

## ePub<sup>WU</sup> Institutional Repository

Abel Guy

fanplot: An R Package for Visualising Sequential Distributions

Article (Published)  
(Refereed)

*Original Citation:*

Guy, Abel (2015) fanplot: An R Package for Visualising Sequential Distributions. *The R Journal*, 7 (1). pp. 15-23. ISSN 2073-4859

This version is available at: <http://epub.wu.ac.at/5910/>

Available in ePub<sup>WU</sup>: December 2017

ePub<sup>WU</sup>, the institutional repository of the WU Vienna University of Economics and Business, is provided by the University Library and the IT-Services. The aim is to enable open access to the scholarly output of the WU.

This document is the publisher-created published version. It is a verbatim copy of the publisher version.

# fanplot: An R Package for Visualising Sequential Distributions

by Guy J. Abel

**Abstract** Fan charts, first developed by the Bank of England in 1996, have become a standard method for visualising forecasts with uncertainty. Using shading fan charts focus the attention towards the whole distribution away from a single central measure. This article describes the basics of plotting fan charts using an R add-on package alongside some additional methods for displaying sequential distributions. Examples are based on distributions of both estimated parameters from a time series model and future values with uncertainty.

## Introduction

Probabilities are notoriously difficult to communicate effectively to lay audiences (Spiegelhalter et al., 2011). Fan charts provide one such method to illustrate either forecasts or past results that are based on probabilistic distributions. Using shading fan charts focus the attention of the reader on the whole distribution away from a single central estimate. Visualising the distribution can aid in communicating the degree of underlying uncertainty in probabilistic forecasts to non-specialists, that might not have been apparent in basic plots and summary statistics.

Fan charts were first introduced by the Bank of England for their inflation forecasts in February 1996 (Britton et al., 1998). Since their initial development fan charts have become a standard method to display uncertainty of future economic indicators by many central banks (Julio, 2007). Their use has also spread to other fields such as climate science (McShane and Wyner, 2011) and demography (Gerland et al., 2014).

Fan charts can be created using various software. Within R, the `vars` package (Pfaff, 2008) has a `fanchart` function for forecasts of confidence regions. It is based solely on “varpred” class objects, i.e., on the predictions of Vector Autoregressive models fitted using other functions within the `vars` package. Similarly, the `forecast` package (Hyndman and Khandakar, 2008) produces fan charts for forecasts based on time series models from the “forecast” class. Julio (2009) provides VBA code in order to plot fan charts for quarterly GDP data in Excel. Alternatively one could use point and click methods in Excel to build customised fan charts based on stacked area charts.<sup>1</sup> Buchmann (2010) provides MATLAB code to create fan charts for user supplied forecast distributions with a limited amount of control for the plotted display.

In any of the fore-mentioned options users are restricted in either their ability to effectively adapt the properties of fan charts or create plots based on alternative models, values or simulated data. The aim of this article is to illustrate R code in the `fanplot` package to create fan charts of different styles and from a range of input data. These are demonstrated on data from sequential Monte Carlo Markov chain (MCMC) simulated distributions of parameters in a stochastic volatility model and expert based forecasts for the Consumer Price Index (CPI) of the Bank of England.

## Fan charts for sequential simulated distributions

The `fanplot` package can be used to display any form of sequential distributions along a plot's x-axis. To illustrate, we use posterior density distributions of the estimated volatility of daily returns ( $y_t$ ) from the Pound/Dollar exchange rate from 02/10/1981 to 28/6/1985. As Meyer and Yu (2002) show, posterior distributions for the volatility process can be estimated in WinBUGS by fitting the stochastic volatility model;

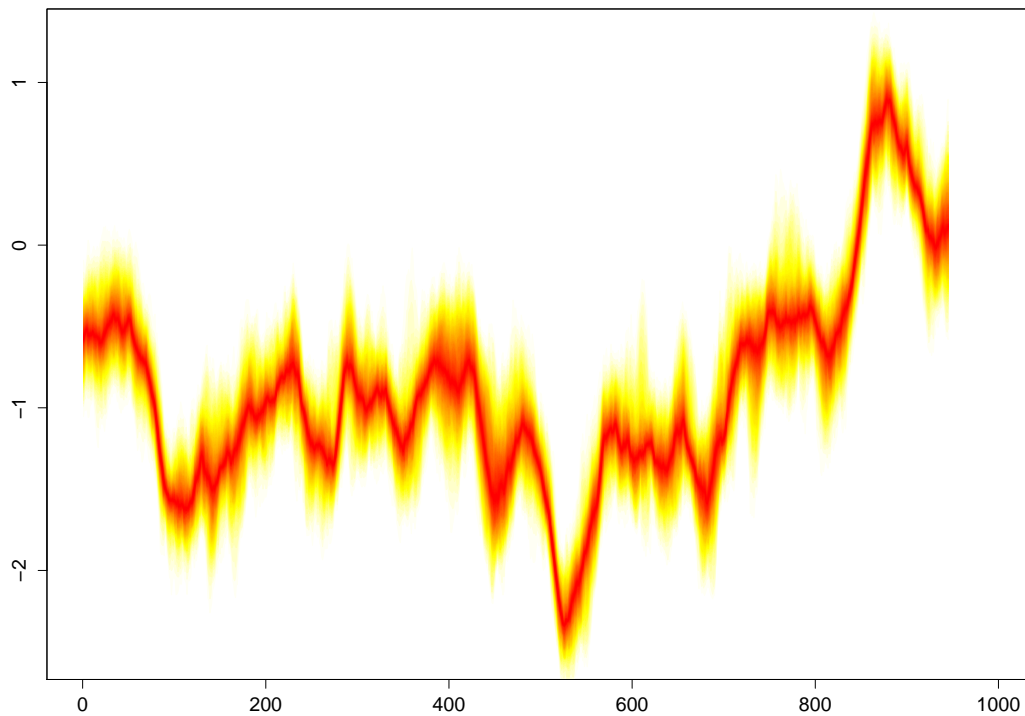
$$y_t | \theta_t = \exp\left(\frac{1}{2}\theta_t\right) u_t \quad u_t \sim N(0, 1) \quad t = 1, \dots, n. \quad (1)$$

The latent volatilities  $\theta_t$ , which are unknown states in a state-space model terminology (Harvey, 1990), are assumed to follow a Markovian transition over time given by the state equations:

$$\theta_t | \theta_{t-1}, \mu, \phi, \tau^2 = \mu + \phi \log \sigma_{t-1}^2 + v_t \quad v_t \sim N(0, \tau^2) \quad t = 1, \dots, n \quad (2)$$

with  $\theta_0 \sim N(\mu, \tau^2)$ .

<sup>1</sup>See <http://peltiertech.com/WordPress/excel-fan-chart-showing-uncertainty-in-projections> for an Excel tutorial.



**Figure 1:** Stochastic Volatility Fan Chart for the Pound-Dollar Exchange Rate Data.

A sample of the posterior distributions of  $\theta_t$  is contained in the `th.mcmc` object of the **fanplot** package. It consists of (1000) rows corresponding to MCMC simulations and (945) columns corresponding to time points  $t$ . Example code to replicate this object using **R2OpenBUGS** (Sturtz et al., 2005) is given in the help file for the `th.mcmc` object. It is based on the BUGS model of Meyer and Yu (2002) replicated in the `my1.R` file of the **fanplot** package. Time ordered simulated distributions, such as `th.mcmc`, can be easily extracted from the `sims.list` element of an **R2OpenBUGS** “bugs” object.

A fan chart of the evolution of the distribution of  $\theta_t$  in Figure 1 can be plotted using either the `fan0` or `fan` function. The `fan0` function, which we will first discuss, provides the simplest representation;

```
library("fanplot")
fan0(data = th.mcmc)
```

The plotting function calculates the values of 100 equally spaced percentiles of each future distribution when the default `data.type = "simulations"` is set. This allows 50 fans to be plotted from the `heat.colours` colour palette, providing darker shadings for the more probable percentiles. The axis limits are determined from the data argument. By default, the y-axis limits to 85 percent of the range of the MCMC distributions to reduce white space in the plot.

Similar plots of sequential distributions from alternative Bayesian models can be easily plotted using the `fan0` or `fan` functions. The data argument accepts objects from a range of classes including “mcmc” which is typically used to handle BUGS or JAGS results via the `read.coda` or `read.jags` commands.

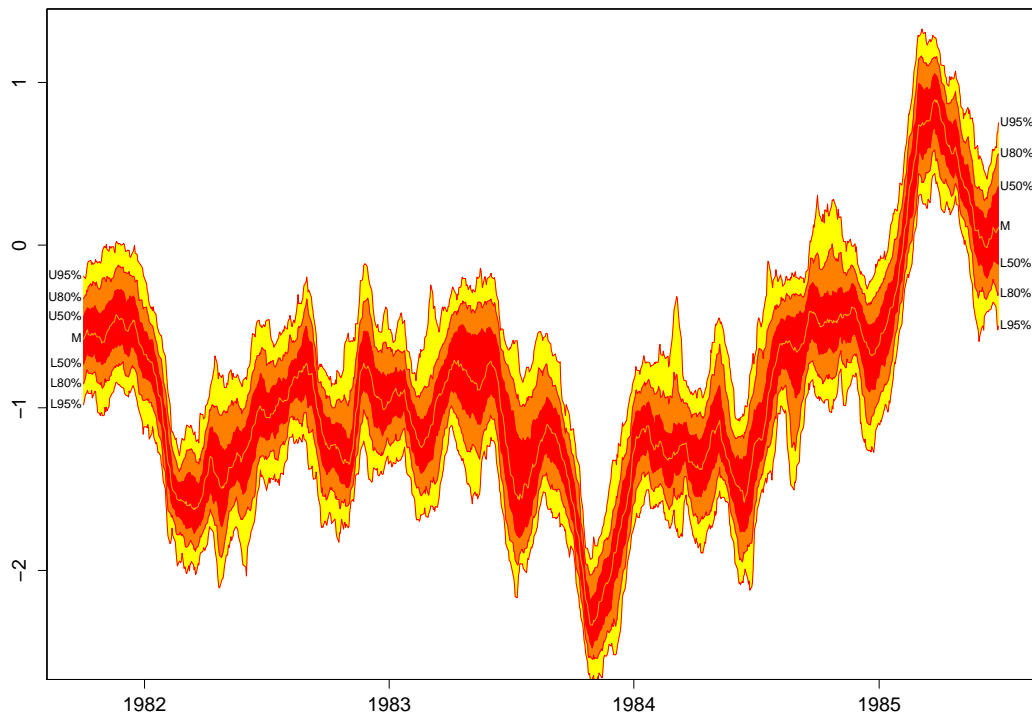
The data in `th.mcmc` are based on trading day observations only. Irregular time series can be handled by passing a **zoo** time series object (Zeileis and Grothendieck, 2005) to the data argument. The trading days are given in the `spvdx` object of the **tsbugs** package (Abel et al., 2013).

```
library("zoo")
library("tsbugs")

# create irregular multiple time series object
th.mcmc2 <- zoo(th.mcmc, order.by = spvdx$date)

# plot
fan0(data = th.mcmc2, type = "interval", ln = c(0.5, 0.8, 0.95),
      llab = TRUE, rcex = 0.6)
```

Basing the fan chart on a **zoo** time series allows the x-axis in Figure 2 to use the corresponding trading days rather than the observations index as in Figure 1. When argument `type = "interval"` is set,



**Figure 2:** Stochastic Volatility Fan Chart for the Pound-Dollar Exchange Rate Data with Prediction Intervals.

the `probs` argument corresponds to prediction intervals. Consequently, the default interval fan chart comprises of three different shades, running from the darkest for the 50th prediction interval to the lightest for the 95th prediction interval. Contour lines are controlled by the `ln` argument, which is set to `NULL` by default for `fan0`. In Figure 2, changes in the default of `ln` overlays lines on the fan chart for the upper and lower bounds of the 50th, 80th and 95th prediction intervals. A further line is plotted along the median of  $\theta_t$  controlled by the `med.ln` argument and shown when `type = "interval"`. Labels on the right hand side are by default added to correspond to the upper and lower bounds of each plotted line. The text size of the labels are controlled by the `rcex` argument. This is set to 0.6 to incorporate the labels without extending the limits of the x-axis. The left labels are added by setting `llab = TRUE` and take the same text properties as the right labels. Users can customize many properties of the fan chart shading, labels contour lines, labels and axis through a range of arguments.

Spaghetti plots are a method of viewing data to visualize possible values through a systems. They are commonly used on geographical data, such as meteorological forecasts (Sanyal et al., 2010) to show possible or realised paths, or over time, such in longitudinal data analysis (Hedeker and Gibbons, 2006). Spaghetti plots can also be used represent uncertainty shown by a range of possible future trajectories or past estimates. For example, using the `th.mcmc2` object Figure 3 displays 20 random sets of  $\theta_t$  simulations plotted by setting the argument `style = "spaghetti"`;

```
# transparent fan with visible lines
fan0(th.mcmc2, ln = c(5, 20, 50, 80, 95), alpha = 0, ln.col = "darkorange", llab = TRUE)

# spaghetti lines
fan(th.mcmc2, style = "spaghetti", n.spag = 20, alpha = 0.3)
```

The initial fan chart is completely transparent from setting the transparency argument `alpha = 0`. In order for the percentile lines to be visible a non-transparent colour is used for the `ln.col` argument. Lines are plotted according the user defined `ln` argument to provide underlying uncertainty measures for the posterior probability distribution. The spaghetti lines, which are semi-transparent, are based on a random selection of simulations. They are superimposed on a fan chart using the `fan` function, which operates in much the similar way as `fan0`. The most important difference between the two is in the default setting of the `add` argument, which controls whether to create a new plot window for a fan chart or add it to an existing device. For the `fan` function, `add` is set to `TRUE` and hence its is more appropriately used to add a fan chart to an existing plotting device. The `fan` function also adds lines and labels on select contours by default as illustrated in the next section.

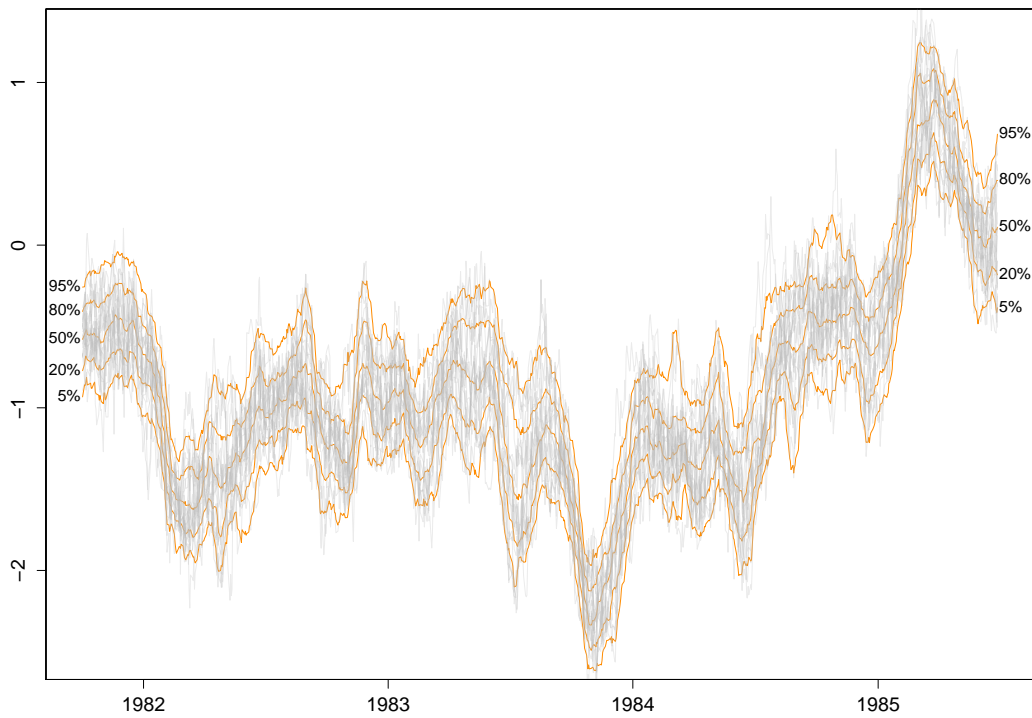


Figure 3: Stochastic Volatility Spaghetti Plot for the Pound-Dollar Exchange Rate Data.

### Bank of England fan charts

The Monetary Policy Committee (MPC) of the Bank of England produces fan charts of forecasts for Consumer Price Index (CPI) of inflation and Gross Domestic Product in their quarterly Inflation Reports. Alongside the fan charts, the Bank of England provides data, in the form of central location, uncertainty and skewness parameters of a split-normal distribution that underlie their fan charts.<sup>2</sup> The probability density of the split-normal distribution is given by [Julio \(2007\)](#)<sup>3</sup> as,

$$f(x; \mu, \sigma_1, \sigma_2) = \begin{cases} \frac{\sqrt{2}}{\sqrt{\pi(\sigma_1 + \sigma_2)}} e^{-\frac{1}{2\sigma_1^2}(x-\mu)^2} & \text{for } -\infty < x \leq \mu \\ \frac{\sqrt{2}}{\sqrt{\pi(\sigma_1 + \sigma_2)}} e^{-\frac{1}{2\sigma_2^2}(x-\mu)^2} & \text{for } \mu < x < \infty \end{cases}, \quad (3)$$

where  $\mu$  represents the mode, and the two standard deviations  $\sigma_1$  and  $\sigma_2$  can be derived given the overall uncertainty parameter,  $\sigma$  and skewness parameters,  $\gamma$ , as;

$$\sigma^2 = \sigma_1^2(1 + \gamma) = \sigma_2^2(1 - \gamma). \quad (4)$$

Functions for the probability density, cumulative distribution, quantiles and random generation for the split-normal distribution can be found in the **fanplot** package.

The `boe` data frame provides historical details on the forecasts of the MPC for CPI inflation between Q1 2004 to Q4 2013.

```
> head(boe)
  time0  time mode uncertainty skew
1 2004 2004.00 1.34      0.2249  0
2 2004 2004.25 1.60      0.3149  0
3 2004 2004.50 1.60      0.3824  0
4 2004 2004.75 1.71      0.4274  0
5 2004 2005.00 1.77      0.4499  0
6 2004 2005.25 1.68      0.4761  0
```

The first column `time0` refers to the base year of forecast, the second, `time` indexes future projections, whilst the remaining three columns provide values for the corresponding projected central location ( $\mu$ ), uncertainty ( $\sigma$ ) and skew ( $\gamma$ ) parameters:

<sup>2</sup>The Bank of England predominately refer to the equivalent, re-parametrised, two-piece normal distribution.  
<sup>3</sup>[Wallis \(2014\)](#) notes the two-piece normal distribution was first introduced by [Fechner \(1897\)](#).

Bank of England style fan charts vary from quarter to quarter but follow a similar theme throughout, which can be replicated in R using the **fanplot** package. The input data given to a fan function to plot a fan chart differs from the simulations of the previous section. Rather than many simulations from distributions in each time period we can pass a “matrix” object of time ordered values from the split-normal quantile function.<sup>4</sup> As is the case for passing simulated values to the fan function, rows of the data object represent a set of user defined probabilities and columns represent a set of time points. For example, in the code below, a subset of the Bank of England future parameters of CPI published in Q1 2013 are first selected. Then a vector of probabilities related to the percentiles, that we ultimately would like to plot different shaded fans for, are created. Finally, in a for loop, the `qsplnorm` function calculates the values for which the time-specific (i) split-normal distribution will be less than or equal to the probabilities of p.

```
# select relevant data
y0 <- 2013
boe0 <- subset(boe, time0 == y0)
k <- nrow(boe0)

# guess work to set percentiles the boe are plotting
p <- seq(0.05, 0.95, 0.05)
p <- c(0.01, p, 0.99)

# quantiles of split-normal distribution for each probability (row) at each future
# time point (column)
cpival <- matrix(NA, nrow = length(p), ncol = k)
for (i in 1:k)
  cpival[, i] <- qsplnorm(p, mode = boe0$mode[i],
                        sd = boe0$uncertainty[i],
                        skew = boe0$skew[i])
```

The new object `cpival` contains values evaluated from the `qsplnorm` function in 6 rows, for our selected probabilities used in the calculation `p`, and 13 columns for successive time periods for which the MPC provide future parameters.

The `cpival` object can be used to add a fan chart to an active R graphic device. In the code below, the area of Figure 4 is set up when plotting the past CPI data, contained in the time series object `cpi`. The `xlim` arguments are set to ensure space on the right hand side of the plotting area for the fan. Following as closely as possible the Bank of England style for plotting fan charts for Q1 2013<sup>5</sup>, the plotting area is set to near square, the background for future values is a gray colour, y-axis are plotted on the right hand side, a horizontal line are added for the CPI target and a vertical line for the two-year ahead point.

```
# past data
plot(cpi, type = "l", col = "tomato", lwd = 2,
     xlim = c(y0 - 5, y0 + 3), ylim = c(-2, 7),
     xaxt = "n", yaxt = "n", ylab = "")

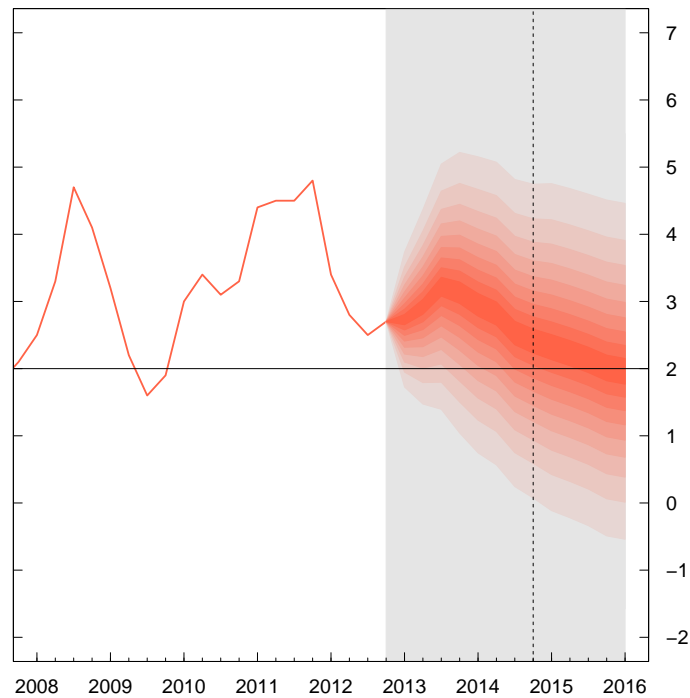
# background shading during forecast period
rect(y0 - 0.25, par("usr")[3] - 1, y0 + 3, par("usr")[4] + 1,
     border = "gray90", col = "gray90")

# add fan
fan(data = cpival, data.type = "values", probs = p,
    start = y0, frequency = 4, anchor = cpi[time(cpi) == y0 - 0.25],
    fan.col = colorRampPalette(c("tomato", "gray90")), ln = NULL, rlab = NULL)

# boe aesthetics
axis(2, at = -2:7, las = 2, tcl = 0.5, labels = FALSE)
axis(4, at = -2:7, las = 2, tcl = 0.5)
axis(1, at = 2008:2016, tcl = 0.5)
axis(1, at = seq(2008, 2016, 0.25), labels = FALSE, tcl = 0.2)
abline(h = 2) # boe cpi target
abline(v = y0 + 1.75, lty = 2) # 2 year line
```

<sup>4</sup>Note, values from any distributions quantile function can be used.

<sup>5</sup>The CPI fan chart for Q1 2013 can be viewed in the February 2013 Inflation Report (Bank of England, 2013), Chart 5.3.



**Figure 4:** Fan chart, in the Bank of England style, for the MPC Q1 2013 forecast of the percentage increase in prices on year earlier.

The fan chart itself is outputted from the `fan` function, where arguments are set to ensure a close resemblance of the R plot to that produced by the Bank of England. The first three arguments in the `fan` function called in the above code, provide the `cpival` data to be plotted, indicate that the data are a set of calculated values (as opposed to simulations as in the previous examples) and provide the probabilities that correspond to each row of `cpival` object. The next two arguments define the start time and frequency of the data. These operate in a similar fashion to those used when defining time series in R with the `ts` function. The `anchor` argument is set to the value of CPI before the start of the fan chart. This allows a join between the value of the Q1 2013 observation and the fan chart. The `fan.col` argument is set to a colour palette for shades between tomato and gray90. The final two arguments are set to NULL to suppress the plotting of contour lines at the boundary of each shaded fan and their labels, as per the Bank of England style.

An alternative plot in Figure 5 is based on a regular time series object of simulated data and some other style settings in the `fan` function. These produce a fan chart with a greater array of coloured fans with labels and contour lines alongside selected percentiles of the future distribution. The input data is based upon 10,000 simulated values produced using the `rsplitnorm` function and the future split-normal distribution parameters from Q1 2013 in the truncated `boe0` data frame;

```
# simulate future values
cpisim <- matrix(NA, nrow = 10000, ncol = k)
for (i in 1:k)
  cpisim[, i] <- rsplitnorm(n = 10000, mode = boe0$mode[i],
                           sd = boe0$uncertainty[i],
                           skew = boe0$skew[i])
```

The fan chart based on the simulations in `cpisim` are then be added to the truncated CPI data plot;

```
# truncate cpi series and plot
cpi0 <- ts(cpi[time(cpi) < 2013], start = start(cpi), frequency = frequency(cpi))
plot(cpi0, type = "l", lwd = 2, las = 1, ylab = "",
      xlim = c(y0 - 5, y0 + 3.5), ylim = c(-2, 7))
```

```
# add fan
library("RColorBrewer")
fan(data = cpisim, type = "interval", probs = seq(0.01, 0.99, 0.01),
     start = y0, frequency = 4, ln = c(50, 80, 95), med.ln = FALSE,
     fan.col = colorRampPalette(colors = rev(brewer.pal(9, "Oranges"))))
```

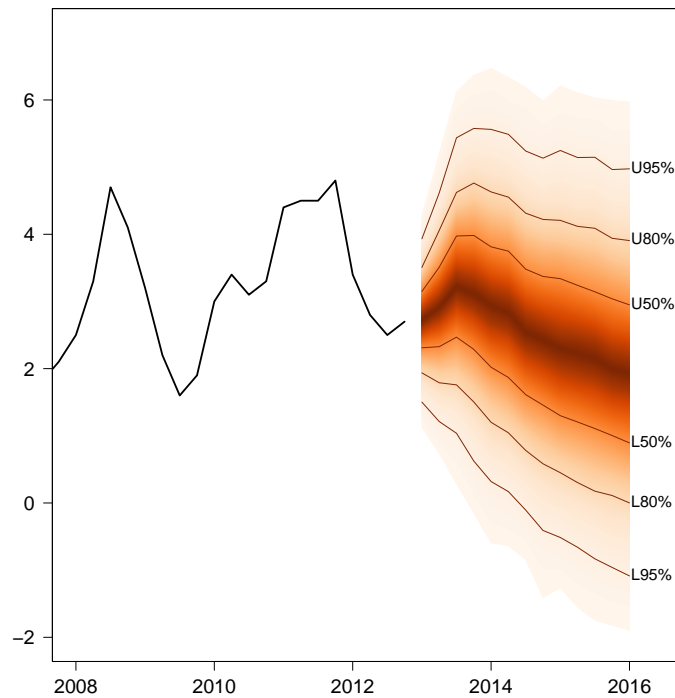


Figure 5: Alternative fan chart for the MPC Q1 2013 CPI forecast.

This code shows how users can control multiple visual elements of the fan chart, not previously illustrated. In Figure 5 the fan is based on 50 shadings for 100 equally spaced percentiles of the future distributions specified through the `probs` argument. The colour scheme is based on the Oranges palette from the `RColorBrewer` package (Neuwirth, 2014). Contour lines for the upper and lower intervals of the 50th, 80th and 95th prediction intervals are imposed using the `ln` argument. The median line, plotted by default when `type = "interval"` is set, is removed using the `med.ln` argument.

Box plots (Tukey, 1977) are commonly used as a simple descriptive statistics to visualise data through their quartiles. The `boxplot` function in R has many options including the display of multiple box plots based on sequential distributions. However, when data is based on a time series with multiple observations during a unit of time, such as quarterly data, fixing the location of the plot on the x-axis can be cumbersome. The fan function overcomes this problem when setting `style = "boxplot"`. In Figure 6 the simulated future CPI data `cpisim` are passed to the `data` argument:

```
# plot past data
plot(cpi0, type = "l", xlim = c(y0-5, y0+3), ylim = c(-2, 7), lwd = 2)

# box plots
fan(cpisim, style = "boxplot", start = y0, frequency = 4, outline = FALSE)
```

The fan function allows users to easily locate sequential distributions of box plots on the x-axis using the `start` and `frequency` arguments. Additional arguments in the fan function are passed to `boxplot`. For example from the code above, outliers are suppressed by setting `outline = FALSE`.

## Summary

The `fanplot` package allows users to easily visualise uncertainty based on either simulations from sequential distributions or values based on pre-calculated quartiles of distribution. Interactive visualisations of fan charts, as demonstrated in the `net_elicit.R` demo using the `shiny` package (RStudio and Inc., 2014), could potentially allow for an intuitive elicitation of experts forecasts<sup>6</sup>. Data of various classes can be incorporated including regular time series ("`mts`"), irregular time series ("`zoo`") and simulations from, say, MCMC via "`matrix`", "`data.frame`" or "`mcmc`" type objects. The `fanplot` package also has a range of options to adjust colour shadings of fan charts, their lines and labels, and specify whether to display prediction intervals or percentiles of distributions.<sup>7</sup>

<sup>6</sup>Run via `demo(net_elicit, package = "fanplot", ask = FALSE)`.

<sup>7</sup>An illustration of many of these options are provided in a demo file, run via `demo(sv_fan, package = "fanplot", ask = FALSE)`.



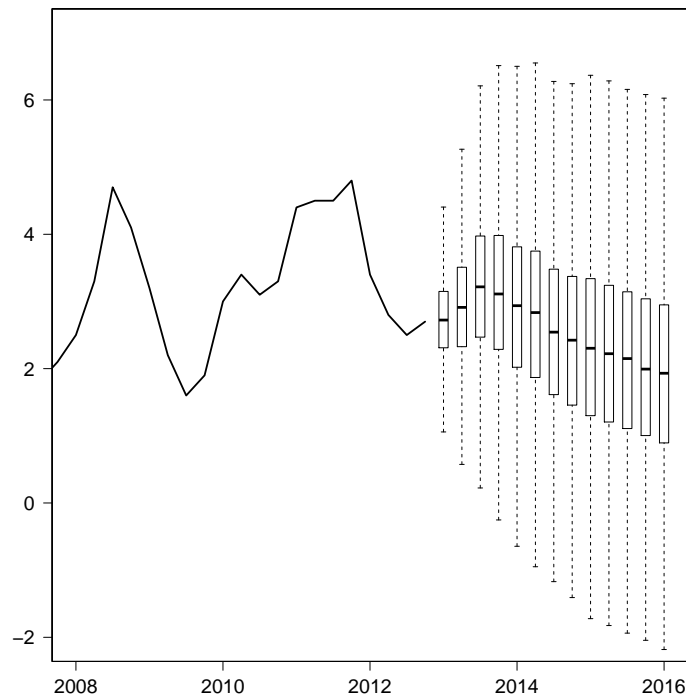


Figure 6: Box plots for the MPC Q1 2013 CPI forecast.

## Bibliography

- G. J. Abel, J. Bijak, J. J. Forster, J. Raymer, P. W. F. Smith, and J. S. Wong. Integrating uncertainty in time series population forecasts: An illustration using a simple projection model. *Demographic Research*, 29:1187–1226, Dec. 2013. [p16]
- Bank of England. Inflation report February 2013. Technical report, Bank of England, London, United Kingdom, 2013. URL <http://www.bankofengland.co.uk/publications/Documents/inflationreport/2013/ir13feb.pdf>. [p19]
- E. Britton, P. Fisher, and J. Whitley. The Inflation Report projections: understanding the fan chart. *Bank of England Quarterly Bulletin*, 38:30–37, 1998. [p15]
- M. Buchmann. MATLAB file exchange: Fan chart, 2010. URL <http://www.mathworks.de/matlabcentral/fileexchange/27702-fan-chart>. [p15]
- G. T. Fechner. *Kollektivmasslehre*. Leipzig, Germany: Wilhelm Engelmann, 1897. [p18]
- P. Gerland, A. E. Raftery, H. Sevčiková, N. Li, D. Gu, T. Spoorenberg, L. Alkema, B. K. Fosdick, J. Chunn, N. Lalic, G. Bay, T. Buettner, G. K. Heilig, and J. Wilmoth. World population stabilization unlikely this century. *Science*, 387(6635):803–805, Sept. 2014. [p15]
- A. C. Harvey. *Forecasting, Structural Time Series Models and the Kalman Filter*. Cambridge University Press, Cambridge, United Kingdom, 1990. [p15]
- D. Hedeker and R. D. Gibbons. *Longitudinal Data Analysis*. John Wiley & Sons, Hoboken, NJ, USA, 2006. [p17]
- R. J. Hyndman and Y. Khandakar. Automatic time series forecasting: The forecast package for R. *Journal of Statistical Software*, 27:1–22, 2008. [p15]
- J. M. Julio. The fan chart: The technical details of the new implementation. 2007. URL <http://www.banrep.gov.co/docum/ftp/borra468.pdf>. [p15, 18]
- J. M. Julio. The HPD fan chart with data revision. 2009. URL <http://mpra.ub.uni-muenchen.de/57714/>. [p15]
- B. B. McShane and A. J. Wyner. A statistical analysis of multiple temperature proxies: Are reconstructions of surface temperatures over the last 1000 years reliable? *The Annals of Applied Statistics*, 5(1): 5–44, Mar. 2011. [p15]

- R. Meyer and J. Yu. BUGS for a Bayesian analysis of stochastic volatility models. *Econometrics Journal*, 3(2):198–215, Dec. 2002. [p15, 16]
- E. Neuwirth. *RColorBrewer: ColorBrewer Palettes*, 2014. URL <http://CRAN.R-project.org/package=RColorBrewer>. [p21]
- B. Pfaff. VAR, SVAR and SVEC models: Implementation within R package vars. *Journal of Statistical Software*, 27(4), 2008. [p15]
- RStudio and Inc. *shiny: Web Application Framework for R*, 2014. URL <http://CRAN.R-project.org/package=shiny>. [p21]
- J. Sanyal, S. Zhang, J. Dyer, A. Mercer, P. Amburn, and R. J. Moorhead. Noodles: a tool for visualization of numerical weather model ensemble uncertainty. *IEEE Transactions on Visualization and Computer Graphics*, 16(6):1421–1430, 2010. [p17]
- D. Spiegelhalter, M. Pearson, and I. Short. Visualizing uncertainty about the future. *Science*, 333(6048): 1393–1400, Sept. 2011. [p15]
- S. Sturtz, U. Ligges, and A. Gelman. R2WinBUGS: a package for running WinBUGS from R. *Journal of Statistical Software*, 12(3):1–16, 2005. [p16]
- J. W. Tukey. *Exploratory Data Analysis*. Addison-Wesley Publishing Company, Reading, MA, USA, 1 edition, 1977. [p21]
- K. F. Wallis. The two-piece normal, binormal, or double Gaussian distribution: Its origin and rediscoveries. *Statistical Science*, 29(1):106–112, Feb. 2014. [p18]
- A. Zeileis and G. Grothendieck. zoo: S3 infrastructure for regular and irregular time series. *Journal of Statistical Software*, 30(6):1–27, 2005. [p16]

Guy J. Abel  
Wittgenstein Centre (IIASA, VID/ÖAW, WU)  
Vienna Institute of Demography/Austrian Academy of Sciences  
Wohllebengasse 12-14, 6th floor,  
1040 Vienna, Austria  
[guy.abel@oeaw.ac.at](mailto:guy.abel@oeaw.ac.at)