

Статистик програмчлалын R хэл

Хэрэгцээт зарим чухал ойлголт болон алдааны эргэн тойронд

Г.Махгал

© 2016 Г.Махгал.

Ашгийн бус зорилгоор хэрэглэгдэх нөхцөлтэйгээр чөлөөтэй түгээгдэнэ.

℥ 2016/10/4



Удиртгал

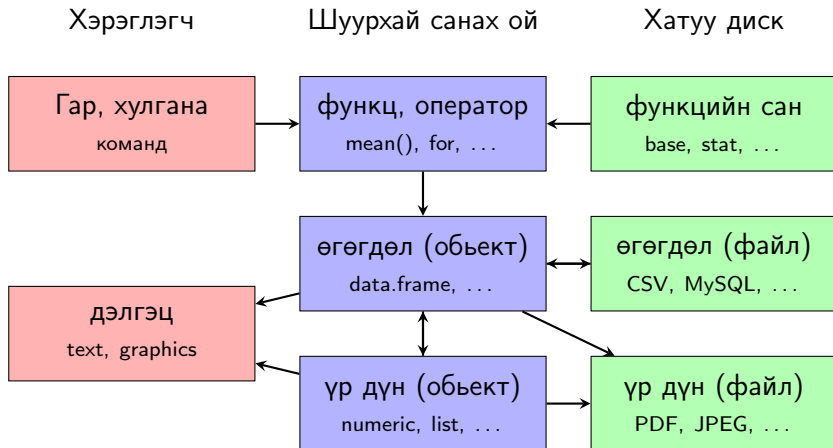
1. R програмын ажиллах зарчмыг схемээр авч үзнэ. Үүний дүнд R програмын талаар илүү тодорхой төсөөлөлтэй болно.
2. Нэрийн огторгуйн тухай ойлголт болон объектийн нэршил, нөөцлөгдсөн үгс, чухал тогтмолуудтай танилцана.
3. Нэр бүхий объектийн оршин буй эсэхийг шалгах болон түүнийг устгах аргыг үзнэ.
4. Код дахь алдааг олох, алдаанаас сэргийлэхтэй холбогдох зөвлөмжийг өгнө.
5. Зарим нийтлэг алдаа, тэдгээрийн шалтгааныг авч үзнэ.

Жишээ код ба зарим слайд интернэтэд:
үзэхээр <https://goo.gl/RNCFHR> татахаар <https://goo.gl/zUs7Hs>



Хэрэгцээт зарим чухал ойлголт

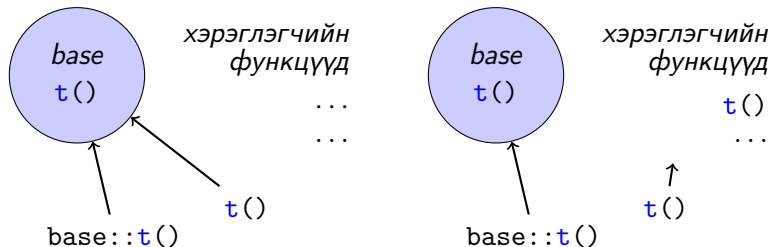
R програмын ажиллах зарчим



Хэрэгцээт зарим чухал ойлголт

Нэрийн огторгуй

Нэрийн огторгуй¹ гэдэг нь багц дахь функц болон бусад объектүүдыг бусдаас ялгаж салгахтай холбогдох ойлголт юм.



base – R програмын үндсэн багц, base::t() – хөрвүүлэх функц

¹namespace



Хэрэгцээт зарим чухал ойлголт

Объектийн нэршил ба нөөцлөгдсөн үгс

- ▶ Объектийн нэр үсгээр эхлэх бөгөөд үсэг, тоо, цэг, доогуур зураасаас тогтоно.
- ▶ R програм нь үсгийн том жижгийг ялгадаг.
- ▶ Нөөцлөгдсөн үгс²:
if else repeat while function for in next break TRUE FALSE NULL Inf
NaN NA NA_integer_ NA_real_ NA_complex_ NA_character_

²reserved words



Хэрэгцээт зарим чухал ойлголт

Зарим чухал тогтмолууд

Тогтмол	Тайлбар
TRUE, FALSE, T, F	логик үнэн ба худал
NULL	хоосон объект
Inf	∞ буюу төгсгөлгүй
NaN	тодорхойгүй
NA	орхигдсон утга



Хэрэгцээт зарим чухал ойлголт

Нэр бүхий объектийн оршин буй эсэхийг шалгах ба устгах

- ▶ Объектийн оршин байгаа эсэхийг шалгах

```
| exists("object.name")
```

- ▶ Объектийг устгах

```
| rm(object.name)
```



Алдааны эргэн тойронд

Код дахь алдааг олох, алдаанаас сэргийлэх

1. Алдааны мэдээллийг анхааралтай унших
2. Кодыг жижиг хэсгүүдэд задлах, завсрын үр дүнгүүдийг хянаж шалгах
3. Функц ба операторын баримтжуулалтыг сайтар уншиж судлах
4. Тухайн функц ба операторын жишээ кодыг ажиллуулж үзэх
5. Нэг зорилготой эсвэл олон дахин ашиглагдах³ бүлэг командыг функц болгох
6. Товч, оновчтой код бичихийг зорих
7. Код дотроо тайлбар⁴ бичих

³ эсвэл өөр өөр хэсэгт байгаа ижилхэн код

⁴ comment, # тэмдэгтийг ашигладаг



Алдааны эргэн тойронд

Зарим нийтлэг алдаа, тэдгээрийн шалтгаан

► could not find function

1. функцийн нэрийг алдаатай бичсэн
2. шаардлагатай багцыг ачаалаагүй
3. объектийн элементийн индексийг дугуй хаалтанд бичсэн

► object not found

1. хувьсагч буюу объектийн нэрийг буруу бичсэн
2. хувьсагч буюу объектийг үүсгээгүй
3. өөр хүрээнд буй хувьсагч буюу объектийг заасан

► cannot open

1. ажлын хавтсыг (working directory) шалгах
2. файлын нэрийг шалгах
3. файлын замыг шалгах
4. файлд хандах эрхийг шалгах



Алдааны эргэн тойронд

Зарим нийтлэг алдаа, тэдгээрийн шалтгаан

► unexpected numeric constant in

1. жишээлбэл, таслал орхих

| c(-1, 0 1)

► unexpected '=' in

1. "==" логик операторыг "=" гэж бичсэн

► object of type '...' is not subscriptable

1. дугуй хаалт (функцийн аргумент өгөх) ба дөрвөлжин хаалтыг (объектийн элементэд индексээр нь хандах) сольж бичсэн

► subscript out of bounds

1. индексийг 0 (тэг) эсвэл үүнээс багаар өгсөн
2. индекс объектийн уртаас хэтэрсэн
3. үл орших индексээр хандсан



Алдааны эргэн тойронд

Зарим нийтлэг алдаа, тэдгээрийн шалтгаан

► not meaningful for factors

1. чанарын өгөгдөл дээр арифметик үйлдэл гүйцэтгэсэн ашигласан, жишээлбэл

```
X = factor(c("X", "Y", "X", "X", "Y"))  
X + 1
```

► invalid 'type' (character) of argument

ЭСВЭЛ

argument is not numeric

1. текст дээр арифметик функцийг ашигласан, жишээлбэл

```
str = "text"  
mean(str)
```



Алдааны эргэн тойронд

Зарим нийтлэг алдаа, тэдгээрийн шалтгаан

► unexpected 'else' in "else"

1. else түлхүүр үгийг шинэ мөрнөөс шууд бичсэн, тухайлбал

```
x <- 3
if (x %% 2 == 0)
  print("even")
else
  print("odd")
```

зөв бичлэг нь

```
if (x %% 2 == 0) {
  print("even")
} else {
  print("odd")
}
```

```
if (x %% 2 == 0) print("even") else print("odd")
```



Алдааны эргэн тойронд

Анхааруулга, алдаа болон мэдэгдэл үүсгэх

- ▶ **Анхааруулга** Командыг үргэлжлүүлэн гүйцэтгэнэ.

```
| warning("Heads Up!")
```

```
| Warning message:  
| Heads Up!
```

- ▶ **Алдаа** Командыг таслан зогсооно.

```
| stop("Fatal Error!")
```

```
| Error: Fatal Error!
```

- ▶ **Мэдэгдэл** Зөвхөн мэдэгдэл өгнө.

```
| message("Done")
```

```
| Done
```





© 2016 Г.Махгал

Ашгийн бус зорилгоор хэрэглэгдэх нөхцөлтэйгээр чөлөөтэй түгээгдэнэ.

