Function یک موجود هوشمند

فانکشن یک موجود هوشمند است که وقتی یک بار چیزی را برایش تعریف کنیم یعنی به آن ورودی بدهیم و یک دستور العمل برای آن تعریف کنیم می­توانیم دفعات بعد بدون نیاز به تعریف دوباره، خروجی مد نظر را بگیریم

Output

خروجی

تعریف / دستورالعمل

Input

ورودی

\*دیتا تایپ (انواع داده)

انواع مختلف داده­هایی که در فضار آر استودیو وجود دارند شامل:

#Numeric -> numbers اعداد

#Character -> words: names, features, types ( “ali”) اسم­ها،ویژگی­ها، انواع

آیتم­های کاراکتر داخل کوتیشن قرار می­گیرند "علی"\*

کوچک و بزرگ بودن حروف داخل کوتیشن مهم است. دو عبارت زیر دو کاراگتر متفاوت در نظر گرفته می­شوند

"ali", “Ali”

#logical -> True or False یک متغیر که می­تواند گزینه­های مختلف داشته باشد

Hint:

اگرعددی داخل کوتیشن قرار بگیرد یک کارکتر محسوب می­شود و نه یک نیومریک.

\*بردار -> مجموعه­ای از مقادیر

c : combine

\*فاکتور شبیه به بردارکاراکتر است ولی با فایده­های دیگر

Factor

فاکتور بردار قبلی را پاک می­کند و عناصر آن را نمایش می­دهد. فاکتور در حجم بالای دیتا و به این منظور که تک تک داده­ها را جدا جدا بررسی نکنیم کاربرد دارد

\*تشخیص کلاس دیتا

Tabulate(field) --run🡪 [3 5 3 ] اینکه از هر داده چند عدد داری

hint:

برای ران کردن لاین در کنسول آر کلیک راست کنید و گزینه­ی ران را انتخاب کنید

-فاصله­ی داخل " " کوتیشن یه کلاس جدید از دیتا بهتون می­ده

\*برای ران کردن برداری که قبلا ساختید از بردار زیر استفاده کنید

Print(students)

\*با این دستور می­توانیم در هر لحظه کلاس دیتا را مشاهده کنیم

Class(age)

\*با این دستور می­توانیم تعداد بردارها را مشاهده کنیم

Length(students)

Data.frame(students, scores,fields)

کنار هم چیدن بردارها به شکل ستونی

هر بردار یک ستون را تشکیل داده و یک جدول ایجاد می­شود

در صورتی می­توان بردارها را در یک جدول قرار داد که طول برابر داشته باشند

دیتا فریم یک جدول دو بعدی است که شامل:

-ردیف­ها -ستون­ها

\*با دستور زیر ابعاد جدول را مشاهده کنید ( اینکه جدول چه تعداد ستون و ردیف دارد)

Dim(data)

[12 5] عدد سمت راست نمایانگر ستون و عدد سمت چپ نمایانگر ردیف است

\*برای انتخاب کردن قسمت محدودی از جدول از دستور زیر استفاده می­کنیم

#Selection

1-برای انتخاب اندیس کاراکترها :

Student[ c(1,3,4)] داخل براکت یک بردار درست کنید

2-برای انتخاب کاراکتر از یک بردار لاجیکال استفاده می­کنیم دو عدد مساوی و اسم عنصر داخل کوتیشن

Students[students == “ali”]

3-برای حذف کاراکتر از مجموعه از نقیض دستور قبل استفاده می­کنیم

Students[students  **!**= “ali”]

4-انتخاب کلاس دیتا و ترکیب آنها

Students [scores> 10] دانش­آموزانی با معدل بالای ده

5-انتخاب ستون­ها در دیتا فریم

Data$ student انتخاب ستون student

6- انتخاب اندیس در دیتا فریم

Data[1,2]عدد سمت راست نمایانگر ستون وعدد سمت چپ نمایانگر ردیف است

Data[1:2, 3:4] برای انتخاب ردیف­ها و ستون­های متوالی از : استفاده کنید( ردیف 1 تا 2)

Data [ **c**(1,3,5) , 3:4] برای انتخاب ردیف یا ستون­هایی که متوالی نیستند از بردار ترکیب استفاده کنید

Data [ bio, ]

\*اگر جای ستون را ردیف را خالی بگذاریم یعنی همه­ی ستون­ها و همه­ی ردیف­ها

7-انتخاب ردیف مشخص با کلاس دیتای مشخص

Data[data$ field == “bio”] انتخاب ردیف­هایی که فیلد آنها بایو است

A smaller dataframe

ستون­ها در دیتا فریم یک ویژگی اند پس انتخاب داده­ها در ردیف معنا پیدا می­کند

8-برای انتخاب دو ستون از & استفاده می­کنیم

Data [data $ field == bio & data$pass == T]

\*برای ایجاد تغییرات در دیتا فریم ابتدا طبق مراحل قبل آن را انخاب می­کنیم و سپس تغییرات را اعمال می­کنیم

Modification

Data [1,2] 🡨 “amirali” انتخاب کردن جایی که علی در جدول قرار دارد

Hint:

از این دو علامت می­توان به جای هم استفاده کرد

🡪 =

داده­های غیر قابل دسترس

NA handling

هنگامی که به دیتای واقعی کار می­کنید گاهی پیش می­اید که یک دیتای به خصوص در دسترس نیست(برای مثال یکی از آزمودنی­ها تست نداده است )

Data$score [data$score < 10] 🡨 NA

\*تعریف یک متغیر جدید برای داده­های غیرقابل دسترس

Nadata🡨 data[is.na(data$scores) , ]

NA🡨 T, F

T, F 🡪 AV

\*تعریف یک متغیر جدید برای داده­های قابل دسترس با نقیض متغیر قبلی

Ava.data 🡨 data [ ! is.na ( data$scores), ]

F,T,F,T

می­تونید یک بردار رو انتخاب کنیم و تغییرش بدیم

Data$scores [ is,na(data scores)] 🡨 9

NA🡨 9

\*چند متغیر دیگر

Mat( nrow = 4 , ncol = 5 ) این بردار شبیه به دیتا فریم است

List 🡨 list(studnets,scores,data) ترکیب سه بردار داخل پرانتز و ایجاد یک جدول