2nd PC-Assignment Wednesday 20 May from 15:30-17:00 hours

Suppose you wish to approximate the integral $I(a,b) \equiv \int_a^b f(x)\,\mathrm{d}x$.

Consider the following numerical integration algorithm:

$R_{1,1} = (b-a)\frac{f(a)+f(b)}{2}$
for *j=2,3,...,n* **do**
  $h_j = \frac{b-a}{2^{j-1}}$
  $S = 0$
  for *i=1,...,*2^{j-2} **do**
    $S = S + f\left(a + (2i-1)h_j\right)$
  end
  $R_{j1} = \frac{1}{2}R_{j-1,1} + h_j S$
  for *k=2,...,j* **do**
    $R_{jk} = \frac{4^{k-1}R_{j,k-1} - R_{j-1,k-1}}{4^{k-1}-1}$
  end
end

The algorithm generates a triangular set of values:

$$\begin{array}{ccccc} R_{11} & & & & \\ R_{21} & R_{22} & & & \\ R_{31} & R_{32} & R_{33} & & \\ \vdots & & & \ddots & \\ R_{n1} & R_{n2} & \cdots & \cdots & R_{nn} \end{array}$$

and R_{nn} can be used as the final numerical approximation for the analytic integral $I(a,b)$.

You are asked to approximate

$$\int_0^{4\log(2)+2\log(5)} \tfrac{1}{2}e^{-x/2}\,\mathrm{d}x = \tfrac{95}{100}$$

Write one R-script that codes the following:

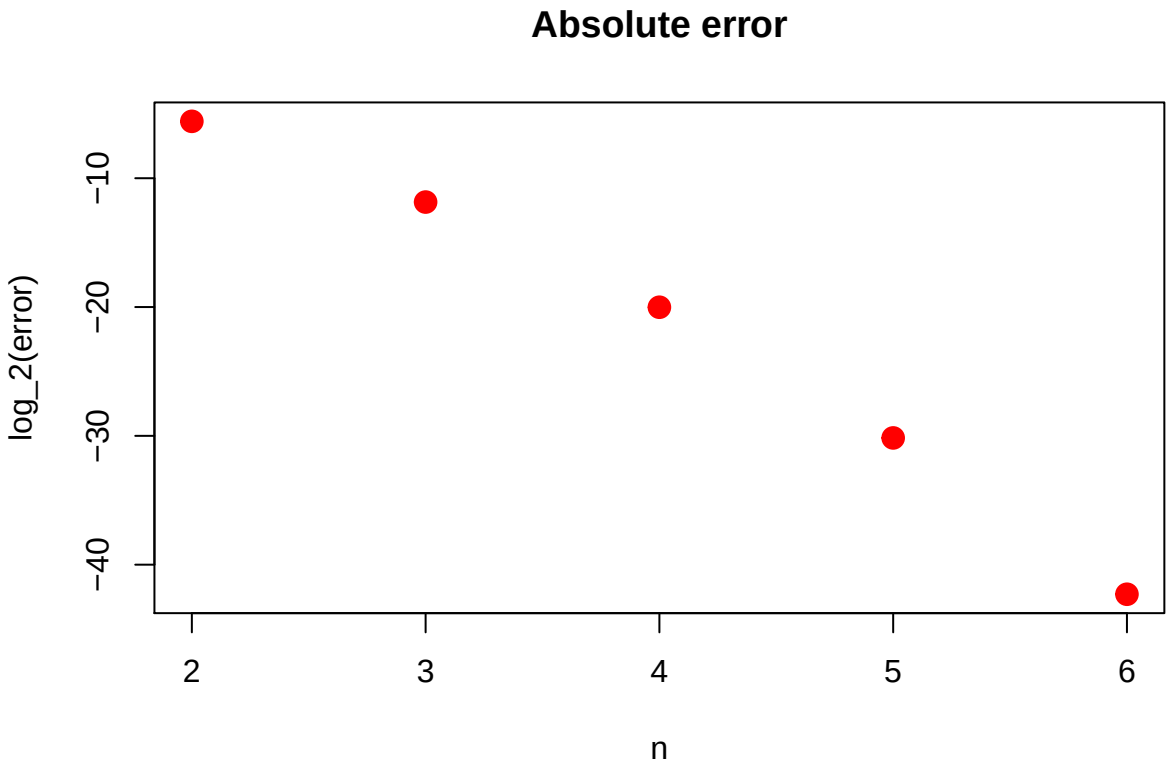
1. `f <- function(x)` that calculates $\frac{1}{2}e^{-x/2}$.
   - The function $f(x)$ is the probability density function of $X \sim \chi^2(2)$.
   - The upper bound $4\log(2) + 2\log(5)$ is the 95th percentile.
2. `NumInt <- function(f,a,b,n)` that implements the algorithm shown above for interval $[a,b]$ and n iterations.
   - You may use an $n \times n$ matrix for R and the function should return R_{nn}

3. Generate a table for $n = 2, ..., 6$ that shows on one line (i) n (ii) the true value of the integral [10 decimals] (iii) its approximation [10 decimals] (iv) the absolute error [13 decimals]
4. Generate a plot with n on the horizontal axis and $\log_2(error)$ on the vertical axis, i.e. the logarithm of the error in base 2 [with red solid circles].
5. Using $n = 6$, determine the value for b such that $\int_0^b \frac{1}{2} e^{-x/2}\, \mathrm{d}x = 99/100$. Use R's build-in function `uniroot()`. Consult its help page for more info about this function.

    - This value of b is the 99th percentile.
    - Display this value in 4 decimals (see Output).
    - Consult the help file to see how to use and call `uniroot`.
    - Use the interval $[8, 10]$ to be searched for the root.
    - Note that `uniroot`'s help file reads: `f` will be called as `f(x, ...)` for a numeric value of `x`. Hence, it might be handy to use the following function:
      `NumInt2 <- function(b,a,f,n){ NumInt(f,a,b,n)-0.99 }`
      in which also the order of the arguments is different from `NumInt`!

6. **General remarks**

    - Put everything in one file and do not make use of other functions outside the script.
    - Download the template from Canvas and enter your name and student number at the first line.
    - When finished, upload your file (PC2.R) to canvas. In case of emergency (Canvas not available), you can also email the file to uvapna@gmail.com
    - For full credits, your output needs to be the same as shown on the next page (apart from small rouding-off errors):




```
n=2: True area=0.9500000000  Approx=0.9708305483   Error=0.0208305482757
n=3: True area=0.9500000000  Approx=0.9502707543   Error=0.0002707543032
n=4: True area=0.9500000000  Approx=0.9500009430   Error=0.0000009430111
n=5: True area=0.9500000000  Approx=0.9500000008   Error=0.0000000008335
n=6: True area=0.9500000000  Approx=0.9500000000   Error=0.0000000000002
For n=6 and b=9.2103: I(0,b)=0.99.
```