Introduction to Financial Engineering (HW5)
Organizer: Albert Ferreiro-Castilla
room: PC email: `aferreiro@mat.uab.cat`

- Please follow the guidelines for assignments given in the Module Handbook.
- All programs should be written in R (compilable without errors or warnings).
- You should submit a write-up (.pdf) of the program as well as the source code (.r).
- File names should be as `yoursurname_yourname_HW5.extension`
- You should submit via moddle.
- Deadline: 24th November 2023 at 10am.

1. We will build up a Brownian motion simulator.

   (a) **Coin tossing sample path**: Build up a function which takes a parameter N and outputs a vector, that when plotted is the path of the profit and loss of a tossing game. Have in mind the following rules:

   - The game performs N bets in the time interval $[0, 1]$.
   - The profit or loss for each bet is $\frac{1}{\sqrt{N}}$.

   i. Plot a sample path for the following values of $N = 5, 10, 50, 100, 10000$.

   (b) **Sample path distribution**: Code a functions which takes the values N and m and calls m times the previous function with the parameter N. Make this function to keep the last value for each sample path.

   i. Call the previous function for the pairs $(N, m) = (100, 100), (1000, 1000), (10000, 10000)$. Plot the density function of the resulting time series and perform a normality test.

2. We will program a Monte Carlo algorithm to compute option prices.

   (a) **Building a sample path under the Black-Scholes model**: Write a function that samples a path from the Black-Scholes stochastic differential equation:

   $$dS_t = rS_t dt + \sigma S_t dW_t$$

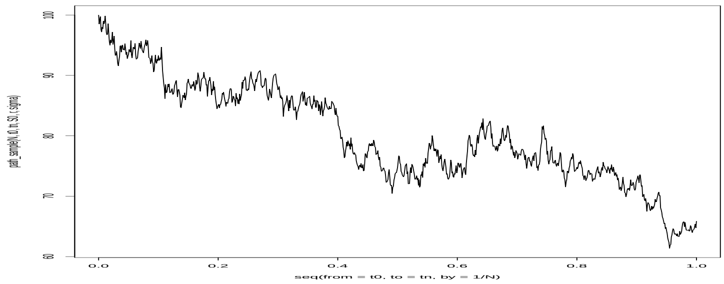
   using an Euler discretization. The function should take the following variables

   - N: number of time steps in the sample.
   - $t0$: initial time expressed as a fraction of year.
   - tn: expire time expressed as a fraction of year.
   - $S0$: initial price of the stock.
   - r: annualized risk free rate expressed as percentage.
   - σ: annualized volatility expressed as percentage.

   Write a function with the following prototype that outputs a path of the stochastic process:

```
path_sample=function(N,t0,tn,S0,r,sigma)
```

   The following is a realization of such path:




```
N=1000
t0=0
tn=1
S0=100
r=0.01
sigma=0.30

plot(seq(from = t0, to = tn, by = 1/N),path_sample(N,t0,tn,S0,r,sigma),type="l")
```

- Explore the path properties depending on the variable r and σ.

(b) **Payoff function**: Implement the following function that evaluates a payoff for a CALL option with a given strike K.

```
payoff_function=function(S,K){
  return(max(S-K,0))
}
```

(c) **Monte Carlo algorithm**: Implement a Monte Carlo algorithm as described in the lecture notes. The function should take a parameter M determining the number of sample paths as well as a function determining the payoff of the option. It should also take all parameters from the function $path\_sample$ in order to be able to call it within the procedure. Use the following prototype:

```
monte_carlo(M,N,t0,tn,S0,r,sigma,payoff_function,K)
```

(d) **Put option**: Modify the function $payoff\_function$ and give the price of a PUT option for 1 year, with a 5% interest rate and 40% volatility. The stock currently trades at 90 EUR and the strike of the PUT option is 75.