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

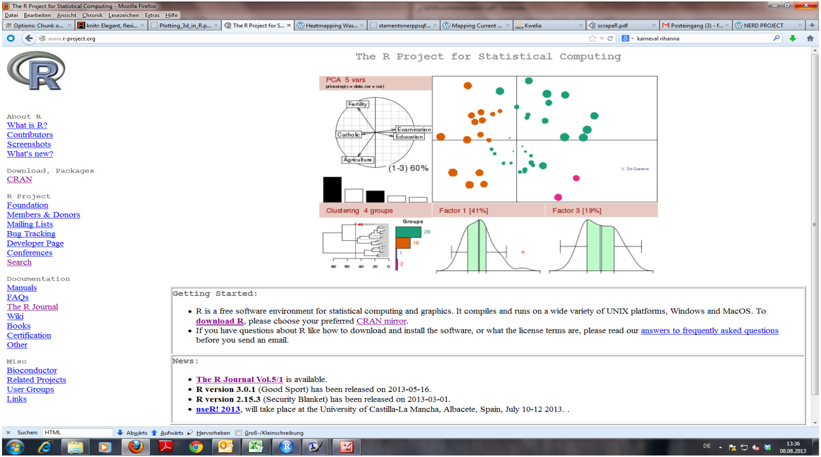
Why R?

- 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

About R
What is R?
Contributors
Screenshots
What's new?

Download, Packages
CRAN

R Project
Foundation
Members & Donors
Mailing Lists
Bug Tracking
Developer Page
Conferences
Search

Documentation
Manuals
FAQs
The R Journal
Wiki
Books
Certification
Other

Misc
Bioconductor
Related Projects
User Groups
Links

Getting Started:

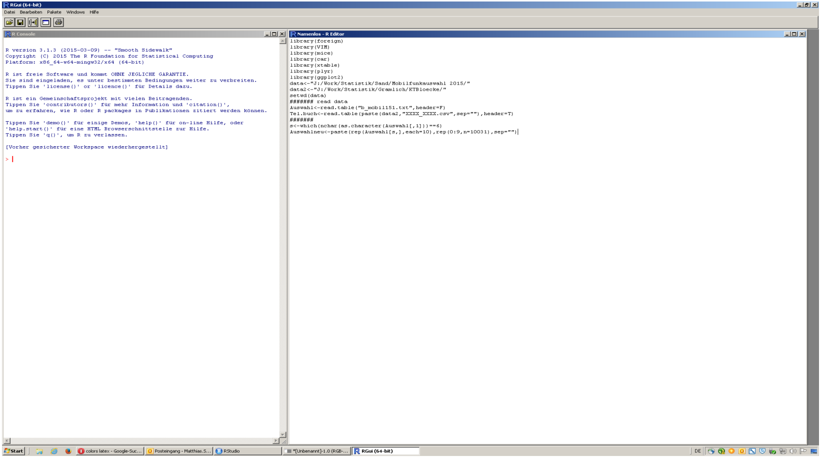
- R is a free software environment for statistical computing and graphics. It compiles and runs on a wide variety of UNIX platforms, Windows and MacOS. To **download R**, please choose your preferred CRAN mirror.
- If you have questions about R like how to download and install the software, or what the 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.
- **useR! 2013**, will take place at the University of Castilla-La Mancha, Albacete, Spain, July 10-12 2013. .

`https://www.r-project.org`

R Basic

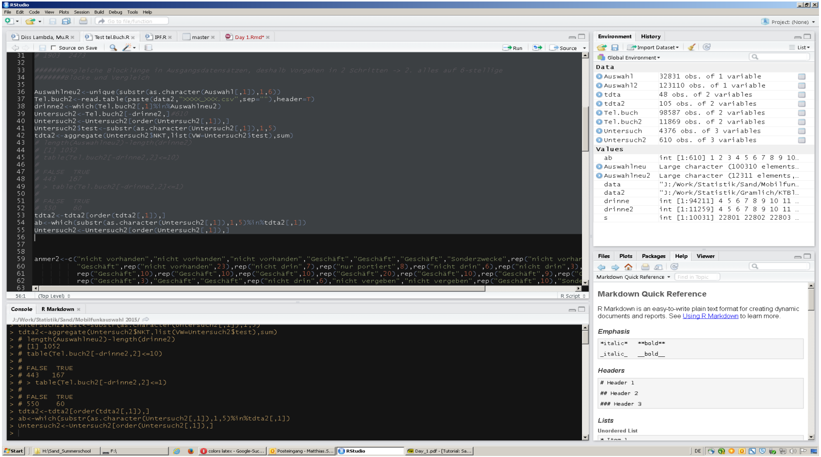


```
R version 3.1.3 (2015-03-09) -- "Smooth Sidewalk"
Copyright (C) 2015 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

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

R ist ein Gemeinschaftsprojekt mit vielen Beitragenden.
Tippen Sie 'contributors()' für mehr Information 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 Browserschnittstelle zur Hilfe.
Tippen Sie 'q()', um R zu verlassen.

[Vorher gesicherter Workspace wiederhergestellt]

> 
```

```
library(foreign)
library(VIM)
library(mice)
library(car)
library(xtable)
library(plyr)
library(ggplot2)
data<-"J:/Work/Statistik/Sand/Mobilfunkauswahl 2015/"
data2<-"J:/Work/Statistik/Gramlich/KTBloecke/"
setwd(data)
####### read data
Auswahl<-read.table("b_mobil151.txt",header=F)
Tel.buch<-read.table(paste(data2,"XXXX_XXXX.csv",sep=""),header=T)
#######
s<-which(nchar(as.character(Auswahl[,1]))==6)
Auswahlneu<-paste(rep(Auswahl[s,],each=10),rep(0:9,n=10031),sep="")
```

- most R-user prefer the graphical user interface (GUI) RStudio

R Studio



```r
# 1903  2473

######Ungleiche Blocklänge in Ausgangsdatensätzen, deshalb Vorgehen in 2 Schritten -> 2. alles auf 6-stellige
######Blöcke und Vergleich

Auswahlneu2<-unique(substr(as.character(Auswahl[,1]),1,6))
Tel.buch2<-read.table(paste(data2,"XXXX_XXX.csv",sep=""),header=T)
drinne2<-which(Tel.buch2[,1]%in%Auswahlneu2)
Untersuch2<-Tel.buch2[-drinne2,]#610
Untersuch2<-Untersuch2[order(Untersuch2[,1]),]
Untersuch2$test<-substr(as.character(Untersuch2[,1]),1,5)
tdta2<-aggregate(Untersuch2$NKT,list(VW=Untersuch2$test),sum)
# length(Auswahlneu2)-length(drinne2)
# [1] 1052
# table(Tel.buch2[-drinne2,2]<=10)
#
# FALSE  TRUE
# 443    167
# > table(Tel.buch2[-drinne2,2]<=1)
#
# FALSE  TRUE
# 550    60
tdta2<-tdta2[order(tdta2[,1]),]
ab<-which(substr(as.character(Untersuch2[,1]),1,5)%in%tdta2[,1])
Untersuch2<-Untersuch2[order(Untersuch2[,1]),]



anmer2<-c("nicht vorhanden","nicht vorhanden","nicht vorhanden","Geschäft","Geschäft","Geschäft","Sonderzwecke",rep("nicht vorhar
          "Geschäft",rep("nicht vorhanden",23),rep("nicht drin",7),rep("nur portiert",8),rep("nicht drin",6),rep("nicht drin",3),
          rep("Geschäft",10),rep("Geschäft",10),rep("Geschäft",10),rep("Geschäft",20),rep("Geschäft",10),rep("Geschäft",9),rep("G
          rep("Geschäft",3),"Geschäft","Geschäft",rep("nicht drin",6),"nicht vergeben","nicht vergeben",rep("Geschäft",10),"Sonde
```

56:1 (Top Level) R Script

J:/Work/Statistik/Sand/Mobilfunkauswahl 2015/

```
> tdta2<-aggregate(Untersuch2$NKT,list(VW=Untersuch2$test),sum)
> # length(Auswahlneu2)-length(drinne2)
> # [1] 1052
> # table(Tel.buch2[-drinne2,2]<=10)
> #
> # FALSE  TRUE
> # 443    167
> # > table(Tel.buch2[-drinne2,2]<=1)
> #
> # FALSE  TRUE
> # 550    60
> tdta2<-tdta2[order(tdta2[,1]),]
> ab<-which(substr(as.character(Untersuch2[,1]),1,5)%in%tdta2[,1])
> Untersuch2<-Untersuch2[order(Untersuch2[,1]),]
> 
```

Global Environment

Data	
Auswahl	32831 obs. of 1 variable
Auswahl2	123110 obs. of 1 variable
tdta	48 obs. of 2 variables
tdta2	105 obs. of 2 variables
Tel.buch	98587 obs. of 2 variables
Tel.buch2	11869 obs. of 2 variables
Untersuch	4376 obs. of 3 variables
Untersuch2	610 obs. of 3 variables
Values	
ab	int [1:610] 1 2 3 4 5 6 7 8 9 10…
Auswahlneu	Large character (100310 elements…
Auswahlneu2	Large character (12311 elements,…
data	"J:/Work/Statistik/Sand/Mobilfun…
data2	"J:/Work/Statistik/Gramlich/KTBl…
drinne	int [1:94211] 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 …

Markdown Quick Reference

R Markdown is an easy-to-write plain text format for creating dynamic documents and reports. See Using R Markdown to learn more.

Emphasis

```
*italic*   **bold**
_italic_   __bold__
```

Headers

```
# Header 1
## Header 2
### Header 3
```

Lists

Unordered List

https://www.rstudio.com

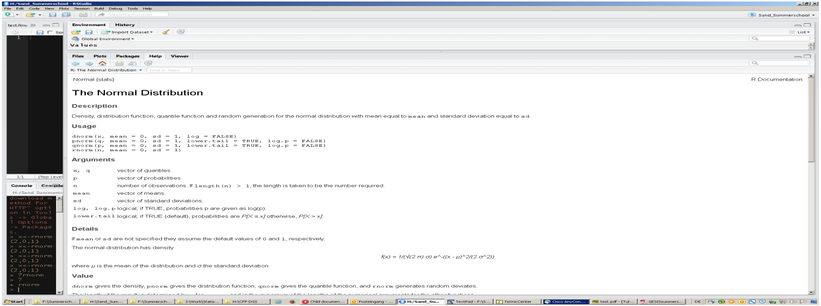
Basic R Commands

- `<-` 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] 0.07789946
length(x)
[1] 10
```

Getting Help

- *?command*



```
1
```

Values

Normal {stats}

R Documentation

The Normal Distribution

Description

Density, distribution function, quantile function and random generation for the normal distribution with mean equal to `mean` and standard deviation equal to `sd`.

Usage

```
dnorm(x, mean = 0, sd = 1, log = FALSE)
pnorm(q, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
qnorm(p, mean = 0, sd = 1, lower.tail = TRUE, log.p = FALSE)
rnorm(n, mean = 0, sd = 1)
```

Arguments

`x, q`	vector of quantiles.
`p`	vector of probabilities.
`n`	number of observations. If `length(n) > 1`, the length is taken to be the number required.
`mean`	vector of means.
`sd`	vector of standard deviations.
`log, log.p`	logical; if TRUE, probabilities p are given as log(p).
`lower.tail`	logical; if TRUE (default), probabilities are $P[X \le x]$ otherwise, $P[X > x]$.

Details

If `mean` or `sd` are not specified they assume the default values of `0` and `1`, respectively.

The normal distribution has density

f(x) = 1/(√(2 π) σ) e^-((x - μ)^2/(2 σ^2))

where μ is the mean of the distribution and σ the standard deviation.

Value

`dnorm` gives the density, `pnorm` gives the distribution function, `qnorm` gives the quantile function, and `rnorm` generates random deviates.

```
download m
ethod for
HTTP" opti
on in Tool
s -> Globa
l Options
-> Package
s.
> x<-rnorm
(2,0,1)
> x<-rnorm
(2,0,1)
> x<-rnorm
(2,0,1)
> x<-rnorm
(2,0,1)
> ?rnorm
> ?
+ rnorm
> 
```

- CRAN
- Quick-R
- stackoverflow.com

Types of Data

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 I

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 II

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 IV

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
```

Sequences

```
 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
```

Random Numbers

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

Important Functions

of the {base}-Package

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

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

`quantile()`	90% Quantile	`quantile(x,.9)`
`sample()`	Draw a sample	`sample(x,1)`

Important Functions
of the {base}-Package

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

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

`quantile()`	90% Quantile	`quantile(x,.9)`
`sample()`	Draw a sample	`sample(x,1)`

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

Important Functions

of the {base}-Package

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

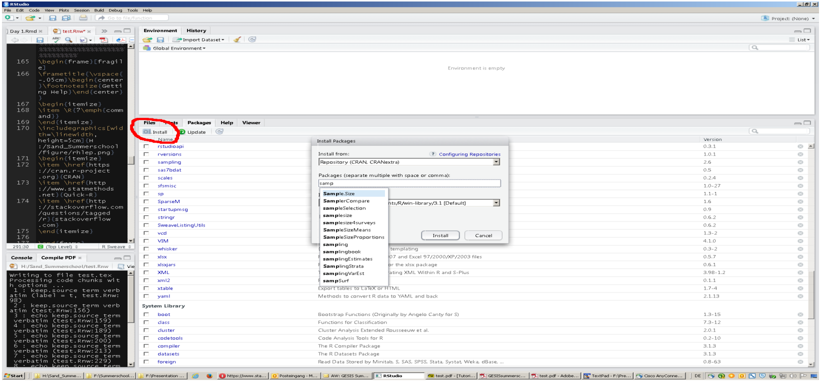
These functions do only need one argument
Other functions need to be specified by further arguments:

`quantile()`	90% Quantile	`quantile(x,.9)`
`sample()`	Draw a sample	`sample(x,1)`

- `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")`



RStudio
File Edit Code View Plots Session Build Debug Tools Help
Go to file/function
Project: (None)
Day 1.Rmd
test.Rnw*
Environment History
Import Dataset
List
Global Environment
Environment is empty
Files Plots Packages Help Viewer
Install Update
291:30
(Top Level)
R Sweave
Console Compile PDF
H:/Sand_Summerschool/test.Rnw
View
Start H:\Sand_Summe... F:\Summerschool... F:\Presentation ... https://www.sta... Posteingang - M... AW: GESIS Sum... RStudio test.pdf - [Tutori... GESISsummersc... test.pdf - Adobe... TextPad - F:\Pre... Cisco AnyConne... DE

```
\begin{frame}[fragile]
\frametitle{\vspace{-.05cm}\begin{center}\footnotesize{Getting Help}\end{center}}
\begin{itemize}
\item \R{?\emph{command}}
\end{itemize}
\includegraphics[width=\linewidth,height=5cm]{H:/Sand_Summerschool/figure/rhlep.png}
\begin{itemize}
\item \href{https://cran.r-project.org}{CRAN}
\item \href{http://www.statmethods.net}{Quick-R}
\item \href{http://stackoverflow.com/questions/tagged/r}{stackoverflow.com}
\end{itemize}
```

```
Writing to file test.tex
Processing code chunks with options ...
 1 : keep.source term verbatim (label = t, test.Rnw:98)
 2 : keep.source term verbatim (test.Rnw:156)
 3 : echo keep.source term verbatim (test.Rnw:159)
 4 : echo keep.source term verbatim (test.Rnw:189)
 5 : echo keep.source term verbatim (test.Rnw:200)
 6 : echo keep.source term verbatim (test.Rnw:213)
 7 : echo keep.source term verbatim (test.Rnw:229)
```

Install Packages

Install from: Repository (CRAN, CRANextra) — Configuring Repositories

Packages (separate multiple with space or comma): samp

Sample.Size
SamplerCompare
sampleSelection
samplesize
samplesize4surveys
SampleSizeMeans
SampleSizeProportions
sampling
samplingbook
samplingEstimates
SamplingStrata
samplingVarEst
sampSurf

...nts/R/win-library/3.1 [Default]

Install Cancel

	Name		Version
□	rstudioapi		0.3.1
□	rversions		1.0.1
□	sampling		2.6
□	sas7bdat		0.5
□	scales		0.2.4
□	sfsmisc		1.0-27
□	sp		1.1-1
□	SparseM		1.6
□	startupmsg		0.9
□	stringr		0.6.2
□	SweaveListingUtils		0.6.2
□	vcd		1.3-2
□	VIM		4.1.0
□	whisker	templating	0.3-2
□	xlsx	07 and Excel 97/2000/XP/2003 files	0.5.7
□	xlsxjars	or the xlsx package	0.6.1
□	XML	ating XML Within R and S-Plus	3.98-1.2
□	xml2		0.1.1
□	xtable	Export tables to LaTeX or HTML	1.7-4
□	yaml	Methods to convert R data to YAML and back	2.1.13
System Library			
□	boot	Bootstrap Functions (Originally by Angelo Canty for S)	1.3-15
□	class	Functions for Classification	7.3-12
□	cluster	Cluster Analysis Extended Rousseeuw et al.	2.0.1
□	codetools	Code Analysis Tools for R	0.2-10
□	compiler	The R Compiler Package	3.1.3
□	datasets	The R Datasets Package	3.1.3
□	foreign	Read Data Stored by Minitab, S, SAS, SPSS, Stata, Systat, Weka, dBase, ...	0.8-63

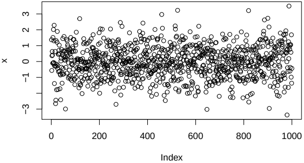
`library(sampling)` *or* `require(sampling)`

Useful Packages

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

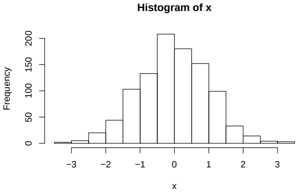
Basic Graphics

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



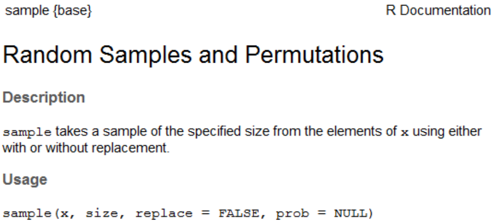
`set.seed()` is used to specify a starting point

```
hist(x)
```



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

The Basic sample Function



sample {base} R Documentation

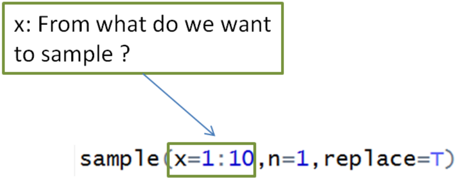
Random Samples and Permutations

Description

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

Usage

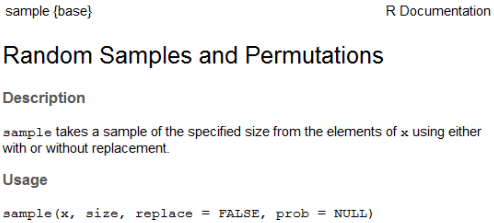
```
sample(x, size, replace = FALSE, prob = NULL)
```



x: From what do we want to sample ?

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

The Basic `sample` Function



sample {base} R Documentation

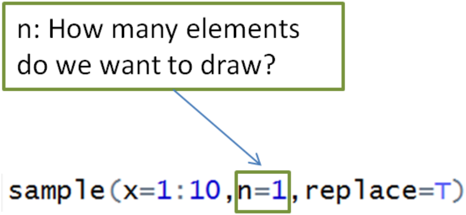
Random Samples and Permutations

Description

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

Usage

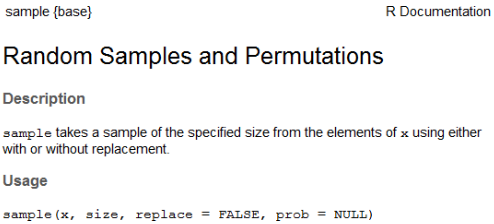
```
sample(x, size, replace = FALSE, prob = NULL)
```



n: How many elements do we want to draw?

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

The Basic `sample` Function



sample {base} R Documentation

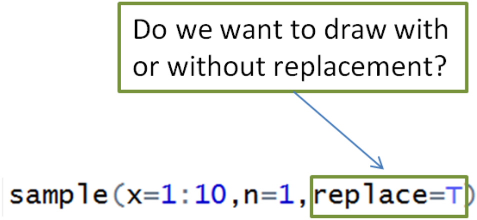
Random Samples and Permutations

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)
```

Simple Example on Sampling

```
 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
```

Simple Example on Sampling

Summary of the dataset

```
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
```

$\Rightarrow \sigma^2$

Simple Example on Sampling

```
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)
```

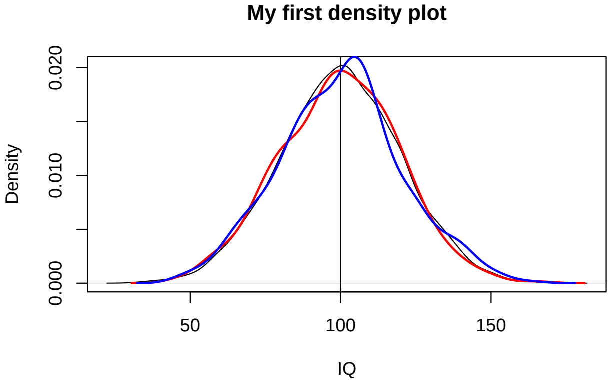
```
      id            gender         education          iq        
 Min.   :   3   female:257   average:132   Min.   : 45.95  
 1st Qu.:2322   male  :243   high   :134   1st Qu.: 85.38  
 Median :4804                low    :192   Median :100.00  
 Mean   :4896                none   : 42   Mean   :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

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

```
[1] 487
```

Simple Example on Sampling

```
 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)
```



Simple Example on Sampling

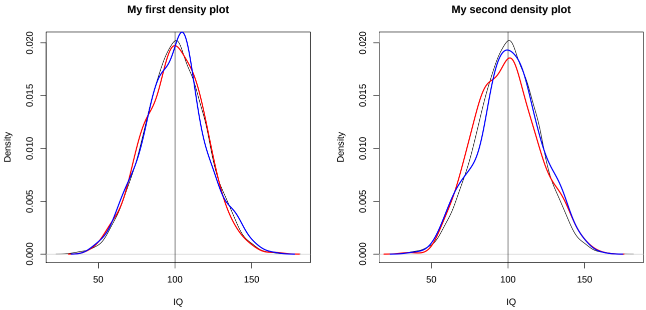
the sampling package

```
 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,]
```

Simple Example on Sampling

the sampling package

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

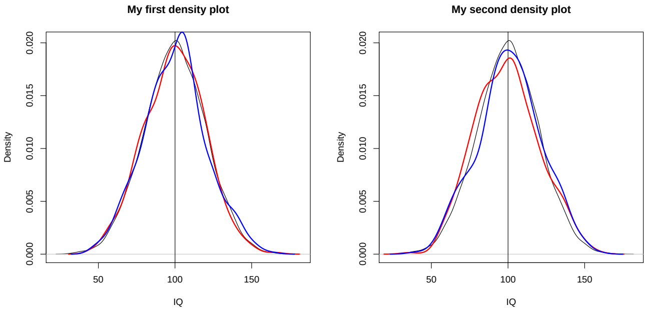


```
dev.off()
```

Simple Example on Sampling

the sampling package

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



```
dev.off()
```

- should yield same results

⇒ routine differs in "starting point"

Working Directory and Workspace

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())
```

Reading and Writing Data

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")
```

Exercise 1

Sample sizes

1. Generate 1000 numbers from a exponential distribution
2. Draw three samples(n1=2,n2=10,n3=100)
3. Plot the density and add the means of the three samples as vertical lines

Exercise 2

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