Report

Ru JIA

2016年6月16日

Read ME

- 我把所有文件都打包在一个 R project 里面，可以用Rstudio打开文件 `quant v1.2.Rproj`
- `Analysis_v1.2.R` 是数据读取和初步分析
- `Trade_v1.2.R` 是交易策略函数
- `dt.Rdata` 是一个读文件+整理后的阶段性成果，考虑到运行比较慢+跨平台问题，我直接把这个中间结果存成Rdata格式，以便在 `Trade_v1.2.R` 中可以直接load

这个版本更新内容:

1. 样本内/外起止的日期作为函数参数，可以调节。
2. 原始数据不用excel了，直接从csv文件读取。文件夹 `csvStockFiles` 里面是合并好的股票数据csv文件，一只股票一个文件。目前只有一只股票“000001SH”，是我手工合并的（excel里复制粘贴,再另存为csv格式）后续可以添加更多。。。
3. 做一个简单的交易策略：在out-of-sample 阶段,寻找奇异点,即超过90% 或者80%的quantile 的时间点作为买入信号。买入之后，5%收益率就卖出，2%亏损止损。

文件`Analysis_v1.2.R`的运行结果

（最后两行代码可以改样本内外起止时间参数： `when = ...` ）

```
# 分析股票交易数据
# By: JiaRu, jiaru2014@126.com
# R version 3.2.3
# Platform: x86_64-apple-darwin13.4.0 (64-bit)
# Date: 2016-06-16
#
# Version 1.2
# 1. 样本内/外起止的日期作为函数参数，可调节
# 2. 原始数据不用excel了，直接从csv文件读取。
# ============================================================
========

rm(list = ls())

require(data.table)
require(magrittr)  # 只是为了用 `%>%`
require(ggplot2)
require(assertthat)  # a tool for Defensive Programming
require(stringr)  # 文字处理


# 1. 读文件 ==========================================

# 函数ReadStockFile()
# Argus:
#   stock_num: 股票代码，要加引号！！！
# Return:
#   a data.table of 5 columns
#     type分别为：股票代码 chr, 日期 chr, 时间 chr, 价格 num, 成交
量 int

ReadStockFile <- function(stock_num)
{
  # 检查参数 ---------------------
  assert_that(stock_num %in% stock_list)

  # 读文件 -------------------------
  tryCatch(
    {
      dt <- read.csv(
        file.path("csvStockFiles", paste0(stock_num, ".cs
v")),
        header = TRUE,
        sep = ";",
        colClasses = c(rep("character", 3), "numeric", "integ
er")
```

```
      ) %>% as.data.table()
      message("读文件成功: ", stock_num)

    },
    error = function(e) {message("读取文件出错! "); e}
  )


  # 整理数据格式 ----------------------------
  dt[, DateTime := as.POSIXct(paste(TDate, MinTime), format =
"%Y%m%d %H%M", tz = "PRC")]
  dt[, TDate := as.Date(TDate, "%Y%m%d")]
  dt[, return := c(NA, diff(EndPrc)/EndPrc[-.N])]  # 计算收益
率
  message("整理数据格式完毕! ")

  setkey(dt, "DateTime")

  return(dt)
}



# 2. 检查数据是否有异常  ======================================
# 函数CheckStockData()
# Arugs:
#   dt: data.table from last step
# Return:
#   cleaned data.table

CheckStockData <- function(dt)
{
  # 看有没有重复的行 ----------------------------
  tryCatch(
    {
      if (uniqueN(dt) == nrow(dt)){
        message("没有重复行。")
      } else {
        dt <- unique(dt)
        message("数据集中有重复行，已删除重复行。")
      }
    },
    error = function(e) message("检查是否有重复行时出错。已跳过此
步骤。")
  )
```

```
  # 看每一天时间是否齐全。----------------------------
  # 正常来说每天应有 4*60+1 = 241 行
  tryCatch(
    {
      checktime <- dt[, .N, by = TDate][N != 241]
      if (nrow(checktime) == 0) {
        message("每个交易日的交易信息都是完整的！")
      } else {
        message("以下交易日缺少部分交易信息：")
        print(checktime)
      }
    },
    error = function(e) message("检查每天时间是否齐全时出错。跳
过。")
  )

  return(dt)
}




# 3. 分析  ======================================
# 函数AnalysisDis()分析样本内/外收益率or成交量的分布
# Argus：
#   dt: 清理好的股票数据。
#   what: "return" or "MinTq"
#   when = c(in_start, in_end, out_start, out_end): 样本内/外
起止点。
#       写成可直接转化成日期格式的形式：“2015-01-05”
#       长度可以是3或4

AnalysisDis <- function(dt, what = "return",
    when = c("2015-01-05", "2015-12-31", "2016-01-04", "2016-
01-29"))
{
  # 检查参数 --------------------------
  when <- as.Date(when)
  assert_that(
    what %in% c("return", "MinTq") && length(what) == 1,
    length(when) == 4,
    when[1] < when[2] && when[2] <= when[3] && when[3] < when
[4]
  )
  # 取样本内/外数据 --------------------
  # 这里对于成交量(all >= 0) 取对数
```

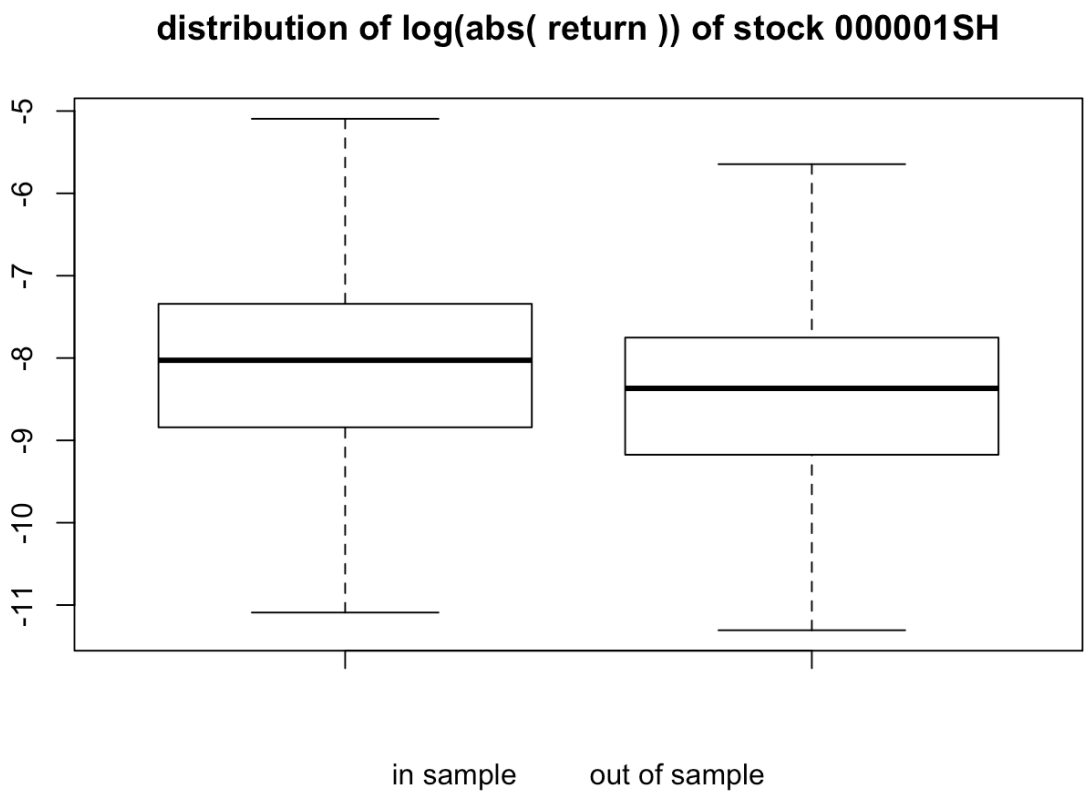
```
  # 对于retrun, 先取绝对值再取对数。
  #
  dt_in <-
    dt[TDate >= when[1] & TDate <= when[2], what, with = FALS
E] %>%
    unlist(use.names = FALSE) %>%
    abs() %>%
    log()
  dt_out <-
    dt[TDate >= when[3] & TDate <= when[4], what, with = FALS
E] %>%
    unlist(use.names = FALSE) %>%
    abs() %>%
    log()

  dt_in_g <<- dt_in  # just for debugging
  dt_out_g <<- dt_out

  # 画图：样本内/外retrun/Vol分布----------------------
  # 这里只用R中的基础绘图系统 boxplot()
  # 暂时先不用histogram,因为根据不同的数据要调整坐标轴，很难写成函数

  boxplot(dt_in, dt_out,
          xlab = "in sample         out of sample",
          main = sprintf("distribution of log(abs( %s )) of s
tock %s", what, stock_num),
          outline = FALSE)

  return(0)
}

# 4. 测试 ==================================================

# 注:
# 文件夹“csvStockFiles“里面是合并好的股票数据csv文件。
# 每个股票只有一个文件，是我手工合并的（excel里复制粘贴），再另存为csv
格式
# 暂时只做了了一个股票，作为测试，后续可以添加更多。。

# 列出数据集中所有的股票代码
stock_list <-
  list.files("csvStockFiles") %>%
  str_replace(pattern = "\\.csv", replacement = "")
cat("以下是数据集中所有可分析的股票代码: ", stock_list, sep = "\n")
```

```
## 以下是数据集中所有可分析的股票代码:
## 000001SH
```

```
# 以"000001SH"为例，测试
stock_num <- "000001SH"
dt <- ReadStockFile(stock_num)
dt <- CheckStockData(dt)
```

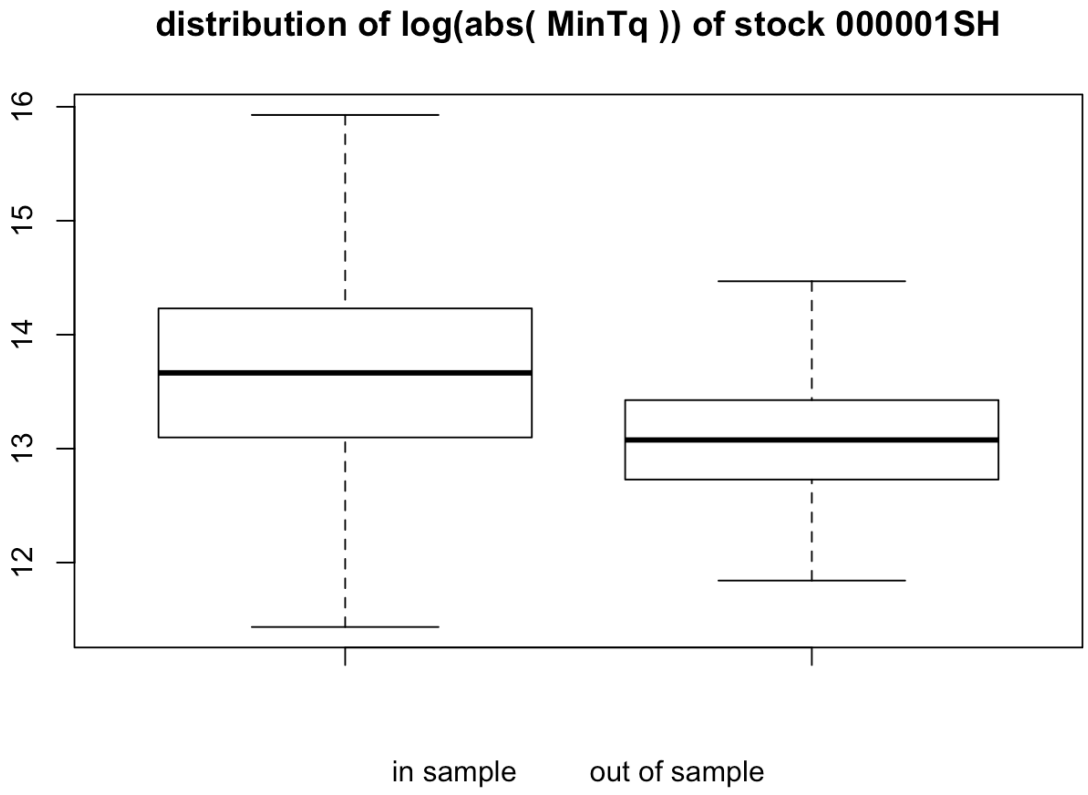
```
##         TDate   N
## 1: 2014-07-15 240
## 2: 2015-08-21 240
```

```
# 以下参数可以改:
AnalysisDis(dt, what = "return",
            when = c("2014-01-01", "2016-05-01", "2016-05-0
2", "2016-05-31"))
```




```
## [1] 0
```

```
AnalysisDis(dt, what = "MinTq",
            when = c("2014-01-01", "2016-05-01", "2016-05-0
2", "2016-05-31"))
```




```
## [1] 0
```

```
# 把数据dt存成Rdata格式，方便后面使用。
# save(dt, file = "dt.Rdata")
```

文件`Trade_v1.2.R`的运行结果

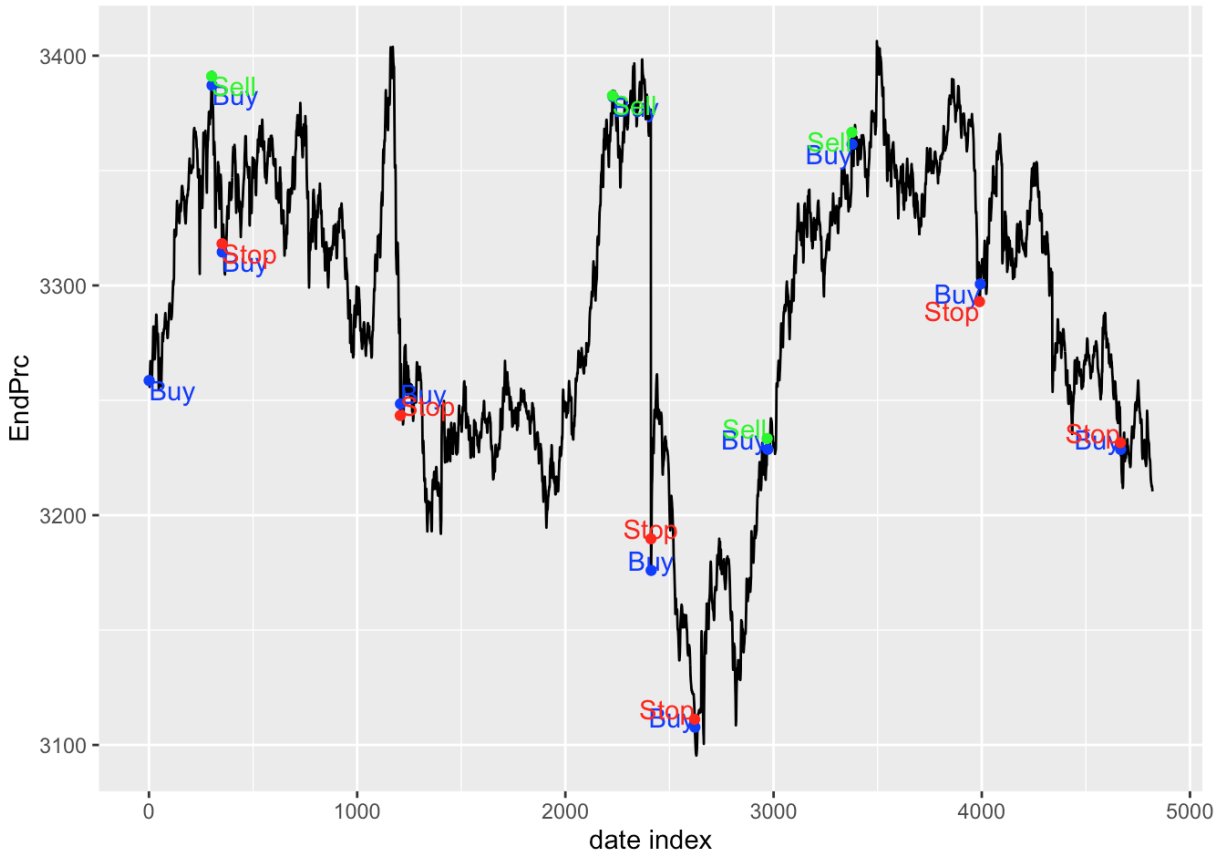
`#` 测试 `========` 后面调用函数 `Trade` 可以使用不同的参数： `sell`，`stop`，`when`）

```
# 设计交易策略
# By: JiaRu, jiaru2014@126.com
# R version 3.2.3
# Platform: x86_64-apple-darwin13.4.0 (64-bit)
# Date: 2016-06-16
#
# Version 1.2
# 做一个简单的交易策略:
# 在out-sample里面找出那些 80% quantile之外的时间,标记出来,
# 利用这些信息来设计buy or sell的signal.
# 对单个股票而言，利用一定时间的样本数据得到一分钟成交量和回报率的联合分布,
# 然后在out-of-sample 阶段,寻找那些奇异点,
# 即超过90% 或者80%的quantile 的时间点，标出买入信号和卖出信号.
# 分析这些时间点对应的下一分钟（或其他形式的时间段）的回报率
# 买入之后，5%收益率就卖出，2%亏损就进行止损，但是当日不能操作
# ============================================================
========


rm(list = ls())

require(data.table)
require(magrittr)
require(ggplot2)
require(assertthat)
require(stringr)

load("dt.Rdata") # 为了方便这里就直接load Rdata格式的数据了。

Trade <- function(
  dt,
  quantile = 0.9,  # 奇异点定义为 90% quantile
  sell = 0.05,  # 5%收益率就卖出
  stop = 0.02,  # 2%亏损就进行止损
  when = c("2015-01-05", "2015-12-31", "2016-01-04", "2016-01
-29")
)
{
  # 定义样本内/样本外数据 ------------------------------------
  when <- as.Date(when)
  dt_in <- dt[TDate >= when[1] & TDate <= when[2]]
  dt_out <- dt[TDate >= when[3] & TDate <= when[4]]

  # 找奇异点 ------------------------------------
  #（此处只看reutrn,暂不考虑return & Volumn 联合分布）
```

```
  # 由于return以0为中心大致呈正态分布，ie. 对称的，奇异点定义为上下10% quantile
  # 如果是看Volumn呢？ Volumn 大致呈对数正态分布。都是正的，不对称。？？？
  q1 <- (1 - quantile) / 2
  q2 <- 1 - q1
  q <- quantile(dt_in$return, probs = c(q1, q2))
  q_up <- q[2]
  q_down <- q[1]

  # 找出买入信号
  buypoint_index <- which(dt_out$return < q_down | dt_out$return > q_up)
  dt_buysignal <- dt_out[buypoint_index][, .(DateTime, Signal = "BuyPoint")]
  dt_trade <- merge(dt_out, dt_buysignal, by = "DateTime", all = TRUE)
  dt_trade[is.na(dt_trade)] <- ""  # 把Signal中的NA换成空字符串，因为NA不能做“==”运算

  # 交易 ------------------------------------------------
  # 交易流程:
  # 按时间顺序遍历整个out sample,
  # 遇到一个买入信号时，如果没有持仓，就（全仓）买入
  # 接下来如果上涨到5%就卖出获利了结，如果跌破2%就止损
  # 持仓期间如果再遇到买入信号则跳过。

  state <- 0  # 0表示没有持仓，1表示满仓
  sellpoint <- NaN  # 获利平仓点
  stoppoint <- NaN  # 止损点
  for (i in 1:nrow(dt_trade)) {

    if (state == 1) { # 有仓位。。。

      if (dt_trade[i, EndPrc] > sellpoint) {
        dt_trade[i, Close := "Sell"]  # 若涨到sellpoint, 卖出获利（sell）
        state <- 0
      } else if (dt_trade[i, EndPrc] < stoppoint) {
        dt_trade[i, Close := "Stop"]  # 若跌到stoppoint, 止损（stop）
        state <- 0
      } else {dt_trade[i, Close := ""]}

    } else {  # (state == 0) # 没有仓位。。。
```

```
      if (dt_trade[i, Signal == "BuyPoint"]) {
        dt_trade[i, Buy := "Buy"]
        state <- 1
        buyprice <- dt_trade[i, EndPrc]
        sellpoint <- buyprice * (1 + sell)
        stoppoint <- buyprice * (1 - stop)
      }

    }
    message("Looping ", round(i/nrow(dt_trade)*100), " %
...") # 进度条
  }
  dt_trade[is.na(dt_trade)] <- ""  # 把Signal中的NA换成空字符串

  # 计算总收益率
  stoptime <- nrow(dt_trade[Close == "Stop"])
  selltime <- nrow(dt_trade[Close == "Sell"])
  totalreturn <- selltime * sell - stoptime * stop
  message("Total return is ", totalreturn, " Sell: ", selltim
e, " Stop: ", stoptime)

  return(dt_trade)
}


# 测试 ====================================================
# 以下参数可以修改:
dt_trade <- Trade(
  dt,
  quantile = 0.9,
  sell = 0.04,
  stop = 0.02,
  when = c("2013-06-01", "2014-12-31", "2015-01-01", "2015-02
-01")
)

dt_trade[, index := seq_along(DateTime)]

g <- ggplot(dt_trade, aes(x = index, y = EndPrc)) +
  geom_line() +
  geom_point(data = dt_trade[Buy == "Buy"], col = "blue") +
  geom_text(data = dt_trade[Buy == "Buy"], aes(label = "Bu
y"), col = "blue", vjust = "inward", hjust = "inward") +
  geom_point(data = dt_trade[Close == "Sell"], col = "green")
+
  geom_text(data = dt_trade[Close == "Sell"], aes(label = "Se
```

```
ll"), col = "green", vjust = "inward", hjust = "inward") +
  geom_point(data = dt_trade[Close == "Stop"], col = "red") +
  geom_text(data = dt_trade[Close == "Stop"], aes(label = "St
op"), col = "red", vjust = "inward", hjust = "inward") +
  xlab("date index")
g
```




```
# 列出交易记录:
dt_trade[
  Buy=="Buy" | Close == "Sell" | Close == "Stop",
  .(DateTime, return, Buy, Close)
  ]
```

```
##                DateTime        return Buy Close
##  1: 2015-01-05 09:30:00  0.2158352782 Buy
##  2: 2015-01-06 10:29:00  0.0012672838      Sell
##  3: 2015-01-06 10:31:00 -0.0017532857 Buy
##  4: 2015-01-06 11:19:00 -0.0019580879      Stop
##  5: 2015-01-06 11:20:00 -0.0010855407 Buy
##  6: 2015-01-12 09:31:00 -0.0045509609      Stop
##  7: 2015-01-12 09:32:00  0.0015656482 Buy
##  8: 2015-01-16 10:26:00  0.0017490418      Sell
##  9: 2015-01-16 10:29:00 -0.0008018275 Buy
## 10: 2015-01-19 09:30:00 -0.0479901770      Stop
## 11: 2015-01-19 09:31:00 -0.0043094597 Buy
## 12: 2015-01-19 14:30:00 -0.0009938149      Stop
## 13: 2015-01-19 14:31:00 -0.0011047302 Buy
## 14: 2015-01-21 10:46:00  0.0013068491      Sell
## 15: 2015-01-21 10:49:00 -0.0008309161 Buy
## 16: 2015-01-23 09:31:00  0.0028494866      Sell
## 17: 2015-01-23 09:32:00 -0.0015418833 Buy
## 18: 2015-01-27 13:12:00 -0.0013910382      Stop
## 19: 2015-01-27 13:16:00  0.0015633719 Buy
## 20: 2015-01-30 10:56:00 -0.0013622020      Stop
## 21: 2015-01-30 10:57:00 -0.0008995742 Buy
##                DateTime        return Buy Close
```