```r
install.packages("Hyde")
```

```
## Warning: package 'Hyde' is not available (for R Under development)
```

```r
library(Hyde)
```

```
## Loading required package: gRbase
## Loading required package: nnet
## Loading required package: rjags
## Loading required package: coda
## Loading required package: lattice
## Linked to JAGS 3.4.0
## Loaded modules: basemod,bugs
```

```r
# summary(DATASET)
```

```r
net <- HydeNetwork(~ x)

plot(net)
```

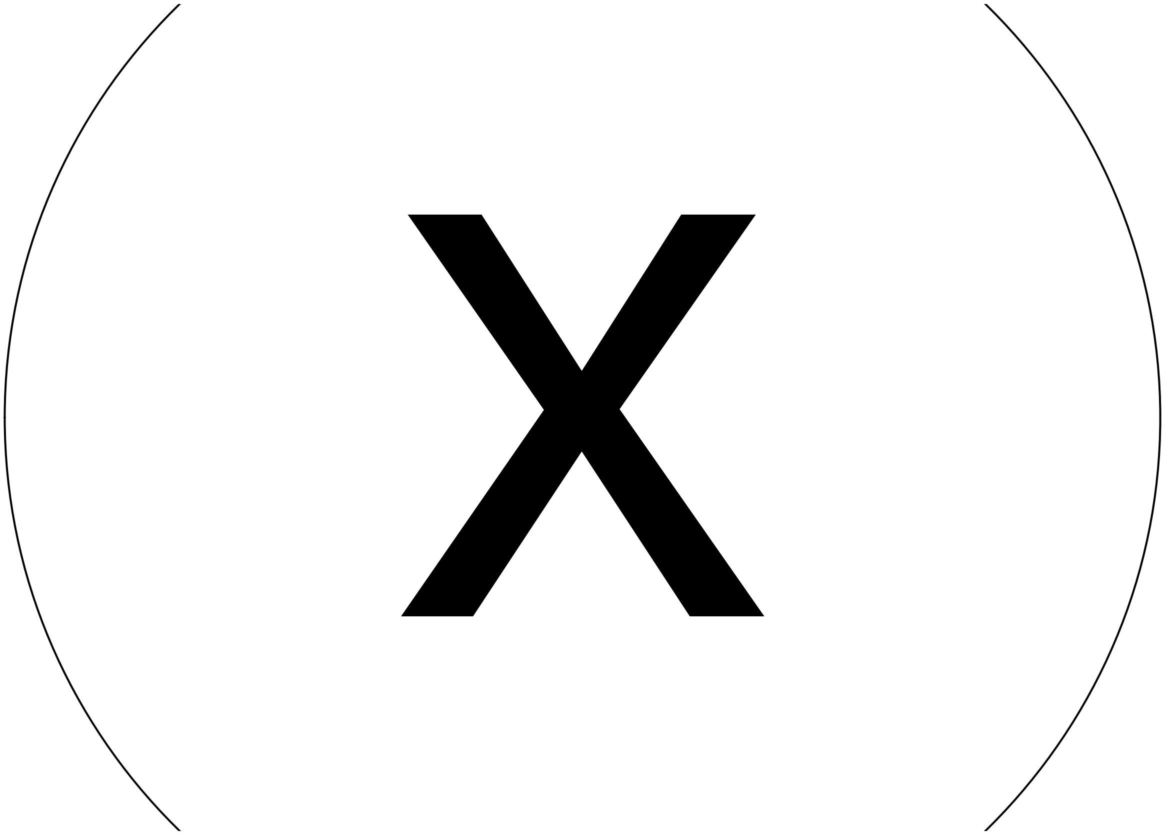

Figure 1: plot of chunk unnamed-chunk-3

```
# (see `vignette("Customizing_Network_Graphs", package = "Hyde")`)

net
```

```
## A Probabilistic Graphical Network
## Has data attached: No
##
## x
## dnorm(mu = Unspecified, tau = Unspecified)
## : x ~ 1
```

1 Root Nodes

1.1 Binary

1.1.1 Using `setNode`

1.1.2 Using JAGS Code

1.2 Categorical

1.2.1 Using `setNode`

1.2.2 Using JAGS code

1.3 Normal

1.3.1 Using `setNode`

1.3.2 Using JAGS Code

1.4 Uniform

1.4.1 Using `setNode`

1.4.2 Using JAGS code

1.5 Weibull

1.5.1 Using `setNode`

1.5.2 Using JAGS code

1.6 Other types:

dgamma, dexp, dpois, dnegbin, etc

2 Nodes With or Without Parents – Manually specifying regression equations

2.1 Nodes in the exponential family – need regression equation + link function

2.1.1 Ordinary Least Squares

2.1.2 Lognormal

2.1.3 Logistic regression nodes . . . I think JAGS has a logit() function that my be useful

2.1.4 Poisson Regression nodes

2.1.5 Gamma regression nodes

2.2 Categorical Nodes – list (?) of multinomial logistic regression equations – this will be tricky

2.3 Continuous Nodes

2.3.1 Regression equation with Gaussian errors

2.4 Manual Specification using JAGS code

2.4.1 Linear regression equation with cauchy errors

2.4.2 Mixture distributions (do this last!)

2.4.2.1 Zero-inflated Poisson

2.4.2.2 Zero inflated negative binomial

3 Node With or Without Parents – Reading formula from existing model objects

3.1 Reading from `lm` objects

3.2 Reading from `glm` objects - can we make this work for lognormal

3.3 Reading from multinom objects

3.4 Reading from survreg objects (note the list component modelObject$dist that stores the type of distribution used for the parametric survival regression model)

4 Learning Node Distributions from Training Data

4.1 Default behavior of `HydeNetwork(.formula, data=trainingData)`

4.2 Tutorial involving user-specified distributions for particular nodes and estimation of regression equations from data

one node may be survival (with censoring), one may be binary, one may be normal, one may be Poisson, and the other may be categorical