# NoteDay3

Diamond

2020/7/8

## chapter 3 Visuation:ggplot2

install first package, ggplot2, by installing tidyverse

Test work place simply.

```r
getwd()
```

```
## [1] "D:/zju/ / /  /DataScienceAndApplications/mynote"
```

### 1.Load data

```r
#National Parks in California
ca <- read_csv("data/ca.csv")
```

```
## Parsed with column specification:
## cols(
##   region = col_character(),
##   state = col_character(),
##   code = col_character(),
##   park_name = col_character(),
##   type = col_character(),
##   visitors = col_double(),
##   year = col_double()
## )
```

```r
#Acadia National Park
acadia <- read_csv("data/acadia.csv")
```

```
## Parsed with column specification:
## cols(
##   region = col_character(),
##   state = col_character(),
##   code = col_character(),
##   park_name = col_character(),
##   type = col_character(),
##   visitors = col_double(),
##   year = col_double()
## )
```

```r
#Southeast US National Parks
se <- read_csv("data/se.csv")
```

```
## Parsed with column specification:
## cols(
##   region = col_character(),
##   state = col_character(),
##   code = col_character(),
##   park_name = col_character(),
##   type = col_character(),
##   visitors = col_double(),
##   year = col_double()
## )
```

```r
#2016 Visitation for all Pacific West National Parks
visit_16 <- read_csv("data/visit_16.csv")
```

```
## Parsed with column specification:
## cols(
##   region = col_character(),
##   state = col_character(),
##   code = col_character(),
##   park_name = col_character(),
##   type = col_character(),
##   visitors = col_double(),
##   year = col_double()
## )
```

```r
#All Nationally designated sites in Massachusetts
mass <- read_csv("data/mass.csv")
```

```
## Parsed with column specification:
## cols(
##   region = col_character(),
##   state = col_character(),
##   code = col_character(),
##   park_name = col_character(),
##   type = col_character(),
##   visitors = col_double(),
##   year = col_double()
## )
```

## 2.A Grammar of Graphics!

ggplot(data = <DATA>) + <GEOM_FUNCTION>(mapping = aes(<MAPPINGS>),stat = <STAT>,position = <POSITION>) + <COORDINATE_FUNCTION> + <FACET_FUNCTION>

You can uniquely describe any plot as a combination of these 7 parameters.

### A simple style

```r
head(ca)
```

```
## # A tibble: 6 x 7
##   region state code  park_name                     type          visitors  year
##   <chr>  <chr> <chr> <chr>                         <chr>            <dbl> <dbl>
## 1 PW     CA    CHIS  Channel Islands National Park National Park     1200  1963
## 2 PW     CA    CHIS  Channel Islands National Park National Park     1500  1964
## 3 PW     CA    CHIS  Channel Islands National Park National Park     1600  1965
## 4 PW     CA    CHIS  Channel Islands National Park National Park      300  1966
## 5 PW     CA    CHIS  Channel Islands National Park National Park    15700  1967
## 6 PW     CA    CHIS  Channel Islands National Park National Park    31000  1968
```

```r
#view(ca) other worksheet will come out
```

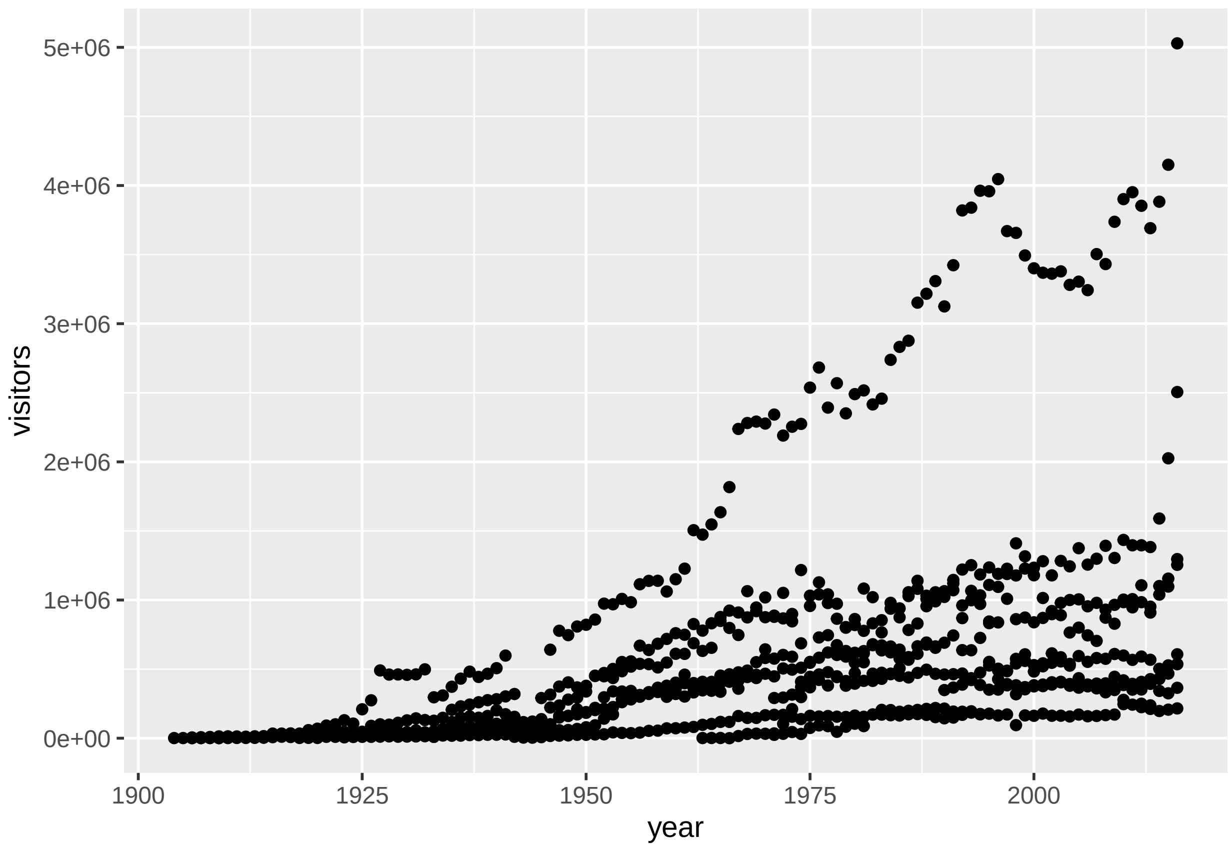
Among the variables in ca are:

1. region, US region where park is located.
2. visitors, the annual visitation for each year

To build a ggplot, we need to:

*use the ggplot() function and bind the plot to a specific data frame using the data argument*

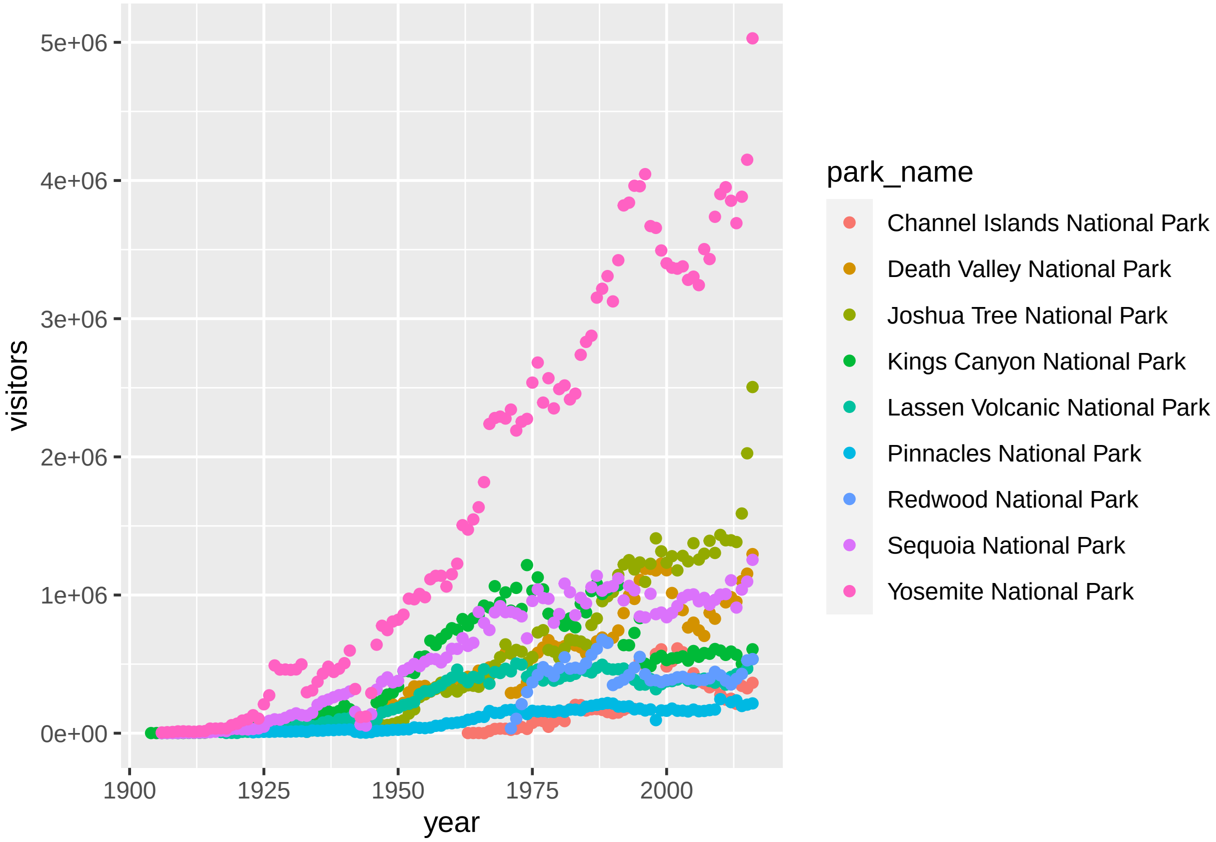
```r
ggplot(data=ca)+
  geom_point(aes(x = year, y = visitors))
```



Notation '+' must be the last in the line (middle also OK)

**Change the style:**

```
ggplot(data = ca) +
geom_point(aes(x = year, y = visitors, color = park_name))
```



Capitalize the x and y axis labels and add a main title to the figure.

Remove that standard gray background using a different theme.

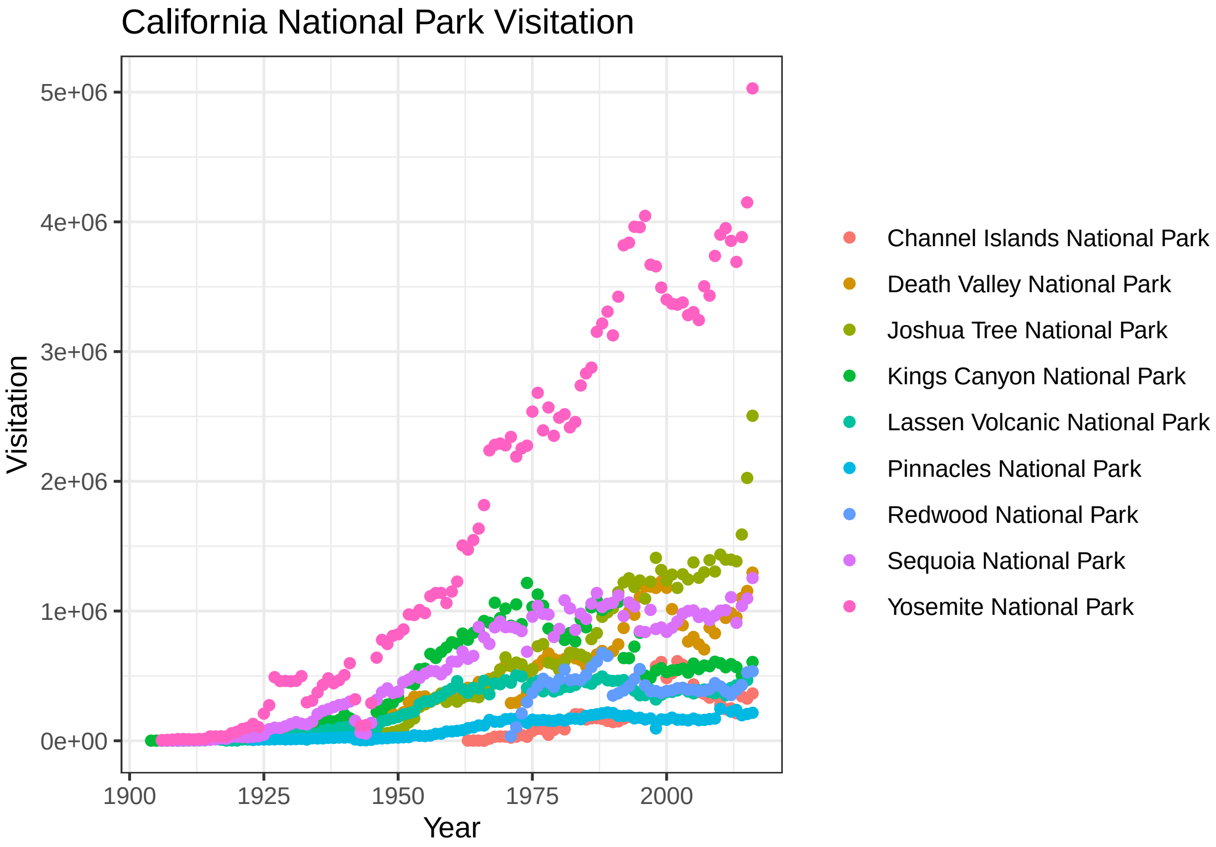
Many themes come built into the ggplot2 package.

*theme_bw*()

Once you start typing theme_ a list of options will pop up.

The last thing to do is remove the legend title.

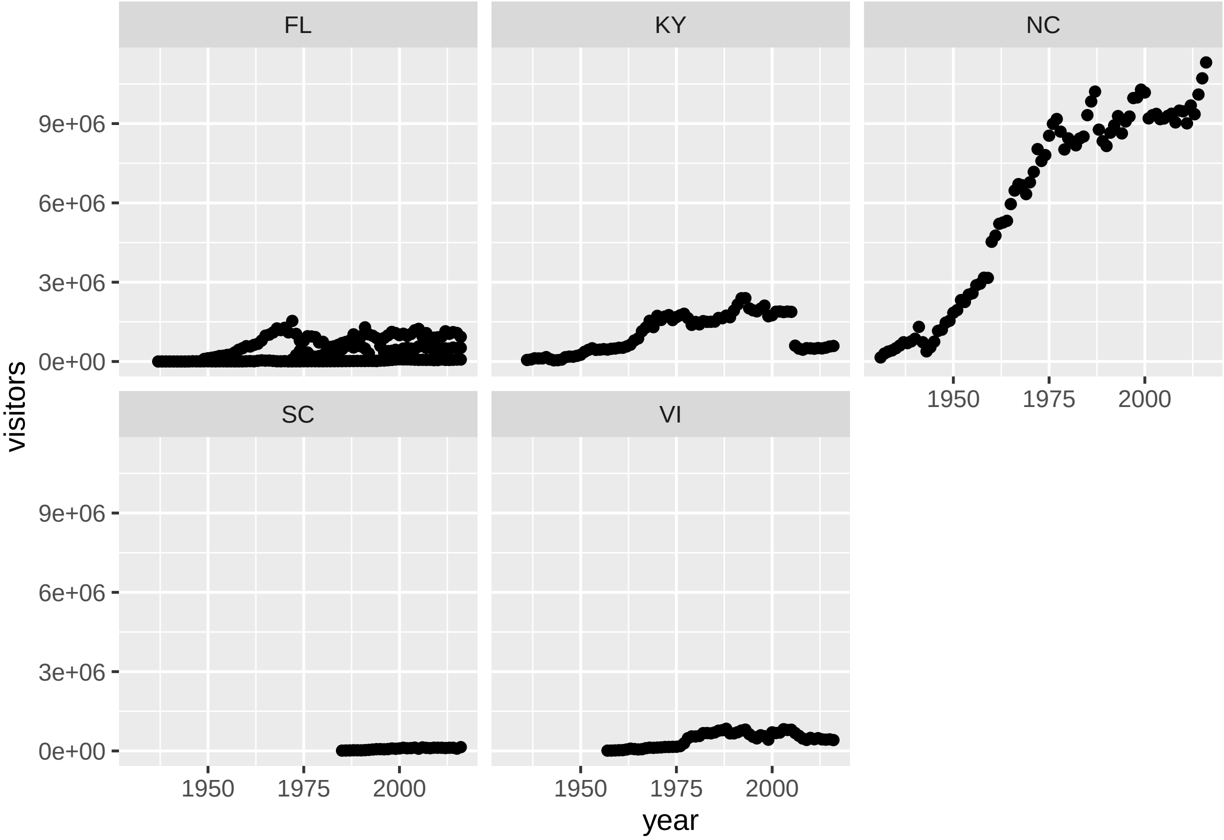
```
ggplot(data = ca) +
geom_point(aes(x = year, y = visitors, color = park_name)) +
labs(x = "Year",
y = "Visitation",
title = "California National Park Visitation") +
theme_bw() +
theme(legend.title=element_blank())
```



### Faceting

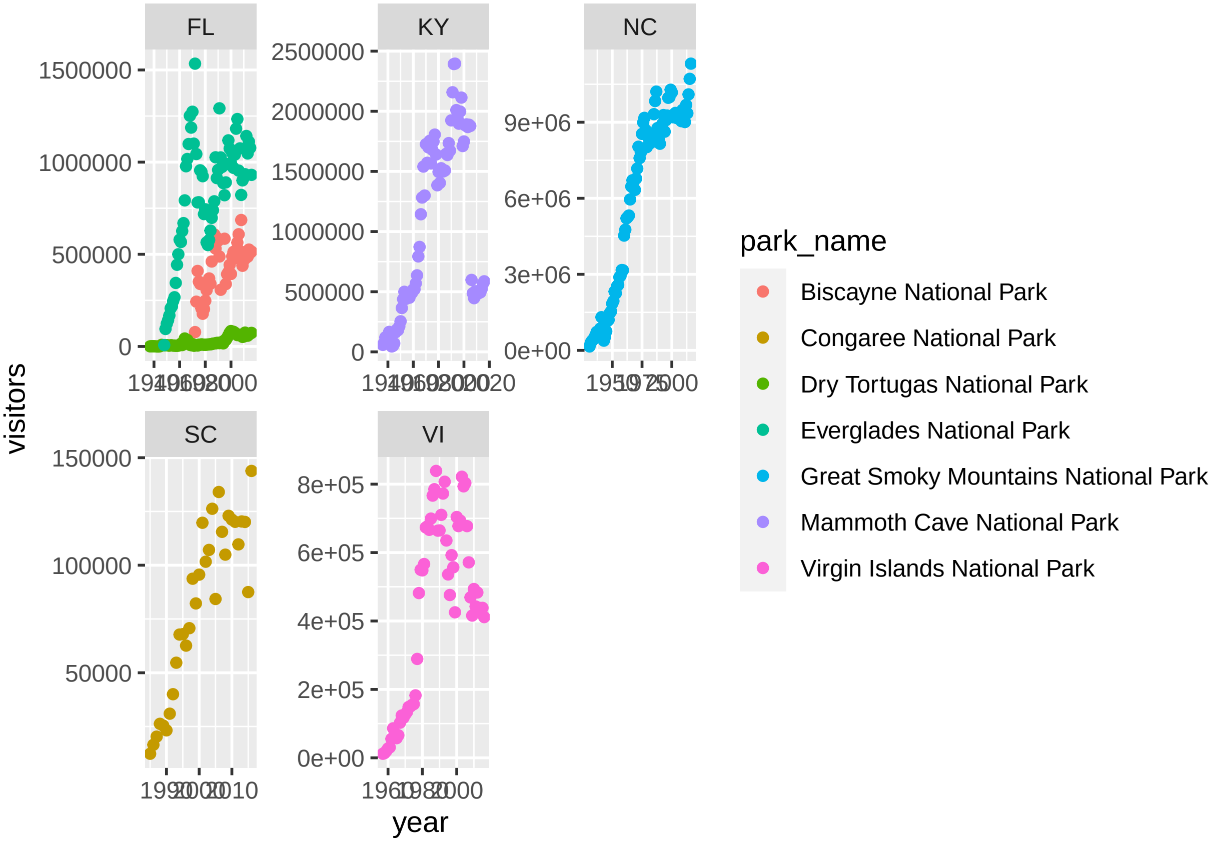
ggplot has a special technique called faceting that allows the user to split one plot into multiple plots based on data in the dataset.

```
ggplot(data = se) +
geom_point(aes(x = year, y = visitors)) +
facet_wrap(~ state)
```



Style change:

```
ggplot(data = se) +
geom_point(aes(x = year, y = visitors, color = park_name)) +
facet_wrap(~ state, scales = "free")
```

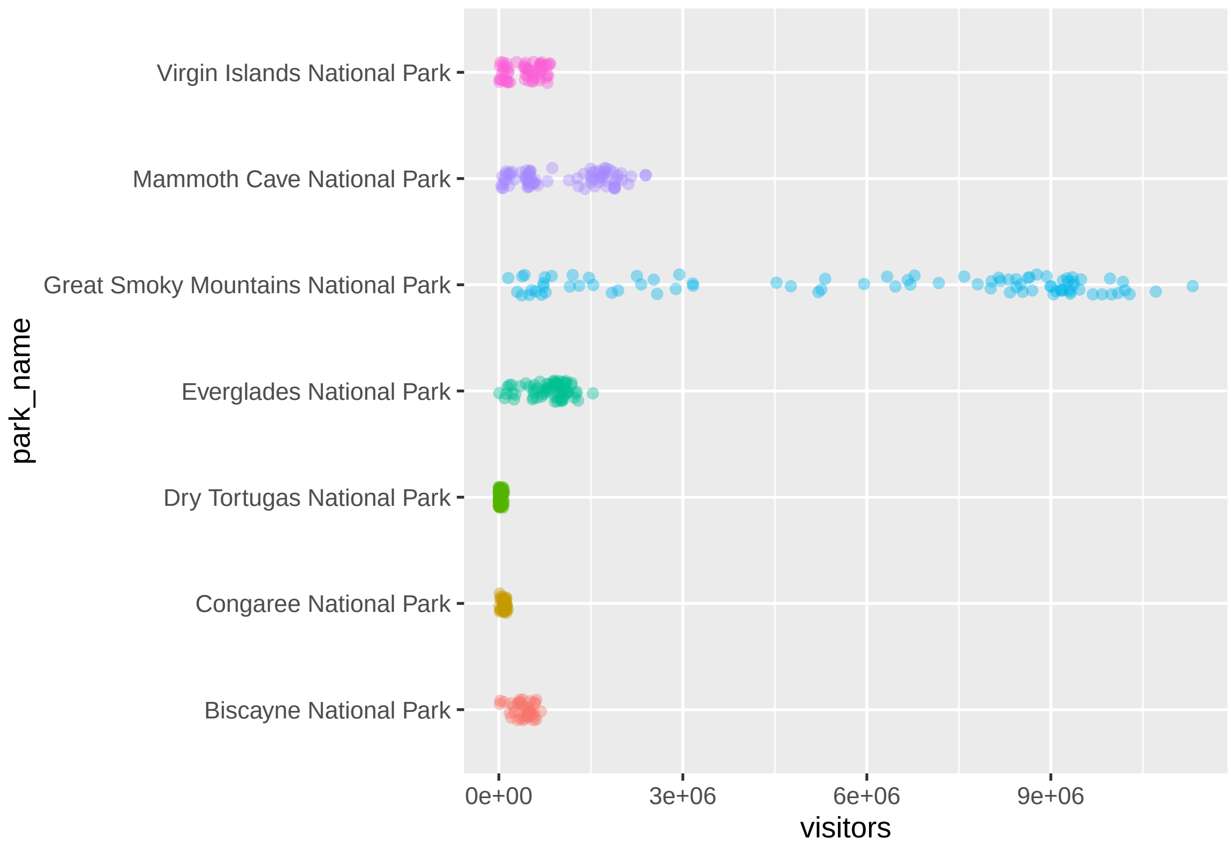


```
#scales = "free" every figure's x-axis is different
```

### Geometric objects (geoms)

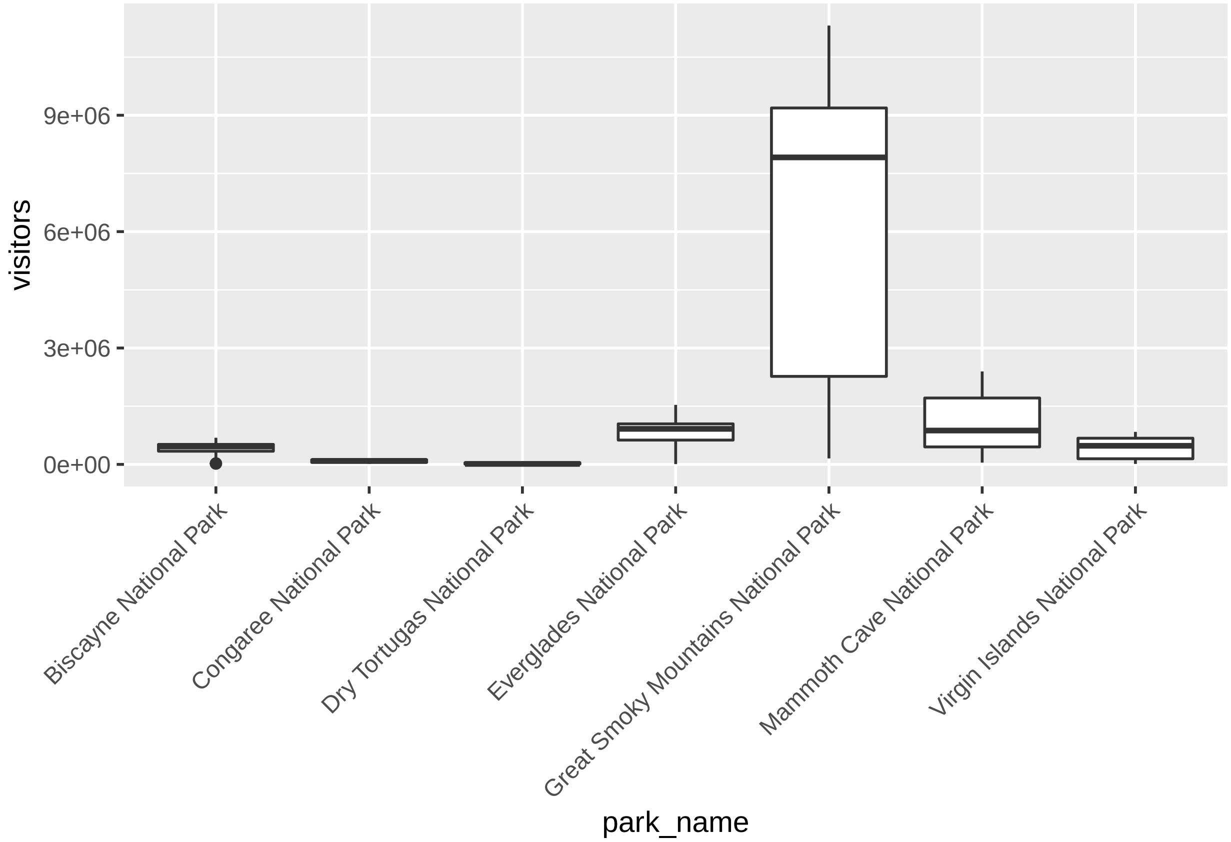
A geom is the geometrical object that a plot uses to represent data. People often describe plots by the type of geom that the plot uses. For example, bar charts use bar geoms, line charts use line geoms, boxplots use boxplot geoms, and so on. Scatterplots break the trend; they use the point geom. You can use different geoms to plot the same data. To change the geom in your plot, change the geom function that you add to ggplot(). Let's look at a few ways of viewing the distribution of annual visitation (visitors) for each park (park_name).

```
ggplot(data = se) +
geom_jitter(aes(x = park_name, y = visitors, color = park_name),
width = 0.1,
alpha = 0.4) +
coord_flip() +
theme(legend.position = "none")
```

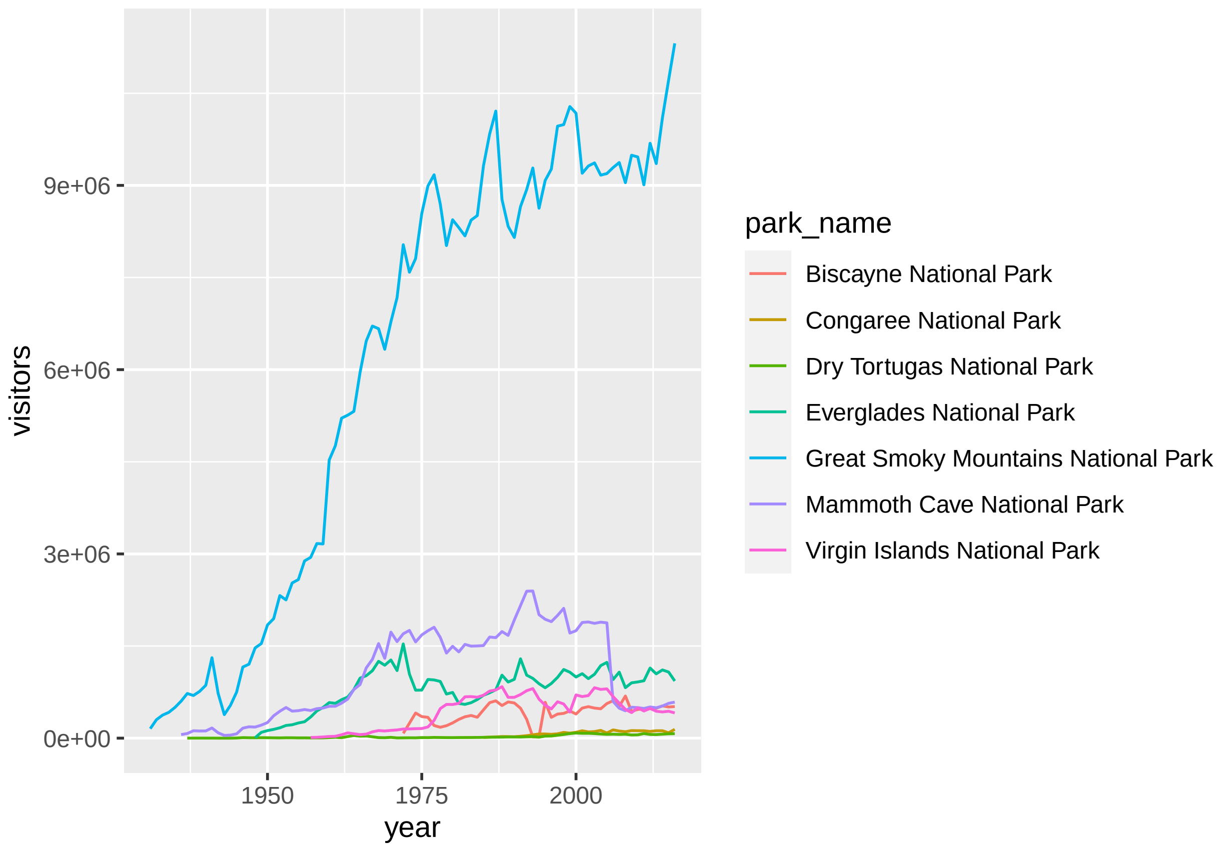


```
#alpha : transparent rate
```

```
ggplot(se, aes(x = park_name, y = visitors)) +
geom_boxplot() +
theme(axis.text.x = element_text(angle = 45, hjust = 1))
```



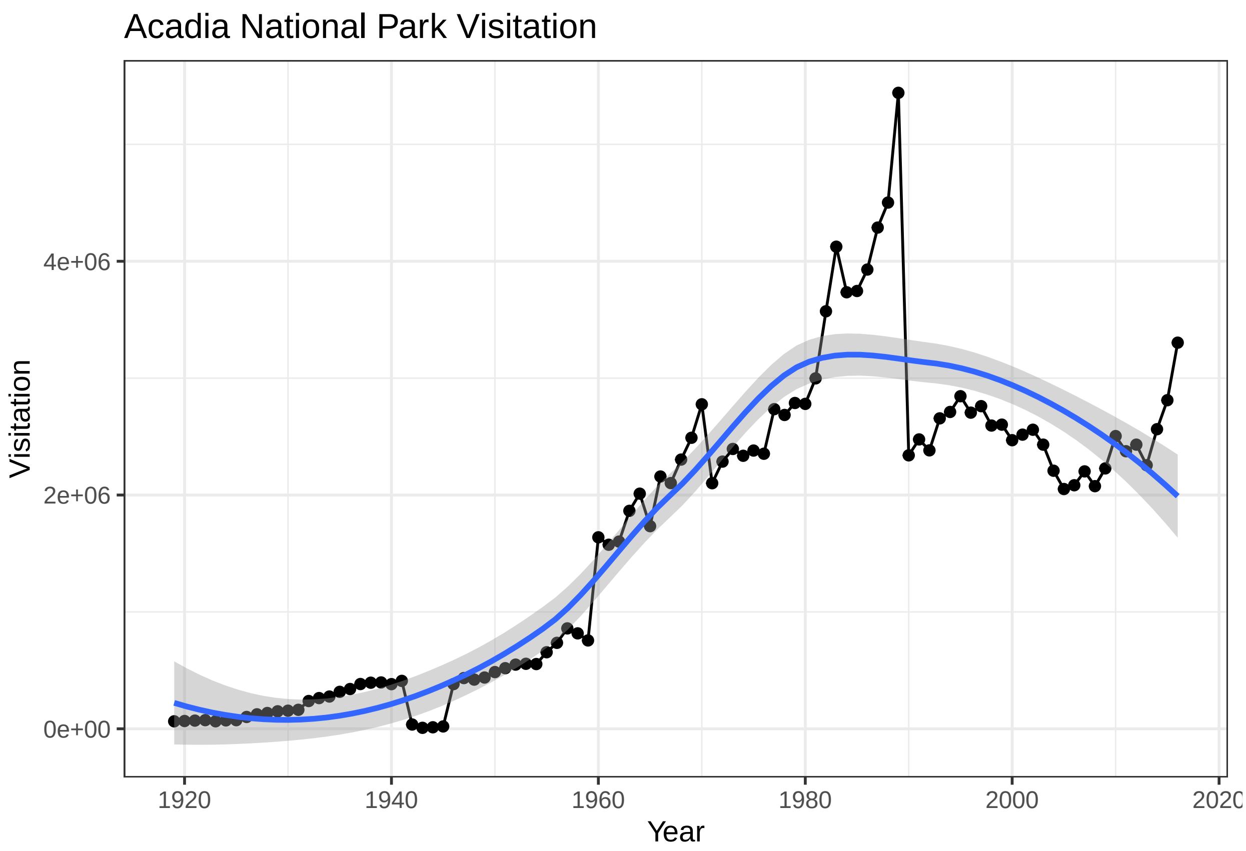
```r
ggplot(se, aes(x = year, y = visitors, color = park_name)) +
geom_line()
```



geom_smooth allows you to view a smoothed mean of data. Here we look at the smooth mean of visitation over time to Acadia National Park:

```
ggplot(data = acadia) +
geom_point(aes(x = year, y = visitors)) +
geom_line(aes(x = year, y = visitors)) +
geom_smooth(aes(x = year, y = visitors)) +
labs(title = "Acadia National Park Visitation",
y = "Visitation",
x = "Year") +
theme_bw()
```

```
## `geom_smooth()` using method = 'loess' and formula 'y ~ x'
```



### Bar chats

```
png('figures/test.png')
ggplot(data = visit_16, aes(x = state)) +
geom_bar()
dev.off()
```

```
## pdf
##   2
```

```
pdf('figures/test.pdf')
ggplot(data = visit_16, aes(x = state)) +
geom_bar()
dev.off()
```

```
## pdf
##   2
```

```
#   PDF PNG
```