Homework 1 Corporate Finance

YiTao Hu

Executive Summary

We computed the gross and excess returns of S&P500 stock index from 1973 January 1st to 2018 December 31st. For the index return data, we used daily index return collected from CRSP database. As for the risk-free rates, we used monthly, annual and five-year Treasury bond rates. Also, we used Federal Fund Overnight borrowing rates as a good approximation.

Homework Text

The daily, monthly, annual and five year gross arithmetic and geometric return is computed using the following formulas respectively:

$$\bar{R}_{arth} = \frac{\sum_t^T R_t}{T} \quad \text{and} \quad \bar{R}_{geo} = \sqrt[T]{\prod_t^T (1 + R_t)} - 1.$$

This gives us the results of the gross returns:

In [21]:

```
gross_rtn_df
```

Out[21]:

	Athematic	Geometric
Daily	0.110001	0.093286
Monthly	0.101907	0.088891
Yearly	0.104561	0.088540
Five_Year	0.096927	0.077514

For the excess return, we subtract the risk-free rate from the stock index return. Mathematically, the arithmetic and geometric mean is computed as follows:

$$\bar{R}_{arth} = \frac{\sum_t^T (R_t - r_f)}{T} \quad \text{and} \quad \bar{R}_{geo} = \sqrt[n]{\frac{\prod_t^T (1 + R_t)}{\prod_t^T (1 + r_{ft})}}.$$

This gives us the results of the gross returns:

In [22]:

```
excess_rtn_df
```

Out[22]:

	Athematic	Geometric
Daily	0.069190	0.053081
Monthly	0.044305	0.031834
Yearly	0.052688	0.035477
Five_Year	0.049050	0.022451

Also, we annualize the returns using the following formula:

$$\left(1+\frac{\bar{R}}{N}\right)^{N}-1$$

Code

The code used to generate the outputs above is as followed:

In [17]:

```
import numpy as np
import pandas as pd
from scipy import stats
```

In [18]:

```
SP500_df=pd.read_csv('S&P500.csv')
Risk_free_Rates=pd.read_csv('risk_free_rates.csv')

SP500_df=SP500_df.set_index(pd.to_datetime(SP500_df['caldt']))
Risk_free_Rates=Risk_free_Rates.set_index(pd.to_datetime(Risk_free_Rates['date'
]))

returns=pd.merge(SP500_df,Risk_free_Rates,left_index=True,right_index=True)
returns=returns.drop(columns=['caldt','date'],axis=1)
returns=returns.dropna()

returns[['ED_M1','FF_O','TCMNOM_Y1','TCMNOM_Y5']]=returns[['ED_M1','FF_O','TCMNO
M_Y1','TCMNOM_Y5']]/100
returns['ED_M1']=(returns['ED_M1']+1)**(1/12)-1
returns['FF_O']=(returns['FF_O']+1)**(1/360)-1
returns['TCMNOM_Y5']=(returns['TCMNOM_Y5']+1)**(5)-1
SP500_returns=returns[['vwretd','ewretd']]
```

In [19]:

```
Risk_free_rates=returns[['FF_O','ED_M1','TCMNOM_Y1','TCMNOM_Y5']]

arthi_d_rtn=SP500_returns.mean()#compute arthimatic mean return
arthi_d_rtn=(arthi_d_rtn+1)**260-1
geo_d_rtn=stats.gmean(SP500_returns+1)-1 #compute daily geometric return
geo_d_rtn=(geo_d_rtn+1)**260-1
arth_d_ex_rtn=SP500_returns.mean()-Risk_free_rates.FF_O.mean() #compute daily ar
ithmatic ex-return mean
arth_d_ex_rtn=(arth_d_ex_rtn+1)**260-1
geo_d_ex_rtn=(stats.gmean(SP500_returns+1)/stats.gmean(Risk_free_rates.FF_O+1)-1
) #compute daily geometric ex-return
geo_d_ex_rtn=(geo_d_ex_rtn+1)**260-1
#convert daily to monthly return
monthly_rtn=SP500_returns.resample('M',label='right').apply(lambda r: ((r+1).cum
prod()-1).last('D'))
monthly_Rf=Risk_free_rates.ED_M1.resample('M').apply(lambda r: r.last('D'))
arth_m_rtn=monthly_rtn.mean()
arth_m_rtn=(arth_m_rtn+1)**12-1
geo_m_rtn=stats.gmean(monthly_rtn+1)-1
geo_m_rtn=(geo_m_rtn+1)**12-1
arth_m_ex_rtn=monthly_rtn.mean()-monthly_Rf.mean()
arth_m_ex_rtn=(arth_m_ex_rtn+1)**12-1
geo_m_ex_rtn=(stats.gmean(monthly_rtn+1)/stats.gmean(monthly_Rf+1)-1)
geo_m_ex_rtn=(geo_m_ex_rtn+1)**12-1
#convert daily to annual return
annual_rtn=SP500_returns.resample('Y',label='right').apply(lambda r: ((r+1).cump
rod()-1).last('D'))
annual_Rf=Risk_free_rates.TCMNOM_Y1.resample('Y').apply(lambda r: r.last('D'))
arth_y_rtn=annual_rtn.mean()
geo_y_rtn=stats.gmean(annual_rtn+1)-1
arth_y_ex_rtn=annual_rtn.mean()-annual_Rf.mean()
geo_y_ex_rtn=(stats.gmean(annual_rtn+1)/stats.gmean(annual_Rf+1)-1)
#convert daily to 5-year return
five_Y_rtn=SP500_returns.resample('5Y',label='right').apply(lambda r: ((r+1).cum
prod()-1).last('D'))
five_yr_Rf=Risk_free_rates.TCMNOM_Y5.resample('5Y').apply(lambda r: r.last('D'))
five_y_arth=five_Y_rtn.mean()
five_y_arth=(1+five_y_arth)**0.2-1
five_y_geo=stats.gmean(five_Y_rtn+1)-1
five_y_geo=(1+five_y_geo)**0.2-1
arth_5y_ex_rtn=five_Y_rtn.mean()-five_yr_Rf.mean()
arth_5y_ex_rtn=(1+arth_5y_ex_rtn)**0.2-1
geo_5y_ex_rtn=(stats.gmean(five_Y_rtn+1)/stats.gmean(five_yr_Rf+1)-1)
geo_5y_ex_rtn=(1+geo_5y_ex_rtn)**0.2-1
```

In [20]:

```
gross_rtn_df=pd.DataFrame(data=[[arthi_d_rtn[0],geo_d_rtn[0]],[arth_m_rtn[0],geo
_m_rtn[0]],[arth_y_rtn[0],geo_y_rtn[0]],
                                [five_y_arth[0],five_y_geo[0]]],index=['Daily','M
onthly','Yearly','Five_Year'],
                           columns=['Athematic','Geometric'])
excess_rtn_df=pd.DataFrame(data=[[arth_d_ex_rtn[0],geo_d_ex_rtn[0]],[arth_m_ex_r
tn[0],geo_m_ex_rtn[0]],[arth_y_ex_rtn[0],geo_y_ex_rtn[0]],
                                [arth_5y_ex_rtn[0],geo_5y_ex_rtn[0]]],index=['Dai
ly','Monthly','Yearly','Five_Year'],
                           columns=['Athematic','Geometric'])
```