Statistical Arbitrage and High Frequency Trading Simulator and Visualizer

Team Winnie Cheng (wwcheng@alum.mit.edu) and Tirto Adji (tirto_adji@yahoo.com)

About this Document:
Documentation for Package TradeStrategyVisualizer (Tirto Adji). Please contact Winnie Cheng for documentation on the strategy simulator portion.

Overview:
Statistical Arbitrage is a set of trading strategies that make use of data mining techniques and observed statistical properties of groups of stocks to forecast future movements. This project aims at developing R packages and associated visualization tools for traders to analyze the effectiveness of their strategies.

The project is divided into two sections: TradeStrategySimulator and TradeStrategyVisualizer. The latter part of this document focuses on describing the TradeStrategyVisualizer portion of the project.

Visualization:
There are 2 R packages used for visualization: ggplot2 and googleVis. They are both powerful visualization tools that help graphically represent sets of data. They act as visual aids to the user to understand the data easier, especially when the data is abstract and large in size. In this case, we use them to generate charts to compare performance between different trading algorithms and stock prices over a time period, as well as movement of stock prices. The charts make use of S&P 500 stock data from the 2009-08-21 to 2010-08-20 time period.

The charts are available at
http://stanford.edu/~tadji/stat290/project.html

ggplot2:
We produce 6 charts using ggplot2.
1. Portfolio Holding Summary Chart
This chart compares the performance of different trading algorithms: mean reverting and random buy hold. It plots the holding value on the Y-axis and date on the X-axis. It shows that the mean reverting algorithm gives a higher overall return compared to the random buy hold algorithm. We also added summary statistics, a linear fit, for each algorithm. The R function that produces this chart also accepts optional caption text to annotate the graphs.
2. Stock Correlation Matrix Chart
This chart shows the degree of relationship (correlation) between two ticker symbols. It plots the top positions of positively correlated stocks. It uses ggplot2's geom_tile() and the computed correlation number as its fill.
3. Stock Correlation Pairs Cartesian Chart
This chart plots the top positions of positively correlated stocks in pairs. Each pair has its own scale and

it displays the price movement of the two tickers which are closely related. As a result, the stock pairs shown here have similar trends. It uses facet_grid() and geom_line() as the grammar of the graphics.

4. Portfolio Stock Performance Cartesian Chart

This chart displays stock price movements in our portfolio, each with a smoothed line that follows each original line, and a smoothed line that applies to the entire group. This way it's easier to identify which stocks are performing well, and which are not.

5. Portfolio Stock Prices BoxPlot

This boxplot chart is useful to identify outliers and the box plots are sorted from low to high stock prices from left to right. The users can quickly glance and detect anomaly, if any, in the stock prices. Larger boxplots mean more skewed distributions.

6. Stock Calendar Heat Map

This chart represents the individual stock prices time series data as a calendar with days filled with colors representing the values. Red means higher prices, green means lower prices. The chart makes it an interesting way to look at financial time series data. The users can quickly scan the chart and see the data partitioned by week, week day, and year. It uses geom_tile(), scale_fill_gradient() and facet_wrap() as the ggplot2's grammar.

googleVis:

As an experiment, we use the googleVis package to display the same data. We use annotated time line and motion charts. Here are their short descriptions:

1. Annotated Time Line Chart

This chart dynamically displays the movement of stock prices and event annotations (BUY/SELL) at any point in time. At the bottom at the chart, there is a slider that can be used to zoom in and out to a specific time period.

2. Motion Chart

This chart dynamically shows the performance of the different algorithms used in the trade strategy simulator. The chart allows users to select and explore several variables over time. It uses animation to depict the movement of the variables. The motion chart measures the growth rate of one portfolio's value using a specific algorithm compared to the other.

Comments on Initial Proposal:

The initial intention was to build a web-based visualization which will call R packages and utilize more visualization techniques, but this proved to be rather challenging. Hence, we limit the scope of this class project to deliver the visualization only from R.

We are happy with the architectural decision to make the sqlite database as the interface between Winnie Cheng's TradeStrategySimulator and Tirto Adji's TradeStrategyVisualizer packages. It decouples our tasks and made our life easier. We agreed upon a pre-defined data elements and structure and work independently. And when the time comes, we just hooked up our data set and it flows nicely from one package to the next one. We opted to store our data in the database to make it easier to exchange data between the two packages. We are doing many disk reads/writes. This is okay since our data size is

small, but may not be scalable for a large data sets. We are cognizant about this and chose to not focus on performance as a trade-offs for ease of implementation.

Comments on visualization tools:

ggplot2 makes it easier to think about data graphically, the real genius is the grammar of the graphics. It is relatively easy to learn and the grammar is very powerful to express the data in many different ways. The documentation is also superb, although it would take some time to master all of the nuances of the tool.

googleVis is also an excellent tool to convert graphics in R to their interactive versions on the web. It provides an interface between R and google's visualization API, allowing users to easily access the data, identify trends, and gain insights from the data. It is a rather novel package, thus we were having minor problems with our version of googleVis. We experimented by opening the source code and applied a minor tweak to it to make it work as we intended.

Examples and Tests included in the package:

To run the examples in the package, please install the package and execute the following:

```
> library(TradeStrategyVisualizer)
> demo(TradeStrategyVisualizer)
```