In [1]:

```
import matplotlib
import matplotlib.pyplot as plt
matplotlib.style.use('ggplot')

from IPython.core.display import display, HTML
display(HTML("<style>.container { width:100% !important; }</style>"))

import numpy as np
import pandas as pd
```

Leveraged Portfolios

https://en.wikipedia.org/wiki/130%E2%80%9330_fund

Thomas Schmelzer

A 130/30 Equity Portfolio

- Allocate capital $C = 1$. Sell short at most $c = 0.3$ to finance a long position of $1 + c$.
- Universe of n assets.

$$
\begin{aligned}
\mathbf{x}^* = \arg\max_{\mathbf{x}\in\mathbb{R}^n} \mu^T \mathbf{x} \\
\text{s.t. } \Sigma\, x_i = 1 \\
\Sigma\, |x_i| \leq 1 + 2c \\
\sqrt{\mathbf{x}^T \mathbf{C}\mathbf{x}} \leq \sigma_{\max}
\end{aligned}
$$

In [2]:

```python
from cvx.util import cvx, maximize

# make some random data, e.g. cov-matrix and expected returns
n = 100
c = 0.9
C = c * np.ones((n, n)) + (1 - c) * np.eye(n)
mu = 0.05 * np.sin(range(0, n))
# maximal volatility and leverage...
sigma_max = 1.0
excess = 0.3

x = cvx.Variable(n)
constraints = [cvx.sum(x)==1, cvx.norm(x,1)<=1+2*excess, cvx.quad_form(x,C)<=sigma_ma
maximize(objective=x.T*mu, constraints=constraints)
f = x.value

print("Sum of positive weights: {0}".format(np.sum(f[f > 0])))
print("Sum of negative weights: {0}".format(np.sum(f[f < 0])))
print("Sum of all weights:      {0}".format(np.sum(f)))
```

```
Sum of positive weights: 1.2999999974815202
Sum of negative weights: -0.29999999748158745
Sum of all weights:      0.9999999999999326
```

Summary

- Leverage is here a constraint for the 1-norm of the weight vector.
- Note that we do not solve two problems for the short and long part of the portfolio.