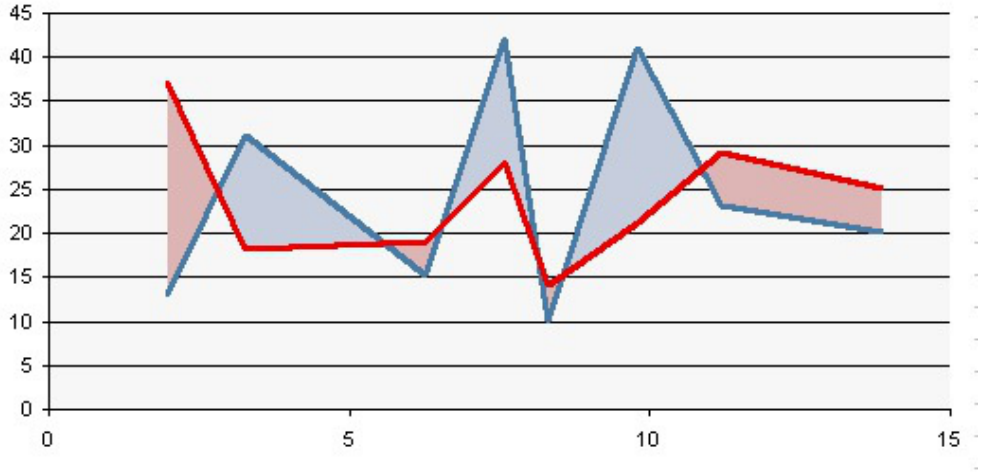
Finding my way around R

ggplot2: Two Color XY-Area Combo Chart

OCTOBER 22, 2009

tags: chart, excel, ggplot2, plot

David@Work blog (http://davidmerlemontgomery.blogspot.com/2009/09/two-color-xy-area-combo-chart.html) shows how to fill in the area between two crossing lines in an Excel chart. This post was also published as a guest-post on **PTS blog (http://peltiertech.com/WordPress/two-color-xy-area-combo-chart-guest-post/)**.



Let's try to replicate this graph in **ggplot2 (http://had.co.nz/ggplot2/)**.

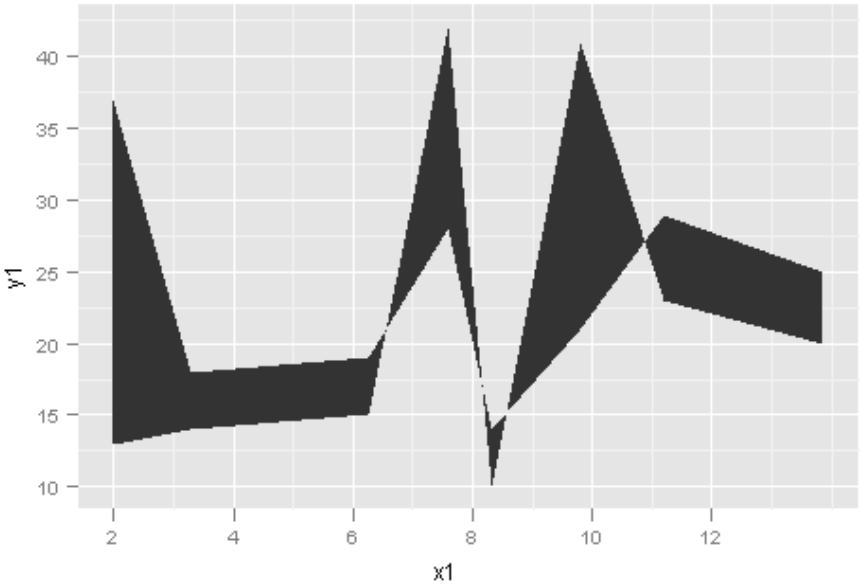
First, load `ggplot2` and generate the data frame to be used in the example (I am using a slightly modified dataset, therefore the final result will differ somewhat from the original graph).

```
> library(ggplot2)
```

```
> cross <- data.frame(x1 = c(2, 3.27, 6.26, 7.58,
      8.33, 9.79, 11.2, 13.86), y1 = c(13, 14,
      15, 42, 10, 41, 23, 20), y2 = c(37, 18,
      19, 28, 14, 21, 29, 25))
```

Filling just the area between the two lines is accomplished easily in `ggplot2`, however as we would need the segments to be of different colour then some extra work is required.

```
> ggplot(cross, aes(x1, ymin = y1, ymax = y2)) +
      geom_ribbon()
```

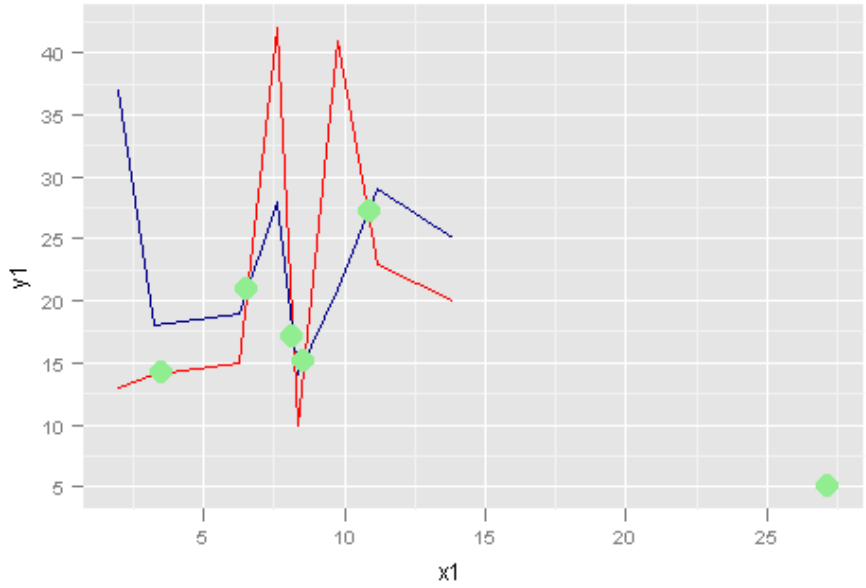


In order to change the fill colour at each point where two lines cross, the points of intersection need to be calculated.

```
> cross$slope1 <- c(NA, with(cross, diff(y1)/diff(x1)))
> cross$slope2 <- c(NA, with(cross, diff(y2)/diff(x1)))
> cross$intcpt1 <- with(cross, y1 - slope1 *
     x1)
> cross$intcpt2 <- with(cross, y2 - slope2 *
     x1)
> cross$x2 <- with(cross, (intcpt1 - intcpt2)/(slope2 -
     slope1))
> cross$y3 <- with(cross, slope1 * x2 + intcpt1)
> cross <- cross[, c(-4:-7)]
```

Now, just as an extra precaution and to make sure that calculations are correct, we check visually the location of the points of intersection:

```
> ggplot(cross) + geom_line(aes(x1, y1), colour = "red") +
     geom_line(aes(x1, y2), colour = "darkblue") +
     geom_point(aes(x2, y3), colour = "lightgreen",
         size = 4)
```



As I am planning to colour the plot above generated using `geom_ribbon` the points of intersection need also to be presented in the form expected by `geom_ribbon` (x, ymin, ymax) – a simple copy of `y3` accomplishes this.

```
> cross$y4 <- cross$y3
```

Additional error-checking is also obviously needed, as is indicated by the position of the left and rightmost green dots on the above graph – any two lines can have a point of intersection which falls outside the limits of the particular plot.

```
> cross[which(cross$x2 > cross$x1), c("x2", "y3",
     "y4")] <- NA
> cross$segment <- findInterval(cross$x1, c(cross$x2[which(!is.na(cross$x2))]))
```

For `ggplot2` to be able to vary the fill colour at each crossing of the lines it needs to know the start and end point of each coloured area. This means that the middle points of intersection need to be duplicated, as they would be part of two adjacent areas filled with different colours.

```
> cross$x3 <- c(tail(cross$x2, -1), NA)
> cross$y5 <- c(tail(cross$y3, -1), NA)
> cross$y6 <- cross$y5
```

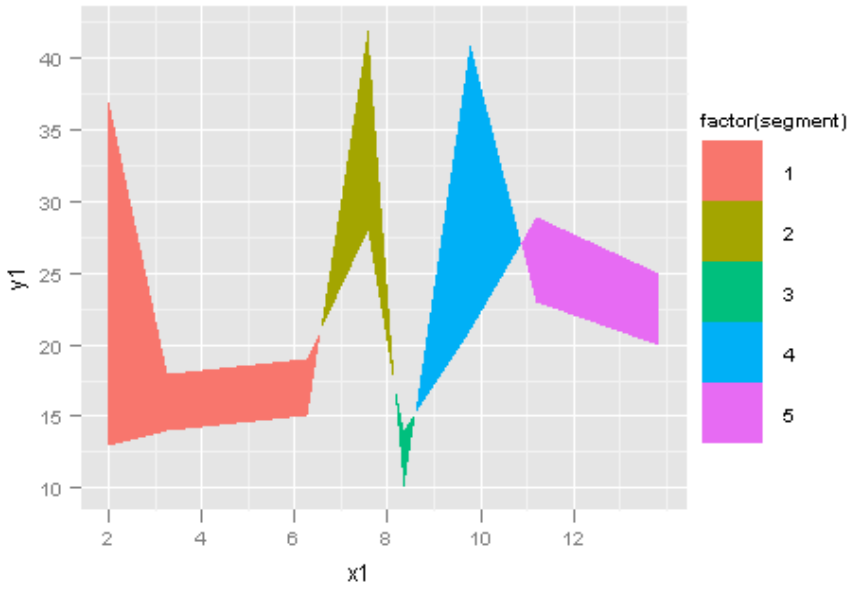
Now the coordinates of two lines and the start/end points of coloured areas need to be combined into one dataframe in a long format.

```
> cross1 <- cross[, c(1:3, 7)]
> cross2 <- cross[!is.na(cross$x2), c(4:6, 7)]
> cross3 <- cross[!is.na(cross$x3), c(8:10, 7)]
```

```
> names(cross2) <- names(cross1)
> names(cross3) <- names(cross1)
```

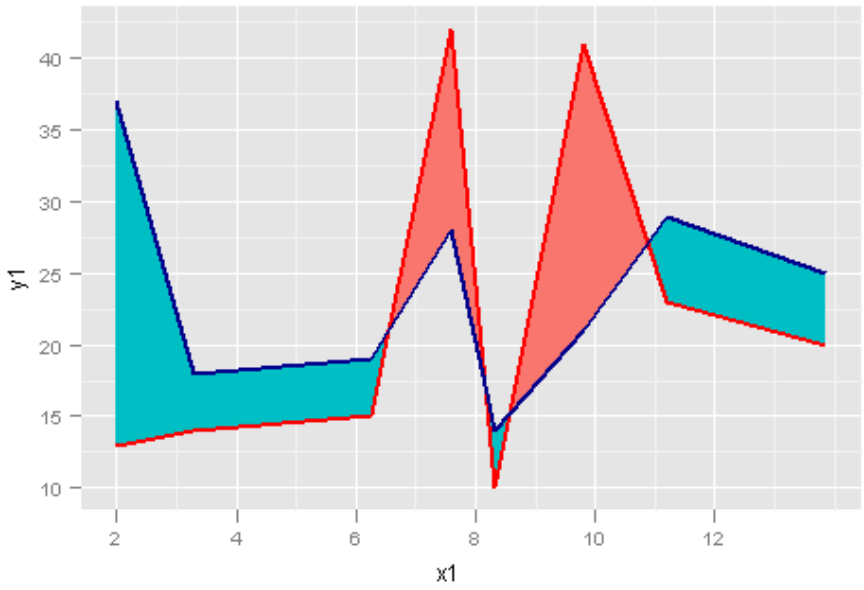
```
> combo <- rbind(cross1, cross2)
> combo <- rbind(combo, cross3)
> combo <- combo[is.finite(combo$y1), ]
> combo <- combo[order(combo$x1), ]
```

```
> ggplot(combo, aes(x1, ymin = y1, ymax = y2)) +
    geom_ribbon(aes(fill = factor(segment)))
```



Each segment is filled with a different colour, but we want to limit the number of fill colours to two.

```
> ggplot(combo, aes(x1, ymin = y1, ymax = y2,
    )) + geom_ribbon(aes(fill = factor(segment%%2))) +
    geom_path(aes(y = y1), colour = "red",
        size = 1) + geom_path(aes(y = y2),
    colour = "darkblue", size = 1) + opts(legend.position = "none")
```



from → R

page: 2

ggplot2: Overplotting In a Faceted Scatterplot

DECEMBER 3, 2009

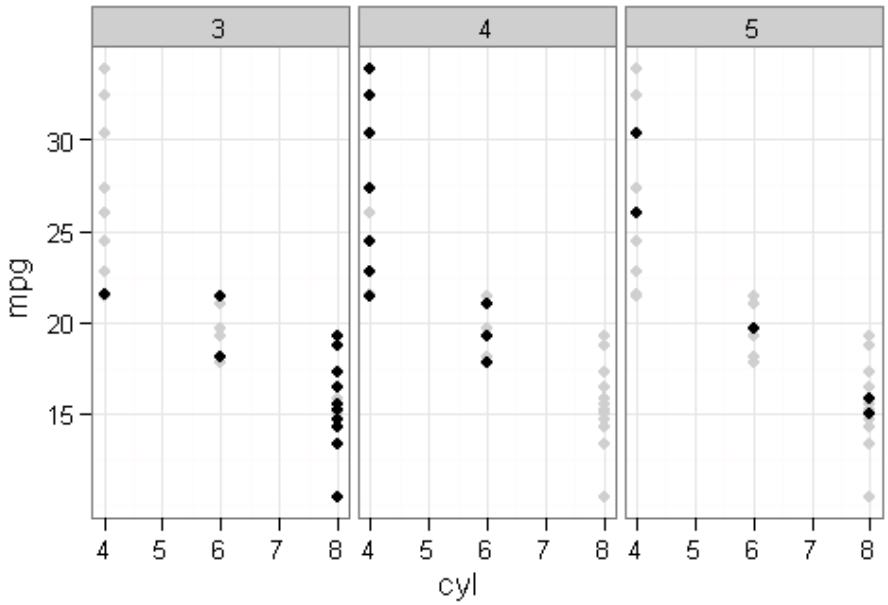
tags: facet, ggplot2, plot, R, scatterplot

Hadley Wickham recently **shared (http://groups.google.com/group/ggplot2/msg/4c0763732ee5f4e1)** a nice tip on how to get a faceted scatterplot plot with all points in the background of each plot.

This technique makes a clever use of setting the faceting variable to NULL so that all points are plotted in light grey in all the facets.

```
> library(ggplot2)
```

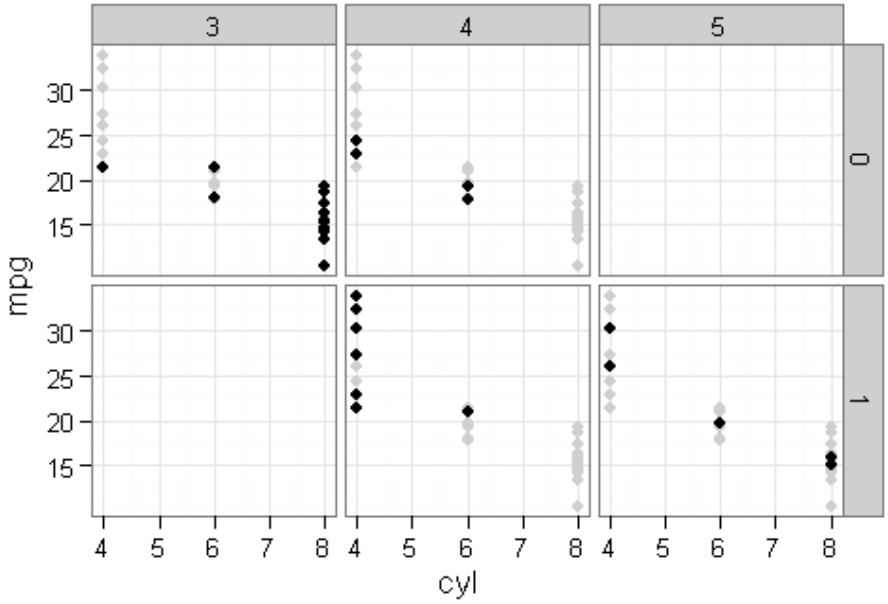
```
> ggplot(mtcars, aes(cyl, mpg)) + geom_point(data = transform(mtcars,
    gear = NULL), colour = "grey80") + geom_point() +
    facet_grid(~gear) + theme_bw()
```



Update 17 May 2010

bch asked in the comments below, how to achieve the same when there are two facets. The method is the same, now one would need to exclude both of the facetting variables from the dataset used to draw the light grey points.

```
> ggplot(mtcars, aes(cyl, mpg)) + geom_point(data = mtcars[,
     !names(mtcars) %in% c("am", "gear")],
     colour = "grey80") + geom_point() + facet_grid(am ~
     gear) + theme_bw()
```



from → R

6 Comments leave one →

1. **squid PERMALINK**
   November 12, 2009 7:10 pm
   I'm trying to modify your graph to color between lines but only when y1 > y2. However, for my dataset I

seem not to be able to get past this point:

cross$segment summary(cross$x1)
Min. 1st Qu. Median Mean 3rd Qu. Max.
1 15390000 30780000 30770000 46160000 61540000

> summary(cross$y1)
Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
0.0001211 0.0057750 0.0078010 0.0079960 0.0098790 0.0488000 9.0000000

> summary(cross$y2)
Min. 1st Qu. Median Mean 3rd Qu. Max. NA's
0.000000 0.001056 0.001706 0.002003 0.002607 0.009346 9.000000

I can't figure out what this error is due to. Do you have any insight?
Thank you.
Squid.

REPLY

2. **squid PERMALINK**
   November 12, 2009 7:13 pm
   Sorry, Let me try that again….I didn't post the error I keep getting. It is this..

   > cross$segment <- findInterval(cross$x1, c(cross$x2[which(!is.na(cross$x2))]))

   Error in findInterval(cross$x1, c(cross$x2[which(!is.na(cross$x2))])) :
   'vec' must be sorted non-decreasingly

   This error occurs even when I remove NAs from the original dataset, so that doesn't seem to be the cause.

   Thanks, again.

   REPLY
   - **learnr PERMALINK***
     November 16, 2009 11:28 pm
     You should do what the error message suggests, i.e. sort vector `c(cross$x2[which(!is.na(cross$x2))])` non-decreasingly.

     REPLY
3. **Aniko PERMALINK**
   November 21, 2009 1:37 am
   I had a fealing that there must be a solution that does not involve finding the crossing points almost by hand. The following code does not use ggplot, though it can be surely converted to use it instead of traditional graphics. The main idea is to use a polygon drawing library, and relegate all the difficulties there. Isn't it simpler?

   Aniko

   cross <- data.frame(x1 = c(2, 3.27, 6.26, 7.58,
   8.33, 9.79, 11.2, 13.86), y1 = c(13, 14,
   15, 42, 10, 41, 23, 20), y2 = c(37, 18,
   19, 28, 14, 21, 29, 25))
   attach(cross)

```
library(gpclib)
ymin <- min(y1,y2)

#vertices for area under y1
mat1 <- cbind(c(x1[1], x1, x1[length(x1)]), c(ymin, y1, ymin))
#vertices for area under y2
mat2 <- cbind(c(x1[1], x1, x1[length(x1)]), c(ymin, y2, ymin))

#create polygons from vertices
pp1 <- as(mat1, "gpc.poly")
pp2 <- as(mat2, "gpc.poly")

#plot
plot(x1, y1, col="red", type="n")
plot(setdiff(pp1,pp2), poly.args=list(col="lightbluerder=NA), add=TRUE)
plot(setdiff(pp2,pp1), poly.args=list(col="pinkrder=NA), add=TRUE)
lines(x1, y1, col="blue")
lines(x1, y2, col="red")
```

REPLY

4. **shariaZes PERMALINK**
   November 25, 2009 8:01 pm
   Nice blogpost, amazing looking blog, added it to my favs.

   REPLY

Trackbacks

1. Shading between two lines – ggplot | Matt's Stats n stuff