Post01-Tiantian-Fu

Data Visualization with ggplot2

Introduction

As you have known,the ability to produce meaningful and beautiful data visualizations is an essential part of your skill set as a data scientist. The main package we normally use is ggplot. Primarily, there are 8 types of objectives you may construct plots. They are *Correlation*,*Deviation*,*Ranking*,*Distribution*,*Composition* ,*Change*,*Groups*,and *Spatial* perspectively. And in each objectives, there are all kinds of graphs you can use. In this post, what I want to do is to introduce you **Area Graphs** , which are used to display the development of quantitative values over an interval or time period. They are most commonly used to show trends, rather than convey specific values.

Data preparation

Fisrt, I am going to generate a data frame sampling a 400 by 2 data frame containing females and males with different weights, different means and standard deviations.

```
# load package and data
library(plyr)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.4.2
```

```
set.seed(1234)
df <- data.frame(
  sex=factor(rep(c("F", "M"), each=200)),
  weight=round(c(rnorm(200, mean=55, sd=5),
                 rnorm(200, mean=65, sd=5)))
  )
head(df)
```

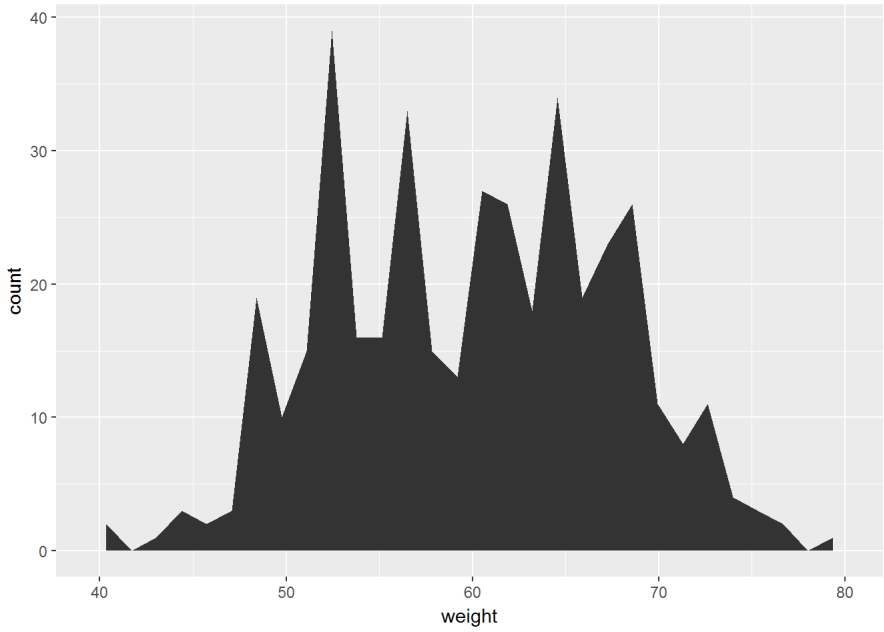
```
##   sex weight
## 1   F     49
## 2   F     56
## 3   F     60
## 4   F     43
## 5   F     57
## 6   F     58
```

Then, Let us start with some basic area plots.

Basic area plots

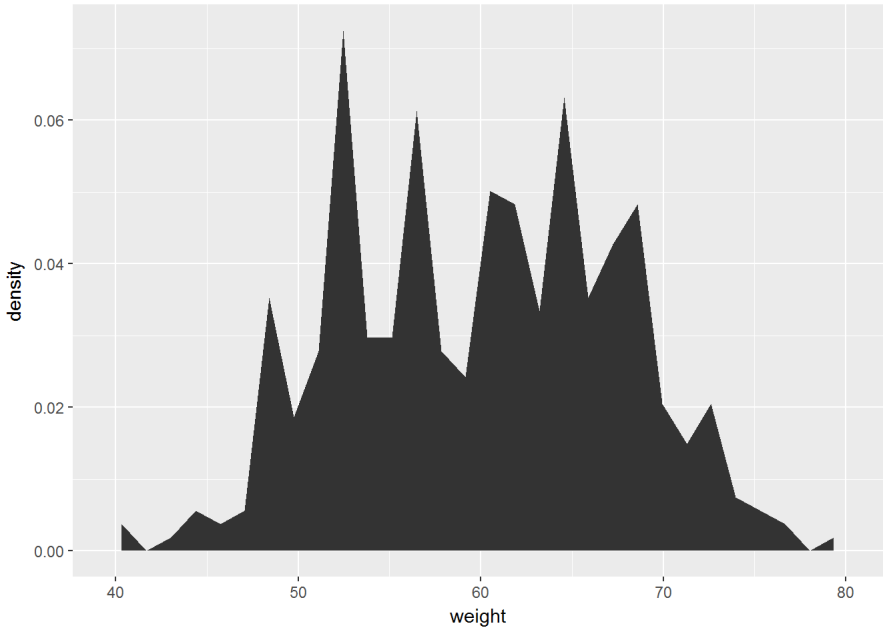
```
p <- ggplot(df, aes(x=weight))
# Basic area plot
p + geom_area(stat = "bin")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



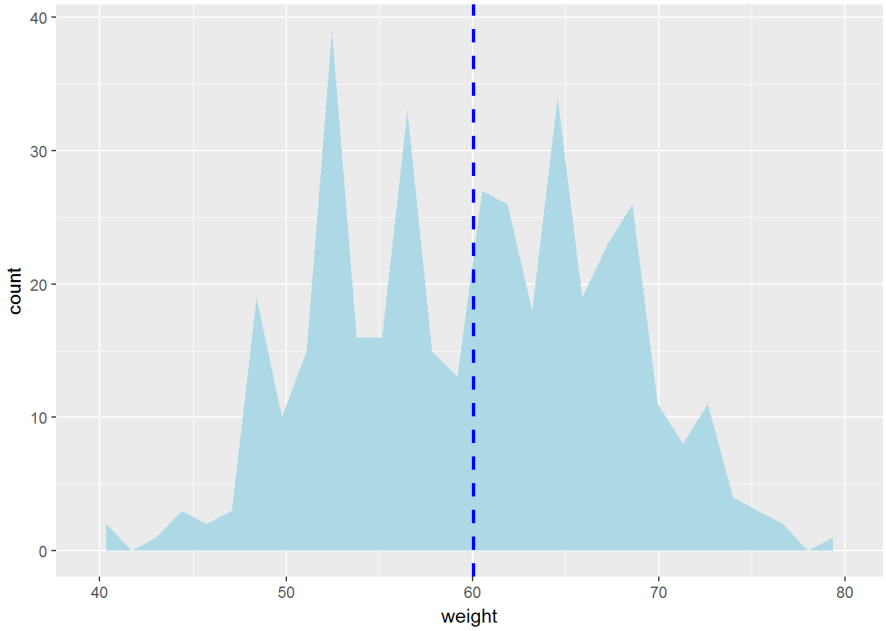

```
# y axis as density value
p + geom_area(aes(y = ..density..), stat = "bin")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Add mean line
p + geom_area(stat = "bin", fill = "lightblue")+
  geom_vline(aes(xintercept=mean(weight)),
             color="blue", linetype="dashed", size=1)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

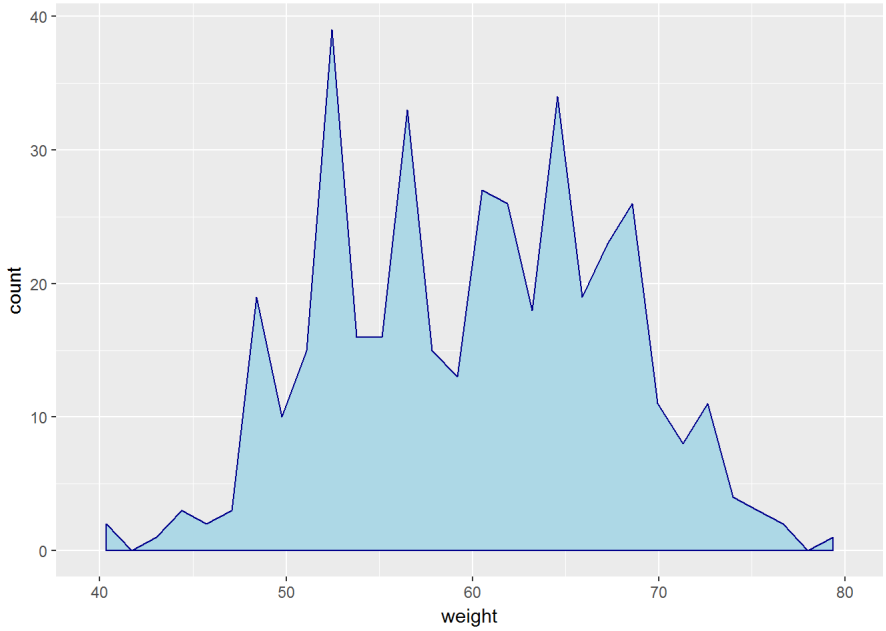


Analysis: For graph one, I graphed the distribution between weight and count by using areas plot. As we can see from the graph, the majority weight of the sample data are distributed between 50 kg to 70 kg and when it gets close to two sides, there are less amount of people being in those areas. Graph two represents the same idea as the first one except that I changed y axis-count to density. For Graph three, I filled in blue color and added a verticle line representing the mean value of weight.

Change line types and colors

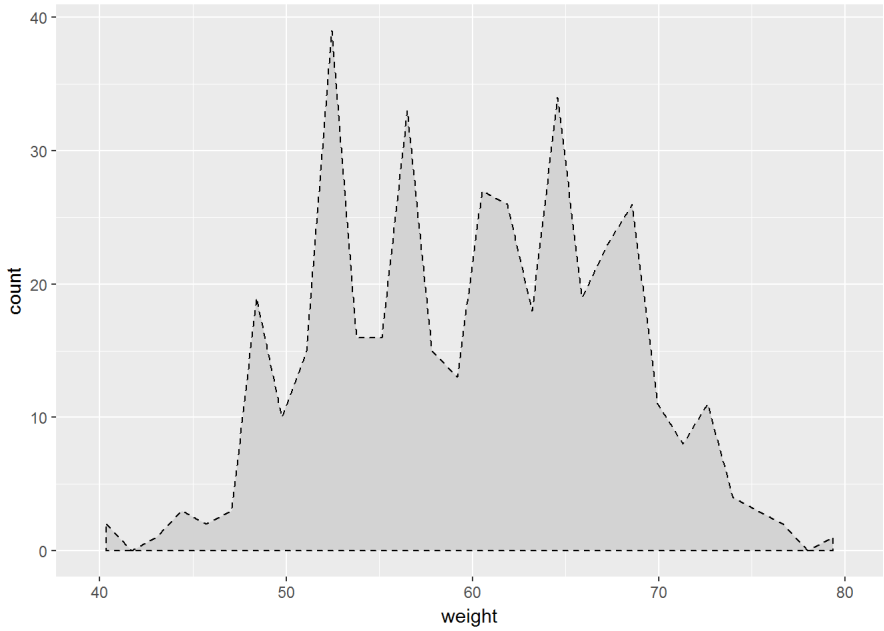
```
# Change line color and fill color
p + geom_area(stat ="bin", color="darkblue",
              fill="lightblue")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Change line type
p + geom_area(stat = "bin", color= "black",
              fill="lightgrey", linetype="dashed")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Analysis: For the first graph, I changed the line color of this graph and kept everthingelse the same. For the second graph, I changed the line type from straight line to dashed line.

Change colors by groups

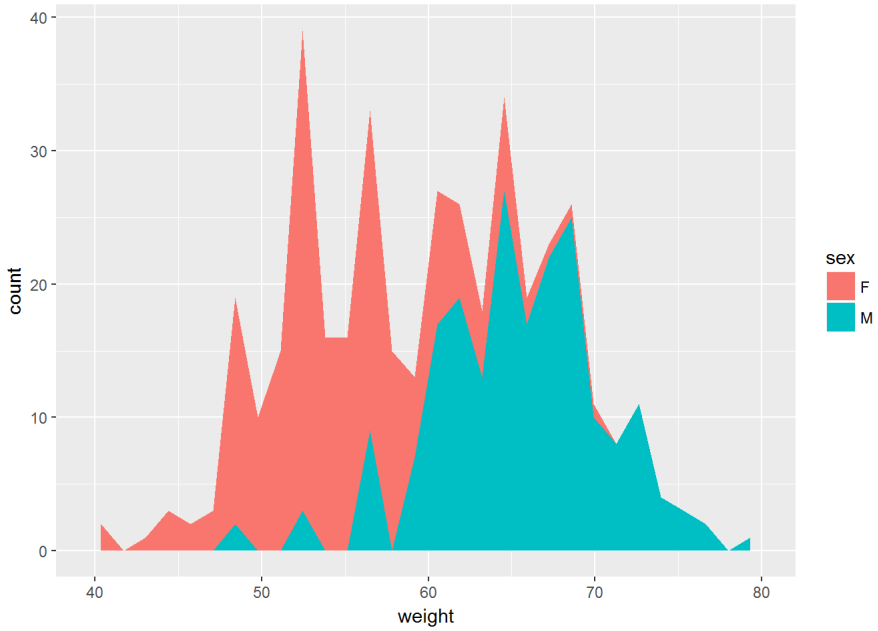
Calculate the mean of each group Change fill colors

```
mu <- ddply(df, "sex", summarise, grp.mean=mean(weight))
head(mu)
```

```
##   sex grp.mean
## 1   F    54.70
## 2   M    65.36
```

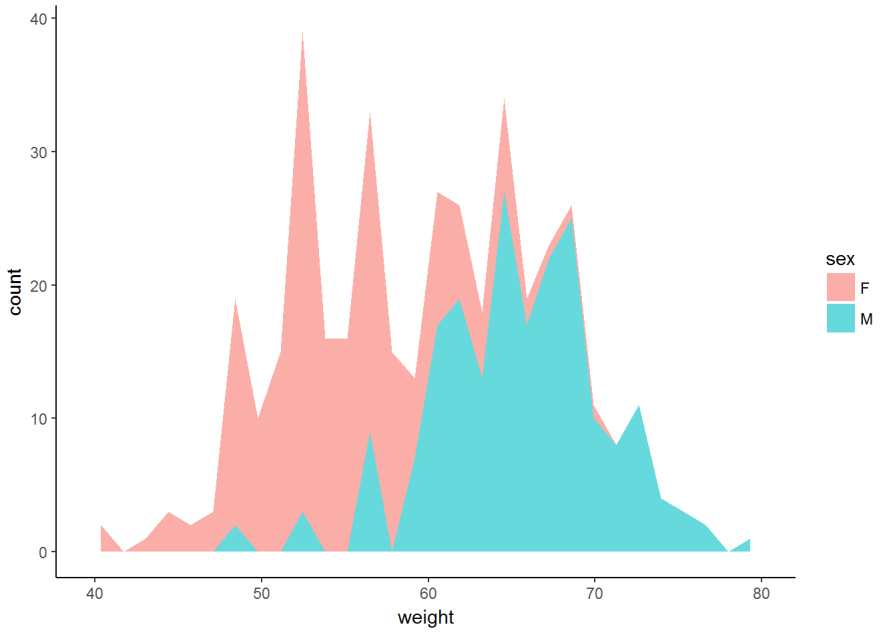
```
# Change area plot fill colors by groups
ggplot(df, aes(x=weight, fill=sex)) +
  geom_area(stat ="bin")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



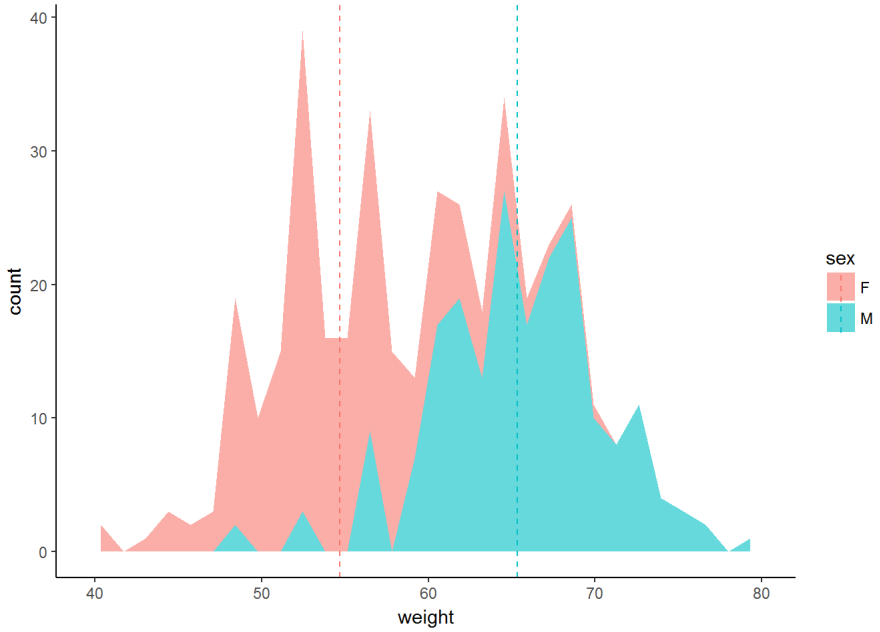

```
# Use semi-transparent fill
p<-ggplot(df, aes(x=weight, fill=sex)) +
  geom_area(stat ="bin", alpha=0.6) +
  theme_classic()
p
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



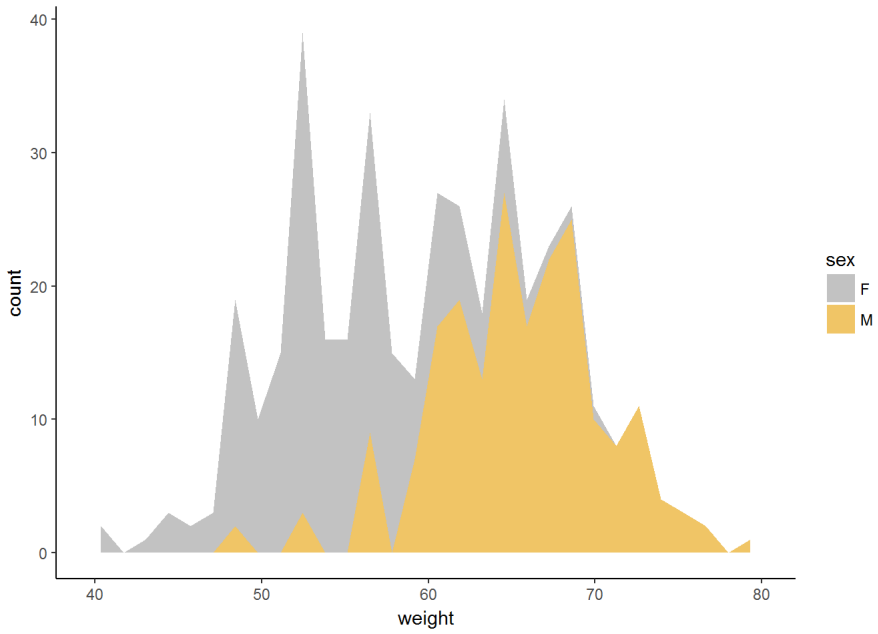

```
# Add mean lines
p+geom_vline(data=mu, aes(xintercept=grp.mean, color=sex),
             linetype="dashed")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



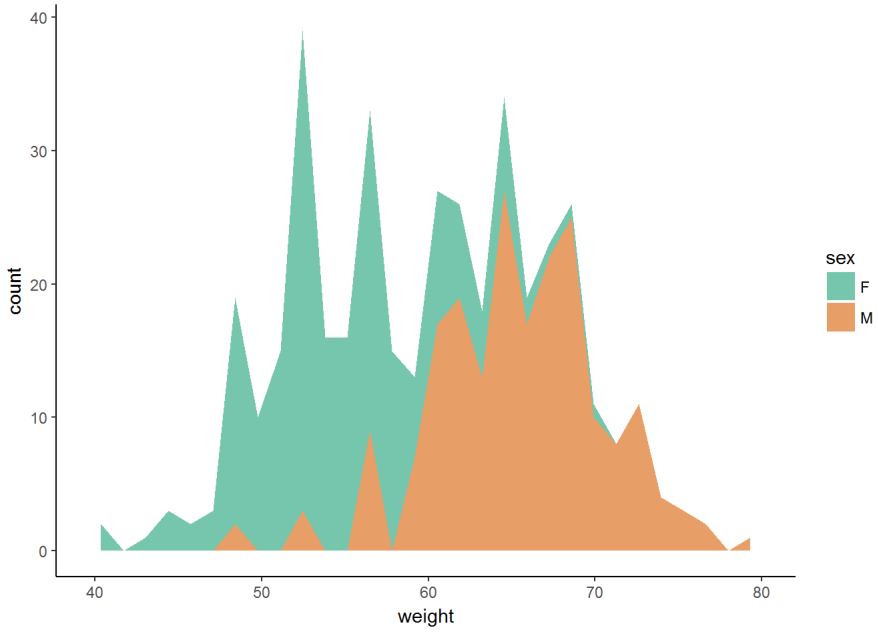

```
# Use custom color palettes
p+scale_fill_manual(values=c("#999999", "#E69F00"))
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



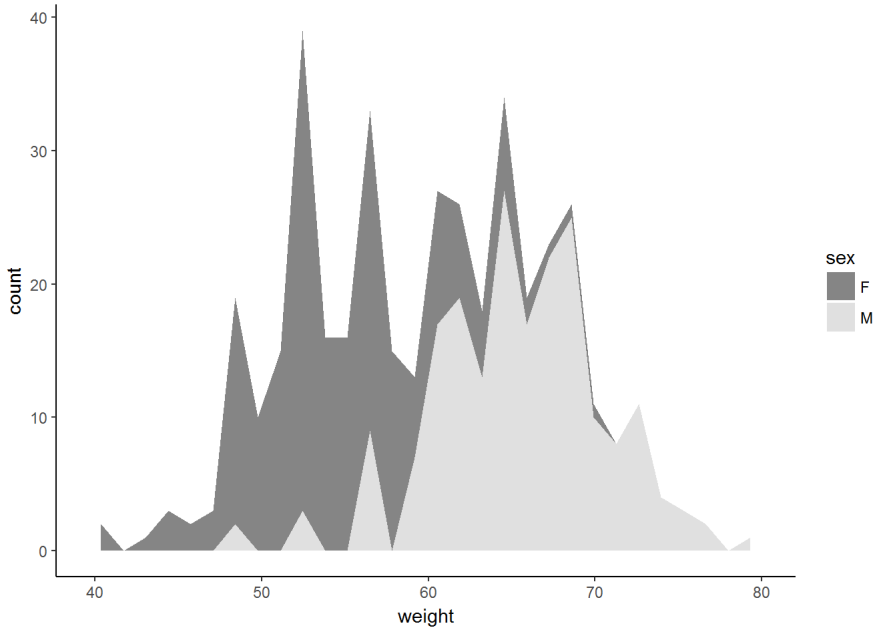

```
# use brewer color palettes
p+scale_fill_brewer(palette="Dark2")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```




```
# Use grey scale
p + scale_fill_grey()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Analysis: for the graphs above, I changed color by different genders, which makes this data more representative and more easily to look at. The bule area that contains males' weights is gathered right next to the pink area, which contains females' weights. And the mean value of each group are located in its own region. Besides, we can also customize the color, change the location of legend.

Next, I am going to introduce another cool area graph, which is stacked area graph.

Stacked Area Graph

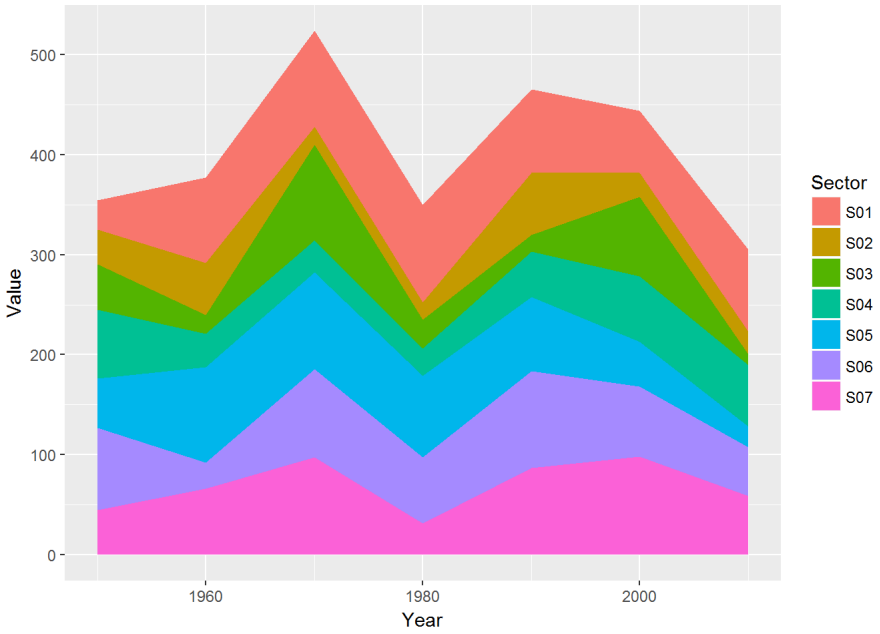
Description: Stacked Area Graphs work in the same way as simple Area Graphs do, except for the use of multiple data series that start each point from the point left by the previous data series.The entire graph represents the total of all the data plotted. Stacked Area Graphs also use the areas to convey whole numbers, so they do not work for negative values. Overall, they are useful for comparing multiple variables changing over an interval.

```
# ggplot2 library
library(ggplot2)

# DATA
set.seed(345)
Sector <- rep(c("S01","S02","S03","S04","S05","S06","S07"),times=7)
Year <- as.numeric(rep(c("1950","1960","1970","1980","1990","2000","2010"),each=7))
Value <- runif(49, 10, 100)
data <- data.frame(Sector,Year,Value)
```

Fisrt, I created a data frame containing seven sectors, years, and values.

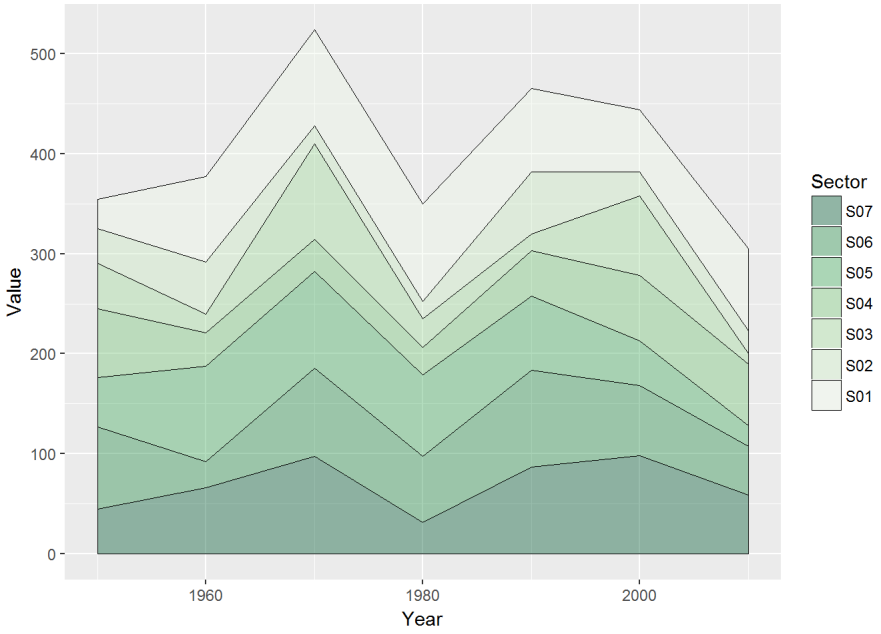
```
ggplot(data, aes(x=Year, y=Value, fill=Sector)) +
    geom_area()
```



Analylis: This is the basic stacked area plot proposed by ggplot2.From the graph, we can tell that different sectors have different values of area. Since each sector is overlaied with each other, it can be easily told that in a certain year which sector has the highest value by comparing each

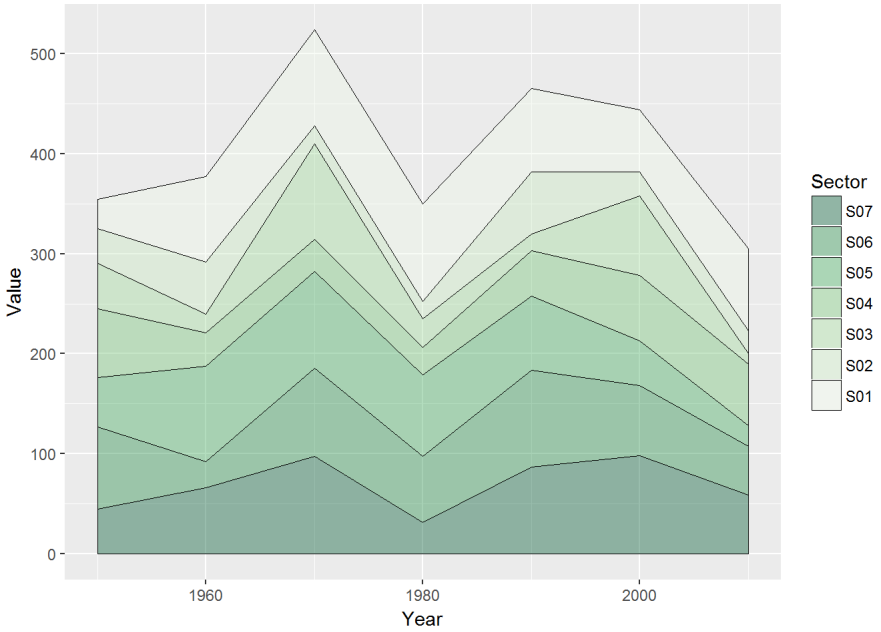
color's height given certain year. For example, in the year 1980 we can tell that Sector 01 has the highest value and Sector 02 has the lowest value.

```
ggplot(data, aes(x=Year, y=Value, fill=Sector)) +
    geom_area(colour="black", size=.2, alpha=.4) +
    scale_fill_brewer(palette="Greens", breaks=rev(levels(data$Sector)))
```



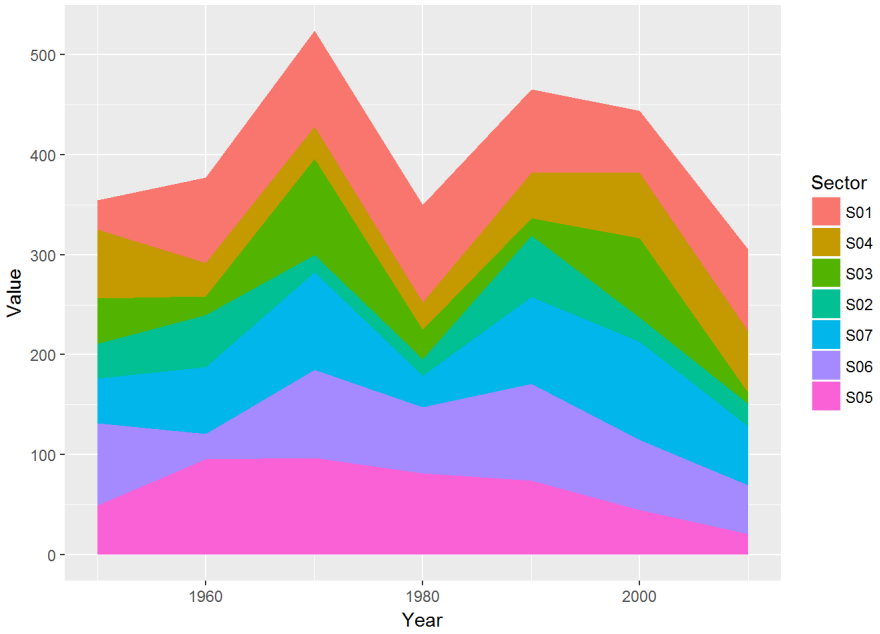
And, You can easily change the color palette.

```
data=data[order(data$Sector, decreasing=T) , ]
ggplot(data, aes(x=Year, y=Value, fill=Sector)) +
    geom_area(colour="black", size=.2, alpha=.4) +
    scale_fill_brewer(palette="Greens", breaks=rev(levels(data$Sector)))
```



To reverse the stacking order, you have to change the order in the initial data frame

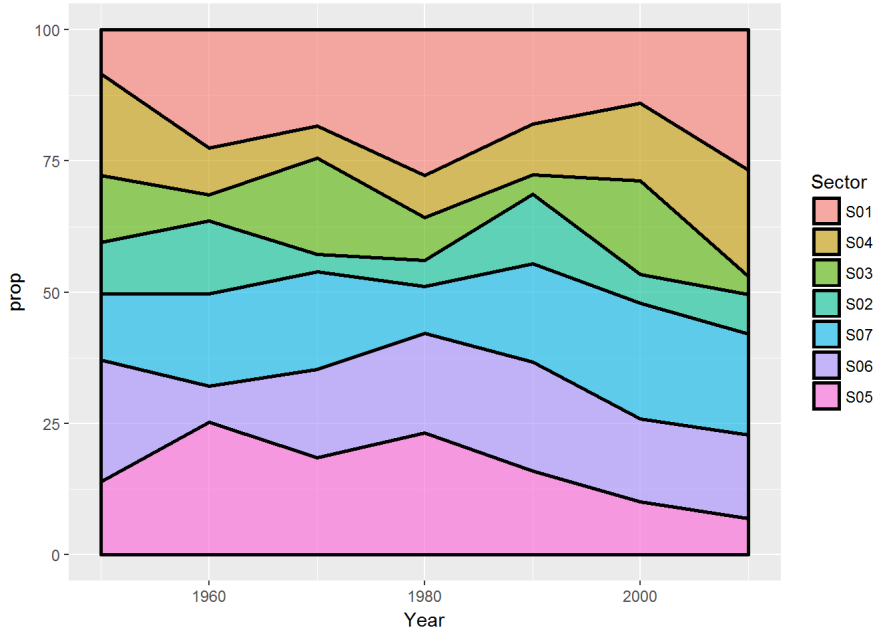
```
data$Sector=factor(data$Sector , levels=levels(data$Sector)[c(1,4,3,2,7,6,5)])
ggplot(data, aes(x=Year, y=Value, fill=Sector)) +
    geom_area()
```



Thus you can give a specific order, still reordering the dataframe as needed.

```
my_fun=function(vec){ as.numeric(vec[3]) / sum(data$Value[data$Year==vec[2]]) *100 }
data$prop=apply(data , 1 , my_fun)

ggplot(data, aes(x=Year, y=prop, fill=Sector)) +
    geom_area(alpha=0.6 , size=1, colour="black")
```



You can also draw a proportional stacked area graph: the sum of each year is always equal to hundred and value of each group is represented through percentages. To make it, you have to calculate these percentages first.

Conclusion:

Throughout this post, We were be able to learn basic area graphs and stacked area graphs. We started off with the basic graphs that hopefully served a good revision. Then we were able to move on to an intermediate level area plot, which is stacked graph. The area graphs gave us some new insights into how we can utilize ggplot and easily visualize the difference among different objects having a certain variable constant.

I hope you will like my post and thank you for reading!

References

http://r-statistics.co/Top50-Ggplot2-Visualizations-MasterList-R-Code.html

https://www.bioinformatics.babraham.ac.uk/training/ggplot_course/Introduction%20to%20ggplot.pdf

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

http://tutorials.iq.harvard.edu/R/Rgraphics/Rgraphics.html

https://www.statistics.com/visualization-in-R-with-ggplot2/?utm_source=R-bloggers.com&utm_medium=blog&utm_campaign=R+courses

http://eriqande.github.io/rep-res-web/lectures/making-maps-with-R.html

http://www.r-graph-gallery.com/136-stacked-area-chart/