Package ‘markovchain’

February 21, 2015

Type Package
Title Easy Handling Discrete Time Markov Chains
Version 0.1.1
Date 2015-02-20
Author Giorgio Alfredo Spedicato
Maintainer Giorgio Alfredo Spedicato <spedicato_giorgio@yahoo.it>

Description
Functions and S4 methods to create and manage discrete time Markov chains (DTMC) more easily. In addition functions to perform statistical (fitting and drawing random variates) and probabilistic (analysis of DTMC proprieties) analysis are provided.

License GPL-2
Depends R (>= 3.0.0), methods
Imports igraph, Matrix, matlab, expm, stats4, parallel
LazyLoad yes
ByteCompile yes
BugReports Giorgio Alfredo Spedicato <spedicato_giorgio@yahoo.it>
Suggests testthat
NeedsCompilation no
Repository CRAN
Date/Publication 2015-02-21 18:10:14

R topics documented:

markovchain-package ...................................................... 2
absorbingStates .......................................................... 3
blanden ...................................................................... 4
conditionalDistribution ................................................... 5
craigsendi ................................................................. 6
firstPassage ................................................................. 7
holson ....................................................................... 8
**Description**

The package contains classes and method to create and manage (plot, print, export for example) discrete time Markov chains (DTMC). In addition it provide functions to perform statistical (fitting and drawing random variates) and probabilistic (analysis of DTMC properties) analysis.

**Details**

- **Package**: markovchain
- **Type**: Package
- **Version**: 0.1.1
- **Date**: 2015-02-20
- **License**: GPL-2
- **Depends**: R (>= 2.14), methods, expm, matlab, igraph, Matrix

**Author(s)**

Giorgio Alfredo Spedicato Maintainer: Giorgio Alfredo Spedicato <spedicato_giorgio@yahoo.it>

**References**

A First Course in Probability (8th Edition), Sheldon Ross, Prentice Hall 2010

**Examples**

```r
# create some markov chains
statesNames=c("a","b")
```
absorbingStates

mca<-new("markovchain", transitionMatrix=matrix(c(0.7,0.3,0.1,0.9), byrow=TRUE, nrow=2, dimnames=list(statesNames, statesNames)))

statesNames=c("a","b","c")
mcb<-new("markovchain", states=statesNames, transitionMatrix=
matrix(c(0.2,0.5,0.3,
0.1,0,
0.1,0.8,0.1), byrow=TRUE, dimnames=list(statesNames, statesNames))

statesNames=c("a","b","c","d")
matrice<-matrix(c(0.25,0.75,0,0,0.4,0.6,0,0,0,0,0.1,0.9,0,0,0.7,0.3), nrow=4, byrow=TRUE)
mcC<-new("markovchain", states=statesNames, transitionMatrix=matrice)
mcd<-new("markovchain", transitionMatrix=matrix(c(0,1,0,1), nrow=2, byrow=TRUE))

#operations with S4 methods
mca^2
steadyStates(mcb)
absorbingStates(mcb)
markovchainSequence(n=20, markovchain=mcC, include=TRUE)

---

absorbingStates | Various function to perform statistical and probabilistic analysis

**Description**

These functions return absorbing and transient states of the markovchain objects.

**Usage**

absorbingStates(object)
transientStates(object)
canonicForm(object)

**Arguments**

object | A markovchain object.

**Value**

A matrix
Author(s)
Giorgio Spedicato

References
Feres, Matlab listing for markov chain.

See Also
markovchain

Examples
```r
statesNames <- c("a", "b", "c")
markovB <- new("markovchain", states = statesNames, transitionMatrix =
  matrix(c(0.2, 0.5, 0.3,
           0.1, 0.8, 0.1), byrow = TRUE, dimnames = list(statesNames, statesNames)
  )
absorbingStates(markovB)
transientStates(markovB)
canoncForm(markovB)
# periodicity analysis
E <- matrix(0, nrow = 4, ncol = 4)
E[1, 2] <- 1
# E[2, c(1, 3)] = 1/2;
E[2, 1] <- 1/3; E[2, 3] <- 2/3
# E[3, c(2, 4)] = 1/2;
E[4, 3] <- 1
mcE <- new("markovchain", states = c("a", "b", "c", "d"),
  transitionMatrix = E,
  name = "E")
is irreducible(mce) # true
period(mce) # 2
```

---

blanden Mobility between income quartiles

Description
This table shows mobility between income quartiles for father and sons for the 1970 cohort born

Usage
data(blanden)
**Format**

The format is: table [1:4, 1:4] 0.38 0.25 0.16 0.29 0.28 0.17 0.26 0.17 ... - attr(*, "dim-names")=List of 2 ..$ : chr [1:4] "Bottom" "2nd" "3rd" "Top" ..$ : chr [1:4] "Bottom" "2nd" "3rd" "Top"

**Details**

The rows represent father's income quartile when the son is aged 16, whilst the columns represent sons' income quartiles when he is aged 30 (in 2000).

**Source**

Giorgio Spedicato from references

**References**

Jo Blanden, Paul Gregg and Stephen Machin, Intergenerational Mobility in Europe and North America, Center for Economic Performances (2005)

**Examples**

```r
data(blanden)
mobilityMc<-as(blanden, "markovchain")
```

---

**Description**

It extracts the conditional distribution of the subsequent state, given current state.

**Usage**

```r
conditionalDistribution(object, state)
```

**Arguments**

- `object` A markovchain object.
- `state` Subsequent state.

**Value**

A named probability vector

**Author(s)**

Giorgio Spedicato
References

A First Course in Probability (8th Edition), Sheldon Ross, Prentice Hall 2010

See Also

markovchain

Examples

# define a markov chain
statesNames <- c("a", "b", "c")
markovB <- new("markovchain", states = statesNames, transitionMatrix = matrix(c(0.2, 0.5, 0.3, 0.1, 0.1, 0.8, 0.1), nrow = 3, byrow = TRUE, dimnames = list(statesNames, statesNames)))
conditionalDistribution(markovB, "b")

craigsendi                  CD4 cells counts on HIV Infects between zero and six month

Description

This is the table shown in Craig and Sendi paper showing zero and six month CD4 cells count in six brackets

Usage

data(craigsendi)

Format

The format is: table [1:3, 1:3] 682 154 19 33 64 19 25 47 43 - attr(*, "dimnames")=List of 2 ..$ : chr [1:3] "0-49" "50-74" "75-UP" ..$: chr [1:3] "0-49" "50-74" "75-UP"

Details

Rows represent counts at the beginning, cols represent counts after six months.

Source


References

See source

Examples

data(craigsendi)
csMc <- as(craigsendi, "markovchain")
steadyStates(csMc)
firstPassage  

First passage across states

Description

This function compute the first passage probability in states

Usage

firstPassage(object, state, n)

Arguments

object    A markovchain object
state     Initial state
n         Number of rows on which compute the distribution

Details

Based on Feres’ Matlab listings

Value

A matrix of size 1:n x number of states showing the probability of the first time of passage in states to be exactly the number in the row.

Author(s)

Giorgio Spedicato

References

Renaldo Feres, Notes for Math 450 Matlab listings for Markov chains

See Also

conditionalDistribution

Examples

# create a simple Markov chain
simpleMc<-new("markovchain", states=c("a","b"),
transitionMatrix=matrix(c(0.4,0.6,0.3,.7),nrow=2,byrow=TRUE))
firstPassage(simpleMc,"b",20)
holson

Holson data set

Description
A data set containing 1000 life histories trajectories and a categorical status (1,2,3) observed on eleven evenly spaced steps.

Usage
data("holson")

Format
A data frame with 1000 observations on the following 12 variables.

- id: unique id
- time1: observed status at i-th time
- time2: observed status at i-th time
- time3: observed status at i-th time
- time4: observed status at i-th time
- time5: observed status at i-th time
- time6: observed status at i-th time
- time7: observed status at i-th time
- time8: observed status at i-th time
- time9: observed status at i-th time
- time10: observed status at i-th time
- time11: observed status at i-th time

Details
The example can be used to fit a markovchain or a markovchainList object.

Source
Private communications

References
Private communications

Examples
data(holson)
head(holson)
is.accessible

Verify if a state \( j \) is reachable from state \( i \).

Description

This function verifies if a state is reachable from another, i.e., if exists a path that leads to state \( j \) leaving from state \( i \) with positive probability.

Usage

\[
is\text{.accessible}(\text{object, from, to})
\]

Arguments

- **object**: A markovchain object.
- **from**: The name of state "i" (beginning state).
- **to**: The name of state "j" (ending state).

Details

If wraps and internal function named .commStatesFinder.

Value

A boolean value.

Author(s)

Giorgio Spedicato

References

James Montgomery, University of Madison

See Also

is.irreducible

Examples

statesNames<-c("a","b","c")
markovB<-new("markovchain", states=statesNames, transitionMatrix=
matrix(c(0.2,0.5,0.3,
0.1,0,0.8,0.1),nrow=3, byrow=TRUE, dimnames=list(statesNames,statesNames))
is.accessible(markovB,"a","c")
**is.irreducible**

*Function to check if a Markov chain is irreducible*

**Description**
This function verifies whether a markovchain object transition matrix is composed by only one communicating class.

**Usage**

```r
is.irreducible(object)
```

**Arguments**

- `object` A markovchain object

**Details**

It is based on `.communicatingClasses` internal function.

**Value**

A boolean values.

**Author(s)**

Giorgio Spedicato

**References**

Feres, Matlab listings for Markov Chains.

**See Also**

`summary`

**Examples**

```r
statesNames=c("a","b")
mcA<-new("markovchain", transitionMatrix=matrix(c(0.7,0.3,0.1,0.9),byrow=TRUE, nrow=2,
                                                dimnames=list(statesNames,statesNames))

is.irreducible(mcA)
```
Description

The S4 class that describes markovchain objects.

Objects from the Class

Objects can be created by calls of the form `new("markovchain", states, byrow, transitionMatrix, ...)`. 

Slots

states: Name of the states. Must be the same of colnames and rownames of the transition matrix
byrow: Binary flag.
transitionMatrix: Square transition matrix
name: Optional character name of the Markov chain

Methods

* signature(e1 = "markovchain", e2 = "markovchain"): multiply two markovchain objects
* signature(e1 = "markovchain", e2 = "matrix"): markovchain by matrix multiplication
* signature(e1 = "markovchain", e2 = "numeric"): markovchain by numeric vector multiplication
* signature(e1 = "matrix", e2 = "markovchain"): matrix by markov chain
* signature(e1 = "numeric", e2 = "markovchain"): numeric vector by markovchain multiplication

[ signature(x = "markovchain", i = "ANY", j = "ANY", drop = "ANY"): ...
^ signature(e1 = "markovchain", e2 = "numeric"): power of a markovchain object
== signature(e1 = "markovchain", e2 = "markovchain"): equality of two markovchain objects

absorbingStates signature(object = "markovchain"): method to get absorbing states
canonicForm signature(object = "markovchain"): return a markovchain object into canonic form
coerce signature(from = "markovchain", to = "data.frame"): coerce method from markovchain to data.frame
conditionalDistribution signature(object = "markovchain"): returns the conditional probability of subsequent states given a state
coerce signature(from = "data.frame", to = "markovchain"): coerce method from data.frame to markovchain
coerce signature(from = "table", to = "markovchain"): coerce method from table to markovchain
coerce signature(from = "markovchain", to = "igraph"): coercing to igraph objects

coerce signature(from = "markovchain", to = "matrix"): coercing to matrix objects

coerce signature(from = "matrix", to = "markovchain"): coercing to markovchain objects from matrix one

dim signature(x = "markovchain"): method to get the size

initialize signature(.Object = "markovchain"): initialize method

plot signature(x = "markovchain", y = "missing"): plot method for markovchain objects

predict signature(object = "markovchain"): predict method

print signature(x = "markovchain"): print method.

show signature(object = "markovchain"): show method.

states signature(object = "markovchain"): states method.

steadyStates signature(object = "markovchain"): method to get the steady vector.

summary signature(object = "markovchain"): method to summarize structure of the markov chain

transientStates signature(object = "markovchain"): method to get the transient states.

t signature(x = "markovchain"): transpose matrix

transitionProbability signature(object = "markovchain"): transition probability

Warning

Validation method is used to assess whether either columns or rows totals to one. Rounding is used up to 5th decimal. If state names are not properly defined for a probability matrix, coercing to markovchain object leads to overriding states name with artificial "s1", "s2", ... sequence

Note

markovchain object are written in S4 Classes.

Author(s)

Giorgio Spedicato

References

A First Course in Probability (8th Edition), Sheldon Ross, Prentice Hall 2010

See Also

markovchainSequence,markovchainFit
**Examples**

```r
# show markovchain definition
desc(markovchain)
# create a simple Markov chain
transmatr <- matrix(c(0.4, 0.6, 0.3, 0.7), nrow = 2, byrow = TRUE)
simpleMc <- new("markovchain", states = c("a", "b"),
                transitionMatrix = transMatr,
                name = "simpleMc")
# power
simpleMc^
# some methods
steadyStates(simpleMc)
absorbingStates(simpleMc)
simpleMc[2, 1]
t(simpleMc)
is.irreducible(simpleMc)
# conditional distributions
conditionalDistribution(simpleMc, "b")
# example for predict method
sequence <- c("a", "b", "a", "a", "b", "a", "b", 
              "a", "a", "b", "a", "b", "b", "b", "b", "b",  
              "a")
mcFit <- markovchainFit(data = sequence)
predict(mcFit$estimate, newdata = "b", n.ahead = 3)
# direct conversion
myMc <- as(transMatr, "markovchain")
# example of summary
summary(simpleMc)
## Not run: plot(simpleMc)
```

---

**markovchainFit**

*Function to fit a discrete Markov chain*

**Description**

Given a sequence of states arising from a stationary state, it fits the underlying Markov chain distribution using either MLE (also using a Laplacian smoother) or bootstrap.

**Usage**

```r
markovchainFit(data, method = "mle", byrow = TRUE, nboot = 10, laplacian = 0, name, parallel = FALSE)
createSequenceMatrix(stringchar, toRowProbs = FALSE, sanitize = TRUE)
```

**Arguments**

- **data**: A character list.
- **method**: Method used to estimate the Markov chain. Either "mle" or "bootstrap" or "laplace"
markovchainFit

byrow  it tells whether the output Markov chain should show the transition probabilities by row.
nboot  Number of bootstrap replicates in case "bootstrap" is used.
laplacian  Laplacian smoothing parameter, default zero. It is only used when "laplace" method is chosen.
name  Optional character for name slot.
parallel  Use parallel processing when performing Boostrap estimates.
stringchar  Equivalent to data
toRowProbs  converts a sequence matrix into a probability matrix
sanitize  put 1 in all rows having rowSum equal to zero

Value

A list containing an estimate and, when "bootstrap" method is used, a matrix of standards deviations and the bootstrap samples.

Warning

"mle" method calls createSequenceMatrix function using sanitize parameter set to TRUE.

Note

When MLE method is called, the lists contains one entry: estimate. Bootstrap algorithm has been defined "euristically". In addition, parallel facility is not complete, involving only a part of the bootstrap process. When data is either a data.frame or a matrix object, only MLE fit is currently available.

Author(s)

Giorgio Spedicato

References

A First Course in Probability (8th Edition), Sheldon Ross, Prentice Hall 2010

See Also

markovchainSequence, markovchainListFit

Examples

sequence<-("a", "b", "a", "a", "a", "b", "a", "b", "a", "b", "a", "b", "a", "a", "a", "b", "b", "b", "a")
sequenceMatr<-createSequenceMatrix(sequence, sanitize=FALSE)
mclMLE<-markovchainFit(data=sequence)
mclBSP<-markovchainFit(data=sequence, method="bootstrap", nboot=5, name="Bootstrap Mc")
```
markovchainList-class

markovchainList-class  Class "markovchainList"

Description

A class to handle non-homogeneous Markov chains

Objects from the Class

Objects can be created by calls of the form `new("markovchainList", ...)`. Each item in the list is a markovchain object.

Slots

markovchains: Object of class "list": a list of markovchains
name: Object of class "character": optional name of the class

Methods

[[ signature(x = "markovchainList"): extract the i-th markovchain
dim signature(x = "markovchainList"): number of markovchain underlying the matrix
predict signature(object = "markovchainList"): predict from a markovchainList
print signature(x = "markovchainList"): prints the list of markovchains
show signature(object = "markovchainList"): same as print

Note

The class consists in a list of markovchain objects. It can help to deal with non-homogeneous Markov chains.

Author(s)

Giorgio Spedicato

References

A First Course in Probability (8th Edition), Sheldon Ross, Prentice Hall 2010

See Also

markovchain
```
Examples

showClass("markovchainList")
#define a markovchainList
statesNames=c("a", "b")

mcA<-new("markovchain", name="MCA", transitionMatrix=matrix(c(0.7, 0.3, 0.1, 0.9),
byrow=TRUE, nrow=2, dimnames=list(statesNames, statesNames)))

mcB<-new("markovchain", states=c("a", "b", "c"), name="MCB",
transitionMatrix=matrix(c(0.2, 0.5, 0.3, 0.1, 0.8, 0.1),
byrow=TRUE))

mcC<-new("markovchain", states=c("a", "b", "c", "d"), name="MCC",
transitionMatrix=matrix(c(0.25, 0.75, 0.0, 0.4, 0.6,
0, 0, 0, 0.1, 0.9, 0.0, 0.7, 0.3), nrow=4, byrow=TRUE))

mclist<-new("markovchainList", markovchains=list(mcA, mcB, mcC),
name="Non-homogeneous Markov Chain")

markovchainListFit  Fit a markovchainList

Description

Fit a markovchainList

Usage

markovchainListFit(data, byrow = TRUE, laplacian = 0, name)

Arguments

data Either a matrix or a data.frame object.
byrow Indicates whether distinct stochastic processes trajectories are shown in distinct rows.
laplacian Laplacian correction (default 0).
name Optional name.

Value

A list containing two slots: estimate (the estimate) name

Examples

#using holson dataset
data(holson)
#fitting a single markovchain
singleMc<-markovchainFit(data=holson[,2:12])
#fitting a markovchainList
mclistFit<-markovchainListFit(data=holson[,2:12], name="holsonMcList")
Preproglucacon DNA protein bases sequences

Description
Sequence of bases for preproglucacon DNA protein

Usage
data(preproglucacon)

Format
A data frame with 1572 observations on the following 2 variables.

v1  a numeric vector, showing original coding
preproglucacon  a character vector, showing initial of DNA bases (Adenine, Cytosine, Guanine, Thymine)

Source
Avery Henderson

References
Avery Henderson, Fitting markov chain models on discrete time series such as DNA sequences

Examples
data(preproglucacon)
preproglucaconMc<-markovchainFit(data=preproglucacon$preproglucacon)

rain Alofi island daily rainfall

Description
Rainfall measured in Alofi Island

Usage
data(rain)
Format
A data frame with 1096 observations on the following 2 variables.

v1  a numeric vector, showing original coding
rain a character vector, showing daily rainfall millilitres brackets

Source
Avery Henderson

References
Avery Henderson, Fitting markov chain models on discrete time series such as DNA sequences

Examples

data(rain)
rainMc<-markovchainFit(data=rain$rain)

rmarkovchain
Function to generate a sequence of states from homogeneous or non-homogeneous Markov chains.

Description
Provided any markovchain or markovchainList objects, it returns a sequence of states coming from the underlying stationary distribution.

Usage
rmarkovchain(n, object, ...)
markovchainSequence(n, markovchain, t0 = sample(markovchain@states, 1),
include.t0 = FALSE)

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>n</td>
<td>Sample size</td>
</tr>
<tr>
<td>object</td>
<td>Either a markovchain or a markovchainList object.</td>
</tr>
<tr>
<td>...</td>
<td>additional parameters passed to the internal sampler</td>
</tr>
<tr>
<td>markovchain</td>
<td>The markovchain object</td>
</tr>
<tr>
<td>t0</td>
<td>The initial state.</td>
</tr>
<tr>
<td>include.t0</td>
<td>Specify if the initial state shall be used.</td>
</tr>
</tbody>
</table>
Details

When an homogeneous process is assumed (markovchain object) a sequence is sampled of size n. When an non-homogeneous process is assumed, n samples are taken but the process is assumed to last from the begin to the end of the non-homogeneous markov process.

Value

Either a character vector or a data frame

Note

Check the type of input

Author(s)

Giorgio Spedicato

References

A First Course in Probability (8th Edition), Sheldon Ross, Prentice Hall 2010

See Also

markovchainFit

Examples

#define the Markov chain
statesNames<-c("a","b","c")
mcB<-new("markovchain", states=statesNames, transitionMatrix=matrix(c(0.2,0.5,0.3,
0,0.2,0.8,0.1,0.8,0.1),nrow=3, byrow=TRUE, dimnames=list(statesNames,statesNames))
)#show the sequence
outs<-markovchainSequence(n=100,markovchain=mcB, t0="a")
outs2<-rmarkovchain(n=20, object=mcB)

---

states 

Defined states of a transition matrix

Description

This method returns the states of a transition matrix.

Usage

states(object)
steadyStates

Arguments

object  A discrete markovchain object

Value

The character vector corresponding to states slot.

Author(s)

Giorgio Spedicato

References

A First Course in Probability (8th Edition), Sheldon Ross, Prentice Hall 2010

See Also

markovchain

Examples

statesNames=c("a","b","c")
markovB<-new("markovchain", states=statesNames, transitionMatrix=
  matrix(c(0.2,0.5,0.3,
          0,1,0,
          0.1,0.8,0.1),nrow=3, byrow=TRUE, dimnames=list(statesNames,statesNames))
states(markovB)

steadyStates  Stationary states of a markovchain object

Description

This method returns the stationary vector in matricial form of a markovchain object.

Usage

steadyStates(object)

Arguments

object  A discrete markovchain object

Value

A matrix corresponding to the stationary states
Note
The steady states are identified starting from which eigenvectors correspond to identity eigenvalues and then normalizing them to sum up to unity.

Author(s)
Giorgio Spedicato

References
A First Course in Probability (8th Edition), Sheldon Ross, Prentice Hall 2010

See Also
markovchain

Examples
```r
statesNames <- c("a", "b", "c")
markovB <- new("markovchain", states=statesNames, transitionMatrix=
  matrix(c(0.2, 0.5, 0.3,
          0, 1, 0,
          0.1, 0.8, 0.1), nrow=3, byrow=TRUE, dimnames=list(statesNames, statesNames))
steadyStates(markovB)
```

---

**transitionProbability**  
*Function to get the transition probabilities from initial to subsequent status.*

Description
This is a convenience function to get transition probabilities.

Usage
```r
transitionProbability(object, t0, t1)
```

Arguments
- **object**: A markovchain object.
- **t0**: Initial state.
- **t1**: Subsequent state.

Value
A matrix
transitionProbability

Author(s)
Giorgio Spedicato

References
A First Course in Probability (8th Edition), Sheldon Ross, Prentice Hall 2010

See Also
markovchain

Examples
#define a markov chain
statesNames=c("a","b","c")
markovB<-new("markovchain",states=statesNames,transitionMatrix=
matrix(c(0.2,0.5,0.3,
0.1,0.8,0.1,nnrow=3,byrow=TRUE,dimnames=list(statesNames,statesNames))
)
transitionProbability(markovB,"b","c")
Index

*Topic classes
  markovchain-class, 11
  markovchainList-class, 15
*Topic datasets
  blanden, 4
  craigsendi, 6
  holson, 8
  preproglucacon, 17
  rain, 17
*Topic package
  markovchain-package, 2
  *,markovchain,markovchain-method
    (markovchain-class), 11
  *,markovchain,matrix-method
    (markovchain-class), 11
  *,markovchain,numeric-method
    (markovchain-class), 11
  *,matrix,markovchain-method
    (markovchain-class), 11
  *,numeric,markovchain-method
    (markovchain-class), 11
  ==,markovchain,markovchain-method
    (markovchain-class), 11
  [,markovchain,ANY,ANY,ANY-method
    (markovchain-class), 11
  [][]*,markovchainList-method
    (markovchainList-class), 15
  ^,markovchain,numeric-method
    (markovchain-class), 11

  absorbingStates, 3
  absorbingStates,markovchain-method
    (markovchain-class), 11
  blanden, 4
  canonicForm (absorbingStates), 3
  canonicForm,markovchain-method
    (markovchain-class), 11
  coerce, data.frame,markovchain-method
    (markovchain-class), 11
  coerce,markovchain,data.frame-method
    (markovchain-class), 11
  coerce,markovchain,igraph-method
    (markovchain-class), 11
  coerce,markovchain,matrix-method
    (markovchain-class), 11
  coerce,matrix,markovchain-method
    (markovchain-class), 11
  coerce,table,markovchain-method
    (markovchain-class), 11
  conditionalDistribution, 5, 7
  conditionalDistribution,markovchain-method
    (markovchain-class), 11
  craigsendi, 6
  createSequenceMatrix (markovchainFit), 13
  dim,markovchain-method
    (markovchain-class), 11
  dim,markovchainList-method
    (markovchainList-class), 15
  firstPassage, 7
  holson, 8
  initialize,markovchain-method
    (markovchain-class), 11
  is.accessible, 9
  is.irreducible, 10
  markovchain, 4, 6, 15, 20–22
  markovchain (markovchain-package), 2
  markovchain-class, 11
  markovchain-package, 2
  markovchainFit, 12, 13, 19
  markovchainList-class, 15
  markovchainListFit, 14, 16
  markovchainSequence, 12, 14
markovchainSequence (rmarkovchain), 18

period (absorbingStates), 3
plot, markovchain, missing-method
   (markovchain-class), 11
predict, markovchain-method
   (markovchain-class), 11
predict, markovchainList-method
   (markovchainList-class), 15
preproglucacon, 17
print, markovchain-method
   (markovchain-class), 11
print, markovchainList-method
   (markovchainList-class), 15

rain, 17
rmarkovchain, 18

show, markovchain-method
   (markovchain-class), 11
show, markovchainList-method
   (markovchainList-class), 15
states, 19
states, markovchain-method
   (markovchain-class), 11
steadyStates, 20
steadyStates, markovchain-method
   (markovchain-class), 11
summary, 10
summary, markovchain-method
   (markovchain-class), 11

t, markovchain-method
   (markovchain-class), 11
transientStates (absorbingStates), 3
transientStates, markovchain-method
   (markovchain-class), 11
transitionProbability, 21
transitionProbability, markovchain-method
   (markovchain-class), 11