

TUTORIAL: SAMPLING, WEIGHTING AND ESTIMATION DAY 1

Stefan Zins, Matthias Sand
and Jan-Philipp Kolb

GESIS - Leibniz Institute
for the Social Sciences

GESIS Summer School

Cologne, Germany

August 24th, 2015

- Open Source
 - You can work with several datasets at the same time
 - You can create your own objects, functions and packages
 - Over 5,000 packages contributed by users available on CRAN
- Rapid implementation of new (scientific) developments
- Quick development of new tools that fit the user's demand

GETTING STARTED - DOWNLOAD R

The R Project for Statistical Computing

PCA 5 vars
prcomp(x = data, cor = TRUE)

Clustering 4 groups

Group

Factor 1 [41%]

Factor 3 [19%]

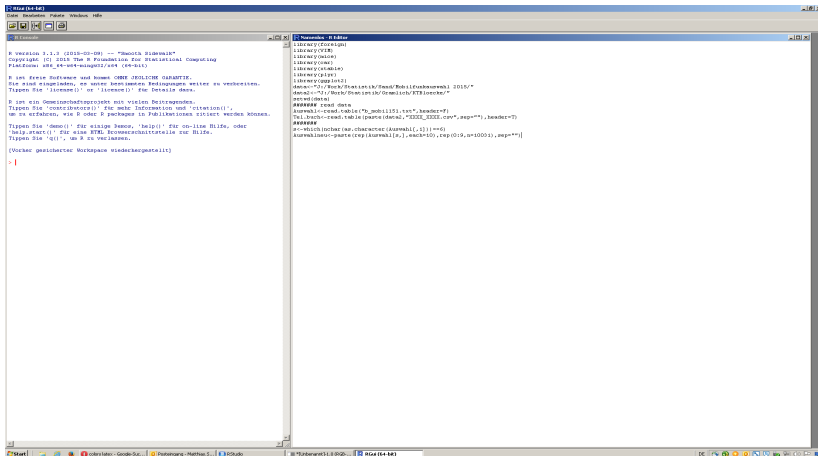
Getting Started:

- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To [download R](#), please choose your preferred [CRAN mirror](#).
- If you have questions about R like how to download and install the software, or what the license terms are, please read our [answers to frequently asked questions](#) before you send an email.

News:

- [The R Journal Vol.5/1](#) is available.
- [R version 3.0.1](#) (Good Sport) has been released on 2013-05-16.
- [R version 2.15.3](#) (Security Blanket) has been released on 2013-03-01.
- [iReR 2013](#) will take place at the University of Castilla-La Mancha, Albacete, Spain, July 10-12 2013.

<https://www.r-project.org>



The screenshot shows the RStudio graphical user interface (GUI) with two main panes. The left pane is the R console, displaying the R version (3.1.3) and copyright information. The right pane is the script editor, showing a script that loads the 'foreign' package and reads a CSV file named 'data2.csv' from a specific path. The script also includes comments in German and a loop to process the data.

```

R version 3.1.3 (2015-03-09) -- "Smooth Sailing"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R ist freie Software und kommt OHNE JEDELICHE GARANTIE.
Sie sind eingeladen, es unter bestimmten Bedingungen weiter zu verbreiten.
Tippen Sie 'license()' or 'license()' für Details dazu.

R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Informationen und 'citation()'
um zu erfahren, wie R oder R packages in Publikationen zitiert werden können.

Tippen Sie 'demo()' für einige Demos, 'help()' für on-line Hilfe, oder
'help.start()' für eine HTML Brownian-Schnittstelle zur Hilfe.
Tippen Sie 'q()' um R zu verlassen.

(Vorher gesicherter Workspace wiederhergestellt)

> |
  
```

```

# RStudio: R Editor
library(foreign)
library(CSV)
library(XML)
library(ggplot2)
library(reshape2)
library(Rcpp)
library(RcppInt)
library(RcppInt2)
library(RcppInt3)
library(RcppInt4)
library(RcppInt8)
library(RcppInt16)
library(RcppInt32)
library(RcppInt64)
library(RcppInt128)
library(RcppInt256)
library(RcppInt512)
library(RcppInt1024)
library(RcppInt2048)
library(RcppInt4096)
library(RcppInt8192)
library(RcppInt16384)
library(RcppInt32768)
library(RcppInt65536)
library(RcppInt131072)
library(RcppInt262144)
library(RcppInt524288)
library(RcppInt1048576)
library(RcppInt2097152)
library(RcppInt4194304)
library(RcppInt8388608)
library(RcppInt16777216)
library(RcppInt33554432)
library(RcppInt67108864)
library(RcppInt134217728)
library(RcppInt268435456)
library(RcppInt536870912)
library(RcppInt1073741824)
library(RcppInt2147483648)
library(RcppInt4294967296)
library(RcppInt8589934592)
library(RcppInt17179869184)
library(RcppInt34359738368)
library(RcppInt68719476736)
library(RcppInt137438953472)
library(RcppInt274877906944)
library(RcppInt549755813888)
library(RcppInt1099511627776)
library(RcppInt2199023255552)
library(RcppInt4398046511104)
library(RcppInt8796093022208)
library(RcppInt17592186044416)
library(RcppInt35184372088832)
library(RcppInt70368744177664)
library(RcppInt140737488355328)
library(RcppInt281474976710656)
library(RcppInt562949953421312)
library(RcppInt1125899906842624)
library(RcppInt2251799813685248)
library(RcppInt4503599627370496)
library(RcppInt9007199254740992)
library(RcppInt18014398509481984)
library(RcppInt36028797018963968)
library(RcppInt72057594037927936)
library(RcppInt144115188075855872)
library(RcppInt288230376151711744)
library(RcppInt576460752303423488)
library(RcppInt1152921504606846976)
library(RcppInt2305843009213693952)
library(RcppInt4611686018427387904)
library(RcppInt9223372036854775808)
library(RcppInt18446744073709551616)
library(RcppInt36893488147419103232)
library(RcppInt73786976294838206464)
library(RcppInt147573952589676412928)
library(RcppInt295147905179352825856)
library(RcppInt590295810358705651712)
library(RcppInt1180591620717411303424)
library(RcppInt2361183241434822606848)
library(RcppInt4722366482869645213696)
library(RcppInt9444732965739290427392)
library(RcppInt18889465931478580854784)
library(RcppInt37778931862957161709568)
library(RcppInt75557863725914323419136)
library(RcppInt151115727451828646838272)
library(RcppInt302231454903657293676544)
library(RcppInt604462909807314587353088)
library(RcppInt1208925819614629174706176)
library(RcppInt2417851639229258349412352)
library(RcppInt4835703278458516698824704)
library(RcppInt9671406556917033397649408)
library(RcppInt19342813113834066795298816)
library(RcppInt38685626227668133590597632)
library(RcppInt77371252455336267181195264)
library(RcppInt154742504910672534362390528)
library(RcppInt309485009821345068724781056)
library(RcppInt618970019642690137449562112)
library(RcppInt1237940039285380274899124224)
library(RcppInt2475880078570760549798248448)
library(RcppInt4951760157141521099596496896)
library(RcppInt9903520314283042199192993792)
library(RcppInt19807040628566084398385987584)
library(RcppInt39614081257132168796771975168)
library(RcppInt79228162514264337593543950336)
library(RcppInt158456325028528675187087900672)
library(RcppInt316912650057057350374175801344)
library(RcppInt633825300114114700748351602688)
library(RcppInt1267650600228229401496703205376)
library(RcppInt2535301200456458802993406410752)
library(RcppInt5070602400912917605986812821504)
library(RcppInt10141204801825835211973625643008)
library(RcppInt20282409603651670423947251286016)
library(RcppInt40564819207303340847894502572032)
library(RcppInt81129638414606681695789005144064)
library(RcppInt162259276832213363391578010288128)
library(RcppInt324518553664426726783156020576256)
library(RcppInt649037107328853453566312041152512)
library(RcppInt1298074214657706907132624082305024)
library(RcppInt2596148429315413814265248164610048)
library(RcppInt5192296858630827628530496329220096)
library(RcppInt10384593717261655257060992658440192)
library(RcppInt20769187434523310514121985316880384)
library(RcppInt41538374869046621028243970633760768)
library(RcppInt83076749738093242056487941267521536)
library(RcppInt166153499476186484112975882535043072)
library(RcppInt332306998952372968225951765070086144)
library(RcppInt664613997904745936451903530140172288)
library(RcppInt1329227995809491872903807060280344576)
library(RcppInt2658455991618983745807614120560689152)
library(RcppInt5316911983237967491615228241121378304)
library(RcppInt10633823966475934983230456482242756608)
library(RcppInt21267647932951869966460912964485513216)
library(RcppInt42535295865903739932921825928971026432)
library(RcppInt85070591731807479865843651857942052864)
library(RcppInt170141183463614959731687303715884105728)
library(RcppInt340282366927229919463374607431768211456)
library(RcppInt680564733854459838926749214863536422912)
library(RcppInt1361129467708919677853498429727072845824)
library(RcppInt2722258935417839355706996859454145691648)
library(RcppInt5444517870835678711413993718908291383296)
library(RcppInt10889035741671357422827987437816582766592)
library(RcppInt21778071483342714845655974875633165533184)
library(RcppInt43556142966685429691311949751266331066368)
library(RcppInt87112285933370859382623899502532662132736)
library(RcppInt174224571866741718765247799005065324265472)
library(RcppInt348449143733483437530495598010130648530944)
library(RcppInt696898287466966875060991196020261297061888)
library(RcppInt1393796574933933750121982392040522594123776)
library(RcppInt2787593149867867500243964784081045188247552)
library(RcppInt5575186299735735000487929568162090376495104)
library(RcppInt11150372599471470000975859136324180752990208)
library(RcppInt22300745198942940001951718272648361505980416)
library(RcppInt44601490397885880003903436545296723011960832)
library(RcppInt89202980795771760007806873090593446023921664)
library(RcppInt178405961591543520015613746181186892047843328)
library(RcppInt356811923183087040031227492362373784095686656)
library(RcppInt713623846366174080062454984724747568191373312)
library(RcppInt1427247692732348160124909969449495136382746624)
library(RcppInt2854495385464696320249819938898990272765493248)
library(RcppInt5708990770929392640499639877797980545530986496)
library(RcppInt11417981541858785280999279755595961091061972992)
library(RcppInt22835963083717570561998559511191922182123945984)
library(RcppInt45671926167435141123997119022383844364247891968)
library(RcppInt91343852334870282247994238044767688728495783936)
library(RcppInt182687704669740564495988476089535377456991567872)
library(RcppInt365375409339481128991976952179070754913983135744)
library(RcppInt730750818678962257983953904358141509827966271488)
library(RcppInt1461501637357924515967907808716283019655932542976)
library(RcppInt2923003274715849031935815617432566039311865085952)
library(RcppInt5846006549431698063871631234865132078623730171904)
library(RcppInt11692013098863396127743262469730264157247460343808)
library(RcppInt23384026197726792255486524939460528314494920687616)
library(RcppInt46768052395453584510973049878921056628989841375232)
library(RcppInt93536104790907169021946099757842113257979682750464)
library(RcppInt187072209581814338043892199515684226515959365500928)
library(RcppInt374144419163628676087784399031368453031918731001856)
library(RcppInt748288838327257352175568798062736906063837462003712)
library(RcppInt1496577676654514704351137596125473812127674924007424)
library(RcppInt2993155353309029408702275192250947624255349848014848)
library(RcppInt5986310706618058817404550384501895248510699696029696)
library(RcppInt11972621413236117634809100769003790497021399392059392)
library(RcppInt23945242826472235269618201538007580994042798784118784)
library(RcppInt47890485652944470539236403076015161988085597568237568)
library(RcppInt95780971305888941078472806152030323976171195136475136)
library(RcppInt191561942611777882156945612304060647952342390272950272)
library(RcppInt383123885223555764313891224608121295904684780545900544)
library(RcppInt766247770447111528627782449216242591809369561091801088)
library(RcppInt1532495540894223057255564898432485183618739122183602176)
library(RcppInt3064991081788446114511129796864970367237478244367204352)
library(RcppInt6129982163576892229022259593729940734474956488734408704)
library(RcppInt122599643271537844580445191874598814689499129774688177216)
library(RcppInt245199286543075689160890383749197629378998259549376354432)
library(RcppInt490398573086151378321780767498395258757996519098752708864)
library(RcppInt980797146172302756643561534996790517515993038197505417728)
library(RcppInt1961594292344605513287123069993581035031986076395010835456)
library(RcppInt3923188584689211026574246139987162070063972152790021670912)
library(RcppInt7846377169378422053148492279974324140127944305580043341824)
library(RcppInt15692754338756844106296984559948648280255888611160086683648)
library(RcppInt31385508677513688212593969119897296560511777222320173367296)
library(RcppInt62771017355027376425187938239794593121023554444640346734592)
library(RcppInt125542034710054752850375876479589186242047108889280693469184)
library(RcppInt251084069420109505700751752959178372484094217778561386938368)
library(RcppInt502168138840219011401503505918356744968188435557122773876736)
library(RcppInt1004336277680438022803007011836713489936376871114245547753472)
library(RcppInt2008672555360876045606014023673426979872753742228491095506944)
library(RcppInt4017345110721752091212028047346853959745507484456982191013888)
library(RcppInt8034690221443504182424056094693707919491014968913964382027776)
library(RcppInt16069380442887008364848112189387415838982029937827928764055552)
library(RcppInt32138760885774016729696224378774831677964059875655857528111104)
library(RcppInt64277521771548033459392448757549663355928119751311715056222208)
library(RcppInt128555043543096066918784897515099326711856239502623430112444416)
library(RcppInt257110087086192133837569795030198653423712479005246860224888832)
library(RcppInt514220174172384267675139590060397306847424958010493720449777664)
library(RcppInt1028440348344768535350279180120794613694849916020987440899555328)
library(RcppInt2056880696689537070700558360241589227389699832041974881799110656)
library(RcppInt4113761393379074141401116720483178454779399664083949763598221312)
library(RcppInt8227522786758148282802233440966356909558799328167899527196442624)
library(RcppInt16455045573516296565604466881932713819117598656335799054392885248)
library(RcppInt32910091147032593131208933763865427638235197312671598108785770496)
library(RcppInt65820182294065186262417867527730854676470394625343196217701540992)
library(RcppInt131640364588130372524835735055461709352940789250686392435403081984)
library(RcppInt263280729176260745049671470110923418705881578501372784870806163968)
library(RcppInt526561458352521490099342940221846837411763157002745569741612327936)
library(RcppInt1053122916705042980198685880443693674823526314005491139483224655872)
library(RcppInt2106245833410085960397371760887387349647052628010982278966449311744)
library(RcppInt4212491666820171920794743521774774699294105256021964557932898623488)
library(RcppInt8424983333640343841589487043549549398588210512043929115865797246976)
library(RcppInt16849966667280687683178974087099097977176421024087858231731594493952)
library(RcppInt33699933334561375366357948174198195954352842048175716463463188987904)
library(RcppInt67399866669122750732715896348396391908705684096351432926926377975808)
library(RcppInt134799733338245501465431792696792783817411368192702865853852755951616)
library(RcppInt269599466676491002930863585393585567634822736385405731707705511903232)
library(RcppInt539198933352982005861727170787171135269645472770811463415411023806464)
library(RcppInt1078397866705964011723454341574342270539290945541622926830822047612928)
library(RcppInt2156795733411928023446908683148684541078581891083245853661644095225856)
library(RcppInt4313591466823856046893817366297369082157163782166491707323288190451712)
library(RcppInt8627182933647712093787634732594738164314327564332983414646576380903424)
library(RcppInt1725436586729542418757526946518947632862865512866596682929315276180672)
library(RcppInt3450873173459084837515053893037895265725731025733193365858630552361344)
library(RcppInt6901746346918169675030107786075790531451462051466386731717261104722688)
library(RcppInt13803492693836339350060215572151581062902924102932773463434522209445376)
library(RcppInt2760698538767267870012043114430316212580584820586554692686904441889072)
library(RcppInt5521397077534535740024086228860632425161169641173109385373808883778144)
library(RcppInt11042794155069071480048172457721264850322339282346218770747617767556288)
library(RcppInt22085588310138142960096344915442529700644678564692437541495235535112576)
library(RcppInt44171176620276285920192689830885059401289357129384875082990471070225152)
library(RcppInt88342353240552571840385379661770118802578714258769750165980942140450304)
library(RcppInt176684706481105143680770759323540237605157428517539500331961884280900608)
library(RcppInt353369412962210287361541518647080475210314857035079000663923768561801216)
library(RcppInt706738825924420574723083037294160950420629714070158001327847537123602432)
library(RcppInt1413477651848841149446166074588321900841259428140316002655695074247204864)
library(RcppInt2826955303697682298892332149176643801682518856280632005311390148494409728)
library(RcppInt5653910607395364597784664298353287603365037712561264010622780296988819456)
library(RcppInt11307821214790729195569328596706575206730075425122528021245560593977738912)
library(RcppInt22615642429581458391138657193413150413460150850245056042491121187955477824)
library(RcppInt4523128485916291678227731438682630082692030170049011208498224237591095568)
library(RcppInt9046256971832583356455462877365260165384060340098022416996448475182191136)
library(RcppInt18092513943665166712910925754730520330768120680196044833992896950364382272)
library(RcppInt36185027887330333425821851509461040661536241360392089667985793900728764544)
library(RcppInt72370055774660666851643703018922081323072482720784179335971587801457529088)
library(RcppInt144740111549321333703287406037844162646144965441568358671943175602915058176)
library(RcppInt289480223098642667406574812075688325292289930883136717343886351205830116352)
library(RcppInt578960446197285334813149624151376650584579861766273434687772702411660232704)
library(RcppInt1157920892394570669626299248302753301169159723532546869375545404823320465408)
library(RcppInt2315841784789141339252598496605506602338319447065093738751090809646640930816)
library(RcppInt4631683569578282678505196993211013204676638894130187477502181619293281861632)
library(RcppInt9263367139156565357010393986422026409353277788260374955004363238586563723264)
library(RcppInt1
```

The screenshot displays the RStudio IDE with the following components:

- Source Editor:** Contains R code for data manipulation. It includes comments in German and uses functions like `unique`, `substr`, `read.table`, `aggregate`, and `order` to process data from 'Auswahl1' and 'Untersuch2'.
- Environment:** Lists loaded data objects:
 - `Auswahl1`: 12811 obs. of 1 variable
 - `Auswahl2`: 121110 obs. of 1 variable
 - `tdta`: 48 obs. of 2 variables
 - `tdta2`: 105 obs. of 2 variables
 - `Tel.buch2`: 98587 obs. of 2 variables
 - `Untersuch`: 11869 obs. of 2 variables
 - `Untersuch2`: 4376 obs. of 3 variables
 - `Untersuch2`: 610 obs. of 3 variables
- Values:** Shows the structure of variables:
 - `ab`: int [1:610] 1 2 3 4 5 6 7 8 9 10...
 - `Auswahl1neu`: Large character [100310] elements...
 - `Auswahl2neu`: Large character [121110] elements...
 - `data2`: "J:/Work/Statistik/Sand/Mobilfun..."
 - `drinne`: int [1:94213] 4 5 6 7 8 9 10 11...
 - `drinne2`: int [1:11259] 4 5 6 7 8 9 10 11...
 - `s`: int [1:10031] 22801 22802 22803...
- Console:** Shows the execution output of the code, including the results of `aggregate`, `length`, `table`, and `order` operations.
- Markdown Quick Reference:** A sidebar panel providing a reference for creating documents and reports using R Markdown, including sections on Emphasis, Headers, and Lists.

<https://www.rstudio.com>

- `<-` assignment operator
- `#` can be used to comment your script
- `x<-rnorm(10,0,1)` creates a vector with ten standardnormal-distributed values
- `mean(x)` calculates the mean of variable `x`; `length(x)` returns the number of observations in `x`

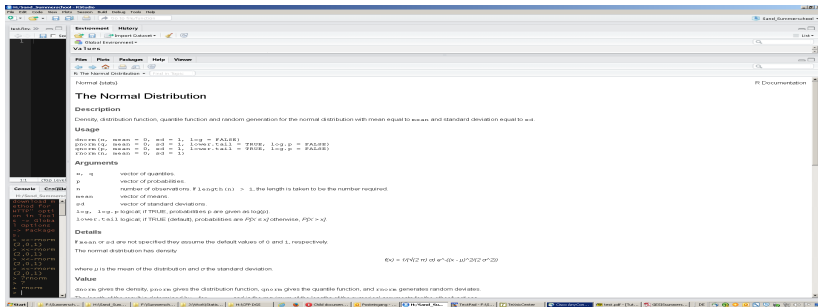
```
mean(x)
```

```
[1] -0.6266963
```

```
length(x)
```

```
[1] 10
```

- *?command*



- CRAN
- Quick-R
- stackoverflow.com

```
numeric    x<-c(1,2)
logical     x<-c(T,F)
character   x<-c("A","B")
factor      x<-as.factor(c("White","Black"))
```

`str()` returns the type of your data

```
x<-as.factor(c("White","Black"))
str(x)
```

```
Factor w/ 2 levels "Black","White": 2 1
```


Indexing a Vector:

```
A1<-c(1,2,3,4)
```

```
A1[1]
```

```
[1] 1
```

```
A1[1:3]
```

```
[1] 1 2 3
```

```
A1[-2]
```

```
[1] 1 3 4
```

Indexing a data frame:

```
A2<-4:1
```

```
AA<-cbind(A1,A2)
```

```
AA[1,]
```

```
A1 A2
```

```
1  4
```

```
AA[,1]
```

```
[1] 1 2 3 4
```

```
AA[1:3,2]
```

```
[1] 4 3 2
```

```
AA[,-1]
```

```
[1] 4 3 2 1
```

Indexing an array

```
A3<-array(1:8,c(2,2,2))
```

```
A3
```

```
, , 1
```

	[,1]	[,2]
[1,]	1	3
[2,]	2	4

```
, , 2
```

	[,1]	[,2]
[1,]	5	7
[2,]	6	8

```
A3[, ,2]
```

	[,1]	[,2]
[1,]	5	7
[2,]	6	8

Indexing a list

```
A4<-list(A1,c("Summer","Winter"))
```

```
A4
```

```
[[1]]
```

```
[1] 1 2 3 4
```

```
[[2]]
```

```
[1] "Summer" "Winter"
```

```
A4[[1]]
```

```
[1] 1 2 3 4
```

```
1:5
```

```
[1] 1 2 3 4 5
```

```
rep("A",times=10)
```

```
[1] "A" "A" "A" "A" "A" "A" "A" "A" "A" "A"
```

```
rep(1:3,times=2,each=3)
```

```
[1] 1 1 1 2 2 2 3 3 3 1 1 1 2 2 2 3 3 3
```

```
seq(-5,5,by=2.5)
```

```
[1] -5.0 -2.5  0.0  2.5  5.0
```

Function	Distribution	Important parameter
<code>runif()</code>	Uniform distribution	n, min, max
<code>rnorm()</code>	Normal distribution	n, mean, sd
<code>rpois()</code>	Poisson distribution	n, lambda
...

Function	Meaning	Example
<code>length()</code>	Length	<code>length(x)</code>
<code>max()</code>	Maximum	<code>max(x)</code>
<code>min()</code>	Minimum	<code>min(x)</code>
<code>sd()</code>	Standard deviation	<code>sd(x)</code>
<code>var()</code>	Variance	<code>var(x)</code>
<code>mean()</code>	Mean	<code>mean(x)</code>
<code>median()</code>	Median	<code>median(x)</code>

These functions do only need one argument
Other functions need to be specified by further arguments:

<code>quantile()</code>	90% Quantile	<code>quantile(x,.9)</code>
<code>sample()</code>	Draw a sample	<code>sample(x,1)</code>

Function	Meaning	Example
<code>length()</code>	Length	<code>length(x)</code>
<code>max()</code>	Maximum	<code>max(x)</code>
<code>min()</code>	Minimum	<code>min(x)</code>
<code>sd()</code>	Standard deviation	<code>sd(x)</code>
<code>var()</code>	Variance	<code>var(x)</code>
<code>mean()</code>	Mean	<code>mean(x)</code>
<code>median()</code>	Median	<code>median(x)</code>

These functions do only need one argument
Other functions need to be specified by further arguments:

<code>quantile()</code>	90% Quantile	<code>quantile(x,.9)</code>
<code>sample()</code>	Draw a sample	<code>sample(x,1)</code>

- R is a modular program with many functions included in basic R

Function	Meaning	Example
<code>length()</code>	Length	<code>length(x)</code>
<code>max()</code>	Maximum	<code>max(x)</code>
<code>min()</code>	Minimum	<code>min(x)</code>
<code>sd()</code>	Standard deviation	<code>sd(x)</code>
<code>var()</code>	Variance	<code>var(x)</code>
<code>mean()</code>	Mean	<code>mean(x)</code>
<code>median()</code>	Median	<code>median(x)</code>

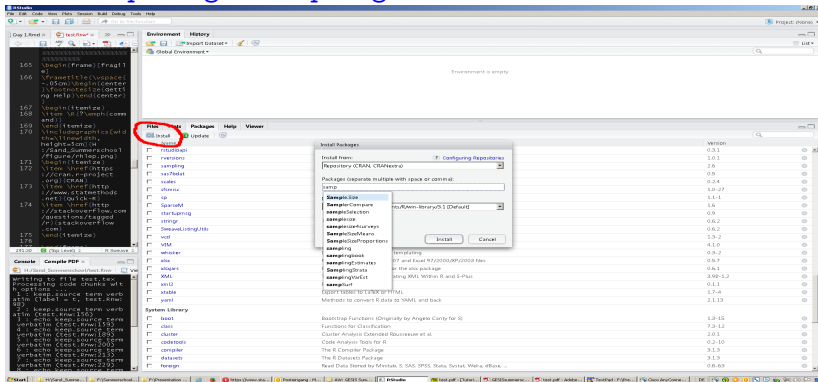
These functions do only need one argument
Other functions need to be specified by further arguments:

<code>quantile()</code>	90% Quantile	<code>quantile(x,.9)</code>
<code>sample()</code>	Draw a sample	<code>sample(x,1)</code>

- R is a modular program with many functions included in basic R
- more specific functions are embedded in further packages

INSTALLING AND LOADING PACKAGES

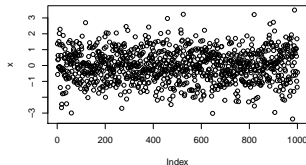
`install.package("sampling")`



`library(sampling)` or `require(sampling)`

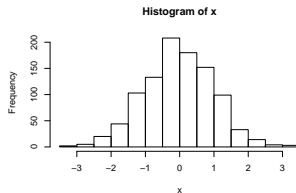
Library	Subject
<code>foreign</code>	reading and writing of data in numerous formats (e.g. <i>.dta</i> , <i>.sav</i>)
<code>sampling</code>	drawing and weighting samples
<code>survey</code>	analysis of complex survey samples
<code>xlsx</code>	read and write data in Excell-Format
<code>xtable</code>	export tables to LaTeX and HTML
<code>mice</code>	multiple imputation by chain equation
<code>reshape</code>	alter structure of datasets
<code>car</code>	applied regressions
<code>VIM</code>	visualization and imputation of Missing Values
<code>lattice</code>	high-level data visualization
<code>ggplot2</code>	grammar for graphics in R

```
set.seed(42)  
x <- rnorm(1000,0,1)  
plot(x)
```



`set.seed()` is used to specify a starting point

```
hist(x)
```



⇒ we will use 42 as seed-value for future exercises to obtain comparable results

`sample (base)`

R Documentation

Random Samples and Permutations


Description

`sample` takes a sample of the specified size from the elements of `x` using either with or without replacement.

Usage

```
sample(x, size, replace = FALSE, prob = NULL)
```

x: From what do we want
to sample ?



```
sample(x=1:10, n=1, replace=T)
```

`sample (base)`

R Documentation

Random Samples and Permutations


Description

`sample` takes a sample of the specified size from the elements of `x` using either with or without replacement.

Usage

```
sample(x, size, replace = FALSE, prob = NULL)
```

`n`: How many elements
do we want to draw?



```
sample(x=1:10, n=1, replace=T)
```

`sample {base}`

R Documentation

Random Samples and Permutations


Description

`sample` takes a sample of the specified size from the elements of `x` using either with or without replacement.

Usage

```
sample(x, size, replace = FALSE, prob = NULL)
```

Do we want to draw with
or without replacement?



```
sample(x=1:10, n=1, replace=T)
```

```
id <- 1:10000
set.seed(42)
education <- sample(c("none", "low", "average", "high"), 10000,
+                   replace = T, prob = c(.072, .356, .289, .283))
gender <- sample(c("male", "female"), 10000,
+               replace = T, prob = c(.488, .512))
iq <- rnorm(10000, 100, 20)
my.pop <- data.frame(id, gender, education, iq)
head(my.pop)
```

	id	gender	education	iq
1	1	male	high	123.26218
2	2	male	none	96.19531
3	3	male	low	94.21088
4	4	female	high	92.02308
5	5	male	average	114.18485
6	6	male	average	67.54705


```
summary(my.pop)
```

id	gender	education	iq
Min. : 1	female:5125	average:2851	Min. : 30.93
1st Qu.: 2501	male :4875	high :2820	1st Qu.: 86.50
Median : 5000		low :3588	Median :100.08
Mean : 5000		none : 741	Mean :100.02
3rd Qu.: 7500			3rd Qu.:113.60
Max. :10000			Max. :173.26

```
prop.table(table(my.pop$gender,my.pop$education))
```

	average	high	low	none
female	0.1449	0.1465	0.1844	0.0367
male	0.1402	0.1355	0.1744	0.0374

```
var(my.pop$iq)*(nrow(my.pop)-1)/nrow(my.pop)
```

```
[1] 406.1684
```

⇒ σ^2

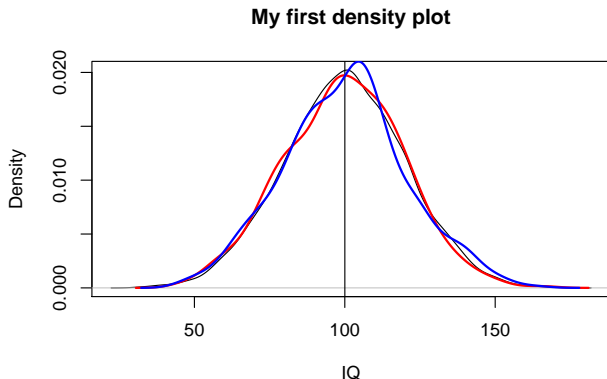
```
set.seed(42)
s.SRS <- sample(1:nrow(my.pop),500,replace=T)
s.SRSWOR <- sample(1:nrow(my.pop),500,replace=F)
my.samp.SRS <- my.pop[s.SRS,]
my.samp.SRSWOR <- my.pop[s.SRSWOR,]
summary(my.samp.SRS)
```

	id	gender	education	iq
Min.	: 3	female:257	average:132	Min. : 45.95
1st Qu.:	2322	male :243	high :134	1st Qu.: 85.38
Median	:4804		low :192	Median :100.00
Mean	:4896		none : 42	Mean : 99.60
3rd Qu.:	7434			3rd Qu.:113.20
Max.	:9966			Max. :165.63

```
nrow(unique(my.samp.SRS))
```

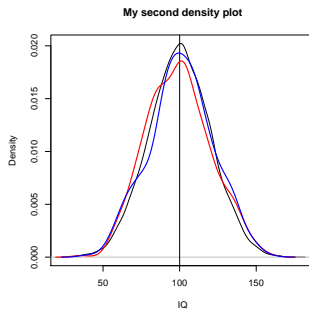
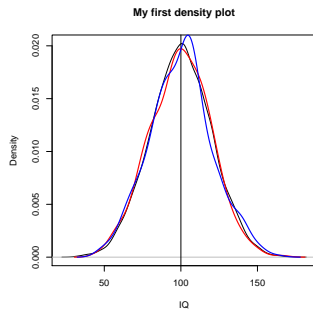
```
[1] 487
```

```
plot(density(my.pop$iq),main = "My first density plot"  
+      , xlab = "IQ")  
abline(v=mean(my.pop$iq), col = "black")  
lines(density(my.samp.SRS$iq),col = "red",lwd=2)  
lines(density(my.samp.SRSWOR$iq),col = "blue",lwd=2)
```



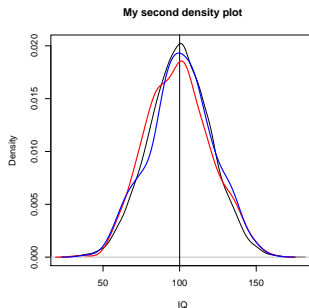
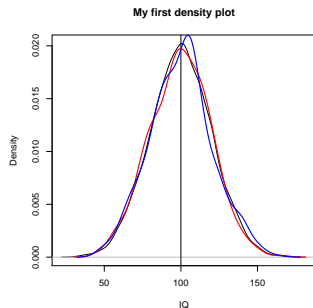
```
library(sampling)
set.seed(42)
s.SRS1 <- srswr(500,nrow(my.pop))
s.SRSWOR1 <- srswor(500,nrow(my.pop))
my.samp.SRS1 <- rbind(my.pop[s.SRS1!=0,]
+                      ,my.pop[s.SRS1>1,])
my.samp.SRSWOR1 <- my.pop[s.SRSWOR1==1,]
```

```
par(mfrow=c(1,2))
```



```
dev.off()
```

```
par(mfrow=c(1,2))
```



```
dev.off()
```

- should yield same results
- ⇒ routine differs in "starting point"

Declaring a working directory

```
path<-"H:/Sand_ Summerschool/Data Day1/"  
setwd(path)
```

- It is always useful to define and set your working directory at the beginning of each script
- `getwd()` displays you your current working directory
- `dir()` shows you all objects in a specific directory
- `ls()` lists all objects in your workspace
- `rm()` removes a object from your workspace

Example:

```
rm(list = ls())
```

Writing/ saving data and results

```
write.table(my.pop,"Synthetic Data Day1.csv",  
row.names = F, quote = F, dec = ".",sep = ",")
```

OR:

```
save(my.samp.SRS,s.SRS,my.samp.SRSWOR1,file = "Day1.Rdata")
```

⇒ See also: `write.csv` and `write.csv2` (`sep = ";"`)

Reading/ loading data and results

```
d1 <- read.table("Synthetic Data Day1.csv",  
header = F, dec = ".",sep = ",")
```

OR:

```
load("Day1.Rdata")
```


SAMPLE SIZES

- 1 Generate 1000 numbers from a exponential distribution
- 2 Draw three samples($n_1=2, n_2=10, n_3=100$)
- 3 Plot the density and add the means of the three samples as vertical lines

BELGIAN MUNICIPALITIES/ VARIANCE DECOMPOSITION

- 1 Load the dataset "belgianmunicipalities" from the `sample`-package using the `data()`-command
- 2 Inspect the structure of the dataset
- 3 Calculate mean and variance of the variable `averageincome`
- 4 Calculate the mean of each province for that variable, plot your results and add the mean of 3
- 5 Recalculate the mean of `averageincome` based on the means by province and compare your results
- 6 Make a boxplot of the variable `averageincome` for each province
- 7 Calculate the variance of `averageincome` using variance decomposition and compare it with 3
 - ⇒ Advice: consider the dataset as the aggregates for the whole population and use the formula for the population variance