

# Package ‘pkgnet’

January 3, 2019

**Type** Package

**Title** Get Network Representation of an R Package

**Version** 0.3.0

**Maintainer** Brian Burns <brian.burns@uptake.com>

**Description** Tools from the domain of graph theory can be used to quantify the complexity and vulnerability to failure of a software package. That is the guiding philosophy of this package. 'pkgnet' provides tools to analyze the dependencies between functions in an R package and between its imported packages.

**Imports** assertthat, covr, data.table, DT, futile.logger, igraph, knitr, magrittr, methods, R6, rmarkdown(>= 1.9), tools, visNetwork

**Suggests** devtools, testthat, withr

**License** BSD\_3\_clause + file LICENSE

**URL** <https://github.com/UptakeOpenSource/pkgnet>

**BugReports** <https://github.com/UptakeOpenSource/pkgnet/issues>

**LazyData** TRUE

**RoxygenNote** 6.1.1

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Brian Burns [aut, cre],  
James Lamb [aut],  
Patrick Boueri [ctb],  
Jay Qi [aut]

**Repository** CRAN

**Date/Publication** 2019-01-03 00:40:02 UTC

## R topics documented:

AbstractGraphReporter . . . . .	2
AbstractPackageReporter . . . . .	3

CreatePackageReport . . . . .	4
DefaultReporters . . . . .	4
DependencyReporter . . . . .	5
doc_shared . . . . .	5
FunctionReporter . . . . .	6
InheritanceReporter . . . . .	7
SummaryReporter . . . . .	8
<b>Index</b>	<b>9</b>

---

AbstractGraphReporter *Abstract Graph Reporter Class*

---

## Description

Defines the Abstract Class for all PackageGraphReporters defined in pkgnet. The class is not meant to be instantiated, but inherited from and its methods overloaded such that each Metric implements certain functionality.

## Usage

AbstractGraphReporter

## Format

An object of class R6ClassGenerator of length 24.

## Public Members

edges A data.table from SOURCE to TARGET nodes describing the connections  
nodes A data.table with node as an identifier, and augmenting information about each node  
pkg\_graph An igraph object describing the package graph  
network\_measures A list of network measures calculated by calculate\_network\_features  
layout\_type Character string indicating currently active graph layout  
graph\_viz visNetwork object of package graph

## Active Bindings

pkg\_graph Returns the graph object  
network\_measures Returns a table of network measures, one row per node  
graph\_viz Returns the graph visualization object  
layout\_type If no value given, the current layout type for the graph visualization is returned. If a valid layout type is given, this function will update the layout\_type field. You can use `grep("^layout_\\S", getNamespaceExports("igraph"), value = TRUE)` to see valid options.

**See Also**

Other AbstractReporters: [AbstractPackageReporter](#)

---

AbstractPackageReporter

*Abstract Package Reporter Class*

---

**Description**

Defines the Abstract Class for all PackageReporters defined in pkgnet. The class is not meant to be instantiated, but inherited from and its methods overloaded such that each Metric implements certain functionality.

**Usage**

AbstractPackageReporter

**Format**

An object of class R6ClassGenerator of length 24.

**Public Methods**

`set_package(pkg_name, pkg_path = NULL)` • Set the package that all operations in the object are done for.

- **Args:**

- `pkg_name`: A string with the name of the package you are analyzing.
- `pkg_path`: Optional directory path to source code of the package. It is used for calculating test coverage. It can be an absolute or relative path.

`get_summary_view()` • Returns a particular reporters summary report on the package for use in a high level view

**See Also**

Other AbstractReporters: [AbstractGraphReporter](#)

CreatePackageReport     *Surface the internal and external dependencies of an R package.*

---

**Description**

Surface the internal and external dependencies of an R package.

**Usage**

```
CreatePackageReport(pkg_name, pkg_reporters = DefaultReporters(),  
  pkg_path = NULL, report_path = tempfile(pattern = pkg_name, fileext =  
  ".html"))
```

**Arguments**

pkg\_name            (string) name of a package  
pkg\_reporters      (list) a list of package reporters  
pkg\_path            (string) The path to the package repository. If given, coverage will be calculated for each function. pkg\_path can be an absolute or relative path.  
report\_path        (string) The path and filename of the output report. Default report will be produced in the temporary directory.

**Value**

A list of instantiated pkg\_reporters fitted to pkg\_name

**Author(s)**

B. Burns

**See Also**

GetPackageGraphs

---

DefaultReporters     *Default Reporters*

---

**Description**

Instantiates a list of default reporters to feed into [CreatePackageReport](#)

**Usage**

```
DefaultReporters()
```

---

DependencyReporter      *Package Dependency Reporter Class*

---

### Description

This Reporter takes a package and uncovers the structure from its other package dependencies, determining which package it relies on is most central, allowing for a developer to determine how to vet its dependency tree

### Usage

```
DependencyReporter
```

### Format

An object of class R6ClassGenerator of length 24.

### See Also

Other PackageReporters: [FunctionReporter](#), [InheritanceReporter](#), [SummaryReporter](#)

### Examples

```
# Instantiate an object
reporter <- DependencyReporter$new()

# Seed it with a package
reporter$set_package("ggplot2")

# plot it up
reporter$plot_network()
```

---

doc\_shared      *NULL Object For Common Documentation*

---

### Description

This is a NULL object with documentation so that later functions can call inheritParams

### Arguments

pkg_name	(string) name of a package
pkg_path	(string) The path to the package repository. If given, coverage will be calculated for each function. pkg_path can be an absolute or relative path.

---

FunctionReporter      *Package Function Reporter Class*

---

## Description

This Reporter takes a package and uncovers the structure from its other functions, determining useful information such as which function is most central to the package. Combined with testing information it can be used as a powerful tool to plan testing efforts.

R6 classes are supported, with methods treated as functions by the Reporter. R6 methods will be named like `<classname>$<methodtype>$<methodname>`, e.g., `FunctionReporter$private_methods$extract_nodes`. Note that the class name used will be the *name of the generator object in the package's namespace*, and *not* the `classname` attribute of the class, which is not required to be defined or to be the same as the generator object name.

## Usage

FunctionReporter

## Format

An object of class `R6ClassGenerator` of length 24.

## Public Methods

`set_package(pkg_name, pkg_path)` • Set properties of this reporter. If `pkg_name` overrides a previously-set package name, any cached data will be removed.

- **Args:**
  - `pkg_name`: String with the name of the package.
  - `pkg_path`: Optional directory path to source code of the package. It is used for calculating test coverage. It can be an absolute or relative path.

## Known Limitations

- Using non-standard evaluation to refer to things (e.g, dataframe column names) that have the same name as a function will trick `FunctionReporter` into thinking the function was called. This can be avoided if you don't use reuse function names for other purposes.
- Functions stored as list items and not assigned to the package namespace will be invisible to `FunctionReporter`.
- Calls to methods of instantiated R6 or reference objects will not be recognized. We don't have a reliable way of identifying instantiated objects, or identifying their class.
- Reference class methods are not yet supported. They will not be identified as nodes by `FunctionReporter`.

## See Also

Other PackageReporters: [DependencyReporter](#), [InheritanceReporter](#), [SummaryReporter](#)

---

InheritanceReporter     *Package Class Inheritance Reporter Class*

---

### Description

This reporter takes a package and traces the class inheritance structure. Currently the following object-oriented systems are supported:

- S4 Classes
- Reference Classes (sometimes informally called "R5")
- R6 Classes

S3 classes are not supported, as their inheritance is defined on an ad hoc basis per object and not formally by class definitions.

Note the following details about class naming:

- Reference Classes : The name passed as `Class` in `setRefClass` is used as the node name by this reporter. This is the class name that is used when specifying inheritance. The generator object returned by `setRefClass` does not have to be assigned and can have a different name.
- R6 Classes : The name of the generator object in the package namespace is used as the node name by this reporter. The generator object returned by `R6::R6Class` is what is used when specifying inheritance. The name passed as `classname` passed to `R6::R6Class` can be a different name or even `NULL`.

For more info about R's built-in object-oriented systems, check out the relevant chapter in [Hadley Wickham's \*Advanced R\*](#). For more info about R6, check out their [docs website](#) or the chapter in [Advanced R's second edition](#).

### Usage

InheritanceReporter

### Format

An object of class `R6ClassGenerator` of length 24.

### Public Methods

`set_package(pkg_name, pkg_path)`     • Set properties of this reporter. If `pkg_name` overrides a previously-set package name, any cached data will be removed.

- **Args:**
  - `pkg_name`: String with the name of the package.
  - `pkg_path`: Optional directory path to source code of the package. It is used for calculating test coverage. It can be an absolute or relative path.

### See Also

Other PackageReporters: [DependencyReporter](#), [FunctionReporter](#), [SummaryReporter](#)

---

SummaryReporter

*Package Summary Reporter Class*

---

### Description

Defines a concrete implementation of [AbstractPackageReporter](#) for a high level overview of a particular package. It will summarize things like lines of code, whether it's on CRAN, etc.

### Usage

```
SummaryReporter
```

### Format

An object of class R6ClassGenerator of length 24.

### Public Methods

`set_package(pkg_name, pkg_path = NULL)` • Set the package that all operations in the object are done for.

- **Args:**

- `pkg_name`: A string with the name of the package you are analyzing.
- `pkg_path`: Optional directory path to source code of the package. It is used for calculating test coverage. It can be an absolute or relative path.

`get_summary_view()` • Returns a particular reporters summary report on the package for use in a high level view

### See Also

Other PackageReporters: [DependencyReporter](#), [FunctionReporter](#), [InheritanceReporter](#)



# Index

## \*Topic **datasets**

- AbstractGraphReporter, [2](#)
- AbstractPackageReporter, [3](#)
- DependencyReporter, [5](#)
- FunctionReporter, [6](#)
- InheritanceReporter, [7](#)
- SummaryReporter, [8](#)

AbstractGraphReporter, [2](#), [3](#)  
AbstractPackageReporter, [3](#), [3](#), [8](#)

CreatePackageReport, [4](#), [4](#)

DefaultReporters, [4](#)  
DependencyReporter, [5](#), [6–8](#)  
doc\_shared, [5](#)

FunctionReporter, [5](#), [6](#), [7](#), [8](#)

InheritanceReporter, [5](#), [6](#), [7](#), [8](#)

R6::R6Class, [7](#)

setRefClass, [7](#)  
SummaryReporter, [5–7](#), [8](#)