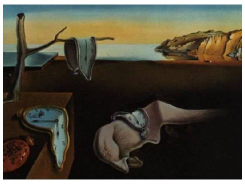

Journal of Statistical Software

March 2008, Volume 25, Code Snippet 1. http://www.jstatsoft.org/

# A Method for Visualizing Multivariate Time Series Data

**Roger D. Peng**
Johns Hopkins Bloomberg School of Public Health

**Abstract**

Visualization and exploratory analysis is an important part of any data analysis and is made more challenging when the data are voluminous and high-dimensional. One such example is environmental monitoring data, which are often collected over time and at multiple locations, resulting in a geographically indexed multivariate time series. Financial data, although not necessarily containing a geographic component, present another source of high-volume multivariate time series data. We present the `mvtsplot` function which provides a method for visualizing multivariate time series data. We outline the basic design concepts and provide some examples of its usage by applying it to a database of ambient air pollution measurements in the United States and to a hypothetical portfolio of stocks.

*Keywords*: multivariate time series, vizualization.

## 1. Introduction

Multivariate time series data are collected in a number of different fields ranging from environmental health to finance and the physical sciences. In some instances, the data may be geographically indexed, so that each individual time series can be thought of as representing a specific location. In other instances, the time series may be grouped together because they share an unobserved characteristic or are correlated for some other reason. For example, time series of stock or mutual fund prices might be grouped together because they fall into a predefined asset class.

Monitoring of environmental factors over time and space is a common component of a number of important applications. Monitoring can provide data to produce descriptive analyses of a given geographic area, to serve as early warning systems for various contaminants, or to assess compliance with regulatory mandates. Often, monitoring is conducted in many different locations using a fixed sampling rate over a period of time. Taken together, the data from each of the monitors can be thought of as a geographically indexed multivariate time series.

Monitoring data for ambient air pollutants in particular is abundant due to the US Environmental Protection Agency's mandate to enforce National Ambient Air Quality Standards throughout the country. The EPA's Air Quality System is a national network of monitors designed to collect near-daily samples of particulate matter, ozone, nitrogen dioxide, sulfur dioxide, carbon monoxide, and lead. Other important monitoring networks include the National Weather Service's network for measuring temperature and humidity and the EPA's Chemical Speciation Network for collecting data on the chemical components of fine particulate matter. Given such available data, it is important to develop tools for exploratory analysis and for summarizing the data.

## 2. The `mvtsplot` function

This code snippet was originally motivated by work in air pollution and health time series studies. Air pollution data for such health studies have a number of features which make visualization and exploratory analysis challenging. These conditions are found in many environmental applications:

1. The data are monitored over time at multiple fixed locations.
2. There may be multiple measurements being taken at a given location at a given time point.
3. There are many missing values, either systematic or random.

With just a handful of time series (perhaps 2 to 10), visualization can usually be accomplished by producing standard time plots of the data, either individually or perhaps simultaneously on a single plot. For example, Figure 1 shows five years of daily ozone levels for three large counties in the United States (Los Angeles, CA, Chicago, IL, and Bronx, NY).

When examining only three time series, it can be useful to make a plot such as Figure 1. However, for examining multiple time series, the amount of screen space required to construct a plot in this fashion becomes prohibitive. For example, in R, the `ts` method for the `plot` generic will only allow a maximum of 10 simultaneous time series to be plotted.

Another element that is lacking in Figure 1 is summary information about the time series. For example, we may want to know what the empirical distribution of values in a given time series looks like or what is the average trend across all time series. Such marginal information might be useful for providing information about higher-level trends and patterns.

We present the `mvtsplot` function for plotting multivariate time series data in R (R Development Core Team 2008). The function does a number of things to provide a useful summary of mutiple parallel time series. The basic plot is an image plot of the time series matrix. The values of each time series are discretized and assigned to distinct categories. By default, the values of each time series are divided into three categories (e.g. "low", "medium", and "high") and plotted via `image` using three colors. The colors used are taken from palettes provided by the **RColorBrewer** package (Harrower and Brewer 2003; Neuwirth 2007). By default, we use the diverging palette `PRGn`, which assigns purple to low values, grey to medium values, and green to high values.

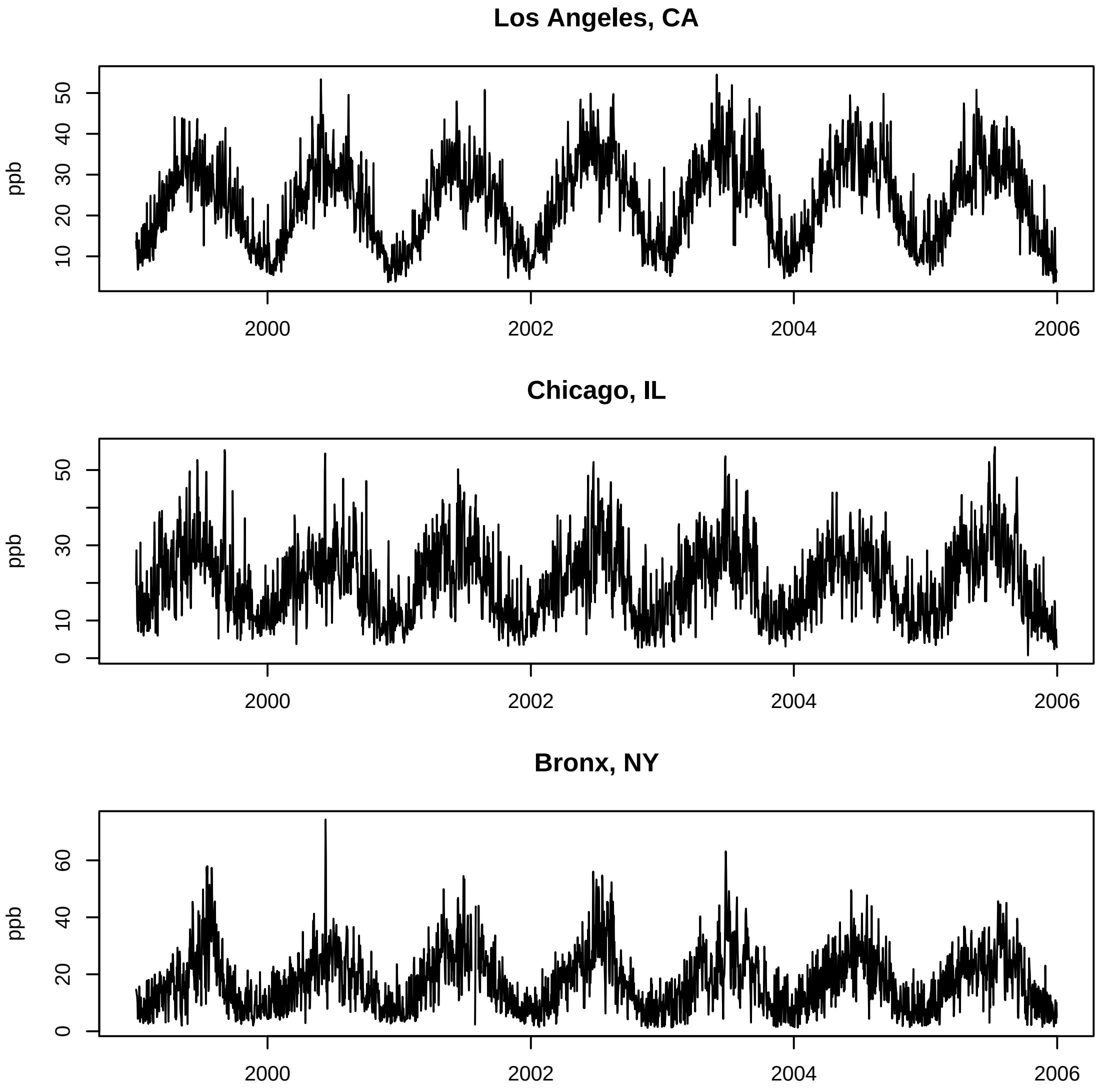

Figure 1: Daily ozone levels in parts per billion (ppb) for Los Angeles, CA, Chicago, IL, and Bronx, NY, for the years 1999–2005.

The discretization of the data values is an important aspect of the plotting code. Our default of three levels is arbitrary, but it provides the user with the ability to visualize variation in the data while also acting as a simple smoother. In particular, the discretization handles highly skewed data and outliers by not allowing those skewed values to push the remaining values into a relatively narrow range of the color spectrum. The choice of the number of levels for discretization depends on the nature of the data and is analogous to choosing the number of degrees of freedom to use in a smoother. Very smooth data can usually be plotted with many levels; very noisy or spikey data typically need to be plotted with fewer levels to be useful in an image plot. Currently, the `mvtsplot` function discretizes the data using quantiles of the time series values. Therefore, using the default of three levels implies dividing the data into tertiles with roughly an equal number of points in each.

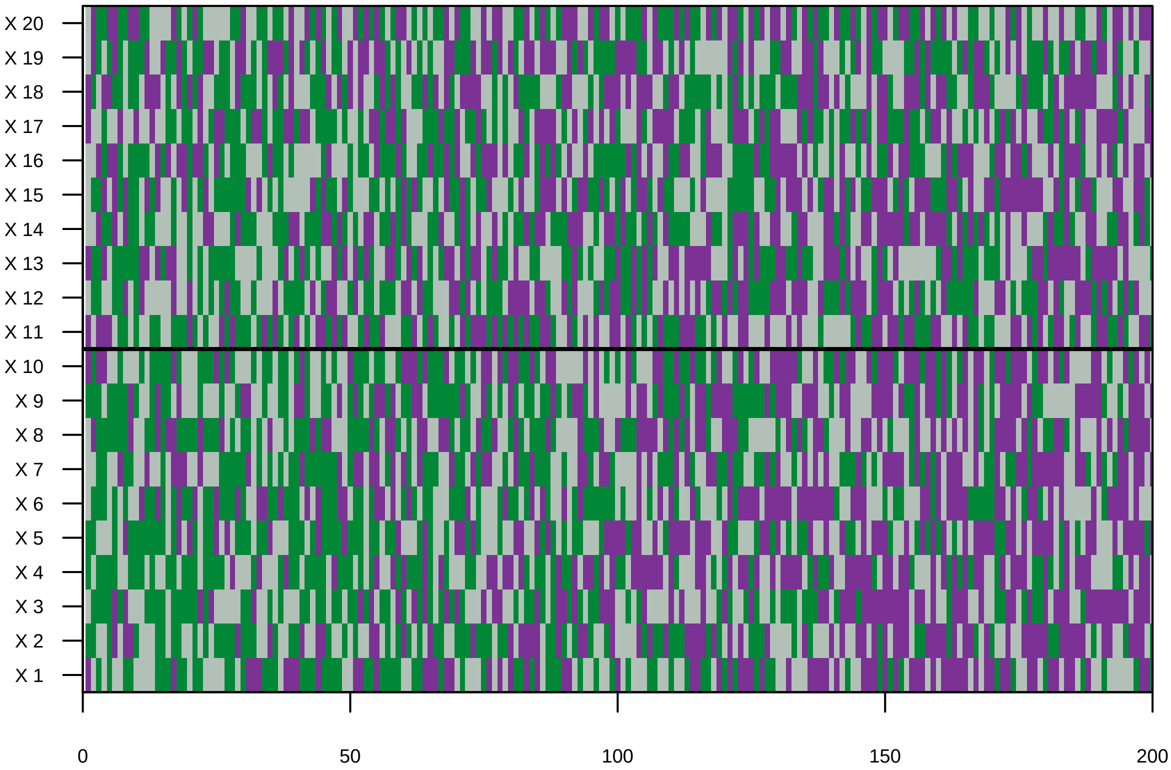

Figure 2: Simulated multivariate time series data using internal normalization.

An example of the default plot created by `mvtsplot` is shown in Figure 2 using simulated data for 20 time series of length 200.

The simulated data actually come from two groups. Ten of the series have an error distribution that is mean 0 and variance 1 while the other 10 come from a distribution that is mean 1 and variance 4. The black horizontal line in the middle of Figure 2 indicates the separation between the two groups of time series. This line is drawn when the `group` argument is non-`NULL`.

Although the two groups of time series come from different distributions, the plot appears the same for both because, by default, each time series is categorized individually. Hence, each time series is assigned to "low", "medium", and "high" using an internal definition of those categories. One can change this behavior by setting the argument `norm = "global"` which normalizes each time series using a definition of the categories based on data from all the time series, not just the time series under consideration. Figure 3 shows the data when they are normalized using the global categories.

One can see now that the top group has an overall higher level, producing mostly green values while the bottom group is lower, producing mostly purple values.

While normalizing the data using global categories can provide some insight into the differences between the time series, it is often more useful to use the internal normalization and to show information about the overall levels in the margins. Setting the `margin` option to `TRUE` (the default) makes `mvtsplot` show summary information about the time series in the bottom and right hand sides of the image plot. Specifically, on the right hand side panel are

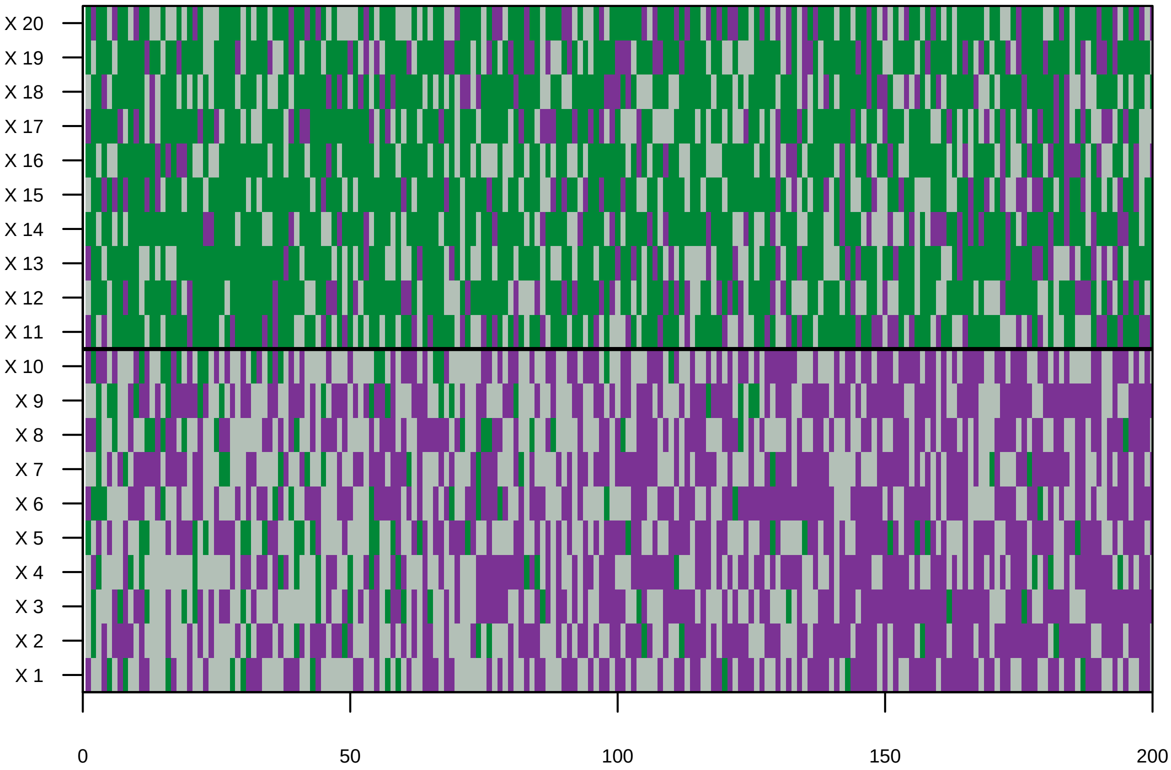

Figure 3: Simulated multivariate time series data using global normalization.

boxplots of the data in each time series and on the bottom panel are median values across all the time series for each time point. Figure 4 shows this plot for the simulated data.

Now we can see in the right hand side panel that the top group not only has a higher level but also a larger variance than the bottom group. In the bottom panel we see that on average across the 40 time series, there is a downward trend that was not as easily seen in the image plot. In Figure 4, we have also increased the number of colors used in the image to 7 from the default of 3.

In many exploratory statistical analyses, it is often useful to smooth the data to obtain a clearer picture of the trends. The `mvtsplot` function has an option called `smooth.df` which can be used to apply a natural spline smoother to each of the time series in the time series matrix. The `smooth.df` argument specifies the number of degrees of freedom to be used in the natural spline smoother for each of the time series (the same number is used for each series). Once the smoother is fit to the data, each observed value is replaced by its fitted value and then plotted. Missing data are replaced by their predicted values, so this option can be used to fill in occasional missing data. While it is probably not wise to fill in long stretches of missing data (and `mvtsplot` attempts to detect such long stretches at the beginning and at the end of the series), missing values could reasonably be filled in if the underlying process were sufficiently smooth. Missing values often obscure the overall picture in a dataset and filling in values can be useful for visualization purposes if not for inference. We make use of this in the example in Section 3.2.

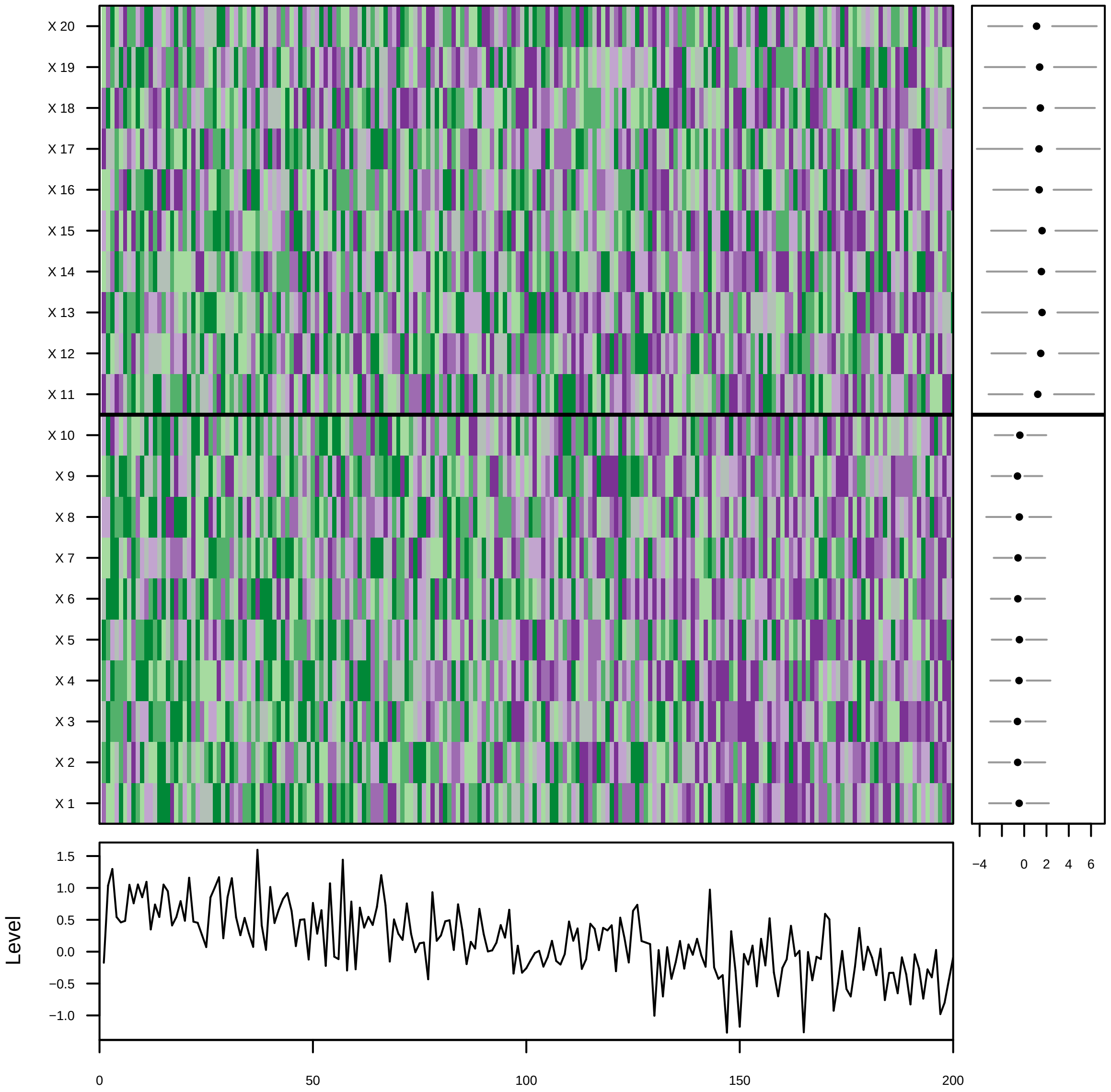

Figure 4: Simulated multivaiate time series data using internal normalization and displaying margin information.

## 2.1. Function arguments

There are a number of arguments for the `mvtsplot` function and they are shown below.

```
mvtsplot(x, group = NULL, xtime = NULL, norm = c("internal", "global"),
    levels = 3, smooth.df = NULL, margin = TRUE, sort = NULL,
    main = "", palette = "PRGn", rowstat = "median", xlim,
    bottom.ylim = NULL, right.xlim = NULL, gcol = 1)
```

These arguments can be used to control the display and the data that are plotted.

- `x`: an $n \times p$ matrix representing a multivariate time series where $n$ is the length of each series and $p$ is the number of time series. The matrix is interpreted such that the first row is the first time point and the last row is the last time point. If `x` is a data frame, it is converted to a matrix using `data.matrix`.

- `group`: a vector of length equal to the number of columns of `x`. Each element of the vector should assign a column of `x` into a group. If `group` is not a factor it will be coerced to be one.

- `xtime`: a vector of dates or times. This vector should inherit from class `Date` or `POSIXt`. If `xtime` is `NULL` (the default) then the vector `seq_len(nrow(x))` is used.

- `norm`: possible values are "internal" and "global". Should categorization of the time series be based on within-series categories ("internal") or using a global set of categories ("global")?

- `levels`: the number of categories or a vector of ranges describing the categories into which each time series should be divided. When specifying ranges, the vector should be in the form of quantiles (i.e. between 0 and 1).

- `smooth.df`: the number of degress of freedom to use in a natural spline smooth of the data. The default is `NULL` which indicates no smoothing.

- `margin`: logical, indicating whether the marginal summary information should be plotted on the bottom and right sides. The default is `TRUE`.

- `sort`: the function for determining the sorting of the columns of the time series matrix. The default (`sort = NULL`) is no sorting but one can specify an arbitrary function to sort the columns in increasing order of the values of that function applied to the columns of the time series matrix.

- `main`: a character string indicating a name for the plot. This name will be plotted in the lower right corner of the overall plot.

- `palette`: the **RColorBrewer** palette to be used for the image plot. The default is `PRGn` which uses green to indicate high values and purple to indicate low values.

- `rowstat`: the function or name of the function to be used for summarizing the rows of the time series matrix, which is subsequently plotted in the bottom margin when `margin = TRUE`. The default is to use the median.

- `xlim`: the range of x values for plotting. If `xtime` is non-`NULL`, then `xlim` should be expressed in the same values.

- `gcol`: the color of the grid lines used to indicate the boundary between different groups (only used when `group` is non-`NULL`).

- `bottom.ylim`: a vector of length 2 indicating the y axis limits to be shown in the bottom panel plot when `margin` is `TRUE`. The default is to show the entire range.

- `right.xlim`: a vector of length 2 indicating the x axis limits to be shown in the right hand side panel plot when `margin` is `TRUE`. The default is to show the entire range.

# 3. Examples

## 3.1. Ozone in the United States

Ozone ($O_3$) is one of the “criteria pollutants” monitored by the US EPA and has been shown in numerous studies to have a short-term effect on daily mortality and hospitalization (see e.g. Samet *et al.* 2000; Bell *et al.* 2004). Figure 5 shows daily ozone in the 100 largest US counties, ordered from top to bottom by decreasing latitude.

These data were obtained from the EPA’s Air Quality System. When multiple monitors for ozone were available on a given day in a given county, the values for those monitors were averaged together using a trimmed mean to create a county-wide average. From the bottom panel we see that ozone is highly seasonal across the US with a summer peak and winter trough. Indeed, ozone development depends on sunlight and is correlated with temperature. In the right hand side panel we see that the overall levels of ozone as well as the within-county ranges of variation are similar across the 100 counties.

Missing data in Figure 5 are denoted by the color white. Many of the counties have blocks of missing data equal to about six months during the late fall, winter, and early spring periods. This is because ozone is often not monitored in areas where the levels are very low during the winter season. Notice that as the latitude decreases (towards the bottom of the plot), the number of counties with all-year monitoring increases. Also, in the lower latitudes, the pattern of ozone variation tends to be less strongly seasonal, indicated by a more scattered pattern of greens and purples as opposed to long blocks of either color. This presumably reflects the relative similarity of the different seasons in the warmer latitudes.

## 3.2. Particulate matter chemical components

Starting in 1999, the US EPA began monitoring the chemical components of particulate matter (PM) in order to obtain a better understanding of how particulate matter pollution varies across the country and ultimately affects human health. Since it is thought that particle mass alone is not injurious to health, information about the levels of individual components of PM is needed to assess which aspect of PM is more or less harmful (National Research Council 2004).

The EPA’s Speciation Trends network collects data on approximately 60 components of PM less than 2.5 $\mu$m in aerodynamic diameter ($PM_{2.5}$). The measurements consist of the mass of each component (in $\mu g/m^3$) measured once every six days. Some locations take measurements once every three days. These data were downloaded from the EPA’s Air Quality System website at `http://www.epa.gov/ttn/airs/airsaqs/detaildata/downloadaqsdata.htm`. Like with the ozone data described in Section 3.1, when multiple monitors are available for a given county on a given day, the component values are averaged across monitors.

In Figure 6 we show the levels of sulfate across 98 US counties for the four year period of 2002–2005. In this plot we have separated the counties into “eastern” and “western” counties. The western counties are on the top of the plot while the eastern counties are on the bottom. Note that there are many more eastern counties than there are western counties.

Sulfate is a major component of $PM_{2.5}$ in that it makes up a substantial fraction of the total $PM_{2.5}$ mass. Hence, its spatial and temporal variation is similar to that of $PM_{2.5}$. In Figure 6 we see also that the eastern counties have on average higher levels of sulfate and

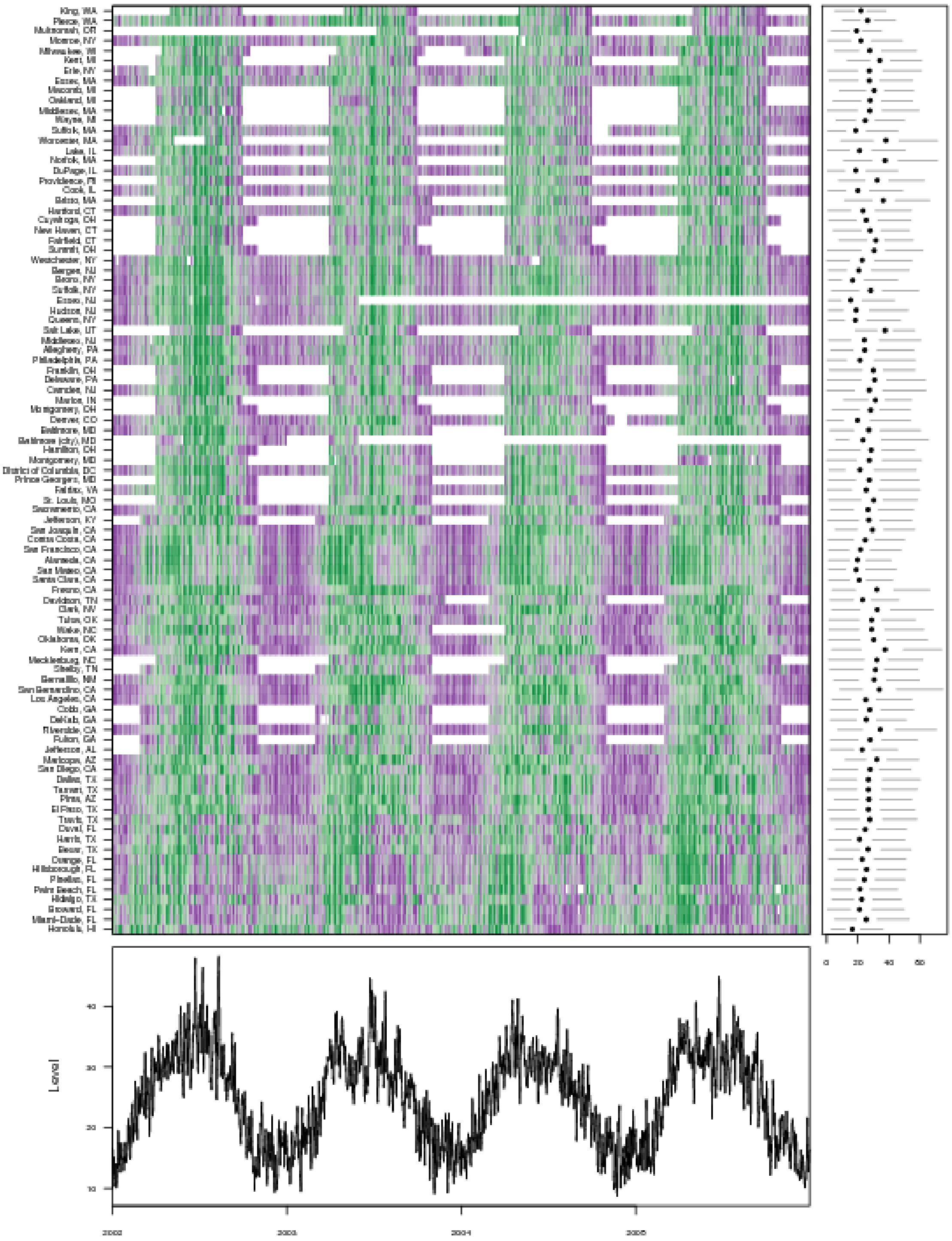

Figure 5: Daily ozone for the largest 100 US counties, 2002–2005. Counties are sorted from top to bottom by decreasing latitude.

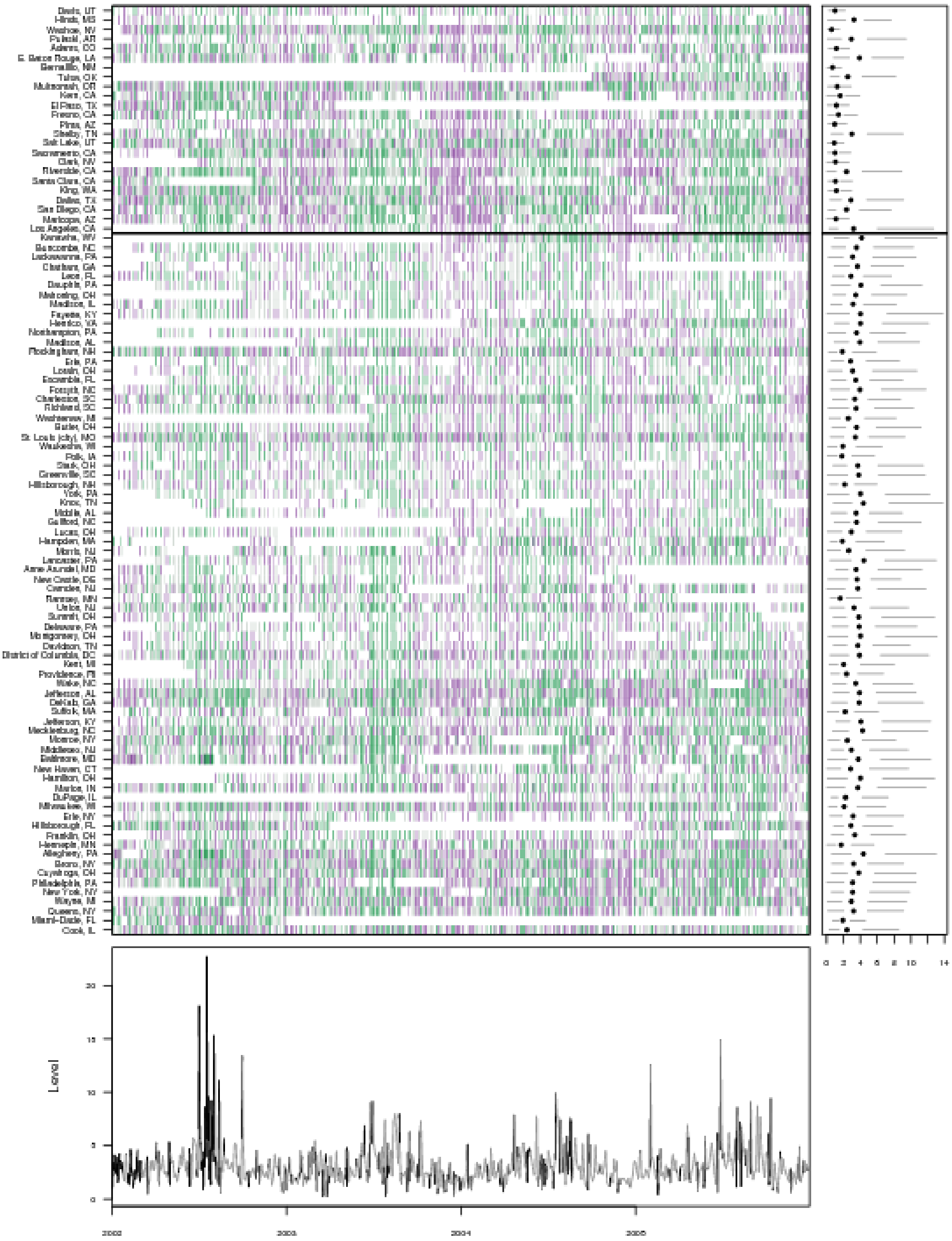

Figure 6: Daily sulfate levels for 98 US counties, 2002–2005.

higher variation around their medians. The western counties generally have lower levels of sulfate as a component of $PM_{2.5}$. This phenomenon is not due to generally higher levels of $PM_{2.5}$ in the east versus the west but is rather a reflection of the different sources of $PM_{2.5}$ in the two regions of the country (Bell *et al.* 2007).

Temporal patterns in sulfate are somewhat difficult to discern in Figure 6 because of the substantial noise in the data. We can smooth the individual time series to get a better sense of the temporal patterns. The `smooth.df` argument to `mvtsplot` allows the user to specify the number of degrees of freedom to use in a natural spline smooth of each time series. The same degrees of freedom is used to smooth each of the time series in the matrix. The result of smoothing the sulfate time series is shown in Figure 7. Here we use a natural spline smoother with 16 degrees of freedom. We see clearly that sulfate tends to peak in the summer time and reach its lowest levels in the winter. The bottom panel shows the median of the smooths across counties. We also see that this general seasonal pattern holds for almost all of the counties, regardless of what region they fall in.

One component of $PM_{2.5}$ that is thought to be particularly harmful is nickel (Lippmann *et al.* 2006). Compared to sulfate, nickel makes up a very small fraction of total $PM_{2.5}$ mass. Figure 8 shows the smoothed daily nickel levels for the 98 US counties, sorted by their overall median level. The data here are smoothed using a natural spline smoother with 32 degrees of freedom.

In contrast to sulfate, we do not see any obvious seasonal or other temporal trend in the data, either overall or for any particular county. Also, we can see that in general, levels of nickel are very low with the exception of a few counties. Bronx, Queens, and New York counties (all in New York state) all have levels of nickel that are much higher than in all other counties. Similarly, these three counties have a much larger range of variation across time. One anomaly in Figure 8 is a large peak in the wither of 2005. It is not immediately clear why such a relatively large peak occurs during this time across many different counties except that nickel tends to be highly variable across time and that such a peak simply occurred by chance.

Figure 9 shows 40 of the 56 components measured by the Speciation Trends Network for Bronx County, New York.

The 40 components shown in Figure 9 are the 40 largest components as a percent of the total $PM_{2.5}$ mass. We see from the right hand side panel that handful of components—sulfate, organic carbon (`OC_K14`), ammonium, nitrate, and elemental carbon—dominate the total $PM_{2.5}$ mass. The lower panel uses a different summary statistic than the previous plots. Since the median component level across components is not interesting, we set the `rowstat` argument to be the `sum` function so that the bottom panel shows the sum of all the component levels, closely approximating the total $PM_{2.5}$ mass. Note that we have applied a smoother with 16 degrees of freedom to each of the columns of the component matrix.

One interesting feature of the data that is shown in the bottom panel is the presence of roughly two peaks in each calendar year, one in summer and one in winter. From the image plot, it would appear that the summer peak is largely driven by the levels of organic carbon and sulfate while the winter peak is driven by nitrate and elemental carbon. Ammonium appears to correlate with both peaks, which is not surprising since it appears in the air as either ammonium nitrate or ammonium sulfate.

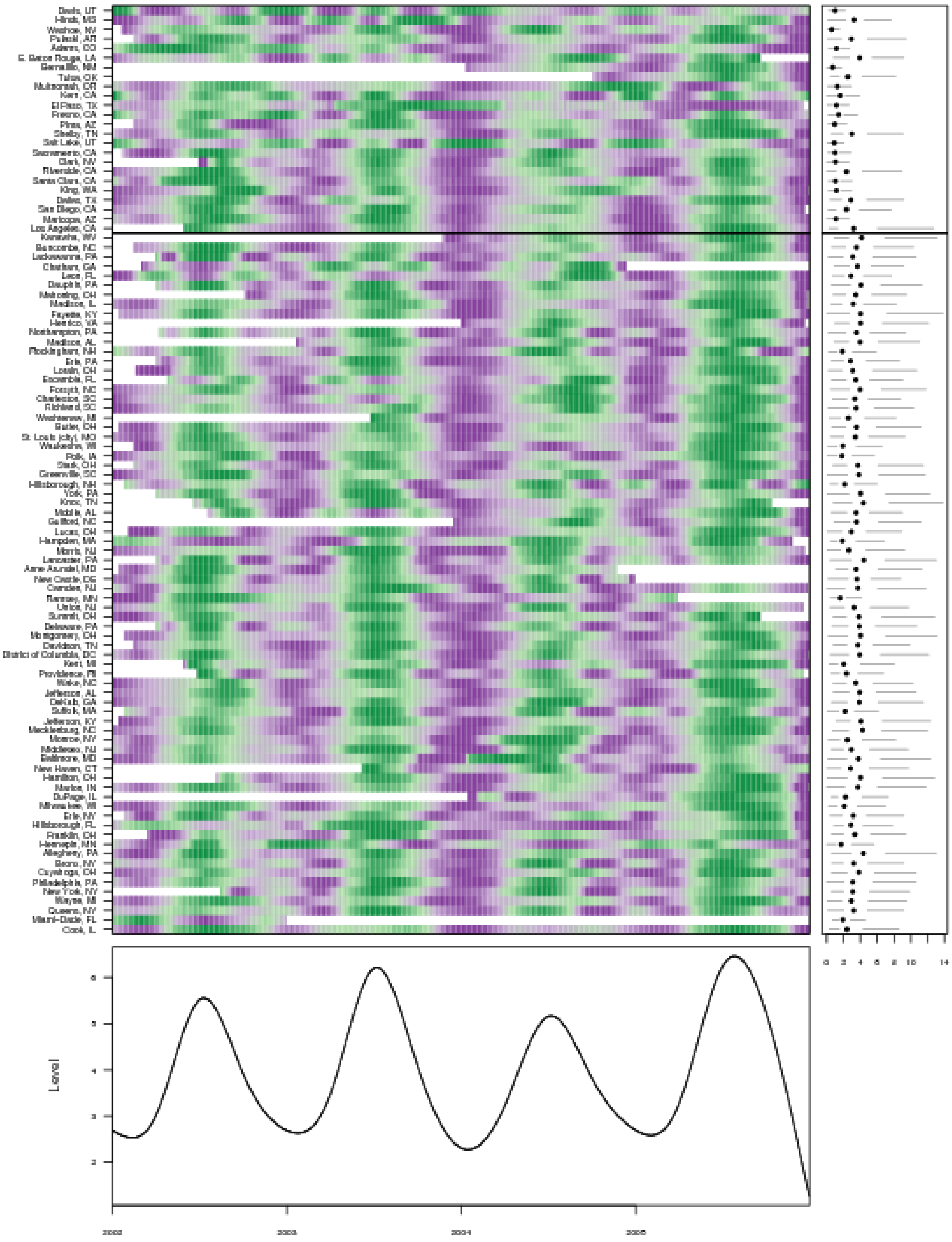

Figure 7: Daily smoothed sulfate levels for 98 US counties, 2002–2005.

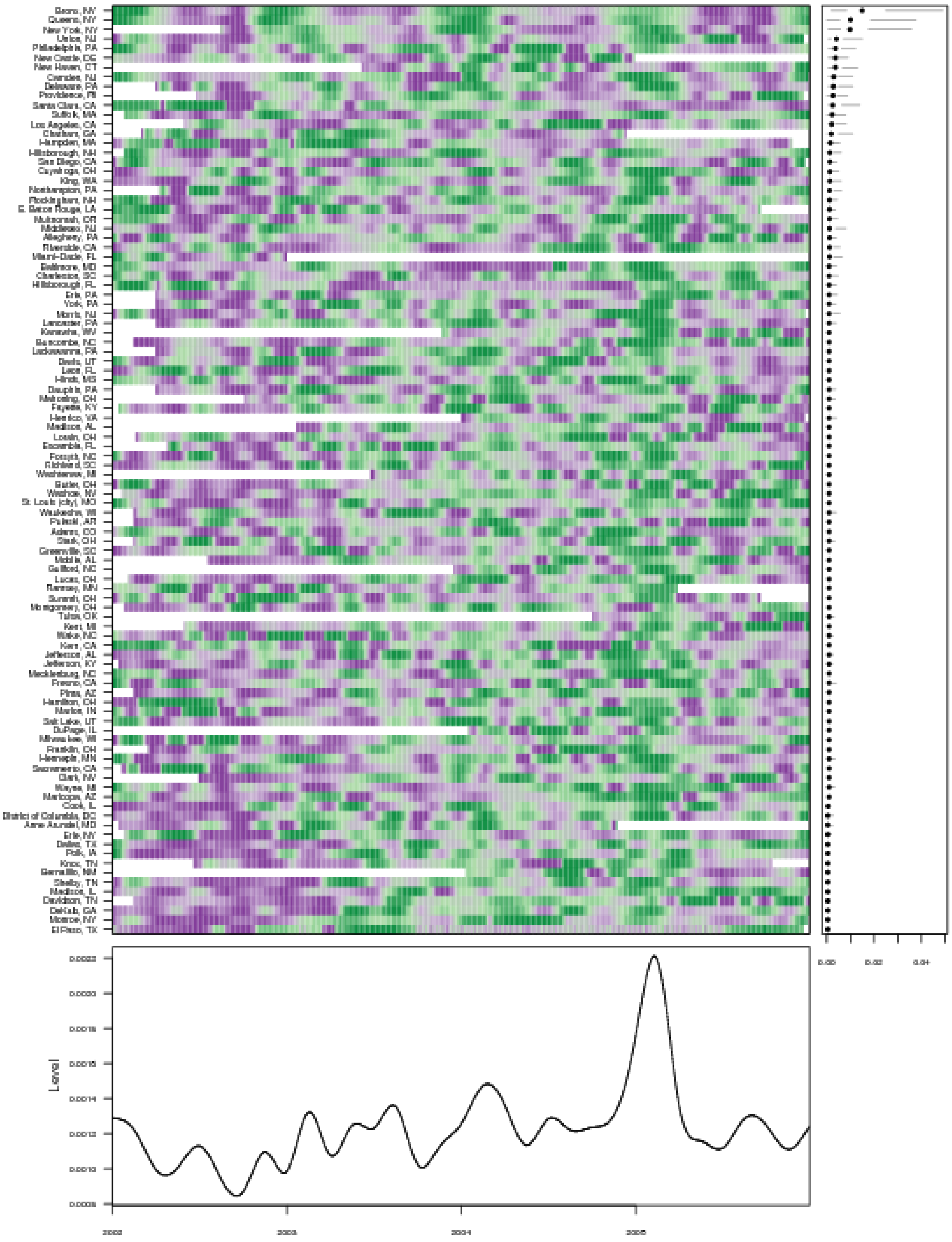

Figure 8: Daily smoothed nickel levels for 98 US counties, 2002–2005.

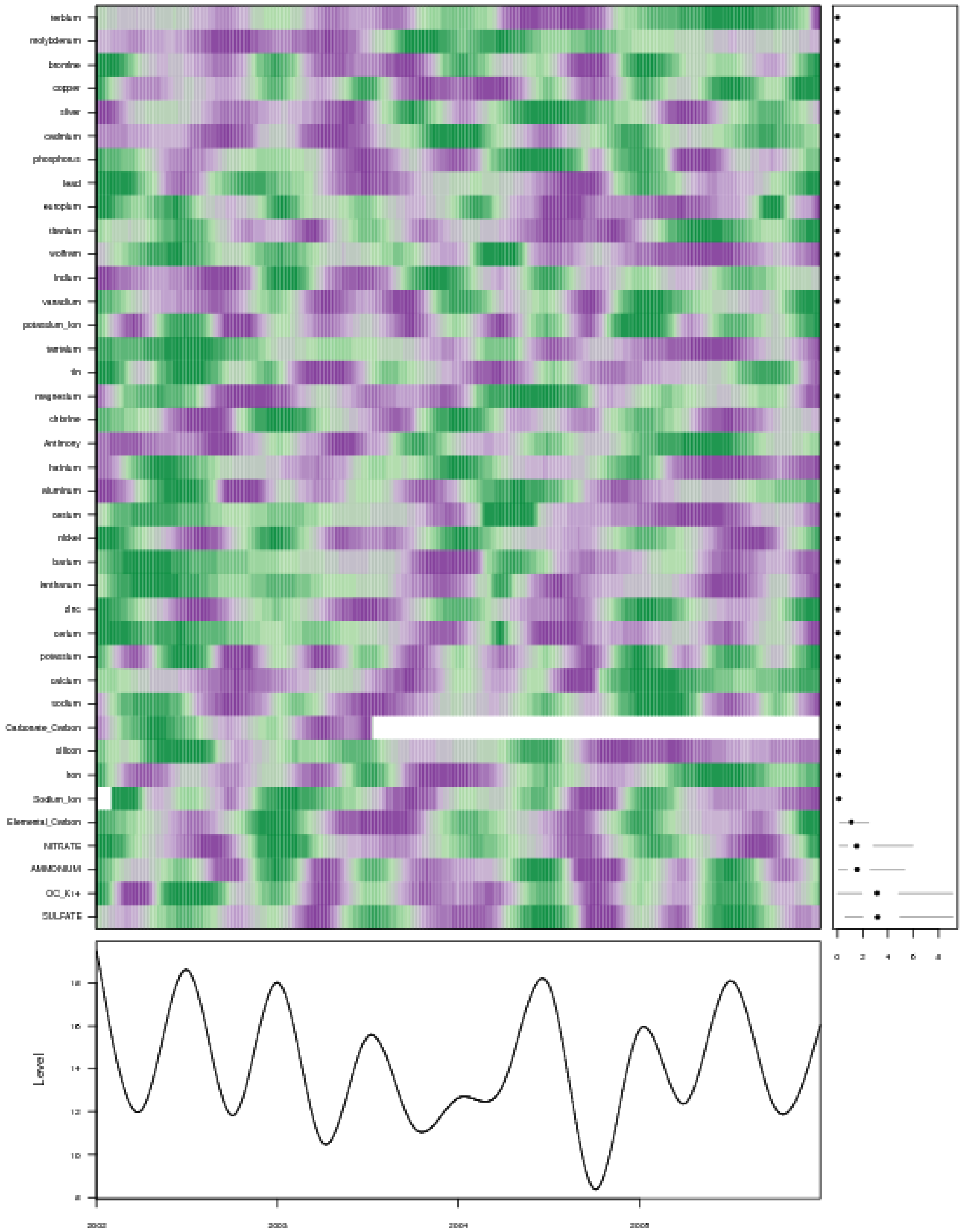

Figure 9: Daily smoothed levels of the largest 40 components of $PM_{2.5}$ (by mass) for Bronx County, New York, 2002–2005.

### 3.3. Hypothetical stock portfolio

Financial time series data are arguably more abundant than environmental data and there are a number of resources freely available on the Web for obtaining these data. In particular, websites like Yahoo! and others provide free historical price data for stocks and mutual funds in the US. The R user already has a number of tools for downloading financial data, including the **tseries** package (Trapletti and Hornik 2007). There are numerous other packages available on CRAN for analyzing financial time series data and for calculating many common statistics. These packages can be found in the "Empirical Finance" task view of CRAN (`http://CRAN.R-project.org/view=Finance`).

In this section we present an example of visualizing an hypothetical portfolio of stocks and exchange traded funds (ETFs). We have collected daily price data for a portfolio consisting of an S&P 500 index exchange traded fund (ticker symbol SPY), an ETF tracking the Lehman Brothers Aggregate Bond Index (AGG), an ETF tracking an index of Treasury Inflation Protected Securities (TIP), and nine large real estate investment trusts (REITs)[1]. The data were obtained using the `get.hist.quote` function from the **tseries** package.

Figure 10 shows the portolio over the period 2006–2007. We took an hypothetical $10,000 investment and divided it equally between the portfolio components. The bottom panel shows the sum of the components, i.e. the market value of the entire portfolio. One can see that the REITs rose dramatically in 2006 and early 2007 and then declined towards the end of the period. The S&P 500 index fund increases in value more or less steadily throughout the period but its relatively small allocation is not sufficient to prop up the market value of the portfolio. The right hand side panel shows the differences in variation between the holdings. As one might expect, the two bond funds indicate the least variation while the REITs indicate much higher levels of variability.

## 4. Discussion

We have presented the `mvtsplot` function for producing plots of large-scale multivariate time series data that can be useful for exploratory analysis prior to formal model fitting. We have demonstrated the function's usage by applying it to data on ambient air pollution monitoring as well as stock prices.

Many choices had to be made in order to make the default plots reasonable. In particular, an attempt is made to calculate the appropriate positioning of the axis labels, but this system is easily thwarted. The `mvtsplot` function also uses the `layout` function to position the different panels when `margin = TRUE`, and so subsequent modification of the individual panels is not possible. We encourage interested users to make their own modifications to the code to suit their unique purposes.

## 5. Code repository

The most recent and future versions of the `mvtsplot` function can be found at `http://www.biostat.jhsph.edu/~rpeng/RR/mvtsplot/`. A history of the changes can be found at the project's git repository located at `http://repo.or.cz/w/mvtsplot.git`.

[1] Ticker symbols SPG, PLD, VNO, BXP, EQR, GGP, HST, KIM, PSA

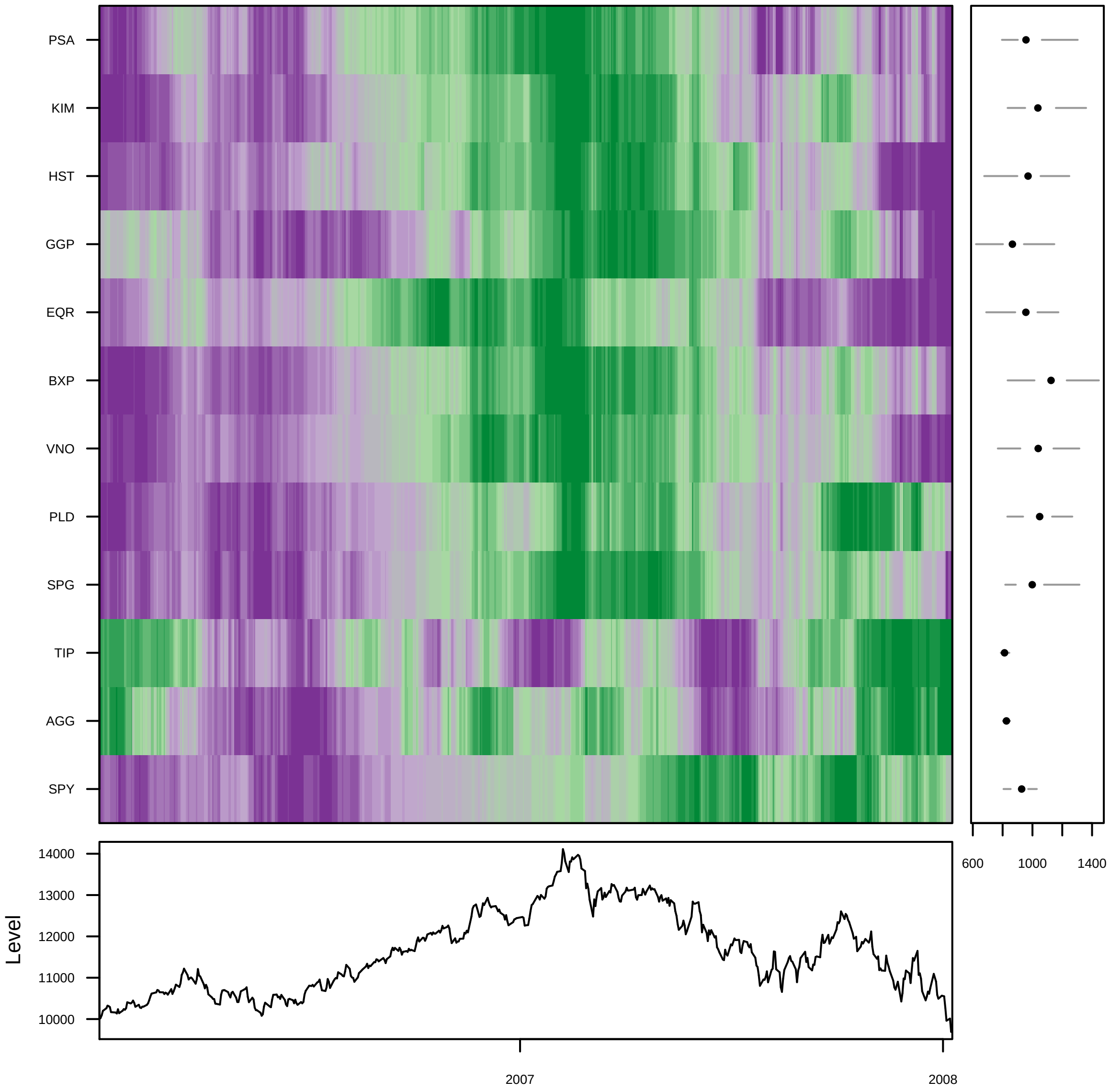

Figure 10: Daily prices and market value for a hypothetical portfolio of an S&P 500 exchange traded fund (ETF), two bond ETFs, and nine real estate investment trusts, 2006–2007.

## References

Bell ML, Dominici F, Ebisu K, Zeger SL, Samet JM (2007). "Spatial and Temporal Variation in $\mathrm{PM}_{2.5}$ Chemical Composition in the United States for Health Effects Studies." *Environmental Health Perspectives*, **115**, 989–995.

Bell ML, McDermott A, Zeger SL, Samet JM, Dominici F (2004). "Ozone and Short-Term Mortality in 95 US Urban Communities, 1987-2000." *Journal of the American Medical Association*, **292**(19), 2372–2378.

Harrower MA, Brewer CA (2003). "**ColorBrewer.org**: An Online Tool for Selecting Color Schemes for Maps." *The Cartographic Journal*, **40**, 27–37. URL http://ColorBrewer.org/.

Lippmann M, Ito K, Hwang JS, Maciejczyk P, Chen LC (2006). "Cardiovascular Effects of Nickel in Ambient Air." *Environmental Health Perspectives*, **114**(11), 1662–1669.

National Research Council (2004). *Research Priorities for Airborne Particulate Matter: IV. Continuing Research Progress.* National Research Council of the National Academies.

Neuwirth E (2007). ***RColorBrewer**: ColorBrewer palettes.* R package version 1.0-2, URL http://CRAN.R-project.org/package=RColorBrewer.

R Development Core Team (2008). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-07-0, URL http://www.R-project.org/.

Samet JM, Zeger SL, Dominici F, Curriero F, Coursac I, Dockery DW, Schwartz J, Zanobetti A (2000). *The National Morbidity, Mortality, and Air Pollution Study, Part II: Morbidity and Mortality from Air Pollution in the United States.* Health Effects Institute, Cambridge.

Trapletti A, Hornik K (2007). ***tseries**: Time Series Analysis and Computational Finance.* R package version 0.10-12, URL http://CRAN.R-project.org/package=tseries.

### Affiliation:

Roger D. Peng
Department of Biostatistics
Johns Hopkins Bloomberg School of Public Health
615 North Wolfe Street
Baltimore MD 21205, United States of America
E-mail: rpeng@jhsph.edu
URL: http://www.biostat.jhsph.edu/~rpeng/

*Journal of Statistical Software* http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume 25, Code Snippet 1 *Submitted:* 2008-01-15
March 2008 *Accepted:* 2008-02-25