Assignment 1

Biomedical Data Science

Due on Wednesday 28 February 2018, 12 noon

The assignment is marked out of **100 points**, and will contribute to **20%** of your final mark.

Please submit a document with your answers to the problems in terms of graphs, tables and text, as required. Mark clearly what problem and sub-problem you are answering.

For each problem submit a text file with .R extension that contains the R code necessary to reproduce your results: you can double check that your code executes correctly by using command `source(filename.R)` from a clean R session.

Marks will be deducted for code that doesn't run or doesn't reproduce the values reported in the previous document. Up to **5 marks** will be awarded for writing clear and reusable code clarity (pay attention to indentation, choice of variable identifiers, comments, error checking, etc).

Overall, you are expected to **submit 5 files via Learn** (1 Word or pdf document, 4 .R files). Alternatively, if you are already familiar with Rmarkdown, you can submit just 1 .Rmd file.

Problem 1 (25 points)

In a longitudinal setting, a reasonable way to impute missing data is by considering a time window around the point in which an observation is missing. If observations occur at regular intervals (say every day, or every week, etc), as in the dataset we will be using below, it is sufficient to consider the neighbouring observations to identify a time window.

(a) Using the "Ozone" variable from the builtin `airquality` dataframe, perform a simple imputation by imputing missing values to the overall mean. **(3 points)**

(b) Write a function that imputes missing values in a vector according to the mean value of the elements within a window of a given size (that is, for a window of size 10, `x[30]` if missing should be imputed to the mean of `x[20:40]`, and `x[5]` if missing should be imputed to the mean of `x[1:15]`). The function should have the following signature:

```
impute.to.window.mean <- function(x, windowsize)
```

where `x` is a vector and `windowsize` is a positive number; the function should return a vector of the same length as input vector `x`. **(8 points)**

(c) Use `impute.to.window.mean()` to repeat the imputation of variable "Ozone" using windows of size 10, 25, 50, 75, 100 and 125. For each window size compute the maximum absolute difference between the corresponding imputation and the imputation to the overall mean obtained at point (a). Collect the results in a dataframe, present them as a table and graphically as a plot, and provide an interpretation of the results you see. **(9 points)**

(d) By using a loop, identify the smallest window size that allows the imputation of all missing values for variables "Ozone" and "Solar.R" (separately). **(5 points)**

Problem 2 (25 points)

Files `longegfr1.csv` and `longegfr2.csv` (available on Learn) contain information regarding a longitudinal dataset containing records on 250 patients. For each subject, eGFR (estimated glomerular filtration rate, a measure of kidney function) was collected at irregularly spaced time points: variable "fu.years" contains the follow-up time (that is, the distance from baseline to the date when each eGFR measurement was taken, expressed in years).

(a) Merge the two files in an appropriate way into a single dataframe, then order the observations according to subject identifier and follow-up time, and add an assertion that the ordering is correct. **(3 points)**

(b) Compute the average eGFR and length of follow-up for each patient, then tabulate the number of patients with average eGFR in the following ranges: (0, 15], (15, 30], (30, 60], (60, 90], (90, ∞). Count and report the number of patients with missing average eGFR. **(4 points)**

(c) For patients with average eGFR in the (0, 15] range, collect in a dataframe their identifier, sex, age at baseline, average eGFR, maximum follow-up time and number of eGFR measurements taken. **(9 points)**

(c) For patients 3, 37, 162 and 223 (one at a time):

- Plot the patient's eGFR measurements as a function of time.
- Fit a linear regression model and add the regression line to the plot.
- Report the 95% confidence interval for the regression coefficients of the fitted model.
- Using a different colour, plot a second regression line computed after removing the extreme eGFR values (the highest and the lowest value). **(9 points)**

Problem 3 (25 points)

The MDRD4 and CKD-EPI equations are two different ways of estimating the glomerular filtration rate (eGFR) in adults:

$$\text{MDRD4} = 175 \times \text{Scr}^{-1.154} \times \text{Age}^{-0.203}\,[\times 0.742 \text{ if female}]\,[\times 1.212 \text{ if black}],$$

and

$$\text{CKD-EPI} = 141 \times \min(\text{Scr}/\kappa, 1)^{\alpha} \times \max(\text{Scr}/\kappa, 1)^{-1.209} \times 0.993^{\text{Age}}\,[\times 1.018 \text{ if female}]\,[\times 1.159 \text{ if black}],$$

where:

- Scr is serum creatinine (in mg/dL)
- κ is 0.7 for females and 0.9 for males
- α is -0.329 for females and -0.411 for males

(a) Write functions that implement the two formulas with the following signatures:

```
egfr.mdrd4 <- function(scr, age, sex, ethnic)
```

and

```
egfr.ckdepi <- function(scr, age, sex, ethnic)
```

where all arguments of the functions are vectors of the same length (you can assume that `sex` is a factor variable with levels “Male” and “Female”, and `ethnic` is a factor variable with levels “Black” and “Other”). **(14 points)**

(b) For the `scr.csv` dataset available on Learn, compute the eGFR according to the two equations. Report (rounded to the second decimal place) mean and standard deviation of the two eGFR vectors and their Pearson correlation coefficient. **(5 points)**

(c) Produce a scatter plot of the two eGFR vectors, and add vertical and horizontal lines corresponding to median, first and third quantiles. Is the relationship between the two eGFR equations linear? Justify your answer. **(6 points)**

Problem 4 (20 points)

The builtin `infert` dataset contains data from a study of secondary infertility (that is, the inability to get pregnant for a woman who had previously been pregnant, recorded in column "case").

(a) Fit a logistic regression model (M1) to predict secondary infertility using age and parity (number of previous successful pregnancies) as predictors. Use the deviance to judge the goodness of fit of the model and report a p-value to 3 significant figures. **(4 points)**

(b) Fit a second model (M2) by adding the number of spontaneous abortions to the set of predictors used in model M1. Report odds ratio and 95% confidence interval for the spontaneuous abortions variable. Perform a likelihood ratio test to compare model M2 to model M1, and report the p-value for the test. **(4 points)**

(c) Implement a function that computes the binomial log-likelihood according to the formula:

$$\log \mathcal{L}(\beta) = \sum_{i \in \text{case}} \log p_i + \sum_{i \in \text{ctrl}} \log(1 - p_i).$$

The function should have the following signature:

```
loglik.binom <- function(y.obs, y.pred)
```

where `y.obs` is a vector of observed outcomes (with values either 0 or 1 to represent controls and cases, respectively), and `y.pred` is a vector of fitted probabilities learnt from a logistic regression model. Use function `loglik.binom()` to compute deviance and null deviance for model M2. **(6 points)**

(d) Using functions `glm.cv()` and `predict.cv()` from Lab 3, perform 10-folds cross-validation for model M2 (set the random seed to 1 before creating the folds). To evaluate the predictive performance of model M2, use function `loglik.binom()` to compute the log-likelihood of the predicted probabilities for each test fold. Report the sum of the test log-likelihoods over all folds. **(6 points)**