Publishing on CRAN

It's easy but requires a lot of work!

Jonas Kristoffer Lindeløv
https://www.linkedin.com/in/jonaskl/
https://x.com/jonaslindeloev

CRAN: the Comprehensive R Archive Network

- CRAN is the archive from which you typically `install.packages(“the_package”)`.
- As of 2024-09-30, CRAN holds 21,416 packages.
- Like PyPi, crates.io, npm, etc. But: CRAN enforces stricter standards.

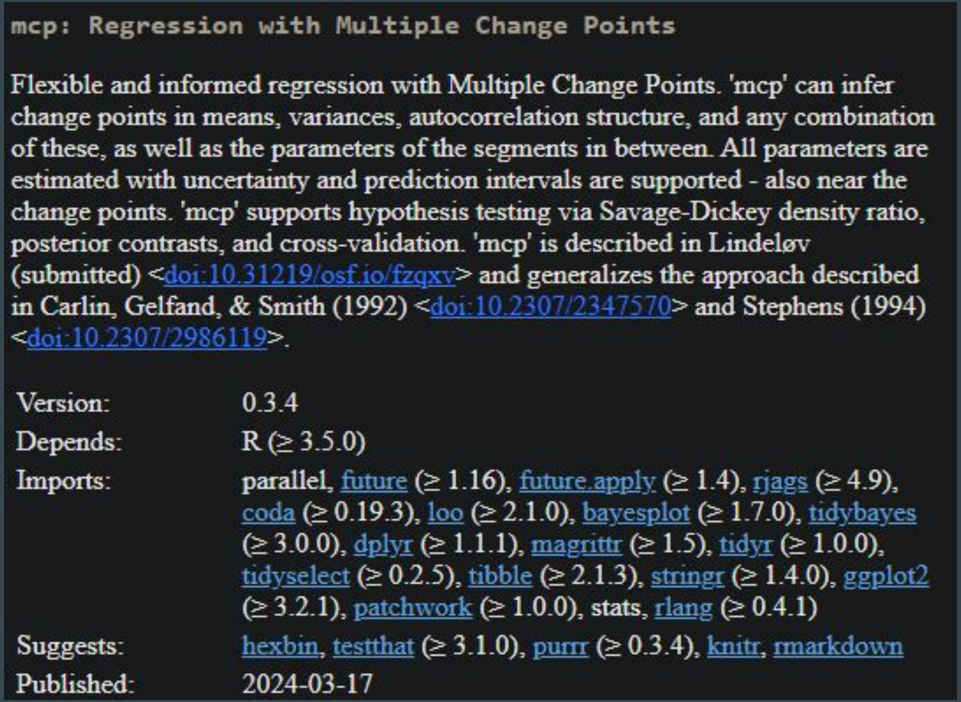
CRAN R
The Comprehensive R
Archive Network

{mcp}

📅 Published 2020-01-09
📑 100+ academic citations
💾 30k+ downloads
⭐ 100+ github stars

Notice:

- Many dependencies
- Requires JAGS (separate binary)
- Generics: summary(), predict(), etc.

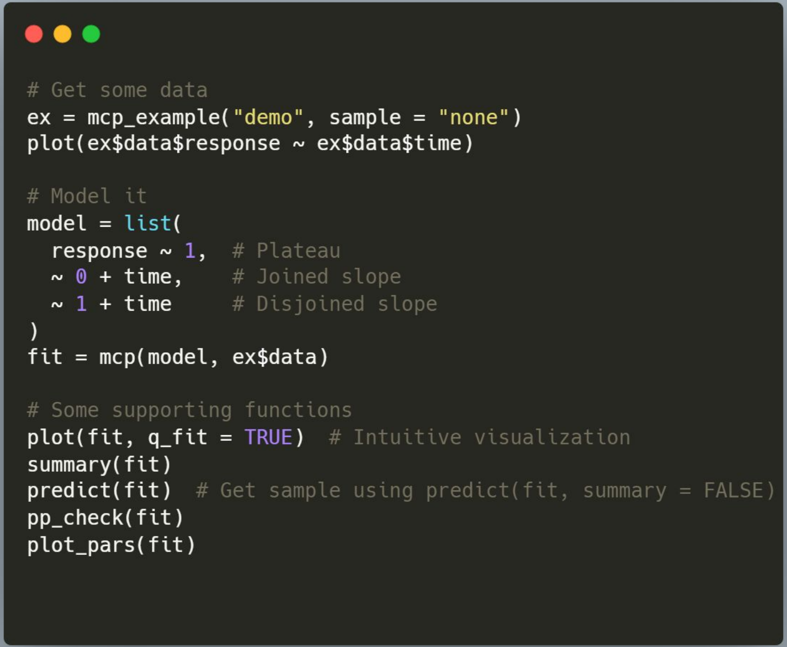


mcp: Regression with Multiple Change Points

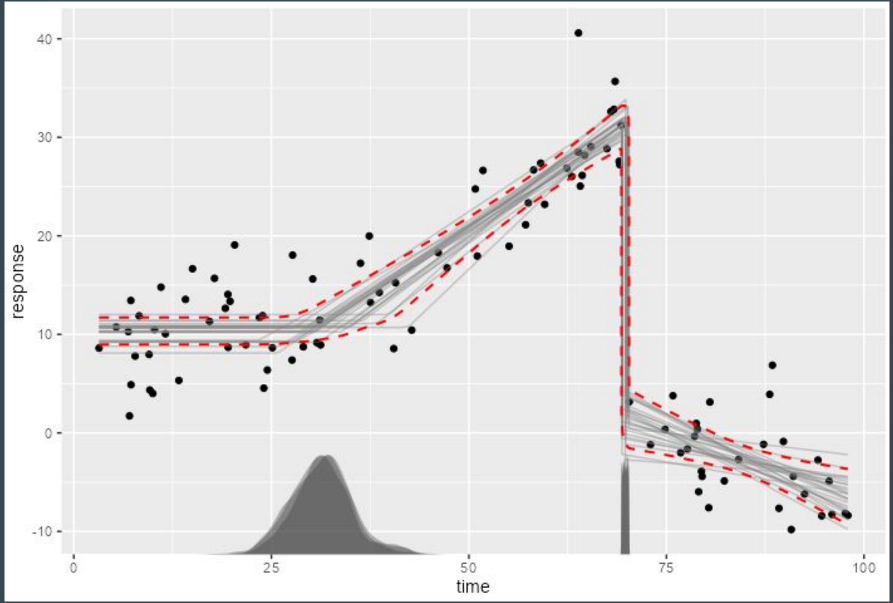
Flexible and informed regression with Multiple Change Points. 'mcp' can infer change points in means, variances, autocorrelation structure, and any combination of these, as well as the parameters of the segments in between. All parameters are estimated with uncertainty and prediction intervals are supported - also near the change points. 'mcp' supports hypothesis testing via Savage-Dickey density ratio, posterior contrasts, and cross-validation. 'mcp' is described in Lindeløv (submitted) <doi:10.31219/osf.io/fzqxv> and generalizes the approach described in Carlin, Gelfand, & Smith (1992) <doi:10.2307/2347570> and Stephens (1994) <doi:10.2307/2986119>.

Version:	0.3.4
Depends:	R (≥ 3.5.0)
Imports:	parallel, future (≥ 1.16), future.apply (≥ 1.4), rjags (≥ 4.9), coda (≥ 0.19.3), loo (≥ 2.1.0), bayesplot (≥ 1.7.0), tidybayes (≥ 3.0.0), dplyr (≥ 1.1.1), magrittr (≥ 1.5), tidyr (≥ 1.0.0), tidyselect (≥ 0.2.5), tibble (≥ 2.1.3), stringr (≥ 1.4.0), ggplot2 (≥ 3.2.1), patchwork (≥ 1.0.0), stats, rlang (≥ 0.4.1)
Suggests:	hexbin, testthat (≥ 3.1.0), purrr (≥ 0.3.4), knitr, rmarkdown
Published:	2024-03-17

mcp demo



```
# Get some data
ex = mcp_example("demo", sample = "none")
plot(ex$data$response ~ ex$data$time)

# Model it
model = list(
  response ~ 1,  # Plateau
  ~ 0 + time,    # Joined slope
  ~ 1 + time     # Disjoined slope
)
fit = mcp(model, ex$data)

# Some supporting functions
plot(fit, q_fit = TRUE)  # Intuitive visualization
summary(fit)
predict(fit)  # Get sample using predict(fit, summary = FALSE)
pp_check(fit)
plot_pars(fit)
```

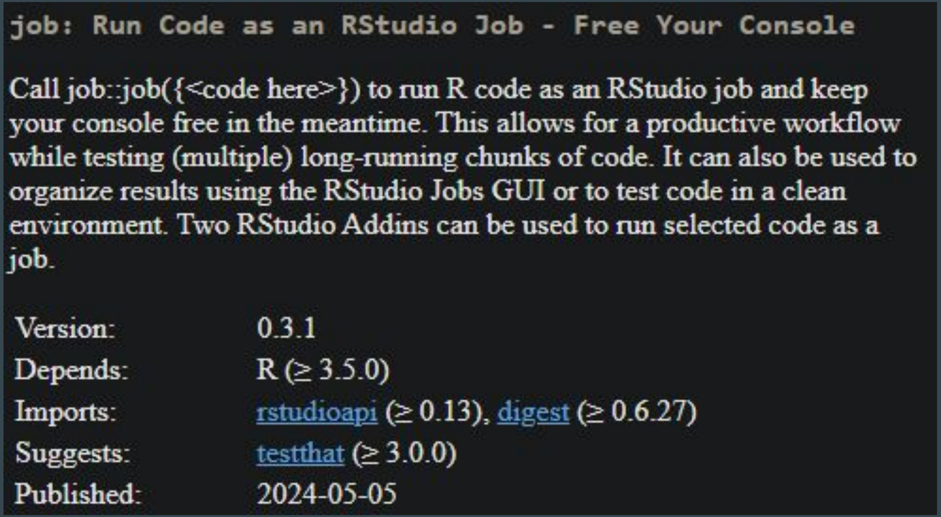


{job}

📅 Published 2021-06-04
💾 20k+ downloads
⭐ 200+ github stars

Notice:

- Few dependencies (but two reverse).
- Only works with RStudio.
- Saves to /tmp/ (external effect).

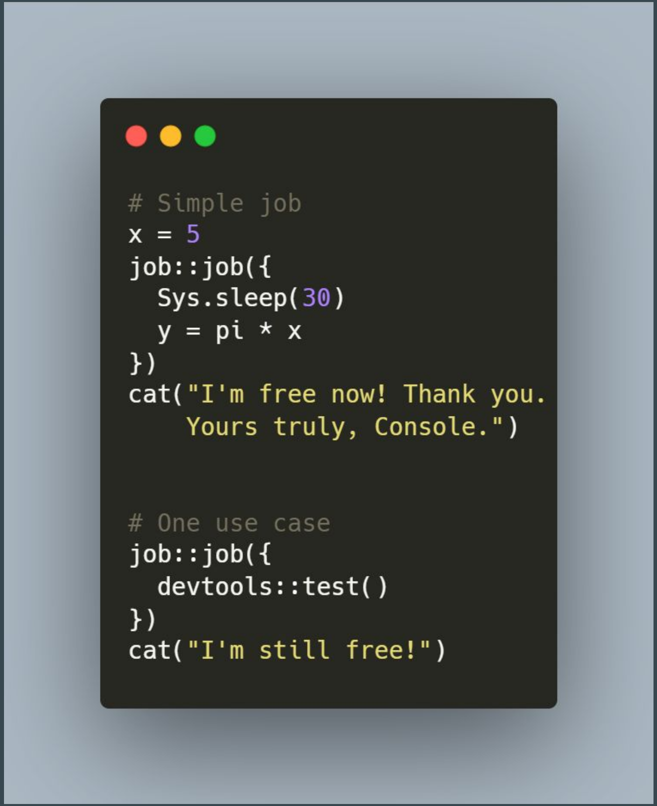


job: Run Code as an RStudio Job - Free Your Console

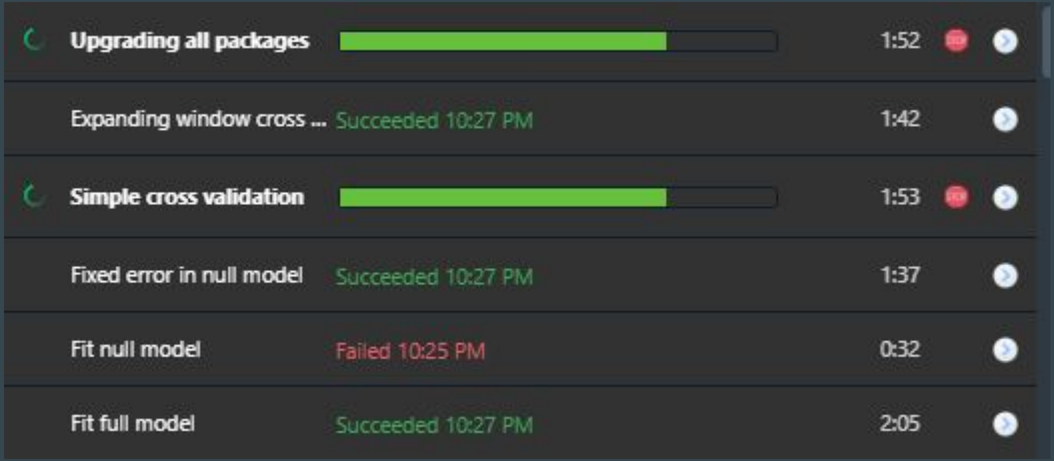
Call job::job({<code here>}) to run R code as an RStudio job and keep your console free in the meantime. This allows for a productive workflow while testing (multiple) long-running chunks of code. It can also be used to organize results using the RStudio Jobs GUI or to test code in a clean environment. Two RStudio Addins can be used to run selected code as a job.

Version:	0.3.1
Depends:	R (≥ 3.5.0)
Imports:	rstudioapi (≥ 0.13), digest (≥ 0.6.27)
Suggests:	testthat (≥ 3.0.0)
Published:	2024-05-05

job demo



```
# Simple job
x = 5
job::job({
  Sys.sleep(30)
  y = pi * x
})
cat("I'm free now! Thank you.
    Yours truly, Console.")


# One use case
job::job({
  devtools::test()
})
cat("I'm still free!")
```



```
Upgrading all packages                                         1:52
Expanding window cross ...   Succeeded 10:27 PM                1:42
Simple cross validation                                        1:53
Fixed error in null model    Succeeded 10:27 PM                1:37
Fit null model               Failed 10:25 PM                   0:32
Fit full model               Succeeded 10:27 PM                2:05
```

Process

Everything is covered in by Hadley Wickham: https://r-pkgs.org/whole-game.html

1. Does the package already exist? See e.g. https://cran.r-project.org/web/views/
2. 📝 Write your package.
3. 🎓 Write tests and pass them.
4. 👩‍🏫 Pass R CMD Check (including test) across systems, R versions, and servers.
5. 💡 (Write good docs and publish)
6. 💌 First CRAN submission!
7. 🔨 Maintenance: new releases and compatibility.

2. Write your package

Set up skeleton:

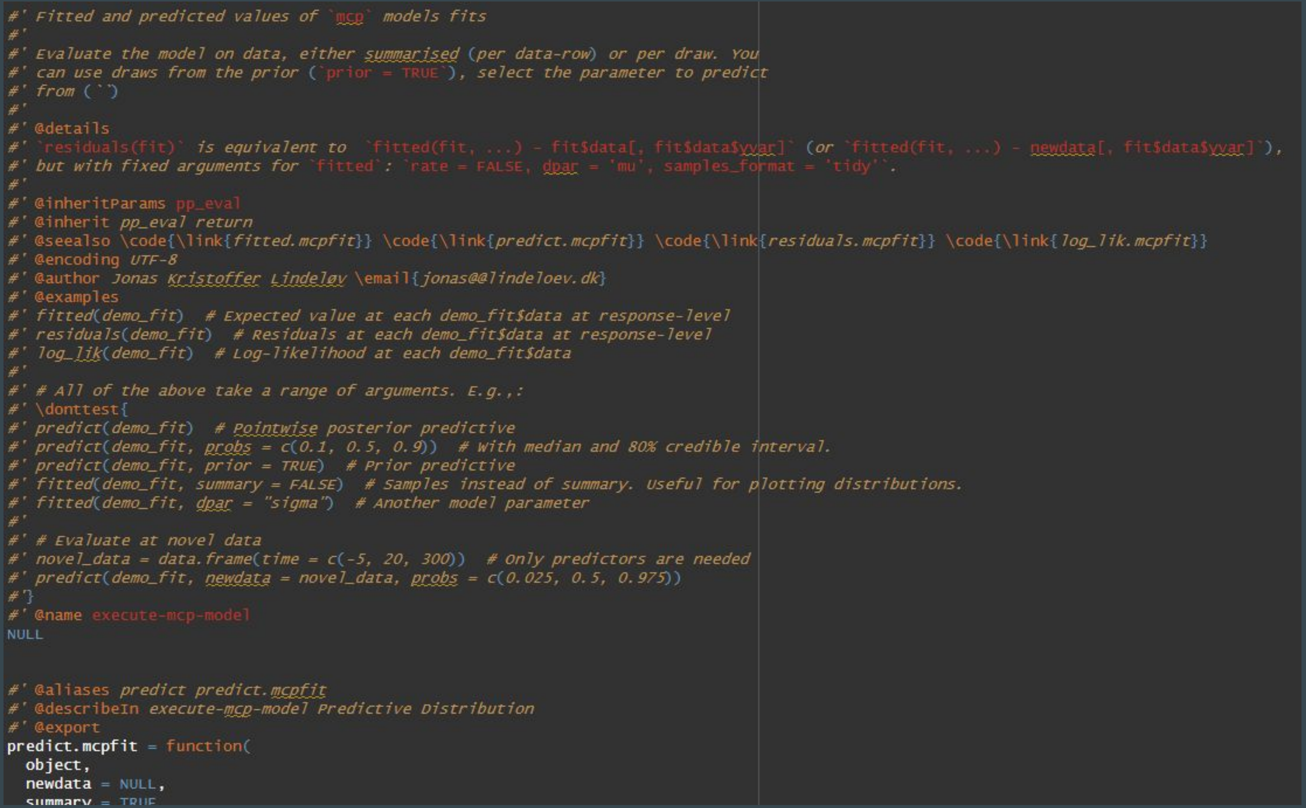
- `usethis::create_package()`
- `usethis::use_git()`
- `usethis::use_testthat()`

Then start coding 🥳

(Tip: CMD+SHIFT+L calls `devtools::load_all()`)

2. Write your package

roxygen2 and generics:



```r
#' Fitted and predicted values of `mcp` models fits
#'
#' Evaluate the model on data, either summarised (per data-row) or per draw. You
#' can use draws from the prior (`prior = TRUE`), select the parameter to predict
#' from (``)
#'
#' @details
#' `residuals(fit)` is equivalent to  `fitted(fit, ...) - fit$data[, fit$data$yvar]` (or `fitted(fit, ...) - newdata[, fit$data$yvar]`),
#' but with fixed arguments for `fitted`: `rate = FALSE, dpar = 'mu', samples_format = 'tidy'`.
#'
#' @inheritParams pp_eval
#' @inherit pp_eval return
#' @seealso \code{\link{fitted.mcpfit}} \code{\link{predict.mcpfit}} \code{\link{residuals.mcpfit}} \code{\link{log_lik.mcpfit}}
#' @encoding UTF-8
#' @author Jonas Kristoffer Lindeløv \email{jonas@@lindeloev.dk}
#' @examples
#' fitted(demo_fit)  # Expected value at each demo_fit$data at response-level
#' residuals(demo_fit)  # Residuals at each demo_fit$data at response-level
#' log_lik(demo_fit)  # Log-likelihood at each demo_fit$data
#'
#' # All of the above take a range of arguments. E.g.,:
#' \donttest{
#' predict(demo_fit)  # Pointwise posterior predictive
#' predict(demo_fit, probs = c(0.1, 0.5, 0.9))  # With median and 80% credible interval.
#' predict(demo_fit, prior = TRUE)  # Prior predictive
#' fitted(demo_fit, summary = FALSE)  # Samples instead of summary. Useful for plotting distributions.
#' fitted(demo_fit, dpar = "sigma")  # Another model parameter
#'
#' # Evaluate at novel data
#' novel_data = data.frame(time = c(-5, 20, 300))  # Only predictors are needed
#' predict(demo_fit, newdata = novel_data, probs = c(0.025, 0.5, 0.975))
#'}
#' @name execute-mcp-model
NULL


#' @aliases predict predict.mcpfit
#' @describeIn execute-mcp-model Predictive Distribution
#' @export
predict.mcpfit = function(
  object,
  newdata = NULL,
  summary = TRUE
```

3. Write tests

- CRAN has no formal requirements for the test. E.g., {job}
- My process (see mcp tests):
    a. Make MVP code work.
    b. Write test for MVP.
    c. Then "grid TDD": test all functions for all possible user calls, including bad calls.
    d. Then fix the code until it passes all tests.
- OBS: tests have max 10 mins runtime on CRAN servers.

4. Pass R CMD Check everywhere

🏠 Local: `devtools::check()`

🌍 Across versions and OSes via Github actions: https://github.com/r-lib/actions/blob/v2-branch/examples/check-standard.yaml

🤨 CRAN-level: Check spelling, URLs, other servers, reverse dependencies, etc: https://github.com/ThinkR-open/prepare-for-cran

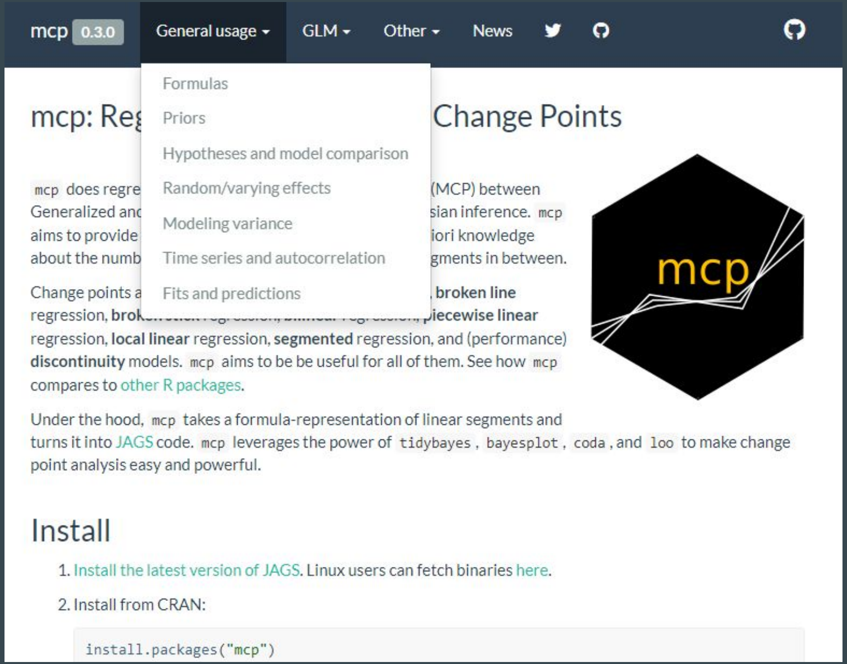
😭 Some bugs are esoteric. E.g., saving options + MacOS + brms + old cpp11: https://github.com/lindeloev/job/issues/27.

5. Write good docs

🌍 Website is good for searchability:

1. usethis::use_pkgdown() and render to a dedicated branch ("docs"?) to keep the commit history clean.
2. Create a github page reflecting that branch.

Vignettes: `usethis::use_vignette()`



mcp: Reg Change Points

mcp does regre (MCP) between Generalized and sian inference. mcp aims to provide iori knowledge about the numb gments in between.

Change points a , broken line regression, brok piecewise linear regression, local linear regression, segmented regression, and (performance) discontinuity models. mcp aims to be be useful for all of them. See how mcp compares to other R packages.

Under the hood, mcp takes a formula-representation of linear segments and turns it into JAGS code. mcp leverages the power of tidybayes, bayesplot, coda, and loo to make change point analysis easy and powerful.

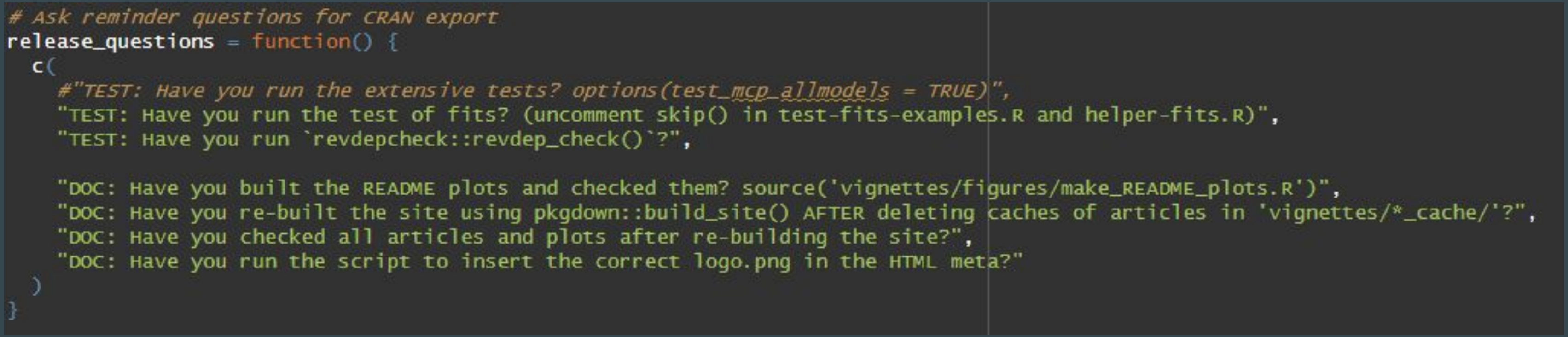
Install

1. Install the latest version of JAGS. Linux users can fetch binaries here.
2. Install from CRAN:

```
install.packages("mcp")
```

6. First CRAN submission!

1. Prepare NEWS, cran-comments, etc.: https://r-pkgs.org/release.html#sec-release-initial
2. 😬 🤞🍀 `devtools::submit_cran()` 🍀🤞😬
3. Follow the progress at https://cran.r-project.org/incoming/ or the Dashboard.
4. Reviewers are pedantic and authoritarian. Accept it and follow orders.



```r
# Ask reminder questions for CRAN export
release_questions = function() {
  c(
    #"TEST: Have you run the extensive tests? options(test_mcp_allmodels = TRUE)",
    "TEST: Have you run the test of fits? (uncomment skip() in test-fits-examples.R and helper-fits.R)",
    "TEST: Have you run `revdepcheck::revdep_check()`?",

    "DOC: Have you built the README plots and checked them? source('vignettes/figures/make_README_plots.R')",
    "DOC: Have you re-built the site using pkgdown::build_site() AFTER deleting caches of articles in 'vignettes/*_cache/'?",
    "DOC: Have you checked all articles and plots after re-building the site?",
    "DOC: Have you run the script to insert the correct logo.png in the HTML meta?"
  )
}
```

mcp: first cran review

Please always write package names, software names and API names in single quotes in title and description. e.g: --> 'mcp'

If there are references describing the methods in your package, please add these in the description field of your DESCRIPTION file in the form
authors (year) <doi:...>
authors (year) <arXiv:...>
authors (year, ISBN:...)
or if those are not available: <https:...>
with no space after 'doi:', 'arXiv:', 'https:' and angle brackets for auto-linking.

Please always add all authors and copyright holders in the Authors@R field with the appropriate roles.
—- A lot about authorship here —-
Please explain in the submission comments what you did about this issue.

\dontrun{} should only be used if the example really cannot be executed (e.g. because of missing additional software, missing API keys, ...) by the user. That's why wrapping examples in \dontrun{} adds the comment ("# Not run:") as a warning for the user.
Does not seem necessary.
Please unwrap the examples if they are executable in < 5 sec, or create additionally small toy examples to allow automatic testing.
(You could also replace \dontrun{} with \donttest if it takes longer than 5 sec to be executed, but it would be preferable to have automatic

Please add \value to .Rd files regarding exported methods and explain the functions results in the documentation.
(See: Writing R Extensions
<https://cran.r-project.org/doc/manuals/r-release/R-exts.html#Documenting-functions>
)
If a function does not return a value, please document that too, e.g. \value{None}.
e.g. ranef.Rd,...

You write information messages to the console that cannot be easily suppressed.
It is more R like to generate objects that can be used to extract the information a user is interested in, and then print() that object.
Instead of print()/cat() rather use message()/warning() or if(verbose)cat(..) if you really have to write text to the console.
(except for print() and summary() functions)
e.g. run_jags(),...

Please fix and resubmit.

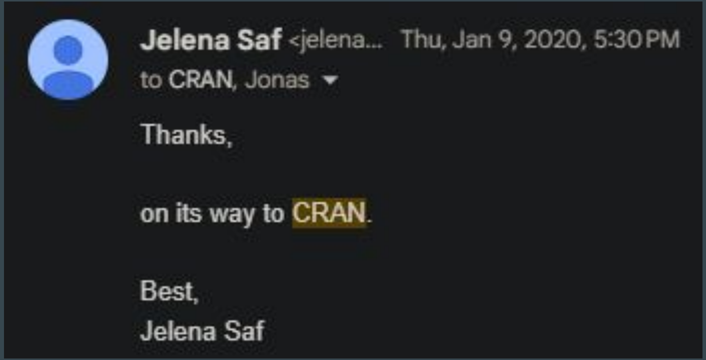
mcp: CRAN review 2+

```
  Possibly mis-spelled words in DESCRIPTION:
    MCP (12:76)
    mcp (12:82, 12:358)

  Found the following (possibly) invalid URLs:
    URL: http://mcmc-jags.sourceforge.net/
      From: README.md
      Status: 503
      Message: Service Unavailable

 Possibly misspelled words in DESCRIPTION:
    Gelfand (13:585)
    Lindeløv (13:486)

—-----------------

ERROR:
The Title field starts with the package name.

—-----------------

You still have
options(mc.cores = 3)
in tests/testthat/test-fit.R. Please reduce.
Please fix and resubmit.
```

🙄

🤬

😭


Jelena Saf <jelena... Thu, Jan 9, 2020, 5:30 PM
to CRAN, Jonas

Thanks,

on its way to CRAN.

Best,
Jelena Saf

🤩

job: CRAN reviews

```
  Found the following (possibly) invalid URLs:
    URL:
https://cran.r-project.org/web/packages/future/vignettes/future-1-overview.html
      From: README.md
      Status: 200
      Message: OK
      CRAN URL not in canonical form
    The canonical URL of the CRAN page for a package is
      https://CRAN.R-project.org/package=pkgname
```

Please change http --> https, add trailing slashes, or follow moved content as appropriate.
Please fix and resubmit.

—------

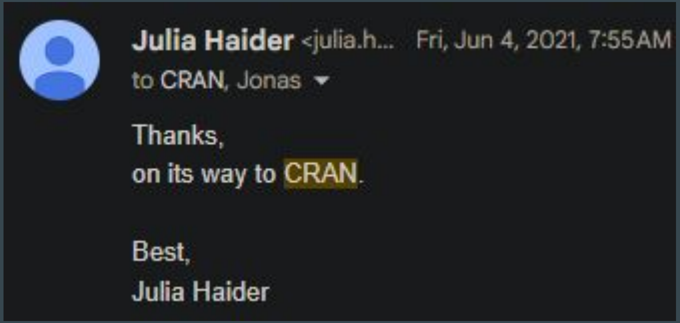
Thanks, we see from the description text:
`job::job({<code here>})`
Please remove the single quotes.

Please do not modify the global environment or the user's home filespace. In your examples or vignettes by deleting objects
rm(list = ls())

👍

You are setting options(warn=-1) in your function. This is not allowed. To avoid unnecessary warning output you could use e.g. suppressWarnings().

Please fix and resubmit.



Julia Haider <julia.h... Fri, Jun 4, 2021, 7:55 AM
to CRAN, Jonas

Thanks,
on its way to CRAN.

Best,
Julia Haider

🤩

7. Maintenance

1. New releases: `devtools::submit_cran()`
2. Breaking dependency updates.
3. New R versions.

7. Maintenance of mcp

—-- Mail, february 29th, 2020 —--

Dear Jonas Kristoffer Lindeløv,

* A new version of tibble is ready to go to CRAN. tibble is
  currently at version 2.99.99.9014 and will become 3.0.0 upon release.

* mcp uses tibble and has problems with the new version.

* We plan to submit tibble to CRAN on Mar 18.

—-- Mail, may 8th, 2020 —--

Dear Jonas Kristoffer Lindeløv,

* A new version of dplyr is ready to go to CRAN. dplyr is
  currently at version 0.8.99.9002 and will become 1.0.0 upon release.

* mcp uses dplyr and has problems with the new version.

* We plan to submit dplyr to CRAN on May 15.

Horror case:
The 2021 {lubridate} incident!

Thanks :-)