Package 'plotmm'

January 8, 2025

Type Package

Title Tidy Tools for Visualizing Mixture Models

Version 0.1.2

BugReports https://github.com/pdwaggoner/plotmm/issues

Maintainer Philip Waggoner <philip.waggoner@gmail.com>

Description The main function, plot_mm(), is used for (gg)plotting output from mixture models, including both densities and overlaying mixture weight component curves from the fit models in line with the
tidy principles. The package includes several additional functions for added plot customization. Supported model objects include: 'mixtools', 'EMCluster', and 'flexmix', with more from each in active dev.
Supported mixture model specifications include mixtures of univariate Gaussians, multivariate Gaussians, Gammas,
logistic regressions, linear regressions, and Poisson regressions.

Imports methods, wesanderson, amerika, ggplot2, mixtools, EMCluster, flexmix

Suggests testthat, graphics, dplyr, patchwork, survival, magrittr, knitr, rmarkdown

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.1

VignetteBuilder knitr

NeedsCompilation no

Author Philip Waggoner [aut, cre], Fong Chan [aut, ctb], Lu Zhang [aut, ctb]

Repository CRAN

Date/Publication 2025-01-08 16:50:06 UTC

Contents

`plot_cut_point` *Tidy Visualization of a Cut Point from a Mixture Model*

Description

Returns a plot of the data density (histogram) with an overlaid cut point generated by the fit mixture model

Usage

```
plot_cut_point(m, plot = TRUE, color = c("grayscale", "amerika", "wesanderson"))
```

Arguments

`m`	An object of class `mixEM` corresponding with the fit mixture model
`plot`	Logical for generating the plot. If FALSE, only the cut point value from the GMM is returned. If TRUE, histogram with the overlaid cut point is returned. Default is set to TRUE.
`color`	A vector of color options including "amerika" (from `amerika` package), "wesanderson" (from `wesanderson` package), and "grayscale", which is the default option.

Details

Mixture models can be used to derive cut points separating clusters via soft assignment (See Benaglia et al. 2009 for more). `plot_cut_point()` plots data density with an overlaid cut point (the mean of the calculated `mu`) from `mixEM` objects via `mixtools`. Note, this function is in its infancy, and at present only works in the limited context of 2-component Gaussian mixture models with equal priors and equal variances. Development for more complex cases is in process.

References

Benaglia, T., Chauveau, D., Hunter, D. and Young, D. 2009. mixtools: An R package for analyzing finite mixture models. Journal of Statistical Software, 32(6), pp.1-29.

Ram, K., and Wickham, H. 2015. wesanderson: a Wes Anderson palette generator. R package version 0.3.

Examples

```
## Not run:
if(require(mixtools)){
mixmdl <- mixtools::normalmixEM(faithful$waiting, k = 2)
}
plot_cut_point(mixmdl, plot = TRUE, color = "amerika") # returns plot, amerika
plot_cut_point(mixmdl, plot = TRUE, color = "wesanderson") # returns plot, wesanderson
plot_cut_point(mixmdl, plot = FALSE) # returns only the cut point value from the GMM

## End(Not run)
```

---

plot_gmm *Plots Mixture Components from Gaussian Mixture Models*

Description

Generates a plot of data densities with overlaid mixture components from a Gaussian mixture model (GMM)

Usage

```
plot_gmm(m, k = NULL)
```

Arguments

m	An object of class `mixEM` corresponding with the fit GMM
k	The number of components specified in the GMM, `m`

Details

Original function from the `plotGMM` package. Retained here for bridging between the packages. We recommend using instead the updated `plot_mm` function.

Note: `plot_gmm` requires a `mixtools` object to be supplied. Users must enter the same component value, `k`, in the `plot_gmm` function, as that which was specified in the original GMM specification (also `k` in `mixtools`).

References

Benaglia, T., Chauveau, D., Hunter, D. and Young, D., 2009. mixtools: An R package for analyzing finite mixture models. Journal of Statistical Software, 32(6), pp.1-29.

Wickham, H., 2016. ggplot2: elegant graphics for data analysis. Springer.

Examples

```
## Not run:
if(require(mixtools)){
mixmdl <- mixtools::normalmixEM(faithful$waiting, k = 2)
}
plot_gmm(mixmdl, 2)

## End(Not run)
```

`plot_mix_comps` *Helper Function for Overlaying Mixture Components*

Description

Allows for plotting mixture components conditioned on a superimposed function meant for passage to ggplot's `stat_function()`

Usage

```
plot_mix_comps(x, mu = NULL, sigma = NULL, lam = 1, beta0 = NULL,
 beta1=NULL, alpha=NULL, beta=NULL,
 normal=FALSE, logisreg=FALSE,
 gamma=FALSE, poisson=FALSE)
```

Arguments

`x`	Input data
`mu`	Component mean
`sigma`	Component variance
`lam`	Component mixture weight
`beta0`	Coefficient values
`beta1`	Coefficient values
`alpha`	Initial shape parameters
`beta`	Initial parameter values
`normal`	Logical for normal distribution
`logisreg`	Logical for logistic regression mixtures
`gamma`	Logical for gamma distribution
`poisson`	Logical for poisson regression mixtures

Details

Allows for component curves to be superimposed over a mixture model plot

Examples

```
## Not run:
if(require(mixtools)){
mixmdl <- mixtools::normalmixEM(faithful$waiting, k = 2)
}
x <- mixmdl$x
x <- data.frame(x)
ggplot2::ggplot(data.frame(x)) +
  ggplot2::geom_density(ggplot2::aes(x), color="black", fill="black") +
  ggplot2::stat_function(geom = "line", fun = plot_mix_comps,
                args = list(mixmdl$mu[1], mixmdl$sigma[1], lam = mixmdl$lambda[1]),
                colour = "red") +
  ggplot2::stat_function(geom = "line", fun = plot_mix_comps,
                args = list(mixmdl$mu[2], mixmdl$sigma[2], lam = mixmdl$lambda[2]),
               colour = "blue")

## End(Not run)
```

---

plot_mix_comps_normal *Custom Function for Overlaying Mixture Components for Normal Distributions*

Description

Plots a mixture component conditioned on a superimposed function

Usage

```
plot_mix_comps_normal(x, mu, sigma, lam)
```

Arguments

x	Input data
mu	Mean of component
sigma	Variance of component
lam	Mixture weight of component

Details

Allows for specifying a custom function to be superimposed when plotting a mixture component assuming a normal distribution. This is the original function for the package, which is also included in the updated `plot_mix_comps()` function.

Examples

```
## Not run:
if(require(mixtools)){
mixmdl <- mixtools::normalmixEM(faithful$waiting, k = 2)
}
x <- mixmdl$x
x <- data.frame(x)
ggplot2::ggplot(data.frame(x)) +
  ggplot2::geom_density(ggplot2::aes(x), color="black", fill="black") +
  ggplot2::stat_function(geom = "line", fun = plot_mix_comps_normal,
                args = list(mixmdl$mu[1], mixmdl$sigma[1], lam = mixmdl$lambda[1]),
                colour = "red") +
  ggplot2::stat_function(geom = "line", fun = plot_mix_comps_normal,
                args = list(mixmdl$mu[2], mixmdl$sigma[2], lam = mixmdl$lambda[2]),
               colour = "blue")

## End(Not run)
```

`plot_mm` *Tidy Visualization of Mixture Models*

Description

Generates a ggplot of data densities with overlaid mixture components from fit mixture models.

Usage

```
plot_mm(m, k = NULL, data = NULL)
```

Arguments

`m`	A mixture model object
`k`	Optional. The number of components specified in the mixture model, `m`
`data`	Name of data object required only for `EMCluster` objects

Details

This is the core function in the package, returning a `ggplot` object for a fit mixture model. The plot includes the data density with overlaid mixture components.

References

Wickham, H., 2016. ggplot2: elegant graphics for data analysis. Springer.

Examples

```
## Not run:
if(require(mixtools)){
mixmdl1 <- mixtools::normalmixEM(faithful$waiting, k = 2)
}
plot_mm(mixmdl1, 2)

if(require(mixtools)){
x <- c(rgamma(200, shape = 50, scale = 11), rgamma(200, shape = 28, scale = 6))
mixmdl2 <- mixtools::gammamixEM(x, lambda = c(1, 1)/2)
}
plot_mm(mixmdl2)

## End(Not run)
```

Index