factorAnalytics: A Concise User Guide

Yi-An Chen

April 10, 2014

1 Introduction

This vignette aims to help users to learn how to use fit factor model with `factorAnalytics` package. We will walk through users a few examples from data input to risk analysis and performance attribution.

2 Factor Model

A factor model is defined as

$$r_t = bf_t + \epsilon_t \;, t = 1 \cdots T \tag{1}$$

Where r_t is N x 1 exress returns, b is N x K and f is K x 1. N is number of variables and K is number of factors. b is usually called factor exposures or factor loadings, and b can be time-varying b_t in fundamental factor model setting. f is factor returns. ϵ_t is serial uncorrelated but may be cross-correlated. The model is useful to fit asset pricing model. The famous CAPM (Capital Assets Pricing Model) is a one factor model with f equal to market returns.

`factorAnalytics` package provides 3 different kinds of factor models. That is fundamental factor model, statistical factor model and time series factor model. We will walk through them one by one.

2.1 Fundamental Factor Model

In the case of fundamental factor model, we assume we know factor exposures b which are assets characteristics, like market capitalization or book-to-market ratio. Therefore, b_t is known and f_t is unknown. We run cross-section OLS or WLS regression to estimate f_t for each time period. In specific,

$$r_t = f_M + b\hat{f}_t + \hat{\epsilon}_t \;, t = 1 \cdots T \tag{2}$$

f_M is normally called market factor or world factor depending on the context on the country level or global level. Econometrically, it is an intercept term of fundamental factor model. f_t is estimated with cross-sectional in each period t.

This approach is also called BARRA type approach since it is initially developed by BARRA and later on been merged by MSCI. The famous Barra global equity model (GEM3) contains more than 50 factors.

2.2 Example 1

We will walk through the first examples in this section where use style factors like size are used.

2.2.1 Loading Data

Let's look at the arguments of `fitFundamentalFactorModel()` which will deal with fundamental factor model in `factorAnalytics`.

```
> library(factorAnalytics)
> args(fitFundamentalFactorModel)

function (data, exposure.names, datevar, returnsvar, assetvar,
    wls = TRUE, regression = "classic", covariance = "classic",
    full.resid.cov = FALSE, robust.scale = FALSE, standardized.factor.exposure = FALSE,
    weight.var)
NULL
```

`data` is in class of `data.frame` and is required to have *assetvar*,*returnvar* and *datevar*. One can image *data* is like panel data setup and need firm variable and time variable. Data has dimension (N x T) and at least 3 consumes to specify information needed.

We download data from CRSP/Compustat quarterly fundamental and name it `equity`. It contains 67 stocks and 106 time period from January 2000 to December 2013.

```
> #equity <- data(equity)
> equity <- read.csv(file="equity.csv")
> names(equity)

 [1] "gvkey"    "datadate" "fyearq"   "fqtr"     "indfmt"   "consol"
 [7] "popsrc"   "datafmt"  "tic"      "conm"     "CURCDQ"   "DATACQTR"
[13] "DATAFQTR" "CEQQ"     "CSHOQ"    "COSTAT"   "DVPSPQ"   "MKVALTQ"
[19] "PRCCQ"    "SPCINDCD" "SPCSECCD"

> length(unique(equity$datadate)) # number of period t

[1] 106

> length(unique(equity$tic)) # number of assets

[1] 63
```

We need asset returns to run our model. We can utilize `Delt()` to calculate price percentage change which is exactly asset returns in `quantmod` package.

```
> library(quantmod)   # for Delt. See Delt for detail
> equity <- cbind(equity,do.call(rbind,lapply(split(equity,equity$tic),
+                                            function(x) Delt(x$PRCCQ))))
> names(equity)[22] <- "RET"
```

We want market value and book-to-market ratio to be our style factors. Market vale can be achieved by common stocks outstanding multiply price and book value is common/ordinary equity value. We take log for market value.

```
> equity$MV <- log(equity$PRCCQ*equity$CSHOQ)
> equity$BM <- equity$CEQQ/equity$MV
```

now we will fit Equation 2 with b = [MV BM].

We will get an error message if `datevar` is not `as.Date` format compatible. In our example, our date variable is *DATACQTR* and looks like "2000Q1". We have to convert it to `as.Date` compatible. We can utilize `as.yearqtr` in `xts` package to do it. Also, we will use character string for asset variable instead of factor.[1]

```
> a <- unlist( lapply(strsplit(as.character(equity$DATACQTR),"Q"),
+                function(x)  paste(x[[1]],"-",x[[2]],sep="") ) )
> equity$yearqtr <- as.yearqtr(a,format="%Y-%q")
> equity$tic <- as.character(equity$tic)
> equity <- subset(equity,yearqtr != "2000 Q1") # delete the first element of each assets
```

2.2.2 Fit the Model

fit the function:

```
> fit.fund <- fitFundamentalFactorModel(exposure.names=c("BM","MV"),datevar="yearqtr",
+                                       returnsvar ="RET",assetvar="tic",wls=TRUE,
+                                       data=equity)

*** Possible outliers found in the factor returns:

                      BM
2000-04-01 -6.558278e-05
                    BM
2000-10-01 -6.2829e-05
                      BM
2001-07-01 -0.0001230401

> names(fit.fund)
```

[1]The best data input is to convert all your data into xts class since we use xts to compute everything in this package, although it is not restricted to it.

```
 [1] "returns.cov"    "factor.cov"     "resids.cov"     "resid.variance"
 [5] "factor.returns" "residuals"      "tstats"         "call"
 [9] "data"           "asset.names"    "beta"           "datevar"
[13] "returnsvar"     "assetvar"       "exposure.names"
```

A few notice for fitting fundamental factor model. So far this function can only deal with balanced panel because we want to extract return covariance and residuals and so on. Second, `datevar` has to be `as.Date` compatible, otherwise the function can not read time index. It is somehow inconvenient but make sure we will not mess up with any time issue.

Default fit method for `fitFundamentalFactorModel()` is classic OLS and covariance matrix is also classic covariance matrix defined by `covClassic()` in `robust` package. One can change to robust estimation and robust covariance matrix estimation.

`returns.cov` contains information about returns covariance. return covariance is

$$\Sigma_x = B\Sigma_f B' + D$$

If `full.resid.cov` is *FALSE*, D is diagonal matrix with variance of residuals in diagonal terms. If *TRUE*, D is covariance matrix of residuals.

```
> names(fit.fund$returns.cov)

[1] "cov"          "mean"         "eigenvalues"
```

Please check out `fit.fund$factor.cov`, `fit.fund$resids.cov` and `fit.fund$resid.variance` yourself for detail.

factor returns, residuals,t-stats are xts class.

```
> fit.fund$factor.returns
> fit.fund$residuals
> fit.fund$tstats
```

Output of `fitFundamentalFactorModel()` is of class *FundamentalFactorModel*. There are generic function `predict`, `summary`, `print` and `plot` can be applied.

```
> summary(fit.fund)
> predict(fit.fund)
> print(fit.fund)
```

If *newdata* is not specified in `predict()`, fitted value of fundamental factor model will be shown, otherwise, predicted value will be shown.

`plot()` method has several option to choose,

```
> plot(fit.fund)
Factor Analytic Plot
Make a plot selection (or 0 to exit):
```

```
numbers of assets are greater than 3 , show only first 3 assets
```

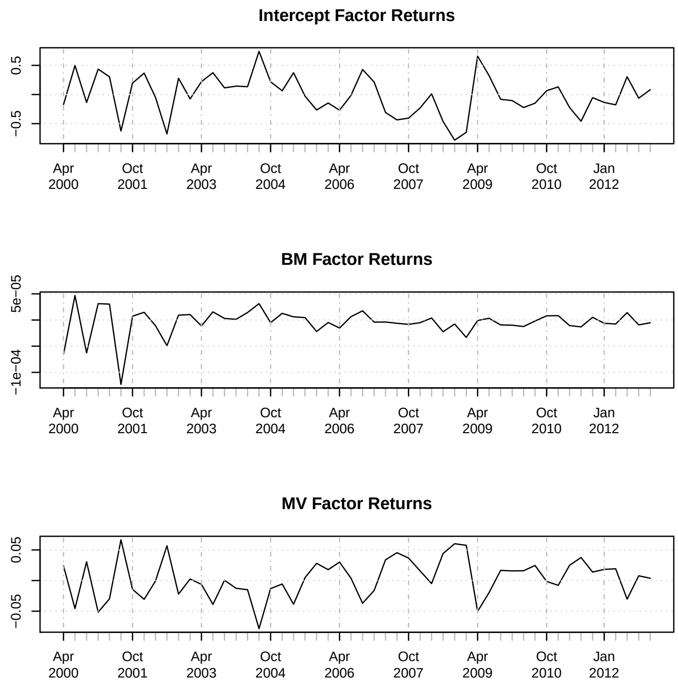

Figure 1: Time Series of factor returns

```
1: Factor returns
2: Residual plots
3: Variance of Residuals
4: Factor Model Correlation
5: Factor Contributions to SD
6: Factor Contributions to ES
7: Factor Contributions to VaR

Selection: plot(fit.fund)
Enter an item from the menu, or 0 to exit
```

For example, choose 1 will give factor returns and it looks like in Figure 1

2.3 Example 2: Barra type industry/country model

In a global equity model or specific country equity model, modelers can use industry/country dummies. In our example, we have 63 stocks in different industry. In specific,

$$x_{it} = \Sigma_{j=1}^{J} b_{i,j} f_{i,t} + \epsilon_{i,t}, \; for\, each\, i\,, t \tag{3}$$

where $b_{i,j} = 1$ if stock i in industry j and $b_{i,j} = 0$ otherwise. In matrix form:

$$x_t = Bf_t + \epsilon_t$$

and B is the N X J matrix of industry dummies.

SPCINDCD in our `equity` contains *S&P* industry codes, we add this variable name into `exposure.names` and we can fit Barra type industry model. Be sure this variable is of class*character* not *numeric*. Otherwise the function will not create dummies.

```
> equity$SPCINDCD <- as.character(equity$SPCINDCD)
> fit.ind <- fitFundamentalFactorModel(exposure.names=c("SPCINDCD"),datevar="yearqtr",
+                                       returnsvar ="RET",assetvar="tic",wls=FALSE,
+                                      data=equity)
```

```
*** Possible outliers found in the factor returns:

           SPCINDCD225
2000-07-01   0.4772727
           SPCINDCD403 SPCINDCD470
2000-10-01  -0.6929461   0.4946921
           SPCINDCD280
2002-04-01   -0.324748
           SPCINDCD470
2003-04-01   0.5166889
           SPCINDCD410 SPCINDCD435
2005-04-01  -0.4928367  -0.4475485
           SPCINDCD470
2008-01-01  -0.7062715
           SPCINDCD120 SPCINDCD160 SPCINDCD400
2009-04-01    1.307985   0.5309631   0.6264418
           SPCINDCD180
2011-01-01    1.997519
           SPCINDCD145
2012-07-01  -0.2566714
           SPCINDCD370
2013-01-01   0.3966165
```

`fitFundamentalFactorModel()` supports mixed model like fit industry/country dummy factor exposures and style factor exposures together. For example,

```
> fit.mix <- fitFundamentalFactorModel(exposure.names=c("BM","MV","SPCINDCD"),
+                                      datevar="yearqtr",returnsvar ="RET",
+                                      assetvar="tic",wls=FALSE,data=equity)
```

2.3.1 Standardizing Factor Exposure

It is common to standardize factor exposure to have weight mean 0 and standard deviation equal to 1. The weight are often taken as proportional to square root of market capitalization, although other weighting schemes are possible.

We will try example 1 but with standardized factor exposure with square root of market capitalization. First we create a weighting variable.

```
> equity$weight <- sqrt(exp(equity$MV)) # we took log for MV before.
```

We can choose `standardized.factor.exposure` to be `TRUE` and `weight.var` equals to weighting variable.

```
> fit.fund2 <- fitFundamentalFactorModel(exposure.names=c("BM","MV"),
+                                      datevar="yearqtr",returnsvar ="RET",
+                                      assetvar="tic",wls=TRUE,data=equity,
+                                      standardized.factor.exposure = TRUE,
+                                        weight.var = "weight"  )

*** Possible outliers found in the factor returns:

                     BM
2000-04-01 -0.001441492
                    BM
2001-10-01 0.001515949
                   MV
2009-04-01 0.00426395
```

The advantage of weight facotr exposures is better interpretation of factor returns. f_t can be interpreted as long-short zero investment portfolio. In our case, f_{MVt} will long big size stocks and short small size stocks.

2.4 Statistical Factor Model

In statistical factor model, neither factor exposure b (normally called factor loadings in statistical factor model) nor factor returns f_t are observed in equation 1. So we can rewrite the model as:

$$r_t = bf_t + \epsilon_t \, , t = 1 \cdots T \tag{4}$$

Factor returns f_t can be calculated as principle components of covariance matrix of assets returns if number of asset N is less than the number of time period T, and factor loadings can be calculated using conventional least square technique.

By default, the first principle component or factor will explain the most variation of returns covariance matrix and so on.

In some cases, when number of assets N is larger than number of time period T. Connor and Korajczyk (1986) develop an alternative method called asymptotic principal components, building on the approximate factor model theory of Chamberlain and Rothschild (1983). Connor and Korajczyk analyze the eigenvector of the T X T cross product of matrix returns rather then N X N covariance matrix of returns. They show the first k eigenvectors of this cross product matrix provide consistent estimates of the k X T matrix of factor returns.

We can use function `fitStatisticalFactorModel` to fit statistical factor model. First, we need asset returns in time series or xts class. We choose xts to work with because time index is easy to handle but this is not restricted to the function.

```
> library(xts)
> tic <- unique(equity$tic)
> ret <- xts(NA,as.yearqtr("2000 Q2",format="%Y Q%q"))
> for (i in tic) {
+ temp <- subset(equity,tic == i)
+ ret.new <- xts(temp$RET,as.yearqtr(temp$yearqtr))
+ names(ret.new) <- i
+ ret <- merge(ret,ret.new)
+ }
> ret <- ret[,-1]
> dim(ret)
```

```
[1] 52 63
```

The data `ret` contians 63 assets and 52 time periods. We will exploit asymptotic principal components analysis to fit statistical model. There are two ways to find numbers of factors, Connor and Korajczyk(1995) and Bai and Ng (2002). Both are provided in our function. We will use Bai and Ng (2002) to choose the numbers of factors.

```
> fit.stat <- fitStatisticalFactorModel(data=ret,
+                                        k= "bn")
> names(fit.stat)
```

```
 [1] "factors"       "loadings"      "k"             "alpha"
 [5] "ret.cov"       "r2"            "eigen"         "residuals"
 [9] "asset.ret"     "asset.fit"     "mimic"         "resid.variance"
[13] "call"          "data"          "assets.names"
```

5 factors is chosen by Bai and Ng (2002). Factor returns can be found `fit.stat$factors`.

```
> fit.stat$k
```

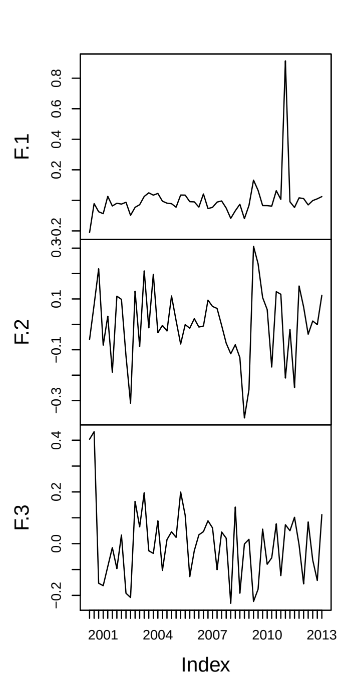

Figure 2: Time Series of statistical factor returns

```
[1] 5
```

We can plot factor returns with generic function `plot`.

Finally, screen plot of eigenvalues shows how much variance can be explained by factors. We can see the first factor explain more than 70 percent of variation of cross-product matrix.

Similar to `fitFundamentalFactorModel`, generic functions like `summary`, `print`, `plot` and `predict` can be used for statistical factor model.

2.5 Time Series Factor Model

In Time Series factor model, factor returns f_t is observed and taken as macroeconomic time series like GDP or other time series like market returns or credit spread. In our package, we provid some common used times series in data set `CommonFactors`. `factors` is monthly time series and `factors.Q` is quarterly time series.

```
> data(CommonFactors)
> names(factors.Q)
```

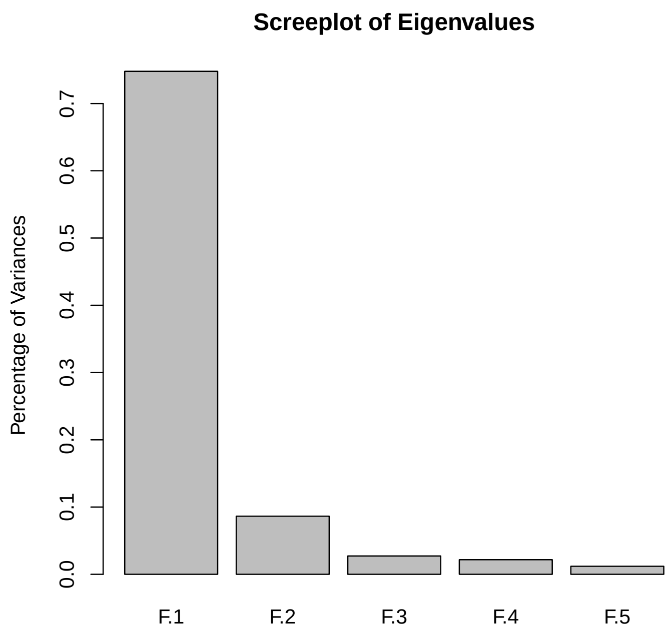

Figure 3: Screen Plot of Eigenvalues

```
 [1] "SP500"          "GS10TR"         "USD.Index"      "Term.Spread"
 [5] "Credit.Spread"  "DJUBSTR"        "dVIX"           "TED.Spread"
 [9] "OILPRICE.Close" "TB3MS"
```

We can combine with our data `ret` and get rid of NA values.

```
> ts.factors <- xts(factors.Q,as.yearqtr(index(factors.Q),format="%Y-%m-%d"))
> ts.data <- na.omit(merge(ret,ts.factors))
```

We will use SP500, 10 years and 3 months term spread and difference of VIX as our common factors.

```
> fit.time <- fitTimeSeriesFactorModel(assets.names=tic,
+                                      factors.names=c("SP500","Term.Spread","dVIX"),
+                                      data=ts.data,fit.method="OLS")
```

`asset.fit` can show model fit for each assets, for example for asset `AA`.

```
> fit.time$asset.fit$AA

Call:
lm(formula = fm.formula, data = reg.df)

Coefficients:
(Intercept)        SP500  Term.Spread         dVIX
 -0.0175480    1.7366404   -0.1423982   -0.0000496
```

`fitTimeSeriesFactorModel` also have various variable selection algorithm to choose. One can include every factor and let the function to decide which one is the best model. For example, we include every common factors and use method `stepwise` which utilizes `step` function in `stat` package

```
> fit.time2 <- fitTimeSeriesFactorModel(assets.names=tic,
+                                       factors.names=names(ts.factors),
+                                       data=ts.data,fit.method="OLS",
+                                       variable.selection = "stepwise")
```

There are 5 factors chosen for asset AA for example.

```
> fit.time2$asset.fit$AA

Call:
lm(formula = AA ~ SP500 + GS10TR + USD.Index + DJUBSTR + OILPRICE.Close,
    data = reg.df)

Coefficients:
   (Intercept)           SP500          GS10TR       USD.Index         DJUBSTR
     -0.005523        1.090955       -1.174835       -1.733384        0.852601
OILPRICE.Close
     -0.429924
```

Generic functions like `summary`, `print`, `plot` and `predict` can also be used for time series factor model as previous section.

3 Risk Analysis

3.1 Factor Model Risk Budgeting

One can do risk analysis with factor model. According to Meucci (2007), factor model can be represented as

$$r_{it} = \alpha_i + \beta_{i1} f_{1t} + \beta_{i2} f_{2t} + \cdots + \beta_{ik} f_{kt} + \sigma i z_{it},\ i = 1 \cdots N,\ t = 1 \cdots T \quad (5)$$

$$= \alpha_i + \tilde{\beta_i}' \tilde{F_t} \quad (6)$$

where z_{it} is the standardized residuals and $\epsilon_{it}/\sigma_i = z_{it}$, $\tilde{\beta_i} = [\beta_{1i}, \ldots, \beta_{ki}, \sigma_i]$, $\tilde{F_t} = [f_{1t}, \ldots, f_{kt}, z_{it}]$

Common risk measures like standard deviation, value-at-risk and expected shortfall are function of homogeneous of degree 1. By Euler theoreom, risk metrics (RM) can be decomposed to

$$RM_i = \beta_{1i} \frac{\partial RM_i}{\partial \beta_{1i}} + \beta_{2i} \frac{\partial RM_i}{\partial \beta_{2i}} + \cdots + \beta_{ki} \frac{\partial RM_i}{\partial \beta_{ki}} + \sigma_i \frac{\partial RM_i}{\partial \sigma_i} \quad (7)$$

where
$\frac{\partial RM_i}{\partial \beta_{ki}}$ is called marginal contribution of factor k to RM_i
$\beta_{ki} \frac{\partial RM_i}{\partial \beta_{ki}}$ is called component contribution of factor k to RM_i
$\beta_{ki} \frac{\partial RM_i}{\partial \beta_{ki}} / RM_i$ is called percentage contribution of factor k to RM_i

`factorAnalytics` package provide 3 different risk metrics decomposition, Standard deviation (Std), Value-at-Risk (VaR) and Expected Shortfall (ES). Each one with different distribution such as historical distribution, Normal distribution and Cornish-Fisher distribution.

This example shows factor model VaR decomposition with Normal distribution of asset AA for a statistical factor model.

```
> data.rd <- cbind(ret[,"AA"],fit.stat$factors,
+                  fit.stat$residuals[,"AA"]/sqrt(fit.stat$resid.variance["AA"]))
> var.decp <- factorModelVaRDecomposition(data.rd,fit.stat$loadings[,"AA"],
+                             fit.stat$resid.variance["AA"],tail.prob=0.05,
+                              VaR.method="gaussian")
> names(var.decp)

[1] "VaR.fm"     "n.exceed"   "idx.exceed" "mVaR.fm"    "cVaR.fm"
[6] "pcVaR.fm"
```

VaR, number of exceed, id of exceed, marginal contribution to VaR, component contribution to VaR and percentage contribution to VaR are computed. Let see VaR and component contribution to VaR

```
> var.decp$VaR.fm

[1] 0.3736797
```

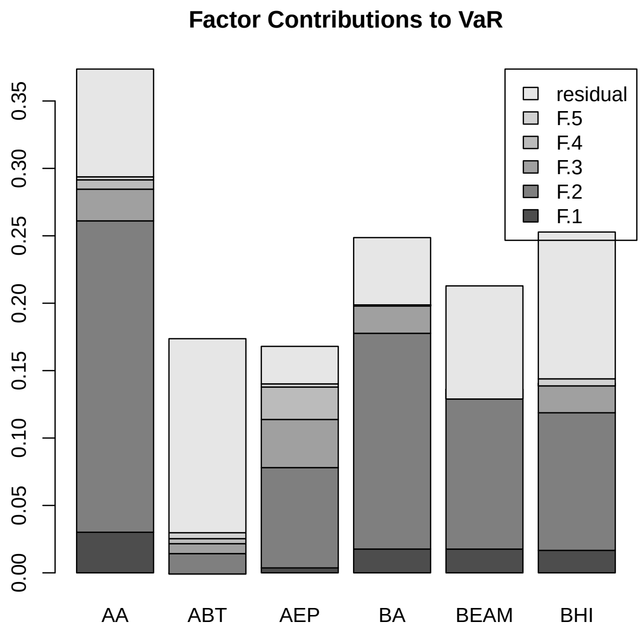

Figure 4: Component Contribution to VaR for Statistical Factor Model.

```
> var.decp$cVaR.fm
```

```
            F.1      F.2       F.3         F.4         F.5       AA.1
CVaR 0.03012737 0.230954 0.0235052 0.006911229 0.002247496 0.07993438
```

It looks like the second factor contributes the largest risk to asset AA.

One can use `plot()` method to see barplot of risk budgeting. Default is to show 6 assets. Figure 4 shows componenet contribution to VaR for several different assets.

3.2 Portfolio Risk Budgeting

Let $Rp_t = Rp_t(w)$ denote the portfolio return based on the vector of portfolio weights w. Let RM(w) denote a portfolio risk measure.

$$RM = w_1 \frac{\partial RM}{\partial w_1} + w_2 \frac{\partial RM}{\partial w_2} + \cdots + w_N \frac{\partial RM}{\partial w_N} \tag{8}$$

where

$\frac{\partial RM}{\partial w_i}$ is called marginal contribution of asset i to RM

$w_i \frac{\partial RM}{\partial w_i}$ is called component contribution of asset i to RM
$w_i \frac{\partial RM}{\partial w_i}/RM$ is called percentage contribution of asset i to RM

we can use function `VaR()` in `PerformanceAnalytics`. Suppose we have an eqaully weighted portfolio of 63 assets in data set `ret`. The following code can compute portfolio VaR, component contribution to VaR and percentage contribution to VaR

```
> VaR(R=ret,method="gaussian",portfolio_method="component")
```

4 Performance Attribution

User can do factor-based performance attribution with `factorAnalytics` package. factor model:

$$r_t = \alpha + Bf_t + e_t,\ t = 1 \cdots T \tag{9}$$

can break down asset returns into two pieces. The first term is *returns atttributed to factors* Bf_t and the second term is called *specific returns* which is simply $\alpha + e_t$.

For example, we can breakdown time series factor model.
Function `factorModelPerformanceAttribution()` can help us to calculate performance attribution.

```
> ts.attr <- factorModelPerformanceAttribution(fit.time)
> names(ts.attr)

[1] "cum.ret.attr.f" "cum.spec.ret"   "attr.list"
```

There are 3 items generated by the function. `cum.ret.attr.f` will return a N x K matrix with cummulative returns attributed to factors. `cum.spec.ret` will return a N x 1 matrix with cummulative specific returns. `attr.list` will return a list which contains returns atttribution to each factors and specific returns asset by asset. In addition, a *FM.attribution* class will be generated and generic function `print()`, `summary()` and `plot()` can be applied to it.

4.1 Benchmark and Active Returns

Portfolio performance is usually compared to similar type of benchmark. US equity portfolio will compare its performance with S&P 500 index for example. Therefore, *active returns* under active management is interested. We define active returns = assets returns - benchmark.

We can also calculate active return attribution just simply fit active return with fundamental factor model, statistical factor model or time series factor model and calculate by `factorModelPerformanceAttribution()`.