



R语言金融数据分析 - 张丹

前言

- 本人程序员、架构师，从编程入门到今天，一直深信着Java是改变世界的语言，Java已经做到了，而且一直很辉煌。但当Java的世界越来越大，变得无所不能的时候，反而不够专业，给了其他语言发展的机会。
- 我已使用Java语言 11年，R语言 3年，Node 1年，我认为“R语言是2014年最具有应用前景的编程语言”。

目录

- R语言介绍
- 时间背景知识
- 金融案例分析
- R语言实现均线模型

R语言介绍

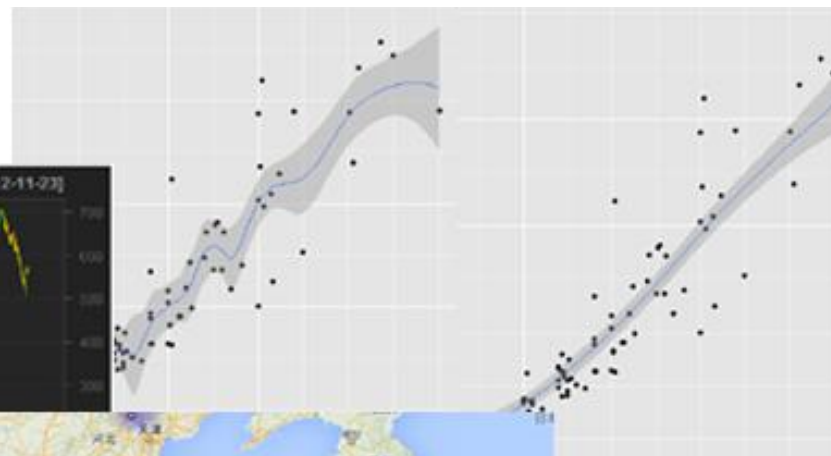
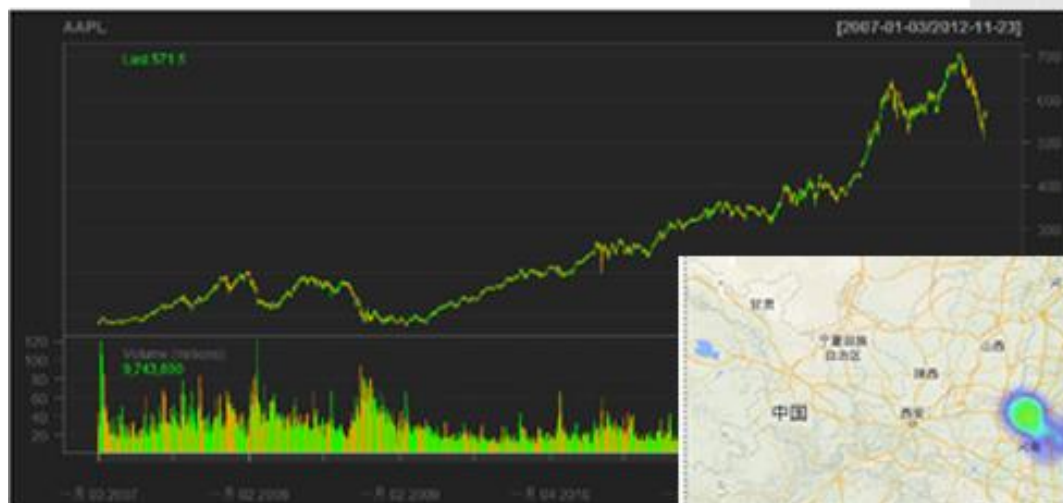
- R是统计学家发明的语言，天生具有统计的基因。
- 从我开始学习R语言，我就开始了知识的跨界思考。统计基于概率论，概率论又基于数学，用计算机的方式编程，解决某个领域的实际问题。简单一算，4个学科知识的交集，决定着我們解决问题的能力。统计的基因，让R语言与众不同！

R语言的应用场景

- R可以做所有SAS做的事情。
- R应用最热门的领域：
 - 统计分析：包括统计分布，假设检验，统计建模
 - 金融分析：量化策略，投资组合，风险控制，时间序列，波动率
 - 数据挖掘：数据挖掘算法，数据建模，机器学习
 - 互联网：推荐系统，消费预测，社交网络
 - 生物信息学：DNA分析，物种分析
 - 生物制药：生存分析，制药过程管理
 - 全球地理科学：天气，气候，遥感数据
 - 数据可视化：静态图，可交互的动态图，社交图，地图，热图，与各种Javascript库的集成

R语言的应用场景

- 回归分析，金融分析，地图分析



金融背景知识：移动平均线

- 移动平均线(MA, Moving average) 是以道·琼斯的“平均成本概念”为理论基础，采用统计学中“移动平均”的原理，将一段时期内的股票价格平均值连成曲线，用来显示股价的历史波动情况，进而反映股价指数未来发展趋势的技术分析方法。它是道氏理论的形象化表述。
- 移动平均线的计算方法就是求连续若干天的收盘价的算术平均。天数就是MA的参数。在技术分析领域中，移动平均线是必不可少的指标工具。移动平均线利用统计学上的“移动平均”原理，将每天的市场价格进行移动平均计算，求出一个趋势值，用来作为价格走势的研判工具。

金融背景知识：计算公式

- 计算公式：
- $MA = (C1+C2+C3+C4+C5+....+Cn)/n$,C为收盘价，n为移动平均周期数.
- 5日移动平均价格计算方法为：
- $MA5 = (前四天收盘价+前三天收盘价+前天收盘价+昨天收盘价+今天收盘价) / 5$

金融背景知识：按时间长短分类

- 移动平均线依时间长短可分为三种，即短期移动平均线，中期移动平均线，长期移动平均线。
- 短期移动平均线，一般以5日或10日为计算期间
- 中期移动平均线，大多以30日、60日为计算期间
- 长期移动平均线，大多以100天和200天为计算期间。

金融背景知识：按数据处理方法分类

- 简单移动平均线(SMA)：又称“算术移动平均线”，是指对特定期间的收盘价进行简单平均化的意思。一般所提及之移动平均线即指简单移动平均线(SMA)。
- 加权移动平均线(WMA):加权移动平均线(Weighted Moving Average 简称WMA)，是一种按时间进行加权运算的移动平均线。时间越近越近的价格，权重越大。计算方式是基于加权移动平均线日数，将每一个之前日数比重提升。每一价格会乘以一个比重，最新的价格会有最大的比重，其之前的每一日的比重将会递减。加权移动平均线是移动平均线(MA)的改良。
- 指数平滑移动平均线(EMA):指数平滑移动平均线EXPMA(Exponential Moving Average)，为解决一旦价格已脱离均线差值扩大，而均线未能立即反应，EXPMA可以减少类似缺点。

金融背景知识：均线定义

- 在日K线图中除了标准的价格K线以外，另外还有4条线，分别是白线、黄线、紫线、绿线依次分别表示：5日、10日、20日和60日移动平均线，通过这4条线与价格K线的交叉，就可以形成不同的均线模型。

金融背景知识：乐视网

- 以乐视网(300104)股票日K线图为例，截取2012年8月到2014年7月的股价数据。



- 最低价是13.91，出现在2012年12月；最高价55.50，出现在2014年1月。这段时期，可以看到乐视网的股价一路震荡向上，绿色线为60日均线平滑了股价，趋势性比较明显。

金融背景知识：均线模型

- 利用均线平滑的特点，可以发现均线与价格K线会有交叉，各均线之间也有交叉，我们可以通过这些交叉点判断交易信号。
- 黄金交叉，当10日均线由下往上穿越30日均线，10日均线在上，30日均线在下，其交叉点就是黄金交叉，黄金交叉是多头的表现，出现黄金交叉后，后市会有一定的涨幅空间，这是进场的最佳时机。
- 死亡交叉，当30日均线与10日平均线交叉时，30日均线由下往上穿越10日平均线，形成30日平均线在上，10日均线在下时，其交点称之为“死亡交叉”，“死亡交叉”预示空头市场来临，股市将下跌此时是出场的最佳时机。

金融背景知识：局限性

- 移动平均线是股价定型后产生的图形，反映较慢，只适用于日间交易。
- 移动平均线不能反映股价在当日的变化及成交量的大小，不适用于日内交易。
- 移动平均线是趋势性模型，如果股价未形成趋势，只是频繁波动，模型不适用。

金融案例分析

- 我们利用R语言对股票数据的进行操作，来实现一个均线模型的实例。
- 以IBM的股票为例，取 2010-01-01 至 2014-07-09，收盘时交易数据。
- 用均线模型进行历史回测。
 - 一条均线模型：20日均线
 - 二条均线模型：5日均线 和20日均线
- 以10W本金进行满仓交易。
- 分别画出2个模型的资金曲线
- 比较2个模型的业绩

用R语言实现均线模型

- R语言本身提供了丰富的金融函数工具包，quantmod包就是最常用的一个，另外还要配合时间序列包zoo和xts，指标计算包TTR，可视包ggplot2等一起使用。
- 关于zoo包和xts包的详细使用，可以参考文章
- [R语言时间序列基础库zoo](#)
- [可扩展的时间序列xts](#)

用R语言实现均线模型：从互联网下载数据

- 我们首先利用quantmod包，从互联网下载股票数据，并以CSV格式保存到本地。
- 利用quantmod包的getSymbols()函数，默认会通过Yahoo的金融开放API下载数据，我们选择IBM的股票数据，从2010-01-01到今天2014-07-09的4年多的日间交易数据。

```
#加载工具包
> library(plyr)
> library(quantmod)
> library(TTR)
> library(ggplot2)
> library(scales)

#下载数据
> download<-function(stock, from="2010-01-01"){
+ df<-getSymbols(stock, from=from, env=environment(), auto.assign=FALSE) #下载数据
+ names(df)<-c("Open", "High", "Low", "Close", "Volume", "Adjusted")
+ write.zoo(df, file=paste(stock, ".csv", sep=""), sep=" ", quote=FALSE) #保存到本地
+}

#本地读数据
> read<-function(stock){
+ as.xts(read.zoo(file=paste(stock, ".csv", sep=""), header = TRUE, sep=" ", format="%Y-%m-%d"))
+}
```

用R语言实现均线模型：从互联网下载数据

- 数据类型为xts格式的时间序列，数据包括7个列，以日期做索引列，其他6列分别为 开盘价(Open), 最高价(High), 最低价(Low), 收盘价(Close), 交易量(Volume), 调整价(Adjusted)。

```
> stock<-"IBM"
> download(stock, from='2010-01-01')
> IBM<-read(stock)

# 查看数据类型
> class(IBM)
[1] "xts" "zoo"

# 查看前6条数据
> head(IBM)
      Open  High  Low  Close  Volume Adjusted
2010-01-04 131.18 132.97 130.85 132.45 6155300    121.91
2010-01-05 131.68 131.85 130.10 130.85 6841400    120.44
2010-01-06 130.68 131.49 129.81 130.00 5805300    119.66
2010-01-07 129.87 130.25 128.91 129.55 5840600    119.24
2010-01-08 129.07 130.92 129.05 130.85 4197200    120.44
2010-01-11 131.06 131.06 128.67 129.48 5730400    119.18
```

用R语言实现均线模型：实现简单的蜡烛图

- 使用quantmod包的chartSeries()函数，我们可以画出可视化效果还不错的蜡烛图。

```
> chartSeries(IBM)
```



用R语言实现均线模型：实现简单的蜡烛图

■ 带指标的蜡烛图

```
> chartSeries(IBM, TA = "addVo(); addSMA(); addEnvelope(); addMACD(); addROC()")
```



- 非常简单的2个函数，就可以实现股票数据的可视化。当然，这个功能是封装好的通用的函数，如果我们要自定策略模型，就需要自己写代码来实现了。

用R语言实现均线模型：自定义均线图

- 通过自定义的方式，我们就可以脱离quantmod包了。
- 我们需要自定义均线指标：
 - 日期时间序列为索引
 - 收盘价做为价格指标
 - 不考虑成交量及其他维度字段
 - 取2010-01-01至2012-01-01，形成趋势的数据
 - 画出价格曲线，5日均线，20日均线，60日是均线

用R语言实现均线模型：自定义均线图

```
#移动平均
> ma<-function(cdata, mas=c(5, 20, 60)){
+   ldata<-cdata
+   for (m in mas){
+     ldata<-merge(ldata, SMA(cdata, m))
+   }
+   ldata<-na.locf(ldata, fromLast=TRUE)
+   names(ldata)<-c('Value', paste('ma', mas, sep=''))
+   return(ldata)
+ }

# 均线图
> drawLine<-function(ldata, title="Stock_MA", sDate=min(index(ldata)), eDate=max(index(ldata)),
+   g<-ggplot(aes(x=Index, y=Value), data=fortify(ldata[, 1], melt=TRUE))
+   g<-g+geom_line()
+   g<-g+geom_line(aes(colour=Series), data=fortify(ldata[, -1], melt=TRUE))
+   g<-g+scale_x_date(labels=date_format("%Y-%m"), breaks=date_breaks("2 months"), limits =
+   g<-g+xlab("") + ylab("Price")+ggtitle(title)
+
+   if(out) ggsave(g, file=paste(title, ".png", sep=""))
+   else g
+ }
```

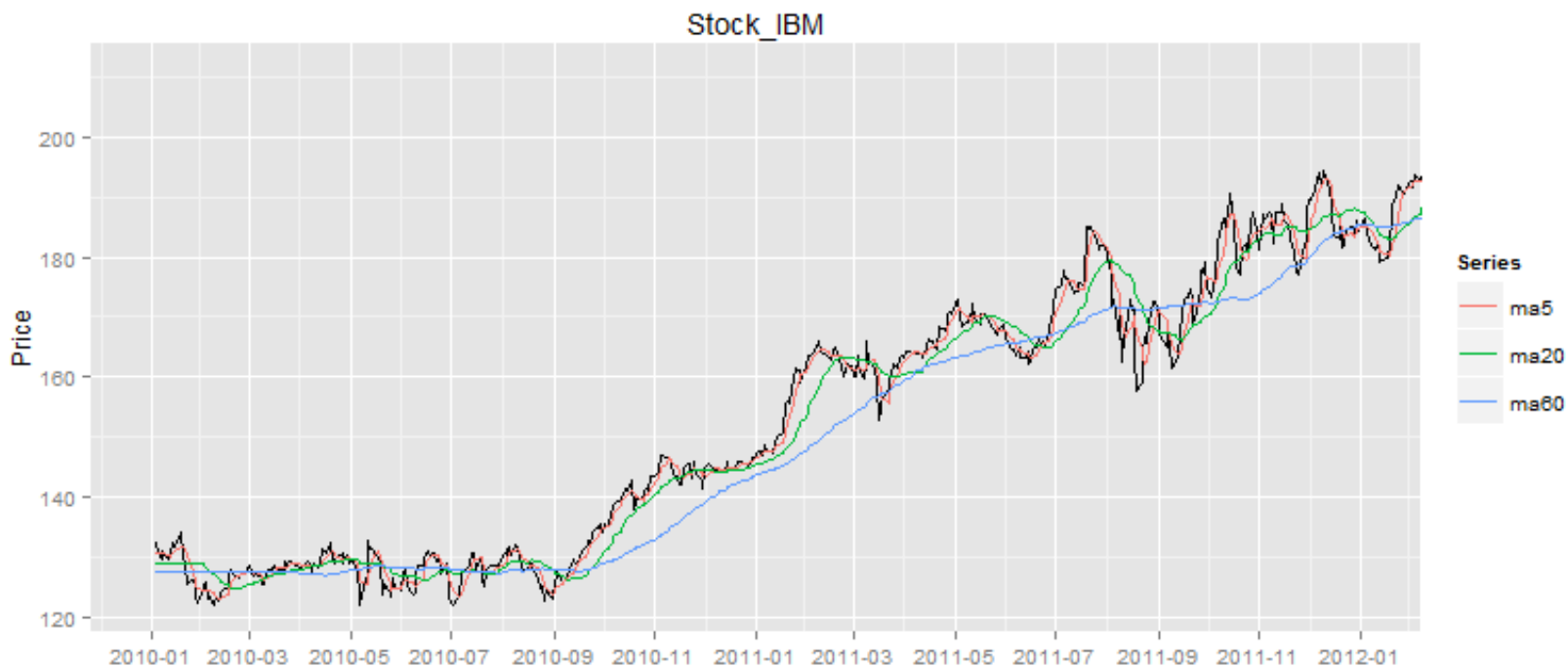
运行程序

```
> ldata<-ma(cdata, c(5, 20, 60)) #选择滑动平均指标
> drawLine(ldata, title, sDate, eDate) #画图
> cdata<-IBM['2010/2012']$Close
> title<-“Stock_IBM” #图片标题
> sDate<-as.Date("2010-1-1") #开始日期
> eDate<-as.Date("2012-1-1") #结束日期

> ldata<-ma(cdata, c(5, 20, 60)) #选择滑动平均指标
> drawLine(ldata, title, sDate, eDate) #画图
```

用R语言实现均线模型：自定义均线图

- 通过自己封装的移动平均函数和可视化函数，就实现了与交易软件中类似的日K线图和多条均线结合的可视化输出。



用R语言实现均线模型：一条均线的交易策略

- 基于上面的定义的均线函数，我们就可以设计自己的交易策略模型了。
- 模型设计思路：
 - 1. 以股价和20日均线的交叉，进行交易信号的判断。
 - 2. 当股价上穿20日均线则买入(红色)，下穿20日均线卖出(蓝色)。

用R语言实现均线模型：一条均线的交易策略

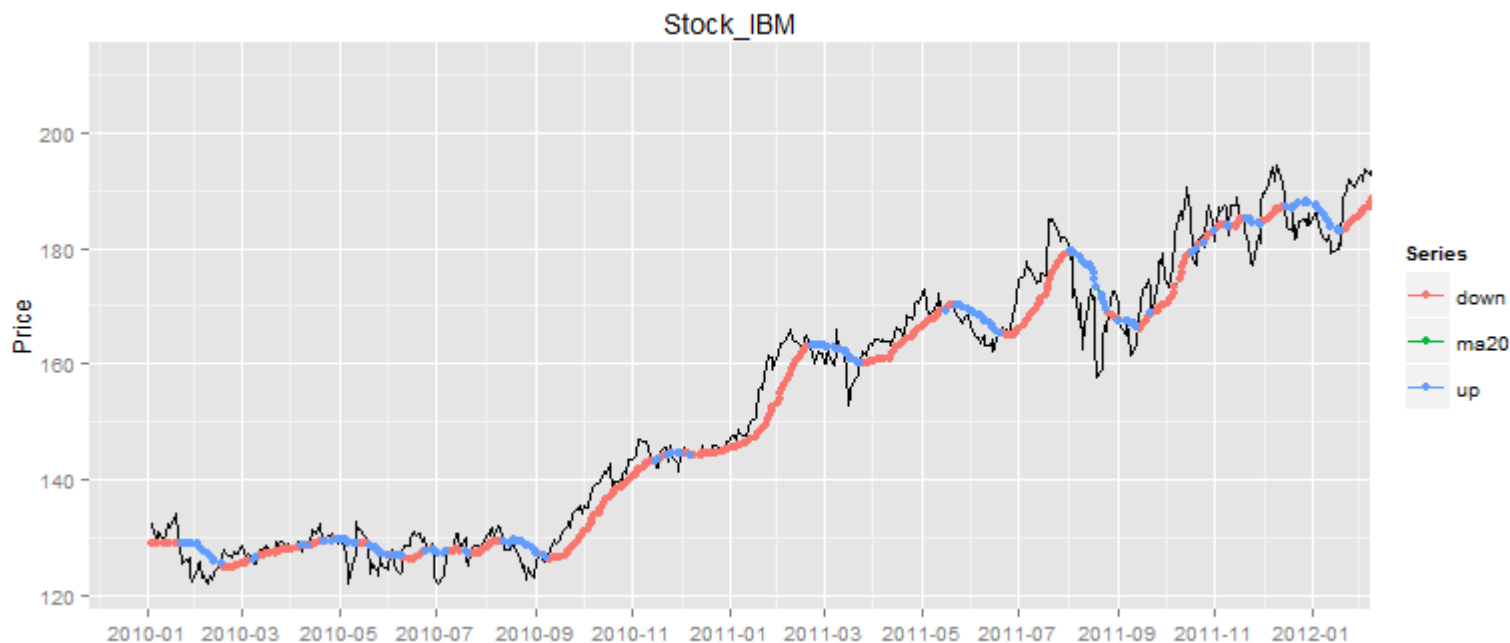
```
# 均线图+散点
> drawPoint<-function(ldata, pdata, titie, sDate, eDate) {
+   g<-ggplot(aes(x=Index, y=Value), data=fortify(ldata[, 1], melt=TRUE))
+   g<-gtgeom_line()
+   g<-gtgeom_line(aes(colour=Series), data=fortify(ldata[, -1], melt=TRUE))
+   g<-gtgeom_point(aes(x=Index, y=Value, colour=Series), data=fortify(pdata, melt=TRUE))
+   g<-gtscale_x_date(labels=date_format("%Y-%m"), breaks=date_breaks("2 months"), limits = c
+   g<-gtxlab("") + ylab("Price")+ggtitle(title)
+   g
+ }
```

```
# 散点数据
> genPoint<-function(pdata, ldata) {} #代码省略
> pdata<-genPoint(pdata, ldata)
> head(pdata)
  Index Series  Value
1 2010-01-04  down 128.7955
2 2010-01-05  down 128.7955
3 2010-01-06  down 128.7955
4 2010-01-07  down 128.7955
5 2010-01-08  down 128.7955
6 2010-01-11  down 128.7955
```

用R语言实现均线模型：一条均线的交易策略

- 以散点覆盖20日均线，红色点为买入持有，蓝色点为卖出空仓。

```
> drawPoint(ldata, pdata, title, sDate, eDate) #画图
```



用R语言实现均线模型：交易信号

- 用股价和20日均线价格做比较，把股价大于均线的部分用蓝色表示，股价小于均线的部分用红色表示。我们看到图中，蓝色点和红色点在20日均线上交替出现，我们可以在每次红色出现的第一个点买入股票，然后在蓝色的第一个点卖出股票，直观看上去的感觉还是不错的。
- 我们要找出这些交易信号点，做量化的统计，看看到底能不能赚钱。

用R语言实现均线模型：交易信号

- 一共有72条交易记录，买卖各占一半。

```
#交易信号
> Signal<-function(cdata,pdata){} #代码省略
> tdata<-Signal(cdata,pdata)
> tdata<-tdata[which(as.Date(row.names(tdata)) head(tdata)
      Value op
2010-01-04 132.45 B
2010-01-22 125.50 S
2010-02-17 126.33 B
2010-03-09 125.55 S
2010-03-11 127.60 B
2010-04-08 127.61 S

# 交易记录
> nrow(tdata)
[1] 72
```

用R语言实现均线模型：模拟交易

- 我们要利用交易信号数据，进行模拟交易。我们设定交易参数，以\$10W为本金，满仓买入或卖出，手续费为0，传入交易信号。

```
#模拟交易
```

```
#参数：交易信号, 本金, 持仓比例, 手续费比例
```

```
> trade<-function(tdata, capital=100000, position=1, fee=0){} #代码省略
```

```
> result1<-trade(tdata, 100000)
```

```
# 查看每笔交易
```

```
> head(result1$ticks)
```

	Value	op	cash	amount	asset	diff
2010-01-04	132.45	B	0.25	755	100000.00	0.00
2010-01-22	125.50	S	94752.75	0	94752.75	-5247.25
2010-02-17	126.33	B	5.25	750	94752.75	0.00
2010-03-09	125.55	S	94167.75	0	94167.75	-585.00
2010-03-11	127.60	B	126.55	737	94167.75	0.00
2010-04-08	127.61	S	94175.12	0	94175.12	7.37

```
# 盈利的交易
```

```
> head(result1$rise)
```

	Value	op	cash	amount	asset	diff
2010-03-11	127.60	B	126.55	737	94167.75	0.00
2010-04-08	127.61	S	94175.12	0	94175.12	7.37
2010-07-22	127.47	B	108.79	633	80797.30	0.00
2010-08-12	128.30	S	81322.69	0	81322.69	525.39
2010-09-09	126.36	B	120.40	632	79979.92	0.00
2010-11-16	142.24	S	90016.08	0	90016.08	10036.16

```
# 亏损的交易
```

```
> head(result1$fall)
```

	Value	op	cash	amount	asset	diff
2010-01-04	132.45	B	0.25	755	100000.00	0.00
2010-01-22	125.50	S	94752.75	0	94752.75	-5247.25
2010-02-17	126.33	B	5.25	750	94752.75	0.00
2010-03-09	125.55	S	94167.75	0	94167.75	-585.00
2010-04-09	128.76	B	51.56	731	94175.12	0.00
2010-04-12	128.36	S	93882.72	0	93882.72	-292.40

用R语言实现均线模型：模拟交易

- 通过模拟交易，我们就能精确地算出每笔交易的盈利情况了。你相信么，有56笔交易其实是亏损的，16笔交易是有盈利的。
- 查看最后的资金情况。
- 最后，资金剩余96363.76元，也就是我们亏了3636.24元。

```
> tail(result1$ticks,1)
      Value op  cash amount  asset  diff
2011-12-21 181.47 S 96363.76    0 96363.76 -3063.87
```

用R语言实现均线模型：资金曲线

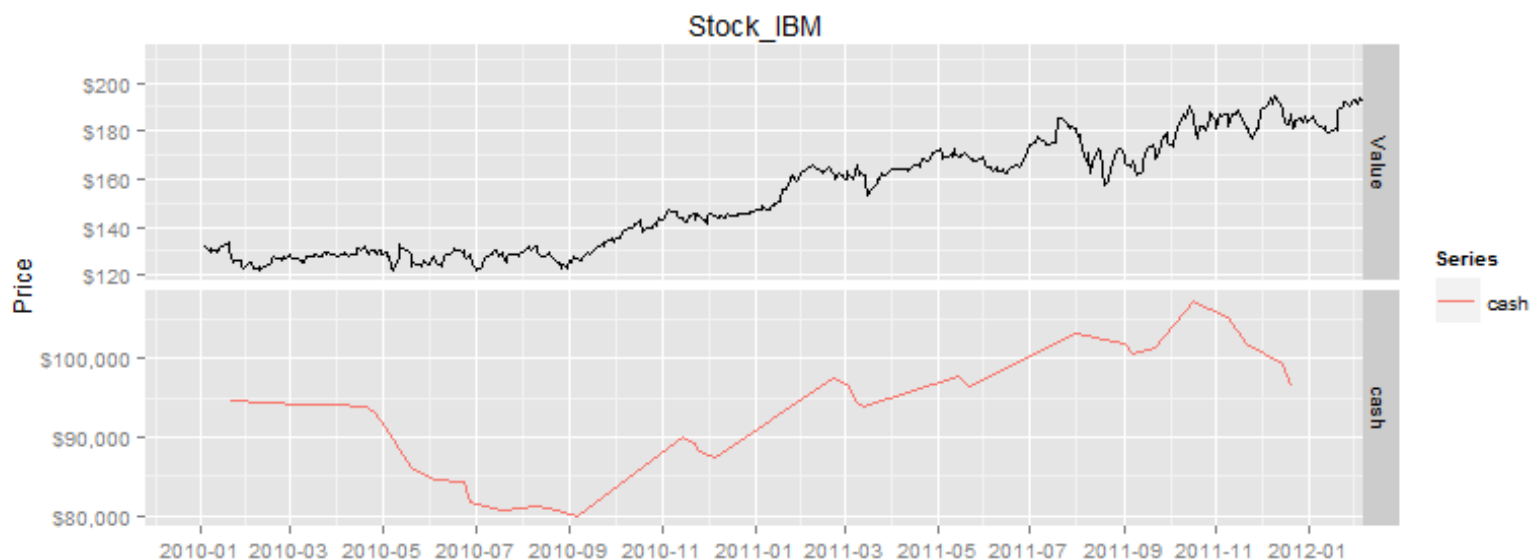
- 为什么最后会亏损呢？中间的大波段应该赚到了足够多的钱。通过资金曲线我们可以找到亏损的原因。
- 画出资金曲线

```
# 股价+现金流量
> drawCash<-function(ldata, adata){
+   g<-ggplot(aes(x=Index, y=Value), data=fortify(ldata[,1], melt=TRUE))
+   g<-g+geom_line()
+   g<-g+geom_line(aes(x=as.Date(Index), y=Value, colour=Series), data=fortify(adata, melt=TRUE))
+   g<-g+facet_grid(Series ~ ., scales = "free_y")
+   g<-g+scale_y_continuous(labels = dollar)
+   g<-g+scale_x_date(labels=date_format("%Y-%m"), breaks=date_breaks("2 months"), limits = c
+   g<-g+xlabs("") + ylab("Price")+ggtitle(title)
+   g
+ }

# 现金流量
> adata<-as.xts(result1$ticks[which(result1$ticks$op=='S'),][['cash']])
> drawCash(ldata, adata)
```

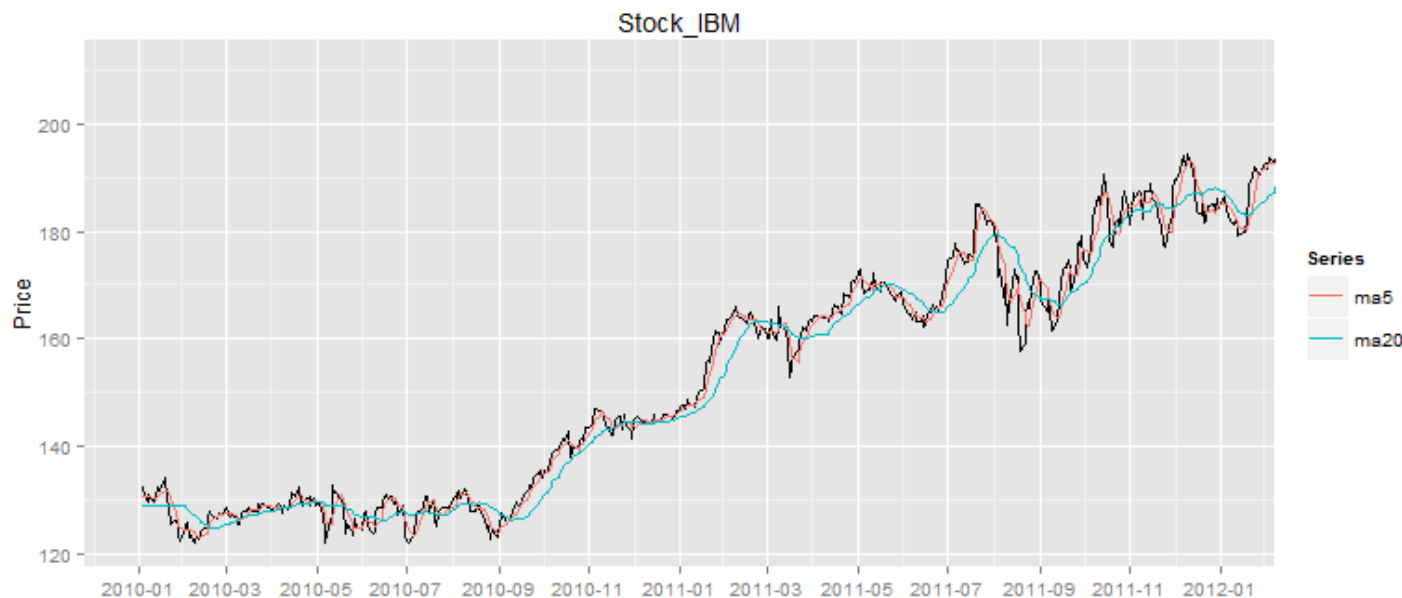
用R语言实现均线模型：资金曲线

- 我们把股价和现金流量并排放置，从2010-09开始均线策略开始大幅赚钱，到2011-10到达最高点，并且超过了本金，然后开始下滑，直截至2012-01亏损3859.86元。这是由于我们把赚到利润继续投资，增大了头寸，以至于2011年底的震荡市让模型失效，从而赔了更多的钱。



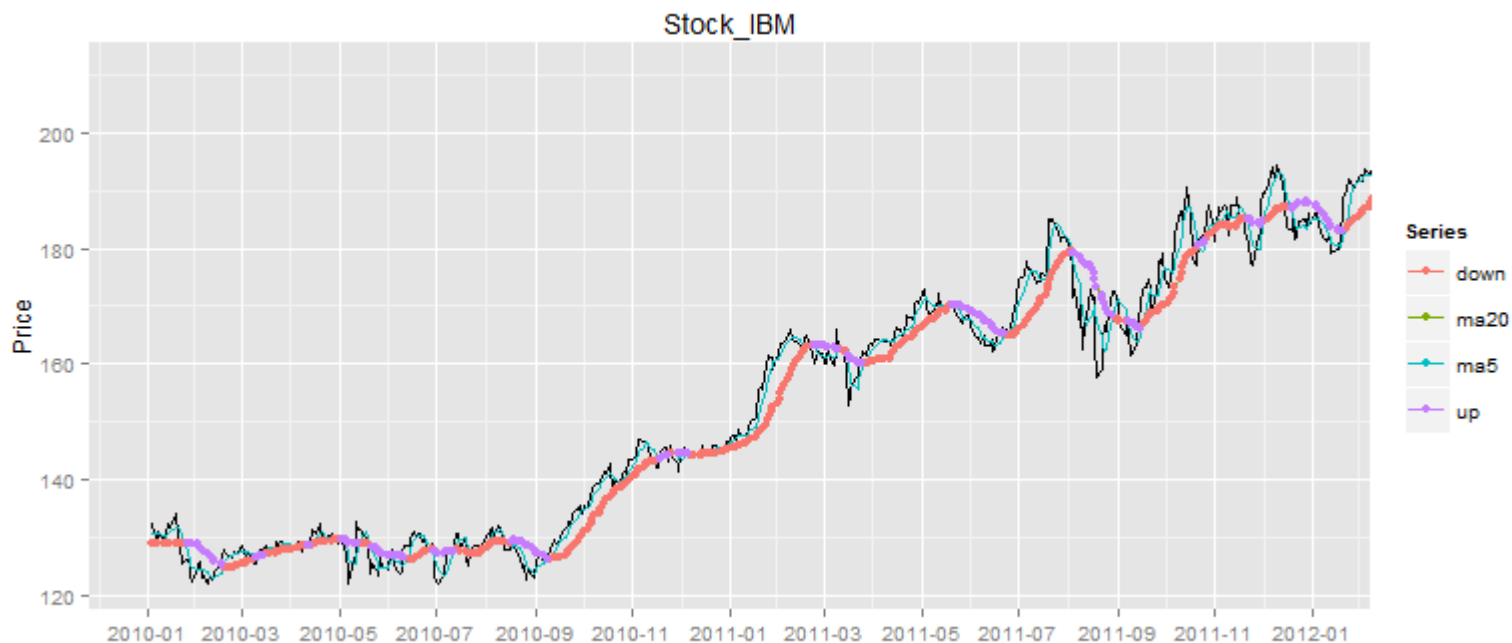
用R语言实现均线模型：二条均线的交易策略

- 一条均线模型，在大的趋势下是可以稳定赚钱的，但由于一条均线对于波动非常敏感性，如果小波动过于频繁，不仅会增加交易次数，而且会让模型失效。然后，就有二条均线的策略模型，可以减低对波动的敏感性。
- 二条均线策略模型，与一条均线模型思路类似，以5日均线价格替换股价，是通过5日均线和20日均线交叉来进行信号交易的。



用R语言实现均线模型：二条均线的交易策略

- 用5日均线 and 20日均线价格做比较，把5日大于均线的部分用紫色表示，股价小于均线的部分用红色表示。我们看到图中，紫色点和红色点在20日均线上交替出现，同样地，我们可以在每次红色出现的第一个点买入股票，然后在紫色的第一个点卖出股票，直观看上去的与一条均线模型类似，都是赚钱的。



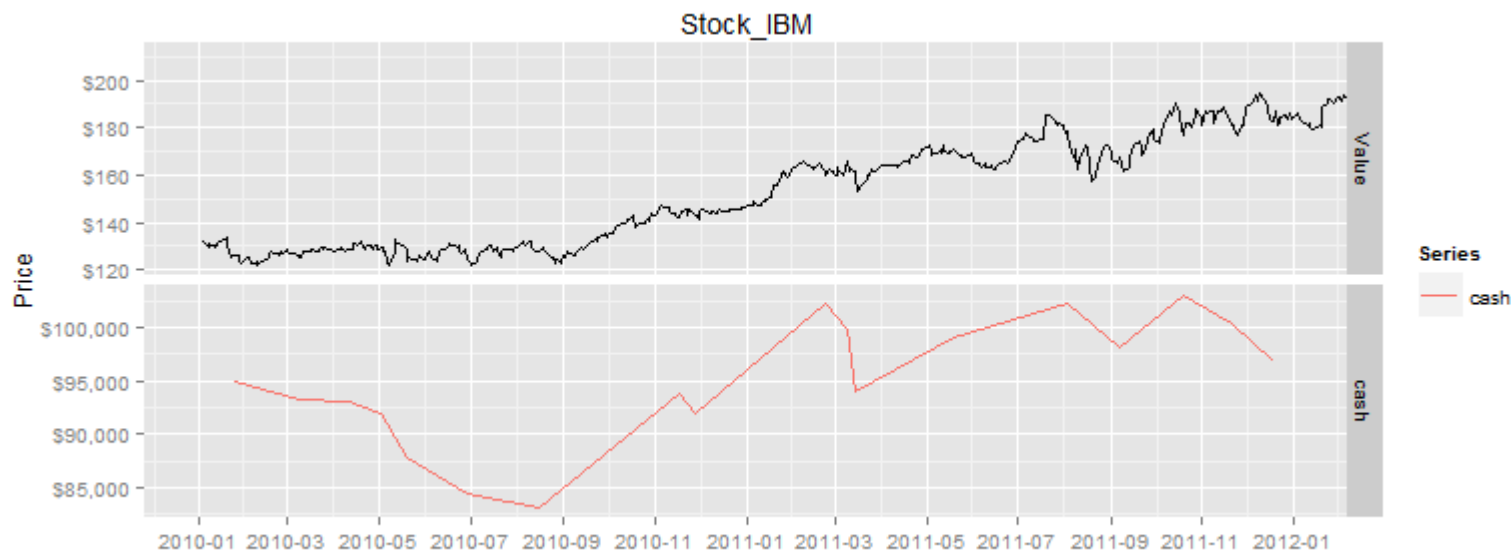
用R语言实现均线模型：二条均线的交易策略

- 一共有36条交易记录，买卖各占一半，比一条均线模型少了18笔交易。
- 利用交易信号数据，进行模拟交易。我们设定交易参数，以\$10W为本金，满仓买入或卖出，手续费为0，传入交易信号。
- 通过模拟交易，我们精确地算出每笔交易的盈利情况了,有26笔交易是亏损的，10笔交易是有盈利的。
- 查看最后的资金情况，资金剩余96828.9元，亏了3171.1元。

```
> tail(result2$ticks,1)
      Value op  cash amount  asset  diff
2011-12-19 182.89 S 96828.9    0 96828.9 -3581.33
```

用R语言实现均线模型：二条均线的交易策略

- 我们可以发现，虽然最后资金也是赔了3171.1，比一条均线策略模型赔的小一点，但二条均线策略模型有3次高于本金的情况，而且最差的情况也比一条均线最差的情况要好。
- 查看资金曲线。



用R语言实现均线模型：对比两个模型的盈利时机

```
# 盈利的交易
> rise<-merge(as.xts(result1$rise[1]), as.xts(result2$rise[1]))
> names(rise)<-c("plan1", "plan2")
```

■ 我们再进一步对比两个模型的盈利情况，找出两个模型中所有赚钱的交易。

```
# 查看数据情况
> rise
      plan1 plan2
2010-03-11 127.60  NA
2010-04-08 127.61  NA
2010-07-22 127.47  NA
2010-08-12 128.30  NA
2010-09-09 126.36  NA
2010-09-10  NA 127.99
2010-11-16 142.24  NA
2010-11-18  NA 144.36
2010-12-07  NA 144.02
2010-12-08 144.98  NA
2011-02-22 161.95  NA
2011-02-23  NA 160.18
2011-03-25 162.18  NA
2011-03-28  NA 161.37
2011-05-16 168.86  NA
2011-05-20  NA 170.16
2011-06-21 166.22  NA
2011-06-23  NA 166.12
2011-08-02 178.05  NA
2011-08-04  NA 171.48
2011-09-14 167.24  NA
```

■ plan1是一条均线模型，plan2是二条均线模型。plan1比plan2多了6次交易，显然多的这几次交易是由于对波动敏感性引起的，反而减少了趋势行情收益。

用R语言实现均线模型：交易区间

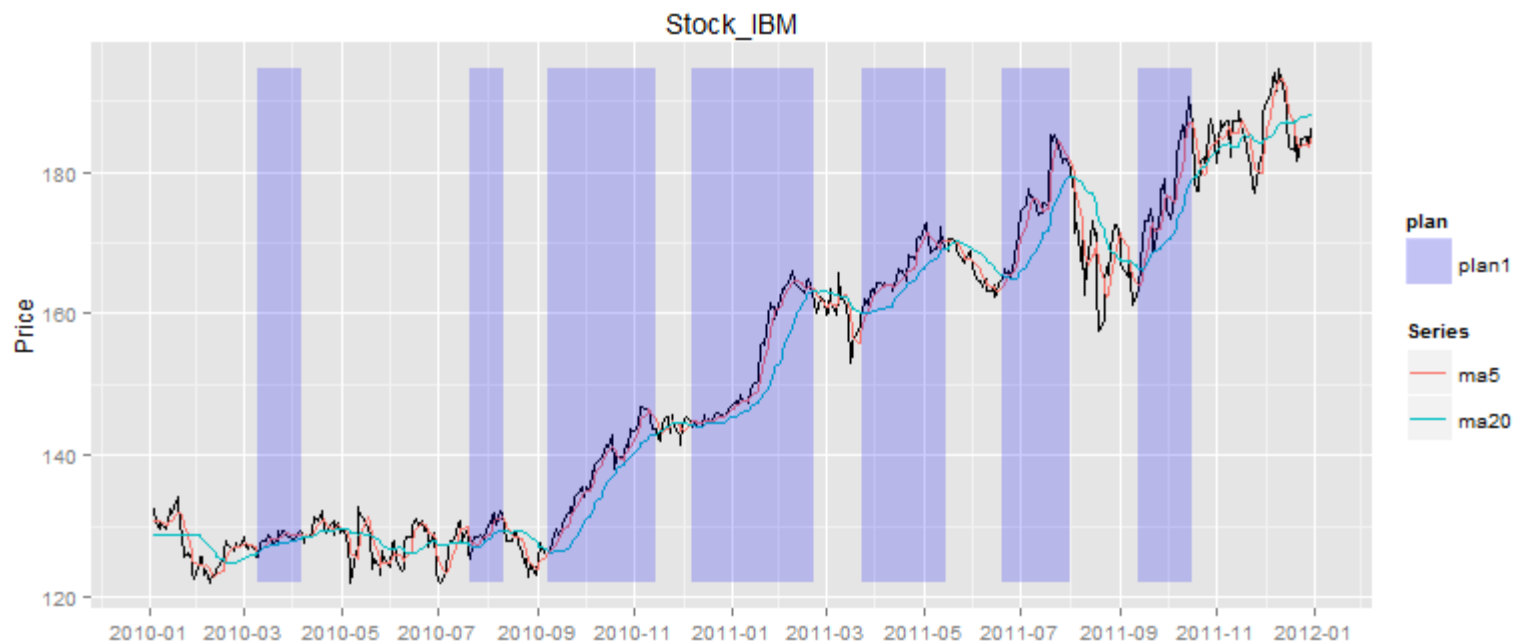
- 最后，我们画一下盈利部分的交易区间

```
> # 均线图+交易区间
> drawRange<-function(ldata, plan, title="Stock_2014", sDate=min(index(ldata)), eDate=max(index
+   g<-ggplot(aes(x=Index, y=Value), data=fortify(ldata[, 1], melt=TRUE))
+   g<-g+geom_line()
+   g<-g+geom_line(aes(colour=Series), data=fortify(ldata[, -1], melt=TRUE))
+   g<-g+geom_rect(aes(NULL, NULL, xmin=start, xmax=end, fill=plan), ymin = yrng[1], ymax = yrng[2])
+   g<-g+scale_fill_manual(values =alpha(c("blue", "red"), 0.2))
+   g<-g+scale_x_date(labels=date_format("%Y-%m"), breaks=date_breaks("2 months"), limits = c
+   g<-g+xlab("") + ylab("Price")+ggtitle(title)
+
+   if(out) ggsave(g, file=paste(title, ".png", sep=""))
+   else g
+ }
```

```
#盈利区间
> plan<-comPlan(ldata, result1, result2){} # 代码省略
> drawRange(ldata, plan, title, sDate, eDate) #画图
```

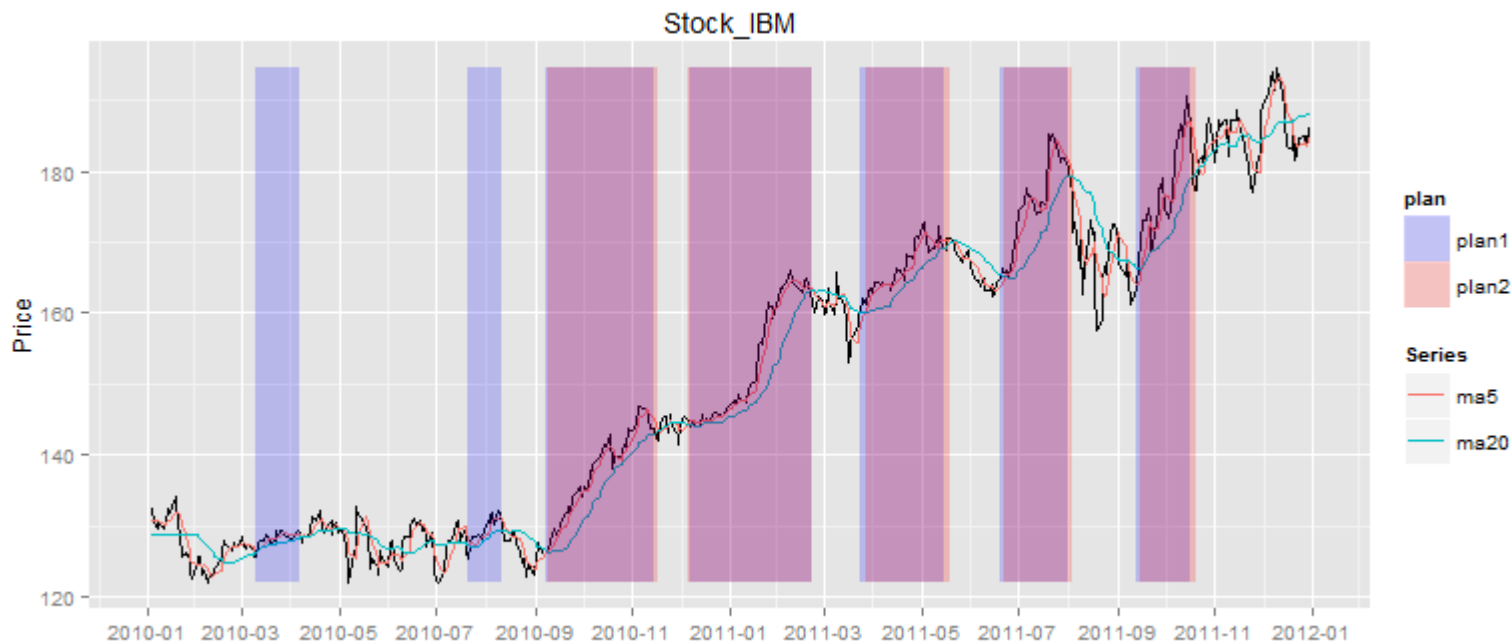
用R语言实现均线模型：交易区间

- plan1的盈利区间。



用R语言实现均线模型：交易区间

- plan1和plan2叠加的盈利区间。



- 从盈利区间我们可以看到，一条均线对波动敏感性的问题，二条均线模型是对一条均线模型的优化，这样我们就一个完整均线模型的实例研发。

用R语言实现均线模型：优化问题

- 如果从交易的角度讲，上面的模型还不能算完成。
- 还有很多的赔钱交易，要进行更多地优化，减少最大回撤，在更确定的时机做多，反向做空等。
- 遗留问题....

- 均线模型虽然如此的简单，但实盘交易时真能在趋势行情中跑赢双均线(优化后)模型，也真不是一件容易的事情。

- **二条均线打天下，不说东方不败，也是独孤求败。**

关于作者

- 张丹，量化投资创业者
- Weibo: @Conan_Z
- Blog : <http://blog.fens.me>
- Email: bsspirit@gmail.com

- 《R的极客理想》即将出版.....





Thanks

FAQ时间