Package and collaboration networks in CRAN

Ioannis Kosmidis

🐦 IKosmidis_

✉ ioannis.kosmidis@warwick.ac.uk

🌐 http://www.ikosmidis.com

Reader in Data Science

University of Warwick & The Alan Turing Institute

05 November 2018
R User Group Oxford
Oxford

Outline

1. CRAN today
2. Exploring CRAN
3. cranly networks
4. cranly summaries
5. Dependence trees

CRAN today

```
R> library("cranly")
R> package_db <- clean_CRAN_db()
R> package_net <- build_network(package_db, perspective = "package")
R> author_net <- build_network(package_db, perspective = "author")
R> author_summaries <- summary(author_net)
R> package_summaries <- summary(package_net)
```

As of today 05 Nov 2018 CRAN has

- 18546 authors
- contributing in 12755 packages

Percentage of CRAN packages by some authors

```
R> library("dplyr")
R> authors <- c("R Core", "Brian Ripley", "Achim Zeileis", "Hadley Wickham",
+            "Dirk Eddelbuettel", "Kurt Hornik",
+            "Yihui Xie", "Ioannis Kosmidis")
R> author_summaries %>% group_by(author) %>%
+     summarize(`%CRAN` = 100 * n_packages / nrow(package_summaries)) %>%
+     filter(author %in% authors) %>%
+     arrange(desc(`%CRAN`))
```

author	%CRAN
Hadley Wickham	1.05
R Core	0.64
Dirk Eddelbuettel	0.59
Kurt Hornik	0.56
Achim Zeileis	0.42
Brian Ripley	0.32
Yihui Xie	0.32
Ioannis Kosmidis	0.06

In the course of a couple of decades CRAN became

- a rich and diverse software ecosystem
- a large database of authors, tools and knowledge which are naturally linked to each other
- hard to explore and keep track of

Outline

1 CRAN today

2 Exploring CRAN

3 cranly networks

4 cranly summaries

5 Dependence trees

Exploring CRAN I

cranberries by Dirk Eddelbuettel

🌎 `http://dirk.eddelbuettel.com/cranberries/`

🐦 `CRANberriesFeed`

Aggregates information about new, updated and removed packages from CRAN available and organises it in an RSS feed and a clean interface

Exploring CRAN II

cranlogs R package by Gabor Csardi

🌐 `https://cran.r-project.org/package=cranlogs`

🌐 `https://cranlogs.r-pkg.org`

API for package downloads from RStudio CRAN mirror, and badges

```
R> cranlogs::cran_top_downloads("last-week")
```

rank	package	count	from	to
1	rlang	212802	2018-10-29	2018-11-04
2	Rcpp	194573	2018-10-29	2018-11-04
3	ggplot2	187144	2018-10-29	2018-11-04
4	glue	150919	2018-10-29	2018-11-04
5	stringi	148461	2018-10-29	2018-11-04
6	dplyr	146945	2018-10-29	2018-11-04
7	digest	143414	2018-10-29	2018-11-04
8	WGCNA	140912	2018-10-29	2018-11-04
9	stringr	137661	2018-10-29	2018-11-04
10	fansi	133979	2018-10-29	2018-11-04

`https://cranlogs.r-pkg.org/badges/tidyverse` ⇒ downloads 1M/month

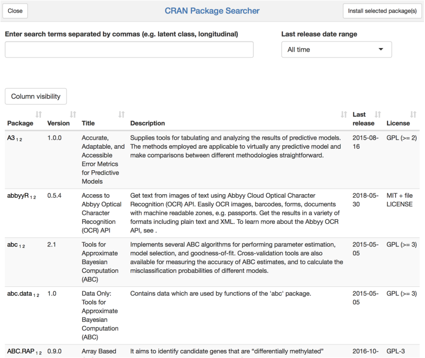
Exploring CRAN III

CRANsearcher R package by Becca Krouse and Agustin Calatroni

🌎 `https://CRAN.R-project.org/package=CRANsearcher`

Shiny interface and RStudio plugin for searching packages by topic

```
R> CRANsearcher::CRANsearcher()
```



Enter search terms separated by commas (e.g. latent class, longitudinal)

Last release date range: All time

Package	Version	Title	Description	Last release	License
A3 $_{1\ 2}$	1.0.0	Accurate, Adaptable, and Accessible Error Metrics for Predictive Models	Supplies tools for tabulating and analyzing the results of predictive models. The methods employed are applicable to virtually any predictive model and make comparisons between different methodologies straightforward.	2015-08-16	GPL (>= 2)
abbyyR $_{1\ 2}$	0.5.4	Access to Abbyy Optical Character Recognition (OCR) API	Get text from images of text using Abbyy Cloud Optical Character Recognition (OCR) API. Easily OCR images, barcodes, forms, documents with machine readable zones, e.g. passports. Get the results in a variety of formats including plain text and XML. To learn more about the Abbyy OCR API, see .	2018-05-30	MIT + file LICENSE
abc $_{1\ 2}$	2.1	Tools for Approximate Bayesian Computation (ABC)	Implements several ABC algorithms for performing parameter estimation, model selection, and goodness-of-fit. Cross-validation tools are also available for measuring the accuracy of ABC estimates, and to calculate the misclassification probabilities of different models.	2015-05-05	GPL (>= 3)
abc.data $_{1\ 2}$	1.0	Data Only: Tools for Approximate Bayesian Computation (ABC)	Contains data which are used by functions of the 'abc' package.	2015-05-05	GPL (>= 3)
ABC.RAP $_{1\ 2}$	0.9.0	Array Based	It aims to identify candidate genes that are “differentially methylated”	2016-10-	GPL-3

Exploring CRAN IV

`available.packages` from **utils**

Returns a matrix of details corresponding to packages currently available at one or more repositories

```
R> library("tibble")
R> available.packages() %>% as.tibble

# A tibble: 13,333 x 17
   Package Version Priority Depends Imports LinkingTo Suggests Enhances
   <chr>   <chr>   <chr>    <chr>   <chr>   <chr>     <chr>    <chr>
 1 A3      1.0.0   <NA>     R (>= ~ <NA>    <NA>      randomF~ <NA>
 2 abbyyR  0.5.4   <NA>     R (>= ~ httr, ~ <NA>      testtha~ <NA>
 3 abc     2.1     <NA>     R (>= ~ <NA>    <NA>      <NA>     <NA>
 4 abc.da~ 1.0     <NA>     R (>= ~ <NA>    <NA>      <NA>     <NA>
 5 ABC.RAP 0.9.0   <NA>     R (>= ~ graphi~ <NA>      knitr, ~ <NA>
 6 ABCana~ 1.2.1   <NA>     R (>= ~ plotrix <NA>      <NA>     <NA>
 7 abcdeF~ 0.4     <NA>     Rglpk,~ <NA>    <NA>      LIM,syb~ <NA>
 8 ABCopt~ 0.15.0  <NA>     <NA>    Rcpp, ~ Rcpp      testtha~ <NA>
 9 ABCp2   1.2     <NA>     MASS    <NA>    <NA>      <NA>     <NA>
10 abcrf   1.7.1   <NA>     R(>= 3~ "readr~ Rcpp, Rc~ <NA>     <NA>
# ... with 13,323 more rows, and 9 more variables: License <chr>,
#   License_is_FOSS <chr>, License_restricts_use <chr>, OS_type <chr>,
#   Archs <chr>, MD5sum <chr>, NeedsCompilation <chr>, File <chr>,
#   Repository <chr>
```

Exploring CRAN V

CRAN_package_db from **tools**

Returns a character data frame with most DESCRIPTION metadata for the current packages in the CRAN package repository

```
R> p_db <- tools::CRAN_package_db()
R> names(p_db)

 [1] "Package"                 "Version"                 "Priority"                "Depends"
 [5] "Imports"                 "LinkingTo"               "Suggests"                "Enhances"
 [9] "License"                 "License_is_FOSS"         "License_restricts_use"   "OS_type"
[13] "Archs"                   "MD5sum"                  "NeedsCompilation"        "Additional_repositories"
[17] "Author"                  "Authors@R"               "Biarch"                  "BugReports"
[21] "BuildKeepEmpty"          "BuildManual"             "BuildResaveData"         "BuildVignettes"
[25] "Built"                   "ByteCompile"             "Classification/ACM"      "Classification/ACM-2012"
[29] "Classification/JEL"      "Classification/MSC"      "Classification/MSC-2010" "Collate"
[33] "Collate.unix"            "Collate.windows"         "Contact"                 "Copyright"
[37] "Date"                    "Description"             "Encoding"                "KeepSource"
[41] "Language"                "LazyData"                "LazyDataCompression"     "LazyLoad"
[45] "MailingList"             "Maintainer"              "Note"                    "Packaged"
[49] "RdMacros"                "SysDataCompression"      "SystemRequirements"      "Title"
[53] "Type"                    "URL"                     "VignetteBuilder"         "ZipData"
[57] "Published"               "Path"                    "X-CRAN-Comment"          "Reverse depends"
[61] "Reverse imports"         "Reverse linking to"      "Reverse suggests"        "Reverse enhances"
[65] "MD5sum"
```

Why bother more and make **cranly**? I

Altruistic reasons

Set of tools for discovery of interconnections in CRAN

Data objects for modelling software networks

Interactive shiny app for package and collaborations discovery
(not there yet but should be easy now)

Why bother more and make **cranly**? II

Less altruistic reasons

Wanted to do what is at

🌎 `ikosmidis.com/software`

and keep track of who/what/why links to my R packages

Wanted a tool that helps me find referees for my editorial work (seriously!)

cranly R package

I am a big fan of "literate programming" principles

`clean_CRAN_db`: "clean up" the CRAN database information

`build_network`: build networks out of it

`build_dependence_tree`: build package dependence trees

`subset`: subset `cranly_network` objects

`summary`: summarize `cranly_network` objects

`plot`: plot `cranly_network` objects or summaries of those

various extractor functions

Outline

1 CRAN today

2 Exploring CRAN

3 cranly networks

4 cranly summaries

5 Dependence trees

The DESCRIPTION file

```
R> (lubridate_desc <- packageDescription("lubridate"))


Package: lubridate
Type: Package
Version: 1.7.4
Title: Make Dealing with Dates a Little Easier
Description: Functions to work with date-times and time-spans: fast and user friendly parsing of date-time data,
             extraction and updating of components of a date-time (years, months, days, hours, minutes, and
             seconds), algebraic manipulation on date-time and time-span objects. The 'lubridate' package has a
             consistent and memorable syntax that makes working with dates easy and fun.  Parts of the 'CCTZ' source
             code, released under the Apache 2.0 License, are included in this package. See
             <https://github.com/google/cctz> for more details.
Authors@R: c( person("Vitalie", "Spinu", email = "spinuvit@gmail.com", role = c("aut","cre")), person("Garrett",
             "Grolemund", role = "aut"), person("Hadley", "Wickham", role = "aut"), person("Ian", "Lyttle",
             role="ctb"), person("Imanuel", "Constigan", role = "ctb"), person("Jason", "Law", role="ctb"),
             person("Doug","Mitarotonda", role="ctb"), person("Joseph", "Larmarange", role="ctb"),
             person("Jonathan", "Boiser", role="ctb"), person("Chel Hee", "Lee", role = "ctb") )
Maintainer: Vitalie Spinu <spinuvit@gmail.com>
License: GPL (>= 2)
Depends: methods, R (>= 3.0.0)
Imports: stringr, Rcpp (>= 0.12.13),
LinkingTo: Rcpp,
Suggests: testthat, knitr, covr
Enhances: chron, fts, timeSeries, timeDate, tis, tseries, xts, zoo
SystemRequirements: A system with zoneinfo data (e.g. /usr/share/zoneinfo) as well as a recent-enough C++11 compiler
             (such as g++-4.8 or later). On Windows the zoneinfo included with R is used.
VignetteBuilder: knitr
LazyData: true
Collate: 'Dates.r' 'POSIXt.r' 'RcppExports.R' 'util.r' 'parse.r' 'timespans.r' 'intervals.r' 'difftimes.r'
             'durations.r' .....
RoxygenNote: 6.0.1
URL: http://lubridate.tidyverse.org, https://github.com/tidyverse/lubridate
BugReports: https://github.com/tidyverse/lubridate/issues
NeedsCompilation: yes
Packaged: 2018-04-10 15:18:02 UTC; vspinu
Author: Vitalie Spinu [aut, cre], Garrett Grolemund [aut], Hadley Wickham [aut], Ian Lyttle [ctb], Imanuel Constigan
             [ctb], Jason Law [ctb], Doug Mitarotonda [ctb], Joseph Larmarange [ctb], Jonathan Boiser [ctb], Chel
             Hee Lee [ctb]
Repository: CRAN
Date/Publication: 2018-04-11 10:08:43 UTC
```

Package directives

```
R> lubridate_desc$Suggests

[1] "testthat, knitr, covr"

R> lubridate_desc$Imports

[1] "stringr, Rcpp (>= 0.12.13),"

R> lubridate_desc$Depends

[1] "methods, R (>= 3.0.0)"

R> lubridate_desc$Enhances

[1] "chron, fts, timeSeries, timeDate, tis, tseries, xts, zoo"

R> lubridate_desc$LinkingTo

[1] "Rcpp,"
```

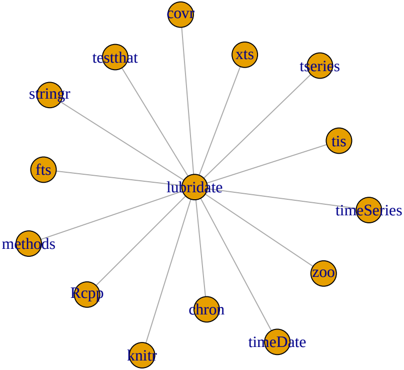
Package directive networks

Package directives "link" packages to each other, in a sense a package citation network by mere definition of what "package dependence" is in R

Package directives define a **package directives network**

Package directive networks

```
R> library("cranly")
R> library("igraph")
R> directives <- with(lubridate_desc, {
+     clean_up_directives(c(Suggests, Imports, Depends, Enhances, LinkingTo))
+ })
R> pdir_net <- data.frame(from = unique(unlist(directives)), to = "lubridate")
R> plot(graph_from_edgelist(as.matrix(pdir_net), directed = FALSE))
```



Author collaboration network

```
R> lubridate_desc$Author

[1] "Vitalie Spinu [aut, cre],\n  Garrett Grolemund [aut],\n  Hadley Wickham [aut],\n  Ian
```

"Author" field specifies a group of collaborators

The author fields from all R packages in CRAN define the CRAN **collaboration network**

Regex, Regex, Regex

CRAN_package_db returns a data frame with current information in most fields in the DESCRIPTION file (and a bit more) for all packages

```
R> p_db$Author[grep("guidance", p_db$Author)][1] %>% strwrap(70)

 [1] "Ravi Varadhan [aut, cph, trl], Paul Gilbert [aut, cre], Marcos Raydan"
 [2] "[ctb] (with co-authors, wrote original algorithms in fortran. These"
 [3] "provided some guidance for implementing R code in the BB package.),"
 [4] "JM Martinez [ctb] (with co-authors, wrote original algorithms in"
 [5] "fortran. These provided some guidance for implementing R code in the"
 [6] "BB package.), EG Birgin [ctb] (with co-authors, wrote original"
 [7] "algorithms in fortran.  These provided some guidance for implementing"
 [8] "R code in the BB package.), W LaCruz [ctb] (with co-authors, wrote"
 [9] "original algorithms in fortran.  These provided some guidance for"
[10] "implementing R code in the BB package.)"

R> p_db$Author[grep("Queen", p_db$Author)][1] %>% strwrap(70)

[1] "Alex M Chubaty [aut, cre], Her Majesty the Queen in Right of Canada,"
[2] "as represented by the Minister of Natural Resources Canada [cph]"
```

For package and collaboration networks, need a reliable systematic way of extracting package and author names from the fields

cranly R package provides clean_up_directives and clean_up_author

clean_up_author

```
R> p_db$Author[grep("guidance", p_db$Author)][1]

[1] "Ravi Varadhan [aut, cph, trl],\n  Paul Gilbert [aut, cre],\n  Marcos Raydan [ctb] (wi

R> p_db$Author[grep("guidance", p_db$Author)][1] %>% clean_up_author

[[1]]
[1] "Ravi Varadhan" "Paul Gilbert"  "Marcos Raydan" "JM Martinez"
[5] "EG Birgin"     "W LaCruz"

R> p_db$Author[grep("Queen", p_db$Author)][1]

[1] "Alex M Chubaty [aut, cre],\n  Her Majesty the Queen in Right of Canada, as represente

R> p_db$Author[grep("Queen", p_db$Author)][1] %>% clean_up_author

[[1]]
[1] "Alex M Chubaty"
```

clean_up_directives

```
R> packageDescription("tidyverse")$Imports

[1] "broom (>= 0.4.2), cli (>= 1.0.0), crayon (>= 1.3.4), dplyr (>=\n0.7.4), dbplyr (>= 1.

R> packageDescription("tidyverse")$Imports %>% clean_up_directives

[[1]]
 [1] "broom"      "cli"        "crayon"     "dplyr"      "dbplyr"
 [6] "forcats"    "ggplot2"    "haven"      "hms"        "httr"
[11] "jsonlite"   "lubridate"  "magrittr"   "modelr"     "purrr"
[16] "readr"      "readxl"     "reprex"     "rlang"      "rstudioapi"
[21] "rvest"      "stringr"    "tibble"     "tidyr"      "xml2"
```

CRAN_package_db() to cranly_db objects

```
R> library("cranly")
R> package_db <- clean_CRAN_db()
R> class(package_db)

[1] "cranly_db"  "data.frame"
```

```
R> lubridate_fields <- package_db %>% filter(package == "lubridate")
R> lubridate_fields$imports

[[1]]
[1] "stringr" "Rcpp"

R> lubridate_fields$suggests

[[1]]
[1] "testthat" "knitr"    "covr"

R> lubridate_fields$author

[[1]]
 [1] "Vitalie Spinu"     "Garrett Grolemund" "Hadley Wickham"
 [4] "Ian Lyttle"        "Imanuel Constigan" "Jason Law"
 [7] "Doug Mitarotonda"  "Joseph Larmarange" "Jonathan Boiser"
[10] "Chel Hee Lee"
```

cranly_network objects

Organise the information in cranly_db objects in networks

There are two obvious network **perspectives** for this information:

Perspective	Nodes	Edge formation	Result
author	authors	same author field	collaboration network
package	packages	in directives fields	package directives network

cranly_network objects

```
R> package_net <- build_network(package_db, perspective = "package")
R> str(package_net, 1)

List of 2
 $ edges:'data.frame': 81845 obs. of  3 variables:
 $ nodes:'data.frame': 12755 obs. of  64 variables:
 - attr(*, "class")= chr [1:2] "cranly_network" "list"
 - attr(*, "timestamp")= POSIXct[1:1], format: "2018-11-05 13:16:38"
 - attr(*, "perspective")= chr "package"
```

```
R> author_net <- build_network(package_db, perspective = "author")
R> str(author_net, 1)

List of 2
 $ edges:'data.frame': 95490 obs. of  10 variables:
 $ nodes:'data.frame': 18546 obs. of  2 variables:
 - attr(*, "class")= chr [1:2] "cranly_network" "list"
 - attr(*, "timestamp")= POSIXct[1:1], format: "2018-11-05 13:16:38"
 - attr(*, "perspective")= chr "author"
```

cranly_network objects

```
R> head(package_net$nodes[, 1:4])

   package version priority                                depends
1       A3   1.0.0     <NA>                        xtable, pbapply
2   a4Base    <NA>     <NA>                                     NA
3   a4Core    <NA>     <NA>                                     NA
4   abbyyR   0.5.4     <NA>
5      abc     2.1     <NA> abc.data, nnet, quantreg, MASS, locfit
6 abc.data     1.0     <NA>

R> head(package_net$edges)

      from     to    type
1     httr abbyyR imports
2      XML abbyyR imports
3     curl abbyyR imports
4    readr abbyyR imports
5     plyr abbyyR imports
6 progress abbyyR imports
```

cranly_network objects

```
R> head(author_net$nodes[, 1:2])

              author
1 Scott Fortmann-Roe
2        Gaurav Sood
3   Csillery Katalin
4  Lemaire Louisiane
5   Francois Olivier
6       Blum Michael
                                                               package
1                                                     A3, adehabitatHR
2 abbyyR, aws.alexa, captr, clarifai, guess, tuber, tubern, virustotal
3                                                        abc, abc.data
4                                                        abc, abc.data
5                                                        abc, abc.data
6                                                        abc, abc.data

R> head(author_net$edges[, 1:3])

                 from                to package
3.1  Csillery Katalin Lemaire Louisiane     abc
3.2  Csillery Katalin  Francois Olivier     abc
3.3  Csillery Katalin      Blum Michael     abc
3.4 Lemaire Louisiane  Francois Olivier     abc
3.5 Lemaire Louisiane      Blum Michael     abc
3.6  Francois Olivier      Blum Michael     abc
```

Interrogating cranly_network objects I

Intuitive, “literate programming”, extractors:

- `package_by(x, author = NULL, exact = FALSE)`
- `package_with(x, name = NULL, exact = FALSE)`
- `author_with(x, name = NULL, exact = FALSE)`
- `author_of(x, package = NULL, exact = FALSE)`
- `suggests(x, package = NULL, exact = FALSE)`
- `imports(x, package = NULL, exact = FALSE)`
- `depends(x, package = NULL, exact = FALSE)`
- `linking_to(x, package = NULL, exact = FALSE)`
- `enhances(x, package = NULL, exact = FALSE)`

Interrogating cranly_network objects II

```
R> package_net %>% author_of("trackeR")

[1] "Yang Liu"                    "Brian C Battaile"
[3] "Andrea Rodriguez-Martinez" "Rafael Ayala"
[5] "Yacine Debbabi"              "Lara Selles Vidal"
[7] "Ioannis Kosmidis"            "Hannah Frick"
[9] "Robin Hornak"

R> package_net %>% author_of("trackeR", exact = TRUE)

[1] "Ioannis Kosmidis" "Hannah Frick"     "Robin Hornak"

R> author_net %>% author_of("brglm")

[1] "Ioannis Kosmidis"            "Kjell Konis"
[3] "Euloge Clovis Kenne Pagui" "Nicola Sartori"
[5] "Alessandra Salvan"
```

Interrogating cranly_network objects III

```
R> package_net %>% package_by("Ioannis Kosmidis")

[1] "betareg"      "brglm"        "brglm2"       "cranly"
[5] "enrichwith"   "PlackettLuce" "profileModel" "trackeR"

R> author_net %>% package_by("Ioannis Kosmidis")

[1] "betareg"      "brglm"        "brglm2"       "cranly"
[5] "enrichwith"   "PlackettLuce" "profileModel" "trackeR"
```

Interrogating cranly_network objects IV

```
R> package_net %>% suggests("trackeR", exact = TRUE)

[1] "testthat"  "knitr"     "rmarkdown" "covr"

R> package_net %>% depends("trackeR", exact = TRUE)

[1] "zoo"

R> package_net %>% imports("trackeR", exact = TRUE)

 [1] "ggplot2"   "ggridges"  "xml2"      "RSQLite"   "jsonlite"
 [6] "raster"    "scam"      "foreach"   "fda"       "sp"
[11] "leaflet"   "ggmap"     "gridExtra" "gtable"

R> package_net %>% linking_to("trackeR", exact = TRUE)

NULL
```

Interrogating cranly_network objects V

All packages that have “bayes” in their name

```
R> package_net %>% package_with("bayes") %>% head(45)

 [1] "acebayes"                "bayesAB"
 [3] "bayesammi"               "BayesBD"
 [5] "BayesBinMix"             "bayesbio"
 [7] "bayesboot"               "BayesCombo"
 [9] "BayesComm"               "bayescount"
[11] "BayesCR"                 "BayesCTDesign"
[13] "BayesDA"                 "bayesDccGarch"
[15] "BAYESDEF"                "bayesDem"
[17] "bayesdfa"                "bayesDP"
[19] "BayesESS"                "BayesFactor"
[21] "BayesFM"                 "bayesGARCH"
[23] "bayesGDS"                "BayesGESM"
[25] "BayesGOF"                "BayesianAnimalTracker"
[27] "Bayesianbetareg"         "BayesianGLasso"
[29] "BayesianNetwork"         "BayesianTools"
[31] "bayesImageS"             "BayesLCA"
[33] "bayesLife"               "bayeslm"
[35] "bayesloglin"             "bayeslongitudinal"
[37] "bayesLopod"              "bayesm"
[39] "BayesMallows"            "BayesMAMS"
[41] "BayesMed"                "bayesmeta"
[43] "bayesmix"                "BayesMixSurv"
[45] "BayesNetBP"
```

Interrogating cranly_network objects VI

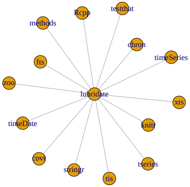
All packages that have “tidy” in their name

```
R> package_net %>% package_with("tidy")

 [1] "htmltidy"       "tidybayes"      "tidyboot"       "tidycensus"
 [5] "tidygenomics"   "tidygraph"      "tidyhydat"      "tidyimpute"
 [9] "tidyLPA"        "tidymodels"     "tidyposterior"  "tidypredict"
[13] "tidyquant"      "tidyr"          "tidyRSS"        "tidyselect"
[17] "tidystats"      "tidystringdist" "tidytext"       "tidytidbits"
[21] "tidytransit"    "tidytree"       "tidyverse"      "tidyxl"
```

Visualising `cranly_network` objects

Static network visualisations are not very useful for exploring **cranly** networks



Need dynamic and interactive network visualisations, with ability to display extra information about nodes and edges on demand

- **node specific**

  package networks: directive summaries, maintainer, downloads

  collaboration networks: number of collaborators, packages
- **edge specific**

  package networks: how are the nodes linked?

  collaboration networks: what do the nodes collaborate on?

visNetwork package: an R interface to `vis.js` JavaScript library

Visualising `cranly_network` objects

```
R> ## Directives network for brglm package
R> plot(package_net, package = "brglm")
R>
R> ## Directives network for packages by me
R> plot(package_net, author = "Ioannis Kosmidis")
R>
R> ## Collaboration network of all authors with Ioannis in their name
R> plot(author_net, author = "Ioannis", exact = FALSE)
R>
R> ## Collaboration network of all authors that have written a package with "glm" in its na
R> plot(author_net, package = "glm", exact = FALSE)
R>
R> ## Directives network of all packages with "glm" in their names
R> plot(package_net, package = "glm", exact = FALSE, title = FALSE, legend = FALSE,
+      width = 1000, height = 800)
```

The magic happens in `subset.cranly_network`

Outline

1. CRAN today
2. Exploring CRAN
3. cranly networks
4. cranly summaries
5. Dependence trees

Talking to **igraph**

cranly has the ability to coerce `cranly_network` objects to `igraph` ones

```
R> as.igraph(package_net)


IGRAPH 3ad7314 DN-- 12755 81845 --
+ attr: name (v/c), version (v/c), author (v/x), date (v/c), url (v/c), license (v/c), maintainer
| (v/c), type (e/c)
+ edges from 3ad7314 (vertex names):
 [1] httr       ->abbyyR       XML        ->abbyyR       curl       ->abbyyR       readr      ->abbyyR
 [5] plyr       ->abbyyR       progress   ->abbyyR       graphics   ->ABC.RAP      stats      ->ABC.RAP
 [9] utils      ->ABC.RAP      plotrix    ->ABCanalysis  Rcpp       ->ABCoptim     graphics   ->ABCoptim
[13] stats      ->ABCoptim     utils      ->ABCoptim     readr      ->abcrf        MASS       ->abcrf
[17] matrixStats->abcrf        ranger     ->abcrf        parallel   ->abcrf        stringr    ->abcrf
[21] Rcpp       ->abcrf        ggplot2    ->ABHgenotypeR reshape2   ->ABHgenotypeR utils      ->ABHgenotypeR
[25] methods    ->abind        utils      ->abind        stringr    ->abjutils     dplyr      ->abjutils
+ ... omitted several edges


R> as.igraph(author_net)


IGRAPH b0b10ae UN-- 18546 95490 --
+ attr: name (v/c), package (v/x), package (e/c), imports (e/x), suggests (e/x), enhances (e/x),
| depends (e/x), linkingto (e/x), version (e/c), maintainer (e/c)
+ edges from b0b10ae (vertex names):
 [1] Csillery Katalin  --Lemaire Louisiane Csillery Katalin  --Francois Olivier
 [3] Csillery Katalin  --Blum Michael      Lemaire Louisiane --Francois Olivier
 [5] Lemaire Louisiane --Blum Michael      Francois Olivier  --Blum Michael
 [7] Csillery Katalin  --Lemaire Louisiane Csillery Katalin  --Francois Olivier
 [9] Csillery Katalin  --Blum Michael      Lemaire Louisiane --Francois Olivier
[11] Lemaire Louisiane --Blum Michael      Francois Olivier  --Blum Michael
[13] Abdulmonem Alsaleh--Robert Weeks      Abdulmonem Alsaleh--Ian Morison
+ ... omitted several edges
```

This gives access to all advanced network summaries provided by igraph, including betweenness, closeness, page rank, degree, eigen centrality.

Summarising cranly_network objects I

```
R> package_summaries <- summary(package_net)
R> package_summaries %>% as.tibble


# A tibble: 12,755 x 17
   package n_authors n_imports n_imported_by n_suggests n_suggested_by n_depends n_depended_by n_enhances
 * <chr>       <int>     <dbl>         <dbl>      <dbl>          <dbl>     <dbl>         <dbl>      <dbl>
 1 A3              1         0             0          2              0         2             0          0
 2 a4Base          1         0             0          0              1         0             0          0
 3 a4Core          1         0             1          0              0         0             1          0
 4 abbyyR          1         6             0          4              0         0             0          0
 5 abc             4         0             1          0              1         5             2          0
 6 abc.da~         4         0             0          0              1         0             1          0
 7 ABC.RAP         4         3             0          2              0         0             0          0
 8 ABCana~         3         1             0          0              3         0             0          0
 9 abcdeF~         2         0             0          2              0         4             0          0
10 ABCopt~         2         4             0          2              0         0             0          0
# ... with 12,745 more rows, and 8 more variables: n_enhanced_by <dbl>, n_linking_to <dbl>,
#   n_linked_by <dbl>, betweenness <dbl>, closeness <dbl>, page_rank <dbl>, degree <dbl>,
#   eigen_centrality <dbl>
```

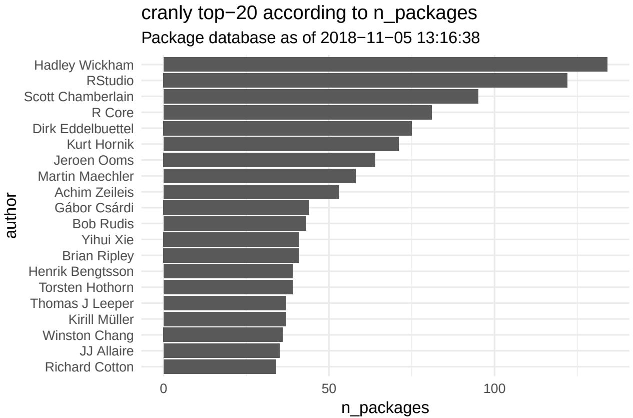
Summarising cranly_network objects II

```
R> author_summaries <- summary(author_net)
R> author_summaries %>% as.tibble


# A tibble: 18,546 x 8
   author             n_packages n_collaborators betweenness     closeness page_rank degree eigen_centrality
 * <chr>                   <int>           <dbl>       <dbl>         <dbl>     <dbl>  <dbl>            <dbl>
 1 Scott Fortmann-Roe          2               1          0  0.00000000551 0.0000170      1     0.0000000567
 2 Gaurav Sood                 8               4       8949. 0.00000000551 0.0000355      4     0.00000537
 3 Csillery Katalin            2               6          0  0.00000000291 0.0000603      6     0
 4 Lemaire Louisiane           2               6          0  0.00000000291 0.0000603      6     0
 5 Francois Olivier            2               6          0  0.00000000291 0.0000603      6     0
 6 Blum Michael                2               6          0  0.00000000291 0.0000603      6     0
 7 Abdulmonem Alsaleh          1               3          0  0.00000000551 0.0000264      3     0.0000149
 8 Robert Weeks                1               3          0  0.00000000551 0.0000264      3     0.0000149
 9 Ian Morison                 1               3          0  0.00000000551 0.0000264      3     0.0000149
10 RStudio                   122             539    1669125. 0.00000000551 0.00152      539     0.00291
# ... with 18,536 more rows
```

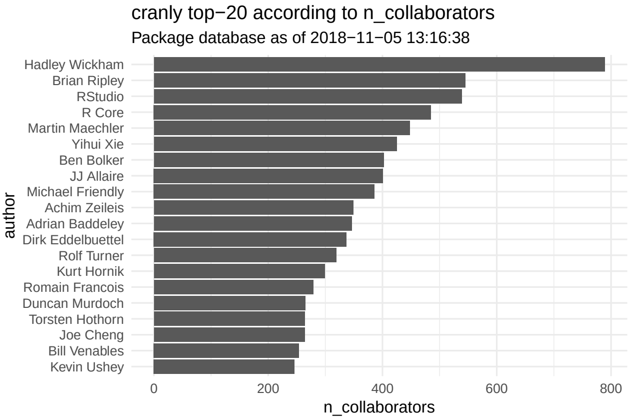
Plotting package summaries = Vanity I

```
R> plot(author_summaries, according_to = "n_packages")
```



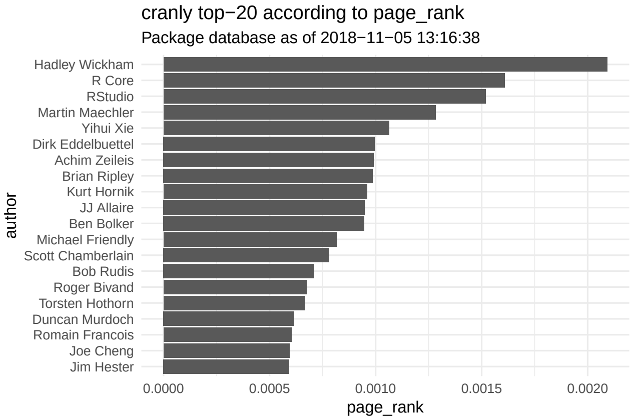
Plotting package summaries = Vanity II

```
R> plot(author_summaries, according_to = "n_collaborators")
```



Plotting package summaries = Vanity III

```
R> plot(author_summaries, according_to = "page_rank", top = 20)
```



Outline

1 CRAN today

2 Exploring CRAN

3 cranly networks

4 cranly summaries

5 Dependence trees

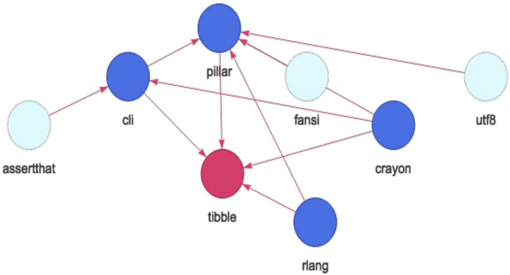
Package dependence tree

A package's dependence tree shows what else needs to be installed with the package in an empty package library with the package

The tree can be constructed neatly using a recursion (see `compute_dependence_tree`), leveraging the advantages of functional programming

tibble dependence tree

```
R> tibble_tree  <- build_dependence_tree(package_net, "tibble")
R> plot(tibble_tree, title = FALSE, legend = FALSE)
```



PNG snapshot

Package dependence index I

The package dependence index is a rough measure of how much "baggage" an R package carries

Average across the generation index of the packages in the tree, with weights that are inversely proportional to the popularity of each package in terms of how many other packages depend on, link to or import it

$$-\frac{\sum_{i \in C_p; i \neq p} \frac{1}{N_i} g_i}{\sum_{i \in C_p; i \neq p} \frac{1}{N_i}}$$

C_p is the dependence tree for the package(s) p

N_i is the total number of packages that depend, link or import package i

g_i is the generation that package i appears in the dependence tree of package(s) p

Package dependence index II

```
R> cranly_tree <- build_dependence_tree(package_net, "cranly")
R> cranly_dep_index <- sapply(cranly_tree$nodes$package, function(package) {
+     tree <- build_dependence_tree(package_net, package = package)
+     s <- summary(tree)
+     s$dependence_index
+ })
R> sort(cranly_dep_index)

  assertthat         cli  colorspace countrycode      crayon
 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
      digest       fansi        glue      gtable   htmltools
 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
    jsonlite    labeling     lattice    lazyeval    magrittr
 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
        MASS      Matrix     munsell        nlme   pkgconfig
 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
        plyr          R6 RColorBrewer       Rcpp       rlang
 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
     stringi     stringr        utf8 viridisLite       withr
 0.000000000 0.000000000 0.000000000 0.000000000 0.000000000
        yaml      pillar      igraph      scales htmlwidgets
 0.000000000 0.004696439 0.012230725 0.018003822 0.193331573
        mgcv  visNetwork      tibble    reshape2     ggplot2
 0.281780864 0.502184168 0.846574672 0.855855051 1.540354944
      cranly
 2.221914406
```

What's next? I

Licence compatibility checks (cudos to Mark Hornick for table)

```
               MIT BSD_3_clause Apache 2.0 LGPL-2.1 LGPL\n(>= 2.1) LGPL-3
MIT              1            1          1        1              1      1
BSD_3_clause     0            1          1        1              1      1
Apache 2.0       0            0          1        0              0      1
LGPL-2.1         0            0          0        1              0      0
LGPL\n(>= 2.1)   0            0          0        1              1      1
LGPL-3           0            0          0        0              0      1
LGPL (>= 3)      0            0          0        0              0      1
GPL-2            0            0          0        0              0      1
GPL (>= 2)       0            0          0        0              0      0
GPL-3            0            0          0        0              0      0
GPL (>=\n3)      0            0          0        0              0      0
Affero GPL-3     0            0          0        0              0      0
               LGPL (>= 3) GPL-2 GPL (>= 2) GPL-3 GPL (>=\n3) Affero GPL-3
MIT                      1     1          1     1          1            1
BSD_3_clause             1     1          1     1          1            1
Apache 2.0               1     0          0     1          1            1
LGPL-2.1                 0     1          1     1          1            1
LGPL\n(>= 2.1)           1     1          1     1          1            1
LGPL-3                   1     0          0     1          1            1
LGPL (>= 3)              1     0          0     1          1            1
GPL-2                    1     0          0     1          1            0
GPL (>= 2)               0     1          1     1          1            1
GPL-3                    0     0          0     1          1            1
GPL (>=\n3)              0     0          0     1          1            1
Affero GPL-3             0     0          0     0          0            1
```

What's next? II

Extractors and topic search in package descriptions

Shiny app

RStudio plugin (anyone?)

Thank you!