Displaying Multiple Graphs on the Same Page

Xander Paul

Introduction

Since being introduced to ggplot earlier on in this course, I have been amazed at the ease with which the novice user can approach data visualization. Whereas other coding languages might intimidate the user with their complexity, R's ggplot package provides users with the firepower they need to create robust, elegant visualizations from the get go.

Likewise, other coding languages (I'm looking at you, STATA) might be a top choice for an economist who needs to examine and display data in a bare bones manner but will dissapoint those who desire to communicate their insights in a more vibrant fashion. R, on the other hand, allows users not only to make a broad spectrum of discoveries but also to present data beautifully and in a way that appeals to both technical and non-technical viewers.

In my previous post I explored a more esoteric, if pretty, function of ggplot, tree map. Here, I aim to bring things back to Earth with a more functional feature of ggplot that still enables the user to create dazzling visualizations. Typically, R does not disappoint in merging functionality with design, and if I take anything from this course it'll be that R beats Python in terms of quick, easy visualizations that still manage to please the eye.

In what I consider to be a pure distillation of R's combination of functionality and beauty, I will explore the various opportunities to display multiple graphs on one page. This function of R allows for a simple grouping of different graphs on a singular display, which allows for quick comparisons of data in a variety of different ways. Though this may seem simple, this empowers the user to combine their data displays into a single area, which gives a strong summary of data analysis and can look more professional and/or clean than presenting data visualizations one after another.

Lastly, I must contrast including different visualizations on one page from facetting. While facetting produces similar plots from different factors, it does not meet one's needs for being able to plot different styles of plots in one place. Instead, we can use other methods to enable oursleves to inlude different styles of plot in one display.

Reproducing This Post

In aiming to make this blog post reproducible for the user who is somewhat acquainted with R, I have constructed this post using the built in dataset airquality. This data frame contains information regarding the ozone concentration, solar radiation in Langleys, wind speed, max temperature, month, and day in New York from May to September 1973. Any user can load the airquality data frame into R. Here is a quick peak at the airquality data frame.

```
##   Ozone Solar.R Wind Temp Month Day
## 1    41     190  7.4   67     5   1
## 2    36     118  8.0   72     5   2
## 3    12     149 12.6   74     5   3
## 4    18     313 11.5   62     5   4
## 5    NA      NA 14.3   56     5   5
## 6    28      NA 14.9   66     5   6
```

```
##      Ozone           Solar.R           Wind             Temp      
##  Min.   :  1.00   Min.   :  7.0   Min.   : 1.700   Min.   :56.00  
##  1st Qu.: 18.00   1st Qu.:115.8   1st Qu.: 7.400   1st Qu.:72.00  
##  Median : 31.50   Median :205.0   Median : 9.700   Median :79.00  
##  Mean   : 42.13   Mean   :185.9   Mean   : 9.958   Mean   :77.88  
##  3rd Qu.: 63.25   3rd Qu.:258.8   3rd Qu.:11.500   3rd Qu.:85.00  
##  Max.   :168.00   Max.   :334.0   Max.   :20.700   Max.   :97.00  
##  NA's   :37       NA's   :7                                       
##      Month            Day      
##  Min.   :5.000   Min.   : 1.0  
##  1st Qu.:6.000   1st Qu.: 8.0  
##  Median :7.000   Median :16.0  
##  Mean   :6.993   Mean   :15.8  
##  3rd Qu.:8.000   3rd Qu.:23.0  
##  Max.   :9.000   Max.   :31.0  
## 
```

Next, I add a new column called TrueDays that represents, starting from 1, the number of the day included in the dataset. I added this column because the Day column included in the dataset resets at 1 at each month. Why I did this will become more apparent later on in the post.

```r
# Add a column giving the numerical day
airquality$TrueDays <- 1:length(airquality$Day)
```

To reproduce the content included in this post, you will need to download the ggplot2 and gridExtra packages. As a refresher, ggplot2 allows for customizable visualizations to be created, while gridExtra equips ggplot's grid function with the ability to produce visuals on a single page.

```r
# Install the requisite libraries
library(ggplot2)
library(gridExtra)
```

I should note that I'm working on version 3.4.2 of R. You will not likely have any problems reproducing my images with any other recent versions of R.

Now that we're all set up and ready to proceed, let's move on to the visualizations.

Creating our Graphs

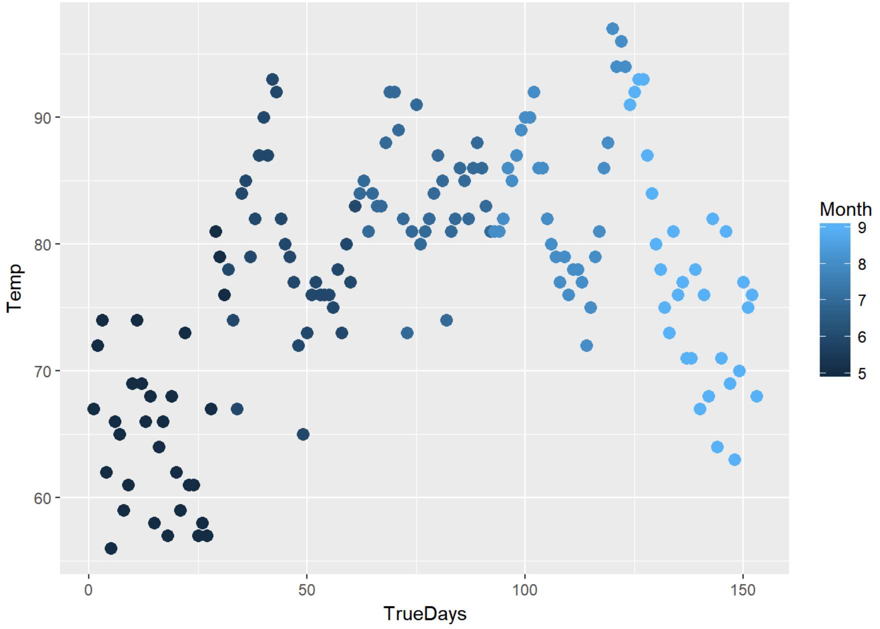
Creating our Graphs

As a way of refreshing the mind and allowing the user to see what individual charts will look like, let's create our individual visualizations as a preliminary step to including them in one page. I'll save each of our individual visualizations as its own variable for easily legible syntax.

Knowing that the average Stat 133 user is likely well versed in making scatterplots, let's plot the temperature and the day for New York in this dataset. Remember that we've created a new column showing the true numerical value for each day in the dataset.

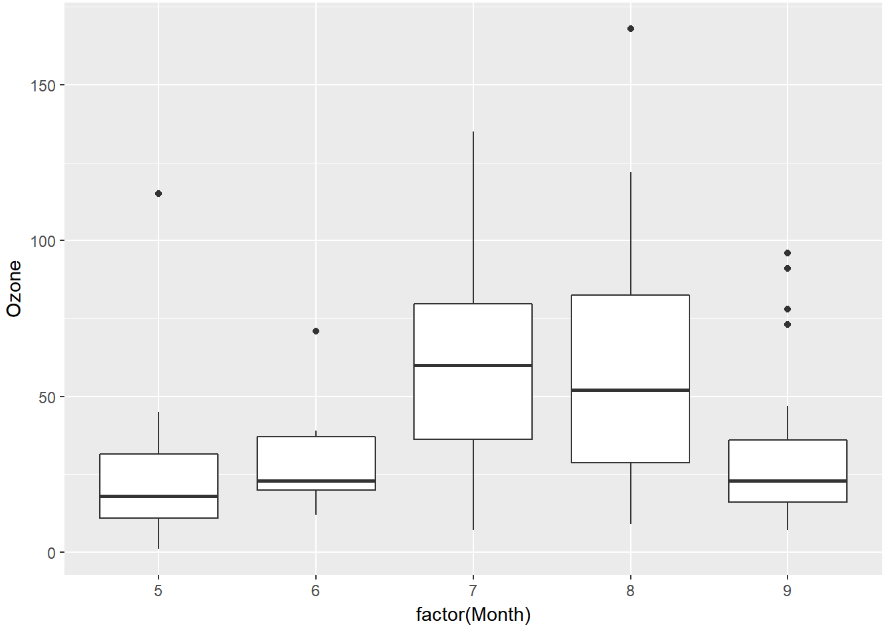
Creating a scatterplot is fairly simple using ggplot and familiar to the reader through this course so I will not elaborate here. Note that the days points are colored based on the month so that the reader can better see the progression of temperature from May to September.

```r
# Create a scatterplot showing the relationship between the day and the temp
temp_scatter <- ggplot(airquality, aes(x=TrueDays, y = Temp, color = Month)) + geom_point(size = 3)
temp_scatter
```



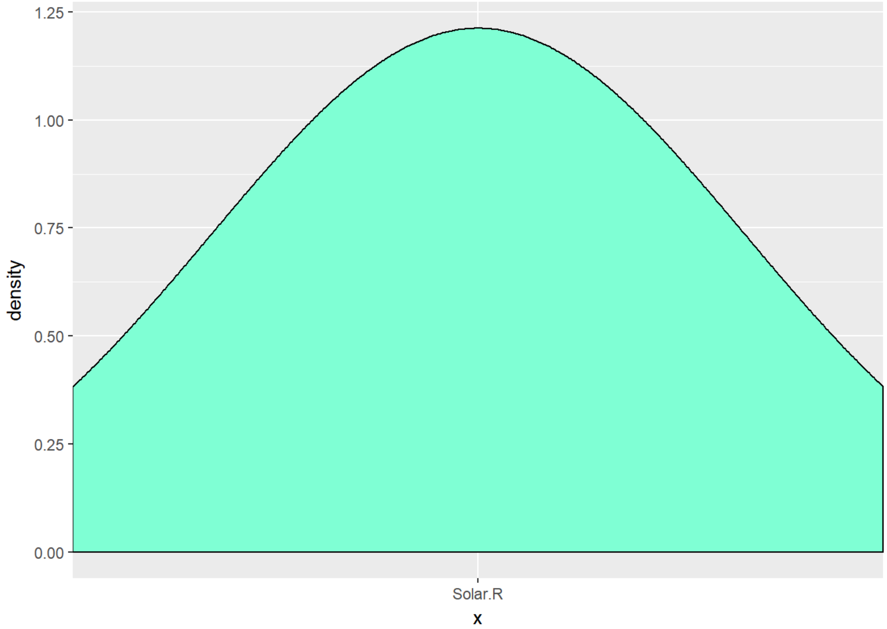
Next, let's create a boxplot showing the relationship between the month and ozone concentration. Note that I converted the Month vector into a factor; this is because Month is represented by integers and is thus continuous. If one does not convert Month into a factor, there will be only one boxplot stretching over the the range 5 to 9, which represents the numerical values of the months contained in this dataset.

```r
# Create a boxplot relating month to ozone levels
ozone_plot <- ggplot(airquality, aes(x = factor(Month), y = Ozone)) + geom_boxplot()
ozone_plot
```



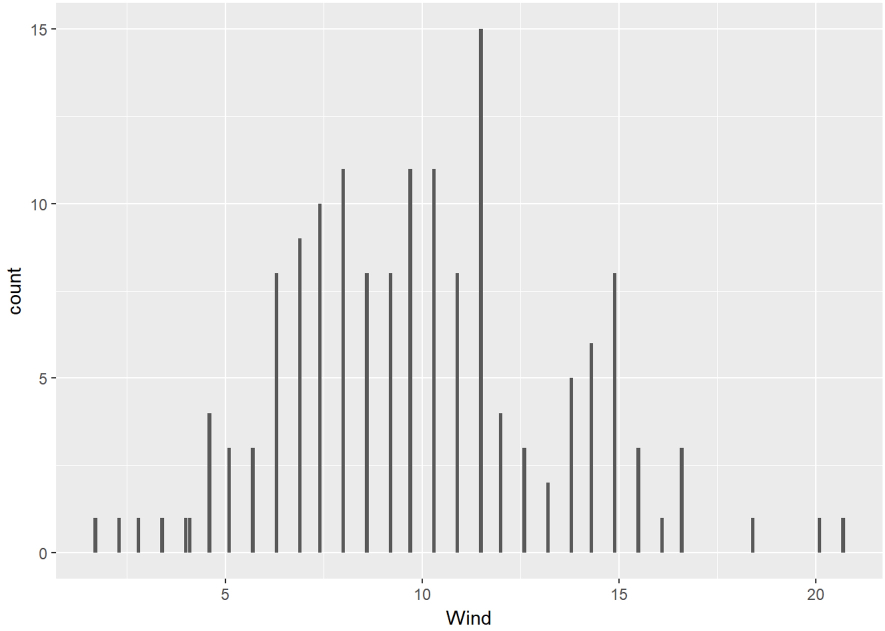
For our next visualization, let's look at the different levels of solar radiation. Similar to a histogram, the density plot will show us the distribution of a particular variable. This visualization is fairly straightforward, with only one variable called.

```r
# Create a density plot of solar radiation levels
solar_density <- ggplot(airquality, aes(x = "Solar.R")) + geom_density(fill = "aquamarine")
solar_density
```



For our final visualization, let's create a bar chart for wind speed. Again, this is a fairly simple visualization, so the code should speak for itself.

```
# Create a bar chart for wind speed.
wind_bar <- ggplot(airquality, aes(x = Wind)) + geom_bar()
wind_bar
```

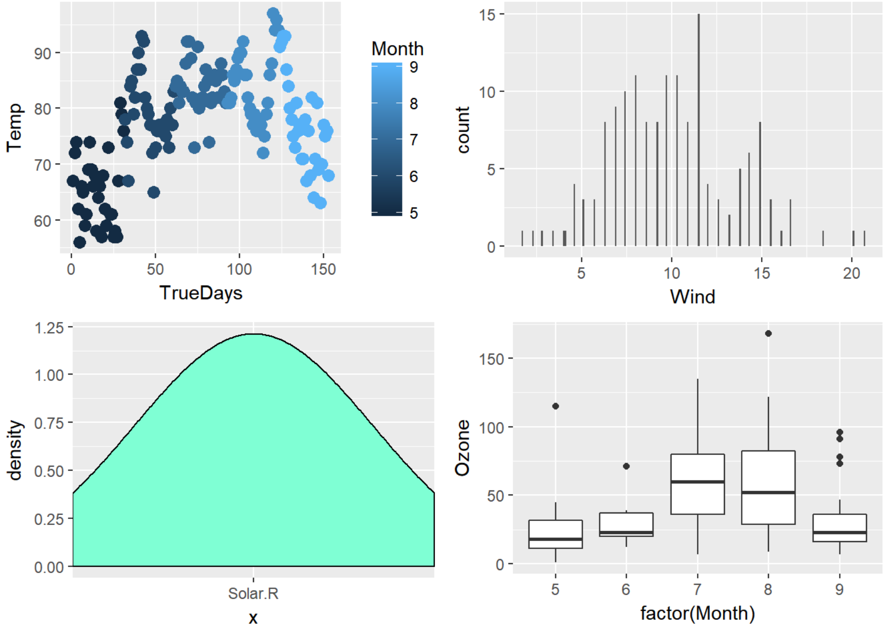


Putting Them All Together

Now that we have our four visualizations, we can merge them into one single display that will give us a simple overview of weather patterns in New York.

Displaying the four charts on one page is fairly straightforward. We call the grid.arrange function, which simply enables the user to display multiple charts on one page. We can customize this display based on number of rows and columns.

```
# Put all four charts into one display
grid.arrange(temp_scatter, wind_bar, solar_density, ozone_plot, ncol = 2, nrow = 2)
```



Great! Now we have a single display that can give us a lot of information about weather and air quality in New York at a single glance. We see a scatter plot displaying temperature patterns from May to September, a bar chart that gives us a distribution of wind speeds, a density plot of solar radiation, and a box plot showing ozone concentration and its relation to month.

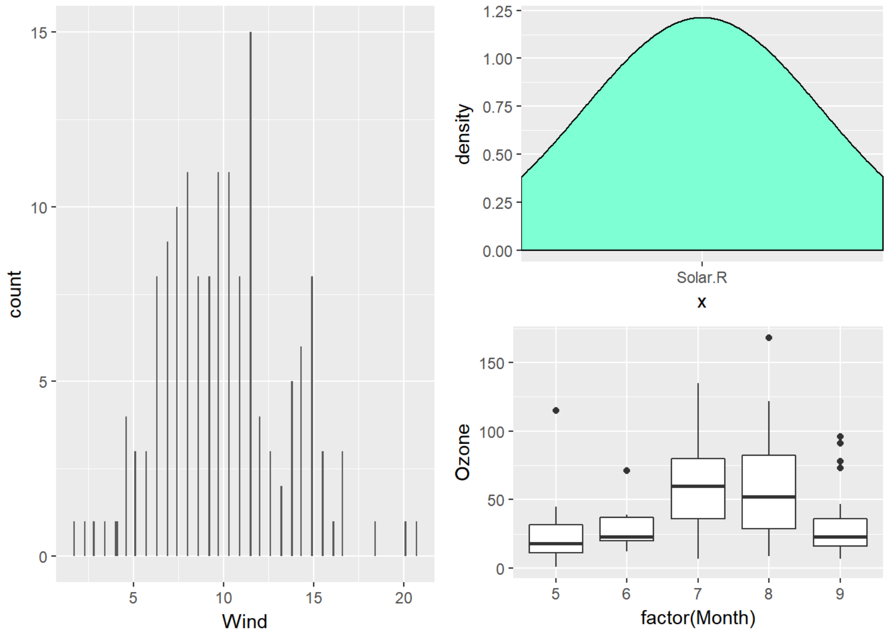
I'll leave the analysis of what each of these plots signifies to the reader, but the bottom line here is that we get a quick overview of all of the information we have decided to plot. This allows the reader to make a large quantity of observations off of visualizations that are close in proximity, rather than forcing them to comb through a tedious report.

Extra Credit

For those of you that want to experiment, let's explore a few other options as far as visualizing multiple plots on the same page. This section will train the reader on using the arrangeGrob argument, which allows for more customizable presentation.

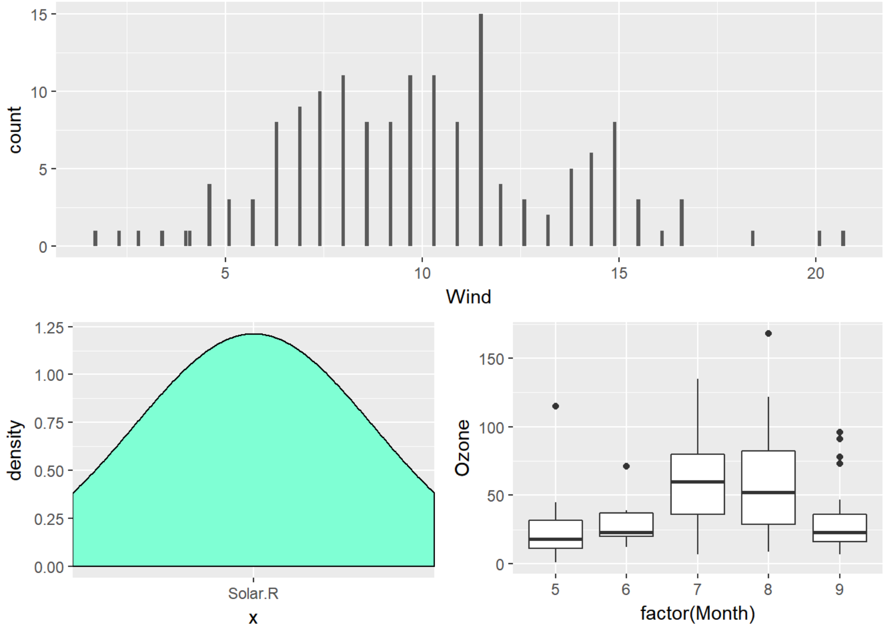
Let's say we want to include only three of our graphs in one display but would like a presentation that takes advantage of the space allotted more fashionably. We can use arrangeGrob to coerce two charts into occupying the same row, and then add this row into another row with our third chart. Note that as we increase the number of rows, we condense the graphs until they are illegible. Please be conscious of your audience's eyes when you are presenting data!

```
# Create a single row-display
grid.arrange(wind_bar, arrangeGrob(solar_density, ozone_plot), nrow = 1)
```



We can present the same three charts in a similar fashion but with a different use of space that might be preferable depending on the user's disrection. By adding in a column argument and upping the row count, we can make the display look more vertical. Depending on what the user is trying to emphasize, this may be a more favorable presentation.

```
# Create a vertically oriented display
grid.arrange(wind_bar, arrangeGrob(solar_density, ozone_plot, ncol = 2), nrow = 2)
```



That’s just about all I have here. Please feel free to experiment with more advanced applications of ggplot, and make use of grid.arrange. Both are strong ways of presenting data, and combined together they are visually appealing and robust in regards to information. I hope this blog post was helpful and inspires you to further experiment with data visualization in R. # References https://www.r-graph-gallery.com/261-multiple-graphs-on-same-page/

http://www.cookbook-r.com/Graphs/Multiple_graphs_on_one_page_(ggplot2)/

http://www.sthda.com/english/articles/24-ggpubr-publication-ready-plots/81-ggplot2-easy-way-to-mix-multiple-graphs-on-the-same-page/

http://rstudio-pubs-static.s3.amazonaws.com/2852_379274d7c5734f979e106dcf019ec46c.html

http://stat545.com/block020_multiple-plots-on-a-page.html

https://stackoverflow.com/questions/1249548/side-by-side-plots-with-ggplot2

https://stat.ethz.ch/R-manual/R-devel/library/datasets/html/airquality.html

https://stackoverflow.com/questions/35504741/multiple-plots-on-one-page-using-ggplot

https://cran.r-project.org/web/packages/egg/vignettes/Ecosystem.html

http://ggplot2.tidyverse.org/reference/geom_density.html