

Package ‘Rlda’

July 13, 2018

Type Package

Title Bayesian LDA for Mixed-Membership Clustering Analysis

Version 0.2.5

Author Pedro Albuquerque and Denis Valle and Daijiang Li

URL <https://github.com/PedroBSB/Rlda>

Maintainer Pedro Albuquerque <pedroa@unb.br>

BugReports <https://github.com/PedroBSB/Rlda/issues>

Description Estimates the Bayesian LDA model for mixed-membership clustering based on different types of data (i.e., Multinomial, Bernoulli, and Binomial entries). Albuquerque, Valle and Li (2017) <doi:10.13140/RG.2.2.34599.96164>.

License GPL-2

LazyData TRUE

Depends R (>= 2.10), Rcpp (>= 0.9.4), RcppProgress (>= 0.1), doParallel (>= 1.0.10), foreach (>= 1.4.3), coda (>= 0.19.1)

LinkingTo Rcpp, RcppArmadillo, RcppProgress

Imports parallel, gtools

SystemRequirements GNU make

Suggests knitr, MCMCpack, rmarkdown, RColorBrewer, reshape2

RoxygenNote 6.0.1

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-07-13 14:20:06 UTC

R topics documented:

birds	2
complaints	13
fishnet	14

generateBernoulliLDA	14
generateBinomialLDA	15
generateMultinomialLDA	17
getPhi	18
getTheta	19
Landsat	20
LocationsBirds	21
logLik	22
plot	23
predict	24
presence	25
print	26
rlda.bernoulli	27
rlda.bernoulliMH	29
rlda.binomial	31
rlda.binomialMH	34
rlda.binomialVB	36
rlda.fastbernoulli	38
rlda.multinomial	40
rlda2mcmc	42
sp500	43
summary	44

Index	46
--------------	-----------

birds	<i>Breeding Bird Survey</i>
-------	-----------------------------

Description

This dataset is a subset of the Breeding Bird Survey.

Usage

```
data("birds")
```

Format

A data frame with 13608 observations and 384 variables.

loc.id Location index

aou530 Bird specie 530 presence (1) or absence (0)

aou540 Bird specie 540 presence (1) or absence (0)

aou590 Bird specie 590 presence (1) or absence (0)

aou1200 Bird specie 1200 presence (1) or absence (0)

aou1320 Bird specie 1320 presence (1) or absence (0)

aou1400 Bird specie 1400 presence (1) or absence (0)

aou1420 Bird specie 1420 presence (1) or absence (0)
aou2140 Bird specie 2140 presence (1) or absence (0)
aou2300 Bird specie 2300 presence (1) or absence (0)
aou2490 Bird specie 2490 presence (1) or absence (0)
aou2580 Bird specie 2580 presence (1) or absence (0)
aou2640 Bird specie 2640 presence (1) or absence (0)
aou2730 Bird specie 2730 presence (1) or absence (0)
aou3091 Bird specie 3091 presence (1) or absence (0)
aou3160 Bird specie 3160 presence (1) or absence (0)
aou3310 Bird specie 3310 presence (1) or absence (0)
aou3370 Bird specie 3370 presence (1) or absence (0)
aou4440 Bird specie 4440 presence (1) or absence (0)
aou4740 Bird specie 4740 presence (1) or absence (0)
aou4750 Bird specie 4750 presence (1) or absence (0)
aou4880 Bird specie 4880 presence (1) or absence (0)
aou4930 Bird specie 4930 presence (1) or absence (0)
aou4950 Bird specie 4950 presence (1) or absence (0)
aou4970 Bird specie 4970 presence (1) or absence (0)
aou4980 Bird specie 4980 presence (1) or absence (0)
aou5011 Bird specie 5011 presence (1) or absence (0)
aou5100 Bird specie 5100 presence (1) or absence (0)
aou5290 Bird specie 5290 presence (1) or absence (0)
aou5380 Bird specie 5380 presence (1) or absence (0)
aou5400 Bird specie 5400 presence (1) or absence (0)
aou5420 Bird specie 5420 presence (1) or absence (0)
aou5610 Bird specie 5610 presence (1) or absence (0)
aou5620 Bird specie 5620 presence (1) or absence (0)
aou5810 Bird specie 5810 presence (1) or absence (0)
aou5880 Bird specie 5880 presence (1) or absence (0)
aou6050 Bird specie 6050 presence (1) or absence (0)
aou6130 Bird specie 6130 presence (1) or absence (0)
aou6270 Bird specie 6270 presence (1) or absence (0)
aou6520 Bird specie 6520 presence (1) or absence (0)
aou6810 Bird specie 6810 presence (1) or absence (0)
aou6882 Bird specie 6882 presence (1) or absence (0)
aou7610 Bird specie 7610 presence (1) or absence (0)
aou510 Bird specie 510 presence (1) or absence (0)

aou2210 Bird specie 2210 presence (1) or absence (0)
aou2250 Bird specie 2250 presence (1) or absence (0)
aou2610 Bird specie 2610 presence (1) or absence (0)
aou3131 Bird specie 3131 presence (1) or absence (0)
aou4470 Bird specie 4470 presence (1) or absence (0)
aou4570 Bird specie 4570 presence (1) or absence (0)
aou6120 Bird specie 6120 presence (1) or absence (0)
aou6140 Bird specie 6140 presence (1) or absence (0)
aou7250 Bird specie 7250 presence (1) or absence (0)
aou1350 Bird specie 1350 presence (1) or absence (0)
aou1410 Bird specie 1410 presence (1) or absence (0)
aou1720 Bird specie 1720 presence (1) or absence (0)
aou2630 Bird specie 2630 presence (1) or absence (0)
aou3080 Bird specie 3080 presence (1) or absence (0)
aou3420 Bird specie 3420 presence (1) or absence (0)
aou4123 Bird specie 4123 presence (1) or absence (0)
aou4670 Bird specie 4670 presence (1) or absence (0)
aou5110 Bird specie 5110 presence (1) or absence (0)
aou5460 Bird specie 5460 presence (1) or absence (0)
aou7040 Bird specie 7040 presence (1) or absence (0)
aou7210 Bird specie 7210 presence (1) or absence (0)
aou22860 Bird specie 22860 presence (1) or absence (0)
aou40 Bird specie 40 presence (1) or absence (0)
aou60 Bird specie 60 presence (1) or absence (0)
aou1250 Bird specie 1250 presence (1) or absence (0)
aou1430 Bird specie 1430 presence (1) or absence (0)
aou1490 Bird specie 1490 presence (1) or absence (0)
aou2240 Bird specie 2240 presence (1) or absence (0)
aou2881 Bird specie 2881 presence (1) or absence (0)
aou3750 Bird specie 3750 presence (1) or absence (0)
aou4620 Bird specie 4620 presence (1) or absence (0)
aou1370 Bird specie 1370 presence (1) or absence (0)
aou4200 Bird specie 4200 presence (1) or absence (0)
aou5520 Bird specie 5520 presence (1) or absence (0)
aou6190 Bird specie 6190 presence (1) or absence (0)
aou7050 Bird specie 7050 presence (1) or absence (0)
aou7150 Bird specie 7150 presence (1) or absence (0)

aou1670 Bird specie 1670 presence (1) or absence (0)
aou1940 Bird specie 1940 presence (1) or absence (0)
aou4120 Bird specie 4120 presence (1) or absence (0)
aou5070 Bird specie 5070 presence (1) or absence (0)
aou3480 Bird specie 3480 presence (1) or absence (0)
aou5190 Bird specie 5190 presence (1) or absence (0)
aou5990 Bird specie 5990 presence (1) or absence (0)
aou10 Bird specie 10 presence (1) or absence (0)
aou6220 Bird specie 6220 presence (1) or absence (0)
aou70 Bird specie 70 presence (1) or absence (0)
aou4860 Bird specie 4860 presence (1) or absence (0)
aou6160 Bird specie 6160 presence (1) or absence (0)
aou1470 Bird specie 1470 presence (1) or absence (0)
aou2260 Bird specie 2260 presence (1) or absence (0)
aou640 Bird specie 640 presence (1) or absence (0)
aou5960 Bird specie 5960 presence (1) or absence (0)
aou1390 Bird specie 1390 presence (1) or absence (0)
aou1460 Bird specie 1460 presence (1) or absence (0)
aou3600 Bird specie 3600 presence (1) or absence (0)
aou6460 Bird specie 6460 presence (1) or absence (0)
aou5600 Bird specie 5600 presence (1) or absence (0)
aou7280 Bird specie 7280 presence (1) or absence (0)
aou7350 Bird specie 7350 presence (1) or absence (0)
aou3640 Bird specie 3640 presence (1) or absence (0)
aou3490 Bird specie 3490 presence (1) or absence (0)
aou3940 Bird specie 3940 presence (1) or absence (0)
aou3550 Bird specie 3550 presence (1) or absence (0)
aou3670 Bird specie 3670 presence (1) or absence (0)
aou7580 Bird specie 7580 presence (1) or absence (0)
aou6070 Bird specie 6070 presence (1) or absence (0)
aou7680 Bird specie 7680 presence (1) or absence (0)
aou4130 Bird specie 4130 presence (1) or absence (0)
aou3930 Bird specie 3930 presence (1) or absence (0)
aou5480 Bird specie 5480 presence (1) or absence (0)
aou3520 Bird specie 3520 presence (1) or absence (0)
aou1290 Bird specie 1290 presence (1) or absence (0)
aou6170 Bird specie 6170 presence (1) or absence (0)

aou3570 Bird specie 3570 presence (1) or absence (0)
aou690 Bird specie 690 presence (1) or absence (0)
aou2020 Bird specie 2020 presence (1) or absence (0)
aou770 Bird specie 770 presence (1) or absence (0)
aou3000 Bird specie 3000 presence (1) or absence (0)
aou3900 Bird specie 3900 presence (1) or absence (0)
aou4641 Bird specie 4641 presence (1) or absence (0)
aou4661 Bird specie 4661 presence (1) or absence (0)
aou4690 Bird specie 4690 presence (1) or absence (0)
aou4840 Bird specie 4840 presence (1) or absence (0)
aou5330 Bird specie 5330 presence (1) or absence (0)
aou5540 Bird specie 5540 presence (1) or absence (0)
aou5670 Bird specie 5670 presence (1) or absence (0)
aou5830 Bird specie 5830 presence (1) or absence (0)
aou5850 Bird specie 5850 presence (1) or absence (0)
aou6560 Bird specie 6560 presence (1) or absence (0)
aou6750 Bird specie 6750 presence (1) or absence (0)
aou6800 Bird specie 6800 presence (1) or absence (0)
aou6850 Bird specie 6850 presence (1) or absence (0)
aou7490 Bird specie 7490 presence (1) or absence (0)
aou7630 Bird specie 7630 presence (1) or absence (0)
aou4021 Bird specie 4021 presence (1) or absence (0)
aou4590 Bird specie 4590 presence (1) or absence (0)
aou4660 Bird specie 4660 presence (1) or absence (0)
aou7380 Bird specie 7380 presence (1) or absence (0)
aou7670 Bird specie 7670 presence (1) or absence (0)
aou4910 Bird specie 4910 presence (1) or absence (0)
aou7540 Bird specie 7540 presence (1) or absence (0)
aou4330 Bird specie 4330 presence (1) or absence (0)
aou6240 Bird specie 6240 presence (1) or absence (0)
aou7480 Bird specie 7480 presence (1) or absence (0)
aou7400 Bird specie 7400 presence (1) or absence (0)
aou5210 Bird specie 5210 presence (1) or absence (0)
aou6680 Bird specie 6680 presence (1) or absence (0)
aou6740 Bird specie 6740 presence (1) or absence (0)
aou5671 Bird specie 5671 presence (1) or absence (0)
aou6291 Bird specie 6291 presence (1) or absence (0)

aou6870 Bird specie 6870 presence (1) or absence (0)
aou4940 Bird specie 4940 presence (1) or absence (0)
aou2120 Bird specie 2120 presence (1) or absence (0)
aou7030 Bird specie 7030 presence (1) or absence (0)
aou2060 Bird specie 2060 presence (1) or absence (0)
aou3330 Bird specie 3330 presence (1) or absence (0)
aou1900 Bird specie 1900 presence (1) or absence (0)
aou3250 Bird specie 3250 presence (1) or absence (0)
aou7590 Bird specie 7590 presence (1) or absence (0)
aou4020 Bird specie 4020 presence (1) or absence (0)
aou5580 Bird specie 5580 presence (1) or absence (0)
aou1440 Bird specie 1440 presence (1) or absence (0)
aou4560 Bird specie 4560 presence (1) or absence (0)
aou4770 Bird specie 4770 presence (1) or absence (0)
aou6550 Bird specie 6550 presence (1) or absence (0)
aou5220 Bird specie 5220 presence (1) or absence (0)
aou5950 Bird specie 5950 presence (1) or absence (0)
aou5140 Bird specie 5140 presence (1) or absence (0)
aou7560 Bird specie 7560 presence (1) or absence (0)
aou6290 Bird specie 6290 presence (1) or absence (0)
aou5170 Bird specie 5170 presence (1) or absence (0)
aou7270 Bird specie 7270 presence (1) or absence (0)
aou1310 Bird specie 1310 presence (1) or absence (0)
aou3920 Bird specie 3920 presence (1) or absence (0)
aou4050 Bird specie 4050 presence (1) or absence (0)
aou4680 Bird specie 4680 presence (1) or absence (0)
aou6150 Bird specie 6150 presence (1) or absence (0)
aou4010 Bird specie 4010 presence (1) or absence (0)
aou7221 Bird specie 7221 presence (1) or absence (0)
aou6610 Bird specie 6610 presence (1) or absence (0)
aou5150 Bird specie 5150 presence (1) or absence (0)
aou7260 Bird specie 7260 presence (1) or absence (0)
aou3880 Bird specie 3880 presence (1) or absence (0)
aou3320 Bird specie 3320 presence (1) or absence (0)
aou1500 Bird specie 1500 presence (1) or absence (0)
aou4520 Bird specie 4520 presence (1) or absence (0)
aou6790 Bird specie 6790 presence (1) or absence (0)

aou6110 Bird specie 6110 presence (1) or absence (0)
aou4280 Bird specie 4280 presence (1) or absence (0)
aou5840 Bird specie 5840 presence (1) or absence (0)
aou1810 Bird specie 1810 presence (1) or absence (0)
aou6570 Bird specie 6570 presence (1) or absence (0)
aou7010 Bird specie 7010 presence (1) or absence (0)
aou3430 Bird specie 3430 presence (1) or absence (0)
aou6360 Bird specie 6360 presence (1) or absence (0)
aou4000 Bird specie 4000 presence (1) or absence (0)
aou6860 Bird specie 6860 presence (1) or absence (0)
aou7222 Bird specie 7222 presence (1) or absence (0)
aou6670 Bird specie 6670 presence (1) or absence (0)
aou4630 Bird specie 4630 presence (1) or absence (0)
aou3680 Bird specie 3680 presence (1) or absence (0)
aou550 Bird specie 550 presence (1) or absence (0)
aou4402 Bird specie 4402 presence (1) or absence (0)
aou6450 Bird specie 6450 presence (1) or absence (0)
aou6830 Bird specie 6830 presence (1) or absence (0)
aou4780 Bird specie 4780 presence (1) or absence (0)
aou3780 Bird specie 3780 presence (1) or absence (0)
aou7240 Bird specie 7240 presence (1) or absence (0)
aou1870 Bird specie 1870 presence (1) or absence (0)
aou5080 Bird specie 5080 presence (1) or absence (0)
aou5012 Bird specie 5012 presence (1) or absence (0)
aou4060 Bird specie 4060 presence (1) or absence (0)
aou4640 Bird specie 4640 presence (1) or absence (0)
aou5590 Bird specie 5590 presence (1) or absence (0)
aou6590 Bird specie 6590 presence (1) or absence (0)
aou3100 Bird specie 3100 presence (1) or absence (0)
aou5180 Bird specie 5180 presence (1) or absence (0)
aou7570 Bird specie 7570 presence (1) or absence (0)
aou7660 Bird specie 7660 presence (1) or absence (0)
aou2010 Bird specie 2010 presence (1) or absence (0)
aou2890 Bird specie 2890 presence (1) or absence (0)
aou3870 Bird specie 3870 presence (1) or absence (0)
aou4090 Bird specie 4090 presence (1) or absence (0)
aou4160 Bird specie 4160 presence (1) or absence (0)

aou4230 Bird specie 4230 presence (1) or absence (0)
aou4610 Bird specie 4610 presence (1) or absence (0)
aou4650 Bird specie 4650 presence (1) or absence (0)
aou5010 Bird specie 5010 presence (1) or absence (0)
aou5630 Bird specie 5630 presence (1) or absence (0)
aou5870 Bird specie 5870 presence (1) or absence (0)
aou5930 Bird specie 5930 presence (1) or absence (0)
aou5970 Bird specie 5970 presence (1) or absence (0)
aou5980 Bird specie 5980 presence (1) or absence (0)
aou6040 Bird specie 6040 presence (1) or absence (0)
aou6080 Bird specie 6080 presence (1) or absence (0)
aou6100 Bird specie 6100 presence (1) or absence (0)
aou6280 Bird specie 6280 presence (1) or absence (0)
aou6310 Bird specie 6310 presence (1) or absence (0)
aou6710 Bird specie 6710 presence (1) or absence (0)
aou6730 Bird specie 6730 presence (1) or absence (0)
aou7180 Bird specie 7180 presence (1) or absence (0)
aou7310 Bird specie 7310 presence (1) or absence (0)
aou7360 Bird specie 7360 presence (1) or absence (0)
aou7510 Bird specie 7510 presence (1) or absence (0)
aou7550 Bird specie 7550 presence (1) or absence (0)
aou5060 Bird specie 5060 presence (1) or absence (0)
aou6480 Bird specie 6480 presence (1) or absence (0)
aou6770 Bird specie 6770 presence (1) or absence (0)
aou6840 Bird specie 6840 presence (1) or absence (0)
aou3390 Bird specie 3390 presence (1) or absence (0)
aou6370 Bird specie 6370 presence (1) or absence (0)
aou6630 Bird specie 6630 presence (1) or absence (0)
aou3260 Bird specie 3260 presence (1) or absence (0)
aou6380 Bird specie 6380 presence (1) or absence (0)
aou7290 Bird specie 7290 presence (1) or absence (0)
aou6390 Bird specie 6390 presence (1) or absence (0)
aou1960 Bird specie 1960 presence (1) or absence (0)
aou2030 Bird specie 2030 presence (1) or absence (0)
aou2000 Bird specie 2000 presence (1) or absence (0)
aou4900 Bird specie 4900 presence (1) or absence (0)
aou3730 Bird specie 3730 presence (1) or absence (0)

aou6760 Bird specie 6760 presence (1) or absence (0)
aou6410 Bird specie 6410 presence (1) or absence (0)
aou2001 Bird specie 2001 presence (1) or absence (0)
aou4171 Bird specie 4171 presence (1) or absence (0)
aou6580 Bird specie 6580 presence (1) or absence (0)
aou1970 Bird specie 1970 presence (1) or absence (0)
aou2190 Bird specie 2190 presence (1) or absence (0)
aou3290 Bird specie 3290 presence (1) or absence (0)
aou1840 Bird specie 1840 presence (1) or absence (0)
aou1180 Bird specie 1180 presence (1) or absence (0)
aou5750 Bird specie 5750 presence (1) or absence (0)
aou3200 Bird specie 3200 presence (1) or absence (0)
aou1880 Bird specie 1880 presence (1) or absence (0)
aou4881 Bird specie 4881 presence (1) or absence (0)
aou3270 Bird specie 3270 presence (1) or absence (0)
aou580 Bird specie 580 presence (1) or absence (0)
aou740 Bird specie 740 presence (1) or absence (0)
aou5130 Bird specie 5130 presence (1) or absence (0)
aou1340 Bird specie 1340 presence (1) or absence (0)
aou1990 Bird specie 1990 presence (1) or absence (0)
aou2110 Bird specie 2110 presence (1) or absence (0)
aou3190 Bird specie 3190 presence (1) or absence (0)
aou6010 Bird specie 6010 presence (1) or absence (0)
aou1770 Bird specie 1770 presence (1) or absence (0)
aou5280 Bird specie 5280 presence (1) or absence (0)
aou440 Bird specie 440 presence (1) or absence (0)
aou4030 Bird specie 4030 presence (1) or absence (0)
aou7410 Bird specie 7410 presence (1) or absence (0)
aou2971 Bird specie 2971 presence (1) or absence (0)
aou4240 Bird specie 4240 presence (1) or absence (0)
aou3120 Bird specie 3120 presence (1) or absence (0)
aou4679 Bird specie 4679 presence (1) or absence (0)
aou4080 Bird specie 4080 presence (1) or absence (0)
aou4540 Bird specie 4540 presence (1) or absence (0)
aou4691 Bird specie 4691 presence (1) or absence (0)
aou4920 Bird specie 4920 presence (1) or absence (0)
aou5040 Bird specie 5040 presence (1) or absence (0)

aou5730 Bird specie 5730 presence (1) or absence (0)
aou5738 Bird specie 5738 presence (1) or absence (0)
aou6292 Bird specie 6292 presence (1) or absence (0)
aou6340 Bird specie 6340 presence (1) or absence (0)
aou6650 Bird specie 6650 presence (1) or absence (0)
aou7190 Bird specie 7190 presence (1) or absence (0)
aou7331 Bird specie 7331 presence (1) or absence (0)
aou7430 Bird specie 7430 presence (1) or absence (0)
aou4320 Bird specie 4320 presence (1) or absence (0)
aou4480 Bird specie 4480 presence (1) or absence (0)
aou4810 Bird specie 4810 presence (1) or absence (0)
aou7020 Bird specie 7020 presence (1) or absence (0)
aou7070 Bird specie 7070 presence (1) or absence (0)
aou4180 Bird specie 4180 presence (1) or absence (0)
aou3960 Bird specie 3960 presence (1) or absence (0)
aou4250 Bird specie 4250 presence (1) or absence (0)
aou5900 Bird specie 5900 presence (1) or absence (0)
aou5650 Bird specie 5650 presence (1) or absence (0)
aou5300 Bird specie 5300 presence (1) or absence (0)
aou5690 Bird specie 5690 presence (1) or absence (0)
aou6090 Bird specie 6090 presence (1) or absence (0)
aou6640 Bird specie 6640 presence (1) or absence (0)
aou7300 Bird specie 7300 presence (1) or absence (0)
aou4070 Bird specie 4070 presence (1) or absence (0)
aou4040 Bird specie 4040 presence (1) or absence (0)
aou6440 Bird specie 6440 presence (1) or absence (0)
aou2950 Bird specie 2950 presence (1) or absence (0)
aou3850 Bird specie 3850 presence (1) or absence (0)
aou5120 Bird specie 5120 presence (1) or absence (0)
aou5800 Bird specie 5800 presence (1) or absence (0)
aou6200 Bird specie 6200 presence (1) or absence (0)
aou7130 Bird specie 7130 presence (1) or absence (0)
aou7460 Bird specie 7460 presence (1) or absence (0)
aou7520 Bird specie 7520 presence (1) or absence (0)
aou4210 Bird specie 4210 presence (1) or absence (0)
aou4290 Bird specie 4290 presence (1) or absence (0)
aou4530 Bird specie 4530 presence (1) or absence (0)

aou3210 Bird specie 3210 presence (1) or absence (0)
aou4310 Bird specie 4310 presence (1) or absence (0)
aou7170 Bird specie 7170 presence (1) or absence (0)
aou4110 Bird specie 4110 presence (1) or absence (0)
aou5910 Bird specie 5910 presence (1) or absence (0)
aou6430 Bird specie 6430 presence (1) or absence (0)
aou3350 Bird specie 3350 presence (1) or absence (0)
aou5050 Bird specie 5050 presence (1) or absence (0)
aou4710 Bird specie 4710 presence (1) or absence (0)
aou6320 Bird specie 6320 presence (1) or absence (0)
aou6330 Bird specie 6330 presence (1) or absence (0)
aou4960 Bird specie 4960 presence (1) or absence (0)
aou5940 Bird specie 5940 presence (1) or absence (0)
aou4580 Bird specie 4580 presence (1) or absence (0)
aou4870 Bird specie 4870 presence (1) or absence (0)
aou2930 Bird specie 2930 presence (1) or absence (0)
aou5780 Bird specie 5780 presence (1) or absence (0)
aou2882 Bird specie 2882 presence (1) or absence (0)
aou3620 Bird specie 3620 presence (1) or absence (0)
aou4430 Bird specie 4430 presence (1) or absence (0)
aou2940 Bird specie 2940 presence (1) or absence (0)
aou4360 Bird specie 4360 presence (1) or absence (0)
aou2920 Bird specie 2920 presence (1) or absence (0)
aou7420 Bird specie 7420 presence (1) or absence (0)
aou3970 Bird specie 3970 presence (1) or absence (0)
aou7330 Bird specie 7330 presence (1) or absence (0)
aou3990 Bird specie 3990 presence (1) or absence (0)
aou6690 Bird specie 6690 presence (1) or absence (0)
aou5911 Bird specie 5911 presence (1) or absence (0)
aou4760 Bird specie 4760 presence (1) or absence (0)
aou7100 Bird specie 7100 presence (1) or absence (0)
aou3090 Bird specie 3090 presence (1) or absence (0)
aou3050 Bird specie 3050 presence (1) or absence (0)
aou6620 Bird specie 6620 presence (1) or absence (0)
aou6540 Bird specie 6540 presence (1) or absence (0)
aou1830 Bird specie 1830 presence (1) or absence (0)
aou5470 Bird specie 5470 presence (1) or absence (0)

aou6883 Bird specie 6883 presence (1) or absence (0)
 aou6420 Bird specie 6420 presence (1) or absence (0)
 aou6121 Bird specie 6121 presence (1) or absence (0)
 aou4100 Bird specie 4100 presence (1) or absence (0)
 aou7320 Bird specie 7320 presence (1) or absence (0)
 aou4461 Bird specie 4461 presence (1) or absence (0)
 aou5860 Bird specie 5860 presence (1) or absence (0)

Source

Pardieck, K.L., D.J. Ziolkowski Jr., M. Lutmerding, K. Campbell and M.-A.R. Hudson. 2017. North American Breeding Bird Survey Dataset 1966 - 2016, version 2016.0. U.S. Geological Survey, Patuxent Wildlife Research Center. <https://www.pwrc.usgs.gov/bbs/RawData/doi:10.5066/F7W0944J>.

Examples

```
data(birds)
```

complaints	<i>Complaints received for the **Bureau of Consumer Financial Protection** in US about financial products and services.</i>
------------	---

Description

Specifically in this dataset we work with only credit card complaint's for the 2015 year.

Usage

```
data("complaints")
```

Format

A data frame with 17301 observations on the following 3 variables.

Product a factor with levels Credit card

Issue a factor with levels Advertising and marketing Application processing delay
 APR or interest rate Arbitration Balance transfer Balance transfer fee Bankruptcy
 Billing disputes Billing statement Cash advance Cash advance fee Closing/Cancelling account
 Convenience checks Credit card protection / Debt protection Credit determination
 Credit line increase/decrease Customer service / Customer relations Delinquent account
 Forbearance / Workout plans Identity theft / Fraud / Embezzlement Late fee
 Other Other fee Overlimit fee Payoff process Privacy Rewards Sale of account
 Transaction issue Unsolicited issuance of credit card

Company a factor variable describing the companies available

Examples

```
data(complaints)
```

fishnet	<i>Latitude and Longitude Fishnet dataset.</i>
---------	--

Description

This dataset is a subset of Fishnet.

Usage

```
data("fishnet")
```

Format

A data frame with 4455 observations and 2 variables.

POINT_X Longitude

POINT_Y Latitude

Examples

```
data(fishnet)
```

generateBernoulliLDA	<i>Simulates a Bernoulli LDA.</i>
----------------------	-----------------------------------

Description

Simulates a Bernoulli LDA.

Usage

```
## S3 method for class 'rlda'
generateBernoulliLDA(seed0, community, variables,
  observations, alpha0, alpha1, gamma, ...)
```

Arguments

seed0	Initial seed to simulate a Bernoulli LDA.
community	Total number of latent clusters. Must be greater than 2.
variables	Total number of variables. Must be greater than the number of communities.
observations	Total number of observations. Must be greater than 1.
alpha0	Scalar hyperparameters that must be positive.
alpha1	Scalar hyperparameters that must be positive.
gamma	Scalar hyperparameters that must be positive.
...	other arguments may be useful.

Details

Generates a list with the simulated Theta and Phi matrix of parameters, Z latent matrix of communities and and Data matrix for the Bernoulli LDA.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>

See Also

[generateMultinomialLDA](#), [generateBinomialLDA](#)

Examples

```
## Not run:
#Generate fake data
res<- generateBernoulliLDA.rlda(seed0=9292, community=3,
                                variables=100, observations=1000,
                                alpha0=0.01, alpha1=0.01, gamma=0.01)

#Show results
res

## End(Not run)
```

generateBinomialLDA *Simulates a Binomial LDA.*

Description

Simulates a Binomial LDA.

Usage

```
## S3 method for class 'rlda'
generateBinomialLDA(seed0, community, variables,
                    observations, totalElements, alpha0, alpha1, ...)
```

Arguments

seed0	Initial seed to simulate a Bernoulli LDA.
community	Total number of latent clusters. Must be greater than 2.
variables	Total number of variables. Must be greater than the number of communities.
observations	Total number of observations. Must be greater than 1.
totalElements	Total expected number of elements for each observation. Must be greater than 1.
alpha0	Scalar hyperparameters that must be positive.
alpha1	Scalar hyperparameters that must be positive.
...	other arguments may be useful.

Details

Generates a list with the simulated Theta and Phi matrix of parameters, Population and and Data matrix for the Binomial LDA.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>

See Also

[generateMultinomialLDA](#), [generateBernoulliLDA](#)

Examples

```
## Not run:
#Generate fake data
res<- generateBinomialLDA.rlda(seed0=9292, community=3,
                              variables=100, observations=1000,
                              alpha0=0.01, alpha1=0.01, gamma=0.01)

#Show results
res

## End(Not run)
```

`generateMultinomialLDA`*Simulates a Multinomial LDA.*

Description

Simulates a Multinomial LDA.

Usage

```
## S3 method for class 'rlda'  
generateMultinomialLDA(seed0, community, variables,  
  observations, totalElements, beta, gamma, ...)
```

Arguments

<code>seed0</code>	Initial seed to simulate a Multinomial LDA.
<code>community</code>	Total number of latent clusters. Must be greater than 2.
<code>variables</code>	Total number of variables. Must be greater than the number of communities.
<code>observations</code>	Total number of observations. Must be greater than 1.
<code>totalElements</code>	Total expected number of elements for each observation. Must be greater than 1.
<code>beta</code>	Vector of positive hyperparameters with dimension equal variables.
<code>gamma</code>	Scalar hyperparameters that must be positive.
<code>...</code>	other arguments may be useful.

Details

Generates a list with the simulated Theta and Phi matrix of parameters, Z latent matrix of communities and and Data matrix for the Multinomial LDA.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>

See Also

[generateBernoulliLDA](#), [generateBinomialLDA](#)

Examples

```
## Not run:
#Generate fake data
res<- generateMultinomialLDA.rlda(seed0=5587, community=4,
    variables=100, observations=1000, totalElements=40,
    beta=rep(1,100), gamma=0.01)
#Show results
res

## End(Not run)
```

getPhi	<i>Provide Phi information.</i>
--------	---------------------------------

Description

Takes a rlda object produced by rlda.binomial, rlda.bernoulli or rlda.multinomial and obtain a Phi estimate from it.

Usage

```
## S3 method for class 'rlda'
getPhi(object, burnin=0.1, ...)
```

Arguments

object	a rlda object as produced by rlda.binomial, rlda.bernoulli or rlda.multinomial.
burnin	a percentual of burn-in observations must be a number between 0 and 1. The default value is burnin=0.1
...	other arguments may be useful.

Details

Get the Phi estimates.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>

See Also

[rlda.binomial](#), [rlda.bernoulli](#), [rlda.multinomial](#)

Examples

```
## Not run:
#Load data
data(presence)
#Set seed
set.seed(9842)
#Hyperparameters for each prior distribution
gamma <-0.01
alpha0<-0.01
alpha1<-0.01
#Execute the LDA for the Binomial entry
res<-rlda.bernoulli(presence, 10, alpha0, alpha1, gamma,
                   5000, TRUE, FALSE)

#getPhi accessor
getPhi(res)

## End(Not run)
```

getTheta

Provide Theta information.

Description

Takes a rlda object produced by `rlda.binomial`, `rlda.bernoulli` or `rlda.multinomial` and obtain a Theta estimate from it.

Usage

```
## S3 method for class 'rlda'
getTheta(object, burnin=0.1, ...)
```

Arguments

object	a rlda object as produced by <code>rlda.binomial</code> , <code>rlda.bernoulli</code> or <code>rlda.multinomial</code> .
burnin	a percentual of burn-in observations must be a number between 0 and 1. The default value is <code>burnin=0.1</code>
...	other arguments may be useful.

Details

Get the Theta estimates.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>

See Also

[rlda.binomial](#), [rlda.bernoulli](#), [rlda.multinomial](#)

Examples

```
## Not run:  
#Load data  
data(presence)  
#Set seed  
set.seed(9842)  
#Hyperparameters for each prior distribution  
gamma <-0.01  
alpha0<-0.01  
alpha1<-0.01  
#Execute the LDA for the Binomial entry  
res<-rlda.bernoulli(presence, 10, alpha0, alpha1, gamma,  
                    5000, TRUE, FALSE)  
  
#getTheta accessor  
getTheta(res)  
  
## End(Not run)
```

Landsat

Landsat TM 5 imagery from 2010 of the Iquitos-Nauta road in the Peruvian Amazon

Description

This data set has Binomial data from Landsat TM 5 imagery from 2010 of the Iquitos-Nauta road in the Peruvian Amazon for 7 bands at 69540 locations.

Usage

```
data(Landsat)
```

Format

A data frame with 69540 observations for 9 columns.

Source

This dataset is from: Valle D, Baiser B, Woodall CW, Chazdon R (2014). "Decomposing biodiversity data using the Latent Dirichlet Allocation model, a probabilistic multivariate statistical method." Ecology letters, 17(12), 1591-1601.

Examples

```
data(Landsat)
```

LocationsBirds	<i>ID variable Latitude and Longitude Locations for Birds dataset.</i>
----------------	--

Description

This dataset is a subset of Birds dataset with ID variable.

Usage

```
data("LocationsBirds")
```

Format

A data frame with 3080 observations and 3 variables.

loc.id Location ID

Latitude Latitude

Longitude Longitude

Examples

```
data(LocationsBirds)
```

logLik *Provide the log-likelihood for the rlda object.*

Description

Takes a rlda object produced and provides the log-likelihood.

Usage

```
## S3 method for class 'rlda'  
logLik(object, ...)
```

Arguments

object a rlda object
... other arguments may be useful.

Details

Get the log-likelihood of the model.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>

See Also

[rlda.binomial](#), [rlda.bernoulli](#), [rlda.multinomial](#)

Examples

```
## Not run:  
library(Rlda)  
#Load data  
data(presence)  
#Set seed  
set.seed(9842)  
#Hyperparameters for each prior distribution  
gamma <-0.01  
alpha0<-0.01
```

```
alpha1<-0.01
#Execute the LDA for the Binomial entry
res<-rlda.bernoulli(presence, 10, alpha0, alpha1, gamma,
                   5000, TRUE, FALSE)

#log-likelihood
logLik(res)

## End(Not run)
```

plot *plot method for rlda object*

Description

Plot a rlda object. The plot function returns three plots based on Theta matrix, Phi matrix and log-likelihood.

Usage

```
## S3 method for class 'rlda'
plot(x, burnin=0.1, maxCluster=NA, ...)
```

Arguments

x	a rlda object created by <code>rlda.binomial</code> , <code>rlda.bernoulli</code> or <code>rlda.multinomial</code> function.
burnin	a percentual of burn-in observations must be a number between 0 and 1. The default value is <code>burnin=0.1</code> .
maxCluster	The maximum number of cluster to be shown. The default value is <code>maxCluster=NA</code> which represents all clusters must be shown.
...	other arguments may be useful.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>

See Also

[rlda.binomial](#), [rlda.bernoulli](#), [rlda.multinomial](#)

Examples

```
## Not run:
#Load data
data(presence)
#Set seed
set.seed(9842)
#Hyperparameters for each prior distribution
gamma <-0.01
alpha0<-0.01
alpha1<-0.01
#Execute the LDA for the Binomial entry
res<-rlda.bernoulli(presence, 10, alpha0, alpha1, gamma,
                   5000, TRUE, FALSE)

#Plot the results
plot(res)

## End(Not run)
```

predict

Provide predictions to the Binomial entry.

Description

Takes a rlda object produced by rlda.binomial and produces a prediction from it.

Usage

```
## S3 method for class 'rlda'
predict(object, data, nclus=NA, burnin=0.1, places.round=0, ...)
```

Arguments

object	a rlda object as produced by rlda.binomial
data	Dataset used to make the predictions. Must have the same number of columns as the dataset used in the rlda.binomial.
nclus	Number of clusters to be used in the prediction. The default value is nclus=NA
burnin	a percentual of burn-in observations must be a number between 0 and 1. The default value is burnin=0.1
places.round	Number decimal places to be rounded. The default value is places.round=0
...	other arguments may be useful.

Details

Predicts the Gibbs Sampling results and arguments.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>

See Also

[rlda.binomial](#), [rlda.bernoulli](#), [rlda.multinomial](#)

Examples

```
## Not run:
library(Rlda)
# Read the SP500 data
data(sp500)
# Create size
spSize <- as.data.frame(matrix(100,
  ncol = ncol(sp500),
  nrow = nrow(sp500)))
# Set seed
set.seed(5874)
# Hyperparameters for each prior distribution
gamma <- 0.01
alpha0 <- 0.01
alpha1 <- 0.01
# Execute the LDA for the Binomial entry
res <- rlda.binomial(data = sp500, pop = spSize, n_community = 10,
  alpha0 = alpha0, alpha1 = alpha1, gamma = gamma,
  n_gibbs = 500, ll_prior = TRUE, display_progress = TRUE)
#Predict
pred<- predict(res, sp500, nclus=3)

## End(Not run)
```

presence

Species Presence/Absence Data

Description

This data set has Presence/Absence predictions for 13 species at 386 forested locations. It consists of species, observed presence-absence values, and the probability predictions of three different models.

Usage

```
data(presence)
```

Format

A data frame with 386 observations for 13 species. Each cell represents one when the specie is presented zero otherwise.

Source

This dataset is from: Moisen, G.G., Freeman, E.A., Blackard, J.A., Frescino, T.S., Zimmerman N.E., Edwards, T.C. Predicting tree species presence and basal area in Utah: A comparison of stochastic gradient boosting, generalized additive models, and tree-based methods. Ecological Modelling, 199 (2006) 176-187.

Examples

```
data(presence)
```

```
print
```

Print information with respect to the model.

Description

Takes a rlda object produced by `rlda.binomial`, `rlda.bernoulli` or `rlda.multinomial` and produces a print from it.

Usage

```
## S3 method for class 'rlda'
print(x, burnin=0.1, ...)
```

Arguments

<code>x</code>	a rlda object as produced by <code>rlda.binomial</code> , <code>rlda.bernoulli</code> or <code>rlda.multinomial</code> .
<code>burnin</code>	a percentual of burn-in observations must be a number between 0 and 1. The default value is <code>burnin=0.1</code>
<code>...</code>	other arguments may be useful.

Details

Print the Gibbs Sampling results.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>

See Also

[rlda.binomial](#), [rlda.bernoulli](#), [rlda.multinomial](#)

Examples

```
## Not run:  
#Load data  
data(presence)  
#Set seed  
set.seed(9842)  
#Hyperparameters for each prior distribution  
gamma <-0.01  
alpha0<-0.01  
alpha1<-0.01  
#Execute the LDA for the Binomial entry  
res<-rlda.bernoulli(presence, 10, alpha0, alpha1, gamma,  
                    5000, TRUE, FALSE)  
  
#print method  
print(res)  
  
## End(Not run)
```

rlda.bernoulli

LDA with bernoulli entry and Stick-Breaking prior.

Description

This method implements the Latent Dirichlet Allocation with Stick-Breaking prior for bernoulli data. `rlda.bernoulli` works with binary data.frame.

Usage

```
rlda.bernoulli(data, n_community, alpha0, alpha1, gamma,  
n_gibbs, ll_prior = TRUE, display_progress = TRUE)
```

Arguments

<code>data</code>	A binary data.frame where each row is a sampling unit (i.e. Plots, Locations, Time, etc.) and each column is a categorical type of element (i.e. Species, Firms, Issues, etc.). The elements inside this data.frame must be Zeros and Ones.
<code>n_community</code>	Total number of communities to return. It must be less than the total number of columns inside the data data.frame.
<code>alpha0</code>	Hyperparameter associated with the Beta prior Beta(alpha0, alpha1).
<code>alpha1</code>	Hyperparameter associated with the Beta prior Beta(alpha0, alpha1).
<code>gamma</code>	Hyperparameter associated with the Stick-Breaking prior.
<code>n_gibbs</code>	Total number of Gibbs Samples.
<code>ll_prior</code>	boolean scalar indicating TRUE if the log-likelihood must be computed using also the priors or FALSE otherwise.
<code>display_progress</code>	boolean scalar TRUE if the Progress Bar must be showed and FALSE otherwise.

Details

`rlda.bernoulli` uses a modified Latent Dirichlet Allocation method to construct Mixed-Membership Clusters using Bayesian Inference. The data must be a non-empty data.frame with the binaries values Zero or Ones for each variable (column) in each observation (row).

Value

A R List with three elements:

<code>Theta</code>	The individual probability for each observation (ex: location) belong in each cluster (ex: community). It is a matrix with dimension equal <code>n_gibbs</code> by <code>nrow(data) * n_community</code>
<code>Phi</code>	The individual probability for each variable (ex: Specie) belong in each cluster (ex: community). It is a matrix with dimension equal <code>n_gibbs</code> by <code>ncol(data) * n_community</code>
<code>LogLikelihood</code>	The vector of Log-Likelihoods compute for each Gibbs Sample.

Note

The Theta and Phi matrix can be obtained for the i-th gibbs sampling using `matrix(Theta[i,], nrow = nrow(data), ncol = ncol(data))` and `matrix(Phi[i,], nrow = n_community, ncol = ncol(data))`, respectively.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>
<http://daijiang.name/>

References

- Blei, David M., Andrew Y. Ng, and Michael I. Jordan.
"Latent dirichlet allocation." *Journal of machine Learning research* 3.Jan (2003): 993-1022.
<http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>
- Valle, Denis, et al.
"Decomposing biodiversity data using the Latent Dirichlet Allocation model, a probabilistic multivariate statistical method." *Ecology letters* 17.12 (2014): 1591-1601.

See Also

[rlda.multinomial](#), [rlda.binomial](#)

Examples

```
## Not run:  
library(Rlda)  
# Presence  
data(presence)  
# Set seed  
set.seed(9842)  
# Hyperparameters for each prior distribution  
gamma <- 0.01  
alpha0 <- 0.01  
alpha1 <- 0.01  
# Execute the LDA for the Bernoulli entry  
res <- rlda.bernoulli(data = presence, n_community = 10,  
alpha0 = alpha0, alpha1 = alpha1, gamma = gamma,  
n_gibbs = 5000, ll_prior = TRUE, display_progress = TRUE)  
  
## End(Not run)
```

rlda.bernoulliMH *LDA with bernoulli entry with Metropolis-Hasting.*

Description

This method implements the Latent Dirichlet Allocation with Stick-Breaking prior for bernoulli data. `rlda.bernoulliMH` works with binary `data.frame`.

Usage

```
rlda.bernoulliMH(data, loc.id, n_community, alpha0, alpha1, gamma,  
n_gibbs, nadapt, ll_prior = TRUE, display_progress = TRUE)
```

Arguments

<code>data</code>	A binary data.frame where each row is a sampling unit (i.e. Plots, Locations, Time, etc.) and each column is a categorical type of element (i.e. Species, Firms, Issues, etc.). The elements inside this data.frame must be Zeros and Ones.
<code>loc.id</code>	Vector column from data with the repeated locations for Presence and Absence data.
<code>n_community</code>	Total number of communities to return. It must be less than the total number of columns inside the data data.frame.
<code>alpha0</code>	Hyperparameter associated with the Beta prior Beta(alpha0, alpha1).
<code>alpha1</code>	Hyperparameter associated with the Beta prior Beta(alpha0, alpha1).
<code>gamma</code>	Hyperparameter associated with the Stick-Breaking prior.
<code>n_gibbs</code>	Total number of Gibbs Samples.
<code>nadapt</code>	Total number of Adaptations.
<code>ll_prior</code>	boolean scalar indicating TRUE if the log-likelihood must be computed using also the priors or FALSE otherwise.
<code>display_progress</code>	boolean scalar TRUE if the Progress Bar must be showed and FALSE otherwise.

Details

`rlda.bernoulliMH` uses a modified Latent Dirichlet Allocation method to construct Mixed-Membership Clusters using Bayesian Inference. The data must be a non-empty data.frame with the binaries values Zero or Ones for each variable (column) in each observation (row).

Value

A R List with three elements:

<code>Theta</code>	The individual probability for each observation (ex: location) belong in each cluster (ex: community). It is a matrix with dimension equal <code>n_gibbs</code> by <code>length(unique(loc.id)) * n_community</code>
<code>Phi</code>	The individual probability for each variable (ex: Specie) belong in each cluster (ex: community). It is a matrix with dimension equal <code>n_gibbs</code> by <code>ncol(data) * n_community</code>
<code>LogLikelihood</code>	The vector of Log-Likelihoods compute for each Gibbs Sample.

Note

The `Theta` and `Phi` matrix can be obtained for the `i`-th gibbs sampling using `matrix(Theta[i,], nrow = length(unique(loc.id)), ncol = length(unique(species)), byrow = TRUE)` and `matrix(Phi[i,], nrow = n_community, ncol = ncol(data)), respectively.`

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>

- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>
<http://daijiang.name/>

References

- Blei, David M., Andrew Y. Ng, and Michael I. Jordan.
"Latent dirichlet allocation." *Journal of machine Learning research* 3.Jan (2003): 993-1022.
<http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>
- Valle, Denis, et al.
"Decomposing biodiversity data using the Latent Dirichlet Allocation model, a probabilistic multivariate statistical method." *Ecology letters* 17.12 (2014): 1591-1601.

See Also

[rlda.multinomial](#), [rlda.binomial](#)

Examples

```
## Not run:
library(Rlda)
# Presence
data(presence)
# Set seed
set.seed(9842)
# Hyperparameters for each prior distribution
gamma <- 0.01
alpha0 <- 0.01
alpha1 <- 0.01
# Execute the LDA for the Bernoulli entry
res <- rlda.bernoulliMH(data=presence,loc.id=seq(1,nrow(presence)),
  n_community=5, alpha0=0.01, alpha1=0.99, gamma=0.1,
  n_gibbs=1000, nadapt=1000, ll_prior = TRUE, display_progress = TRUE)

## End(Not run)
```

rlda.binomial

LDA with binomial entry and Stick-Breaking prior.

Description

This method implements the Latent Dirichlet Allocation with Stick-Breaking prior for binomial data. `rlda.binomial` works with frequency data.frame and also a population data.frame.

Usage

```
rlda.binomial(data, pop, n_community, alpha0, alpha1, gamma,
              n_gibbs, ll_prior = TRUE, display_progress = TRUE)
```

Arguments

<code>data</code>	A abundance data.frame where each row is a sampling unit (i.e. Plots, Locations, Time, etc.) and each column is a categorical type of element (i.e. Species, Firms, Issues, etc.).
<code>pop</code>	A population data.frame where each row is a sampling unit (i.e. Plots, Locations, Time, etc.) and each column is a categorical type of element (i.e. Species, Firms, Issues, etc.). The elements inside this data.frame must all be greater than the elements inside the data data.frame.
<code>n_community</code>	Total number of communities to return. It must be less than the total number of columns inside the data and pop data.frame.
<code>alpha0</code>	Hyperparameter associated with the Beta prior Beta(alpha0, alpha1).
<code>alpha1</code>	Hyperparameter associated with the Beta prior Beta(alpha0, alpha1).
<code>gamma</code>	Hyperparameter associated with the Stick-Breaking prior.
<code>n_gibbs</code>	Total number of Gibbs Samples.
<code>ll_prior</code>	boolean scalar, TRUE if the log-likelihood must be computed using also the priors or FALSE otherwise.
<code>display_progress</code>	boolean scalar, TRUE if the Progress Bar must be showed and FALSE otherwise.

Details

`rlda.binomial` uses a modified Latent Dirichlet Allocation method to construct Mixed-Membership Clusters using Bayesian Inference. The data must be a non-empty data.frame with the frequencies for each variable (column) in each observation (row). The pop must be a non-empty data.frame with the frequencies for each variable (column) in each observation (row) greater than the entries inside data data.frame.

Value

A R List with three elements:

<code>Theta</code>	The individual probability for each observation (ex: location) belong in each cluster (ex: community). It is a matrix with dimension equal <code>n_gibbs</code> by <code>nrow(data) * n_community</code>
<code>Phi</code>	The individual probability for each variable (ex: Specie) belong in each cluster (ex: community). It is a matrix with dimension equal <code>n_gibbs</code> by <code>ncol(data) * n_community</code>
<code>LogLikelihood</code>	The vector of Log-Likelihoods compute for each Gibbs Sample.

Note

The Theta and Phi matrix can be obtained for the i-th gibbs sampling using `matrix(Theta[i,], nrow = nrow(data), ncol = ncol(data))` and `matrix(Phi[i,], nrow = n_community, ncol = ncol(data))`, respectively.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>
<http://daijiang.name>

References

- Blei, David M., Andrew Y. Ng, and Michael I. Jordan.
"Latent dirichlet allocation." *Journal of machine Learning research* 3.Jan (2003): 993-1022.
<http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>
- Valle, Denis, et al.
"Decomposing biodiversity data using the Latent Dirichlet Allocation model, a probabilistic multivariate statistical method." *Ecology letters* 17.12 (2014): 1591-1601.

See Also

[rlda.multinomial](#), [rlda.bernoulli](#)

Examples

```
## Not run:
library(Rlda)
# Read the SP500 data
data(sp500)
# Create size
spSize <- as.data.frame(matrix(100,
  ncol = ncol(sp500),
  nrow = nrow(sp500)))
# Set seed
set.seed(5874)
# Hyperparameters for each prior distribution
gamma <- 0.01
alpha0 <- 0.01
alpha1 <- 0.01
# Execute the LDA for the Binomial entry
res <- rlda.binomial(data = sp500, pop = spSize, n_community = 10,
  alpha0 = alpha0, alpha1 = alpha1, gamma = gamma,
  n_gibbs = 500, ll_prior = TRUE, display_progress = TRUE)

## End(Not run)
```

rlda.binomialMH *LDA with binomial with Metropolis-Hasting.*

Description

This method implements the Latent Dirichlet Allocation with Stick-Breaking prior for binomial data and Remote Sensing. `rlda.binomialMH` works with frequency `data.frame` and also a population `data.frame`.

Usage

```
rlda.binomialMH(data, pop, n_community, alpha0, alpha1, gamma,
n_gibbs, ll_prior = TRUE, display_progress = TRUE)
```

Arguments

<code>data</code>	A abundance <code>data.frame</code> where each row is a sampling unit (i.e. Plots, Locations, Time, etc.) and each column is a categorical type of element (i.e. Species, Firms, Issues, etc.).
<code>pop</code>	A population <code>data.frame</code> where each row is a sampling unit (i.e. Plots, Locations, Time, etc.) and each column is a categorical type of element (i.e. Species, Firms, Issues, etc.). The elements inside this <code>data.frame</code> must all be greater than the elements inside the <code>data</code> <code>data.frame</code> .
<code>n_community</code>	Total number of communities to return. It must be less than the total number of columns inside the <code>data</code> and <code>pop</code> <code>data.frame</code> .
<code>alpha0</code>	Hyperparameter associated with the Beta prior $\text{Beta}(\alpha_0, \alpha_1)$.
<code>alpha1</code>	Hyperparameter associated with the Beta prior $\text{Beta}(\alpha_0, \alpha_1)$.
<code>gamma</code>	Hyperparameter associated with the Stick-Breaking prior.
<code>n_gibbs</code>	Total number of Gibbs Samples.
<code>ll_prior</code>	boolean scalar, TRUE if the log-likelihood must be computed using also the priors or FALSE otherwise.
<code>display_progress</code>	boolean scalar, TRUE if the Progress Bar must be showed and FALSE otherwise.

Details

`rlda.binomialMH` uses a modified Latent Dirichlet Allocation method to construct Mixed-Membership Clusters using Bayesian Inference. The `data` must be a non-empty `data.frame` with the frequencies for each variable (column) in each observation (row). The `pop` must be a non-empty `data.frame` with the frequencies for each variable (column) in each observation (row) greater than the entries inside `data` `data.frame`.

Value

A R List with three elements:

Theta	The individual probability for each observation (ex: location) belong in each cluster (ex: community). It is a matrix with dimension equal n_gibbs by $nrow(data) * n_community$
Phi	The individual probability for each variable (ex: Specie) belong in each cluster (ex: community). It is a matrix with dimension equal n_gibbs by $ncol(data) * n_community$
LogLikelihood	The vector of Log-Likelihoods compute for each Gibbs Sample.

Note

The Theta and Phi matrix can be obtained for the i-th gibbs sampling using `matrix(Theta[i,], nrow = nrow(data), ncol = ncol(data))` and `matrix(Phi[i,], nrow = n_community, ncol = ncol(data))`, respectively.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>
<http://daijiang.name>

References

- Blei, David M., Andrew Y. Ng, and Michael I. Jordan.
"Latent dirichlet allocation." *Journal of machine Learning research* 3.Jan (2003): 993-1022.
<http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>
- Valle, Denis, et al.
"Decomposing biodiversity data using the Latent Dirichlet Allocation model, a probabilistic multivariate statistical method." *Ecology letters* 17.12 (2014): 1591-1601.

See Also

[rlda.multinomial](#), [rlda.bernoulli](#)

Examples

```
## Not run:
library(Rlda)
# Read the SP500 data
data(sp500)
# Create size
spSize <- as.data.frame(matrix(100,
```

```

    ncol = ncol(sp500),
    nrow = nrow(sp500)))
# Set seed
set.seed(5874)
# Hyperparameters for each prior distribution
gamma <- 0.01
alpha0 <- 0.01
alpha1 <- 0.01
# Execute the LDA for the Binomial entry
res <- rlda.binomialMH(data = sp500, pop = spSize, n_community = 10,
alpha0 = alpha0, alpha1 = alpha1, gamma = gamma,
n_gibbs = 500, ll_prior = TRUE, display_progress = TRUE)

## End(Not run)

```

rlda.binomialVB	<i>LDA with binomial entry and Stick-Breaking prior and Variational Inference.</i>
-----------------	--

Description

This method implements the Latent Dirichlet Allocation with Stick-Breaking prior for binomial data with Variational Inference. `rlda.binomialVB` works with frequency data.frame.

Usage

```
rlda.binomialVB(data, loc.id, n_community, alpha0, alpha1,
gamma, maxit=1000, thresh=0.0001)
```

Arguments

<code>data</code>	A abundance data.frame where each row is a sampling unit (i.e. Plots, Locations, Time, etc.) and each column is a categorical type of element (i.e. Species, Firms, Issues, etc.).
<code>loc.id</code>	ID variable in data.frame.
<code>n_community</code>	Total number of communities to return. It must be less than the total number of columns inside the data.
<code>alpha0</code>	Hyperparameter associated with the Beta prior $\text{Beta}(\alpha_0, \alpha_1)$.
<code>alpha1</code>	Hyperparameter associated with the Beta prior $\text{Beta}(\alpha_0, \alpha_1)$.
<code>gamma</code>	Hyperparameter associated with the Stick-Breaking prior.
<code>maxit</code>	Maximum number of iterations.
<code>thresh</code>	Threshold for ELBO convergence.

Details

`rlda.binomialVB` uses a modified Latent Dirichlet Allocation method to construct Mixed-Membership Clusters using Bayesian Variational Inference. The data must be a non-empty data.frame with the frequencies for each variable (column) in each observation (row).

Value

A R List with three elements:

Theta	The individual probability for each observation (ex: location) belong in each cluster (ex: community). It is a matrix with dimension equal <code>n_gibbs</code> by <code>nrow(data) * n_community</code>
Phi	The individual probability for each variable (ex: Specie) belong in each cluster (ex: community). It is a matrix with dimension equal <code>n_gibbs</code> by <code>ncol(data) * n_community</code>
ELBO	The vector of ELBO compute for each iteration.

Note

The Theta and Phi matrix can be obtained for the i-th gibbs sampling using `matrix(Theta[i,], nrow = nrow(data), ncol = ncol(data))` and `matrix(Phi[i,], nrow = n_community, ncol = ncol(data))`, respectively.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>
<http://daijiang.name>

References

- Blei, David M., Andrew Y. Ng, and Michael I. Jordan.
"Latent dirichlet allocation." *Journal of machine Learning research* 3.Jan (2003): 993-1022.
<http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>
- Valle, Denis, et al.
"Decomposing biodiversity data using the Latent Dirichlet Allocation model, a probabilistic multivariate statistical method." *Ecology letters* 17.12 (2014): 1591-1601.

See Also

[rlda.multinomial](#), [rlda.bernoulli](#)

Examples

```
## Not run:
library(Rlda)
# Set seed
set.seed(5874)
# Hyperparameters for each prior distribution
gamma <- 0.01
```

```

alpha0 <- 0.01
alpha1 <- 0.01
# Execute the LDA for the Binomial entry
res<- rlda.binomialVB(data=tmp, loc.id='loc.id', n_community=10,
alpha0=0.01, alpha1=0.99, gamma=0.1, maxit=100, thresh=0.0001)

## End(Not run)

```

rlda.fastbernoulli *LDA with bernoulli entry and Stick-Breaking prior.*

Description

This method implements the Latent Dirichlet Allocation with Stick-Breaking prior for bernoulli data. `rlda.fastbernoulli` works with binary data.frame.

Usage

```

rlda.fastbernoulli(data, loc.id, n_community, alpha0, alpha1, gamma,
n_gibbs, ll_prior = TRUE, display_progress = TRUE)

```

Arguments

<code>data</code>	A binary data.frame where each row is a sampling unit (i.e. Plots, Locations, Time, etc.) and each column is a categorical type of element (i.e. Species, Firms, Issues, etc.). The elements inside this data.frame must be Zeros and Ones.
<code>loc.id</code>	ID variable for repeated measure.
<code>n_community</code>	Total number of communities to return. It must be less than the total number of columns inside the data data.frame.
<code>alpha0</code>	Hyperparameter associated with the Beta prior Beta(alpha0, alpha1).
<code>alpha1</code>	Hyperparameter associated with the Beta prior Beta(alpha0, alpha1).
<code>gamma</code>	Hyperparameter associated with the Stick-Breaking prior.
<code>n_gibbs</code>	Total number of Gibbs Samples.
<code>ll_prior</code>	boolean scalar indicating TRUE if the log-likelihood must be computed using also the priors or FALSE otherwise.
<code>display_progress</code>	boolean scalar TRUE if the Progress Bar must be showed and FALSE otherwise.

Details

`rlda.fastbernoulli` uses a modified Latent Dirichlet Allocation method to construct Mixed-Membership Clusters using Bayesian Inference. The data must be a non-empty data.frame with the binaries values Zero or Ones for each variable (column) in each observation (row).

Value

A R List with three elements:

Theta	The individual probability for each observation (ex: location) belong in each cluster (ex: community). It is a matrix with dimension equal n_gibbs by $nrow(data) * n_community$
Phi	The individual probability for each variable (ex: Specie) belong in each cluster (ex: community). It is a matrix with dimension equal n_gibbs by $ncol(data) * n_community$
LogLikelihood	The vector of Log-Likelihoods compute for each Gibbs Sample.

Note

The Theta and Phi matrix can be obtained for the i-th gibbs sampling using `matrix(Theta[i,], nrow = nrow(data), ncol = ncol(data))` and `matrix(Phi[i,], nrow = n_community, ncol = ncol(data))`, respectively.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>
<http://daijiang.name/>

References

- Blei, David M., Andrew Y. Ng, and Michael I. Jordan.
"Latent dirichlet allocation." *Journal of machine Learning research* 3.Jan (2003): 993-1022.
<http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>
- Valle, Denis, et al.
"Decomposing biodiversity data using the Latent Dirichlet Allocation model, a probabilistic multivariate statistical method." *Ecology letters* 17.12 (2014): 1591-1601.

See Also

[rlda.multinomial](#), [rlda.binomial](#)

Examples

```
## Not run:
library(Rlda)
# Presence
data(presence)
# Set seed
set.seed(9842)
```

```

# Hyperparameters for each prior distribution
gamma <- 0.01
alpha0 <- 0.01
alpha1 <- 0.01
presence$loc.id<-seq(1,nrow(presence))
# Execute the LDA for the Bernoulli entry
res <- rlda.fastbernoulli(data = presence, loc.id='loc.id', n_community = 10,
alpha0 = alpha0, alpha1 = alpha1, gamma = gamma,
n_gibbs = 5000,ll_prior = TRUE, display_progress = TRUE)

## End(Not run)

```

rlda.multinomial *LDA with multinomial entry and Stick-Breaking prior.*

Description

This method implements the Latent Dirichlet Allocation with Stick-Breaking prior for multinomial data. `rlda.multinomial` works with frequency data.frame.

Usage

```

rlda.multinomial(data, n_community, beta, gamma,
n_gibbs, ll_prior = TRUE, display_progress = TRUE)

```

Arguments

<code>data</code>	A abundance data.frame where each row is a sampling unit (i.e. Plots, Locations, Time, etc.) and each column is a categorical type of element (i.e. Species, Firms, Issues, etc.).
<code>n_community</code>	Total number of communities to return. It must be less than the total number of columns inside the data data.frame.
<code>beta</code>	Hyperparameter associated with the Dirichlet Phi matrix.
<code>gamma</code>	Hyperparameter associated with the Stick-Breaking prior.
<code>n_gibbs</code>	Total number of Gibbs Samples.
<code>ll_prior</code>	boolean scalar, TRUE if the log-likelihood must be computed using also the priors or FALSE otherwise.
<code>display_progress</code>	boolean scalar, TRUE if the Progress Bar must be showed and FALSE otherwise.

Details

`rlda.multinomial` uses a modified Latent Dirichlet Allocation method to construct Mixed-Membership Clusters using Bayesian Inference. The data must be a non-empty data.frame with the frequencies for each variable (column) in each observation (row).

Value

A R List with three elements:

Theta	The individual probability for each observation (ex: location) belong in each cluster (ex: community). It is a matrix with dimension equal n_gibbs by $nrow(data) * n_community$
Phi	The individual probability for each variable (ex: Specie) belong in each cluster (ex: community). It is a matrix with dimension equal n_gibbs by $ncol(data) * n_community$
LogLikelihood	The vector of Log-Likelihoods compute for each Gibbs Sample.

Note

The Theta and Phi matrix can be obtained for the i-th gibbs sampling using `matrix(Theta[i,], nrow = nrow(data), ncol = ncol(data))` and `matrix(Phi[i,], nrow = n_community, ncol = ncol(data))`, respectively.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>

References

- Blei, David M., Andrew Y. Ng, and Michael I. Jordan.
"Latent dirichlet allocation." *Journal of machine Learning research* 3.Jan (2003): 993-1022.
<http://www.jmlr.org/papers/volume3/blei03a/blei03a.pdf>
- Valle, Denis, et al.
"Decomposing biodiversity data using the Latent Dirichlet Allocation model, a probabilistic multivariate statistical method." *Ecology letters* 17.12 (2014): 1591-1601.

See Also

[rlda.binomial](#), [rlda.bernoulli](#)

Examples

```
## Not run:  
# Invoke the library  
library(Rlda)  
# Read the Complaints data  
data(complaints)
```

```
# Create the abundance matrix
library(reshape2)
mat1 <- dcast(complaints[, c("Company","Issue")],
  Company ~ Issue, fun.aggregate = length,
  value.var = "Issue")
# Create the rowname
rownames(mat1) <- mat1[, 1]
# Remove the ID variable
mat1 <- mat1[, -1]

# Set seed
set.seed(9292)
# Hyperparameters for each prior distribution
beta <- rep(1,ncol(mat1))
gamma <- 0.01

#Execute the LDA for the Multinomial entry
res <- rlda.multinomial(data = mat1, n_community = 30,
  beta = beta, gamma = gamma, n_gibbs = 1000,
  ll_prior = TRUE, display_progress = TRUE)

## End(Not run)
```

rlda2mcmc

Conversion between rlda to mcmc object from coda package

Description

Convert rlda object to mcmc object from coda package The rlda2mcmc function create a list of two elements including the Theta matrix and Phi matrix in a mcmc class.

Usage

```
## S3 method for class 'rlda'
rlda2mcmc(object, ...)
```

Arguments

object	a rlda object created by rlda.binomial, rlda.bernoulli or rlda.multinomial function.
...	other arguments may be useful.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>

- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>

See Also

[rlda.binomial](#), [rlda.bernoulli](#), [rlda.multinomial](#)

Examples

```
## Not run:
#Load data
data(presence)
#Set seed
set.seed(9842)
#Hyperparameters for each prior distribution
gamma <-0.01
alpha0<-0.01
alpha1<-0.01
#Execute the LDA for the Binomial entry
res<-rlda.bernoulli(presence, 10, alpha0, alpha1, gamma,
                   5000, TRUE, FALSE)
#rlda2mcmc the results
listParams <- rlda2mcmc(res)

## End(Not run)
```

sp500

Daily transactions Sp500

Description

Daily transactions for 46 firms of the Sp500 Index in 2015.

Usage

```
data(sp500)
```

Format

A data frame with 249 observations for 46 firms. Each cell represents one when some transactions occurred and zero otherwise.

Examples

```
data(sp500)
```

summary	<i>Provide summary information.</i>
---------	-------------------------------------

Description

Takes a `rlda` object produced by `rlda.binomial`, `rlda.bernoulli` or `rlda.multinomial` and produces a summary from it.

Usage

```
## S3 method for class 'rlda'  
summary(object, burnin=0.1, quantile = 0.95, silent=FALSE, ...)
```

Arguments

<code>object</code>	a <code>rlda</code> object as produced by <code>rlda.binomial</code> , <code>rlda.bernoulli</code> or <code>rlda.multinomial</code> .
<code>burnin</code>	a percentual of burn-in observations must be a number between 0 and 1. The default value is <code>burnin=0.1</code>
<code>quantile</code>	a quantile for the credible interval. The default value is <code>quantile=0.95</code>
<code>silent</code>	a logical value specifying if the number of clusters, variables and gibbs interactions must be printed. The default value is <code>silent=FALSE</code>
<code>...</code>	other arguments may be useful.

Details

Summarizes the Gibbs Sampling results and arguments.

Author(s)

- Pedro Albuquerque.
<pedroa@unb.br>
<http://pedrounb.blogspot.com/>
- Denis Valle.
<drvalle@ufl.edu>
<http://denisvalle.weebly.com/>
- Daijiang Li.
<daijianglee@gmail.com>

See Also

[rlda.binomial](#), [rlda.bernoulli](#), [rlda.multinomial](#)

Examples

```
## Not run:
#Load data
data(presence)
#Set seed
set.seed(9842)
#Hyperparameters for each prior distribution
gamma <-0.01
alpha0<-0.01
alpha1<-0.01
#Execute the LDA for the Binomial entry
res<-rlda.bernoulli(presence, 10, alpha0, alpha1, gamma,
                    5000, TRUE, FALSE)

#Summary method
Theta<-summary(res)$Theta

## End(Not run)
```

Index

- *Topic **Bernoulli**
 - rlda.bernoulli, 27
 - rlda.bernoulliMH, 29
 - rlda.fastbernoulli, 38
- *Topic **Binomial**
 - rlda.binomial, 31
 - rlda.binomialMH, 34
 - rlda.binomialVB, 36
- *Topic **Cluster**
 - getPhi, 18
 - getTheta, 19
 - logLik, 22
 - plot, 23
 - predict, 24
 - print, 26
 - rlda.bernoulli, 27
 - rlda.bernoulliMH, 29
 - rlda.binomial, 31
 - rlda.binomialMH, 34
 - rlda.binomialVB, 36
 - rlda.fastbernoulli, 38
 - rlda.multinomial, 40
 - rlda2mcmc, 42
 - summary, 44
- *Topic **Fuzzy**
 - getPhi, 18
 - getTheta, 19
 - logLik, 22
 - plot, 23
 - predict, 24
 - print, 26
 - rlda2mcmc, 42
 - summary, 44
- *Topic **LDA**
 - getPhi, 18
 - getTheta, 19
 - logLik, 22
 - plot, 23
 - predict, 24
 - print, 26
 - rlda.bernoulli, 27
 - rlda.bernoulliMH, 29
 - rlda.binomial, 31
 - rlda.binomialMH, 34
 - rlda.binomialVB, 36
 - rlda.fastbernoulli, 38
 - rlda.multinomial, 40
 - rlda2mcmc, 42
 - summary, 44
- *Topic **Mixed-Membership**
 - rlda.bernoulli, 27
 - rlda.bernoulliMH, 29
 - rlda.binomial, 31
 - rlda.binomialMH, 34
 - rlda.binomialVB, 36
 - rlda.fastbernoulli, 38
 - rlda.multinomial, 40
- *Topic **Multinomial**
 - rlda.multinomial, 40
- *Topic **Rlda**
 - generateBernoulliLDA, 14
 - generateBinomialLDA, 15
 - generateMultinomialLDA, 17
- *Topic **Variational Inference**
 - rlda.binomialVB, 36
- *Topic **datasets**
 - birds, 2
 - complaints, 13
 - fishnet, 14
 - Landsat, 20
 - LocationsBirds, 21
 - presence, 25
 - sp500, 43
- *Topic **generateBernoulliLDA**
 - generateBernoulliLDA, 14
 - generateBinomialLDA, 15
 - generateMultinomialLDA, 17
- *Topic **generateBinomialLDA**

- generateBernoulliLDA, [14](#)
 - generateBinomialLDA, [15](#)
 - generateMultinomialLDA, [17](#)
 - *Topic **generateMultinomialLDA**
 - generateBernoulliLDA, [14](#)
 - generateBinomialLDA, [15](#)
 - generateMultinomialLDA, [17](#)
 - *Topic **getPhi**
 - getPhi, [18](#)
 - *Topic **getTheta**
 - getTheta, [19](#)
 - *Topic **plot**
 - plot, [23](#)
 - *Topic **print**
 - print, [26](#)
 - *Topic **rlda2mcmc**
 - rlda2mcmc, [42](#)
 - *Topic **summary**
 - logLik, [22](#)
 - predict, [24](#)
 - summary, [44](#)
- birds, [2](#)
 - complaints, [13](#)
 - fishnet, [14](#)
 - generateBernoulliLDA, [14](#), [16](#), [17](#)
 - generateBinomialLDA, [15](#), [15](#), [17](#)
 - generateMultinomialLDA, [15](#), [16](#), [17](#)
 - getPhi, [18](#)
 - getTheta, [19](#)
 - Landsat, [20](#)
 - LocationsBirds, [21](#)
 - logLik, [22](#)
 - plot, [23](#)
 - plot, Rlda (plot), [23](#)
 - plot.rlda (plot), [23](#)
 - predict, [24](#)
 - presence, [25](#)
 - print, [26](#)
 - rlda.bernoulli, [19](#), [20](#), [22](#), [23](#), [25](#), [27](#), [27](#), [33](#), [35](#), [37](#), [41](#), [43](#), [44](#)
 - rlda.bernoulliMH, [29](#)
 - rlda.binomial, [19](#), [20](#), [22](#), [23](#), [25](#), [27](#), [29](#), [31](#), [31](#), [39](#), [41](#), [43](#), [44](#)
 - rlda.binomialMH, [34](#)
 - rlda.binomialVB, [36](#)
 - rlda.fastbernoulli, [38](#)
 - rlda.multinomial, [19](#), [20](#), [22](#), [23](#), [25](#), [27](#), [29](#), [31](#), [33](#), [35](#), [37](#), [39](#), [40](#), [43](#), [44](#)
 - rlda2mcmc, [42](#)
 - sp500, [43](#)
 - summary, [44](#)