Factor Model Risk Analysis in R

Scottish Financial Risk Academy, March 15, 2011

Eric Zivot
Robert Richards Chaired Professor of Economics
Adjunct Professor, Departments of Applied Mathematics, Finance and Statistics
University of Washington
BlackRock Alternative Advisors, Seattle WA

Outline

- Data for examples
- Risk measures
- Factor risk budgeting
- Portfolio risk budgeting
- Factor model Monte Carlo

Set Options and Load Packages

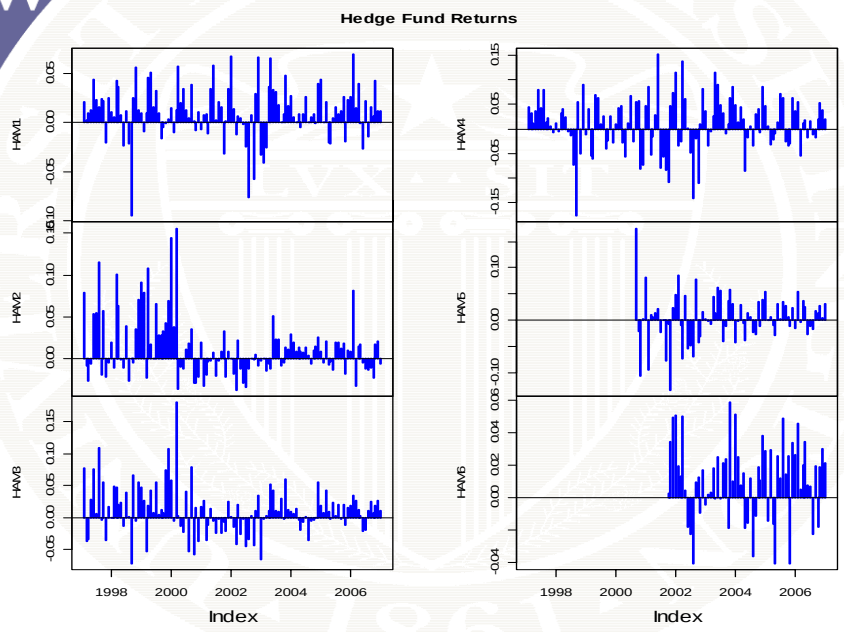
```
# set output options
> options(width = 70, digits=4)

# load required packages
> library(ellipse)              # plot ellipses
> library(fEcofin)              # various economic and
                                # financial data sets
> library(PerformanceAnalytics) # performance and risk
                                # analysis functions
> library(tseries)              # MISC time series funs
> library(xts)                  # time series objects
> library(zoo)                  # and utility functions
```

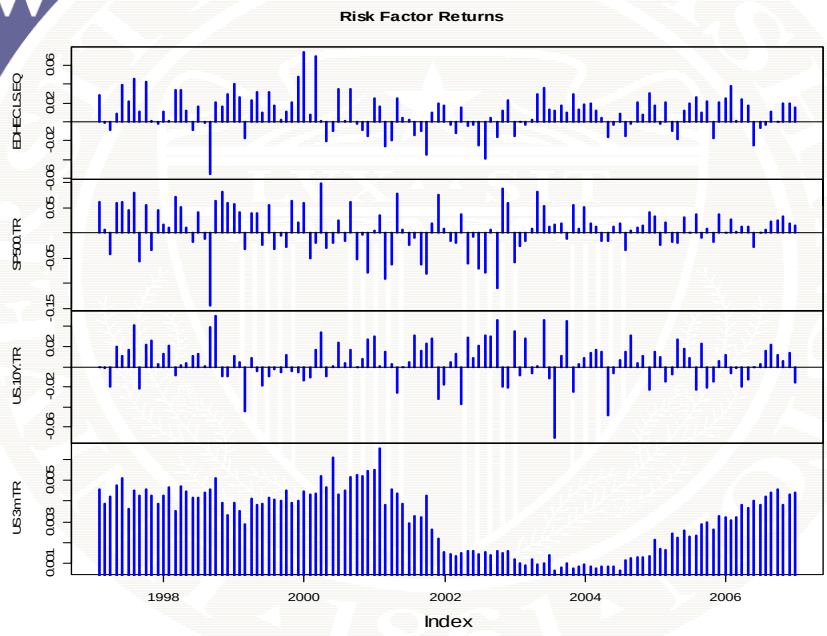

Hedge Fund Data

```
# load hypothetical long-short equity asset managers data
# from PerformanceAnalytics package
> data(managers)
> class(managers)
[1] "xts" "zoo"
> start(managers)
[1] "1996-01-30"
> end(managers)
[1] "2006-12-30"
> colnames(managers)
 [1] "HAM1"        "HAM2"        "HAM3"        "HAM4"
 [5] "HAM5"        "HAM6"        "EDHEC LS EQ" "SP500 TR"
 [9] "US 10Y TR"   "US 3m TR"
# remove data prior to 1997-01-30 due to missing vals
> managers = managers["1997::2006"]
```

Plot Hedge Fund and Factor Returns

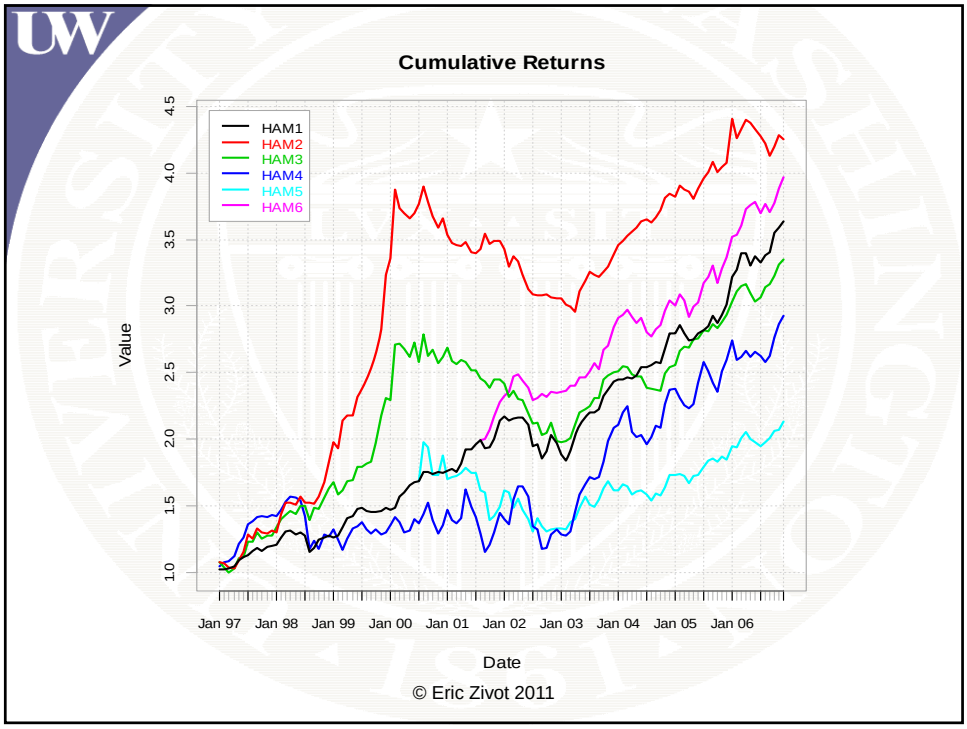
```
> my.panel <- function(...) {
+   lines(...)
+   abline(h=0)
+ }

# use plot.zoo() from zoo package for multi-panel plots
> plot.zoo(managers[, 1:6], main="Hedge Fund Returns",
+          plot.type="multiple", lwd=2, col="blue",
+          panel=my.panel)

> plot.zoo(managers[, 7:10], main="Risk Factor Returns",
+          plot.type="multiple", lwd=2, col="blue",
+          panel=my.panel)
```

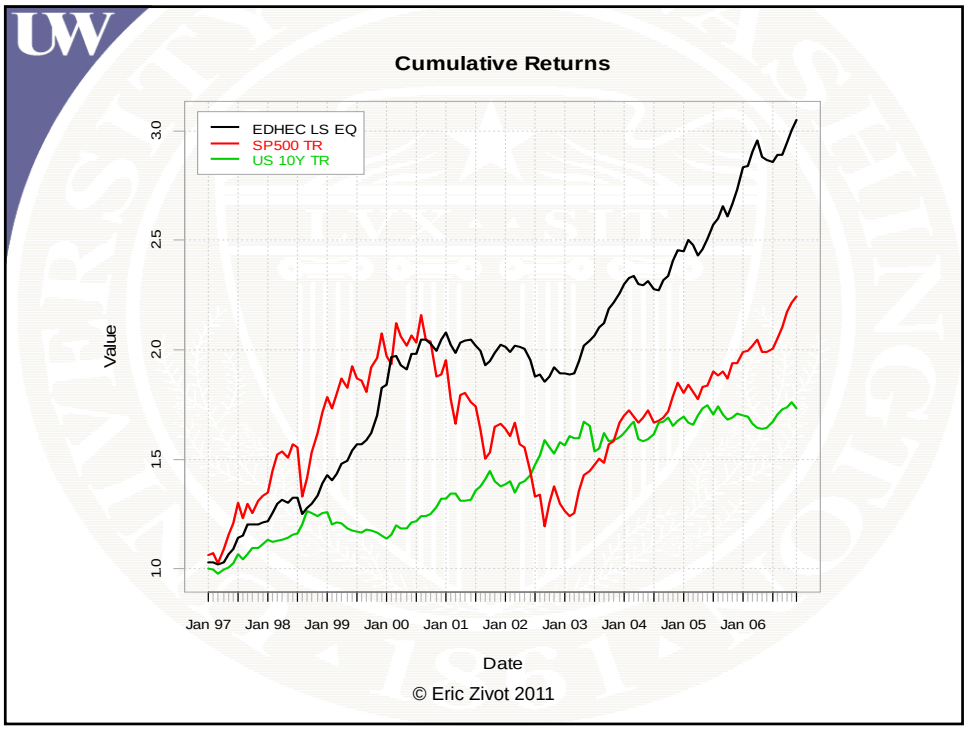





Plot Cumulative Returns

```
# plot cumulative returns using PerformanceAnalytics
# function chart.CumReturns()

# hedge funds
> chart.CumReturns(managers[,1:6], main="Cumulative Returns",
+                  wealth.index=TRUE, legend.loc="topleft")

# risk factors
> chart.CumReturns(managers[,7:9], main="Cumulative Returns",
+                  wealth.index=TRUE, legend.loc="topleft")
```





Descriptive Statistics: Funds

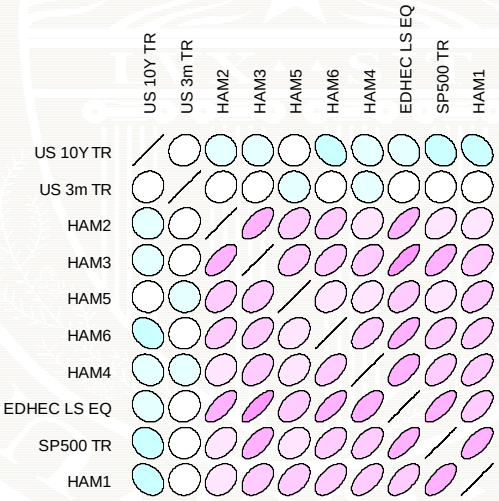
```
# Use table.Stats() function from PerformanceAnalytics package
> table.Stats(managers[, 1:6])
                     HAM1     HAM2     HAM3     HAM4    HAM5    HAM6
Observations    120.0000 120.0000 120.0000 120.0000 77.0000 64.0000
NAs               0.0000   0.0000   0.0000   0.0000 43.0000 56.0000
Minimum          -0.0944  -0.0371  -0.0718  -0.1759 -0.1320 -0.0404
Quartile 1        0.0000  -0.0108  -0.0059  -0.0236 -0.0164 -0.0016
Median            0.0107   0.0075   0.0082   0.0128  0.0038  0.0128
Arithmetic Mean   0.0112   0.0128   0.0108   0.0105  0.0041  0.0111
Geometric Mean    0.0108   0.0121   0.0101   0.0090  0.0031  0.0108
Quartile 3        0.0252   0.0224   0.0263   0.0468  0.0309  0.0255
Maximum           0.0692   0.1556   0.1796   0.1508  0.1747  0.0583
SE Mean           0.0024   0.0033   0.0033   0.0050  0.0052  0.0030
LCL Mean (0.95)   0.0064   0.0062   0.0041   0.0006 -0.0063  0.0051
UCL Mean (0.95)   0.0159   0.0193   0.0174   0.0204  0.0145  0.0170
Variance          0.0007   0.0013   0.0013   0.0030  0.0021  0.0006
Stdev             0.0264   0.0361   0.0367   0.0549  0.0457  0.0238
Skewness         -0.6488   1.5406   0.9423  -0.4064  0.0724 -0.2735
Excess Kurtosis   2.1223   2.7923   3.0910   0.6453  2.1772 -0.4311
```


Descriptive Statistics: Factors

```
> table.Stats(managers[, 7:9])
                EDHEC LS EQ SP500 TR US 10Y TR
Observations      120.0000  120.0000   120.0000
NAs                 0.0000    0.0000     0.0000
Minimum            -0.0552   -0.1446    -0.0709
Quartile 1         -0.0032   -0.0180    -0.0075
Median              0.0110    0.0105     0.0051
Arithmetic Mean     0.0095    0.0078     0.0048
Geometric Mean      0.0093    0.0068     0.0046
Quartile 3          0.0214    0.0390     0.0167
Maximum             0.0745    0.0978     0.0506
SE Mean             0.0019    0.0040     0.0019
LCL Mean (0.95)     0.0058   -0.0003     0.0011
UCL Mean (0.95)     0.0132    0.0158     0.0085
Variance            0.0004    0.0020     0.0004
Stdev               0.0205    0.0443     0.0204
Skewness            0.0175   -0.5254    -0.4389
Kurtosis            0.8456    0.3965     0.9054
```

Sample Correlations

(pairwise complete obs)



Macroeconomic Factor Model (FM)

$$R_{it} = \alpha_i + \beta_{1i} EDHEC.LS.EQ_t + \beta_{2i} SP500.TR_t + \beta_{3i} US.10Y.TR_t + \varepsilon_{it}$$

- R_{it} = return in excess of T-Bill rate on hedge fund *i* in month *t*.
- $EDEC.LS.EQ_t$ = excess total return on EDHEC long-short equity index ("exotic risk factor")
- $SP500.TR_t$ = excess total return on S&P 500 index (traditional equity risk factor)
- $US.10.YR_t$ = excess total return on US 10 year T-Note (traditional rates risk factor)

Prepare Data for Regression

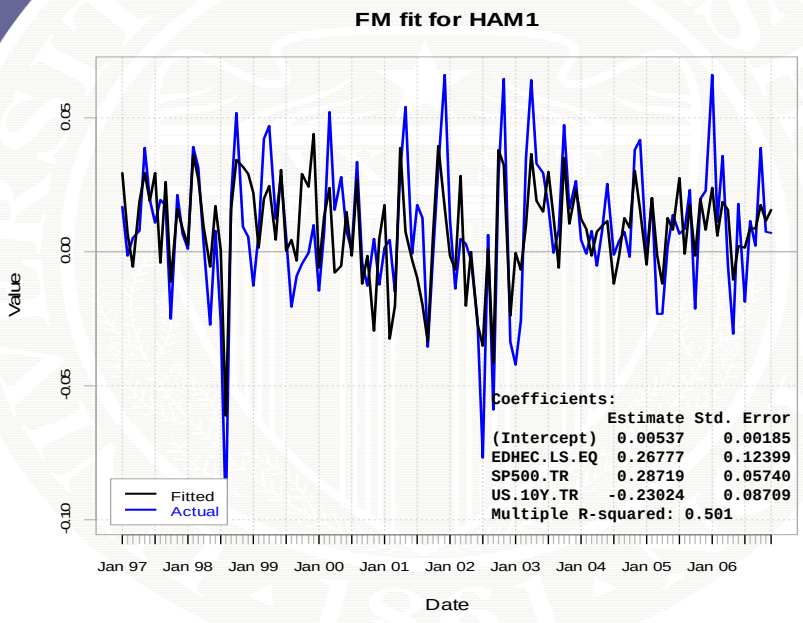
```
# subtract "US 3m TR" (risk free rate) from all
# returns. note: apply() changes managers.df to class
# "matrix"
> managers.df = apply(managers.df, 2,
+       function(x) {x - managers.df[,"US 3m TR"]})
> managers.df = as.data.frame(managers.df)
# remove US 3m TR from data.frame
> managers.df = managers.df[, -10]
# extract variable names for later use
> manager.names = colnames(managers.df)[1:6]
# eliminate spaces in factor names
> factor.names = c("EDHEC.LS.EQ", "SP500.TR",
+                  "US.10Y.TR")
> colnames(managers.df)[7:9] = colnames(managers)[7:9]
+                            = factor.names
>managers.zoo = as.zoo(na.omit(managers[,
+                               manager.names]))
```


Fit FM by Least Squares

```
# initialize list object to hold regression objects
> reg.list = list()

# initialize matrices and vectors to hold regression
# alphas, betas, residual variances and r-squared values
> Betas = matrix(0, length(manager.names),
+                length(factor.names))
> colnames(Betas) = factor.names
> rownames(Betas) = manager.names
> Alphas = ResidVars = R2values =
+         rep(0,length(manager.names))
> names(Alphas) = names(ResidVars) = names(R2values) =
+                 manager.names
```

Fit FM by Least Squares

```
# loop over all assets and estimate time series
# regression
> for (i in manager.names) {
+  reg.df = na.omit(managers.df[, c(i, factor.names)])
+  fm.formula = as.formula(paste(i,"~", ".", sep=" "))
+  fm.fit = lm(fm.formula, data=reg.df)
+  fm.summary = summary(fm.fit)
+  reg.list[[i]] = fm.fit
+  Alphas[i] = coef(fm.fit)[1]
+  Betas[i, ] = coef(fm.fit)[-1]
+  ResidVars[i] = fm.summary$sigma^2
+  R2values[i] =  fm.summary$r.squared
+ }

> names(reg.list)
[1] "HAM1" "HAM2" "HAM3" "HAM4" "HAM5" "HAM6"
```

Regression Results

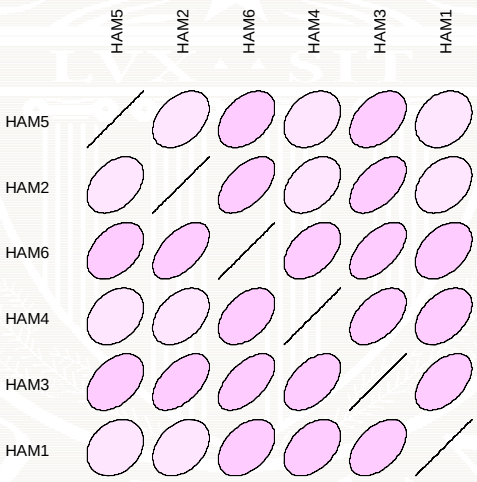
Fund	Intercept	LS.EQ	SP500	US.10YR	σ	R^2
HAM1	0.005***	0.268**	0.287***	-0.230***	0.019	0.501
HAM2	0.001	1.547***	-0.195**	0.050	0.025	0.514
HAM3	-0.001	1.251***	0.131**	0.144	0.022	0.657
HAM4	-0.002	1.222***	0.273**	-0.139	0.043	0.413
HAM5	-0.005	1.621***	-0.184	0.271	0.040	0.232
HAM6	0.004***	1.250***	-0.175*	-0.174*	0.016	0.564

***, **, * denote significance at the 1%, 5% and 10% level, respectively

FM Covariance Matrix

```
# risk factor sample covariance matrix
> cov.factors = var(managers.df[, factor.names])

# FM covariance matrix
> cov.fm = Betas%*%cov.factors%*%t(Betas) +
+          diag(ResidVars)

# FM correlation matrix
> cor.fm = cov2cor(cov.fm)
```

FM Correlations



Fund of Hedge Funds (FoHF)

Equally weighted portfolio (fund of hedge funds):

$$w_i = \frac{1}{6},\ i = \text{HAM1},\ldots,\text{HAM6}$$

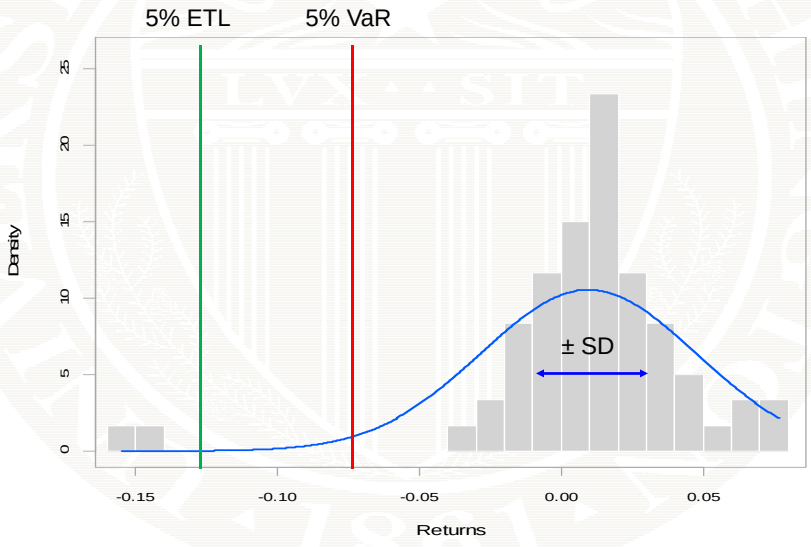
```
> w.vec = rep(1,6)/6
> names(w.vec) = manager.names
> w.vec
 HAM1  HAM2  HAM3  HAM4  HAM5  HAM6
0.167 0.167 0.167 0.167 0.167 0.167
```

```
# portfolio returns. Note: need to eliminate NA values
# from HAM5 and HAM6
> r.p = as.matrix(na.omit(managers.df[,
                          manager.names]))%*%w.vec
> r.p.zoo = zoo(r.p, as.Date(rownames(r.p)))
```

FoHF (Portfolio) FM

```
# portfolio factor model
> alpha.p = as.numeric(crossprod(Alphas,w.vec))
> beta.p = t(Betas)%*%w.vec

> var.systematic = t(beta.p)%*%cov.factors%*%beta.p
> var.specific = t(w.vec)%*%diag(ResidVars)%*%w.vec
> var.fm.p = var.systematic + var.specific
> var.fm.p = as.numeric(var.fm.p)
> r.square.p = as.numeric(var.systematic/var.fm.p)

> fm.p = c(alpha.p, beta.p, sqrt(var.fm.p), r.square.p)
> names(fm.p) = c("intercept", factor.names, "sd", "r2")
> fm.p
  intercept EDHEC.LS.EQ    SP500.TR   US.10Y.TR          sd
   0.000455    1.193067    0.022973   -0.012990    0.027817
  r-squared
   0.812435
```


Risk Measures



Testing for Normality

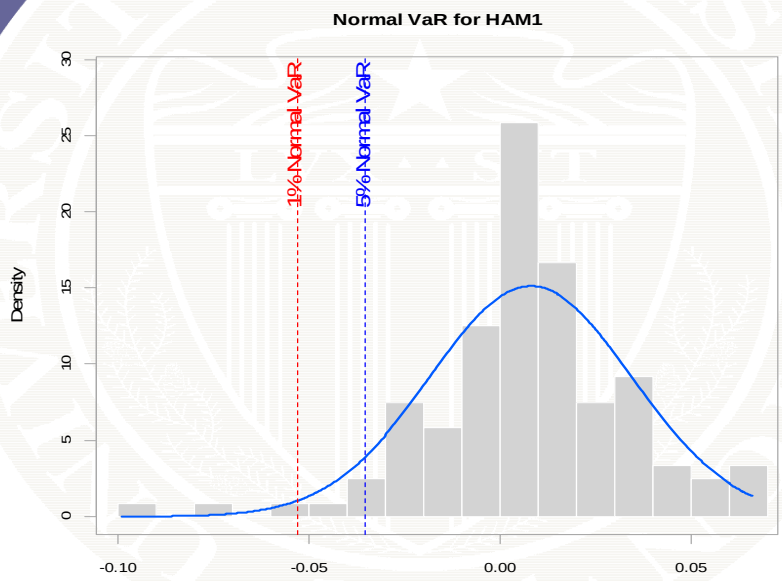
```
# use jarque.bera.test() function from tseries package
> jarque.bera.test(managers.df$HAM1)

        Jarque Bera Test

data:  managers.df$HAM1
X-squared = 33.7, df = 2, p-value = 4.787e-08
```

	HAM1	HAM2	HAM3	HAM4	HAM5	HAM6
statistic	33.7	85.4	67.2	5.88	15.2	1.02
P-value	0.000	0.000	0.000	0.053	0.000	0.602

Conclusion: All assets non-normal except HAM6

Normal VaR

```
# use VaR() function from PerformanceAnalytics package
> args(VaR)
function (R = NULL, p = 0.95, ..., method = c("modified",
  "gaussian", "historical", "kernel"), clean = c("none",
  "boudt", "geltner"), portfolio_method = c("single",
  "component", "marginal"), weights = NULL, mu = NULL,
  sigma = NULL, m3 = NULL, m4 = NULL, invert = TRUE)
# Normal 5% and 1% VaR
> VaR(managers.df[, manager.names], p=0.95, method="gaussian")
        HAM1    HAM2   HAM3    HAM4    HAM5    HAM6
VaR -0.0352 -0.0492 -0.052 -0.0827 -0.0732 -0.0298

> VaR(managers.df[, manager.names], p=0.99, method="gaussian")
        HAM1    HAM2    HAM3  HAM4   HAM5    HAM6
VaR -0.0531 -0.0736 -0.0768 -0.12 -0.104 -0.0459
```



Normal ETL (aka ES)

```
# use ES() function from PerformanceAnalytics package
> args(ES)
function (R = NULL, p = 0.95, ..., method = c("modified",
 "gaussian", "historical", "kernel"), clean = c("none",
 "boudt", "geltner"), portfolio_method = c("single",
 "component"), weights = NULL, mu = NULL, sigma = NULL,
  m3 = NULL, m4 = NULL, invert = TRUE, operational = TRUE)

# Normal 5% and 1% ETL
> ES(managers.df[, manager.names], p=0.95, method="gaussian")
      HAM1    HAM2    HAM3   HAM4    HAM5    HAM6
ES -0.0461 -0.0641 -0.0672 -0.106 -0.0922 -0.0396

> ES(managers.df[, manager.names], p=0.99, method="gaussian")
     HAM1    HAM2   HAM3   HAM4   HAM5    HAM6
ES -0.062 -0.0857 -0.089 -0.139 -0.120 -0.0539
```

FM Normal VaR

```
# compute FM means and standard deviations
> mu.factors = colMeans(managers.df[, factor.names])
> mu.fm = Betas%*%mu.factors
> sigma.fm = sqrt(diag(cov.fm))

# compute VaR using factorAnalytics function normalVaR
> args(normalVaR)
function (mu, sigma, tail.prob = 0.01, invert=FALSE)

> VaR.05.fm = t(normalVaR(mu.fm, sigma.fm, 0.05))
> VaR.01.fm = t(normalVaR(mu.fm, sigma.fm, 0.01))
> rbind(VaR.05.fm, VaR.01.fm)
        HAM1    HAM2    HAM3    HAM4   HAM5    HAM6
[1,] -0.0410 -0.0503 -0.0513 -0.0823 -0.071 -0.0369
[2,] -0.0591 -0.0750 -0.0762 -0.1201 -0.105 -0.0551
```


FM Normal ETL

```
# compute VaR using factorAnalytics function normalES
> args(normalES)
function (mu, sigma, tail.prob = 0.01, invert=FALSE)

> ES.05.fm = t(normalES(mu.fm, sigma.fm, 0.05))
> ES.01.fm = t(normalES(mu.fm, sigma.fm, 0.01))

> rbind(ES.05.fm, ES.01.fm)
        HAM1    HAM2    HAM3   HAM4    HAM5    HAM6
[1,] -0.0521 -0.0654 -0.0666 -0.105 -0.0916 -0.0481
[2,] -0.0681 -0.0872 -0.0886 -0.139 -0.1212 -0.0641
```

Cornish-Fisher VaR

```
# use VaR() function with method="modified" for
# Cornish-Fisher VaR
> VaR(managers.df[, manager.names], p=0.95,
      method="modified")
       HAM1    HAM2    HAM3   HAM4    HAM5    HAM6
VaR -0.0385 -0.0302 -0.0396 -0.088 -0.0708 -0.0317

> VaR(managers.df[, manager.names], p=0.99,
      method="modified")
       HAM1    HAM2    HAM3   HAM4   HAM5    HAM6
VaR -0.0753 -0.0262 -0.0674 -0.142 -0.127 -0.0482
```


Cornish-Fisher ETL

```
# use ES() function with method="modified" for
# Cornish-Fisher ES
> ES(managers.df[, manager.names], p=0.95,
+    method="modified")
       HAM1    HAM2    HAM3   HAM4   HAM5    HAM6
ES -0.0654 -0.0662 -0.0453 -0.123 -0.102 -0.0412

> ES(managers.df[, manager.names], p=0.99,
+    method="modified")
       HAM1   HAM2    HAM3   HAM4   HAM5    HAM6
ES -0.0816 -0.542 -0.0674 -0.178 -0.173 -0.0561
```

Nonparametric VaR

```
# nonparametric VaR is based on empirical quantile
> quantile(managers.df$HAM1, probs=c(0.01, 0.05))
     1%      5%
-0.0736 -0.0309

# use VaR() function with method="historical"
> VaR(managers.df[, manager.names], p=0.95,
+     method="historical")
       HAM1    HAM2    HAM3   HAM4    HAM5    HAM6
VaR -0.0309 -0.0340 -0.0456 -0.086 -0.0752 -0.0357

> VaR(managers.df[, manager.names], p=0.99,
+     method="historical")
       HAM1    HAM2    HAM3   HAM4   HAM5    HAM6
VaR -0.0736 -0.0390 -0.0656 -0.137 -0.117 -0.0426
```


Nonparametric ETL

```
# nonparametric ETL is sample mean below nonparametric
# VaR
> q.hat.05 = quantile(managers.df$HAM1, probs=0.05)
> smpl = managers.df$HAM1 <= q.hat.05
> mean(managers.df$HAM1[smpl])
[1] -0.0577

# use ES() function with method="historical"
> ES(managers.df[, manager.names], p=0.95,
+    method="historical")
      HAM1    HAM2    HAM3   HAM4   HAM5    HAM6
ES -0.0577 -0.0375 -0.0608 -0.120 -0.107 -0.0411

> ES(managers.df[, manager.names], p=0.99,
+    method="historical")
     HAM1    HAM2    HAM3   HAM4   HAM5   HAM6
ES -0.088 -0.0403 -0.0714 -0.161 -0.135 -0.043
```

VaR Summary

Fund	Normal	FM Normal	Modified	Empirical
		5% VaR		
HAM1	**-3.5**	**-4.1**	**-3.9**	**-3.1**
HAM2	**-4.9**	**-5.0**	**-3.0**	**-3.4**
HAM3	**-5.2**	**-5.1**	**-4.0**	**-4.6**
HAM4	**-8.3**	**-8.2**	**-8.8**	**-8.6**
HAM5	**-7.3**	**-7.1**	**-7.1**	**-7.5**
HAM6	**-3.0**	**-3.7**	**-3.1**	**-3.6**
		1% VaR		
HAM1	**-5.3**	**-5.9**	**-7.5**	**-7.4**
HAM2	**-7.4**	**-7.5**	**-2.6**	**-3.9**
HAM3	**-7.7**	**-7.6**	**-6.7**	**-6.6**
HAM4	**-12.0**	**-12.0**	**-14.2**	**-13.7**
HAM5	**-10.4**	**-10.5**	**-12.7**	**-11.7**
HAM6	**-4.6**	**-5.5**	**-4.8**	**-4.3**

ETL Summary

Fund	Normal	FM Normal	Modified	Empirical
		5% ETL ×100		
HAM1	**-4.6**	**-5.2**	**-6.5**	**-5.8**
HAM2	**-6.4**	**-6.5**	**-6.6**	**-3.8**
HAM3	**-6.7**	**-6.7**	**-4.5**	**-6.1**
HAM4	**-10.1**	**-10.5**	**-12.3**	**-12.0**
HAM5	**-9.2**	**-9.2**	**-10.2**	**-10.7**
HAM6	**-3.7**	**-4.8**	**-4.1**	**-4.1**
		1% ETL×100		
HAM1	**-6.2**	**-6.8**	**-8.2**	**-8.8**
HAM2	**-8.6**	**-8.7**	**-54.2**	**-4.0**
HAM3	**-8.9**	**-8.9**	**-6.7**	**-7.1**
HAM4	**-13.9**	**-13.9**	**-17.8**	**-16.1**
HAM5	**-12.0**	**-12.1**	**-17.3**	**-13.5**
HAM6	**-5.4**	**-6.4**	**-5.6**	**-4.3**

VaR and ETL for Portfolio

	Normal	FM Normal	Modified	Empirical
5% VaR	-3.2	-3.8	-4.4	-2.7
1% VaR	-4.8	-5.7	-5.3	-5.1
5% ETL	-4.2	-5.0	-4.6	-4.3
1% ETL	-5.6	-6.6	-6.4	-6.4

Factor Risk Budgeting

```
# use factorModelFactorSdDecomposition() function
# from factorAnalytics package
> args(factorModelFactorSdDecomposition)
function (beta.vec, factor.cov, sig2.e)

# Compute factor SD decomposition for HAM1
> factor.sd.decomp.HAM1 =
factorModelFactorSdDecomposition(Betas["HAM1",],
+                cov.factors, ResidVars["HAM1"])

> names(factor.sd.decomp.HAM1)
[1] "sd.fm"  "mcr.fm" "cr.fm"  "pcr.fm"
```

Factor Contributions to SD

```
> factor.sd.decomp.HAM1
$sd.fm
[1] 0.0265

$mcr.fm
    EDHEC.LS.EQ SP500.TR US.10Y.TR residual
MCR      0.0119   0.0295  -0.00638    0.711

$cr.fm
   EDHEC.LS.EQ SP500.TR US.10Y.TR residual
CR     0.00318  0.00847   0.00147   0.0134

$pcr.fm
    EDHEC.LS.EQ SP500.TR US.10Y.TR residual
PCR        0.12    0.319    0.0553    0.506
```

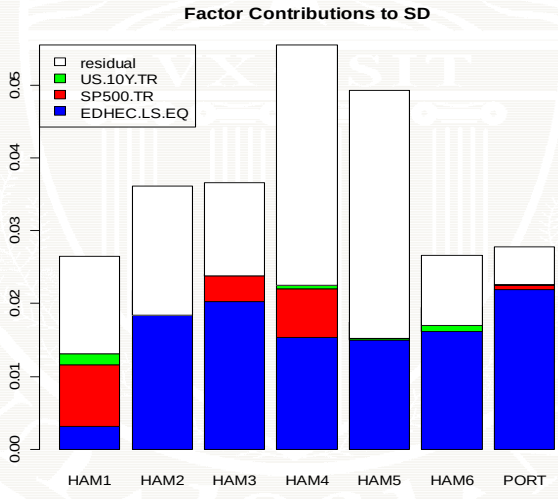

Factor Contributions to SD

```
# loop over all assets and store results in list
> factor.sd.decomp.list = list()
> for (i in manager.names) {
+  factor.sd.decomp.list[[i]] =
factorModelFactorSdDecomposition(Betas[i,],
+                cov.factors, ResidVars[i])
+ }

# add portfolio factor SD decomposition to list
> factor.sd.decomp.list[["PORT"]] =
factorModelFactorSdDecomposition(beta.p,
+             cov.factors, var.p.resid)

> names(factor.sd.decomp.list)
[1] "HAM1" "HAM2" "HAM3" "HAM4" "HAM5" "HAM6" "PORT"
```

Factor Contributions to SD

```
# function to extract contribution to sd from list
> getCSD = function(x) {
+   x$cr.fm
+}

# extract contributions to SD from list
> cr.sd = sapply(factor.sd.decomp.list, getCSD)
> rownames(cr.sd) = c(factor.names, "residual")


# create stacked barchart
> barplot(cr.sd, main="Factor Contributions to SD",
+         legend.text=T, args.legend=list(x="topleft"),
+         col=c("blue","red","green","white"))
```


Factor Contributions to SD



Factor Contributions to ETL

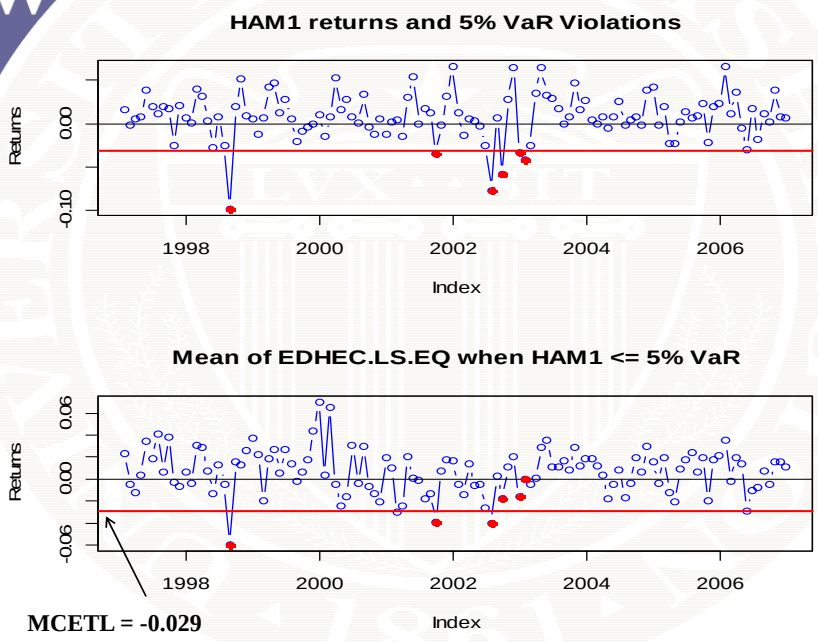
```
# first combine HAM1 returns, factors and std residuals
> tmpData = cbind(managers.df[,1],
+                 managers.df[,factor.names],
+           residuals(reg.list[[1]])/sqrt(ResidVars[1]))
> colnames(tmpData)[c(1,5)] = c(manager.names[1],
                                "residual")

> factor.es.decomp.HAM1 =
factorModelFactorEsDecomposition(tmpData, Betas[1,],
+                      ResidVars[1], tail.prob=0.05)
> names(factor.es.decomp.HAM1)
[1] "VaR.fm"     "n.exceed"   "idx.exceed" "ES.fm"
[5] "mcES.fm"    "cES.fm"     "pcES.fm"
```

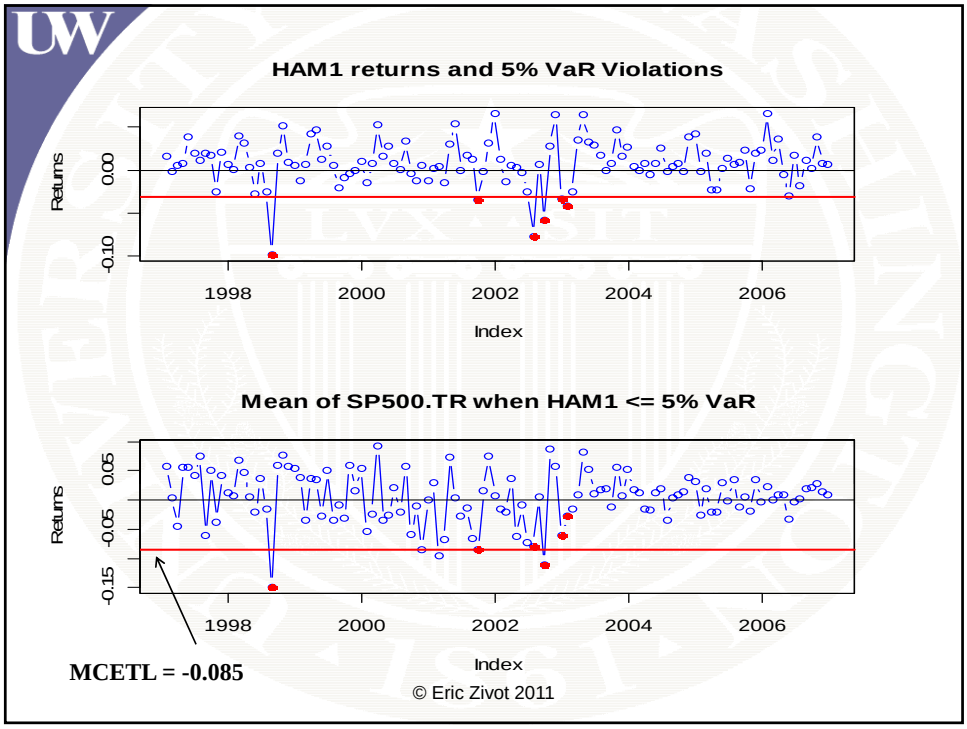

Factor Contributions to ETL

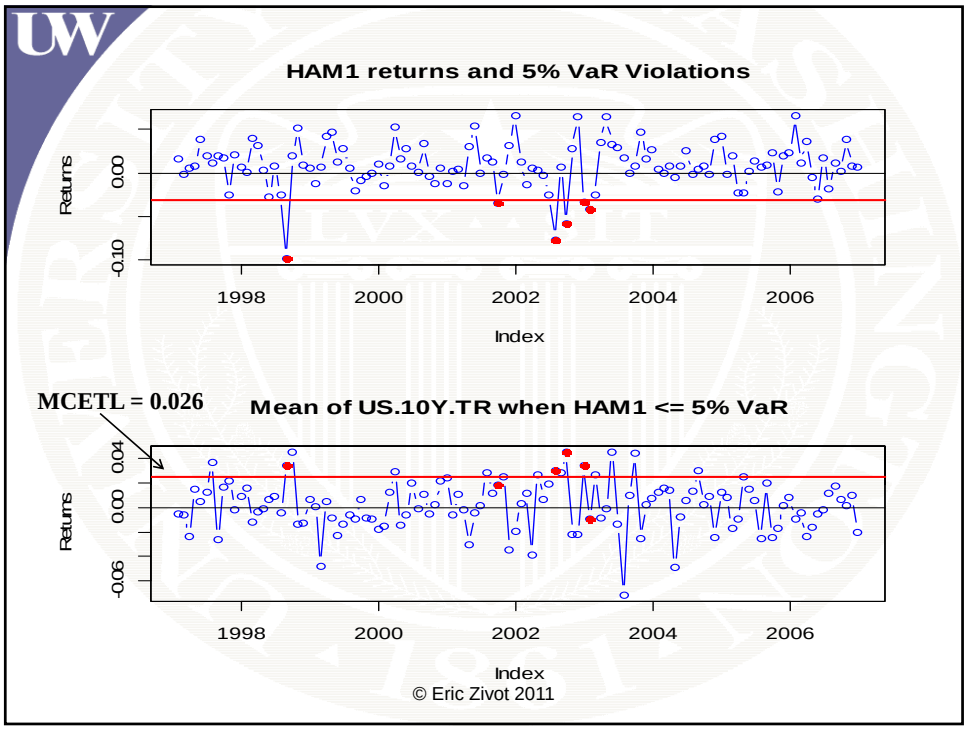
```
> factor.es.decomp.HAM1
$VaR.fm
    5%
0.0309

$n.exceed
[1] 6

$idx.exceed
1998-08-30 2001-09-29 2002-07-30 2002-09-29 2002-12-30
        20         57         67         69         72
2003-01-30
        73
```

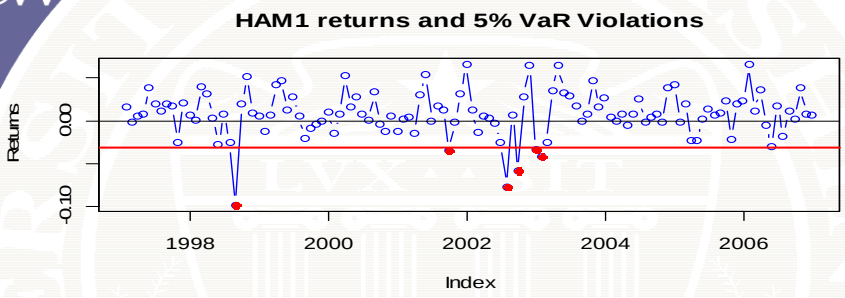
Factor Contributions to ETL

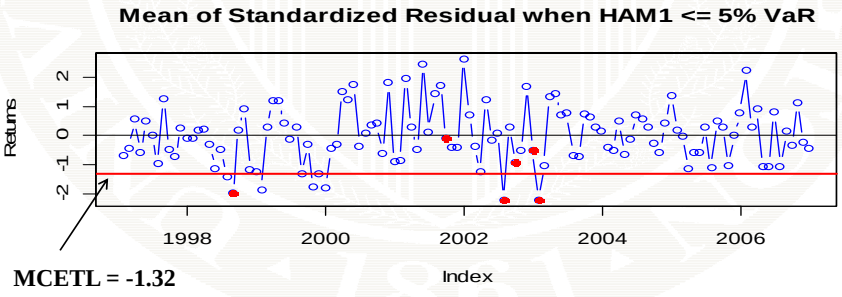
```
$ES.fm
[1] 0.0577

$mcES.fm
     EDHEC.LS.EQ SP500.TR US.10Y.TR residual
MCES      0.0289   0.0852   -0.0255     1.32

$cES.fm
    EDHEC.LS.EQ SP500.TR US.10Y.TR residual
CES     0.00774   0.0245   0.00587    0.025

$pcES.fm
     EDHEC.LS.EQ SP500.TR US.10Y.TR residual
PCES       0.134    0.424     0.102    0.433
```


HAM1 returns and 5% VaR Violations
Returns
0.00
-0.10
1998
2000
2002
2004
2006
Index
Mean of SP500.TR when HAM1 <= 5% VaR
Returns
0.05
-0.05
-0.15
1998
2000
2002
2004
2006
Index
MCETL = -0.085
© Eric Zivot 2011

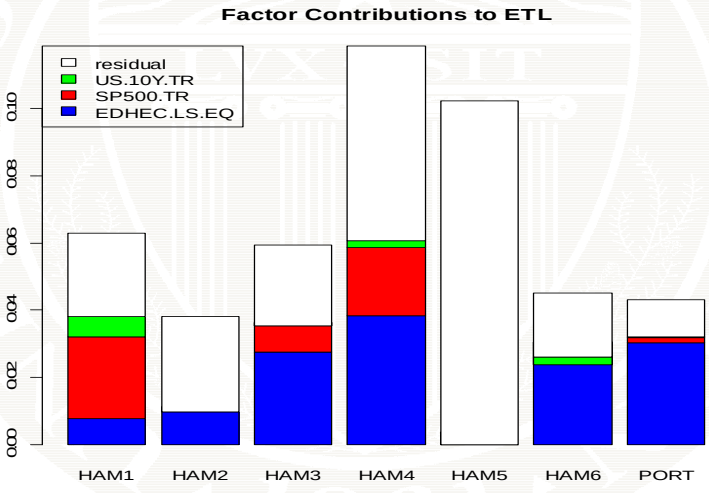

HAM1 returns and 5% VaR Violations
Returns
0.00
-0.10
1998
2000
2002
2004
2006
Index
MCETL = 0.026
Mean of US.10Y.TR when HAM1 <= 5% VaR
Returns
0.04
0.00
-0.06
1998
2000
2002
2004
2006
Index
© Eric Zivot 2011





MCETL = -1.32

Factor Contributions to ETL



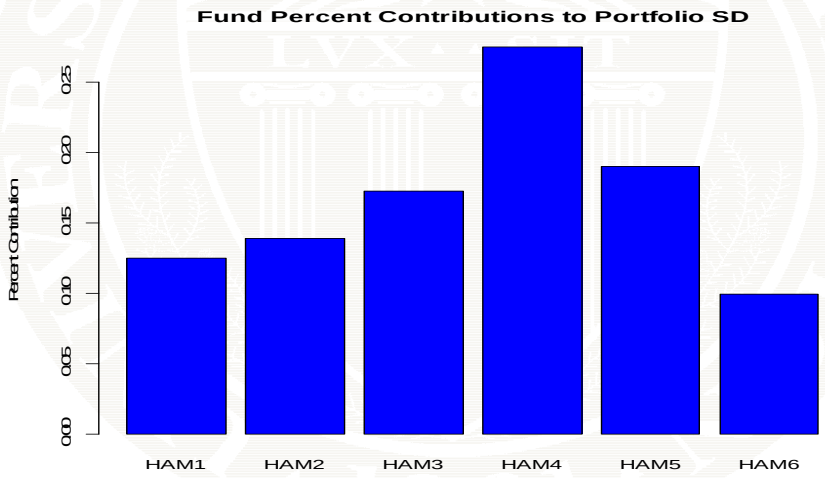
Portfolio Risk Budgeting

```
# use portfolioSdDecomposition() function from
# factorAnalytics package
> args(portfolioSdDecomposition)
function (w.vec, cov.assets)
# compute with sample covariance matrix (pairwise
# complete observations)
> cov.sample = cov(managers.df[,manager.names],
+                  use="pairwise.complete.obs")
> port.sd.decomp.sample =
+ portfolioSdDecomposition(w.vec, cov.sample)

> names(port.sd.decomp.sample)
[1] "sd.p"   "mcsd.p" "csd.p"  "pcsd.p"
```


Portfolio SD Decomposition

```
> port.sd.decomp.sample
$sd.p
[1] 0.0261

$mcsd.p
       HAM1   HAM2   HAM3   HAM4   HAM5   HAM6
MCSD 0.0196 0.0218 0.0270 0.0431 0.0298 0.0155

$csd.p
       HAM1    HAM2    HAM3    HAM4    HAM5    HAM6
CSD 0.00327 0.00363 0.00451 0.00718 0.00497 0.00259

$pcsd.p
      HAM1  HAM2  HAM3  HAM4 HAM5  HAM6
PCSD 0.125 0.139 0.172 0.275 0.19 0.099
```

Fund Contributions to Portfolio SD



Portfolio ETL Decomposition

```
# use ES() function in PerformanceAnalytics package
> port.ES.decomp =
ES(na.omit(managers.df[,manager.names]),
+                    p=0.95, method="historical",
+                    portfolio_method = "component",
+                    weights = w.vec)
> port.ES.decomp
$`-r_exceed/c_exceed`
[1] 0.0479

$c_exceed
[1] 3

$realizedcontrib
  HAM1   HAM2   HAM3   HAM4   HAM5   HAM6
0.1874 0.0608 0.1479 0.3518 0.1886 0.0635
```

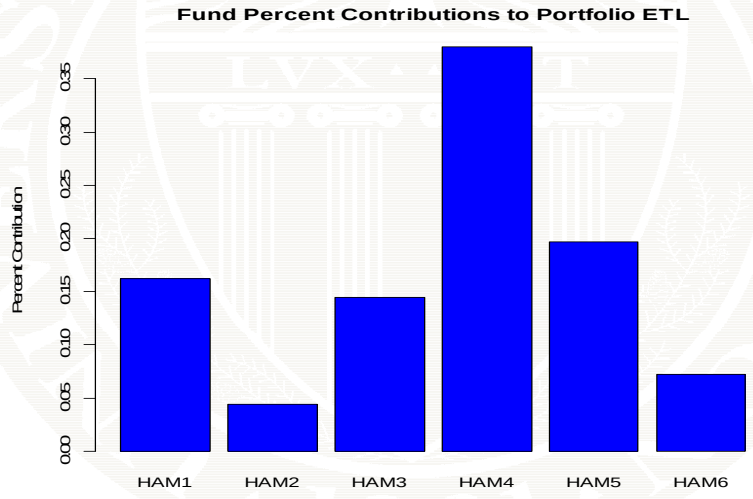
Portfolio ETL Decomposition

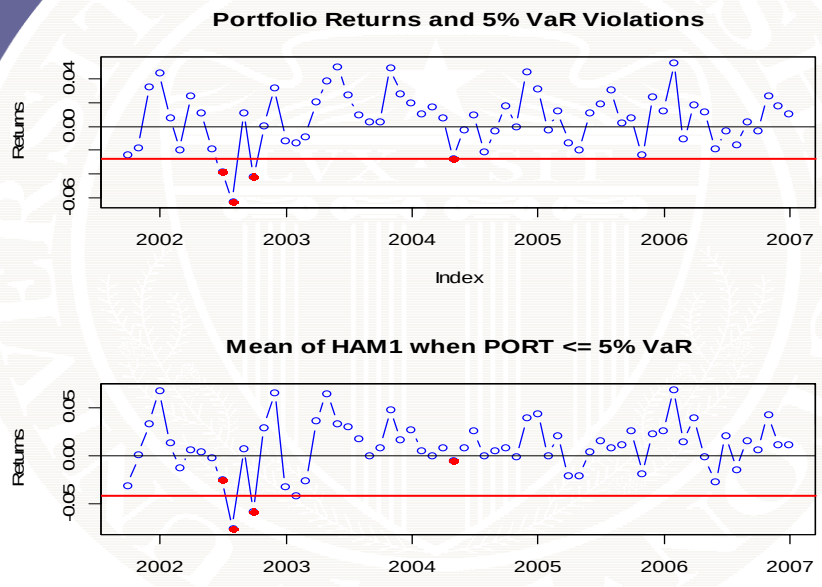
```
# use portfolioEsDecomposition from factorAnalytics
# package.
> args(portfolioEsDecomposition)
function (bootData, w, delta.w = 0.001, tail.prob =
    0.01, method = c("derivative",
    "average"), VaR.method = c("HS", "CornishFisher"))

> port.ES.decomp =
portfolioEsDecomposition(na.omit(managers.df[,manager.
                        names]),w.vec, tail.prob=0.05)

> names(port.ES.decomp)
[1] "VaR.fm"     "ES.fm"      "n.exceed"   "idx.exceed"
[5] "MCES"       "CES"        "PCES"
```

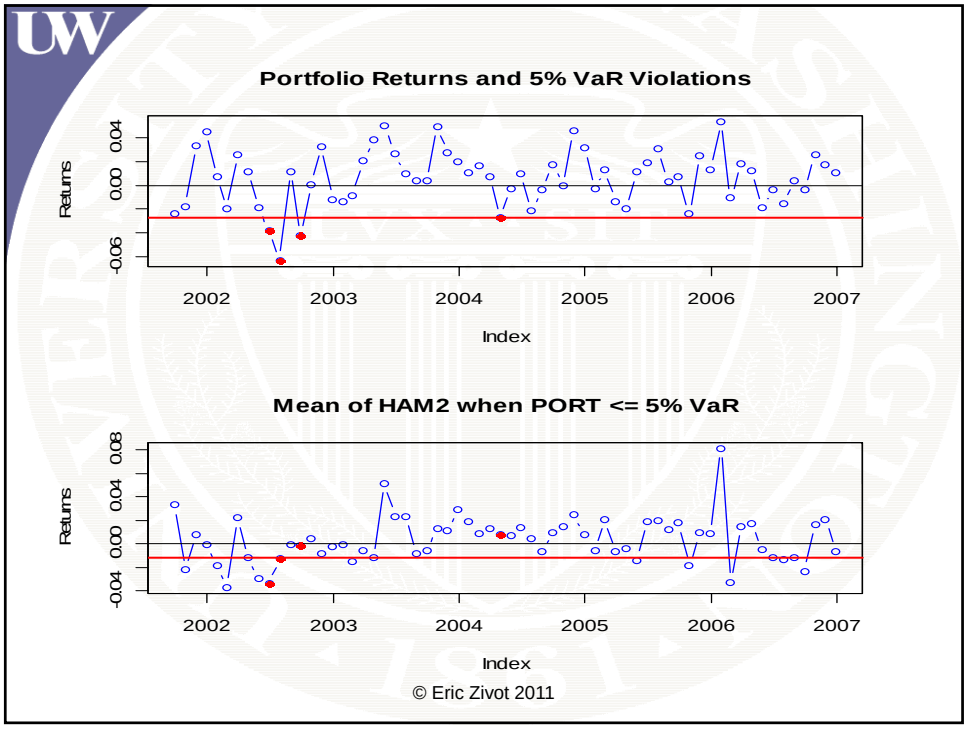

Portfolio ETL Decomposition

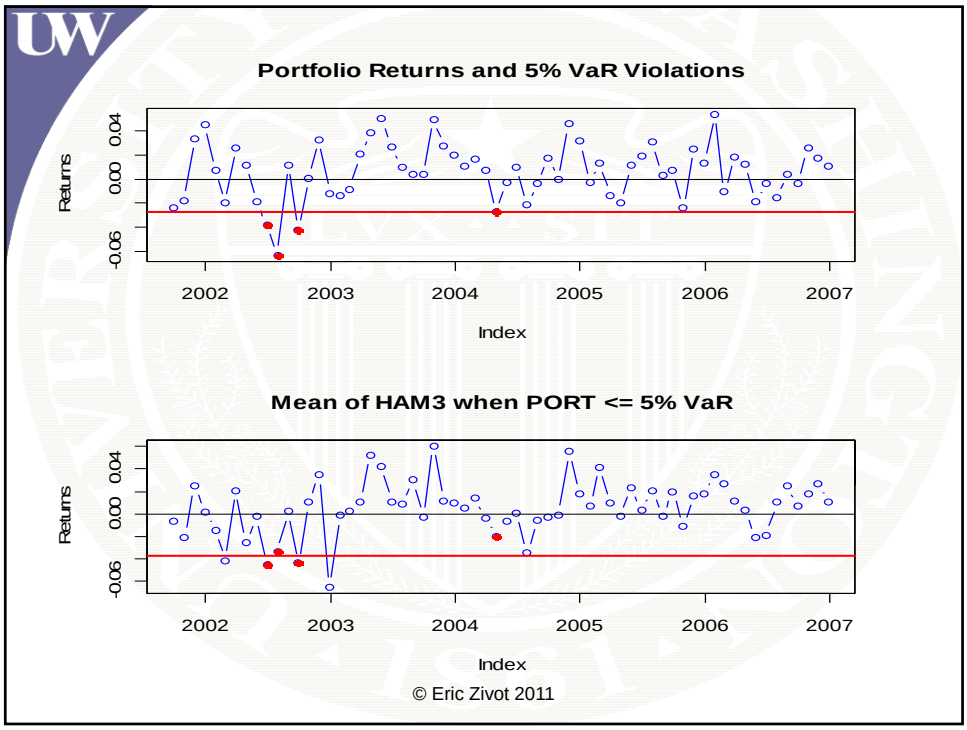
```
> port.ES.decomp
$VaR.fm     $ES.fm      $n.exceed   $idx.exceed
     5%     [1] 0.0428  [1] 4       [1] 10 11 13 32
 0.0269

$MCES
       HAM1   HAM2   HAM3   HAM4   HAM5   HAM6
MCES 0.0417 0.0113 0.0371 0.0976 0.0505 0.0186

$CES
       HAM1    HAM2    HAM3   HAM4    HAM5    HAM6
CES 0.00695 0.00188 0.00618 0.0163 0.00842 0.00310

$PCES
      HAM1   HAM2  HAM3 HAM4  HAM5   HAM6
PCES 0.162 0.0439 0.145 0.38 0.197 0.0724
```

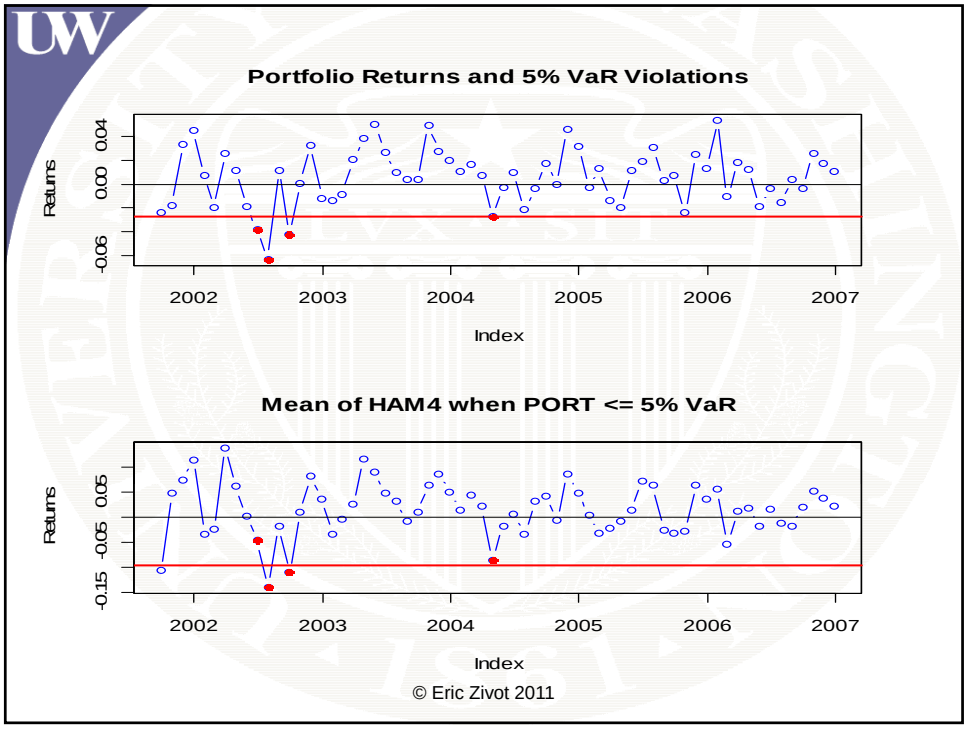
Fund Contributions to Portfolio ETL

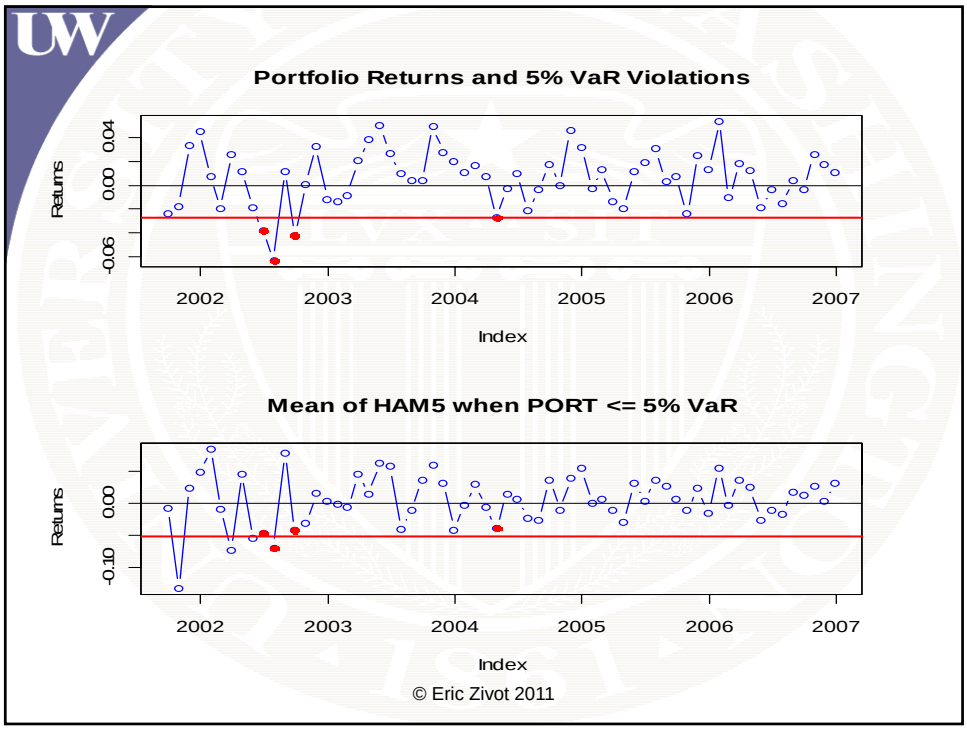


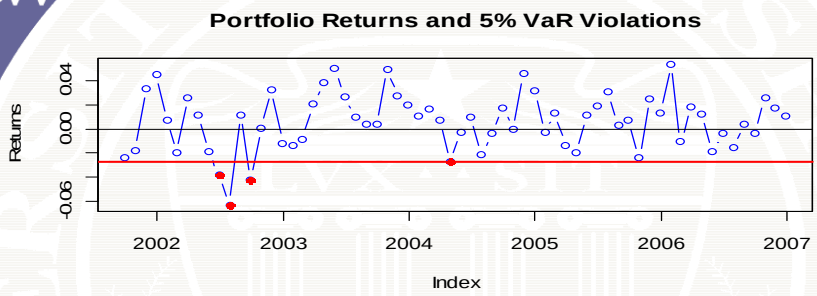


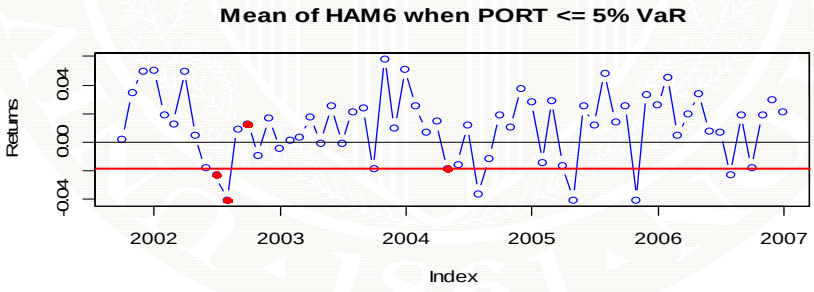

Portfolio Returns and 5% VaR Violations
Returns
0.04
0.00
-0.06
2002
2003
2004
2005
2006
2007
Index
Mean of HAM2 when PORT <= 5% VaR
Returns
0.08
0.04
0.00
-0.04
2002
2003
2004
2005
2006
2007
Index
© Eric Zivot 2011


Portfolio Returns and 5% VaR Violations
Returns
0.04
0.00
-0.06
2002
2003
2004
2005
2006
2007
Index
Mean of HAM3 when PORT <= 5% VaR
Returns
0.04
0.00
-0.06
2002
2003
2004
2005
2006
2007
Index
© Eric Zivot 2011









Factor Model Monte Carlo (FMMC)

```
# resample from historical factors
> n.boot = 5000

# set random number seed
> set.seed(123)

# n.boot reshuffled indices
> bootIdx = sample(nrow(managers.df),  n.boot,
+                  replace=TRUE)

# resampled factor data
> factorDataBoot.mat = as.matrix(managers.df[bootIdx,
+                                factor.names])
```

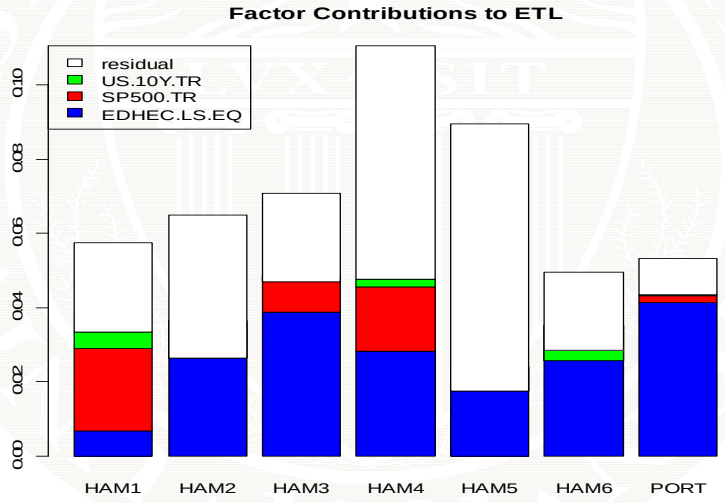
FMMC with Normal Residuals

```
# FMMC using normal distribution for residuals and
# alpha = 0
> returns.boot = matrix(0, n.boot, length(manager.names))
> resid.sim = matrix(0, n.boot, length(manager.names))
> colnames(returns.boot) = colnames(resid.sim) =
+                          manager.names
# FMMC loop
for (i in manager.names) {
   returns.fm = factorDataBoot.mat%*%Betas[i, ]
   resid.sim[, i] = rnorm(n.boot,sd=sqrt(ResidVars[i]))
   returns.boot[, i] = returns.fm + resid.sim[, i]
}

# compute portfolio return and fm residual
> return.p.boot = returns.boot%*%wvec
> resid.fm.p = resid.sim%*%w.vec
```


FMMC Factor Contribution to ETL

```
# compute decomposition in loop
> factor.es.decomp.list = list()
> for (i in manager.names) {
+ tmpData = cbind(returns.boot[, i], factorDataBoot.mat,
+                 resid.sim[, i]/sqrt(ResidVars[i]))
> colnames(tmpData)[c(1,5)] = c(manager.names[i], "residual")
> factor.es.decomp.list[[i]] =
  factorModelFactorEsDecomposition(tmpData, Betas[i,],
+                       ResidVars[i], tail.prob=0.05)
}
# add portfolo retsults - need factor model residuals
> tmpData = cbind(r.p.boot, factorDataBoot.mat,
+                 resid.fm.p/sqrt(as.numeric(var.p.resid)))
> colnames(tmpData)[c(1,5)] = c("PORT", "residual")
> factor.es.decomp.list[["PORT"]] =
factorModelFactorEsDecomposition(tmpData, beta.p,
+                     var.p.resid, tail.prob=0.05)
```

FMMC Factor Contributions to ETL



FMMC Fund Contribution to Portfolio ETL

```
> port.ES.decomp.fmmc =
portfolioEsDecomposition(returns.boot,
+                         w.vec, tail.prob=0.05)
```

FMMC Fund Contributions to Portfolio ETL