RISK_MNGM_HW7_Problem2&3

October 27, 2016

In [7]:
```
######## Homework 7 Problem 2  Part 1 --- AMD & INTC's Historic & formula VaR ##############
import numpy as np
import scipy.stats as ss
import pandas as pd
import math
import matplotlib.pyplot as plt
import datetime as dt


path = '~/Documents/Semester3/M5320/homework/HW4/INTC-yahoo.csv'
sto_AMD= pd.read_csv(path, header = 0)
sto_AMD.shape
#data1 = sto_AMD.values[1:252*20,:]
data1 = sto_AMD.values
#data1 = data1[::-1]
AMD_close = list(data1[:,6])


A_log_rtn = []
A_log_rtn_sq = []

for i in range(1 ,len(data1)-1):
    log_return = math.log( AMD_close[i]/AMD_close[i+1] )
    A_log_rtn.append(log_return)
    A_log_rtn_sq.append(log_return**2)

def vol_and_mu(years):
    S0 = 10000
    T = 5/252
    p = 0.99

    A_vol_years = []
    A_mu_years = []
    A_VaR_years = []
    A_ES_years =[]

##### GBM Formula VaR and ES #######
    for i in range(len(data1)-252*years):
        vol_years = np.std(A_log_rtn[i:i+252*years]) * np.sqrt(252)
        mu_years = np.mean(A_log_rtn[i:i+252*years])*252 + (vol_years**2)/2
        VaR_years = S0 - S0 * np.exp( vol_years * T**(0.5)* ss.norm.ppf(1-p) + (mu_years - pow(
        ES_years = S0 * (1 - np.exp(mu_years *T)/(1-0.975) * ss.norm.cdf(ss.norm.ppf(1-0.975) -
```

```
        A_vol_years.append(vol_years)
        A_mu_years.append(mu_years)
        A_VaR_years.append(VaR_years)
        A_ES_years.append(ES_years)

    return(A_vol_years, A_mu_years, A_VaR_years, A_ES_years)


A_VaR_5years = vol_and_mu(5)[2][1:252*20]
A_ES_5years = vol_and_mu(5)[3][1:252*20]

###### Historic VaR and ES ########

A_hist_log_rtn = []
A_hist_log_rtn_sq = []

for i in range(1, len(data1)-5):
    hist_log_return = math.log( AMD_close[i]/AMD_close[i+5] )
    A_hist_log_rtn.append(hist_log_return)
    A_hist_log_rtn_sq.append(hist_log_return**2)



def His_VaR_and_ES(years):
    S0 = 10000
    T = 5/252
    p = 0.99

    A_his_VaR_years = []
    A_his_ES_years = []

    #A_his_ES_years = []

    for i in range(len(data1)-252*5):

        his_VaR_list = sorted(A_hist_log_rtn[i:i+252*5])
        his_VaR_years = (-S0) * his_VaR_list[13]
        A_his_VaR_years.append(his_VaR_years)

        hist_ES_list = [(-S0)*x for x in his_VaR_list[1:32]]
        hist_ES_years = np.mean(hist_ES_list)
        A_his_ES_years .append(hist_ES_years)

    return(A_his_VaR_years, A_his_ES_years)

A_his_VaR_years = His_VaR_and_ES(5)[0][1:252*20]
A_his_ES_years  = His_VaR_and_ES(5)[1][1:252*20]



timeline = data1[1:252*20,0]
timeline = [dt.datetime.strptime(d,'%Y-%m-%d').date() for d in timeline]

fig, ax = plt.subplots()
```

```
ax.plot(timeline, A_VaR_5years,'g-', label='formula VaR 5 year')
ax.plot(timeline, A_ES_5years,'c-', label='formula ES 5 year')


ax.plot(timeline, A_his_VaR_years,'r-', label='historic VaR 5 year')
ax.plot(timeline, A_his_ES_years,'m-', label='historic ES 5 year')



legend = ax.legend(loc='upper right', shadow=True)
ax.set_title('INTC VaRs and ESs formula VS historic')
plt.show()
```

In [11]:
```
######## Homework 7 Problem 2  Part 2 --- portfolio's Historic & formula VaR ###############
import pandas as pd
import operator
import itertools
from operator import add
import numpy as np
import scipy.stats as ss
import math
import matplotlib.pyplot as plt
import datetime as dt

path1 = '~/Documents/Semester3/M5320/homework/HW4/AMD-yahoo.csv'
path2 = '~/Documents/Semester3/M5320/homework/HW4/INTC-yahoo.csv'

AMD= pd.read_csv(path1, header = 0)
INTC= pd.read_csv(path2, header = 0)
#data1 = sto_AMD.values[1:252*20,:]
data1 = AMD.values


AMD_INTC = pd.concat([AMD, INTC], axis = 1, join='inner', keys = 'Date' )
AMD_INTC_price  = AMD_INTC[[('D', 'Date'),('D', 'Adj Close'), ('a', 'Adj Close')]]

df = pd.DataFrame(AMD_INTC_price)


amd = df[[1]].values.tolist()
intc = df[[2]].values.tolist()

flat_amd= list(itertools.chain.from_iterable(amd))
flat_intc = list(itertools.chain.from_iterable(intc))

list1 = [x*640 for x in flat_amd]
list2 = [x*546 for x in flat_intc]

portfolio = [sum(x) for x in zip(list1, list2)]
AMD_close = portfolio


A_log_rtn = []
```

```
A_log_rtn_sq = []

for i in range(1 ,len(data1)-1):
    log_return = math.log( AMD_close[i]/AMD_close[i+1] )
    A_log_rtn.append(log_return)
    A_log_rtn_sq.append(log_return**2)

def vol_and_mu(years):
    S0 = 10000
    T = 5/252
    p = 0.99

    A_vol_years = []
    A_mu_years = []
    A_VaR_years = []
    A_ES_years =[]

##### GBM Formula VaR and ES #######
    for i in range(len(data1)-252*years):
        vol_years = np.std(A_log_rtn[i:i+252*years]) * np.sqrt(252)
        mu_years = np.mean(A_log_rtn[i:i+252*years])*252 + (vol_years**2)/2
        VaR_years = S0 - S0 * np.exp( vol_years * T**(0.5)* ss.norm.ppf(1-p) + (mu_years - pow
        ES_years = S0 * (1 - np.exp(mu_years *T)/(1-0.975) * ss.norm.cdf(ss.norm.ppf(1-0.975)

        A_vol_years.append(vol_years)
        A_mu_years.append(mu_years)
        A_VaR_years.append(VaR_years)
        A_ES_years.append(ES_years)

    return(A_vol_years, A_mu_years, A_VaR_years, A_ES_years)


A_VaR_5years = vol_and_mu(5)[2][1:252*20]
A_ES_5years = vol_and_mu(5)[3][1:252*20]


A_hist_log_rtn = []
A_hist_log_rtn_sq = []

for i in range(1, len(data1)-5):
    hist_log_return = math.log( AMD_close[i]/AMD_close[i+5] )
    A_hist_log_rtn.append(hist_log_return)
    A_hist_log_rtn_sq.append(hist_log_return**2)


def His_VaR_and_ES(years):
    S0 = 10000
    T = 5/252
    p = 0.99

    A_his_VaR_years = []
    A_his_ES_years = []

    #A_his_ES_years = []
```

```
    for i in range(len(data1)-252*5):

        his_VaR_list = sorted(A_hist_log_rtn[i:i+252*5])
        his_VaR_years = (-S0) * his_VaR_list[13]
        A_his_VaR_years.append(his_VaR_years)

        hist_ES_list = [(-S0)*x for x in his_VaR_list[1:32]]
        hist_ES_years = np.mean(hist_ES_list)
        A_his_ES_years .append(hist_ES_years)

    return(A_his_VaR_years, A_his_ES_years)

A_his_VaR_years = His_VaR_and_ES(5)[0][1:252*20]
A_his_ES_years  = His_VaR_and_ES(5)[1][1:252*20]



timeline = data1[1:252*20,0]
timeline = [dt.datetime.strptime(d,'%Y-%m-%d').date() for d in timeline]

fig, ax = plt.subplots()

ax.plot(timeline, A_VaR_5years,'g-', label='formula VaR 5 year')
ax.plot(timeline, A_ES_5years,'c-', label='formula ES 5 year')


ax.plot(timeline, A_his_VaR_years,'r-', label='historic VaR 5 year')
ax.plot(timeline, A_his_ES_years,'m-', label='historic ES 5 year')



legend = ax.legend(loc='upper right', shadow=True)
ax.set_title('portfolio VaRs and ESs formula VS historic')
plt.show()
```

In [12]:
```
######## Homework 7 Problem 3  Part 1 --- AMD & INTC's Historic abs VS log VaR ##############
import numpy as np
import scipy.stats as ss
import pandas as pd
import math
import matplotlib.pyplot as plt
import datetime as dt


path = '~/Documents/Semester3/M5320/homework/HW4/AMD-yahoo.csv'
sto_AMD= pd.read_csv(path, header = 0)
sto_AMD.shape
#data1 = sto_AMD.values[1:252*20,:]
data1 = sto_AMD.values
#data1 = data1[::-1]
AMD_close = list(data1[:,6])
```

```
###### Historic VaR and ES ########

##### abs return #######
A_hist_rtn = []
A_hist_rtn_sq = []

for i in range(1, len(data1)-5):
    hist_return = (AMD_close[i] - AMD_close[i+5])
    A_hist_rtn.append(hist_return)
    A_hist_rtn_sq.append(hist_return**2)



def abs_His_VaR_and_ES(years):
    S0 = 10000

    A_his_VaR_years = []
    A_his_ES_years = []

    for i in range(len(data1)-252*5):

        his_VaR_list = sorted(A_hist_rtn[i:i+252*5])
        his_VaR_years = (-S0) * his_VaR_list[13]/AMD_close[i]
        A_his_VaR_years.append(his_VaR_years)

        hist_ES_list = [(-S0)*x for x in his_VaR_list[1:32]]
        hist_ES_years = np.mean(hist_ES_list)
        A_his_ES_years .append(hist_ES_years)

    return(A_his_VaR_years, A_his_ES_years)

abs_A_his_VaR_years = abs_His_VaR_and_ES(5)[0][1:252*20]
abs_A_his_ES_years  = abs_His_VaR_and_ES(5)[1][1:252*20]

##### log return #######
A_hist_log_rtn = []
A_hist_log_rtn_sq = []

for i in range(1, len(data1)-5):
    hist_log_return = math.log( AMD_close[i]/AMD_close[i+5] )
    A_hist_log_rtn.append(hist_log_return)
    A_hist_log_rtn_sq.append(hist_log_return**2)


def His_VaR_and_ES(years):
    S0 = 10000

    A_his_VaR_years = []
    A_his_ES_years = []

    for i in range(len(data1)-252*5):

        his_VaR_list = sorted(A_hist_log_rtn[i:i+252*5])
        his_VaR_years = (-S0) * his_VaR_list[13]
```

```
        A_his_VaR_years.append(his_VaR_years)

        hist_ES_list = [(-S0)*x for x in his_VaR_list[1:32]]
        hist_ES_years = np.mean(hist_ES_list)
        A_his_ES_years .append(hist_ES_years)

    return(A_his_VaR_years, A_his_ES_years)

A_his_VaR_years = His_VaR_and_ES(5)[0][1:252*20]

##### formula VaR #########

A_log_rtn = []
A_log_rtn_sq = []

for i in range(1 ,len(data1)-1):
    log_return = math.log( AMD_close[i]/AMD_close[i+1] )
    A_log_rtn.append(log_return)
    A_log_rtn_sq.append(log_return**2)

def vol_and_mu(years):
    S0 = 10000
    T = 5/252
    p = 0.99

    A_vol_years = []
    A_mu_years = []
    A_VaR_years = []
    A_ES_years =[]

##### GBM Formula VaR and ES #######
    for i in range(len(data1)-252*years):
        vol_years = np.std(A_log_rtn[i:i+252*years]) * np.sqrt(252)
        mu_years = np.mean(A_log_rtn[i:i+252*years])*252 + (vol_years**2)/2
        VaR_years = S0 - S0 * np.exp( vol_years * T**(0.5)* ss.norm.ppf(1-p) + (mu_years - pow
        ES_years = S0 * (1 - np.exp(mu_years *T)/(1-0.975) * ss.norm.cdf(ss.norm.ppf(1-0.975)

        A_vol_years.append(vol_years)
        A_mu_years.append(mu_years)
        A_VaR_years.append(VaR_years)
        A_ES_years.append(ES_years)

    return(A_vol_years, A_mu_years, A_VaR_years, A_ES_years)


A_VaR_5years = vol_and_mu(5)[2][1:252*20]


timeline = data1[1:252*20,0]
timeline = [dt.datetime.strptime(d,'%Y-%m-%d').date() for d in timeline]

fig, ax = plt.subplots()

ax.plot(timeline, abs_A_his_VaR_years,'g-', label='Historical VaR (abs rtns)')
```

```
ax.plot(timeline, A_VaR_5years,'c-', label='formula VaR')
ax.plot(timeline, A_his_VaR_years,'r-', label='Historical VaR (log rtns)')


legend = ax.legend(loc='upper right', shadow=True)
ax.set_title('AMD VaRs and ESs log VS absolute')
plt.show()
```

In [1]:
```
######## Homework 7 Problem 3  Part 2 --- portfolio's Historic abs VS log VaR ###############
import pandas as pd
import operator
import itertools
from operator import add
import numpy as np
import scipy.stats as ss
import math
import matplotlib.pyplot as plt
import datetime as dt

path1 = '~/Documents/Semester3/M5320/homework/HW4/AMD-yahoo.csv'
path2 = '~/Documents/Semester3/M5320/homework/HW4/INTC-yahoo.csv'

AMD= pd.read_csv(path1, header = 0)
INTC= pd.read_csv(path2, header = 0)
#data1 = sto_AMD.values[1:252*20,:]
data1 = AMD.values


AMD_INTC = pd.concat([AMD, INTC], axis = 1, join='inner', keys = 'Date' )
AMD_INTC_price  = AMD_INTC[[('D', 'Date'),('D', 'Adj Close'), ('a', 'Adj Close')]]

df = pd.DataFrame(AMD_INTC_price)


amd = df[[1]].values.tolist()
intc = df[[2]].values.tolist()

flat_amd= list(itertools.chain.from_iterable(amd))
flat_intc = list(itertools.chain.from_iterable(intc))

list1 = [x*640 for x in flat_amd]
list2 = [x*546 for x in flat_intc]

portfolio = [sum(x) for x in zip(list1, list2)]
AMD_close = portfolio

###### Historic VaR and ES ########

##### abs return #######
A_hist_rtn = []
A_hist_rtn_sq = []

for i in range(1, len(data1)-5):
    hist_return = (AMD_close[i] - AMD_close[i+5])
```

```
    A_hist_rtn.append(hist_return)
    A_hist_rtn_sq.append(hist_return**2)



def abs_His_VaR_and_ES(years):
    S0 = 10000

    A_his_VaR_years = []
    A_his_ES_years = []

    for i in range(len(data1)-252*5):

        his_VaR_list = sorted(A_hist_rtn[i:i+252*5])
        his_VaR_years = (-S0) * his_VaR_list[13]/AMD_close[i]
        A_his_VaR_years.append(his_VaR_years)

        hist_ES_list = [(-S0)*x for x in his_VaR_list[1:32]]
        hist_ES_years = np.mean(hist_ES_list)
        A_his_ES_years .append(hist_ES_years)

    return(A_his_VaR_years, A_his_ES_years)

abs_A_his_VaR_years = abs_His_VaR_and_ES(5)[0][1:252*20]
abs_A_his_ES_years  = abs_His_VaR_and_ES(5)[1][1:252*20]

##### log return #######
A_hist_log_rtn = []
A_hist_log_rtn_sq = []

for i in range(1, len(data1)-5):
    hist_log_return = math.log( AMD_close[i]/AMD_close[i+5] )
    A_hist_log_rtn.append(hist_log_return)
    A_hist_log_rtn_sq.append(hist_log_return**2)


def His_VaR_and_ES(years):
    S0 = 10000

    A_his_VaR_years = []
    A_his_ES_years = []

    for i in range(len(data1)-252*5):

        his_VaR_list = sorted(A_hist_log_rtn[i:i+252*5])
        his_VaR_years = (-S0) * his_VaR_list[13]
        A_his_VaR_years.append(his_VaR_years)

        hist_ES_list = [(-S0)*x for x in his_VaR_list[1:32]]
        hist_ES_years = np.mean(hist_ES_list)
        A_his_ES_years .append(hist_ES_years)

    return(A_his_VaR_years, A_his_ES_years)
```

```
A_his_VaR_years = His_VaR_and_ES(5)[0][1:252*20]

##### formula VaR #########

A_log_rtn = []
A_log_rtn_sq = []

for i in range(1 ,len(data1)-1):
    log_return = math.log( AMD_close[i]/AMD_close[i+1] )
    A_log_rtn.append(log_return)
    A_log_rtn_sq.append(log_return**2)

def vol_and_mu(years):
    S0 = 10000
    T = 5/252
    p = 0.99

    A_vol_years = []
    A_mu_years = []
    A_VaR_years = []
    A_ES_years =[]

##### GBM Formula VaR and ES #######
    for i in range(len(data1)-252*years):
        vol_years = np.std(A_log_rtn[i:i+252*years]) * np.sqrt(252)
        mu_years = np.mean(A_log_rtn[i:i+252*years])*252 + (vol_years**2)/2
        VaR_years = S0 - S0 * np.exp( vol_years * T**(0.5)* ss.norm.ppf(1-p) + (mu_years - pow(
        ES_years = S0 * (1 - np.exp(mu_years *T)/(1-0.975) * ss.norm.cdf(ss.norm.ppf(1-0.975) -

        A_vol_years.append(vol_years)
        A_mu_years.append(mu_years)
        A_VaR_years.append(VaR_years)
        A_ES_years.append(ES_years)

    return(A_vol_years, A_mu_years, A_VaR_years, A_ES_years)


A_VaR_5years = vol_and_mu(5)[2][1:252*20]


timeline = data1[1:252*20,0]
timeline = [dt.datetime.strptime(d,'%Y-%m-%d').date() for d in timeline]

fig, ax = plt.subplots()

ax.plot(timeline, abs_A_his_VaR_years,'g-', label='Historical VaR (abs rtns)')
ax.plot(timeline, A_VaR_5years,'c-', label='formula VaR')
ax.plot(timeline, A_his_VaR_years,'r-', label='Historical VaR (log rtns)')


legend = ax.legend(loc='upper right', shadow=True)
ax.set_title('portfolio VaRs and ESs log VS absolute')
plt.show()
```

In [ ]: