Extending R through packages: There’s a package for everything
R packages are available on CRAN (Comprehensive R Archive Network)

Contributed Packages

Available Packages

Currently, the CRAN package repository features 7826 available packages.

Table of available packages, sorted by date of publication

Table of available packages, sorted by name

Installation of Packages

Please type `help("INSTALL")` or `help("install.packages")` in R for information on how to install packages from this repository. The manual R Installation and Administration (also contained in the R base sources) explains the process in detail.

CRAN Task Views allow you to browse packages by topic and provide tools to automatically install all packages for special areas of interest. Currently, 33 views are available.

Package Check Results

All packages are tested regularly on machines running Debian GNU/Linux, Fedora, OS X, Solaris and Windows.
R packages are available on CRAN (Comprehensive R Archive Network)

Available CRAN Packages By Name

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A3 Accurate, Adaptable, and Accessible Error Metrics for Predictive Models
abbyyR Access to Abbyy Optical Character Recognition (OCR) API
abc Tools for Approximate Bayesian Computation (ABC)
ABCAnalysis Computed ABC Analysis
abc.data Data Only: Tools for Approximate Bayesian Computation (ABC)
abcdeFBA ABCDE_FBA: A-Biologist-Can-Do-Everything of Flux Balance Analysis with this package
ABCoptim Implementation of Artificial Bee Colony (ABC) Optimization
abcrf Approximate Bayesian Computation via Random Forests
abctools Tools for ABC Analyses
abd The Analysis of Biological Data
abf2 Load Can-Free Armor ABF2 Files
We’ll be working with the package ggplot2

ggplot2: An Implementation of the Grammar of Graphics

An implementation of the grammar of graphics in R. It combines the advantages of both base and lattice graphics: conditioning and shared axes are handled automatically, and you can still build up a plot step by step from multiple data sources. It also implements a sophisticated multidimensional conditioning system and a consistent interface to map data to aesthetic attributes. See http://ggplot2.org for more information, documentation and examples.

Version: 2.0.0
Depends: R (≥ 2.14)
Imports: digest, grid, gtable (≥ 0.1.1), MASS, plyr (≥ 1.7.1), reshape2, scales (≥ 0.3.0), stats
Suggests: ggplot2movies, hexbin, Hmisc, mapproj, maps, maptools, mgcv, multcomp, nlme, testthat, quantreg, knitr
Enhances: sp
Published: 2015-12-18
Author: Hadley Wickham [aut, cre], Winston Chang [aut], RStudio [cph]
Maintainer: Hadley Wickham <hadley at rstudio.com>
BugReports: https://github.com/hadley/ggplot2/issues
License: GPL-2
You can install this package using `install.packages()` in RStudio.
Traditional plotting: You are a painter
   – Manually place individual graphical elements

ggplot2: You employ a painter
   – Describe conceptually how data should be visualized
Most confusing key concept: aesthetic mapping

Maps data values to visual elements of the plot
A few examples of aesthetics

- **position**
- **shape**
- **size**
- **angle**
- **color**
Let’s go over a simple example: mean height and weight of boys/girls ages 10-20

Data from: http://www.cdc.gov/growthcharts/

<table>
<thead>
<tr>
<th>age (yrs)</th>
<th>height (cm)</th>
<th>weight (kg)</th>
<th>sex</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>138</td>
<td>32</td>
<td>M</td>
</tr>
<tr>
<td>15</td>
<td>170</td>
<td>56</td>
<td>M</td>
</tr>
<tr>
<td>20</td>
<td>177</td>
<td>71</td>
<td>M</td>
</tr>
<tr>
<td>10</td>
<td>138</td>
<td>33</td>
<td>F</td>
</tr>
<tr>
<td>15</td>
<td>162</td>
<td>52</td>
<td>F</td>
</tr>
<tr>
<td>20</td>
<td>163</td>
<td>53</td>
<td>F</td>
</tr>
</tbody>
</table>
Map age to $x$, height to $y$, visualize using points

ggplot(data, aes(x=age, y=height)) + geom_point()
Let’s color the points by sex

ggplot(data, aes(x=age, y=height, 
    color=sex)) + geom_point()
And change point size by weight

ggplot(data, aes(x=age, y=height, color=sex, size=weight)) + geom_point()
And connect the points with lines

```r
ggplot(data, aes(x=age, y=height, color=sex, size=weight)) + geom_point() + geom_line()
```

Oops!
The weight-to-size mapping should only be applied to points

ggplot(data, aes(x=age, y=height, color=sex)) + geom_point(aes(size=weight)) + geom_line()
We can also make side-by-side plots (called facets)

```r
ggplot(data, aes(x=age, y=height, color=sex)) + geom_point(aes(size=weight)) + geom_line() + facet_wrap(~sex)
```
Now let’s facet by age, color by weight, and use bars (columns) to plot height

```r
ggplot(data, aes(x=sex, y=height, fill=weight)) + geom_col() + facet_wrap(~age)
```
Let’s plot the sex also at the top of the bar

```r
ggplot(data, aes(x=sex, y=height, fill=weight)) +
geom_col() +
geom_text(aes(label=sex), vjust=1.3, color='white') +
facet_wrap(~age)
```
All the geom’s with all their options are described on the ggplot2 web page

http://ggplot2.tidyverse.org/reference/