Pair Trading and Portfolio Construction

APPLICATION NOTE

Insightful

www.insightful.com

Table of Contents

Abstract ____ 3

Introduction ____ 3
- The Hedge Fund Industry ____ 3
- Pair Trading ____ 3
- Market Neutral Long-Short Portfolio ____ 4
- Overview of this Application Note ____ 4

Section 1: Reading and Preparing Historical Stock Price Data ____ 5

Section 2: Constructing and Exploring Stock Pairs Data ____ 12
- Interpolating Missing Values ____ 12
- Creating Pair Price Ratios ____ 14
- Plotting Pairs ____ 16

Section 3: Constructing an Optimal Long-Short Portfolio ____ 19
- Selecting Weekly Data ____ 20
- Handling Missing Values ____ 21
- Optimizing Long-Short Portfolios ____ 23

Section 4: Assessing Portfolio Performance ____ 25
- Assessing Historical Portfolio Performance ____ 27
- Assessing Future Portfolio Performance ____ 29

Conclusion and Additional Possibilities ____ 31

References ____ 32
- References used in this Application Note ____ 32
- General References ____ 32
- Selected S and S-PLUS Related Books ____ 33
- Web Links ____ 33

DISCLAIMER ____ 34

About Infotec ____ 34

About Oxquant Consulting ____ 34

About Insightful Corporation ____ 34

Contact Us ____ 35

Pair Trading and Portfolio Construction

Dr. Frank Block

Abstract

This Application Note addresses pair trading strategies and long short portfolio construction in the context of hedge fund businesses. The goal is to demonstrate how to explore stock pairs data, detect pair trading signals and construct a mean-variance efficient long-short portfolio using Insightful analytical software. Further extensions of the application are discussed in the concluding remarks.

Introduction

The Hedge Fund Industry

According to Hedge Fund Research Inc.'s 2002 Industry Report, worldwide hedge funds have grown from $39 billion in assets under management (AUM) in 1990 to more than $622 billion in 2002. The number of funds has increased from roughly 530 in 1990 to more than 4,600 in 2002. (Both dollar AUM and number of funds exclude fund-of-funds operations.) Clearly, hedge funds are a rapidly growing industry.

As hedge funds have evolved, so too have the trading strategies employed. In the early 1980s, long/short equity, managed futures and equity hedge were some of the most popular trading strategies employed. The most successful players usually ended-up evolving into global macro funds, e.g. George Soros (Quantum Fund) and Julian Robertson (Tiger Fund); see Lhabitant (2002, Ch.10). More recently, new strategies such as quantitative credit arbitrage and macro equilibrium models have appeared on the scene, We also note the emergence of fund-of-funds, for which the correct hedge fund strategy mix is of prime importance (see Indjic, 2002).

Pair Trading

Pair trading is a popular strategy in the market-neutral-funds sector of hedge funds. It is particularly powerful when markets are volatile and do not show an overall trend.

The idea behind pair trading is that two stocks that have shown a high correlation in the past are expected to continue to do so in the near future. Therefore, when the spread between the pair increases, we expect it to diminish after some time. Consequently, for our trading behavior, this implies that when the spread widens we should short the outperformer and buy the underperformer. When the spread decreases, again we should profit from our positions.

Implicit in this idea is the concept of mean reversion, i.e. asset prices will revert to their long term means if long term means exist (or the time series are stationary statistically). Given mean reversion we can construct a signal detection mechanism that tells us when an asset's price has drifted too far from its mean and it is time for us to trade.

In this Application Note, we will not show in detail how the pairs are found (stock picking). There are many approaches for doing this and high performing methods are well kept secrets. We'll give a brief description of some ideas for the pair selection process following Frazzo et al. (2002), according to whom potential trading pairs can be selected in a four-step process:

1. Look at the historical correlation of stocks within the same sector. A pair candidate should have a high long-term correlation (6 months – 1 year) and a lower short-term (1-2 weeks) correlation.
2. Check whether the current price ratio of the two stocks is about two standard deviations away from its long-term mean.
3. Test whether the ratio is mean-reverting and stationary. This looks at the likelihood for this process to return to the mean once it's about two standard deviations away from it. Frazzo et al. suggest a Vasicek test, which not only tests for mean reversion but also for stationarity.
4. Check that the analysis of fundamentals of the selected pair supports the statistical analysis. This includes knowledge that goes beyond pure history and statistical analysis into the stock picking process, since history may sometimes not repeat itself.

For illustration purposes, we will explore the time series behavior of stock pairs data, and use two standard deviations away from a long-term mean as the signal to trigger pair trading.

Market Neutral Long-Short Portfolio

Classic hedge funds are characterized by their use of short-selling, leverage, derivatives and portfolio concentration. Use of a quantitative strategy such as pair trading usually results in high degree of leverage and capital concentration. Risk control from a portfolio perspective is therefore important.

A market neutral long-short portfolio typically balances the longs and shorts carefully to eliminate all the risk exposures. For example, it holds long positions together with offsetting short positions so that the net position is cash neutral. The example dataset used in this Application Note has nine long stock positions paired with nine short positions. We demonstrate how to construct such a portfolio so that the fund risk is minimized with zero net cash position.

Overview of this Application Note

This Application Note closely follows an example shown in the Insightful webcast called "Enhanced Hedge Fund Performance via Quantitative Risk Management" by Dr. Richard Saldanha (Oxquant) in June 2003.

The four sections of this Application Note cover:

- Reading and preparing historical stock price data
- Constructing and exploring stock pairs data
- Constructing an optimal long-short portfolio
- Assessing the performance of optimized versus non-optimized portfolios

A new approach is shown for solving the pair trading problem based on the integrated visual and scalable programming environment consisting of Insightful Miner, S-PLUS, S+FinMetrics, and S+NuOPT.

We take a completely new approach to solve this problem by using Insightful products: Insightful Miner, S-PLUS®, S+FinMetrics™ and S+NuOPT™. This enables us to integrate financial data with cutting-edge techniques for exploration and analysis. The vast range of available graphical and modeling capabilities are used to explore universes of tens of thousands investment instruments, understand causes and effects, and predict future behavior. Based on the comprehensive set of cutting-edge statistical and financial analysis functions, predictive models and large portfolio optimizations can be easily run and assessed with back-testing techniques. Insightful products also provide an easy-to-use visual programming environment, making the process of transforming and integrating multiple data sources a simple task for very large data sets.

The combined data analysis capabilities of all Insightful finance products can be accessed through Insightful Miner. This note shows the intuitive ease and flexibility of Insightful Miner's visual programming paradigm, which makes complex data analysis applications easier to understand and share with others. Using Insightful Miner's highly scalable architecture, many large-data problems can be handled with ease. When the example in this note is complete, we obtain a workflow that looks like Figure 1, below.

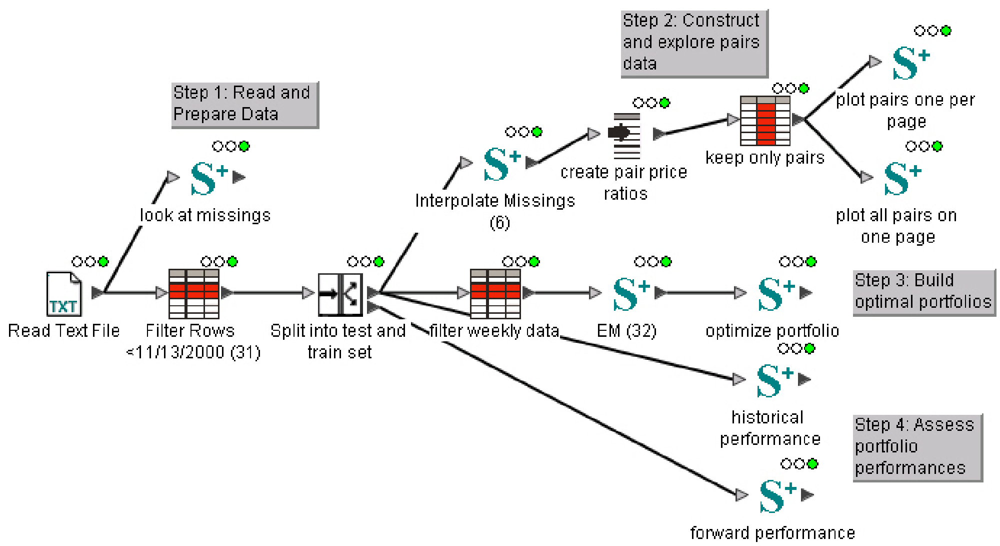


Figure 1

Section 1: Reading and Preparing Historical Stock Price Data

First, we read historical stock price data from a data source[1], which is used for subsequent analysis. After reading the data, some data preparation and transformation are required. Specifically, we will:

- Read stock price data from **daily.csv**, a text file with historical price data on 18 stocks from March 31, 1997 until May 27, 2003. These 18 stocks are in pairs, i.e. half of them are for long only and the other half for short only[2]. The data is comma separated, with the first column containing the date.

---

[1] The data used in our analysis and in the file **daily.csv** is provided by Infotec SA, Geneva, Switzerland.
[2] We suppose the direction of trade for the 18 stocks are pre-determined based on fundamental research results and / or some other external constraints.

- Analyze missing values for each stock price time series. This determines how we subset our data to avoid large gaps in available data.
- Select a subset of the time series for subsequent analysis. This is a consequence of the preceding missing value analysis, which reveals that prior to November 13, 2000, data is missing for most stocks.
- Split the data into test and train data sets. The train set is used for constructing and exploring pairs behavior, building optimal portfolios (under the assumption that the stocks are to be traded in long-short pairs), and assessing historical portfolio performances. The test set is used for out-of-sample testing of the portfolio performance.

Open up a new Insightful Miner worksheet, and double-click a **Read Text File** node (Read Text File) to move it into the empty worksheet. Double-click the node to bring up the Properties dialog, and then click the **Browse** button and navigate to **daily.csv**. Set the rest of the dialog parameters[3] to match those of the screen shot in Figure 2, and then press the **Update Preview** button.

With Insightful Miner's powerful data reading capabilities, it is easy to access data from various sources and to transform raw data into formats suited for analysis.

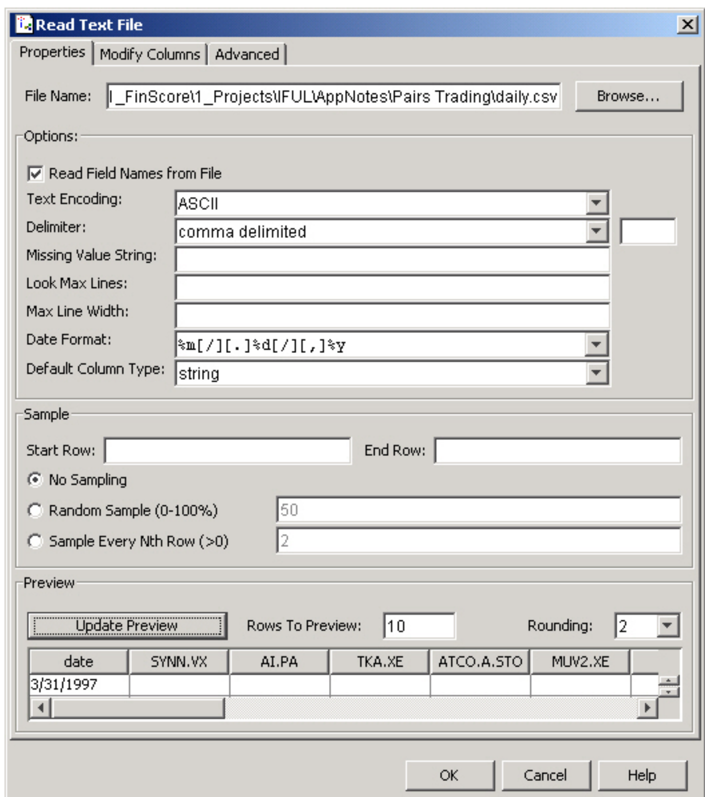


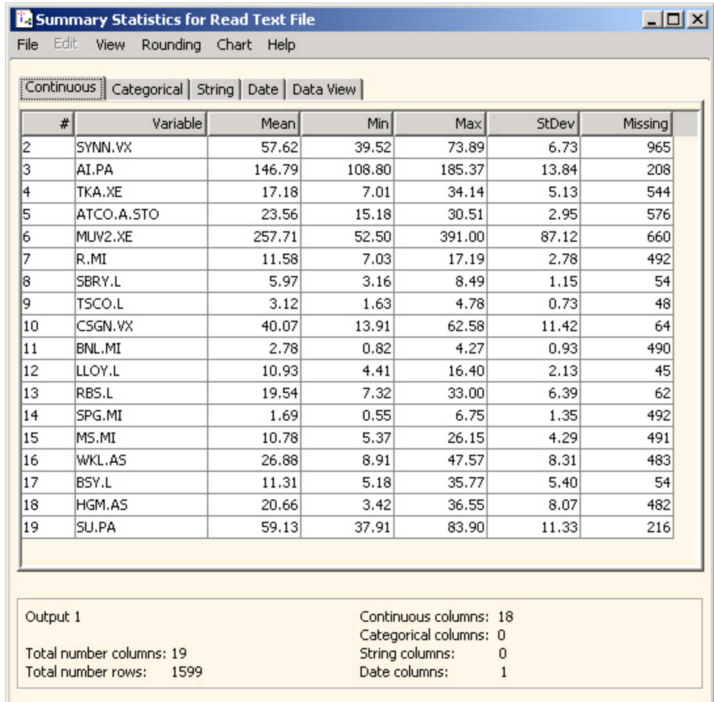
Figure 2

In the preview pane, we notice, with the exception of the **date** field, all other fields appear to be missing. We provide more details on this later.

Run the **Read Text File** node by right-clicking it and selecting **Run to Here**. Check the data was correctly read by right-clicking the node and selecting **Viewer**. The summary statistics for this node are shown in Figure 3.

---

[3] Please note that we have defined a date format different from the suggested default to match the format found in **daily.csv**.

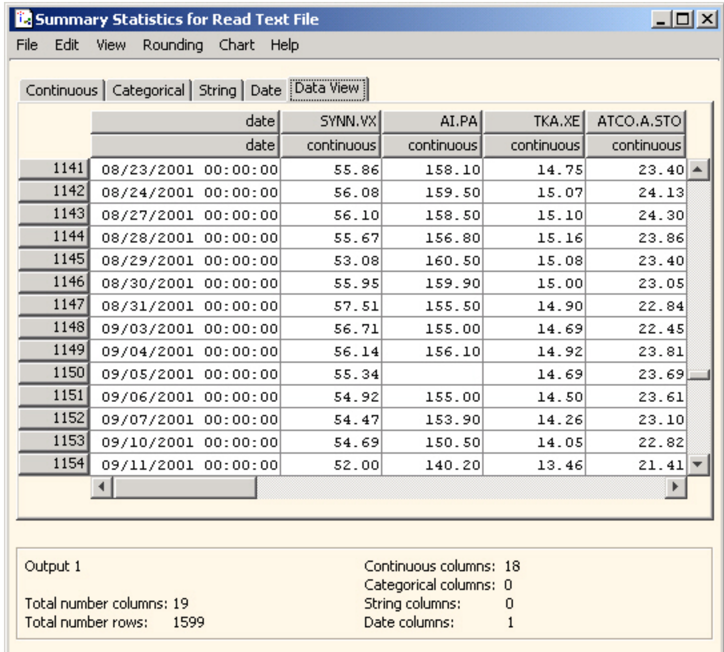


#	Variable	Mean	Min	Max	StDev	Missing
2	SYNN.VX	57.62	39.52	73.89	6.73	965
3	AI.PA	146.79	108.80	185.37	13.84	208
4	TKA.XE	17.18	7.01	34.14	5.13	544
5	ATCO.A.STO	23.56	15.18	30.51	2.95	576
6	MUV2.XE	257.71	52.50	391.00	87.12	660
7	R.MI	11.58	7.03	17.19	2.78	492
8	SBRY.L	5.97	3.16	8.49	1.15	54
9	TSCO.L	3.12	1.63	4.78	0.73	48
10	CSGN.VX	40.07	13.91	62.58	11.42	64
11	BNL.MI	2.78	0.82	4.27	0.93	490
12	LLOY.L	10.93	4.41	16.40	2.13	45
13	RBS.L	19.54	7.32	33.00	6.39	62
14	SPG.MI	1.69	0.55	6.75	1.35	492
15	MS.MI	10.78	5.37	26.15	4.29	491
16	WKL.AS	26.88	8.91	47.57	8.31	483
17	BSY.L	11.31	5.18	35.77	5.40	54
18	HGM.AS	20.66	3.42	36.55	8.07	482
19	SU.PA	59.13	37.91	83.90	11.33	216

Figure 3

This summary shows there are 19 data columns, of which 18 are of type **continuous** (the stock price data) and one is of type **date**. We also see there are 1599 rows of data available. Clicking the **Date** tab reveals that the time span of the data is March 31, 1997 to May 27, 2003. We also notice that each data column has a large number of missing values.

Next, we look at a snapshot of the data read. To this end, we click the **Data View** tab and obtain Figure 4.



	date	SYNN.VX	AI.PA	TKA.XE	ATCO.A.STO
	date	continuous	continuous	continuous	continuous
1141	08/23/2001 00:00:00	55.86	158.10	14.75	23.40
1142	08/24/2001 00:00:00	56.08	159.50	15.07	24.13
1143	08/27/2001 00:00:00	56.10	158.50	15.10	24.30
1144	08/28/2001 00:00:00	55.67	156.80	15.16	23.86
1145	08/29/2001 00:00:00	53.08	160.50	15.08	23.40
1146	08/30/2001 00:00:00	55.95	159.90	15.00	23.05
1147	08/31/2001 00:00:00	57.51	155.50	14.90	22.84
1148	09/03/2001 00:00:00	56.71	155.00	14.69	22.45
1149	09/04/2001 00:00:00	56.14	156.10	14.92	23.81
1150	09/05/2001 00:00:00	55.34		14.69	23.69
1151	09/06/2001 00:00:00	54.92	155.00	14.50	23.61
1152	09/07/2001 00:00:00	54.47	153.90	14.26	23.10
1153	09/10/2001 00:00:00	54.69	150.50	14.05	22.82
1154	09/11/2001 00:00:00	52.00	140.20	13.46	21.41

Figure 4

In this snapshot, there is an example of a missing value on September 9, 2001 for the stock **AI.PA**.

Let's look at missing values in detail. We'll create an S-PLUS node extension that visualizes missing data in the stock price time series using an "image plot" representation.

On Insightful Miner's tool tab labeled **S-PLUS**, we find an icon named **S-PLUS Script** (S+ S-PLUS Script), which we use to create the S-PLUS node for creating this special graph. Drag and drop it onto your worksheet and connect it to the **Read Text File** node. Use the following code to define this node.[4]

```
#------------------------
# visualize missing values
#------------------------

#-----------
# initialize
#-----------
 if (IM$test) {
   return(list(simple=T))
 }

#-------------------------------------------------
# Create a timeSeries() using the first timeDate()
# col as the positions and dropping other date cols
#-------------------------------------------------
 isDateCol <- sapply(IM$in1, function(x) {
   data.class(x) == "timeDate" })
 whichDateCols <- seq(1, ncol(IM$in1))[isDateCol]
 if (length(whichDateCols)==0) {
   warning(
     "No date column is available. Using default origin.")
```

[4] Please refer to the corresponding Insightful Miner and S-PLUS *User's Guide* and *Programmer's Guide* for more details.

```
    pos <- NULL
  }
  else {
    pos <- IM$in1[,whichDateCols[1]]
  }

#-------------------------------
# Create the timeSeries() object
#-------------------------------
pos<-substring(pos,1,10)
ts <- timeSeries(IM$in1[,!isDateCol, drop=F], position = pos)
count.missing <- function(X) {
  nr <- dim(X)[1]
  nc <- dim(X)[2]
  X <- as.numeric(!apply(X, 2, is.na)) *
                   matrix(rep(1:nr, nc), ncol=nc)
  X[X==0] <- NA
  index <- apply(X, 2, FUN=min, na.rm=T) - 1
  return(apply(is.na(X), 2, sum) - index)
}

#----------------------------------------
# Image plot of missing values in dataset
#----------------------------------------
DATES = as.character(dates(pos, out.format=" mon yy"))
xr = rep(-100, ncol(ts@data))
yr = 1:ncol(ts@data)
xx = seq(1, length(DATES), by=56)
daily.names=unlist(dimnames(IM$in1)[2])

image(as.matrix(ts@data), axes=F, xlab="Observations", cex=0.8)
text(xr, yr, daily.names[2:length(daily.names)], cex=0.7)
text(xx, rep(0.25, length(xx)), DATES[xx], srt=90, cex=0.6, adj=1)
title("Pattern of Missing Values in Weekly Return Data used for Model
Training\n(whitespace indicates missing value)", cex=0.7)

# Don't return anything
return(NULL)
```

We run this node and obtain the following graph in Figure 5 by right-clicking on the node and selecting **Viewer**.

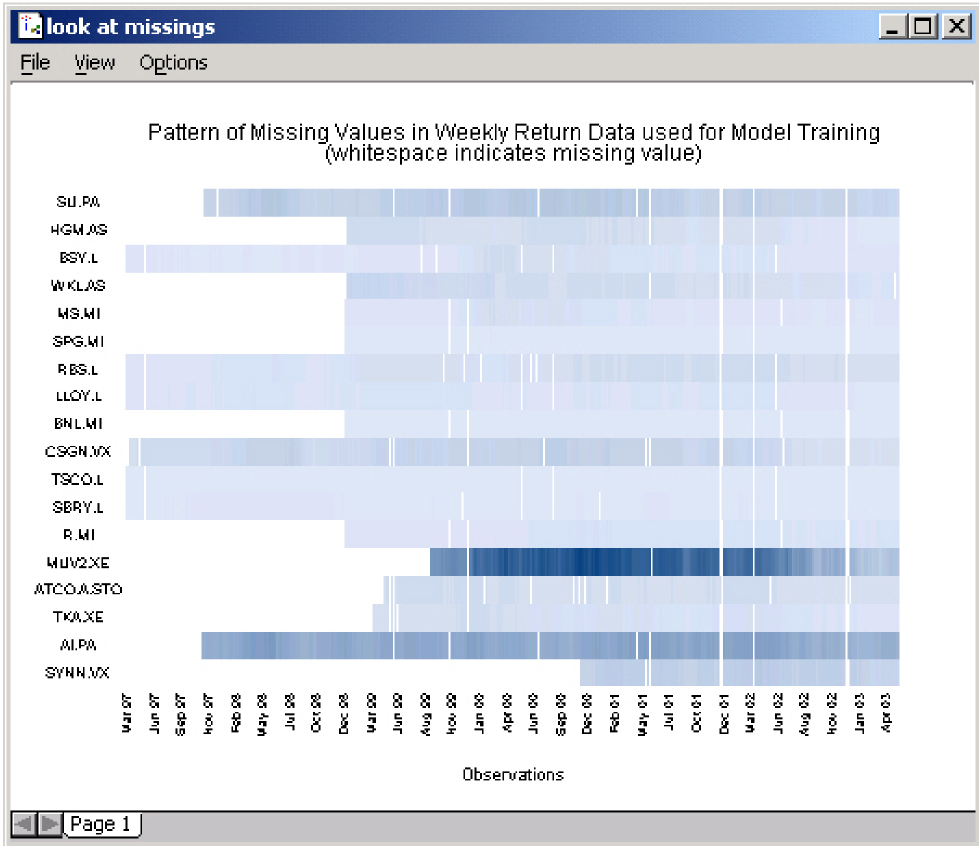


Figure 5

We see that large gaps of missing values dominate until end of 1998; only after roughly November 2000 are all values present. Therefore, we select the data to be used for the present analysis to start at November 11, 2000. We accomplish this by double-clicking a **Filter Rows** node (Filter Rows) and then connecting it to the **Read Text File** node. We define the selection criterion displayed in the **Qualifier** section of Figure 6.

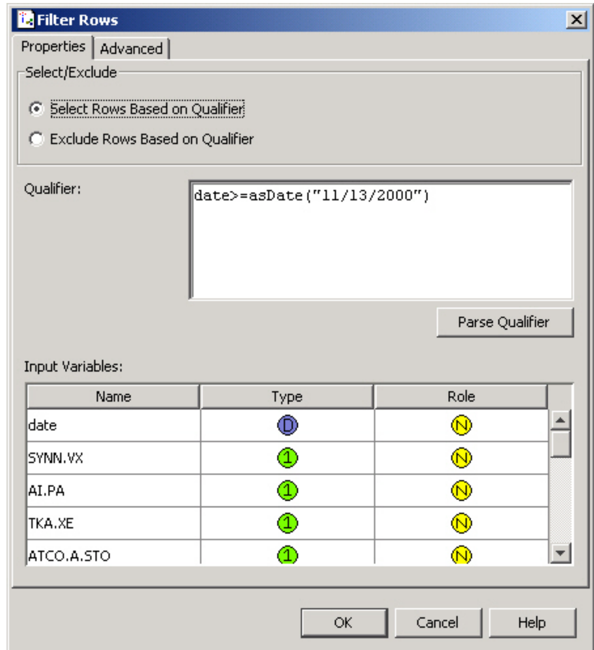


Figure 6

*The **Filter Rows** node allows to easily define selection criteria based on Insightful Miner's expression language.*

Finally, we split the data into two independent sets: One from Nov 13, 2000 until March 31, 2002, and the other from April 1, 2003 until May 27, 2003. Using a **Split** node (

), we define the corresponding splitting criterion.

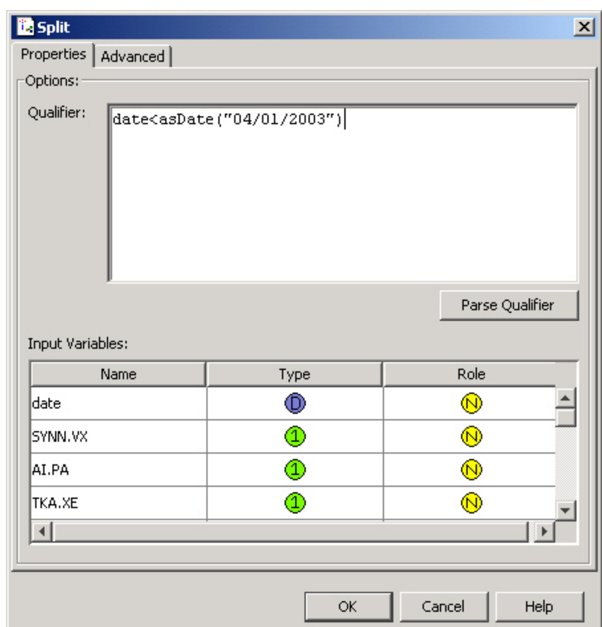


Figure 7

*The **Split** node divides the data in two as defined by the splitting criteria.*

We run this node and obtain all data before April 1, 2003 on the first output and all data including and after April 1, 2003 on the second output. Using the data viewer, we verify that there are 620 records in output 1 (in the **Data Viewer** menu, select **View | Rows**

matching) and 41 records in output 2 (in the **Data Viewer** menu, select **View | Non-matching rows**).

The worksheet so far looks like Figure 8:

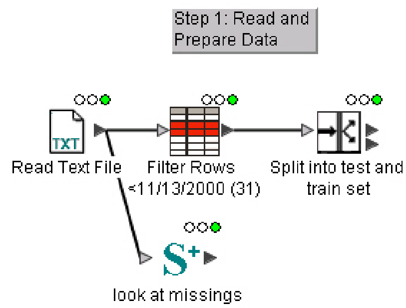


Figure 8

Section 2: Constructing and Exploring Stock Pairs Data

There are any number of ways to select those pairs we wish to trade. For example, methods such as factor analysis, GARCH and neural networks have all been employed with the aim of selecting pairs. See Alexander (2001) for general use of these techniques in finance.

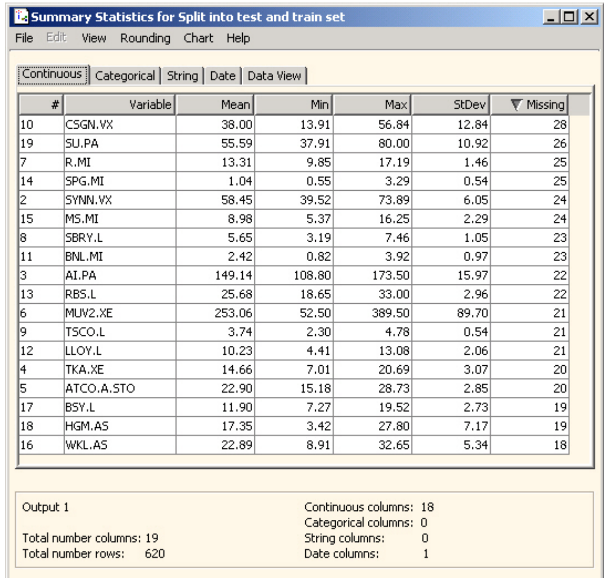
As stated in the introduction to this Application Note, we assume that a pair picking process has already taken place, the stocks are marked as long or short trade only, and we now now want to visualize their behavior over time. To this end we do the following steps:

- **Interpolate missing values.** As we had seen before, although most missing values are eliminated after November 2000, some remain and have to be handled.
- **Create pair price ratios.** We'll create new columns of data containing the ratio of the stock we hold long and the stock we hold short.
- **Plot pair price ratios.** This visualizes the evolution of the price ratio over time for each selected pair, along with information about their mean, one- and two-standard deviation limits.

Interpolating Missing Values

Interpolating data is easy. In fact, we'll use the same S-PLUS node that has already been described in another Insightful Application Note titled "Effective Portfolio Optimization." For completeness, we repeat the description.

Looking at the data viewer of the **Split** node, we note that each time series still has between 18 and 28 missing values, as shown in Figure 9 (after sorting in descending order by clicking once on the **Missing** column header).



#	Variable	Mean	Min	Max	StDev	Missing
10	CSGN.VX	38.00	13.91	56.84	12.84	28
19	SU.PA	55.59	37.91	80.00	10.92	26
7	R.MI	13.31	9.85	17.19	1.46	25
14	SPG.MI	1.04	0.55	3.29	0.54	25
2	SYNN.VX	58.45	39.52	73.89	6.05	24
15	MS.MI	8.98	5.37	16.25	2.29	24
8	SBRY.L	5.65	3.19	7.46	1.05	23
11	BNL.MI	2.42	0.82	3.92	0.97	23
3	AI.PA	149.14	108.80	173.50	15.97	22
13	RBS.L	25.68	18.65	33.00	2.96	22
6	MUV2.XE	253.06	52.50	389.50	89.70	21
9	TSCO.L	3.74	2.30	4.78	0.54	21
12	LLOY.L	10.23	4.41	13.08	2.06	21
4	TKA.XE	14.66	7.01	20.69	3.07	20
5	ATCO.A.STO	22.90	15.18	28.73	2.85	20
17	BSY.L	11.90	7.27	19.52	2.73	19
18	HGM.AS	17.35	3.42	27.80	7.17	19
16	WKL.AS	22.89	8.91	32.65	5.34	18

Figure 9

Interpolation of missing values needs to be done prior to plotting the results and is readily available with S+FinMetrics.

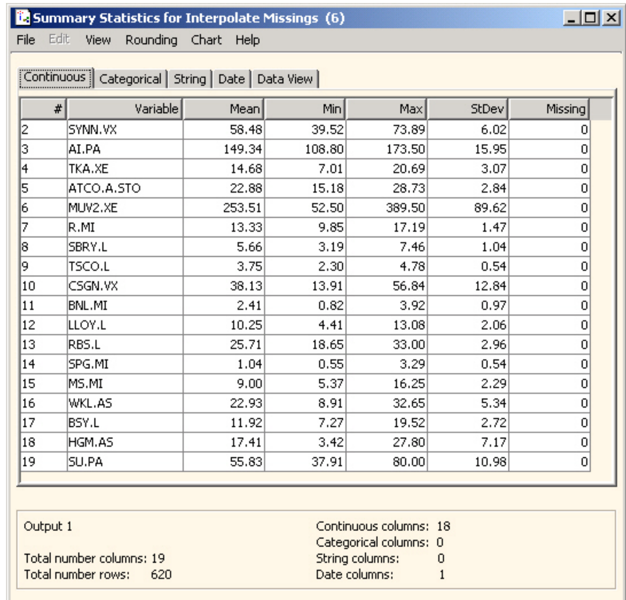
Simple interpolation schemes, as provided by the `interpNA` function in the S+FinMetrics module, are usually enough to effectively handle missing values. We therefore create another **S-PLUS Script** node and call `interpNA` (available by attaching the S+FinMetrics module), designed to interpolate missing values in time series. The content of the second **S-PLUS Script** node is shown below:

```
#--------------------------
# Interpolate missing values
#--------------------------
if (IM$test) { module(finmetrics) }

# interpNA() doesn't handle timeDate cols
# only do interp for the other cols
isDateCol <- sapply(IM$in1, function(x) { data.class(x) == "timeDate" })
for (i in seq(along=isDateCol)) {
  if (!isDateCol[i])
     IM$in1[,i] <- interpNA(IM$in1[,i], method="spline", maxStartNA=5)
  }

# return the full data.frame
IM$in1
```

For each time series column with missing values, replacement values are interpolated using a spline method. After running this node, the data has no more missing values (Figure 10).



Summary Statistics for Interpolate Missings (6)

File Edit View Rounding Chart Help

Continuous | Categorical | String | Date | Data View

#	Variable	Mean	Min	Max	StDev	Missing
2	SYNN.VX	58.48	39.52	73.89	6.02	0
3	AI.PA	149.34	108.80	173.50	15.95	0
4	TKA.XE	14.68	7.01	20.69	3.07	0
5	ATCO.A.STO	22.88	15.18	28.73	2.84	0
6	MUV2.XE	253.51	52.50	389.50	89.62	0
7	R.MI	13.33	9.85	17.19	1.47	0
8	SBRY.L	5.66	3.19	7.46	1.04	0
9	TSCO.L	3.75	2.30	4.78	0.54	0
10	CSGN.VX	38.13	13.91	56.84	12.84	0
11	BNL.MI	2.41	0.82	3.92	0.97	0
12	LLOY.L	10.25	4.41	13.08	2.06	0
13	RBS.L	25.71	18.65	33.00	2.96	0
14	SPG.MI	1.04	0.55	3.29	0.54	0
15	MS.MI	9.00	5.37	16.25	2.29	0
16	WKL.AS	22.93	8.91	32.65	5.34	0
17	BSY.L	11.92	7.27	19.52	2.72	0
18	HGM.AS	17.41	3.42	27.80	7.17	0
19	SU.PA	55.83	37.91	80.00	10.98	0

Output 1

Total number columns: 19
Total number rows: 620

Continuous columns: 18
Categorical columns: 0
String columns: 0
Date columns: 1

Figure 10

Creating Pair Price Ratios

The pairs we're going to base the price ratio calculations on are:

- SYNN.VX (long), AI.PA (short)
- TKA.XE (long), ATCO.A.STO (short)
- MUV2.XE (long), R.MI (short)
- SBRY.L (long), TSCO.L (short)
- CSGN.VX (long), BNL.MI (short)
- LLOY.L (long), RBS.L (short)
- SPG.MI (long), MS.MI (short)
- WKL.AS (long), BSY.L (short)
- HGM.AS (long), SU.PA (short)

Next, we compute, for each pair, the ratio of the long and the short component at each point in time of the train set. To do so, we use a **Create Column** node (Create Columns) and define the new column for the ratios shown in the top part of the screen (Figure 11)

*The **Create Columns** node is used to define the calculation of the pair price ratios.*

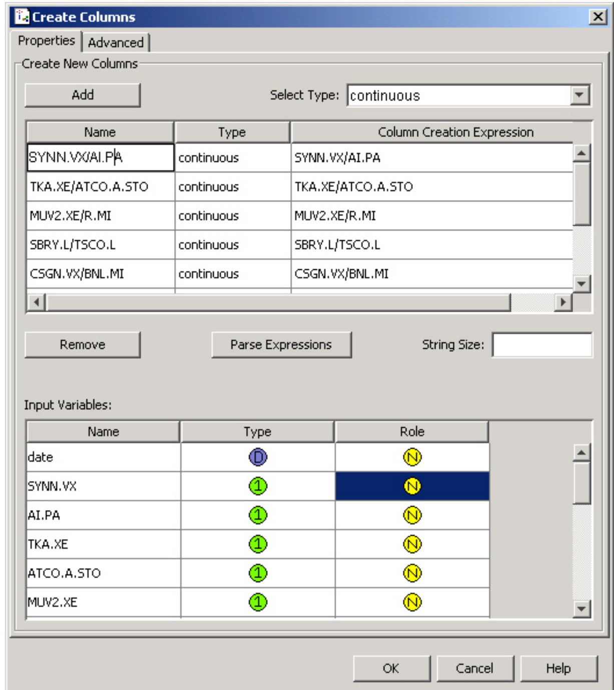
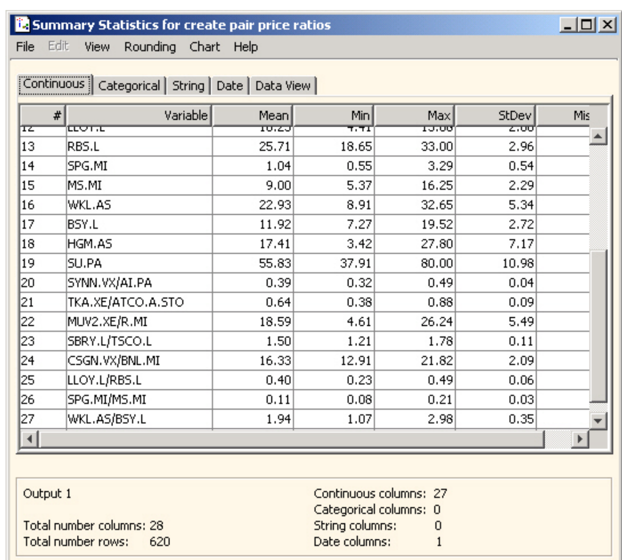


Figure 11

We know from the summary statistics in the **Interpolate Missings** node above that no stock ever took on zero as a value (the minimum value taken on was with **BNL.MI** at 0.82). Therefore, we have to take no further caution in handling divide-by-zero exceptions. We run the **Create Column** node, examine the output using the summary statistics, and obtain Figure 12:



#	Variable	Mean	Min	Max	StDev	Mis
13	RBS.L	25.71	18.65	33.00	2.96	
14	SPG.MI	1.04	0.55	3.29	0.54	
15	MS.MI	9.00	5.37	16.25	2.29	
16	WKL.AS	22.93	8.91	32.65	5.34	
17	BSY.L	11.92	7.27	19.52	2.72	
18	HGM.AS	17.41	3.42	27.80	7.17	
19	SU.PA	55.83	37.91	80.00	10.98	
20	SYNN.VX/AI.PA	0.39	0.32	0.49	0.04	
21	TKA.XE/ATCO.A.STO	0.64	0.38	0.88	0.09	
22	MUV2.XE/R.MI	18.59	4.61	26.24	5.49	
23	SBRY.L/TSCO.L	1.50	1.21	1.78	0.11	
24	CSGN.VX/BNL.MI	16.33	12.91	21.82	2.09	
25	LLOY.L/RBS.L	0.40	0.23	0.49	0.06	
26	SPG.MI/MS.MI	0.11	0.08	0.21	0.03	
27	WKL.AS/BSY.L	1.94	1.07	2.98	0.35	

Figure 12

We have now created nine additional columns, corresponding to the pair price ratios. To proceed, we want to exclude the other 18 columns referring to single stock price and not to pairs. To do this, we use a **Filter Columns** node (Filter Columns) and check all columns we want to keep, as in Figure 13

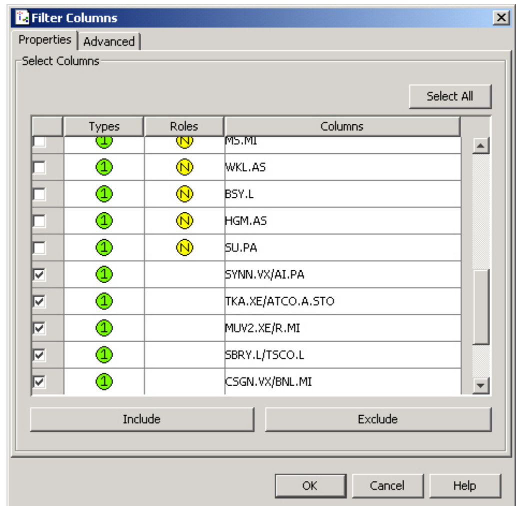
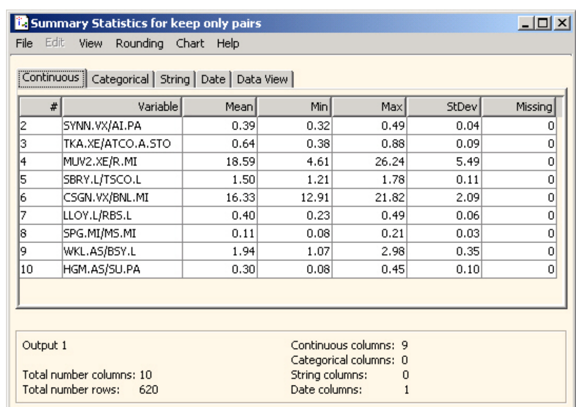


Figure 13

Finally, we run the node and obtain a data set with only nine stock columns and one date column, shown in Figure 14



#	Variable	Mean	Min	Max	StDev	Missing
2	SYNN.VX/AI.PA	0.39	0.32	0.49	0.04	0
3	TKA.XE/ATCO.A.STO	0.64	0.38	0.88	0.09	0
4	MUV2.XE/R.MI	18.59	4.61	26.24	5.49	0
5	SBRY.L/TSCO.L	1.50	1.21	1.78	0.11	0
6	CSGN.VX/BNL.MI	16.33	12.91	21.82	2.09	0
7	LLOY.L/RBS.L	0.40	0.23	0.49	0.06	0
8	SPG.MI/MS.MI	0.11	0.08	0.21	0.03	0
9	WKL.AS/BSY.L	1.94	1.07	2.98	0.35	0
10	HGM.AS/SU.PA	0.30	0.08	0.45	0.10	0

Output 1
Total number columns: 10
Total number rows: 620
Continuous columns: 9
Categorical columns: 0
String columns: 0
Date columns: 1

Figure 14

Plotting Pairs

We would now like to get a better view of the pairs' behavior over time along with some statistical properties about their mean values and standard deviations. To this end, we create two S-PLUS nodes, one plotting all nine pairs on one page, and the other plotting one pair per page to look at the details.

The following S-PLUS script does the plotting of nine graphs on one page:

```
#----------------------------------------------------------
# This script creates multiple time series plots one per page
#----------------------------------------------------------
 if (IM$test) {
   return(list(simple=T))
 }
#------------------------------------------------
# Create a timeSeries() using the first timeDate()
# col as the positions and dropping other date cols
#------------------------------------------------
isDateCol <- sapply(IM$in1, function(x) { data.class(x) == "timeDate" })
whichDateCols <- seq(1, ncol(IM$in1))[isDateCol]
if (length(whichDateCols)==0) {
   warning("No date column is available. Using default origin.")
   pos <- NULL
  }
else {
  pos <- IM$in1[,whichDateCols[1]]
  }
ts <- timeSeries(IM$in1[,!isDateCol, drop=F], position=pos)
tsx <- IM$in1[,!isDateCol, drop=F]

#------------
# plot series
#------------
par(mfrow=c(3,3))
for (x in colIds(ts)) {
  x.mean = mean(tsx[,x])
  x.sd = stdev(tsx[,x])
  xL1 = x.sd
  xL2 = 2 * x.sd
  yu = x.mean + 3 * x.sd
  yl = x.mean - 3 * x.sd
  plot(ts[,x], reference.grid=F, bty="n", ylim = c(yl, yu), cex = .05)

  xc <- 1
  lines(c(0, 300), c(x.mean + xL2, x.mean + xL2), col=1, lwd=1)
  text(xc, 1.01*(x.mean + xL2), "+2sd", col=1)
  lines(c(0, 300), c(x.mean + xL1, x.mean + xL1), col=4, lty=2, lwd=2)
  text(xc, 1.01*(x.mean + xL1), "+1sd", col=4)
  lines(c(0, 300), c(x.mean, x.mean), col=8, lwd=2)
  text(xc, 1.1*x.mean, "Mean", col=8)
  lines(c(0, 300), c(x.mean - xL1, x.mean - xL1), col=4, lty=2, lwd=2)
  text(xc, 1.1*(x.mean - xL1), "-1sd", col=4)
  lines(c(0, 300), c(x.mean - xL2, x.mean - xL2), col=1, lwd=1)
  text(xc, 1.1*(x.mean - xL2), "-2sd", col=1)
  title(x, cex=0.6)
 }
text(.1, -.34, "Source: Infotec (original price series)", cex=0.6)

# Don't return anything
return(NULL)
```

It also displays the mean and +/- one and two standard deviations for each time series. The result obtained is shown in the Figure 15.

The power of S-PLUS graphics can be fully leveraged within Insightful Miner.

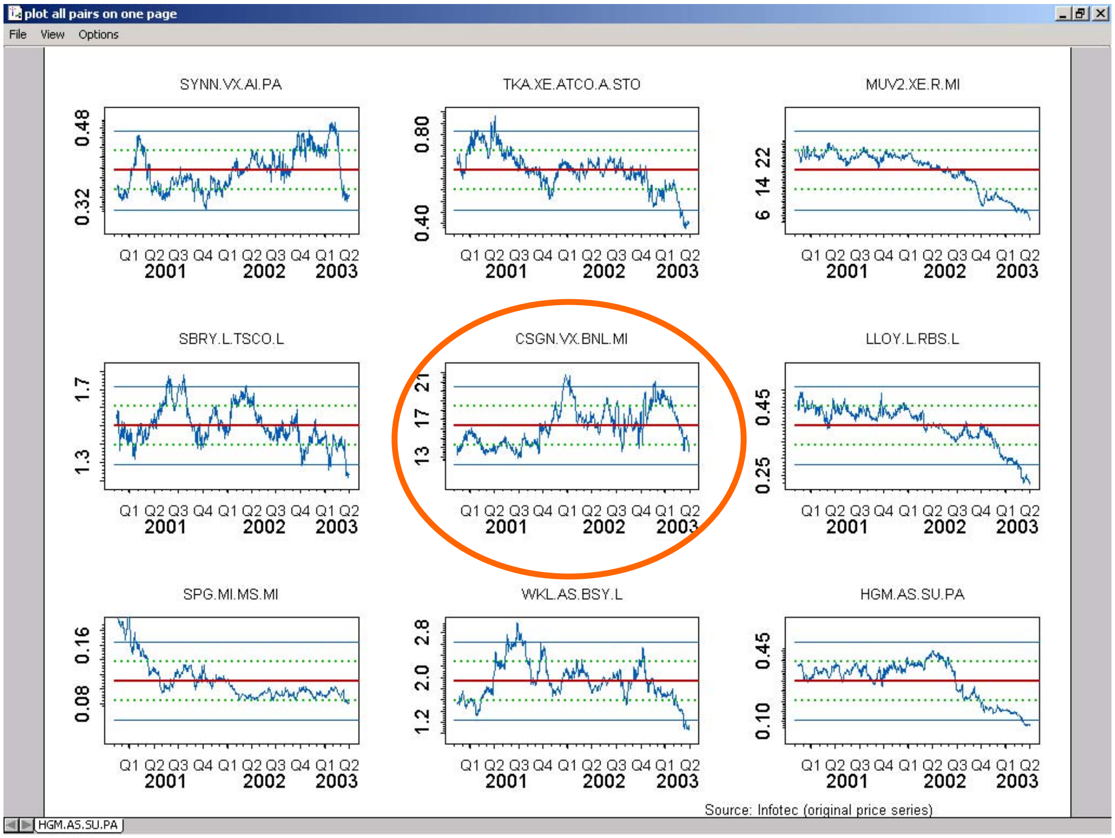


Figure 15

We observe that these pairs are approaching their negative two-standard deviation level, and so we expect them to quickly revert to their mean Therefore, we should trade each of these pairs.

We choose the highlighted pair in Figure 15 consisting of Credit Suisse and Banca Nazionale del Lavoro to look at more details. We create another S-PLUS node and use the same S-PLUS script as above with only a minor change: Instead of using the S-PLUS command `par(mfrow=c(3,3))` defining the output page consists of 3 by 3 plots, we now use the command `par(mfrow=c(1,1))` to define that now we only want one plot per output page.

We run the node and then view the results. Selecting the **CSGN.VX.BNL.MI** tab on the graphical output window, we obtain Figure 16, below.

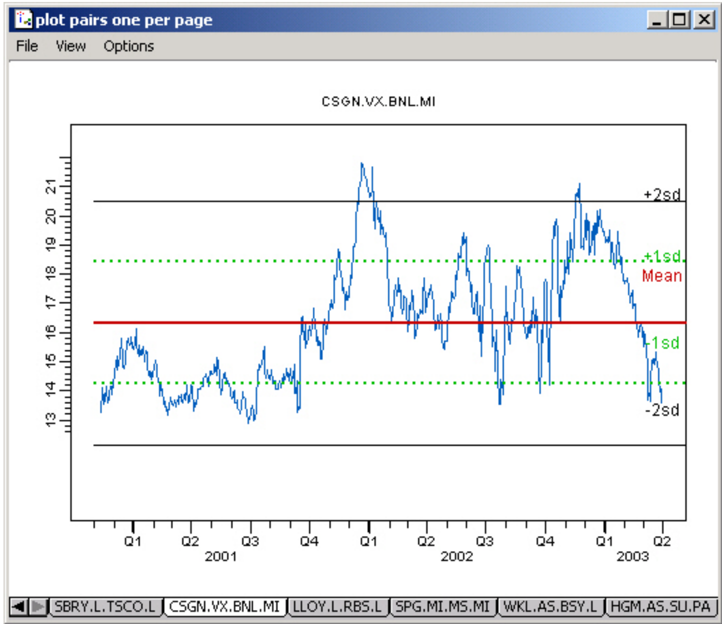


Figure 16

The red line shows the long term mean over the displayed time period, the upper and lower dotted green lines show the +/- one standard deviation levels, and the solid top and bottom black lines show the +/- two standard deviations levels.

Our view is that once the graph has breached the two standard deviation level this is a clear signal to enter into our pair trade. We simply assume that the ratio of the two stock prices will now head back towards its long run average.

The current worksheet for this project should look like Figure 17:

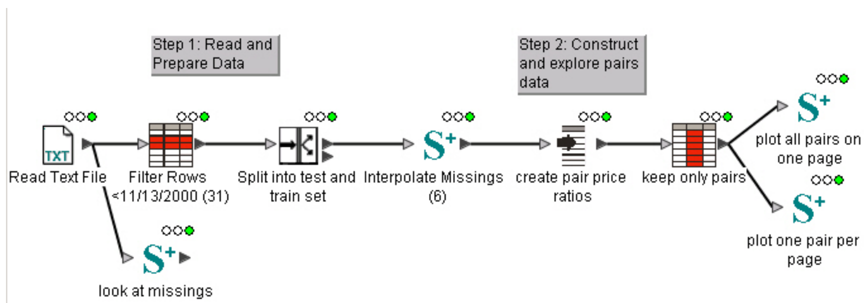


Figure 17

Section 3: Constructing an Optimal Long-Short Portfolio

Pair trading used together with leverage implies an unbounded level of return and risk. A typical long-short hedged fund would construct a portfolio with risk and return characteristics tailored to specific investors. To construct such a risk-controlled portfolio, we complete the following steps:

- Select weekly data for the subsequent analysis.
- Use expectation maximization algorithm to handle missing values.

- Optimize the long-short portfolio.

Selecting Weekly Data

*Aggregating a time series to contain only Friday's closing prices is accomplished by defining a simple selection rule in the **Filter Rows** node.*

Up to this point, we've had daily price data for each stock. Since we would like to optimize our portfolio using weekly returns data to alleviate serial correlation, we now build new time series, which only contain price data for each Friday. This can easily be accomplished by using a **Filter Rows** node and defining the corresponding selection rule, as shown in Figure 18:

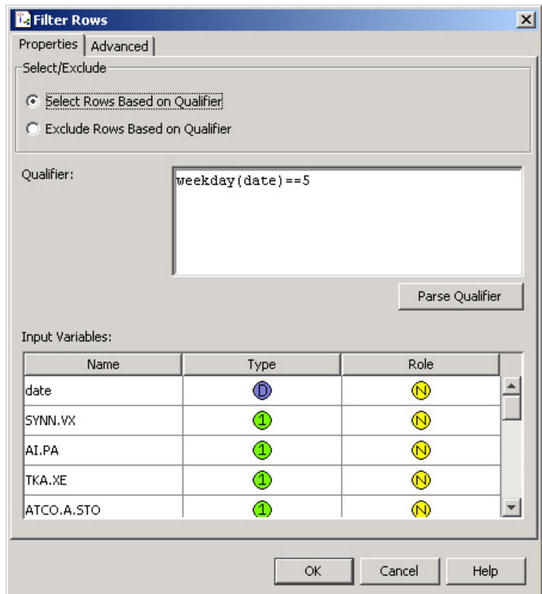
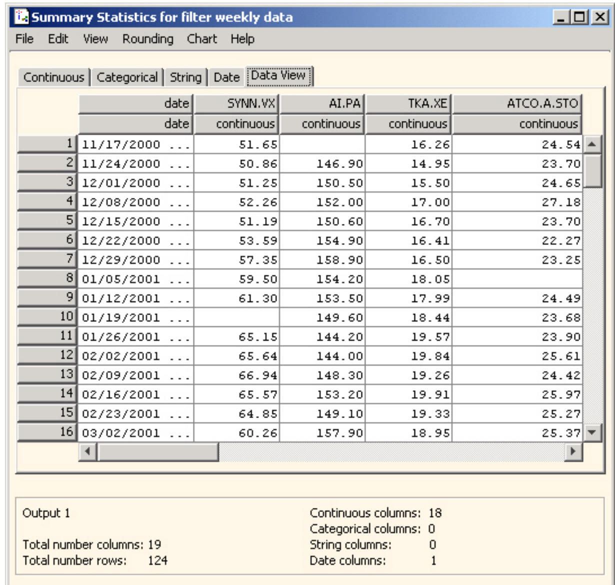


Figure 18

The result of running this node can be explored with the data viewer, as shown in Figure 19:



	date	SYNN.VX	AI.PA	TKA.XE	ATCO.A.STO
	date	continuous	continuous	continuous	continuous
1	11/17/2000 ...	51.65		16.26	24.54
2	11/24/2000 ...	50.86	146.90	14.95	23.70
3	12/01/2000 ...	51.25	150.50	15.50	24.65
4	12/08/2000 ...	52.26	152.00	17.00	27.18
5	12/15/2000 ...	51.19	150.60	16.70	23.70
6	12/22/2000 ...	53.59	154.90	16.41	22.27
7	12/29/2000 ...	57.35	158.90	16.50	23.25
8	01/05/2001 ...	59.50	154.20	18.05	
9	01/12/2001 ...	61.30	153.50	17.99	24.49
10	01/19/2001 ...		149.60	18.44	23.68
11	01/26/2001 ...	65.15	144.20	19.57	23.90
12	02/02/2001 ...	65.64	144.00	19.84	25.61
13	02/09/2001 ...	66.94	148.30	19.26	24.42
14	02/16/2001 ...	65.57	153.20	19.91	25.97
15	02/23/2001 ...	64.85	149.10	19.33	25.27
16	03/02/2001 ...	60.26	157.90	18.95	25.37

Figure 19

We have now reduced the number of rows from 620 to 124, and the dates displayed correspond exactly to Friday of each week.

Handling Missing Values

In this section, we use a different approach to missing value replacement than the one used in Step 2 above. Instead of using the S+FinMetrics function `interpNA`, we use a multivariate approach that allows a more accurate imputation of missing values under the assumption of multivariate normality. We use the EM (expectation maximization) algorithm available in the S-PLUS **missing** library, which is Bayesian in style and provides expectation and covariance as parameters as well as non-negative definite covariance by construction. In particular, we choose the `emGauss` function, which computes estimates of the parameters in a multivariate normal model. `emGauss` handles missing values by using the EM algorithm to compute the modes of the posterior probability distribution of the estimates given the specified normal inverse Wishart prior (when a non-informative prior is used, maximum likelihood estimates are computed). Given the results from the EM algorithm, missing values can be imputed with the `impGauss` function (not shown in this example) as well as obtaining means and covariances of the underlying distributions. Since we are only interested in the optimization of our portfolio, we need the means and covariance matrix, and are not as concerned with the missing value imputation itself. Therefore, the only functions we need are `preGauss` (see the **missing** library manual for more details) and `emGauss`.

The EM algorithm will be applied to the time series of returns, which is obtained by using the S+FinMetrics function `getReturns`.

```
# EM algorithm
 if (IM$test) {
   if (!is.element("missing", search())) {
     library(missing, first=T)
     return(list(simple=T))
     }
 }
library(missing)
```

Through the integration of Insightful Miner with numerous S-PLUS libraries (such as the missing library), you can access cutting-edge data analysis methods. .

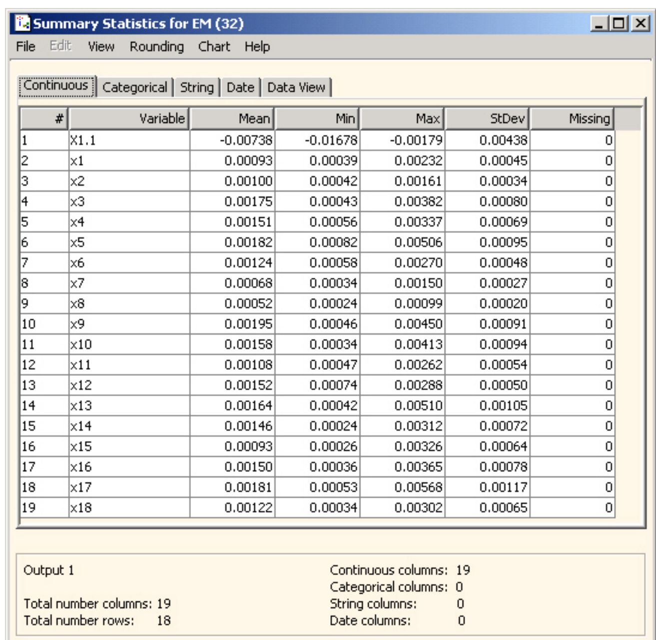
```
# Create a timeSeries() using the first timeDate()
# col as the positions and dropping other date cols
 isDateCol <- sapply(IM$in1, function(x) {
   data.class(x) == "timeDate" })
 whichDateCols <- seq(1, ncol(IM$in1))[isDateCol]
 if (length(whichDateCols)==0) {
   warning(
     "No date column is available. Using default origin.")
   pos <- NULL
 }
 else {
   pos <- IM$in1[,whichDateCols[1]]
 }

# Create the timeSeries() object
 pos<-substring(pos,1,10)
 ret <- getReturns(timeSeries(IM$in1[,!isDateCol, drop=F], position = pos))

# Use EM algorithm expectation maximization ("missing values")
ret.mat = preGauss(matrix(ret, ncol=ncol(ret)))
ret.emGauss = emGauss(ret.mat)
ret.VCOV2 = paramIter(ret.emGauss, expand=T)$sigma
ret.ERS = paramIter(ret.emGauss, expand=T)$mu

out <- data.frame(cbind(as.vector(ret.ERS),Mean=ret.VCOV2))
return(list(out1=out))
```

We run the node and look at the summary statistics of the output, as shown in Figure 20:


Summary Statistics for EM (32)

File Edit View Rounding Chart Help

Continuous | Categorical | String | Date | Data View

#	Variable	Mean	Min	Max	StDev	Missing
1	X1.1	-0.00738	-0.01678	-0.00179	0.00438	0
2	x1	0.00093	0.00039	0.00232	0.00045	0
3	x2	0.00100	0.00042	0.00161	0.00034	0
4	x3	0.00175	0.00043	0.00382	0.00080	0
5	x4	0.00151	0.00056	0.00337	0.00069	0
6	x5	0.00182	0.00082	0.00506	0.00095	0
7	x6	0.00124	0.00058	0.00270	0.00048	0
8	x7	0.00068	0.00034	0.00150	0.00027	0
9	x8	0.00052	0.00024	0.00099	0.00020	0
10	x9	0.00195	0.00046	0.00450	0.00091	0
11	x10	0.00158	0.00034	0.00413	0.00094	0
12	x11	0.00108	0.00047	0.00262	0.00054	0
13	x12	0.00152	0.00074	0.00288	0.00050	0
14	x13	0.00164	0.00042	0.00510	0.00105	0
15	x14	0.00146	0.00024	0.00312	0.00072	0
16	x15	0.00093	0.00026	0.00326	0.00064	0
17	x16	0.00150	0.00036	0.00365	0.00078	0
18	x17	0.00181	0.00053	0.00568	0.00117	0
19	x18	0.00122	0.00034	0.00302	0.00065	0

Output 1

Total number columns: 19
Total number rows: 18

Continuous columns: 19
Categorical columns: 0
String columns: 0
Date columns: 0

Figure 20

After changing the rounding to five decimal places, (select **Rounding |5 Decimal Places**), we recognize that the first variable `X1.1` contains the mean values and the following 18 variables (`x1`…`x18`) contain the covariance matrix. This is the input format required to optimize the portfolio, based on the current mean-covariance matrix.

Optimizing Long-Short Portfolios

The final aim of this exercise is to maximize the expected return of our zero-net-cash position portfolio while at the same time controlling for the variability of the returns. To the fund manager, variability is synonymous with risk. The aim is to satisfy our risk-reward tradeoff at some point along the efficient frontier with longs and shorts offsetting each other. In this example, we don't want to move too far away from our equal-weighted solution. For this reason we constrain our individual stocks weights so that they don't go too near zero or become too large and dominate our portfolio.

We use the quadratic programming technique to optimize our portfolio in the following way:

We want to minimize the objective function $-\lambda\mu^T w + w^T S w$, where

> λ = A parameter controlling the trade-off between risk and return.
> μ = The (18x1) vector of expected returns (means).
> w = The (18x1) vector of stock weights.
> S = The (18x18) estimated stock covariance matrix.

This optimization will be subject to the constraint that the sum of all weights associated with the long stocks must be equal to one ($\Sigma w_i^+ = 1$) and the sum of the weights associated with the short stocks must be equal to minus one ($\Sigma w_i^- = -1$). Furthermore, we restrain the weights between lower and upper boundaries ($a \le |w_i| \le b$ for all i).

The function `solveQP` in the optimization module S+NuOPT solves this type of problem in a very efficient way. The following S-PLUS code defines the objective function and the constraints, and executes the optimization for various values of λ. In this way, we obtain for each value of λ an optimal portfolio along the efficient frontier. Additionally, we create some illustrative graphs such as the weights for each stock in the portfolio along the efficient frontier and the efficient frontier itself.

To setup the optimization, we define the starting values for w to be equal. We expect a quarter (factor 0.25) of the observed weekly return (averaged over one year) as a reversal (factor –1). We take the negative of the observed negative returns and therefore expect to lose on the shorts; they hedge our bets at present. λ will run over the interval (0, 3) to obtain efficient frontier. The weight vector will be constrained (|w|) between a = 0.05 and b = 0.15 and finally, we use the EM-estimated return and covariance matrix obtained in the previous step:

```
if (IM$test) {
   if (!is.element("nuopt", search())) { module(nuopt) }
   return(list(in1.requirements="one.block",
     out1=data.frame(weights=0, returns=0, asset=0, type='')))
}
module(nuopt)

#-----------------------------
# initialize various variables
#-----------------------------
first=T
RISK = NULL
ERS = NULL
LAMBDA = c(0, 0.1, 3)
WEIGHT = c(1.0, 0.5, 0.2, .15)
max.weight = WEIGHT[4]
min.weight = 0.05
w.seq=matrix(0,18,length(LAMBDA))
VCOV = as.matrix(IM$in1[,-1, drop=F])
ret.ERS = IM$in1[,1]
```

```
pos = rep(c(1, -1), 9)
mu = -ret.ERS * pos * .25  # trade reversal
track = T
VCOV.names <- dimnames(VCOV)[[1]]
pos.names <- dimnames(VCOV)[[2]][pos!=0]
indx <- 1:length(pos)
indx <- indx[pos!=0]
mu = as.numeric(mu[pos!=0])
test.val <- all(eigen(VCOV)$values > 0.0)
nonzero <- pos[indx]
nonzero <- rep(1, length(nonzero)) * sign(nonzero)

#--------------------------------
# set constraints for optimization
#--------------------------------
bLO <- nonzero * max.weight
bLO[bLO > 0] <-  min.weight
bUP <- nonzero * max.weight
bUP[bUP < 0] <- -min.weight
a1 <- nonzero
a1[nonzero < 0] <- 1
a1[nonzero > 0] <- 0
a2 <- nonzero
a2[nonzero > 0] <- 1
a2[nonzero < 0] <- 0
A <- rbind(a1, a2)
cLO <- c(-1, 1)
cUP <- c(-1, 1)
x0 <- rep(1.0/length(mu), length(mu))

#-----------------------------------------------------------------
# loop over lambda to construct weights along the efficient frontier
#-----------------------------------------------------------------
if (test.val) {
  for (i in 1:length(LAMBDA)) {
    lambda = LAMBDA[i]
    QP.soln <- solveQP(objQ = VCOV, objL = -mu * lambda, A = A,
                       cLO = cLO, cUP = cUP, bLO = bLO, bUP = bUP,
                      x0 = x0, type = minimize, trace = track)

    # Solution is set of stock weights, update weights
    w.new <- QP.soln$variables$x$current
    w <- NULL
    w[indx] <- w.new
    w.seq[,i] <- w.new

    #------------------------------------
    # plot weights as a function of lambda
    #------------------------------------
    if (track) {
      if(first){
        par(mfrow=c(2,2))
        first=F
      }
      barplot(w.seq[,i], type="h", xlab="", ylab="weight", col=6)
      xr = 0:(ncol(VCOV)-1)
      xr = xr + 0.24 * 1:ncol(VCOV)
      yr = rep(-.175, ncol(VCOV))
      text(xr, yr, dimnames(VCOV)[[2]], cex=1, srt=90, adj=1)
      title(paste("lambda =", lambda), font=8)
    }
  tracking.error    <- t(w) %*% VCOV %*% w
  marg.contrib.risk <- VCOV %*% w
  RISK = c(RISK, sqrt(tracking.error))
  ERS = c(ERS, mu %*% w)
  }
}
#-----------------------
# plot efficient frontier
#-----------------------
par(mfrow=c(1,1))
```

```
tdays = 52
ERS.ann = 100 * ((1 + (ERS))^tdays - 1)
RISK.ann = 100 * RISK * sqrt(tdays)
plot(RISK.ann, ERS.ann, xlim=c(13,18), ylim=c(19,25), xlab="", pch = " ", cex=1.2)
lines(spline(RISK.ann, ERS.ann), lwd=3, col=2)
title("Efficient Frontier", ylab="", cex=1.2)
text(12.4, 22, "Annualised Return%", srt=90, cex=1.5)
text(15.5, 18, "Annualised Volatility%", cex=1.5)

#----------------------------
# plot expected weekly returns
#----------------------------
par(mfrow=c(1,2))
barplot(-ret.ERS * .25)
title("Adjusted Expected Weekly Return")
xr = 0:(ncol(VCOV)-1)
xr = xr + 0.24 * 1:ncol(VCOV)
yr = rep(-0.0003, ncol(VCOV))
text(xr, yr, dimnames(VCOV)[[2]], cex=0.8, srt=90, adj=1)

barplot(mu)
title("Adjusted Expected Weekly Return \nx Sign(Position)")
xr = 0:(ncol(VCOV)-1)
xr = xr + 0.24 * 1:ncol(VCOV)
yr = rep(-.003, ncol(VCOV))
text(xr, yr, dimnames(VCOV)[[2]], cex=0.8, srt=90, adj=1)

# make weight matrix available to other I-Miner
# S-PLUS nodes (e.g. for performance measurement)
wperm<<-w.seq
```

At the end of the script, we write the weights obtained for the various iterations into a permanent object in an S-PLUS database, using the assignment operator <<-). We use this to illustrate another interesting technique for passing on data and other S-PLUS object types (such as functions and models) to other Insightful Miner nodes. We discuss this in the next step about assessing the portfolio performance.

We have now concluded the portfolio optimization part, and we summarize our progress by showing the current worksheet up to this point (Figure 21):

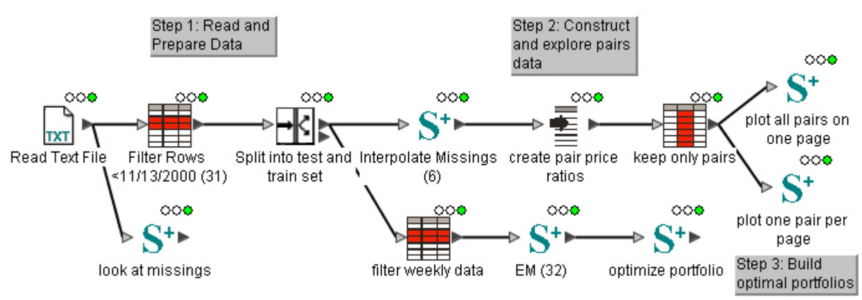


Figure 21

Section 4: Assessing Portfolio Performance

Next, we look at the efficient frontier, which we obtain by selecting the **Viewer** for the portfolio optimization node (see previous section), as shown in Figure 22.

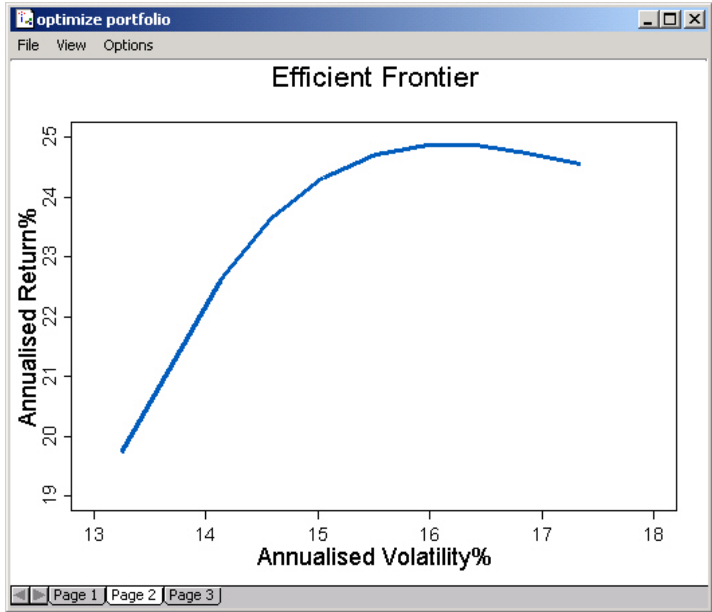


Figure 22

and observe that the performance is rather good! We notice that the double-digit volatility observed is high and would therefore probably prefer to select a portfolio with minimum risk, which would correspond to a value of λ=0 (corresponds to the lower left part of the efficient frontier).

Let’s have a look at the weights of the portfolio at λ=0, shown in Figure 23:

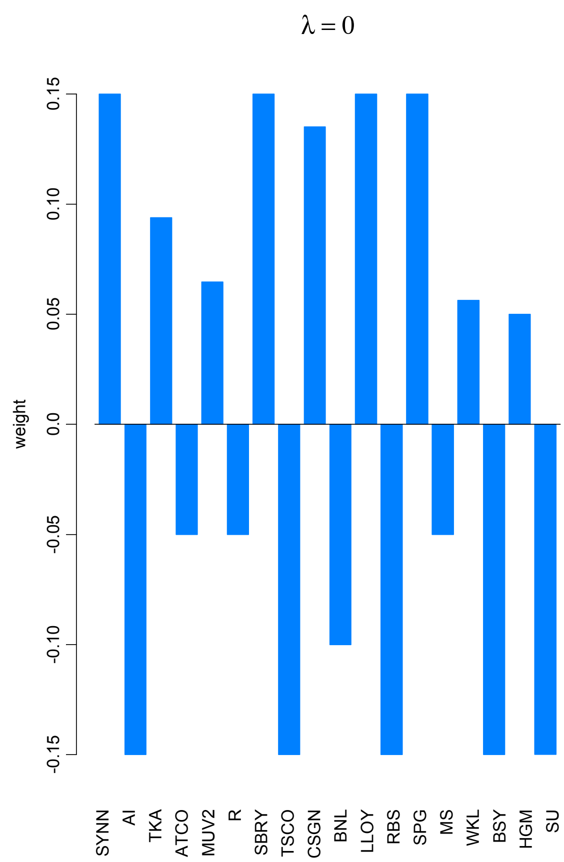


Figure 23

We’ll analyze the performance of this portfolio in two ways: (1) back-testing the historical performance on the train set, and (2) forward-testing the portfolio out-of-sample and observe how it would have performed in the later data. Additionally, we compare the performance of the optimized portfolio to an equal-weighted portfolio.

Assessing Historical Portfolio Performance

We now want to understand how both portfolios (optimized and non-optimized) would have performed in the past. We use the original train set with daily price data, available at the upper output of the previously discussed **Split** node. The comparison will be done with an S-PLUS script. We use a very powerful function for time series aggregation called `aggregateSeries`; with only one function call, we compute the time series of portfolio returns. For a better comparison, we compute the time series of net asset value (NAV), index it to 100 at the beginning of the time series (November 13, 2000), and plot its development over time.

```
if (IM$test) {
   if (!is.element("finmetrics", search())) { module(finmetrics) }
   return(list(in1.requirements="one.block",
     out1=data.frame(weights=0, returns=0, asset=0, type='')))
}
module("finmetrics")
```

```
# Create a timeSeries() using the first timeDate()
# col as the positions and dropping other date cols
 isDateCol <- sapply(IM$in1, function(x) {
   data.class(x) == "timeDate" })
 whichDateCols <- seq(1, ncol(IM$in1))[isDateCol]
 if (length(whichDateCols)==0) {
   warning(
     "No date column is available. Using default origin.")
   pos <- NULL
 }
 else {
   pos <- IM$in1[,whichDateCols[1]]
 }

# Create the timeSeries() object
pos<-substring(pos,1,10)
ts.train.new=timeSeries(IM$in1[,!isDateCol, drop=F], position = pos)

#interpolate price time series for the case there are still some NAs
ts.train.new=interpNA(ts.train.new)

#-------------------------------------
# Assess the equal-weighted portfolio
#-------------------------------------
# Set the portfolio weights
longs <- c(1,3,5,7,9,11,13,15,17)
shorts <- c(2,4,6,8,10,12,14,16,18)
long.weights <- 1 / ts.train.new@data[dim(ts.train.new@data)[1],longs]
long.weights <- long.weights / sum(long.weights)
short.weights <- 1 / ts.train.new@data[dim(ts.train.new@data)[1],shorts]
short.weights <- short.weights / sum(short.weights)
weights <- rep(0,18)
weights[longs] <- long.weights
weights[shorts] <- -1 * short.weights
nweights <<- weights

# Function to calculate the L/S returns on a given day
# ( x[1,] is yesterday's prices, x[2,] is today's)
long.short.returns <- function(x)
{
  assign("weights", nweights, frame=1)
  # Calculate the unweighted returns
  rets <- x[2,] / x[1,] - 1
  # Calculate the portfolio return
  t(rets) %*% weights
}
ls.rets <- aggregateSeries(ts.train.new, FUN=long.short.returns, moving=2,
together=T)

# Calculate the "NAV" over time
index <- cumprod(c(100,1+ls.rets@data))
index <- timeSeries(data=index, pos=ts.train.new@positions)

#-----------------------------------------------------------------
# Assess optimal portfolio with weights corresponding to LAMBDA=0
#-----------------------------------------------------------------
nweights <<- wperm[,1]
ls.rets.opt <- aggregateSeries(ts.train.new, FUN=long.short.returns, moving=2,
together=T)

# Calculate the "NAV" over time
index.opt <- cumprod(c(100, 1+ls.rets.opt@data))
index.opt <- timeSeries(data=index.opt, pos=ts.train.new@positions)

#------------------------------
# plot both performance curves
#------------------------------
par(mfrow=c(1,1))
plot(index, index.opt, reference.grid=F)
```

```
title("Historical Performance of Portfolio Chosen on 11 April 2003\n(Equal-
weighted versus optimized)", ylab="Index", cex=0.75)
text(.5, 50, "Equal-weighted", col=2, cex=0.8)
text(.5, 90, "Optimized", col=3, cex=0.8)
```

Looking at the results of this node we obtain the following graph in Figure 24:

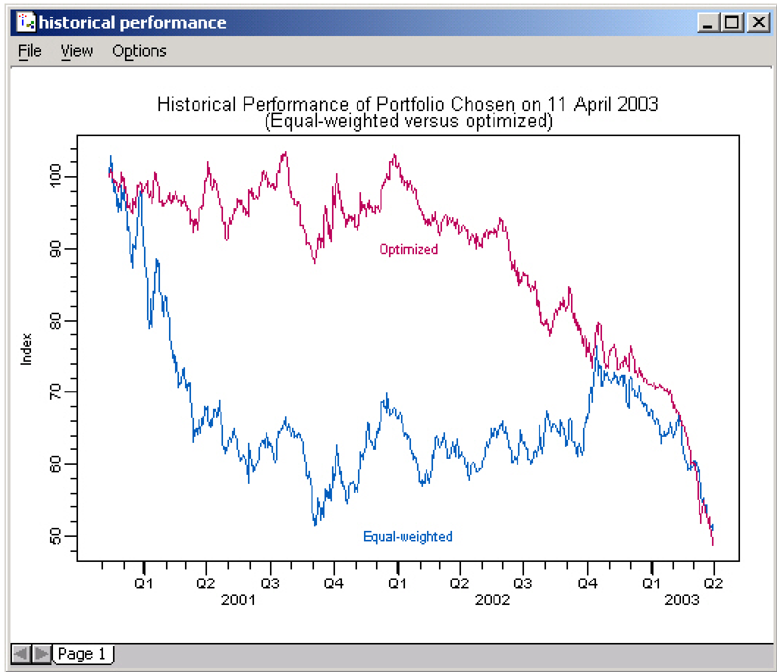


Figure 24

We observe that both portfolios would have lost money, but we already expected that from previous price and returns data exploration. Still, the optimized portfolio would have lost most of its value much later than the equal-weighted (non-optimized) portfolio. If we had observed these portfolios over a longer historical period, we would also have seen another main effect of the optimized portfolio: a much smaller volatility than the equal weighted portfolio!

Assessing Future Portfolio Performance

“Future” is used in a relative sense here, indicating that we want to see how the portfolio performances developed beyond the time limit of the train data (March 31, 2003). To this end, we repeat the exercise in the previous section (historical performance) but apply the model to the test set. We create a second S-PLUS node with the same code as above (minor changes only such as adapting the graph titles), connect it to the lower output of the **Split** node, and obtain the graph in Figure 25.

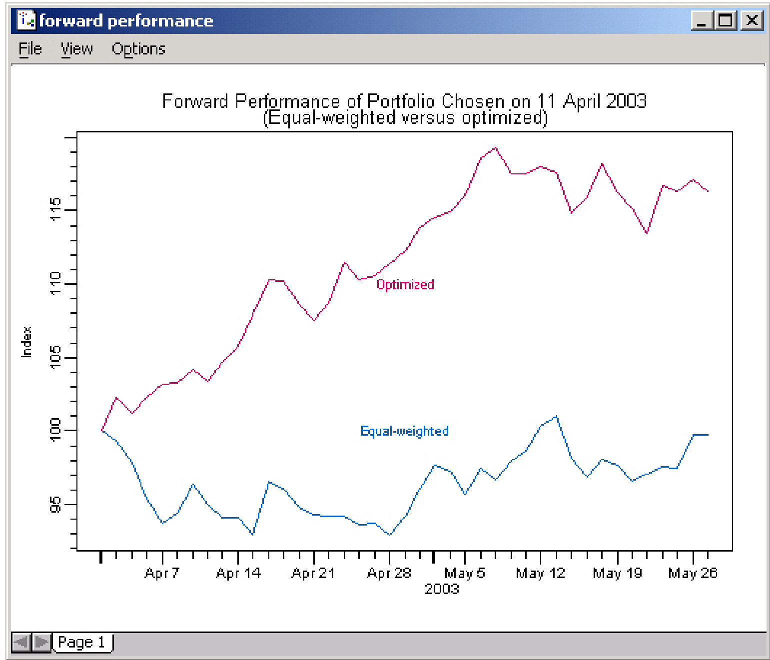


Figure 25

The difference in performance is impressive. The optimized portfolio would have increased its value by 10-15% in only one month, while the equal-weighted portfolio remains at its original value and even loses up to almost 10%.

At this point, our worksheet looks like Figure 26

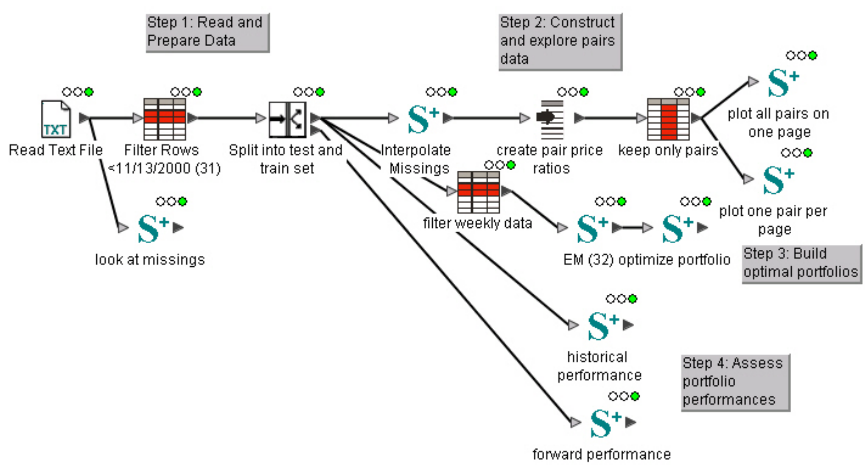


Figure 26

Conclusion and Additional Possibilities

In this Application Note, we demonstrated how to explore time series behavior of stock pairs and how to use Insightful products to design and detect a trading signal. Much more could be said regarding sophisticated signal design and detection mechanisms. For example, statistical / econometric models can be used to generate conditional forecasts of price pairs. Deviations from the model forecasts are then a signal to trade. Usually these models are back tested before actually being deploying to trigger pairs trades. Furthermore, formal unit root tests can be used to establish if a price-pair time series is stationary or not. S+FinMetrics offers a rich set of functionality that allows users to do these types of time series analysis.

Using a small sample of stock pairs, we have shown how to effectively construct an efficient long-short portfolio by optimizing the weight for each stock in the stock pairs to obtain desired risk-return characteristics. Our optimized portfolio tested out-of-sample appears to work well but same caution is required. This is a rather simplified example and a portfolio of only nine pairs is still very concentrated. We also note that annualized return for out-of-sample performance is a gross extrapolation since it is unlikely we'll hold a portfolio chosen in the manner we've outlined for such a long period.

It would be natural to consider controlling for sector and country exposures as additional constraints in our S+NuOPT quadratic programming solution. The beta neutral strategy can be easily implemented in the same fashion.

Another enhancement to our portfolio construction procedure could be modeling additional features such as illiquidity/liquidity constraints, transaction costs, minimal trading lots, etc. S+NuOPT provides powerful support for mixed-integer programming (see Bertsimas et al. 1999), which is essential for any real-life application.

Constructing a mean-variance efficient long-short portfolio is an important practice in hedge fund management. The integration of Insightful Miner, S-PLUS, S+NuOPT and S+FinMetrics results in a very powerful data analysis and modeling platform for this type of application. Pre-built analytic components and S-PLUS-based custom components combined can be used to solve typical asset management problems related to hedge funds in a completely new way.

The intuitive ease and flexibility of Insightful Miner's visual programming paradigm makes such data analysis applications easier to understand and share with others. Further, the highly scalable architecture allows handling large-data problems with no difficulty. Insightful Miner, S-PLUS, S+FinMetrics, and S+NuOPT form a powerful and seamlessly integrated analytical environment for advanced financial analysis, modeling, and deployment.

References

The following list of references has been kindly compiled by Dr. Richard Saldanha, Oxquant.

References used in this Application Note

Richard Saldanha, Insightful webcast titled “Enhanced Hedge Fund Performance via Quantitative Risk Management”, Oxquant, June 2003, available for download at http://www.insightful.com/company/events.asp

Alexander, C. (2001) Market Models: A Guide to Financial Data Analysis. John Wiley & Sons.

Bertsimas, D., Darnell, C. & Soucy, R. (Jan/Feb 1999). “Portfolio construction through mixed-integer programming at Grantham, Mayo, Van Otterloo and Company.” Interfaces 29, No. 1, pp. 49–66.

Frazzo, J., Mougeot, N. & Smith, I (2002). Initiation of European pair trading research coverage. Lehman Brothers.

Indjic, D. (2002). Strategic asset allocation with portfolios of hedge funds. *AIMA Journal*, Dec 2002. [On constructing portfolios of hedge funds their strategy mix and optimization.]
Lhabitant, F. (2002) Hedge Funds – Myths and Limits. John Wiley & Sons.

General References

Barham, S. & Hallsworth, I. (Eds) (2002). *Starting a Hedge Fund – a European Perspective* ,ISI Publications. [A practical guide for HF start-ups. The book’s principal sponsor is Morgan Stanley with co-sponsors Dechert, Hemisphere and Ernst & Young.]

Ineichen, A.M. (2003*). Absolute Returns: The Risk and Opportunities of Hedge Fund Investing*. John Wiley & Sons. [A comprehensive description of the HF industry. Aimed at investors and would-be HF managers.]

Lhabitant, F. (2002). *Hedge Funds – Myths and Limits.* John Wiley & Sons. [This book fills the gap between Barham & Hallsworth (2002) and Ineichen (2003).]

Markowitz, H.M. (1987). *Mean-Variance Analysis in Portfolio Choice and Capital Markets.* Frank J. Fabozzi Associates. (Revised reissue 2000.)

Scherer, B. (2002). *Portfolio Construction and Risk Budgeting.* Risk Books.

Selected S and S-PLUS Related Books

Bruce, A. & Gao, H. (1996). *Applied Wavelet Analysis with S-PLUS*. Springer.

Scherer, B. & Martin, D. (2003). *Portfolio Optimization using NUOPT for S-PLUS*. Springer. (In preparation.)

Venables, W.N. & Ripley, B.D. (2002). *Modern Applied Statistics with S* (4th Edition). Springer. [This is a very good textbook for S and S-PLUS users alike.]

Zivot, E. & Wang, J. (2003). *Modeling Financial Time Series with S-PLUS*. Springer. [An applied book on analyzing financial time series and certainly more than just a user's guide to S-PLUS and S+FinMetrics.]

(See also http://www.insightful.com/support/splusbooks.asp)

Web Links

http://www.math.ethz.ch/~mcneil/	Professor Alexander McNeil's homepage
http://www.infotecnet.com/	Market data provider
http://www.hfr.com/	Hedge Fund Research

DISCLAIMER

About Infotec

Infotec specialises in the acquisition, integration and delivery of real-time financial information. Founded in 1982, the company has an international client base that includes banks, brokerage houses, asset management firms and other financial institutions. Head office is in Geneva, Switzerland, with business operations also in Zürich, London, Paris and Frankfurt. For more information, visit www.infotecnet.com, email info@infotecnet.com or call +41 (0) 22 715 4433.

About Oxquant Consulting

Oxquant Consulting offers quantitative analysis, general advice and programming expertise in mathematical finance, data analysis, computation and related fields. Our clients are investment banks, fund management operations, insurance and re-insurance brokers. We talk with risk managers, heads of proprietary trading desks, private client services and hedge-fund operations. Oxquant also provides advice to software vendors and solutions providers. For more information, visit www.oxquant.com, email info@oxquant.com or call +44 (0) 1865 811160.

About Insightful Corporation

Insightful Corporation (NASDAQ: IFUL) provides enterprises with scalable data analysis solutions that drive better decisions faster by revealing patterns, trends and relationships. The company is a leading supplier of software and services for statistical data analysis, data mining and knowledge access enabling clients to gain intelligence from numeric and text data. The company's products include S-PLUS®, Insightful Miner, StatServer®, S-PLUS Analytic Server®, S+FinMetrics™, S+Wavelets®, and S+NuOPT™. Headquartered in Seattle, Insightful has offices in New York City, North Carolina, France, Switzerland, and the United Kingdom with distributors around the world. For more information, visit http://www.insightful.com/, email info@insightful.com or call 1-800-569-0123.

Contact Us

For more information contact Insightful at the information below or email info@insightful.com.

Global Headquarters
1700 Westlake Ave N
Suite 500
Seattle, WA 98103
Tel: 206.283.8802
Fax: 206.283.6310
info@insightful.com

Insightful France
7, rue Auber
31000 Toulouse
France
Tel: +33 0 5 62 27 70 60
Fax: +33 0 5 62 27 70 61
info.fr@insightful.com

Insightful Switzerland
Christoph Merian-Ring 11
4153 Reinach
Switzerland
Tel: +41 61 717 9340
Fax: +41 61 717 9341
info.ch@insightful.com

Insightful UK
5th Floor
Network House
Basing View
Basingstoke, Hampshire
RG21 4HG
Tel: +44 (0) 1256 339800
Fax: +44 (0) 1256 339839
info.uk@insightful.com