

Post01: Illustrating Data of Locations Using ggplot2

Yijing Zhu

October 30, 2017

Introduction

In Stat 133 we have learnt `ggplot()` to draw graphs on data. Certain types of data, like numeric data, can be described with basic `ggplot2` functions, like scatterplots and histograms. My motivation of writing this article is to explore ways to illustrate data of locations to get a closer look of data of my favorite tennis players. Data used here are the match results of tennis ATP in 2016. Methods on making a frequency table, a pie chart, and drawing a map will be introduced.

Audience

This post is for students from Stat 133 who know the basics of `ggplot2`, but would love to learn other aspects of the package based on their interests.

Data Preparations

The first step of analyzing data is to load necessary packages and read tables.

```
library("ggmap")
```

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

```
## Warning: package 'ggplot2' was built under R version 3.3.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.3.2

##
## 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

library(countrycode)

## Warning: package 'countrycode' was built under R version 3.3.2

# Don't forget to set the working directory to an appropriate folder.
dat<-read.csv('atp_matches_2016.csv', stringsAsFactors = FALSE)
```

Frequency Counts

```
frequencytable<-table(filter(dat,round=='F')$winner_ioc)

## Warning: package 'bindrcpp' was built under R version 3.3.2

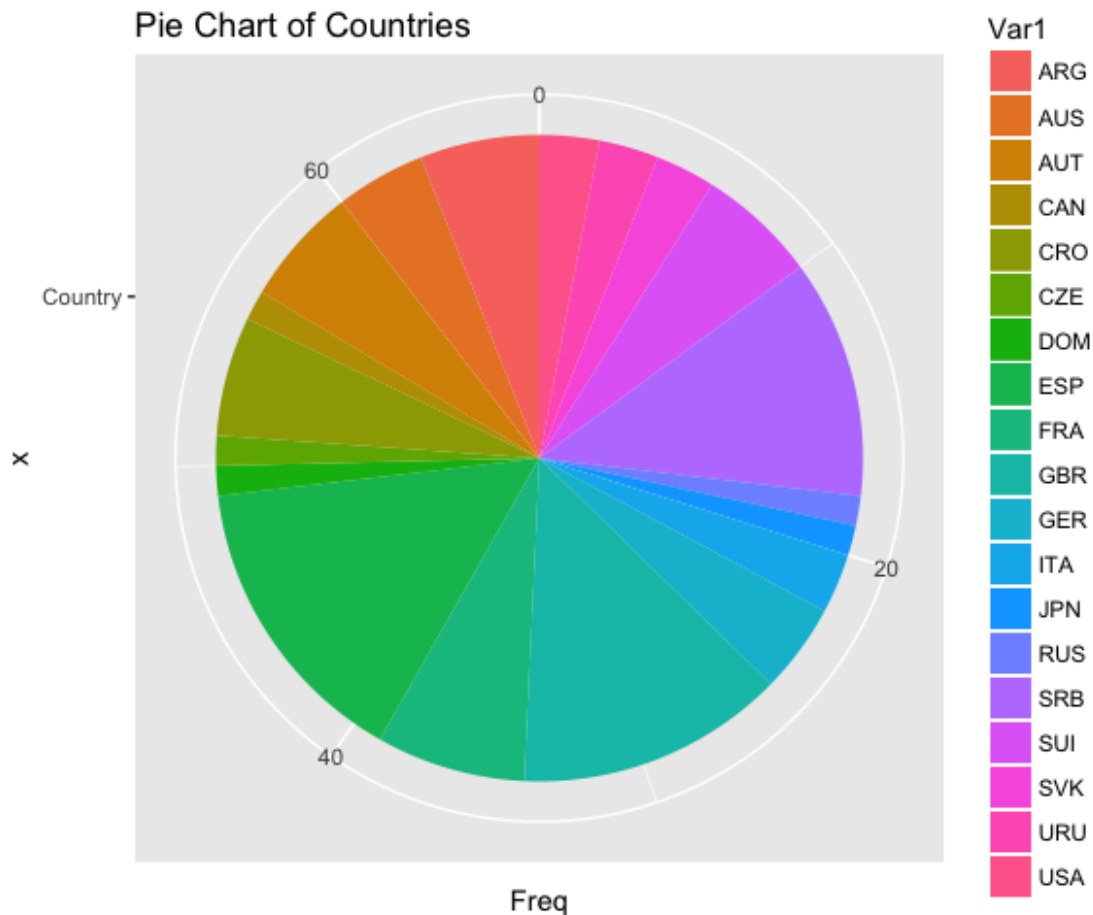
w<-as.data.frame(frequencytable)
head(w)

##   Var1 Freq
## 1  ARG     4
## 2  AUS     3
## 3  AUT     4
## 4  CAN     1
## 5  CRO     4
## 6  CZE     1
```

Pie Chart

Drawing pie charts require add more descriptions of ggplots.

```
pie<-ggplot(w, aes(x="Country",y=Freq,fill=Var1))+  
geom_bar(width = 1, stat = "identity")+coord_polar("y", start=0)+ggtitle("Pie  
pie
```



Mapping Locations

Name of locations can be transformed into longitudes and latitudes with `geocode()`. Google maps can then draw these locations on a map for you.

```
# show locations on a map  
map<-select(dat,round,winner_ioc)
```

```
map<-filter(map,round=='F')
map<-group_by(map,winner_ioc)
map<-head(map)

country<-countrycode(map$winner_ioc,"ioc","country.name")
country
```

```
## [1] "Canada"      "Switzerland" "Serbia"      "Spain"      "Serbia"
## [6] "Serbia"
```

```
ll.country <- geocode(country)
```

```
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?adc
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?adc
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?adc
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?adc
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?adc
## Information from URL : http://maps.googleapis.com/maps/api/geocode/json?adc
```

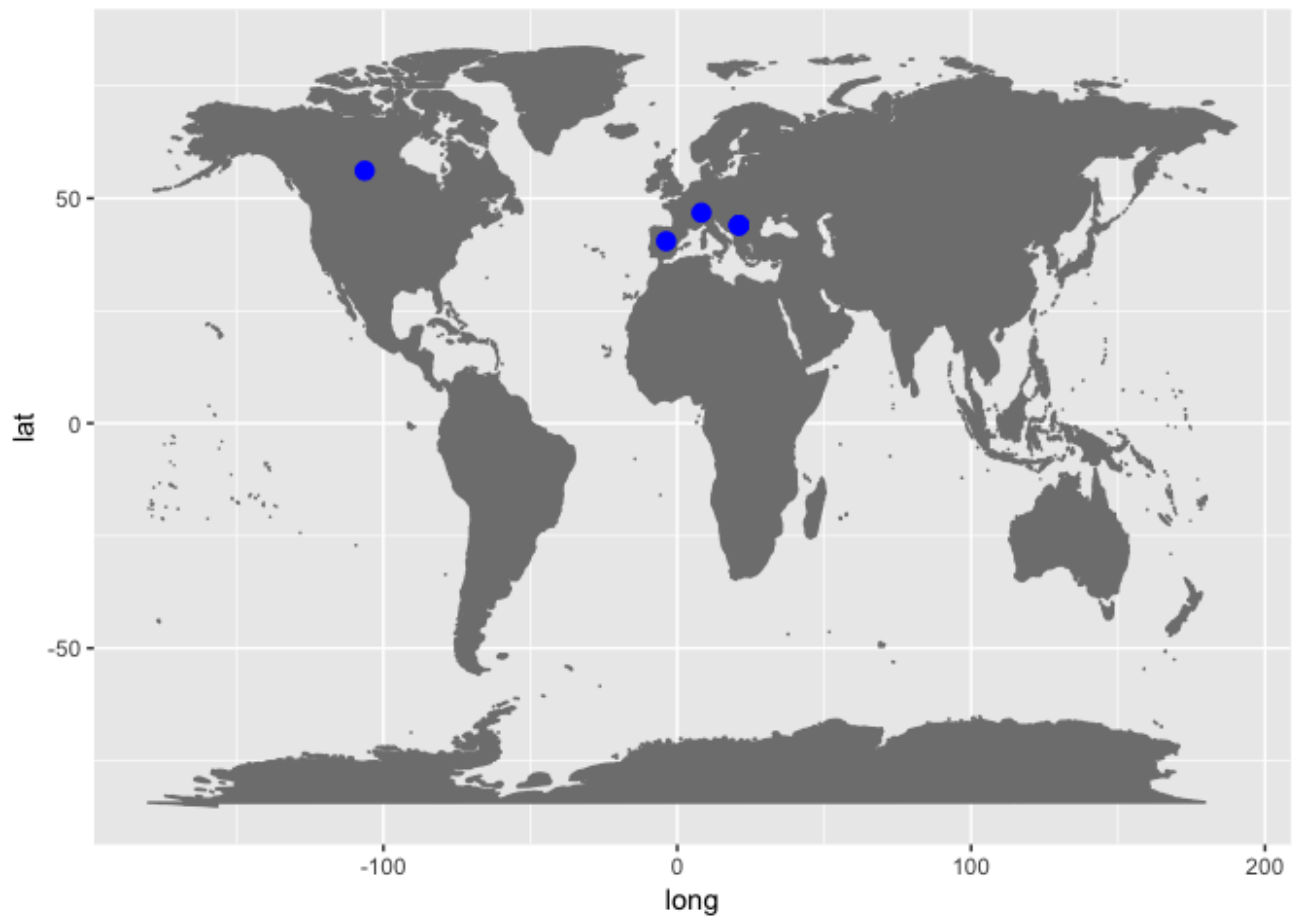
```
country.x <- ll.country$lon
country.y <- ll.country$lat
```

```
mp <- NULL
mapWorld <- borders("world", colour="gray50", fill="gray50") # create a layer
```

```
## Warning: package 'maps' was built under R version 3.3.2
```

```
mp <- ggplot() + mapWorld
```

```
#Now Layer the cities on top
mp <- mp+ geom_point(aes(x=country.x, y=country.y) ,color="blue", size=3)
mp
```



Message

Data of locations can be illustrated with ggplot2 by pie charts and drawing on maps with methods shown above.

References

- 1 <https://www.statmethods.net/advgraphs/ggplot2.html> 2
- https://github.com/JeffSackmann/tennis_atp/blob/master/atp_matches_2016.csv 3
- <https://www.r-bloggers.com/r-beginners-plotting-locations-on-to-a-world-map/> 4
- <http://www.r-graph-gallery.com/128-ring-or-donut-plot/> 5
- <http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html> 6
- <http://swirlstats.com/blog/scn.html> 7 https://plot.ly/ggplot2/geom_polygon/