R and R-Studio

Bjarki Þór Elvarsson and Einar Hjörleifsson

Marine Research Institute

Overview

Overview

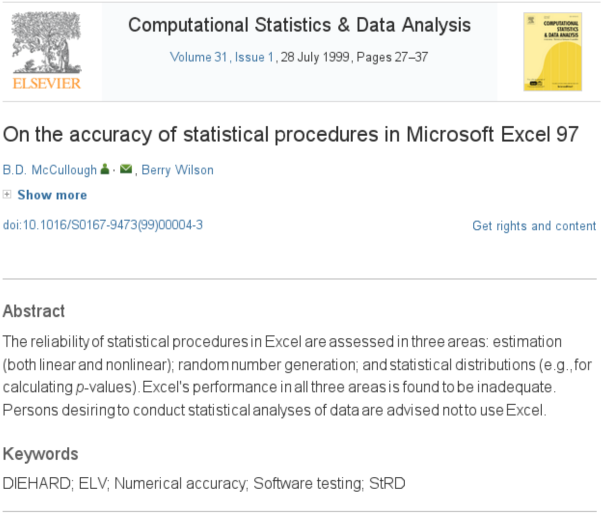
KEEP
CALM
AND
DON'T
PANIC

Why R?

R has become the lingua france of statistical analysis and data wrangling

- Its free! If you are a teacher or a student, the benefits are obvious
- It runs on a variety of platforms including Windows, Unix and MacOS
- It provides an unparalleled platform for programming new statistical methods in an easy and straightforward manner
- It offers powerful tools for data exploration and presentation

Why not Excel



Computational Statistics & Data Analysis

Volume 31, Issue 1, 28 July 1999, Pages 27–37

On the accuracy of statistical procedures in Microsoft Excel 97

B.D. McCullough, Berry Wilson

Show more

doi:10.1016/S0167-9473(99)00004-3

Get rights and content

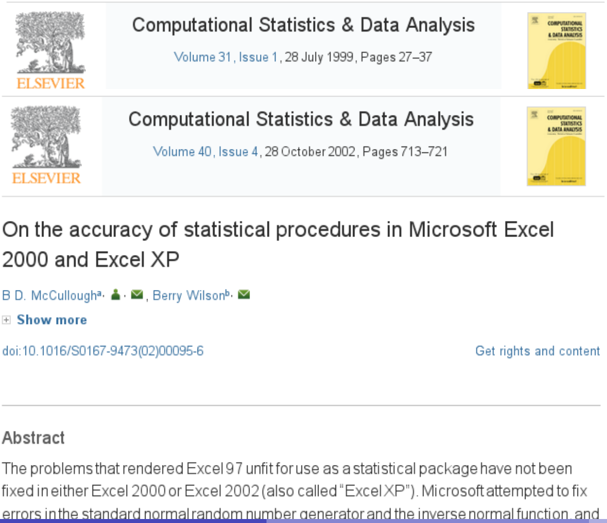
Abstract

The reliability of statistical procedures in Excel are assessed in three areas: estimation (both linear and nonlinear); random number generation; and statistical distributions (e.g., for calculating *p*-values). Excel's performance in all three areas is found to be inadequate. Persons desiring to conduct statistical analyses of data are advised not to use Excel.

Keywords

DIEHARD; ELV; Numerical accuracy; Software testing; StRD

Why not Excel



Computational Statistics & Data Analysis

Volume 31, Issue 1, 28 July 1999, Pages 27–37

Computational Statistics & Data Analysis

Volume 40, Issue 4, 28 October 2002, Pages 713–721

On the accuracy of statistical procedures in Microsoft Excel 2000 and Excel XP

B D. McCullough[a], Berry Wilson[b]

Show more

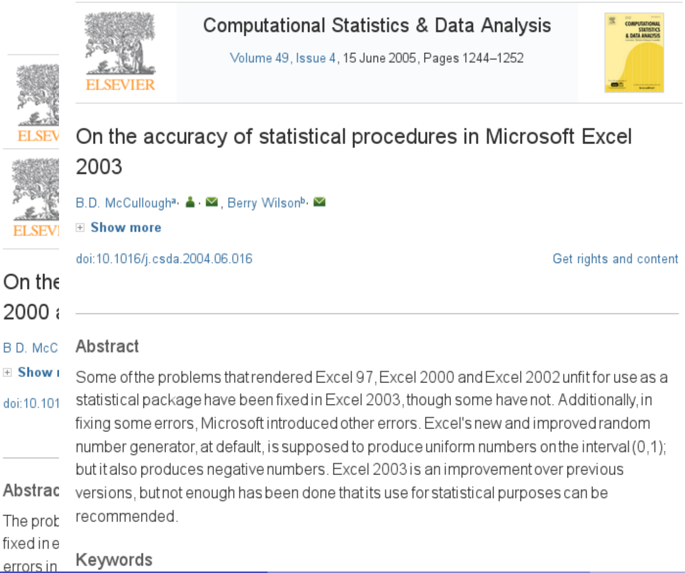
doi:10.1016/S0167-9473(02)00095-6

Get rights and content

Abstract

The problems that rendered Excel 97 unfit for use as a statistical package have not been fixed in either Excel 2000 or Excel 2002 (also called "Excel XP"). Microsoft attempted to fix errors in the standard normal random number generator and the inverse normal function, and

Why not Excel



Computational Statistics & Data Analysis

Volume 49, Issue 4, 15 June 2005, Pages 1244–1252

On the accuracy of statistical procedures in Microsoft Excel 2003

B.D. McCullough[a], Berry Wilson[b]

Show more

doi:10.1016/j.csda.2004.06.016

Get rights and content

Abstract

Some of the problems that rendered Excel 97, Excel 2000 and Excel 2002 unfit for use as a statistical package have been fixed in Excel 2003, though some have not. Additionally, in fixing some errors, Microsoft introduced other errors. Excel's new and improved random number generator, at default, is supposed to produce uniform numbers on the interval (0,1); but it also produces negative numbers. Excel 2003 is an improvement over previous versions, but not enough has been done that its use for statistical purposes can be recommended.

Keywords

ELSEV

ELSEVI

On the

2000 a

B D. McC

Show

doi:10.101

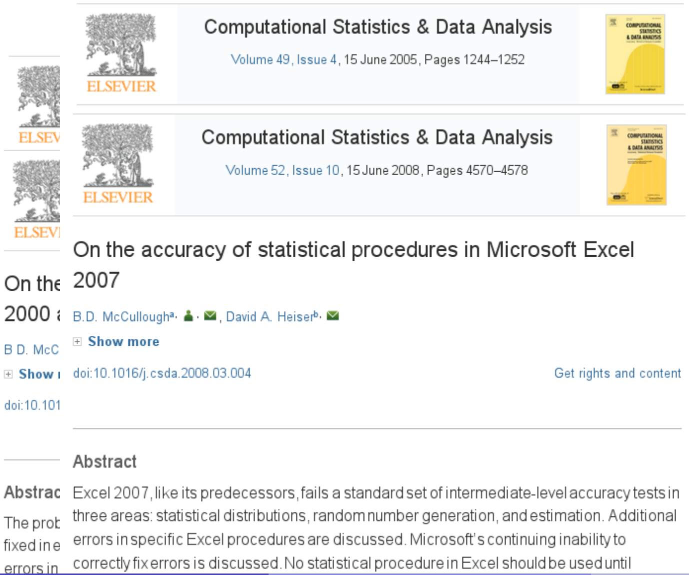
Abstrac

The prob

fixed in e

errors in

Why not Excel



Computational Statistics & Data Analysis

Volume 49, Issue 4, 15 June 2005, Pages 1244–1252

On the
2000 a

B D. McC

⊞ Show

doi:10.101

Abstrac

The prob
fixed in e
errors in

Computational Statistics & Data Analysis

Volume 52, Issue 10, 15 June 2008, Pages 4570–4578

On the accuracy of statistical procedures in Microsoft Excel 2007

B.D. McCullough[a], David A. Heiser[b]

⊞ Show more

doi:10.1016/j.csda.2008.03.004

Get rights and content

Abstract

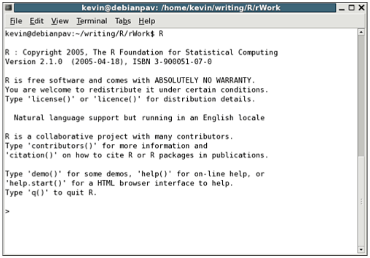
Excel 2007, like its predecessors, fails a standard set of intermediate-level accuracy tests in three areas: statistical distributions, random number generation, and estimation. Additional errors in specific Excel procedures are discussed. Microsoft's continuing inability to correctly fix errors is discussed. No statistical procedure in Excel should be used until

What is R?

What is R (cont)

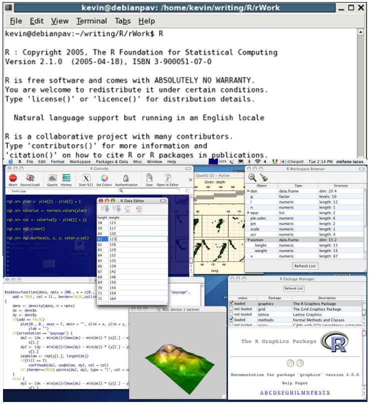
- R is commandline driven:
    - its biggest appeal as one can reuse commands
    - its biggest hurdle in widespread use
- R is open-source:
    - Other statistical software packages can be extremely expensive
    - Large user base with almost all statistical methods implemented

R - interfaces


kevin@debianpav: /home/kevin/writing/R/rWork
File Edit View Terminal Tabs Help

```
kevin@debianpav:~/writing/R/rWork$ R

R : Copyright 2005, The R Foundation for Statistical Computing
Version 2.1.0  (2005-04-18), ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for a HTML browser interface to help.
Type 'q()' to quit R.

>
```

R - interfaces


kevin@debianpav: /home/kevin/writing/R/rWork
File Edit View Terminal Tabs Help
R File Edit Format Workspace Packages & Data Misc Window Help
(Charged) Tue 2:14 PM stefano iacus
R Console
Abort Source/Load Quartz History Start X11 Set Colors Authentication Save Open In Editor
/Users/jago
R Data Editor
Quartz (2) - Active
Given : depth
long
R Workspace Browser
Refresh List
R Package Manager
RGL device 1 (active)

```
kevin@debianpav:~/writing/R/rWork$ R

R : Copyright 2005, The R Foundation for Statistical Computing
Version 2.1.0  (2005-04-18), ISBN 3-900051-07-0

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
```

```
rgl.sr> ylen <- ylim[2] - ylim[1] + 1

rgl.sr> colorlut <- terrain.colors(ylen)

rgl.sr> col <- colorlut[y - ylim[1] + 1]

rgl.sr> rgl.clear()

rgl.sr> rgl.surface(x, z, y, color = col)
>
>
>
>
>
>
```

height	weight
58	115
59	117
60	120
61	123
62	126
63	129
64	132
65	135
66	139
67	142
68	146
69	150
70	154
71	159
72	164

Object	Type	Structure
▶dati	data.frame	dim: 20 4
g	factor	levels: 10
l	numeric	length: 12
n	numeric	length: 1
▶opar	list	length: 2
pie.sales	numeric	length: 6
pin	numeric	length: 2
scale	numeric	length: 1
usr	numeric	length: 4
▼women	data.frame	dim: 15 2
height	numeric	length: 15
weight	numeric	length: 15
x	numeric	length: 87

status	Package	Description
✓ loaded	graphics	The R Graphics Package
not loaded	grid	The Grid Graphics Package
not loaded	lattice	Lattice Graphics
✓ loaded	methods	Formal Methods and Classes

The R Graphics Package

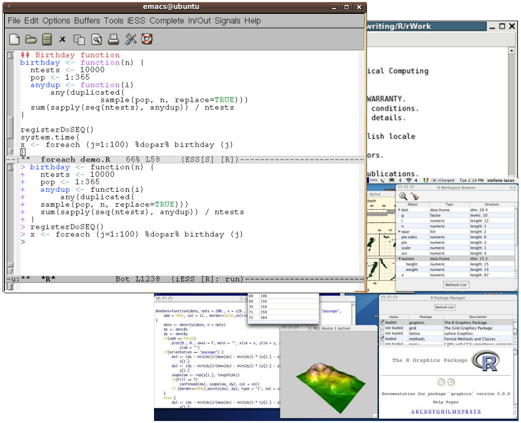
Documentation for package `graphics' version 2.0.0

Help Pages

A B C D E F G H I L M N P R S T X

```
BoxDens=function(data, npts = 200., x = c(0.,                          "paysage",
    add = TRUE, col = 11., border=FALSE,collin
{
    dens <- density(data, n = npts)
    dx <- dens$x
    dy <- dens$y
    if(add == FALSE)
        plot(0., 0., axes = F, main = "", xlim = x, ylim = y,
            ylab = "")
    if(orientation == "paysage") {
        dx2 <- (dx - min(dx))/(max(dx) - min(dx)) * (x[2.] - x
            x[1.]
        dy2 <- (dy - min(dy))/(max(dy) - min(dy)) * (y[2.] - y
            y[1.]
        seqbelow <- rep(y[1.], length(dx))
        if(Fill == T)
            confshade(dx2, seqbelow, dy2, col = col)
        if (border==TRUE) points(dx2, dy2, type = "l", col = c
    }
    else {
        dy2 <- (dx - min(dx))/(max(dx) - min(dx)) * (y[2.] - y
```

R - interfaces



```
## Birthday function
birthday <- function(n) {
  ntests <- 10000
  pop <- 1:365
  anydup <- function(i)
      any(duplicated(
                sample(pop, n, replace=TRUE)))
  sum(sapply(seq(ntests), anydup)) / ntests
}

registerDoSEQ()
system.time(
x <- foreach (j=1:100) %dopar% birthday (j)
)
--:**  foreach demo.R    66% L58    (ESS[S] [R])------------------------
> birthday <- function(n) {
+   ntests <- 10000
+   pop <- 1:365
+   anydup <- function(i)
+       any(duplicated(
+   sample(pop, n, replace=TRUE)))
+   sum(sapply(seq(ntests), anydup)) / ntests
+ }
> registerDoSEQ()
> x <- foreach (j=1:100) %dopar% birthday (j)
>
-u:**  *R*           Bot L1238   (iESS [R]: run)------------------------
```

```
ical Computing

WARRANTY.
 conditions.
 details.

lish locale

ors.

ublications.
```

Object	Type	Structure
▶ dati	data.frame	dim: 20 4
g	factor	levels: 10
l	numeric	length: 12
n	numeric	length: 1
▶ opar	list	length: 2
pie.sales	numeric	length: 6
pin	numeric	length: 2
scale	numeric	length: 1
usr	numeric	length: 4
▼ women	data.frame	dim: 15 2
height	numeric	length: 15
weight	numeric	length: 15
x	numeric	length: 87

status	Package	Description
✓ loaded	graphics	The R Graphics Package
not loaded	grid	The Grid Graphics Package
not loaded	lattice	Lattice Graphics
✓ loaded	methods	Formal Methods and Classes

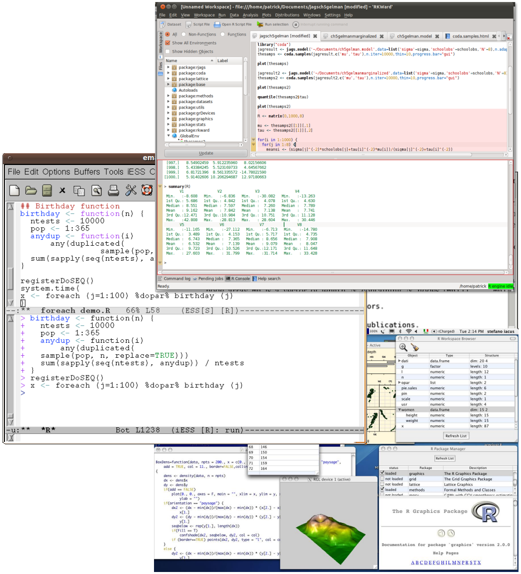
The R Graphics Package

Documentation for package `graphics' version 2.0.0

Help Pages

A B C D E F G H I L M N P R S T X

R - interfaces



```
library("coda")
jagresult <- jags.model('~/Documents/ch5gelman.model',data=list('sigma'=sigma,'schoolobs'=schoolobs,'N'=8),n.adap
thesamps <- coda.samples(jagresult,c('mu','tau'),n.iter=10000,thin=10,progress.bar="gui")

plot(thesamps)

jagresult2 <- jags.model('~/Documents/ch5gelmanmarginalized',data=list('sigma'=sigma,'schoolobs'=schoolobs,'N'=8
thesamps2 <- coda.samples(jagresult2,c('mu','tau'),n.iter=10000,thin=10,progress.bar="gui")

plot(thesamps2)

quantile(thesamps2$tau)

plot(thesamps2)

R <- matrix(0,1000,8)

mu <- thesamps2[[1]][,1]
tau <- thesamps2[[1]][,2]

for(i in 1:1000) {
  for(j in 1:8) {
    meansi <- (sigma[j]^(-2)*schoolobs[j]+tau[i]^(-2)*mu[i])/(sigma[j]^(-2)+tau[i]^(-2))
```

```
 [997,]   8.54902459  5.912235960   8.02156606
 [998,]   5.43384245  5.523169733   4.64567662
 [999,]   6.81721396  8.561335572 -14.78021590
[1000,]   5.91402606 10.206294687  12.97180663

> summary(R)
      V1               V2               V3                V4
 Min.   :-8.608   Min.   :-6.836   Min.   :-30.082   Min.   :-13.263
 1st Qu.: 5.686   1st Qu.: 4.842   1st Qu.:  4.078   1st Qu.:  4.630
 Median : 8.551   Median : 7.597   Median :  7.260   Median :  7.789
 Mean   : 9.162   Mean   : 7.842   Mean   :  7.138   Mean   :  7.741
 3rd Qu.:12.471   3rd Qu.:10.984   3rd Qu.: 10.751   3rd Qu.: 11.128
 Max.   :42.808   Max.   :28.813   Max.   : 28.604   Max.   : 30.446
      V5                V6                V7               V8
 Min.   :-11.165   Min.   :-27.112   Min.   :-6.713   Min.   :-14.780
 1st Qu.:  3.489   1st Qu.:  4.153   1st Qu.: 5.717   1st Qu.:  4.735
 Median :  6.743   Median :  7.365   Median : 8.656   Median :  7.908
 Mean   :  6.532   Mean   :  7.139   Mean   : 9.079   Mean   :  8.047
 3rd Qu.:  9.723   3rd Qu.: 10.526   3rd Qu.:12.171   3rd Qu.: 11.648
 Max.   : 27.603   Max.   : 31.799   Max.   :31.714   Max.   : 33.428

>
```

```
## Birthday function
birthday <- function(n) {
  ntests <- 10000
  pop <- 1:365
  anydup <- function(i)
      any(duplicated(
           sample(pop,
  sum(sapply(seq(ntests), a
}

registerDoSEQ()
system.time(
x <- foreach (j=1:100) %dopar% birthday (j)
)
--:**  foreach_demo.R    66% L58    (ESS[S]  [R])-----------------------
> birthday <- function(n) {
+   ntests <- 10000
+   pop <- 1:365
+   anydup <- function(i)
+       any(duplicated(
+   sample(pop, n, replace=TRUE)))
+   sum(sapply(seq(ntests), anydup)) / ntests
+ }
> registerDoSEQ()
> x <- foreach (j=1:100) %dopar% birthday (j)
>
-u:**  *R*          Bot L1238   (iESS [R]: run)---------------------
```

Object	Type	Structure
dati	data.frame	dim: 20 4
g	factor	levels: 10
l	numeric	length: 12
n	numeric	length: 1
opar	list	length: 2
pie.sales	numeric	length: 6
pin	numeric	length: 2
scale	numeric	length: 1
usr	numeric	length: 4
women	data.frame	dim: 15 2
height	numeric	length: 15
weight	numeric	length: 15
x	numeric	length: 87

status	Package	Description
loaded	graphics	The R Graphics Package
not loaded	grid	The Grid Graphics Package
not loaded	lattice	Lattice Graphics
loaded	methods	Formal Methods and Classes

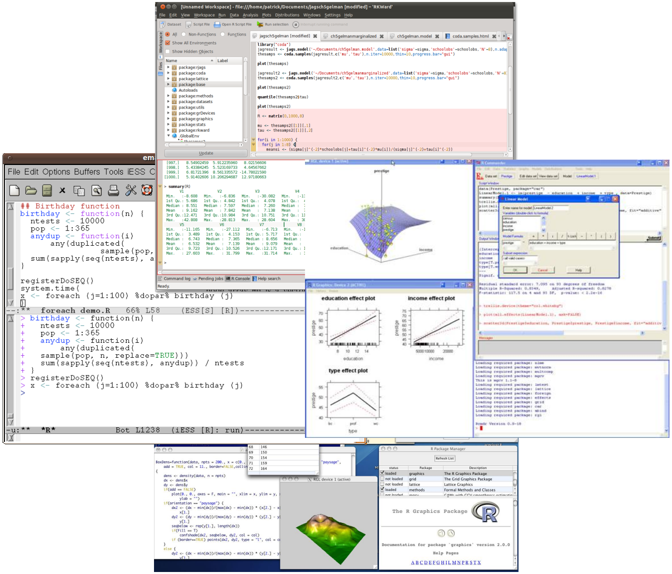
The R Graphics Package

Documentation for package `graphics' version 2.0.0

Help Pages

A B C D E F G H I L M N P R S T X

R - interfaces



```
library("coda")
jagresult <- jags.model('~/Documents/ch5gelman.model',data=list('sigma'=sigma,'schoolobs'=schoolobs,'N'=8),n.adap
thesamps <- coda.samples(jagresult,c('mu','tau'),n.iter=10000,thin=10,progress.bar="gui")

plot(thesamps)

jagresult2 <- jags.model('~/Documents/ch5gelmanmarginalized',data=list('sigma'=sigma,'schoolobs'=schoolobs,'N'=8)
thesamps2 <- coda.samples(jagresult2,c('mu','tau'),n.iter=10000,thin=10,progress.bar="gui")

plot(thesamps2)

quantile(thesamps2$tau)

plot(thesamps2)

R <- matrix(0,1000,8)

mu <- thesamps2[[1]][,1]
tau <- thesamps2[[1]][,2]

for(i in 1:1000) {
  for(j in 1:8) {
   meansi <- (sigma[j]^(-2)*schoolobs[j]+tau[i]^(-2)*mu[i])/(sigma[j]^(-2)+tau[i]^(-2))
```

```
[997,]   8.54902459  5.912235960   8.02156606
[998,]   5.43384245  5.523169733   4.64567662
[999,]   6.81721396  8.561335572 -14.78021590
[1000,]  5.91402606 10.206294687  12.97180663

> summary(R)
       V1               V2               V3                V4
 Min.   :-8.608   Min.   :-6.836   Min.   :-30.082   Min.   :-1
 1st Qu.: 5.686   1st Qu.: 4.842   1st Qu.:  4.078   1st Qu.:
 Median : 8.551   Median : 7.597   Median :  7.260   Median :
 Mean   : 9.162   Mean   : 7.842   Mean   :  7.138   Mean   :
 3rd Qu.:12.471   3rd Qu.:10.984   3rd Qu.: 10.751   3rd Qu.: 1
 Max.   :42.808   Max.   :28.813   Max.   : 28.604   Max.   : 3
       V5                V6                V7               V8
 Min.   :-11.165   Min.   :-27.112   Min.   :-6.713   Min.   :-1
 1st Qu.:  3.489   1st Qu.:  4.153   1st Qu.: 5.717   1st Qu.:
 Median :  6.743   Median :  7.365   Median : 8.656   Median :
 Mean   :  6.532   Mean   :  7.139   Mean   : 9.079   Mean   :
 3rd Qu.:  9.723   3rd Qu.: 10.526   3rd Qu.:12.171   3rd Qu.: 1
 Max.   : 27.603   Max.   : 31.799   Max.   :31.714   Max.   : 3
>
```

```
## Birthday function
birthday <- function(n) {
  ntests <- 10000
  pop <- 1:365
  anydup <- function(i)
      any(duplicated(
                 sample(pop,
  sum(sapply(seq(ntests), a
}

registerDoSEQ()
system.time(
x <- foreach (j=1:100) %dopar% birthday (j)
)
--:**  foreach demo.R    66% L58     (ESS[S] [R])-----------
> birthday <- function(n) {
+   ntests <- 10000
+   pop <- 1:365
+   anydup <- function(i)
+       any(duplicated(
+   sample(pop, n, replace=TRUE)))
+   sum(sapply(seq(ntests), anydup)) / ntests
+ }
> registerDoSEQ()
> x <- foreach (j=1:100) %dopar% birthday (j)
>
-u:**  *R*            Bot L1238   (iESS [R]: run)-----------
```

```
data(Prestige, package="car")
LinearModel.1 <- lm(prestige ~ education + income + type , data=Prestige)
```

Enter name for model: LinearModel.2
Model Formula: prestige ~ education + income + type
Subset expression: <all valid cases>

```
Residual standard error: 7.095 on 93 degrees of freedom
Multiple R-Squared: 0.8349,     Adjusted R-squared: 0.8278
F-statistic: 117.5 on 4 and 93 DF,  p-value: < 2.2e-16

> trellis.device(theme="col.whitebg")

> plot(all.effects(LinearModel.1), ask=FALSE)

> scatter3d(Prestige$education, Prestige$prestige, Prestige$income, fit="additiv
```

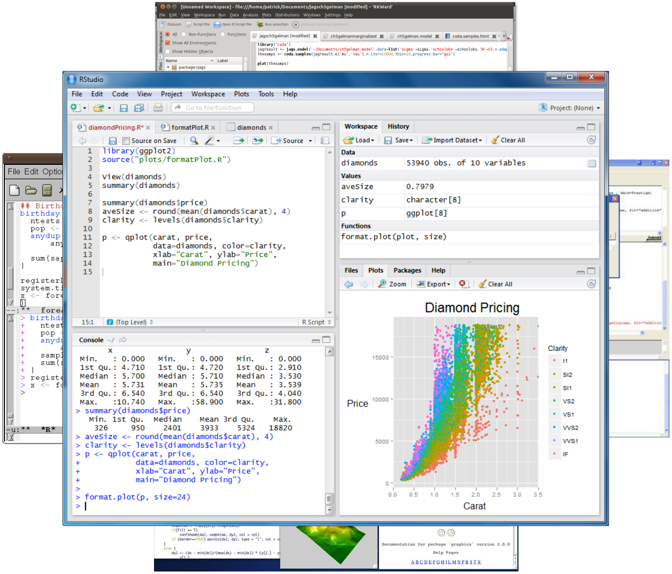
```
Loading required package: nlme
Loading required package: mvtnorm
Loading required package: multcomp
Loading required package: mgcv
This is mgcv 1.1-8
Loading required package: lmtest
Loading required package: lattice
Loading required package: foreign
Loading required package: effects
Loading required package: grid
Loading required package: car
Loading required package: abind
Loading required package: rgl

Rcmdr Version 0.9-18
>
```

```
BoxDens=function(data, npts = 200., x = c(0.,
    add = TRUE, col = 11., border=FALSE,collin
{
    dens <- density(data, n = npts)
    dx <- dens$x
    dy <- dens$y
    if(add == FALSE)
        plot(0., 0., axes = F, main = "", xlim = x, ylim = y,
            ylab = "")
    if(orientation == "paysage") {
        dx2 <- (dx - min(dx))/(max(dx) - min(dx)) * (x[2.] - x
            x[1.]
        dy2 <- (dy - min(dy))/(max(dy) - min(dy)) * (y[2.] - y
            y[1.]
        seqbelow <- rep(y[1.], length(dx))
        if(Fill == T)
            confshade(dx2, seqbelow, dy2, col = col)
        if (border==TRUE) points(dx2, dy2, type = "l", col = c
    }
    else {
        dy2 <- (dx - min(dx))/(max(dx) - min(dx)) * (y[2.] - y
```

68	146
69	150
70	154
71	159
72	164

status	Package	Description
✓ loaded	graphics	The R Graphics Package
not loaded	grid	The Grid Graphics Package
not loaded	lattice	Lattice Graphics
✓ loaded	methods	Formal Methods and Classes

R - interfaces



```
library("coda")
jagresult <- jags.model('~/Documents/ch5gelman.model',data=list('sigma'=sigma,'schoolobs'=schoolobs,'N'=8),n.adap
thesamps <- coda.samples(jagresult,c('mu','tau'),n.iter=10000,thin=10,progress.bar="gui")

plot(thesamps)
```

```
library(ggplot2)
source("plots/formatPlot.R")

View(diamonds)
summary(diamonds)

summary(diamonds$price)
aveSize <- round(mean(diamonds$carat), 4)
clarity <- levels(diamonds$clarity)

p <- qplot(carat, price,
           data=diamonds, color=clarity,
           xlab="Carat", ylab="Price",
           main="Diamond Pricing")
```

```
Data
diamonds        53940 obs. of 10 variables
Values
aveSize         0.7979
clarity         character[8]
p               ggplot[8]
Functions
format.plot(plot, size)
```

```
         x                y                z
 Min.   : 0.000   Min.   : 0.000   Min.   : 0.000
 1st Qu.: 4.710   1st Qu.: 4.720   1st Qu.: 2.910
 Median : 5.700   Median : 5.710   Median : 3.530
 Mean   : 5.731   Mean   : 5.735   Mean   : 3.539
 3rd Qu.: 6.540   3rd Qu.: 6.540   3rd Qu.: 4.040
 Max.   :10.740   Max.   :58.900   Max.   :31.800
> summary(diamonds$price)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
    326     950    2401    3933    5324   18820
> aveSize <- round(mean(diamonds$carat), 4)
> clarity <- levels(diamonds$clarity)
> p <- qplot(carat, price,
+            data=diamonds, color=clarity,
+            xlab="Carat", ylab="Price",
+            main="Diamond Pricing")
>
> format.plot(p, size=24)
>
```

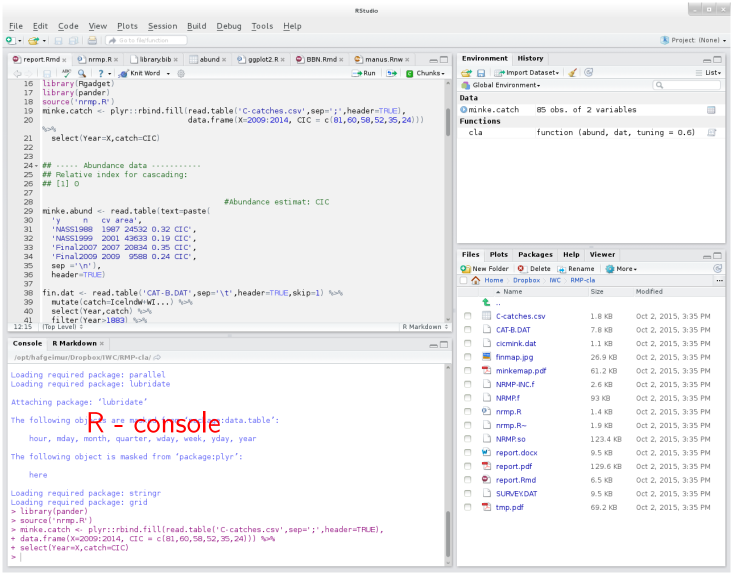
[Unnamed Workspace] - file:///home/patrick/Documents/jagsch5gelman [modified] – 'RKWard'
RStudio
File Edit Code View Project Workspace Plots Tools Help
diamondPricing.R*
formatPlot.R
diamonds
Source on Save
Source
Workspace
History
Import Dataset
Clear All
Console ~/
Files Plots Packages Help
Zoom
Export
Diamond Pricing
Price
Carat
Clarity
I1
SI2
SI1
VS2
VS1
VVS2
VVS1
IF
Project: (None)
Documentation for package `graphics' version 2.0.0
Help Pages

Overview

RStudio

- RStudio allows the user to run R in a more user-friendly environment. It is open- source (i.e. free) and available at `www.rstudio.com`
- built to help you write R code, run R code, and analyze data with R
- text editor, version control, project handling, markdown support, keyboard shortcuts, debugging tools, and much more

Using R-studio



```
library(Rgadget)
library(pander)
source('nrmp.R')
minke.catch <- plyr::rbind.fill(read.table('C-catches.csv',sep=';',header=TRUE),
                                data.frame(X=2009:2014, CIC = c(81,60,58,52,35,24)))
%>%
  select(Year=X,catch=CIC)


## ----- Abundance data -----------
## Relative index for cascading:
## [1] 0

                                        #Abundance estimat: CIC
minke.abund <- read.table(text=paste(
  'y      n    cv area',
  'NASS1988  1987 24532 0.32 CIC',
  'NASS1999  2001 43633 0.19 CIC',
  'Final2007 2007 20834 0.35 CIC',
  'Final2009 2009  9588 0.24 CIC',
  sep ='\n'),
  header=TRUE)

fin.dat <- read.table('CAT-B.DAT',sep='\t',header=TRUE,skip=1) %>%
  mutate(catch=IcelndW+WI...) %>%
  select(Year,catch) %>%
  filter(Year>1883) %>%
```

Data	
minke.catch	85 obs. of 2 variables
Functions	
cla	function (abund, dat, tuning = 0.6)

Name	Size	Modified
..		
C-catches.csv	1.8 KB	Oct 2, 2015, 3:35 PM
CAT-B.DAT	7.8 KB	Oct 2, 2015, 3:35 PM
cicmink.dat	1.1 KB	Oct 2, 2015, 3:35 PM
finmap.jpg	26.9 KB	Oct 2, 2015, 3:35 PM
minkemap.pdf	61.2 KB	Oct 2, 2015, 3:35 PM
NRMP-INC.f	2.6 KB	Oct 2, 2015, 3:35 PM
NRMP.f	93 KB	Oct 2, 2015, 3:35 PM
nrmp.R	1.4 KB	Oct 2, 2015, 3:35 PM
nrmp.R~	1.9 KB	Oct 2, 2015, 3:35 PM
NRMP.so	123.4 KB	Oct 2, 2015, 3:35 PM
report.docx	9.5 KB	Oct 2, 2015, 3:35 PM
report.pdf	129.6 KB	Oct 2, 2015, 3:35 PM
report.Rmd	6.5 KB	Oct 2, 2015, 3:35 PM
SURVEY.DAT	9.5 KB	Oct 2, 2015, 3:35 PM
tmp.pdf	69.2 KB	Oct 2, 2015, 3:35 PM

Console　R Markdown

/opt/hafgeimur/Dropbox/IWC/RMP-cla/

```
Loading required package: parallel
Loading required package: lubridate

Attaching package: ‘lubridate’

The following objects are masked from ‘package:data.table’:

    hour, mday, month, quarter, wday, week, yday, year

The following object is masked from ‘package:plyr’:

    here

Loading required package: stringr
Loading required package: grid
> library(pander)
> source('nrmp.R')
> minke.catch <- plyr::rbind.fill(read.table('C-catches.csv',sep=';',header=TRUE),
+ data.frame(X=2009:2014, CIC = c(81,60,58,52,35,24))) %>%
+ select(Year=X,catch=CIC)
> 
```

R - console

R-console

One can type commands directly into the console to perform calculations:

```
1+1 ## this should be 2

## [1] 2

mean(c(1,2,3,4,5)) ## mean of all numbers from 1 to 5

## [1] 3
```

and assign the results with the "<-" sign:

```
x <- 1+1 ## save the result into a variable named x
x        ## print out the value of x to screen

## [1] 2
```

Using R as a calculator

R can be used as calculator:

```
1+1         # 1 plus 1 equals 2;)
2*2         # 2 times 2 equals 4
3^2.5       # 3 to the power of 2.5
exp(5)      # e to the power of 5
log(5)      # natural logarithm (ln) of 5
log10(5)    # Base 10 log of 5
1/2         # 1 over 2 equals 0.5
```

Data types

R has a number of data types to handle the various inputs from the user:

- Numbers: integers, reals and complex
- Strings: letters, words, files etc..
- Logicals: TRUE or FALSE
- Factors: integer numbers that correspond to a fixed (limited) set of values
- NA's: Not Available, used when data is missing

Data types

```r
x <- 1.2          ## number
y <- 'A'          ## string
z <- TRUE         ## logical
w <- factor('A') ## factor
```

Data structures

One of R's many strengths are multiple data structures:

- **vectors** (1d)
- matricies (2d)
- arrays (nd)
- **dataframes** (2d)
- lists (1d, sort of)
- ...

Vectors

Vectors can contain:

- Numbers
- Strings
- Logicals
- NA's

```
x <- c(1,2,3)       ## vector of numbers
y <- c('a','b','c') ## vector of strings
z <- c(TRUE,FALSE)  ## vector of locigals
class(x)            ## returns the type of vector
u <- NA             ## NA
```

Useful commands for vectors

```
seq(1,10)         ## makes a vector from 1 to 10
1:10              ## same
rep(2,3)          ## makes a vector that contains 2, 3 times
length(y)         ## returns the length of a vecor named y
sort(y)           ## sorts a vector into ascending or
                  ## descending order
cut(y)            ## divides the range of y into intervals and
                  ## makes a factor variable
as.character(y)   ## changes y to a character vector
cbind(x1,x2)      ## binds vectors by columns
rbind(x1,x2)      ## binds vectors by rows
```

Useful commands for vectors (II)

```
mean()          # mean
median()        # median
quantile()      # quantiles
summary()       # depends on what we feed it with...
sd()            # standard deviation
var()           # variance
range()         # range
min()           # smallest value
max()           # largest value
```

Logical statements

```r
a == b    ## a is equal to b
a != b    ## a is not equal to b
a > b     ## a is greater than b
a >= b    ## a is greater or equal to b
a < b     ## a is less than b
a <= b    ## a is less or equal to b
a & b     ## a and b
a | b     ## a or b
!a        ## not a
is.na(a) ## is a equal to NA (missing)
```

Selecting parts of a vector

```
x[1]          ## get the first element of a vector
x[c(1,2)]     ## get the first and second element
x[-1]         ## get every element except the first one
x[-c(1,2)]    ## get every element except 1 and 2
x[x==3]       ## all elements that are equal to 3
x[!is.na(x)]  ## all elements that are not NA
x[x!=3&x<5]   ## all elements that not equal to 3 and <5
```

Data-frames

- Data frames are collections of vectors (columns) of the same length, similar to a table in a database
- The columns do not need to be of the same type
- One can access the columns:

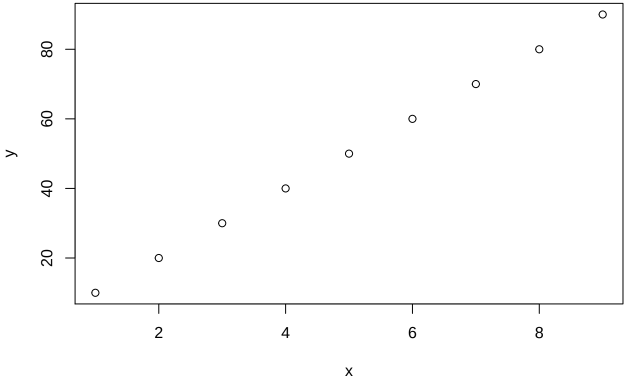
```r
data$Col.name              ## get column named Col.name
data[,1]                   ## get column number 1
```

- and the entries in the column:

```r
data$Col.name[1]        ## get entry number 1 from Col.name
data[2,1]               ## get entry number 2 from column 1
```

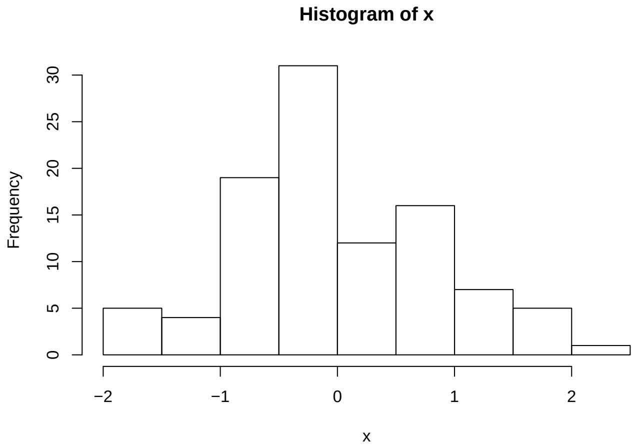
Base R scatter plot

```
x <- 1:9
y <- 10*(1:9)
plot(x,y)
```

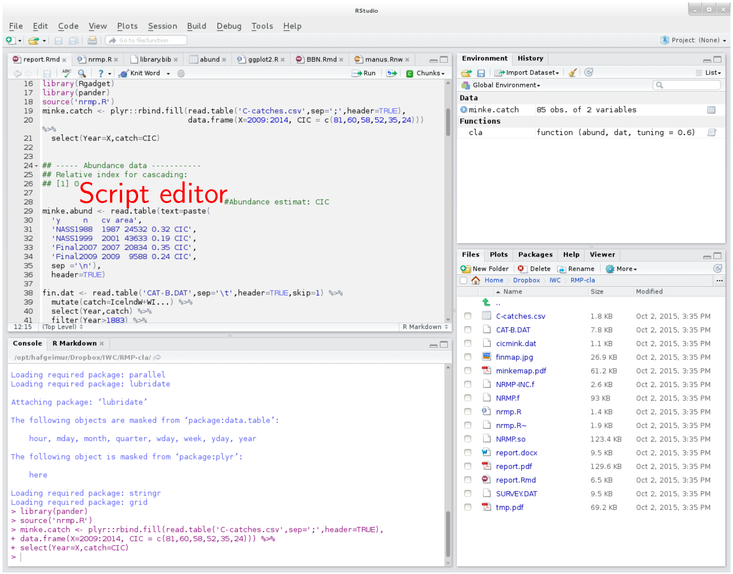


Base R histogram

```r
x <- rnorm(100)
hist(x)
```



Using R-studio



```
library(Rgadget)
library(pander)
source('nrmp.R')
minke.catch <- plyr::rbind.fill(read.table('C-catches.csv',sep=';',header=TRUE),
                                data.frame(X=2009:2014, CIC = c(81,60,58,52,35,24)))
%>%
  select(Year=X,catch=CIC)


## ----- Abundance data -----------
## Relative index for cascading:
## [1] 0

                                     #Abundance estimat: CIC
minke.abund <- read.table(text=paste(
  'y      n   cv area',
  'NASS1988  1987 24532 0.32 CIC',
  'NASS1999  2001 43633 0.19 CIC',
  'Final2007 2007 20834 0.35 CIC',
  'Final2009 2009  9588 0.24 CIC',
  sep ='\n'),
  header=TRUE)

fin.dat <- read.table('CAT-B.DAT',sep='\t',header=TRUE,skip=1) %>%
  mutate(catch=IcelndW+WI...) %>%
  select(Year,catch) %>%
  filter(Year>1883) %>%
```

```
/opt/hafgeimur/Dropbox/IWC/RMP-cla/

Loading required package: parallel
Loading required package: lubridate

Attaching package: ‘lubridate’

The following objects are masked from ‘package:data.table’:

    hour, mday, month, quarter, wday, week, yday, year

The following object is masked from ‘package:plyr’:

    here

Loading required package: stringr
Loading required package: grid
> library(pander)
> source('nrmp.R')
> minke.catch <- plyr::rbind.fill(read.table('C-catches.csv',sep=';',header=TRUE),
+ data.frame(X=2009:2014, CIC = c(81,60,58,52,35,24))) %>%
+ select(Year=X,catch=CIC)
>
```

```
Data
minke.catch    85 obs. of 2 variables
Functions
  cla          function (abund, dat, tuning = 0.6)
```

Name	Size	Modified
..		
C-catches.csv	1.8 KB	Oct 2, 2015, 3:35 PM
CAT-B.DAT	7.8 KB	Oct 2, 2015, 3:35 PM
cicmink.dat	1.1 KB	Oct 2, 2015, 3:35 PM
finmap.jpg	26.9 KB	Oct 2, 2015, 3:35 PM
minkemap.pdf	61.2 KB	Oct 2, 2015, 3:35 PM
NRMP-INC.f	2.6 KB	Oct 2, 2015, 3:35 PM
NRMP.f	93 KB	Oct 2, 2015, 3:35 PM
nrmp.R	1.4 KB	Oct 2, 2015, 3:35 PM
nrmp.R~	1.9 KB	Oct 2, 2015, 3:35 PM
NRMP.so	123.4 KB	Oct 2, 2015, 3:35 PM
report.docx	9.5 KB	Oct 2, 2015, 3:35 PM
report.pdf	129.6 KB	Oct 2, 2015, 3:35 PM
report.Rmd	6.5 KB	Oct 2, 2015, 3:35 PM
SURVEY.DAT	9.5 KB	Oct 2, 2015, 3:35 PM
tmp.pdf	69.2 KB	Oct 2, 2015, 3:35 PM

Working in R

- In R we use *commands* to create new *objects*. When we do that we have a choice to do two things:
  - The command spits out the results to the screen and then these results forgotten
  - The command saves the output so we can reuse it.
- R can't save the object without giving it a name

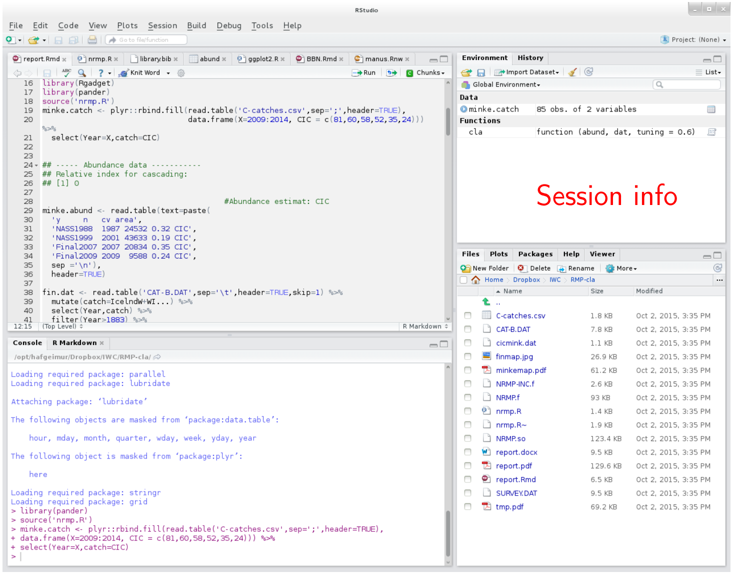
```
name <- command
```

- Note that, like in real life, there is not undo button
- One should rather "remember" the commands used to create the object

Script editor

- Is opened when selecting File->New file->R script from the menu panel
- A R script is basically a series of R commands that can be run in the console
- It is recommended to store all commands that "work" in a script for later reconstruction of a particular analysis
- It is possible to run a single line in a script by pressing [ctrl]+Enter or the Run button in the top right corner

Using R-studio



```
library(Rgadget)
library(pander)
source('nrmp.R')
minke.catch <- plyr::rbind.fill(read.table('C-catches.csv',sep=';',header=TRUE),
                                data.frame(X=2009:2014, CIC = c(81,60,58,52,35,24)))
%>%
  select(Year=X,catch=CIC)


## ----- Abundance data -----------
## Relative index for cascading:
## [1] 0

                                         #Abundance estimat: CIC
minke.abund <- read.table(text=paste(
  'y      n    cv area',
  'NASS1988  1987 24532 0.32 CIC',
  'NASS1999  2001 43633 0.19 CIC',
  'Final2007 2007 20834 0.35 CIC',
  'Final2009 2009  9588 0.24 CIC',
  sep ='\n'),
  header=TRUE)

fin.dat <- read.table('CAT-B.DAT',sep='\t',header=TRUE,skip=1) %>%
  mutate(catch=IcelndW+WI...) %>%
  select(Year,catch) %>%
  filter(Year>1883) %>%
```

Data	
minke.catch	85 obs. of 2 variables
Functions	
cla	function (abund, dat, tuning = 0.6)

```
/opt/hafgeimur/Dropbox/IWC/RMP-cla/

Loading required package: parallel
Loading required package: lubridate

Attaching package: ‘lubridate’

The following objects are masked from ‘package:data.table’:

    hour, mday, month, quarter, wday, week, yday, year

The following object is masked from ‘package:plyr’:

    here

Loading required package: stringr
Loading required package: grid
> library(pander)
> source('nrmp.R')
> minke.catch <- plyr::rbind.fill(read.table('C-catches.csv',sep=';',header=TRUE),
+ data.frame(X=2009:2014, CIC = c(81,60,58,52,35,24))) %>%
+ select(Year=X,catch=CIC)
> 
```

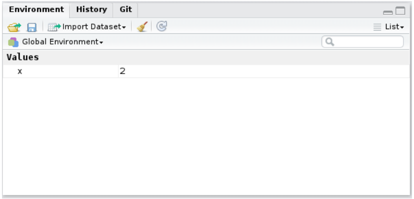
Name	Size	Modified
..		
C-catches.csv	1.8 KB	Oct 2, 2015, 3:35 PM
CAT-B.DAT	7.8 KB	Oct 2, 2015, 3:35 PM
cicmink.dat	1.1 KB	Oct 2, 2015, 3:35 PM
finmap.jpg	26.9 KB	Oct 2, 2015, 3:35 PM
minkemap.pdf	61.2 KB	Oct 2, 2015, 3:35 PM
NRMP-INC.f	2.6 KB	Oct 2, 2015, 3:35 PM
NRMP.f	93 KB	Oct 2, 2015, 3:35 PM
nrmp.R	1.4 KB	Oct 2, 2015, 3:35 PM
nrmp.R~	1.9 KB	Oct 2, 2015, 3:35 PM
NRMP.so	123.4 KB	Oct 2, 2015, 3:35 PM
report.docx	9.5 KB	Oct 2, 2015, 3:35 PM
report.pdf	129.6 KB	Oct 2, 2015, 3:35 PM
report.Rmd	6.5 KB	Oct 2, 2015, 3:35 PM
SURVEY.DAT	9.5 KB	Oct 2, 2015, 3:35 PM
tmp.pdf	69.2 KB	Oct 2, 2015, 3:35 PM

Session info

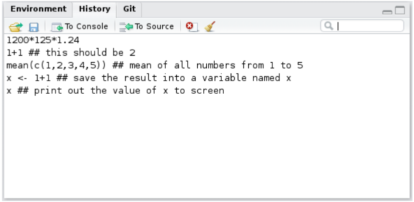
Session info

- **Environment**: Contains a list of all declared variables. If you have a dataset you can double click the variable to view the data.

- **History**: Contains a list of all previous commands entered into the console.



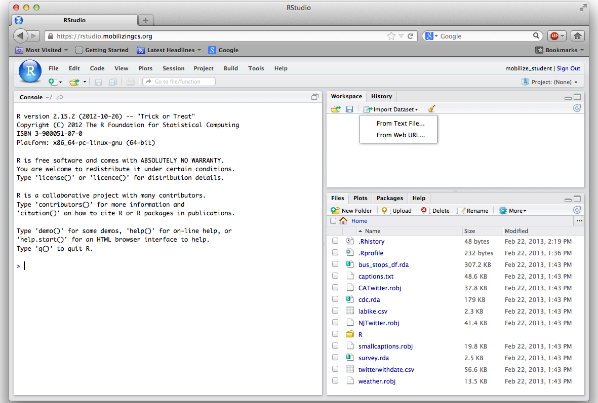
```
Values
  x                2
```



```
1200*125*1.24
1+1 ## this should be 2
mean(c(1,2,3,4,5)) ## mean of all numbers from 1 to 5
x <- 1+1 ## save the result into a variable named x
x ## print out the value of x to screen
```

Getting data into R

The environment tab allows you also to import data into R:

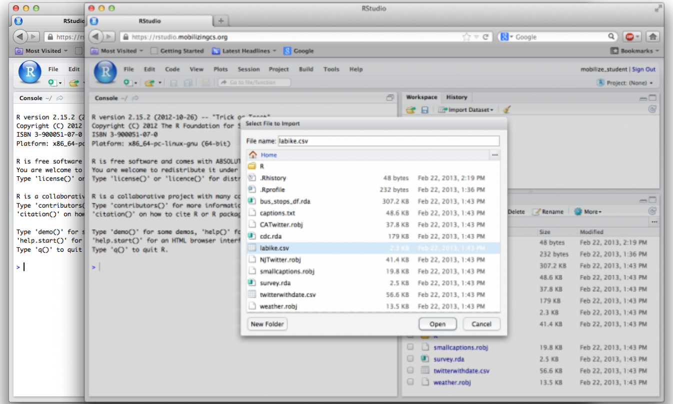

RStudio
https://rstudio.mobilizingcs.org
Most Visited
Getting Started
Latest Headlines
Google
Bookmarks
File Edit Code View Plots Session Project Build Tools Help
mobilize_student | Sign Out
Go to file/function
Project: (None)
Console ~/
Workspace History
Import Dataset
From Text File...
From Web URL...
Files Plots Packages Help
New Folder Upload Delete Rename More
Home

```
R version 2.15.2 (2012-10-26) -- "Trick or Treat"
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

>
```

Name	Size	Modified
.Rhistory	48 bytes	Feb 22, 2013, 2:19 PM
.Rprofile	232 bytes	Feb 22, 2013, 1:36 PM
bus_stops_df.rda	307.2 KB	Feb 22, 2013, 1:43 PM
captions.txt	48.6 KB	Feb 22, 2013, 1:43 PM
CATwitter.robj	37.8 KB	Feb 22, 2013, 1:43 PM
cdc.rda	179 KB	Feb 22, 2013, 1:43 PM
labike.csv	2.3 KB	Feb 22, 2013, 1:43 PM
NJTwitter.robj	41.4 KB	Feb 22, 2013, 1:43 PM
R		
smallcaptions.robj	19.8 KB	Feb 22, 2013, 1:43 PM
survey.rda	2.5 KB	Feb 22, 2013, 1:43 PM
twitterwithdate.csv	56.6 KB	Feb 22, 2013, 1:43 PM
weather.robj	13.5 KB	Feb 22, 2013, 1:43 PM

Getting data into R

The environment tab allows you also to import data into R:



```
R version 2.15.2 (2012-10-26) -- "Trick or Treat"
Copyright (C) 2012 The R Foundation for S
ISBN 3-900051-07-0
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUT
You are welcome to redistribute it under
Type 'license()' or 'licence()' for distr

R is a collaborative project with many co
Type 'contributors()' for more informatio
'citation()' on how to cite R or R packag

Type 'demo()' for some demos, 'help()' fo
'help.start()' for an HTML browser interf
Type 'q()' to quit R.

>
```

Select File to Import

File name: labike.csv

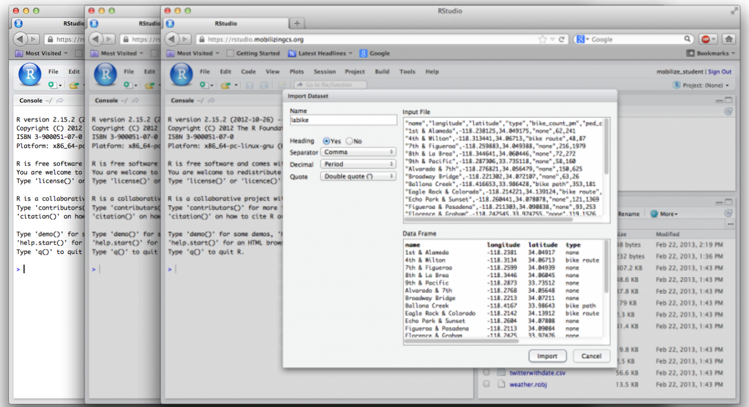
Home

R		
.Rhistory	48 bytes	Feb 22, 2013, 2:19 PM
.Rprofile	232 bytes	Feb 22, 2013, 1:36 PM
bus_stops_df.rda	307.2 KB	Feb 22, 2013, 1:43 PM
captions.txt	48.6 KB	Feb 22, 2013, 1:43 PM
CATwitter.robj	37.8 KB	Feb 22, 2013, 1:43 PM
cdc.rda	179 KB	Feb 22, 2013, 1:43 PM
labike.csv	2.3 KB	Feb 22, 2013, 1:43 PM
NJTwitter.robj	41.4 KB	Feb 22, 2013, 1:43 PM
smallcaptions.robj	19.8 KB	Feb 22, 2013, 1:43 PM
survey.rda	2.5 KB	Feb 22, 2013, 1:43 PM
twitterwithdate.csv	56.6 KB	Feb 22, 2013, 1:43 PM
weather.robj	13.5 KB	Feb 22, 2013, 1:43 PM

New Folder | Open | Cancel

Getting data into R

The environment tab allows you also to import data into R:


RStudio
File Edit Code View Plots Session Project Build Tools Help
mobilize_student | Sign Out
Project: (None)
Console ~/
Import Dataset
Import
Cancel
Rename
More

```
R version 2.15.2 (2012-10-26) --
Copyright (C) 2012 The R Foundat
ISBN 3-900051-07-0
Platform: x86_64-pc-linux-gnu (6

R is free software and comes wit
You are welcome to redistribute
Type 'license()' or 'licence()'

R is a collaborative project wit
Type 'contributors()' for more i
'citation()' on how to cite R or

Type 'demo()' for some demos, 'h
'help.start()' for an HTML brows
Type 'q()' to quit R.

> |
```

Name: labike

Heading: [✓ Yes] No

Separator: Comma

Decimal: Period

Quote: Double quote (")

Input File

```
"name","longitude","latitude","type","bike_count_pm","ped_c
"1st & Alameda",-118.238125,34.049175,"none",62,241
"4th & Wilton",-118.313441,34.06713,"bike route",48,87
"7th & Figueroa",-118.259883,34.049388,"none",216,1979
"8th & La Brea",-118.344641,34.060446,"none",72,272
"9th & Pacific",-118.287306,33.735118,"none",58,160
"Alvarado & 7th",-118.276821,34.056479,"none",150,625
"Broadway Bridge",-118.221302,34.072107,"none",63,26
"Ballona Creek",-118.416653,33.986428,"bike path",353,181
"Eagle Rock & Colorado",-118.214221,34.139124,"bike route",
"Echo Park & Sunset",-118.260441,34.078078,"none",121,1369
"Figueroa & Pasadena",-118.211303,34.090838,"none",93,253
"Florence & Graham",-118.242545,33.924755,"none",119,1526
```

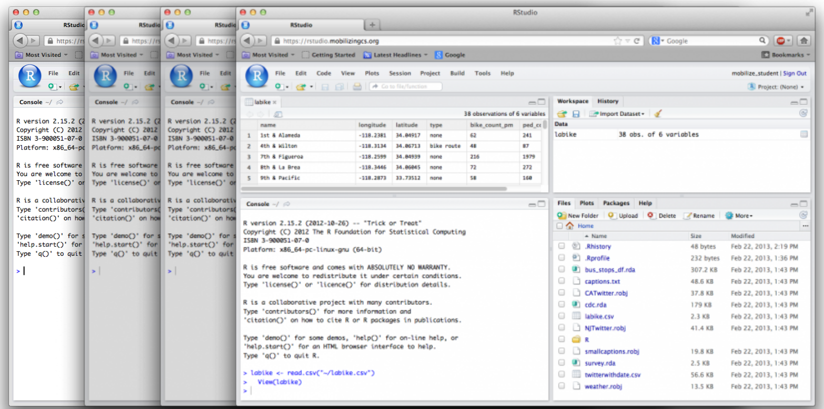
Data Frame

```
name                   longitude  latitude  type
1st & Alameda          -118.2381  34.04917  none
4th & Wilton           -118.3134  34.06713  bike route
7th & Figueroa         -118.2599  34.04939  none
8th & La Brea          -118.3446  34.06045  none
9th & Pacific          -118.2873  33.73512  none
Alvarado & 7th         -118.2768  34.05648  none
Broadway Bridge        -118.2213  34.07211  none
Ballona Creek          -118.4167  33.98643  bike path
Eagle Rock & Colorado  -118.2142  34.13912  bike route
Echo Park & Sunset     -118.2604  34.07808  none
Figueroa & Pasadena    -118.2113  34.09084  none
Florence & Graham      -118.2425  33.97476  none
```

twitterwithdate.csv	56.6 KB	Feb 22, 2013, 1:43 PM
weather.robj	13.5 KB	Feb 22, 2013, 1:43 PM

Getting data into R

The environment tab allows you also to import data into R:



38 observations of 6 variables

	name	longitude	latitude	type	bike_count_pm	ped_cc
1	1st & Alameda	-118.2381	34.04917	none	62	241
2	4th & Wilton	-118.3134	34.06713	bike route	48	87
3	7th & Figueroa	-118.2599	34.04939	none	216	1979
4	8th & La Brea	-118.3446	34.06045	none	72	272
5	9th & Pacific	-118.2873	33.73512	none	58	160

Data

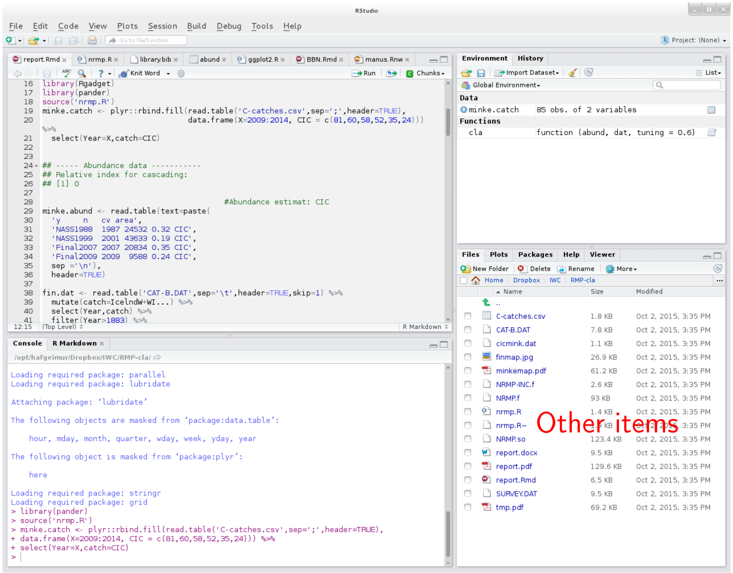
```
labike          38 obs. of 6 variables
```

Console ~/

```
R version 2.15.2 (2012-10-26) -- "Trick or Treat"
Copyright (C) 2012 The R Foundation for Statistical Computing
ISBN 3-900051-07-0
Platform: x86_64-pc-linux-gnu (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> labike <- read.csv("~/labike.csv")
>   View(labike)
> 
```

Name	Size	Modified
.Rhistory	48 bytes	Feb 22, 2013, 2:19 PM
.Rprofile	232 bytes	Feb 22, 2013, 1:36 PM
bus_stops_df.rda	307.2 KB	Feb 22, 2013, 1:43 PM
captions.txt	48.6 KB	Feb 22, 2013, 1:43 PM
CATwitter.robj	37.8 KB	Feb 22, 2013, 1:43 PM
cdc.rda	179 KB	Feb 22, 2013, 1:43 PM
labike.csv	2.3 KB	Feb 22, 2013, 1:43 PM
NJTwitter.robj	41.4 KB	Feb 22, 2013, 1:43 PM
R		
smallcaptions.robj	19.8 KB	Feb 22, 2013, 1:43 PM
survey.rda	2.5 KB	Feb 22, 2013, 1:43 PM
twitterwithdate.csv	56.6 KB	Feb 22, 2013, 1:43 PM
weather.robj	13.5 KB	Feb 22, 2013, 1:43 PM

Using R-studio



```
library(Rgadget)
library(pander)
source('nrmp.R')
minke.catch <- plyr::rbind.fill(read.table('C-catches.csv',sep=';',header=TRUE),
                                data.frame(X=2009:2014, CIC = c(81,60,58,52,35,24)))
%>%
  select(Year=X,catch=CIC)


## ----- Abundance data -----------
## Relative index for cascading:
## [1] 0

                                        #Abundance estimat: CIC
minke.abund <- read.table(text=paste(
  'y      n   cv area',
  'NASS1988  1987 24532 0.32 CIC',
  'NASS1999  2001 43633 0.19 CIC',
  'Final2007 2007 20834 0.35 CIC',
  'Final2009 2009  9588 0.24 CIC',
  sep ='\n'),
  header=TRUE)

fin.dat <- read.table('CAT-B.DAT',sep='\t',header=TRUE,skip=1) %>%
  mutate(catch=IcelndW+WI...) %>%
  select(Year,catch) %>%
  filter(Year>1883) %>%
```

Console | R Markdown

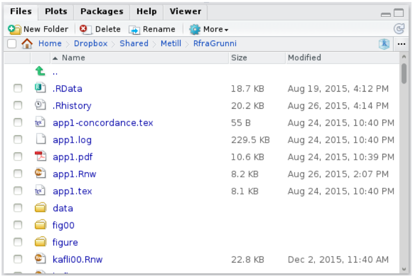
```
/opt/hafgeimur/Dropbox/IWC/RMP-cla/

Loading required package: parallel
Loading required package: lubridate

Attaching package: ‘lubridate’

The following objects are masked from ‘package:data.table’:

    hour, mday, month, quarter, wday, week, yday, year

The following object is masked from ‘package:plyr’:

    here

Loading required package: stringr
Loading required package: grid
> library(pander)
> source('nrmp.R')
> minke.catch <- plyr::rbind.fill(read.table('C-catches.csv',sep=';',header=TRUE),
+ data.frame(X=2009:2014, CIC = c(81,60,58,52,35,24))) %>%
+ select(Year=X,catch=CIC)
> 
```

```
Data
minke.catch      85 obs. of 2 variables
Functions
  cla            function (abund, dat, tuning = 0.6)
```

Name	Size	Modified
..		
C-catches.csv	1.8 KB	Oct 2, 2015, 3:35 PM
CAT-B.DAT	7.8 KB	Oct 2, 2015, 3:35 PM
cicmink.dat	1.1 KB	Oct 2, 2015, 3:35 PM
finmap.jpg	26.9 KB	Oct 2, 2015, 3:35 PM
minkemap.pdf	61.2 KB	Oct 2, 2015, 3:35 PM
NRMP-INC.f	2.6 KB	Oct 2, 2015, 3:35 PM
NRMP.f	93 KB	Oct 2, 2015, 3:35 PM
nrmp.R	1.4 KB	Oct 2, 2015, 3:35 PM
nrmp.R~	1.3 KB	Oct 2, 2015, 3:35 PM
NRMP.so	123.4 KB	Oct 2, 2015, 3:35 PM
report.docx	9.5 KB	Oct 2, 2015, 3:35 PM
report.pdf	129.6 KB	Oct 2, 2015, 3:35 PM
report.Rmd	6.5 KB	Oct 2, 2015, 3:35 PM
SURVEY.DAT	9.5 KB	Oct 2, 2015, 3:35 PM
tmp.pdf	69.2 KB	Oct 2, 2015, 3:35 PM

Other items–files

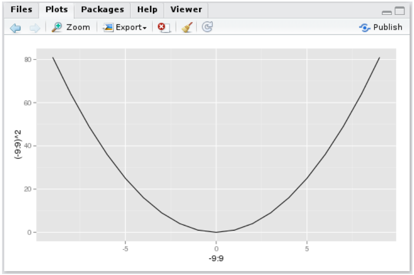
Files: Shows files on the computer. The user can change the working directory (where R reads data-files and scripts from) by selecting "more->Set as working directory"



Name	Size	Modified
..		
.RData	18.7 KB	Aug 19, 2015, 4:12 PM
.Rhistory	20.2 KB	Aug 26, 2015, 4:14 PM
appl-concordance.tex	55 B	Aug 24, 2015, 10:40 PM
appl.log	229.5 KB	Aug 24, 2015, 10:40 PM
appl.pdf	10.6 KB	Aug 24, 2015, 10:39 PM
appl.Rnw	8.2 KB	Aug 26, 2015, 2:07 PM
appl.tex	8.1 KB	Aug 24, 2015, 10:40 PM
data		
fig00		
figure		
kafli00.Rnw	22.8 KB	Dec 2, 2015, 11:40 AM

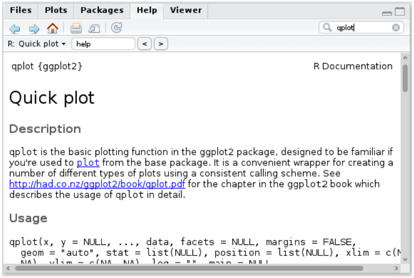
Other items–plots

Plots: Graphical output from R. The user can export these the figures to file (as jpeg, png or pdf) or to clipboard



Other items–help

Help: Gives a browsable interface to R's in-line help pages. The user can search for a topic or a specific function



qplot {ggplot2} R Documentation

Quick plot

Description

qplot is the basic plotting function in the ggplot2 package, designed to be familiar if you're used to plot from the base package. It is a convenient wrapper for creating a number of different types of plots using a consistent calling scheme. See http://had.co.nz/ggplot2/book/qplot.pdf for the chapter in the ggplot2 book which describes the usage of qplot in detail.

Usage

```
qplot(x, y = NULL, ..., data, facets = NULL, margins = FALSE,
  geom = "auto", stat = list(NULL), position = list(NULL), xlim = c(N
```

On help

A help page for a function usually has the following categories:

- **Description**: short description of the function
- **Usage**: how the function is used/called from R
- **Arguments**: What are the function inputs
- **Details**: The nitty gritty on how the function should work
- **Examples**: example uses
- **See also**: other useful functions that are similar

Help commands

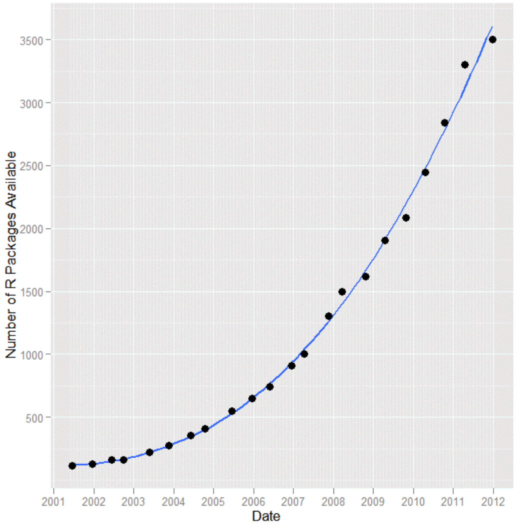
One can also get help from the console:

```
help.start()           # General help page
help(nameOffuntion)    # Gives the help page needed
?nameOffunction        # shorthand
help.search('topic')   # searches for specific topic
```

Packages

- One of R's greatest strengths comes from specialized add-ons, called *packages* that the user can download and install
- Users download packages from CRAN's website (Comprehensive R Archive Network)
- Currently at 7811 packages (January 20th 2016)
- All packages comes with both an in-line and pdf manual

Package growth



MRI's R packages

- geo – nautical charts (also available on CRAN)
- ora – data base connectivity
- fjolst – all samples from commercial and survey vessels
- Logbooks – logbooks from the commercial fishery
- Rgadget – specialised package to interact with Gadget

Using packages

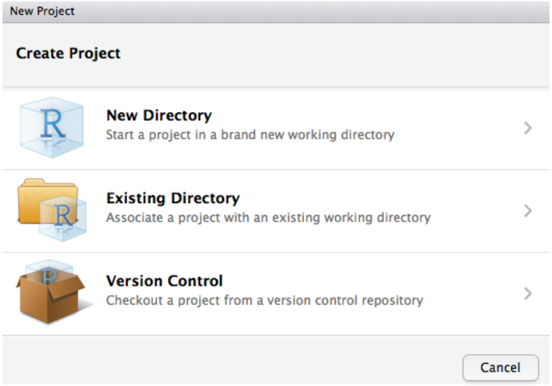
- One can use the `install.packages` command to download and install packages
- To use the package in an R session one needs to load it into memory using the `library` command
- In R-studio one can also select, load and install packages in the "other items" part

```
install.packages('ggplot2') ## install package named ggplot2
library(ggplot2)            ## load it into memory
```

Projects

- Rstudio allows us to make things a little bit easier to isolate the task we are working with a any given point by defining specific **projects**
- Projects save the state between sessions. This includes:
  - Working directories
  - Open files/scripts
  - Workspaces
  - Color scheme
- Projects are created by selecting File->New project

Creating projects


New Project

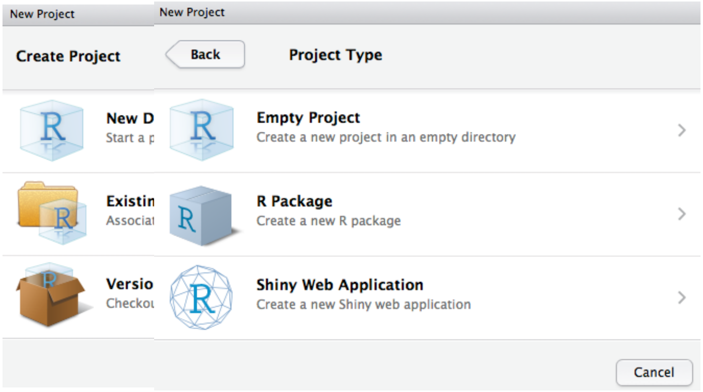
Create Project

New Directory
Start a project in a brand new working directory

Existing Directory
Associate a project with an existing working directory

Version Control
Checkout a project from a version control repository

Creating projects



New Project

Create Project

New D
Start a p

Existin
Associat

Versio
Checkou

New Project

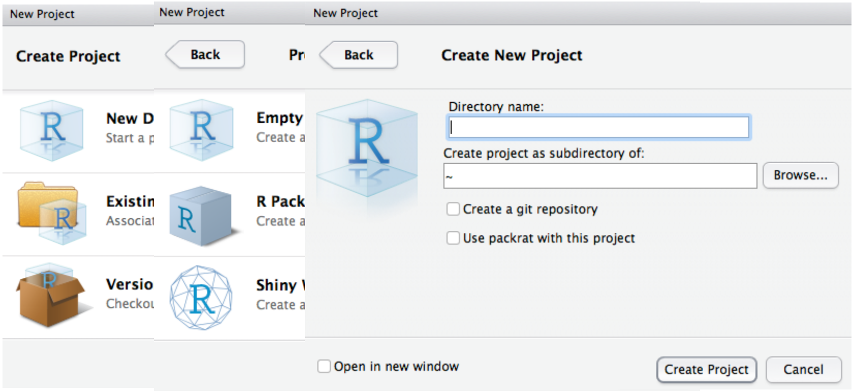
Project Type

Empty Project
Create a new project in an empty directory

R Package
Create a new R package

Shiny Web Application
Create a new Shiny web application

Creating projects


New Project

Create Project

New D
Start a p

Existin
Associat

Versio
Checkou

New Project

Back

Pr

Empty
Create a

R Pack
Create a

Shiny
Create a

New Project

Back

Create New Project

Directory name:

Create project as subdirectory of:

~

Browse...

Create a git repository

Use packrat with this project

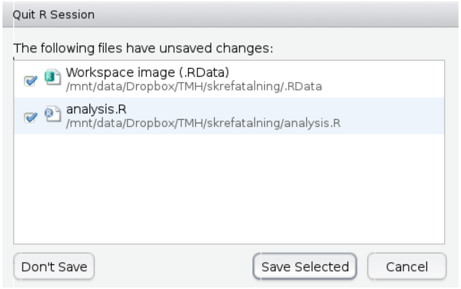
Open in new window

Create Project

Cancel

Turning R off

When shutting down R and Rstudio it reminds you to save your work:


Quit R Session

The following files have unsaved changes:

- ✓ Workspace image (.RData) /mnt/data/Dropbox/TMH/skrefatalning/.RData
- ✓ analysis.R /mnt/data/Dropbox/TMH/skrefatalning/analysis.R

In general you should **not** save your workspace unless the analysis takes a long time to reproduce.

Don't save your workspace

Excercise 1

- Create a new directory anywhere on your computer and names it "Classwork"
- Open R-studio and create a new project called "classEx1" and associate it with the newly created directory
- Install lubridate, stringr,geo, ggplot2, tidyr, dplyr and gridExtra if you haven't done so already
- Create a new script named 'Ex1.R'
- Look at the help page for 'sum', 'mean' and 'sd'
- Read in the minke whale data using the "import dataset" menu
- Write a script that calculates the sum, mean and standard deviation of the stomach.weight of the minke whales

Overview

if sentences

- It is often so that we would like some parts of our code to run only if some condition is fulfilled
- We use `if` sentences for that
- When setting up the conditions the operators come in handy

```
if(condition){
  ## this runs if the condition(s) are fulfilled
}
```

if sentences - example

```
x <- 10
## check if x is greater than 0
if(x>0){
    print('x is positive')
}
```

if and else sentences

- Sometimes we would like to check for more than one condition
- We use a combination of `if` and `else` sentences for that

```
if(condition_1){
  ## this runs if condition 1 is fulfilled,
  ## then we skip the else-sentences below
} else if(condition_2){
  ## this runs if condition 2 is fulfilled,
  ## then we skip the else-sentence below
} else{
  ## this runs of neither condition 1 or 2 are fulfilled
}
```

if - else sentences - example

```r
x <- 10

## check if x > 10
if(x>10){
    print('x is greater than 10')
} else if(x>=0){
    print('x is less than 10 but greater than or equal to 0')
} else {
  print('x is negative')
}
```

ifelse sentences

- It is often so that we only have two conditions
- We use `ifelse` sentences for these cases

```
ifelse(condition,
       'this runs if the condition is fulfilled',
       'this runs if the condition is not fulfilled')
```

ifelse sentences - example

Replace NA's from data

```
x <- c(1,2,3,NA,4,5)
## find all missing entries and replace them
x <- ifelse(is.na(x),0,x)
```

for-loop

- We use a `for`-loop if we need some part of our code to run more than one time and we know how often it sould run
- We use an index/counter that counts how often we enter the loop
- We try to avoid `for`-loops because they are rather slow
- There are several commands in R that help us to avoid for-loops - see the end of the slides

```
for(index in vector){
  ## This code runs as often as the length of the vector
}
```

while-loop

- While-loops are similar to for-loops
- We use a `while`-loop if we need some part of our code to run more than one time but we dont know how often it sould run
- The loop runs while the condition stated is fulfilled

```
while(condition){
  ## This code runs while the condition(s) are fulfilled
}
```

Loops, examples

```r
## find the number of entries in the data
n <- nrow(minke)

## calculate the mean length old fashion way
ML <- 0
for(i in 1:n){
  ML <- ML + dat$length[i]
}
ML <- ML/n
```

User–defined functions in R

- Even though there are a great number of built in functions/commands/methods in R we sometimes need to write our own functions
- We use `function` for that
- We can write a name of a built in function in the console to to get the code behind it
- We can take the code and change it as we please

function

- A function has a name, takes arguments/settings and returns an object
- Object within the function are local within the function
- Objects the function returns can be of any type

```
nameOfFunction <- function(argument1, argument2, ... ){
  ## The code of the function
  return(object)
}
```

Note that you can name your function whatever you want

function

```
add1 <- function(x){
  x <- x + 1
  return(x)
}

add1(10)

## [1] 11

x

## Error in eval(expr, envir, enclos):  object 'x' not found
```

Useful string operations

The 'stringr' package adds a number of string operations:

```
str_c()          ## glues strings together
str_length()     ## measure the length of a string
str_sub()        ## select parts of the string
str_str() <-     ## assign parts of the string with new values
str_dup()        ## duplicates string
str_trim()       ## removes trailing white space
str_pad()        ## adds whitespace
```

Find and replace

Stringr also does find and replace:

```
str_detect()
str_subset()
str_locate()
str_locate_all()
str_extract()
str_extract_all()
str_match()
str_match_all()
str_split()
```

Class excercise 2

- Read in the minke whale data
- Define a function that calculates the condition factor of a whale using the formula:
$$C = 100 * (W/L^3)$$
- Use the function to calculate the condition factor for all whales that were weighed
- Investigate the distribution of the condition factor
- Assign classification to all whales with body condition less than 0.8 'Malnourished', for 0.8 and above assign "OK" (hint use ifelse)
- Impute the weight for those whales that were not weighed