Short R Tutorial

Vivian Ngo

2020-03-21

Contents

Ways to view or summarize a dataframe

```
# I am just creating some sample data here
n <- 1000
time<- 1:n
lat<- runif(n,-90,90)
long<- rbeta(n/2,0.5,0.5)
long<- c(long,rnorm(n/2,mean=4,sd=2) )
df <- data.frame(time,long,lat)
```

```
# you can see some of the data
# remember to NOT print out a whole dataframe!
head(df)
```

```
##   time      long       lat
## 1    1 0.9654910 -38.94197
## 2    2 0.8381534 -74.51122
## 3    3 0.8980769 -48.94307
## 4    4 0.9508913 -58.42503
## 5    5 0.1363910 -69.00644
## 6    6 0.3021185  49.83711
```

```
# you can also "glimpse" the data
# remember you need hte dplyr package
library(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
```

```
glimpse(df)
```

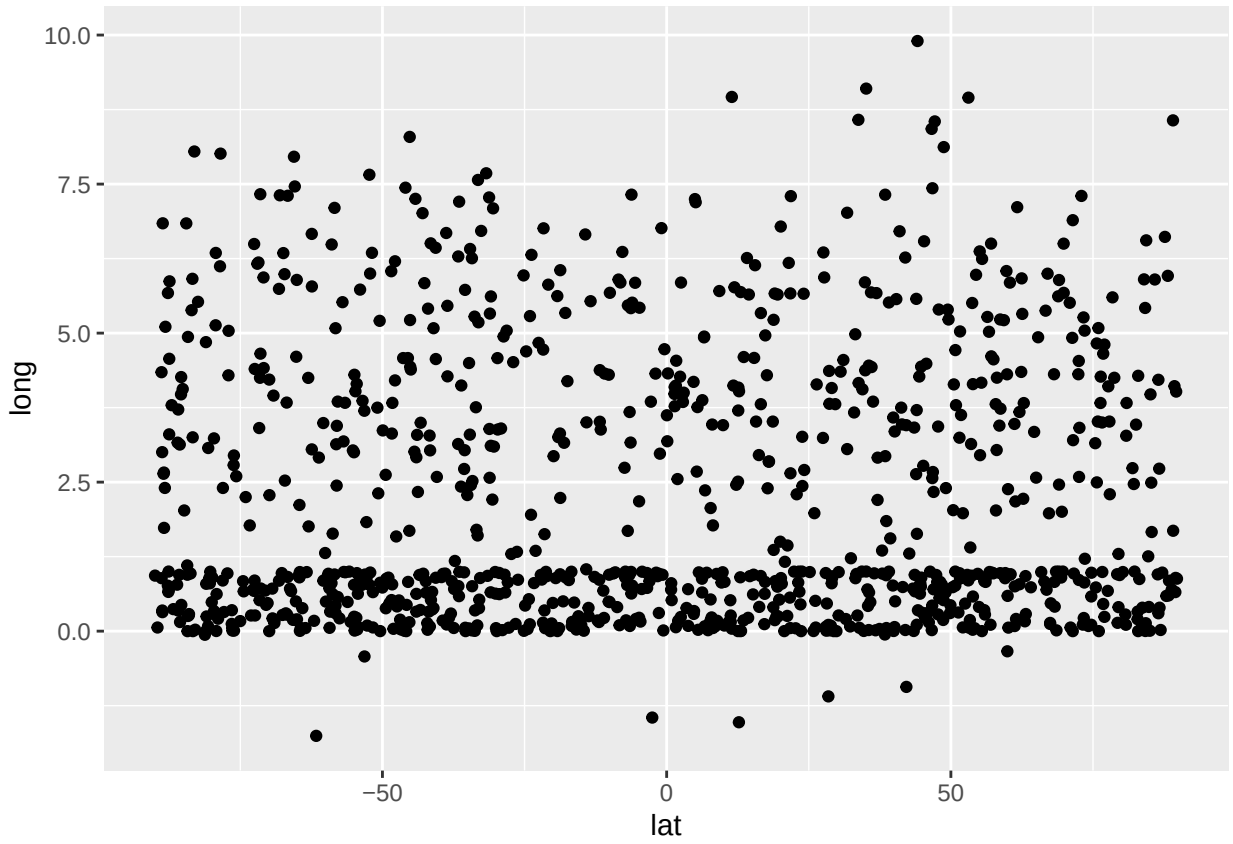
```
## Observations: 1,000
## Variables: 3
## $ time <int> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18,...
## $ long <dbl> 0.96549098, 0.83815336, 0.89807685, 0.95089132, 0.13639096, 0....
## $ lat  <dbl> -38.941973, -74.511222, -48.943068, -58.425032, -69.006441, 49...
```

```
summary(df)
```

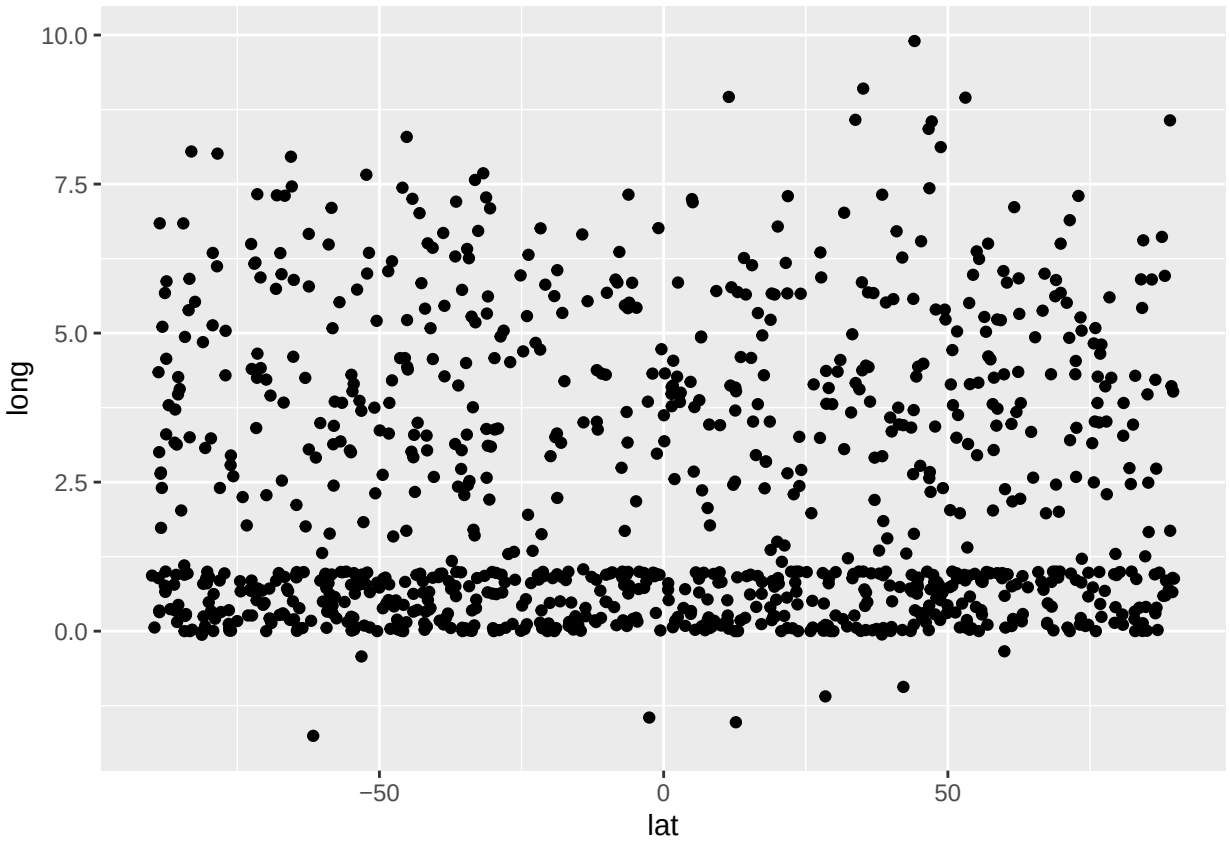
```
##       time             long              lat
##  Min.   :   1.0   Min.   :-1.7565   Min.   :-89.9996
##  1st Qu.: 250.8   1st Qu.: 0.4301   1st Qu.:-46.4538
##  Median : 500.5   Median : 0.9909   Median : -0.1701
##  Mean   : 500.5   Mean   : 2.2923   Mean   : -1.3656
##  3rd Qu.: 750.2   3rd Qu.: 4.0884   3rd Qu.: 44.6490
##  Max.   :1000.0   Max.   : 9.9001   Max.   : 89.7784
```

Showing observations by location, such as using longitude and latitude

```
# how to do a map using long and lat
# so let's say you have coordinates and you just want to plot points
# at the coordinates
# you can do this:
library(ggplot2)
df %>% ggplot(aes(x=lat, y=long)) + geom_point()
```




```
# you can also code it like this:
ggplot(data=df, aes(x=lat, y=long)) + geom_point()
```

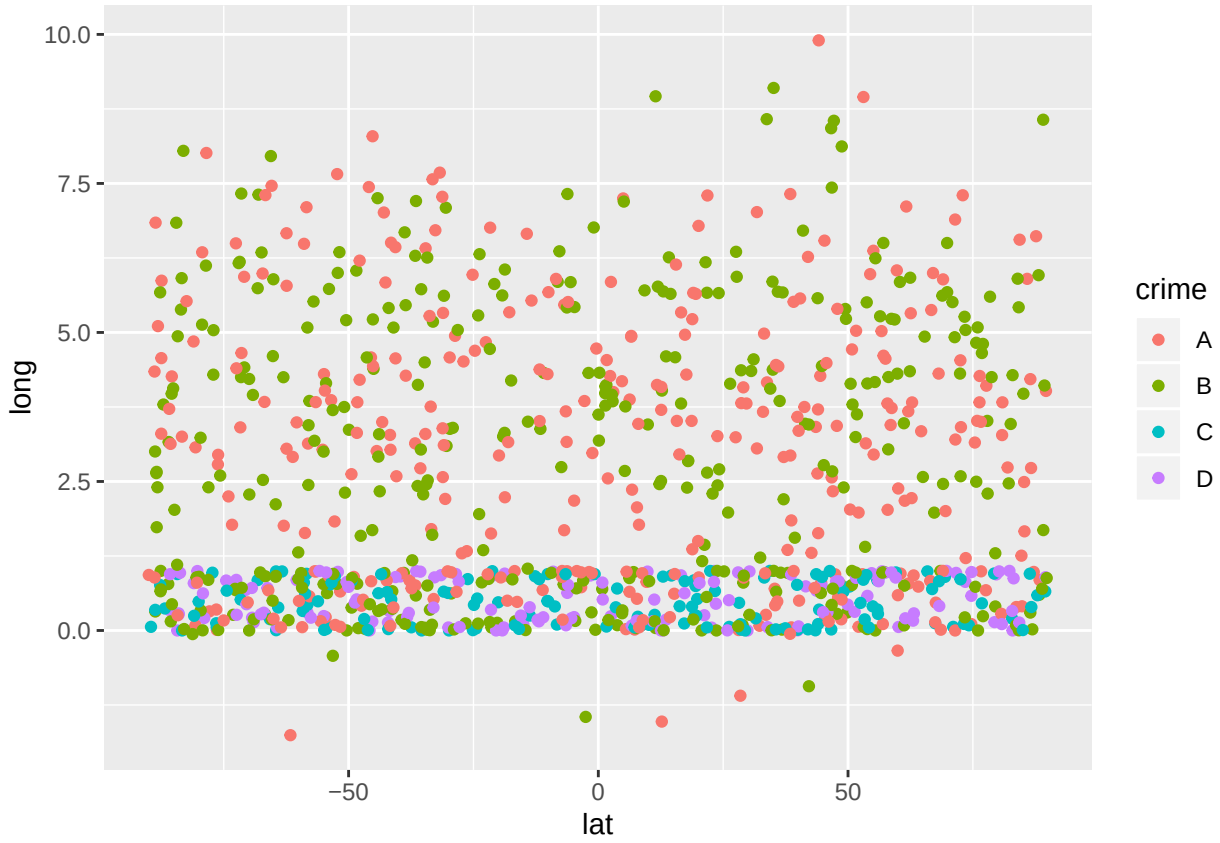


Different colors, shapes, and sizes for different types of observations

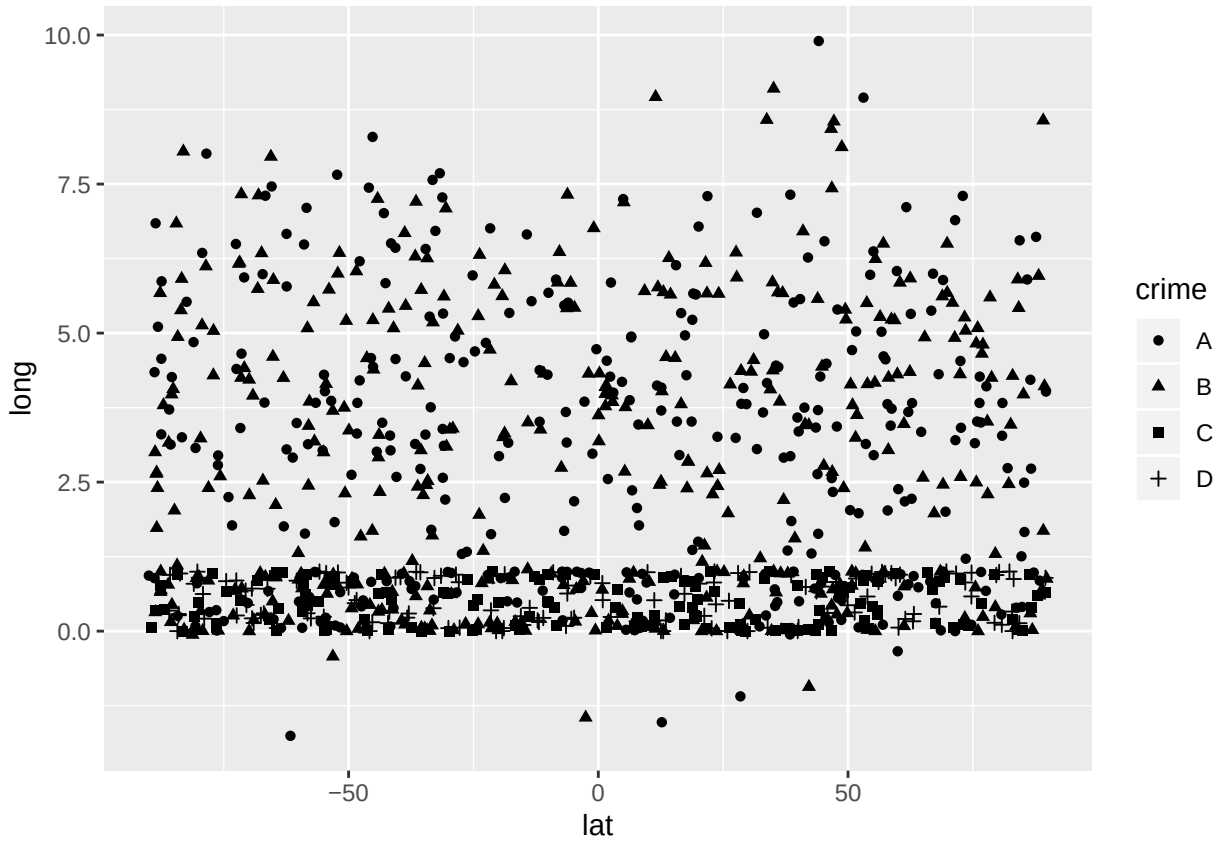
```
# let's say you want to further distinguish these locations by a type of crime
# or something

# e.g. let's add in some crime data where crime can be A,B,C, or D
crime<- sample(c("A","B","C","D"), size=n/2, replace=TRUE)
# for the second half, I want to make the crimes just A's and B's
# so we can see some differences in the graph
crime<-c(crime, sample(c("A","B"), size=n/2, replace=TRUE))
df$crime<- crime
# let's also add in some numerical data about, for example, how much the crime
# cost for an incident at that location
cost<- runif(n, 0 ,9000) # this generates n random numbers between 0 and 9000
df$cost <- cost

head(df) # let's see what it looks like now
```

```
##   time      long       lat crime      cost
## 1    1 0.9654910 -38.94197     B 4949.8777
## 2    2 0.8381534 -74.51122     D 4272.9071
## 3    3 0.8980769 -48.94307     A 6713.8392
## 4    4 0.9508913 -58.42503     C 1228.2722
## 5    5 0.1363910 -69.00644     D 2726.0172
## 6    6 0.3021185  49.83711     B  261.8778
```

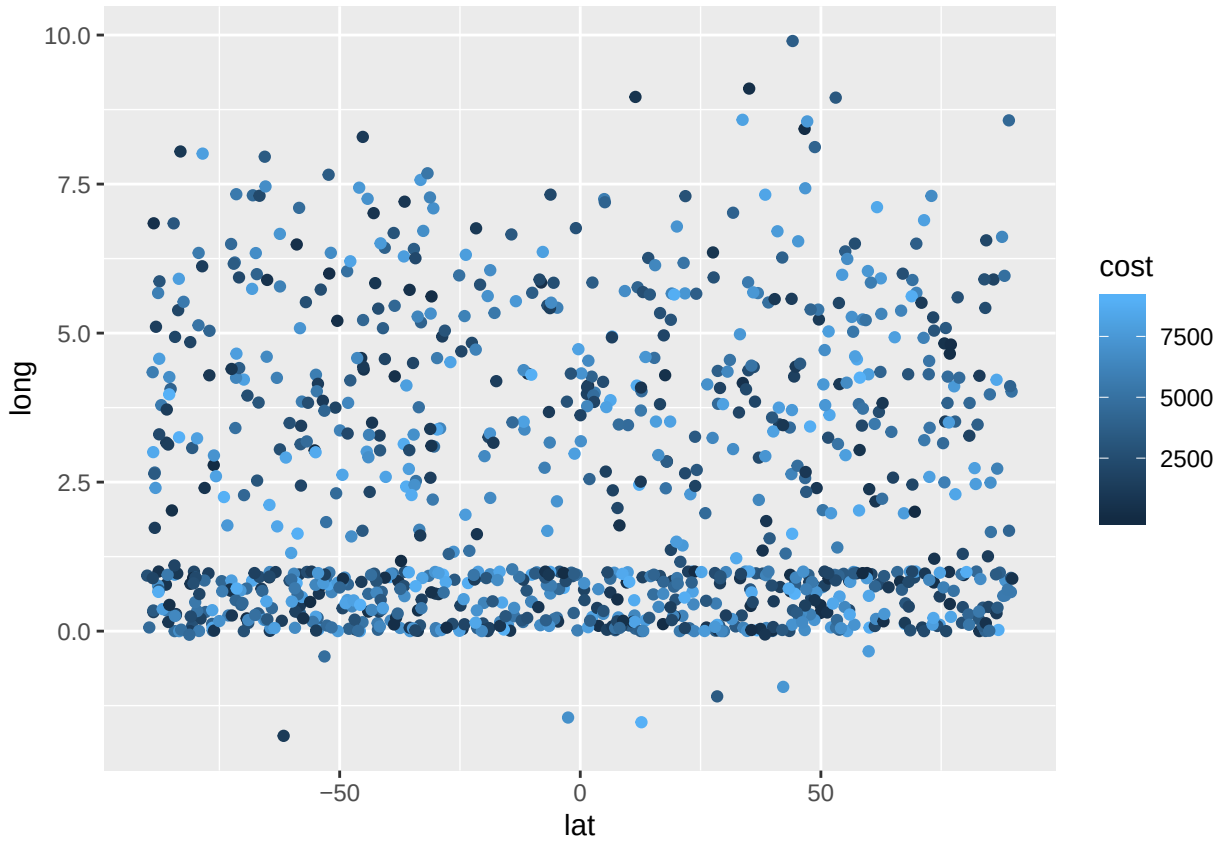
```
# We can differ the points based on categorical variables such as crime type

# color the plots by crime type
df %>% ggplot(aes(x=lat, y=long, color=crime)) +
  geom_point()
```



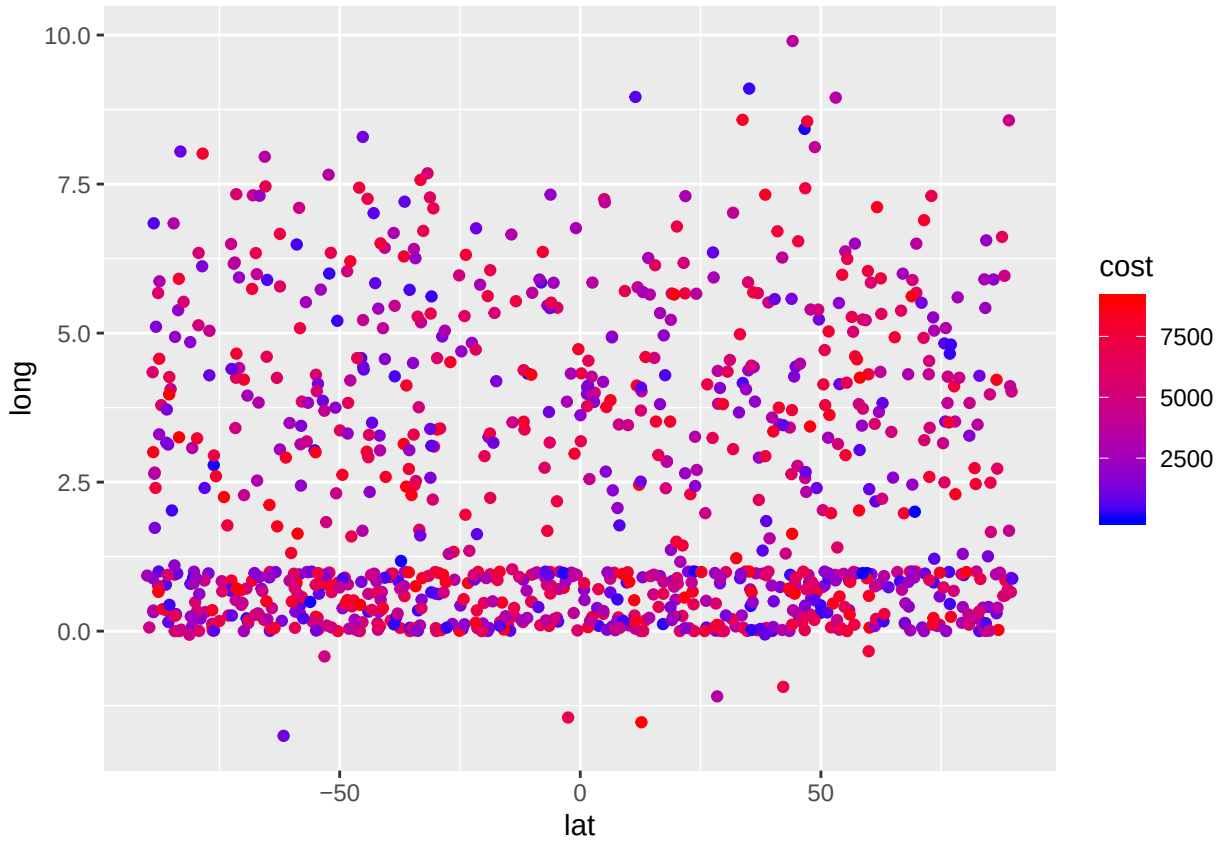

```
# change the shape of the points by crime type
df %>% ggplot(aes(x=lat, y=long, shape=crime)) +
  geom_point()
```



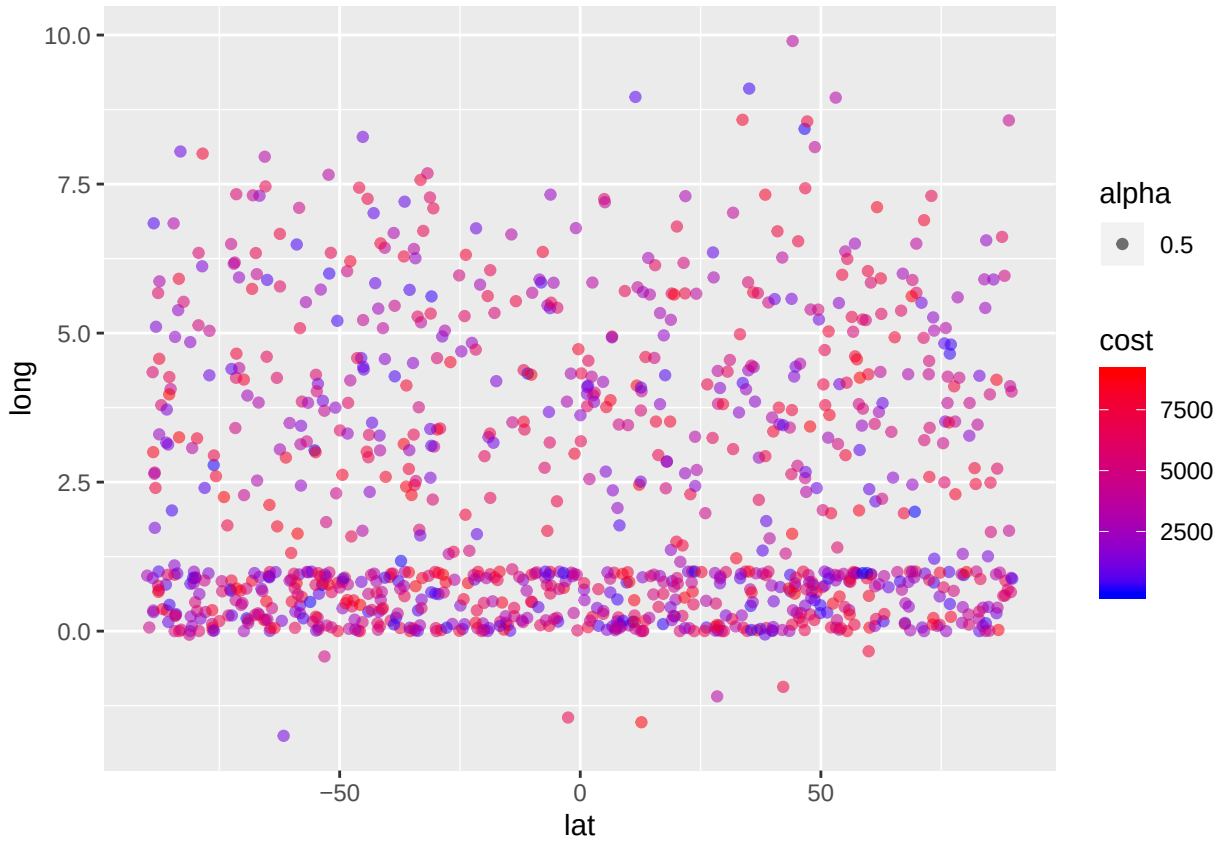

```
# What about numerical values, such as cost?

# we can change the color of the points based on the cost
df %>% ggplot(aes(x=lat, y=long, color=cost)) +
  geom_point()
```



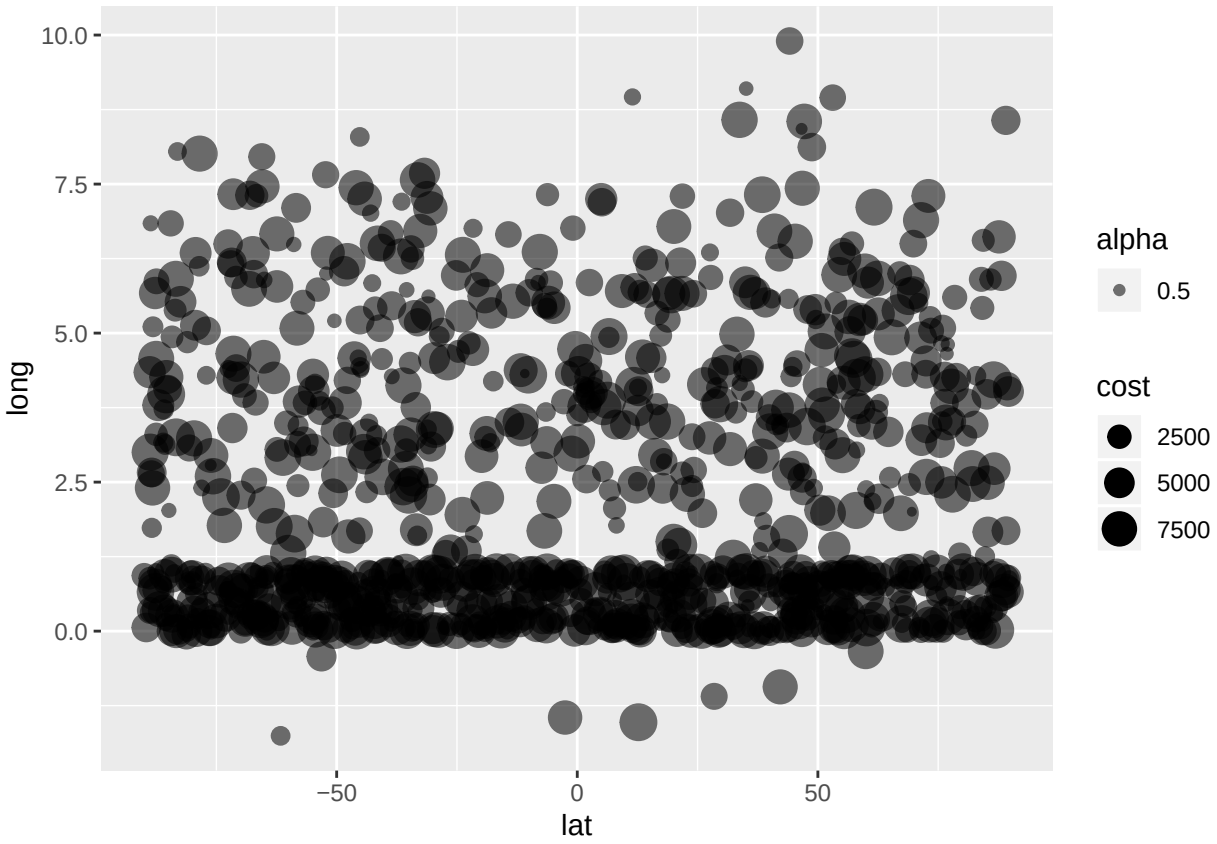

```
# we can also use different color schemes
df %>% ggplot(aes(x=lat, y=long, color=cost)) +
  geom_point() +
  scale_color_gradient(low="blue", high="red")
```




```
# visit http://www.sthda.com/english/wiki/ggplot2-colors-how-to-change-colors-automatically-and-manually
# for more information about color schemes

# we can change the transparency of the points using "alpha"
df %>% ggplot(aes(x=lat, y=long, color=cost, alpha=0.5)) +
  geom_point() +
  scale_color_gradient(low="blue", high="red")
```

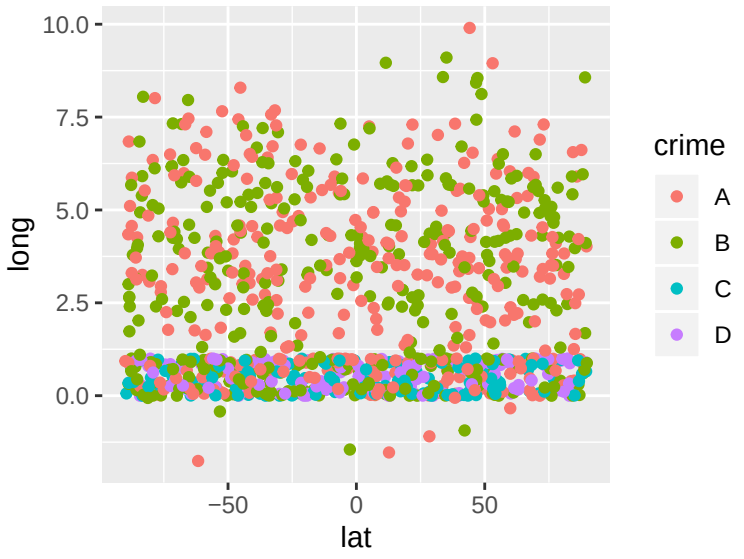



```
# we can change the size of the points
df %>% ggplot(aes(x=lat, y=long, alpha=0.5, size=cost)) +
  geom_point() +
  scale_color_gradient(low="blue", high="red")
```



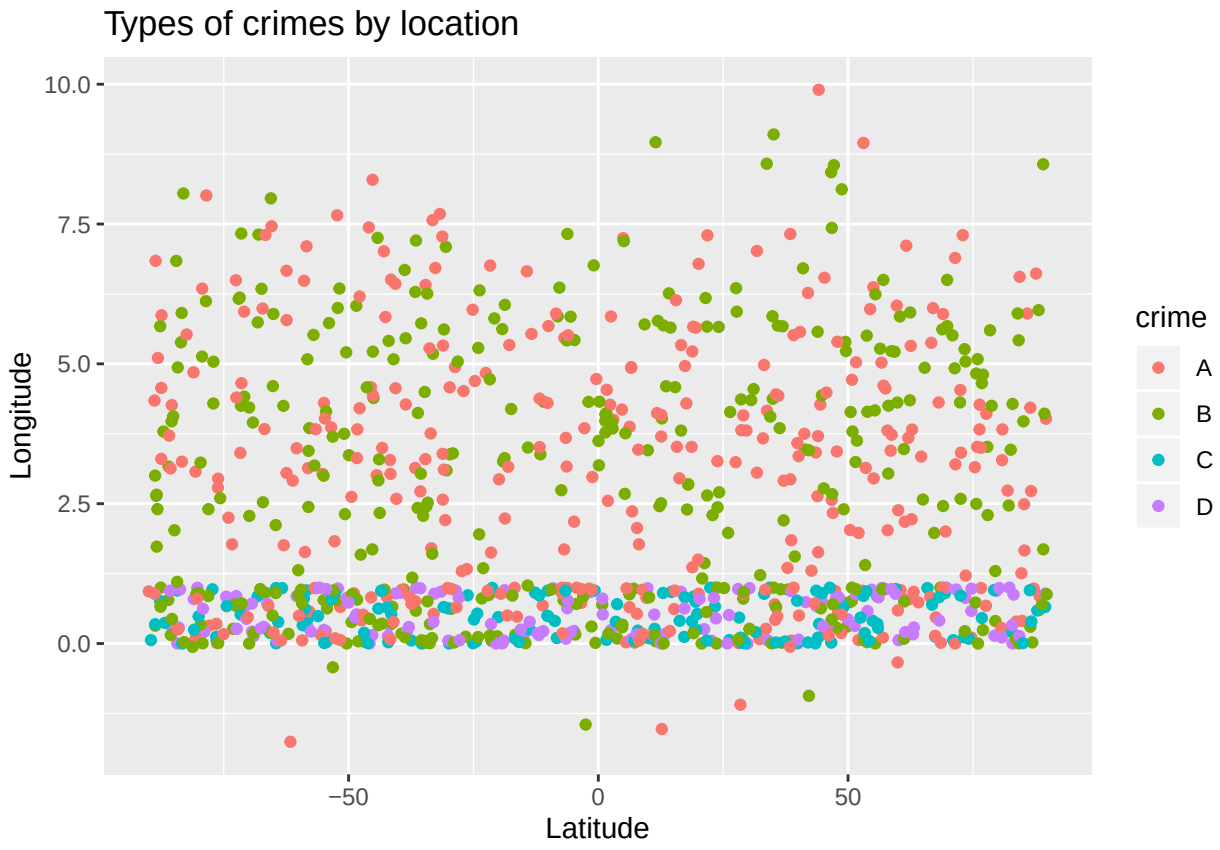
Changing the display size of your figures

```
# changing the size of your figures.
# to control the size of your figures, you can use fig.width and fig.height in the chunk header.

# for this chunk, I have put:
# ```{r fig.width=4, fig.height=3, fig.align="center"}
# you can adjust fig.width and fig.height to your liking
df %>% ggplot(aes(x=lat, y=long, color=crime)) +
  geom_point()
```



Don't forget to label and title your plots!

```
df %>% ggplot(aes(x=lat, y=long, color=crime)) +
  geom_point() +
  labs(title="Types of crimes by location") + # title the graph
  xlab("Latitude") +  # x axis label
  ylab("Longitude") # y axis label
```



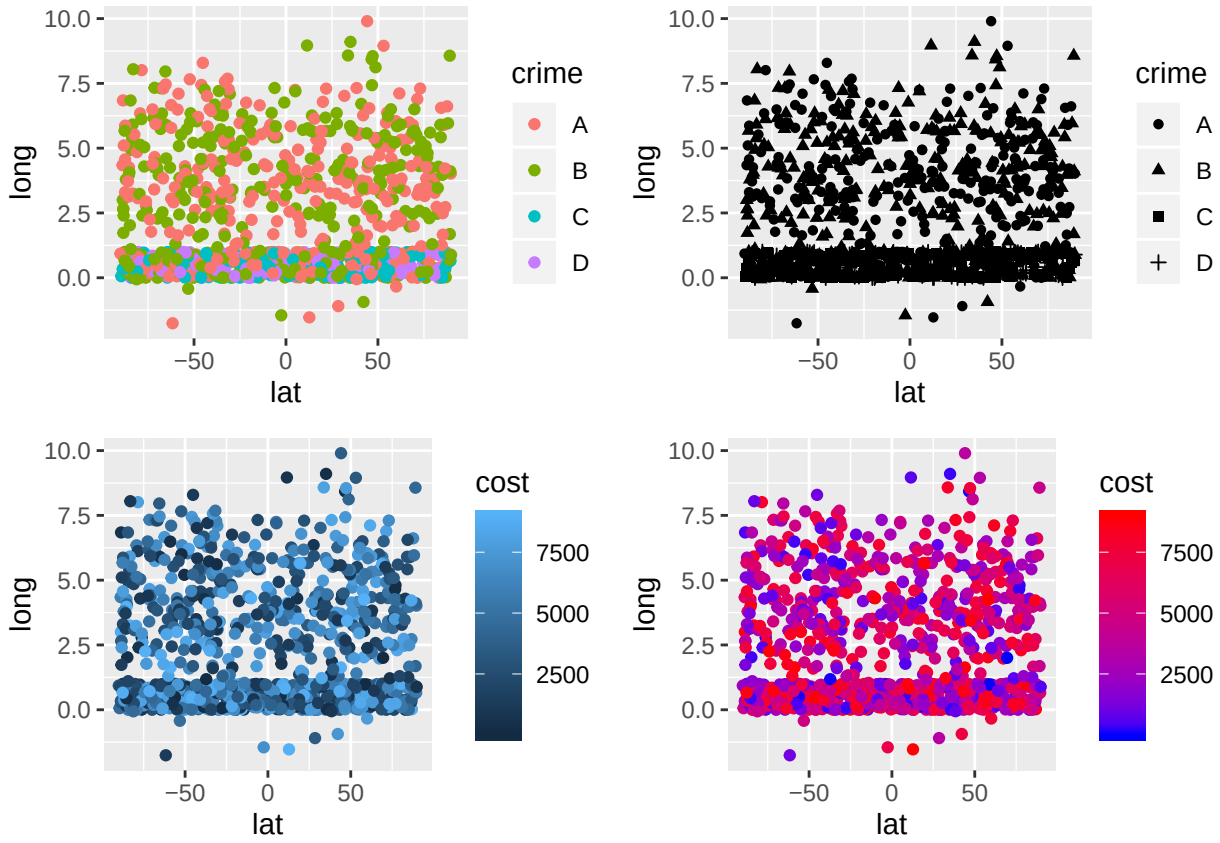
Showing multiple plots at once

```
# this might be outside the scope of this course, but...
# you can use something called grid.arrange to show multiple plots at once.
# you just need to name your plots first.
# e.g.:
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
p1<- df %>% ggplot(aes(x=lat, y=long, color=crime)) +
  geom_point()
p2<- df %>% ggplot(aes(x=lat, y=long, shape=crime)) +
  geom_point()
p3<-df %>% ggplot(aes(x=lat, y=long, color=cost)) +
  geom_point()
p4<-df %>% ggplot(aes(x=lat, y=long, color=cost)) +
  geom_point() +
  scale_color_gradient(low="blue", high="red")
# nrow specifies the number of rows you want in this grid
grid.arrange(p1,p2,p3,p4,nrow=2)
```



Notes

- Not all plots work well for your data. In some cases, changing the color may be useful but not the size. Sometimes it will be the opposite. You should pick the graph that you use based on what is more visually helpful to the audience.
- Always title and label your graphs.
- When you have knitted your work, remember to look over the document to make sure that nothing is out of place.