Basic stock modeling `stockPortfolio` in R

David Diez
UCLA Statistics

Overview

There are three simple steps in `stockPortfolio` to identifying an optimal portfolio:

1. Download data.

```
> stockData <- getReturns(vectorOfTickers,
+   start='2004-09-01', end='2009-09-30')
```

2. Model the stock behavior using the data.

```
> model1 <- stockModel(stockData)
> model2 <- stockModel(stockData, model='CCM')
...
```

3. Identify the optimal portfolio suggested by the model.

```
> optPort <- optimalPort(model1)
```

Overview

Presentation topics:

- Stock data, its format and where we get it.
- Basic investment terminology.
- Review several models offered in `stockPortfolio`.
- Simple but useful plots for investing.

These topics will be discussed in the context of `stockPortfolio`.

Obtaining data

We want stock returns, i.e. percent returns.

- Place stock tickers in a vector (`ticker`).
- `getReturns` retrieves stock returns from Yahoo Finance.
- Returns are in decimal form, i.e. a 10.3% gain means a return of 0.103.

```
> ticker <- c('C', 'BAC',
+   'JPM', 'WFC')
```

```
> gR <- getReturns(ticker,
+   start='2005-01-01',
+   end='2009-10-01')
```

Frequency of observations

The `getReturns` function can get daily, weekly, or monthly data.

```
> byDay <- getReturns(ticker, 'd', start='2005-01-01')
> byWk  <- getReturns(ticker, 'w', start='2005-01-01')
> byMth <- getReturns(ticker, 'm', start='2005-01-01')
```

The default is monthly data.

Warning: daily downloads = lots of data = longer download time.

Function declaration: `getReturns`

```
getReturns(ticker,
      freq = c("month", "week", "day"),
      get = c("overlapOnly", "all"),
      start = "1970-01-01",
      end = NULL)
```

Warning: setting `get="all"` often results in missing value problems.

Example

Some tickers are available in the `stock94Info` data.

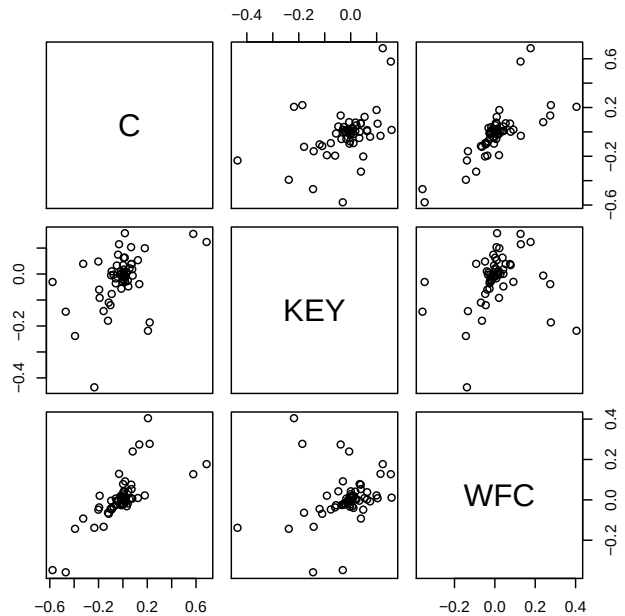
```
> data(stock94Info)
> (ticker <- stock94Info$ticker)[c(1:12, 25)]
 [1] "C"     "KEY"   "WFC"   "JPM"   "SO"    "DUK"
 [7] "D"     "HE"    "EIX"   "LUV"   "CAL"   "AMR"
[13] "^GSPC"

> ind <- stock94Info$industry # for later


> theData <- getReturns(ticker,
+   start='2004-09-31', end='2009-09-31')
```

The `print`, `summary`, and `plot` methods can be applied to `theData`.

Example, continued

The `print`, `summary`, and `plot` methods can be applied to `theData`:

```
> theData
...  matrix of stock returns ...
> summary(theData)
25 stocks, observed once per month
between 2004-10-01 and 2009-09-01

                    C    KEY    WFC ...
Mean Return -0.016 -0.018 0.0076 ...
> plot(theData) # ugly, too many
stocks
> pairs(theData$R[,1:3])
```



Risk-free investments

Other investments also exist, and the `stockPortfolio` takes one into account: risk-free investments. (Nearly) risk-free investments exist as

- Insured Certificates of Deposit
- US Treasuries
- Insured Savings Accounts

The risk-free rate of return, R_f, is the return attainable by these risk-free investments.

Argument name in `stockPortfolio`: `Rf`. The value of `Rf` is standardized for the period, e.g. 3% annual return equates to setting `Rf=0.0025` for monthly data.

Short selling

A portfolio is the allocation of money to stocks.

Investors are also permitted to borrow stocks from other investors, sell them, and use that money to invest in other stocks. This is called *short selling*.

Short selling will be referred to via `shortSelling` in function arguments, and it takes values `'y'` and `'n'`.

RIP Bear Stearns.

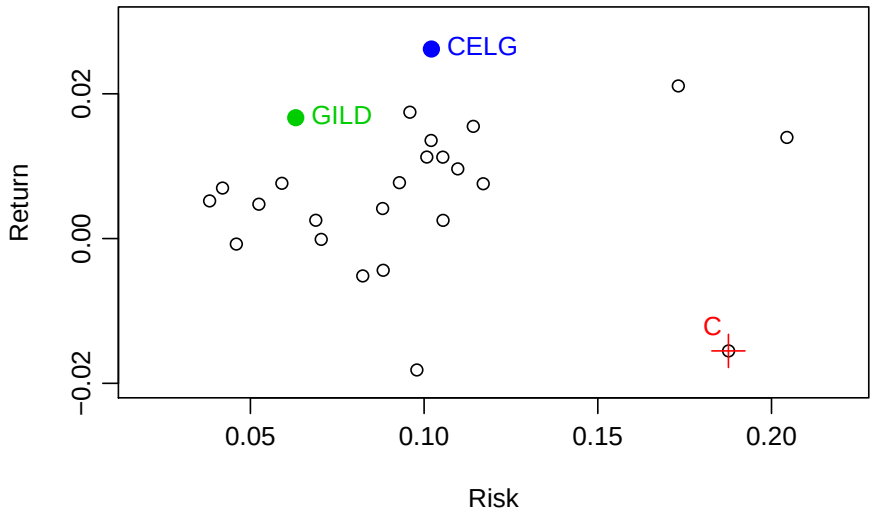
Minimize risk, maximize return

To gain an idea of how to minimize risk and maximize return, a first step is to look at two summary statistics:

- $\bar{R}$ vector of average returns for each stock.
- Σ variance-covariance matrix of returns.

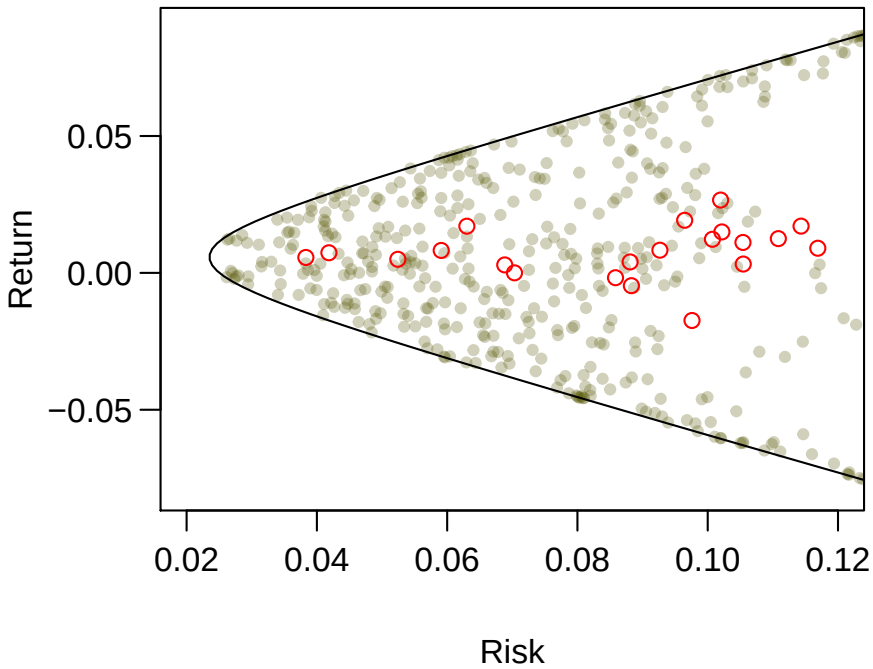
Risk is quantified as the standard deviation of the return.

To minimize risk while maximizing reward, $\bar{R}$ and Σ will be useful.



Portfolio possibiliites

By randomly construct many portfolios, a general view of the Risk-Return possibilities becomes apparent. The curve encapsulating the *portfolio cloud* (`portCloud`) is the *portfolio possibilities curve* (`portPossCurve`). The top half of this curve yields the best possible investments: the *efficient frontier*.



Modeling stocks

There are four models offered:

- **Variance covariance (default).** Use $\bar{R}$, Σ, and R_f to select a portfolio that minimizes risk but maximizes return.
- **Constant correlation model (CCM).** Smooth Σ and then do variance covariance method.
- **Multigroup model (MGM).** Compromise strategy: do some smoothing on Σ (less than CCM) and then optimize.
- **Single index model (SIM).** Use a linear model to analyze stock behavior, where we regress stock returns against some stock index.

Implementation

The 25^{th} "stock" – the S&P500 – is dropped for the first three models.

```
> model1 <- stockModel(theData, drop=25)
> model2 <- stockModel(theData, drop=25, model='CCM')
> model3 <- stockModel(theData, drop=25, model='MGM',
+   industry=ind)
> model4 <- stockModel(theData, model='SIM', index=25)
```

By default, `Rf=0` and `shortSelling='y'`. Short selling is always permitted when the model is the variance-covariance or multigroup model.

Single index model

The *Single Index Model* is the most well-known of the four models. If X_M describes the returns of the stock index and X_i describes the returns of stock i, then we propose a linear model that relates the two:

$$X_i = \alpha_i + \beta_i X_M + \epsilon_i$$

where α_i and β_i are constants and ϵ_i is a vector of the model errors for stock i. Example where no short selling is permitted:

```
> data(stock94Info)
> sim <- stockModel(theData, model='SIM', index=25,
+   industry=ind, shortSelling='n')
```

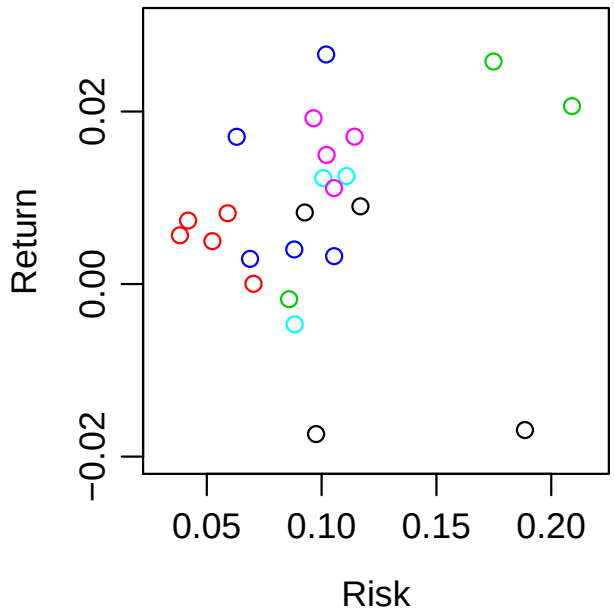
Examining a stockModel object

Basic information about a model is easily accessible.

```
> summary(sim)
Model:  SIM
60 observations, each one month apart
Dates:  2004-10-01 to 2009-09-01
Short selling is not permitted
```

```
> plot(sim)
```

Stock colors correspond to the industries.



Finding the optimal portfolio

For any model, the goal is to minimize risk while maximizing return. There is a single function to identify the optimal portfolio of a model: `optimalPort`.

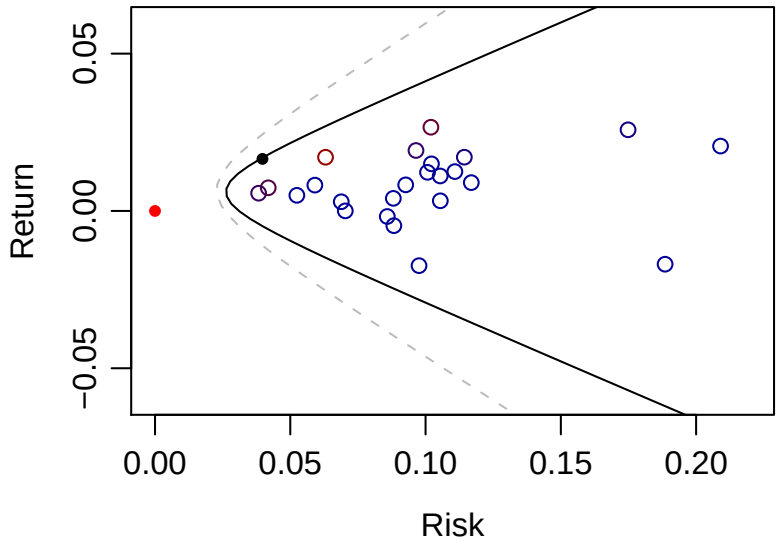
The first argument is an output of `stockModel`. The next two arguments permit adjustments to the model (`Rf` and `shortSelling`).

```
> simOP <- optimalPort(sim)

> summary(simOP)
Model:  single index model
Expected return:  0.0159
Risk estimate:    0.0399

> simOP
...  same output as above ...

Portfolio allocation:
...
```

Visualization of optimal portfolio

The optimal stock portfolio is shown by the black dot on the efficient frontier when no short selling is permitted. Allocation shown by heat coloring.

```
> plot(simOP, xlim=c(0,.2),
+   ylim=0.06*c(-1,1))
> portPossCurve(sim, 10,
+   add=TRUE)
> points(0, 0, pch=20, col=2)
```



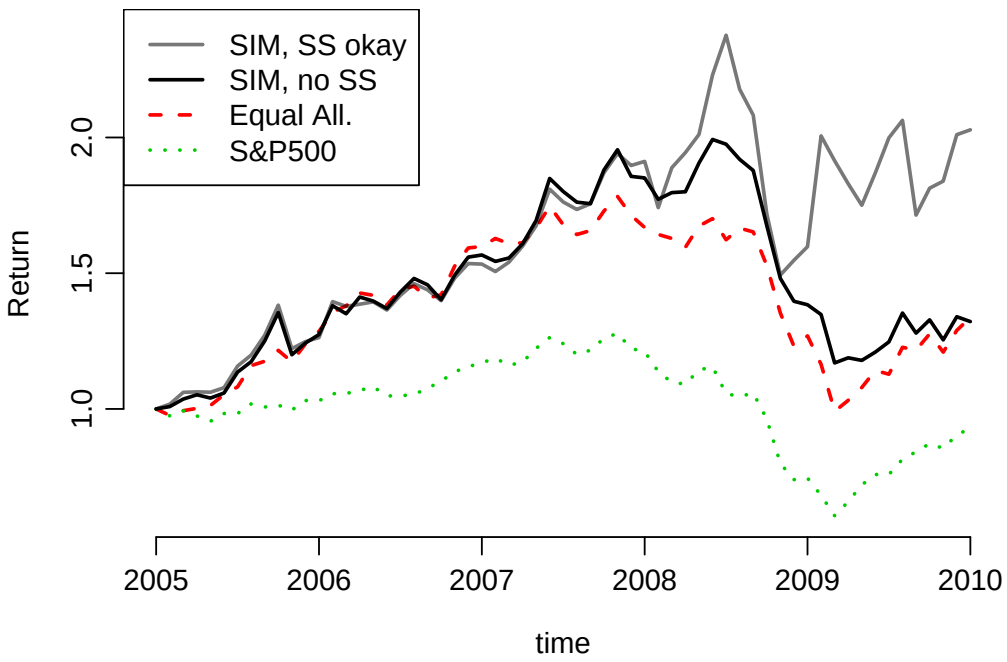
Omitted topics

Hopefully at this point creating a stock portfolio and finding the optimal allocation using one of the models would be a relatively simple task using `getReturns`, `stockModel`, and `optimalPort`. What was not covered:

- Finer details of the models.
- Comparison of models (`testPort` is useful in this respect).
- Creation of portfolio clouds (`portCloud`) and portfolio possibilities curves (`portPossCurve`).

Sample of testPort

Before farewells, a brief examination of the utility of these models (using testPort).



Models were fit annually and based on the most recent five years of data.