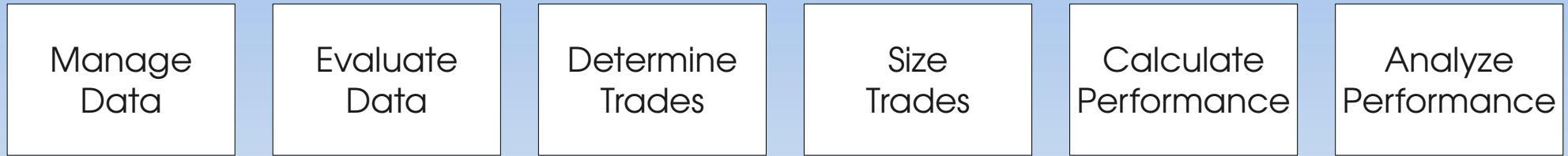


Quantitative Strategy Development in R

R/Finance Chicago 2011
Brian G. Peterson
brian@braverock.com

Trade Simulation Tool Chain



Types of Activities

Connect to database

Download historical data

Clean and align data

Graph prices and indicators

Calculate indicators

Transform prices

Estimate volatility

Calculate trailing volume

Estimate pre-trade pricing

Forecast return

Forecast risk

Evaluate rules

Generate signals

Optimize portfolio

Budget risk

Calculate target position

Calculate trade size

Evaluate trading costs

Calculate Performance

Specify contract specs

Capture trades

Calculate positions

Calculate P&L

Aggregate portfolio

Analyze Performance

Calculate returns and risk

Compare to benchmarks

Provide attribution

Analyze risk

Example R Packages

quantmod
indexes
RTAQ
xts
...

TTR
signal-extraction
...

quantstrat
quantmod

lspm
Portfolio-Analytics

blotter
Financial-Instrument

Performance-Analytics

About the Faber Example

- A very simple trend following strategy:
 - Faber, Mebane T., "*A Quantitative Approach to Tactical Asset Allocation.*" *Journal of Risk Management* (Spring 2007).
- Buy when monthly price $>$ 10-month SMA.
- Sell and move to cash when monthly price $<$ 10-month SMA.
- 10 years of monthly data, S&P Sector ETFs.
- No shorting, 'sell' goes to cash.
- Positions are fixed.

Faber in R Code

```
currency('USD')
symbols = c("XLF", "XLP", "XLE", "XLY", "XLV", "XLI", "XLB", "XLK", "XLU")
for(symbol in symbols){ stock(symbol, currency="USD",multiplier=1) }
getSymbols(symbols, src='yahoo', index.class=c("POSIXt","POSIXct"),
from='1998-01-01')
for(symbol in symbols) {
  x<-get(symbol)
  x<-to.monthly(x,indexAt='lastof',drop.time=TRUE)
  colnames(x)<-gsub("x",symbol,colnames(x))
  assign(symbol,x)
}
initPortf('faber', symbols=symbols, initDate='1997-12-31')
initAcct('faber', portfolios='faber', initDate='1997-12-31')
initOrders(portfolio='faber', initDate='1997-12-31')
s <- strategy("faber")
s <- add.indicator(strategy = s, name = "SMA", arguments = list(x =
quote(Cl(mktdata)), n=10), label="SMA10")
s <- add.signal(s, name="sigCrossover", arguments = list(data=quote(mktdata),
columns=c("Close","SMA"), relationship="gt"), label="Cl.gt.SMA")
s <- add.signal(s,name="sigCrossover", arguments = list(data=quote(mktdata),
columns=c("Close","SMA"), relationship="lt"),label="Cl.lt.SMA")
s <- add.rule(s, name='ruleSignal', arguments = list(data=quote(mktdata),
sigcol="Cl.gt.SMA", sigval=TRUE, orderqty=100, ordertype='market',
orderside=NULL, threshold=NULL), type='enter')
s <- add.rule(s, name='ruleSignal', arguments = list(data=quote(mktdata),
sigcol="Cl.lt.SMA", sigval=TRUE, orderqty='all', ordertype='market',
orderside=NULL, threshold=NULL), type='exit')
out <- try(applyStrategy(strategy='s' , portfolios='faber'))
updatePortf(Portfolio='faber')
```

Code Color Key:

Financial-
Instrument
quantmod
blotter
quantstrat
TTR
xts

Hidden:

xts
TTR
blotter

No custom code

Faber Results

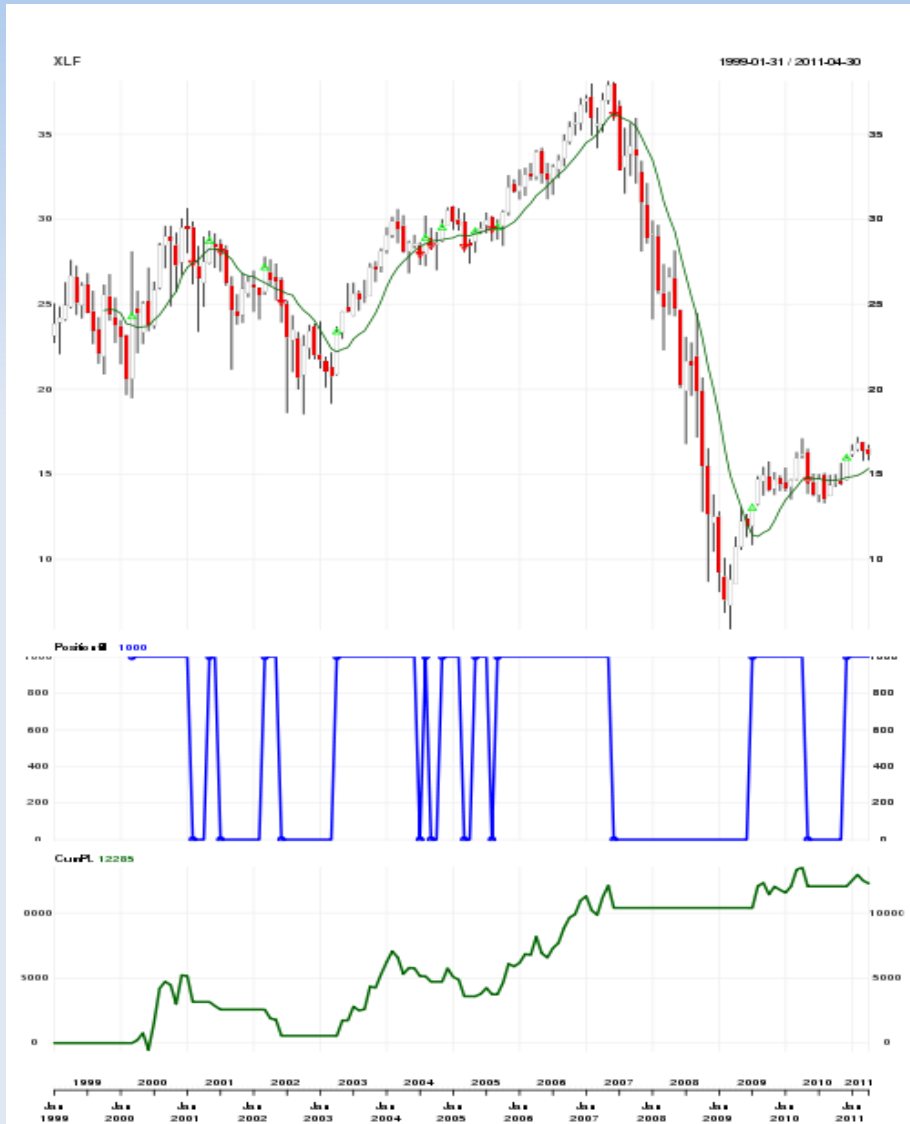
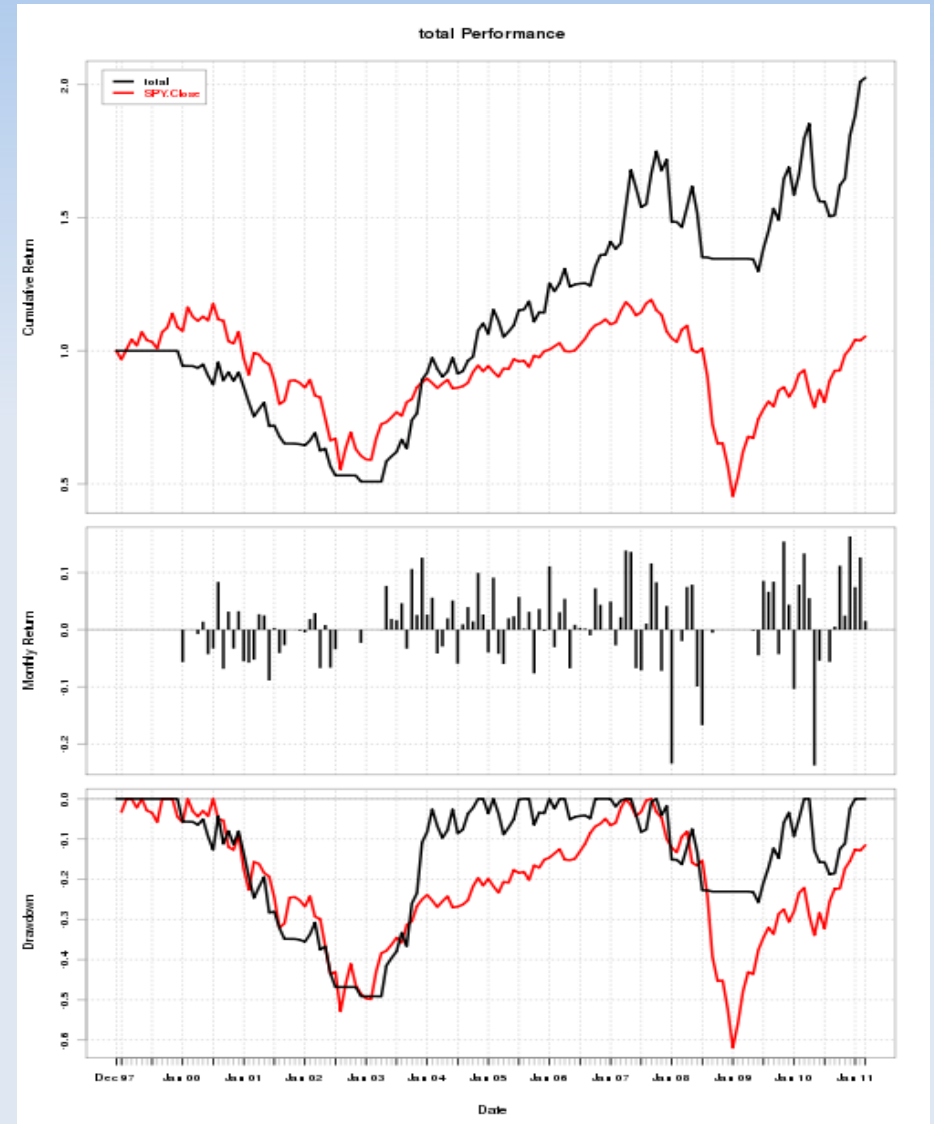


chart.Posn()



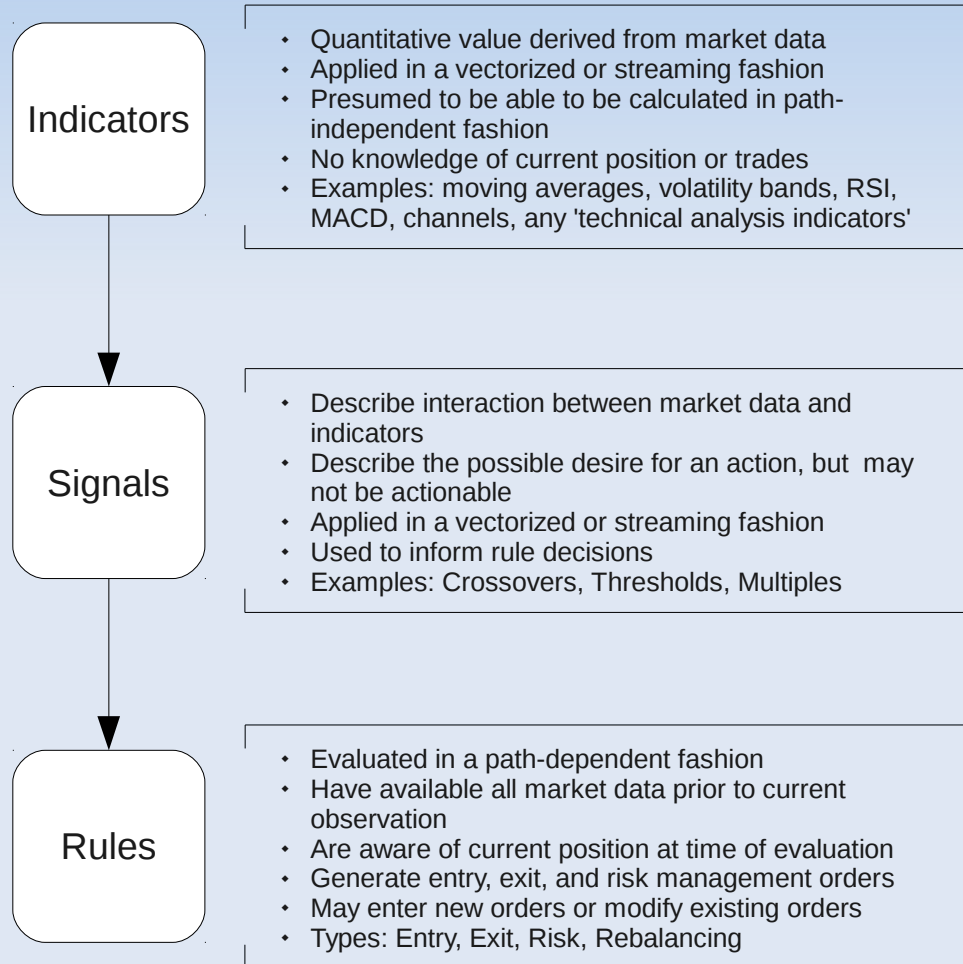
charts.PerformanceSummary()

Faber Results, cont.

- Strategy is profitable over a decade on all but one sector
- Performance could be improved by capital aware position sizing, and more sophisticated limit-based exits
- Selection of output from tradeStats() function:

	Net Trading PL	Max Drawdown	# Trades	Profit Factor	Std Dev Trade PL	Largest Winner	Largest Loser	Max Equity	Min Equity
XLF	12285	-4670	20	3.9	2061	6655	-2015	13545	-525
XLP	7185	-4730	22	2.0	1035	3195	-1745	7245	-115
XLE	47255	-17730	22	2.4	6927	28965	-3635	47655	-9135
XLY	8645	-8030	26	1.2	1852	5835	-2205	9075	-4655
XLV	1015	-5805	26	0.6	958	2555	-1655	1015	-5805
XLI	5685	-8930	30	0.9	2201	7915	-4835	6225	-8235
XLB	15885	-9305	24	1.7	2533	7965	-3255	15885	-9305
XLK	-7355	-15365	26	0.5	2540	3985	-10155	0	-15365
XLU	17255	-6800	16	4.9	3226	10835	-1805	21525	-1245

Using quantstrat



- Designed and used for 'real' quantitative strategies at all frequencies
- Many strategies may be constructed from all open source components
- Proprietary strategies add custom
 - Indicators
 - Signal Functions
 - Order Sizing Logic
- Framework may be extended for live execution
- GSoC 2011 project