

# PerformanceAnalytics Changed by Kirk Li

kirkli@u.washington.edu

February 24, 2014

## Contents

<b>1</b>	<b>Changes on chart.Boxplot</b>	<b>1</b>
<b>2</b>	<b>Changes on chart.QQplot</b>	<b>6</b>

## 1 Changes on chart.Boxplot

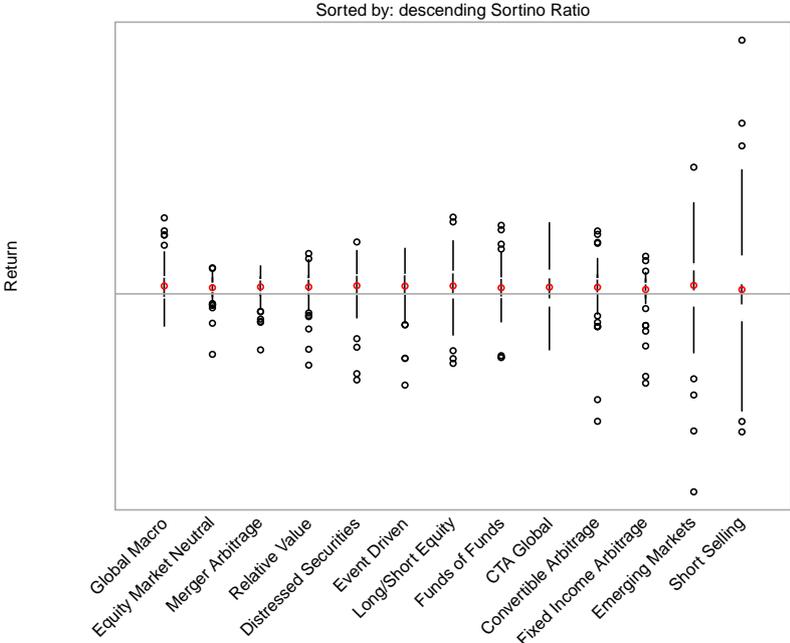
Remarks:

- Sorting boxplot by different risk measure
  - Enable the ascending sorting and descending sorting
  - Enable one of 18 measures that adopted from table.Distributions, table.DrawdownsRatio, table.DownsideRiskRatio, table.AnnualizedReturns.
- Fix the horizontal and vertical display of boxplot
  - Adjust the par value per horizontal=TRUE or FALSE
  - Adjust the labeling per horizontal=TRUE or FALSE
  - Modify the angle of axis label to reduce the space
- Add base measure that records the base ordering, relative to the specified ordering.
  - vertical
  - horizontal

```
# read data
data(edhec)

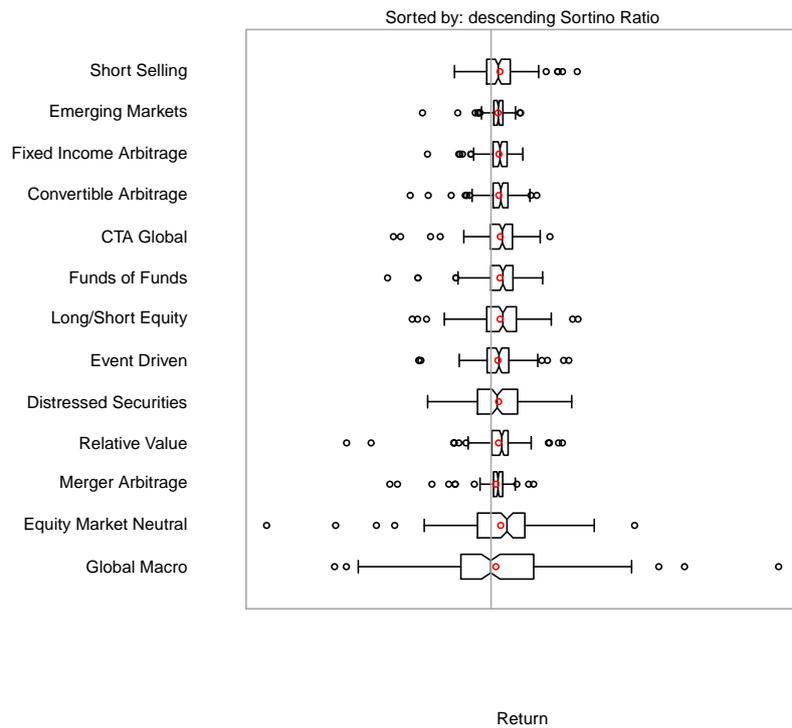
# vertical box, sort by Sortino ratio,
chart.Boxplot(R=edhec, sort.by="Sortino ratio",
             horizontal=FALSE, as.Tufte=TRUE)
```

# Return Distribution Comparison



```
# horizontal box, sort by Sortino ratio,
chart.Boxplot(R=edhec, sort.by="Sortino ratio",
             horizontal=TRUE, as.Notch=TRUE)
```

### Return Distribution Comparison



```

# vertical box, sort by mean, with base order variance
# ascending sort

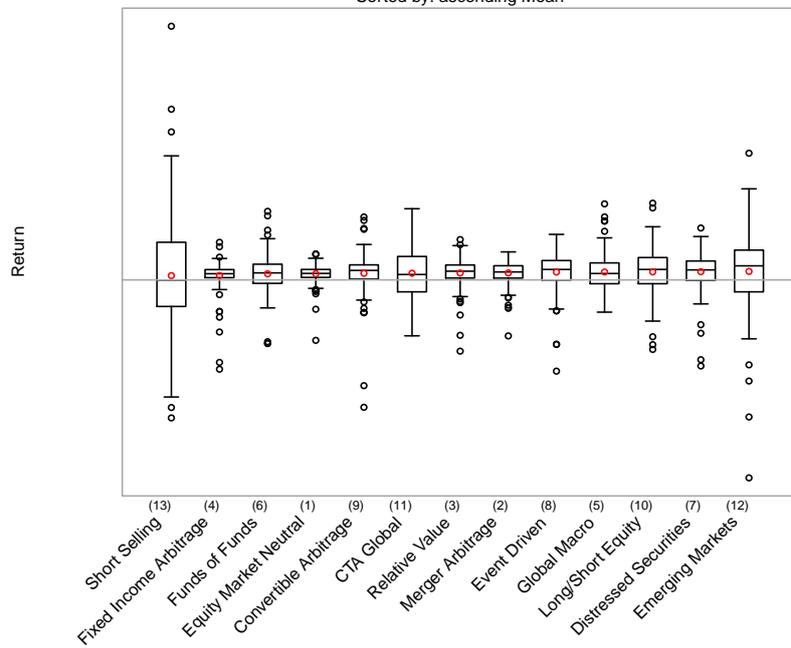
chart.Boxplot(R=edhec, sort.by="mean", horizontal=FALSE
, sort.base="variance", sort.ascending=TRUE)

```

### Return Distribution Comparison

Base order: ascending Variance

Sorted by: ascending Mean



```

# horizontal box, sort by mean, with base order
  variance

# descending sort

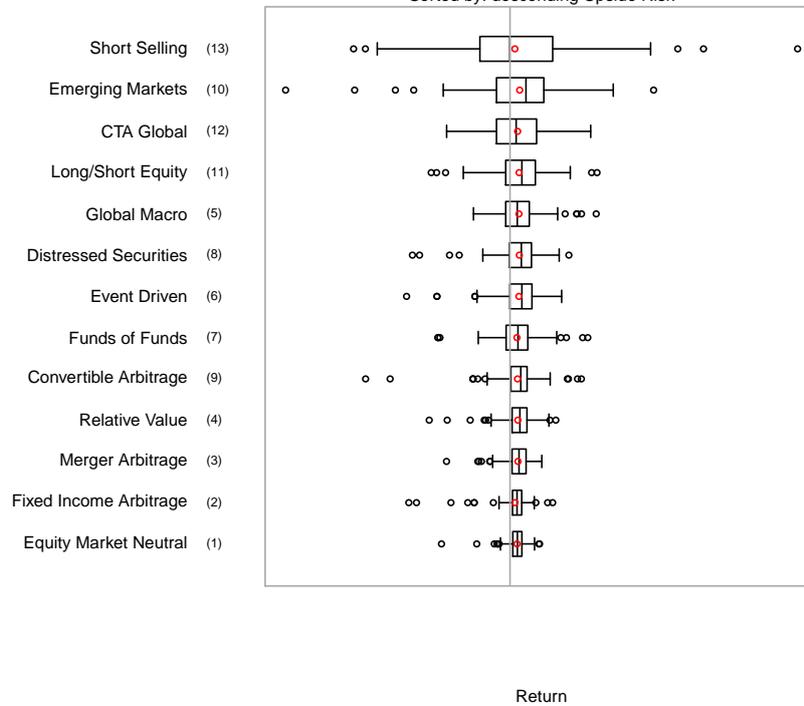
chart.Boxplot(R=edhec,sort.by="upside risk",
  horizontal=TRUE, sort.base="std dev", sort.
  ascending=FALSE)

```

### Return Distribution Comparison

Base order: descending Std Dev

Sorted by: descending Upside Risk



```

# horizontal box, sort by mean, with base order
  variance

# descending sort

measure_set <- c("NULL", "mean", "median", "variance",
  "sharp ratio",

  "mean absolute deviation", "std dev",
  "sterling ratio", "calmar ratio", "
  burke ratio", "pain index", "ulcer
  index", "martin ratio", "downside
  risk", "omega ratio", "sortino
  ratio", "upside risk", "upside
  potential ratio", "omega sharpe
  ratio")

try(

  sapply(measure_set, function(x){

    cat(x, "\n")

    chart.Boxplot(
      R=edhec,
      sort.by=x)

  })

)

```

## 2 Changes on chart.QQplot

Remarks:

- add normal mixture distribution

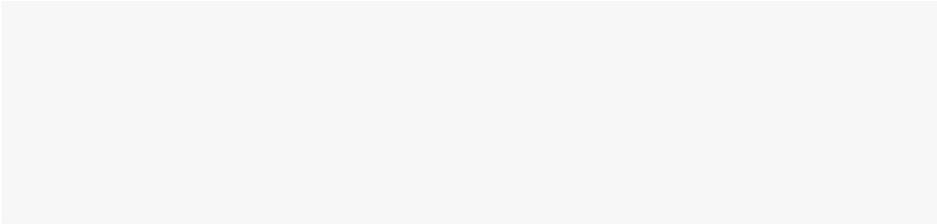
```

data(managers)

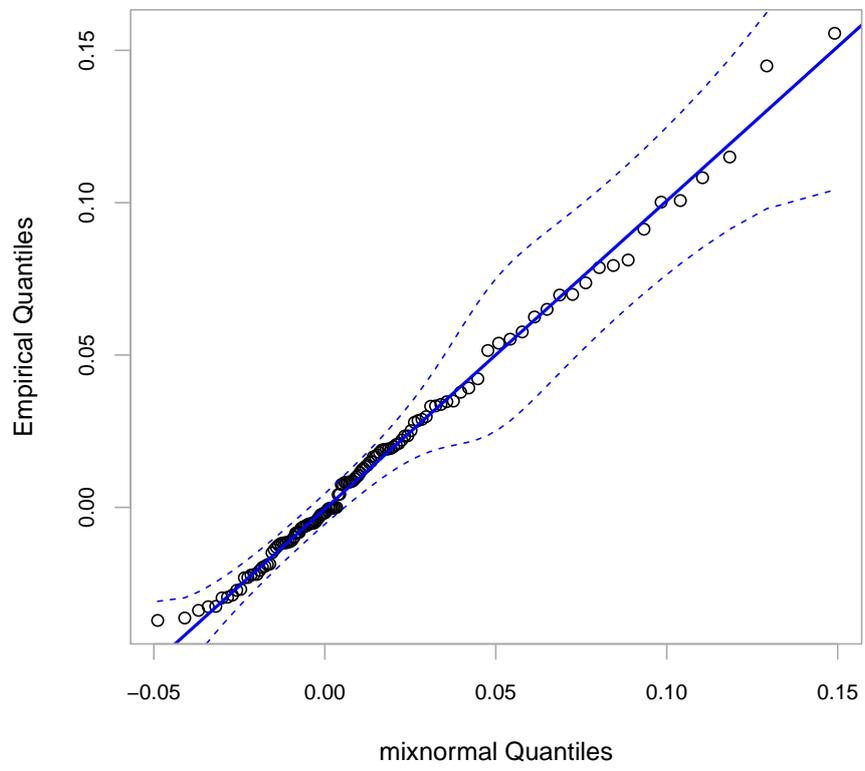
# fit with real data, using fitted distribution as
  theoretical quantiles

chart.QQPlot( checkData(managers[,2, drop = FALSE], na
  .rm = TRUE, method = "vector"), main = "Normal
  Mixture Distribution",
  line=c("quartiles"), para=list(m=2),
  distribution = 'mixnormal',
  envelope=0.95)

```



**Normal Mixture Distribution**



```

# fit with simulated data, using fitted distribution
  as theoretical quantiles

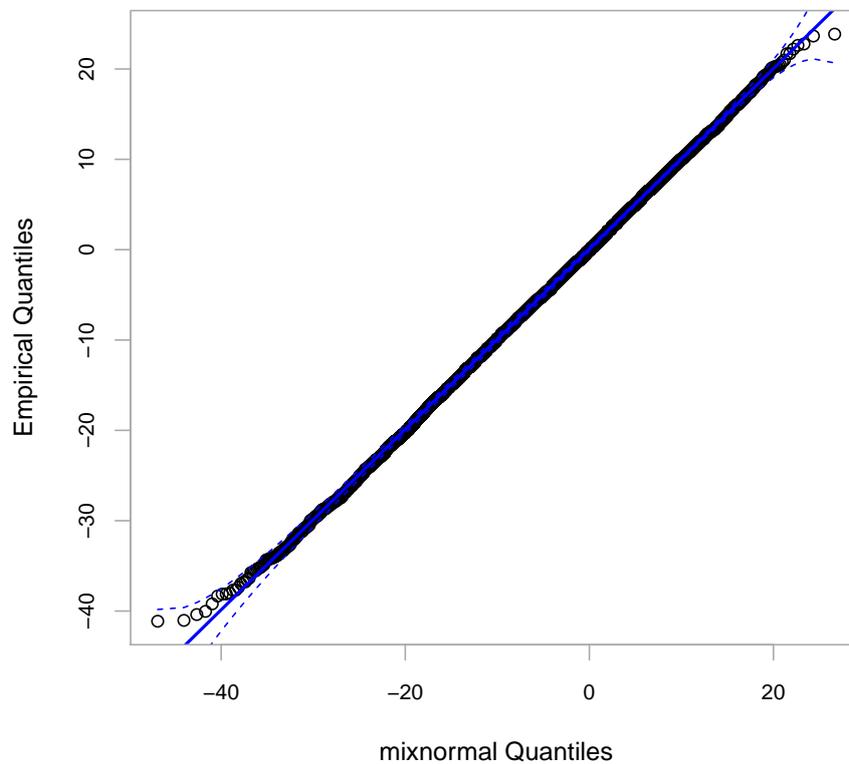
x <- replicate(10000, ifelse(runif(1)>0.5, rnorm(1,5,5),
  rnorm(1,-10,10)))

chart.QQPlot(x, main = "Normal Mixture Distribution",
  line=c("quartiles"), para=list(m=2),
  distribution = 'mixnormal',
  envelope=0.95)

## [1] "fitted model:"
##           mu sig2      w
## [1,]  -10.29 97.5 0.476
## [2,]   4.89 26.5 0.524
## [1] "using fitted model as theoretical distribution
"

```

**Normal Mixture Distribution**



```

# fit with simulated data, using true distribution as
  theoretical quantiles

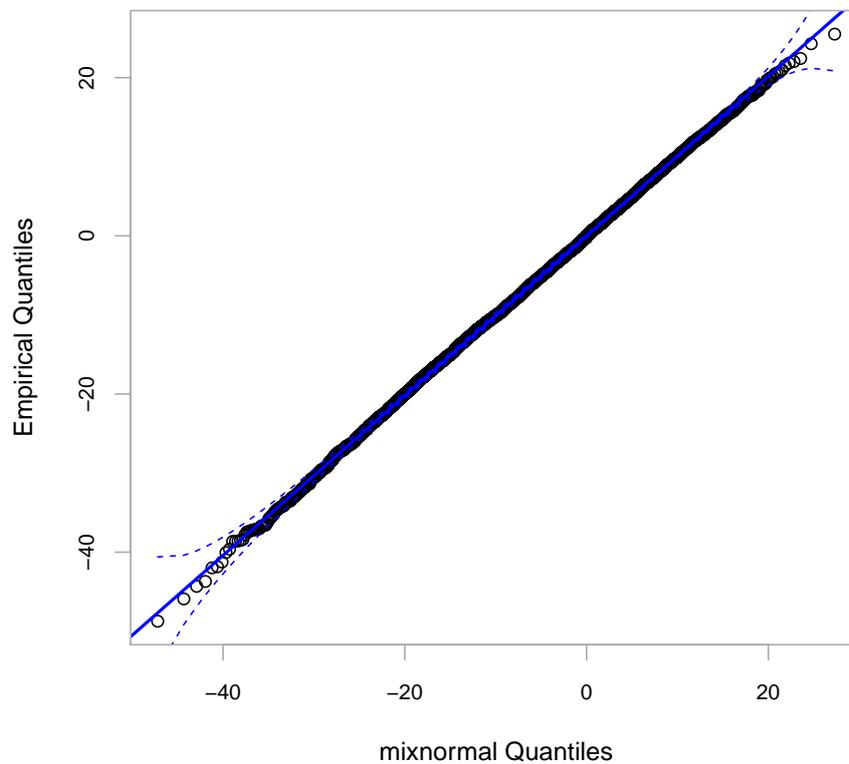
x <- replicate(10000,ifelse(runif(1)>0.5,rnorm(1,5,5),
  rnorm(1,-10,10)))

chart.QQPlot(x, main = "Normal Mixture Distribution",
  line=c("quartiles"), para=list(m=2,mu=
    c(5,-10),sig2=c(25,100),w=c
    (0.5,0.5)), distribution = '
    mixnormal',
    envelope=0.95)

## [1] "fitted model:"
##      mu sig2 w
## [1,] -9.77 102.3 0.514
## [2,]  5.08  24.3 0.486

```

**Normal Mixture Distribution**



```

# fit with simulated data, using wrong distribution as
  theoretical quantiles

x <- replicate(10000, ifelse(runif(1)>0.5, rnorm(1,5,5),
  rnorm(1,-10,10)))

chart.QQPlot(x, main = "Normal Mixture Distribution",
  line=c("quartiles"), para=list(m=2, mu=
    c(0,20), sig2=c(25,100), w=c(0.5,0.5)
  ), distribution = 'mixnormal',
  envelope=0.95)

## [1] "fitted model:"
##           mu sig2  w
## [1,]  -10.05 99.5 0.49
## [2,]   4.81 25.8 0.51

```

**Normal Mixture Distribution**

