SIGNALS

UNIVERSE = TOP 500

market\_data = pd.read\_parquet('data/US/mkt\_data.pq')

balance\_sheet = pd.read\_parquet('data/US/balance\_sheets.pq')

GICS = pd.read\_parquet('data/US/GICS.pq')

P = pd.read\_parquet('data/US/universe\_table.pq')

R = market\_data['close'].unstack().reindex\_like(P).pct\_change()

income\_statement = pd.read\_parquet('data/US/income\_statement.pq')

mkt\_cap = market\_data['MktCap'].unstack().reindex\_like(P).astype(float)

ltdebt = balance\_sheet['longTermDebt'].unstack().reindex\_like(P).ffill().astype(float)

stdebt = balance\_sheet['shortTermDebt'].unstack().reindex\_like(P).ffill().astype(float)

cash\_equiv = balance\_sheet['cashAndEquivalents'].unstack().reindex\_like(P).ffill().astype(float)

EV = mkt\_cap + (ltdebt+stdebt)  - cash\_equiv.fillna(0)

ebitda = income\_statement['ebitda'].unstack().reindex\_like(P).ffill().astype(float)

EM = EV / ebitda

EM = EM.sub(EM.mean(1),0).div(EM.std(1),0)

signal = EM[P].rank(1,pct=True,ascending=False).clip(0.01,0.99).apply(norm.ppf).ffill()

signal = signal[P].rolling(252).rank(axis=0,pct=True,ascending=True).clip(0.01,0.99)

signal = signal[P].rank(axis=1,pct=True,ascending=True).clip(0.01,0.99).apply(norm.ppf)

signal[signal.abs()<0.2] = 0

signal = signal[P].groupby(GICS['gicgrp'],axis=1).apply(center)

signal = signal[P].div(signal.abs().sum(1),0)

pnl = (signal.shift()\*R).sum(1)

sr = sharpe(pnl)

pnl.cumsum().plot()

pd.Series({'Sharpe':sr})

