Introduction to Statistics and R

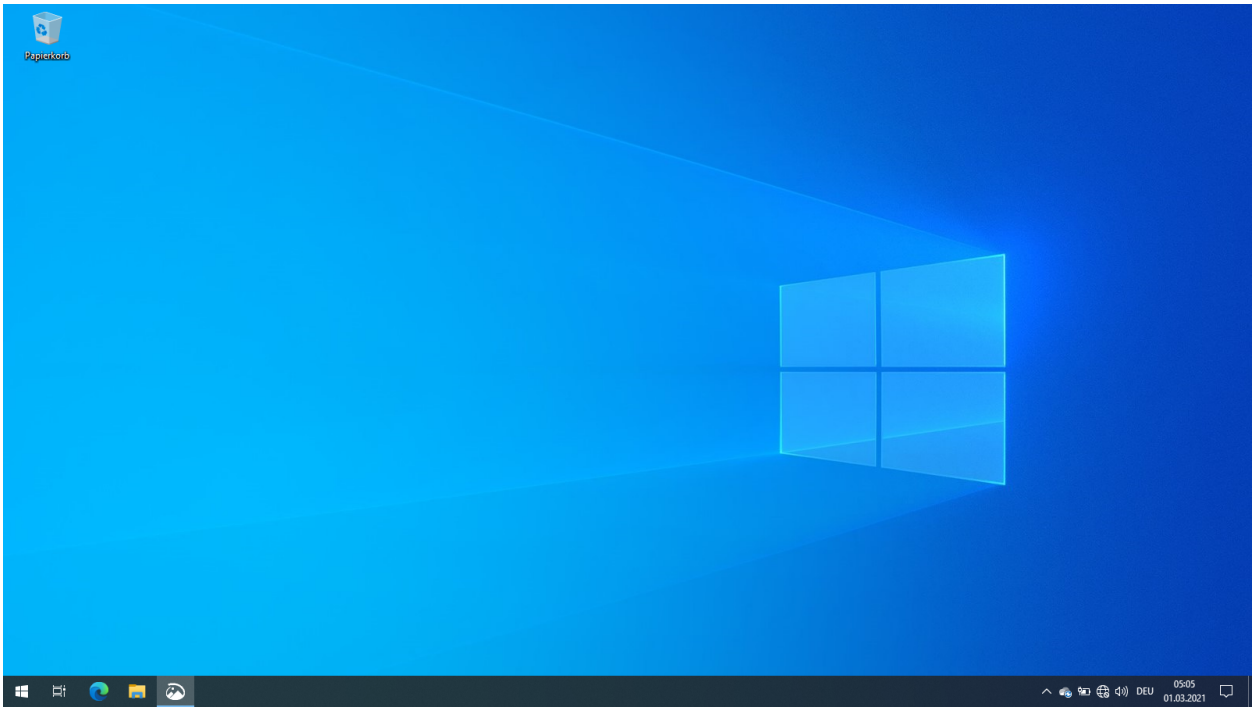
Accessing and managing subsets of data

Eric Stemmler

03.03.2021

Contents

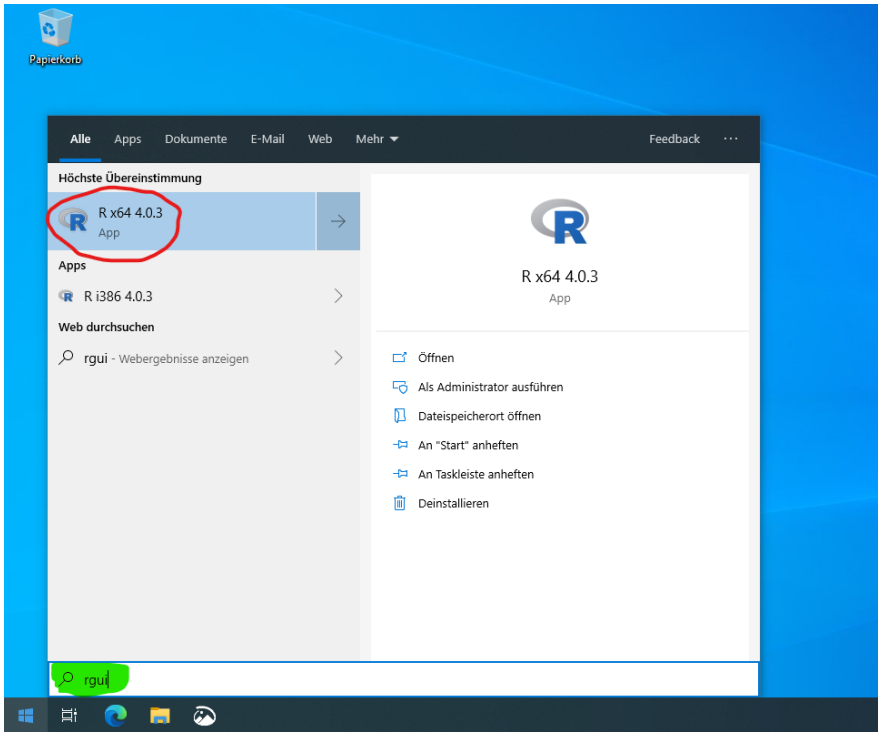
1 Recap: Using RGui



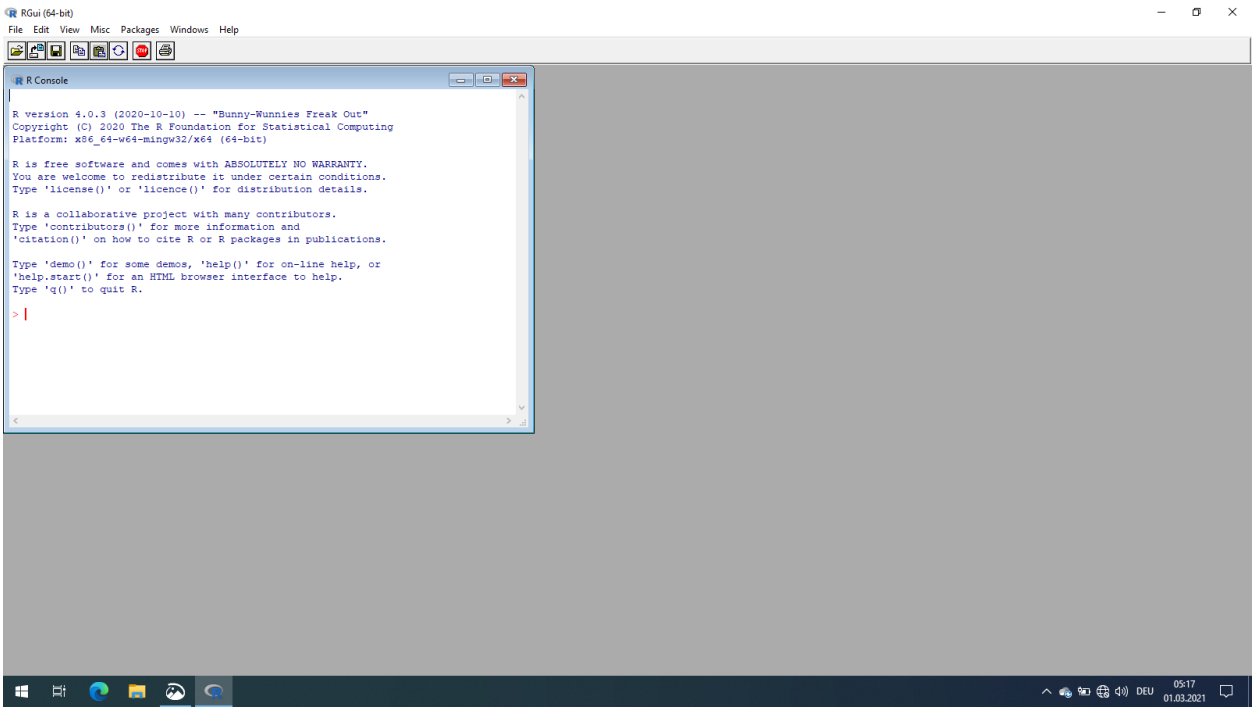
Манай статистикийн сургалтанд дахин тавтай морилно уу! Өнгөрсөн долоо хоногт бид R-ийн давтлага хичээлийг эхлүүлсэн бөгөөд тус хичээлийн үеэр R програм болон түүний график хэрэглэгчийн интерфэйсийг ашиглахад зарим бэрхшээл тулгарч байгааг анзаарсан. Энэ нь тус програмыг хэрхэн ашиглахаа сайн тайлбарлаагүйгээс болсон гэж бодож байна. Тиймээс өнөөдөр би өнгөрсөн долоо хоногийн тойм эхлэхээс өмнө үүнийг ашиглах талаар илүү нарийвчлан ярьж эргэн давтахыг хүсч байна.

Бүр эхнээс нь эхлэхийн тулд бид таны Windows-ийн desktop-оос эхлэх болно. R програмыг эхлүүлэхийн

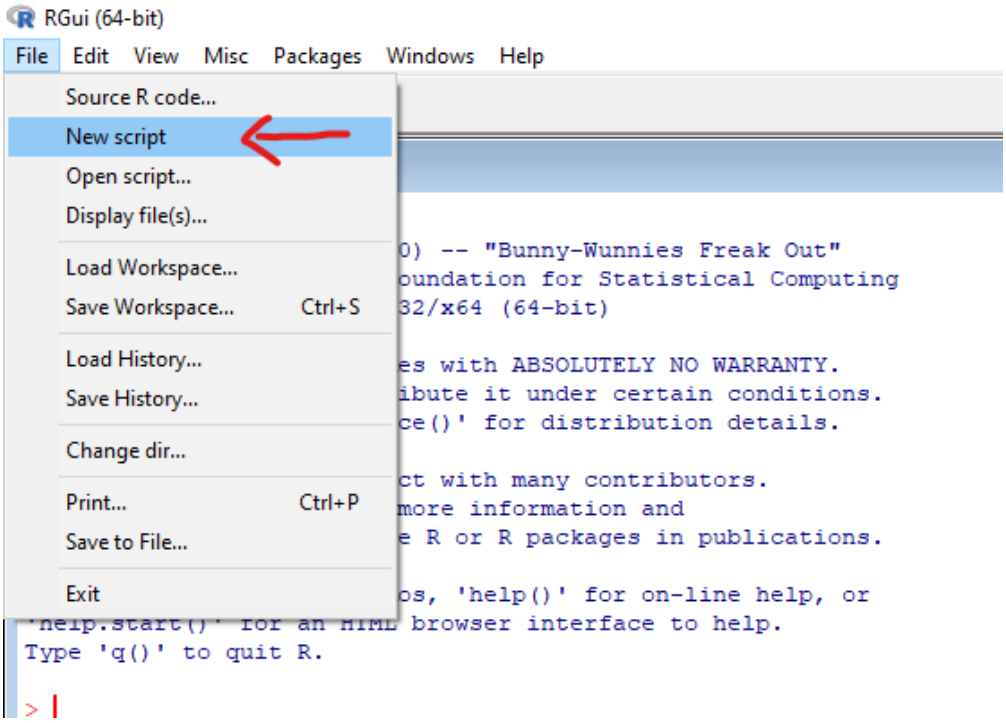
тулд ихэвчлэн дэлгэцийнхээ зүүн доод буланд байдаг цонхны дүрс дээр дарна. Мөн гар дээрх windows цонхны товчлуурыг дарахад л хангалттай



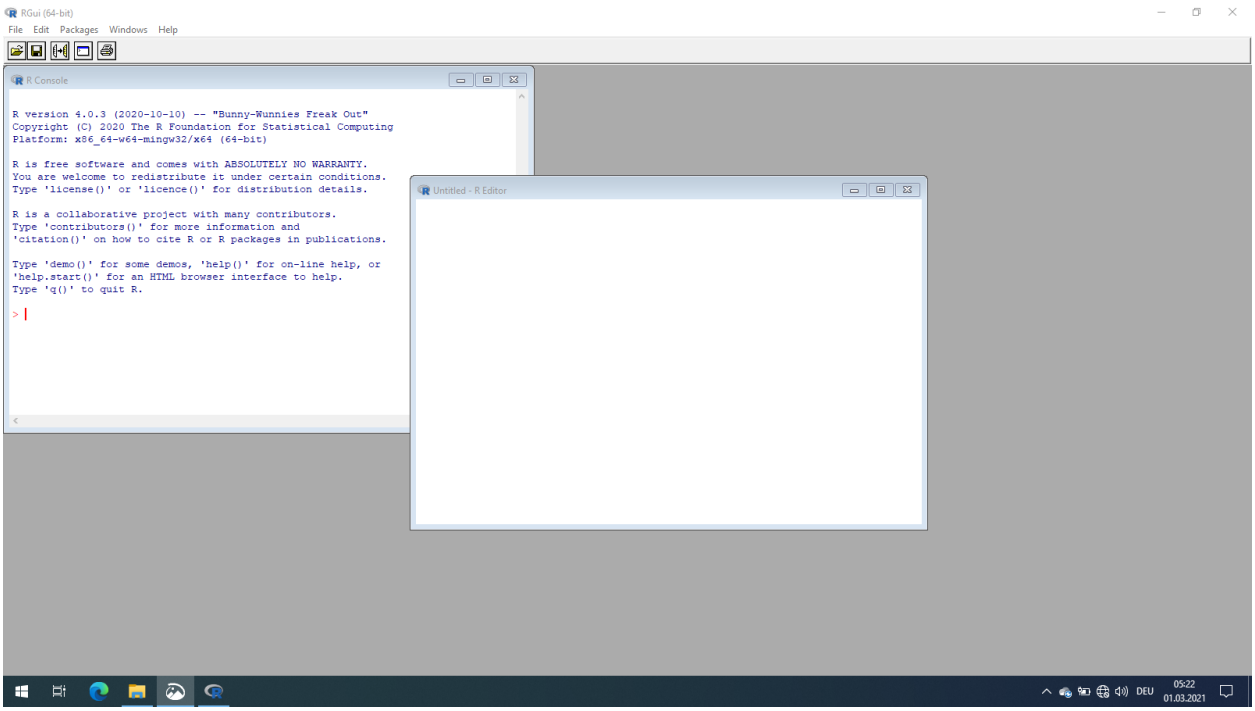
Дараа нь гар дээрээ Rgui гэж бичиж оруулна уу. Таныг бичиж эхэлмэгц Windows нь хайлтын талбарыг автоматаар нээгээд таны оруулсан мэдээлэл тэнд бичигдэнэ. Үүний дараа та enter товчийг дарах шаардлагагүй, бичиж байх үед цонх хайж эхэлдэг. Хайлтын үр дүнг бичиж дуусахад “R x64 4.0.3” гэсэн нэг сонголт байх ёстой. Түүнчлэн "R i386 4.0.3" гэсэн бас нэг сонголт байх ёстой. Аль ч тохиолдолд “4.0.3” нь таны суулгасан R-ын хувилбарын дугаар гэж нэрлэгддэг. R хувилбарыг татаж суулгахтай холбоотойгоор таны хувилбарын дугаар өөр байж болно. “X64” нь 64 бит, “i386” нь 32 битийн инсталл юм. Сүүлийнх нь зөвхөн маш эртний компьютеруудад зориулагдсан тул ашиглах шаардлагагүй юм. Та R-г эхлүүлэхийн тулд "R x64" дээр дарна уу.



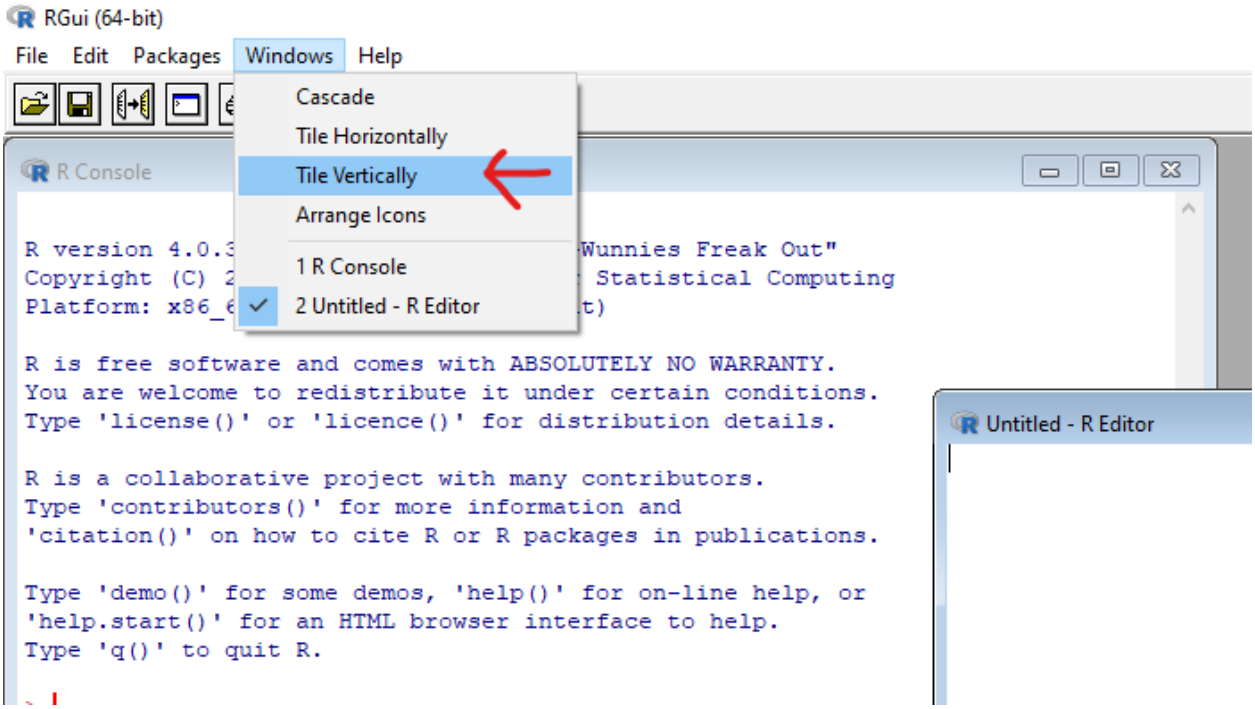
Одоо дэлгэцэн дээр R консолыг харуулсан ганц нээлттэй цонх бүхий R график хэрэглэгчийн интерфэйс нь ажиллаж эхлээд байна. Бид R-ийг эхэлсэнийг тэнд бичигдсэн "greeting" текст, хувилбарын дугаар болон харгалзах нэрийг хараад энэ нь консол гэдгийг таньж болно. Бидний програмчлалыг R дээр хийх 2 арга байдгийг санаж байгаа байх. 1-р арга нь бид консол руу R командыг оруулж болно. Ингэснээр “R interpreter” /"R тайлбарлагч" гэж нэрлэгддэг хэрэглэгчийн график интерфейсийн ард байгаа R програм нь ганц командыг уншаад шууд гүйцэтгэдэг бөгөөд ямар ч үр дүнг буцаана гаргадаг. Энэ нь өөр өөр команд хэрхэн ажилладагийг судлахад ашигтай арга байж болох юм. Гэхдээ эрдэмтэд бид өөрсдийн ажлыг дахин давтагдах чадвартай байгаасай гэж хүсч байдаг тул бусад эрдэмтэн судлаачдад үр дүнгээ зүгээр л танилцуулахын оронд статистик дүн шинжилгээнд хийсэн алхамуудаа ойлгуулахыг хүсч байдаг. Мөн R-д бид R- script бичих замаар статистик дүн шинжилгээ хийх гэсэн 2 дахь аргыг ашиглаж болно. R-script нь дараа дахин гүйцэтгэгдэх болон хадгалагдах боломжтой R-ын командуудыг хадгалдаг энгийн текст файл юм. Та R дээр дүн шинжилгээ хийж ажиллахдаа болон мөн энд ажилладаг дасгалууд дээр үүнийг ашиглахыг танд зөвлөж байна. R -script-үүдийг өгөгдлийн багцын хамт имэйлээр хялбархан хуваалцах боломжтой тул та бусад хамт ажиллагсадтайгаа ижил статистик дүн шинжилгээ хийх эсвэл үр дүнг харьцуулах гэх мэт зүйл дээр хамтран ажиллах боломжтой юм.



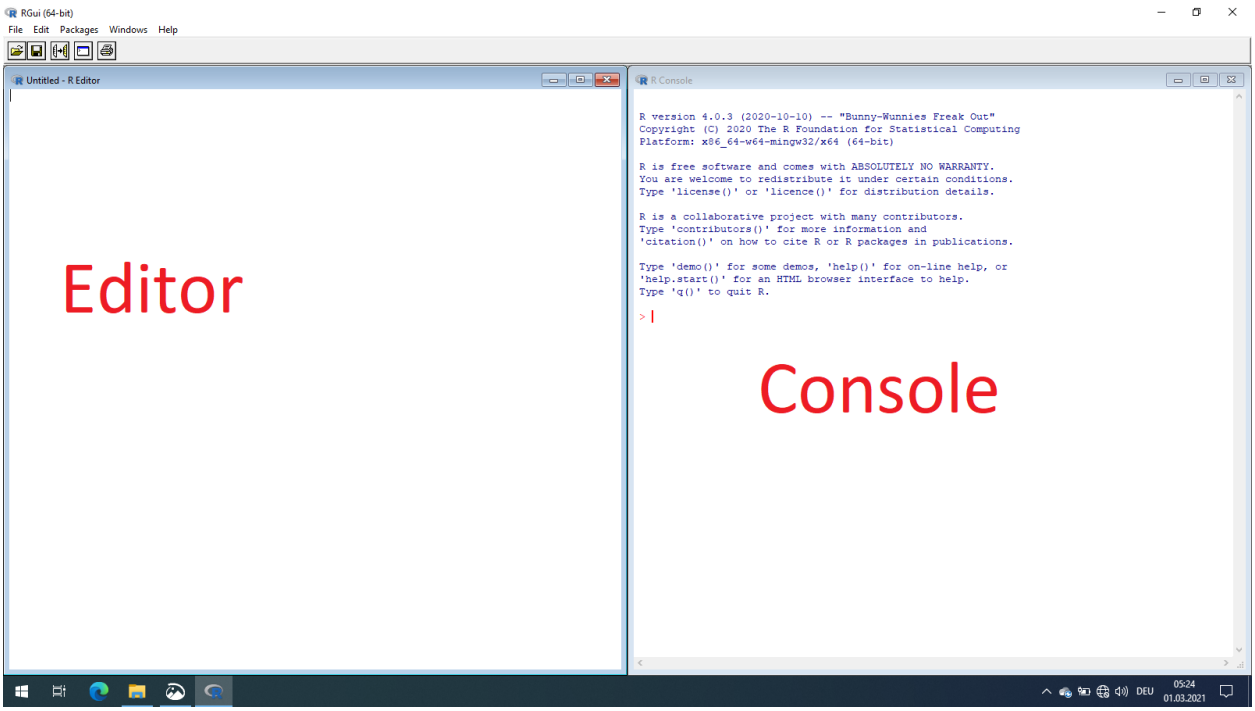
Шинэ R-script нээхийн тулд дээд цэсний мөрөнд байгаа File цэсний New script дээр дарна.



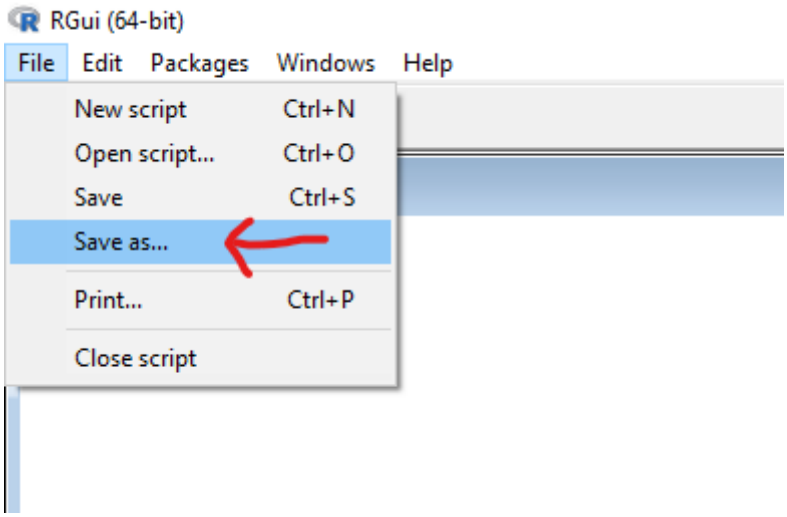
Энэ нь шинэ цонхонд шинэ R- script нээх болно



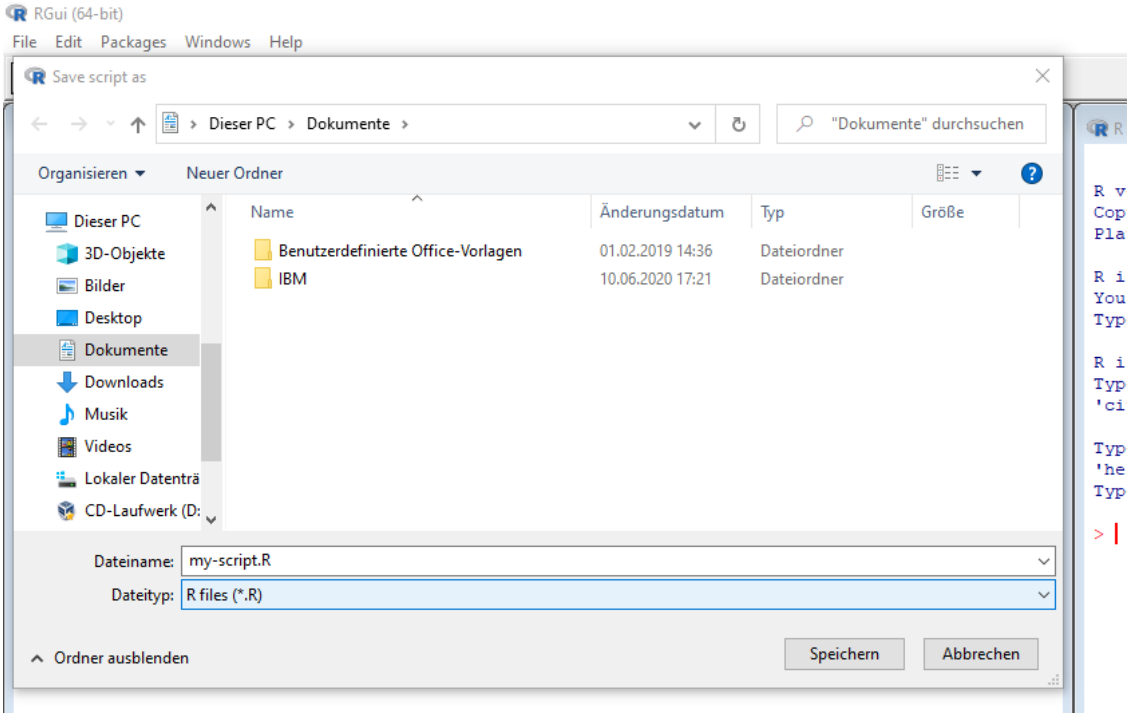
Эдгээр хоёр цонхыг зэрэгцээ автоматаар сайхан байрлуулахын тулд та Windows Tile Vertically эсвэл Windows Tile Horizontally дээр дарж сонгож болно. Гэхдээ энэ бол та бүхэнд зориулсан зөвлөмж юм. Би зүгээр л дэлгэцийг цэгцтэй зохион байгуулах дуртай.



Энэ слайд дээр бид зүүн гар талд нь R- script editor, баруун гар талд нь консол байрласан байгаа 2 цонхыг харж байна.

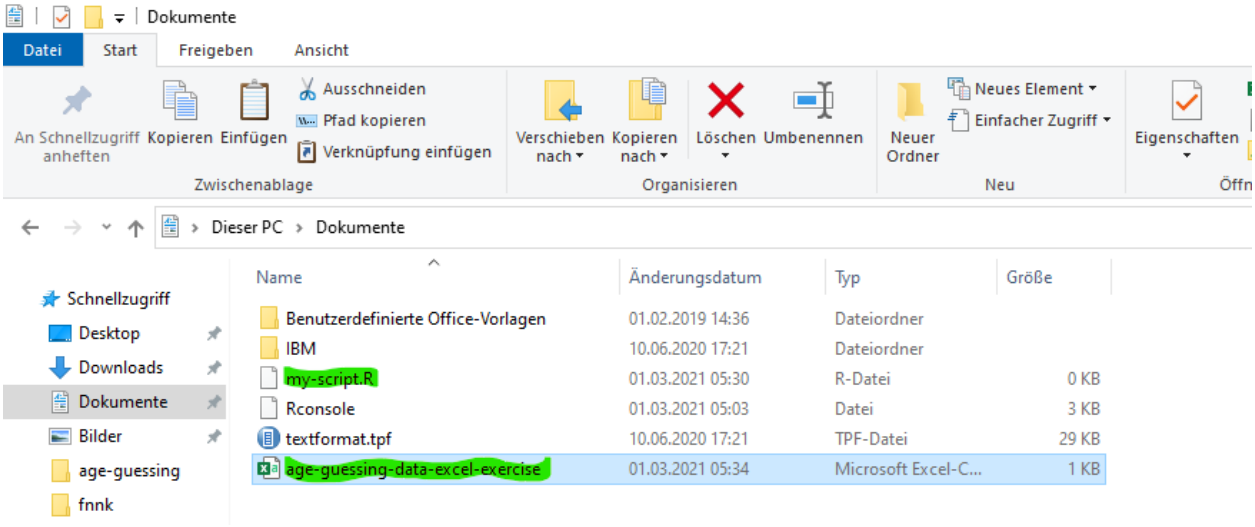


Бид R-script-ээ хараахан хадгалаагүй байна. Editor-ийн гарчиг дээр “Untitled” буюу "Нэргүй" гэж байгаа тул дүн шинжилгээндээ нэр өгье. Шинэ файлын хэлбэрээр хадгалахын тулд File цэсний Save As дээр дарна уу. Хэрэв та өөрийн R-script-ийг анх удаа хадгалж байгаа бол File цэсний Save дээр дарах ба R автоматаар хаана хадгалахыг асууна. Эсвэл та компьютерийн гарынхаа Ctrl + S товчлуур дээр дарж болно

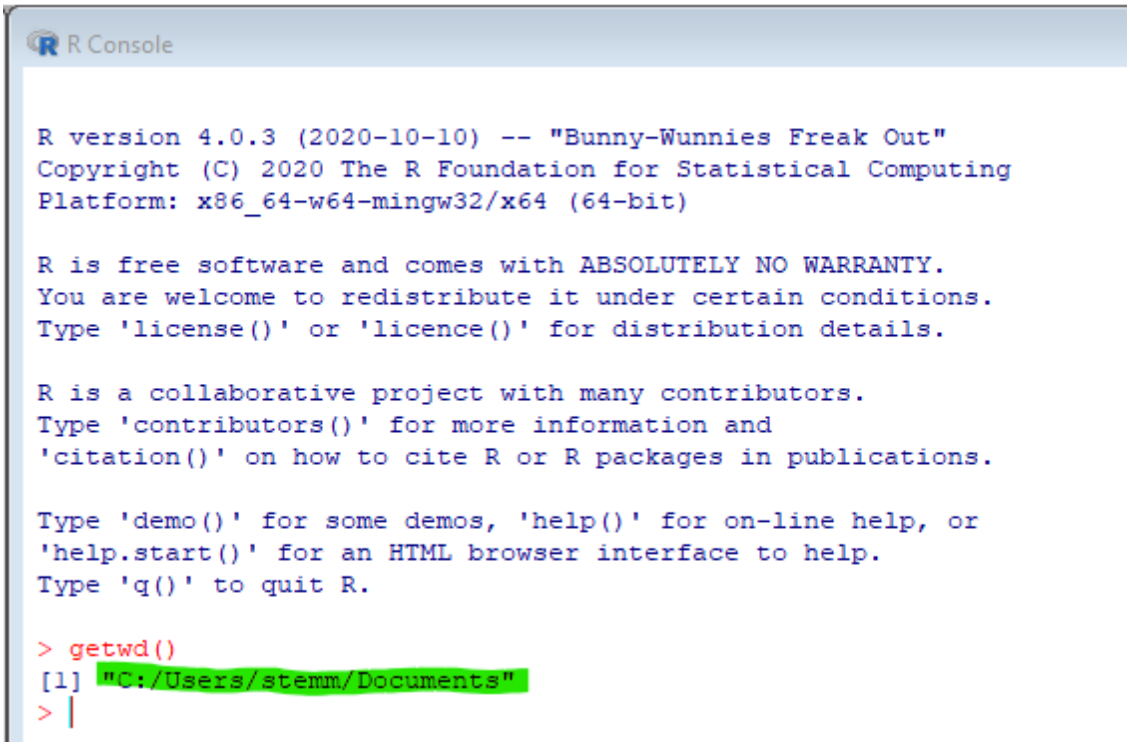


Одоо бид R-script-ийнхээ нэрийг сонгоно. Би өөрийнхийгөө “my-script. R”гэж нэрлэсэн байгаа. Та файлынхаа нэрийг “. R” -аар дуусгах ёстойг анхаарна уу. Эс тэгвээс R-script гэж танигддаггүй. Мөн R нь таны хэрэглэгчийн баримт бичгийн хавтсыг нээж байгааг анзааараарай. Яагаад ингэж байгааг бид удахгүй мэдэх болно. Хялбар байх үүднээс өгөгдөл болон R-script-ээ хадгалахдаа энэ хавтсыг ашиглахыг зөвлөж байна. Хэрэв та өөр хавтас сонговол та өгөгдлийнхөө файлын бүтэн замыг олох хэрэгтэй болно. Хамгийн сүүлд нь мэдээж R програм руу буцахын тулд save товч дээр дарна.

Одоо бид R-script-ээ хадгалсан тул editor-ийн нэр нь бидний хадгалагдсан script-ийн нэр, байршлыг унших болно.

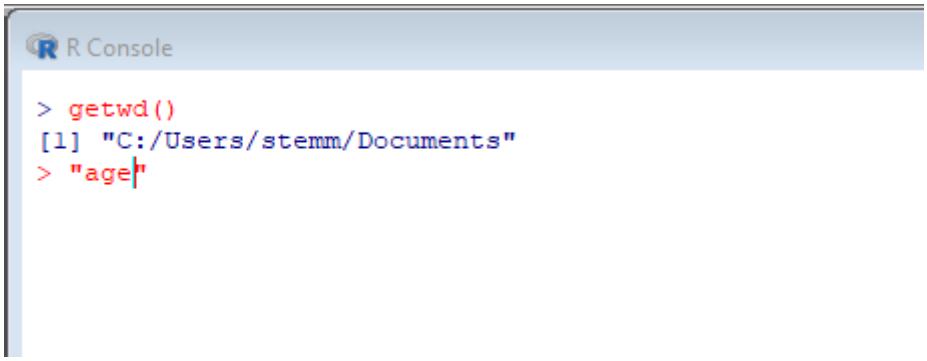


Та Windows File Explorer программаа нээгээд R-script хадгалагдсан хавтсыг нээж болно.Таны харж байгаачлан яг энэ хавтсанд би нас таах CSV файлыг хуулсан байгаа. Хэрэв би энэ хавтсанд CSV файлаа хуулчих юм бол R дээр бүтэн замын оронд зүгээр л файлын нэрийг бичиж оруулаад шууд унших боломжтой юм.



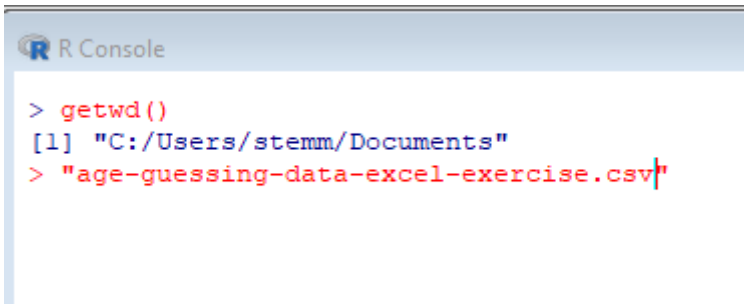
R-д буцаж ороод R-ийн ""“working directory”/”ажлын лавлах" гэснийг шалгаж болно. “working directory” нь таны компьютер дээрх “R харж байгаа” хавтас юм. Та getwd () функцийг ашиглан “working directory”

гэж юу болохыг шалгаж болно. Таны харж байгаачлан R-ийн “working directory” нь R-script-ээ хадгалсан мөн насыг таах өгөгдөлийн CSV файл байрлаж байгаа хавтастай ижилхэн хавтас юм.

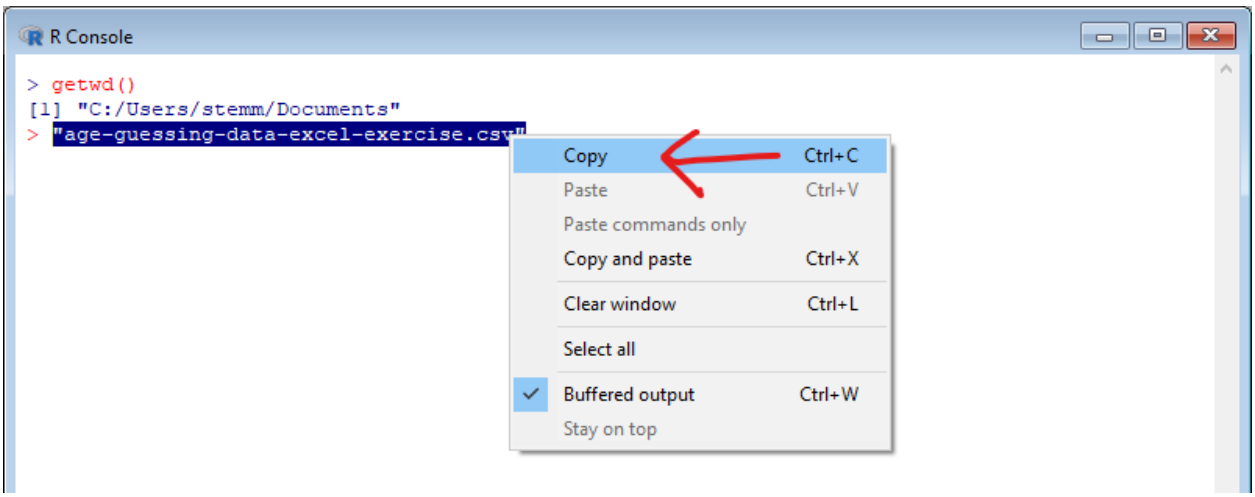


Press ⇆ x2 to complete automatically

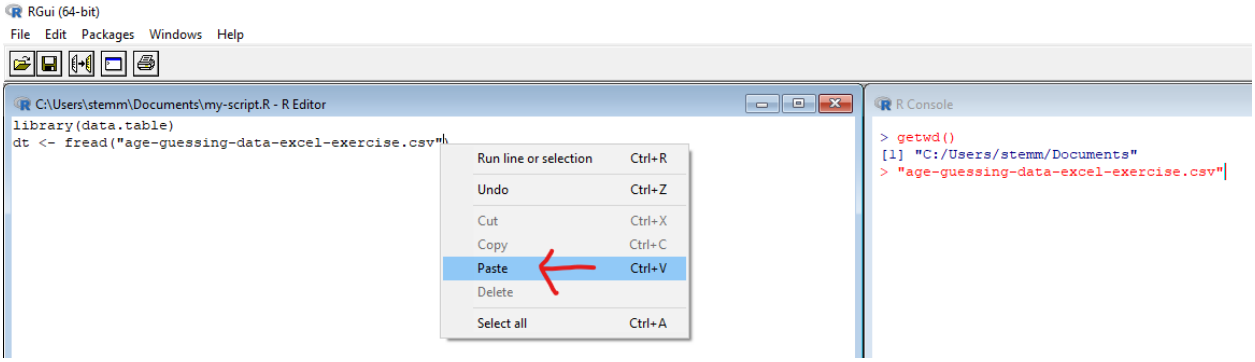
“working directory” -д уншихыг хүссэн хадгалсан файлуудтай байх нь файлын нэрийг бүтэн замгүйгээр бичихэд л давуу талтай юм. Түүнчлэн, энэ нь R-ийн консолын “auto-completion”/ "автоматаар бөглөх" онцлогийг ашиглах боломжийг бидэнд олгодог. Хэрэв та хоёр цэгтэй хаалтанд тэмдэгт мөр/файлын нэр/-ийг оруулж, дараа нь компьютерийнхээ гар дээр байгаа TAB товчлуурыг хоёр удаа дарвал R нь working directory дотроос тохирох файлын нэрийг тааруулж хайж олох болно. Энэ арга нь маш их хэрэгтэй бөгөөд бид файлын нэрийг яг бүрэн санах шаардлагагүй, зөвхөн эхний хэдэн тэмдэгтийг л санахад болно.



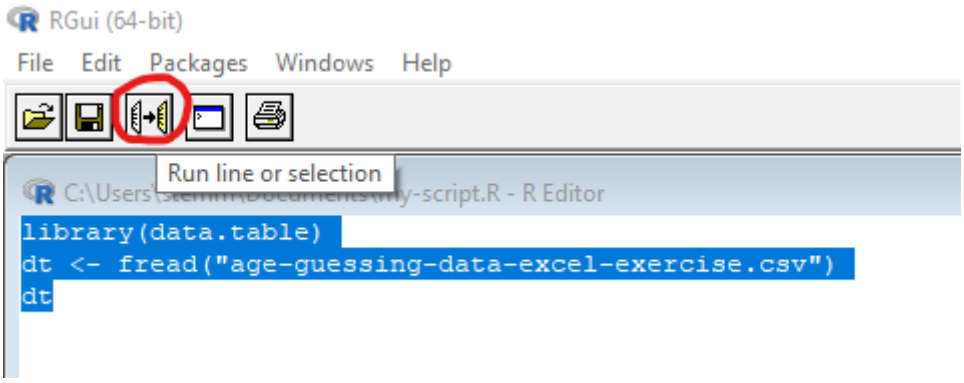
Таны харж байгаачлан энэ слайдан дээр R нь миний хайж байгаа файлын нэрийг автоматаар бөглөсөн байна.



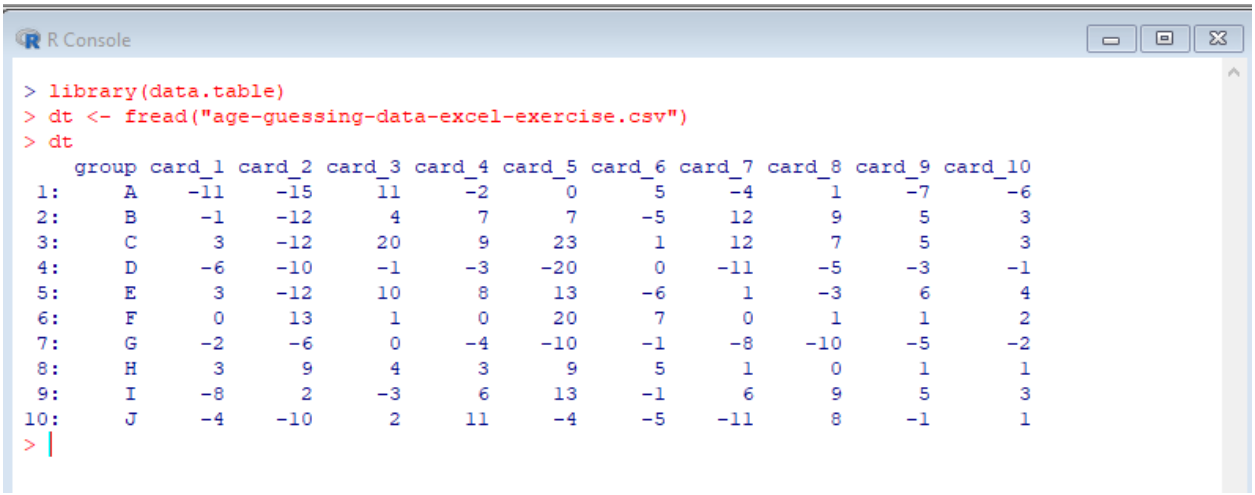
Харамсалтай нь дээрх “auto-completion”/ "автоматаар бөглөх" онцлог нь зөвхөн R-ийн консолын хувьд боломжтой бөгөөд editor -ийн хувьд боломжгүй юм. Editor дээр TAB товчийг дарахад таб буюу зай л оруулна. Гэсэн хэдий ч, бид одоо консолоос текстийг хуулж аваад Editor-т хэрэгтэй байгаа газраа paste /буулгах боломжтой юм.



Энэ тохиолдолд өнгөрсөн долоо хоногийн жишээг ашиглая. Бид уншихыг хүсч буй файлдаа параметрийн файлын set бүхий fread () функцийг дуудая. Бид өгөгдлийн файлынхаа нэрийг оруулахын тулд Editor дээр хулганы баруун товчийг дараад "Paste" товчийг дарахад л хангалттай.



Бид ийнхүү өнгөрсөн долоо хоногийн script-ээ хийж дуусгалаа. Ctrl + R товчийг дарж курсорын доор байгаа мөрөн дэх командыг гүйцэтгэх боломжтойг бид аль хэдийн сурч мэдсэн байгаа. Гэхдээ script -ийг бүхэлд нь нэг дор, өөрөөр хэлбэл script дэх мөр бүрийг бичигдсэн дарааллаар нь гүйцэтгэж болно. Үүнийг хийхийн тулд бид script -ийн бүх командыг Ctrl + A товчийг дарж идэвхжүүлээд, дараа нь слайд дээр тэмдэглэгдсэн R- script editor-ийн дээд талд байрлах "Run line or selection" гэсэн тайлбар бүхий жижиг дүрс дээр дарна.



Хэрэв дээрх бүх үйлдэл хийгдсэн бол editor-оос ирсэн гурван командыг консол дээр шалгаснаар нэг дор гүйцэтгэгдсэн болохыг харж болно. Энэ 3 командын сүүлийн команд нь нас таамаглах өгөгдлийг агуулсан data.table/өгөгдлийн хүснэгт/-ийг агуулсан dt хувьсагчийн агуулгыг хэвлэх явдал байсан.

2 Recap: Working with data sets

- Prepare data set: from Excel (`*.xlsx`) to CSV (`*.csv`)
- Read-in data set: `fread(file = ...)`
- Analyse, modify, plot data sets
- Save modified data set: `fwrite(x=..., file=..., sep=',')`

Өнгөрсөн долоо хоногт бид CSV файлын форматыг ашиглан өгөгдлийг R-руу хэрхэн унших талаар авч үзсэн байгаа. Microsoft Excel файлуудаа CSV файлын форматтай болгон өөрчлөхийн тулд хэрхэн ашиглаж болох талаар ярилцсан. CSV файлыг үүсгэсний дараа би та бүхэнд өгөгдлийг R-д унших хоёр аргыг харуулсан байгаа. R-ийн багцын data.table-ээс fread () функцийг ашиглан өгөгдлийг унших болон data.table гэж нэрлэгддэг өгөгдлийн төрөлд хадгалж болно. Нэгэнт өгөгдлийг data.table-ээр уншаад хадгаласан бол бид өгөгдлөө өөрчлөх, дүн шинжилгээ хийх эсвэл визуалчлах/дүрслэх боломжтой болж байгаа юм. Эцэст нь бид мөн түүнчлэн fwrite() функцийг ашиглан өөрчилсөн өгөгдлөө файлд хадгалах боломжтой юм.

3 Accessing variables from a data.table

3.0.1 Column Access - $

Using $ to access a column

```
library(data.table)
dt <- fread("age-guessing-data-excel-exercise.csv")
dt$group
```

```
##  [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J"
```

```
dt$card_1
```

```
##  [1] -11  -1   3  -6   3   0  -2   3  -8  -4
```

R-console: Press ⇆ behind $ to list all available column names

Одоогийн байдлаар бид data.table объектыг хэрхэн яаж хэвлэх, түүнд хэрхэн багана нэмж оруулахыг л үзсэн байгаа. Слайд дээр харуулсан кодонд бид data.table-д зориулж багцыг дахин ачааллаад data.table

объектыг өөрийн өгөгдлийн багц дээр уншиж үүсгээд үр дүнг dt нэртэй хувьсагч дотор хадгална. Хэрэв бид dt-ээс нэг багана хэвлэх эсвэл ашиглахыг хүсч байвал $ тэмдэгийг ашигладаг бөгөөд үүнийг баганад хандах оператор гэж нэрлэдэг. Энэ жишээнд би эхлээд group баганын утгуудыг хэвлээд дараа нь карт_1 баганы эхний гэрэл зургийн таамаглалын алдааны утгуудыг хэвлэж байна. Хэрэв бид хандахыг хүсч буй баганы нэр дээр эргэлзэж байгаа бол консол дээр “auto-completion”/ "автоматаар бөглөх" аргыг ашиглах боломжтой юм.

3.0.2 Selecting multiple columns

We can also use [] to access columns

```
dt[, c("group", "card_1")]
```

```
##     group card_1
##  1:     A    -11
##  2:     B     -1
##  3:     C      3
##  4:     D     -6
##  5:     E      3
##  6:     F      0
##  7:     G     -2
##  8:     H      3
##  9:     I     -8
## 10:     J     -4
```

Баганад хандах $ операторыг ашиглах нь дан ганц нэг баганад хандах боломжийг олгодог. Хэрэв бид олон баганад хандахыг хүсвэл dt-ийн дөрвөлжин хаалт [] операторыг ашиглах шаардлагатай. Бид дөрвөлжин хаалтыг оруулаад, хаалтан дотроо ганц таслал бичээд дараа нь бидний мэддэг болсон c() функцийг ашиглан баганууудын нэрийг тэмдэгт вектор байдлаар бичиж өгнө.

. . . or a combination of both . . .

```
dt[, c("group", "card_1")]$group
```

```
##  [1] "A" "B" "C" "D" "E" "F" "G" "H" "I" "J"
```

Ийм байдлаар data.table-ээс баганад хандах нь үнэн хэрэгтээ зөвхөн заасан баганууудыг агуулсан шинэ data.table обьектийг гаргаж ирдэг. Сонирхолтой нь бид слайд дээр харуулсанчлан багана хандалтын $ операторыг шууд ард нь дахин ашиглаж болдог.

4 Accessing subsets from data.table

Subset: Selection of rows by *row number*

```
# first row
dt[1]
```

```
##    group card_1 card_2 card_3 card_4 card_5 card_6
## 1:     A    -11    -15     11     -2      0      5
##    card_7 card_8 card_9 card_10
## 1:     -4      1     -7      -6
```

Бид баганууддад хэрхэн хандахыг үзлээ. Гэхдээ бид бас тодорхой мөрүүдэд хандах боломжтой юм. Үүнийг бид data.table-ийн "subsetting" гэж нэрлэдэг. Үүний тулд бид хандах оператор дөрвөлжин хаалтыг дахин ашигладаг. Гэхдээ энэ удаа бид таслалыг бичихгүй. Энгийн дэд багцын хувьд зөвхөн эхний мөрөнд хандах, хэвлэх үйлдлийг хялбархан хийдэг. Ердөө л хандах мөрийн дугаараа л зааж өгнө.

Subset: Selection of rows by *row number*

```
# first three rows
dt[1:3]
```

```
##    group card_1 card_2 card_3 card_4 card_5 card_6
## 1:     A    -11    -15     11     -2      0      5
## 2:     B     -1    -12      4      7      7     -5
## 3:     C      3    -12     20      9     23      1
##    card_7 card_8 card_9 card_10
## 1:     -4      1     -7      -6
## 2:     12      9      5       3
## 3:     12      7      5       3
```

Олон мөрөнд хандахын тулд бид ижил зарчмыг баримтална. Бид мөрүүдийн дугаарыг зааж өгнө. Мөрийн дугааруудыг үргэлжилсэн тасралтгүй байдлаар оруулах шаардлагагүй. Мөрийн дугаарууд гарцаагүй байгаа тохиолдолд бид бүхэл тоонуудын ямар ч векторыг бэлтгэж болно.

```
# all but the row 1 to 5
dt[-(1:5)]
```

```
##    group card_1 card_2 card_3 card_4 card_5 card_6
## 1:     F      0     13      1      0     20      7
## 2:     G     -2     -6      0     -4    -10     -1
## 3:     H      3      9      4      3      9      5
## 4:     I     -8      2     -3      6     13     -1
## 5:     J     -4    -10      2     11     -4     -5
##    card_7 card_8 card_9 card_10
## 1:      0      1      1       2
## 2:     -8    -10     -5      -2
## 3:      1      0      1       1
## 4:      6      9      5       3
## 5:    -11      8     -1       1
```

Гэсэн хэдий ч гажих зүйл байдаг ба энэ нь сөрөг тоонууд юм. Мэдээжийн хэрэг сөрөг мөрний дугаар гэж байхгүй, гэхдээ энэ нь юу гэсэн үг вэ гэхээр сөрөг тэмдгийг мөрний дугаарын өмнө бичсэнээр бид dt обьектдоо заасан сөрөг тэмдэгтэй мөрийн дугаартай мөрнөөс бусад бүх мөрийг хэвлэхийг хэлнэ. Тиймээс слайд дээрх хоёрдахь жишээ нь эхний 5 мөргүйгээр өгөгдлийн хүснэгтийг харуулж байна

Subset: Selection of rows by *variable value*

```
# Only rows where group equals "A" or "D"
dt[group == "A" | group == "D"]
```

```
##    group card_1 card_2 card_3 card_4 card_5 card_6
## 1:     A    -11    -15     11     -2      0      5
## 2:     D     -6    -10     -1     -3    -20      0
##    card_7 card_8 card_9 card_10
## 1:     -4      1     -7      -6
## 2:    -11     -5     -3      -1
```

Symbol | is a logical “or”: group can be “A” or “D”

Мөрүүдийн дугаарыг зааж өгөх нь нэг их хэрэгцээтэй биш байж болох юм. Учир нь зөв мөрүүдийг сонгохын тулд мөрүүдийн дугаарууд яг таг мэдэж байх ёстой бөгөөд нэмээд хэлэхэд мөрүүдийн дарааллаас хамаардаг. Илүү практик арга бол хувьсагч утгууд дээр үндэслэн дэд багцуудыг зааж өгөх явдал юм. Слайд дээрх эхний жишээн дээр бид тодорхой A, D утга бүхий хувьсагч group-ийн мөрүүдийг сонгож байна. Бид мөр сонгох нэг биш хоёр хэмжүүрийг зааж өгч байгаа тул логик эсвэл boolean операторыг ашиглан тэдгээрийг нэгтгэх шаардлагатай юм. R-д boolean операторуудыг логик операторууд гэж нэрлэдэг боловч бодит ялгаа байдаггүй. Хоолой гэж нэрлэгддэг | тэмдэг нь boolean

OR/ЭСВЭЛ гэсэн утгатай. Бодит байдал дээр үүнийг дараахь байдлаар уншиж болно: “A эсвэл D утгатай бүлгийн мөрүүдийг сонгоно уу”. Шулуухан хэлэхэд энэ мэдэгдэл нь “A эсвэл D эсвэл хоёуланг ийнх нь утга бүхий group-ийн мөрүүдийг сонгоно уу” гэсэн үг юм. Хэдий тийм ч бид зөвхөн нэг хувьсагчийг хэлж байгаа тул энэ хувьсагч хоёр утгыг зэрэг агуулж чадахгүй.

Subset: Selection of rows by *variable value*

```
# Only rows where the absolute value
# of error is larger than 5
dt[abs(card_1) > 5]
```

```
##    group card_1 card_2 card_3 card_4 card_5 card_6
## 1:     A    -11    -15     11     -2      0      5
## 2:     D     -6    -10     -1     -3    -20      0
## 3:     I     -8      2     -3      6     13     -1
##    card_7 card_8 card_9 card_10
## 1:     -4      1     -7      -6
## 2:    -11     -5     -3      -1
## 3:      6      9      5       3
```

Үүнтэй ижил логик нь тоон хувьсагчуудад мөн хамаарна. Бид энэ жишээнд заасны дагуу абсолют утга нь 5-аас их хувьсагч карт 1-ийн бүх мөрийг сонгох хэмжүүрийг зааж өгч болно.

Subset: Selection of rows by *variable value*

```
# Only rows where group is one of "A", "B" or "C"
# and the absolute value of error for the 1st photograph
# is larger than 5
dt[group %in% c("A", "B", "C") & abs(card_1) > 5]
```

```
##    group card_1 card_2 card_3 card_4 card_5 card_6
## 1:     A    -11    -15     11     -2      0      5
##    card_7 card_8 card_9 card_10
## 1:     -4      1     -7      -6
```

- `group %in% c("A", "B", "C")` means `group == "A" | group == "B" | group == "C"`
- Symbol `&` is a logical “and”, i.e. group must be "A", "B" or "C" AND error is larger than 5

Бид хэмжүүрийг нэг удаад зөвхөн нэг хувьсагч дээр ашиглахаар хязгаарлагдахгүй. Логик эсвэл boolean операторуудаар нэгтгэгдсэн тохиолдолд дурын хувьсагч дээр хэмжүүрийн дурын хослолуудыг сонгож болно. Тохиолдож болох хамгийн муу зүйл нь бидний мөрүүдийн аль нь ч өгөгдсөн хэмжүүрт нийцэхгүй байх явдал бөгөөд үр дүнд нь 0 мөр бүхий data.table гарч ирнэ. Энэ жишээнд логик операторын хэрэглээг дахин харуулахын тулд, boolean AND гэсэн & операторыг ашиглан хэмжүүрүүдийг нэгтгэж байна. Эхний хэмжүүрт бүлэг нь A, B, эсвэл C байх ёстой гэсэн бол хоёр дахь хэмжүүрт card_1-ийн таамаглалын алдааны үнэмлэхүй утга 5-аас их байх ёстой гэж заасан байна. boolean AND/ &-ийг ашиглаж байгаагийн учир нь бид энд хоёр хэмжүүрт хоёуланд нь тохирч байгаа мөрүүдийг сонгож байгаа юм.

Subset: Selection of rows by *variable value*

```
# same as above, but using logical OR instead of AND
dt[group %in% c("A", "B", "C") | abs(card_1) > 5]
```

```
##    group card_1 card_2 card_3 card_4 card_5 card_6
## 1:     A    -11    -15     11     -2      0      5
## 2:     B     -1    -12      4      7      7     -5
## 3:     C      3    -12     20      9     23      1
## 4:     D     -6    -10     -1     -3    -20      0
## 5:     I     -8      2     -3      6     13     -1
```

```
##    card_7 card_8 card_9 card_10
## 1:     -4      1     -7      -6
## 2:     12      9      5       3
## 3:     12      7      5       3
## 4:    -11     -5     -3      -1
## 5:      6      9      5       3
```

Логик OR эсвэл логик AND-ийн ялгааг харуулахын тулд логик AND-ийг логик OR-оор орлуулбал юу болохыг шалгаж үзье. Үр дүн нь эхний ба хоёр дахь хэмжүүрийг хангасан мөрүүд гүйцэтгэгдэнэ. Гэхдээ зөвхөн нэг хэмжүүрийг л хангасан тохиолдолд аль хэдийн хангалттай юм. Энэ нь илүү олон мөрийг гаргаж ирэх ёстой.

5 Sorting

To sort means to change the order of rows

Sorting: ascending vs descending

```
# ascending
dt[order(card_2)]
```

```
##     group card_1 card_2 card_3 card_4 card_5 card_6
##  1:     A    -11    -15     11     -2      0      5
##  2:     B     -1    -12      4      7      7     -5
##  3:     C      3    -12     20      9     23      1
##  4:     E      3    -12     10      8     13     -6
##  5:     D     -6    -10     -1     -3    -20      0
##  6:     J     -4    -10      2     11     -4     -5
##  7:     G     -2     -6      0     -4    -10     -1
##  8:     I     -8      2     -3      6     13     -1
##  9:     H      3      9      4      3      9      5
## 10:     F      0     13      1      0     20      7
##     card_7 card_8 card_9 card_10
##  1:     -4      1     -7      -6
##  2:     12      9      5       3
##  3:     12      7      5       3
##  4:      1     -3      6       4
##  5:    -11     -5     -3      -1
##  6:    -11      8     -1       1
##  7:     -8    -10     -5      -2
##  8:      6      9      5       3
##  9:      1      0      1       1
## 10:      0      1      1       2
```

Зарим тохиолдолд бид data.table-ээс мөрүүдэд хандахын оронд мөрүүдийн эрэмбэ, байрлалыг илүү сонирхдог. Жишээлбэл, хэрэв бид зарим хувьсагчийн хамгийн бага эсвэл хамгийн их утгатай мөрийг мэдэхийг хүсч байвал. Үүнийг хийх нэг арга бол data.table дэх мөрүүдийн дарааллыг нэг буюу хэд хэдэн хувьсагчийн утга дээр үндэслэн эрэмбэлж өөрчлөх явдал юм. Эхлээд дөрвөлжин хаалт оператороор дамжуулан мөрийн конголтод дахин хандаж, 2-рт order() функцыг ашиглан хувьсагчийн нэрээр бэлтгэх эрэмбэлэх замаар үүнийг хийж болно. Аливаа зүйлийг эрэмбэлэхэд үндсэндээ өсөх дараалал ба буурах дараалал гэсэн хоёр арга байдаг болохыг анхаарна уу. Өсөх тоон утгууд нь багаас эхэлж, мөрийн их тоогоор нэмэгдэх болно. Буурах нь эсрэгээрээ, хамгийн их утгаас эхэлж, мөрийн их тоо буурна. order() функц нь утгыг өсөх дараалаар эрэмбэлдэг.

Sorting: ascending vs descending

```
# descending
dt[order(card_2, decreasing = TRUE)]
```

```
##     group card_1 card_2 card_3 card_4 card_5 card_6
##  1:     F      0     13      1      0     20      7
##  2:     H      3      9      4      3      9      5
##  3:     I     -8      2     -3      6     13     -1
##  4:     G     -2     -6      0     -4    -10     -1
##  5:     D     -6    -10     -1     -3    -20      0
##  6:     J     -4    -10      2     11     -4     -5
##  7:     B     -1    -12      4      7      7     -5
##  8:     C      3    -12     20      9     23      1
##  9:     E      3    -12     10      8     13     -6
## 10:     A    -11    -15     11     -2      0      5
##     card_7 card_8 card_9 card_10
##  1:      0      1      1       2
##  2:      1      0      1       1
##  3:      6      9      5       3
##  4:     -8    -10     -5      -2
##  5:    -11     -5     -3      -1
##  6:    -11      8     -1       1
##  7:     12      9      5       3
##  8:     12      7      5       3
##  9:      1     -3      6       4
## 10:     -4      1     -7      -6
```

Хэрэв бид эхний мөрөнд байгаа зарим хувьсагчийн хамгийн их утгыг харахыг хүсвэл decreasing нэртэй параметр бүхий order() функцийг бэлтгэж decreasing=TRUE болгож тохируулах ёстой.

6 Summary: What functions did we learn?

- `dt\$column`: get column from data.table
- `dt[, c("columnA", "columnB")]`: get one or more columns from data.table
- `order(column)`: Sort values in `column` in ascending order
- `order(column, decreasing = TRUE)`: Sort values in `column` in decreasing order
- `A \%in\% c(B, C; D)`: shortcut for A = B or A = C or A = D
- Logical Or-operator `|`: A=="normal" | A=="small"
- Logical And-operator `\&`: `A>1 \& A<5`

Өнөөдөр сурч мэдсэн зүйлээ нэгтгэн дүгнэе. Бид баганад хандах оператор гэж нэрлэвэл илүү тохиромжтой долларын тэмдгэн оператор болон дөрвөлжин хаалт операторыг ашиглан data.table -ээс багануудад хандах талаар авч үзлээ. Шаардлагатай бол энэ хоёрыг нэгтгэж болно гэдгийг бид бас үзсэн байгаа. Дараа нь бид мөрийн хэмжүүрийг зааж өгөхөд дөрвөлжин хаалт [] операторыг хэрхэн ашиглах талаар ярилцсан. Миний дурдаагүй зүйл бол дөрвөлжин хаалт [] операторыг ашиглан мөр, баганы зэрэг сонгох боломжтой юм. Гэхдээ бид үүнийг маргааш R-ийн давтлага хичээл дээр авч үзэх болно. Дараа нь мөрүүдийн дарааллыг хэрхэн өөрчлөхийг үзүүлсэн. order() функцийг ашиглах нь эгнээний сонголтыг хийхтэй адилхан гэдгийг харлаа. Бид үр дүнгийн дараалал буурч байгаа тул decreasing гэж нэрлэгдэх параметрийг зааж өгөх замаар order() функцийн үйлдлийг өөрчилж болно. While discussing row criteria, we also introduced logical operators in R: OR and AND and how to use them for selecting rows. One special form of OR is the in-operator, which provides a handy shortcut for specifying multiple OR conditions.

7 Exercises

7.0.1 Age Guessing

Analyse the age guessing data set. For each question, write one line of R code to determine the answer.

1. Which team had the lowest absolute error for card no. 8?
2. Which photograph card was the most difficult one, i.e. the one with the highest total error

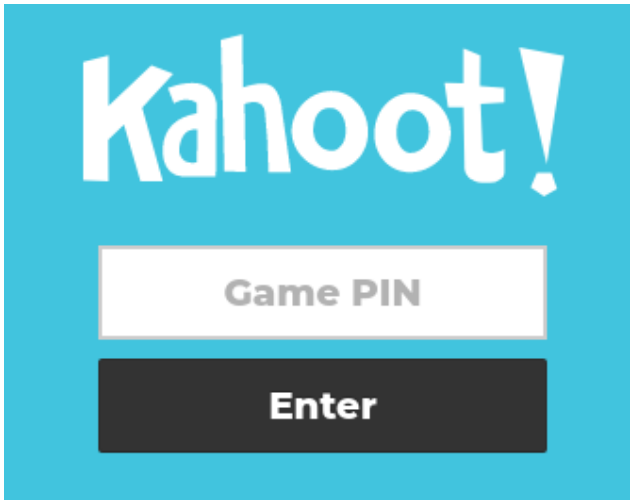
7.0.2 Bird Flue - Avian Influenza A/ H5N1

Zuur et al. (2009): The file `BirdFlu.xls` contains the annual number of confirmed cases of human Avian Influenza A/(H5N1) for several countries reported to the World Health Organization (WHO). The data were taken from the WHO website (www.who.int/en/).

Download the data set and copy file `BirdFlu.xls` from http://highstat.com/Books/Book3/MoreData.zip. Convert `BirdFlu.xls` into a csv format, ie. `birdflu.csv`. Write a R-script to answer the following questions:

1. What is the total number of bird flu cases in 2003 and in 2005?
2. Which country has had the *most* cases?
3. Which country has had the *least* bird flu deaths?

8 Quiz



- Phone or Computer: https://www.kahoot.it
- Wifi: Platinum
- Password: H2SvsH2O

References

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