HW4Q4.R

board

Mon Nov 14 16:32:24 2016

library(fExtremes)

## Warning: package 'fExtremes' was built under R version 3.1.3

## Loading required package: timeDate

## Warning: package 'timeDate' was built under R version 3.1.3

## Loading required package: timeSeries

## Warning: package 'timeSeries' was built under R version 3.1.3

## Loading required package: fBasics

## Warning: package 'fBasics' was built under R version 3.1.3

##   
##   
## Rmetrics Package fBasics  
## Analysing Markets and calculating Basic Statistics  
## Copyright (C) 2005-2014 Rmetrics Association Zurich  
## Educational Software for Financial Engineering and Computational Science  
## Rmetrics is free software and comes with ABSOLUTELY NO WARRANTY.  
## https://www.rmetrics.org --- Mail to: info@rmetrics.org  
## Loading required package: fGarch

## Warning: package 'fGarch' was built under R version 3.1.3

## Loading required package: fTrading

## Warning: package 'fTrading' was built under R version 3.1.3

setwd("C:/Users/board/Desktop/Kaggle/Scenario Analysis")  
data <- read.csv("SP\_500\_Log\_Returns\_19600610\_19871016.csv")  
  
# Create full loss portfolios   
linear\_loss <- -1000000\* na.omit(data[[4]])  
#full\_loss <- 1000000\*(exp(na.omit(data[[4]]) ) -1)  
  
# Fit GEV parameters to the data   
GEV\_fit <- gevFit(linear\_loss, block = 125)  
GEV\_fit

##   
## Title:  
## GEV Parameter Estimation   
##   
## Call:  
## gevFit(x = linear\_loss, block = 125)  
##   
## Estimation Type:  
## gev mle   
##   
## Estimated Parameters:  
## xi mu beta   
## 1.969227e-01 1.741177e+04 6.612363e+03   
##   
## Description  
## Mon Nov 14 16:32:27 2016

GEV\_params <- NULL  
GEV\_params[1:3] <- c(-0.075, 8529, 3694)  
  
#Slide 24 Parameters xi = 0.3614 mu = 8430 sigma = 3337  
qgev(0.95, xi = 0.3614, mu = 8430, beta = 3337)

## [1] 26208.18  
## attr(,"control")  
## xi mu beta lower.tail  
## 0.3614 8430 3337 TRUE

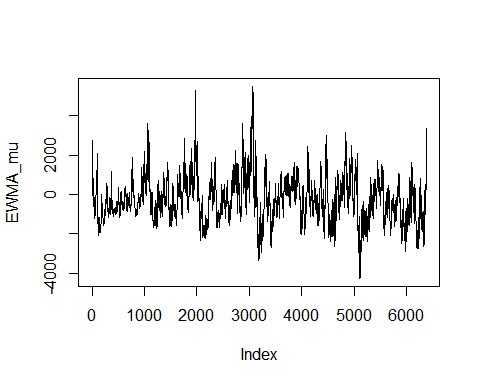
# Fit the GP parameters to data   
GP\_fit <- gpdFit(linear\_loss)  
GP\_fit

##   
## Title:  
## GPD Parameter Estimation   
##   
## Call:  
## gpdFit(x = linear\_loss)  
##   
## Estimation Method:  
## gpd mle   
##   
## Estimated Parameters:  
## xi beta   
## 0.1196947 4502.7225545   
##   
## Description  
## Mon Nov 14 16:32:27 2016 by user: board

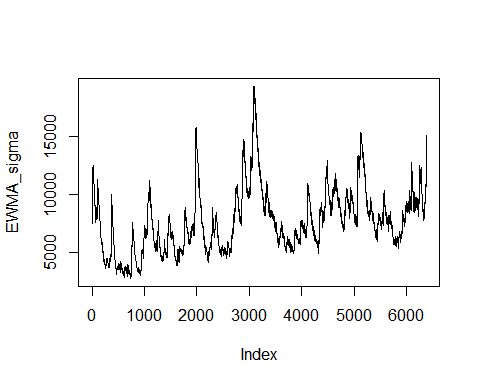
GP\_params <- NULL  
GP\_params[1:2] <- c(5.65067e-2, 2.39357e3)  
  
# Slide 37 xi = 0.2622, beta = 2549   
qgpd(0.95, xi = 0.2622, beta = 2549)

## [1] 11602.34  
## attr(,"control")  
## xi mu beta lower.tail  
## 0.2622 0 2549 TRUE

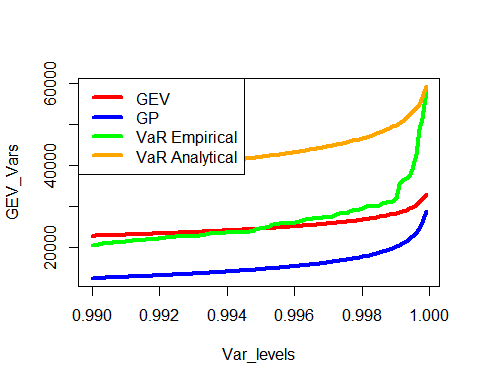
# Calculate EWMA mu and sigma   
numb\_obs <- length(linear\_loss)  
M <- 500   
  
#   
EWMA\_mu <- rep(0,(numb\_obs - M))  
EWMA\_mu[1] <- mean(linear\_loss[1:M])  
for (i in 1:(numb\_obs-M)){  
 EWMA\_mu[i+1]<- EWMA\_mu[i]\*0.97 + linear\_loss[(M+i)]\*0.03  
}  
plot(EWMA\_mu, type = 'l')



final\_EWMA\_mu = EWMA\_mu[(numb\_obs-M+1)]  
  
  
# EWMA Sigma   
EWMA\_sigma <- rep(0,(numb\_obs - M))  
EWMA\_sigma[1] <- var(linear\_loss[1:M])  
for (i in 1:(numb\_obs-M)){  
 EWMA\_sigma[i+1] <- 0.97\*EWMA\_sigma[i] + 0.03\*(EWMA\_mu[i] - linear\_loss[(M+i)])^2  
   
}  
EWMA\_sigma <- sqrt(EWMA\_sigma)  
plot(EWMA\_sigma, type = 'l')



final\_EWMA\_sigma <- EWMA\_sigma[(numb\_obs-M+1)]  
  
# VaR levels specified in question  
Var\_levels = seq(from = 0.99, to = 0.9999,by = 0.000099)  
  
# Compute quantiles for the specified VaR levels given the parameters from distributions above   
Var\_reg = quantile(linear\_loss, Var\_levels)  
Var\_EWMA = qnorm(Var\_levels, mean = final\_EWMA\_mu, sd = final\_EWMA\_sigma)  
GEV\_Vars = qgev(Var\_levels, xi = GEV\_params[1], mu = GEV\_params[2], beta = GEV\_params[3])  
GP\_Vars = qgpd(Var\_levels, xi = GP\_params[1], beta = GP\_params[2] )  
  
  
# Aggregate Var's for limits for plotting  
all\_data <- c(GEV\_Vars, GP\_Vars, Var\_reg, Var\_EWMA)  
  
# Plotting  
plot(Var\_levels, GEV\_Vars, ylim = c(min(all\_data), max(all\_data)),type = 'l', col= 'red', lwd = 4)  
lines(Var\_levels ,GP\_Vars,type = 'l', col = 'blue' , lwd = 4)  
lines(Var\_levels, Var\_reg, type = 'l', col = 'green', lwd =4)  
lines(Var\_levels, Var\_EWMA, type = 'l', col = 'orange', lwd = 4)  
  
legend("topleft", c("GEV", "GP", "VaR Empirical", "VaR Analytical"), lwd = 4, col = c("red", "blue", "green", "orange"))



# VaR at alpha = 0.9999  
Var\_reg\_9 = quantile(linear\_loss, 0.9999)  
Var\_EWMA\_9 = qnorm(0.9999, mean = final\_EWMA\_mu, sd = final\_EWMA\_sigma)  
GEV\_Vars\_9 = qgev(0.9999, xi = GEV\_params[1], mu = GEV\_params[2], beta = GEV\_params[3])  
GP\_Vars\_9 = qgpd(0.9999, xi = GP\_params[1], beta = GP\_params[2] )  
  
Var\_reg\_9

## 99.99%   
## 58012.61

Var\_EWMA\_9

## [1] 59361.88

GEV\_Vars\_9

## [1] 33097.1  
## attr(,"control")  
## xi mu beta lower.tail  
## -0.075 8529 3694 TRUE

GP\_Vars\_9

## [1] 28921.82  
## attr(,"control")  
## xi mu beta lower.tail  
## 0.0565067 0 2393.57 TRUE

# GEV has highest VaR 0.9999 at 33097  
  
  
#L' = 99,452   
pVar\_reg\_9 = max(linear\_loss)  
pVar\_EWMA\_9 = pnorm(99452, mean = final\_EWMA\_mu, sd = final\_EWMA\_sigma)  
pGEV\_Vars\_9 = pgev(99452, xi = GEV\_params[1], mu = GEV\_params[2], beta = GEV\_params[3])  
pGP\_Vars\_9 = pgpd(99452, xi = GP\_params[1], beta = GP\_params[2] )  
  
pVar\_reg\_9

## [1] 69088.98

pVar\_EWMA\_9

## [1] 1

pGEV\_Vars\_9

## [1] 1  
## attr(,"control")  
## xi mu beta lower.tail  
## -0.075 8529 3694 TRUE

pGP\_Vars\_9

## [1] 1  
## attr(,"control")  
## xi mu beta lower.tail  
## 0.0565067 0 2393.57 TRUE

# Need alpha of 1 for all measures...