Introduction to Statistics and R

Computing with and plotting of data tables

Eric Stemmler

24.03.2021

Contents

1 Recap: Accessing data tables

- Accessing columns from a `data.table`
  - Single columns: e.g. `dt\$colA`
  - Multiple columns: e.g. `dt[, c("colA", colB")]`
- Accessing rows/ subsetting a data.table
  - By row number: e.g. `dt[1:3]` returns first 3 rows
  - By variable values: e.g. `dt[colA > 3 \& colB < 2]`

Бид өнгөрсөн хичээлээр data.table объектод хадгалагдсан өгөгдлийг хэрхэн сонгох, ашиглах боломжуудийн талаар ярилцсан. Тухайн хүн нь хэд хэдэн хувьсагч болон хэмжилтүүдийн өгөгдлийн багцтай байдаг тул энэ нь нэлээд чухал зүйл юм. Өгөгдлийн багцад дүн шинжилгээ хийхдээ бид гар дор байгаа өгөгдлийнхөө зарим хэсгийг сонгох шаардлагатай болдог. Жишээ нь сонгосон хоёр хувьсагчийн хоорондын хамаарлыг визуалчлах/дүрслэх-д. Тодорхой өгөгдлийн багцтай танилцаад тухайн хүн нь ихэвчлэн илүү чухал гэж үзэхүйц цөөн хэдэн хувьсагчид дүн шинжилгээ хийж эхлээд, дараагийн шатанд бусдыг нь авч үзэх замаар үргэлжлүүлэн судалдаг. Бид data.table-ээс дан ганц баганад хандахад $-operator-ийг хэрхэн ашиглахыг үзсэн байгаа. R-консол дээр ажиллахдаа тодорхой баганы нэрийг олоход туслахын тулд TAB товчийг дахин ашиглаж болно. Нэгээс илүү баганыг сонгохын тулд бид арай илүү үйлдлийг хийх шаардлагатай. Үүний нэг арга нь бол баганын нэрнүүдийн векторыг өгөөд, зөвхөн эдгээр хоёр баганыг багтаасан шинэ хүснэгтийг гаргаж ирэхийг data.table объектоос хүсэх явдал юм. Дөрвөлжин хаалтанд бид таслалын тэмдгийн дараа баганын нэрүүдийг өгдөг болохыг анхаарна уу. Таслал тэмдгийн өмнөх мэдэгдлийг data.table -ийн мөрөнд, ардах мэдэгдлийг баганад хэрэглэдэг гэдгийг санаарай. Үүний дагуу бид data.table-ээс хэд хэдэн мөрийг таслалын өмнө мэдэгдэл өгөх замаар сонгож болно. Жишээ нь мөрийн дугаар болох багц тоог бэлтгэж өгөх. Эсвэл баганууд нь то-

дорхой утгатай мөрүүдийг сонгож болно. Үүнийг хийхийн тулд бид boolean эсвэл логик операторуудыг ашиглаж болно.

2 Aggregating grouped data

2.0.1 Calculating measures of columns

```
library(data.table)
dt <- fread("age-guessing.csv")
dt[, mean(card_1)]
```

```
## [1] -2.3
```

```
dt[abs(card_2) > 5, mean(card_1)]
```

```
## [1] -1.666667
```

```
dt[, var(card_1)]
```

```
## [1] 24.01111
```

```
dt[, var(card_1) + var(card_2)]
```

```
## [1] 120.2444
```

data.table объектоос өгөгдлийг сонгож, шүүх боломжтой болсон тул үүгээр юу хийж чадахаа мэдэх цаг болжээ. Өмнө дурьдсанчлан, data.table-ийн доторх таслалын дараа өгдөг мэдэгдлүүд баганууд ашиглагддаг. Энэ нь бид сонгосон баганууд дээрээ тодорхой тооцооллыг шууд data.table дотор хийх боломжтой гэсэн үг юм. Та бидний нас таах өгөдлийн багц дээрх ердийн статистик тооцооллын сонголтыг энд харж байна. Кодын жишээний хоёрдахь мөрийг бас анхаарна уу: Бид card_1 баганын дундажийг тооцох боловч бүх утгыг ашиглахын оронд зөвхөн card_2-ийн абсолют алдаа 5-аас их байх утгын дундажыг тооцно. Тиймээс бид кодын эхний мөрөөс өөр үр дүнд хүрч байна.

2.0.2 Using factors

- How to categorize data? Use factors!
- factors are categories. Example: “mammals”
  - Horse
  - Sheep
  - Cow
  - Goat
- different categories in a factor are called “levels”
- In R we say: factor `mammals` has levels `Horse`, `Sheep`, `Cow`, `Goat`

Нас таах өгөгдлийн жишээнд бидэнд тодорхой нэг гэрэл зургийн хувьд бүлэг тус бүрт үргэлж нэг утга байна. Гэхдээ нэг бүлэгт нэг л хэмжилт байх нь онцгой тохиолдол юм. Ихэнх тохиолдолд нэг бүлэгт хэд хэдэн хэмжилт байдаг. Бүлэг тус бүрээр нь харсанчлан эдгээр төрлийн тооцоог хийх маш үр дүнтэй арга бол factor/фактор-ийг ашиглах явдал юм. Factors/факторууд нь үндсэндээ категориудыг тодорхойлдог R дахь өөр нэг өгөгдлийн төрөл юм. Хувьсагчийг factor/ фактор болгон мэдэгдэх нь R-д энэ хувьсагчийг арай өөрөөр зохицуулахыг хэлж өгдөг. Жишээлбэл: Бид малын талаархи зарим өгөгдлийг цуглуулаад, зүйлийн төрлийг “mammal”/ “хөхтөн” гэсэн нэртэй хувьсагч болгон тэмдэглэсэн гэж үзье. Янз бүрийн категориудыг R-д factor/фактор-ийн levels/ түвшингүүд гэж нэрлэдэг. Тиймээс өгөгдсөн жишээнд бид "Морь", "Хонь", "Үхэр", "Ямаа" гэсэн levels-тэй mammal нэртэй factor /фактор хувьсагчтай байх болно.

mammal	age	weight
horse	1	359.46
horse	1	404.33
horse	1	421.24
sheep	2	92.78
sheep	2	87.12
sheep	3	63.69
cow	1	1047.44
cow	3	402.14
cow	3	720.19
goat	3	41.97
goat	1	-31.23
goat	1	85.48

Ийм өгөгдлийн багцын жишээг энд харуулсан байна. Бидэнд насыг нь жилээр нь тооцсон болон, жинг нь хэмжсэн 12 төрлийн амьтан байна.

Example: Create a factor in R

```
my_factor <- factor(x = c("horse", "sheep", "cow", "goat"))
class(my_factor)
```

```
## [1] "factor"
```

```
my_factor
```

```
## [1] horse sheep cow   goat
## Levels: cow goat horse sheep
```

R-д факторыг хэрхэн үүсгэх вэ? Бид factor() нэртэй функцийг ашигладаг нь гайхах зүйл биш юм. Энэ функц нь фактор болох ёстой утгуудыг агуулсан x параметрийг авдаг. factor() функц нь x-ийн бэлтгэж өгсөн ялгаатай утгууд дээр үндэслэн өөр өөр levels буюу категориудыг өөрөө автоматаар олж мэдэх болно. Бид энэ тохиолдолд my_example гэж нэрлэдэг хувьсагч дээр үр дүнг хадгалж болох бөгөөд энэ нь үнэхээр фактор болохыг мэдэхийн тулд class()-ийг дуудаж өгөгдлийн төрлийг шалгаж болно. Энэ хувьсагчийг хэвлэхэд үр дүн нь энгийн тэмдэгт векторийг хэвлэхээс арай өөр харагдаж байна. R нь векторын утгуудыг ердийнхөөрөө дахин хэвлэнэ. Гэхдээ үүнээс гадна factor() функцийн тодорхойлсон янз бүрийн levels-ийг хэвлэнэ. Эдгээр levels-ийг үргэлж цагаан толгойн дарааллаар эрэмбэлдэг болохыг анхаарна уу.

Тиймээс хүснэгт үүсгэх үед категори тус бүрт өгөгдлийн утгатай байх тусам давтагдах категорийн вектор бидэнд үнэхээр хэрэгтэй болно. Энэ нь маш их бичиж шивэх гэсэн санаа байж болох юм. Учир нь бид жишээ нь x параметрийн вектор луу 5 удаа "морь" гэсэн үгийг шивэх хэрэгтэй болно. Гэхдээ үүнийг илүү хурдан хийх арга нь энд байна: Бид үүнийг хийхийн тулд давтах функц (rep())-ийг ашиглаж болно. rep() -ийг ашигласнаар энэ нь цөөхөн хэдэн утгыг үндэслэн илүү том векторуудыг үүсгэх явдлыг маш энгийн болгодог. Үндсэндээ each болон times гэсэн параметрүүдээрээ ялгагдах rep() функцийн хоёр хувилбар байдаг. Тиймээс бид үндсэн утгуудын x параметрийг бэлтгэн өгч, дараа нь each параметрийг өгч үүндээ давталтын тоог зааж өгнө. rep () нь дараа нь утга тус бүрийг x-ээс хувилан бичиж үүнийг анхны тохиолдлынх нь дарааллаар нэгтгэх болно.

... or using parameter `times` to repeat the whole sequence:

```
# use rep() to avoid too much typing:
my_categories <- rep(x = c("horse", "sheep", "cow", "goat"),
    times = 3L)
my_categories
```

```
##  [1] "horse" "sheep" "cow"   "goat"  "horse" "sheep"
##  [7] "cow"   "goat"  "horse" "sheep" "cow"   "goat"
```

rep() -ийн хоёрдахь хувилбар нь times параметрийг ашигладаг. Жишээнээс ялгааг нь та харж байна: times нь rep() -ийг вектор х-ийг бүхэлд нь n удаа давтаж, бүгдэнгийн хамтад нь дараалуулан нэгтгэхэд хүргэдэг.

... and then convert to factor

```
my_categories <- factor(my_categories)
my_categories
```

```
##  [1] horse sheep cow   goat  horse sheep cow   goat
##  [9] horse sheep cow   goat
## Levels: cow goat horse sheep
```

Note: Levels are sorted alphabetically

Одоо бид rep() -ийн үр дүнг ашиглан категори тус бүрийг бичихгүйгээр тэдгээрийг аятайхан бөгөөд амархан фактор болгон хувиргаж чадна. factor() функц нь my_categories хувьсагчид өгөгдсөн олон утгуудын дотроос ялгаатай өвөрмөц levels-ийг тодорхойлж байгааг дахин анхаарна уу.

2.0.3 Calculating measures per groups

Calculating e.g. the mean of column `weight`:

```
dt[, mean(weight)]
```

```
## [1] 307.8838
```

... not very meaningful, since these are different mammals. How to calculate for each mammal type?:

```
dt[, mean(weight), by = mammal]
```

```
##    mammal        V1
## 1:  horse 395.00913
## 2:  sheep  81.19747
## 3:    cow 723.25598
## 4:   goat  32.07251
```

Одоо бидний data.table-ийн товч хураангуй статистикийг тооцоолох талаар дахин авч үзье. Бид малын өгөгдлийнхөө дундаж жинг сонирхож байна гэж бодъё. Өмнө нь үзсэнчлэн бид weight багана дээр mean() функцийг дуудах замаар data.table-ийн доторх дундажыг тооцоолж болно. Гэсэн хэдий ч, энэ функц нь янз бүрийн амьтдыг ялгаж салгахгүй тул энэ аргаар тооцоолсон дундаж утга нь хэрэггүй зүйл юм. Угаасаа ямаа үхрээс хөнгөн, морь нь үхрээс хөнгөн байж болох ч ямаанаас хүнд юм. Тиймээс янз бүрийн амьтдын өгөгдлийг хамтад нь хэрэглэж амьтны дундаж жинг тооцоолох нь утгагүй юм. Статистикт энэ төрлийн харьцуулалтыг заримдаа "алим, жүрж хоёрыг харьцуулах" гэж хэлдэг бөгөөд хоёр зүйл огт өөр болохыг илэрхийлдэг. Ямар ч байсан одоо бид R-д факторыг хэрхэн мэдэгдэхээ мэддэг болсон тул data.table дотор by параметрийг ашиглан үүнээс зайлсхийх боломжтой юм. Хэрэв бид өгөгдлийн хүснэгтийн баганын дундажийг дахин тооцоолох болон үүнээс гадна by параметрт өөр баганыг зааж өгөх юм бол бидний data.table нь энэ баганад байгаа категориудтай адил олон тооны дундаж утгыг тооцоолно.

Using `by`: Do anything you do in `rows` on `columns` for each different value in ***group*** separately

```
dt[rows, columns, by = group]
```

Энэ талаар дахин бодож үзье. Бид одоо data.table доторх өгөгдөлд хандах өөр аргыг сурч байна. Мөр сонгох, багана сонгох, тооцоолохыг дамжуулдаг код бичих боломжтойг бид аль хэдийн үзсэн бөгөөд одоо гуравдахь хувилбарын хувьд бид нэг ба түүнээс дээш баганын нэрийг бүлэгт өгөх замаар

бүлэгүүдийг юу бүрдүүлдэг болохыг тодорхойлж дараа нь бүлэг эсвэл категори тус бүрийн баганад тусад нь зааж өгсөн ямар ч тооцооллыг гүйцэтгэх боломжтой юм.

Calculating e.g. the mean of column `weight`:

```
dt[, mean(weight)]
```

```
## [1] 307.8838
```

... not very meaningful, since these are different mammals. How to calculate for each mammal type?:

```
dt[, mean(weight), by = mammal]
```

```
##    mammal        V1
## 1:  horse 395.00913
## 2:  sheep  81.19747
## 3:    cow 723.25598
## 4:   goat  32.07251
```

Тэгэхээр энэ жишээнд харагдаж байгаачлан by-ийг ашиглах үед data.table нь параметр by-д өгсөн багана дахь ижил утгатай дундаж утга тус бүрт мөрүүдийг автоматаар ашигладаг. Ийнхүү бид бүлэг тус бүрийн дундаж тооцооллын чухал ач холбогдолтой үр дүнг олж авна. Өөрөөр хэлбэл бүх адуу, ямаа, үхэр, хонины дундаж жинг.

Calculating e.g. the mean of column `weight`, grouping by **two variables**:

```
dt[, mean(weight), by = c("mammal", "age")]
```

```
##    mammal age         V1
## 1:  horse   1  395.00913
## 2:  sheep   2   89.94874
## 3:  sheep   3   63.69492
## 4:    cow   1 1047.43971
## 5:    cow   3  561.16411
## 6:   goat   3   41.96568
## 7:   goat   1   27.12593
```

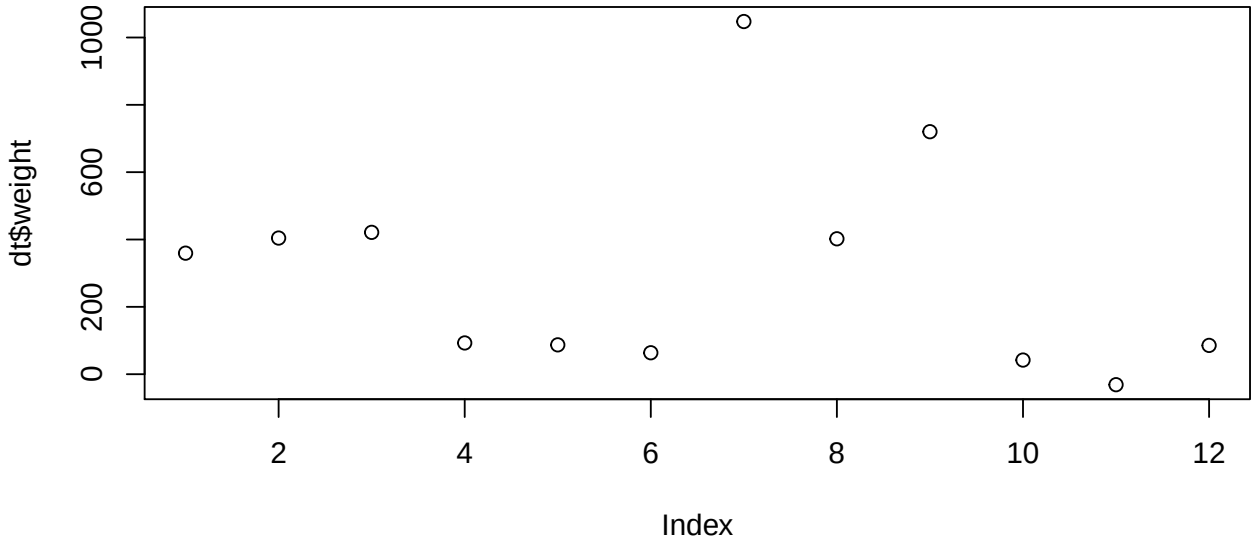
Үнэн хэрэгтээ data.table нь бүлэглэхэд ямар хувьсагч ашиглах вэ гэдгийг үнэхээр чухалчилж үздэггүй. Бид өгөгдлөө бүлэглэхийн тулд дурын тоо эсвэл баганыг ашиглаж болно. Бид үүнийг баганын нэрүүдийн тэмдэгт вектороор хангах замаар арай өөрөөр бичих л шаардлагатай.

3 Data Visualization: Plotting

3.0.1 Plotting in R

Scatterplot: Plot values from **one** or two columns

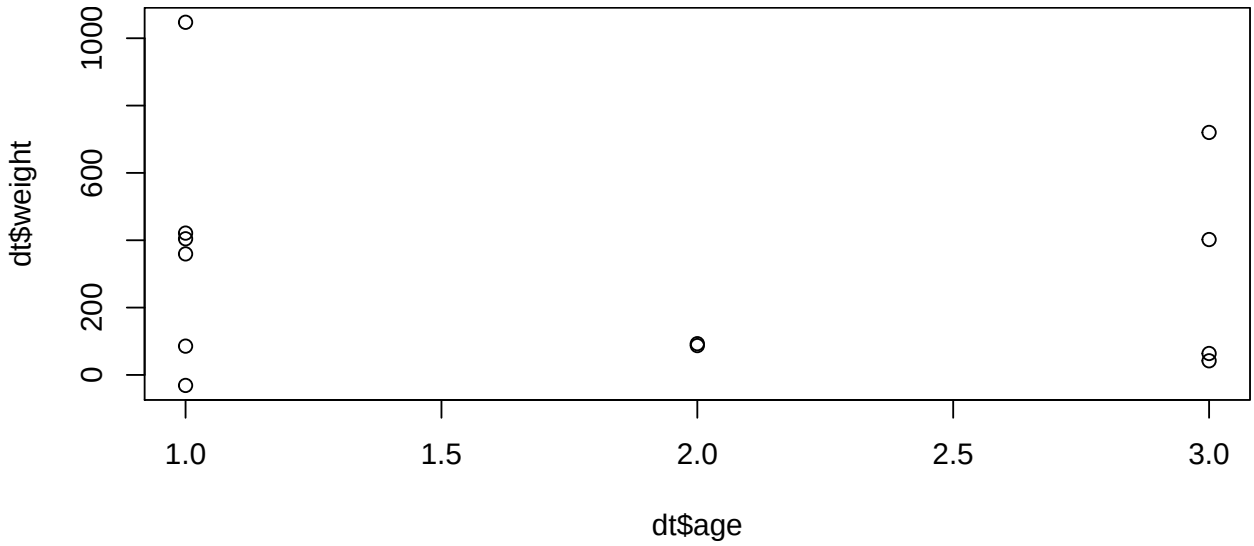
```
plot(dt$weight)
```



Судалгааны өгөгдөлд дүн шинжилгээ хийхэд хамгийн ашигтай графикийн төрлүүдийн нэг бол тархалтын график юм. Та plot() командыг ашиглан тархалтын графикийг хурдан зурж болох бөгөөд утгын нэг эсвэл хоёр вектороор хангах боломжтой. Хэрэв та утгуудын дан ганц нэг векторийг өгвөл эдгээр утгыг у тэнхлэгт дүрсэлж, х бүрэлдэхүүн хэсэг нь индексийн дугаар эсвэл дарааллын тоог гаргах болно. Өөрөөр хэлбэл векторын эхний цэг нь 1-тэй тэнцүү х бүрэлдэхүүн хэсэгтэй байх ба хоёр дахь утга нь 2-той тэнцүү х бүрэлдэхүүн хэсэгтэй байх болно гэх мэт.

Scatterplot: Plot values from one or **two** columns

```
plot(y = dt$weight, x = dt$age)
```

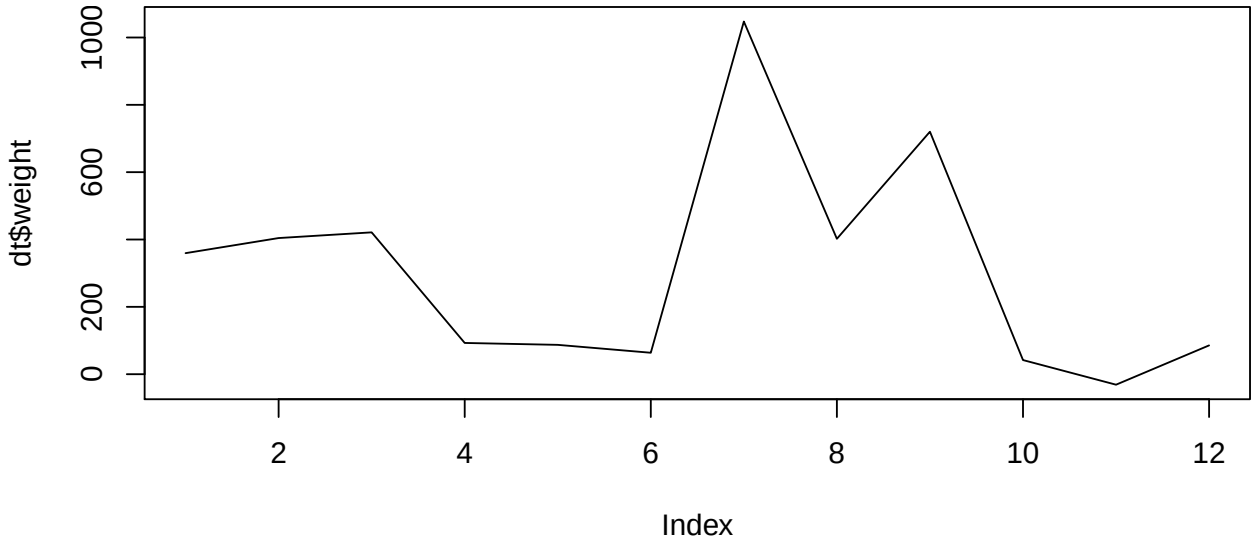


Хэрэв бид утгын хоёр векторыг өгвөл х бүрэлдэхүүн хэсгийн индексжүүлэлтийг алгасах болно. Илүү тодорхой код бичихийн тулд ямар хувьсагчийг хэрхэн зураглахыг хүсч байгаагаа тодорхойлохын тулд х, у параметрийн нэрүүдийг байнга ашиглахыг зөвлөе.

3.0.2 Different plot types

Line plot: Like scatterplot where points are connected by lines

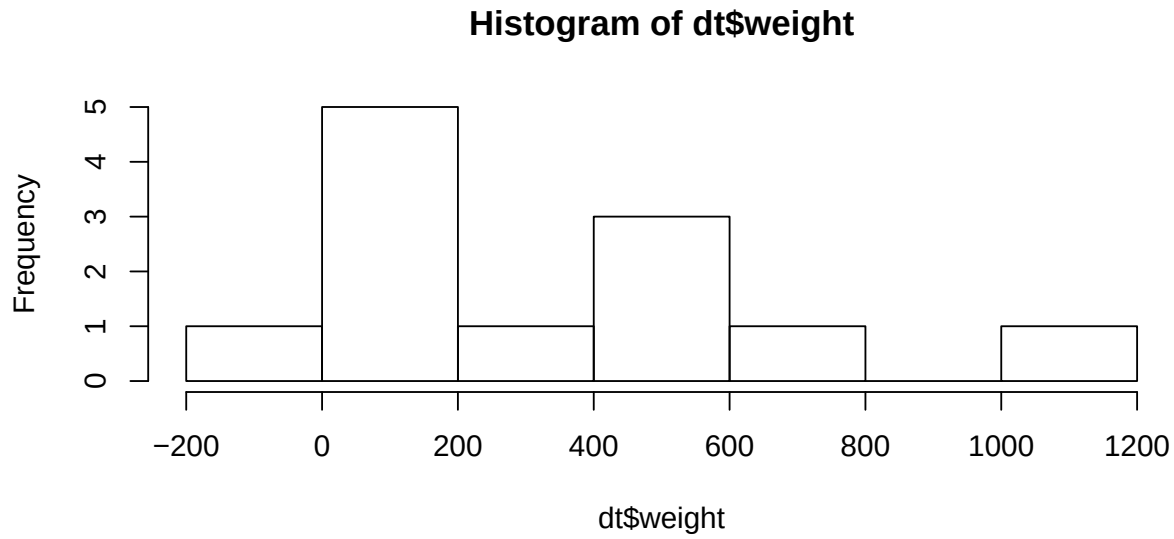
```
plot(dt$weight, type = "l")
```



Тархалтын графиктай ижил төстэй энгийн боловч өөр хувилбар бол шугаман график ашиглах явдал юм. Шугаман график нь өгөгдлийн цэгүүдийг шугамаар холбодог. Энэ нь дараалсан цэгүүдийн хоорондох харьцангуй өөрчлөлтийг сонирхож байгаа үед хэрэгтэй байдаг. Энэ нь шугаман сегмент нь эерэг эсвэл сөрөг налуутай эсэхийг харахад хялбар бөгөөд тиймээс утгууд нэгээс нөгөөд буурах эсвэл нэмэгдэх эсэх талаарх бүр жижиг өөрчлөлтүүдийг ч гэсэн харахад хялбар байдаг.

Histogram

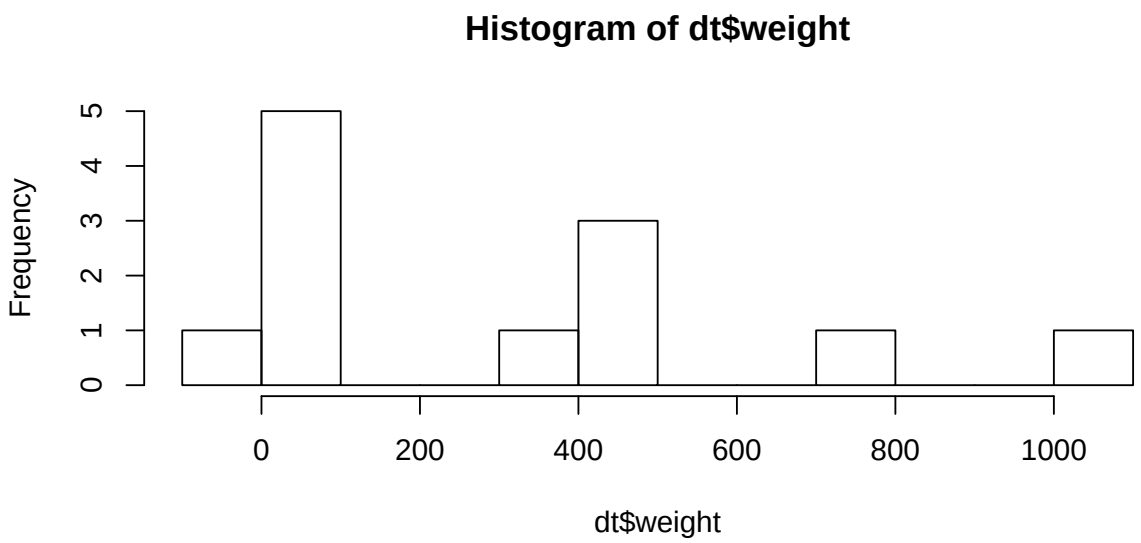
```
hist(dt$weight)
```



Гистограм зурах нь тархалтын графиктай адил хялбар байдаг. Бид hist() командыг ашиглаж, утгуудын дан ганц векторыг бэлдэж өгнө. Интервалын хэмжээг автоматаар сонгох боловч параметр breaks-ийг өгөх замаар үүнийг бас тохируулж болно.

Histogram

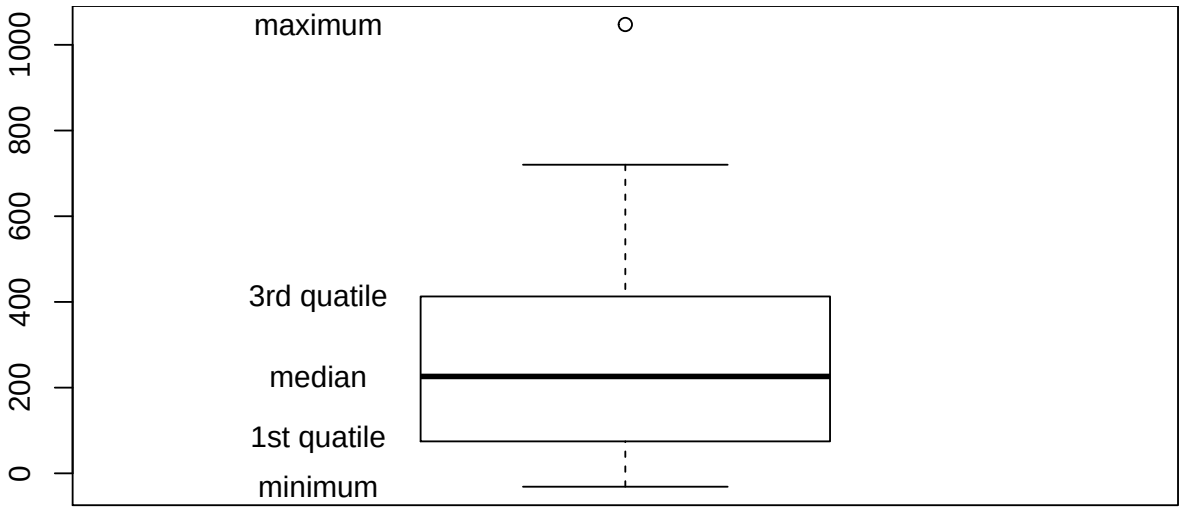
```
hist(dt$weight, breaks = 8)
```



Хэрэв бид параметр breaks-ийг зааж өгвөл энэ жишээн дээрх шиг интервалын тоог тохируулахын тулд аль нэгийг нь сонгож болно. Интервал бүр дээр илүү их хяналт тавихын тулд бид интервал тус бүрийг болон байршлыг нь тусад нь зааж өгсөн breakpoints/таслах цэг гэж нэрлэгдэх векторыг өгч болох байсан. Хэрэв та энэ функцийг хэрхэн ашиглах талаар илүү ихийг мэдэхийг хүсч байвал R-ийн консол дотор ? hist -ийг дуудах замаар R-ийн баримт бичгийг үзэхийг зөвлөж байна.

Box-wishker plot

```
boxplot(dt$weight)
```



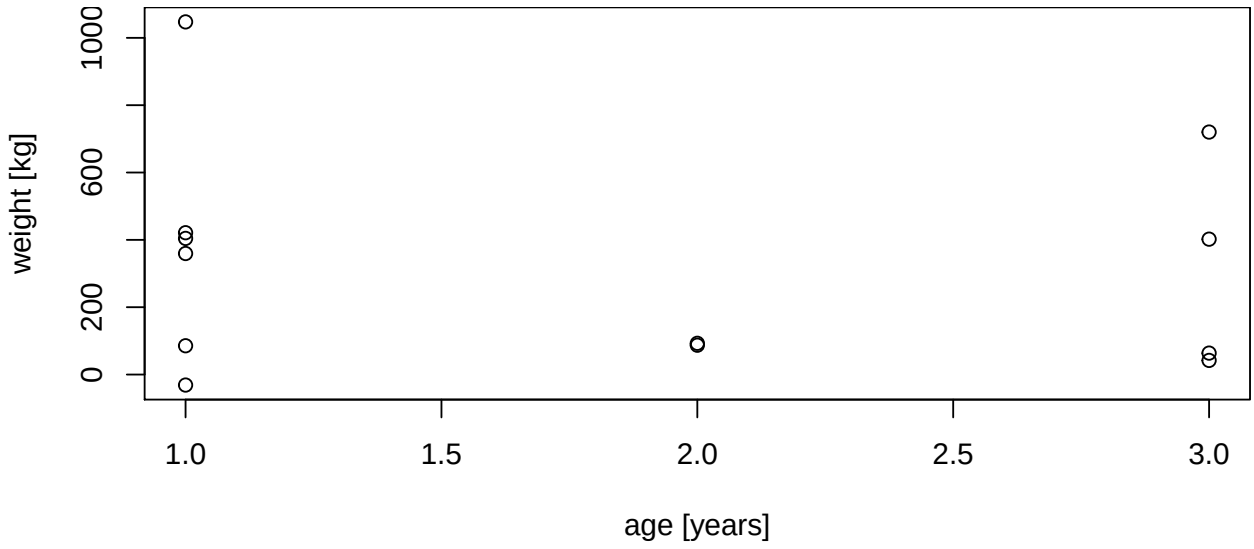
Энэ удаагийн хичээлээр та бүхэнд харуулахыг хүссэн хамгийн сүүлийн графикийн төрөл бол box-whiskers гэж нэрлэгддэг график юм. Энэ нь бидний энэ сургалтаар үзсэн ихэнх хураангуй статистикийг графикаар харуулдаг. box-whiskers графикийг boxplot() функцийг дуудаж нэг ба түүнээс дээш утгын вектороор хангах замаар зурж болно. boxplot функц нь 1-рт медиан буюу голчыг, 1, 3-р квартилийг, хамгийн бага ба хамгийн их утгыг тооцоолж, хоёрдугаарт, эдгээр утгыг илэрхийлэх хайрцгийг зурна. Хамгийн бага болон их утга нь whiskers хэмээгчээр илэрхийлэгддэг. Энэ нь багц утгыг графикаар нэгтгэх, байршил, тархалтыг хоёуланг нь харуулах үр дүнтэй арга юм.

3.1 Styling a plot

3.1.1 Titles and lables

Scatterplot: Changing **axis labels**

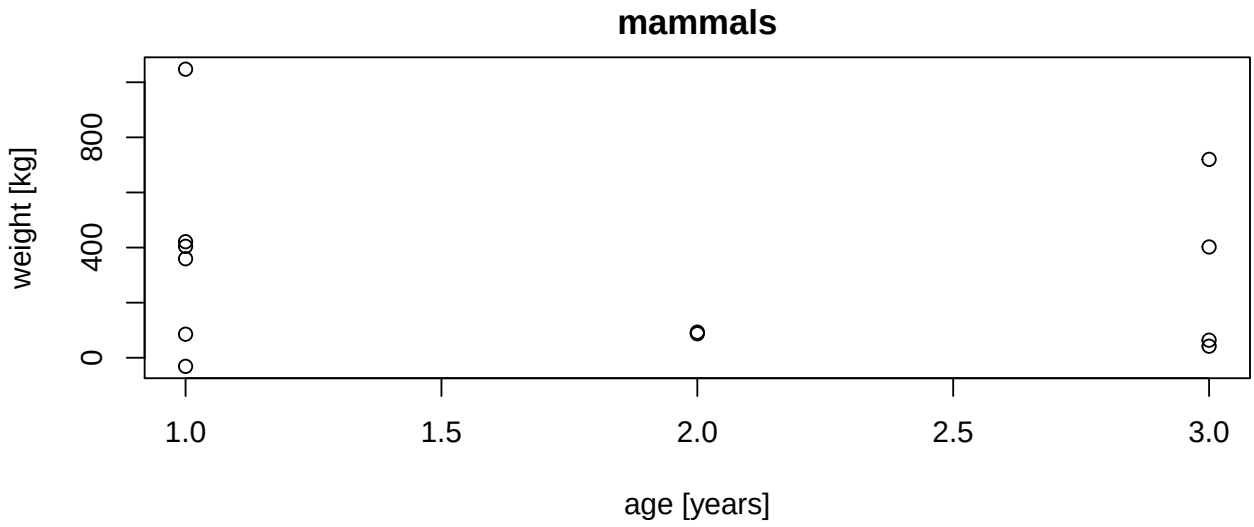
```
plot(y = dt$weight, x = dt$age,
     xlab = "age [years]", ylab = "weight [kg]")
```



Сая үзсэнчлэн график зурах нь өгөгдлийн багцыг судлах, шинжлэхэд сайн арга боловч хэвлэлтэнд ашиглах нь бас чухал юм. Бидний box графикуудаас гадна их бага хэмжээгээр зурсан графикууд нь хэвлэлтэнд хараахан бэлэн болоогүй байна. Энэ хэсэгт би эдгээр графикийг хэрхэн илүү мэдээлэл сайтай, сэтгэл татам болгохыг харуулья. Анхдагчаар plot() функц нь R хувьсагчийн нэрсийн нэр дээр үндэслэн y ба x тэнхлэгийн label-ийг тохируулна. Эдгээр нэрс нь ихэвчлэн богино, нууцлаг байдаг тул бодитоор зурагласан зүйлийн талаар илүү уншууштай, тодорхой тайлбар өгөх нь дээр юм. X ба y тэнхлэгийн label-ийг тохируулахын тулд жишээн дээр байгаачлан xlab ба ylab гэсэн хоёр параметрийг бэлтгэж өгөх боломжтой юм.

Scatterplot: adding a **title**

```
plot(y = dt$weight, x = dt$age,
     xlab = "age [years]", ylab = "weight [kg]",
     main = "mammals")
```

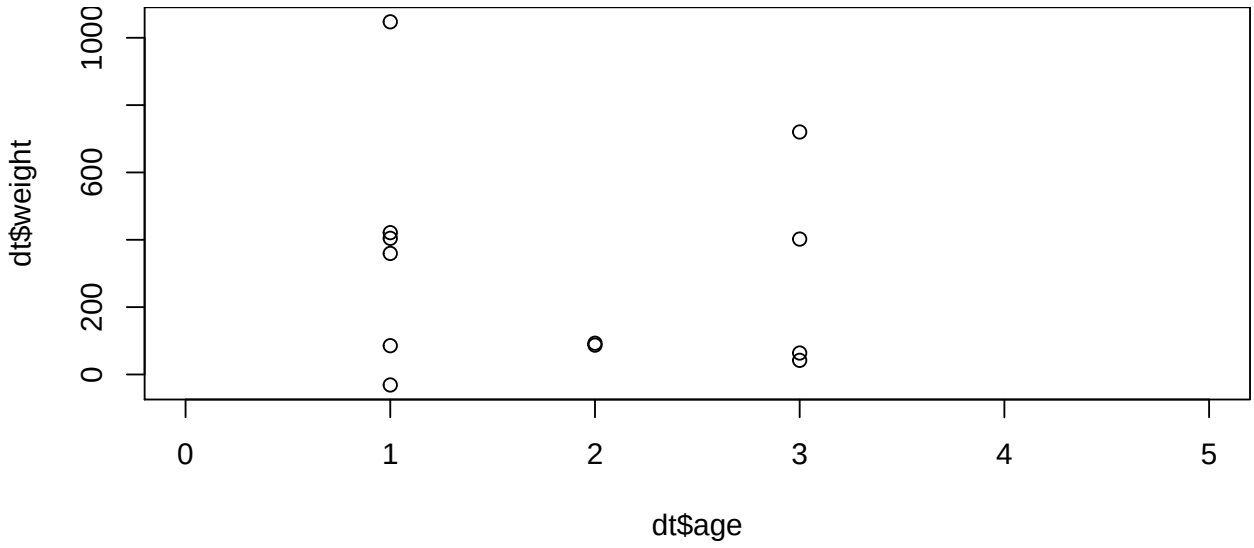


Нэмж хэлэхэд бид main параметрийг зааж өгснөөр таны графикт гарчиг өгөх боломжтой юм.

3.1.2 Adjusting the axis

Scatterplot: Changing **axis limits**

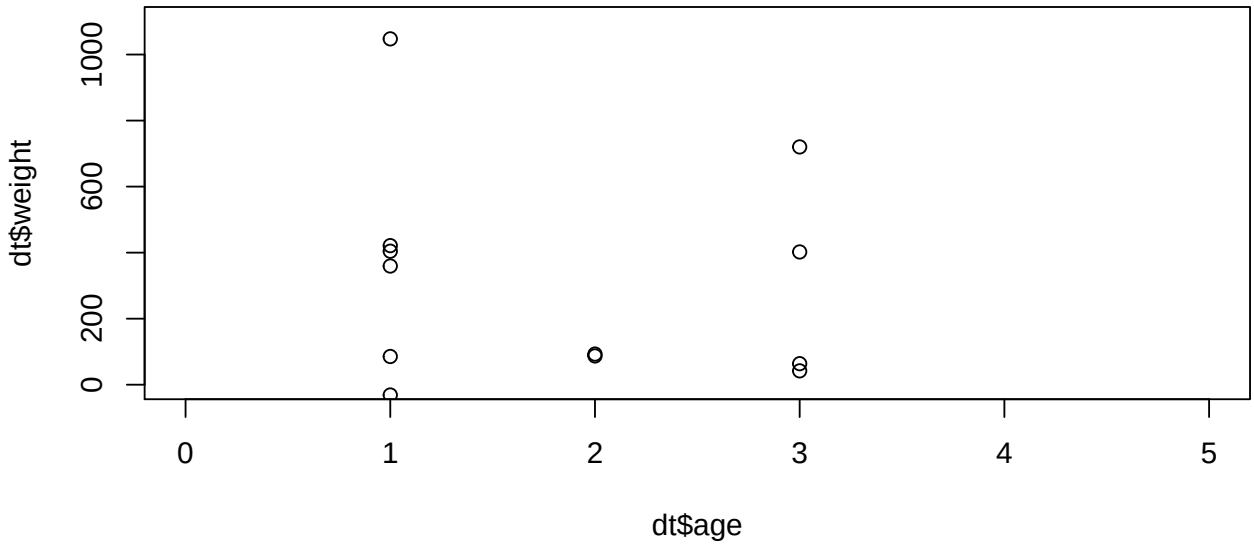
```
plot(y = dt$weight, x = dt$age,
     xlim = c(0, 5))
```



R-ийн зураглалын функцууд нь ихэвчлэн өгөгдөлийн дэлгэцийн цар хүрээг өөрсдөө сонгодог. Хэрэв та анхдагч тохиргоонд сэтгэл ханагалуун бус байвал xlim ба ylim параметрүүдийн аль нэгийг эсвэл хоёуланг нь зааж өгснөөр дэлгэцийн хязгаарыг өөрчлөх боломжтой. Эдгээрийн хувьд та тус бүрдээ доод ба дээд хязгаар гэсэн хоёр утгыг агуулсан тоон векторыг бэлтгэж өгөх ёстой.

Scatterplot: Changing **axis limits**

```
plot(y = dt$weight, x = dt$age,
     xlim = c(0, 5), ylim = c(0, 1100))
```

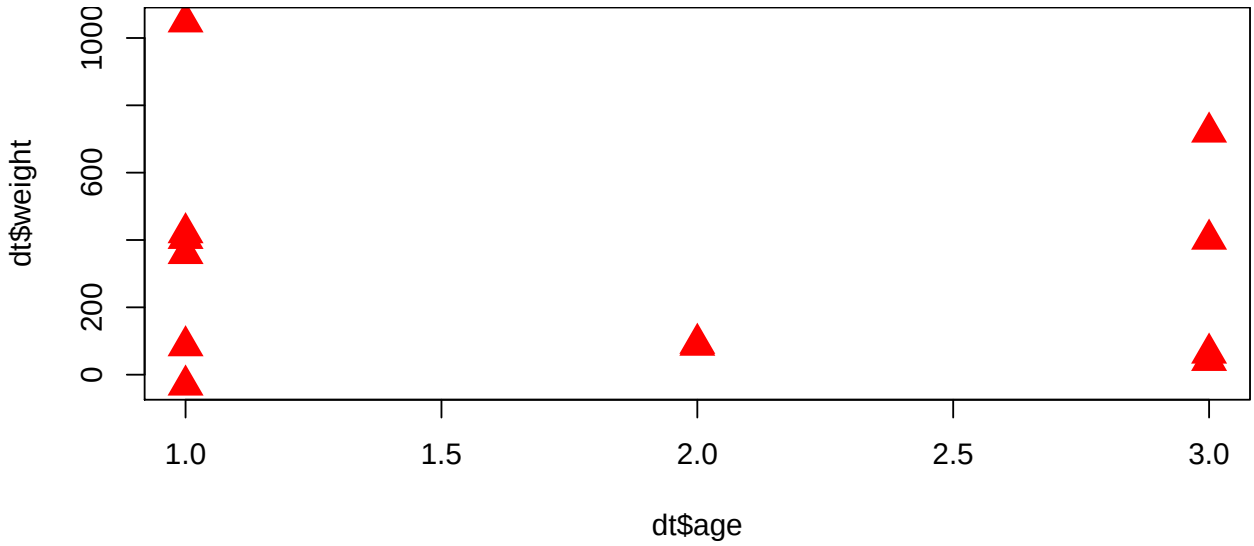


Энэ жишээнд x ба y тэнхлэгийн аль алинд нь хязгаарууд хэрхэн ашиглахыг харуулав.

3.1.3 Color, shape and size

Scatterplot: Changing **color**, **shape** and **size**

```
plot(y = dt$weight, x = dt$age,
     col = "red", pch = 17, cex = 2)
```

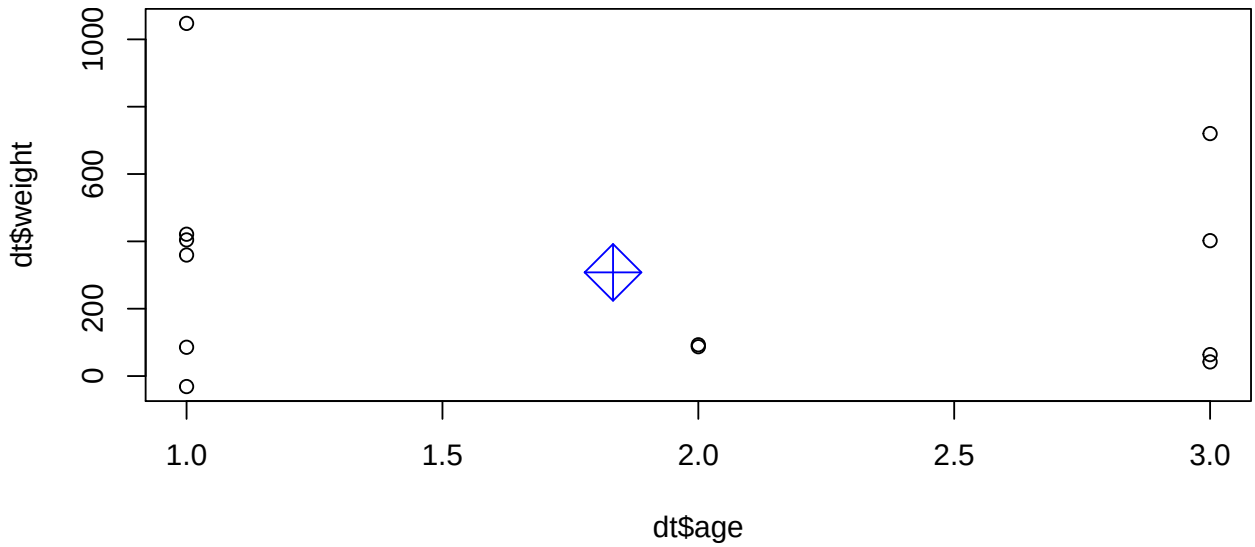


Бид үнэндээ энэ төрлийн графикийн бараг бүх шинж чанарыг өөрчилж чадна. Гэсэн хэдий ч зураглалыг хэрхэн өөрчлөх талаархи бүх боломж нэг бүрийг хамрах боломжгүй бөгөөд бидний анхаарч үзэх графикийн өөр нэг үндсэн тал бол өнгө юм. Түүнчлэн цэгэн тэмдгийн төрөл, хэмжээг өөрчлөх боломжтой. Жишээлбэл, хэрэв бид янз бүрийн бүлэг цэгүүдийг визуалчлахыг хүсч байвал энэ нь хэрэгтэй байдаг. Бүлэг бүрийн хувьд бид өөр өнгө сонгох боломжтой. Өөр нэг боломжтой зүйл бол түүхий буюу ажиллаагүй өгөгдлийн цэгийг стандарт өнгө, хэлбэр, хэмжээгээр зураглах, дараа нь байршлын хураангуй статистикийг нэмэх, жишээ нь өөр өнгө, хэлбэр, хэмжээ бүхий дундаж утгыг нэмэх явдал юм. Үүнийг эхлүүлэхийн тулд энэ жишээн дэх plot() командыг харна уу. Цэгүүдийн өнгө нь улаан бөгөөд үүнийг col параметрийг ашиглан тохируулсан болно. Нэмж дурдахад тойргийг цэгэн хэлбэр болгон ашиглахын оронд бид гурвалжинг ашигласан. “Point character”/ "Цэгэн тэмдэгт" гэсэн үгний товчлол болох pch параметрийг тохируулснаар үүнийг гүйцэтгэж болно. Түүнчлэн, “character

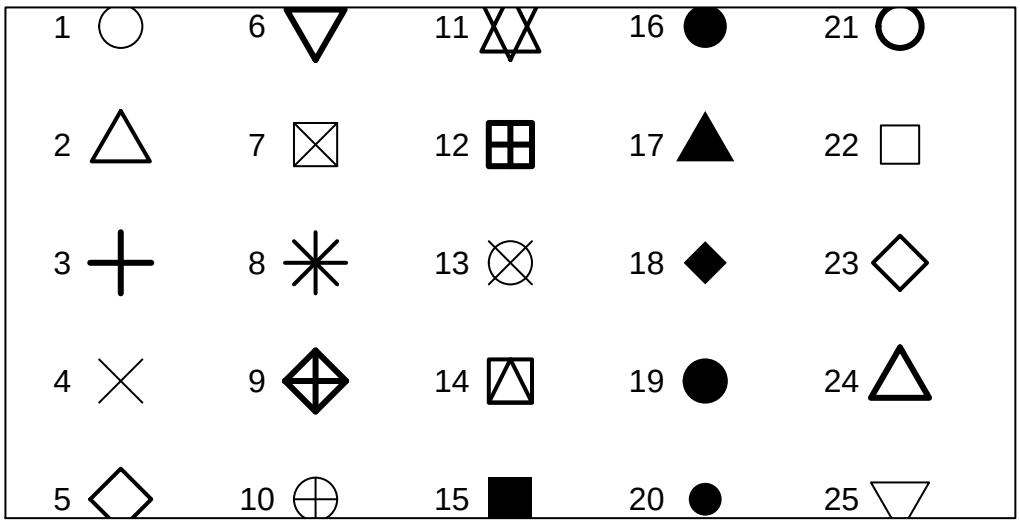
expansion” гэсэн үгний товчлол болох cex параметрийг нэгээс их утгад тохируулсан тул гурвалжин нь ердийнхөөс арай том байна.

Scatterplot: Adding points

```
plot(y = dt$weight, x = dt$age)
points(y = mean(dt$weight), x = mean(dt$age),
       col = "blue", pch = 9, cex = 3)
```



Ижилхэн график дээр янз бүрийн өнгө, хэлбэр, хэмжээтэй цэгүүдийг визуалчлахын тулд дөнгөж сая харуулсан plot ()-ийг ашиглан эхний цэгийн багцыг эхлээд зурах хэрэгтэй. Эхний графикийг зурсны дараа points() функцийг дуудаж хоёр дахь цэгийн багцыг нэмж болно. Өгөгдсөн жишээнд бид дундаж жин хэмээн у бүрэлдэхүүн хэсэг бүхий цэгийг, дундаж нас хэмээн х бүрэлдэхүүн хэсэг бүхий дан ганц нэг цэгийг нэмж оруулав. Цэгийн өнгийг цэнхэр гэж тохируулж, хэлбэрийг нь 9-р дугаарын хэлбэрээр өгсөн бөгөөд энэ нь хөндлөн огтлолцсон зураас бүхий хөндлөн тэгш өнцөгт бөгөөд анхдагч хэмжээнээс 3 дахин том хэмжээтэй байна.



Хэлбэрийн параметрийг ийм байдлаар сонгох нь жаахан нууцлаг тул таны сонгох боломжтой бүх цэгийн дүрсийг энд зурж харууллаа.

4 Summary

4.0.1 What functions did we learn?

- `mean()`, `median()`, `var()`: aggregation functions
- `rep()`: repeat input
- `factor()`: create variable of categories
- `plot()`: scatterplot
- `plot(..., type = 'l')`: line plot
- `points()`: add points to plot
- `hist()`: histogram
- `boxplot()`: draw a box-wishkers plot

Өнөөдөр бид нэгтгэх болон график үүсгэхэд хоёуланд нь хэрэгтэй олон функцийн талаар сурч мэдлээ. Тодорхой хураангуй статистикийг тооцоолохын тулд та дундаж, медиан/голч эсвэл дисперсийг тооцоолох функцийг ашиглаж болно. Хураангуй статистикийг тооцоолохыг заримдаа бас нэгтгэх гэж нэрлэдэг. rep функцийг ашиглан үндсэн утгуудын давталт болох том хэмжээний утгын векторуудыг хэрхэн хялбархан үүсгэж болохыг бид харлаа. Бид ялгаатай өөр өөр категориуд болох levels-ийг агуулсан фактор хэмээн нэрлэгддэг категориудыг R-д хэрхэн мэдэгдэх талаар сурлаа. Бид тархалтын болон шугаман графикийг үүсгэхэд plot функцийг хэрхэн ашиглаж болохыг үзлээ. Одоо байгаа график дээр өөр өнгө, хэлбэр, хэмжээтэй илүү олон цэг нэмж оруулахын тулд points функцийг ашиглаж болно. Гистограм эсвэл box-whiskers график гэх мэт бусад төрлийн графикуудыг үүсгэхийн тулд бид R бэлтгэж өгсөн hist() болон boxplot() функцуудыг ашиглаж болно.

5 Exercises

Use of the `plot` function using terrestial ecology data:

1. In Chapter 16 of Zuur et al. (2009), a study is presented analysing numbers of amphibians killed along a road in Portugal using generalised additive mixed modelling techniques. In this exercise, we use the `plot` command to visualise a segment of the data. Open the file `Amphibian\_road\_Kills.xls`, prepare a spreadsheet, and import the data into R. Download: http://highstat.com/Books/Book3/MoreData.zip
2. The variable, `TOT\_N`, is the number of dead animals at a sampling site, `OLIVE` is the number of olive groves at a sampling site, and `D Park` is the distance from each sampling point to the nearby natural park. Create a plot of `TOT\_N` versus `D\_park`. Use appropriate labels.

References

Alain Zuur, Elena N Ieno, and Erik Meesters. *A Beginner's Guide to R*. Springer Science & Business Media, 2009.