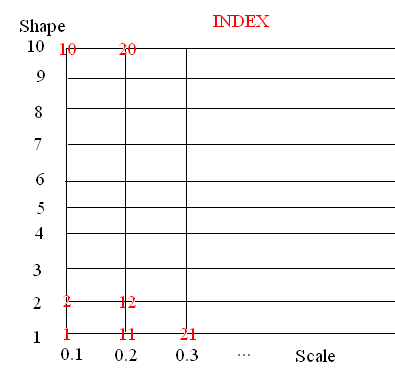
***Preliminaries***

The solutions to the assignments involving R coding should be reported in details, all R code that you write should be **included** into the report. Also, all appropriate pictures or diagrams should be included.

***Assignment 1: Genetic algorithm***

It is known that the vector *x* stored in *x.RData* comes from a gamma distribution. The purpose of this assignment is to find the maximum likelihood estimates of the parameters of this distribution.

1. Load the data set to R.
2. Write your own function *myfun( )* dependent on parameters *shape*  and *scale*  that computes the log-likelihood of observing vector *x*.(Use *dgamma() with log=TRUE, sum()*). Think about which values *shape* and *scale* can take and comment why it is a problem to use conventional optimization methods like BFGS.
3. Assume that your want now find the optimal log-likelihood using the grid (*Shape=1,2,..10, Scale=0.1, 0.2, … 10*). This implies the grid with 1000 points. Each point on this grid is given by its index as it is shown on the figure
4. Load functions from function.R and read commented information about them
5. Perform the genetic optimization by doing the following. Write your own function dependent on parameters *maxiter* and *mutprob* that
   1. Sets initial population by random sampling of 10 indexes from the total set of indexes (1…1000) (Use *sample()*).
   2. Computes the objective function for these indexes and stores it in some vector *values* (use *getC(), myfun()*)
   3. Performs *maxiter* iterations where at each iteration
      1. Two indexes are randomly sampled from the current population, they are further used as parents(use *sample().*)
      2. Two indexes with smallest objective function are selected from the current population, they are further referred to as victims(use *order()*)
      3. Use parents to produce two new children by crossover. Mutate both children with probability *mutprob*. (use *crossover(), mutate()*)
      4. Replace two victims by two new children in the population and update the list *values*
      5. The current maximal value of the objective function is saved
   4. Plots the initial population and resulting population in the coordinates (*Shape, Scale*) (use *getC(), plot(), points()*)
   5. Returns the list of the maximal objective function values for each iteration.
6. Run your function with different combinations of *maxiter=10, 100* and *mutprob=0.1, 0.5,0.9.* Observe the initial population and final population. Conclusions? For each of combination of the parameters plot also the time series showing how the maximal objective function was changing from iteration 1 to iteration *maxiter.*
7. The true values are: *shape =5, scale=5, log-likelihood=* *-746.5996.* What can you say about the performance of your genetic algorithm?

## Submission procedure

If you are neither speaker or opponent, you should just send-in the solutions via It’s learning. **Note that there is a deadline for submission!** If you are *speaker*, you must also do the following:

* Put your report and PowerPoint presentation to the folder *Lab 7* which is in the folder *Seminars*. Specify Active : “Set time span”. Specify activation time as **20 maj 17:00. Make sure that the time settings are specified!**

If you are *opponent*, you may find the report and PPT-file for revision available immediately after **20 maj 17:00** in the folder*Seminars->Lab 7*