Report

Dian Zhang, ID: 2016110592

The report shows answers of the first assignment. The answer was calculated by 2 different ways.

In Case A the return rates of stock (RRS) was calculated by geometrical mean, and RRS of those companies which have exited the market are set as -100. In Case B, however, I suppose that all people have sold the stock before the stock has been dropped out. The specific way the RSS is progressed will be explain in the case.

Besides, methods of calculating the weight are also totally different. The necessary explanation of all the method will be given in each case. If you want to furtherly know how the dataset is progressed, the Appendix illustrates all the code.

I. Case A

The table below demonstrates the **variance**, **skewness** and **kurtosis** of ***rrpercent(RRS)***, and the **weighted mean** is given below the table.

The necessary explanation of the table is given below. If you want to furtherly know how the dataset is progressed, ***Appendix I*** illustrates the ***Python code***, which is used to get the using data, and the ***Stata code***, which is used to analyze the using data and create the table.

Table 1.1:

```
. sum rrpercent,d
```

rrpercent

Percentiles		Smallest		
1%	-100	-100		
5%	-100	-100		
10%	-.0225136	-100	Obs	3,595
25%	.0028961	-100	Sum of Wgt.	3,595
50%	.0137453		Mean	-9.608471
		Largest	Std. Dev.	29.502
75%	.0252646	.0929311		
90%	.0383742	.0946137	Variance	870.3678
95%	.0466051	.103005	Skewness	-2.738003
99%	.0694001	.1103066	Kurtosis	8.496663

Weighted mean of rrpercent(%) = Σ(rrpercent * weight) / Σweight = -7.49242%

weight = start_trade_market_value / start_adjust_price

rrpercent = power ((final_ adjust_price/ start_adjust_price) ,1/T)-1

rrpercent: return rate% (percent),

T = 250*(17-0+1)

II. Case B

The tables below demonstrate **mean**, **variance**, **skewness** and **kurtosis** of ***rrpercent*** ***(return rate%)***. In table 2.1 I don't use weight while in table 2.2 I use.

In Case B I use a different method. I use log transformation of adjusted prices and calculate the first difference (DF) which represents the daily RRS (DRRS). Calculating the numerical mean of DF, then we get the numerical mean of DRRS of one stock.

I also use different method to calculate weight. In this case, weight equals to start tarde market value. If you want to furtherly know how I progress the dataset, ***Appendix II*** shows you all the ***Stata code*** (I wrote this case in stata), and I will explain my code step by step.

Table 2.1:

```
. sum rr_percent, d

                    rr_percent, unweighted
-------------------------------------------------------------
      Percentiles      Smallest
 1%    -.2562088      -36.19199
 5%    -.0656926      -17.7731
10%    -.0170969      -3.947043       Obs                3,557
25%     .0125904      -3.714307       Sum of Wgt.        3,557

50%     .0359999                      Mean            .1140143
                       Largest        Std. Dev.       .8794515
75%     .0893568       9.529829
90%     .2944857       9.532267       Variance        .7734349
95%     .5354348       9.533882       Skewness       -19.34681
99%      1.80911       9.536128       Kurtosis        893.4509

. sum rr_percent [fweight = intwei], d
```

Table 2.2:

```
. sum rr_percent [fweight = intwei], d

                    rr_percent, unweighted
-------------------------------------------------------------
      Percentiles      Smallest
 1%    -.2115781      -36.19199
 5%    -.1121904      -17.7731
10%    -.0586203      -3.947043       Obs           4.1987e+12
25%    -.0144448      -3.714307       Sum of Wgt.   4.1987e+12

50%     .0190532                      Mean            .0700882
                       Largest        Std. Dev.       .6227696
75%     .0480365       9.529829
90%     .1356285       9.532267       Variance        .3878419
95%     .2884372       9.533882       Skewness       -11.98815
99%      1.56559       9.536128       Kurtosis        1147.484

.
```

Appendix I

Python Code: To change sourcing data to using data

```
# -*- coding: utf-8 -*-

import csv
import os
import pandas as pd
#批处理板块
path = r'C:\Users\johnz\Desktop\数据工作\Python\投资经济学作业 1\投资经济学作业
1\DataSource\stock data'
dirs = os.listdir(path)#获取 CSV 文件名字

data_output = []
data_output1 = []
for file in dirs:
    filename = path + '\\'+file
    file = os.path.splitext(file)[0]#将文件名字的扩展名去掉，直接得到文件名字

    with open(filename) as f:
        reader = csv.reader(f)
        data1 = {}
        data2 = []

        data_output = []
        for row in reader:
            data1[row[1]] = row
            data2.append(row)
    #data 中 data[12]是后复权价格。[9]是交易市值
    #权重=交易市值 / 后复权价格
    #日均回报率 ：start_price*(1+rr)^T=final_price
    #所以 rr = (final_price / start_price) ^(1/T)-1      T = 4500(day)

    if ('2017-12-29'in data1.keys()) == True:
        Final = data1['2017-12-29']
    else:
        Final = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
    if ('2000-01-04'in data1.keys()) == True:
        Start = data1['2000-01-04']
    else:
        Start = data2[-1]
    final_traded_market_value = float(Final[9])
```

```
        final_adjust_price = float(Final[12])
        start_adjust_price = float(Start[12])
        start_traded_market_value = float(Start[9])

        weight = start_traded_market_value/start_adjust_price
        rr = pow( (final_adjust_price / start_adjust_price),(1/4500))-1
        rrpercent = rr*100
        #print (rrpercent)  测试成功

        this_code = {'code':Start[0],'weight':weight,'rrpercent':rrpercent}
        this_code1 = [weight,rrpercent]
        this_code1.insert(0,Start[0])
        data_output1.append(this_code1)

        print('code:'+Start[0])

name = ['code','weight','rrpercent']
Data_Using = pd.DataFrame(columns = name,data = data_output1)
#print (Data_Using)
Data_Using.to_csv(r'C:\Users\johnz\Desktop\数据工作\Python\投资经济学作业 1\投资经济学作业 1\DataUsing\stock_weight_rrprecent.csv')
```

Stata Code: Analyze using data (In this part I only use stata to analyze the means, variance, skewness and kurtosis. To calculate the weighted mean, I use Excel and easily get the answer.)

```
. use "C:\Users\johnz\Desktop\数据工作\Python\投资经济学作业 1\投资经济学作业 1\DataUsing\stock_weight_rrprecent.csv"
. set more off
. sum rrpercent, d
```

Appendix II

Data sourcing (1st_part) :

In this part I change files in form of csv to dta (a form of dataset in stata).

```
set more off

cd "C:\Users\johnz\Desktop\数据工作\Python\投资经济学作业1\投资经济学作业1\DataSo
clear
local myfilelist : dir . files"*.csv"
/*The first part*/
foreach file of local myfilelist {
drop _all
insheet using `file'
local outfile = subinstr("`file'",".csv","",.)

save "`outfile'", replace

}
```

Data sourcing (2nd_part) :

In this part I generate date variable and replace the AP in order to furtherly progress the data.

```
set more off

clear
local myfilelist : dir . files"*.dta"
/*The second part*/
foreach file of local myfilelist {
drop _all
    use `file'
    gen date1 = date(date,"YMD")
        label variable date1 "unconsist date"
    tsset date1
    replace adjust_price = 1 if (date1 <14613 | date1 >21182)

save `file', replace

}

```

Data sourcing (3rd part) :

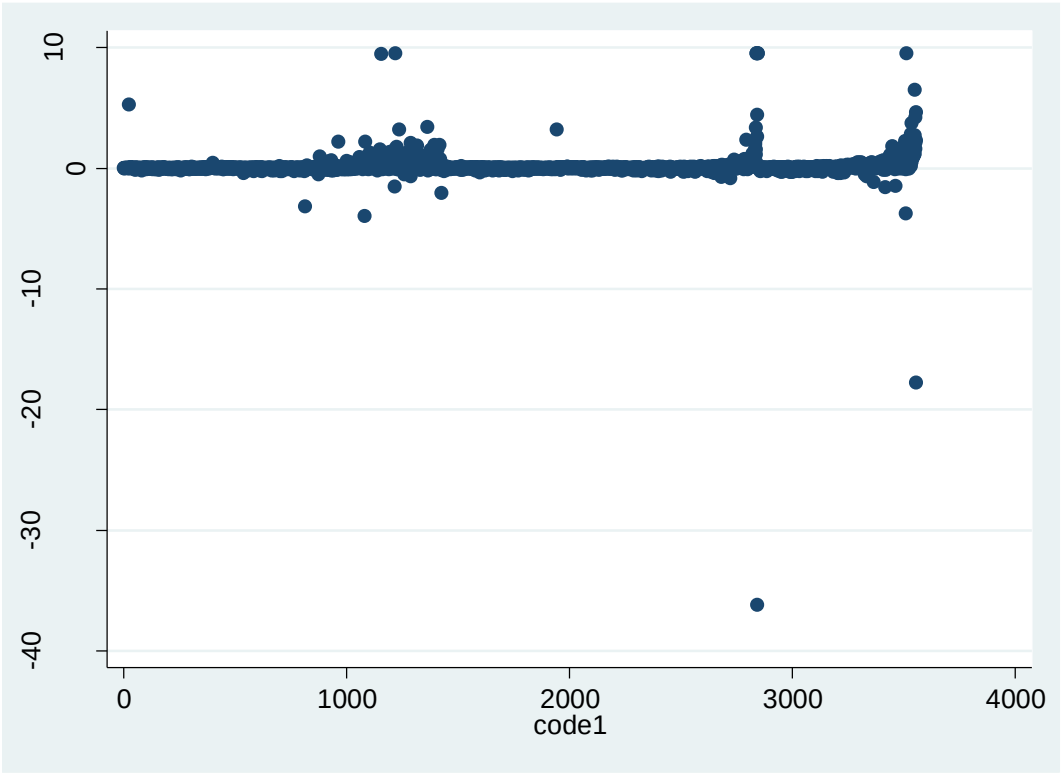
In this part I gain the DRRS.

```
foreach file of local myfilelist {
drop _all
    use `file'

    generate in_adjprice = log(adjust_price)
        label variable in_adjprice "log(adjust_price)"
    sort date1

    generate dinad_price = in_adjprice[_n+1] - in_adjprice[_n] if (date1 >14613
        label variable dinad_price "daily growth rate of adjprice"
    egen sumdinad_price = mean(dinad_price) if (date1 >14613 & date1 <21182)
        label variable sumdinad_price "mean(dinad_price)"

    /*This is In(price_first)-In(Price_last)
      Or you can consider this var as the return rate of this stock.
    */
    gen weight = traded_market_value
    gen intwei = int(weight)
        label variable weight "This is TMV."
        label variable intwei "This is TMV.int"

    gen wei_rrper = sumdinad_price * weight if (date1 >14613 & date1 <21182)
        label variable wei_rrper "This is the weighted return rate"
    /*This is the weighted return rate we need to use later. */
    gen rr_percent = sumdinad_price * 100 if (date1 >14613 & date1 <21182)
        label variable rr_percent "rr_percent, unweighted"

    keep  code date1 weight wei_rrper rr_percent intwei
    /*Now we just keep some Var that we need to use.*/


    /*Now we just need to keep one obs which contains all the data we need
    of one stock. */
save `file',replace
}

clear
```

Data appending and motherfile analyzing:

In this part I append all the ".dta" files to a mother file and analyze it.

```
clear
local myfilelist : dir . files"*.dta"

foreach file of local myfilelist{
appen using `file'
}/*Now we have dataset we need in memory.*/

save motherfile/*We save the dataset we want to use in a new file
                  called `motherfile'. */

clear

use motherfile
drop if wei_rrper == .

encode code, gen(code1)

sort code1
by code1: egen min_date = min(date1)
by code1: keep if date1 == min_date

/*by code1: keep if date1 == max_date*/

save motherfile, replace


sum rr_percent, d
sum rr_percent [fweight = intwei], d
```


10
0
-10
-20
-30
-40
0
1000
2000
3000
4000
code1