Example

For the most convenient testing experience use a Jupyter Notebook

Included in the python ZCAPM package and ZCAPM github repository is data used for testing the model. The data is as follows

1. ff_factors.csv is a file containing the Fama French factors
2. ff25_day.csv is a file containing returns for 25 size - Book-to-Market sorter portfolios
3. ind47.csv is a file containing returns for 47 industry portfolios
4. mu_sigma.csv contains returns for the equal weight market return and market sigma as discussed in the ZCAPM book

The methods included in the Testing class of the ZCAPM package are as follows

- estLinearModel(), _rollapplyLM(), and _LMRegression() are all used for constructing and estimating time series factor loadings for linear factor models such as the Fama French 3 factor model

- estZCAPM(), _rollapplyEM(), _EMRegression(), and _EM_loop() are all methods used for estimating time series factor loadings for our proposed ZCAPM model with the Expectation Maxization Algorithm

- FamaMacBeth() is used for running the Fama-MacBeth test

for more information on each method use `__doc__`

Import ZCAPM Package

In [1]:

```
import ZCAPM
```

Create class instance

In [2]:

```
#Use test = Testing(False) if you do not want progress updates while the code is running
test = ZCAPM.Testing()
```

Load and prepare data for testing

In [3]:

```
#Set Parameters for rolling estimations and EM Algorithm. For information on parameters see doc string of _EM_loop met
tol = .001
MaxIter = 1000
criterion = 1
width = 12

######## IMPORTANT########
#trims off all of the first "width" amount of monthly returns to ensure that the Fama-MacBeth test is performed
```

```
#OUT OF SAMPLE
monthly_excess_return = test.sorted_portfolio_monthly_excess_return.iloc[width:,:]
portfolio_excess_return = test.sorted_portfolio_daily_excess_return
factor_return = test.factor_daily_return

YearMonth = monthly_excess_return.index

#create pandas series for mkt ret, mkt sigma, and factors. Convert indices of these series and portfolio return datafr
#to be the YearMonth list. Useful for indexing purposes while testing
mu = (factor_return.loc[:,"R_a.R_f"])
sigma = (factor_return.loc[:,"sigma_a"])
facs_ret = factor_return.loc[:,['YearMonth','R_a.R_f','SMB','HML']]

mu.index = factor_return.YearMonth
sigma.index = factor_return.YearMonth
portfolio_excess_return.set_index('YearMonth',inplace = True)
facs_ret.set_index('YearMonth',inplace =True)
```

Time series estimations of ZCAPM and linear factor models

In [4]:

```
#calculates time series factor loadings for each portfolio. See Testing class for information on the arguments of each
#method
zcapm_results = test.estZCAPM(portfolio_excess_return,mu,sigma,tol,MaxIter,criterion,width)
ff3_results = test.estLinearModel(portfolio_excess_return,facs_ret,width)
capm_results = test.estLinearModel(portfolio_excess_return,facs_ret.loc[:,['R_a.R_f']],width)

#adjust the zeta estimates for each portfolio to monthly estimates
zeta_cols = zcapm_results.columns[zcapm_results.columns.str.contains('zeta')]
zcapm_results.loc[:,zeta_cols] = zcapm_results.loc[:,zeta_cols]*21
```

```
Fitting linear model for BIG.HiBMBM
```

Out of sample cross-sectional Fama MacBeth test

In [5]:

```
#Runs the Fama-MacBeth Test for each portfolio
ZCAPM = test.FamaMacBeth(monthly_excess_return,zcapm_results,['beta','zeta'],'ZCAPM')
FF3 = test.FamaMacBeth(monthly_excess_return,ff3_results,['R_a.R_f','SMB','HML'],'Fama-French 3 Factor')
CAPM = test.FamaMacBeth(monthly_excess_return,capm_results,['R_a.R_f',],'CAPM')
```

In [6]:

```
ZCAPM
```

Out[6]:

	coefficients	t-values
ZCAPM		
intercept	0.7593262880472027	3.0875681939793744
beta	-0.17833179614591696	-0.7265999715044811
zeta	0.4885823411522055	4.299951332644644
Single Regression Approach R-squared	0.9690609647879033	

In [7]:

```
FF3
```

Out[7]:

	coefficients	t-values
Fama-French 3 Factor		
intercept	0.8899926349195385	4.625249967909042
R_a.R_f	-0.3742676286214678	-1.7778283203036678
SMB	0.18226260438326772	1.3752332078248815
HML	0.3025136631944702	2.544038141241701
Single Regression Approach R-squared	0.6470600399973218	

In [8]:

```
CAPM
```

Out[8]:

	coefficients	t-values
CAPM		
intercept	0.9215175378932576	3.7385395290180075
R_a.R_f	-0.29867579468565364	-1.1877174962510977
Single Regression Approach R-squared	0.5198257918234421	