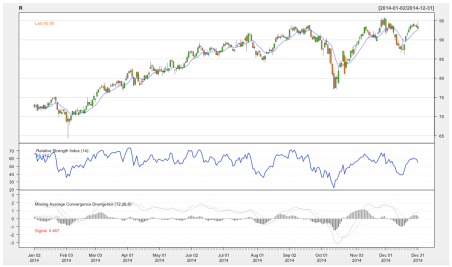
SuRf – February 2015

# Performance Measurement in R

Quantmod & PerformanceAnalytics

Justin Castagna



# About me - My background with “R”

- My name is Justin Castagna
- http://au.linkedin.com/in/justincastagna

- Accountant – Not a programmer or statistician.
- Background with Superannuation, Funds management.

- Largely self taught with ‘R’.
- Began this journey by being involved in quality program, Lean-Six Sigma at AMP. Discovered the value of applied stats with Minitab.
- Changed jobs - no statistics tools.
- Discovered open-source “R” environment in 2006 – right price
- Slow learning curve with command line interface. Often frustrated by my lack of progress it getting ‘R’ to do what I could imagine and what I had been able to do with Minitab. Perseverance and lots of small steps.
- “R” is like a Swiss army knife for your work toolkit

# Agenda

- Introduction
- Process
- Quantmod
  - Commands & Charts
- Returns
- PerformanceAnalytics
  - Commands & Charts
- Attribution
- Questions

# Disclosure

This presentation involves money and shares so…

The information shown is for general information only, it does not constitute any recommendation or advice; it has been prepared without taking into account your personal objectives, financial situation or needs and so you should consider its appropriateness having regard to these factors before acting on it. Any taxation position described is a general statement and should only be used as a guide. It does not constitute tax advice and is based on current tax laws and our interpretation. Your individual situation may differ and you should seek independent professional tax advice. You should also consider obtaining personalised advice from a professional financial adviser before making any financial decisions in relation to the matters discussed hereto.

I may (or may not) hold some of the listed shares directly or indirectly (via fund managers) and have long (or short) economic exposure to these listed shares.

All mistakes are my own!

# Purpose of Talk

This is an introductory talk - No investment background knowledge expected.

1. Will focus on two packages – Give you an overview of
    1. Quantmod
    2. PerformanceAnalytics

2. Learn some ‘R’.

3. Introduce some concepts on Performance Measurement & Attribution.

4. Improve your financial literacy.

Slides available on SURF Meetup site & my LinkedIn profile.

# Financial Literacy

- Last point; Improving your financial Literacy is the most important.

- Presentation a real success if
  - Gives you some new ideas to think about
  - Helps you improve your comprehension of financial related documents such as
    - Newspaper Business pages
    - Investment manager communications
    - Superannuation fund communications
  - Helps you be more skeptical of some advertising of Investment returns
  - Enables you to ask better questions of your financial advisor(s)
  - Helps you understand different types of returns

  - Appreciate the potential complexity of the subject area
  - Sparks an interest in learning more about this sort of stuff

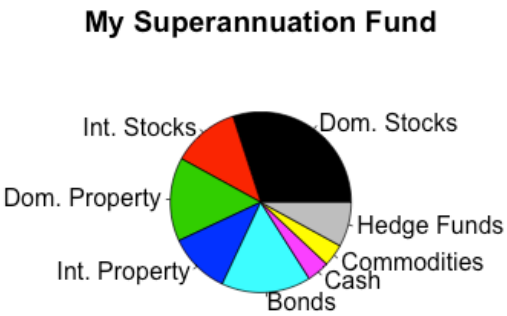
# Definition

- Performance measurement is the quality control of the investment decision process providing necessary information to enable the client (YOU!) and asset managers to assess exactly how the money has been invested and the results of the process.

- Helps answer the questions
  - What is return of the investments?
  - Why has the portfolio performed that way?
  - How can we improve performance?

- What's this to do with me? I'm not a portfolio fund manager.

# Everyone is a Fund Manager!

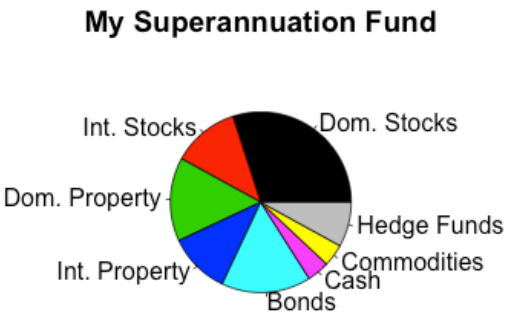




Cash Shares Property Superannuation

- Everyone has investments – you all own a portfolio of assets (& perhaps liabilities to!)
- Directly like a house, shares or managed investments
- Indirectly via their superannuation fund

- Your super fund is likely to have fund managers remunerated based on their performance. Indirectly you are employing them!
- Your Financial advisor may recommend investments based on financial performance

# Portfolio Construction & Valuation

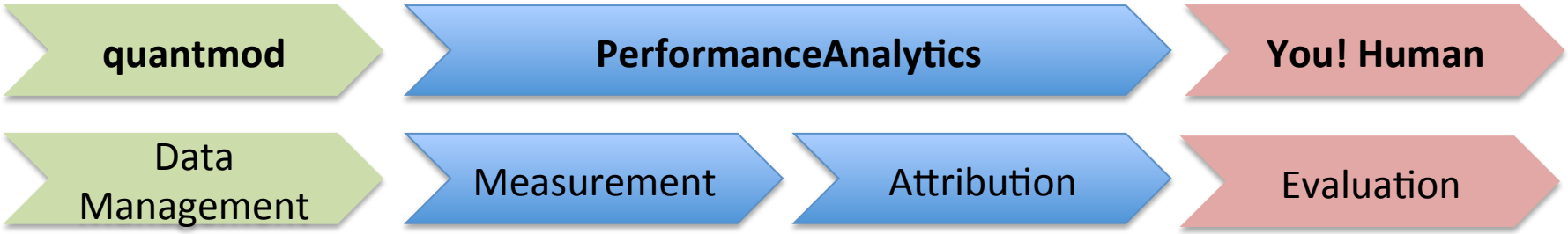




Cash Shares Property Superannuation

- Asset Allocation how investor divides her portfolio into various asset classes.
- Security Selection is the choice of specific securities within an asset class. Based on risk considerations, the investor establishes the asset allocation strategy.

- Valuation methods vary for different asset types.
- AASB ; Type 1 – stock exchange, Type 2 – unlisted investment, Type 3 – opaque model
- Look at your member communications from a super fund – this will be disclosed

# Performance Measurement Process




| | | | |
|---|---|---|---|
| Get Price Data | Calculation of returns | Return Attribution | Feedback |
| Valuation data | Benchmarks | Risk Analysis | Control |
| Financial Info. | Distribution of Info | (ex post / ex ante) | Investment Decision |

# QUANTMOD

*What quantmod IS*

A rapid prototyping environment, with comprehensive tools for data management and visualization.

Can manage and get data from several sources

Yahoo finance; csv file, MySQL, Interactive Brokers, OANDA, Bloomberg…

Able to source several types of data

Share prices, FX, commodities prices, dividends, financial information

Caveat Emptor – You get what you pay for if you use free data ☺

# Lets get started with some R!

```
# My colour conventions; Commentry Text in green; Input in Blue; Output in Black

Most useful command for getting share price data is getSymbols()

> args(getSymbols)

function (Symbols = NULL, env = parent.frame(), reload.Symbols = FALSE,
    verbose = FALSE, warnings = TRUE, src = "yahoo", symbol.lookup = TRUE,
    auto.assign = getOption("getSymbols.auto.assign", TRUE),
    ...)

# Other useful arguments not listed are ‘to=‘ and ‘from=‘

# getSymbols()
# getSymbols(’x', from='2010-01-01', to='2014-12-31', src='yahoo')
```

# Lets get some Data!

```
> library(quantmod)

> getSymbols('R')    # Appropriate stock ticker!  (BTW It's Ryder System, Inc.)
[1] "R"
> head(R,3)
             R.Open  R.High  R.Low  R.Close  R.Volume  R.Adjusted
2007-01-03  51.55    53.42   51.55  52.90    885300    44.41
2007-01-04  52.34    53.70   52.34  53.59    578300    44.99
2007-01-05  53.54    53.58   52.45  52.62    605300    44.18

# Result is an OHLC [Open, High, Low, Close] data set.
# This is an 'Extensible Time Series' ('xts') object

# Note the indexing by time in YYYY-MM-DD format
# R.Adjusted takes into account corporate events such as share splits etc
```

# Get more Financial Data!

```
> R.d <- getDividends('R')
                [,1]
1980-03-03  0.08333
1980-05-23  0.27267
…
2014-08-14 0.37000
2014-11-13 0.37000

> names(R.d)<-"R.Div”
> head(R.d)
                R.Div
1980-03-03   0.08333
1980-05-23   0.27267
```

```
> getFX('AUD/USD')
[1] "AUDUSD”

> tail(AUDUSD)
                AUD.USD
2015-01-10  0.8149
2015-01-11  0.8207
2015-01-12  0.8207
2015-01-13  0.8194

# Try … getMetals()
```

```
> getFinancials("R")                    # Hey - I am an Accountant!
[1] "R.f”
> viewFin(R.f,"BS","A")                 # Can get also get Cashflows, P&L - quarterly
Annual Balance Sheet for R
```

| | 2013-12-31 | 2012-12-31 | 2011-12-31 | 2010-12-31 |
|---|---|---|---|---|
| Cash & Equivalents | 61.56 | 66.39 | 104.57 | 213.05 |
| Cash and Short Term Investments | 61.56 | 66.39 | 104.57 | 213.05 |
| Accounts Receivable - Trade, Net | 777.37 | 655.29 | 647.10 | 520.98 |
| Total Receivables, Net | 777.37 | 775.76 | 754.64 | 615.00 |
| Total Inventory | 64.30 | 64.15 | 65.91 | 58.70 |
| Prepaid Expenses | 159.26 | 91.20 | 99.04 | 84.99 |
| Other Current Assets, Total | NA | 42.73 | 64.00 | 51.55 |
| Total Current Assets | 1062.49 | 1040.24 | 1088.17 | 1023.30 |
| Property/Plant/Equipment, Total - Gross | 11711.88 | 10860.59 | 10047.93 | 8936.22 |
| Accumulated Depreciation, Total | -4587.22 | -4481.13 | -4374.08 | -4128.16 |
| Goodwill, Net | 383.72 | 384.22 | 377.31 | 355.84 |
| Intangibles, Net | 72.41 | 80.47 | 84.82 | 72.27 |
| Other Long Term Assets, Total | 460.50 | 434.59 | 393.69 | 392.90 |
| Total Assets | 9103.78 | 8318.98 | 7617.84 | 6652.37 |
| ….. | | | | |
| Total Liabilities | 7207.07 | 6851.49 | 6299.68 | 5248.06 |
| ….. | | | | |
| Total Equity | 1896.71 | 1467.49 | 1318.15 | 1404.31 |
| Total Liabilities & Shareholders' Equity | 9103.78 | 8318.98 | 7617.84 | 6652.37 |
| Total Common Shares Outstanding | 53.34 | 51.37 | 51.14 | 51.17 |

```
attr(,"col_desc")
[1] "As of 2013-12-31" "As of 2012-12-31" "As of 2011-12-31" "As of 2010-12-31"
```

# Give me some ‘R’ Candy!

```
> chartSeries(R)
```

# Tinkering with the Chart

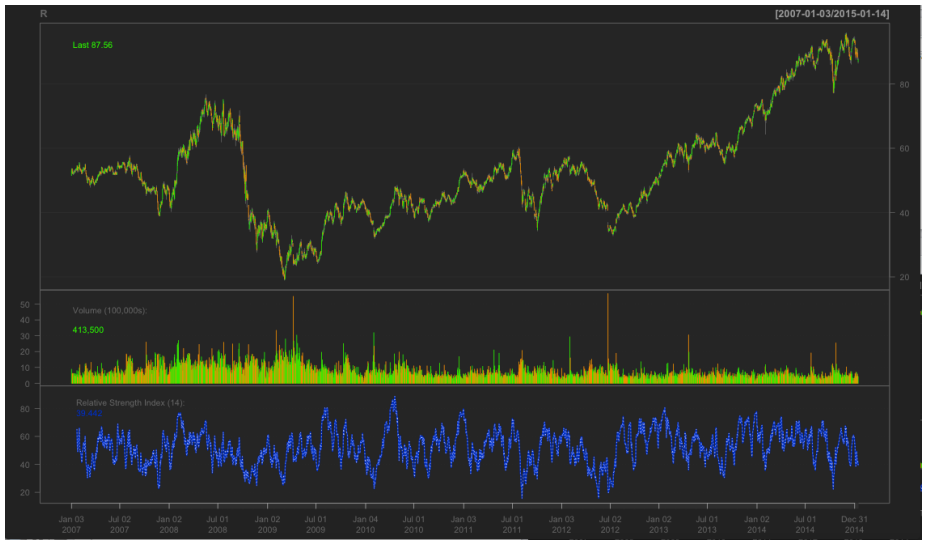
# Add Technical Indicators with TA Argument

> chartSeries(R,TA='addVo();addRSI()')

# Volume and Relative Strength Indicators

# 20+ indicators available

# Refer to TTR package for additional info

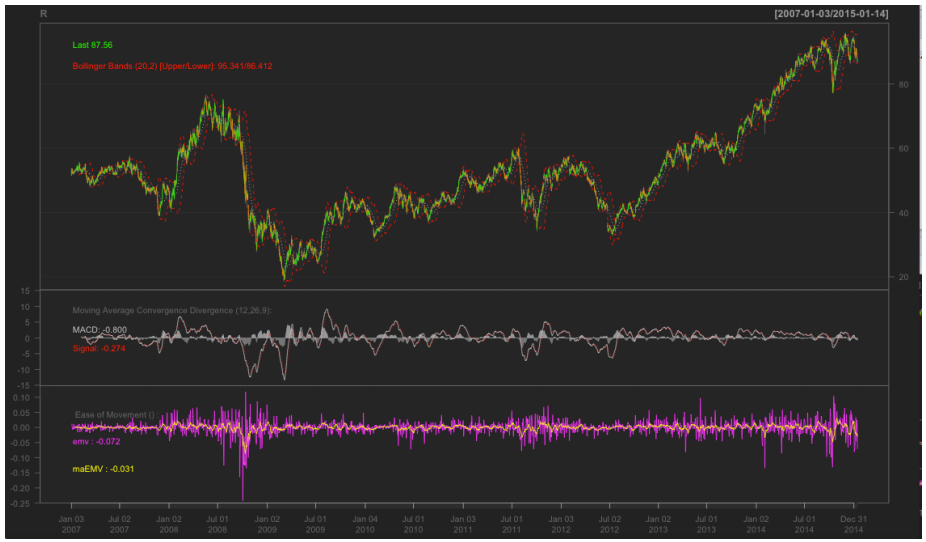



> chartSeries(R,TA='addMACD();addEMV();
addBBands()')

#Moving Average Convergence Divergence

#Bollinger Bands

# Try addZigZag();addTDI();addROC();addCMF()

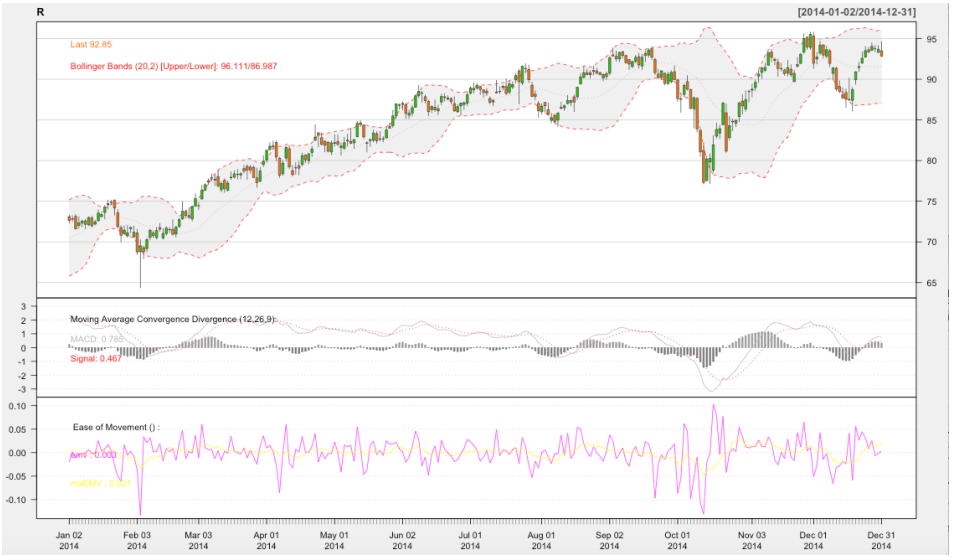


# Use reChart

# No need to reenter all commands

# Use reChart when modifing existing chart

> reChart(subset='2014', theme='white')




# note subsetting data examples.

reChart(subset='last 3 months',
, type='candles')

# Lets make it Local!

# Can't trade Ryder System "R" on the Australian Stock Exchange
# https://au.finance.yahoo.com for lookups

# Note getXxxx() command may not work well on Australia Stocks e.g. Dividends/Financials
# I assume its Yahoo's licensing - You get what you pay for!

# Note the ".AX" suffix
# The stocks **'WBC'** and **'WBC.AX'** are quite different! (Wabco vs Westpac Bank)

```
> getSymbols(c('^AXJO','WBC.AX'),from='2010-01-01')
[1] "AXJO"  "WBC.AX"          # ^AXJOP is index
> head(WBC.AX,3)
```

| | WBC.AX.Open | WBC.AX.High | WBC.AX.Low | WBC.AX.Close | WBC.AX.Volume | WBC.AX.Adjusted |
|---|---|---|---|---|---|---|
| 2010-01-04 | 25.35 | 25.38 | 25.25 | 25.30 | 2746500 | 18.69 |
| 2010-01-05 | 25.60 | 25.60 | 25.39 | 25.51 | 4741300 | 18.84 |
| 2010-01-06 | 25.39 | 25.49 | 25.27 | 25.39 | 3088000 | 18.75 |

# So how did ‘R’ go?

- To understand how ‘R went we need to know its return.
- Return
  - A profit from an investment.
  - change in wealth over time.
- Total Return = Capital return plus Income return
  - (ie share price & dividends)
- Many competing methodologies for calculating returns.
- PerformanceAnalytics package is used for return data

# PerformanceAnalytics

- PerformanceAnalytics provides an **R** package of econometric functions for performance and risk analysis of financial instruments or portfolios.

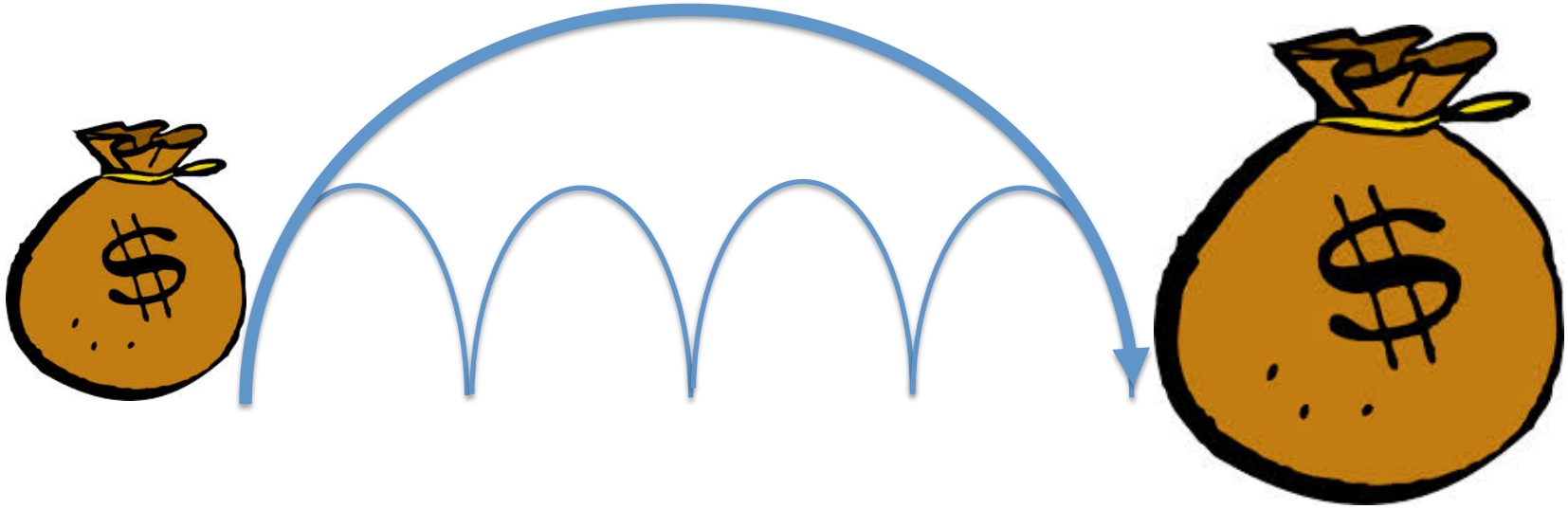
- In general, this package requires return (rather than price) data. Almost all of the functions will work with any periodicity, from annual, monthly, daily, to even minutes and seconds, either regular or irregular.

- Use function Return.calculate for calculating returns from prices.

- Using the 'correct' Return is important.

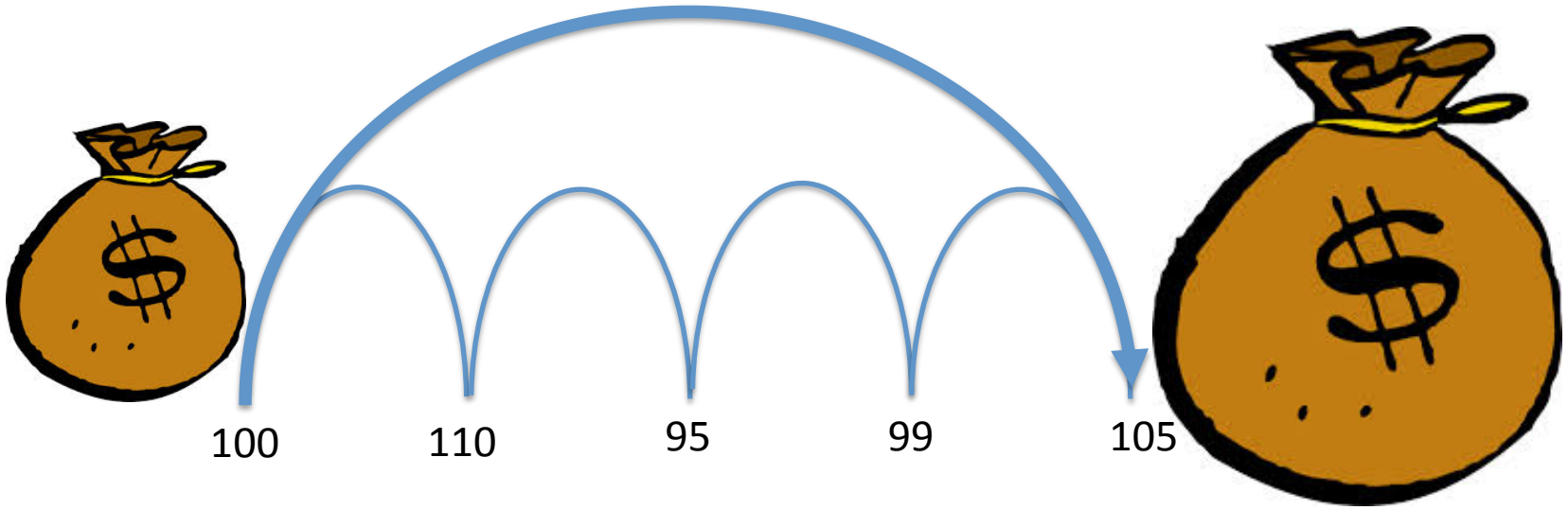
- From a control /evaluation perspective;  some issues to consider are…

# Periods



- Single period vs. multiple periods chained together

# Periods



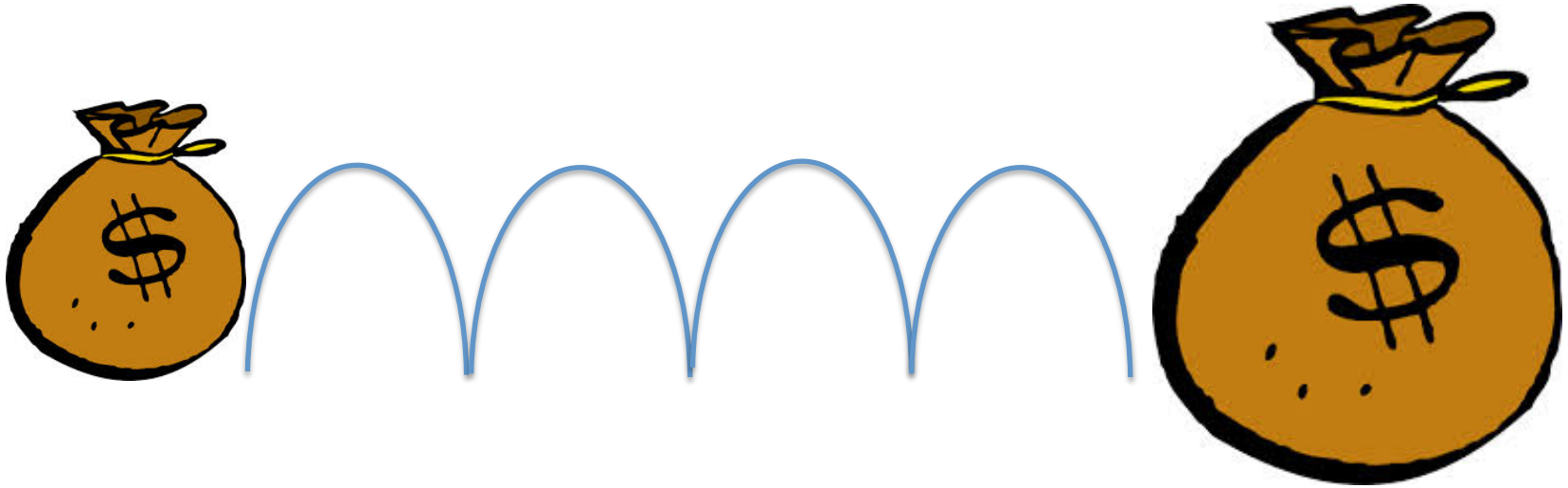
100 110 95 99 105

$$\frac{110}{100} \times \frac{95}{110} \times \frac{99}{95} \times \frac{105}{99} = 105\%$$

$$1.1000 \times 0.8636 \times 1.0421 \times 1.0606 = 1.0499$$

```
> MyReturns=c(1.1, .8636, 1.0421, 1.0606)
> cumprod(MyReturns)
[1] 1.1000000 0.9499600 0.9899533 1.0499445
```

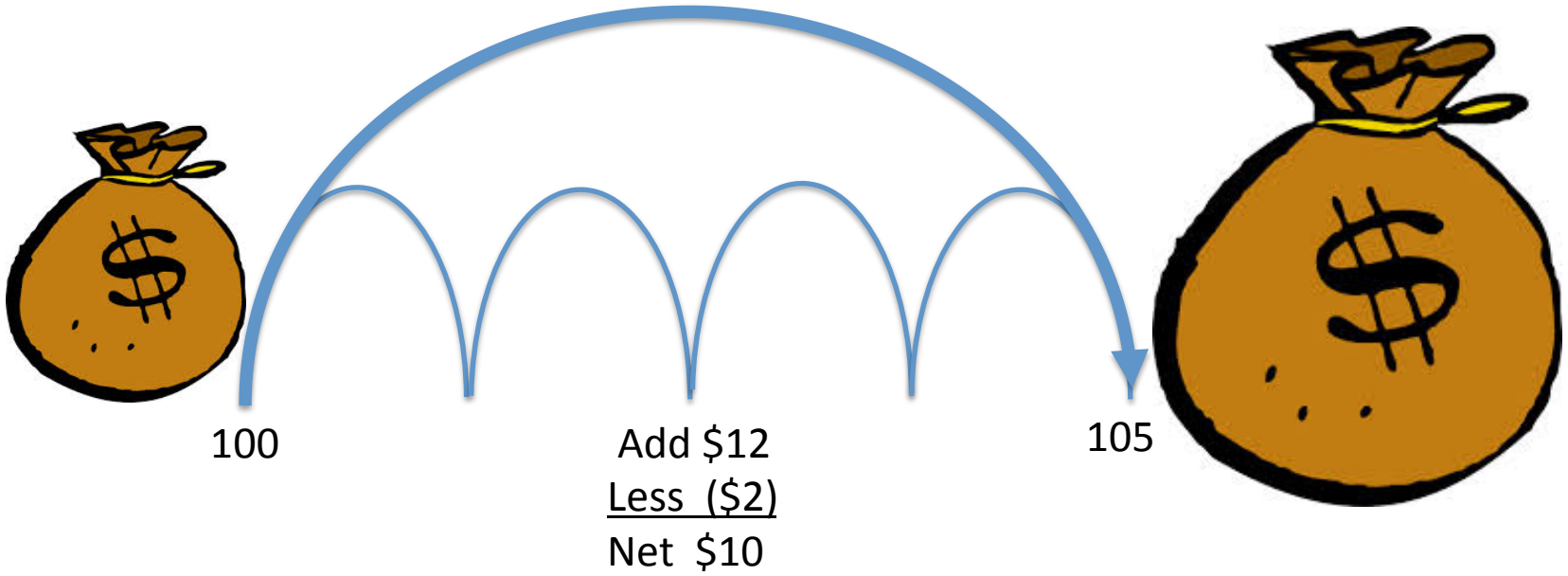
# Arithmetic vs Geometric



| | | |
|---|---|---|
| Cumulative.Rtn= (1.1*.8636*1.0421*1.0606)-1 | [1] 0.04994449 | 5.0% |
| # What's the 'Average' return over each period | | |
| Arithmetic.Ave= (.10-.1362+.0421+.0606)/4 | [1] 0.016625 | |
| Arithmetic.Compounded = (1.016625^4)-1 | [1] 0.0681768 | 6.8% |
| Geometric.Ave=((1.1 * .8636* 1.0421 *1.0606)^.25)-1 | [1] 0.01225885 | |
| Geometric.Compounded=((1.01225)^4)-1 | [1] 0.04990775 | 5.0% |

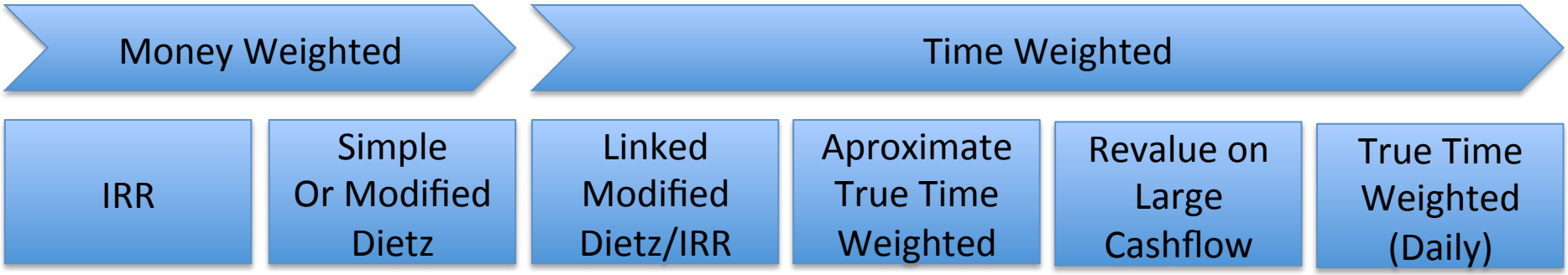
# Arithmetic will always have a higher rate – Which rate is advertised?

# Cashflow Complication




- Investments often have cashflows.
  - Additional contribution of principal on a property
  - A management fee taken out
  - A contribution or withdrawal of cash to/from a managed fund
- Cashflow's complicate the return calculation
- Cashflows impact the valuation of an investment

# Money vs Time weighted




- Many different methodologies to deal with Cashflows
- Different methodology will give you a different return!
- Money Weighted Return is similar to concept of IRR; Investment amounts matter.
- Time Weighted Return measures return of asset irrespective of amounts invested.
- Time of Cashflow event is important
  - Is cash available beginning of day, end of day or do you assume midday?

# Different Return Examples

Valuation.Start = 60.8  Valuation.End = 115.3  External.CF = 45.1

```
# Irr.rate= .1136 manually calculated – Anyone know a good package for IRR?
IRR = (Valuation.Start *(1+Irr.rate)) + (External.CF * (1+Irr.rate)^.5)
[1] 115.2997
```

11.36%

```
Dietz.Simple= (Valuation.End - Valuation.Start - External.CF)/ (Valuation.Start + (External.CF/2))
[1] 0.1127774          # Assumes cashflow middle of day
```

11.27%

---

```
# Dietz Modified
#Cashflow on available on 14th morning   Days.Month =31 Days.Used=13

Dietz.Modified = (Valuation.End - Valuation.Start - External.CF)/
  (Valuation.Start + (((Days.Month - Days.Used)/Days.Month) * External.CF))
[1] 0.108062
```

10.80%

```
#Cashflow on available on 14th evening   Days.Month =31 Days.Used=14
[1] 0.1099001
```

10.99%

# Calculate a Return

# Both Quantmod and PerformanceAnalytics can calculate returns.

# Note these are price returns only; do not include income component from dividends

# Quite flexible to produce various period type or geometric returns

```
library(quantmod)
```

```
> monthlyReturn(R,subset='2014')
            monthly.returns
2014-01-31  -0.035104364
2014-02-28   0.058013766
2014-03-31   0.061072756
2014-04-30   0.028278278
2014-05-30   0.056096374
2014-06-30   0.014978684
2014-07-31  -0.022249972
2014-08-29   0.048879601
2014-09-30  -0.004095639
2014-10-31  -0.016672224
2014-11-28   0.079688030
2014-12-31  -0.027952261
```

```
>library(PerformanceAnalytics)
```

```
>periodReturn(R,subset='2014',period='monthly',type='arithmetic')
            monthly.returns
2014-01-31  -0.035104364
2014-02-28   0.058013766
2014-03-31   0.061072756
2014-04-30   0.028278278
2014-05-30   0.056096374
2014-06-30   0.014978684
2014-07-31  -0.022249972
2014-08-29   0.048879601
2014-09-30  -0.004095639
2014-10-31  -0.016672224
2014-11-28   0.079688030
2014-12-31  -0.027952261
```

# the Return calculate functions assumes regular price data.

# If corporate actions, dividends, or other adjustments such as time- or money-weighting are to be taken into account, those calculations must be made separately. Use adjusted returns, specify quote="AdjClose"

# Benchmark

So you have measured a return of an investment or portfolio.

How do you know if the return is good?

You need to compare against a benchmark – a standard point of reference.

There are different types of benchmarks and these can be constructed quite differently.

Benchmark should be appropriate for investment. Many benchmark indexes available.

```
> getSymbols(c('^AXJO', '^AORD', '^GSPC', '^DJI')
# Indexes S&P/ASX200, All Ordinaries, S&P500, Dow Jones

# What should 'R' – Ryder System be compared against?
```

# ‘R’ Performance Pics

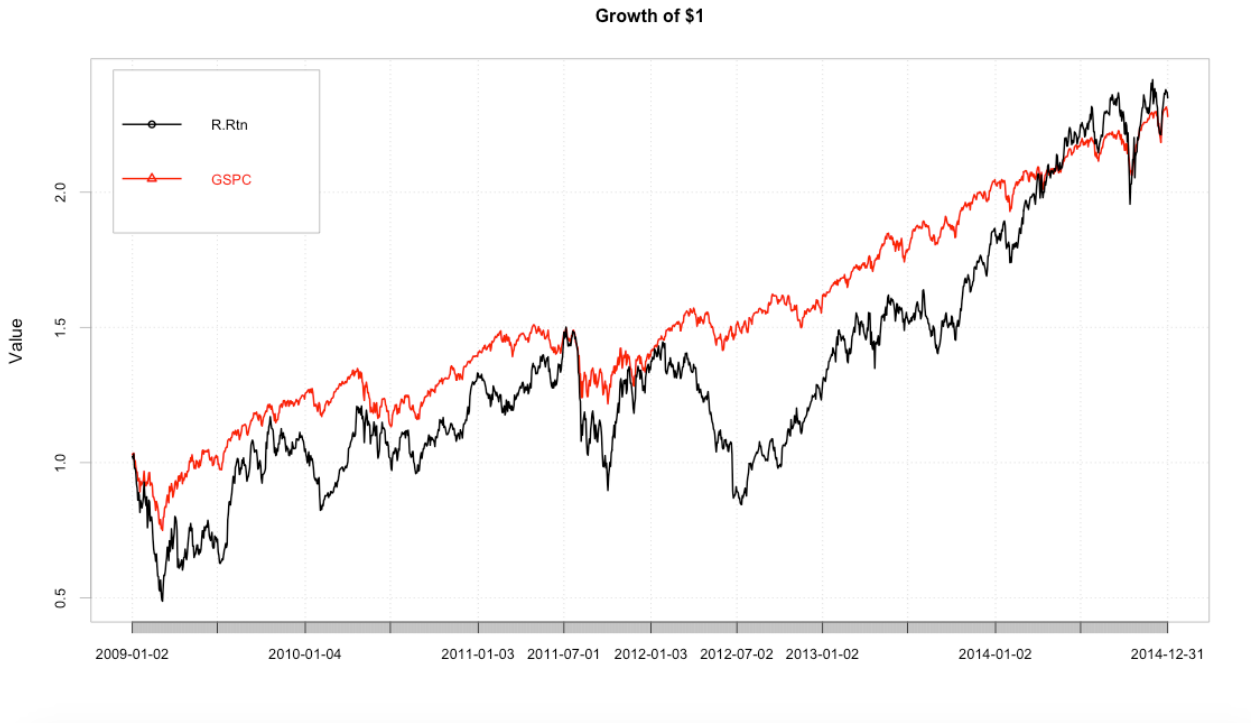



chart.CumReturns(R.Rtn,wealth.index=TRUE,main="Growth of $1", legend.loc="topleft")

# Create a local portfolio

```
# Some LIC's on ASX ;
> getSymbols(c("ARG.AX", "MLT.AX","^AXJO"),
            from="2007-01-01", to= "2014-12-31" ,src='yahoo')

# Create some Monthly Returns & adjust column names
# Note these are price returns only; do not include income component from dividends
> SuRf.Rtn.M=cbind(monthlyReturn(ARG.AX, type='log'),
            monthlyReturn(MLT.AX, type='log'),
            monthlyReturn(AXJO,   type='log'))

> names(SuRf.Rtn.M)<-c("ARG.Rtn.M", "MLT.Rtn.M","AXJO.Rtn.M")

> head(SuRf.Rtn.M)
```

| | ARG.Rtn.M | MLT.Rtn.M | AXJO.Rtn.M |
|---|---|---|---|
| 2007-01-31 | 0.12889101 | -1.510480834 | 0.017595967 |
| 2007-02-28 | -0.11054187 | 0.006403437 | 0.010184563 |

# Tables

# Lots of summary tables available

# Perhaps something odd with MLT.Rtn or original MLT price data

# You get what you pay for with Free data!

> table.Stats(SuRf.Rtn.M)

| | ARG.Rtn.M | MLT.Rtn.M | AXJO.Rtn.M |
|---|---|---|---|
| Observations | 95.0000 | 95.0000 | 96.0000 |
| NAs | 3.0000 | 3.0000 | 2.0000 |
| Minimum | -0.1433 | -1.6317 | -0.1354 |
| Quartile 1 | -0.0256 | -0.0306 | -0.0260 |
| Median | 0.0026 | 0.0000 | 0.0084 |
| Arithmetic Mean | -0.0004 | -0.0163 | -0.0005 |
| Geometric Mean | -0.0017 | NaN | -0.0014 |
| Quartile 3 | 0.0271 | 0.0282 | 0.0321 |
| Maximum | 0.1289 | 1.6378 | 0.0706 |
| SE Mean | 0.0052 | 0.0296 | 0.0044 |
| LCL Mean (0.95) | -0.0108 | -0.0750 | -0.0092 |
| UCL Mean (0.95) | 0.0099 | 0.0424 | 0.0082 |
| Variance | 0.0026 | 0.0830 | 0.0018 |
| Stdev | 0.0509 | 0.2882 | 0.0429 |
| Skewness | -0.1778 | -1.3497 | -0.7513 |
| Kurtosis | 0.6404 | 27.0688 | 0.3457 |

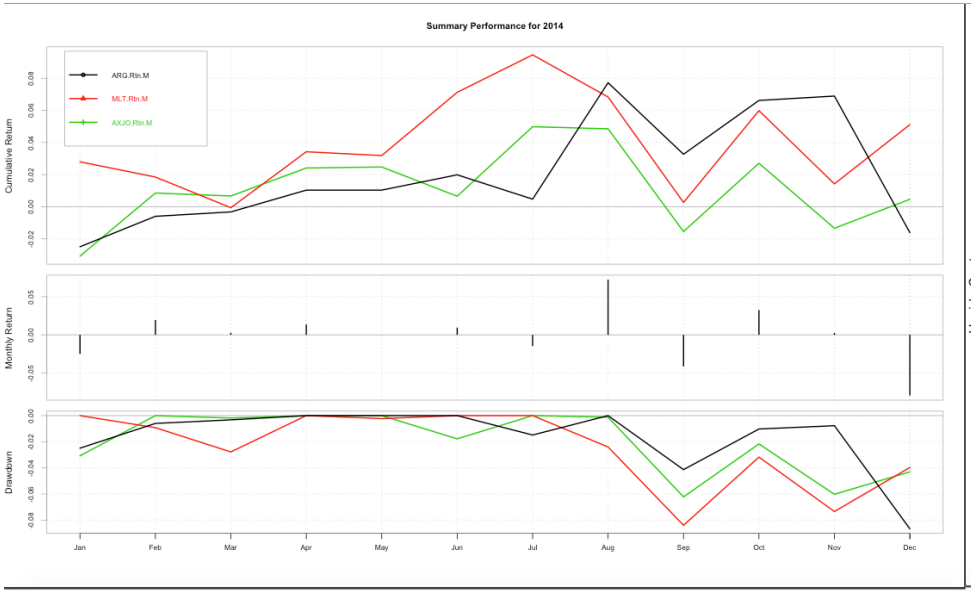
# Some Risk Info

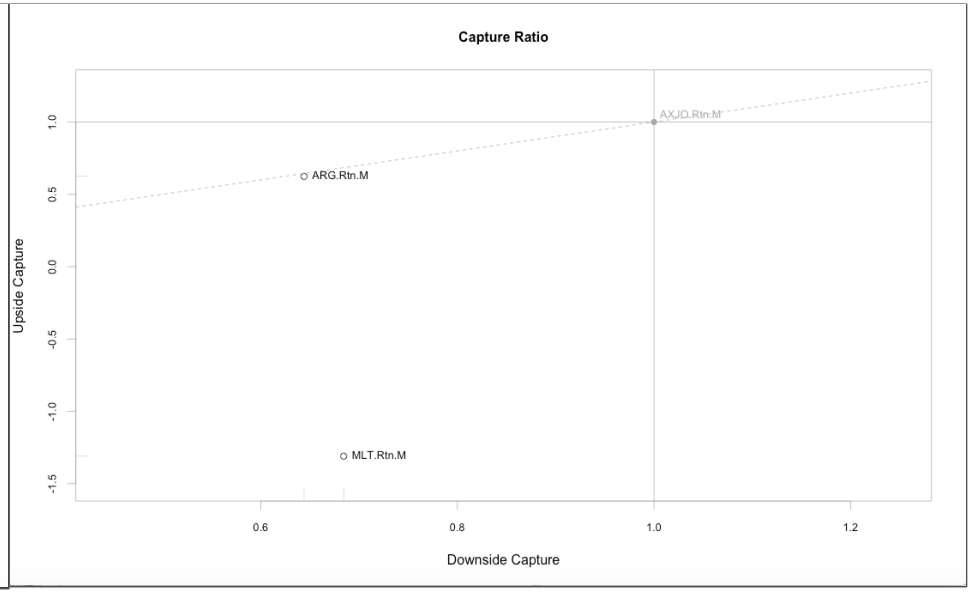
```
> ActiveReturn(SuRf.Rtn.M[,1:2],SuRf.Rtn.M[,3],scale=NA)
                                ARG.Rtn.M          MLT.Rtn.M
Active Premium: AXJO.Rtn.M      -0.006565933       -0.1314682

> table.AnnualizedReturns(SuRf.Rtn.M)
                                ARG.Rtn.M     MLT.Rtn.M     AXJO.Rtn.M
Annualized Return               -0.0206       -0.0388        -0.0170
Annualized Std Dev              0.1762         0.9982         0.1486
Annualized Sharpe (Rf=0%)      -0.1170        -0.0389        -0.1142
# Sharpe; return  per unit of risk; higher ratio better (absolute returns/risks)

> InformationRatio(SuRf.Rtn.M[,1:2],SuRf.Rtn.M[,3],scale=NA) # Ex Post Relative Risk
                                ARG.Rtn.M          MLT.Rtn.M
Information Ratio: AXJO.Rtn.M   -0.04375041        -0.1325988
# Excess Rtn/tracking error ~ Measure of fund managers skill
```

# Summaries

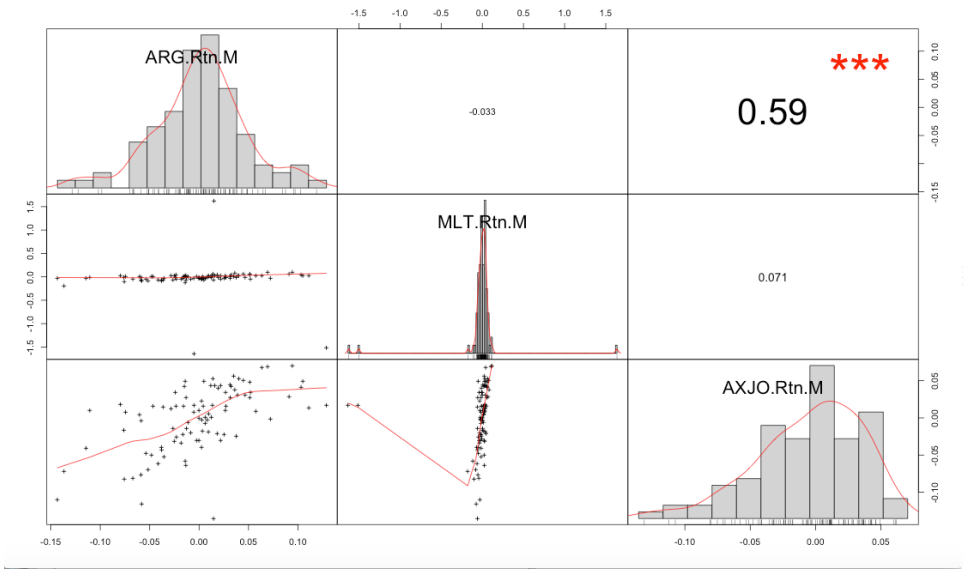


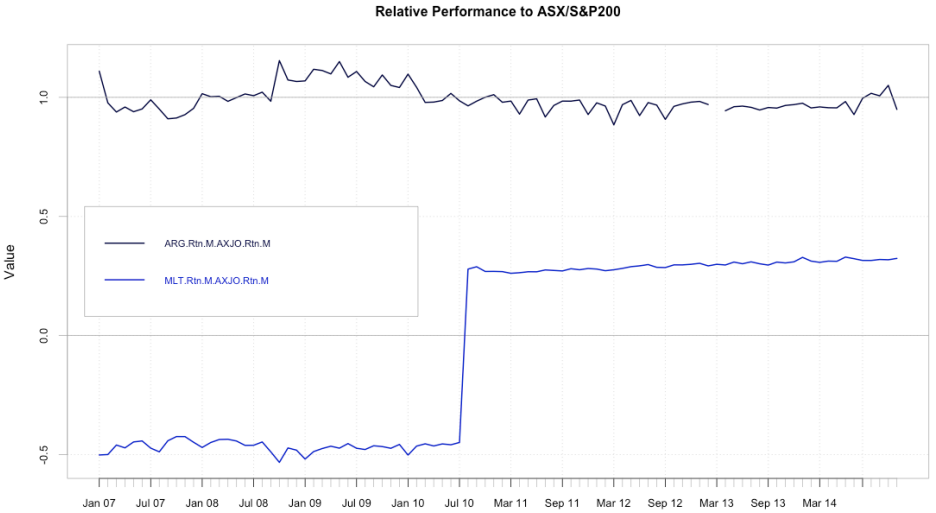




```
> charts.PerformanceSummary(SuRf.Rtn.M['2014',c(1,2,3)])
```

```
> chart.CaptureRatios(SuRf.Rtn.M[,1:2], SuRf.Rtn.M[,3,drop=FALSE])
```

# Charts

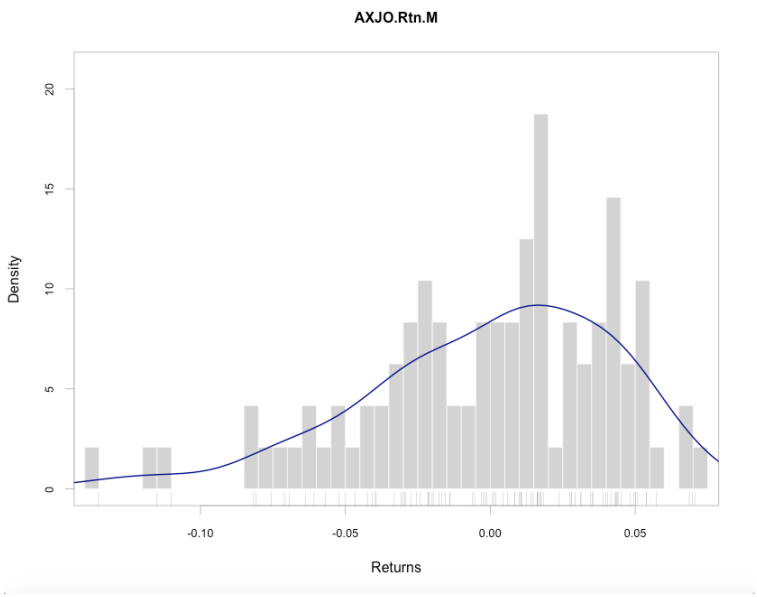


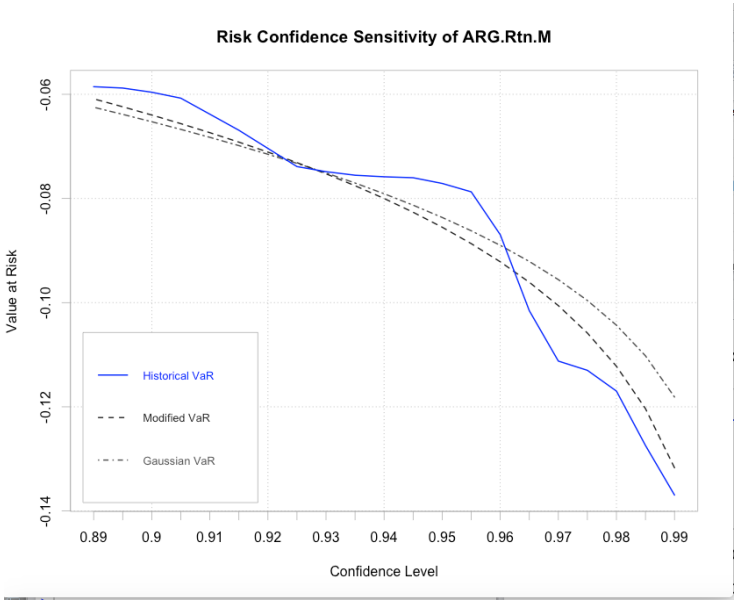




```
chart.Correlation(SuRf.Rtn.M[,1:3], histogram=TRUE, pch="+")
# significance stars

chart.RelativePerformance(SuRf.Rtn.M[, 1:2, drop=FALSE],
      SuRf.Rtn.M[, 3, drop=FALSE], colorset=rich8equal, legend.loc="left”)
```

# Charts




```
chart.Histogram(SuRf.Rtn.M[,3,drop=FALSE], breaks=50, methods = c("add.density", "add.rug") )

chart.VaRSensitivity(SuRf.Rtn.M[,1,drop=FALSE],
                     methods=c("HistoricalVaR", "ModifiedVaR", "GaussianVaR"), colorset=bluefocus, lwd=2)
# different types of VaR
```

# Return.portfolio

```
# Use to calculate returns of multiple multiple securites
x1=Return.portfolio(SuRf.Rtn.M[94:97,1:2])  # Default Return.portfolio; equal weighting, no rebalancing;
x2=Return.portfolio(SuRf.Rtn.M[94:97,1:2],  rebalance_on='months')
# With Monthly rebalanceing  - Note return differences
x3=Return.portfolio(SuRf.Rtn.M[94:97,1:2], weights=c(.5,.5), rebalance_on='months')
#  With monthly balancing; weights explicit, Same as above
x4=Return.portfolio(SuRf.Rtn.M[94:97,1:2], weights=c(.6,.4), rebalance_on=NA)  # Changing weights mix
x5=Return.portfolio(SuRf.Rtn.M[94:97,1:2], weights=c(.6,.4), rebalance_on='months')
# monthly balancing; changing weights mix
x6=Return.portfolio(SuRf.Rtn.M[94:97,1:2], weights=c(.6,.4), rebalance_on='quarters')
# Quarterly balancing; changed weight mix
```

x1 portfolio.returns

| | |
|---|---|
| 2014-08-29 | 0.02408501 |
| 2014-09-30 | -0.05092928 |
| 2014-10-31 | 0.04403184 |
| 2014-11-28 | -0.01920169 |

x2 portfolio.returns

| | |
|---|---|
| 2014-08-29 | 0.02408501 |
| 2014-09-30 | -0.05140117 |
| 2014-10-31 | 0.04473865 |
| 2014-11-28 | -0.02024536 |

x3 portfolio.returns

| | |
|---|---|
| 2014-08-29 | 0.02408501 |
| 2014-09-30 | -0.05140117 |
| 2014-10-31 | 0.04473865 |
| 2014-11-28 | -0.02024536 |

x4 portfolio.returns

| | |
|---|---|
| 2014-08-29 | 0.03370050 |
| 2014-09-30 | -0.04894205 |
| 2014-10-31 | 0.04160996 |
| 2014-11-28 | -0.01469325 |

x5 portfolio.returns

| | |
|---|---|
| 2014-08-29 | 0.03370050 |
| 2014-09-30 | -0.04939085 |
| 2014-10-31 | 0.04228078 |
| 2014-11-28 | -0.01568609 |

x6 portfolio.returns

| | |
|---|---|
| 2014-08-29 | 0.03370050 |
| 2014-09-30 | -0.04894205 |
| 2014-10-31 | 0.04228078 |
| 2014-11-28 | -0.01594412 |

# Portfolio Return.Verbose

```
x6=Return.portfolio(SuRf.Rtn.M[94:97,1:2], verbose=TRUE, weights=c(.6,.4), rebalance_on='quarters')
# Verbose = TRUE; Shows more than just returns
```

$returns

| | portfolio.returns |
|---|---|
| 2014-08-29 | 0.03370050 |
| 2014-09-30 | -0.04894205 |
| 2014-10-31 | 0.04228078 |
| 2014-11-28 | -0.01594412 |

$contribution

| | ARG.Rtn.M | MLT.Rtn.M |
|---|---|---|
| 2014-08-29 | 0.043297468 | -0.009596971 |
| 2014-09-30 | -0.025732857 | -0.023209190 |
| 2014-10-31 | 0.019469586 | 0.022811197 |
| 2014-11-28 | 0.001516175 | -0.017460299 |

$BOP.Weight

| | ARG.Rtn.M | MLT.Rtn.M |
|---|---|---|
| 2014-08-29 | 0.6000000 | 0.4000000 |
| 2014-09-30 | 0.6223248 | 0.3776752 |
| 2014-10-31 | 0.6000000 | 0.4000000 |
| 2014-11-28 | 0.5943404 | 0.4056596 |

$EOP.Value

| | ARG.Rtn.M | MLT.Rtn.M |
|---|---|---|
| 2014-08-29 | 0.6432975 | 0.3904030 |
| 2014-09-30 | 0.6166974 | 0.3664117 |
| 2014-10-31 | 0.6090062 | 0.4156695 |
| 2014-11-28 | 0.6105598 | 0.3977784 |

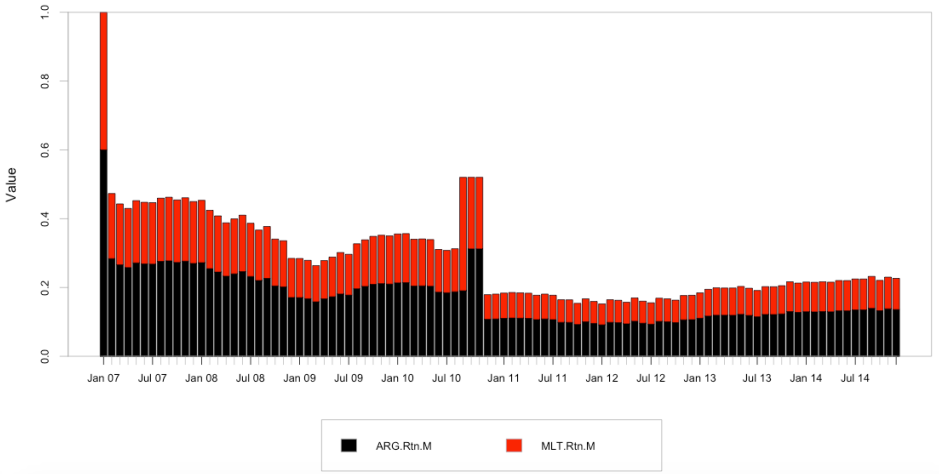
$BOP.Value

| | ARG.Rtn.M | MLT.Rtn.M |
|---|---|---|
| 2014-08-29 | 0.6000000 | 0.4000000 |
| 2014-09-30 | 0.6432975 | 0.3904030 |
| 2014-10-31 | 0.5898654 | 0.3932436 |
| 2014-11-28 | 0.6090062 | 0.4156695 |

$EOP.Weight

| | ARG.Rtn.M | MLT.Rtn.M |
|---|---|---|
| 2014-08-29 | 0.6223248 | 0.3776752 |
| 2014-09-30 | 0.6272930 | 0.3727070 |
| 2014-10-31 | 0.5943404 | 0.4056596 |
| 2014-11-28 | 0.6055109 | 0.3944891 |

# Portfolio Pictures


```
x7=Return.portfolio(SuRf.Rtn.M2[,1:2], verbose=TRUE, weights=c(.6,.4), rebalance_on='months')
# Entire data  set; spike is where problem source data is like likely to be

chart.CumReturns(x7$returns)  #recall issue with historical MLT data

chart.StackedBar(x7$BOP.Value)
```

# Attribution

# Attribution > Act of determining the contributors of a result

# Model discussed below is based on Domestic Equities only

# More complicated approaches for

International Equities (i.e. Currency effects)

Other asset class types (i.e. Fixed Interest)

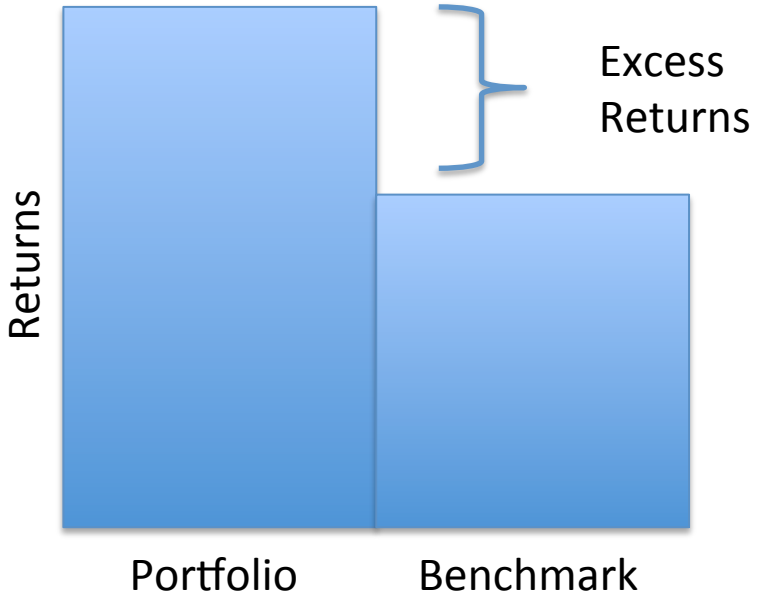
Multi Level Attribution (Fund of Funds)

Derivatives (i.e. Futures, Forwards, Swaps, overlays)

```
mp= c(.5,.3,.2,.5,.4,.1,.2,-.08,.6,.1,-.04,.08)
dim(mp) <- c(3,4)   mp=data.frame(mp)
colnames(mp)= c( 'Portfolio.W', 'Benchmark.W', 'Portfolio.Rtn', 'Benchmark.Rtn')
```

| | Portfolio.W | Benchmark.W | Portfolio.Rtn | Benchmark.Rtn |
|---|---|---|---|---|
| 1 | 0.5 | 0.5 | 0.20 | 0.10 |
| 2 | 0.3 | 0.4 | -0.08 | -0.04 |
| 3 | 0.2 | 0.1 | 0.60 | 0.08 |

# Brinson Model(s)




| Timing | Selection: Portfolio | Selection: Benchmark |
|---|---|---|
| Portfolio | (IV) Actual Portfolio Return | (II) Asset Allocation Return |
| Benchmark | (III) Security Selection Rtn. | (I) Benchmark Rtn. |

- Brinson model(s) were the among the first to decompose returns.
- Focus of attribution is to determine the source of the excess return or alternatively the active decsions of the investment manager.
- Terminology - Beta is benchmark or market return, Alpha is excess return of portfolio.

```
# Brinson Model
> Port.Rtn.Arith = Portfolio.W * Portfolio.Rtn
> Port.Rtn.Arith [1]  0.100 -0.024  0.120   #Component IV (Portfolio Return) contribution at security level
> sum(Port.Rtn.Arith) [1] 0.196              # Component IV (Portfolio Return) for entire portfolio.

> Benchmk.Rtn.Arith = Benchmark.W * Benchmark.Rtn
> Benchmk.Rtn.Arith [1]  0.050 -0.016  0.008  # Component I
> sum(Benchmk.Rtn.Arith) [1] 0.042

> Excess.Rtn= sum(Port.Rtn.Arith) - sum(Benchmk.Rtn.Arith)  # Components IV - I
> Excess.Rtn [1] 0.154 #Arithmetic
```

```
> Asset.Allocation = (Portfolio.W - Benchmark.W) * Benchmark.Rtn
> sum(Asset.Allocation) [1] 0.012

> Security.Selection = Benchmark.W * (Portfolio.Rtn - Benchmark.Rtn)
> sum(Security.Selection) [1] 0.086

> Interaction = (Portfolio.W - Benchmark.W) * (Portfolio.Rtn - Benchmark.Rtn)  # No one likes “Other”
> sum(Interaction) [1] 0.056

> Excess.Rtn.2= Asset.Allocation + Security.Selection + Interaction
> sum(Excess.Rtn.2) [1] 0.154  #Ties to above
```

# Brinson Falcher Model

```
> Port.Rtn.Arith=Portfolio.W * Portfolio.Rtn
> sum(Port.Rtn.Arith) [1] 0.196

> Benchmk.Rtn.Arith = Benchmark.W * Benchmark.Rtn
> sum(Benchmk.Rtn.Arith) [1] 0.042
```

---

```
> Asset.Alloc.2 = (Portfolio.W - Benchmark.W) * (Benchmark.Rtn - sum(Benchmk.Rtn.Arith))
> Asset.Alloc.2 [1] 0.0000 0.0082 0.0038
> sum(Asset.Alloc.2) [1] 0.012

> Security.Selection = Portfolio.W * (Portfolio.Rtn - Benchmark.Rtn)
> Security.Selection [1]  0.050 -0.012  0.104
> sum(Security.Selection) [1] 0.142

> sum(Port.Rtn.Arith)-sum(Benchmk.Rtn.Arith) [1] 0.154
> sum(Asset.Alloc.2)+sum(Security.Selection)     [1] 0.154
```

```
#geometric asset alloc
> Asset.Alloc.Geo= (Portfolio.W - Benchmark.W) * (((1+Benchmark.Rtn)/
(1+sum(Benchmk.Rtn.Arith)) -1))
> Asset.Alloc.Geo
[1] 0.000000000 0.007869482 0.003646833
> sum(Asset.Alloc.Geo)
[1] 0.01151631

> Security.Selection.Geo = Portfolio.W * (((1+Portfolio.Rtn )/(1+Benchmark.Rtn))-1) *
                                    (1+Benchmark.Rtn)/ (((sum(Benchmk.Rtn.Arith) + sum(Asset.Alloc.2)))+1)
> Security.Selection.Geo
[1]  0.04743833 -0.01138520  0.09867173
> sum(Security.Selection.Geo)
[1] 0.1347249

> sum(Asset.Alloc.Geo + Security.Selection.Geo)
[1] 0.1462412
# Note Geometric results less than Arithmetic of 15.4%
```

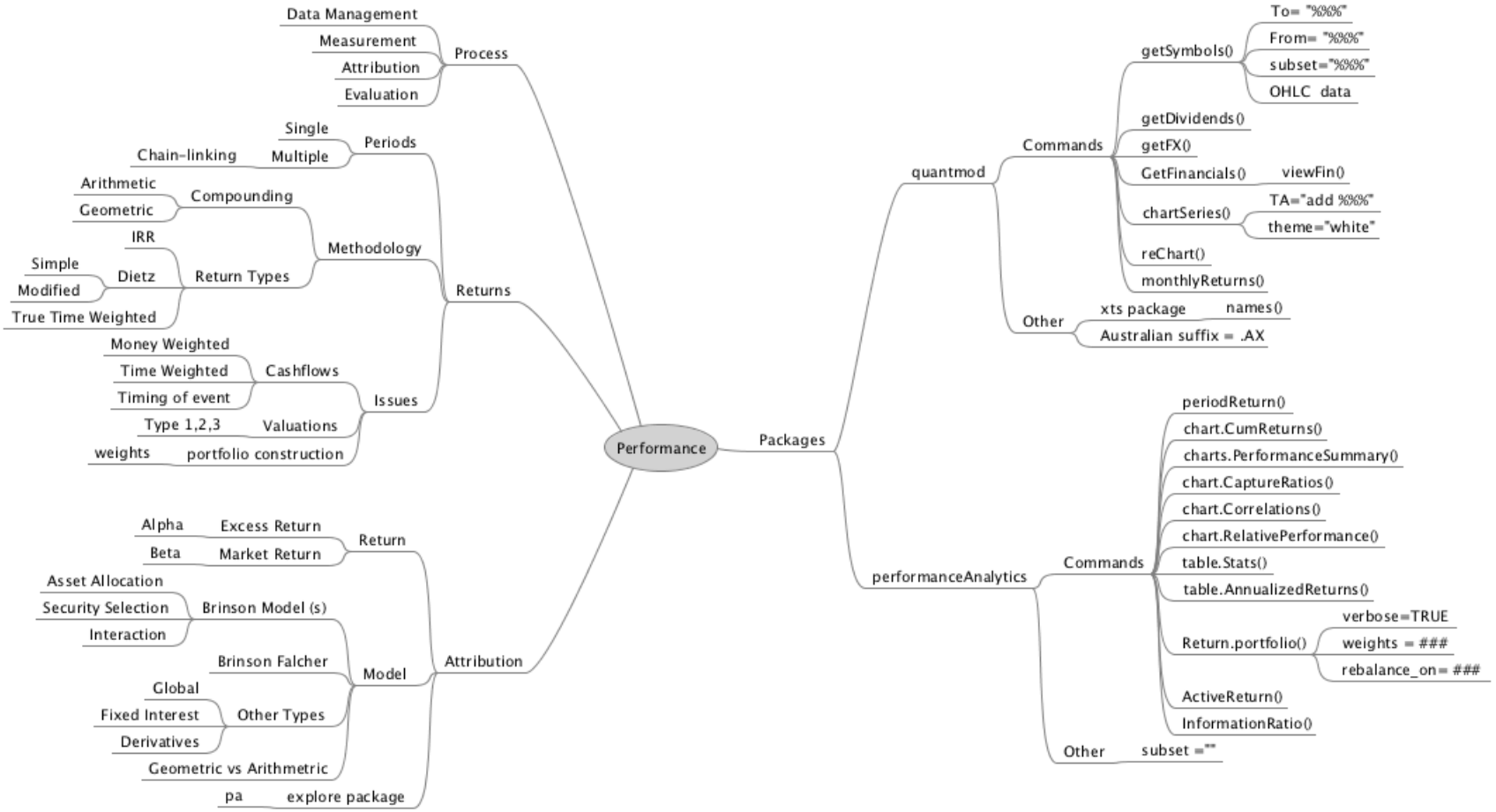
# Wrap up

- Hopefully, presentation has helped you understand different types of returns and issues involved in calculating
  - Helps you be more skeptical of some advertising of Investment returns.
  - Enables you to ask better questions of your financial advisor(s).
  - Appreciate concepts related to performance measurement & attribution
  - You would feel comfortable using the quantmod and perfromanceAnalytics pkgs.

- Broadly speaking best practice is
  - Valuations are done at market value.
  - Total returns (price + income) used.
  - Time weighted.
  - Geometric.
  - Risk is measured as well as return.

- Explore further the breadth or quantmod, perfromanceAnalytics packages
  - Related packages like pa, xts and TTR

# Summary


Performance
Process
Data Management
Measurement
Attribution
Evaluation
Returns
Periods
Single
Multiple
Chain-linking
Methodology
Compounding
Arithmetic
Geometric
Return Types
IRR
Dietz
Simple
Modified
True Time Weighted
Issues
Cashflows
Money Weighted
Time Weighted
Timing of event
Valuations
Type 1,2,3
portfolio construction
weights
Attribution
Return
Excess Return
Alpha
Market Return
Beta
Model
Brinson Model (s)
Asset Allocation
Security Selection
Interaction
Brinson Falcher
Other Types
Global
Fixed Interest
Derivatives
Geometric vs Arithmetric
explore package
pa
Packages
quantmod
Commands
getSymbols()
To= "%%%"
From= "%%%"
subset="%%%"
OHLC data
getDividends()
getFX()
GetFinancials()
viewFin()
chartSeries()
TA="add %%%"
theme="white"
reChart()
monthlyReturns()
Other
xts package
names()
Australian suffix = .AX
performanceAnalytics
Commands
periodReturn()
chart.CumReturns()
charts.PerformanceSummary()
chart.CaptureRatios()
chart.Correlations()
chart.RelativePerformance()
table.Stats()
table.AnnualizedReturns()
Return.portfolio()
verbose=TRUE
weights = ###
rebalance_on= ###
ActiveReturn()
InformationRatio()
Other
subset =""

# Further Reading

- http://www.financialliteracy.gov.au
- https://www.moneysmart.gov.au
- http://en.wikipedia.org/wiki/Financial_literacy

- http://www.asx.com.au/products/etf/managed-funds-etp-product-list.htm

- Practical Portfolio Performance Measurement and Attribution: Carl Bacon
- Investment Performance Attribution: David Spaulding

- R Vignets for Quantmod, PerformanceAnalytics, TTR, xts, zoo, pa

R?’s