

Outline

day 3 – April 16th

- Input / Output
- Packages

QUIZZ

- Starting from vector **x** as:

```
x <- c(23, 1, 0, 10, 6)
```

```
x <- x - 2
```

```
x[2] <- x[4]
```

```
x[-2]
```

```
x[ x < 8 ] <- 0
```

What is **x** now?

QUIZZ

- How do you change the name of the 2nd column of data frame **df** to “kcal”?

dessert	cal
brownie	466
cheesecake	321
cookies	360
apple_pie	237

1. `colnames(df[,2]) <- “kcal”`
2. `colnames(df[2]) <- “kcal”`
3. `colnames(df)[2] <- “kcal”`
4. `names(df[, 2]) <- “kcal”`

QUIZZ

- Which way(s) do you select the desserts that **have less than 350 kcal?** (the data frame is **df**)

dessert	kcal
brownie	466
cheesecake	321
cookies	360
apple_pie	237

1. `df$dessert[df$kcal < 350]`
2. `df[df[,"kcal"] < 350, "dessert"]`
3. `df[df[,"kcal"] < 350,]$dessert`

QUIZZ

- Starting from matrix **x** as:

13	1
21	7

x <- x+3

x[2,] <- x[2,]/2

x[, -1]

x[x<5] <- 0

What is **x** now?

Input / Output

Basic Input/output

- **reading in a file**
- **writing out a file**
- **saving a graph in a file**

Read in a file into a vector

- **scan("file.txt")**

By default, scans "double" elements.

→ can fail if the input contains characters

- **scan("file.txt", what="character")**

→ specify the type of data to scan!

- **scan("/users/bi/sbonnin/Dir/file.txt",)**

→ If file is not in the current directory:

Write a vector into a file

- **write**(ourfile, file="ourfile.txt")
- **write**(ourfile, file="ourfile.txt", ncolumns=1)
→ organizes the data in one column

I/O

Input : read in a table into a data frame

- `a <- read.table("file.txt")`

I/O

Input : read in a table into a data frame
→ adapt to your input data

- `a <- read.table("file.txt",`
`sep=","`, **Field separator**
`header=TRUE`, **First row of file read as header / column names**
`skip=2`, **Number of rows to skip before reading the file**
`comment.char="%")` **Character preceding comment lines (those lines will be skipped)**

Input / Output

- **write.table**(a, "file.txt",
 sep="\t", **# Field separator**
 col.names=TRUE, **# Column names written in output file**
 row.names=TRUE, **# Row names written in output file**
 quote=FALSE) **# Quote around character elements:**
 disable

Exercise 6:

Input / Output

Libraries / packages

Library/packages

- **Packages** are collections of R functions, data, and compiled code in a well-defined format.
- The directory where packages are stored is called the **library**.

Definitions from <http://www.statmethods.net/interface/packages.html>

R-base: standard packages

- **R-base:**

about 25 standard packages supplied with R by default (example: base, stats, graphics).

R-contrib: all other packages

- 2 main repositories:
 - CRAN: Comprehensive R Archive Network
11038 packages available
<https://cran.r-project.org>
 - Bioconductor:
1383 packages available
<https://www.bioconductor.org/>

Find packages

- R packages in CRAN:

<https://cran.r-project.org/web/packages/>

Available CRAN Packages By Date of Publication

Date	Package	Title
2017-05-21	aSPU	Adaptive Sum of Powered Score Test
2017-05-21	d3r	'd3.js' Utilities for R
2017-05-21	dynr	Dynamic Modeling in R
2017-05-21	cdfReader	Reading EDF(+) and BDF(+) Files
2017-05-21	GlobalOptions	Generate Functions to Get or Set Global Options
2017-05-21	GWmodel	Geographically-Weighted Models
2017-05-21	IgorR	Read Binary Files Saved by 'Igor Pro' (Including 'Neuromatic' Data)
2017-05-21	imputeTS	Time Series Missing Value Imputation
2017-05-21	lagged	Classes and Methods for Lagged Objects
2017-05-21	mclust	Gaussian Mixture Modelling for Model-Based Clustering, Classification, and Density Estimation
2017-05-21	NPMOD	Non Parametric Module
2017-05-21	numGen	Number Series Generator
2017-05-21	penRvine	Flexible R-Vines Estimation Using Bivariate Penalized Splines
2017-05-21	PhenotypeSimulator	Flexible Phenotype Simulation from Different Genetic and Noise Models
2017-05-21	plfMA	A GUI to View, Design and Export Various Graphs of Data
2017-05-21	RANN	Fast Nearest Neighbour Search (Wraps ANN Library) Using L2 Metric
2017-05-21	regnet	Network-Based Regularization for Generalized Linear Models
2017-05-21	timereg	Flexible Regression Models for Survival Data
2017-05-20	AIG	Automatic Item Generator

Find packages

- R packages in Bioconductor:
<https://bioconductor.org/packages>

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

Home Install Help Developers About

Search:

Home » BiocViews

All Packages

Bioconductor version 3.5 (Release)

Autocomplete biocViews search:

▼ Software (1381)

- ▶ AssayDomain (525)
- ▶ BiologicalQuestion (507)
- ▶ Infrastructure (297)
- ▶ ResearchField (373)
- ▶ StatisticalMethod (441)
- ▶ Technology (872)
- ▶ WorkflowStep (735)
- ▶ AnnotationData (912)
- ▶ ExperimentData (316)

Packages found under Software:

Show entries

Search table:

Package	Maintainer	Title
a4	Tobias Verbeke, Willem Ligtenberg	Automated Affymetrix Array Analysis Umbrella Package
a4Base	Tobias Verbeke, Willem Ligtenberg	Automated Affymetrix Array Analysis Base Package
a4Classif	Tobias Verbeke, Willem Ligtenberg	Automated Affymetrix Array Analysis Classification Package
a4Core	Tobias Verbeke, Willem Ligtenberg	Automated Affymetrix Array Analysis Core Package
a4Preproc	Tobias Verbeke, Willem Ligtenberg	Automated Affymetrix Array Analysis Preprocessing Package
a4Reporting	Tobias Verbeke, Willem Ligtenberg	Automated Affymetrix Array Analysis Reporting Package
ABAEnrichment	Steffi Grote	Gene expression enrichment in human brain regions
...	Yongming Andrew	Microarray QA and statistical data analysis for Applied Biosystems Genome

Bioconductor

- R packages specialized in bioinformatics analysis
 - Supports most types of genomics and NGS data (e.g. limma, DESeq2, BayesPeak)
 - Specific data classes (e.g. Granges from GenomicRanges)
 - Integrates command line tools (e.g Rsamtools)
 - Annotation tools (e.g. biomaRt)

Types of Bioconductor packages

- **Software:** set of functions
e.g. DESeq2(NGS data analysis)
- **Annotation:** annotation of specific arrays, organisms, events, etc.
e.g. BSgenome.Hsapiens.UCSC.hg38
- **Experiment:** data that can be loaded and used
e.g. ALL (acute lymphoblastic leukemia dataset)

Install a package with R studio

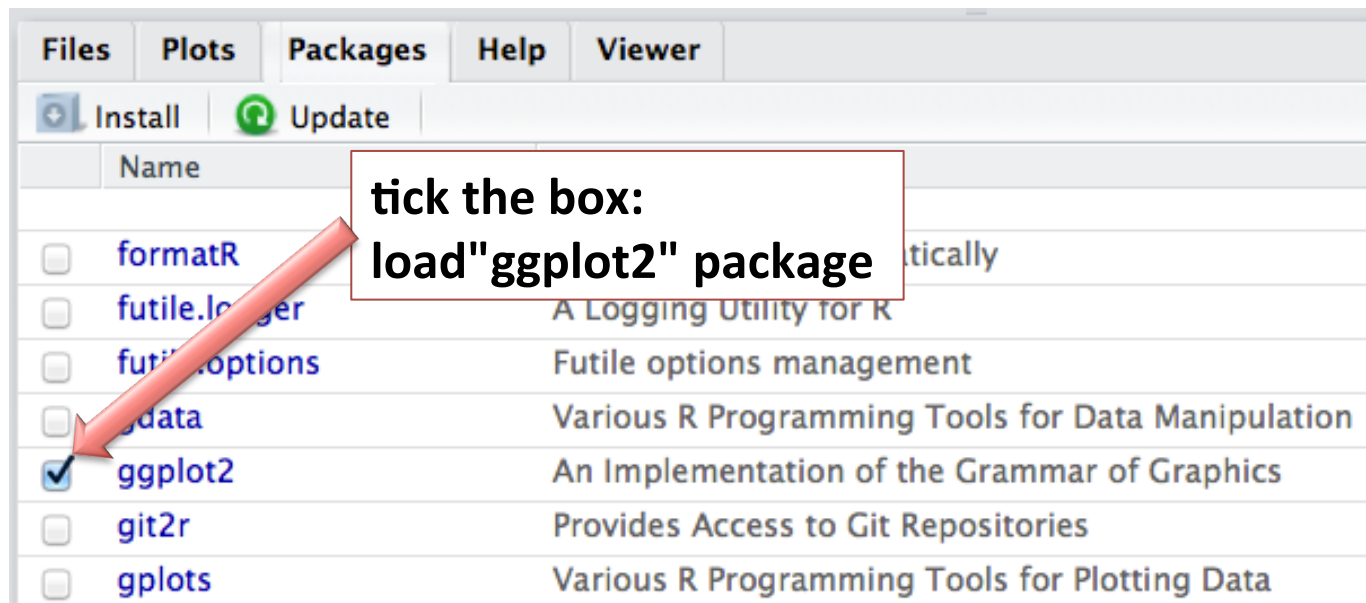
Install "ggplot2" package

The screenshot shows the R Studio interface. The 'Packages' pane is visible, showing a list of installed and available packages. A dialog box titled 'Install Packages' is open, allowing the user to specify the source repository and the packages to install. The 'Install from:' dropdown is set to 'Repository (CRAN)'. The 'Packages (separate multiple with space or comma):' text box is empty. The 'Install to Library:' dropdown is set to '/Library/Frameworks/R.framework/Versions/3.2/Resources/'. The 'Install dependencies' checkbox is checked. The 'Install' and 'Cancel' buttons are at the bottom of the dialog.

Name	Description	Version
base64enc	Tools	0.1-3
BiocInstaller	Install	1.20.1
bitops	Bitwis	1.0-6
boot	Boots	1.3-18
BSDA	Bas	1.01
car	Comp	2.1-2
caTools	Tools	1.17.1
class	Funct	7.3-14
cluster	"Findi	2.0.3
codetools	Code	0.2-14
colorspace	Color	1.2-6
compiler	The R	3.2.4
curl	A Mod	0.9.7

Load a package

- from console
`library(ggplot2)`
- with RStudio



Listing functions from packages

- `ls("package:ggplot2")`

```
[1] "%+%"           "+replace%"
[3] "Coord"         "CoordCartesian"
[5] "CoordFixed"    "CoordFlip"
[7] "CoordMap"      "CoordPolar"
[9] "CoordQuickmap" "CoordTrans"
[11] "Geom"          "GeomAblines"
[13] "GeomAnnotationMap" "GeomArea"
[15] "GeomBar"       "GeomBlank"
```

...

R Studio server at CRG

- You need to specify a writeable directory to install packages into!
- Follow these steps:

```
setwd("~/") # go to home directory
```

```
dir.create("R_packages") # create directory where to store  
                        packages
```

```
.libPaths("~/R_packages/") # add path to library path
```

Exercise 7:

Packages