Template(?) for a `pairs()` option in `ggplot2`

...also showing a good choice for `knit_theme` and \\`lslset` definitions

Henrique Ap. Laureano

henrique.laureano@kaust.edu.sa ∧ http://mynameislaure.github.io/

/KAUST/CEMSE/STAT

Summer Semester
2018

Contents

Code ∧ graph

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

```r
# <r code> -----
# other option for the function pairs()
library(brinla) ; data(usair, package = "brinla")
library(ggplot2); library(GGally)

ggpairs(usair
        , lower = list(continuous = "cor")
        , upper = list(continuous = "points")
        , axisLabels = "none") +
  theme_bw()
# </r code> -----
```

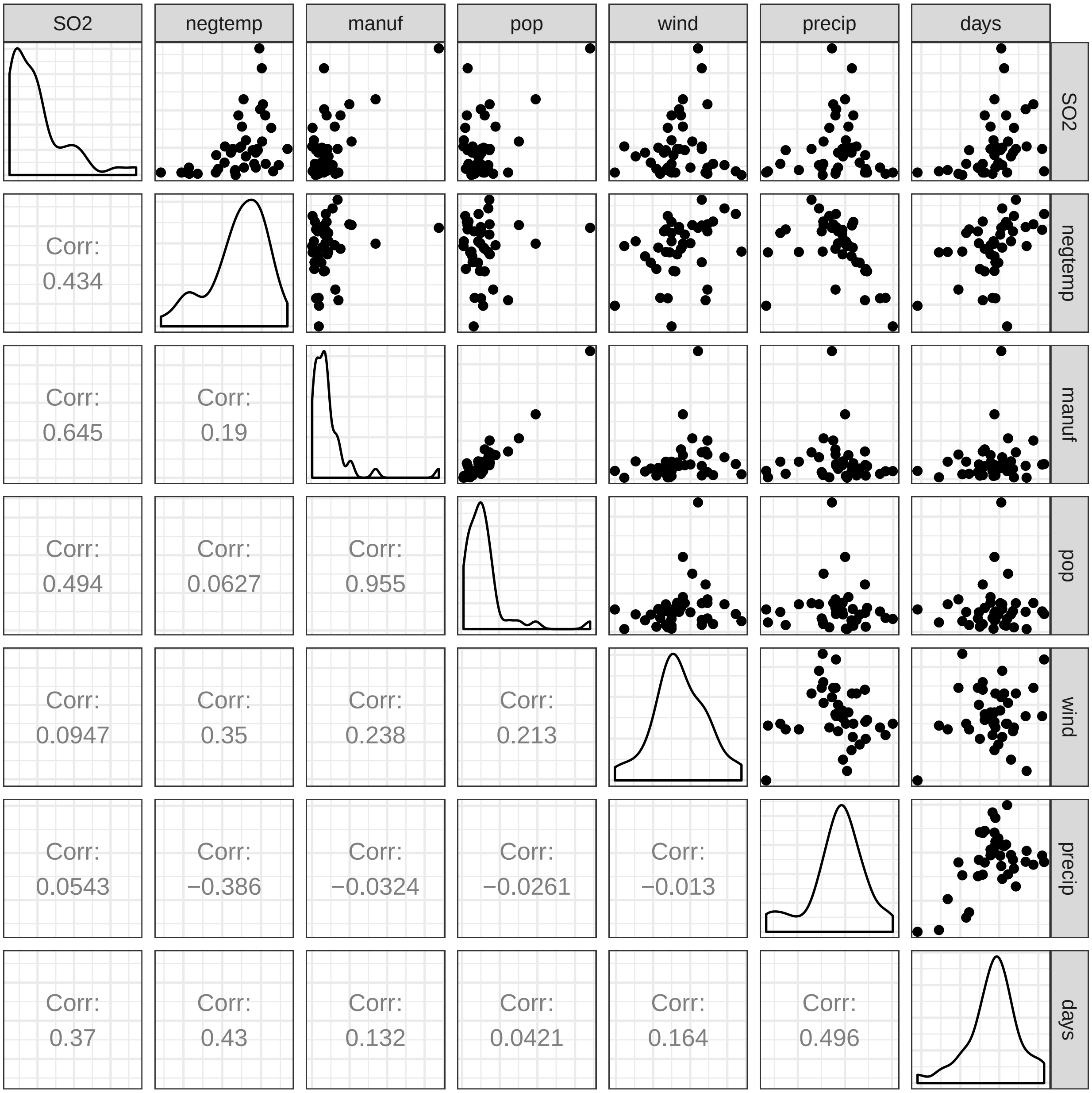

Figure 1: Descriptive analysis of the quantitative features. upper triangular matrix: scatterplots; lower triangular matrix: correlations.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

```r
# <r code> -----
library(INLA)
# simulating data, 100 points
n <- 100 ; x <- seq(0, 1, length.out = n)

f.true <- (sin(2*pi*x**3))**3 ; y <- f.true + rnorm(n, sd = .2)

data.inla <- list(y = y, x = x)

# fitting the random walk 1 model for smoothing splines,
# without the intercept
formula <- y ~ -1 + f(x, model = "rw1", constr = FALSE)
result <- inla(formula, data = data.inla)

f.hat <- result$summary.random$x$mean # posterior mean
f.lb <- result$summary.random$x$'0.025quant' # 2.5% percentile
f.ub <- result$summary.random$x$'0.975quant' # 97.5% percentile

data.plot <- data.frame(y = y, x = x,
                        f.true = f.true, f.hat = f.hat,
                        f.lb = f.lb, f.ub = f.ub)
ggplot(data.plot, aes(x = x, y = y)) +
  geom_line(aes(y = f.hat), col = "#0080ff", size = .75) +
  geom_line(aes(y = f.true), linetype = 2, size = .75) +
  geom_ribbon(aes(ymin = f.lb, ymax = f.ub),
              alpha = .25, fill = "orange", col = "#0080ff") +
  geom_point(aes(y = y)) +
  theme_minimal() +
  labs(title = "Random walk 1 model for smoothing splines")
# </r code> -----
```

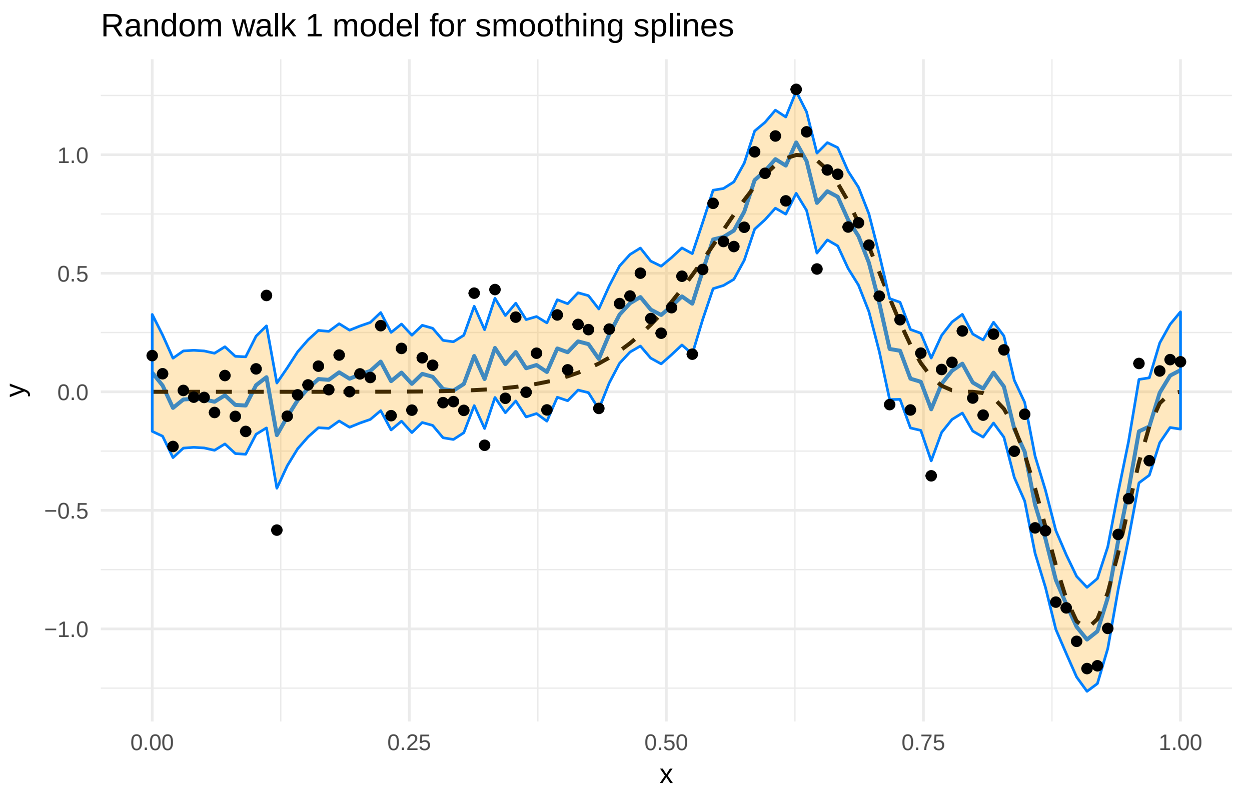

Figure 2: Posterior mean and 95% credible interval of a random walk 1 model for smoothing splines without intercept.

lstlisting

```
# <r code> -----
library(knitr)

tema <- knit_theme$get("clarity") # acid

knit_theme$set(tema)

opts_chunk$set(size='small'
               , cache=TRUE
               , cache.path='cache/'
               , comment=NA
               , warning=FALSE
               , message=FALSE
               , fig.align='center'
               , dpi=100
               , fig.path='iBagens/'
               , fig.pos='H'
               , results='hold'
               , fig.show='hold')
# </r code> -----
# <r code> -----
# other option for the function pairs()
library(brinla) ; data(usair, package = "brinla")

library(ggplot2); library(GGally)

ggpairs(usair
        , lower = list(continuous = "cor")
        , upper = list(continuous = "points")
        , axisLabels = "none") +
  theme_bw()
# </r code> -----
# <r code> -----
library(INLA)
# simulating data, 100 points
n <- 100 ; x <- seq(0, 1, length.out = n)

f.true <- (sin(2*pi*x**3))**3 ; y <- f.true + rnorm(n, sd = .2)

data.inla <- list(y = y, x = x)

# fitting the random walk 1 model for smoothing splines,
# without the intercept
formula <- y ~ -1 + f(x, model = "rw1", constr = FALSE)
result <- inla(formula, data = data.inla)

f.hat <- result$summary.random$x$mean # posterior mean
f.lb <- result$summary.random$x$'0.025quant' # 2.5% percentile
f.ub <- result$summary.random$x$'0.975quant' # 97.5% percentile

data.plot <- data.frame(y = y, x = x,
                        f.true = f.true, f.hat = f.hat,
                        f.lb = f.lb, f.ub = f.ub)
ggplot(data.plot, aes(x = x, y = y)) +
  geom_line(aes(y = f.hat), col = "#0080ff", size = .75) +
  geom_line(aes(y = f.true), linetype = 2, size = .75) +
  geom_ribbon(aes(ymin = f.lb, ymax = f.ub),
              alpha = .25, fill = "orange", col = "#0080ff") +
  geom_point(aes(y = y)) +
  theme_minimal() +
  labs(title = "Random walk 1 model for smoothing splines")
# </r code> -----
## # <r code> -----
## # extracting R code to insert after with \lstinputlisting{}
## purl("~/Dropbox/stuff.Rnw", documentation = 0)
## # kl-17766:~ laureaha$ mv stuff.R ~/Dropbox/
## # </r code> -----
```