Exam

Linköpings Universitet, Institutionen för datavetenskap, Statistik

Course:	732A33 Advanced Programming in R
Date and time:	2013-08-20, 8-12.
Teacher:	Patrik Waldmann
Material:	The Art of R Programming (Matloff). The book should be free from notes and comments, but can contain underlined lines and indicators for quickly finding chapters/sections etc. Slides from all lectures, the lab reports as well as the three papers on data mining in R can be included in a personal folder.

Grades:	Maximum is 20 points.
	A=19-20 points.
	B=17-18 points.
	C=12-16 points.
	D=10-11 points.
	E=8-9 points.
	F=0-7 points.

Solutions should be written in a Word file, do not include the questions.

The R code should be complete and readable code, possible to run by copying directly into a script.

Comment directly in the code whenever something needs to be explained or discussed.

Follow the instructions carefully.

1 Data and programming structure (4p)

a). Create a list named **computer** containing the following four elements:

The character vector **brand** containing (‘HP’,’Asus’,’Dell’,’Apple’)

The integer vector **price** containing (12000,11500,10300,13100)

The numeric vector **speed** containing (100.6,97.5,101.8,95.0)

The logical vector **Linux** containing (‘FALSE’,’FALSE’,’FALSE’,’TRUE’)

Sort the whole list according to **speed** (lowest value first). Only present the code. (2p)

b). Write a for-loop that loops over the numbers 1; 3; 6; 10; 15 and 21, and that prints

the product of the current number (not number 1) with the prior number on the screen. (2p)

2 Simulation and graphics (6p)

a). Simulate a data set of 1000 observations times 4 variables (denoted a,b,c,d) from the multivariate normal distribution with mean 0 and covariance matrix sigma, where the diagonal of sigma should be 1 and covariance between a and b 0.95, between c and d 0.5 (the other covariances 0). Visualize the data with a Trellis plot labeled with the names of the variables and title 'Trellis plot'. Present your code and plot. (3p)

b). Write a function that reads one vector (named **response** of size 1 x n) and one matrix (named **predictor** of size n rows x p columns). Then performs one linear model for each column in **predictor** and saves the regression coefficients into vector **coef**, and permute (sampling without replacement) the values within each column **x** times and for each sample calculate **x** times p new regression coefficients that should be saved into matrix **permcoef**. Sort the values within the columns of **permcoef** and calculate how many regression coefficients that is larger in each column when compared with **coef (**store in vector **pval**). Finally, divide the values in **pval** with **x**, and output **coef** and **pval** to the screen. (3p)

3 Strings and input/output (3p)

a). Write a function that reads a text file, finds the number of times a given word occurs in the text, and outputs to the screen the word and how many times this word occur in the text. The input to the function should be the name of the text file and the word that should be found. (3p)

4 Interfacing and parallel computations (4p)

a). Write code that reads matrix x (of dimension n x p) and calculates the maximum of each row in parallel (using `parapply`). The code should be so flexible that it detects the number of cores automatically, and the output should be a vector with maximum values. (2p) b). Compare the computation time of your code in a) with an alternative version that performs the same task based on the `clusterSplit` function. The timing should be over the complete parallel code. Provide code and timing output. (2p)

5 Data mining (3 p)

a). Your task is to employ `mboost` to study the linear relationship between the discrete response variable disease (where 1 indicates disease and 0 indicates no-disease) and 16 variables believed to influence the disease. The data should be read from the `Disease.txt` file. Fit a generalized linear boosting model to this data by using disease as a binomial response variable. Find the optimal number of iterations based on cross-validation. Provide a plot with all the selected variables. Explain your code carefully. (3p)