# Ggplots Galore!

*Sumedhaa Kothari*

*10/31/2017*

```
There is so much to learn with r, that if you're anything like me, the complexities of some key features may have
slipped over your head! For me, one of these features is ggplot. Try as I may to really understand what can be don
e with it, I find myself doing the bare minimum. In this post I will be explaining what ggplot does and some of th
e cool things that can be done with it!

To begin with a basic foundation, it should be mentioned that ggplot is a function within the ggplot2 package. Sur
prisingly, ggplot2 is not a second version of ggplot1. There is no ggplot1. I KNOW. Let that sink in. What a fake-
out.

Ggplot2 was created by Hadley Wickham as an alternative to the base graphing packages in R. It differs from the ba
se visualization package by allowing you to layer together the different graphical components (ex: data, the x and
y variable, bar graph vs. points vs. histogram etc.). This diverges from the standard package, which has the user
define exactly which graph they intend to work with in the first place, and then call the columns from the data fr
ame(s) they desire.

While there are a handful of different things that can be done with ggplot2 in comparison to the base graphics, th
ere are a few basics differences worth mentioning right off the bat. First, ggplot provides a standard grey gridde
d background, which some find to be more aesthetically pleasing than the white background provided with the base g
raphing package. Second, the graphs in ggplot come with a legend, which is standard, and you would have to impleme
nt separately in base graphics. Ggplot also requires you to use the + operator to input a display a "geometric" ty
pe for the data. For example, for a bar graph one would use geom_bar or for a boxplot, one would use geom_boxplot.
There are so many varieties of graphs that are worth playing around with such as geom_dotplot, geom_point and geom
_label to list a few. For a complete list, I suggest visiting http://ggplot2.tidyverse.org/reference/. Now its tim
e for a disclaimer: if you briefly scroll through the webpage provided you will see so many functions that the com
binations of all of these would be simply too much for me to dive into. However I found a few cool ones that I wan
t to highlight below!

To start with, let me take the opportunity to do a few interesting things with ggplot, that hopefully you have not
seen before!
```

Here are the basic steps to create a ggplot:

Preliminarily: Download the packages

```
library(ggplot2)
```

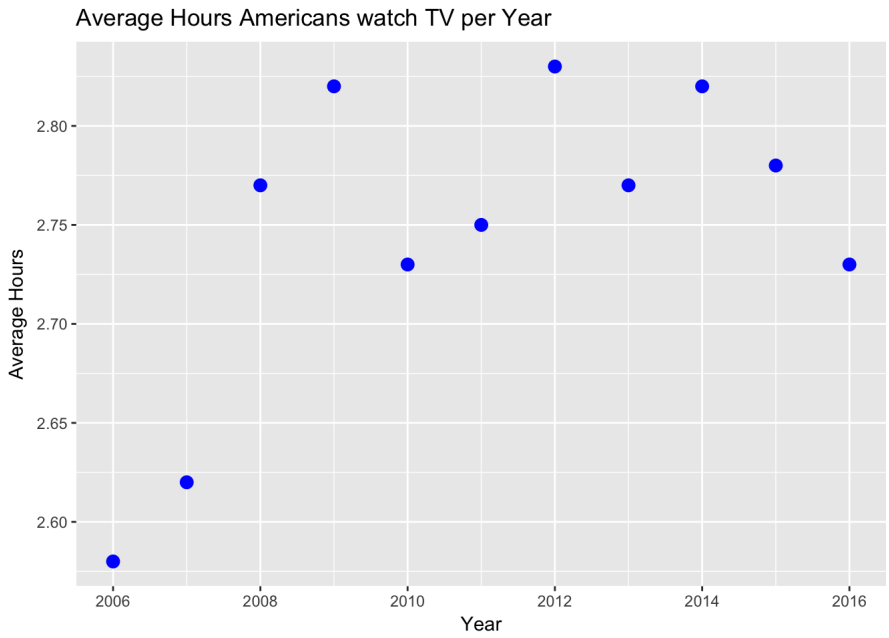
Step 1: Assemble your data into a data frame. In this example, I have pulled data from the Bureau of Labor Statistics on the average number of hours of TV are watched per day by Americans each year.

```
year <- c(2006, 2007, 2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015, 2016)
avg_hrs <- c(2.58, 2.62, 2.77, 2.82, 2.73, 2.75, 2.83, 2.77, 2.82, 2.78, 2.73)
period <- c(rep("annual", 11))
dat<- data.frame(year, avg_hrs, period)
dat
```

```
##    year avg_hrs period
## 1  2006    2.58 annual
## 2  2007    2.62 annual
## 3  2008    2.77 annual
## 4  2009    2.82 annual
## 5  2010    2.73 annual
## 6  2011    2.75 annual
## 7  2012    2.83 annual
## 8  2013    2.77 annual
## 9  2014    2.82 annual
## 10 2015    2.78 annual
## 11 2016    2.73 annual
```

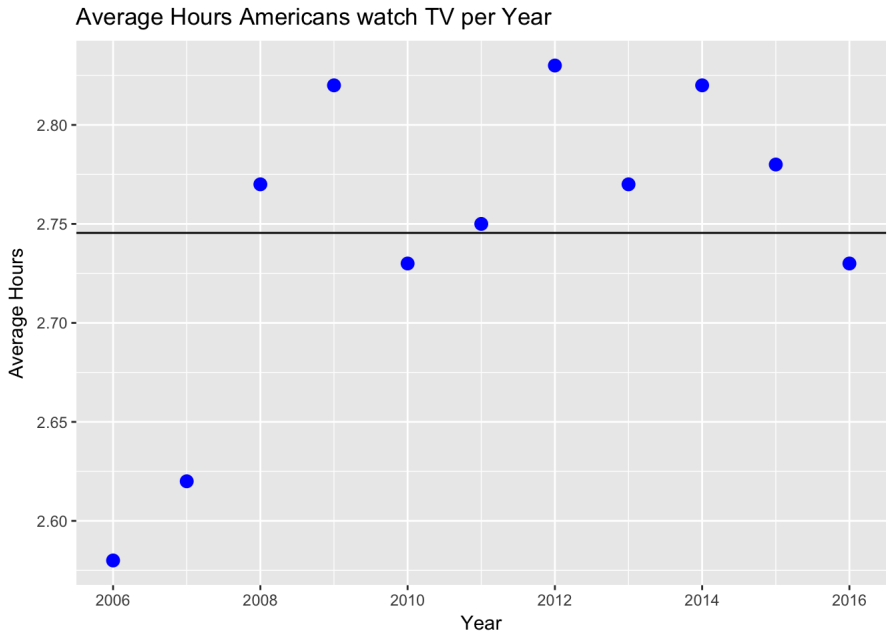
Step 2: Use the ggplot2 function. This takes in 3 parameters (your_data, aes(x_variable, y_variable)). Beyond these, you must add the geometric type that want your graph to take on. The geometric type allows you to play with color and size, among many other things more stylistic options like fill, or shape.

```
new_plot <- ggplot(dat,
       aes(x=dat$year,
           y=dat$avg_hrs)) +
  geom_point(data = dat, colour = 'blue', size = 3) + labs(x = "Year", y= "Average Hours", title= "Average Hours A
mericans watch TV per Year")

new_plot
```



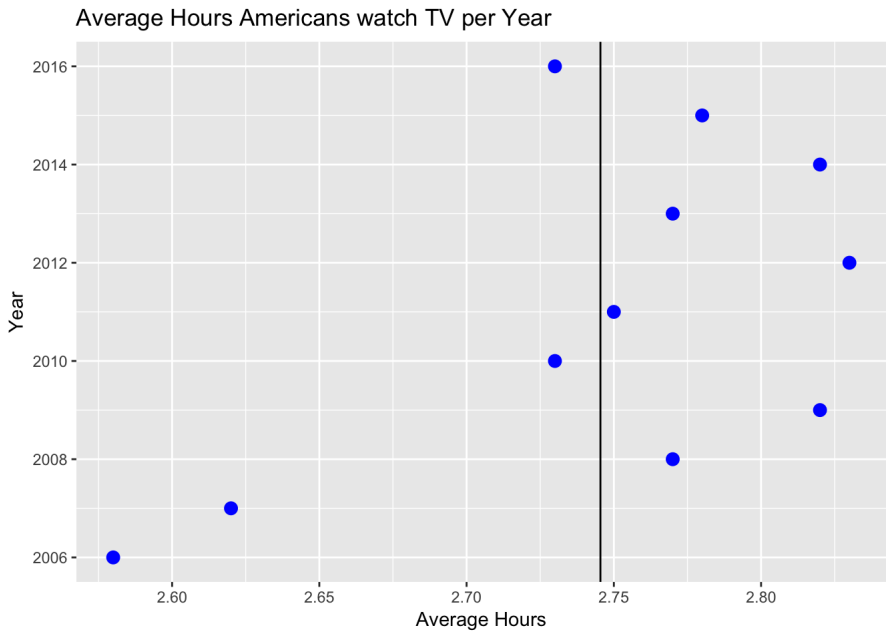

Cool feature #1: You can layer multiple graphs on top of each other! For instance, if you were looking at the graph above and wanted to see how many years were over the average number of hours for that 10 year period, you could graph this mean line over the previous graph! Take a look.

```
mean_hrs<- mean(avg_hrs)

rly_new_plot <- new_plot + geom_hline(aes(yintercept = mean_hrs))

rly_new_plot
```



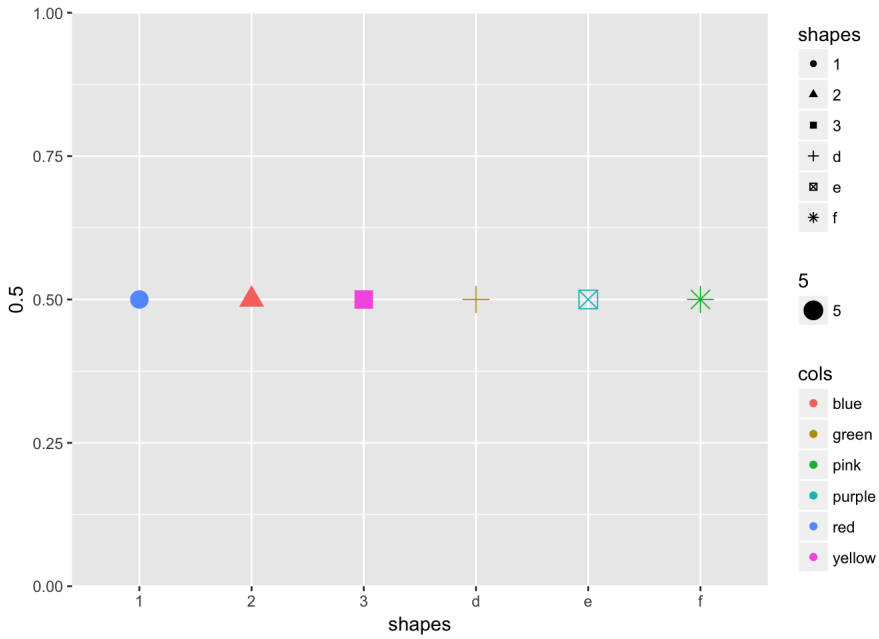

Cool feature #2: Want to just flip the coordinates to see your data with a new perspective? Call “coord_flip()” on it! Sometimes changing your data around can help you see new patterns in the exploraty phase that you might not have seen before.

```
rly_new_plot +
  coord_flip()
```



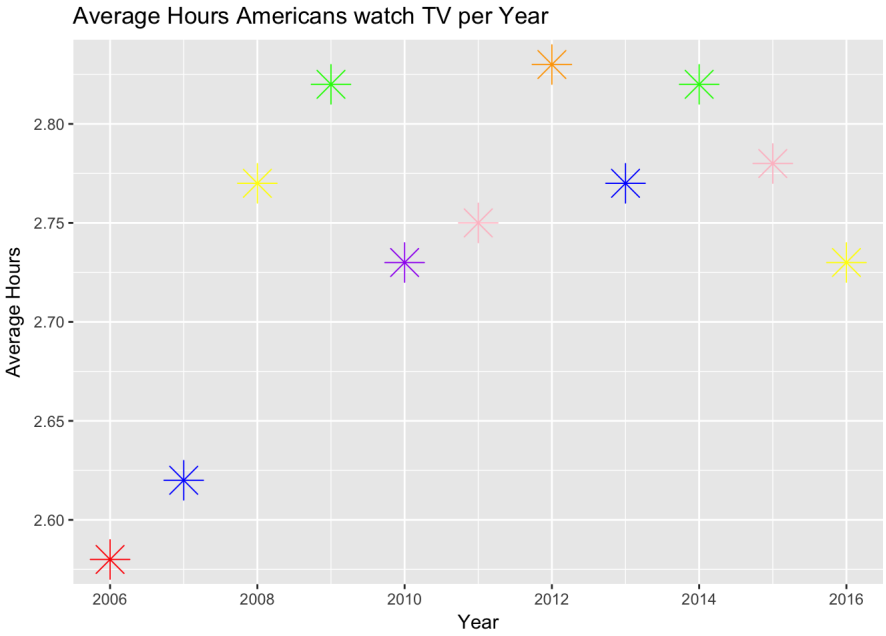

Cool feature #3: What kind of shapes does ggplot have to offer, you ask? Look no further at these cool options!

```
c=data.frame(cols <- c("red","blue","yellow","green","purple", "pink"))
d=data.frame(shapes <- c(1,2,3,"d","e","f"))
ggplot(d) +
geom_point(data=d, mapping=aes(x= shapes, y=0.5, shape= shapes, size = 5, color= cols))
```



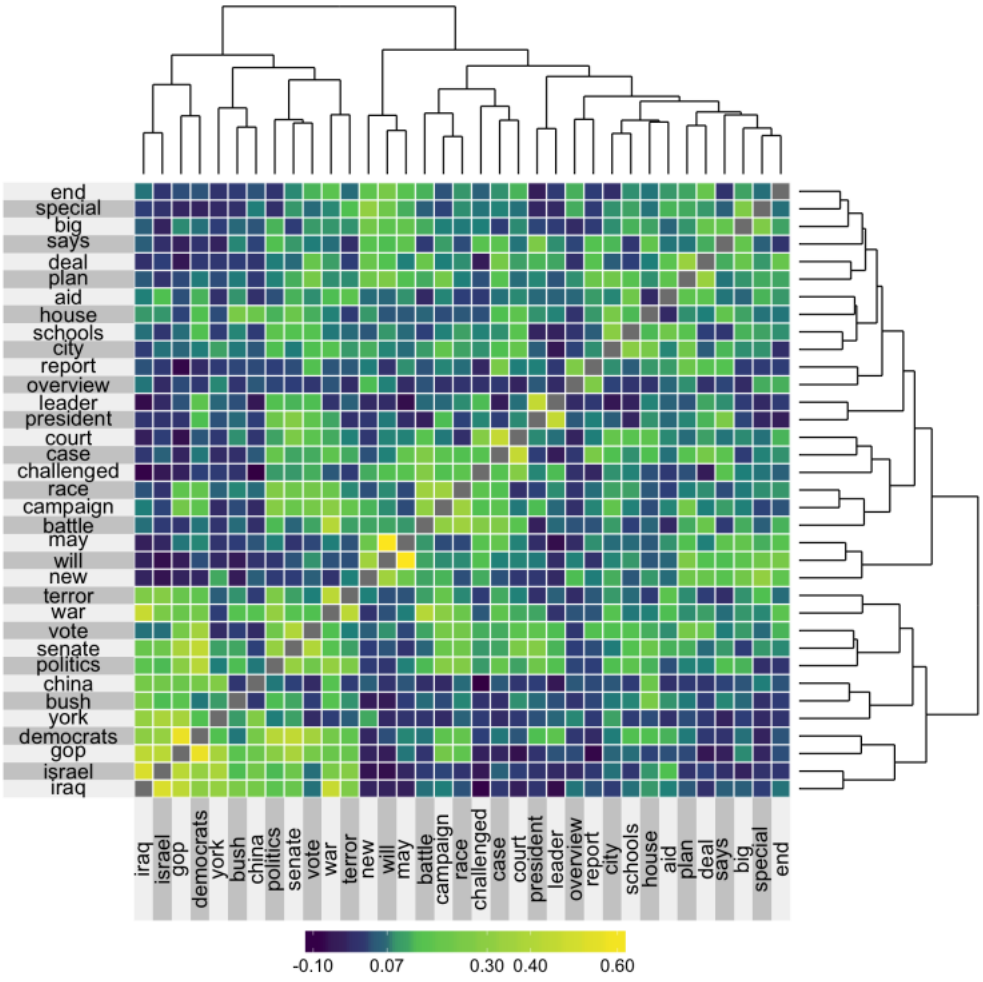

Cool Feature #3.5: Lets throw one of the shapes (that we havent seen in class before) into our original graph of Avg Hours of TV per Year. How cool is that!!

```
c=data.frame(cols <- c("red","blue","yellow","green","purple", "pink", "orange", "blue", "green", "pink", "yellow"
))

newest_plot <- ggplot(dat,
       aes(x=dat$year,
           y=dat$avg_hrs)) +
  geom_point(data = dat, colour = cols, size = 7, shape = 8) + labs(x = "Year", y= "Average Hours", title= "Averag
e Hours Americans watch TV per Year")

newest_plot
```




Before I conclude, I think it is definitely important to mention that there are opposing views to the adoption of ggplot in one's arsenal. On one hand, there are proponents for ggplot's seemingly expanded level of capabilities s uch as David Robinson (varianceexplained.com), but on the other hand, there are statisticians such as Jeff Leek wh o choose not to use ggplot in there work (simplystatistics.org). There are a few main reasons for this.

First and most commonly, ggplot requires more code to implement than a regular basic graph in the base package. Th is is a deterrent from some teachers even teaching ggplot to their students, as the base package can already do mo st of what ggplot can. Next, while the default formats are visually pleasing, the majority of the more complex gra phs you can make with ggplot you can do with the base package. And sometimes the base visual packages require less code to do it! Along these lines, many statisticians argue that in order for a graph to be publishable in the firs t place the defaults for graph should be changed anyway and therefore the defaults from ggplot are rendered redund ant. Finally, you cannot make clustered heat maps (view image below) with ggplot, like you can with the base packa ge. For these reasons there remains a debate as to which graphing package is better, and this is where you get to be the master of your own ship! Ggplot or base? Which one will it be? You decide!




```
I really hope you enjoyed reading this post and exploring some fun ggplot characteristics! If you need clarificati
on on any of the tools or plots used in my post, my resources are below! Cheers!
```

References:


- https://flowingdata.com/2016/03/22/comparing-ggplot2-and-r-base-graphics/
- https://en.wikipedia.org/wiki/Ggplot2
- http://ggplot.yhathq.com/
- http://ggplot2.tidyverse.org/reference/
- https://simplystatistics.org/2016/02/11/why-i-dont-use-ggplot2/
- http://varianceexplained.org/r/why-I-use-ggplot2/
- https://data.bls.gov/cgi-bin/surveymost
- http://rstudio-pubs-static.s3.amazonaws.com/3365_9573f6d661b444499365fe1841ee65d3.html