در R نمی توان بخشی از یک رشته کاراکتری (`character strings`) را بوسیله `subscripting` استخراج نمود. برای استخراج بخشی از یک رشته کاراکتری از تابع `substring` استفاده می شود.

مدیریت
کاراکتر

R

Pattern
matching

مدیریت
تاریخی

ساختارهای
کنترلی

ساختارهای
تکرار

بخشی از مدیریت داده های آماری در R شامل پردازش رشته های کاراکتری (character strings) می باشد. هدف از pattern matching یافتن رشته های کاراکتری است که با یک الگو خاص انطباق دارند. دو تابع grep و grepl در این بخش مورد بررسی قرار میگیرند. grep که مخفف global regular expression print است، یک بردار را بر اساس یک pattern خاص جست و جو میکند. تابع grepl مانند تابع grep است با این تفاوت که خروجی آن از نوع logical میباشد.

- `^` Start of line anchor \Rightarrow `^Hello`
- `$` End of line anchor \Rightarrow `World$`
- `.` Any single character \Rightarrow `a.e`
- `|` Alternation operator \Rightarrow `the|this`
- `[]` Character set, any one character listed inside `[]` \Rightarrow `[hc]at`
- `[^]` Negative character set, any one character not listed inside `[]` \Rightarrow `[^aeiou]`
- `-` Range indicator \Rightarrow `[A-Z]`
- `*` The preceding item can occur zero or more times \Rightarrow `file1*`
- `+` The preceding item can occur one or more times \Rightarrow `gro+ve`
- `?` The preceding item can occur zero or one times \Rightarrow `colou?r`
- `{n}` The preceding item is matched exactly n times \Rightarrow `Moham{2}ad`
- `{n, }` The preceding item is matched n or more times \Rightarrow `[a-z]{4, }`
- `{n, m}` The preceding item is matched at least n times, but not more than m times \Rightarrow `[0-9]{4, 6}`
- `\` Escape the next character \Rightarrow `*\ .txt`

- $\backslash d$ Any digit : $[0-9]$
- $\backslash D$ Anything not a digit : $[^0-9]$
- $\backslash w$ Any word character : $[A-Za-z_0-9]$
- $\backslash W$ Anything not a word character : $[^\backslash w]$
- $\backslash s$ White space : $[\text{space}, \backslash t, \backslash n, \backslash v, \backslash f, \backslash r]$
- $\backslash S$ Anything not whitespace : $[^\backslash s]$

تابع `as.Date` رشته کاراکتری با فرمت سال، ماه و روز را که با `dash` و یا `slash` جدا شده باشد به عنوان ورودی دریافت و آن را به تاریخ تبدیل می کند. چنانچه فرمت تاریخ ورودی متفاوت از فرمت پیش فرض باشد، جهت بیان فرمت تاریخ ورودی از آرگومان `format` تابع `as.Date` استفاده می شود. در این آرگومان از گروهی از نمادها که با `%` شروع می شود استفاده می گردد. تاریخ در R به صورت تعداد روزهای از اول ژانویه سال ۱۹۷۰ ذخیره می شود.

با استفاده از سه تابع months، weekdays و quarters می توان روز هفته، ماه و فصل تاریخ مورد نظر را استخراج نمود

جهت مدیریت تاریخ به همراه زمان از دو کلاس POSIX (Portable Operating System Interface, primary for UNIX) که شامل POSIXct مخفف continuous time و POSIXlt مخفف list time استفاده می شود. کلاس POSIXct تاریخ و زمان را بر حسب ثانیه از اول ژانویه سال ۱۹۷۰ ذخیره می کند. کلاس POSIXlt تاریخ و زمان را به صورت لیستی از اجزا زمان ذخیره می کند. فرمت پیش فرض برای این تابع برای تاریخ مشابه تابع gas.Date و برای زمان به صورت ثانیه:دقیقه:ساعت است

در صورتیکه تاریخ و زمان ورودی دارای فرمت دیگری باشد از تابع strptime استفاده می شود. تابع strptime رشته کاراکتری تاریخ و زمان و فرمت ورودی را دریافت و متغیری از کلاس POSIXlt بر می گرداند.

مدیریت
کاراکتر

Pattern
matching

مدیریت
تاریخی

ساختار
های
کنترلی

ساختارهای
تکرار

برخی اوقات جریان اجرای کدها میبایست بر ایای شرایط و معیارها کنترل شود. ساختارهای کنترلی (flow of control) به صورت مدیریت نحوه اجرای کدها تعریف میگردد که به صورت کلی شامل ساختارهای تصمیم یا شرطی (conditional statements) و تکرار (loops) میشود. ساختارهای شرطی بر مبنای عملگرهای مقایسه ای (relational operators) و منطقی (logical operators) طراحی میشوند.

شرط

ساده ترین شکل ساختار شرطی بررسی يك شرط با استفاده از if است. حالت صحیح کد با دستور if به این صورت است. شرط ما در مقابل if و دستوری که در صورت درست بودن شرط اجرا می شود داخل آکولاد قرار میگیرد. در صورت true بودن شرط بلوک if اجرا خواهد شد. بر همین اساس می توان با استفاده از ساختار if else در بخش else تعیین نمود که در صورت false بودن شرط چه رویدادی رخ دهد.

مدیریت
کاراکتر

Pattern
matching

مدیریت
تاریخی

ساختار
های
کنترلی

ساختارهای
تکرار

مدیریت
کاراکتر

Pattern
matching

مدیریت
تاریخی

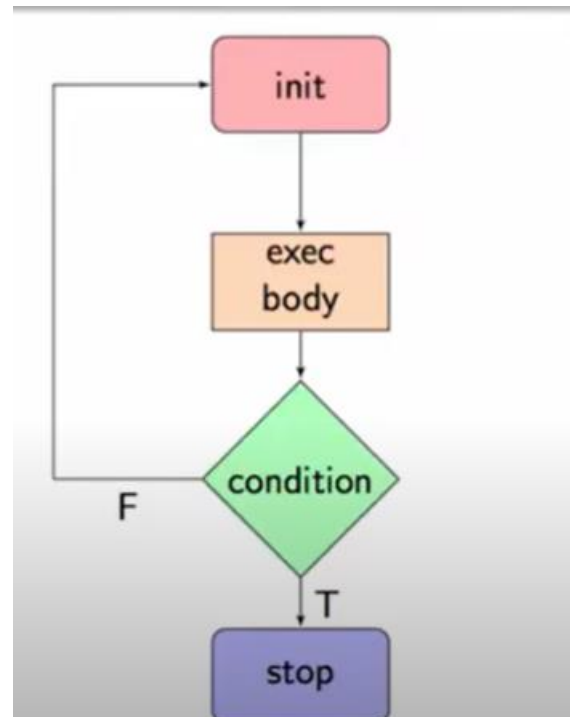
ساختار
های
کنترلی

ساختارهای
تکرار

تابع switch نیز در مواردی جهت بررسی چندین شرط استفاده می گردد. در این تابع آرگومان داخل پرانتز عبارت مورد ارزیابی است که میتواند رشته یا عددی صحیح باشد. اگر مقدار عبارت برابر option1 باشد کد مربوط به آن اجرا خواهد شد. این مورد برای هریک از option ها صادق است و هر تعداد که مورد نیاز ما باشد قابل تعریف خواهد بود.

while loops & repeat statement

حلقه repeat بطورپیش فرض فاقد هرگونه ساختار تصمیم است و در صورت عدم تعریف ساختار شرطی این حلقه به صورت بی نهایت تکرار خواهد شد.



مدیریت
کاراکتر

Pattern
matching

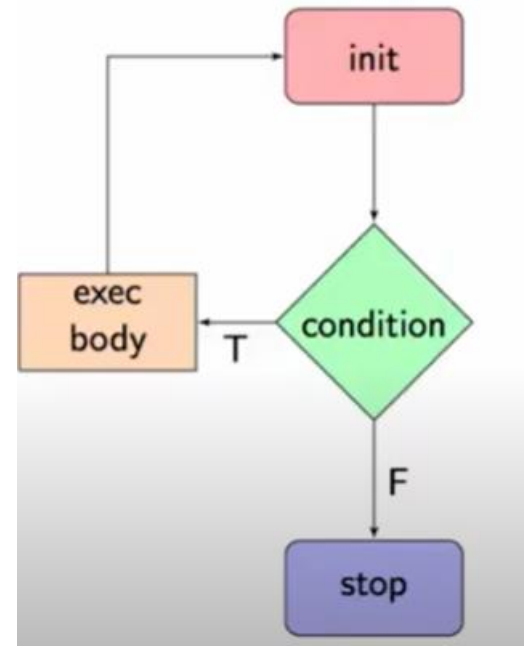
مدیریت
تاریخی

ساختارهای
کنترلی

ساختار
های
تکرار

R

حلقه repeat در نرم افزار R متناظر با ساختار do while در c است. ساختار تکرار دیگر while است. while loop مشابه ساختار repeat با عبارت break است با این تفاوت که در while loop قبل از ورود به حلقه ابتدا شرط بررسی می گردد. در while loop حتماً لازم است که در داخل حلقه شرط نقض گردد در غیر اینصورت حلقه تکرار بی نهایت ایجاد می شود.



مدیریت
کاراکتر

Pattern
matching

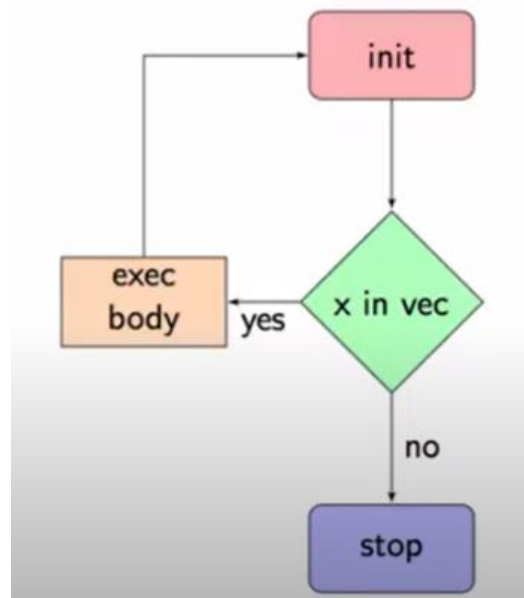
مدیریت
تاریخی

ساختارهای
کنترلی

ساختار
های
تکرار

R

در ساختار حلقه for که دیگر ساختار از جنس تکرار است، یک قطعه کد و دستور به تعداد دفعات مشخص تکرار می‌شود. در نرم افزار R معمولاً از این امکان جهت پیمایش بردار استفاده می‌شود. در پرانتز مقابل for ابتدا متغیر سپس کلمه کلیدی in و در نهایت بردار قرار می‌گیرد.



مدیریت
کاراکتر

Pattern
matching

مدیریت
تاریخی

ساختارهای
کنترلی

ساختار
های
تکرار

R



خسته نباشید!