

```

# -*- coding: utf-8 -*-
"""
Created on Wed Nov 25 14:57:15 2015

@author: Logan
"""

# Load packages
import pandas.io.data as web
import datetime
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from datetime import datetime
import matplotlib
matplotlib.style.use('ggplot')

#directory= '/'
directory = '/'

# Make a dataframe for announcement dates and company tickers
df = pd.read_csv(directory + 'Updated_Ratings.csv')

# make a list of tickers and dates separately
#dates = list(df['Date_Announced'])
#tickers = list(df['Ticker'])

# make a list of just dates and company tickers
keep_columns = ['TICKER', 'DATE_ANNOUNCED']
df2 = df[keep_columns]
df2.index = pd.to_datetime(df['DATE_ANNOUNCED'])

# make a list of just dates and company tickers
keep_columns = ['TICKER', 'DATE_ANNOUNCED']
df2 = df[keep_columns]
df2.index = pd.to_datetime(df['DATE_ANNOUNCED'])
df.index = pd.to_datetime(df['DATE_ANNOUNCED'])

#load file
crsp = pd.read_csv(directory + 'crsp_Month_80_14_SP.csv')

# Date reformatting
crsp['year'] = np.round(crsp.date/10000)
crsp['mth'] = [int(str(i)[4:6]) for i in crsp.date.values]
crsp['day'] = [int(str(i)[6:]) for i in crsp.date.values]
crsp['date2'] = [int(str(i)[:6]) for i in crsp.date.values] # to
merge w/ FF

```

```

# Get the realized excess return from the Fama French files MONTHLY
ff = pd.read_csv(directory + 'F-F_Research_Data_Factors.csv')
ff.set_index(ff['DATE'], inplace=True)
ff.drop('DATE', 1, inplace=1)
ff.columns = ['ExMkt', 'SMB', 'HML', 'RF'] # Rename the columns
ff = ff/100
ff['date2'] = ff.index
ff = ff[['date2', 'ExMkt', 'SMB', 'HML', 'RF']]

# Merge the Fama-French results to the CRSP file
crsp = pd.merge(crsp, ff, on='date2')
crsp.drop('date2', 1, inplace=1)

crsp['RET'] = np.where((crsp.RET == 'B') | (crsp.RET == 'C'), np.NaN,
crsp.RET)
crsp['RET'] = [float(c) for c in crsp.RET if c != np.NaN]
crsp['ExRet'] = crsp['RET'] - crsp['RF']
crsp['MktComparison'] = crsp['ExRet'] - crsp['ExMkt']

crsp = crsp[['PERMNO', 'TICKER', 'date', 'ExRet', 'RET', 'ExMkt',
'SMB', 'HML', 'RF', 'MktComparison']]
crsp.index = [datetime.strptime(str(i), '%Y%m%d') for i in
crsp.date.values]

crsp.to_csv(directory + 'Merged_FF.csv')

# make a dictionary for tickers
df_tickers = pd.DataFrame()

# make new dataframe with dates and tickers to use for CRSP
new_df = pd.DataFrame()
for rows in df2.iterrows():
    # print rows
    tick = rows[1]['TICKER']
    dt = rows[0].date()
    start = dt - pd.DateOffset(months=0)
    end = dt + pd.DateOffset(months=24)
    temp = crsp[crsp.TICKER == tick]
    temp = temp.sort()
    temp = temp[start:end]
    temp['event'] = np.arange(len(temp))
    new_df = pd.concat([new_df, temp])

#merge
result = pd.merge(new_df, df, how='outer', on=['TICKER'])
result.to_csv(directory + 'Merged_Data.csv')

#group by trials - 3 yr avg ret per firm
co_ret = result.groupby(['TICKER'])['RET'].mean()
co_ret.describe()

```

```

#group by gender
gender = result.groupby(['GENDER', 'event'])['RET'].mean()
gender.plot()

#Quartiles
#test= result.describe()

wow = result.quantile([.25, .5, .75, 1])[['ExRet', 'MktComparison']]
wow.to_csv(directory + 'quartiles.csv')

result['quartile'] = pd.qcut(result['RATING'], 4, labels=False)

#Returns against market
new2 = pd.DataFrame()
for q in range(4):
    temp = result[result['quartile'] == q][['event', 'MktComparison',
'TICKER']]
    temp = temp.pivot(index='event', columns='TICKER',
values='MktComparison')
    temp = temp.dropna(axis=1, how='all')
    temp = temp.dropna(axis=0, how='all')
    temp = temp.fillna(0)
    temp = np.log(1+temp)
    temp = temp.cumsum()
    temp = temp.mean(axis=1)
    temp = pd.DataFrame(temp).rename(columns={0: q})
    new2 = pd.concat([new2, temp], axis=1)

plt.figure()
new2.plot()
plt.title('Market Returns')
plt.ylabel('Percent')
plt.savefig(directory + 'market_returns.pdf')

#returns based on gender
new3 = pd.DataFrame()
for q in ['M', 'F']:
    temp = result[result['GENDER'] == q][['event', 'RET', 'TICKER']]
    temp = temp.pivot(index='event', columns='TICKER', values='RET')
    temp = temp.dropna(axis=1, how='all')
    temp = temp.dropna(axis=0, how='all')
    temp = temp.fillna(0)
    temp = np.log(1+temp)
    temp = temp.cumsum()
    temp = temp.mean(axis=1)
    temp = pd.DataFrame(temp).rename(columns={0: q})
    new3 = pd.concat([new3, temp], axis=1)
plt.figure()

```

```

new3.plot()
plt.title('Gender Returns')
plt.ylabel('Percent')
plt.savefig(directory + 'gender_returns.pdf')

#returns of quartiles
new4 = pd.DataFrame()
for q in range(4):
    temp = result[result['quartile'] == q][['event', 'ExRet',
'TICKER']]
    temp = temp.pivot(index='event', columns='TICKER', values='ExRet')
    temp = temp.dropna(axis=1, how='all')
    temp = temp.dropna(axis=0, how='all')
    temp = temp.fillna(0)
    temp = np.log(1+temp)
    temp = temp.cumsum()
    temp = temp.mean(axis=1)
    temp = pd.DataFrame(temp).rename(columns={0: q})
    new4 = pd.concat([new4, temp], axis=1)

plt.figure()
new4.plot()
plt.title('ExRet Returns')
plt.ylabel('Percent')
plt.savefig(directory + 'exret.pdf')

```

```

# OPTION 1
new = result.groupby(['quartile', 'event'])['RET'].mean()

new = new.reset_index()
new = new.pivot(index='event', columns='quartile', values='RET')

new = np.log(1+new)
new = new.cumsum()

```

```

#get risk free rate
#subtract risk free from returns, plot
#then subtract market returns, plot
#merge on permos

```

