Predictive Modeling Lab 2020-01-27
BSc in Data Science and Engineering

Eduardo García Portugués

We follow the materials at https://bookdown.org/egarpor/PM-UC3M/app-softw.html

Introduction to R

- Simple computations
- Variables and assignment
- Vectors
- Some functions
- Matrices, data frames, and lists
- More on data frames
- Vector-related functions
- Logical conditions and subsetting
- Plotting functions
- Distributions
- Functions
- Control structures

Exercises

- Exercise 1. Compute:
  - \( \frac{e^2 + \sin(2)}{\cos^{-1}\left(\frac{1}{2}\right)} + 2 \). Answer: 2.723274.
  - \( \sqrt{3^{2.5} + \log(10)} \). Answer: 4.22978.
  - \( (26.93 - \log_2(3 + \sqrt{2 + \sin(1)}) \cdot 10^{\tan(1/3)})) \sqrt{3^{2.5} + \log(10)} \). Answer: -3.032108.

- Exercise 2. Do the following:
  - Store -123 in the variable \( y \).
  - Store the log of the square of \( y \) in \( z \).
  - Store \( \frac{y - z}{y + z} \) in \( y \) and remove \( z \).
  - Output the value of \( y \). Answer: 4.366734.

- Exercise 3. Do the following:
  - Create the vector \( x = (1, 7, 3, 4) \).
  - Create the vector \( y = (100, 99, 98, ..., 2, 1) \).
  - Create the vector \( z = (4, 8, 16, 32, 96) \).
  - Compute \( x_2 + y_4 \) and \( \cos(x_3) + \sin(x_2) e^{-y_2} \). Answers: 104 and -0.9899925.
  - Set \( x_2 = 0 \) and \( y_2 = -1 \). Recompute the previous expressions. Answers: 97 and 2.785875.
  - Index \( y \) by \( x + 1 \) and store it as \( z \). What is the output? Answer: \( z \) is \( c(-1, 100, 97, 96) \).

- Exercise 4. Do the following:
  - Compute the mean, median and variance of \( y \). Answers: 49.5, 49.5, 843.6869.
  - Do the same for \( y + 1 \). What are the differences?
What is the maximum of $y$? Where is it placed?
- Sort $y$ increasingly and obtain the 5th and 76th positions. Answer: $c(4, 75)$.
- Compute the covariance between $y$ and $y$. Compute the variance of $y$. Why do you get the same result?

**Exercise 5.** Do the following:
- Create a matrix called $M$ with rows given by $y[3:5], y[3:5]^2$, and $\log(y[3:5])$.
- Create a data frame called myDataFrame with column names “y”, “y2”, and “logy” containing the vectors $y[3:5], y[3:5]^2$ and $\log(y[3:5])$, respectively.
- Create a list, called $l$, with entries for $x$ and $M$. Access the elements by their names.
- Compute the squares of myDataFrame and save the result as myDataFrame2.
- Compute the log of the sum of myDataFrame and myDataFrame2. 

<table>
<thead>
<tr>
<th>#</th>
<th>y</th>
<th>y2</th>
<th>logy</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9.180087</td>
<td>18.33997</td>
<td>3.242862</td>
</tr>
<tr>
<td>2</td>
<td>9.159678</td>
<td>18.29895</td>
<td>3.238784</td>
</tr>
<tr>
<td>3</td>
<td>9.139059</td>
<td>18.25750</td>
<td>3.234656</td>
</tr>
</tbody>
</table>

**Exercise 6.** Do the following:
- Load the faithful dataset into R.
- Get the dimensions of faithful and show beginning of the data.
- Retrieve the fifth observation of eruptions in two different ways.
- Obtain a summary of waiting.

**Exercise 7.** Do the following:
- Create the vector $x = (0.3, 0.6, 0.9, 1.2)$.
- Create a vector of length 100 ranging from 0 to 1 with entries equally separated.
- Compute the amount of zeros and ones in $x <- c(0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 1, 0, 1, 0)$.
- Compute the vector $(0.1, 1.1, 2.1, ..., 100.1)$ in four different ways using seq and rev. Do the same but using : instead of seq. 

**Exercise 8.** Do the following for the iris dataset:
- Compute the subset corresponding to Petal.Length either smaller than 1.5 or larger than 2. Save this dataset as irisPetal.
- Compute and summarize a linear regression of Sepal.Width into Petal.Width + Petal.Length for the dataset irisPetal. What is the $R^2$? Solution: 0.101.
- Check that the previous model is the same as regressing Sepal.Width into Petal.Width + Petal.Length for the dataset iris with the appropriate subset expression.
- Compute the variance for Petal.Width when Petal.Width is smaller or equal that 1.5 and larger than 0.3. Solution: 0.1266541.

**Exercise 9.** Do the following:
- Plot the faithful dataset.
- Add the straight line $y = 110 - 15x$ (red).
- Make a new plot for the function $y = \sin(x)$ (black). Add $y = \sin(2x)$ (red), $y = \sin(3x)$ (blue), and $y = \sin(4x)$ (orange).

**Exercise 10.** Do the following:
- Compute the 90%, 95% and 99% quantiles of a $F$ distribution with $df1 = 1$ and $df2 = 5$. Answer: $c(4.060420, 6.607891, 16.258177)$.
- Plot the distribution function of a $U(0, 1)$. Does it make sense with its density function?
Sample 100 points from a Poisson with \(\lambda = 5\).
- Sample 100 points from a \(U(-1, 1)\) and compute its mean.
- Plot the density of a \(t\) distribution with \(df = 1\) (use a sequence spanning from -4 to 4). Add lines of different colors with the densities for \(df = 5\), \(df = 10\), \(df = 50\), and \(df = 100\). Do you see any pattern?

**Exercise 11.** Do the following:
- Create a function that takes as argument \(n\) and returns the value of \(\sum_{i=1}^{n} i^2\).
- Create a function that takes as input the argument \(N\) and then plots the curve \((n, \sum_{i=1}^{\sqrt{n}} i)\) for \(n = 1, \ldots, N\). **Hint:** use `sapply`.

**Exercise 12.** Do the following:
- Compute \(C_{n \times k} = A_{n \times m} B_{m \times k}\) from \(A\) and \(B\). Use that \(c_{i,j} = \sum_{l=1}^{m} a_{i,l} b_{l,j}\). Test the implementation with simple examples.
- Create a function that samples a \(N(0, 1)\) and returns the first sampled point that is larger than 4.
- Create a function that simulates \(N\) samples from the distribution of \(\max(X_1, \ldots, X_n)\) where \(X_1, \ldots, X_n\) are iid \(U(0, 1)\).

**Exercise 13.** Create a routine for approximating by Monte Carlo integration the following integrals:
- \(\int_{0}^{1} x^2 \, dx = 1/3\).
- \(\int_{1}^{5} \log(x) \, dx = \sin(5) - \sin(1)\).
- \(\int_{-1}^{1} \int_{-1}^{1} xy^2 \, dx \, dy = 0\).
- \(\int_{-1}^{1} \int_{-1}^{1} \sin(xy) \, dx \, dy = 0\).

**Exercise 14.** Create a function that implements the Kolmogorov–Smirnov test as described in the exercise in https://bookdown.org/egarpor/PM-UC3M/app-ext-ht.html

**Exercise 15.** Create a routine that implements the bisection method. It must find the (unique) root \(f(x^*) = 0, x^* \in [0, 1]\) of an arbitrary function \(f : [0, 1] \rightarrow \mathbb{R}\) such that \(\text{sign}(f(0)) \neq \text{sign}(f(1))\). The routine must take as input the function \(f\) and the maximum number of iterations \(N\) of the algorithm.