**R语言第六章——Kurtosis**

第六小组成员：龚文珺 高婉 姜汝艺

1. **项目介绍**

通过r语言来求得一个资产组合的峰度，来观察数据的离散程度，度量资产组合偏离某分布的情况，正态分布的峰度为3，当时间序列的曲线峰值比正态分布的高时，峰度大于3，当比正态分布的低时，峰度小于3，峰度越大，说明该数据系列中的极端值越多。

峰度峰度是描述总体中所有取值分布形态陡缓程度的统计量，涉及到了投资组合理论，将收益率的标准差称为波动率，衡量风险，利用均值--方差模型分析得出通过投资组合可以有效降低风险的结论。Konno和Suzuki(1995)研究了收益不对称情况下的均值-方差-偏度模型，该模型在收益率分布不对称的情况下具有价值，因为具有相同均值和方差的资产组合很可能具有不同的偏度，偏度大的资产组合获得较大收益率的可能性也相应增加。Athayde，Flores（2002）考虑了非对称分布条件下的资产配置情况：在前两阶奇数矩限定的情况下，分别最小化方差与峰度并将其推广到最小化任一奇数矩阵；Jondeau，Rockinger（2002）在投资者效用函数为常数相对风险厌恶（CRRA）效用函数的假定下将期末期望收益Taylor展开取前4阶高阶矩，运用一阶条件来最优化资产配置；Jondeau，Rockinger（2005）考虑收益率的联合非正态分布和时变特征，包括了波动聚集性、非对称和肥尾特征。将期末期望收益Taylor展开并取前4阶高阶矩，运用一阶条件来最优化资产配置；Sahu等（2001，2003）提出偏正态分布来衡量高阶矩的影响，能充分考虑偏度与协偏度，同时处理“肥尾”的影响；Campbell R等（2004偏正态分布估计高阶矩的影响，贝叶斯方法处理收益分布的参数不确定性情况，在上述基础之上处理最优化问题。

参考以上理论以及资料，我们将上证综合指数，上证综合A股指数，上证综合B股指数，沪深300，中小板指五个指数构建一个组合，通过r语言计算其单个以及组合的峰值，构建时间序列，将其可视化。具体步骤可详见第二部分代码解释。

1. **代码展示**

## 1、导入价格数据，计算月收益率

1，安装所需要的包

install.packages("tidyverse")

install.packages("lubridate")

install.packages("readxl")

install.packages("highcharter")

install.packages("tidyquant")

install.packages("timetk")

install.packages("tibbletime")

install.packages("quantmod")

install.packages("PerformanceAnalytics")

install.packages("scales")

#加载包

library(tidyverse)

library(lubridate)

library(readxl)

library(highcharter)

library(tidyquant)

library(timetk)

library(tibbletime)

library(quantmod)

library(PerformanceAnalytics)

library(scales)

#创建对象

symbols <- c("SZ","AG", "BG", "HS","ZXB")

#选择上证综合指数，上证综合A股指数，上证综合B股指数，沪深300，中小板指，2012年12月31日至2017年12月31日的日收盘价格，数据来源：国泰安数据库

#数据保存至Excel文件，并提取数据，并新增一列日期，格式为年月日

#将数据转换成时间序列格式数据

prices <-

read\_excel("mydata.xlsx",

col\_types = c("text", "numeric",

"numeric", "numeric",

"numeric", "numeric")) %>%

mutate(date = ymd(date)) %>%

tk\_xts(date\_var = date)

#将日度数据转为月度数据，并至每月最后一天

prices\_monthly <- to.monthly(prices,

indexAt = "lastof",

OHLC = FALSE)

head(prices\_monthly, 3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | SZ | AG | BG | HS | ZXB |
| 2012-12-31 | 2269.128 | 2376.039 | 244.915 | 2522.952 | 719.479 |
| 2013-01-31 | 2385.422 | 2496.833 | 277.773 | 2686.882 | 766.152 |
| 2013-02-28 | 2365.593 | 2475.974 | 277.624 | 2673.327 | 799.095 |

#计算数据的收益率-第一种方法

asset\_returns\_xts <-

Return.calculate(prices\_monthly,

method = "log") %>%

na.omit()

head(asset\_returns\_xts, 3)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | SZ | AG | BG | HS | ZXB |
| 2013-01-31 | 0.0499804 | 0.0495883 | 0.1258930 | 0.062952 | 0.0628532 |
| 2013-02-28 | -0.008347 | -0.008389 | -0.000536 | -0.00506 | 0.0420993 |
| 2013-03-31 | -0.056063 | -0.056089 | -0.051095 | -0.06900 | -0.040021 |

#计算数据的收益率-第二种方法

asset\_returns\_dplyr\_byhand <-

prices %>%

to.monthly(indexAt = "lastof", OHLC = FALSE) %>%

data.frame(date = index(.)) %>%

# 移除指数因为指数转换成了行名

remove\_rownames() %>%

gather(asset, prices, -date) %>%

group\_by(asset) %>%

mutate(returns = (log(prices) - log(lag(prices)))) %>%

select(-prices) %>%

spread(asset, returns)%>%

select(date, symbols)

#删除空缺值，即第一行

asset\_returns\_dplyr\_byhand <-

asset\_returns\_dplyr\_byhand %>%

na.omit()

#计算数据的收益率-第三种方法

asset\_returns\_tq\_builtin <-

prices %>%

tk\_tbl(preserve\_index = TRUE,

rename\_index = "date") %>%

gather(asset, prices, -date) %>%

group\_by(asset) %>%

tq\_transmute(mutate\_fun = periodReturn,

period = "monthly",

type = "log") %>%

spread(asset, monthly.returns)%>%

select(date, symbols) %>%

slice(-1)

#计算数据的收益率-第四种方法

asset\_returns\_tbltime <-

prices %>%

tk\_tbl(preserve\_index = TRUE,

rename\_index = "date") %>%

as\_tbl\_time(index = date) %>%

as\_period(period = "month",

side = "end") %>%

gather(asset, returns, -date) %>%

group\_by(asset) %>%

tq\_transmute(mutate\_fun = periodReturn,

type = "log") %>%

spread(asset, monthly.returns) %>%

select(date, symbols) %>%

slice(-1)

#将数据整理为3列

asset\_returns\_long <-

asset\_returns\_dplyr\_byhand %>%

gather(asset, returns, -date) %>%

group\_by(asset)

#画出收益率曲线-折线图

highchart(type = "stock") %>%

hc\_title(text = "Monthly Log Returns") %>%

hc\_add\_series(asset\_returns\_xts[, symbols[1]],

name = symbols[1]) %>%

hc\_add\_series(asset\_returns\_xts[, symbols[2]],

name = symbols[2]) %>%

hc\_add\_series(asset\_returns\_xts[, symbols[3]],

name = symbols[3]) %>%

hc\_add\_series(asset\_returns\_xts[, symbols[4]],

name = symbols[4]) %>%

hc\_add\_series(asset\_returns\_xts[, symbols[5]],

name = symbols[5]) %>%

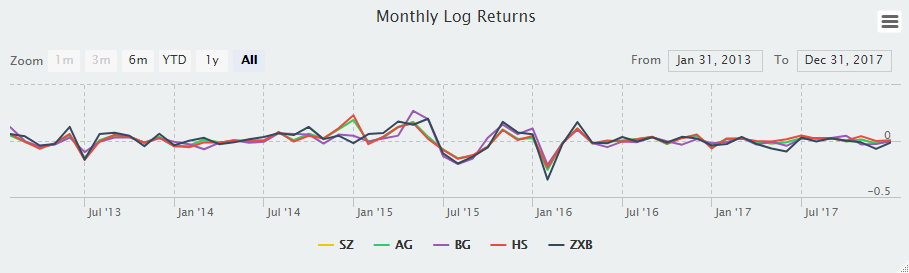
hc\_add\_theme(hc\_theme\_flat()) %>%

hc\_navigator(enabled = FALSE) %>%

hc\_scrollbar(enabled = FALSE) %>%

hc\_exporting(enabled = TRUE) %>%

hc\_legend(enabled = TRUE)



#画出收益率曲线-直方图

hchart(hc\_hist, color = "cornflowerblue") %>%

hc\_title(text =

paste(symbols[1],

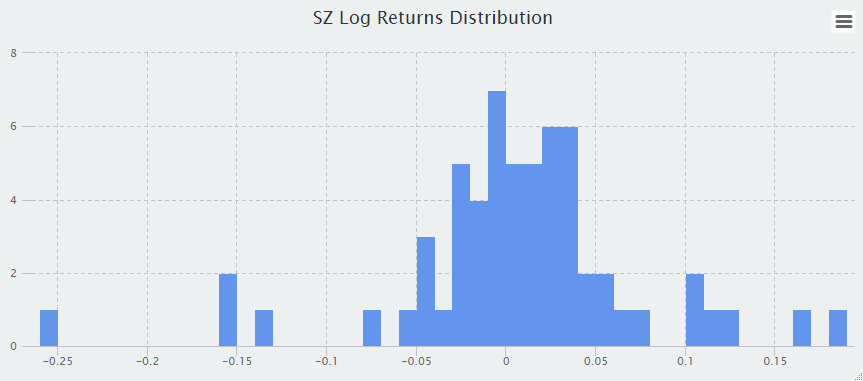
"Log Returns Distribution",

sep = " ")) %>%

hc\_add\_theme(hc\_theme\_flat()) %>%

hc\_exporting(enabled = TRUE) %>%

hc\_legend(enabled = FALSE)



## 2、创建资产组合

#创建权重向量数据

w <- c(0.25,0.25,0.20,0.20,0.10)

#创建简单的数据集将权重和资产组合进行匹配，检查匹配情况

tibble(w, symbols)

#计算加权投资组合每月的收益率，提取资产组合收益率数据，进行加权，每个月对权重进行重新调节，使权重每月都固定，将列变量名称改为“returns”

portfolio\_returns\_xts\_rebalanced\_monthly <-

Return.portfolio(asset\_returns\_xts,

weights = w,

rebalance\_on = "months") %>%

`colnames<-`("returns")#将列变量名称改为“returns”

head(portfolio\_returns\_xts\_rebalanced\_monthly, 3)

|  |  |
| --- | --- |
|  | returns |
| 2013-01-31 | 0.068946469 |
| 2013-02-28 | -0.001093063 |
| 2013-03-31 | -0.056059389 |

#使用tidyverse计算资产组合月收益率

asset\_returns\_long %>%

group\_by(asset) %>%

mutate(weights = case\_when(asset == symbols[1] ~ w[1],

asset == symbols[2] ~ w[2],

asset == symbols[3] ~ w[3],

asset == symbols[4] ~ w[4],

asset == symbols[5] ~ w[5])) %>%

head(3)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| date | asset | returns | weights |  |
|  | *<date>* | *<chr>* | *<dbl>* | *<dbl>* |
| 1 | 2013-01-31 | SZ | 0.0500 | 0.25 |
| 2 | 2013-02-28 | SZ | -0.00835 | 0.25 |

portfolio\_returns\_dplyr\_byhand <-

asset\_returns\_long %>%

group\_by(asset) %>%

mutate(weights = case\_when(asset == symbols[1] ~ w[1],

asset == symbols[2] ~ w[2],

asset == symbols[3] ~ w[3],

asset == symbols[4] ~ w[4],

asset == symbols[5] ~ w[5]),

weighted\_returns = returns \* weights) %>%

group\_by(date) %>%

summarise(returns = sum(weighted\_returns))

head(portfolio\_returns\_dplyr\_byhand, 3)

|  |  |  |
| --- | --- | --- |
|  | date | returns |
|  | *<date>* | *<dbl>* |
| 1 | 2013-01-31 | 0.0689 |
| 2 | 2013-02-28 | -0.00109 |

#使用tidyquant包实现资产组合月平衡收益率的计算

portfolio\_returns\_tq\_rebalanced\_monthly <-

asset\_returns\_long %>%

tq\_portfolio(assets\_col = asset,

returns\_col = returns,

weights = w,

col\_rename = "returns",

rebalance\_on = "months")

#画出资产组合月收益率的散点图

portfolio\_returns\_tq\_rebalanced\_monthly %>%

ggplot(aes(x = date, y = returns)) +

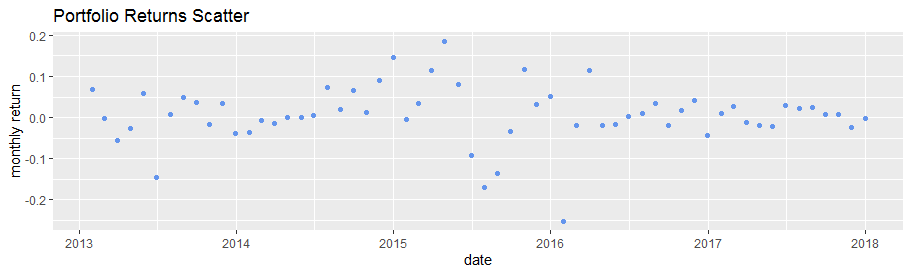
geom\_point(colour = "cornflowerblue")+

xlab("date") +

ylab("monthly return") +

theme\_update(plot.title = element\_text(hjust = 0.5)) +

ggtitle("Portfolio Returns Scatter") +

scale\_x\_date(breaks = pretty\_breaks(n=6)) 

#画出资产组合月收益直方图

portfolio\_returns\_tq\_rebalanced\_monthly %>%

ggplot(aes(x = returns)) +

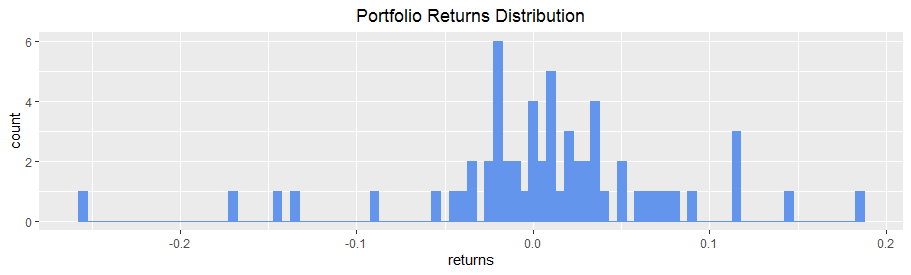
geom\_histogram(binwidth = .005,

fill = "cornflowerblue",

color = "cornflowerblue") +

ggtitle("Portfolio Returns Distribution") +

theme\_update(plot.title = element\_text(hjust = 0.5))



#画出资产组合的月收益率时间序列折线图

highchart(type = "stock") %>%

hc\_title(text = "Portfolio Monthly Returns") %>%

hc\_add\_series(portfolio\_returns\_xts\_rebalanced\_monthly$returns,

name = "Rebalanced Monthly",

color = "cornflowerblue") %>%

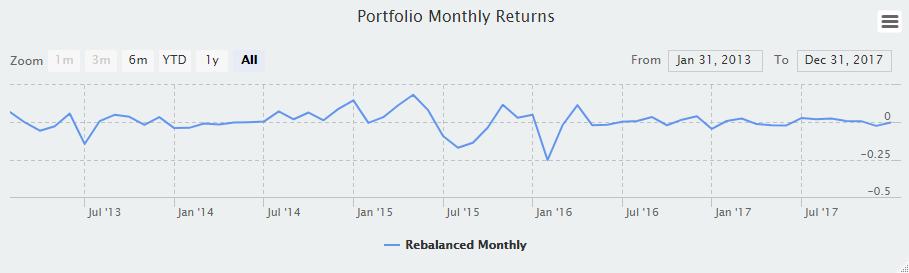
hc\_add\_theme(hc\_theme\_flat()) %>%

hc\_navigator(enabled = FALSE) %>%

hc\_scrollbar(enabled = FALSE) %>%

hc\_legend(enabled = TRUE) %>%

hc\_exporting(enabled = TRUE)



#画出单个资产的分布直方图与资产组合进行比较

asset\_returns\_long %>%

ggplot(aes(x = returns,

fill = asset)) +

geom\_histogram(alpha = 0.15,

binwidth = .01) +

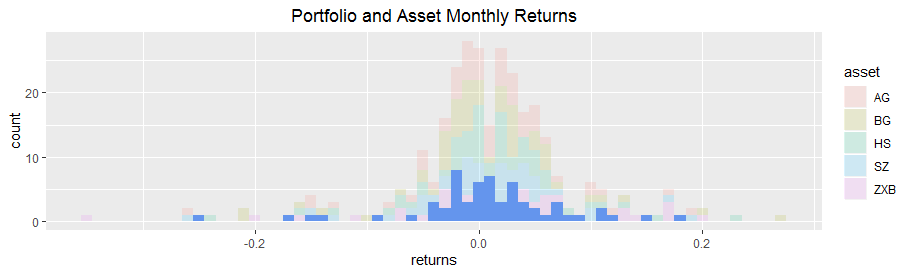
geom\_histogram(data = portfolio\_returns\_tq\_rebalanced\_monthly,

fill = "cornflowerblue",

binwidth = .01) +

ggtitle("Portfolio and Asset Monthly Returns") +

theme\_update(plot.title = element\_text(hjust = 0.5))



#画出资产组合收益率的密度分布图和直方图

portfolio\_returns\_tq\_rebalanced\_monthly %>%

ggplot(aes(x = returns)) +

geom\_histogram(binwidth = .01,

colour = "cornflowerblue",

fill = "cornflowerblue") +

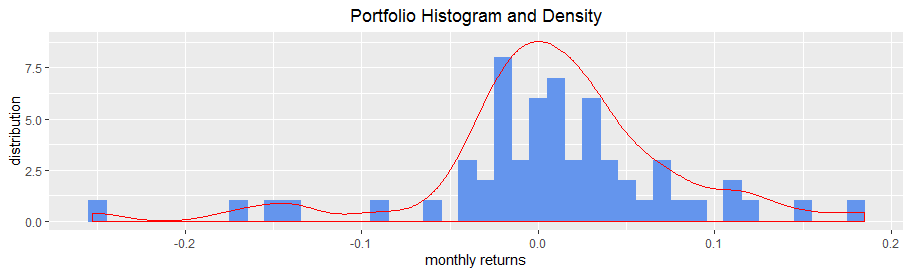
geom\_density(alpha = 1, color = "red") +

xlab("monthly returns") +

ylab("distribution") +

theme\_update(plot.title = element\_text(hjust = 0.5)) +

ggtitle("Portfolio Histogram and Density")



## 3、计算组合与单个资产峰度

#用kurtosis()函数计算其峰度

kurt\_xts <-

kurtosis(portfolio\_returns\_xts\_rebalanced\_monthly$returns)

#利用kurtosis的计算公式和直接用kurtosis(）函数相比较，发现计算结果是一致的

kurt\_tidy <-

portfolio\_returns\_tq\_rebalanced\_monthly %>%

summarise(

kurt\_builtin = kurtosis(returns),

kurt\_byhand =

((sum((returns - mean(returns))^4)/

length(returns))/

((sum((returns - mean(returns))^2)/

length(returns))^2)) - 3) %>%

select(kurt\_builtin, kurt\_byhand)

#新增一列

kurt\_tidy %>%

mutate(xts = kurt\_xts)

#画出收益率的密度分布图

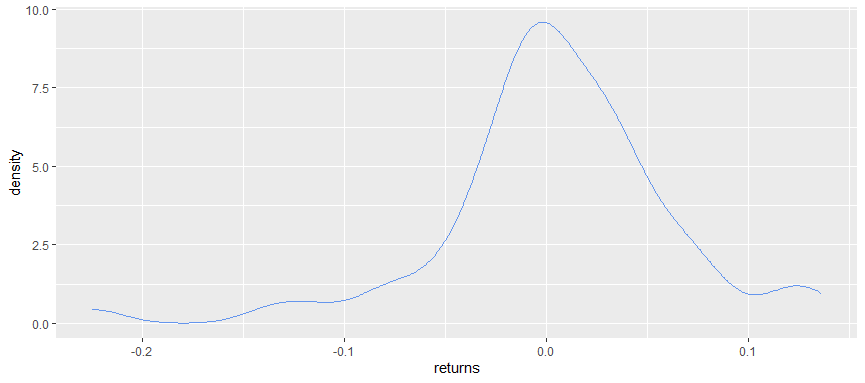
portfolio\_density\_plot <-

portfolio\_returns\_tq\_rebalanced\_monthly %>%

ggplot(aes(x = returns)) +

stat\_density(geom = "line", alpha = 1, colour = "cornflowerblue")

portfolio\_density\_plot



#将峰度可视化，使用ggplot将投资组合f峰度密度函数展示成图形，并将高于和低于收益率均值两个标准差的尾部用粉色阴影标示

sd\_pos <-

mean + (2\* sd(portfolio\_returns\_tq\_rebalanced\_monthly$returns))

sd\_neg <-

mean - (2\* sd(portfolio\_returns\_tq\_rebalanced\_monthly$returns))

sd\_pos\_shaded\_area <-

ggplot\_build(portfolio\_density\_plot)$data[[1]] %>%

filter(x > sd\_pos )

sd\_neg\_shaded\_area <-

ggplot\_build(portfolio\_density\_plot)$data[[1]] %>%

filter(x < sd\_neg)

portfolio\_density\_plot +

geom\_area(data = sd\_pos\_shaded\_area,

aes(x = x, y = y),

fill="pink",

alpha = 0.5) +

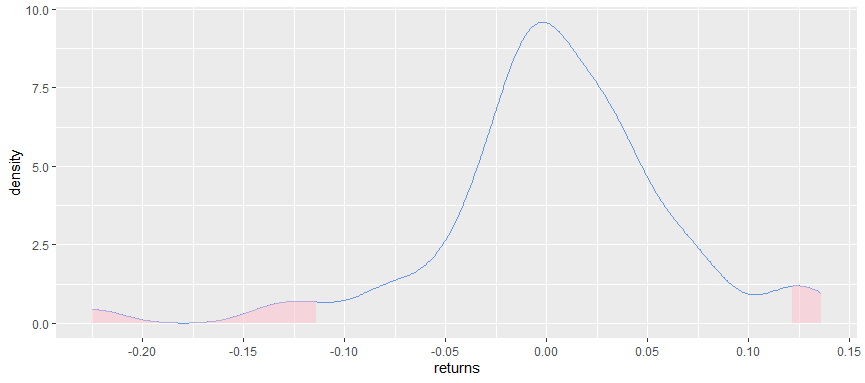
geom\_area(data = sd\_neg\_shaded\_area,

aes(x = x, y = y),

fill="pink",

alpha = 0.5) +

scale\_x\_continuous(breaks = pretty\_breaks(n = 10))



#将上述图形美化，增加更多图形元素，更加标准化

#增加加入均值和中位数统计变量，将均值用红色虚线标示，将中位数用黑色虚线标示

mean <- mean(portfolio\_returns\_tq\_rebalanced\_monthly$returns)

mean\_line\_data <-

ggplot\_build(portfolio\_density\_plot)$data[[1]] %>%

filter(x <= mean)

median<- median(portfolio\_returns\_tq\_rebalanced\_monthly$returns)

median\_line\_data <-

ggplot\_build(portfolio\_density\_plot)$data[[1]] %>%

filter(x <= median)

portfolio\_density\_plot +

geom\_area(data = sd\_pos\_shaded\_area,

aes(x = x, y = y),

fill="pink",

alpha = 0.5) +

geom\_area(data = sd\_neg\_shaded\_area,

aes(x = x, y = y),

fill="pink",

alpha = 0.5) +

geom\_segment(data = shaded\_area\_data,

aes(x = mean,

y = 0,

xend = mean,

yend = density),

color = "red",

linetype = "dotted") +

annotate(geom = "text",

x = mean,

y = 5,

label = "mean",

color = "red",

fontface = "plain",

angle = 90,

alpha = .8,

vjust = -1.75) +

geom\_segment(data = median\_line\_data,

aes(x = median,

y = 0,

xend = median,

yend = density),

color = "black",

linetype = "dotted") +

annotate(geom = "text",

x = median,

y = 5,

label = "median",

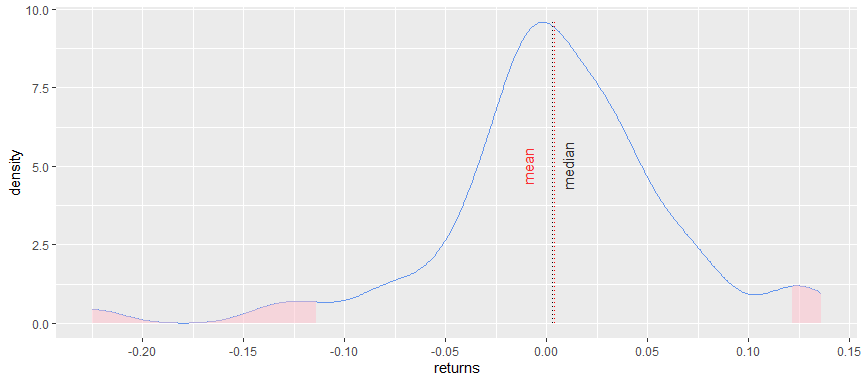
fontface = "plain",

angle = 90,

alpha = .8,

vjust = 1.75) +

scale\_x\_continuous(breaks = pretty\_breaks(n = 10))



#将资产组合的峰度与单个资产的峰度进行比较，X轴为资产，y轴为峰度

asset\_returns\_long %>%

summarize(kurt\_assets = kurtosis(returns)) %>%

add\_row(asset = "Portfolio",

kurt\_assets = kurt\_tidy$kurt\_byhand) %>%

ggplot(aes(x = asset,

y = kurt\_assets,

colour = asset)) +

geom\_point() +

geom\_text(

aes(x = "Portfolio",

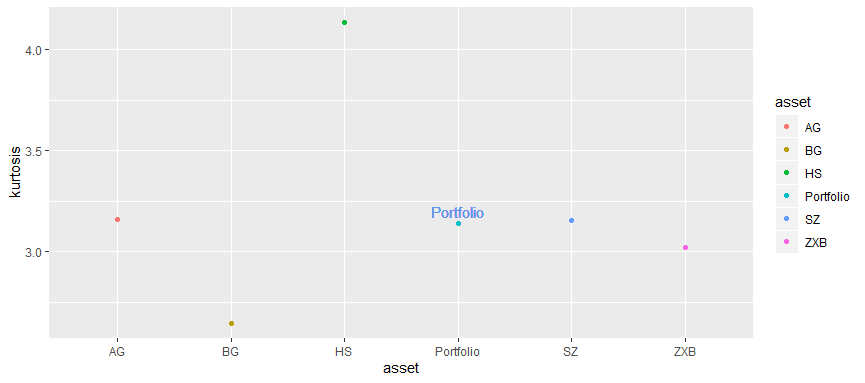
y =

kurt\_tidy$kurt\_byhand + .06),

label = "Portfolio",

color = "cornflowerblue") +

labs(y = "kurtosis")



#计算时间序列中的滚动峰度，并删除缺失值

window <- 24

rolling\_kurt\_xts <-

rollapply(portfolio\_returns\_xts\_rebalanced\_monthly,

FUN = kurtosis,

width = window) %>%

na.omit()

#使用rollify()函数，将任何函数转换为自身的版本

kurt\_roll\_24 <-

rollify(kurtosis,

window = window)

roll\_kurt\_tibbletime <-

portfolio\_returns\_tq\_rebalanced\_monthly %>%

as\_tbl\_time(index = date) %>%

mutate(kurt = kurt\_roll\_24(returns)) %>%

select(-returns) %>%

na.omit()

#使用tq\_mutate() 函数

rolling\_kurt\_tq <-

portfolio\_returns\_tq\_rebalanced\_monthly %>%

tq\_mutate(select = returns,

mutate\_fun = rollapply,

width = window,

FUN = kurtosis,

col\_rename = "tq") %>%

select(-returns) %>%

na.omit()



#将滚动峰值可视化

rolling\_kurt\_tq %>%

mutate(xts = coredata(rolling\_kurt\_xts),

tbltime = roll\_kurt\_tibbletime$kurt) %>%

mutate\_if(is.numeric, funs(round(.,3))) %>%

tail(3)

|  |  |  |  |
| --- | --- | --- | --- |
| <date> | <db1> | <db1> | <db1> |
| 2017-10-31 | 7.88 | 7.88 | 7.88 |
| 2017-11-30 | 7.79 | 7.79 | 7.79 |
| 2017-12-31 | 8.11 | 8.11 | 8.11 |

#使用 highcharter包来可视化滚动峰值

highchart(type = "stock") %>%

hc\_title(text = "Rolling 24-Month kurtosis") %>%

hc\_add\_series(rolling\_kurt\_xts,

name = "Rolling 24-Month kurtosis",

color = "cornflowerblue") %>%

hc\_yAxis(title = list(text = "kurtosis"),

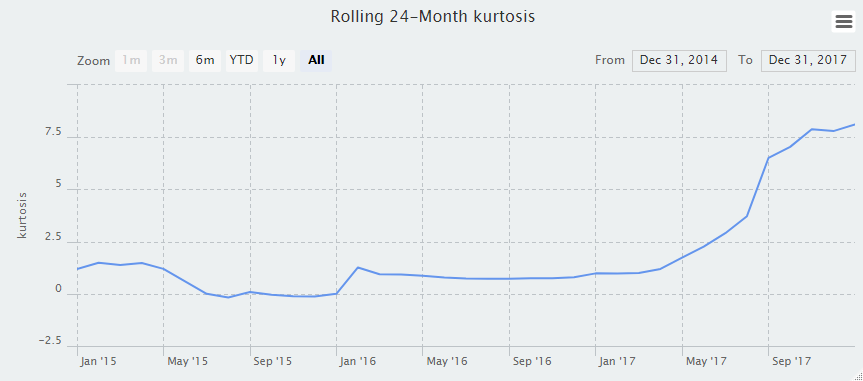
opposite = FALSE) %>%

hc\_add\_theme(hc\_theme\_flat()) %>%

hc\_navigator(enabled = FALSE) %>%

hc\_scrollbar(enabled = FALSE) %>%

hc\_exporting(enabled = TRUE)



#使用ggplot包使图更加完善

rolling\_kurt\_tq %>%

ggplot(aes(x = date, y = tq)) +

geom\_line(color = "cornflowerblue") +

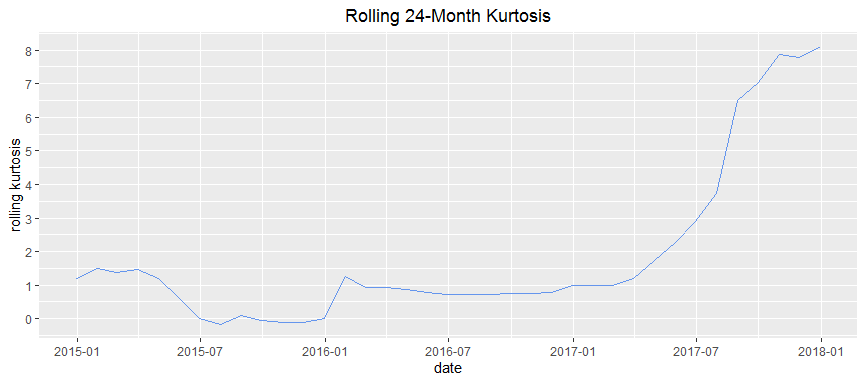
scale\_y\_continuous(breaks = pretty\_breaks(n = 8)) +

scale\_x\_date(breaks = pretty\_breaks(n = 8)) +

ggtitle("Rolling 24-Month Kurtosis") +

labs(y = "rolling kurtosis") +

theme\_update(plot.title = element\_text(hjust = 0.5))



附录：数据（截取部分数据，完整数据在EXCEL文件）

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **date** | **SZ** | **AG** | **BG** | **HS** | **ZXB** |
| 2012-12-31 | 2269.128 | 2376.039 | 244.915 | 4030.855 | 719.479 |
| 2013-01-04 | 2276.992 | 2384.189 | 247.496 | 4018.897 | 715.712 |
| 2013-01-07 | 2285.364 | 2392.591 | 255.955 | 3991.207 | 724.128 |
| 2013-01-08 | 2276.07 | 2382.488 | 262.653 | 4053.622 | 735.959 |
| 2013-01-09 | 2275.34 | 2381.873 | 259.474 | 4041.54 | 736.772 |
| 2013-01-10 | 2283.658 | 2390.587 | 260.267 | 4054.598 | 740.599 |
| 2013-01-11 | 2242.997 | 2348.101 | 254.02 | 4067.849 | 723.19 |
| 2013-01-14 | 2311.74 | 2420.006 | 263.008 | 4030.49 | 747.707 |
| 2013-01-15 | 2325.682 | 2434.615 | 264.322 | 4035.329 | 761.14 |
| 2013-01-16 | 2309.499 | 2417.627 | 263.45 | 3985.291 | 762.794 |
| 2013-01-17 | 2284.909 | 2391.813 | 262.167 | 3980.856 | 754.78 |
| 2013-01-18 | 2317.07 | 2425.509 | 265.247 | 4026.152 | 764.837 |
| 2013-01-21 | 2328.221 | 2436.856 | 273.301 | 4050.093 | 768.962 |
| 2013-01-22 | 2315.139 | 2423.059 | 273.946 | 4016.016 | 757.088 |
| 2013-01-23 | 2320.911 | 2428.999 | 276.728 | 4069.5 | 760.054 |
| 2013-01-24 | 2302.598 | 2410.053 | 269.964 | 4003.379 | 744.166 |
| 2013-01-25 | 2291.304 | 2398.293 | 267.382 | 3971.057 | 742.949 |
| 2013-01-28 | 2346.505 | 2456.025 | 274.781 | 4015.821 | 760.398 |
| 2013-01-29 | 2358.977 | 2469.041 | 277.025 | 4040.17 | 765.706 |
| 2013-01-30 | 2382.475 | 2493.778 | 276.819 | 4018.857 | 766.849 |
| 2013-01-31 | 2385.422 | 2496.833 | 277.773 | 3998.137 | 766.152 |
| 2013-02-01 | 2419.02 | 2532.125 | 279.096 | 4006.099 | 772.888 |
| 2013-02-04 | 2428.154 | 2541.786 | 278.062 | 4053.753 | 766.646 |
| 2013-02-05 | 2433.13 | 2547.005 | 278.424 | 4055.823 | 773.484 |
| 2013-02-06 | 2434.477 | 2548.304 | 280.895 | 4049.948 | 778.673 |
| 2013-02-07 | 2418.53 | 2531.355 | 284.397 | 4104.203 | 779.728 |
| 2013-02-08 | 2432.402 | 2545.855 | 286.405 | 4102.397 | 790.501 |
| 2013-02-18 | 2421.558 | 2534.522 | 284.797 | 4227.567 | 795.161 |
| 2013-02-19 | 2382.914 | 2494.531 | 270.748 | 4217.702 | 783.939 |
| 2013-02-20 | 2397.178 | 2509.359 | 274.533 | 4143.833 | 798.134 |
| 2013-02-21 | 2325.951 | 2434.644 | 269.603 | 4120.851 | 783.085 |
| 2013-02-22 | 2314.164 | 2422.297 | 268.433 | 4105.012 | 784.607 |
| 2013-02-25 | 2325.819 | 2434.371 | 272.404 | 4073.67 | 792.072 |
| 2013-02-26 | 2293.341 | 2400.301 | 270.18 | 4099.351 | 781.865 |
| 2013-02-27 | 2313.22 | 2421.106 | 272.559 | 4128.073 | 783.228 |
| 2013-02-28 | 2365.593 | 2475.974 | 277.624 | 4111.911 | 799.095 |
| 2013-03-01 | 2359.506 | 2469.53 | 278.432 | 4075.9 | 807.521 |
| 2013-03-04 | 2273.404 | 2379.378 | 269.01 | 4048.006 | 782.605 |
| 2013-03-05 | 2326.306 | 2434.878 | 272.509 | 4054.247 | 800.567 |
| 2013-03-06 | 2347.179 | 2456.712 | 275.235 | 4020.89 | 813.273 |
| 2013-03-07 | 2324.293 | 2432.66 | 274.597 | 3992.698 | 805.859 |
| 2013-03-08 | 2318.611 | 2426.809 | 271.924 | 3997.134 | 797.057 |
| 2013-03-11 | 2310.593 | 2418.428 | 270.742 | 3996.622 | 797.45 |
| 2013-03-12 | 2286.605 | 2393.442 | 265.403 | 4006.718 | 781.261 |
| 2013-03-13 | 2263.968 | 2369.804 | 261.595 | 4009.722 | 775.259 |
| 2013-03-14 | 2270.277 | 2376.382 | 262.874 | 4021.968 | 777.089 |
| 2013-03-15 | 2278.401 | 2384.889 | 263.728 | 3993.575 | 775.601 |
| 2013-03-18 | 2240.016 | 2344.646 | 260.619 | 3976.949 | 765.497 |
| 2013-03-19 | 2257.434 | 2362.878 | 262.642 | 3959.395 | 769.509 |
| 2013-03-20 | 2317.374 | 2425.671 | 268.513 | 3930.798 | 788.3 |
| 2013-03-21 | 2324.242 | 2432.872 | 269.043 | 3926.852 | 796.787 |
| 2013-03-22 | 2328.278 | 2437.094 | 269.566 | 3931.249 | 799.644 |
| 2013-03-25 | 2326.715 | 2435.445 | 269.661 | 3944.163 | 799.156 |
| 2013-03-26 | 2297.669 | 2404.935 | 268.541 | 3913.069 | 793.36 |
| 2013-03-27 | 2301.259 | 2408.697 | 268.857 | 3913.446 | 794.387 |
| 2013-03-28 | 2236.302 | 2340.527 | 265.031 | 3921.002 | 769.084 |
| 2013-03-29 | 2236.621 | 2340.922 | 263.795 | 3912.954 | 767.746 |
| 2013-04-01 | 2234.395 | 2338.531 | 264.821 | 3902.687 | 773.378 |
| 2013-04-02 | 2227.74 | 2331.492 | 265.563 | 3889.864 | 766.012 |
| 2013-04-03 | 2225.295 | 2328.988 | 264.115 | 3882.208 | 760.94 |
| 2013-04-08 | 2211.592 | 2314.728 | 260.794 | 3836.501 | 763.214 |
| 2013-04-09 | 2225.775 | 2329.547 | 262.982 | 3822.539 | 769.007 |
| 2013-04-10 | 2226.126 | 2330.009 | 261.056 | 3821.2 | 766.955 |
| 2013-04-11 | 2219.553 | 2323.098 | 260.926 | 3820.782 | 762.323 |
| 2013-04-12 | 2206.78 | 2309.667 | 260.734 | 3817.791 | 755.221 |
| 2013-04-15 | 2181.942 | 2283.896 | 253.066 | 3837.73 | 744.189 |
| 2013-04-16 | 2194.846 | 2297.38 | 255.053 | 3837.818 | 749.581 |
| 2013-04-17 | 2193.796 | 2296.194 | 256.746 | 3842.435 | 754.989 |
| 2013-04-18 | 2197.602 | 2300.198 | 256.793 | 3832.116 | 757.593 |
| 2013-04-19 | 2244.643 | 2349.366 | 263.722 | 3843.143 | 772.114 |
| 2013-04-22 | 2242.169 | 2346.766 | 263.666 | 3831.296 | 776.954 |
| 2013-04-23 | 2184.538 | 2286.363 | 258.637 | 3829.955 | 756.686 |
| 2013-04-24 | 2218.318 | 2321.733 | 262.32 | 3842.606 | 771.725 |
| 2013-04-25 | 2199.307 | 2301.744 | 261.982 | 3837.934 | 760.389 |
| 2013-04-26 | 2177.912 | 2279.54 | 255.502 | 3825.645 | 747.835 |
| 2013-05-02 | 2174.123 | 2275.669 | 253.054 | 3825.99 | 750.068 |
| 2013-05-03 | 2205.497 | 2308.326 | 260.569 | 3829.871 | 764.32 |
| 2013-05-06 | 2231.166 | 2335.172 | 264.002 | 3849.45 | 778.317 |
| 2013-05-07 | 2235.575 | 2339.802 | 264.197 | 3857.046 | 782.295 |
| 2013-05-08 | 2246.3 | 2351.067 | 264.619 | 3845.616 | 787.467 |
| 2013-05-09 | 2232.971 | 2337.069 | 264.057 | 3830.538 | 786.477 |
| 2013-05-10 | 2246.831 | 2351.609 | 264.973 | 3822.093 | 792.497 |
| 2013-05-13 | 2241.92 | 2346.492 | 263.914 | 3834.3 | 796.029 |
| 2013-05-14 | 2217.01 | 2320.356 | 262.35 | 3834.538 | 787.773 |
| 2013-05-15 | 2224.797 | 2328.506 | 263.262 | 3842.712 | 795.367 |
| 2013-05-16 | 2251.806 | 2356.807 | 265.754 | 3795.754 | 804.065 |
| 2013-05-17 | 2282.87 | 2389.407 | 267.557 | 3734.646 | 814.639 |
| 2013-05-20 | 2299.986 | 2407.362 | 268.738 | 3756.088 | 822.587 |
| 2013-05-21 | 2305.114 | 2412.729 | 269.334 | 3752.298 | 832.684 |
| 2013-05-22 | 2302.403 | 2409.891 | 269.033 | 3740.994 | 830.416 |
| 2013-05-23 | 2275.667 | 2381.915 | 265.714 | 3724.675 | 823.382 |
| 2013-05-24 | 2288.533 | 2395.361 | 267.653 | 3721.277 | 835.04 |
| 2013-05-27 | 2293.077 | 2400.137 | 267.772 | 3701.421 | 842.934 |
| 2013-05-28 | 2321.318 | 2429.763 | 269.669 | 3706.057 | 847.584 |
| 2013-05-29 | 2324.017 | 2432.596 | 269.817 | 3694.683 | 854.055 |
| 2013-05-30 | 2317.749 | 2426.039 | 268.992 | 3647.35 | 854.889 |
| 2013-05-31 | 2300.595 | 2408.246 | 263.56 | 3715.921 | 848.664 |
| 2013-06-03 | 2299.252 | 2406.692 | 266.568 | 3731.044 | 844.966 |
| 2013-06-04 | 2272.416 | 2378.645 | 262.545 | 3732.213 | 828.128 |
| 2013-06-05 | 2270.931 | 2377.077 | 262.628 | 3726.795 | 831.075 |
| 2013-06-06 | 2242.11 | 2346.815 | 261.292 | 3707.58 | 816.979 |
| 2013-06-07 | 2210.898 | 2314.045 | 259.805 | 3727.826 | 801.748 |
| 2013-06-13 | 2148.355 | 2248.562 | 252.931 | 3760.852 | 779.393 |
| 2013-06-14 | 2162.041 | 2262.836 | 255.628 | 3770.383 | 793.174 |
| 2013-06-17 | 2156.215 | 2256.775 | 254.146 | 3737.873 | 795.418 |
| 2013-06-18 | 2159.291 | 2260.059 | 253.126 | 3721.891 | 799.017 |
| 2013-06-19 | 2143.454 | 2243.452 | 251.926 | 3712.195 | 795.468 |
| 2013-06-20 | 2084.021 | 2181.115 | 247.758 | 3705.388 | 772.078 |
| 2013-06-21 | 2073.095 | 2169.684 | 246.377 | 3719.559 | 767.688 |
| 2013-06-24 | 1963.235 | 2054.621 | 235.102 | 3743.469 | 722.109 |
| 2013-06-25 | 1959.508 | 2050.785 | 233.287 | 3728.598 | 719.363 |
| 2013-06-26 | 1951.495 | 2042.126 | 238.152 | 3747.884 | 732.191 |
| 2013-06-27 | 1950.013 | 2040.674 | 235.858 | 3729.747 | 719.582 |
| 2013-06-28 | 1979.206 | 2071.259 | 238.663 | 3667.181 | 719.61 |

（来源于国泰安数据库）