

# 风险计量分析与工具

## R 语言期末作业

### ——简单的交互式网页

小组成员：

李 情 2019100424

李 想 2019100425

许沐明 2019102487

# 目录

- 1. 综述 ..... 3
  - 1.1. 作业思路 ..... 3
  - 1.2. 数据来源 ..... 3
- 2. 功能实现 ..... 4
  - 2.1.复合 K 线图 ..... 4
  - 2.2. 树状图 ..... 7
  - 2.3. 横向条形图 ..... 9
  - 2.4. 价格变化密度图 ..... 14
  - 2.5. 个股日行情表 ..... 15
  - 2.6. 个股财务信息表 ..... 16
- 3. 网页实现 ..... 17
- 4. 总结 ..... 23

# 1. 综述

## 1.1. 作业思路

我们的目的是为了实现一个简单的交互式网页，我们将 Finviz 网站上的具体功能写成一个函数，再利用 shiny 包写成一个多页面式的交互式网页，涉及到具体功能时则调用这些创建的函数，尽量将代码更具有可读性。

压缩包文件解释：

表 1 R 文档解释

R 文档	作用
runshiny.R	简单交互式网站的实现，运行可调用所有的函数
final_barchart.R	函数的创建，用来实现上证指数的成分股的四种横向条形图
density.R	函数的创建，用来实现商业指数的五分钟的价格变化密度图
treemap.R	函数的创建，用来实现上证行业涨跌幅的树状图
StockCandlestick.R	函数的创建，用来实现包含成交量柱状图与成交量均线及价格移动均线的复合 K 线图

表 2 数据文档解释

csv/xlsx 文档	作用
bar_figure1.csv	用来制作 final_barchart.R 中的横向条形图一
bar_figure2.csv	用来制作 final_barchart.R 中的横向条形图二
bar_figure3.csv	用来制作 final_barchart.R 中的横向条形图三
bar_figure4.csv	用来制作 final_barchart.R 中的横向条形图四
density.csv	用来制作 density.R 中的价格变化密度图
C.xlsx	用来制作 treemap.R 中的树状图
DRESSTK.csv	用来制作 StockCandlestick.R 中的复合 K 线图
table1.xlsx	用来制作 runshiny.R 中的个股日行情 DT 表
table1.xlsx	用来制作 runshiny.R 中的个股财务信息表

## 1.2. 数据来源

由于实验室的数据库不稳定，我们的数据都来自于 Resset 金融研究数据库，下载成 csv 或 xts 文件。R 语言的 shiny 运行速度较慢，因此复合 K 线图、个股行情 DT 表、个股财务信息的数据都只选取了一部分上证个股。

表 3 数据选择解释

数据文档	股票池	数据特征	时间范围
bar_figure1.csv	上证成分股	当日收盘价与前一天收盘价对比	2019 年 9 月 30 日
bar_figure2.csv	上证成分股	当日当日开盘价与今年以来的成交均价对比	2019 年 9 月 30 日
bar_figure3.csv	上证成分股	当日涨跌幅与 60 日移动平均对比	2019 年 9 月 30 日
bar_figure4.csv	上证成分股	当日涨跌幅与 120 日移动平均对比	2019 年 9 月 30 日
density.csv	商业指数	每五分钟的价格变化	2017 年 9 月 30 日,
C.xlsx	上证成分股	行业分类、市值及当日涨跌幅	2019 年 9 月 30 日
DRESSTK.csv	600000-	每日开盘价、收盘价、最高价、最低价、成交量	2018 年 10 月 1 日-
	600100		2019 年 9 月 30 日
table1.xlsx	600000-600202	每日收盘价、涨跌幅及成交量数据	2019 年 9 月 30 日
table1.xlsx	600000-600202	22 项财务指标	2019 年 9 月 30 日

注：商业指数的数据最新只有 2017 年 9 月 30 日

表 4 财务信息指标的选取

指标类别	指标名称	
市值指标	总市值	自由流通市值
	市盈率 PE	市净率 PB
	市销率 PS	
每股指标	股息率	每股收益 EPS
波动率指标	波动率:100 周	波动率:24 个月
	波动率:60 个月	
成长指标	年化净资产收益率	年化总资产报酬率
	年化总资产净利率	营业总收入
	营业收入	净利润
财务指标	权益乘数	流动比率
	速动比率	现金比率
	现金流量利息保障倍数	现金净流量

## 2. 功能实现

### 2.1.复合 K 线图

文字说明：

复合 K 线图由 K 线图和成交量图两部分构成，本文都用 ggplot()函数创建这两幅图，再用 grid.arrange()函数将两幅图组合起来。

K线图：由每日的开盘价、收盘价、最高价、最低价构成的蜡烛图，在包含三条移动平均曲线，其中红线为 20 日移动平均曲线，黄色为 50 日移动平均曲线，蓝色为 200 日移动平均曲线。

成交量图：成交量绿柱表示收盘价低于开盘价，红柱表示收盘价高于开盘价。即日 K 线收绿，则成交量也收绿，反之也成立。蓝线表示成交量五日均线，红线表示成交量十日均线。

### 代码：

---

```
StockCandlestick.R
```

---

```
stockplot <- function(x){
```

```
  fulldata <- read.csv("DRESSTK.csv", header=TRUE)
```

```
  data<-fulldata[fulldata$Stkcd == x,3:8]
```

```
  data<- na.omit(data)
```

```
  class(data)
```

```
  data$date<-as.Date(data$date)
```

```
  class(data$date)
```

```
  # 为了解决日期作为横坐标会出现休市日的情况，需要以下的变量辅助
```

```
  row_len <- nrow(data)
```

```
  breaks <- seq(1, row_len, 30)
```

```
  labels <- data$date[breaks]
```

---

```
  # 作 K 线图
```

```
  p1 <- data %>%
```

```
    arrange(data$date) %>%
```

```
    mutate(ma20 = SMA(close, n = 20, align = "right"),
```

```
           ma50 = SMA(close, n = 50, align = "right"),
```

```
           ma200 = SMA(close, n = 200, align = "right"),
```

```
           date_axis = row_number()) %>%
```

```
  ggplot(aes(x = date_axis)) +
```

```
  geom_boxplot(aes(lower = pmin(close, open),
```

```
                  middle = close,
```

```
                  upper = pmax(close, open),
```

```
                  ymin = low,
```

```
                  ymax = high,
```

```
                  group = date_axis,
```

---

---

```
      fill = open > close),
      stat = "identity",
      show.legend = FALSE) +
  geom_line(aes(y = ma20), color = "red") +
  geom_line(aes(y = ma50), color = "yellow") +
  geom_line(aes(y = ma200), color = "blue3") +
  scale_x_continuous(breaks = breaks,
                     labels = NULL,
                     expand = c(0, 0)) +
  theme(axis.ticks.x = element_blank(),
        axis.title = element_blank(),
        axis.text.y = element_text(margin = margin(l = 8)))
```

---

#### # 作成交量图

```
p2 <- data %>%
  arrange(date) %>%
  mutate(vol_ma5 = SMA(volume, n = 5, align = "right"),
         vol_ma10 = SMA(volume, n = 10, align = "right"),
         date_axis = row_number()) %>%
  ggplot(aes(x = date_axis, y = volume)) +
  geom_bar(stat = "identity",
          aes(fill = open > close),
          show.legend = FALSE) +
  geom_line(aes(y = vol_ma5), color = "blue3") +
  geom_line(aes(y = vol_ma10), color = "red") +
  scale_x_continuous(breaks = breaks,
                     labels = format(labels, "%Y-%m"),
                     expand = c(0, 0)) +
  scale_y_continuous(expand = c(0, 0.5)) +
  theme(axis.title = element_blank())
```

---

#### # 组合

```
grid.arrange(p1, p2, nrow = 2, heights = 2:1)
}
```

---

**结果：**

**图 1 复合 K 线图**



## 2.2.树状图

**文字说明：**

本文利用 `treemap()` 函数来创建树状图，分为全行业树状图及分行业树状图，构建树状图的代码一致，只是在构建行业树状图是进行了数据筛选工作。

树状图中，股票面积的大小对应股票市值的大小，股票颜色的变化对应股票当日涨跌幅的变化。

**代码：**

---

```
treemap.R
```

---

```
#全行业树状图
```

```
TreeMap1<-function(){
  data <- read.xlsx("C.xlsx",1,encoding = "UTF-8")
  data$stcok<- as.integer(data$stcok)
  data$color <- rainbow(nlevels(data$stock))[data$stock] #更改颜色
  color <- c( "#9E3649", "#FF0017", "#008B4F", "#00AE51")

  palette.HCL.options <- list(hue_start=270, hue_end=360+150)
  treemap(data,
    index=c("class", "stock1"),
    vSize="market_cap",
    vColor="percent",
```

---

---

```

        type="value",
        #palette.HCL.options=palette.HCL.options, #设置颜色
        palette=(values=color),
        #palette="-RdGy",
        format.legend = list(scientific = FALSE, big.mark = " "),
        fontsize.title=0, #去掉标题
        fontsize.legend=0, #去掉图例
        fontcolor.labels="white",fontface.labels=2,fontsize.labels = c(12,20), #标签
颜色, 加粗

        #fontfamily.labels="STSong", #解决中文乱码
        bg.labels=0, #聚类标签的背景颜色
        align.labels=list(c("left", "top"), c("center", "center")), #改变标签位置, 前面
为聚类标签, 后面为每个 stock 标签

        overlap.labels=0, #0-1, 确定标签之间重叠程度, 0 表示如果较高级别的标签重
叠, 则不打印较低级别的标签

    )
}

```

---

```

#各行业树状图
TreeMap2<-function(x){
data <- read.xlsx("C.xlsx",1,encoding = "UTF-8")
data$stcok<- as.integer(data$stcok)
data<-data[data$class==x,]
data$color <- rainbow(nlevels(data$stock))[data$stock] #更改颜色
color <- c( "#9E3649", "#FF0017", "#008B4F", "#00AE51")

palette.HCL.options <- list(hue_start=270, hue_end=360+150)

treemap(data,
        index=c("class", "stock1"),
        vSize="market_cap",
        vColor="percent",
        type="value",
        #palette.HCL.options=palette.HCL.options, #设置颜色
        palette=(values=color),
        #palette="-RdGy",
        format.legend = list(scientific = FALSE, big.mark = " "),
        fontsize.title=0, #去掉标题
        fontsize.legend=0, #去掉图例
        fontcolor.labels="white",fontface.labels=2,fontsize.labels = c(12,20), #标签颜色, 加粗
        #fontfamily.labels="STSong", #解决中文乱码
        bg.labels=0, #聚类标签的背景颜色
        align.labels=list(c("left", "top"), c("center", "center")), #改变标签位置, 前面为聚类标签,
后面为每个 stock 标签

        overlap.labels=0, #0-1, 确定标签之间重叠程度, 0 表示如果较高级别的标签重叠, 则不打

```

---



印较低级别的标签

)

}

结果图:

图 2 全行业树状图

全行业树状图

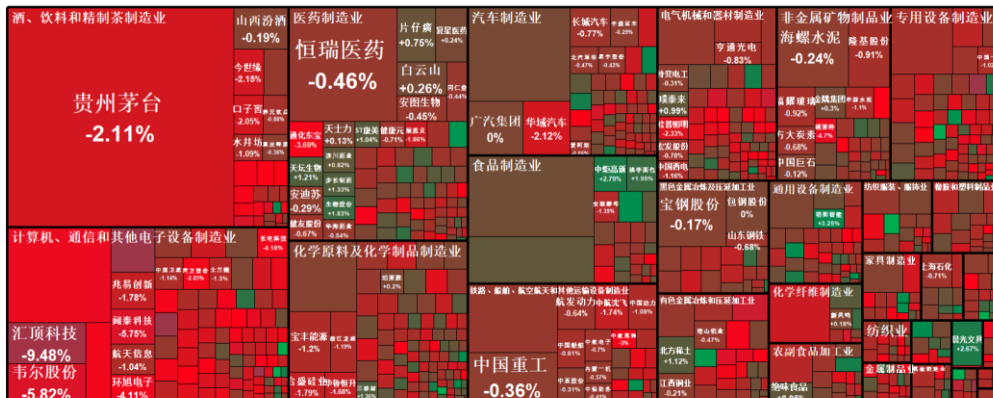
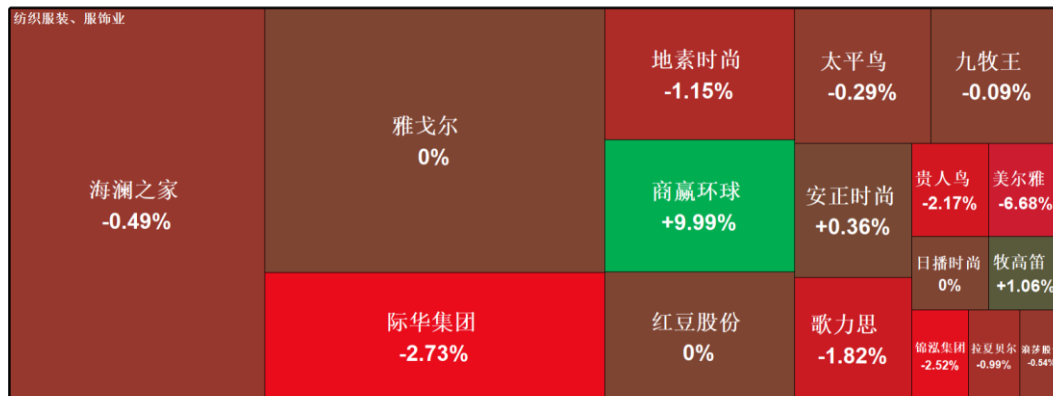


图 3 各行业树状图

各行业树状图



## 2.3. 横向条形图

文字说明:

本文四张条形图都是用 ggplot()函数完成的

创建的 Barchart1()函数: 是衡量上证指数成分股中当天的收盘价和前一天的收盘价相比,

绿色的表示高于前一天收盘价的股票数，红色的表示低于前一天收盘价的股票数，灰色表示不变。

创建的 Barchart2()函数：衡量上证指数成分股中当日开盘价和 2019 年以来的成交均价相比，绿色的表示高于 2019 年以来的成交均价，红色的表示低于 2019 年以来的成交均价。

创建的 Barchart3()函数：衡量上证指数成分股中当日涨跌幅和 60 日移动平均变化相比，绿色的表示高于 60 日移动平均变化，红色的表示低于 60 日移动平均变化。

创建的 Barchart4()函数：衡量上证指数成分股中当日涨跌幅和 120 日移动平均变化相比，绿色的表示高于 120 日移动平均变化，红色的表示低于 120 日移动平均变化。

#### 代码：

---

```
final_barchart.R
Barchart1 <- function(){
##### figure1

#上涨、不变、下降的股票比例

data1 <- read.csv('bar_figure1.csv',header=T)

data1$trend = factor(data1$trend, levels=c('declining','stay','advancing'))

ggplot(data1,aes(' ',stock,fill=trend))+

  geom_bar(stat="identity",position="stack",width = 0.05)+ #stack:堆积柱状图, width:柱状图
  宽度

  theme_bw()+

  #scale_fill_brewer(palette='Set1')+ #柱状图颜色, 使用 RColorBrewer

  scale_fill_manual(values=c("#FF6666","#CCCCCC","#66CC33"))+

  guides(fill=guide_legend(title=NULL))+ #不显示标题

  guides(fill=F) + #不显示图例

  #不显示各种图标

  theme(panel.grid = element_blank(),panel.border = element_blank(),
```

---

---

```

axis.text = element_blank(),axis.ticks = element_blank(),
text = element_blank())+

#显示数字

geom_text(aes(y=position,label=paste(stock,'% ',sep = ")),size=3,col='white')+
geom_text(aes(y=position_num,label=number),size=3,
          col=c('#66CC33','#FF6666'),
          vjust=-1.5)+
geom_text(aes(y=position_trend,label=trend1),size=3,col='black',vjust=-1.5)+

coord_flip() #将图形横纵坐标互换
}

```

---

```

Barchart2<-function(){
#### figure2

#总发行创下新高/新低的股票比例

data2 <- read.csv('bar_figure2.csv',header=T)
data2$trend = factor(data2$trend, levels=c('low','high'))

ggplot(data2,aes(' ',stock,fill=trend))+

  geom_bar(stat="identity",position="stack",width = 0.05)+ #stack:堆积柱状图, width:柱状图
  宽度

  theme_bw()+
  scale_fill_manual(values=c("#FF6666","#66CC33"))+

  guides(fill=guide_legend(title=NULL))+ #不显示标题

  guides(fill=F)  + #不显示图例

#不显示各种图标

theme(panel.grid = element_blank(),panel.border = element_blank(),
      axis.text = element_blank(),axis.ticks = element_blank(),
      text = element_blank())+

#显示数字

geom_text(aes(y=position,label=paste(stock,'% ',sep = ")),size=3,col='white')+
geom_text(aes(y=position_num,label=number),size=3,
          col=c('#66CC33','#FF6666'),
          vjust=-1.5)+
geom_text(aes(y=position_trend,label=trend1),size=3,col='black',vjust=-1.5)+

coord_flip() #将图形横纵坐标互换

```

---

---

```

}
Barchart3<-function(){
#### figure3

#日振幅高于/低于 60 天移动平均的股票数比例

data3 <- read.csv('bar_figure3.csv',header=T)
data3$trend = factor(data3$trend, levels=c('below','above'))

ggplot(data3,aes(' ',stock,fill=trend))+

  geom_bar(stat="identity",position="stack",width = 0.05)+ #stack:堆积柱状图, width:柱状图

  宽度

  theme_bw()+
  scale_fill_manual(values=c("#FF6666","#66CC33"))+

  guides(fill=guide_legend(title=NULL))+ #不显示标题

  guides(fill=F)  + #不显示图例

  #不显示各种图标

  theme(panel.grid = element_blank(),panel.border = element_blank(),
        axis.text = element_blank(),axis.ticks = element_blank(),
        text = element_blank())+

  #显示数字

  geom_text(aes(y=position,label=paste(stock,'% ',sep = " ")),size=3,col='white')+
  geom_text(aes(y=position_num,label=number),size=3,
            col=c('#66CC33','#FF6666'),
            vjust=-1.5)+
  geom_text(aes(y=position_trend,label=trend1),size=3,col='black',vjust=-1.5)+

  coord_flip() #将图形横纵坐标互换

}

```

---

```

Barchart4<-function(){
#### figure4

#日振幅高于/低于 120 天移动平均的股票数比例

data4 <- read.csv('bar_figure4.csv',header=T)
data3$trend = factor(data4$trend, levels=c('below','above'))

ggplot(data4,aes(' ',stock,fill=trend))+

  geom_bar(stat="identity",position="stack",width = 0.05)+ #stack:堆积柱状图, width:柱状图

```

---

---

## 宽度

```
theme_bw()+
scale_fill_manual(values=c("#FF6666","#66CC33"))+

guides(fill=guide_legend(title=NULL))+ #不显示标题

guides(fill=F)  + #不显示图例

#不显示各种图标

theme(panel.grid = element_blank(),panel.border = element_blank(),
      axis.text = element_blank(),axis.ticks = element_blank(),
      text = element_blank())+

#显示数字

geom_text(aes(y=position,label=paste(stock,'% ',sep = " ")),size=3,col='white')+
geom_text(aes(y=position_num,label=number),size=3,
          col=c('#66CC33','#FF6666'),
          vjust=-1.5)+
geom_text(aes(y=position_trend,label=trend1),size=3,col='black',vjust=-1.5)+

coord_flip() #将图形横纵坐标互换

}
```

---

## 结果图：

图 4 横向条形图一



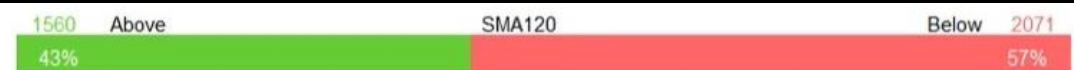
图 5 横向条形图二



图 6 横向条形图三



图 7 横向条形图四



## 2.4.价格变化密度图

### 文字说明:

本文的价格变化密度图是用 ggplot()函数完成的, 是对于商业指数 (SH.000005) 于 2017 年 9 月 30 日 (由于 Resset 数据库最新一天的数据只有这天) 每五分钟的价格变化图, 绿色代表增长, 红色代表下跌。

### 代码:

---

```
density.R
densityplot <- function(){

density <- read.csv('density.csv', header=T)

# create some fake data with zero-crossings
a= data.frame(x=seq(1,length(density$price1)),y=density$price1)
b <- do.call("rbind",
             sapply(1:(nrow(a)-1), function(i){
               f <- lm(x~y, a[i:(i+1),])
               if (f$qr$rank < 2) return(NULL)
               r <- predict(f, newdata=data.frame(y=0))
               if(a[i,]$x < r & r < a[i+1,]$x)
                 return(data.frame(x=r,y=0))
               else return(NULL)
             }))
a2 <- rbind(a,b)

mytheme<-theme(panel.grid.major=element_line(),
               axis.ticks = element_blank(),
               axis.text = element_blank(),
               axis.title=element_blank())

ggplot(a2,aes(x,y)) +
  geom_area(data=subset(a2, y<=0), fill="#FF6666") +
  geom_area(data=subset(a2, y>=0), fill="#66CC33") +
  scale_y_continuous(breaks=NULL)+
  labs(fill="")+
  mytheme+
  ggtitle("Where will Shanghai composite index be in the next 5 minutes?")+
```

---

---

```
theme(plot.title = element_text(lineheight=.8, face="bold",hjust = 0.5))
}
```

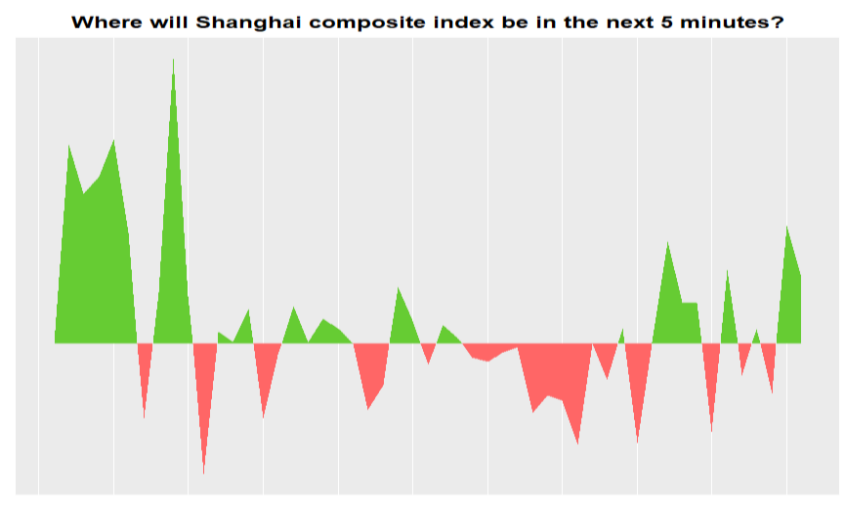
---

**结果图：**

---

**图 8 商业指数价格变化密度图**

---



2.5.个股日行情表

**文字说明：**

本文的个股日行情表，是用 shiny 包中的 DT 函数完成的，这一部分的代码直接写在了 runshiny.R 中的 server 部分。该表格可以对所有变量进行升序或降序排列，也可以根据每个变量进行筛选查找。

**代码：**

---

```
table1 <- read.xlsx("C:\\Users\\Irene\\Desktop\\last homework\\table1.xlsx",1,encoding = "UTF-8")
table1<-table1[1:170,]
output$overview <- renderDataTable(table1)
```

---

结果图：

图 9 上证个股日行情表

上证个股行情表				
Show	25	▼	entries	Search: <input type="text"/>
证券代码	证券简称	收盘价	涨跌幅	成交量
600000	浦发银行	11.99	0.2508	29531099
600004	白云机场	16.71	-2.3378	19955213
600006	东风汽车	4.34	-0.4587	5420184
600007	中国国贸	16.72	-2.7341	1298202
600008	首创股份	3.24	0.0000	8561200
600009	上海机场	74.77	-1.6184	7784250
600010	包钢股份	1.30	0.7752	243821532
600011	华能国际	5.87	0.3419	22015797
600012	皖通高速	5.55	0.3617	1291064
600022	山东钢铁	1.36	0.7407	76021193
600023	浙能电力	3.80	1.0638	15438203
600025	华能水电	4.19	-0.2381	18481412
600026	中远海能	5.85	0.6885	17321622
600027	华电国际	3.64	1.9608	29833762
600028	中国石化	4.94	0.0000	67022030
600029	南方航空	6.55	-0.4559	30629495
600030	中信证券	21.49	-0.6932	65110057
600031	三一重工	14.47	-0.8904	49216262
<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Showing 1 to 25 of 170 entries			Previous 1 2 3 4 5 6 7 Next	

## 2.6.个股财务信息表

文字说明：

本文的个股财务信息表，是用 shiny 包中的 table 函数完成的，这一部分的代码直接写在了 runshiny.R 中的 server 部分。根据交互式网页输入的股票代码，返回该股票的财务信息表。

代码：

```
table2 <- read.xlsx("C:\\Users\\Irene\\Desktop\\last homework\\table2.xlsx",1,encoding = "UTF-8")
table2<-table2[1:170,]
stockfactor <- eventReactive(input$go, {
```



```
#等待按钮被点击，得到股票代码

stocktable <- table2[table2[,1]==input$stockcode,]
stock <- cbind(colnames(stocktable),t(stocktable))
showstock <- cbind(stock[1:8,],stock[9:16,],stock[17:24,])
})
output$stockview <- renderTable(
  stockfactor(),colnames = FALSE
)
```

**结果图：**

**图 10 上证个股财务信息表**

证券代码	600000	权益乘数.	13.1477	每股收益EPS	2.0802
证券简称	浦发银行	流动比率.	NA	现金净流量	-6.0281e+10
总市值	351931443960	速动比率.	NA	年化净资产收益率.	12.4674
自由流通市值.	183983251395	现金比率.	NA	年化总资产报酬率.	NA
市盈率PE	5.764	现金流量利息保障倍数.	NA	年化总资产净利率.	0.9096
市净率PB	0.7329	营业总收入	1.90819e+11	波动率.100周	20.6613
市销率PS	1.8443	营业收入	1.90819e+11	波动率.24个月	21.1996
股息率	2.9191	净利润	6.1715e+10	波动率.60个月	25.6905

### 3. 网页实现

**文字说明：**

本文的交互式页面，分别利用 ui 和 server 定义 shiny 的页面形式和内置交互式函数。

ui 中，利用 navbarPage()函数创建导航栏，分三个页面进行战术，其中 Home 页面输出 K 线图、四个横向条形图、价格变化密度图、个股日行情表及全行业树状图；Maps 页面，设置单选栏，用户可选择行业名称，输出该行业的树状图；Elite 页面，设置文本框，用户可输入想要查询的股票代码，输出该股票的复合 K 线图及该股的财务信息表。

**代码：**

```
runshiny.R
```

```
#设置环境
```

```
setwd('/Users/Irene/Desktop/last homework')
```

---

```
library(shiny)
source("density.R")
source("final_barchart.R")
source("treemap.R")
source("StockCandlestick.R")
library(shiny)
library("xlsx")
library(ggplot2)
library(dplyr)
library(purrr)
#install.packages("treemap")
library(treemap)
#install.packages("treemapify")
library(treemapify)
library(RColorBrewer)
library(ggplot2)
#library(tidyverse)
library(gridExtra)
library(quantmod)
library(zoo)
library(dplyr)
```

---

#定义 ui 展示

```
ui <- navbarPage('FINVIZ',
  tabPanel('Home',
    #定义第一个页面： Home 页面
    fluidPage(
      h1("上证指数 K 线图",align="center"),
      fluidRow(
        column(12,plotOutput("szindex"))
      ),
      h1("横向条形图",align="center"),
      #条形图
      h3("图一及图二"),
      fluidRow(
        column(6,plotOutput("barchart1")),
        column(6,plotOutput("barchart2"))),
      h3("图三及图四"),
      fluidRow(
        column(6,plotOutput("barchart3")),
        column(6,plotOutput("barchart4"))),
      h3("图五"),
      fluidRow(
        column(6,plotOutput("BBplot"))
      ),
      #输出表格与树状图
```

---

---

```

h1("上证个股行情表",align="center"),
fluidRow(
  column(12,dataTableOutput("overview"))

),
h1("全行业树状图",align="center"),
fluidRow(
  column(12,plotOutput("treeMap"))
)
)),
tabPanel('Maps',
  #定义第二个页面，树状图页面
  fluidPage(
    #树状图
    p("选择想要查看的行业的树状图"),
    fluidRow(
      column(6,selectInput("industryclass", "行业分类",
        c("农副食品加工业","食品制
造业","酒、饮料和精制茶制造业",
        "纺织业","纺织服装、服饰
业","家具制造业","造纸及纸制品业",
        "医药制造业")))),
      h1("各行业树状图",align="center"),
      fluidRow(
        column(12,plotOutput("industrytreeMap"))
      )
    )),
tabPanel('Elite',
  #定义第三个页面，Elite 页面
  tab4 <- fluidPage(
    #查找个股信息
    p("选择想要查看的个股的 K 线图及财务信息数据"),
    fluidRow(
      column(3,textInput("stockcode", "证券代码", "600000")),
      column(3,actionButton("go", "确定"))
    ),
    h1("个股 K 线图",align="center"),
    #日线图
    fluidRow(
      column(12,plotOutput("stockcandlestick"))
    ),
    h1("个股财务信息表格",align="center"),
    #财务信息表格
    fluidRow(
      column(12,tableOutput("stockview"))
    )
  )

```

---

---

```

    ))
  )
}

#读取数据
table1 <- read.xlsx("C:\\Users\\Irene\\Desktop\\last homework\\table1.xlsx",1,encoding = "UTF-8")
table2 <- read.xlsx("C:\\Users\\Irene\\Desktop\\last homework\\table2.xlsx",1,encoding = "UTF-8")
table1<-table1[1:170,]
table2<-table2[1:170,]
#定义 server 内置交互式
server <- function(input, output) {

  stockfactor <- eventReactive(input$go, {
    #等待按钮被点击，得到股票代码
    stocktable <- table2[table2[,1]==input$stockcode,]
    stock <- cbind(colnames(stocktable),t(stocktable))
    showstock <- cbind(stock[1:8,],stock[9:16,],stock[17:24,])
  })
  output$overview <- renderDataTable(table1)
  output$stockview <- renderTable(
    stockfactor(),colnames = FALSE
  )
  output$BBplot <- renderPlot(densityplot())
  output$barchart1 <- renderPlot(Barchart1())
  output$barchart2 <- renderPlot(Barchart2())
  output$barchart3 <- renderPlot(Barchart3())
  output$barchart4 <- renderPlot(Barchart4())
  output$treeMap <- renderPlot(TreeMap1())
  Xstock <- eventReactive(input$go, {
    #等待按钮被点击，得到股票代码
    input$stockcode
  })
  output$stockcandlestick <- renderPlot(stockplot(Xstock()))
  output$szindex <- renderPlot(stockplot("600000"))
  output$industrytreeMap <-renderPlot(TreeMap2(input$industryclass))
}

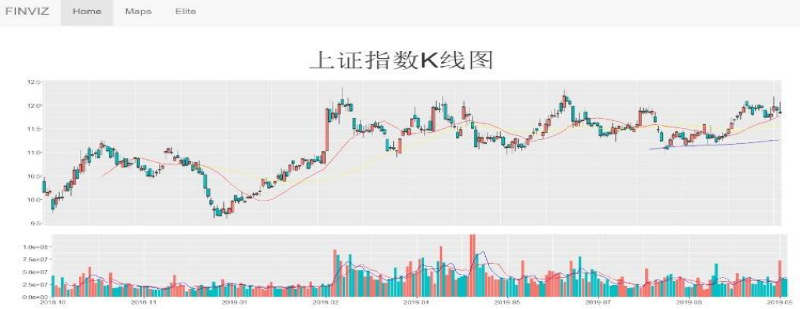
#运行 shiny 页面
shinyApp(ui, server)

```

---

结果图：

图 11 页面一(Home 页面)

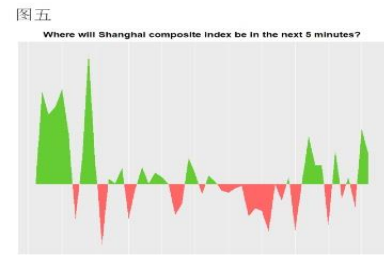
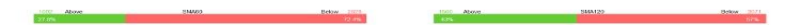


横向条形图

图一及图二



图三及图四



上证个股行情表

Show 25 entries Search:

证券代码	证券简称	收盘价	涨跌幅	成交量
600000	浦发银行	11.99	0.2508	29531099
600004	白云机场	16.71	-2.3378	19955213
600006	东风汽车	4.34	-0.4587	5420184
600007	中国国贸	16.72	-2.7341	1298202
600008	首创股份	3.24	0.0000	8561200
600009	上海机场	74.77	-1.6184	7784250
600010	包钢股份	1.30	0.7752	243821532
600011	华能国际	5.87	0.3419	22015797
600012	皖通高速	5.55	0.3617	1291064
600022	山东钢铁	1.38	0.7407	76021193
600023	浙能电力	3.80	1.0638	15438203
600025	华能水电	4.19	-0.2381	18481412
600026	中远海能	5.85	0.6885	17321622
600027	华电国际	3.64	1.9608	29833762
600028	中国石化	4.94	0.0000	67022030
600029	南方航空	6.55	-0.4559	30829495
600030	中信证券	21.49	-0.6932	65110057
600031	三一重工	14.47	-0.8904	49216262

Showing 1 to 25 of 170 entries

Previous 1 2 3 4 5 6 7 Next

全行业树状图



图 12 页面二(Maps 页面)



图 13 页面三(Elite 页面)



## 4. 总结

从这份期末作业来说，我们可以改进的地方还有许多，首先，利用数据接口的方法而不是读取数据的方法会使得 shiny 交互式页面的运行速度更快，也可以将可以查询的股票池真正意义上扩大到上证甚至是深证等全 A 股，真正实现数据的可视化。其次，Finviz 网站上还有更多的功能值得我们去学习去挖掘，该网站上的自定义数据处理服务是个非常有效的功能，可能仅仅利用 shiny 包在交互式页面的实现上会有局限，我们应再自我研究型学习其他 R 语言中制作网站更全面的包和函数方法。然后，在交互式页面的排版和外观上，我们也还有改进的空间，jsp 语言是非常好的实现展示页面排版的语句，R 语言中也有应用 jsp 语言的接口。最后，在金融方面，本次作业我们更多的是实现数据的可视化，未来应将我们所学习到的金融知识、大数据方法也可以应用到交互式页面中。

经过一个学期的 R 语言学习，我们从各个角度学习了如何使用 R 语言的基础语句、如何创建函数、如何进行数据的可视化、如何创建一个交互式网站、如何利用 R 语言成为金融分析工具等等，收益匪浅，未来真正在实际工作中应用 R 语言提高工作效率提高工作能力是我们需要不断去思考、改进的方向。

最后非常感谢王老师一学期以来的教学!!!