Post 01 - ggplot2

Stella Park

Introduction

Throughout the Stat 133 course so far, the topic that interested me the most was ggplot2. I was fascinated by how you can plot such aesthetic visual representations of data including plots, charts, and so much more using simple codes. Since I believe that visual representation is crucial in analyzing statistical data, I felt that ggplot was a superb tool that you can use to compare data, visualize abstract data, or even create something aesthetically pleasing visuals just for the beauty of it!

In this post, I will be introducing ggplot2 basics and then get into some of the coolest things you can do with the package. It's really not at all complicated to use ggplot2.

What is ggplot?

I'll start off by explaining what ggplot is. Borrowing technical words from the book ggplot2: Elegant Graphics for Data Analysis by Hadley Wickham, ggplot is an R package for producing statistical, or data, graphics based on a particular grammar. It is especially convenient because it provides beautiful, hassle-free plots that take care of fiddly details like drawing legends.

Grammar of Graphics

- data
- aesthetic mapping
- geometric object
- statistical transformations
- scales
- coordinate system
- position adjustments
- faceting

Loading the Data

Let's download some data to get started! Make sure you have R and RStudio installed in your computer, and, of course, ggplot2 package in Rstudio. I will load it because I have it already installed.

```
library(ggplot2)
```

We will be working with the NBA Players Data as a start. Follow the instructions below to download RData file to your working directory:

```
github <- "https://github.com/ucb-stat133/stat133-fall-2017/raw/master/"
csv <- "data/nba2017-players.csv"
download.file(url = paste0(github, csv), destfile = 'nba2017-players.csv')
```

Then import the data and label it as "nba" for us to easily access it when using ggplot2.

Make sure you have the readr package installed in your computer so you can read the csv file on Rstudio.

```
#load the data of 2017 NBA players
nba <- read.csv('nba2017-players.csv', stringsAsFactors = FALSE)
```

Here is a sneak peek of what the nba data file looks like:

```
head(nba)
```

```
##              player team position height weight age experience
## 1        Al Horford  BOS        C     82    245  30          9
## 2      Amir Johnson  BOS       PF     81    240  29         11
## 3     Avery Bradley  BOS       SG     74    180  26          6
## 4 Demetrius Jackson  BOS       PG     73    201  22          0
## 5      Gerald Green  BOS       SF     79    205  31          9
## 6     Isaiah Thomas  BOS       PG     69    185  27          5
##                         college   salary games minutes points points3
## 1         University of Florida 26540100    68    2193    952      86
## 2                               12000000    80    1608    520      27
## 3 University of Texas at Austin  8269663    55    1835    894     108
## 4      University of Notre Dame  1450000     5      17     10       1
## 5                               1410598    47     538    262      39
## 6      University of Washington  6587132    76    2569   2199     245
##   points2 points1
## 1     293     108
## 2     186      67
## 3     251      68
## 4       2       3
## 5      56      33
## 6     437     590
```

```
#variables of data, as established in columns of the table
colnames(nba)
```

```
##  [1] "player"     "team"       "position"   "height"     "weight"
##  [6] "age"        "experience" "college"    "salary"     "games"
## [11] "minutes"    "points"     "points3"    "points2"    "points1"
```

Below is the most basic format of ggplot2 usage:

```
ggplot(data, aes(displ, hwy, colour=class)) + layers
```

Structure:

- data: pecify the data you are using in the defining it
- aes: supply aesthetic mapping, such as x and y variables
- layers: add on layers such as types of graphs such as geom_point() for scatterplots, scales, facets, or coordinate systems

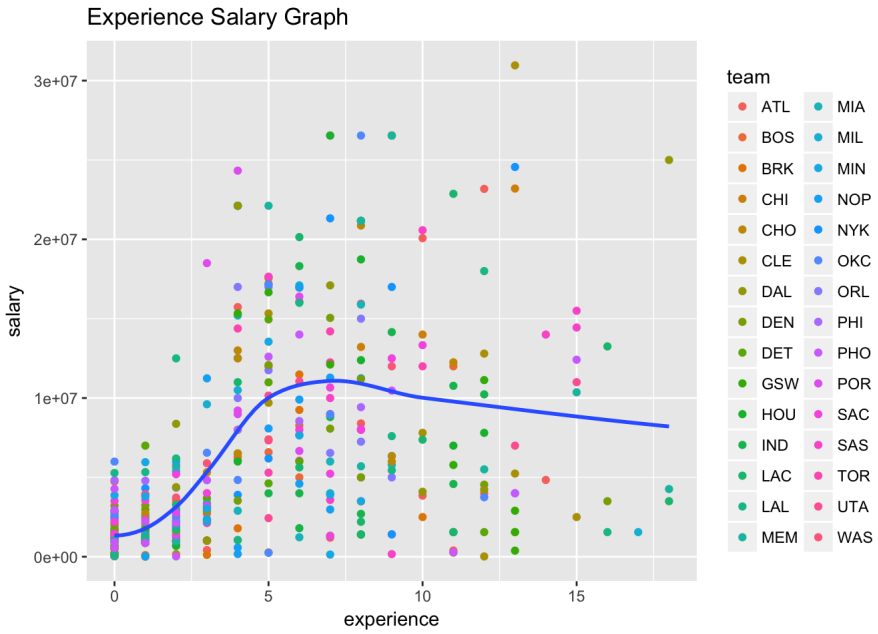
Access https://www.rdocumentation.org/packages/ggplot2/versions/2.2.1 for ggplot2 documentation and functions.

Graphing with ggplot2

1) Scatterplot

Scatterplots can be drawn using geom_point(). Refer to the website above to look up for the functions of the extra aesthetics and layers that I will add aside from the original function.

```
ggplot(nba, aes(x=experience, y=salary)) +
  geom_point(aes(color=team)) +                    #color-code by teams for easier comparison
  geom_smooth(se=FALSE, method="loess") +          #add the loess line to observe the trend of data
  labs(
    title="Experience Salary Graph",
    xlab="Experience (years)",
    ylab="Salary ($)"
  )
```

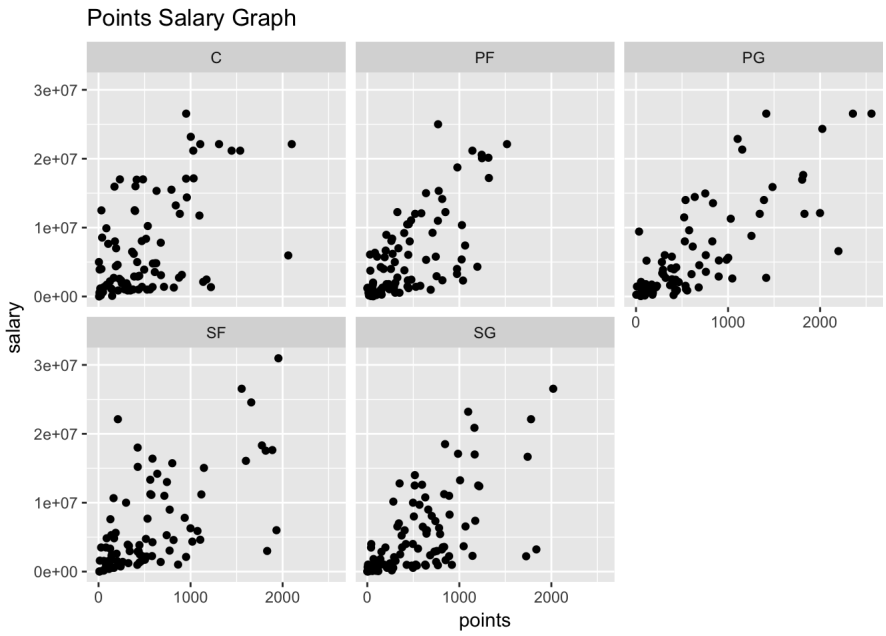


Notice how you can play around with the layers such as geom_point and distinguish the data according to teams, for instance.

2) Facets

The facet layer is useful in comparing data by grouping categorically. For instance, take this example of the relationship between points and salary, according to players' position.

```
#facet by different position to compare each position's correlation of points and salary
ggplot(nba, aes(x=points, y=salary)) +
  geom_point() +
  facet_wrap(~ position) +
  labs(
    title="Points Salary Graph",
    xlab="Points",
    ylab="Salary ($)"
  )
```



Notice how you can perform much more efficient and easy analysis of data comparing the graphs at one glance by faceting.

3) Stacking Bars

Now load the dplyr package if you have not done so already, because we will be dealing with more complex functions. This time, we will graph stacking bar charts to plot the average points, grouped by position and team, color-coded by teams.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

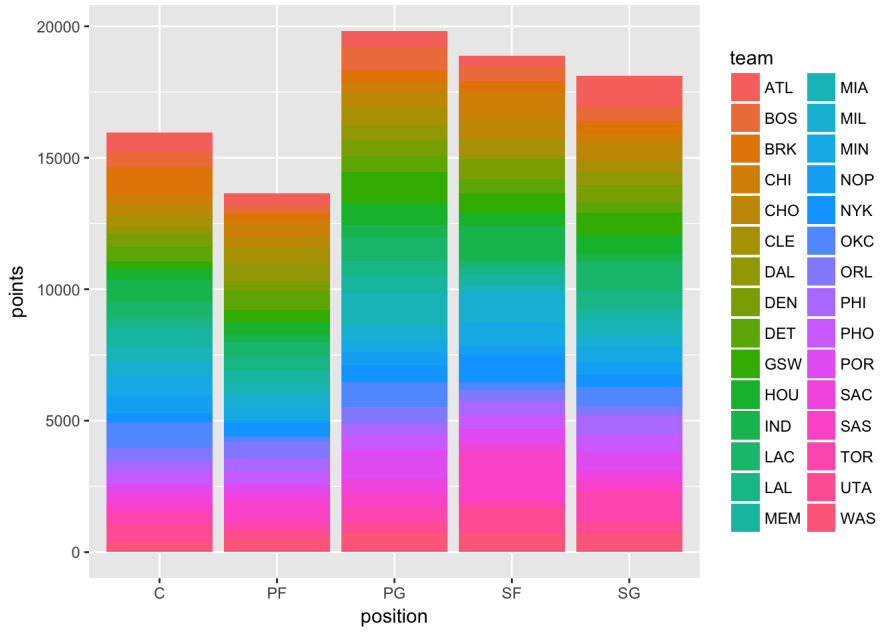
```
##
## Attaching package: 'dplyr'
```

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

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

```
#compute the average points because we are going to group it by position and team
avg_points <- nba %>%
  group_by(position, team) %>%
  summarise(points=mean(points)) %>%
  ungroup()

#fill in different colors for different teams
ggplot(avg_points) +
  geom_col(aes(x=position, y=points, fill=team))
```



It is great to be able to compare such categorical data, differentiating by color. Sophisticated and detailed graphs like these cannot be plotted without ggplot. There are so many options to choose from, so feel free to explore the various functions, referencing the url above!

Introducing ggmap

Since I was so intrigued by ggplot and visualizing data, I explored material not covered at my Stat 133 course–ggmap. It is a package of spacial visualization with ggplot2, containing a collection of functions to visualize spatial data and models on top of static maps from various online sources such as Google Maps.

Prerequisites

First, install the packages below if you have not already, in order to play with ggmap. I will load them because I have already downloaded them.

```
library(ggmap)
library(maps)
library(mapdata)
```

About the packages:

- maps: contains outlines of continents, countries, states, and counties
- mapdata: contains a few more higher-resolution outlines
- map_data(): function provided by ggplot2 that turns a series of points along an outline into data frame

Examples of the packages:

- USA map from maps package

```
usa <- map_data("usa")
head(usa)
```

```
##        long      lat group order region subregion
## 1 -101.4078 29.74224     1     1   main      <NA>
## 2 -101.3906 29.74224     1     2   main      <NA>
## 3 -101.3620 29.65056     1     3   main      <NA>
## 4 -101.3505 29.63911     1     4   main      <NA>
## 5 -101.3219 29.63338     1     5   main      <NA>
## 6 -101.3047 29.64484     1     6   main      <NA>
```

- World map centered on the Pacific Ocean from mapdata package

```
dat <- map_data("world2Hires")
head(dat)
```

```
##       long      lat group order region subregion
## 1 226.6336 58.42416     1     1 Canada      <NA>
## 2 226.6314 58.42336     1     2 Canada      <NA>
## 3 226.6122 58.41196     1     3 Canada      <NA>
## 4 226.5911 58.40027     1     4 Canada      <NA>
## 5 226.5719 58.38864     1     5 Canada      <NA>
## 6 226.5528 58.37724     1     6 Canada      <NA>
```

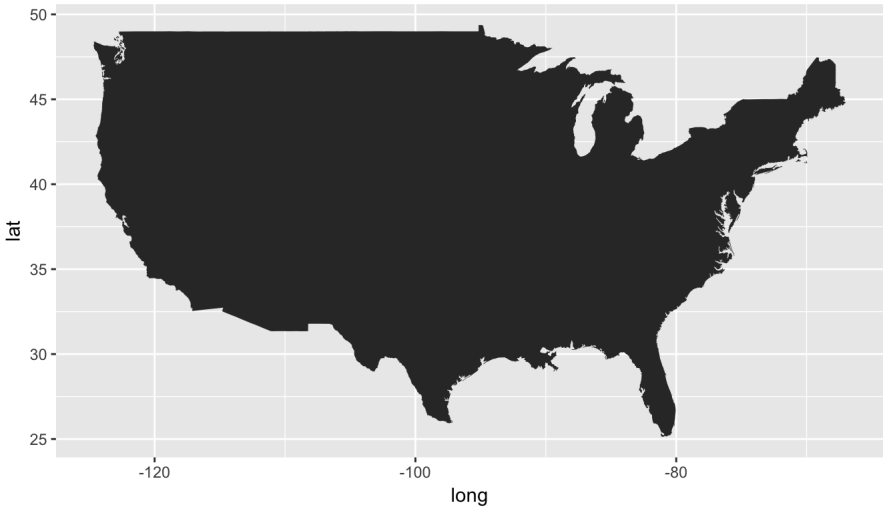
Plotting with ggmap

Now let's have fun using the various functions of ggmap! Refer to the website https://cran.r-project.org/web/packages/ggmap/ggmap.pdf which

provides detailed resources regarding ggmap and its use.

Map of the USA

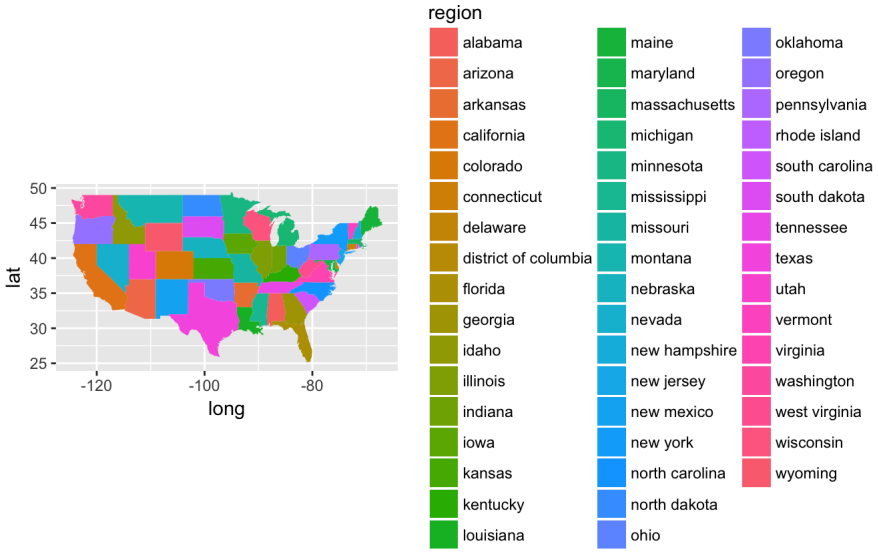
```
ggplot() + geom_polygon(data=usa, aes(x=long, y=lat, group=group)) +
  coord_fixed(1.3)      #it is essential to fix the coordinates for the map to look proportional
```



State Map of the USA

```
#define the data frame "states"
states <- map_data("state")

#plot the map of USA with states, with different colors
ggplot(data=states) +
  geom_polygon(aes(x=long, y=lat, fill=region, group=group)) +      #color by region
  coord_fixed(1.3)
```



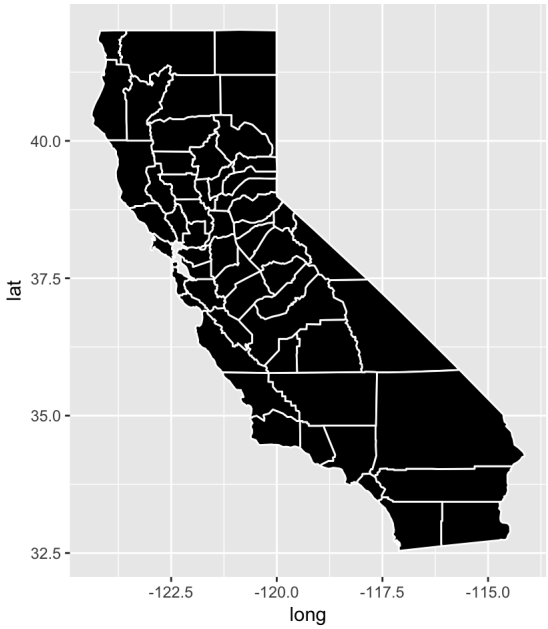
Map of California and its Counties

You can be so detailed with ggmap that you can even include boundaries of counties of the state of California!

```
#subset California from all the 50 states
cal <- subset(states, region=="california")

#subset the counties of California only
ca_county <- subset(map_data("county"), region=="california")

ca <- ggplot(data=cal, mapping=aes(x=long, y=lat, group=group)) +
  coord_fixed(1.3) +
  geom_polygon(data=ca_county, fill='black', color="white")    #add counties to the map of California
ca
```



Adding Bike Stations to ggmap

You can even encorporate data into maps created with ggmaps! I will load data of bike stations in the Bay Area and plot it on the map to show you how powerful ggplot2 is with ggmap.

```
a <- "https://github.com/data-8/materials-fa17/raw/master/"
b <- "lec/station.csv"
download.file(url = paste0(a, b), destfile = 'station.csv')
station <- read.csv('station.csv')
```

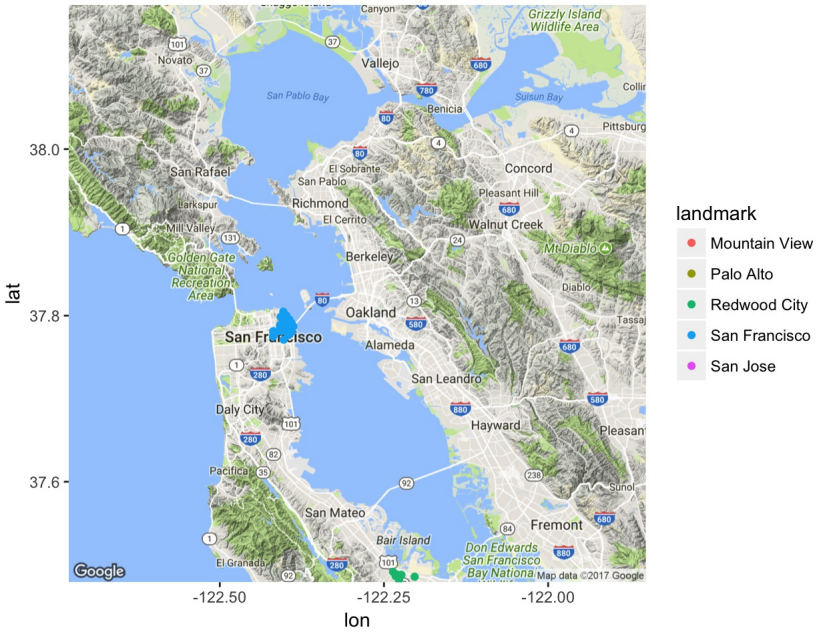
```
#get the coordinates of Bay Area
bay = get_map(location = 'Bay Area')
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=Bay+Area&zoom=10&size=640x640&scale=2&maptype=terrain&language=en-EN&sensor=false
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=Bay%20Area&sensor=false
```

```
#get map from Google Maps and plot the bike stations from the loaded data frame, station.csv
ggmap(bay) +
  geom_point(data = station, mapping = aes(x = long, y = lat, color = landmark)) #color code by landmarks
```

```
## Warning: Removed 28 rows containing missing values (geom_point).
```



Zoom into the region of San Francisco to visualize the bike stops!

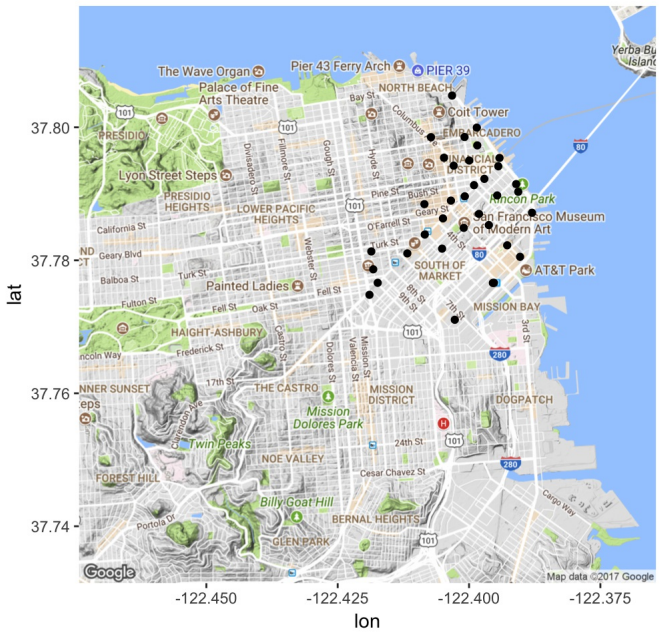
```
#select the San Francisco region this time and zoom into it
sf = get_map(location = 'San Francisco', zoom=13)
```

```
## Map from URL : http://maps.googleapis.com/maps/api/staticmap?center=San+Francisco&zoom=13&size=640x640&scale=2&
maptype=terrain&language=en-EN&sensor=false
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?address=San%20Francisco&sensor=false
```

```
ggmap(sf) +
  geom_point(data = station, mapping = aes(x = long, y = lat))
```

```
## Warning: Removed 35 rows containing missing values (geom_point).
```



Concluding

That's a wrap! It was fascinating to research about such diverse functions and operations that ggplot2 can do, particularly in combination with other packages such as ggmap. I hope you also get to explore the fun stuff that I tried with ggplot2.

As it is crucial to compute data, I believe that it is as pivotal to observe the data by visualizing, since comparison becomes far more convenient and efficient if done with visualization. It is prudent to use tools such as ggplot2, not only because of the aesthetics of it, but also because of the diversity of application and power it encompasses.

Sources

- https://books.google.com/books?

hl=en&lr=&id=XgFkDAAAQBAJ&oi=fnd&pg=PR8&dq=ggplot2&ots=soY85Qc_6N&sig=XoYKVfd6LhTOD29pf7zLibzC-a8#v=onepage&q=ggplot2&f=false
- https://www.youtube.com/watch?v=49fADBfcDD4
- https://link.springer.com/chapter/10.1007/978-3-319-24277-4_11/fulltext.html
- https://github.com/tidyverse/ggplot2/wiki/Why-use-ggplot2
- https://www.rdocumentation.org/packages/ggplot2/versions/2.2.1
- https://blog.rstudio.com/2016/11/14/ggplot2-2-2-0/
- https://cran.r-project.org/web/packages/ggmap/ggmap.pdf