Table of Contents

notes of data

'FF3': monthly Fama-French five factors, 1926/07-2018/12 order of variables: 'rmrf','smb','hml','rf' 'FF5': monthly Fama-French five factors, 1967/07-2018/12 order of variables: 'rmrf','smb','hml','rmw','cma','rf' 'ind_value_daily': daily value-weighted returns of 48 industry portfolios 'ind_value': monthly value-weighted returns of 48 industry portfolios 'NBER_rec': NBER recession indicator in the same sample period

```
addpath('./functions');
```

part 2 excess return of volatility-managed portfolios

2.1 realized variance of each month

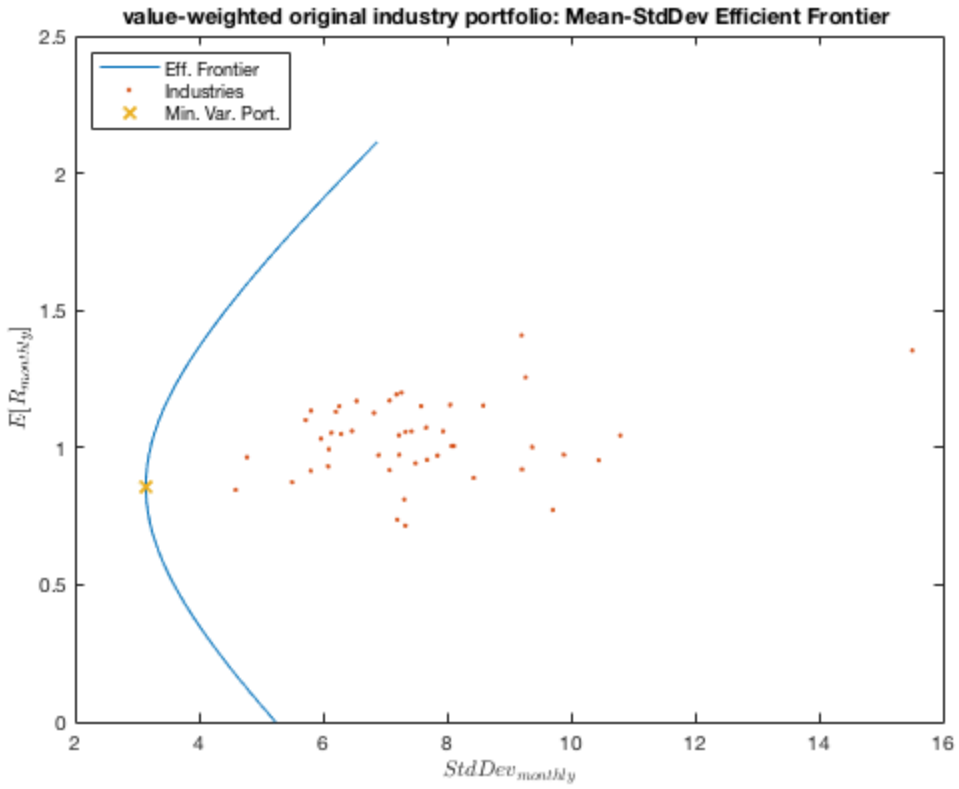
```
RV_value = getRV(ind_value_daily,date);
% 2.2 volatility-managed excessreturn
VMret_value = getVMret(ind_value,RV_value,FF3(:,4));
```

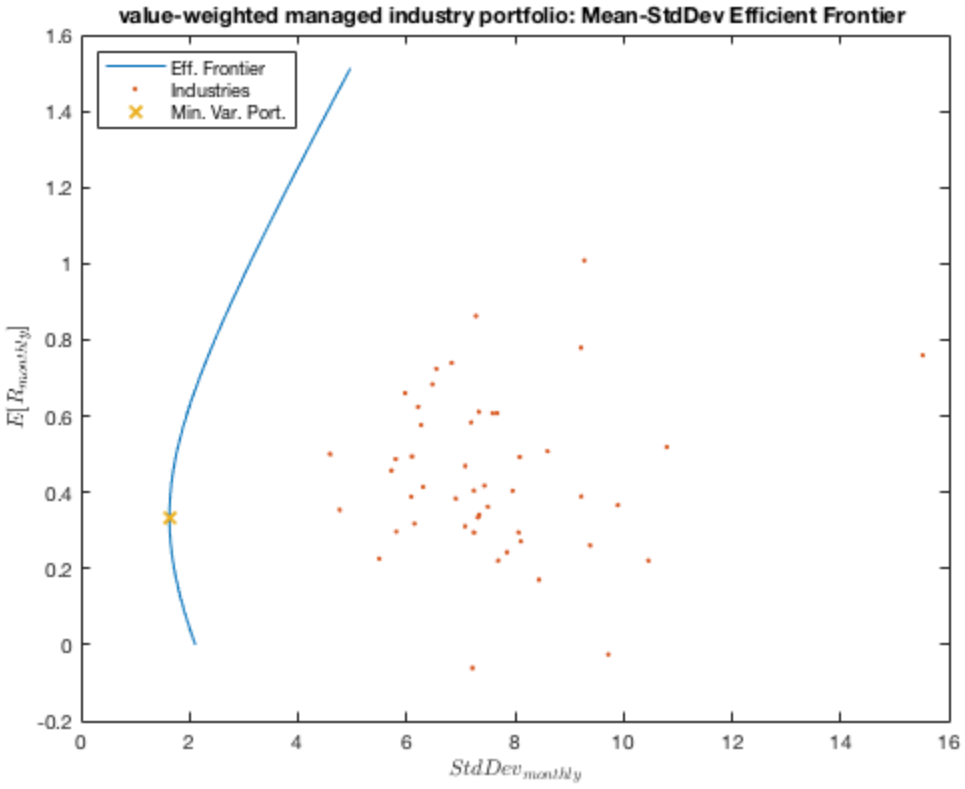
part 3 comparison between managed & original portfolios

3.1 CAPM, FF3 and FF5 pricing on original portfolio

```
FF1_ind_original = pricing([FF3(:,1),FF3(:,4)],ind_value,1,0);
FF3_ind_original = pricing(FF3,ind_value,3,0);
FF5_ind_original = pricing(FF5,ind_value,5,0);
% 3.2 CAPM, FF3 and FF5 pricing on managed portfolio
FF1_ind_managed =
 pricing([FF3(:,1),FF3(:,4)],VMret_value.port_normalize,1,0);
FF3_ind_managed = pricing(FF3,VMret_value.port_normalize,3,0);
FF5_ind_managed = pricing(FF5,VMret_value.port_normalize,5,0);
% 3.3 excess Sharpe ratio and utility gain
Sharpe_value =
 getSharpe(ind_value,VMret_value.port_normalize,mean(FF3(:,4)));
% 3.4 return difference during NBER defined recession
DiffRec_value =
 getDiffRec(ind_value,VMret_value.port_normalize,NBER_rec);
```

```
% 3.5 efficient frontier of original/managed portfolios
EF_value_original = mvp(nanmean(ind_value), ...
    cov(ind_value,'partialrows'), mean(FF3(:,4)), 1,...
    'value-weighted original industry
 portfolio', 'EF_value_original');
EF_value_managed = mvp(nanmean(VMret_value.port_normalize), ...
    cov(VMret_value.port_normalize,'partialrows'), mean(FF3(:,4)),
 1,...
    'value-weighted managed industry portfolio', 'EF_value_managed');
```





part 4 risk-parity portfolio

4.1 inverse of volatility

```
volinv_value = getVolInv(ind_value,FF3(:,4));
% 4.2 construct risk parity portfolio (use original portfolio)
retRP1_value = getRP(ind_value,FF3(:,4),volinv_value,1,FF3(:,1)); %
 unlever
retRP2_value = getRP(ind_value,FF3(:,4),volinv_value,2,FF3(:,1)); %
 lever
% 4.3 inverse of three-year rolling volatility (use managed portfolio)
volinv_valueVM = getVolInv(ind_value,FF3(:,4));
% 4.4 construct risk parity portfolio (use managed portfolio)
retRP1_valueVM = getRP(VMret_value.port_normalize,FF3(:,4),...
    volinv_valueVM,1,FF3(:,1)); % unlever
retRP2_valueVM = getRP(VMret_value.port_normalize,FF3(:,4),...
    volinv_valueVM,2,FF3(:,1)); % lever
```

4.5 CAPM, FF3 and FF5 pricing on risk-parity using original returns

levered risk parity portfolios

```
FF1_rp1_value_original =
 pricing([FF3(:,1),FF3(:,4)],retRP1_value.rpRet_unlever,1,0);
```

```
FF3_rp1_value_original = pricing(FF3,retRP1_value.rpRet_unlever,3,0);
FF5_rp1_value_original = pricing(FF5,retRP1_value.rpRet_unlever,5,0);
% unlevered risk parity portfolios
FF1_rp2_value_original =
 pricing([FF3(:,1),FF3(:,4)],retRP2_value.rpRet_lever,1,0);
FF3_rp2_value_original = pricing(FF3,retRP2_value.rpRet_lever,3,0);
FF5_rp2_value_original = pricing(FF5,retRP2_value.rpRet_lever,5,0);
% 4.6 CAPM, FF3 and FF5 pricing on risk-parity using original returns
% levered risk parity portfolios
FF1_rp1_value_managed =
 pricing([FF3(:,1),FF3(:,4)],retRP1_valueVM.rpRet_unlever,1,0);
FF3_rp1_value_managed = pricing(FF3,retRP1_valueVM.rpRet_unlever,3,0);
FF5_rp1_value_managed = pricing(FF5,retRP1_valueVM.rpRet_unlever,5,0);
% unlevered risk parity portfolios
FF1_rp2_value_managed =
 pricing([FF3(:,1),FF3(:,4)],retRP2_valueVM.rpRet_lever,1,0);
FF3_rp2_value_managed = pricing(FF3,retRP2_valueVM.rpRet_lever,3,0);
FF5_rp2_value_managed = pricing(FF5,retRP2_valueVM.rpRet_lever,5,0);
```

Published with MATLAB® R2018b