

Backtest Graphics

by David Kane, Ziqi Lu, Karan Tibrewal, Fan Zhang, Miller Zijie Zhu

Abstract The **backtestGraphics** package provides an interactive graphical interface to visualize backtests results for a variety of financial instruments (equities, futures, credit default swaps, et cetera). It provides handy facilities for visualizing backtest data, including:

- A panel to display summary performance statistics, like number of instruments, cumulative and annualized profit and loss, Sharpe ratio and the best and worst performing months.
- A panel to display detailed performance statistics, like top three drawdowns, and the three best and worst performers in the backtest.
- Interactive plots of profit and loss (P&L), net market value (NMV) and gross market value (GMV).
- Seamless subsetting of backtest data to accommodate for visualization of multiple strategies, sub-strategies, and overlapping portfolios that may be present in a single backtest result.

Introduction

Backtesting is the process of testing trading strategies on prior time horizons to measure the effectiveness of the given strategy. It helps investors understand and optimize their trading strategies [Campbell et al. (2008)]. For more on backtesting, *Quantitative Value* [Gray and Carlisle (2013)] is a good introductory book.

Data visualization is not a substitute to analysis; instead, it is a tool to enable users to employ their human perception to process lots of data quickly and approximately [Bostock (2012)]. Accordingly, **backtestGraphics** is not designed to run backtests, but to visualize backtest data efficiently. We provide fast and flexible tools that help in detecting trends and anomalies in the data. Our package returns a **Shiny** interface that allows the user to visually explore her backtest results [Chang et al. (2015)]. The interface contains a sidebar panel which includes “Summary” and “Detail” tabs. The “Summary Tab” provides the user with summary statistics, such as average gross market value (GMV), number of instruments, cumulative and annualized profit and loss (P&L), Sharpe ratio and best and worst performing months. The “Detail” tab provides the user with information about the top three drawdowns and the three best and worst performers in the backtest. The main panel of the interface houses **dygraph** plots of cumulative P&L, daily P&L, NMV and GMV [Vanderkam and Allaire (2015)]. The interactive nature of these plots provide the user with additional flexibility in exploring her backtest results by enabling her to zoom into specific time periods, or learn the value of a given response variable on a specific date.

Backtests vary in complexity. A simple backtest employs a single strategy in a single portfolio, while a more complex backtest might use multiple strategies and sub-strategies. In order to accommodate for this, our package provides three dropdown menus that allow the user to subset her backtest results in accordance to such overlapping portfolios for different (sub)strategies at different time periods.

Data

The package comes with three sample data frames: **commodity**, **equity**, and **credit**. These are backtest results for commodity futures, equities, and credit default swaps (CDS), respectively. We use these data frames to demonstrate the capabilities of the backtestGraphics package. (*Note: The user may also use her own data frame, as long as it adheres to the format specified in the documentation*).

Let us begin with some details of **equity**:

Source: local data frame [9 x 5]

| | name (chr) | date (date) | sector (chr) | nmv (dbl) | pnl (dbl) |
|---|---------------|----------------|-----------------|--------------|--------------|
| 1 | A.Z | 2005-05-02 | MAT | -131926.1 | 3158.4525 |

```

2 ABFS 2005-05-02 IND -131926.1 -4058.6214
3 ABS. 2005-05-02 CNS -131926.1 -733.2929
4 A.Z 2005-05-03 MAT -128767.7 3644.3684
5 ABFS 2005-05-03 IND -135984.7 -335.9721
6 ABS. 2005-05-03 CNS -132659.4 733.2935
7 A.Z 2005-05-04 MAT -125123.3 -1457.7478
8 ABFS 2005-05-04 IND -136320.7 -1595.8684
9 ABS. 2005-05-04 CNS -131926.1 -2466.5317

```

- [,1] **name** column contains the name of each stock.
- [,2] **date** column contains the trading date. (*Note: this column must be of Date type*).
- [,3] **sector** column contains the sector of each stock.
- [,4] **nmv** column contains the NMV at the beginning of the trading day for each stock. (*Note: all values must be converted to the same currency*).
- [,5] **pnl** column contains P&L of each stock. (*Note: all values must be converted to the same currency*).

Now, let us look at **commodity**:

Source: local data frame [5 x 11]

| | name (chr) | id (chr) | date (date) | sector (chr) | portfolio (int) | strategy (chr) | substrategy (chr) |
|---|---------------|-------------|----------------|-----------------|--------------------|-------------------|----------------------|
| 1 | brent | CO | 2003-02-03 | energy | 1 | Strategy 1 | Strategy 1.1 |
| 2 | brent | CO | 2003-02-03 | energy | 6 | Strategy 1 | Strategy 1.2 |
| 3 | cattle feeder | FC | 2003-02-03 | livestock | 1 | Strategy 3 | Strategy 3.1 |
| 4 | cbt wheat | W | 2003-02-03 | agriculture | 1 | Strategy 3 | Strategy 3.2 |
| 5 | cbt wheat | W | 2003-02-07 | agriculture | 1 | Strategy 3 | Strategy 3.2 |

Variables not shown: **gmV** (dbl), **nmv** (dbl), **pnl** (dbl), **contract** (dbl)

The columns **name**, **date**, **sector**, **nmv**, and **pnl** have the same meaning as the ones in **equity**. The remaining columns are listed below: (*Note: these columns are optional*).

- [,2] **ID** column contains ID information for different commodities.
- [,6] **strategy** column contains the strategy number.
- [,7] **substrategy** column contains the sub-strategy number.
- [,8] **gmV** column contains the GMV of a given commodity at the start of each trading day (*Note: all values must be converted to the same currency*).
- [,11] **contract** column contains the number of contracts of a given commodity at the start of each trading day.

Finally, we look at **credit**:

Source: local data frame [10 x 7]

| | name (chr) | date (date) | sector (chr) | strategy (chr) | gmV (dbl) | nmv (dbl) |
|----|---------------|----------------|-----------------|-------------------|--------------|--------------|
| 1 | AA.Senior.5 | 2008-10-01 | Basic Materials | daily | 9e+06 | 259259.3 |
| 2 | AA.Senior.5 | 2008-10-01 | Basic Materials | monthly | 9e+06 | 259259.3 |
| 3 | AA.Senior.5 | 2008-10-01 | Basic Materials | quarterly | 9e+06 | 259259.3 |
| 4 | ABY.Senior.5 | 2008-10-01 | Basic Materials | daily | 9e+06 | 259259.3 |
| 5 | ABY.Senior.5 | 2008-10-01 | Basic Materials | monthly | 9e+06 | 259259.3 |
| 6 | ABY.Senior.5 | 2008-10-01 | Basic Materials | quarterly | 9e+06 | 259259.3 |
| 7 | AA.Senior.5 | 2008-10-02 | Basic Materials | daily | 9e+06 | 170731.7 |
| 8 | ABY.Senior.5 | 2008-10-02 | Basic Materials | daily | 9e+06 | 170731.7 |
| 9 | AA.Senior.5 | 2008-10-03 | Basic Materials | daily | 9e+06 | 170731.7 |
| 10 | ABY.Senior.5 | 2008-10-03 | Basic Materials | daily | 9e+06 | 170731.7 |

Variables not shown: **pnl** (dbl)

Most of the columns here are the same as those in **equity** and **commodity**. Notice that the **strategy** column here is used to denote backtest results for different trading frequencies. Towards this, the user must provide all backtest observations for each instrument, even when the position on the instrument is held and unchanged. This is because **backtestGraphics** uses the time gaps between each observation to determine the trading frequencies, and to calculate the annualized statistics.

Using the backtestGraphics Interface

In this section, we discuss the functionality of various components of the **Shiny** interface returned by the **backtestGraphics** function. To use **backtestGraphics**, the user is required to pass in a data frame with **date**, **ID/name**, **NMV** and **P&L** columns to the **backtestGraphics** function. Additionally, the user can also pass in optional columns, such as **sector**, **GMV**, **contracts**, **strategy**, **substrategy**, and **portfolio**.

As an example, let us look at the interface for the **commodity** data frame. Type the following command, and click the “Visualize” button on the Shiny interface returned by the function call.

```
library(backtestGraphics)
backtestGraphics(x = commodity)
```

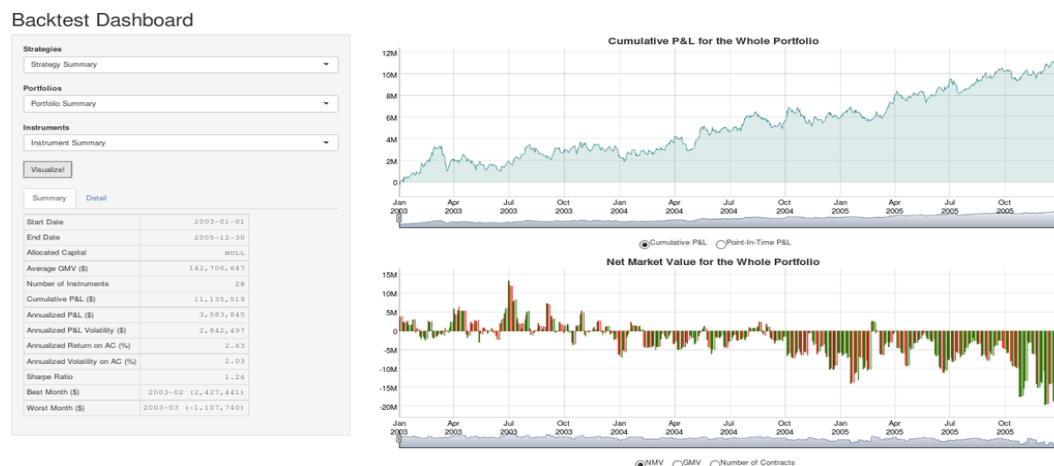


Figure 1: The **backtestGraphics** interface for the **commodity** data frame. The interface houses three dropdown menus on the top-left to toggle between Strategies, Portfolios, and Instruments. The user may click on the “Visualize” button to look at the summary statistics and the interactive plots. Finally, the user can also switch between plots with the radio buttons at the bottom of the plots.

Now, let us individually examine the various components of the Shiny interface:

Dropdown Menus

The interface contains three dropdown menus on the top-left to toggle between different Strategies, Portfolios, and Instruments. By selecting different combinations of these menus, the user can visualize different subsets of strategies, portfolios and instruments.

If the given data frame does not have a **strategy** column, or a **portfolio** column, the respective dropdown menus will be fixed to “Strategy Summary” and “Portfolio Summary”. If the user selects a combination that is incompatible with the given data frame, the interface will generate an error message and prompt the user to select a different combination.

Figure 2 provides a screenshot of the interface’s dropdown menus.

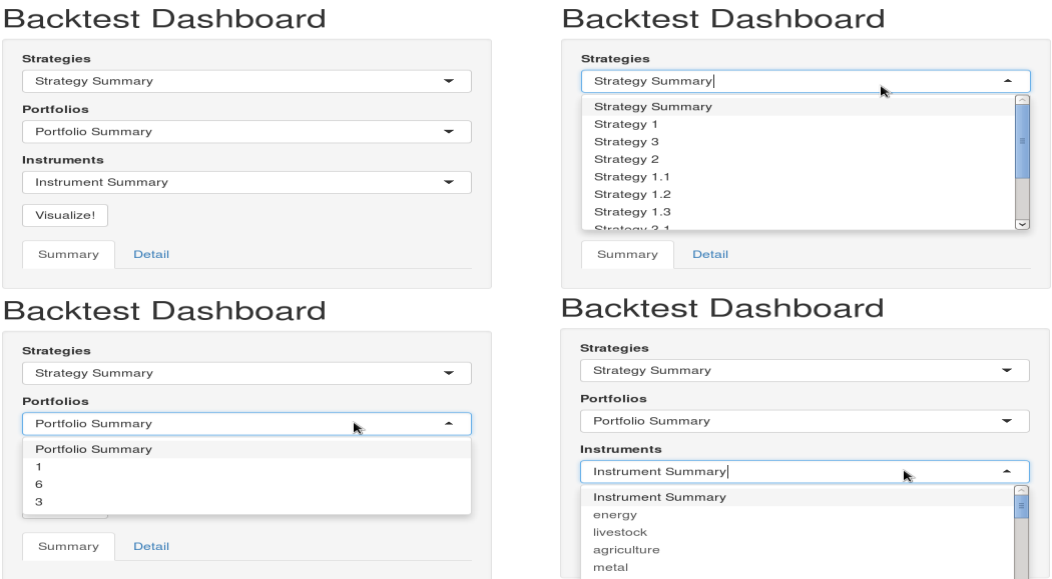


Figure 2: Dropdown menus: The user can subset her backtest results along different strategies, portfolios and instruments by selecting appropriate choices from the dropdown menus.

The “Summary” Tab

The “Summary” tab is present in the left-panel of the **Shiny** interface. It provides summary statistics for the backtest result. Below is a screenshot of the “Summary” tab for our commodity data frame, along with corresponding documentation.

| | |
|---------------------------------|--------------------------|
| Start Date | 2003-01-01 |
| End Date | 2005-12-30 |
| Allocated Capital | NULL |
| Average GMV (\$) | 142,706,647 |
| Number of Instruments | 28 |
| Cumulative P&L (\$) | 11,135,519 |
| Annualized P&L (\$) | 3,583,845 |
| Annualized P&L Volatility (\$) | 2,842,497 |
| Annualized Return on AC (%) | 2.63 |
| Annualized Volatility on AC (%) | 2.03 |
| Sharpe Ratio | 1.26 |
| Best Month (\$) | 2003-02 (2,427,441) |
| Worst Month (\$) | 2003-03 (- 1,107,740) |

Figure 3: The “Summary” tab for the `commodity` data frame. On clicking the “Visualize” button, the function slices the data frame according to inputs in the drop down menus to calculate summary statistics for the specified data subset.

Documentation of summary statistics:

- Start and End date : The backtest period.
- Allocated Capital : The amount of capital allocated to the portfolio. We use this number to calculate return on allocated capital. If the user does not specify this value, the corresponding entry will be NULL and our package will use the highest GMV to calculate return on allocated capital.
- Average GMV : Average GMV of the portfolio over the backtest period. GMV is calculated by taking the absolute value of NMV of each instrument.
- Number of Instruments : The number of different instruments in the given subset of the data frame.
- Cumulative P&L : Sum of all P&L over the backtest period.
- Annualized P&L : This is calculated as the Average P&L times an annualization factor. Our package observes the date interval between observations in the data frame to determine the trading frequency. The assumed annualization factor of daily trading is $\sqrt{252}$, while that for other trading frequency is $\sqrt{\frac{365}{\text{date gap}}}$.
- Annualized P&L volatility : The standard deviation of all annualized P&L's.
- Annualized Return : Annualized average return on allocated capital. We calculate average return rates by dividing P&L by the allocated capital amount. The average return rates are then annualized by multiplying them with the annualization factor.
- Sharpe Ratio : A measure of risk-adjusted return. We calculate this by dividing the mean P&L by the standard deviation of P&L. We assume that risk-free rate = 0.
- Best Month and Worst Month : The month with the highest P&L, and the month with the lowest P&L

The “Detail” Tab

The “Detail” tab is also present in the left-panel of the **Shiny** interface. It provides information about the best and worst performers, as well as the biggest drawdowns. Below is a screenshot of the “Detail” tab for our `commodity` data frame, along with corresponding documentation.

| Top Three Drawdowns: | | |
|-------------------------|------------|------------|
| Start Date | End Date | P&L (\$) |
| 2003-03-11 | 2003-06-18 | -2,306,913 |
| 2003-10-30 | 2004-01-09 | -1,770,827 |
| 2004-10-11 | 2004-11-15 | -1,732,102 |
| Three Best Performers: | | |
| Tickers | P&L (\$) | |
| JX: tcom kerosene | 2,780,786 | |
| FC: cattle feeder | 1,730,847 | |
| CL: wti | 1,285,581 | |
| Three Worst Performers: | | |
| Tickers | P&L (\$) | |
| SB: sugar | -1,085,512 | |
| QW: white sugar | -838,242 | |
| LH: lean hogs | -579,077 | |

Figure 4: “Detail” tab with the `commodity` data frame.

Documentation of detailed statistics:

- Top three drawdowns : The three biggest declines in P&L from peak to trough. The table contains the start and end dates of the drawdowns, along with the actual value of the drawdowns.
- Best and worst three performers : The three best performers are the instruments with the highest three cumulative P&L's. Similarly, the three worst performers are the instruments with the lowest three cumulative P&L's. (Note: even if the user is looking at a subset of individual instruments, we will display information about the best and worst performers across all instruments. However, if the user is looking at data from a specific sector, we will display information about the best and worst performers in the specified sector).

Plots

The right-panel of our **Shiny** interface provides interactive plots for cumulative and point-in-time P&L, NMV, GMV and number of contracts.

Figure 5 provides a screenshot of the plots.



Figure 5: A screenshot of the plots with the `commodity` data frame. The P&L plots is displayed on the top, while the market value plots are on the bottom. Radio buttons at the bottom of the plots allow the user to seamlessly switch between plots.

We have made the plots interactive using the **dygraphs** package [Vanderkam and Allaire (2015)]. The user can zoom in to a specific time periods in a plot by dragging the corresponding region with the mouse, or changing the time slider bar, which is located below the plots. To go back to the original time scale, the user can merely double click on the respective plot. Additionally, by hovering the mouse over the plot, the user can get specific values of the response variable of the plot, on the specific date. The color of the plots is based on the sign of the point-in-time P&L. Green bars represent profits, while the red bars represent losses.

Conclusion

The **backtestGraphics** package provides a simple **Shiny** interface to visualize backtest results. Inside the package, we have provided three sample data frames. The user can use the code provided in this document to test the package with these data frames.

Our interface provides the user with summary, as well as detailed statistics about the backtest result. The user can subset her data according to strategies, sub-strategies, and instruments. Additionally, we also provide interactive plots that show movement of important variables over the backtest period.

These are powerful tools to uncover trends and anomalies in the data. By decomposing the data frame into different sectors and instruments, the user can gain important insights into his trading strategies over different sectors and time periods. The interactive plots provide additional flexibility to dive into the nitty-gritty details of the backtest.

Going forward, the **backtestGraphics** package can easily be customized to suit the needs of a specific user. Such customization may include, among others, adding new response variables to plots, redesigning plots, or introducing additional summary statistics. We invite users to take advantage of this, and redesign the **backtestGraphics** interface according to their preferences.

Authors

David Kane
Harvard University
IQSS
1737 Cambridge Street
CGIS Knafel Building, Room 350
Cambridge, MA, USA
dave.kane@gmail.com

Ziqi Lu
Economics and Mathematics
Williams College
Williamstown, MA, USA
ziqi.lu@williams.edu

Karan Tibrewal
Mathematics and Computer Science
Williams College
Williamstown, MA, USA
karan.tibrewal@williams.edu

Fan Zhang
Economics and Statistics
Williams College
Williamstown, MA, USA
fan.zhang@williams.edu

Miller Zijie Zhu
Computer Science and Economics
Williams College
Williamstown, MA, USA
zijie.miller.zhu@gmail.com

Bibliography

- M. Bostock. *Time Series Visualization with Cubism.js*, 2012. URL <https://bost.ocks.org/mike/cubism/intro/#0>. [p1]
- K. Campbell, J. Enos, D. Gerlanc, and D. Kane. Backtests. *R News*, 7(1), MAY 2008. [p1]
- W. Chang, J. Cheng, J. Allaire, Y. Xie, and J. McPherson. *shiny: Web Application Framework for R*, 2015. URL <http://CRAN.R-project.org/package=shiny>. R package version 0.12.1. [p1]
- W. Gray and T. Carlisle. *Quantitative Value*. John Wiley & Sons, Inc., Hoboken, New Jersey, 2013. [p1]
- D. Vanderkam and J. Allaire. *dygraphs: Interface to Dygraphs Interactive Time Series Charting Library*, 2015. URL <http://CRAN.R-project.org/package=dygraphs>. R package version 0.4.5. [p1, 6]