Factor Analyzer User Guide

Version control

Version	Date	Note	Author
1.0	2012-06-13		Louis Luo

Table of Contents

1 Introduction

What is it?

Factor Analyzer is an integrated tailor-made tool for EIS factor model process. It can be used to analyze individual factors and assist in factor selection and performance monitoring.

How is it implemented?

Factor Analyzer is implemented in matlab as classes, and integrated with database for data retrieval. It is built on top of EIS myfints(xts) toolbox.

What can it do?

Its main functionality includes:

- Load relevant data for factor analysis from DB and store them in an object
- Calculate statistics and generate user-customized PDF report which displays:
  - Basic descriptive statistics: coverage, autocorrelation, median, and quintile.
  - Rolling IC, average IC, IR of IC when neutralized to different dimensions
  - Cumulative factor portfolio return when neutralized to different dimensions
  - Factor performance grouped by sectors or countries
  - Factor performance in different volatility regimes
  - Return contribution and factor score distribution grouped by MCAP
  - Return contribution and factor score distribution grouped by ADV
  - Return contribution and factor score distribution grouped by borrowing cost
- Generate statistics summary in excel file
- Fit cross-sectional univariate / multivariate regression for factors with the choice of standard and/or customized risk factors, and build multi-factor model using multivariate regression approach
- Generate PDF report and excel file from regression analysis

How flexible is it?

It employs a flexible design so that users can have a wide range of customization to make, such as the choice of a specific subset of the universe to perform analysis, the choice of neutralization style, the choice of subplots on the report, the choice of risk factors to use, etc.

Where can I find it?

The TFS path of the code is:

$/QuantStrategy/Analytics/Utility/FactorAnalytics/

$/QuantStrategy/Analytics/Utility/FactorAnalytics/@FactorAnalyzer/

2 Quick-start Demo

This session presents a quick-start demo for the users of the factor analyzer.

User should launch Matlab, add the path of **$/QuantStrategy/Analytics/Utility/FactorAnalytics/**, and then follow the steps below to try out the functions:

Step 1) Set parameters used to create FactorAnalyzer object

```
% parameter for creating object
aggid = '0064990100'; % the aggid of the index - define the universe
startdate = '2000-01-31'; % start date of data
enddate = '2012-04-30'; % end date of data
isprod = 0; % 1 for getting data from production (live data), 0 for getting data from development (backtest data)
freq = 'M'; % frequency of data
dateParam = {'BusDays',0}; % extra date parameters to control the dates of time series
nbucket = 5; % number of neutralization bucket used for styles such as b2p, beta, and mcap
univname = 'MSWO'; % name of universe, arbitraryly decided by users
ctrylist = {}; % a country list that defines a subset from the universe
facinfo.name = {'F00001','F00005','F00018'}; % set the factorid that will be included in the object
```

Step 2) Create FactorAnalyzer object

```
% construct the object
o = FactorAnalyzer(facinfo,aggid,startdate,enddate,isprod,freq,ctrylist,nbucket,dateParam,univname);
```

Step 3) Generate factor report with default settings, report will be generated in current directory

```
% calculate statistics and get report with default settings
o = CalcStatistics(o,startdate,enddate);
GetSummary(o);
```

Step 4) Report customization

```
% basic customization
gicslevel = 2; % GICS level 2 neutral
neutralstyle = {'gics','ctry','ctrysect'}; % customize neutralization style
facOrAlpha = {'F00001','F00018'}; % pick a subset of factors to run
custplot = {'ByCtry', 'LongShort', 'ScoreByMcap', 'RtnByMcap', 'ScoreByLiq', 'RtnByLiq'}; % choose plots on the
report, starting from the 5th plots
Savepath = 'Y:\Louis.Luo\'; % save path of the report
summaryfile = 'Summary001'; % name of the summary excel file


% advanced customization
secids = fieldnames(o.bmhd,1); % get security list
CurCtryGICS = LoadSecInfo(secids,'Country,IsoCurId,SubIndustId','','',0); % retrieve current GICS, country, and
currency info
Ctry = CurCtryGICS.Country;
Cur = CurCtryGICS.IsoCurId;
GICS = CurCtryGICS.SubIndustId;

% 1. user customized universe - financial sector of the original benchmark
custuniv.name = {'MSWOFIN'};
custuniv.data = o.bmhd(:,floor(GICS/10^6) == 40);

% 2. user customized neutralization style - currency neutral
uniqCur = unique(Cur);
currency = o.bmhd;
```

```
currency(:,:) = NaN;
for i = 1:numel(uniqCur)
    ccyidx = ismember(Cur, uniqCur{i});
    currency(:,ccyidx) = i;
end

% 3. user customized neutralization style - currency neutral
regionctrymap.region =
{'USA','EUR','EUR','EUR','EUR','EUR','EUR','EUR','EUR','EUR','EUR','JAP','RES','RES','RES'};
regionctrymap.ctry  =
{'USA','CHE','GBR','DEU','FRA','DNK','IRL','SWE','BEL','ESP','FIN','JPN','AUS','ISR','CAN'};
region = o.bmhd;
region(:,:) = NaN;
uniqRegion = unique(regionctrymap.region);
for i = 1:numel(uniqRegion)
    regionidx = ismember(Ctry, regionctrymap.ctry(ismember(regionctrymap.region, uniqRegion(i)))); % find the
index of region i based on the region ctry mapping and the country id of benchmark
    region(:,regionidx) = i;
end

% create customized style parameter
custstyle.name = {'currency', 'region'};
custstyle.data = {currency, region};
```

Step 5) Regenerate factor report with customizations

```
[o, o_new] =
CalcStatistics(o,startdate,enddate,'gicslevel',1,'facOrAlpha',facOrAlpha,'neutralstyle',neutralstyle ...
    ,'custstyle',custstyle,'custuniv',custuniv,'custplot',custplot,'savepath',Savepath); % o_new will only
contain the stocks from customized universe
GetSummary(o, facOrAlpha, Savepath, summaryfile);
```

Step 6) Construct alpha weight selected factors and examine its long-short performance

```
selectfaclist = {'F00001','F00018'};
result = AnalyzeAlpha(o, 'faclist', selectfaclist, 'normdata', o.gics, 'normlevel', 1, 'wgtmethod', 'EW');
```

Step 7) Load EM risk model (optional)

EM risk model can be used in regression analysis when you want to neutralize some of the risk factors when examining the performance of alpha factors. The data loading takes a long time, you don't have to load the risk factor if you set the 'riskneutral' parameter to be 0 when calling RegressionAnalysis.

```
o = LoadRiskModel(o);
```

Step 8) Run regression analysis, and generate PDF / excel report.

```
% selected factor list to perform regression analysis
selectfaclist = {'F00001','F00018'};
% individual alpha factor + sector dummy
[o, stat, model] = RegressionAnalysis(o, 'faclist', selectfaclist, 'reportname', 'Regression001','sectordummy',
1, 'riskneutral', 0);
% individual alpha factor + sector dummy + 5 EM risk factor
[o, stat, model] = RegressionAnalysis(o, 'faclist', selectfaclist, 'reportname', 'Regression002','sectordummy',
1, 'riskneutral', 1, 'numriskfac', 5);

% create currency dummy variables
CurDummy = cell(1, numel(uniqCur));
```

```
for i = 1:numel(uniqCur)
    CurDummy{i} = o.bmhd;
    CurDummy{i}(:,:) = 0;
    CurDummy{i}(:,ismember(Cur, uniqCur{i})) = 1;
    CurDummy{i}(isnan(o.bmhd)) = 0;
end

% individual alpha factor + sector dummy + 5 EM risk factor + customized risk factor
[o, stat, model] = RegressionAnalysis(o, 'faclist', selectfaclist, 'reportname', 'Regression003','sectordummy',
1, 'riskneutral', 1, 'numriskfac', 5, 'otherriskfac', CurDummy);

% multiple alpha factors + sector dummy + 5 EM risk factor + customized risk factor
[o, stat, model] = RegressionAnalysis(o, 'faclist', selectfaclist, 'reportname', 'Regression004','sectordummy',
1, 'riskneutral', 1, 'numriskfac', 5, 'otherriskfac', CurDummy, 'buildmodel', 1);
```

3 Function List

Functions available to users are listed in this session.

FactorAnalyzer

- Command

```
o = FactorAnalyzer(facinfo, aggid, startdate, enddate, isprod, freq, ctrylist, nbucket, dateParam, univname)
```

- Description
  - Constructor of the factor analyzer object
- Input
  - facinfo: structure with mandatory field ‘name’, and optional field ‘ishigh’
  - aggid: string for universe (index) id
  - startdate: string for start date of data
  - enddate: string for end date of data
  - isprod: 1 for production (live) mode, 0 for development (backtest) mode
  - freq: character (M, W, Q, D) for data frequency
  - ctrylist: a cell array of 3-letter country code for a subset of the universe
  - nbucket: numeric for number of bucket used for style neutralization
  - dateParam: cell array for parameters used in myfints.genDateSeries
  - univname: string for name of the universe defined by users
- Output
  - o: factor analyzer object

AddFactor

- Command

```
o = AddFactor(o, facinfo, islive)
```

- Description
  - Add new factors into the object
- Input
  - o: the original FactorAnalyzer object
  - facinfo: structure with field ‘name’ as the factorids to add
  - isprod: 1 for production (live), 0 for development (backtest)
- Output
  - o: the FactorAnalyzer object with new factors added

AnalyzeAlpha

- Command

```
result = AnalyzeAlpha(o, varargin)
```

- Description
  - Construct alpha and calculate long-short performance
- Input
  - o: FactorAnalyzer object
  - varargin: cell array of variables in the format of {'varname1', varvalue1, 'varname2', varvalue2,…}
    - faclist: cell array of factorids selected for alpha
    - normdata: myfints used for normalization
    - normlevel: = 1,2,3,4 when normdata is GICS, = 'customized' when normdata is anything else
    - wgtmethod: string for weighting method of factors, possible choices are: 'EW', 'IC', 'IR'
    - window: numeric for window of calculating IC and IR according to the wgtmethod
    - facgrp: cell array for classify factors into groups when applying weighting method, e.g. {[1,2],[3,4,],[5]} will group factor 1 & 2 together, 3 & 4 together, and 5 separately.
- Output
  - Result: structure with following fields:
    - Alpha: myfints for alpha
    - IC: myfints for cross-section IC
    - LS: myfints for long-short return
    - Long: myfints for long side return contribution
    - Short: myfints for short side return contribution
    - AC: myfints for auto-correlation
    - TO: myfints for long short portfolio turnover

CalcStatistics

- Command

```
[o_old, o_new] = CalcStatistics(o, startdate, enddate, varargin)
```

- Description
  - Calculate statistics and generate pdf report for each factors
  - Dynamic input provide users with plenty of flexibility to customized their analysis
- Input
  - o: FactorAnalyzer object
  - startdate
  - enddate
  - varargin: cell array of variables in the format of {'varname1', varvalue1, 'varname2', varvalue2,…}
    - gicslevel: numeric for level of GICS (1,2,34) to neutralize on
    - facOrAlpha: NaN for all factors; cell array to specify a list of factorid; myfints for external signal
    - custstyle: structure for customized neutralization style, see FA_example.m for example
    - custuniv: structure for customized universe, see FA_example.m for example

        - custplot: cell array of strings for customized plots, possible choices are: 'ByGICS', 'ByCtry', 'ByRegime', 'LongShort', 'RtnByMcap', 'ScoreByMcap', 'RtnByCost', 'ScoreByCost', 'RtnByLiq', 'ScoreByLiq'.

- Output
    - o_old: FactorAnalyzer object which is identical to the input object but with updated field: 'statistics'
    - o_new: FactorAnalyzer object with is modified according to the input customized universe

CalculateCorrMat

- Command

```
o = CalculateCorrMat(o, faclist)
```

- Description
    - Calculate cross-section correlation among raw factor values and take the mean over time
- Input
    - o: factorAnalyzer object
    - faclist: cell array for list of factor id selected
- Output
    - o: FactorAnalyzer object with updated field: corrmat

GetSummary

- Command:

```
GetSummary(o, faclist, savepath, filename)
```

- Description
    - Generate a excel file with factor summaries according to the 'statistics' field of the object
- Input
    - o: FactorAnalyzer object
    - faclist: cell array for list of factor id selected
    - savepath: string for the path to save for the file
    - filename: string for the name of the file
- Output
    - An excel file with multiple sheets, each corresponding to one neutralization style

LoadRiskModel

- Command

```
o = LoadRiskModel(o)
```

- Description
  - Load the EM risk model from database
- Input
  - o: FactorAnalyzer object
- Output
  - o: FactorAnalyzer object with updated field: 'riskmodel'

RegressionAnalysis

- Command

```
[o, stat, model] = RegressionAnalysis(o, varargin)
```

- Description
  - Perform regression analysis and generate PDF / excel reports with various user customization
- Input
  - o: FactorAnalyzer object
  - varargin: cell array of variables in the format of {'varname1', varvalue1, 'varname2', varvalue2,…}
    - reportname: string for report name (user can also add savepath in front of it)
    - faclist: cell array of factorid
    - facgrp: numeric vector for factor group classification. e.g. [1,1,1,2,2,3] classify first three factors to group 1, the latter two factors to group 2, and the last factor to group 3
    - grpwgtmethod: weighting method for factors within the group. Choices are 'EW', 'IC', 'IR'
    - grpwgtwindow: window of calculating IC and IR when applying inner-group weighting method
    - sectordummy: 1 for including sector dummy in regression, 0 for not
    - sectorlevel: 1,2,3,4 for GICS level used to create sector dummy
    - riskneutral: 1 for including EM risk factors in regression, 0 for not
    - numriskfac: number of risk factors to include in regression
    - otherriskfac: cell array of myfints for other risk factors input by users
    - buildmodel: 1 for building a multi-factor model and output corresponding result
    - modelmethod: method of building multi-factor model: 'ALL' - using all factors in faclist, 'Forward' - keep adding factors as long as the R2 contributed by alpha factors are increasing. 'Backward'- keep dropping factors as long as they are not significant (T-stat < 1.5)
    - predmethod: string for predicting next period factor returns: 'EMA' for exponential moving average and 'SMA' for simple moving average
    - predwindow: numeric for window used in predicting factor returns.
- Output
  - o: FactorAnalyzer object with updated field 'riskmodel' if riskneutral = 1, otherwise identical to input o.
  - stat: structure for individual factor statistics with following fields
    - option: the input option of user

        - factor_norm: normalized factor scores
        - unigrp: unique factor group based on user input
        - grpfacts: cell array of grouped factor score, including all dummy variables and risk factors
        - grpfac_ac: average auto-correlation of grouped factors
        - uvrbeta: beta from univariate regression (return vs. single alpha factor)
        - uvr_t: fama-macbeth t-stat from univariate regression (return vs. single alpha factor)
        - mvrbeta: beta from multivariate regression (return vs. single alpha factor + risk factors)
        - mvr_t: fama-macbeth t-stat from multivariate regression (return vs. single alpha factor + risk factors)
        - mvrFacPF: factor portfolio implied from multivariate regression (return vs. single alpha factor + risk factors), which is neutralized to all other risk factors
    - model: structure for model statitstics with following fields:
        - beta: beta from multivariate regression (return vs. multiple alpha factors + risk factors)
        - tstat: fama-macbeth t-stat from multivariate regression (return vs. multiple alpha factors + risk factors)
        - adjR2: myfints of adjusted R-square of regression
        - R2contrib: R-square contribution from alpha factors
        - facPF: factor portfolio for each factor from multivariate regression (return vs. multiple alpha factors + risk factors)
        - rethat: myfints of estimated return from regression
        - predbeta: myfints of predicted beta based on user input
        - alphafacidx: index of factors that are used to construct alpha
        - alpha: myfints for predicted alpha = sum(predbeta * factor portfolio score)
    - PDF and excel file with the input file name will also be generated. On the report:
        - Raw (univariate) beta is from return vs. single alpha factor regression
        - Pure (multivariate) beta is from return vs. (single alpha factor + risk factors) regression
        - Model (multivariate) beta is from return vs. (multiple alpha factor + risk factors) regression