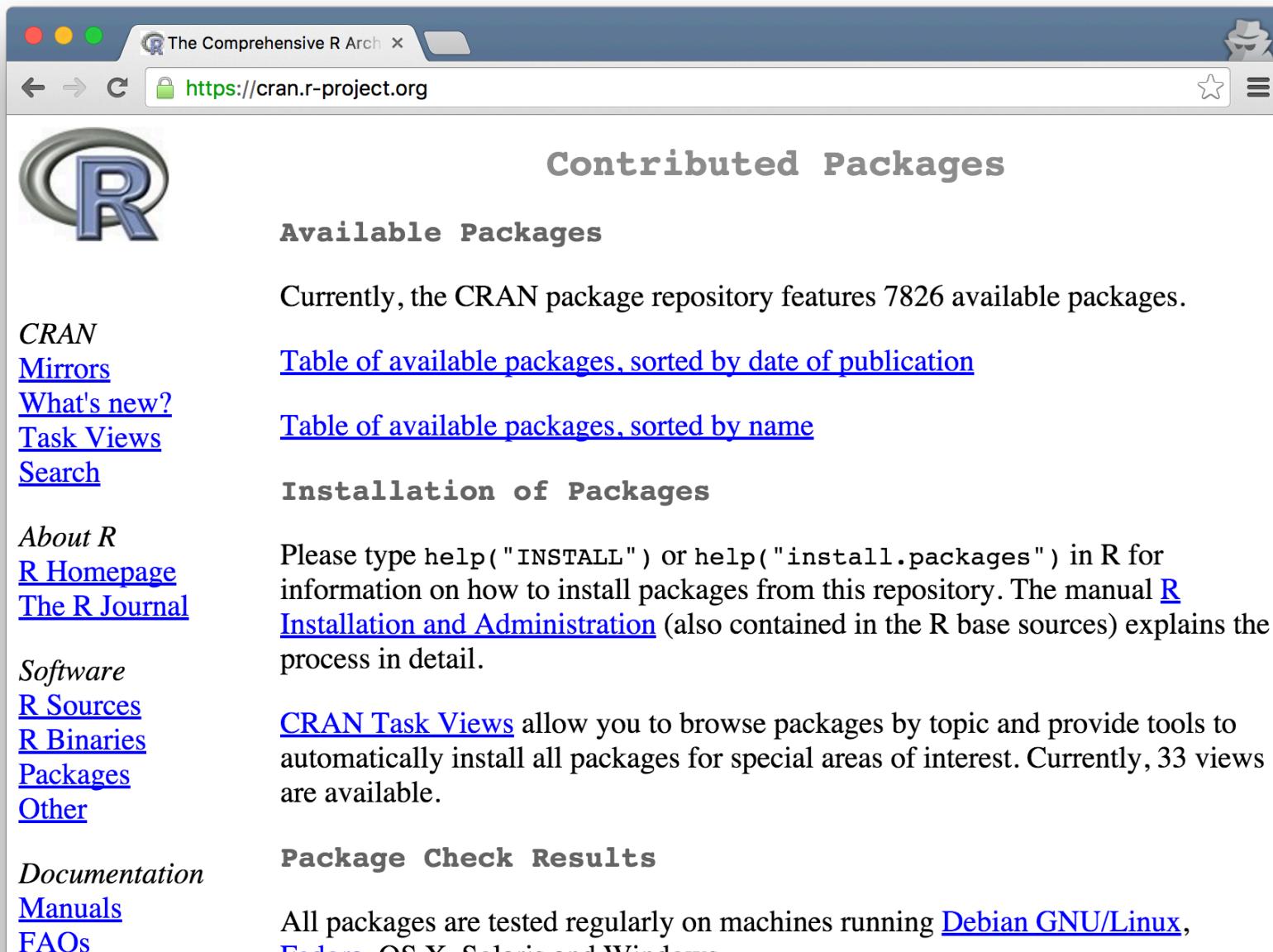


Extending R through packages:
There's a package for everything

R packages are available on CRAN (Comprehensive R Archive Network)



The screenshot shows a web browser window with the title bar "The Comprehensive R Arch". The address bar displays the URL "https://cran.r-project.org". The page content is titled "Contributed Packages" and features a large blue "R" logo. Below the logo, there are sections for "Available Packages" (stating 7826 available packages) and "Installation of Packages" (with instructions to use R functions like "help('INSTALL')"). There are also links to "Table of available packages, sorted by date of publication" and "Table of available packages, sorted by name". On the left side, there is a sidebar with links for "CRAN", "About R", "Software", "Documentation", and "FAQs".

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](#).

CRAN

[Mirrors](#)

[What's new?](#)

[Task Views](#)

[Search](#)

About R

[R Homepage](#)

[The R Journal](#)

Software

[R Sources](#)

[R Binaries](#)

[Packages](#)

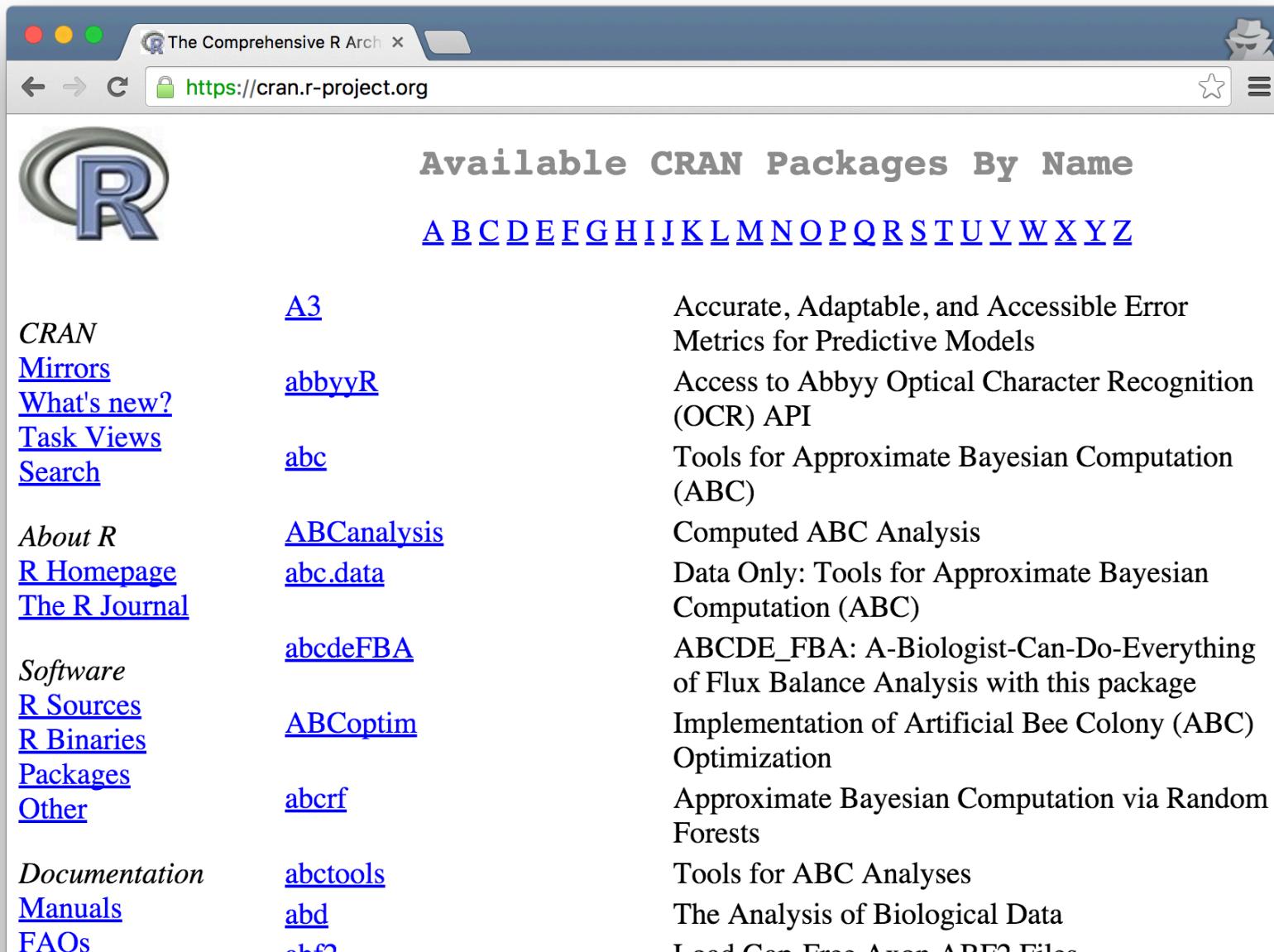
[Other](#)

Documentation

[Manuals](#)

[FAQs](#)

R packages are available on CRAN (Comprehensive R Archive Network)



The screenshot shows a web browser window with the title "The Comprehensive R Arch" and the URL "https://cran.r-project.org". The page displays the "Available CRAN Packages By Name" section, featuring a large blue "R" logo on the left. Below the logo, there are links for "CRAN", "Mirrors", "What's new?", "Task Views", and "Search". On the right, there are links for "A", "B", "C", "D", "E", "F", "G", "H", "I", "J", "K", "L", "M", "N", "O", "P", "Q", "R", "S", "T", "U", "V", "W", "X", and "Y". The "A3" package is highlighted with a brief description: "Accurate, Adaptable, and Accessible Error Metrics for Predictive Models". Other packages listed include "abbyyR", "abc", "ABCanalysis", "abc.data", "abcdeFBA", "ABCOptim", "abcrf", "abctools", "abd", and "abc2".

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	

CRAN

[Mirrors](#)

[What's new?](#)

[Task Views](#)

[Search](#)

About R

[R Homepage](#)

[The R Journal](#)

Software

[R Sources](#)

[R Binaries](#)

[Packages](#)

[Other](#)

Documentation

[Manuals](#)

[FAQs](#)

[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

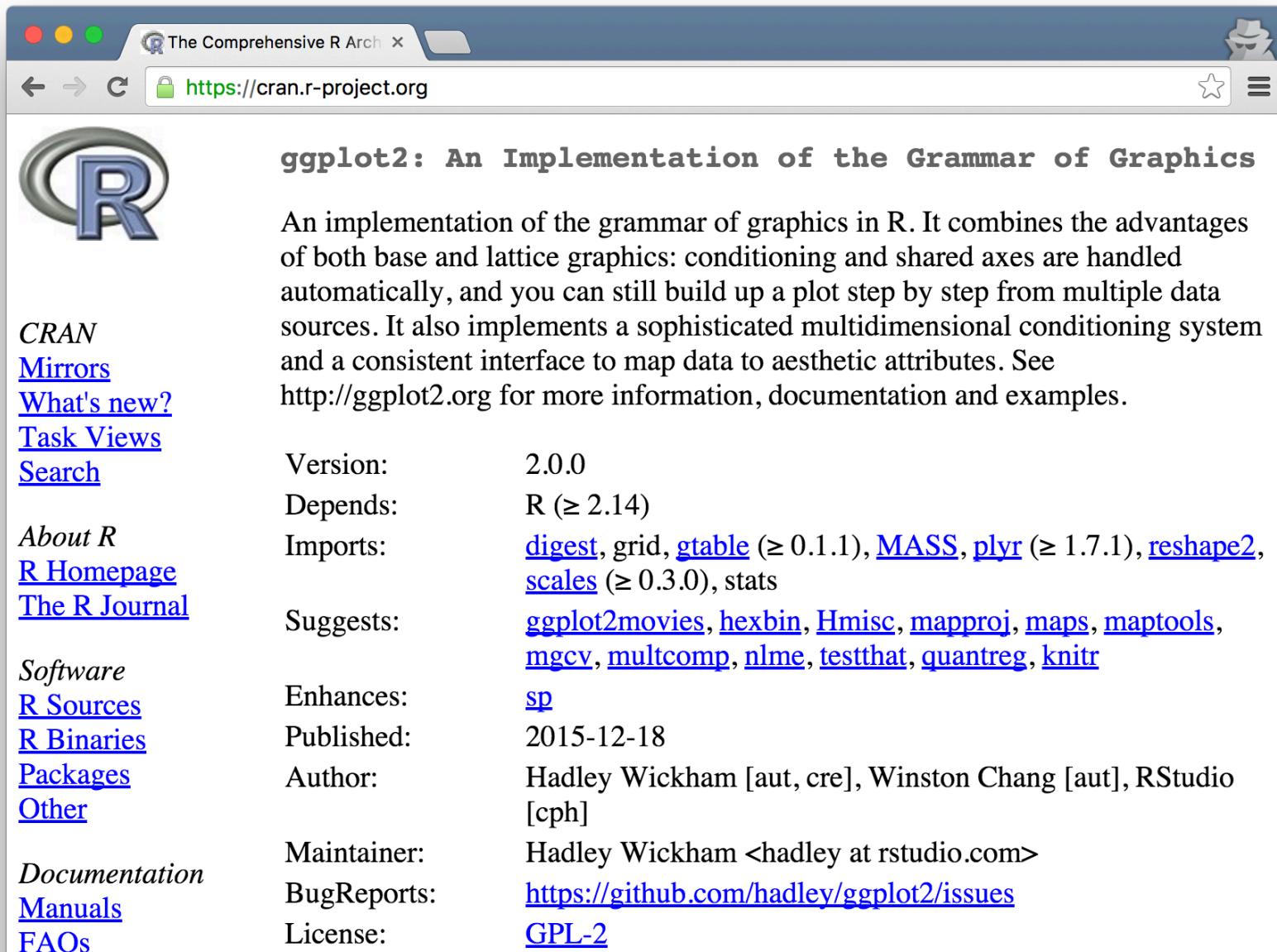
[abd](#)

The Analysis of Biological Data

[abc2](#)

Load Gen-Easy Axon ARF2 Files

We'll be working with the package `ggplot2`



The screenshot shows a web browser window with the title "The Comprehensive R Arch" and the URL "https://cran.r-project.org". The main content is the package page for "ggplot2: An Implementation of the Grammar of Graphics". On the left, there's a sidebar with links for CRAN, Mirrors, What's new?, Task Views, Search, About R, R Homepage, and The R Journal. Below that are links for Software, R Sources, R Binaries, Packages, and Other. At the bottom are links for Documentation, Manuals, and FAQs. The right side of the page contains detailed information about the package, including its version (2.0.0), dependencies (R >= 2.14), imports (digest, grid, gtable, MASS, plyr, reshape2, scales, stats), suggests (ggplot2movies, hexbin, Hmisc, mapproj, maps, maptools, mgcv, multcomp, nlme, testthat, quantreg, knitr), enhances (sp), published date (2015-12-18), author (Hadley Wickham [aut, cre], Winston Chang [aut], RStudio [cph]), maintainer (Hadley Wickham <hadley at rstudio.com>), bug reports (https://github.com/hadley/ggplot2/issues), and license (GPL-2).

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

```
Console ~/ ↗
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 an HTML browser interface to help.
Type 'q()' to quit R.

> install.packages("ggplot2") 
% Total % Received % Xferd Average Speed Time Time Time Current
          Dload Upload Total Spent Left Speed
0       0      0      0      0      0      0 --::-- --::-- --::-- 0 38 1932k
38  751k    0      0  1529k      0  0:00:01 --::-- 0:00:01 1527k 100 1932k
0       0  2918k      0 --::-- --::-- --::-- 2918k

The downloaded binary packages are in
/var/folders/q8/wptgtbdn1pz0cfgrz39gq00m0000gn/T//RtmpvQgw1u/downloaded_packages
> |
```

ggplot2: A grammar of graphics

Traditional plotting: You **are** a painter

- Manually place individual graphical elements

ggplot2: You **employ** a painter

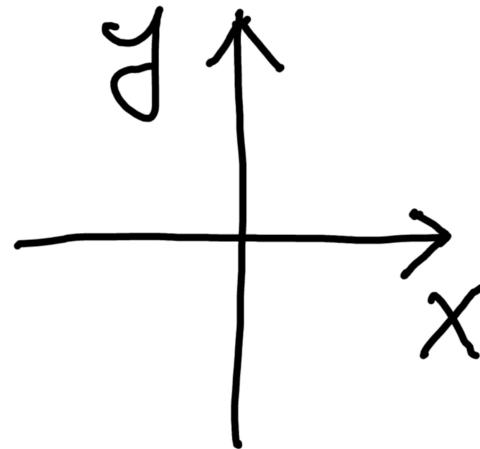
- Describe conceptually how data should be visualized

Most confusing key concept: aesthetic mapping

Maps data values to visual elements of the plot

A few examples of aesthetics

position



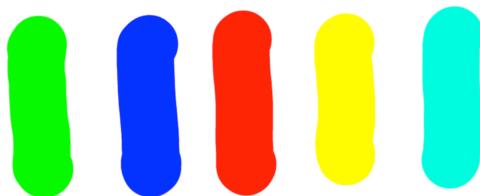
shape



size



color



angle



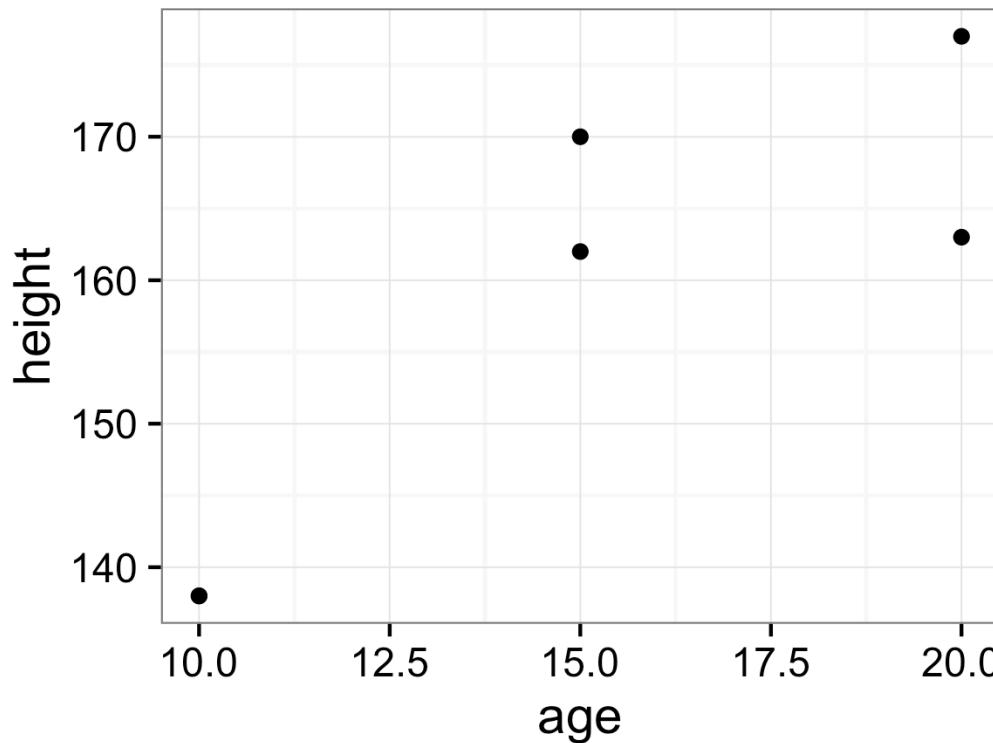
Let's go over a simple example: mean height and weight of boys/girls ages 10-20

age (yrs)	height (cm)	weight (kg)	sex
10	138	32	M
15	170	56	M
20	177	71	M
10	138	33	F
15	162	52	F
20	163	53	F

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

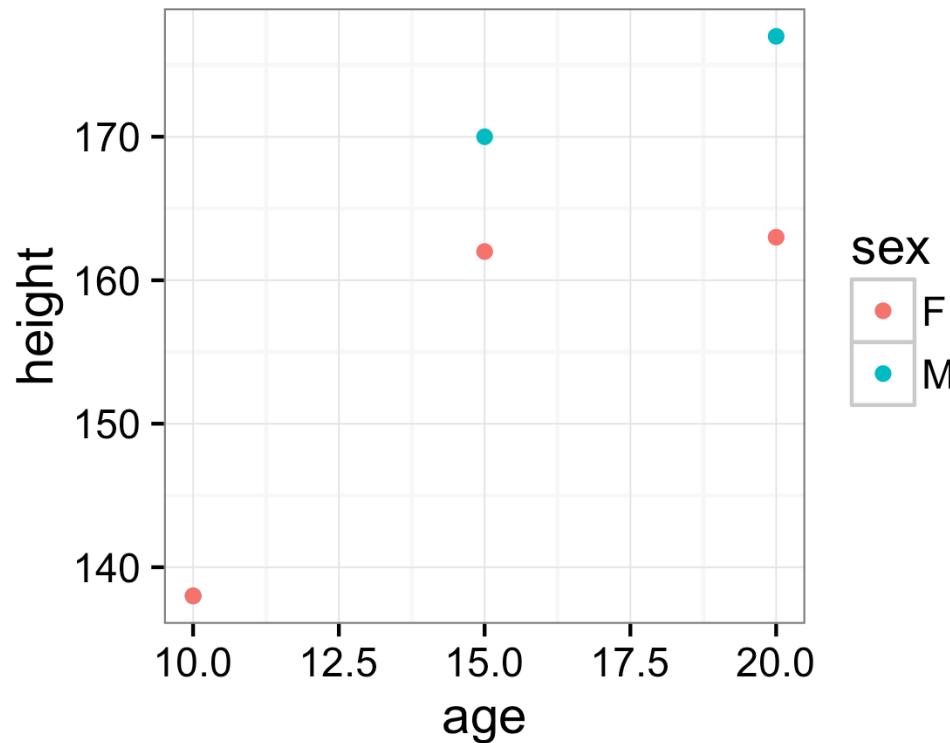
Map age to x, height to y, visualize using points

```
ggplot(data, aes(x=age, y=height)) +  
  geom_point()
```



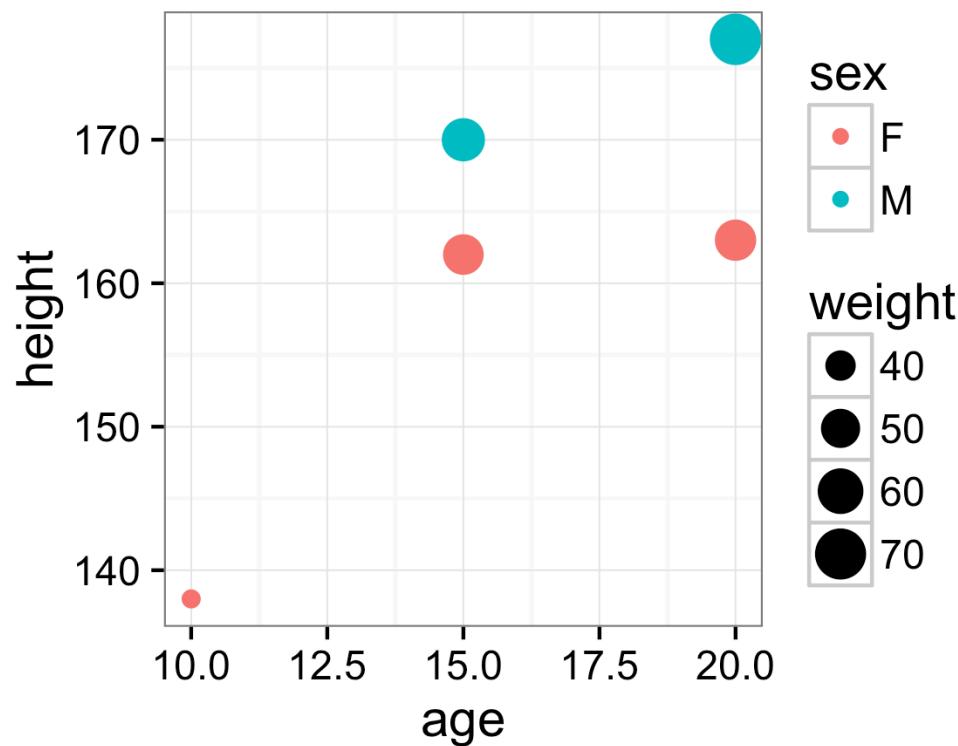
Let's color the points by sex

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



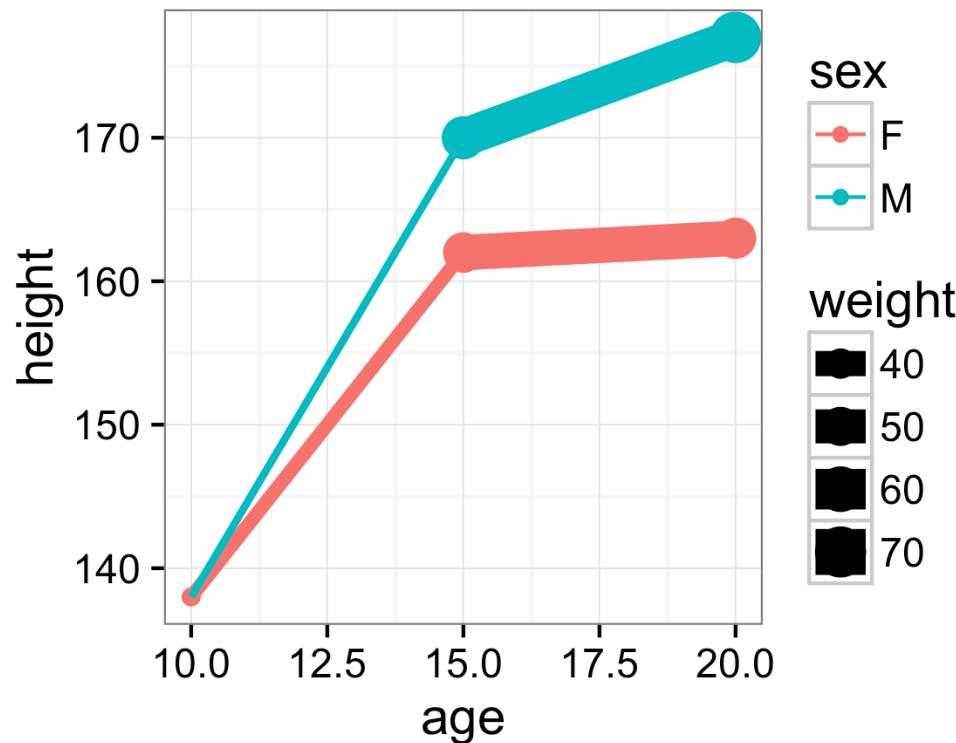
And change point size by weight

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



And connect the points with lines

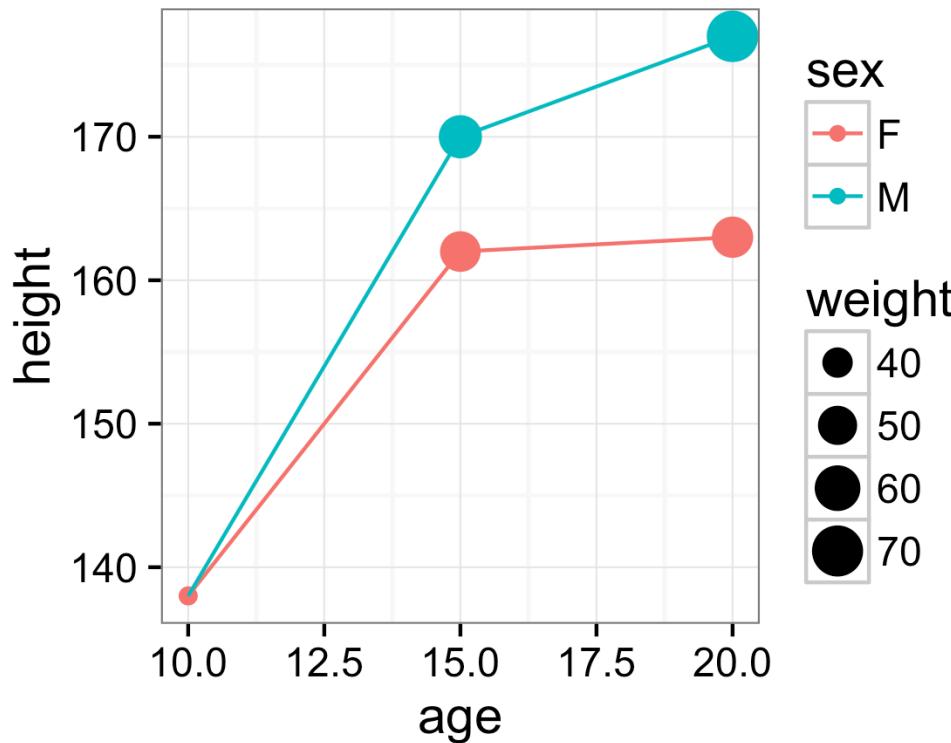
```
ggplot(data, aes(x=age, y=height,  
color=sex, size=weight)) +  
  geom_point() + geom_line()
```



Oops!

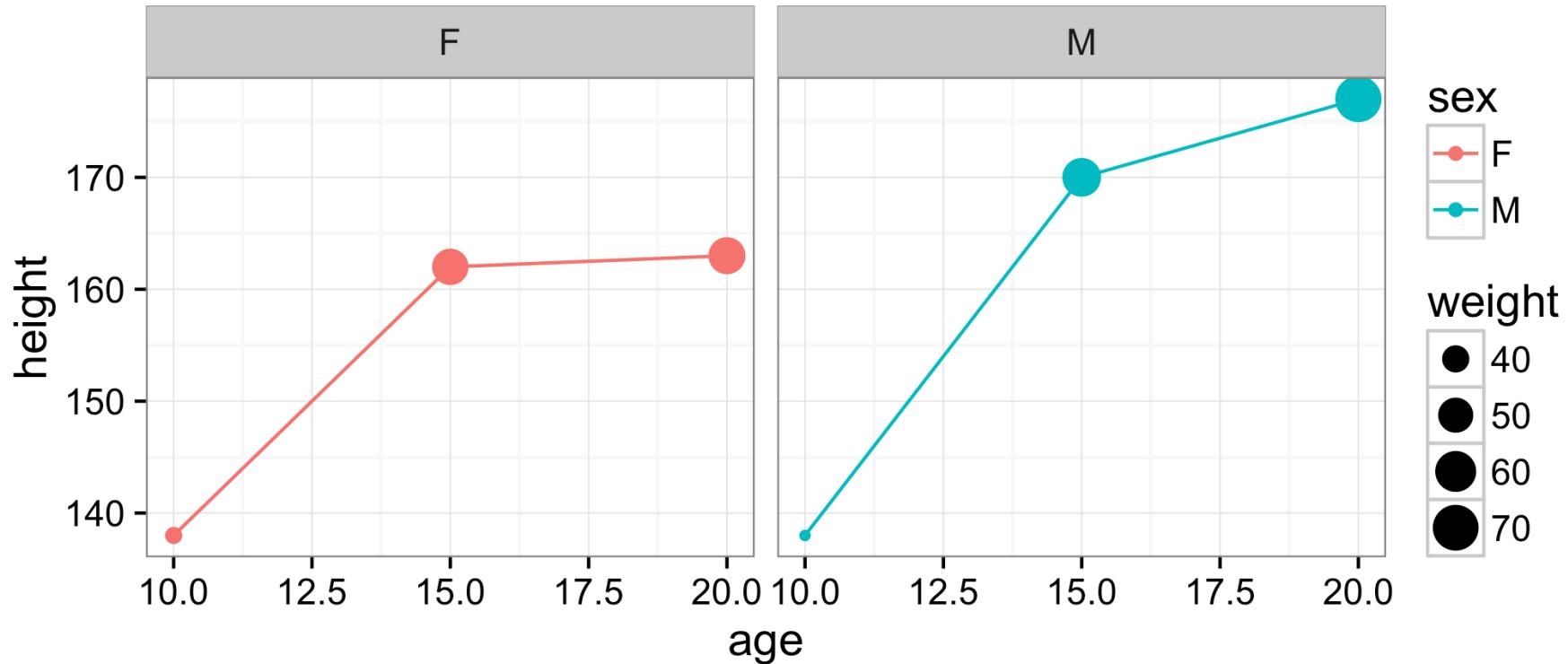
The weight-to-size mapping should only be applied to points

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



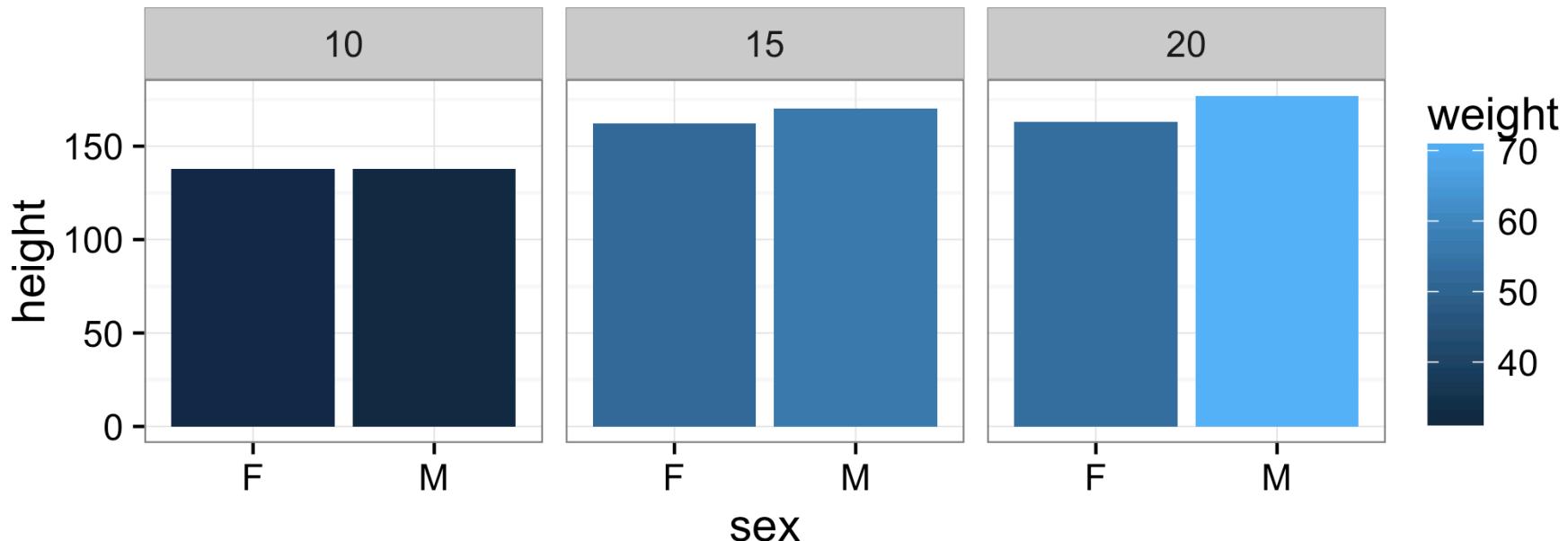
We can also make side-by-side plots (called facets)

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



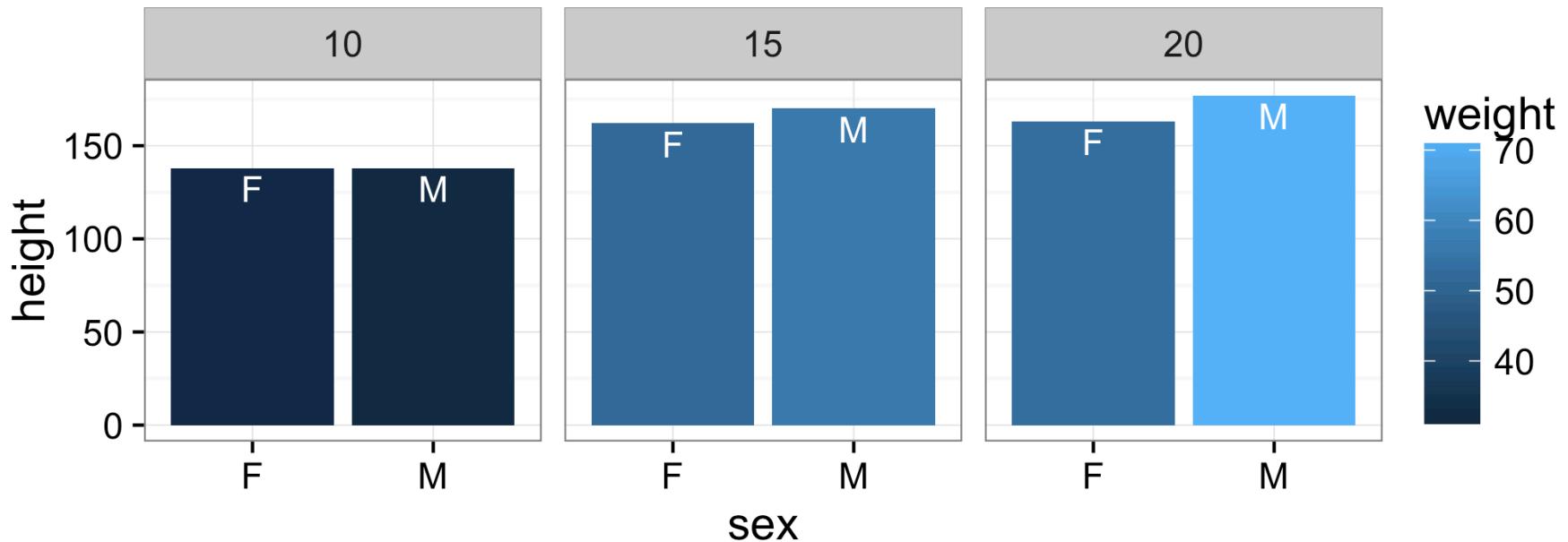
Now let's facet by age, color by weight, and use bars (columns) to plot height

```
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 geom's with all their options are described on the ggplot2 web page

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