Metaheuristics_mmarjan1

Marianne Marjanovic

November 7, 2018

```
knitr::opts_chunk$set(echo = TRUE)


#Load Libraries
library('ggplot2')
library('quantmod')
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
## 
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
## 
##     as.Date, as.Date.numeric
```

```
## Loading required package: TTR
```

```
## Version 0.4-0 included new data defaults. See ?getSymbols.
```

```
library('GA')
```

```
## Loading required package: foreach
```

```
## Loading required package: iterators
```

```
## Package 'GA' version 3.1.1
## Type 'citation("GA")' for citing this R package in publications.
```

```
library('GenSA')
library('knitr')
```

Problem 1

```r
#Feeding real-time stock data into the system

tickers <- c("F", "MCD", "NKE", "TGT", "GIS")
getSymbols(tickers, from = "2008-12-01", to = "2018-10-31", src =  "yahoo", adjust =
TRUE)
```

```
## 'getSymbols' currently uses auto.assign=TRUE by default, but will
## use auto.assign=FALSE in 0.5-0. You will still be able to use
## 'loadSymbols' to automatically load data. getOption("getSymbols.env")
## and getOption("getSymbols.auto.assign") will still be checked for
## alternate defaults.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.warning4.0"=FALSE). See ?getSymbols for details.
```

```
##
## WARNING: There have been significant changes to Yahoo Finance data.
## Please see the Warning section of '?getSymbols.yahoo' for details.
##
## This message is shown once per session and may be disabled by setting
## options("getSymbols.yahoo.warning"=FALSE).
```

```
## [1] "F"   "MCD" "NKE" "TGT" "GIS"
```

```r
P <- NULL
for(ticker in tickers) {
 tmp <- Cl(to.monthly(eval(parse(text = ticker))))
 P <- cbind(P, tmp)
}
colnames(P) <- tickers
R <- diff(log(P))
R <- R[-1,]
mu <- colMeans(R)
sigma <- cov(R)
library("PerformanceAnalytics")
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
##
##     legend
```

```
pContribCVaR <- ES(weights = rep(0.2, 5), method = "gaussian", portfolio_method = "com
ponent", mu = mu, sigma = sigma)$pct_contrib_ES
obj <- function(w) {
 fn.call <<- fn.call + 1
 if (sum(w) == 0) { w <- w + 1e-2 }
 w <- w / sum(w)
 CVaR <- ES(weights = w, method = "gaussian", portfolio_method = "component", mu = m
u, sigma = sigma)
 tmp1 <- CVaR$ES
 tmp2 <- max(CVaR$pct_contrib_ES - 0.225, 0)
 out <- tmp1 - 1e+3 * tmp2
 return(out)
}
obj1 <- function(w) {
 fn.call <<- fn.call + 1
 if (sum(w) == 0) { w <- w + 1e-2 }
 w <- w / sum(w)
 CVaR <- ES(weights = w, method = "gaussian", portfolio_method = "component", mu = m
u, sigma = sigma)
 tmp1 <- CVaR$ES
 tmp2 <- max(CVaR$pct_contrib_ES - 0.225, 0)
 out1 <- tmp1 + 1e+3 * tmp2
 return(out1)
}
```

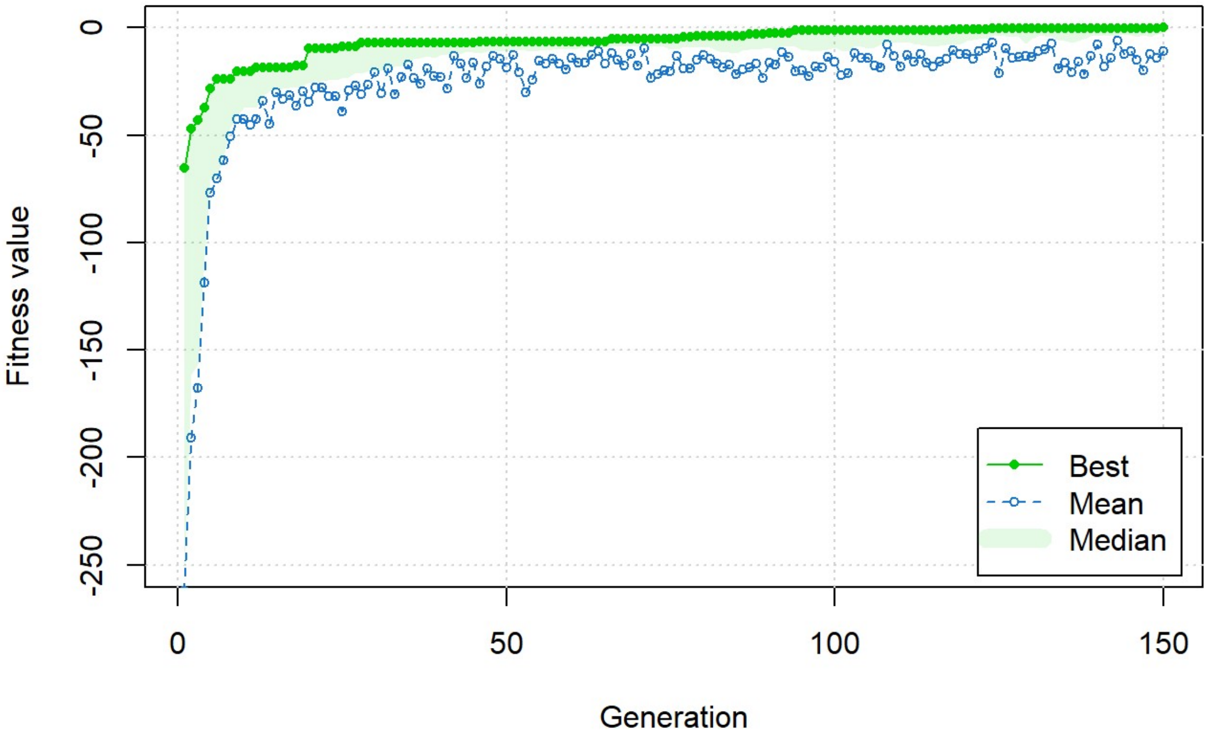
```
#Applying GA to assign weights for optimum portfolio using min objective
# Where x1 = BAC, x2 = F, x3 = GE, x4 = AAPL and x5 = WMT
# Solution is the weights assigned to each of the stocks for optimum portfolio
set.seed(1234)
fn.call <<- 0
gap <- ga(type = "real-valued",fitness=obj,lower=rep(0,5),upper=rep(1,5), popSize = 5
0,maxiter = 150, pcrossover = 0.75, pmutation = 0.1)
nsol <- gap@solution
nsol <- nsol / sum(nsol)
fn.call.gap <- fn.call
nsol
```

```
##             x1        x2        x3        x4        x5
## [1,] 0.1398025 0.3042789 0.1391726 0.2236401 0.1931059
```

```
summary(gap)
```

```
## -- Genetic Algorithm -------------------
##
## GA settings:
## Type                  =  real-valued
## Population size       =  50
## Number of generations =  150
## Elitism               =  2
## Crossover probability =  0.75
## Mutation probability  =  0.1
## Search domain =
##       x1 x2 x3 x4 x5
## lower  0  0  0  0  0
## upper  1  1  1  1  1
##
## GA results:
## Iterations             = 150
## Fitness function value = -0.07122402
## Solution =
##              x1        x2        x3       x4       x5
## [1,] 0.4463806 0.9715436 0.4443694 0.714069 0.616575
```

```
plot(gap)
```



```
#Applying GenSA to assign weights to the stocks for optimum portfolio using max object
ive
set.seed(1234)
fn.call <<- 0
out.GenSA <- GenSA(fn = obj1, lower = rep(0, 5), upper = rep(1, 5), control = list(smo
oth = FALSE, max.call = 3000))
fn.call.GenSA <- fn.call
out.GenSA$value
```

```
## [1] 8.52277
```

```
out.GenSA$counts
```

```
## [1] 3000
```

```
cat("GenSA call functions", fn.call.GenSA, "times.\n")
```

```
## GenSA call functions 3000 times.
```

```
wstar.GenSA <- out.GenSA$par
wstar.GenSA <- wstar.GenSA / sum(wstar.GenSA)
rbind(tickers, round(100 * wstar.GenSA, 2))
```

```
##         [,1] [,2]    [,3]    [,4]    [,5]   
## tickers "F"  "MCD"   "NKE"   "TGT"   "GIS"  
##         "15" "24.86" "15.01" "24.12" "21.02"
```

```
100 * (sum(wstar.GenSA * mu) - mean(mu))
```

```
## [1] 0.003838162
```

```
wstar.GenSA
```

```
## [1] 0.1499819 0.2485596 0.1500737 0.2412150 0.2101697
```

```
#comparing the weights as per both the algorithms
b <- matrix(c(nsol, wstar.GenSA), nrow = 5, ncol = 2)
rownames(b) = c("F", "MCD", "NKE", "TGT", "GIS")
colnames(b) = c("GAPortfolio", "GenSAPortfolio")
b
```

```
##     GAPortfolio GenSAPortfolio
## F     0.1398025      0.1499819
## MCD   0.3042789      0.2485596
## NKE   0.1391726      0.1500737
## TGT   0.2236401      0.2412150
## GIS   0.1931059      0.2101697
```

```
TickerSymbol <- (c("F","MCD","NKE","TGT","GIS","______","TOTAL"))
Company <- (c("Ford","McDonalds","Nike","Target","General Mills","",""))
GA_percent  <- (c(11,21,19,18,31,"_____ ",100))
GenSA_percent <- (c(6,31,18,17,28,"_____ ",100))
TData <- data.frame(TickerSymbol, Company, GA_percent, GenSA_percent)
TData
```

```
##   TickerSymbol       Company GA_percent GenSA_percent
## 1            F          Ford         11             6
## 2          MCD     McDonalds         21            31
## 3          NKE          Nike         19            18
## 4          TGT        Target         18            17
## 5          GIS General Mills         31            28
## 6       ______                   _____         _____ 
## 7        TOTAL                      100           100
```

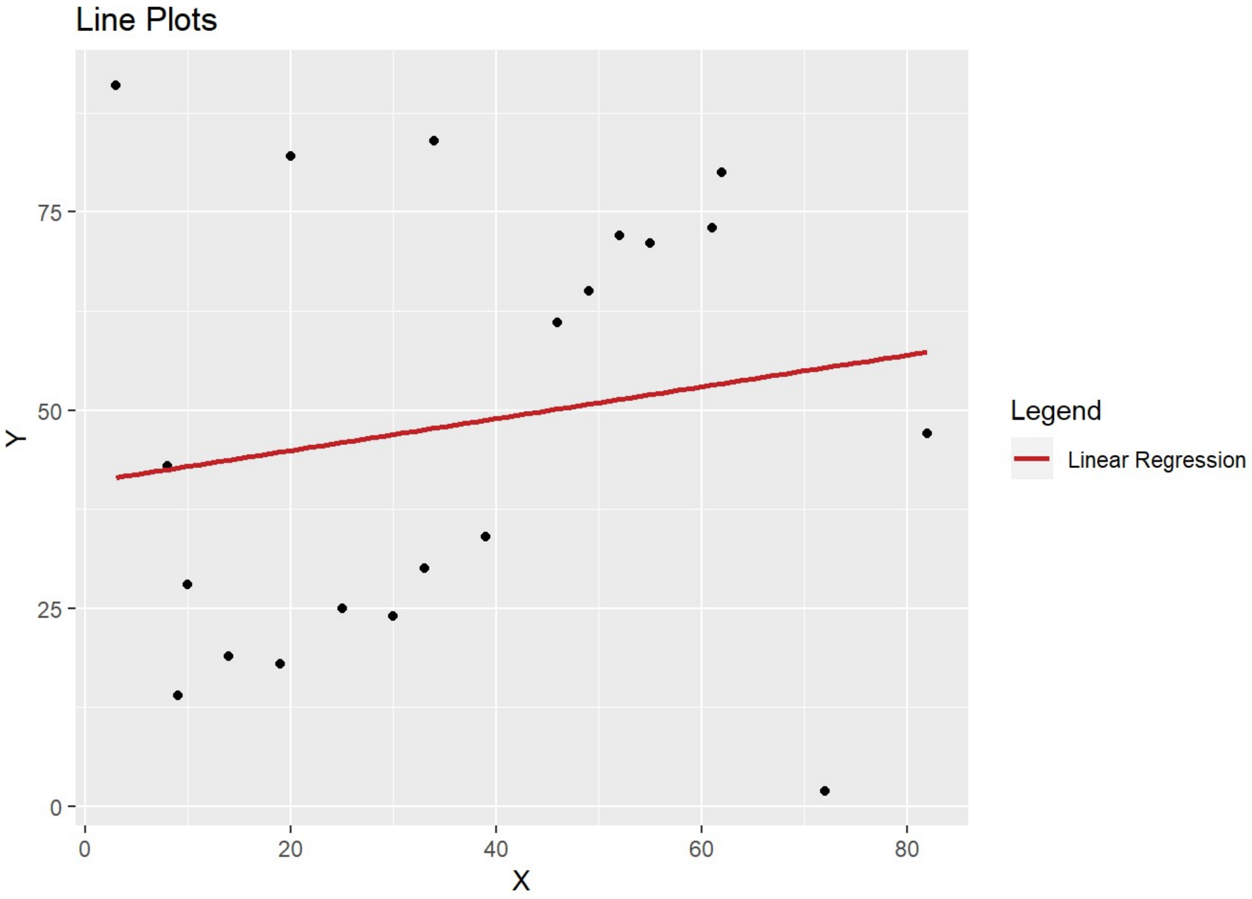
Problem 2

```
X <- (c(25,8,9,19,34,46,72,39,3,10,61,82,14,20,52,49,62,55,30,33))
Y <- (c(25,43,14,18,84,61,2,34,91,28,73,47,19,82,72,65,80,71,24,30))
data2 <- data.frame(X,Y)

lm(formula = Y ~ X)
```

```
## 
## Call:
## lm(formula = Y ~ X)
## 
## Coefficients:
## (Intercept)            X  
##      40.885        0.201
```

```
p <- ggplot(data=data2,aes(X,Y)) + geom_point()
p <- p + stat_smooth(method = "lm", aes(color="Linear Regression"),
        se = FALSE,
        size = 1) +
  scale_color_manual(name="Legend",values= c("Linear Regression"="#C42126")) + ggtitle
("Line Plots")
p
```


Line Plots
Y
X
Legend
Linear Regression