

Visualization

Stefan Glogger

August 2017

Visualization

One Dispersion, different weights

We visualize the different portfolio returns of each time window of each dispersion in a histogram.

The results can (also) be found in “\IR-Phase FIM-Statistik\R-Research Project Statistics\Plot Optimization\Dispersion Const”.

on its own

not so interesting, nicer below

```
for(d in datesNames){
  retOverTime <- apply(1+ret[get(d),], 2, prod)

  for(i in names(xDispConst[[d]])){
    retDispTime <- numeric(length(xDispConst[[d]][[i]]))
    names(retDispTime) <- names(xDispConst[[d]][[i]])
    for(j in 1:length(retDispTime)){
      retDispTime[j] <- crossprod(xDispConst[[d]][[i]][[j]]$x, retOverTime)
    }

    t <- paste(d, i, sep = " - ")
    pdf(file.path(getwd(), "Plot Optimization", "Dispersion Const", paste0(t, ".pdf")), width = 10,
        plot(retDispTime, main = t)
        dev.off()
    }
}
```

together (all different dispersions)

```
for(d in datesNames){
  cols <- rainbow(length(xDispConst[[d]]))
  retOverTime <- apply(1+ret[get(d),], 2, prod)
  retDispTime <- data.frame(w = names(xDispConst[[d]][[1]]))

  for(i in names(xDispConst[[d]])){
    for(j in 1:nrow(retDispTime)){
      retDispTime[j,i] <- crossprod(xDispConst[[d]][[i]][[j]]$x, retOverTime)
    }
  }

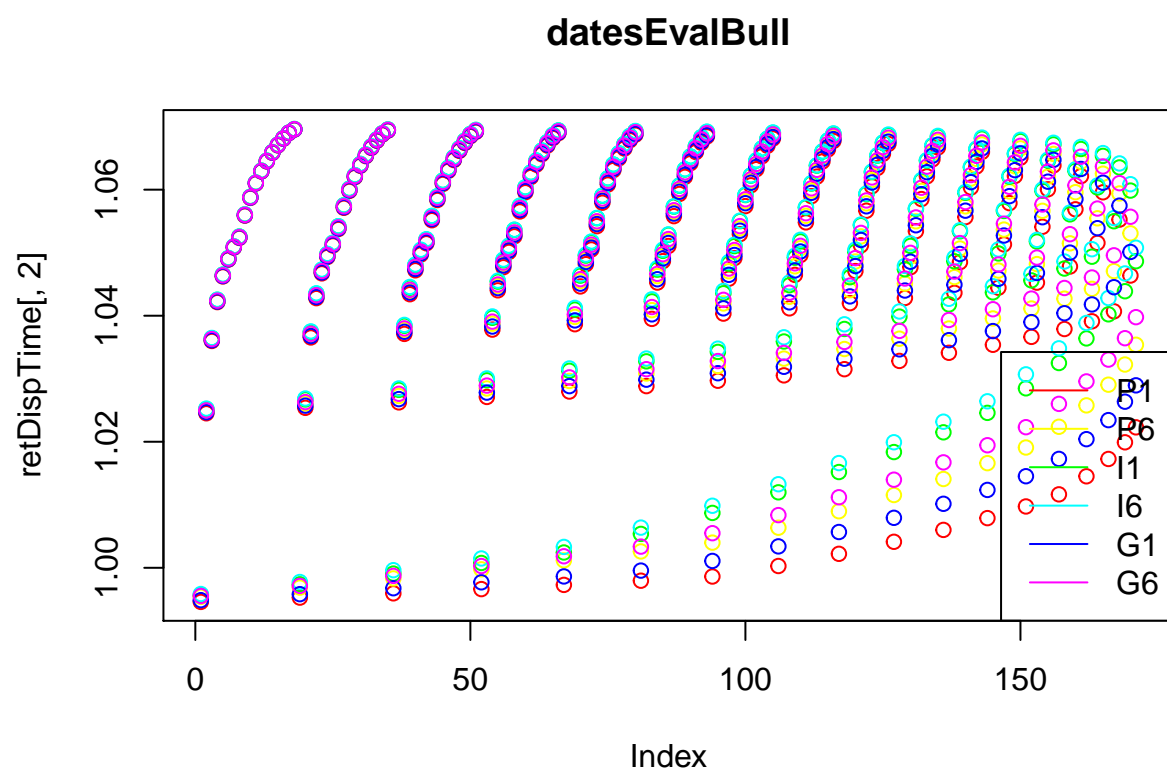
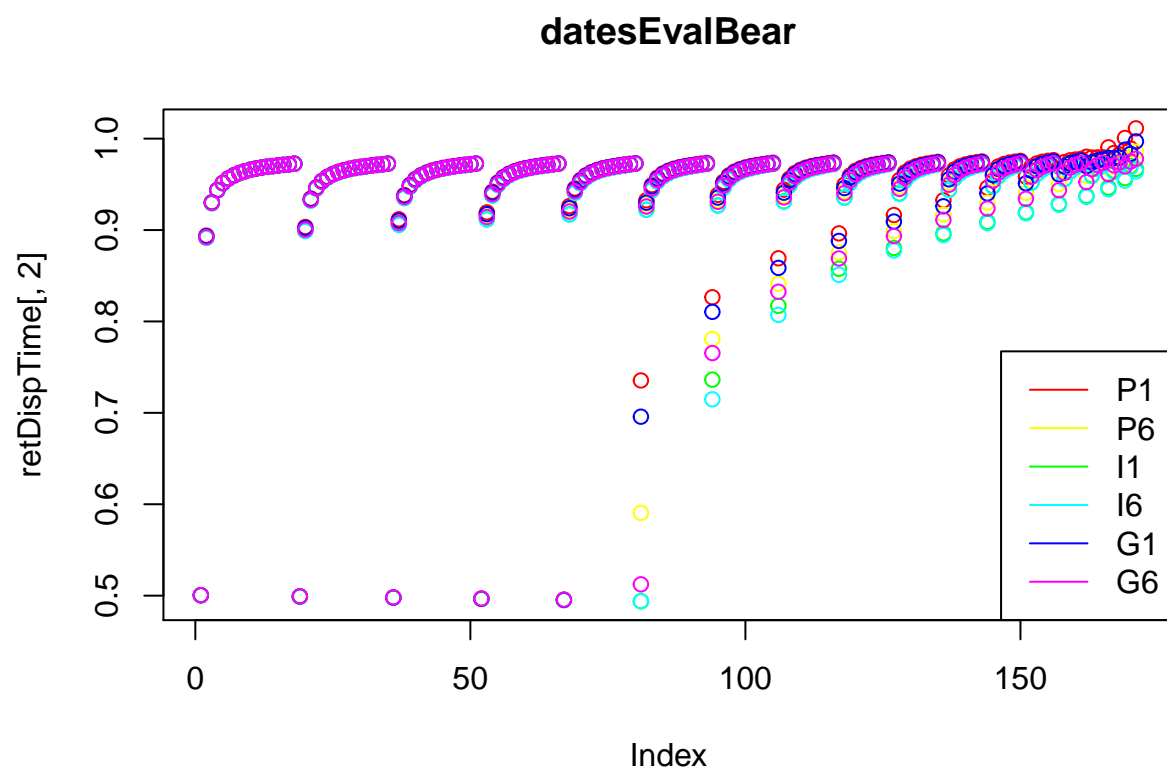
  ylim = c(min(retDispTime[,1]), max(retDispTime[,1]))
  plot(retDispTime[,2], ylim = ylim, col = cols[1], main = d)
```

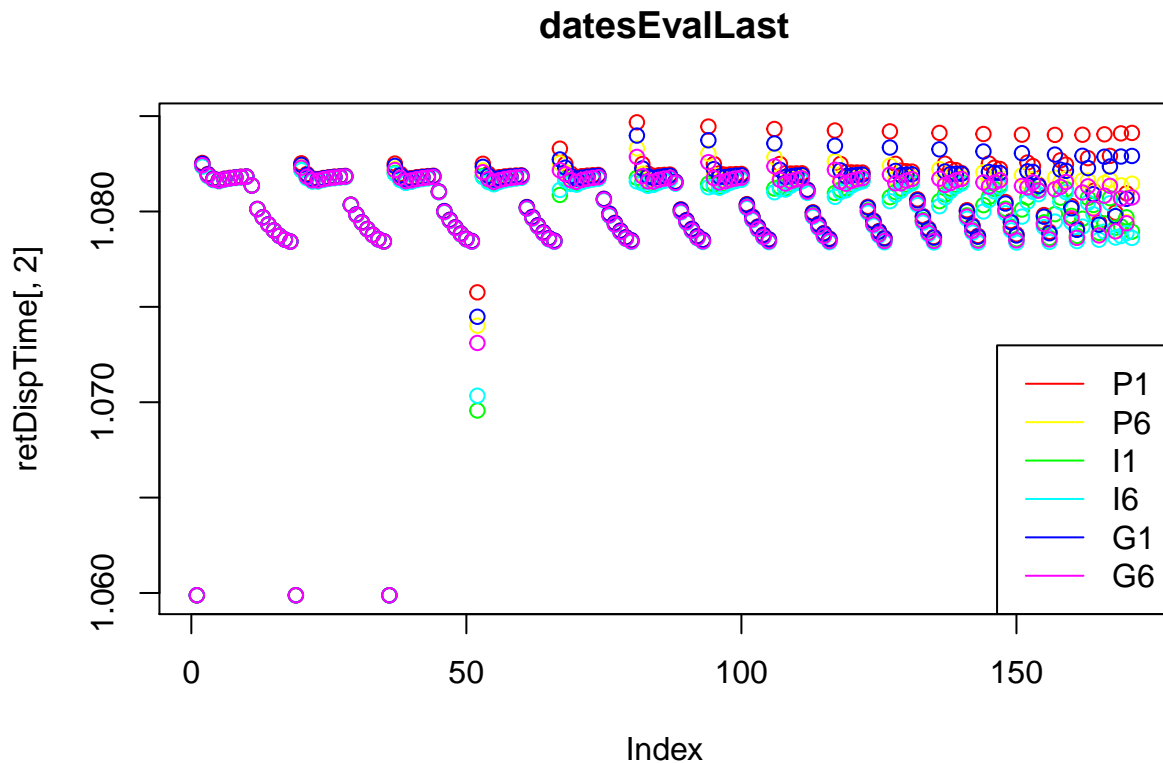
```

for(i in 3:ncol(retDispTime)){
  par(new=T)
  plot(retDispTime[,i], ylim = ylim, axes = F, xlab = "", ylab = "", col = cols[i-1])
}
legend("bottomright", legend = names(xDispConst[[d]]), col = cols, lty = 1)

pdf(file.path(getwd(), "Plot Optimization", "Dispersion Const", paste0("0", d, ".pdf")), width = 10)
plot(retDispTime[,2], ylim = ylim, col = cols[1], main = d)
for(i in 3:ncol(retDispTime)){
  par(new=T)
  plot(retDispTime[,i], ylim = ylim, axes = F, xlab = "", ylab = "", col = cols[i-1])
}
legend("bottomright", legend = names(xDispConst[[d]]), col = cols, lty = 1)
dev.off()
}

```





Classic Optimization

Constant weights over window

We want to visualize the evolvement of a portfolio over each time window.

Be aware of the index shifting: `retPlot[j-1, i]` take wealth of previous day `retOverTime[j-1,]` take return of today (`j` is one step ahead)

Remove numbering of x-axis by `xaxt='n'`.

```
for(d in datesNames){
  cols <- rainbow(length(xClassicConst[[d]]))
  retOverTime <- 1+ret[get(d),]
  retPlotDates <- get(d)
  retPlotDates <- c(datesAll[which(datesAll==min(retPlotDates))-1], retPlotDates)
  retPlot <- data.frame(Datum = retPlotDates)

  for(i in names(xClassicConst[[d]])){
    retPlot[1,i] <- 100
    for(j in 2:nrow(retPlot)){
      retPlot[j,i] <- retPlot[j-1,i]*crossprod(xClassicConst[[d]][[i]], retOverTime[j-1,])
    }
  }

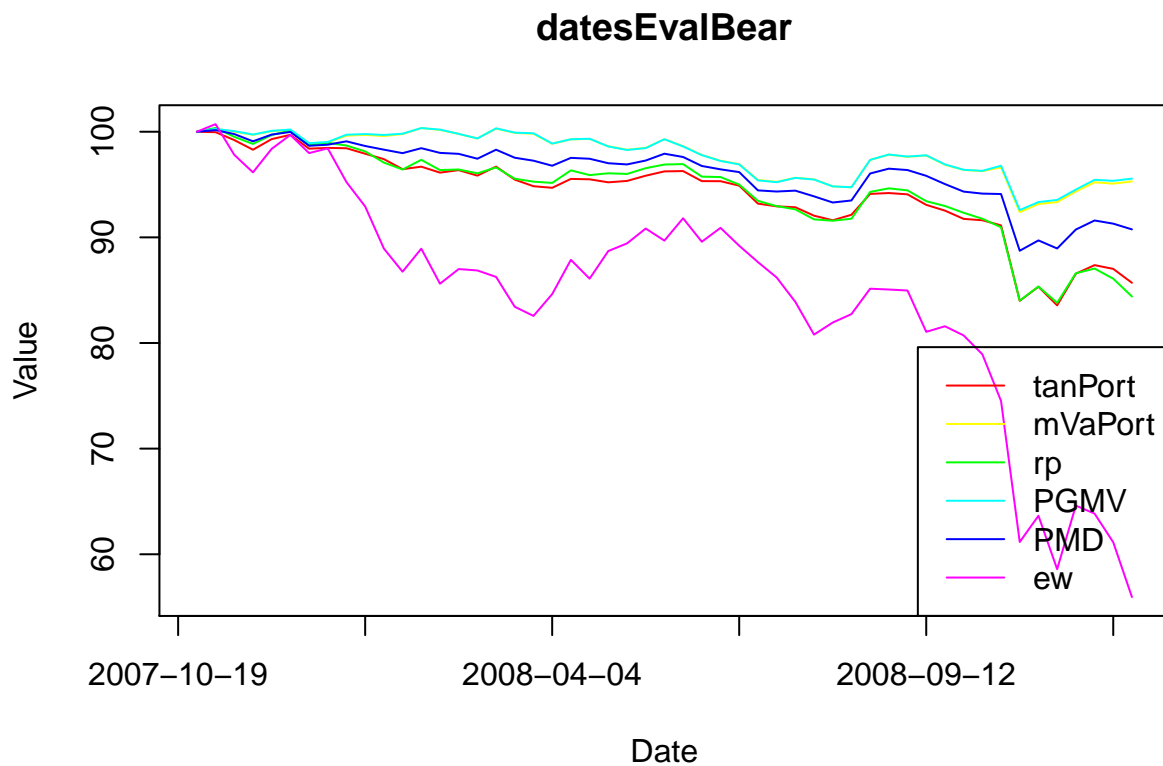
  ylim = c(min(retPlot[,-1]), max(retPlot[,-1]))
}
```

```

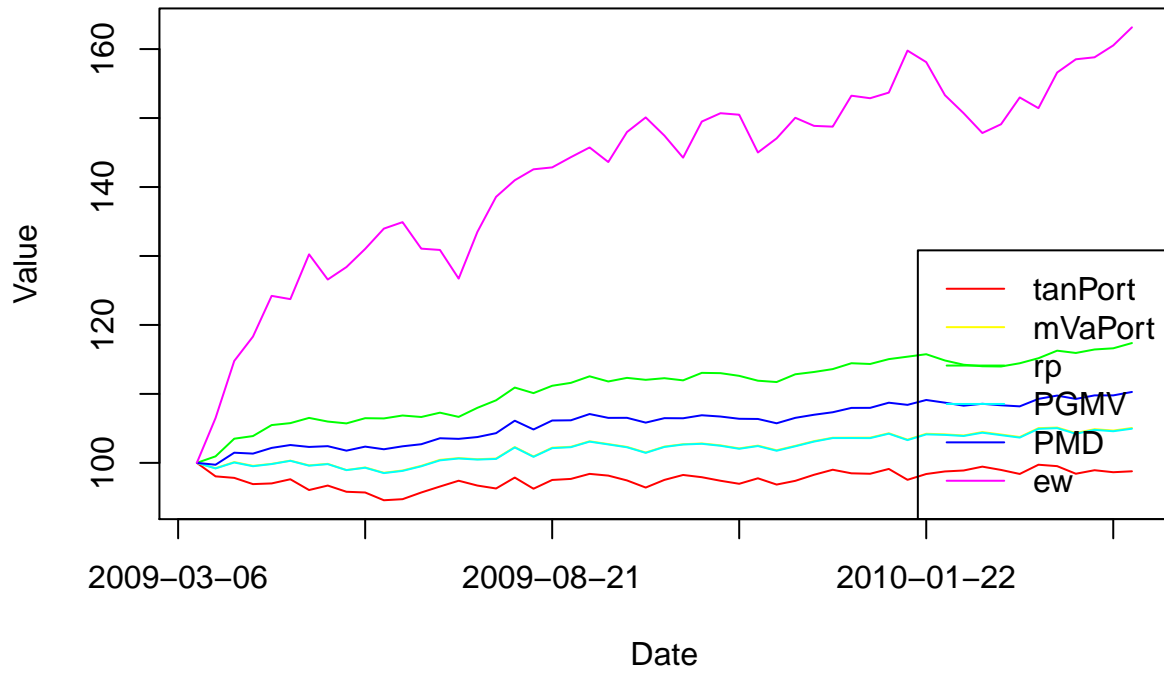
plot(retPlot[,2], type = "l", ylim = ylim, col = cols[1], main = d, xlab = "Date", ylab = "Value",
for(i in 3:ncol(retPlot)){
  par(new=T)
  plot(retPlot[,i], type = "l", ylim = ylim, axes = F, xlab = "", ylab = "", col = cols[i-1])
}
axis(1, at = c(0, 10, 20, 30, 40, 50), labels = retPlot[c(0, 10, 20, 30, 40, 50)+1,1])
legend("bottomright", legend = names(xClassicConst[[d]]), col = cols, lty = 1)

pdf(file.path(getwd(), "Plot Optimization", "Classical Const", paste0(d, ".pdf")), width = 10, height = 10)
plot(retPlot[,2], type = "l", ylim = ylim, col = cols[1], main = d, xlab = "Date", ylab = "Value",
for(i in 3:ncol(retPlot)){
  par(new=T)
  plot(retPlot[,i], type = "l", ylim = ylim, axes = F, xlab = "", ylab = "", col = cols[i-1])
}
axis(1, at = c(0, 10, 20, 30, 40, 50), labels = retPlot[c(0, 10, 20, 30, 40, 50)+1,1])
legend("bottomright", legend = names(xClassicConst[[d]]), col = cols, lty = 1)
dev.off()
}

```



datesEvalBull



datesEvalLast

