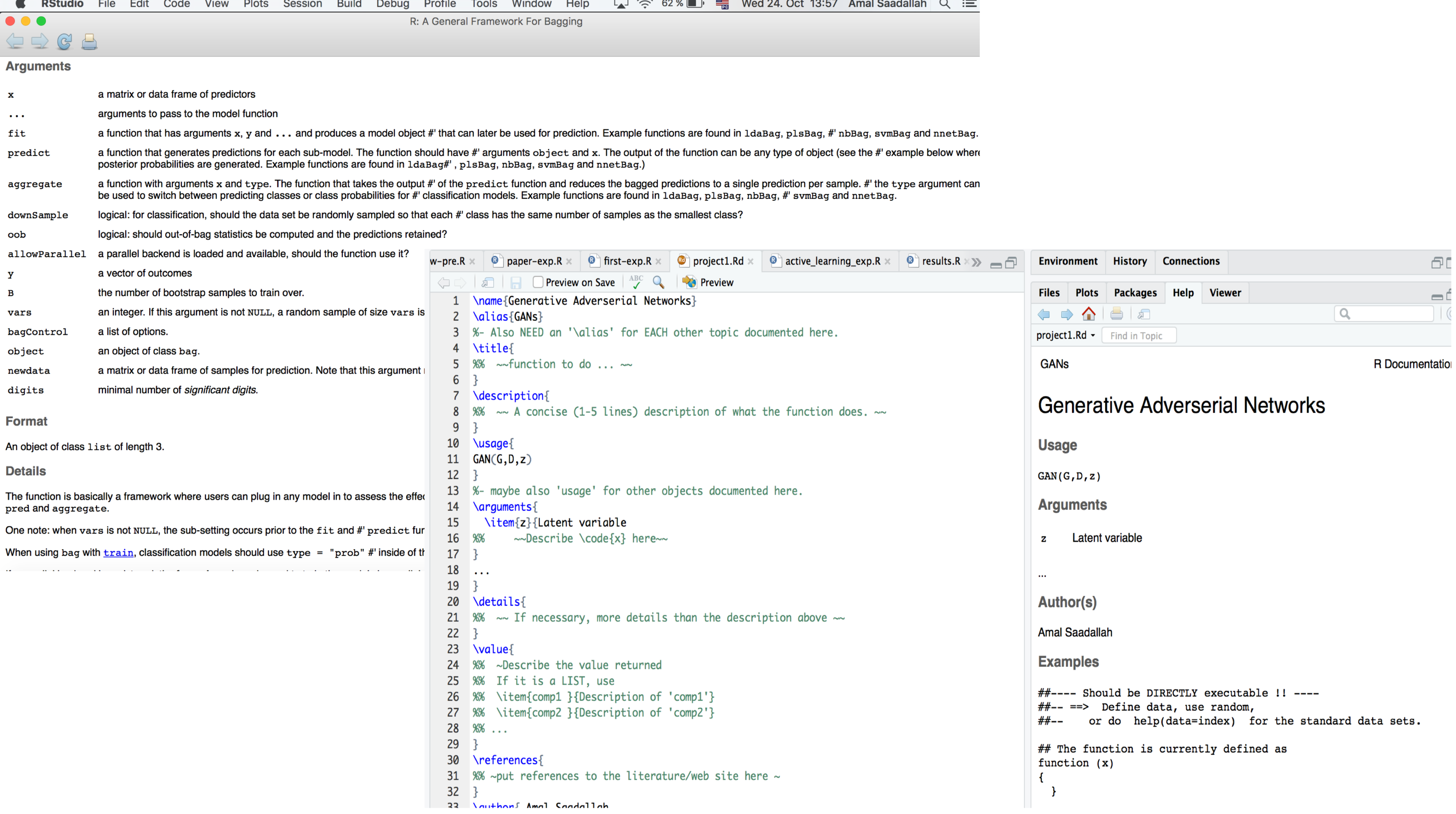

Arguments

`x` a matrix or data frame of predictors

`...` arguments to pass to the model function

`fit` a function that has arguments `x`, `y` and `...` and produces a model object #' that can later be used for prediction. Example functions are found in `ldaBag`, `plsBag`, #' `nbBag`, `svmBag` and `nnetBag`.

`predict` a function that generates predictions for each sub-model. The function should have #' arguments `object` and `x`. The output of the function can be any type of object (see the #' example below where posterior probabilities are generated. Example functions are found in `ldaBag#'` , `plsBag`, `nbBag`, `svmBag` and `nnetBag`.)

`aggregate` a function with arguments `x` and `type`. The function that takes the output #' of the `predict` function and reduces the bagged predictions to a single prediction per sample. #' the `type` argument can be used to switch between predicting classes or class probabilities for #' classification models. Example functions are found in `ldaBag`, `plsBag`, `nbBag`, #' `svmBag` and `nnetBag`.

`downSample` logical: for classification, should the data set be randomly sampled so that each #' class has the same number of samples as the smallest class?

`oob` logical: should out-of-bag statistics be computed and the predictions retained?

`allowParallel` a parallel backend is loaded and available, should the function use it?

`y` a vector of outcomes

`B` the number of bootstrap samples to train over.

`vars` an integer. If this argument is not `NULL`, a random sample of size `vars` is

`bagControl` a list of options.

`object` an object of class `bag`.

`newdata` a matrix or data frame of samples for prediction. Note that this argument

`digits` minimal number of *significant digits*.

Format

An object of class `list` of length 3.

Details

The function is basically a framework where users can plug in any model in to assess the effe `pred` and `aggregate`.

One note: when `vars` is not `NULL`, the sub-setting occurs prior to the `fit` and #' `predict` fur

When using `bag` with train, classification models should use `type = "prob"` #' inside of tł

```
\name{Generative Adverserial Networks}
\alias{GANs}
%- Also NEED an '\alias' for EACH other topic documented here.
\title{
%%  ~~function to do ... ~~
}
\description{
%%  ~~ A concise (1-5 lines) description of what the function does. ~~
}
\usage{
GAN(G,D,z)
}
%- maybe also 'usage' for other objects documented here.
\arguments{
  \item{z}{Latent variable
%%     ~~Describe \code{x} here~~
}
...
}
\details{
%%  ~~ If necessary, more details than the description above ~~
}
\value{
%%  ~Describe the value returned
%%  If it is a LIST, use
%%  \item{comp1 }{Description of 'comp1'}
%%  \item{comp2 }{Description of 'comp2'}
%% ...
}
\references{
%% ~put references to the literature/web site here ~
}
\author{ Amal Saadallah
```

GANs R Documentatio

Generative Adverserial Networks

Usage

```
GAN(G,D,z)
```

Arguments

`z` Latent variable

...

Author(s)

Amal Saadallah

Examples

```
##---- Should be DIRECTLY executable !! ----
##-- ==>  Define data, use random,
##--      or do  help(data=index)  for the standard data sets.

## The function is currently defined as
function (x)
{
  }
```