Exploratory Data Analysis in Finance Using PerformanceAnalytics

Brian G. Peterson & Peter Carl

[1] Diamond Management & Technology Consultants
Chicago, IL
brian@braverock.com

[2] Guidance Capital
Chicago, IL
peter@braverock.com

UseR! International User and Developer Conference, Ames, Iowa, 8-10 Aug 2007

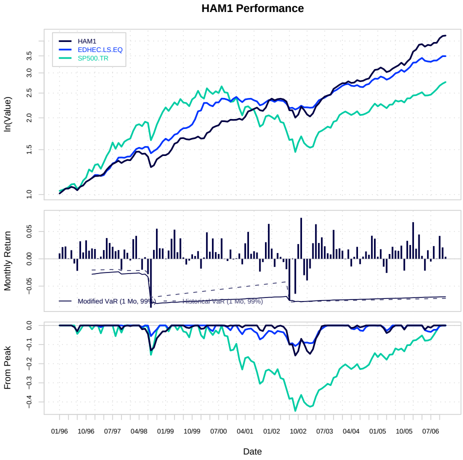
Outline

Overview

- Exploratory data analysis with finance data often starts with visual examination to:
  - examine properties of asset returns
  - compare an asset to other similar assets
  - compare an asset to one or more benchmarks
- Application of performance and risk measures can build a set of statistics for comparing possible investments
- Examples are developed using data for six (hypothetical) managers, a peer index, and an asset class index
- Hypothetical manager data was developed from real manager timeseries using *accuracy* and *perturb* packages to disguise the data while maintaining some of the statistical properties of the original data.

Draw a Performance Summary Chart.

```
> charts.PerformanceSummary(managers[, c(manager.col, indexes.cols)],
+     colorset = rich6equal, lwd = 2, ylog = TRUE)
```



Show Calendar Performance.

```
> t(table.CalendarReturns(managers[, c(manager.col, indexes.cols)]))

             1996 1997 1998 1999 2000  2001  2002 2003 2004 2005 2006
Jan           1.0  1.8 -0.3  0.0 -1.8   0.1   1.9 -4.0  1.5  0.4  6.7
Feb           2.1  0.1  3.6  1.5  0.2   1.0  -1.5 -1.8 -0.1  1.8  1.8
Mar           2.3  0.4  4.2  3.7  4.9  -1.0   1.1  2.9  1.7 -1.4  4.5
Apr           0.1  1.6  0.1  5.3  1.3   2.8   0.4  6.3 -1.4 -2.6  0.5
May           1.6  3.8 -2.0  1.2  3.7   4.9  -0.6  2.9  0.4  0.9 -2.2
Jun          -0.9  2.9  0.3  3.8  1.2   0.9  -1.9  3.9  2.2  2.2  1.6
Jul          -2.2  2.2 -2.8  0.2  0.9   1.4  -7.6  2.3 -1.0  1.5 -0.5
Aug           3.2  1.4 -8.9 -1.1  3.8   1.2   0.0  1.0  0.4  1.5  2.3
Sep           1.2  1.6  1.6 -0.3  0.0  -2.3  -6.4  0.8  1.4  2.4  0.0
Oct           3.4 -2.0  5.5  0.8 -0.4  -0.6   2.7  5.3  0.7 -2.2  4.2
Nov           1.5  1.7  1.9  0.5  1.7   3.0   7.5  1.8  4.2  3.3  2.1
Dec           1.9  1.1  1.9  1.4 -0.1   6.4  -3.0  1.9  3.7  2.5  0.4
HAM1         16.1 17.8  4.4 18.3 16.2  18.9  -8.1 25.5 14.4 10.5 23.3
EDHEC.LS.EQ    NA 21.4 14.6 31.4 12.0  -1.2  -6.4 19.3  8.6 11.3 10.1
SP500.TR     23.0 33.4 28.6 21.0 -9.1 -11.9 -22.1 28.7 10.9  4.9 15.8
```

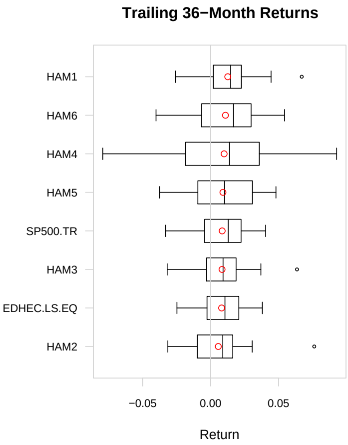
Calculate Statistics.

```
> table.MonthlyReturns(managers[, c(manager.col, peers.cols)])

                    HAM1     HAM2     HAM3     HAM4    HAM5    HAM6
Observations    132.0000 125.0000 132.0000 132.0000 77.0000 64.0000
NAs               0.0000   7.0000   0.0000   0.0000 55.0000 68.0000
Minimum          -0.0895  -0.0429  -0.0738  -0.1800 -0.1386 -0.0402
Quartile 1        0.0000  -0.0105  -0.0066  -0.0213 -0.0184 -0.0034
Median            0.0132   0.0060   0.0107   0.0139  0.0045  0.0146
Arithmetic Mean   0.0112   0.0138   0.0122   0.0105  0.0034  0.0121
Geometric Mean    0.0109   0.0131   0.0115   0.0091  0.0025  0.0118
Quartile 3        0.0231   0.0248   0.0312   0.0440  0.0298  0.0276
Maximum           0.0750   0.1521   0.1774   0.1583  0.1660  0.0544
SE Mean           0.0022   0.0033   0.0032   0.0047  0.0051  0.0030
LCL Mean (0.95)   0.0069   0.0072   0.0058   0.0013 -0.0067  0.0062
UCL Mean (0.95)   0.0156   0.0203   0.0186   0.0197  0.0136  0.0180
Variance          0.0006   0.0014   0.0014   0.0029  0.0020  0.0006
Stdev             0.0251   0.0369   0.0371   0.0536  0.0447  0.0238
Skewness         -0.6871   1.4564   0.8091  -0.4198 -0.0131 -0.2312
Kurtosis          2.4001   2.4099   2.3632   0.8703  2.1288 -0.5305
```

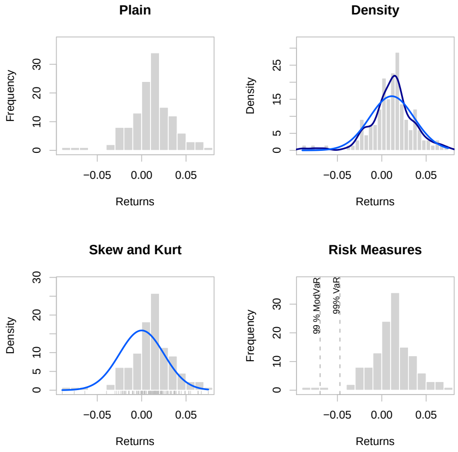
Compare Distributions.

```
> chart.Boxplot(managers[trailing36.rows, c(manager.col, peers.cols,
+     indexes.cols)], main = "Trailing 36-Month Returns")
```



Compare Distributions.

```
> layout(rbind(c(1, 2), c(3, 4)))
> chart.Histogram(managers[, 1, drop = F], main = "Plain", methods = NULL)
> chart.Histogram(managers[, 1, drop = F], main = "Density", breaks = 40,
+     methods = c("add.density", "add.normal"))
> chart.Histogram(managers[, 1, drop = F], main = "Skew and Kurt",
+     methods = c("add.centered", "add.rug"))
> chart.Histogram(managers[, 1, drop = F], main = "Risk Measures",
+     methods = c("add.risk"))
```

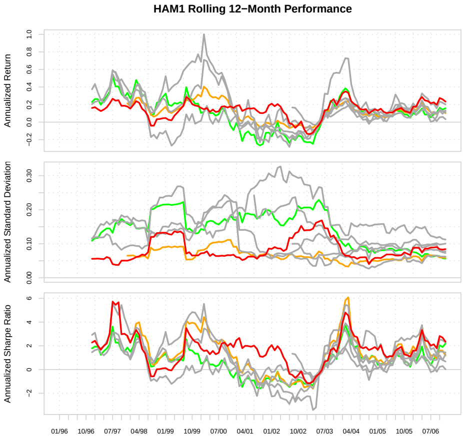


Show Relative Return and Risk.

```
> chart.RiskReturnScatter(managers[trailing36.rows, 1:8], rf = 0.03/12,
+     main = "Trailing 36-Month Performance", colorset = c("red",
+         rep("black", 5), "orange", "green"))
```

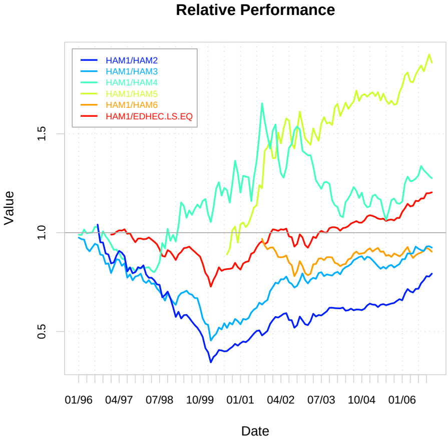
Examine Performance Consistency.

```
> charts.RollingPerformance(managers[, c(manager.col, peers.cols,
+     indexes.cols)], rf = 0.03/12, colorset = c("red", rep("darkgray",
+     5), "orange", "green"), lwd = 2)
```



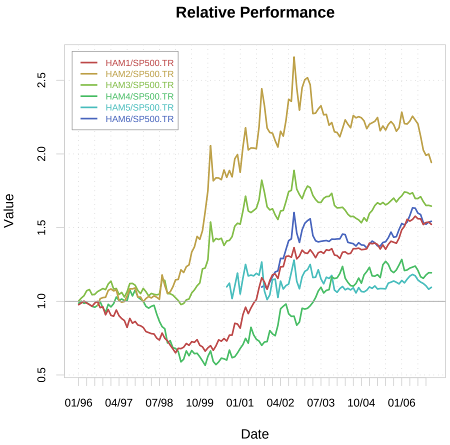
Display Relative Performance.

```
> chart.RelativePerformance(managers[, manager.col, drop = FALSE],
+     managers[, c(peers.cols, 7)], colorset = tim8equal[-1], lwd = 2,
+     legend.loc = "topleft")
```



Compare to a Benchmark.

```
> chart.RelativePerformance(managers[, c(manager.col, peers.cols)],
+     managers[, 8, drop = F], colorset = rainbow8equal, lwd = 2,
+     legend.loc = "topleft")
```



Compare to a Benchmark.

```
> table.CAPM(managers[trailing36.rows, c(manager.col, peers.cols)],
+     managers[trailing36.rows, 8, drop = FALSE], rf = managers[trailing36.rows,
+         Rf.col, drop = F])

                    HAM1 to SP500.TR HAM2 to SP500.TR HAM3 to SP500.TR
Alpha                         0.0061           0.0006           0.0015
Beta                          0.6713           0.4178           0.7349
R-squared                     0.4397           0.1715           0.5907
Annualized Alpha              0.0755           0.0076           0.0180
Correlation                   0.6631           0.4142           0.7686
Correlation p-value           0.0000           0.0120           0.0000
Tracking Error                0.0868           0.0601           0.0021
Active Premium                0.0538          -0.0359          -0.0010
Information Ratio             0.6201          -0.5974          -0.4973
Treynor Ratio                 0.1870           0.0857           0.0962
                    HAM4 to SP500.TR HAM5 to SP500.TR HAM6 to SP500.TR
Alpha                         0.0005           0.0015           0.0033
Beta                          1.1570           0.8442           0.8574
R-squared                     0.3697           0.4887           0.4830
Annualized Alpha              0.0059           0.0181           0.0399
Correlation                   0.6080           0.6991           0.6950
Correlation p-value           0.0001           0.0000           0.0000
Tracking Error                0.0302           0.0119           0.0508
Active Premium                0.0120           0.0061           0.0299
Information Ratio             0.3984           0.5148           0.5889
Treynor Ratio                 0.0724           0.0922           0.1186
```

Calculate Returns.

- The single-period arithmetic return, or simple return, can be calculated as

$$R_t = \frac{P_t}{P_{t-1}} - 1 = \frac{P_t - P_{t-1}}{P_{t-1}} \tag{1}$$

- Simple returns, cannot be added together. A multiple-period simple return is calculated as:

$$R_t = \frac{P_t}{P_{t-k}} - 1 = \frac{P_t - P_{t-k}}{P_{t-k}} \tag{2}$$

- The natural logarithm of the simple return of an asset is referred to as the continuously compounded return, or *log return*:

$$r_t = ln(1 + R_t) = ln\frac{P_t}{P_{t-1}} = p_t - p_{t-1} \tag{3}$$

- Calculating log returns from simple gross return, or vice versa:

$$r_t = ln(1 + R_t), R_t = exp(r_t) - 1. \tag{4}$$

- *Return.calculate* or *CalculateReturns* (now deprecated) may be used to compute discrete and continuously compounded returns for data containing asset prices.

table.CAPM underlying techniques

- Return.annualized — Annualized return using

$$prod(1+R_a)^{\frac{scale}{n}} - 1 = \sqrt[n]{prod(1+R_a)^{scale}} - 1 \tag{5}$$

- TreynorRatio — ratio of asset's Excess Return to Beta β of the benchmark

$$\frac{(\overline{R_a - R_f})}{\beta_{a,b}} \tag{6}$$

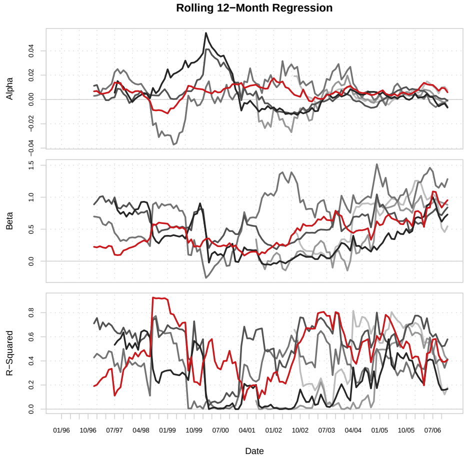
- ActivePremium — investment's annualized return minus the benchmark's annualized return
- Tracking Error — A measure of the unexplained portion of performance relative to a benchmark, given by

$$TrackingError = \sqrt{\sum \frac{(R_a - R_b)^2}{len(R_a)\sqrt{scale}}} \tag{7}$$

- InformationRatio — ActivePremium/TrackingError

Compare to a Benchmark.

```
> charts.RollingRegression(managers[, c(manager.col, peers.cols),
+     drop = FALSE], managers[, 8, drop = FALSE], rf = 0.03/12,
+     colorset = redfocus, lwd = 2)
```



Calculate Downside Risk.

```
> table.DownsideRisk(managers[, 1:6], rf = 0.03/12)
```

	HAM1	HAM2	HAM3	HAM4	HAM5	HAM6
Semi Deviation	0.0277	0.0266	0.0331	0.0584	0.0469	0.0252
Gain Deviation	0.0164	0.0347	0.0296	0.0314	0.0298	0.0157
Loss Deviation	0.0209	0.0099	0.0187	0.0371	0.0321	0.0132
Downside Deviation (MAR=10%)	0.0276	0.0235	0.0318	0.0572	0.0480	0.0227
Downside Deviation (rf=3%)	0.0275	0.0197	0.0309	0.0564	0.0462	0.0229
Downside Deviation (0%)	0.0289	0.0185	0.0295	0.0550	0.0457	0.0214
Maximum Drawdown	-0.1573	-0.2240	-0.2786	-0.2913	-0.3775	-0.0707
VaR (99%)	0.0471	0.0721	0.0741	0.1142	0.1006	0.0432
Beyond VaR	0.0477	0.0731	0.0750	0.1154	0.1009	0.0437
Modified VaR (99%)	0.0695	0.0239	0.0634	0.1380	0.1232	0.0438

Semivariance and Downside Deviation

- Downside Deviation as proposed by Sharpe is a generalization of semivariance which calculates bases on the deviation below a Minimumn Acceptable Return(MAR)

$$\delta_{MAR} = \sqrt{\frac{\sum_{t=1}^{n}(R_t - MAR)^2}{n}} \tag{8}$$

- Downside Deviation may be used to calculate semideviation by setting MAR=mean(R) or may also be used with MAR=0
- Downside Deviation (and its special cases semideviation and semivariance) is useful in several performance to risk ratios, and in several portfolio optimization problems.

Value at Risk

- Value at Risk (VaR) has become a required standard risk measure recognized by Basel II and MiFID
- Traditional mean-VaR may be derived historically, or estimated parametrically using

$$z_c = q_p = qnorm(p) \tag{9}$$

$$VaR = \bar{R} - z_c \cdot \sqrt{\sigma} \tag{10}$$

- Even with robust covariance matrix or Monte Carlo simulation, mean-VaR is not reliable for non-normal asset distributions
- For non-normal assets, VaR estimates calculated using GPD (as in VaR.GPD) or Cornish Fisher perform best
- Modified Cornish Fisher VaR takes higher moments of the distribution into account:

$$z_{cf} = z_c + \frac{(z_c^2 - 1)S}{6} + \frac{(z_c^3 - 3z_c)K}{24} + \frac{(2z_c^3 - 5z_c)S^2}{36} \tag{11}$$

$$modVaR = \bar{R} - z_{cf}\sqrt{\sigma} \tag{12}$$

- Modified VaR also meets the definition of a coherent risk measure per Artzner,et.al.(1997)

Risk/Reward Ratios in *PerformanceAnalytics*

- SharpeRatio — return per unit of risk represented by variance, may also be annualized by

$$\frac{\sqrt[n]{prod(1+R_a)^{scale}}-1}{\sqrt{scale}\cdot\sqrt{\sigma}} \tag{13}$$

- Sortino Ratio — improvement on Sharpe Ratio utilizing downside deviation as the measure of risk

$$\frac{(\overline{R_a - MAR})}{\delta_{MAR}} \tag{14}$$

- Calmar and Sterling Ratios — ratio of annualized return (Eq. 1) over the absolute value of the maximum drawdown
- Sortino's Upside Potential Ratio — upside semdiviation from MAR over downside deviation from MAR

$$\frac{\sum_{t=1}^{n}(R_t - MAR)}{\delta_{MAR}} \tag{15}$$

- Favre's modified Sharpe Ratio — ratio of excess return over Cornish-Fisher VaR

$$\frac{(\overline{R_a - R_f})}{modVaR_{R_a,p}} \tag{16}$$

Summary

- Performance and risk analysis are greatly facilitated by the use of charts and tables.
- The display of your infomation is in many cases as important as the analysis.
- *PerformanceAnalytics* contains several tool for measuring and visualizing data that may be used to aid investment decision making.

- Further Work
  - Additional parameterization to make charts and tables more useful.
  - Pertrac or Morningstar-style sample reports.
  - Functions and graphics for more complicated topics such as factor analysis and optimization.

Install PerformanceAnalytics.

- As of version 0.9.4, PerformanceAnalytics is available in CRAN
- Version 0.9.5 was released at the beginning of July
- Install with:
  `> install.packages("PerformanceAnalytics")`
- Required packages include `Hmisc`, `zoo`, and Rmetrics packages such as `fExtremes`.
- Load the library into your active R session using:
  `> library("PerformanceAnalytics").`

Load and Review Data.

```
> data(managers)
> head(managers)

            HAM1 HAM2    HAM3    HAM4 HAM5 HAM6 EDHEC.LS.EQ SP500.TR US.10Y.TR
Jan 1996  0.0100   NA  0.0359  0.0208   NA   NA          NA   0.0340   0.00380
Feb 1996  0.0215   NA  0.0295  0.0231   NA   NA          NA   0.0093  -0.03532
Mar 1996  0.0226   NA  0.0253 -0.0053   NA   NA          NA   0.0096  -0.01057
Apr 1996  0.0008   NA  0.0478  0.0200   NA   NA          NA   0.0147  -0.01739
May 1996  0.0158   NA  0.0337  0.0122   NA   NA          NA   0.0258  -0.00543
Jun 1996 -0.0086   NA -0.0293 -0.0089   NA   NA          NA   0.0038   0.01507
         US.3m.TR
Jan 1996  0.00456
Feb 1996  0.00398
Mar 1996  0.00371
Apr 1996  0.00428
May 1996  0.00443
Jun 1996  0.00412
```

Set Up Data for Analysis.

```
> dim(managers)

[1] 132  10

> managers.length = dim(managers)[1]
> colnames(managers)

 [1] "HAM1"        "HAM2"        "HAM3"        "HAM4"        "HAM5"
 [6] "HAM6"        "EDHEC.LS.EQ" "SP500.TR"    "US.10Y.TR"   "US.3m.TR"

> manager.col = 1
> peers.cols = c(2, 3, 4, 5, 6)
> indexes.cols = c(7, 8)
> Rf.col = 10
> trailing12.rows = ((managers.length - 11):managers.length)
> trailing12.rows

 [1] 121 122 123 124 125 126 127 128 129 130 131 132

> trailing36.rows = ((managers.length - 35):managers.length)
> trailing60.rows = ((managers.length - 59):managers.length)
> frInception.rows = (length(managers[, 1]) - length(managers[,
+     1][!is.na(managers[, 1])]) + 1):length(managers[, 1])
```

Draw a Performance Summary Chart.

```
> charts.PerformanceSummary(managers[, c(manager.col, indexes.cols)],
+     colorset = rich6equal, lwd = 2, ylog = TRUE)
```