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 15364 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, 41 views are available.

Package Check Results

All packages are tested regularly on machines running Debian GNU/Linux, Fedora, OS X, Solaris and Windows.
We’ll be working with the package `ggplot2`

`ggplot2`: Create Elegant Data Visualisations Using the Grammar of Graphics

A system for 'declaratively' creating graphics, based on "The Grammar of Graphics". You provide the data, tell 'ggplot2' how to map variables to aesthetics, what graphical primitives to use, and it takes care of the details.

Version: 3.2.1
Depends: R (≥ 3.2)
Imports: `digest`, `grDevices`, `grid`, `gtable` (≥ 0.1.1), `lazyeval`, `MASS`, `mgcv`, `reshape2`, `rlang` (≥ 0.3.0), `scales` (≥ 0.5.0), `stats`, `tibble`, `viridisLite`, `withr` (≥ 2.0.0)
Suggests: `covr`, `dplyr`, `ggplot2movies`, `hexbin`, `Hmisc`, `knitr`, `lattice`, `mapproj`, `maps`, `maptools`, `multcomp`, `munsell`, `nlme`, `profvis`, `quantreg`, `rgeos`, `rmarkdown`, `rpart`, `sf` (≥ 0.7-3), `svglite` (≥ 1.2.0.9001), `testthat` (≥ 0.11.0), `vdiffir` (≥ 0.3.0)
Enhances: `sp`
Published: 2019-08-10
Author: Hadley Wickham [aut, cre], Winston Chang [aut], Lionel Henry [aut], Thomas Lin Pedersen [aut], Kohske Takahashi [aut], Claus Wilke [aut], Kara Woo [aut], Hiroaki Yutani [aut], RStudio [cph]
Maintainer: Hadley Wickham <hadley at rstudio.com>
BugReports: [https://github.com/tidyverse/ggplot2/issues](https://github.com/tidyverse/ggplot2/issues)
License: GPL-2 | file LICENSE
URL: [http://ggplot2.tidyverse.org](http://ggplot2.tidyverse.org), [https://github.com/tidyverse/ggplot2](https://github.com/tidyverse/ggplot2)
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**
  - Diagram showing position in a 2D coordinate system.

- **shape**
  - Examples of shapes: circle, square, triangle, star.

- **size**
  - Examples of size: varying sizes of dots.

- **color**
  - Examples of color: green, blue, red, yellow, cyan.

- **angle**
  - Examples of angle: various angles represented graphically.
Let’s go over a simple example: mean height and weight of boys/girls ages 10-20

<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>

Data from: http://www.cdc.gov/growthcharts/
Map age to x, height to y, visualize using points

```r
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

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

```
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

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 geoms with all their options are described on the ggplot2 web page

https://ggplot2.tidyverse.org/reference/