#-------------------------------------

# Simulation

#-------------------------------------

# number of months to simulate

n.obs = 12 \* n.obs

# monthly Investment and Inflation assumptions

monthly.mean.return = annual.mean.return / 12

monthly.ret.std.dev = annual.ret.std.dev / sqrt(12)

monthly.inflation = annual.inflation / 12

monthly.inf.std.dev = annual.inf.std.dev / sqrt(12)

# simulate Returns

monthly.invest.returns = matrix(0, n.obs, n.sim)

monthly.inflation.returns = matrix(0, n.obs, n.sim)

monthly.invest.returns[] = rnorm(n.obs \* n.sim, mean = monthly.mean.return, sd = monthly.ret.std.dev)

monthly.inflation.returns[] = rnorm(n.obs \* n.sim, mean = monthly.inflation, sd = monthly.inf.std.dev)

# simulate Withdrawals Time to Maturity

# <https://dqydj.com/scripts/automated/scripts/taylor_rule.html>

# <http://vazoof.com/the-5-components-of-interest-rates/>

# Five Components of Interest Rates

#Real-Risk Free Rate, Expected Inflation, Default Risk Premium, Liquidity Premium, Maturity Premium

#Taylor Rule for short term interest rates

nav = matrix(Short.Term.Interest.Rate, n.obs + 1, n.sim)

for (j in 1:n.obs) {

nav[j + 1, ] = nav[j, ] \* (1 + monthly.invest.returns[j, ] - monthly.inflation.returns[j, ]) - Time.to.Maturity

}

# once nav is below 0 => run out of money

nav[ nav < 0 ] = NA

# convert to millions

nav = nav / 1000000

return(nav)

}

plot\_nav <- function(nav) {

layout(matrix(c(1,2,1,3),2,2))

palette(c("black", "grey50", "grey30", "grey70", "#d9230f"))

# plot all scenarios

matplot(nav,

type = 'l', lwd = .25, lty = 1, col = 1:5,

xlab = 'Months', ylab = 'Interest Rate',

main = 'Interest Rates')

# plot % of scenarios that are still paying

p.alive = 1 - rowSums(is.na(nav)) / ncol(nav)

plot(100 \* p.alive, las = 1, xlab = 'Months', ylab = 'Percentage Paying',

main = 'Percentage of Paying Scenarios', ylim=c(0,100))

grid()

last.period = nrow(nav)

# plot distribution of final wealth

final.nav = nav[last.period, ]

final.nav = final.nav[!is.na(final.nav)]

if(length(final.nav) == 0) return()

plot(density(final.nav, from=0, to=max(final.nav)), las = 1, xlab = 'Final Capital',

main = paste0('Distribution of Final Capital\n', 100 \* p.alive[last.period], '% are still paying'))

grid()

}