In all tasks specified below you will work with a part of Forbes2000 data set, which comes from HSAUR package. You should prepare your data using the following function:

```r
prepare_data = function(){
  library(HSAUR)  # load the library with Forbes data set
  F = Forbes2000  # just to make a copy
  F = F[!is.na(F$profits), ]  # rows with NA values are removed
  # now we select only countries with at least 25 companies in the data
  selected.countries = names(table(F$country)[table(F$country) >= 25])
  F = F[F$country %in% selected.countries, ]
  F$country = droplevels(F$country)
  cat(nrow(F), "observations selected from Forbes2000 data set.\n")
  cat("Selected countries: ",
      paste(selected.countries, collapse=" ", ", ", sep=" ")
  # to randomly split the data into two disjoint subsets
  set.seed(123); s = sample(1710)
  forbes.train <<- F[s[1:1200], ]  # training examples
  forbes.test <<- F[s[1201:1710], ]  # test examples
}
```

When you run `prepare_data()`, you will get two data frames, namely `forbes.train` and `forbes.test` with the same structure:

```r
> str(forbes.train)
'data.frame': 1200 obs. of  8 variables:
$ rank     : int 555 1568 795 1762 1873 81 1026 1774 1076 882 ... 
$ name     : chr "KeySpan" "M6-Metropole Television" "Zions Bancorp" "Buderus" ... 
$ country  : Factor w/ 16 levels "Australia","Canada",...: 16 4 16 5 5 3 8 15 16 12 ... 
$ category : Factor w/ 27 levels "Aerospace & defense",...: 27 18 2 7 9 19 18 27 2 3 ... 
$ sales    : num 6.85 1.48 1.89 1.51 2.14 ... 
$ profits  : num 0.4 0.17 0.34 0.25 0.16 1.94 0.16 -0.1 0.23 0.17 ... 
$ assets   : num 13 1.2 28.56 1.51 2.46 5.79 4.37 5.25 2.46 3.01 ... 
$ marketvalue: num 5.79 4.37 5.25 2.46 3.01 ... 
```

Variable `profits` will be considered as an output attribute. Look at its distribution. Is it similarly distributed in the training and the test set?
Task 1 – Estimating probability distributions and entropy

- Read file `xy.100.csv` – there are 100 observations of two random variables X and Y.
  - hint: `read.table("xy.100.csv", header=T)`
- Compute the estimate of marginal distributions p(x), p(y).
  - i.e. estimate the probabilities p(x), p(y) for all values of X and Y
- Draw histograms for both variables.
  - hints: `plot(<factor>), barplot(table(...)), hist(...)`
- Compute the estimate of joint and conditional distributions p(x,y), p(x|y), p(y|x).
  - i.e. estimate those probabilities for all pairs (x,y)
- Write your own functions for computing entropy and conditional entropy in R. Both functions `entropy(x)` and `entropy.cond(x,y)` should take factors as their input; `entropy.cond(x,y)` will take two factors of the same length and compute $H(x \mid y)$.
  - Example use: `entropy.cond(observations$x, observations$y)`
- Compute estimations of entropy and mutual information
  - $H(X), H(Y), H(X \mid Y), H(Y \mid X), I(X;Y)$
- Question: Are distributions X and Y statistically independent?
- Compare discrete features `country` and `category` in training data set `forbes.train`. Transform output attribute `profits` to a binary variable
  - `> forbes.train$profits = factor(forbes.train$profits > 0.2)`

Then use conditional entropy and compare how the two features can contribute to the prediction of binary profits.

Task 2 – Evaluation of classification Random Forests (RF)

Work with the above mentioned data frames `forbes.train` and `forbes.test`. First transform output attribute `profits` to a binary variable in both training and test data sets

- `> forbes.train$profits = factor(forbes.train$profits > 0.2)`
- `> forbes.test$profits  = factor(forbes.test$profits > 0.2)`

Build and evaluate RF models to predict binary `profits` using R package `randomForest`. There are 5 features that you can use for prediction: `category`, `sales`, `assets`, `marketvalue`, `country`.

- Learn 20 random forests with different number of trees using `ntree` in `seq(100, 2000, 100)`.
- For each random forest compute error rate estimate using 6-fold cross validation. In each cross validation run you will have 1000 examples for training and 200 examples for test. Compute mean and standard deviation of the error rate.
- Then compare the cross validation results with the OOB error estimates, and also with test error rates measured using `forbes.test`.
- Arrange all your results in a nice table and plot a chart.
Task 3 – Regression Random Forest (RF)

Work with the above mentioned data frames `forbes.train` and `forbes.test`. Build and evaluate a regression RF model to predict profits using R package `randomForest`. There are 5 features that you can use for prediction: category, sales, assets, marketvalue, country. For differences between classification and regression RF, see `help(randomForest)`.

- During the development process use only your development data in `forbes.train`. Choose a good value of `ntree` parameter and estimate the generalization error using cross validation. Report on your work.
- Only when you finish whole development, take the test set and evaluate your model. Compare the result with the error estimated during the development.
- Develop also a single decision tree and compare its performance to your random forest.