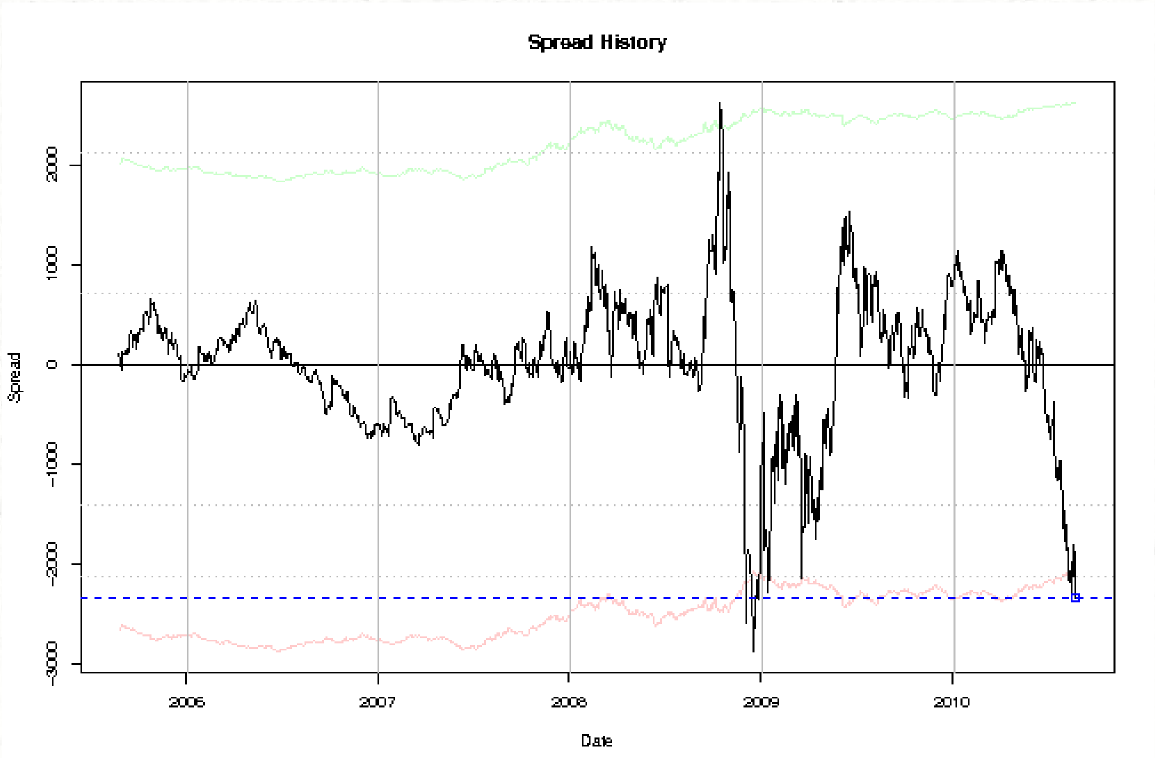
R Analytics for Spread Trading: Some Interesting Bits

Paul Teetor



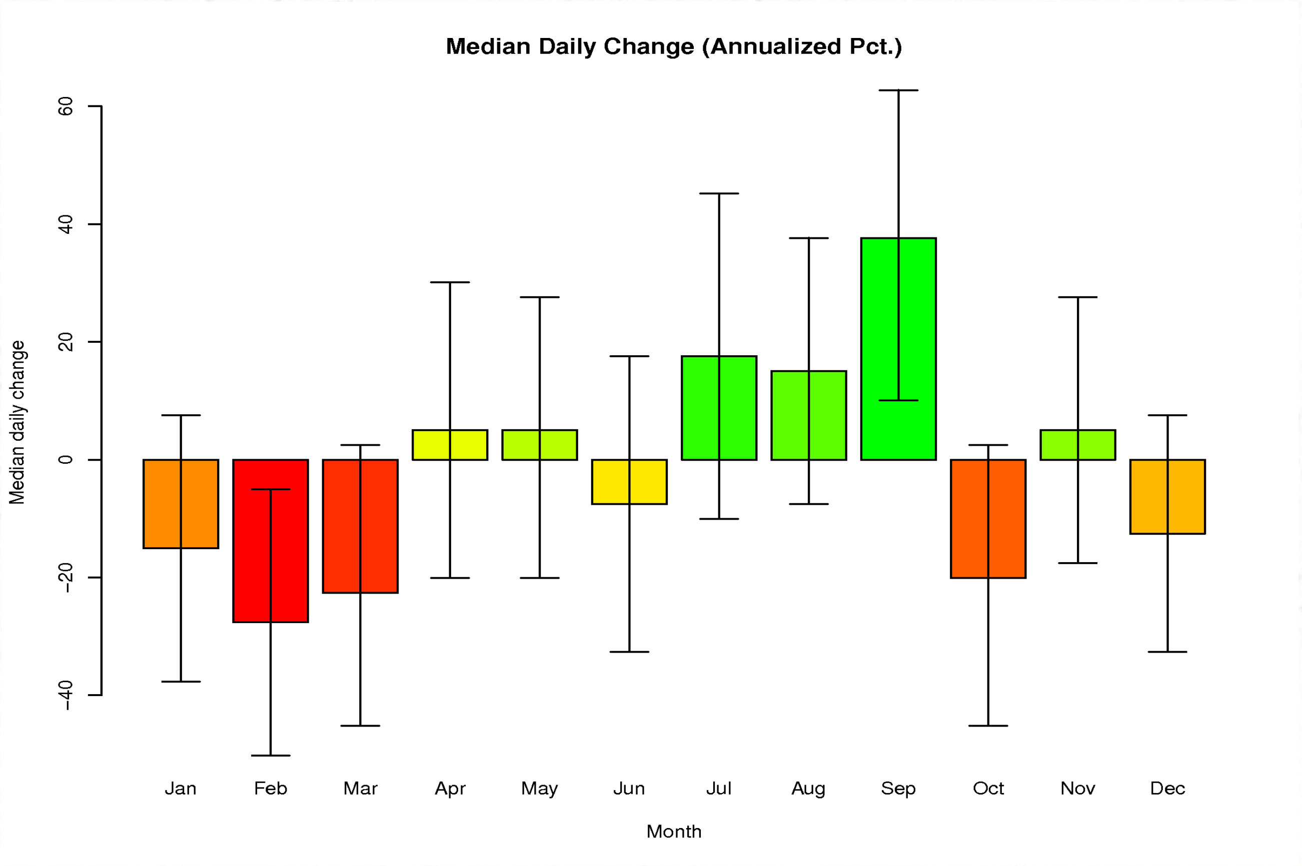
What the heck is a “spread”?

- A spread is the difference between two prices.
  - Crude oil vs. gasoline
  - Corn vs. soybeans
  - Treasury bonds vs. Treasury notes
- Some spreads have predictable properties.
  - Mean reversion: Tend back to their avg.
  - Seasonality: Price patterns by season

Software for analyzing spreads

- Apache web server on an Ubuntu box
- Perl middleware (5,000 lines)
- **R for analytics** (7,000 lines) using off-the-shelf packages and a small local library
- MySQL back-end (2,800 lines)
- Batch jobs, nightly and weekly
- Accessed entirely through browser

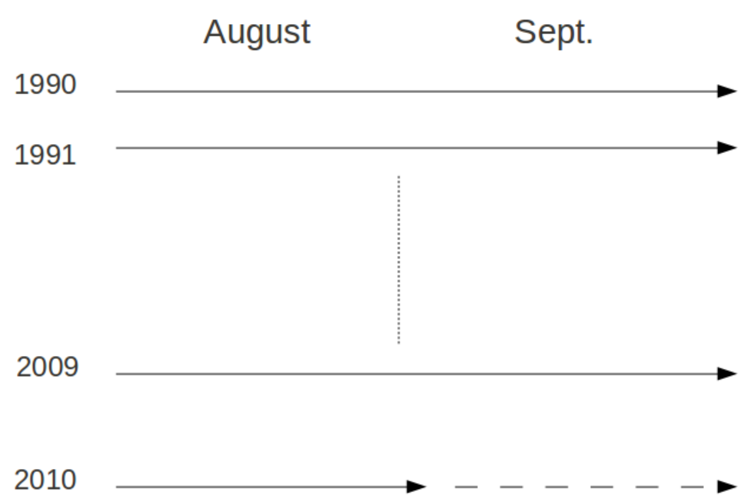
Finding Seasonal Patterns: ANOVA & Bar Charts



Bar Charts: Confidence Intervals and Colors

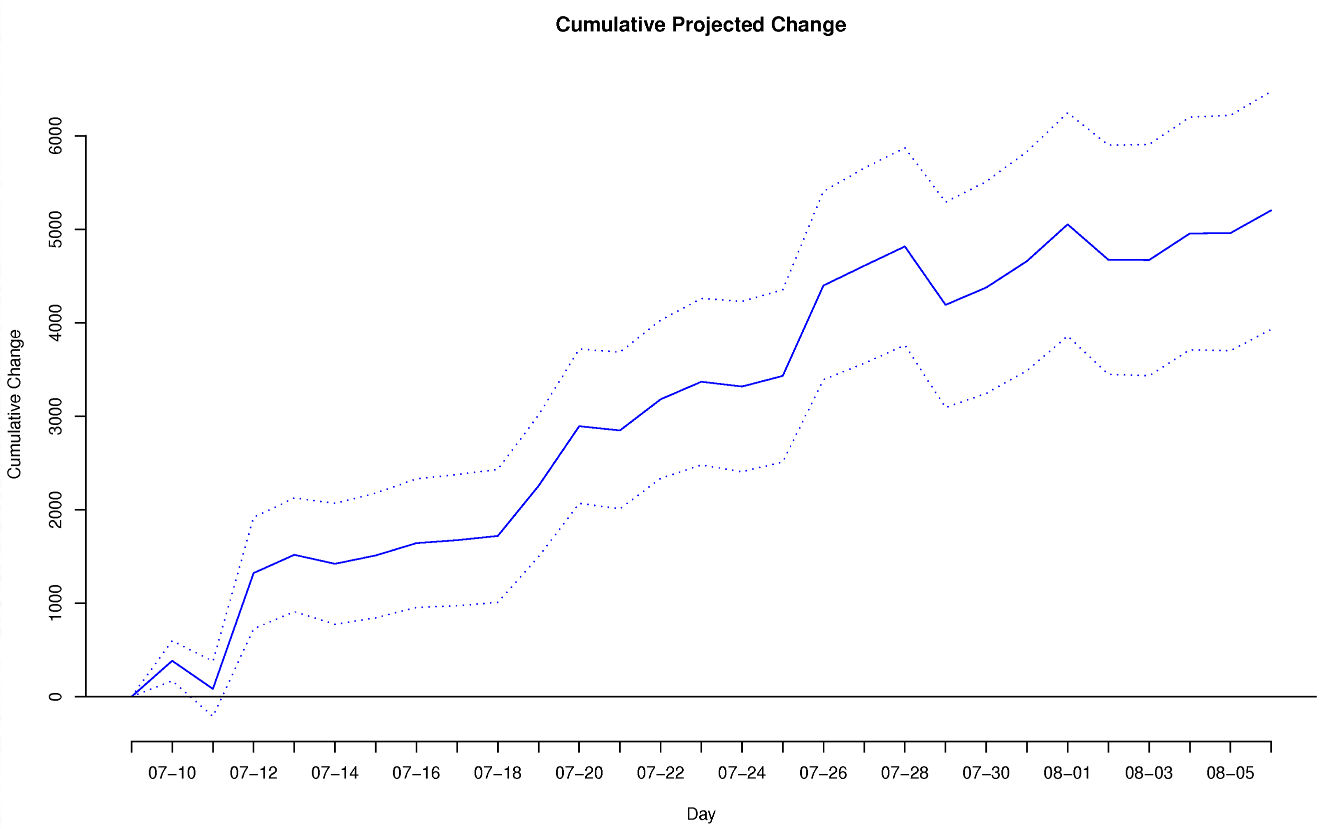
```
library(gplots)            # For barplot2 function

# . . . compute medians (bar heights) and confidence
# intervals using wilcox.test function . . .

# Build a vector of colors from red to green
cols <- rainbow(heights,start=0,end=2/6)[rank(heights)]

# Plot heights with confidence intervals and colors
barplot2(heights,
    plot.ci = TRUE, ci.u = ci.upper, ci.l = ci.lower,
   col=cols,
   . . .  )
```

Predicting a Seasonal Spread



- Want to project daily changes for Sept.
- Build a linear regression model of recent August changes using data from previous Augusts.
- Use backwards stepwise regr. to eliminate uninformative history.
- With that model and data from previous Septs, project spread changes for this Sept.

Typical Seasonal Prediction



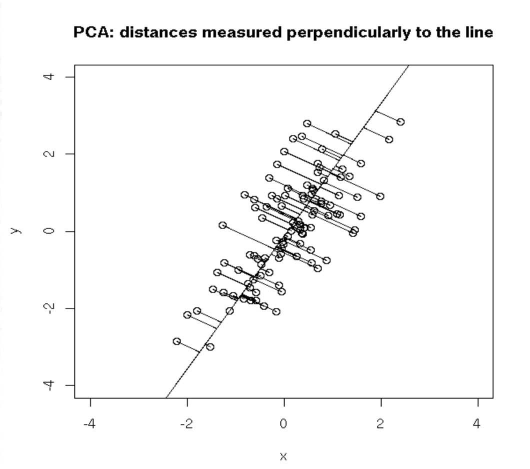
Outline of R Code

```
# Build the model
#
m <- lm(recent ~ ., data=augustHistory)

# Eliminate unhelpful years
#
m.red <- step(m, direction="backward")

# Project into September
#
pred <- predict(m.red, newdata=septHistory)
```

Hedge Ratio: Finding the ratio of two random variables

- Spread calculated as $S_i = Y_i - \beta \times X_i + \varepsilon_i$
- β is needed ratio between *Y* and *X*. How calculate?
- Tried extracting *X* coefficient from `lm(y ~ x)`
- Doesn't work well: *Ord. least squares assumes Y random, X not random (zero variance).*
- Result is that ratio for *X*/*Y* is not simply inverse of ratio for *Y*/*X*. (Oops!)

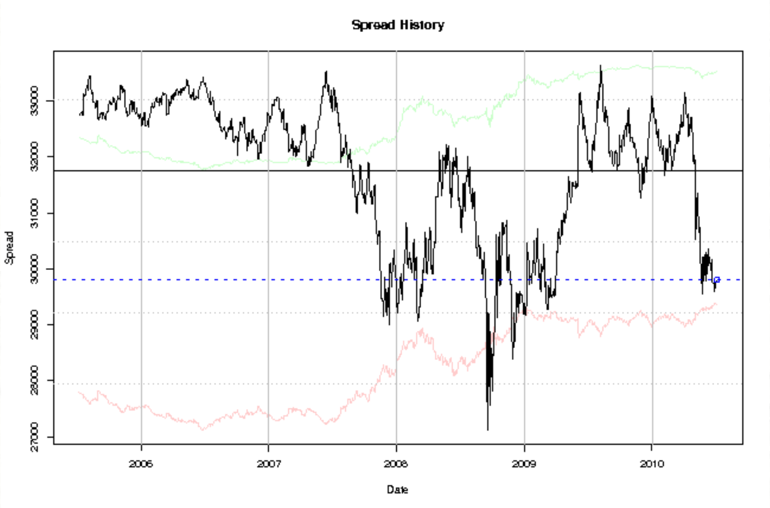
Ord. Least Squares vs. Total Least Squares

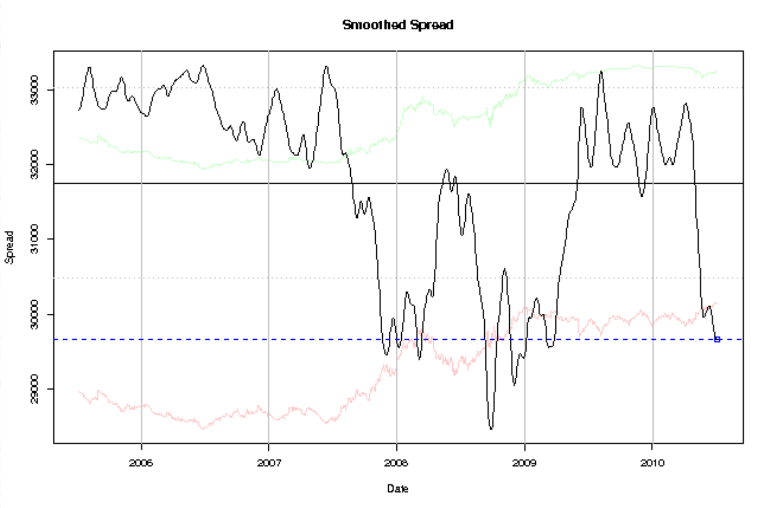


R Code for Total Least Squares
(a.k.a. Orthogonal Least Squares)

```
# Do principle components analysis (PCA)
#
r <- princomp( ~ x + y)

# Slope of orthogonal regression
#
slope <- r$loadings[2,1] / r$loadings[1,1]

# Intercept of orthogonal regression
#
inter <- r$center[2] - slope * r$center[1]
```

Removing Noise





- “Noisy” data complicates decisions
- Smoothing reveals the underlying pattern

Local Polynomial Smoothing

via KernSmooth Package

```
library(KernSmooth)

# . . . given t (time) and y . . .

N <- length(t)

# Calculate bandwidth, the degree of smoothing
# (dpill = “Direct Plug-In Meth. for Local Lin. Regr.”)
#
bw <- dpill(t, y, gridsize=N)

# Perform smoothing
#
smooth <- locpoly(x=t, y=y, bandwidth=bw, gridsize=N)
```

You, too, can build trading analytics using simple tools found around your house!

- Built mostly from off-the-shelf R packages
- Consistent application of developed statistical tools to financial data.
- Not rocket science. *It just works.*

R Analytics for Spread Trading

Paul Teetor

`paulteetor` *at* `yahoo.com`
`@pteetor`
`http://quanttrader.info/public`