Risk_Factors_Simulation

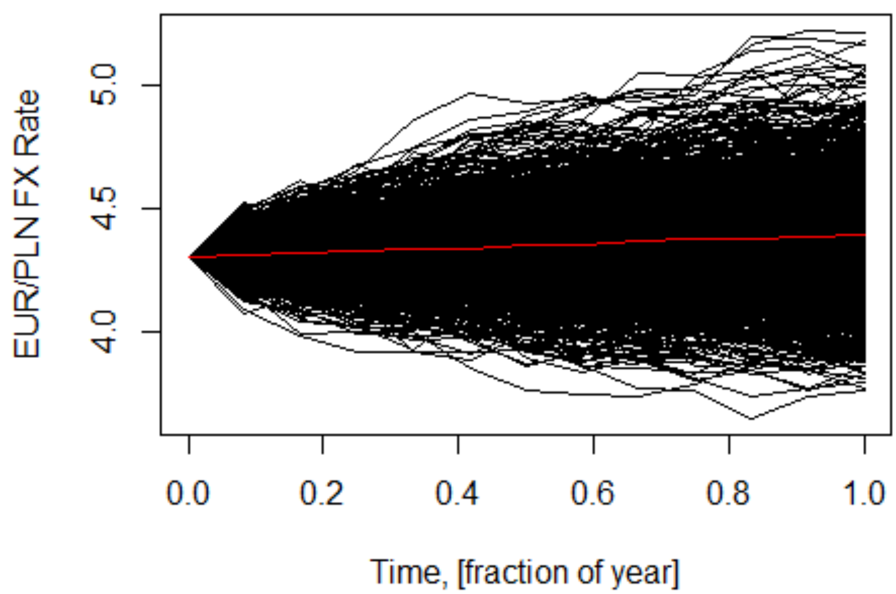
Input data

```
# Specify calibration time horizon
start_date <- "20160101"
# We assume that today is 31 Dec 2018
today <- "20181231"
# Input historical EUR/PLN data
FX <- stooq(ticker = "EURPLN", from = start_date, to = today, interval = "d")
# Input other historical data
WIBOR_3M <- stooq(ticker = "plopln3m", from = start_date, to = today,
interval = "d")
EURIBOR_3M <- stooq(ticker = "EUOEUR3M", from = start_date, to = today,
interval = "d")

# # # It is possible to save the data on your hard drive in order not to
exceed daily limit of queries from stooq.pl
# # save(FX, file = "FX_test.RData")
# # save(WIBOR_3M, file = "WIBOR_3M_test.RData")
# # save(EURIBOR_3M, file = "EURIBOR_3M_test.RData")
# load("EURIBOR_3M_test.RData")
# load("WIBOR_3M_test.RData")
# load("FX_test.RData")
```

FX Forward Risk Factor simulation

```
# maturity of the FX Forward instrument (in years)
T <- 1
# timestep of simulation is monthly (in years)
dt <- 1/12
# number of simulations
M <- 1000

# risk-free domestic interest rate (WIBOR 3M)
r_d <- WIBOR_3M[WIBOR_3M$Date=="2018-12-31",2]/100
# risk-free foreign interest rate (EURIBOR 3M)
r_f <- EURIBOR_3M[EURIBOR_3M$Date=="2018-12-31",2]/100
# drift parameter
mu <- r_d-r_f
# Calibration to historical volatility
vol <- sd(diff(log(FX[,2])))*sqrt(252)
# current EUR/PLN rate
FX0 <- 4.3
# Simulation matrix
FX_sim <- matrix(NA, ncol = M, nrow = T/dt+1)
```

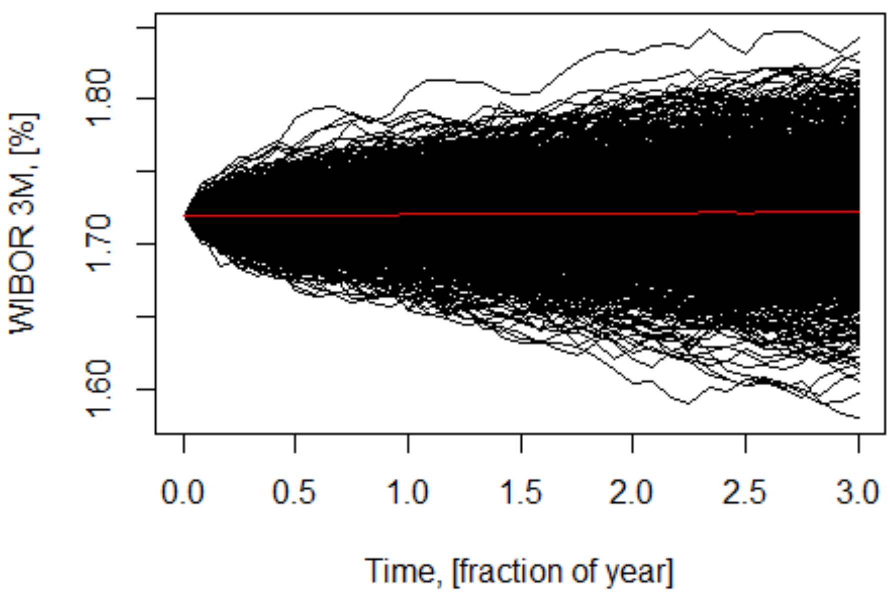
```
FX_sim[1,] <- FX0

for (n in 1:M) {
  for (i in 2:(T/dt+1)) {
    FX_sim[i,n] <- FX_sim[i-1, n]*exp((mu-(vol^2)/2)*dt +
vol*sqrt(dt)*rnorm(1))
  }
}
# Plot the output
ts_FX_sim <- ts(data = FX_sim, start = 0, end = T, frequency = 1/dt, deltat =
dt)
plot(ts_FX_sim, plot.type="single", ylab="EUR/PLN FX Rate", xlab="Time,
[fraction of year]")
lines(as.vector(time(ts_FX_sim)), rowMeans(ts_FX_sim), col='red')
```



IRS Risk Factor Simulation

```
# maturity of the IRS instrument (in years)
T <- 3
# timestep of simulation is monthly (in years)
dt <- 1/12
# number of simulations
M <- 1000
# Interest rate as of today. Starting point for simuations.
X0 <- WIBOR_3M[WIBOR_3M$Date=="2018-12-31", 2]
# drift parameter historical calibration
```

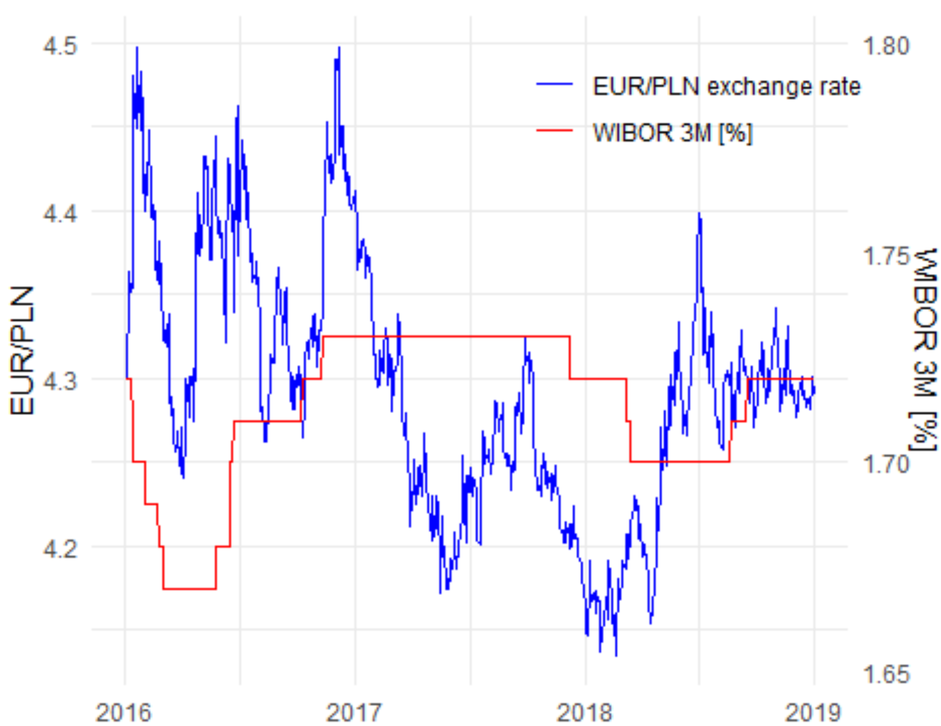
```r
mu <- mean(diff(WIBOR_3M[,2]))*252
# volatility parameter cistorical calibration
vol <- sd(diff(WIBOR_3M[,2]))*sqrt(252)
# Simulation matrix
r_sim <- matrix(NA, ncol = M, nrow = T/dt+1)
r_sim[1,] <- X0
for (n in 1:M) {
  for (i in 2:(T/dt+1)) {
    r_sim[i,n] <- r_sim[i-1, n] + mu*dt + vol*dt^0.5*rnorm(1)
  }
}
# Plot the output
ts_r_sim <- ts(data = r_sim, start = 0, end = T, frequency = 1/dt, deltat =
dt)
plot(ts_r_sim, plot.type="single", ylab="WIBOR 3M, [%]", xlab="Time,
[fraction of year]")
lines(as.vector(time(ts_r_sim)), rowMeans(ts_r_sim), col='red')
```



CIRS Risk Factor Simulation

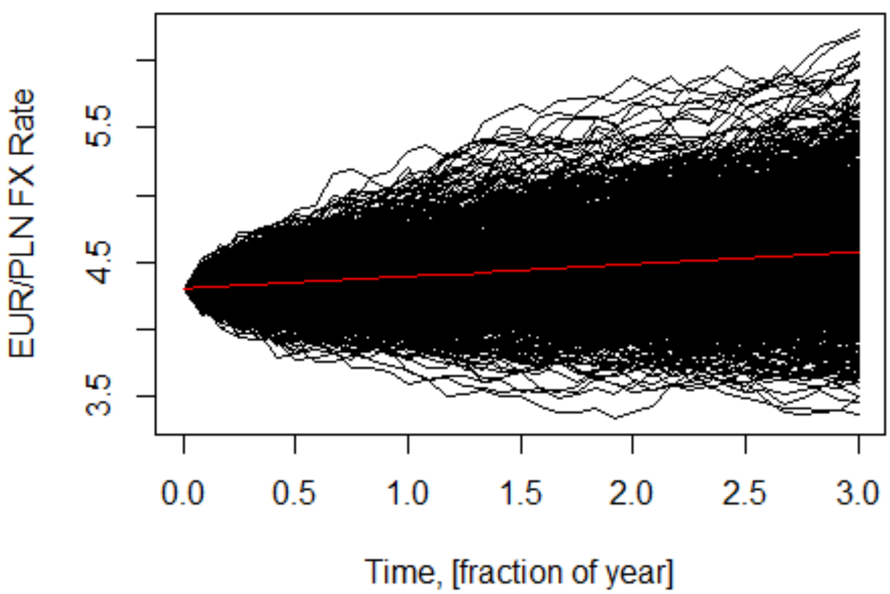
```r
# maturity of the CIRS instrument (in years)
T <- 3
# timestep of simulation is monthly (in years)
dt <- 1/12
# number of simulations
M <- 1000
```

```r
# Handling correlation ====

# Making sure that the observation dates are matching
FX_filt <- FX$Date %in% WIBOR_3M$Date
WIBOR_filt <- WIBOR_3M$Date %in% FX$Date
# Plot of correlated random variables
plt <- data.frame(FX[FX_filt,], WIBOR_3M = WIBOR_3M[WIBOR_filt, 2])
library(ggplot2)

## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang

ggplot(plt, aes(x=Date)) +
  geom_line(aes(y=EURPLN, colour="EUR/PLN exchange rate")) +
  geom_line(aes(y=WIBOR_3M*2.5, colour="WIBOR 3M [%]")) +
  scale_y_continuous(sec.axis = sec_axis(~./2.5, name = "WIBOR 3M [%]")) +
  scale_colour_manual(values = c("blue", "red")) +
  labs(y = "EUR/PLN",
       x = "",
       colour = "") +
  theme_minimal() +
  theme(legend.position = c(0.8, 0.9))
```

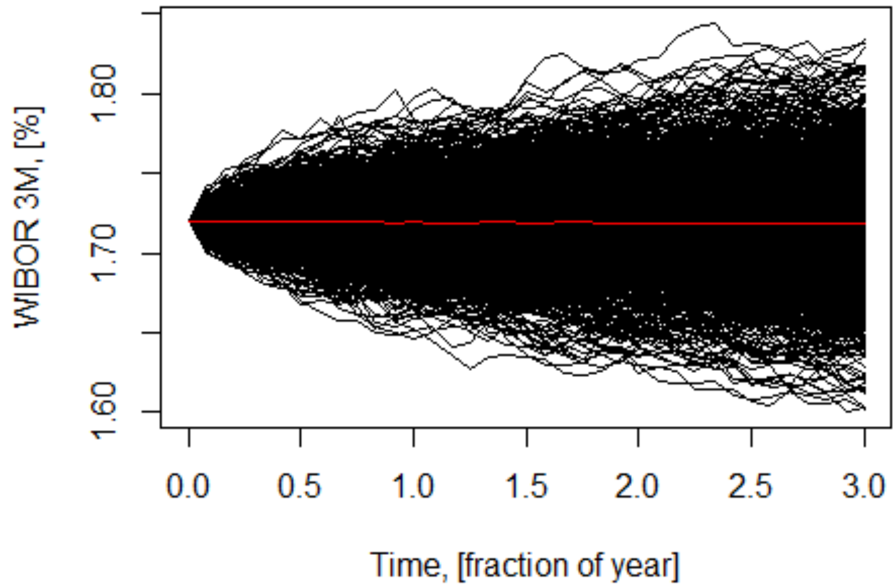


```r
# Risk factors are slightly negatively correlated
rho <- cor(FX[FX_filt,2], WIBOR_3M[WIBOR_filt, 2])

# # Example of Cholesky Factorization
# fact <- t(chol(matrix(c(1,rho,rho,1),nrow = 2)))
# eps_FX <- rnorm(M)
# eps_WIBOR <- rnorm(M)
# eps <- rbind(eps_FX, eps_WIBOR)
# z <- fact %*% eps
# z_FX <- z[1,]
# z_WIBOR <- z[2,]
# plot(z_FX, type = 'l')
# lines(z_WIBOR, col='red')
# cor(z_FX,z_WIBOR)


# EUR/PLN simulation inputs ====
# risk-free domestic interest rate (WIBOR 3M)
r_d <- WIBOR_3M[WIBOR_3M$Date=="2018-12-31",2]/100
# risk-free foreign interest rate (EURIBOR 3M)
r_f <- EURIBOR_3M[EURIBOR_3M$Date=="2018-12-31",2]/100
# drift parameter
FX_mu <- r_d-r_f
# Calibration to historical volatility
FX_vol <- sd(diff(log(FX[,2])))*sqrt(252)
# current EUR/PLN rate
FX0 <- 4.3
# Simulation matrix
FX_sim <- matrix(NA, ncol = M, nrow = T/dt+1)
FX_sim[1,] <- FX0

# WIBOR 3M simulation inputs ====
# Interest rate as of today. Starting point for simuations.
X0 <- WIBOR_3M[WIBOR_3M$Date=="2018-12-31", 2]
# drift parameter historical calibration
WIBOR_mu <- mean(diff(WIBOR_3M[,2]))*252
# volatility parameter historical calibration
WIBOR_vol <- sd(diff(WIBOR_3M[,2]))*sqrt(252)
# Simulation matrix
WIBOR_sim <- matrix(NA, ncol = M, nrow = T/dt+1)
WIBOR_sim[1,] <- X0

# The Common Loop ====
for (n in 1:M) {
  for (i in 2:(T/dt+1)) {
    # Cholesky factorization
    eps_FX <- rnorm(1)
    eps_WIBOR <- rnorm(1)
    z_FX <- eps_FX
```

```
    z_WIBOR <- eps_FX*rho+eps_WIBOR*sqrt(1-rho^2)

    FX_sim[i,n] <- FX_sim[i-1, n]*exp((FX_mu-(FX_vol^2)/2)*dt + 
FX_vol*sqrt(dt)*z_FX)
    WIBOR_sim[i,n] <- WIBOR_sim[i-1, n] + WIBOR_mu*dt + 
WIBOR_vol*sqrt(dt)*z_WIBOR
  }
}

# Plot the output for FX simulation
ts_FX_sim <- ts(data = FX_sim, start = 0, end = T, frequency = 1/dt, deltat = 
dt)
plot(ts_FX_sim, plot.type="single", ylab="EUR/PLN FX Rate", xlab="Time, 
[fraction of year]")
lines(as.vector(time(ts_FX_sim)), rowMeans(ts_FX_sim), col='red')
```




```
# Plot the output for WIBOR 3M simulation
ts_WIBOR_sim <- ts(data = WIBOR_sim, start = 0, end = T, frequency = 1/dt, 
deltat = dt)
plot(ts_WIBOR_sim, plot.type="single", ylab="WIBOR 3M, [%]", xlab="Time, 
[fraction of year]")
lines(as.vector(time(ts_WIBOR_sim)), rowMeans(ts_WIBOR_sim), col='red')
```




```
xyz <- mean(diag(cor(apply(ts_WIBOR_sim, MARGIN = 2, diff),
apply(apply(ts_FX_sim, MARGIN = 2, log), MARGIN = 2, diff))))
print(paste0("Average correlation between differences of simulated WIBOR 3M
and log-returns of FX rate converges to correlation between the historical
data of risk factors and is equal to: ", round(xyz, 3)))

## [1] "Average correlation between differences of simulated WIBOR 3M and
log-returns of FX rate converges to correlation between the historical data
of risk factors and is equal to: -0.283"
```