CPPI

March 16, 2025

1 Implementing Portofolio Insurance (CPPI) and Drawdown Constraints

[28]:

```
%load_ext autoreload
%autoreload 2
%matplotlib inline

import edhec_risk_kit_129 as erk
import pandas as pd
import numpy as np

# load the industry returns and the total market index that we previoulsy create
ind_return = erk.get_ind_returns ()
tmi_return = erk.get_total_market_index_returns ()
```

```
The autoreload extension is already loaded. To reload it, use:
  %reload_ext autoreload
```

[29]:

```
risky_r = ind_return ["2000":][["Steel","Fin","Beer"]]
# Safe asset, same shape as risky asset
safe_r = pd.DataFrame().reindex_like(risky_r)
```

[30]:

```
risky_r.shape
```

[30]:

```
(228, 3)
```

[31]:

```
safe_r.shape
```

[31]:

```
(228, 3)
```

[32]:

```
safe_r[:]= 0.03/12
start = 1000
floor = 0.8
```

2 1 . Cushion - (Asset value - Floor Value)

3 2 . Compute an Allocation to Safe and Risky Assets –> m * risk budget

4 3 . Recompute the Asset Value based on the returns

[33]:

```
def compound (r):
    return (1+r).prod()-1

def compound_2 (r):
    return np.expm1(np.log1p(r).sum())
```

[34]:

```
compound_2 (risky_r)
```

[34]:

```
Steel   -0.051696
Fin      1.773937
Beer     3.361349
dtype: float64
```

5 Back to CPPI

[84]:

```
dates = risky_r.index
n_steps = len (dates)
account_value = start
floor_value = start*floor
m = 3
account_history = pd.DataFrame().reindex_like(risky_r)
cushion_history = pd.DataFrame().reindex_like(risky_r)
risky_w_history = pd.DataFrame().reindex_like(risky_r)

for step in range(n_steps):
    cushion = (account_value - floor_value)/account_value
    risky_w = m*cushion
    risky_w = np.minimum(risky_w, 1)
    risky_w = np.maximum(risky_w, 0)
    safe_w = 1-risky_w
    risky_alloc = account_value*risky_w
    safe_alloc = account_value*safe_w
    # recompute the new account value at the end of this step
    account_value = risky_alloc*(1+risky_r.iloc[step]) + safe_alloc*(1+safe_r.
 ↪iloc[step])
    # save the histories for analysis and plotting
    cushion_history.iloc[step] = cushion
    risky_w_history.iloc[step] = risky_w
    account_history.iloc[step] = account_value
    risky_wealth = start*(1+risky_r).cumprod()
```

[85]:

```
account_history.head()
```

[85]:

```
                Steel          Fin        Beer
2000-01    984.380000   974.480000  987.320000
2000-02   1023.292876   931.167544  922.971256
2000-03   1047.555176   998.187296  924.835988
2000-04   1042.079009   973.927479  939.993701
2000-05   1007.137753  1001.460033  991.145489
```

[86]:

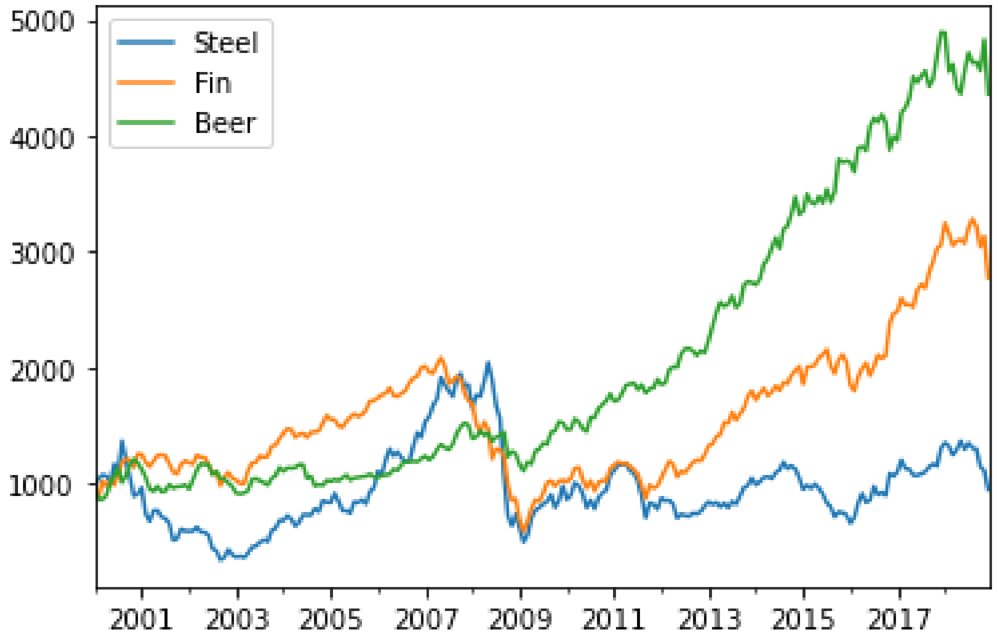
```
risky_wealth = start* (1+risky_r).cumprod ()
```

[87]:

```
risky_wealth.plot ()
```

[87]:

```
<matplotlib.axes._subplots.AxesSubplot at 0x70d9ae813e50>
```

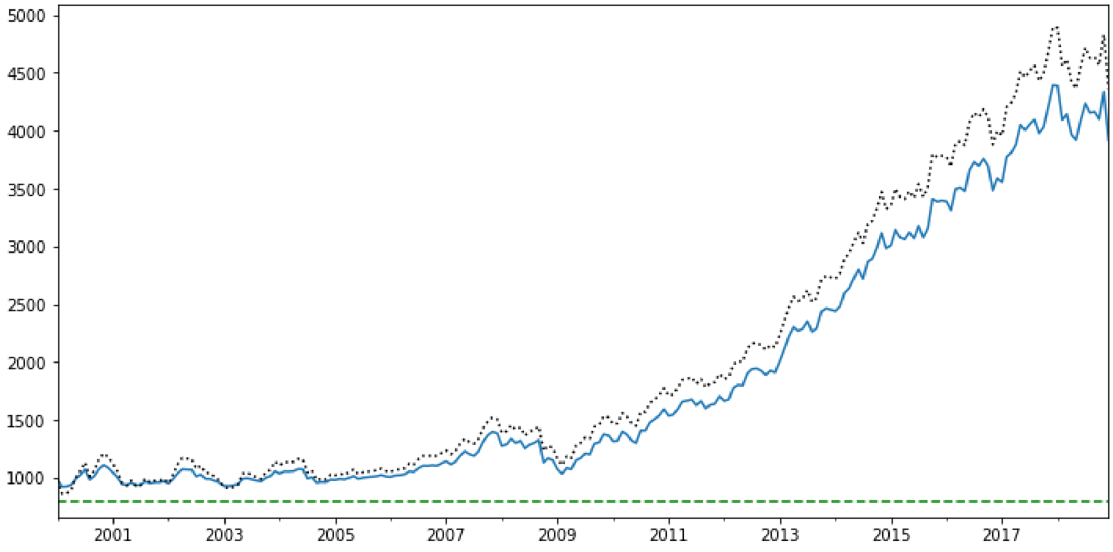


[95]:

```
ind = "Beer"
ax = account_history [ind].plot(figsize=(12,6))
risky_wealth [ind].plot(style = "k:")
ax.axhline (y=floor_value, color='green', linestyle = "--")
```

[95]:

```
<matplotlib.lines.Line2D at 0x70d9afa55790>
```

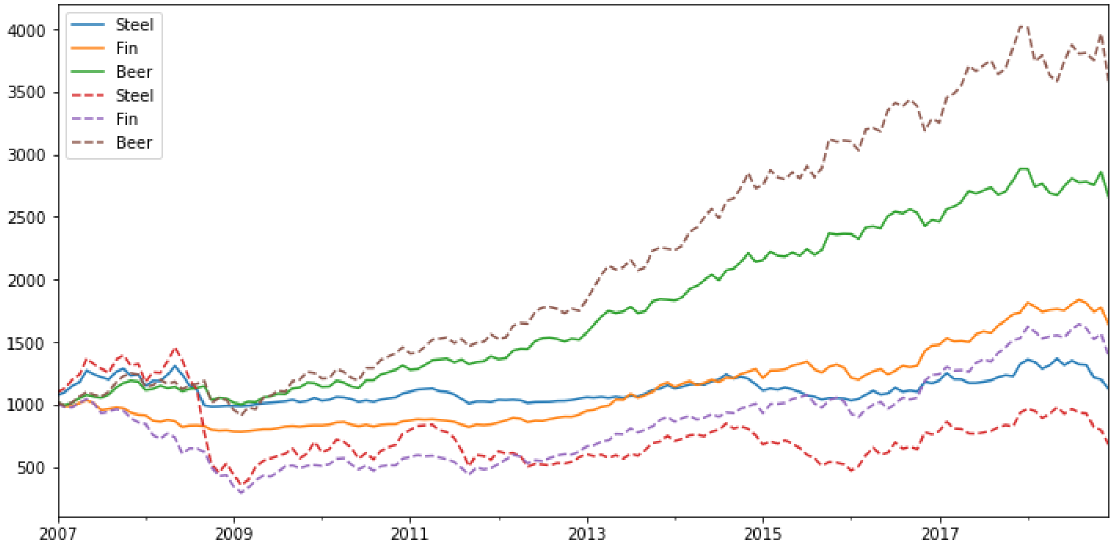


6 Drawdown Constraint

```
[99]: btr = erk.run_cppi(ind_return["2007":][["Steel","Fin","Beer"]], drawdown = 0.25)
```

```
[101]: ax = btr ["Wealth"].plot(figsize=(12,6))
       btr ["Risky Wealth"].plot(ax=ax, style = "--")
```

```
[101]: <matplotlib.axes._subplots.AxesSubplot at 0x70d9aea4ab50>
```



[ ]: