List of distributions in hmmTMB

Théo Michelot

The hidden Markov models implemented in hmmTMB comprise two stochastic processes: an unobserved state process (S_t), defined as a discrete-valued Markov chain, and an observation process (Z_t). The observation depends on the hidden state as follows,

$$Z_t|\{S_t = j\} \sim \mathcal{D}(\theta_j),$$

where $\mathcal{D}$ is a distribution, and θ_j is the vector of its parameters in state $j \in \{1, 2, \dots\}$.

In practice, the choice of $\mathcal{D}$ depends on the observed variable; for example, a beta variable might be appropriate for a variable defined between 0 and 1, or a gamma distribution for a strictly positive variable. The distributions that are currently included in hmmTMB can be accessed using `hmmTMB:::dist_list`, and a list of distributions together with their parameters is shown below in alphabetical order. The word in quotes is the name of the distribution used in the package, which should be used when defining the observation model.

```
## "beta": beta(shape1, shape2)
## "binom": binomial(size, prob)
## "cat": categorical(p1)
## "dir": Dirichlet(alpha1, alpha2)
## "exp": exponential(rate)
## "foldednorm": folded normal(mean, sd)
## "gamma": gamma(shape, scale)
## "gamma2": gamma2(mean, sd)
## "lnorm": log-normal(meanlog, sdlog)
## "mvnorm": multivariate normal(mu1, mu2, sd1, sd2, corr12)
## "nbinom": negative binomial(size, prob)
## "norm": normal(mean, sd)
## "pois": Poisson(lambda)
## "t": Student's t(mean, scale)
## "truncnorm": truncated normal(mean, sd, min, max)
## "tweedie": Tweedie(mean, p, phi)
```

```
## "vm": von Mises(mu, kappa)
## "weibull": Weibull(shape, scale)
## "wrpcauchy": wrapped Cauchy(mu, rho)
## "zib": zero-inflated binomial(size, prob, z)
## "zigamma": zero-inflated gamma(shape, scale, z)
## "zigamma2": zero-inflated gamma2(mean, sd, z)
## "zinb": zero-inflated negative binomial(size, prob, z)
## "zip": zero-inflated Poisson(lambda, z)
## "ztnb": zero-truncated negative binomial(size, prob)
## "ztp": zero-truncated Poisson(lambda)
```