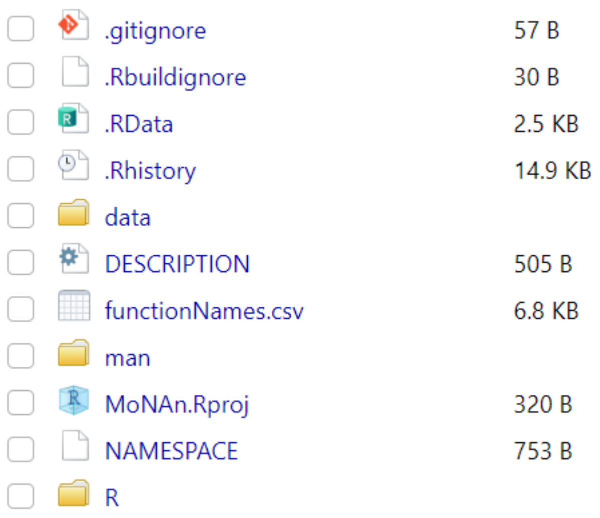
R Package Documentation

Create an R Package Project

- Install packages *devtools, roxygen2, knitr* and *testthat*
- *New Project* > *R Package* > creates basic package structure (see below)
  or when linking to **GitHub**: *New Project* > *Version Control* > *Git* > specify repository (*stackoverflow*-topic for cloning private repositories) > *devtools::create()* or *usthis::create_package()*

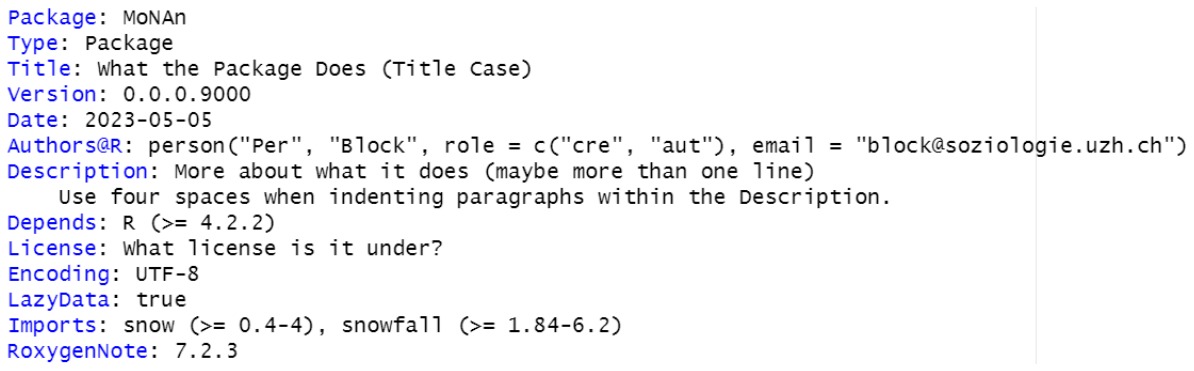


.gitignore	57 B
.Rbuildignore	30 B
.RData	2.5 KB
.Rhistory	14.9 KB
data	
DESCRIPTION	505 B
functionNames.csv	6.8 KB
man	
MoNAn.Rproj	320 B
NAMESPACE	753 B
R	

Description

Contains important metadata such as:

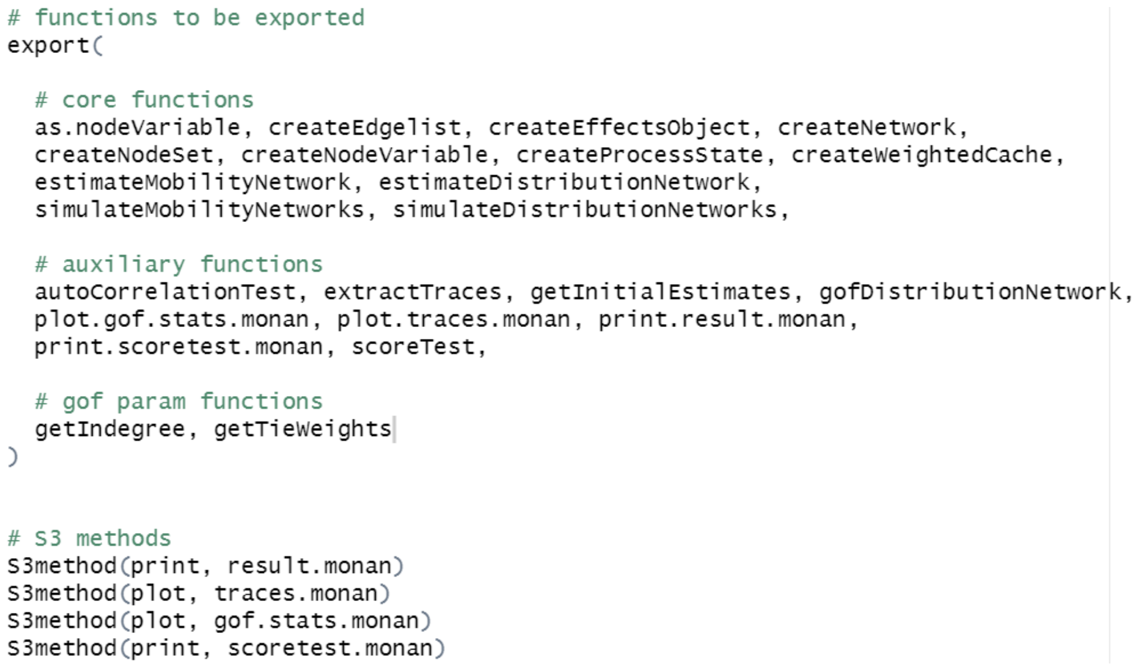
- *Version*: version number (always in format x.x.x, in which <major>.<minor>.<patch> - in „development version“ 0.0.0.9xxx) => more information
- *Date*
- *Authors@R*: name of the authors and contact details
- *Description*: short description of the package
- *Depends*: minimum version of R that is needed to run the package (better to be conservative and specify newer version)
- *License*: how people are allowed to use the package (e.g. *MIT* or *GPL-3*) => more information
- *Lazy Data*: specifies whether data provided with the package should only be loaded when it is actively used (saves memory)
- *Imports*: required other packages => ideally specify a minimum version, e.g. *dplyr (>= 0.3.0.1)*
- *Roxygen: list(markdown = TRUE)*: specifies whether *Markdown* syntax is used in Roxygen comments



```
Package: MoNAn
Type: Package
Title: What the Package Does (Title Case)
Version: 0.0.0.9000
Date: 2023-05-05
Authors@R: person("Per", "Block", role = c("cre", "aut"), email = "block@soziologie.uzh.ch")
Description: More about what it does (maybe more than one line)
    Use four spaces when indenting paragraphs within the Description.
Depends: R (>= 4.2.2)
License: What license is it under?
Encoding: UTF-8
LazyData: true
Imports: snow (>= 0.4-4), snowfall (>= 1.84-6.2)
RoxygenNote: 7.2.3
```

Namespace

- *export()*: list functions which should be exported and publicly available
- S3methods: use S3methods to define that specific printing/plotting functions are executed for certain object classes, e.g. *S3method(plot, traces.monan)* => more information



```
# functions to be exported
export(

  # core functions
  as.nodeVariable, createEdgelist, createEffectsObject, createNetwork,
  createNodeSet, createNodeVariable, createProcessState, createWeightedCache,
  estimateMobilityNetwork, estimateDistributionNetwork,
  simulateMobilityNetworks, simulateDistributionNetworks,

  # auxiliary functions
  autoCorrelationTest, extractTraces, getInitialEstimates, gofDistributionNetwork,
  plot.gof.stats.monan, plot.traces.monan, print.result.monan,
  print.scoretest.monan, scoreTest,

  # gof param functions
  getIndegree, getTieWeights
)


# S3 methods
S3method(print, result.monan)
S3method(plot, traces.monan)
S3method(plot, gof.stats.monan)
S3method(print, scoretest.monan)
```

man

Contains help-files for functions (and data) => are generated automatically using *roxygen2* in the *R*-folder (see below), no files should be edited/created manually in this folder

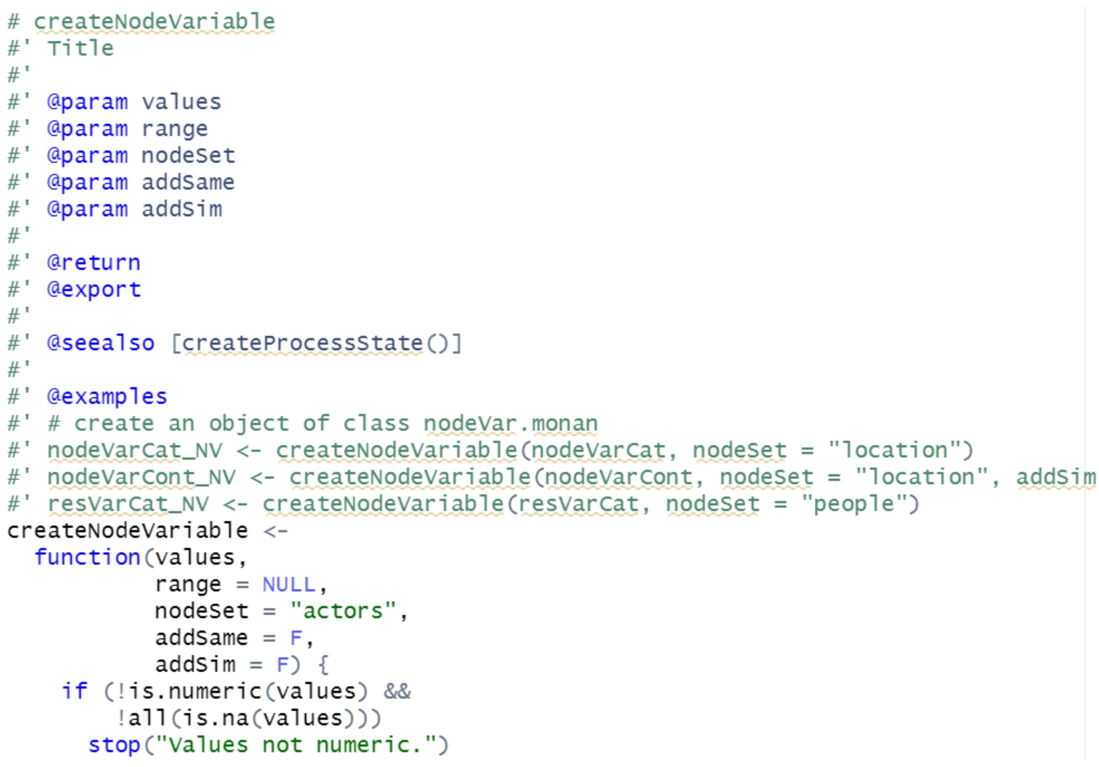
R

Contains .*R*-scripts in which functions are defined (several functions can be defined in the same script => more information)

If a function is not exported (hidden/accessed only in background), no documentation needed => if documentation needed: add a **Roxygen-Skeleton** by pressing *ctrl + alt + shift + R* **or** *ctrl + shift + P* **or** *Code > Insert Roxygen Skeleton* (see below: Roxygen-comments are always added with *#'* and parameters etc. are inserted automatically)

- *@aliases*: Name of an "alias" function which should also refer to a certain help file (e.g. estimateDistributionNetwork for which *estimateDistributionNetwork <- estimateMobilityNetwork*)

- *@rdname*: merges documentation of two different functions into one help-file (for the second function, indicate the name of the first (base) function under *@rdname*) => more information
- *@export*: specifies that a function should be exported
- *@seealso*: links similar/related functions (also from other packages), use *@seealso [function-Name()]*
- *@examples*: example code to show the user how the function can be used (must be specified as Roxygen comment with #' => use this helpful add-in to quickly convert multiple lines into Roxygen-comment)



```r
# createNodeVariable
#' Title
#'
#' @param values
#' @param range
#' @param nodeSet
#' @param addSame
#' @param addSim
#'
#' @return
#' @export
#'
#' @seealso [createProcessState()]
#'
#' @examples
#' # create an object of class nodeVar.monan
#' nodeVarCat_NV <- createNodeVariable(nodeVarCat, nodeSet = "location")
#' nodeVarCont_NV <- createNodeVariable(nodeVarCont, nodeSet = "location", addSim = TRUE)
#' resVarCat_NV <- createNodeVariable(resVarCat, nodeSet = "people")
createNodeVariable <-
  function(values,
           range = NULL,
           nodeSet = "actors",
           addSame = F,
           addSim = F) {
    if (!is.numeric(values) &&
        !all(is.na(values)))
      stop("Values not numeric.")
```

data

Contains data (as *.rda*-files) which can be used to run/calculate examples from the help-files; to add data, load it into the environment => execute usethis::use_data() in project file to save data as *.rda*-file (automatically in *data*-folder) => execute usethis::use_r("") to create associated *.R*-script (in *R*-folder) based on which the help-file for data documentation is created (works similar to documentation of functions) => parameter must be added manually => more information (YouTube)
Recommended: Provide code with which (example) data was created for later changes/updates => more information

- *@name*: define name of the help-file (if desired, otherwise name of the *.rda*-file will be used)
- *@description*: description of the data
- *@format*: specification of the format in which the data are available
- *@rdname*: merges documentation of two different data objects into one help-file (for second data object specify name of first (base) data object)
- *@source*: data source (e.g. URL)

- **Important**: name of the data object must be specified below the parameters in quotes

```
#' Example Data for the MoNAn Package
#'
#' @name exampleData
#' @description
#' These are example data for the MoNAn package and can be used to estimate a
#' mobility network. The following objects are provided for this purpose:
#'
#' \describe{
#'   \item{mobilityEdgelist}{blabla}
#'   \item{nodeVarCat}{blabla}
#'   \item{nodeVarCont}{blabla}
#'   \item{resVarCat}{blabla}
#' }
#'
#' @rdname exampleData
#' @format `mobilityEdgelist`
#' A data frame with 743 rows and 2 columns.
"mobilityEdgelist"


#' @rdname exampleData
#' @format `nodeVarCat`
#' An object with 17 values.
"nodeVarCat"


#' @rdname exampleData
#' @format `nodeVarCont`
#' An object with 17 values.
"nodeVarCont"
```

.gitignore

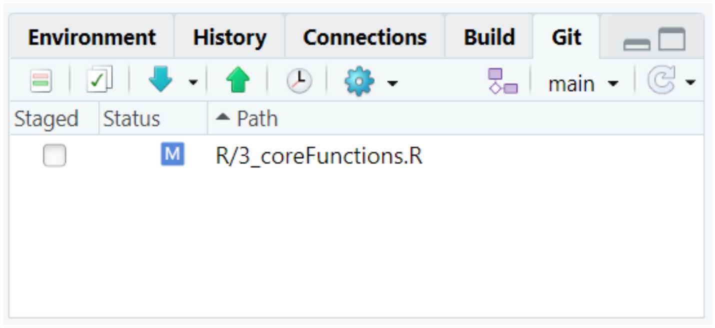
Lists files that should not be visible in GitHub => files can be added automatically executing *usethis::use_git_ignore()*

.Rbuildignore

Lists files that should not be considered when the package is created => files can be added automatically executing *usethis::use_build_ignore()*

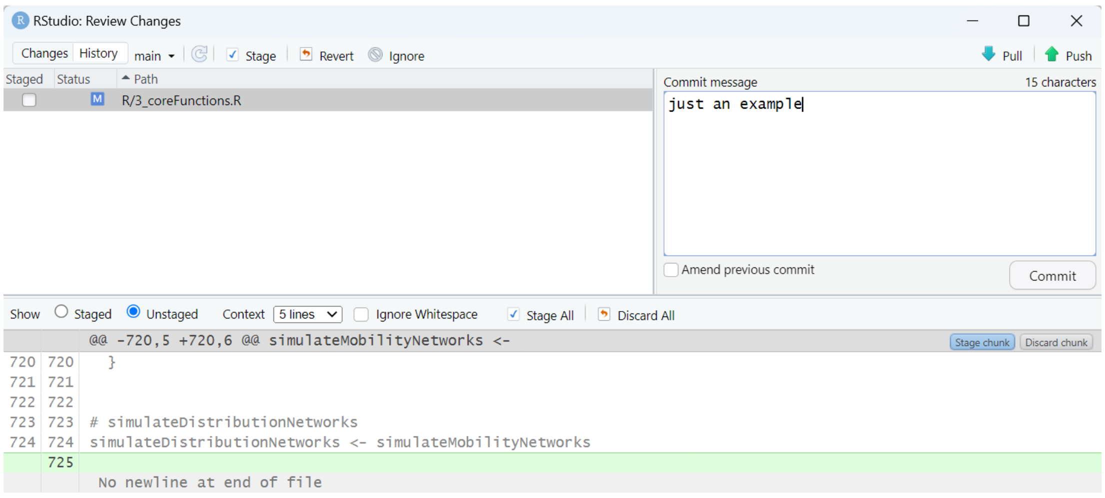
GitHub

GitHub is recommended as version control system => more information



Staged	Status	Path
☐	M	R/3_coreFunctions.R

Changes to the package can be made both in R or directly in the GitHub repository; when working in R, always "push" the latest version first (blue arrow) => "commit" changes with a short description (see below) => finally "push" changes (green arrow), so that the version in the GitHub repository is the latest version



```
just an example
```

```
@@ -720,5 +720,6 @@ simulateMobilityNetworks <-
  }


# simulateDistributionNetworks
simulateDistributionNetworks <- simulateMobilityNetworks

 No newline at end of file
```

Workflow

Procedure to edit the package:

1. Pull latest package version from GitHub (!)
2. Make adjustments directly in *.R*-scripts (also for the help-files – do not change them in the automatically generated *.Rd*-files)
3. Reload package pressing *ctrl + shift + B* **or** *Build > Install > Clean and Install* => update documentation/help-files executing *devtools::document()*
4. Commit changes to GitHub and push them

Other useful information

- *usethis::use_package_doc()*: creates a help-file for the package based on the description-file
- Code-Restyling: formats code into common R notation => more information
- R-scripts are loaded in alphabetical order and from top to bottom => can lead to problems if a certain auxiliary function (e.g. plotting function) is loaded before the actual main function => therefore: if necessary, adjust order manually by numbering scripts or adjusting order of functions within the script

General literature, videos, links

- *R Packages – Organize, test, document and share your code* by Wickham (2015) as PDF or online (up-to-date): very detailed, everything included, also introduction to GitHub
- *Package Development:: Cheat Sheet*: rough overview, useful to look up basic commands/tags/ workflow
- *Building R Packages*: rough overview
- *Building R packages with devtools and usethis | RStudio:* YouTube-video, rather basic but useful when having no idea on how to create a package
- GitHub repositories of *RSiena*, *tidyr*: useful to look up how other packages are created, tidyr also uses *roxygen2*