

金融数据特征分析与R软件入门

院 系: 经管学院电子商务系

班 级: 电商2017-1班

姓 名: 卢大佐

学 号: 09173268

教 师: 王许

2020年6月25日

# 0 作业说明

本次作业所有代码托管于GitHub

项目地址为：<https://github.com/HanSuhhi/Rwork>

# 1 运用所学习的三种网络爬虫方法（quantmod程序包、网页爬取技术、Tushare数据库）任意抓取深圳证券交易所上市的股票，并分别进行绘图和基本统计分析

## 1.1 代码演示

辅助方法代码：

# 定义获取股票函数

# @params: stock\_name 股票名称

# @return: stock 获得的股票

getStockByName <- function(stock\_name, stock, type, label) {

    # quantmod 方法

    if (type == 'quantmod') {

        return(getStockByQuantmod(stock\_name))

    }

    # web 方法

    else if (type == 'web') {

        if (label == 'sz') {

            return(getSpecialStockByWeb(stock, label))

        } else {

            return(getStockByWeb(stock))

        }

    }

    # Tushare 方法

    else if (type == 'Tushare') {

        print(stock\_name)

        return(getStockByTushare(stock\_name))

    }

    # 报错

    else {

        throwWrong()

    }

}

# Tushare 获取股票

    # @params: stock\_name 股票名称

    # @return: stock 获得的股票

getStockByTushare <- function(stock\_name) {

    api <- pro\_api(token="afc328b8c1724efce0fdfdc00126e4cd0e72eaef28f64428e11c0dfb")

    # 获取数据

    stock <- api(api\_name = 'daily', ts\_code=stock\_name, start\_date='20170525', end\_date='20200529')

    stock\_new\_1<-stock[order(stock[,2],decreasing=F),]

    head(stock\_new\_1)

}

getSZ <- function() {

}

# Quantmod 获取股票

    # @params: stock\_name 股票名称

    # @return: stock 获得的股票

getStockByQuantmod <- function(stock\_name) {

    print(stock\_name)

    # 获取该股票交易数据

    stock <- getSymbols(stock\_name, src = "yahoo", from = "2017-01-01", auto.assign = F)

    # 检查是否获取成功

    if (is.OHLC(stock) == TRUE) {

        print("股票获取成功")

        return(stock)

    } else {

        return(NULL)

        throwWrong()

    }

}

# web 获取股票

    # @params: stock\_name 股票名称

    # @return: OHLC\_ful 获得的股票

getStockByWeb <- function(stock\_name) {

    symbol = stock\_name # stock code

    from = as.Date("2003-07-03")

    to=Sys.Date()+90 #设置最后日期，seq函数不能识别当季值，截止日期向前推一个季度

    time.index=seq.Date(from=from,to=to,by="quarter") #设置季度节点的数据索引

    year.id=year(time.index)

    quarter.id=quarter(time.index) #设置日期元素，与网页数据特征对应

    OHLC=list()

    for(t in 1:length(time.index)){

        year=year.id[t]

        season=quarter.id[t]

        url=paste0("http://quotes.money.163.com/trade/lsjysj\_",symbol,".html?year=",year,"&season=",season)

        web=read\_html(url) #Read HTML or XML

        xpath="/html/body/div[2]/div[4]/table"

        web.table=html\_table(html\_nodes(web,xpath=xpath))

        OHLC[[t]]= web.table[[1]]

    }

    OHLC\_1<-OHLC[[1]]

    OHLC\_ful<-OHLC\_1[order(OHLC\_1[,1],decreasing=F),]

    for(i in 2:length(OHLC)){

        OHLC\_conti=OHLC[[i]]

        OHLC\_conti\_ful=OHLC\_conti[order(OHLC\_conti[,1],decreasing=F),]

        OHLC\_ful<-rbind(OHLC\_ful,OHLC\_conti\_ful)

    }

    return(OHLC\_ful)

}

# web 获取特殊股票f

    # @params: stock\_name 股票名称

    # @return: OHLC\_ful 获得的股票

getSpecialStockByWeb <- function(stock\_name, label) {

    symbol = stock\_name # stock code

    from = as.Date("2003-07-03")

    to=Sys.Date()+90 #设置最后日期，seq函数不能识别当季值，截止日期向前推一个季度

    time.index=seq.Date(from=from,to=to,by="quarter") #设置季度节点的数据索引

    year.id=year(time.index)

    quarter.id=quarter(time.index) #设置日期元素，与网页数据特征对应

    OHLC=list()

    for(t in 1:length(time.index)){

        year=year.id[t]

        season=quarter.id[t]

        url=paste0("http://quotes.money.163.com/trade/lsjysj\_zhishu\_","399001",".html?year=",year,"&season=",season)

        web=read\_html(url) #Read HTML or XML

        xpath="/html/body/div[2]/div[3]/table"

        web.table=html\_table(html\_nodes(web,xpath=xpath))

        OHLC[[t]]= web.table[[1]]

    }

    OHLC\_1<-OHLC[[1]]

    OHLC\_ful<-OHLC\_1[order(OHLC\_1[,1],decreasing=F),]

    for(i in 2:length(OHLC)){

        OHLC\_conti=OHLC[[i]]

        OHLC\_conti\_ful=OHLC\_conti[order(OHLC\_conti[,1],decreasing=F),]

        OHLC\_ful<-rbind(OHLC\_ful,OHLC\_conti\_ful)

    }

    return(OHLC\_ful)

}

# 定义绘图函数

    # @params: stock 绘制股票

    # @params: needs 需要绘制哪些图

    # @return:

paintNeedGraphs <- function(stock, type) {

    # 日数据图和成交量

    # 取名个推是因为选取股票是300766

    if (type == 'quantmod') {

        chartSeries(stock, type = "line", minor.ticks = FALSE, theme = "white", name="STOCK")

    }

    else if(type == 'web') {

        stock.heavy.ohlc<-as.data.frame(lapply(stock,as.numeric))

        stock.heavy.ohlc<-as.xts(stock.heavy.ohlc,order.by=as.Date(stock[,1]))

        vec<-c(1,8,9)

        stock.heavy.ohlc\_2<-stock.heavy.ohlc[,-vec]

        stock.heavy.price<-stock.heavy.ohlc\_2[,4]

        plot(stock.heavy.price,type="l")

        hist(stock.heavy.price,nclass=30,main="STOCK")

    }

}

# 基本数据分析

    # @params: stock 股票

    # @return:

BasicAnalysis <- function(stock, stock\_data) {

    print("基本数据分析信息：")

    a <- summary(stock)

    b <- basicStats(stock\_data)

    c <- stat.desc(stock\_data, basic = TRUE, desc = TRUE, norm = TRUE, p = 0.9)

    print(a)

    print('调用fBasics--------------------------------------')

    print(b)

    print('调用pastecs 数据分析--------------------------------------')

    print(c)

}

# 报错

    # @return: string

throwWrong <- function() {

    print("出现异常失败，请重试")

}

主逻辑代码展示

    Q1 <- function(stock, v, type, label) {

        # 拼接股票名称

        stock\_name <- paste(stock, '.', v, sep = "", NULL)

        paste('所需获取的股票名称为', stock\_name, "", NULL)

        print(label)

        # 获取股票

        stock <- getStockByName(stock\_name, stock, type, label)

        Sys.sleep(10)

        # 绘图

        paintNeedGraphs(stock, type)

        # 基本数据分析

        if (label == "sz") {

            BasicAnalysis(stock, stock[, 5])

        } else {

            BasicAnalysis(stock, stock[, 4])

        }

    }

    # ------------------------------------------------

    # | <quantmod> | <web> | <Tushare> |

    Q1(stock, 'sz', 'quantmod', 'undefind')

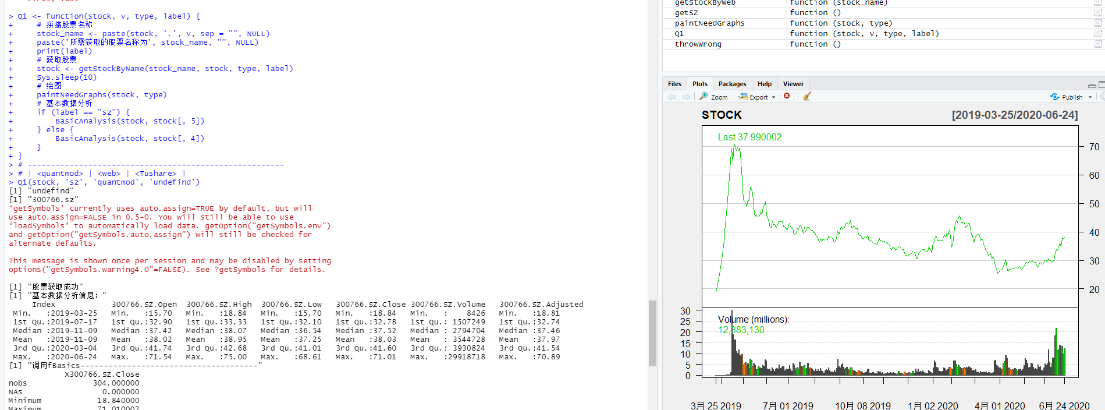
# Q1(stock, 'sz', 'web', 'undefind')

# Q1(stock, 'sz', 'Tushare', 'undefind')

# ------------------------------------------------

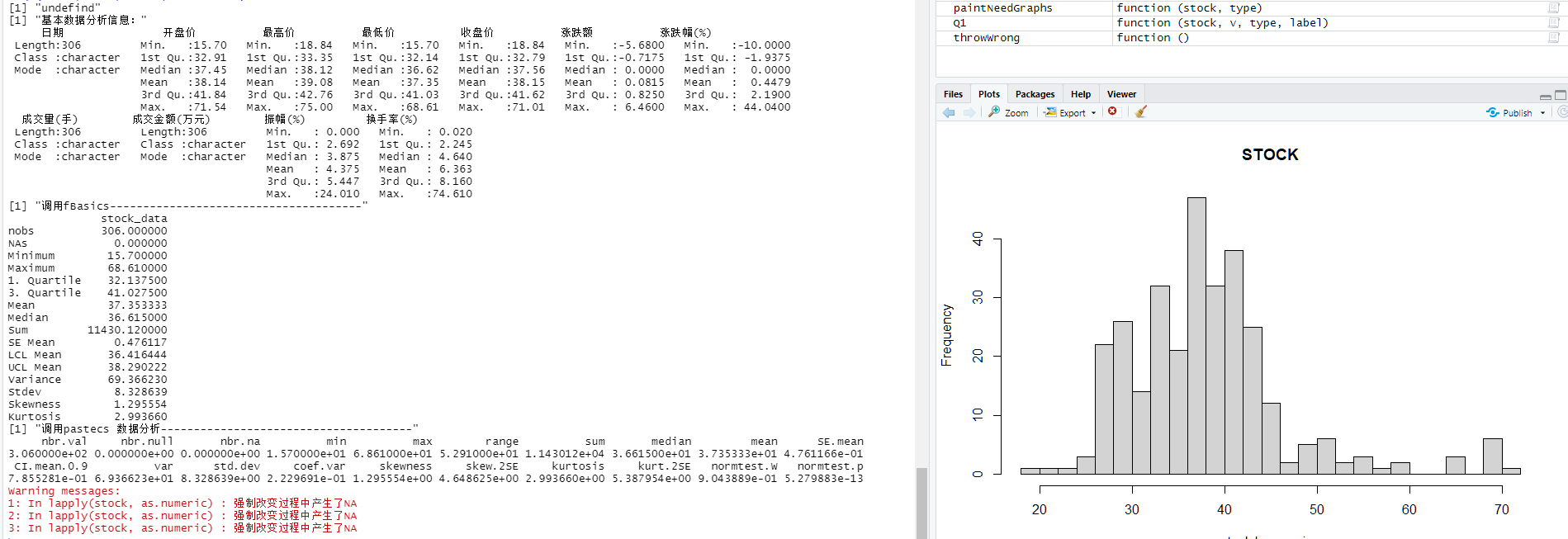
## 1.2 运行结果

### 1.2.1 使用quantmod 包

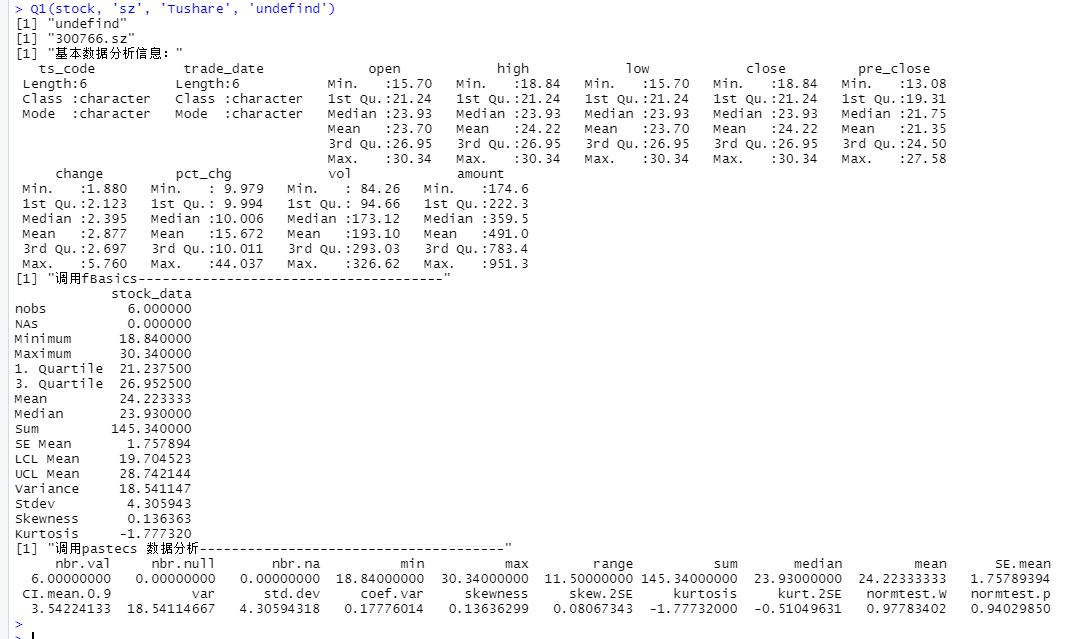




### 1.2.2 使用web爬虫



### 1.2.3 使用Tushare



## 1.3 差异分析

使用quantmod包进行数据抓取，安全快捷，直观可视。但是存在适配性问题，如果存在一些原包没有提供方法的需求，需要开发者进行二次封装，降低代码可读性。

使用web爬虫爬取数据，需要对网页结构，相关协议有一定了解，爬取难度视网页安全程度而有所补益。同时还存在爬取数据因网页数据结构的不同而不同，需要开发者进行数据接口适配。优点在于完整。原始数据可信度高。

使用Tushare包获取数据本质上与使用quantmod包获取数据一致，只是数据采用了Tushare数据库中数据，有商业公司信誉为数据做担保，可信度高。

# 运用上述三种网络爬虫方法抓取深圳市场指数数据

## 代码演示

# -------------------------------------------------------

# 2 运用上述三种网络爬虫方法抓取深圳市场指数数据

# -------------------------------------------------------

    # @params: stock 股票代码

    # @params: v 股票板块

    # @return:

    Q2 <- function(stock, v, type) {

        # 方法与第一题一致

        # 为表不同，闭包调用

        Q1(stock, v, type, 'sz')

    }

    # -------------------------------------------------------

    # | <quantmod> | <web> | <Tushare> |

    SZZS <- "399001"

    Q2(SZZS, 'sz', 'quantmod')

    # Q2(SZZS, 'sz', 'web')

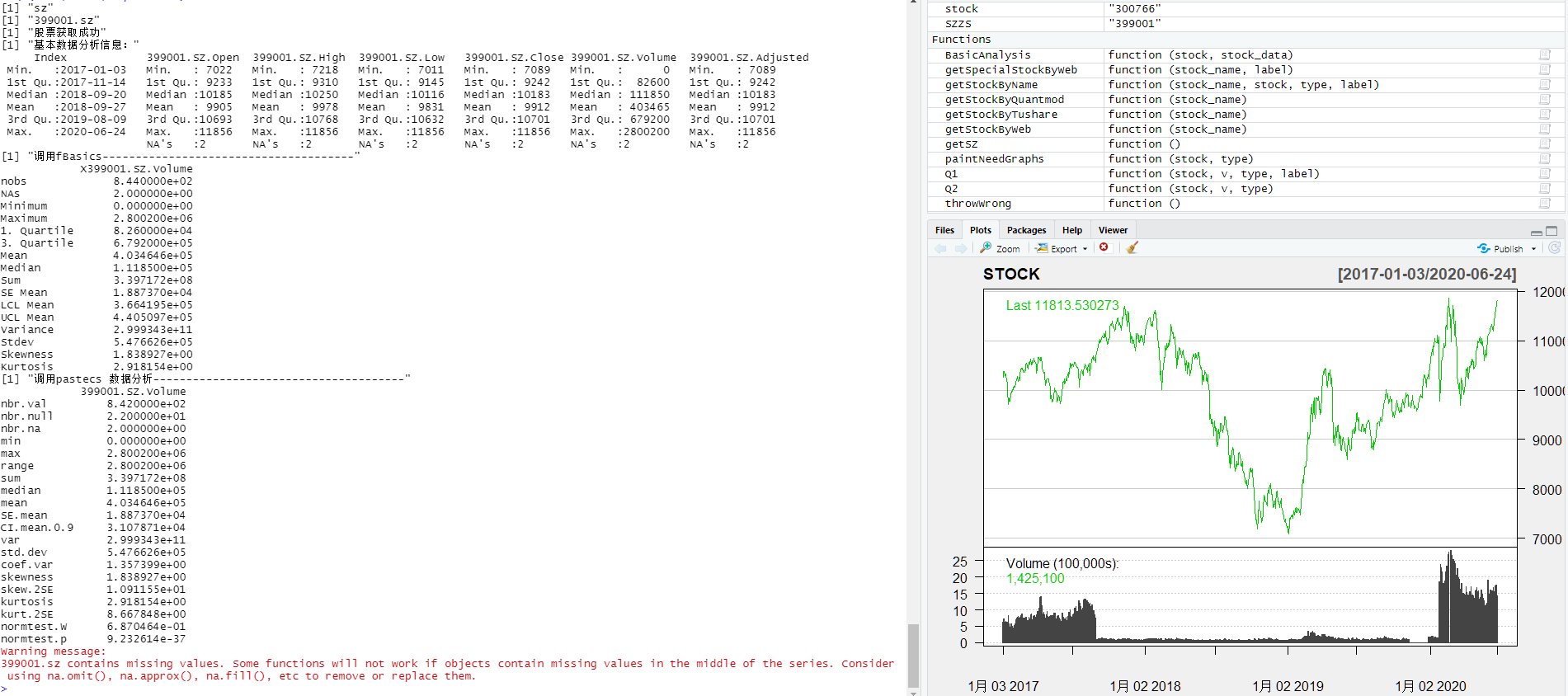
    # Tushare 暂未实现

    # Q2(SZZS, 'sz', 'Tushare')

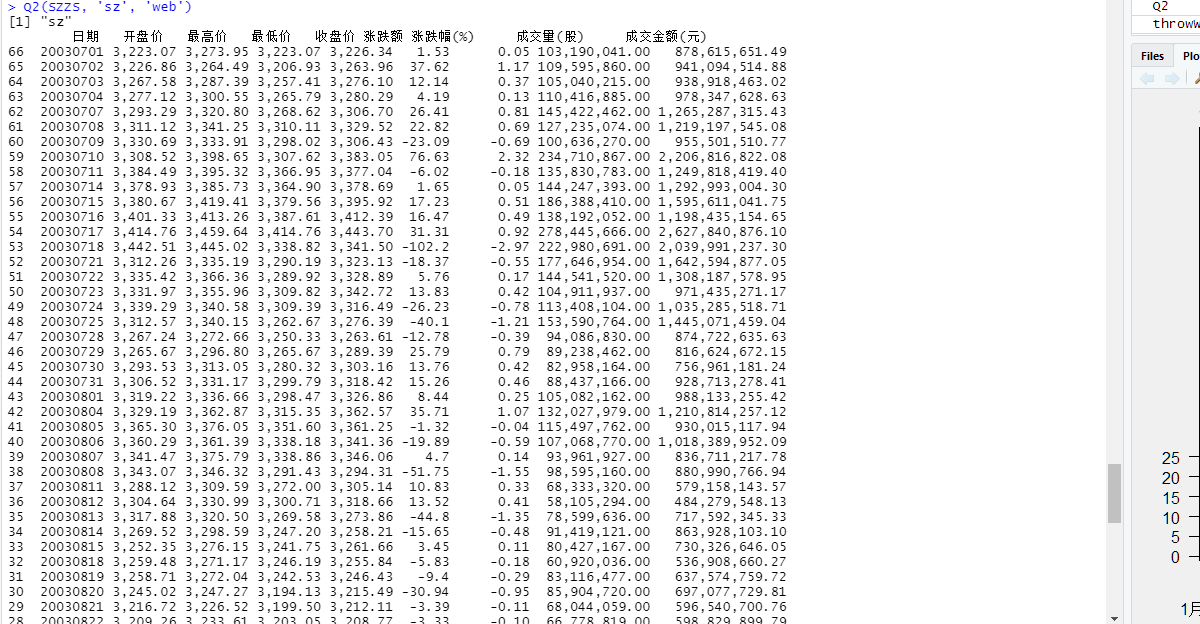
    # -------------------------------------------------------

## 运行结果

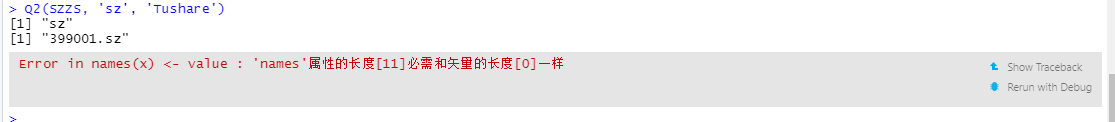
### 2.2.1 使用quantmod包



### 2.2.2 使用web爬虫



### 2.2.3 使用Tushare



# 并运用单一指数模型测度所选取股票的风险

## 3.1 代码演示

# -------------------------------------------------------

# 3 运用单一指数模型测度所选取股票的风险

# -------------------------------------------------------

    title <- "300766.sz"

    # 计算调整收盘价

    # 每日互动收盘价

    GETUI\_price <- getStockByQuantmod(title)

    GETUI\_price <- na.approx(GETUI\_price)[, 6]

    # 深圳成指收盘价

    SZCZ\_price <- getStockByQuantmod("399001.sz")

    SZCZ\_price <- na.approx(SZCZ\_price)[, 6]

    Sys.sleep(2)

    # 依据调整收盘价计算收益率

    GETUI.r <- dailyReturn(GETUI\_price, subset = "2019-03-25/2020-06-24",type = "log")

    SZCZ.r <- dailyReturn(SZCZ\_price, subset = "2019-03-25/2020-06-24",type = "log")

    # 将序列转换为时间序列对象

    GETUI.r<-ts(GETUI.r, start = c(2019,03,25), freq = 60)

    SZCZ.r<-ts(SZCZ.r, start = c(2019,03,25), freq = 60)

    print("单一资产和市场收益率的相关性分析")

    # 运用cor函数计算两序列的相关系数

    cor(GETUI.r, SZCZ.r)

    # 绘制散点图

    plot(GETUI.r, SZCZ.r)

    # 构建市场模型并估计参数

    market <- lm(GETUI.r~SZCZ.r)

    summary(market)

    market<-lm(GETUI.r~0+SZCZ.r)

    summary(market)

    # 模型残差序列的检验

    # 观察残差序列及其自相关性

    par(mfrow=c(2,1))

    plot(market$residuals,type="l")

    acf(market$residuals,lag=36)

    # 对残差序列进行白噪声检验

    Box.test(market$residuals,lag=36)

    # 单一资产与市场指数收益率最小二乘线性拟合图解

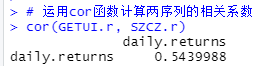
    par(mfrow=c(1,1))

    plot(GETUI.r, SZCZ.r, cex = 1)

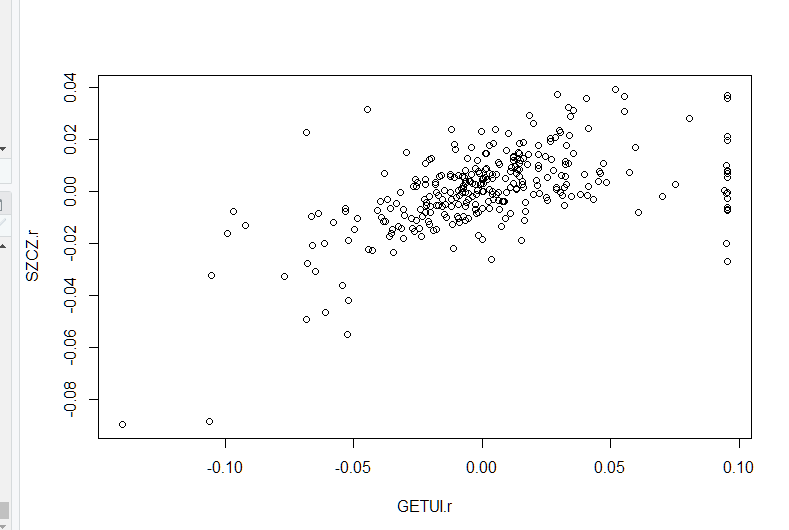
    abline(0, market$coefficients)

## 3.2 运行结果

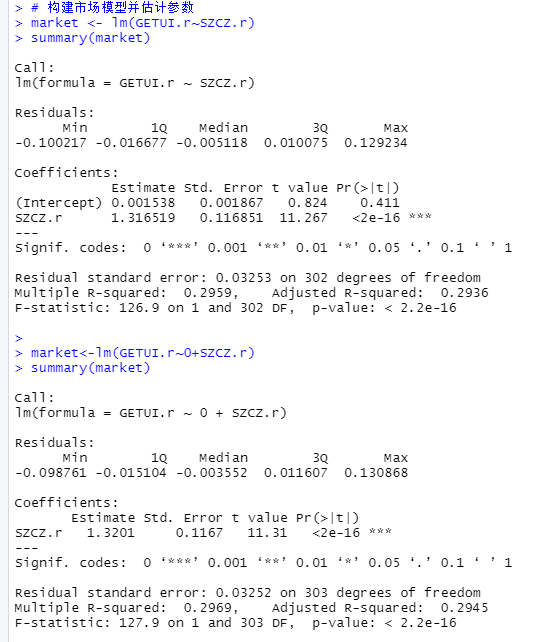
### 3.2.1 两序列相关系数



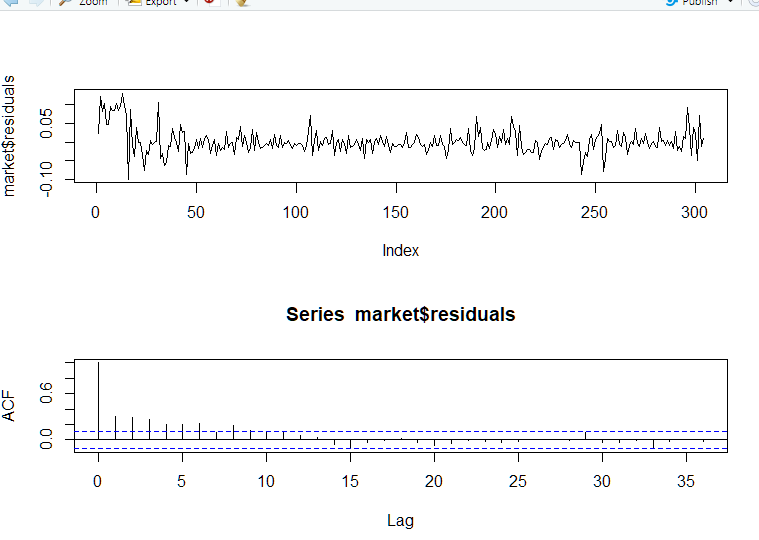
### 3.2.2 散点图

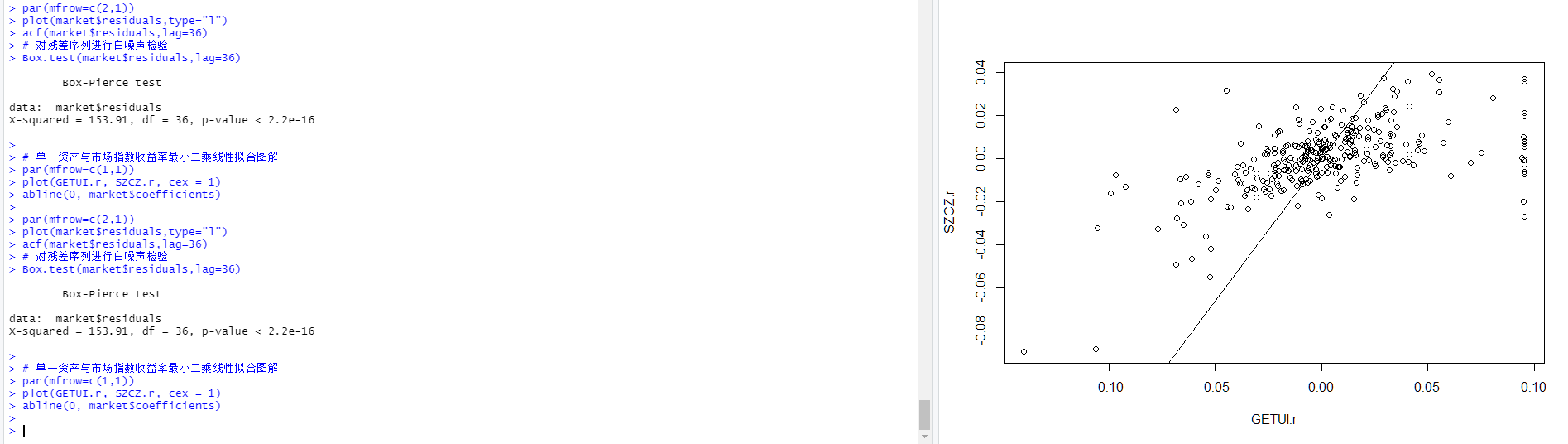


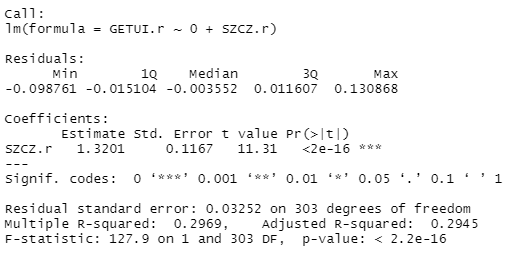
### 3.2.3 市场模型并估计参数



### 3.2.4 模型残差序列的检验







## 3.3 结论

因为单一资产与市场指数收益率最小二乘线性拟合图较散，代表单一资产面临的系统性风险在其总体风险中的占比较低。

同时单一资产资格变量相对市场价格变动的敏感度亦较低。股票风险相对较小。