```
## Registering fonts with R
## Warning:  package 'ggplot2' was built under R version 3.3.3
## Warning:  package 'xtable' was built under R version 3.3.3
## Warning:  package 'dplyr' was built under R version 3.3.3
##
## Attaching package:  'dplyr'
## The following objects are masked from 'package:plyr':
##
##    arrange, count, desc, failwith, id, mutate, rename, summarise,
##    summarize
## The following objects are masked from 'package:stats':
##
##    filter, lag
## The following objects are masked from 'package:base':
##
##    intersect, setdiff, setequal, union
```

1 Description

The symetric random walk will be described in this document (Mt). it covers the theory of "Stochastic Calculus for finance" Tome 2 chapter 3 section 1.

The construction of the random walk depend on the evolution of a random variable X_i. The previous RV can take two value at each time, like tossing a coin. X_i can take the value 1 or -1.

$$X_i = \begin{cases} 1 \\ -1 \end{cases} \tag{1}$$

The Symetric Random Walk is constructed by summing up the different outcome of the random variable X_i from k experiments:

$$M_k = \sum_{j=1}^{k} X_j \tag{2}$$

In the following lines of code, X_i is randomly difined. The variable k ensure to have a sufficent number of periods to further generate the scaled random walk. It refers to the k of equation 2. p and q are the probability measure, respectively p chance to get value 1 and q chance to get -1 from random variable X_i.

After creating the random variable X_i it suffices to add up all the differente output we get from time 1 up to k to get a specific Symetric Random Walk.

The following outcome present a randomly generated 300 steps symmetric random walk.

Table 1: 300 steps Symmetric Random Walk

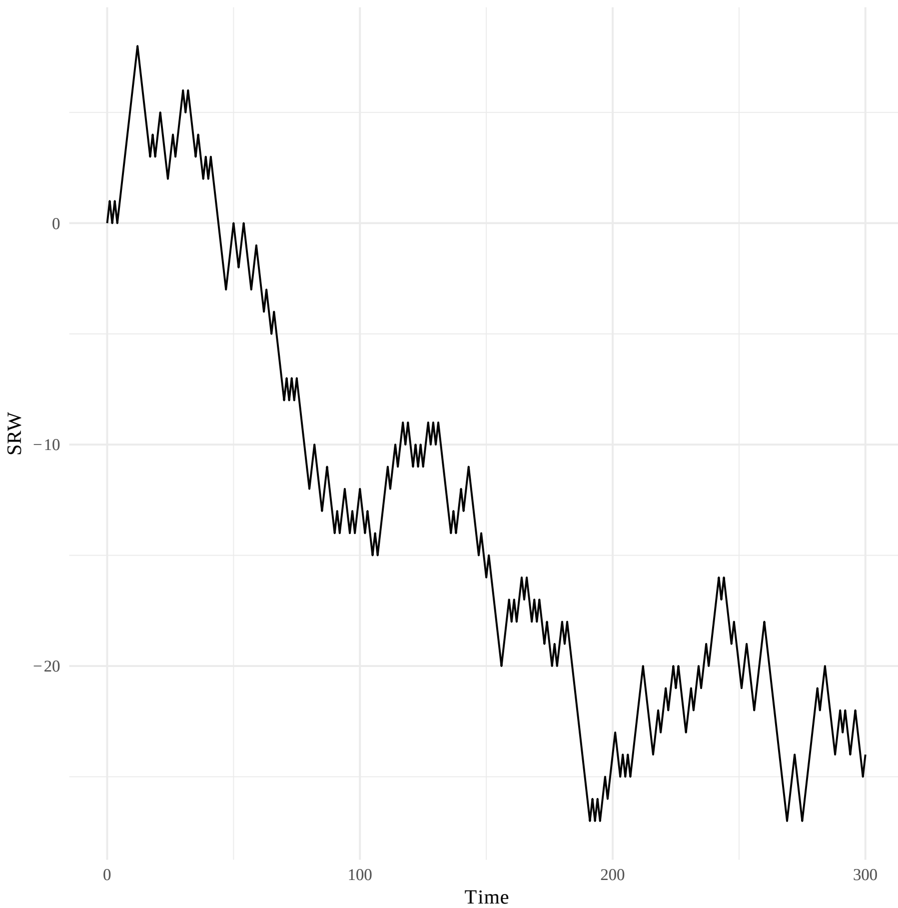

Figure 1: Symmetric Random Walk

```r
# Because squared matrix dim(y) = dim(x):
dim_x <- dim_y <- 1:(k + 1) # from 1 to k+1 because we start to time zero nonrandom which e
o <- outer(dim_x, dim_y, FUN=function(r,c){(r-c) + (1-c)} )
to <- t(o)
subset <- upper.tri(to, diag = T)
Mk <- to * subset
colnames(Mk) <- paste0("F(", 1:ncol(Mk) - 1, ")")
# Transform Mk to better suit the table:
Mk_print <- apply(Mk, 2, as.character)
Mk_print[Mk_print == '0'] <- ''
Mk.tab <- xtable(Mk_print[1:10, 1:10], digits = 0, format = "latex")
align(Mk.tab) <- rep("r", 11)
```

```
Mk.tab
```

	F(0)	F(1)	F(2)	F(3)	F(4)	F(5)	F(6)	F(7)	F(8)	F(9)
1		1	2	3	4	5	6	7	8	9
2		-1		1	2	3	4	5	6	7
3			-2	-1		1	2	3	4	5
4				-3	-2	-1		1	2	3
5					-4	-3	-2	-1		1
6						-5	-4	-3	-2	-1
7							-6	-5	-4	-3
8								-7	-6	-5
9									-8	-7
10										-9