Post 1: Exploring ggPlot

Timothy Tan

October 30, 2017

Introduction

In class we have been introduced to a library, ggplot, that implements the "Grammer of Graphics" and creates diverse and even beautiful graphs and charts to display data. However, while what we have covered so far presents a solid introduction to ggplot, there are far more extravagant functions and graphs to be utilized. In this post, we will cover a few functions of ggplot in more depth.

Setup

```
#loading packages
require(readr)
```

```
## Loading required package: readr
```

```
require(dplyr)
```

```
## Loading required package: dplyr
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
#creating data set
nbaData <- read_csv('data/nba2017-player-statistics.csv',
                    col_types = list(Position = col_factor(
                      levels = c('C', 'PF', 'PG', 'SF', 'SG')),
                      Salary = col_double()))

#Fixing Experience
nbaData$Experience <- replace(nbaData$Experience, nbaData$Experience == 'R', '0')
nbaData$Experience <- as.integer(nbaData$Experience)

#editing and adding some data
nbaData$Salary <- round(nbaData$Salary/1000000, 2)
nbaData
```

```
## # A tibble: 441 x 24
##               Player  Team Position Experience Salary  Rank   Age    GP
##                <chr> <chr>   <fctr>      <int>  <dbl> <int> <int> <int>
##  1        Al Horford   BOS        C          9  26.54     4    30    68
##  2      Amir Johnson   BOS       PF         11  12.00     6    29    80
##  3     Avery Bradley   BOS       SG          6   8.27     5    26    55
##  4 Demetrius Jackson   BOS       PG          0   1.45    15    22     5
##  5      Gerald Green   BOS       SF          9   1.41    11    31    47
##  6     Isaiah Thomas   BOS       PG          5   6.59     1    27    76
##  7       Jae Crowder   BOS       SF          4   6.29     3    26    72
##  8       James Young   BOS       SG          2   1.83    13    21    29
##  9      Jaylen Brown   BOS       SF          0   4.74     8    20    78
## 10     Jonas Jerebko   BOS       PF          6   5.00    10    29    78
## # ... with 431 more rows, and 16 more variables: GS <int>, MIN <int>,
## #   FGM <int>, FGA <int>, Points3 <int>, Points3_atts <int>,
## #   Points2 <int>, Points2_atts <int>, FTM <int>, FTA <int>, OREB <int>,
## #   DREB <int>, AST <int>, STL <int>, BLK <int>, TO <int>
```

```
nbaData <- mutate(nbaData, Missed_FG = nbaData$FGA - nbaData$FGM,
                  Missed_FT = nbaData$FTA - nbaData$FTM,
                  PTS = nbaData$FTM + 2 * nbaData$Points2 + 3 * nbaData$Points3,
                  REB = nbaData$OREB + nbaData$DREB,
                  MPG = round(nbaData$MIN/nbaData$GP, 2))

nbaData <- mutate(nbaData, EFF = (nbaData$PTS + nbaData$REB + nbaData$AST + nbaData$STL
                                  + nbaData$BLK - nbaData$Missed_FG - nbaData$Missed_FT
                                  - nbaData$TO) / nbaData$GP)
```

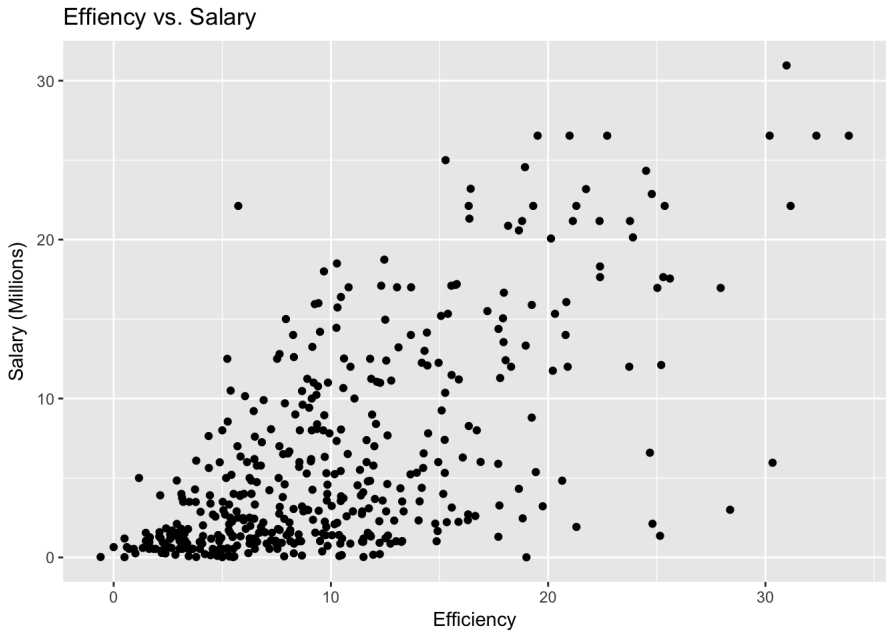
Basics

First, we will have a short review of basic plots.

Scatterplot

This is the basic scatterplot showing the correlation of Salary and Efficiency.

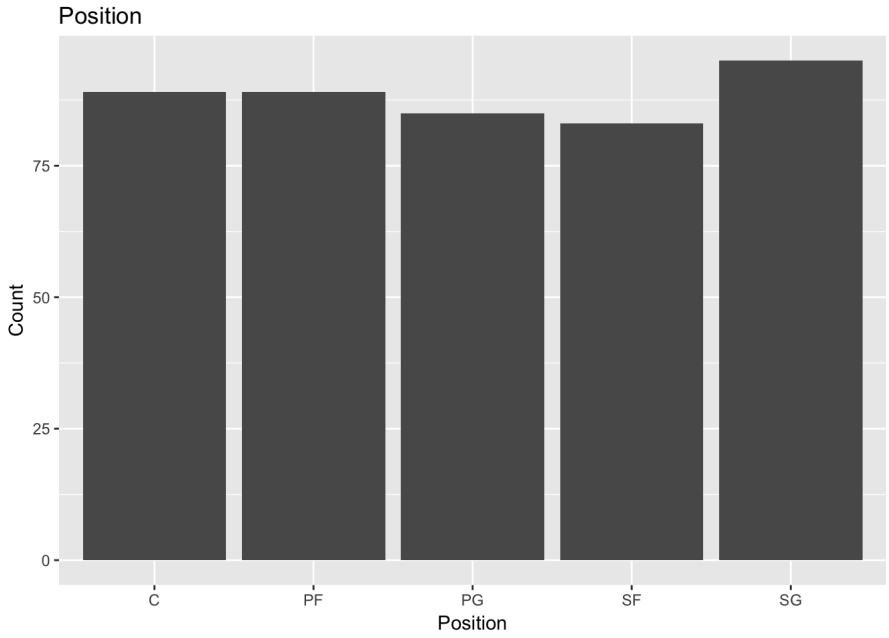
```
ggplot(data = nbaData) +
  geom_point(aes(x = EFF, y = Salary)) +
  labs(title = 'Effiency vs. Salary', x = 'Efficiency', y = 'Salary (Millions)')
```



Bar Graph

Here we have a basic bar graph showing number of players per position

```
ggplot(data = nbaData) +
  geom_bar(aes(x = Position))  +
  labs(title = 'Position', x = 'Position', y = 'Count')
```



New Plots!

Now that we have reviewed some of the basic plots, let's explore some different, more advanced plots!

Paths

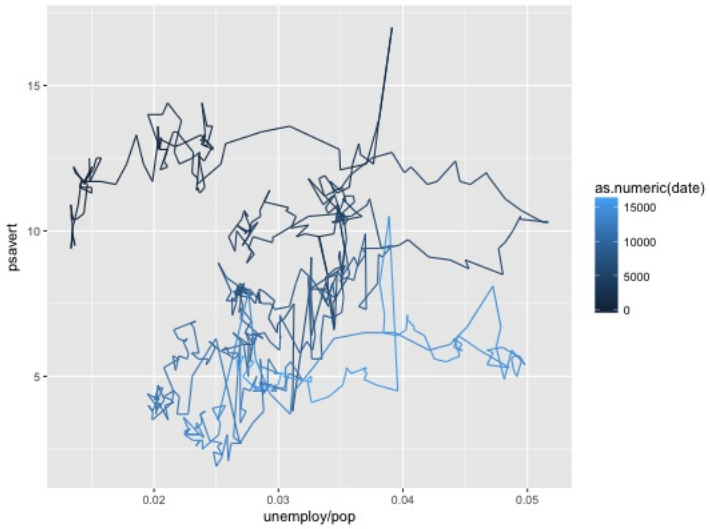
Paths allow one to explore how two variabels are related with regards to a third variable. In this case, we look how points and salary are related with regards to experience. For this section we will focus on a smaller data set, looking into the Golden State Warriors.

```
#creating Golden State Warriors data frame
gsw <- filter(nbaData, Team == 'GSW')
```

```
ggplot(gsw, aes(x = PTS, y = Salary)) +
  geom_path(aes(color = Experience)) +
  labs(title = 'Points vs. Salary with regards to Experience',
       x = 'Points', y = 'Salary (Millions)')
```

As we can see here, low experience generally means low points and low salary, while mid-range experience leads to high salary and higher scoring output, and high experience shows mid-range or low points and salary. This makes sense, as the best scoring and highest-paid players are generally those in the middle of their careers at the top of their physical primes, while the veterans make more than rookies.

Generally, a path is used more often when considering two variables over time. Below we see an economic example of unemployment rate vs. personal savings rate over time. Even though the graph looks a little jagged, one can see how the the increase of time leads to lower personal savings rates and higher unemployment rates.



Unemployment vs. Personal Savings Rate over Time

Ribbon

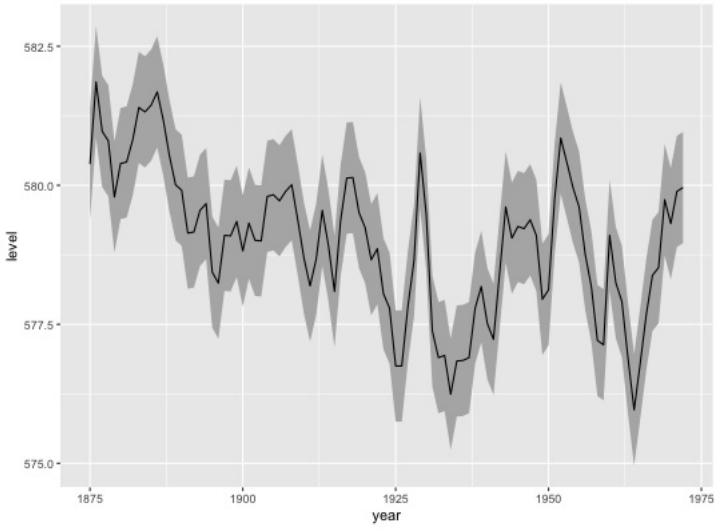
The second plot we are exploring is the ribbon plot. A ribbon plot utilizes an interval for each point over an x-variable. It is very similar to a regular graph. In the example below we are looking at efficiency as a function of minutes played. We will shorten the time range to see more of the data more clearly.

```
#creating selective minutes played data frame
shortNBAData <- filter(nbaData, MIN > 1000 & MIN < 2000)

ggplot(shortNBAData, aes(MIN)) +
  geom_ribbon(aes(ymin = EFF - 1, ymax = EFF + 1), fill = "yellow") +
  geom_line(aes(y = EFF)) +
  labs(title = 'Points vs. Salary with regards to Experience',
       x = 'Points', y = 'Salary (Millions)')
```

Efficiency oscillates quite a bit over minutes played, which makes sense as some subpar players still get a lot of playing time. Let's look at another example.

In this biology example below we see the change in water level of Lake Huron over a hundred-year period. Since water level is not constant over the course of a year, a ribbon plot is best suited to represent the data. We can see the range of water level each year as well as the change in water level over many years.



Water level of Lake Huron over time

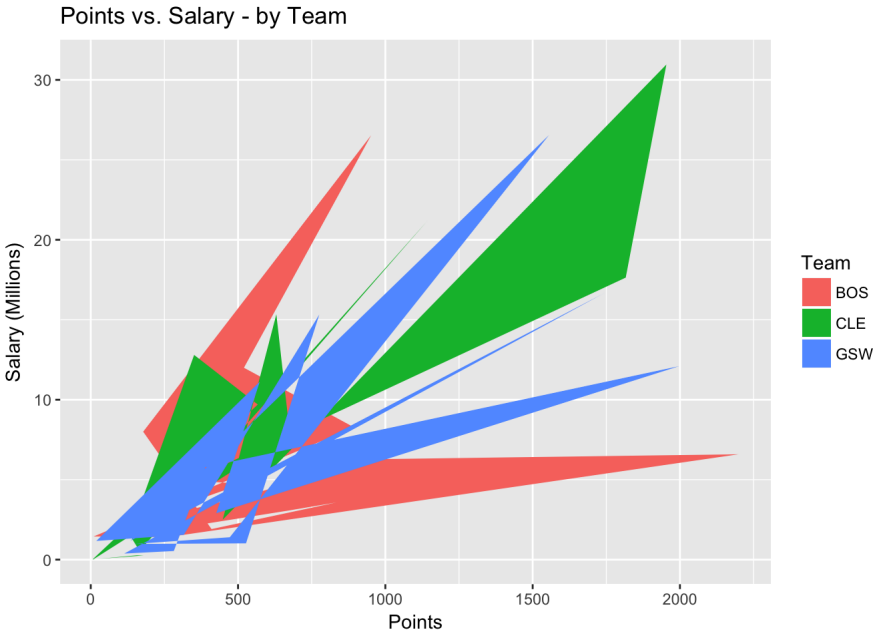
Polygon

The third and final plot we are looking at is the polygon plot. The polygon plot is similar to the path plot, except the start and end points are connected and the shape filled by the aesthetic fill. They can also be grouped by the aesthetic group.

In the example below we see points versus salary, grouped and filled by team. With a huge data set like nbaData, and having 32 teams, the polygons can become slightly confusing, so we will use a smaller set including only 3 teams.

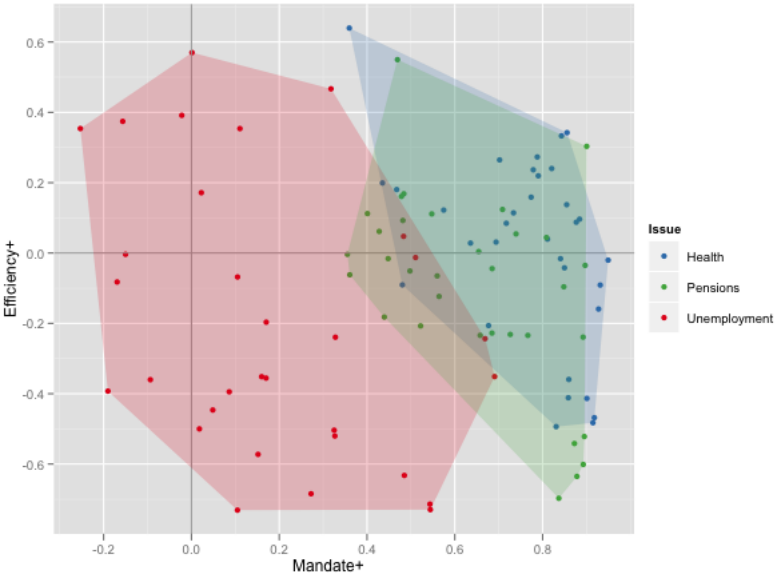
```
#creating three team data frame
gswcle <- filter(nbaData, Team == 'CLE' | Team == 'BOS' |
                  Team == 'GSW')

ggplot(gswcle, aes(x = PTS, y = Salary)) +
  geom_polygon(aes(fill = Team, group = Team)) +
  labs(title = 'Points vs. Salary - by Team',
       x = 'Points', y = 'Salary (Millions)')
```



The shapes still seem a little clustered toward the low points and low salary side of the plot, so let's look at another example.

In this political example below we see how issue mandates and efficiency are related. Based upon the different issues, we can see the area covered by each government issue, when considering possible efficiency and mandate combinations.



Effiency vs. Mandate with regards to Issue

Layers!

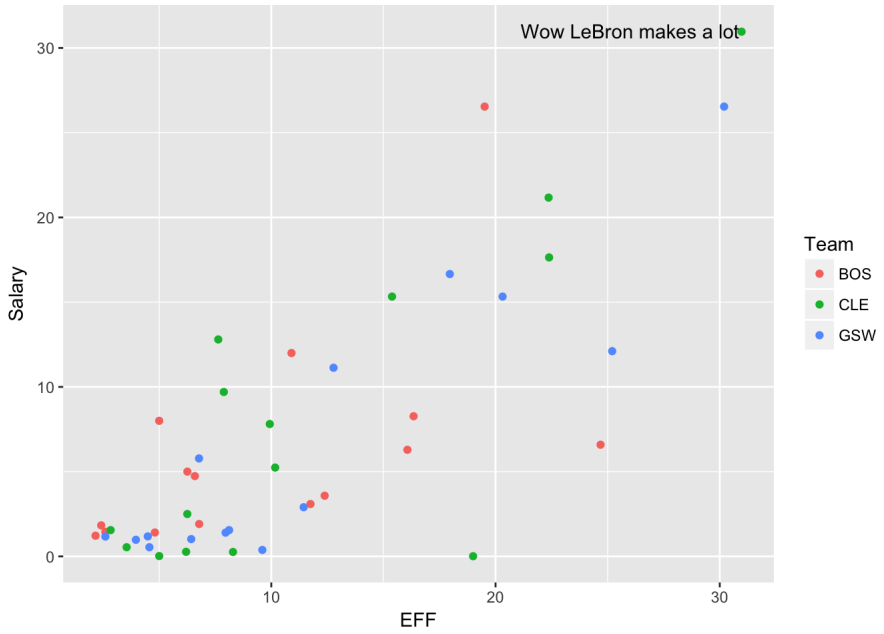
In this section we will examine some layers that can be added on top of plots. These will help explain some of the data in the plots.

Annotate

Annotate allows for some text additions or some aesthetics that don't affect the data, but can help describe the data.

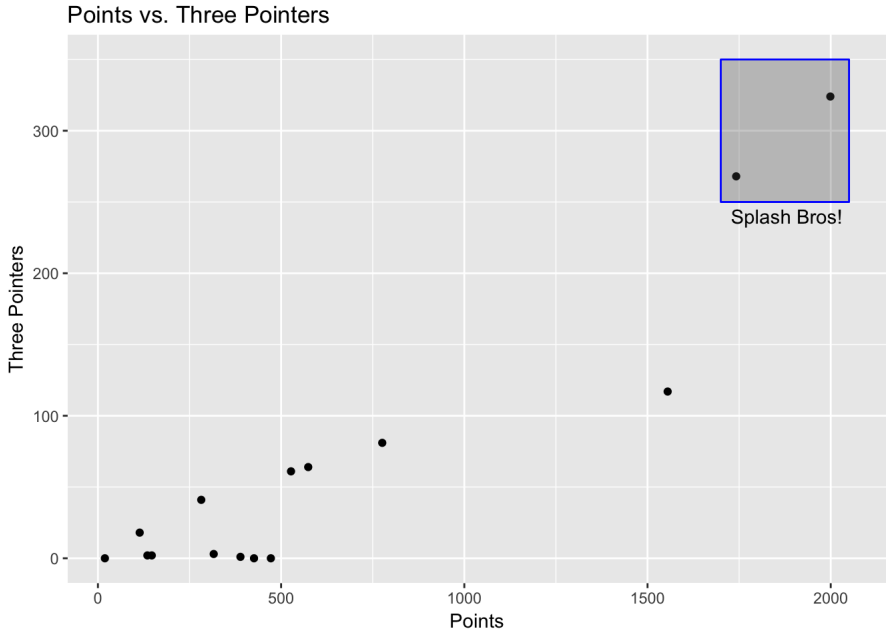
The first annotation we see is the basic text addition. You have to specify the coordinates for these, as they are plotted on top of the data, not within.

```
ggplot(gswcle, aes(x = EFF, y = Salary)) +
  geom_point(aes(color = Team)) +
  annotate("text", x = 26, y = 31, label = "Wow LeBron makes a lot")
```



The second annotation we look at is a shaded rectangle. This can be useful in emphasizing certain points. Again, you will need to specify where to create the rectangle by inputting coordinates, and you can customize it with shading depth and border color. Here we see the rectangle created with a blue border showing the so-called “Splash Brothers” of the Golden State Warriors: Steph Curry and Klay Thompson.

```
ggplot(gsw, aes(x = PTS, y = Points3)) +
  geom_point() +
  annotate("rect", xmin = 1700, xmax = 2050, ymin = 250, ymax = 350,
  alpha = .3, color = 'blue') +
  annotate("text", x = 1880, y = 240, label = "Splash Bros!") +
  labs(title = 'Points vs. Three Pointers',
       x = 'Points', y = 'Three Pointers')
```



Conclusion

We’ve seen ggplot be used in various ways, including the path, polygon, and ribbon plots, as well as the ones that have been learned in class but we’ve only scratched the surfaces to the depth of plots that can be achieved through ggplot. There are various other plots like the hexagonal heatmap, and the jitter function that can be applied to your data. To summarize, this post is meant to give you a glimpse of what ggplot can do and encourage you to further dive into the depths of all that ggplot can do.

References

http://ggplot2.tidyverse.org/reference/ http://ggplot2.tidyverse.org/reference/annotate.html
http://ggplot2.tidyverse.org/reference/geom_polygon.html http://ggplot2.tidyverse.org/reference/geom_path.html
http://ggplot2.tidyverse.org/reference/geom_ribbon.html http://ggplot2.tidyverse.org/reference/geom_path-12.png
http://ggplot2.tidyverse.org/reference/geom_ribbon-6.png https://i.stack.imgur.com/hvh2j.png
https://stats.stackexchange.com/questions/22805/how-to-draw-neat-polygons-around-scatterplot-regions-in-ggplot2
http://rmarkdown.rstudio.com/authoring_basics.html http://sape.inf.usi.ch/quick-reference/ggplot2/geom_polygon https://www.rstats-tips.net/2016/02/using-geom_ribbon-to-visualize-a-corridor-for-your-data/ https://www.r-bloggers.com/how-to-annotate-a-plot-in-ggplot2/