TP4

In [1]:
```
# Load packages
import pandas as pd
import numpy as np
from statistics import *
from scipy.stats import *
from math import *

# Get data from excel
# SP100 & SP500 price index
spindex = pd.read_excel("/files/exercises/04/TP4.xls", sheet_name = 'S&P100Index', skiprows = 4)
spindex.columns = ['date','sp100','sp500']
spindex = spindex.set_index('date')

# SP100 constitutes
sp100_const = pd.read_excel("/files/exercises/04/TP4.xls", sheet_name = 'S&P100Constituents', skiprows = 4)
# Rename columns
const_names = sp100_const.columns[1:].tolist()
const_names = [st.replace("(P)", "") for st in const_names]
sp100_const.columns = ['date'] + const_names

sp100_const = sp100_const.set_index('date')

# T-Bill 3-month rates
tbill_3m = pd.read_excel("/files/exercises/04/TP4.xls", sheet_name = 'TBill3Months', skiprows = 4)
tbill_3m.columns = ['date','tbill']
tbill_3m = tbill_3m.set_index('date')
```

Test of the CAPM: Time-Series Approach

Question 1:

The standard Sharpe-Lintner CAPM assumes the existence of a constant risk-free rate and is written as:

$$E[r_i] = r_f + \beta_i(E[r_m] - r_f).$$

For each stock, estimate the betas on the 12/11/1992 - 15/8/2001 period, using the S&P 500 as a market proxy.

Issue: how can you form the expectations?

In [2]:
```
print('=====Problem 1: Time-Series Approach=====\n')

# Calculation of the arithmetic returns
RA_SP100Constituents = (sp100_const - sp100_const.shift(1))/sp100_const.shift(1)
RA_SP100Constituents = RA_SP100Constituents.dropna()

RA_SP500 = (spindex - spindex.shift(1))/spindex.shift(1)
RA_SP500 = RA_SP500.dropna()
RA_SP500 = RA_SP500['sp500']

# The TBill returns are given annualized, to get return per week we divide by 100 and by 52 because there are 52 weeks in a year
tbill = tbill_3m[1:]/5200
```

```
=====Problem 1: Time-Series Approach=====
```

In [3]:
```
nb_titres = sp100_const.shape[1]
```

In [4]:
```
# Create dataframe for y and x
x = pd.concat([RA_SP500, tbill], axis = 1)
x['rm_rf'] = x['sp500'] - x['tbill']

y = RA_SP100Constituents.sub(tbill['tbill'], axis = 0)
```

In [5]:
```
import statsmodels.api as sm
# Remember, in this excersise we go until 16/8/2001, so we need to filter the date range
x_ts = x[x.index <= '2001-08-16']
y_ts = y[y.index <= '2001-08-16']

betas = np.zeros(nb_titres)
```

```
x['rm_rf'] = x['sp500'] - x['tbill']

y = RA_SP100Constituents.sub(tbill['tbill'], axis = 0)
```

In [5]:
```
import statsmodels.api as sm
# Remember, in this excersise we go until 16/8/2001, so we need to filter the date range
x_ts = x[x.index <= '2001-08-16']
y_ts = y[y.index <= '2001-08-16']

betas = np.zeros(nb_titres)

x_ts = sm.add_constant(x_ts['rm_rf'])

for i in range(0,nb_titres):
    reg = sm.OLS(endog = y_ts.iloc[:,i], exog = x_ts)
    results = reg.fit()
    betas[i] = results.params['rm_rf']
```

Question 2:

A first method to test the CAPM is based on the following time-series regression:

$$z_{i,t} = \alpha_i + \beta_i z_{m,t} + \epsilon_{i,t},$$

where $z_{i,t} = r_{i,t} - r_{f,t}$, and $\epsilon_{i,t}$ is a zero mean noise. The null hypothesis $H_0 : \alpha_i = 0$ can then be tested for each asset separately using standard t-statistics.

Using the remaining data, perform this naive test of the CAPM and check how well the market explains individual stock returns.

Solution approach: We test the CAPM stock by stock. Multivariate tests allow you to test the CAPM on all stocks at once.

For this first test, it is as well crucial to suppose that the underlying distribution doesn't vary during time (returns iid), and that we use a good proxy for the market portfolio.

In [6]:
```
# Initialize the variables (makes the execution of the program more efficient)
alphas = np.zeros(nb_titres)
betas_p2 = np.zeros(nb_titres)
testsalphas = np.zeros(nb_titres)
```

In [7]:
```
for i in range(0,nb_titres):
    reg = sm.OLS(endog = y_ts.iloc[:,i], exog = x_ts)
    results = reg.fit()
    alphas[i] = results.params['const']
    betas_p2[i] = results.params['rm_rf']
    alpha_ci = results.conf_int(alpha=0.05, cols=None)


    if alpha_ci.iloc[0,0]*alpha_ci.iloc[0,1] > 0:
        # To test at 95# level, we look if 0 belongs to the confidence interval or not.
        # 0 doesn't belong to the confidence interval, so the CAPM model is not valid.
        testsalphas[i] = 0
    else:
        testsalphas[i] = 1
```

In [8]:
```
# you can then for example look at the number of times in which the CAPM is accepted/rejected :
nb_valides = sum(testsalphas)
nb_rejets = -sum(testsalphas-1)
```

Remarks:

- the CAPM doesn't include any temporal evolution of the model (only one period). Are the betas really constant through time?
- We assumed that the SP500 is representative of the market return

Fama MacBeth Procedure

Question 1:

In [9]:
```
print('----- Problem 2: Fama MacBeth Procedure -----')

beta_fm = np.zeros(nb_titres)
y_fm_1 = y[(y.index >= '1992-11-12') & (y.index <= '1996-11-14')]
x_fm_1 = x[(x.index >= '1992-11-12') & (x.index <= '1996-11-14')]
x_fm_1 = sm.add_constant(x_fm_1['rm_rf'])
```