Week 3: Data Visualization

DSC 365: Introduction to Data Science

2024-09-03

Recommended Reading:

- *Modern Data Science with R* Ch. 2: Data Visualization
- *Modern Data Science with R* Ch. 3: A Grammar for Graphics

`ggplot2`

In the lecture, we showed that statistics alone may lead to a misunderstanding of the data. Therefore, when working with new data, we should always make some visualizations to help us understand the data. A common way for plotting in R today is through `ggplot2`.

`ggplot2` is an R package (located in `tidyverse`) for "decoratively creating graphics"

- https://ggplot2.tidyverse.org/reference/

```r
library(tidyverse)
```

Example: Hate crimes and income inequality

A FiveThirtyEight article published in 2017 claimed that higher rates of hate crimes were tied to greater income inequality.

https://fivethirtyeight.com/features/higher-rates-of-hate-crimes-are-tied-to-income-inequality/

- FiveThirtyEight publishes their data sets - let's investigate for ourselves.
- Data set is posted in BlueLine. Download this data set, and save it to your computer.

Follow these steps to read the data into RStudio:

1. In the Environment tab, click "Import Dataset". Since this is a CSV document, you want to import a text file.

2. Navigate to your CSV data set. Make sure that the first row contains column names.
3. Import the data.

Another way to do this:

1. Put the data file next to the Rmd file.
2. In the console, print`read.csv(hate.crimes.csv)` See ?read.csv for more information about this function

```
hate_crimes <- read.csv("./data/hate_crimes.csv")
#glimpse(hate_crimes)
#head(hate_crimes)
```

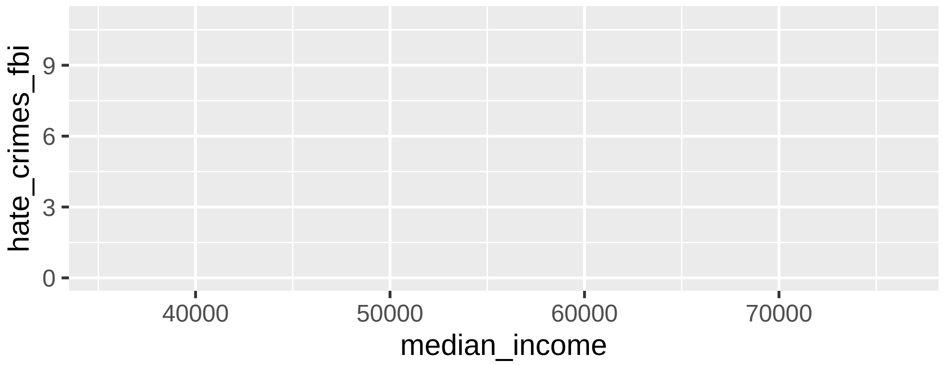
A simple plot?

Basic Format of a Plot:

`data and aesthetics + plot type + options`

The data, aesthetics, and plot type are necessary to create a plot. For example, below is what happens when we just specify the data and aesthetics.

```
ggplot(hate_crimes, aes(x=median_income, y=hate_crimes_fbi))
```



Variable type

After specifying the data and aesthetics, we need to decide the plot type. In order to do that, we need to know the variable type(s). There are two different ways to distinguish the variables.

By function:

1. response variable
2. explanatory variable

By value type:

1. continuous variable
2. categorical variable

Type of plots

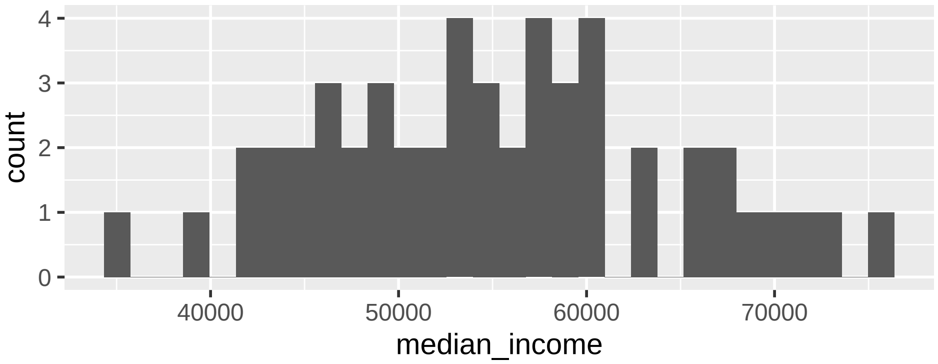
1. Visualize one continous variable.

Usually for the response variable using histograms and density plots

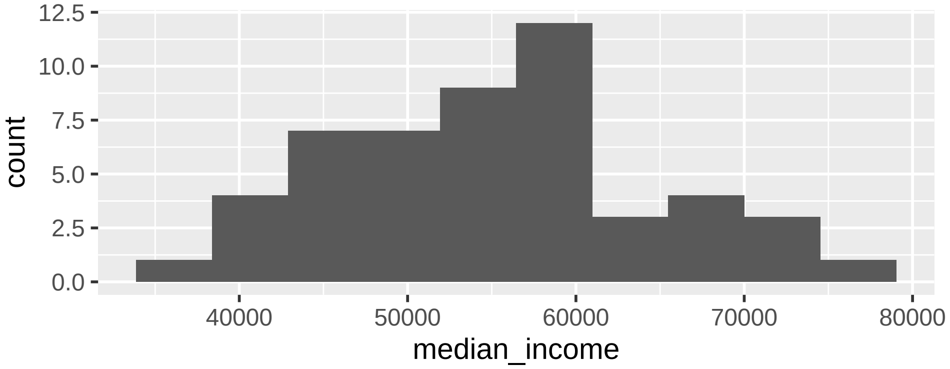
(a) Histograms

```
ggplot(hate_crimes, aes(x=median_income)) +
  geom_histogram()
```

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

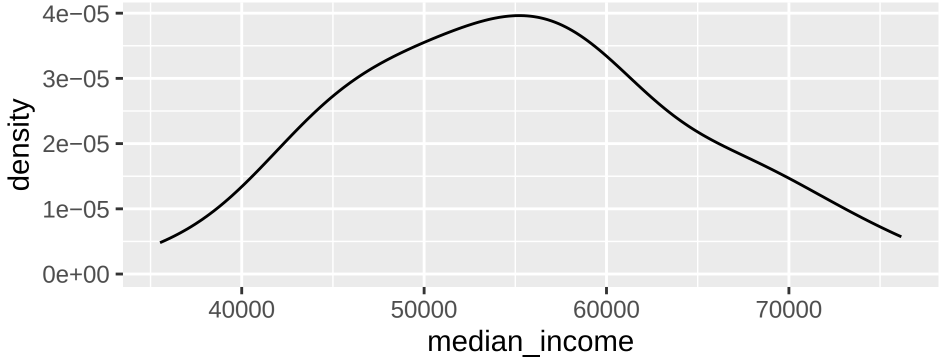


```
ggplot(hate_crimes, aes(x=median_income)) +
  geom_histogram(bins = 10)
```

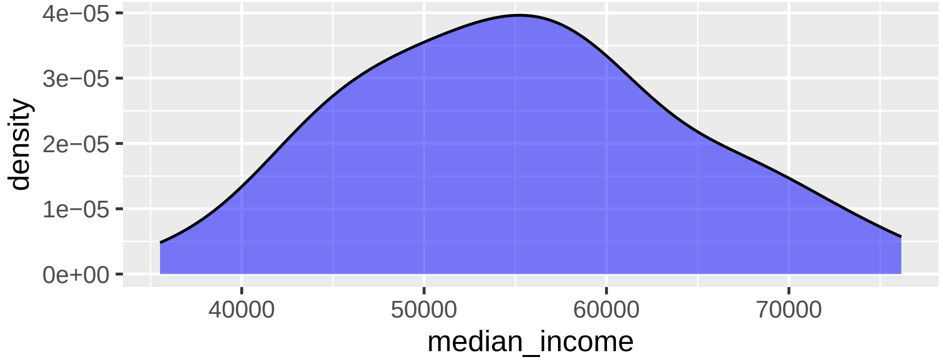


(b) Density Plots

```
ggplot(hate_crimes, aes(x=median_income)) +
  geom_density()
```



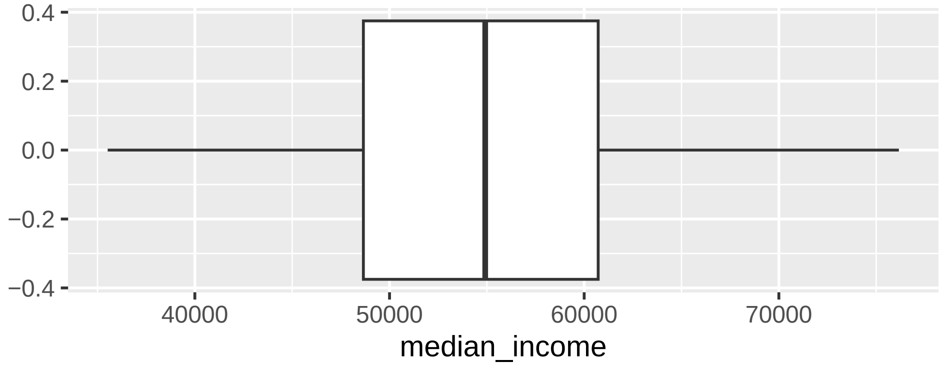
```
ggplot(hate_crimes, aes(x=median_income)) +
  geom_density(fill = "blue", alpha = 0.5)
```



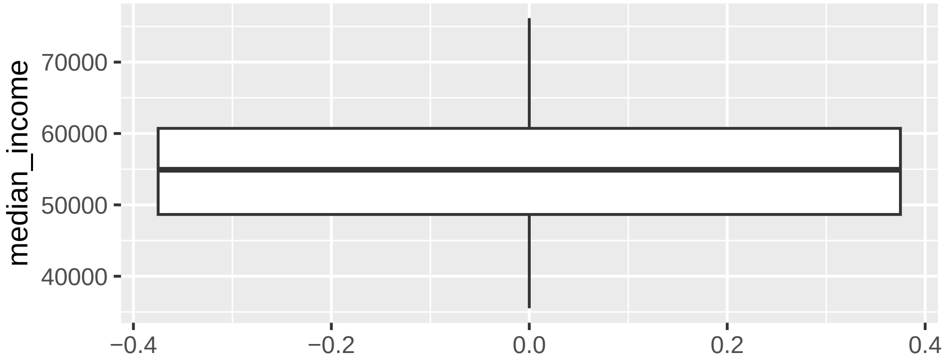
(c) Box Plots

What is the difference between box-plot and histogram/density plot? Box-plot shows the median but not the distribution.

```
ggplot(hate_crimes, aes(x=median_income)) +
  geom_boxplot()
```

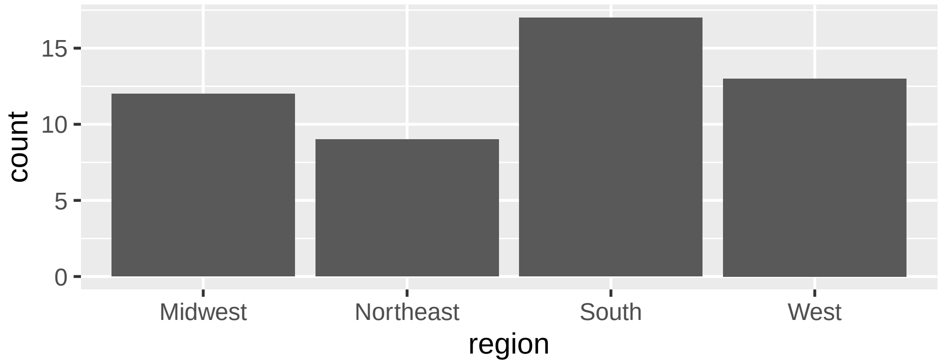


```
ggplot(hate_crimes, aes(y=median_income)) +
  geom_boxplot()
```

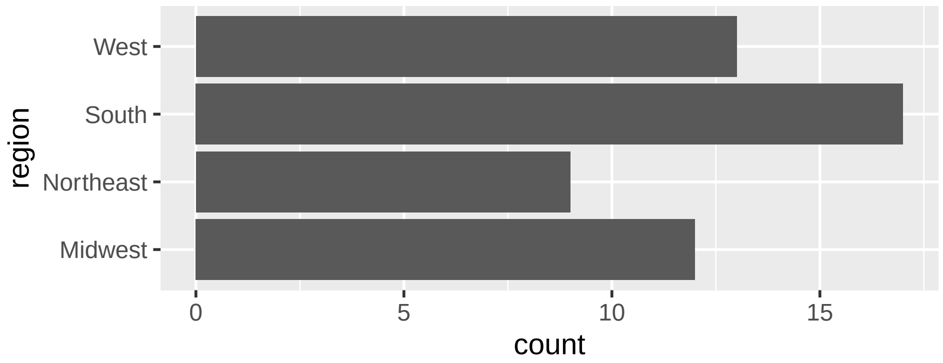


2. Visualize one categorical variable, usually for the response variable using a bar-plot

```
ggplot(hate_crimes, aes(x=region)) +
  geom_bar()
```



```
ggplot(hate_crimes, aes(x=region)) +
  geom_bar() +
  coord_flip()
```

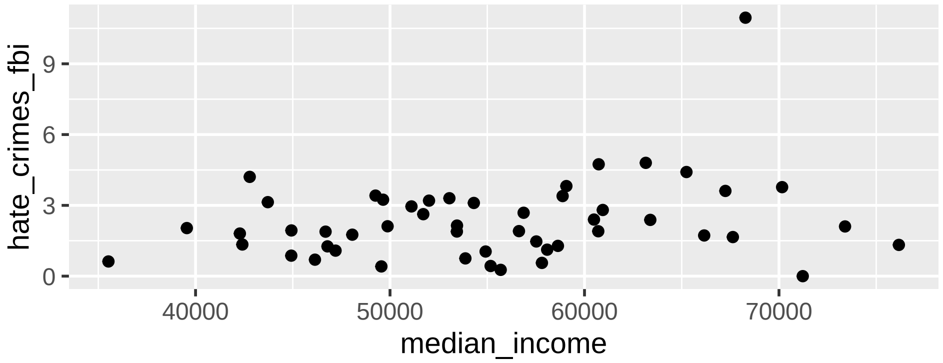


3. Visualize two continous variables.

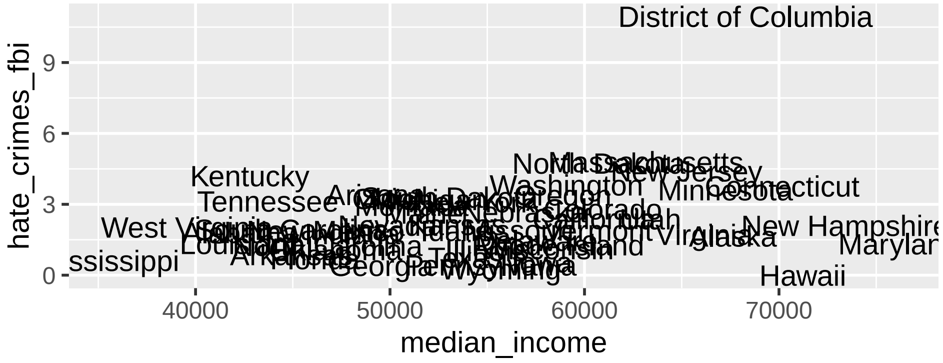
Focus on showing the relation between them. Can be response variable + explanatroy variable. Can also be explanatory variable + explanatory variable.

(a) Scatterplots

```
ggplot(hate_crimes, aes(x=median_income, y=hate_crimes_fbi)) +
  geom_point()
```



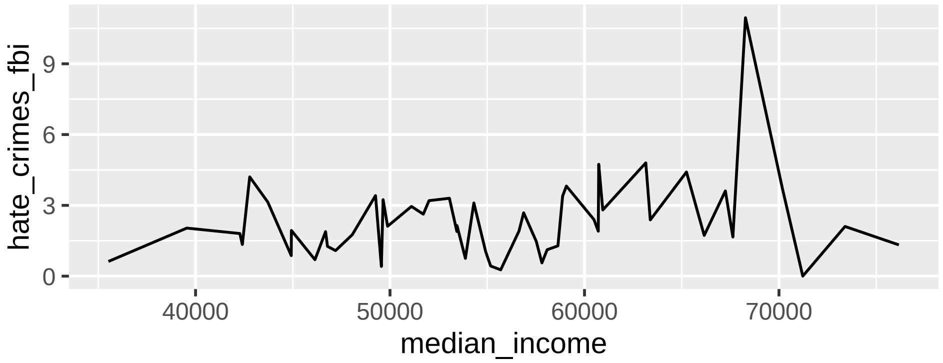
```
ggplot(hate_crimes, aes(x=median_income, y=hate_crimes_fbi)) +
  geom_text(aes(label=state))
```



(b) Line plots and Smooth Line Plots to connect the points in the scatterplot.

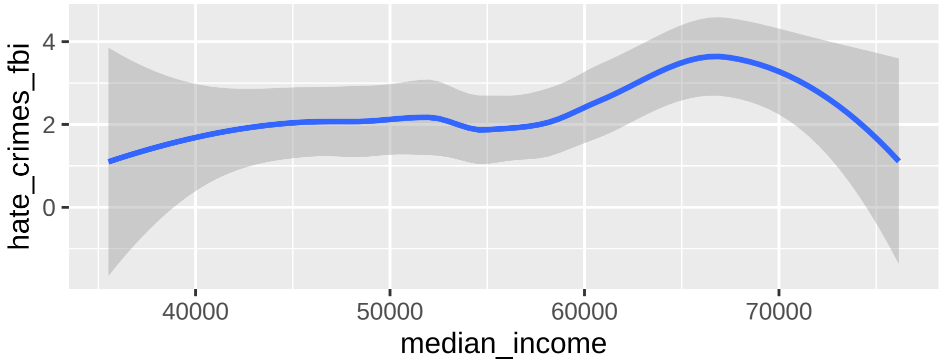
The Smooth Line Plots help show the trend due to smoothness

```
ggplot(hate_crimes, aes(x=median_income, y=hate_crimes_fbi)) +
  geom_line()
```



```
ggplot(hate_crimes, aes(x=median_income, y=hate_crimes_fbi)) +
  geom_smooth()
```

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

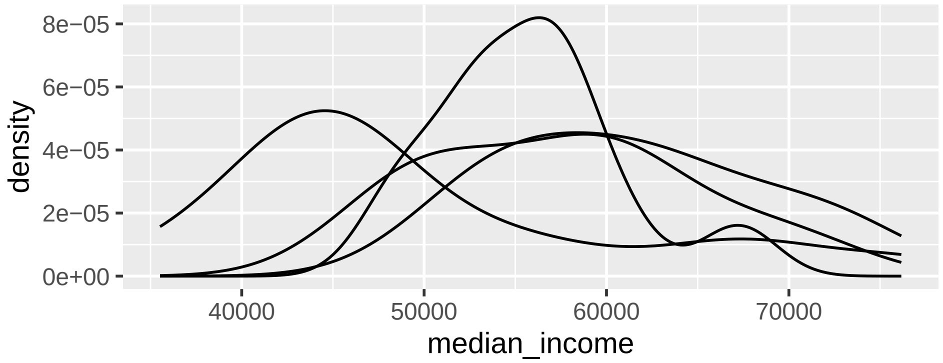


5. Visualize one continuous variable and one categorical variable (Multiple groups)

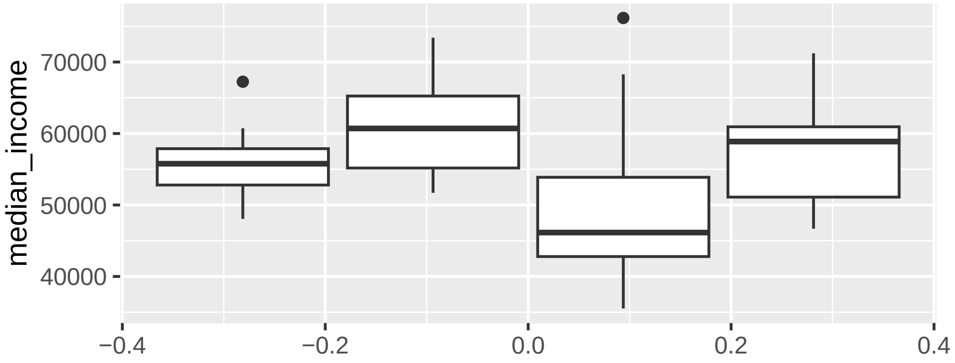
Sometimes we want to compare the variable(s) across multiple groups. eg: compare median income across different region. Which plots can compare multiple group?

Theses are called side-by-side plots.

```
ggplot(hate_crimes, aes(x=median_income, group = region)) +
  geom_density()
```



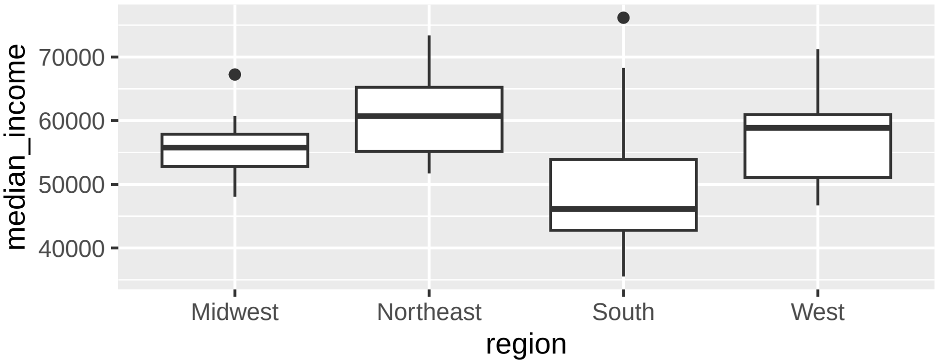
```
ggplot(hate_crimes, aes(y=median_income, group = region)) +
  geom_boxplot()
```



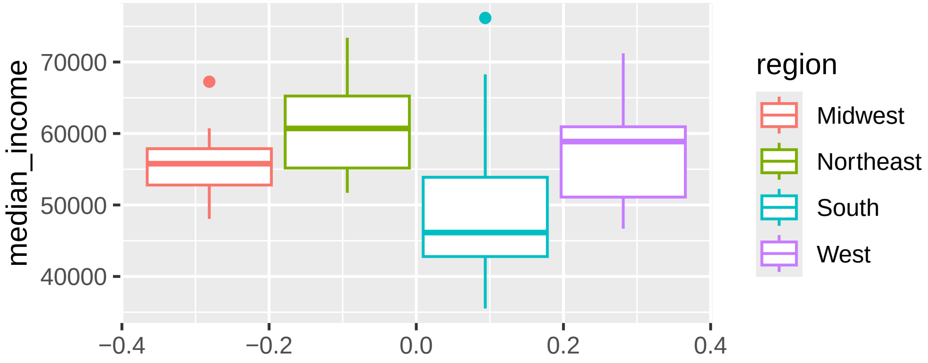
Wait, how can I know which group is which group?

Include options like color and size

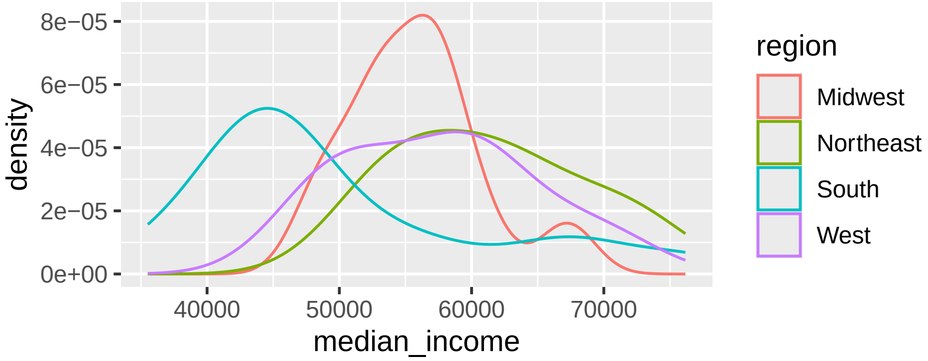
```
ggplot(hate_crimes, aes(y=median_income, x = region)) +
  geom_boxplot()
```



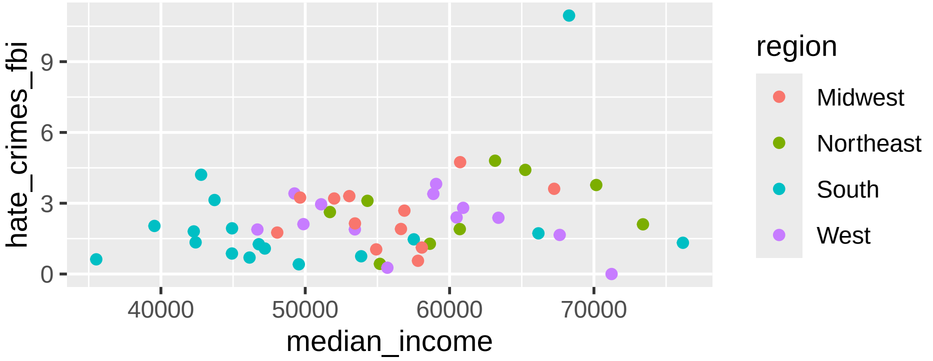
```
ggplot(hate_crimes, aes(y=median_income, group = region)) +
  geom_boxplot(aes(color = region))
```



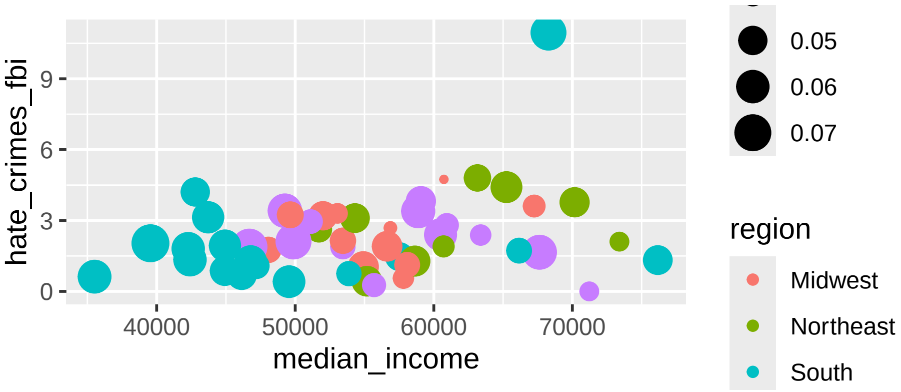
```
ggplot(hate_crimes, aes(x=median_income, group = region)) +
  geom_density(aes(color = region))
```



```
ggplot(hate_crimes, aes(x=median_income, y=hate_crimes_fbi, group = region)) +
  geom_point(aes(color = region))
```



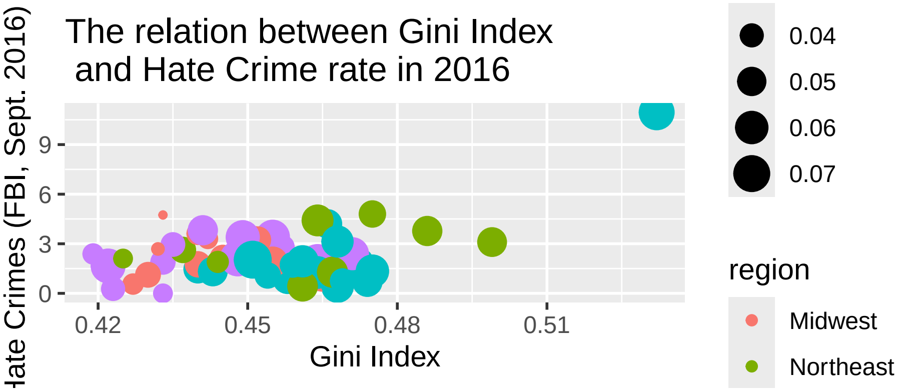
```
ggplot(hate_crimes, aes(x=median_income, y=hate_crimes_fbi, group = region)) +
  geom_point(aes(color = region, size = unemployment))
```



Adding plot title and changing axis titles

Add x-axis, y-axis labels and title

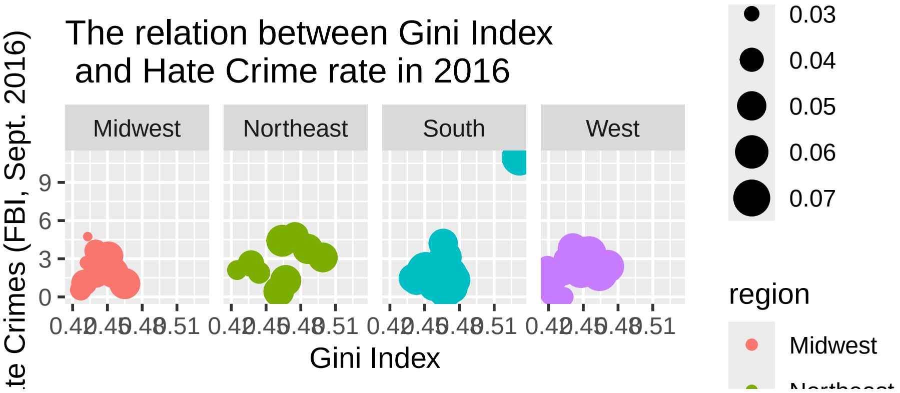
```
ggplot(hate_crimes, aes(x=gini_index, y=hate_crimes_fbi)) +
  geom_point(aes(color=region, size=unemployment)) +
  xlab('Gini Index') +
  ylab('Hate Crimes (FBI, Sept. 2016)') +
  ggtitle('The relation between Gini Index \n and Hate Crime rate in 2016')
```



Faceting by groups

Instead of putting all groups information into one page, you can do by each panel.

```
ggplot(hate_crimes, aes(x=gini_index, y=hate_crimes_fbi)) +
  geom_point(aes(color=region, size=unemployment)) +
  xlab('Gini Index') +
  ylab('Hate Crimes (FBI, Sept. 2016)') +
  ggtitle('The relation between Gini Index \n and Hate Crime rate in 2016')+
  facet_wrap(~region, nrow=1)
```

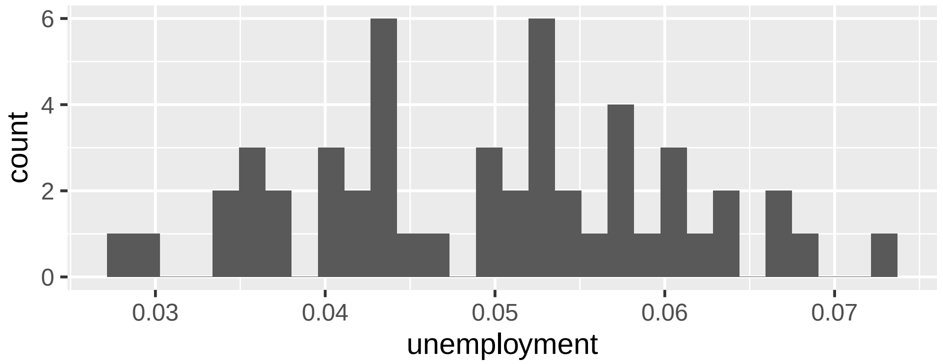


Try it for yourself

1. Suppose we are interested in the unemployment rate and want to see its distribution.

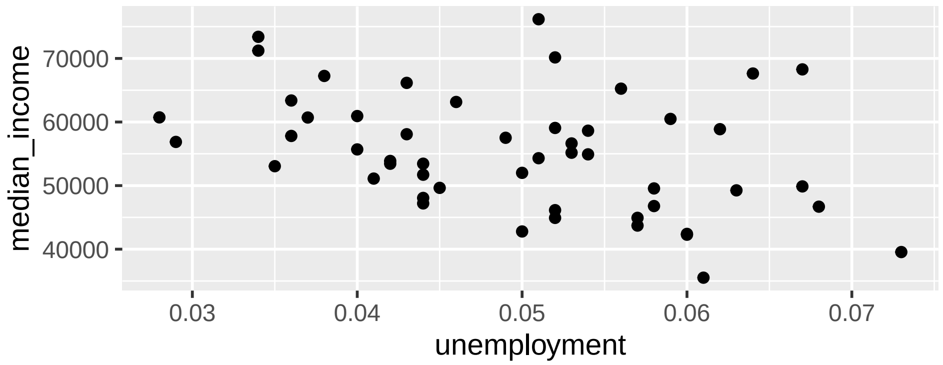
```
ggplot(hate_crimes, aes(x=unemployment)) +
  geom_histogram()
```

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



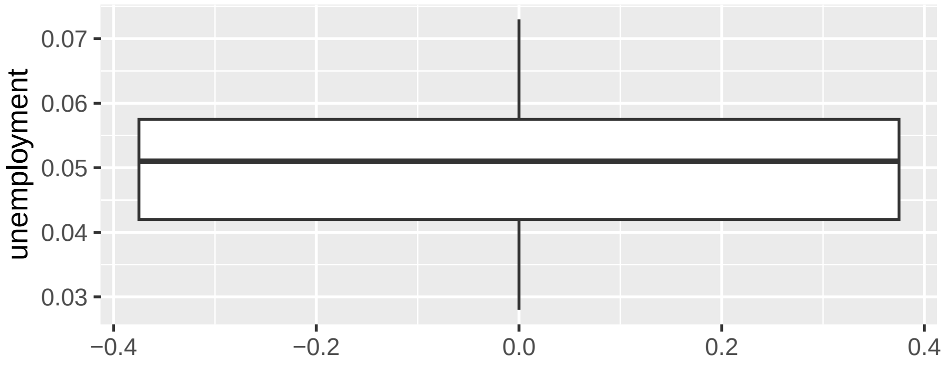
2. Suppose we want to show the relation between unemployment rate and median income.

```
ggplot(hate_crimes, aes(x=unemployment, y = median_income)) +
  geom_point()
```



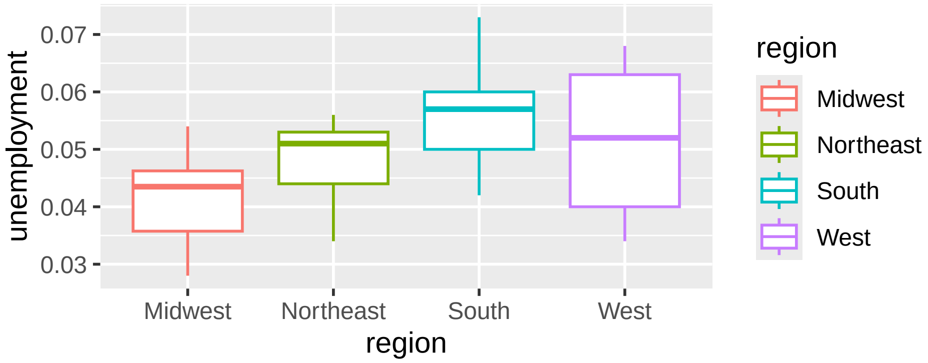
3. Report an approximate median for the unemployment rate.

```
ggplot(hate_crimes, aes(y=unemployment)) +
  geom_boxplot()
```



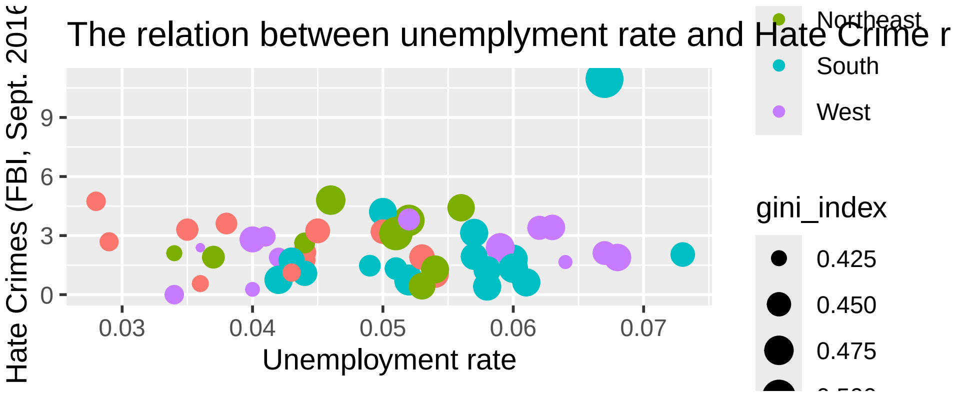
4. Show the unemployment rate across different region. Use color to indicate different regions.

```
ggplot(hate_crimes, aes(y = unemployment, x = region)) +
  geom_boxplot(aes(color = region))
```



5. Show the relation between unemployment rate and FBI hate crime rate. Use size to indicate gini index. Make sure to include axis labels and title.

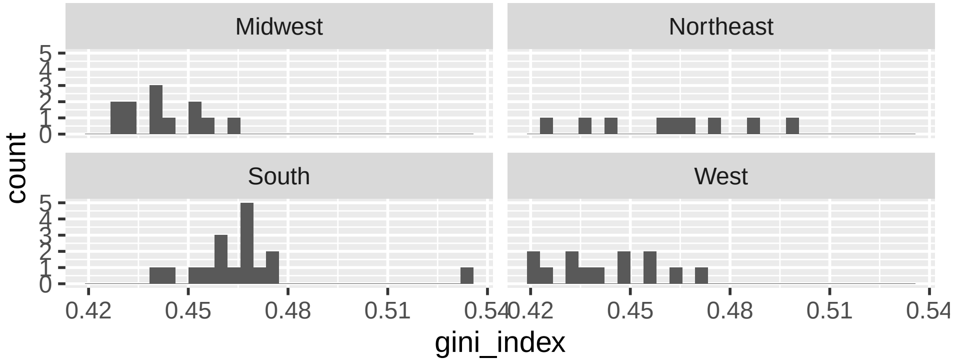
```
ggplot(hate_crimes, aes(x=unemployment, y=hate_crimes_fbi)) +
  geom_point(aes(color = region, size=gini_index)) +
  xlab('Unemployment rate') +
  ylab('Hate Crimes (FBI, Sept. 2016)') +
  ggtitle('The relation between unemplyment rate and Hate Crime rate in 2016')
```



6. Plot the distribution of gini index and put differnet region on different panel.

```
ggplot(hate_crimes, aes(x = gini_index)) +
  geom_histogram() +
  facet_wrap(~region)
```

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



Weather patterns

The data set `Weather` contains data on weather-related variables for several world cities.

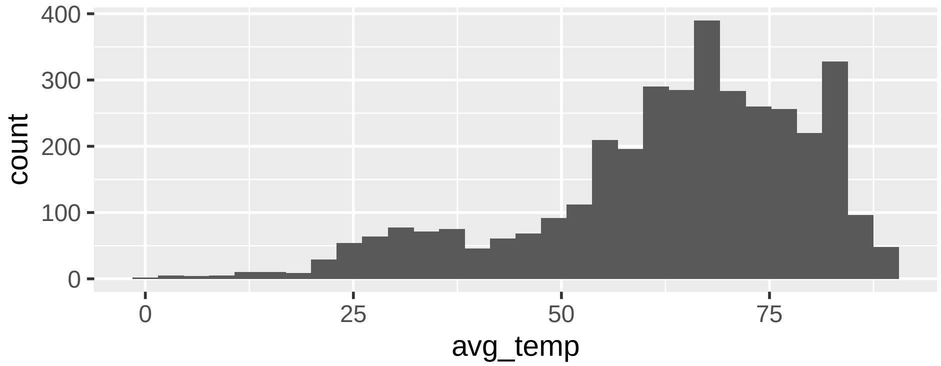
```
#install.packages(mosaicData)
library(mosaicData)
data(Weather)
glimpse(Weather)
```

```
Rows: 3,655
Columns: 25
$ city          <chr> "Auckland", "Auckland", "Auckland", "Auckland", "Aucklan~
$ date          <date> 2016-01-01, 2016-01-02, 2016-01-03, 2016-01-04, 2016-01~
$ year          <dbl> 2016, 2016, 2016, 2016, 2016, 2016, 2016, 2016, 2016, 20~
$ month         <int> 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,~
$ day           <dbl> 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 1~
$ high_temp     <dbl> 68, 68, 77, 73, 69, 69, 71, 77, 69, 71, 75, 69, 71, 75, ~
$ avg_temp      <dbl> 65, 66, 72, 66, 62, 63, 66, 70, 66, 66, 67, 66, 66, 68, ~
$ low_temp      <dbl> 62, 64, 66, 60, 55, 57, 60, 64, 64, 62, 59, 62, 62, 62, ~
$ high_dewpt    <dbl> 64, 64, 70, 66, 55, 54, 59, 72, 68, 63, 61, 66, 61, 63, ~
$ avg_dewpt     <dbl> 60, 63, 67, 60, 52, 51, 54, 67, 61, 58, 58, 62, 57, 61, ~
$ low_dewpt     <dbl> 55, 61, 64, 54, 48, 46, 50, 59, 55, 55, 54, 59, 54, 59, ~
$ high_humidity <dbl> 100, 100, 100, 100, 82, 88, 83, 100, 100, 88, 94, 100, 8~
$ avg_humidity  <dbl> 82, 94, 91, 76, 69, 65, 65, 92, 81, 76, 72, 87, 73, 80, ~
$ low_humidity  <dbl> 68, 88, 74, 53, 56, 46, 53, 83, 64, 64, 53, 78, 64, 65, ~
$ high_hg       <dbl> 30.15, 30.04, 29.80, 30.12, 30.21, 30.24, 30.24, 30.01, ~
$ avg_hg        <dbl> 30.09, 29.90, 29.73, 29.90, 30.14, 30.22, 30.13, 29.79, ~
$ low_hg        <dbl> 30.01, 29.80, 29.68, 29.77, 30.09, 30.18, 30.04, 29.62, ~
$ high_vis      <dbl> 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,~
$ avg_vis       <dbl> 6, 5, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6,~
$ low_vis       <dbl> 4, 1, 1, 6, 6, 6, 6, 2, 4, 6, 6, 3, 6, 6, 6, 6, 6, 2, 6,~
$ high_wind     <dbl> 21, 33, 18, 15, 13, 17, 22, 21, 18, 20, 17, 16, 18, 17, ~
$ avg_wind      <dbl> 15, 21, 12, 10, 7, 8, 12, 14, 11, 15, 10, 8, 13, 9, 7, 4~
$ low_wind      <dbl> 28, 46, NA, NA, NA, 28, 25, 28, 29, NA, 26, NA, NA, NA, ~
$ precip        <chr> "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "0", "~
$ events        <chr> "Rain", "Rain", "Rain", "Rain", NA, NA, "Rain", "Rain", ~
```

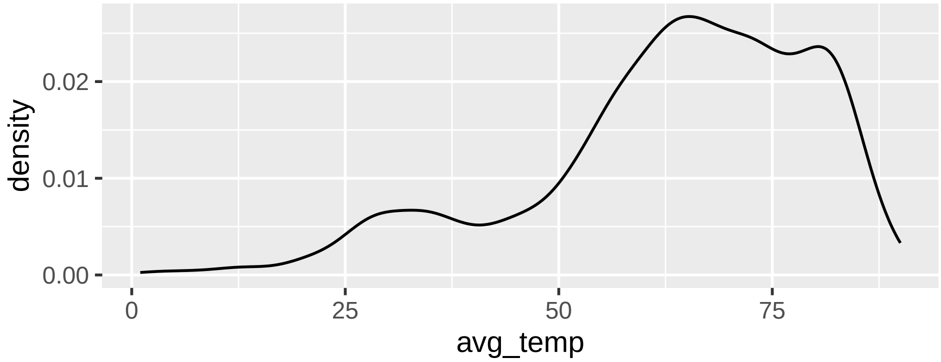
(a). We want to study the average temperature. First, we want to see its distribution. How would we do this?

```r
ggplot(Weather, aes(x=avg_temp)) +
  geom_histogram()
```

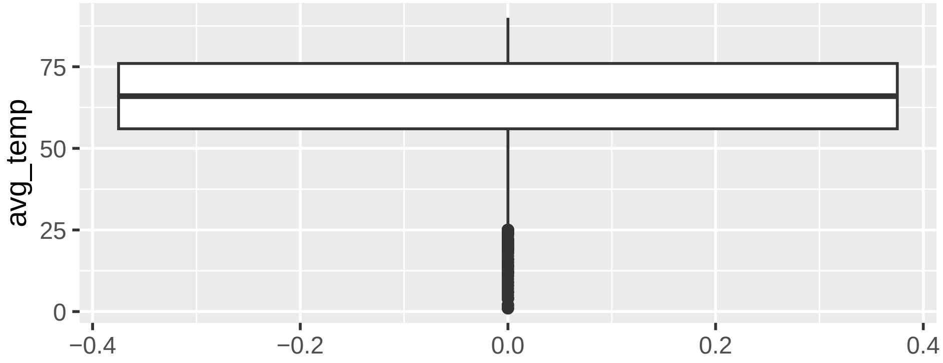
```
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



```
ggplot(Weather, aes(x=avg_temp)) +
  geom_density()
```



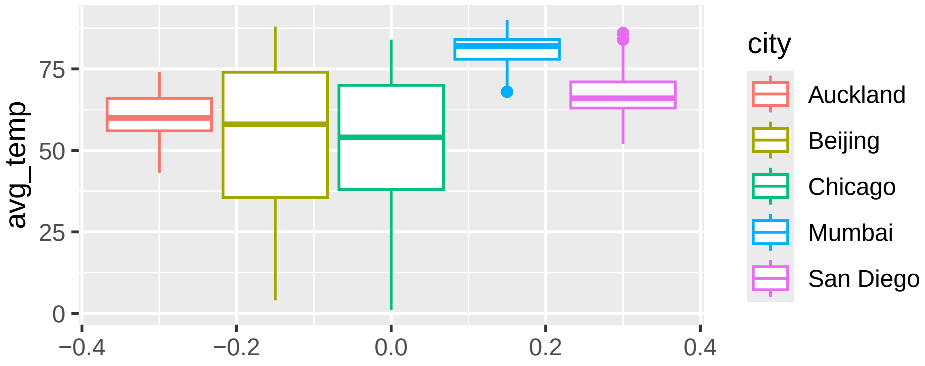
```
ggplot(Weather, aes(y=avg_temp)) +
  geom_boxplot()
```



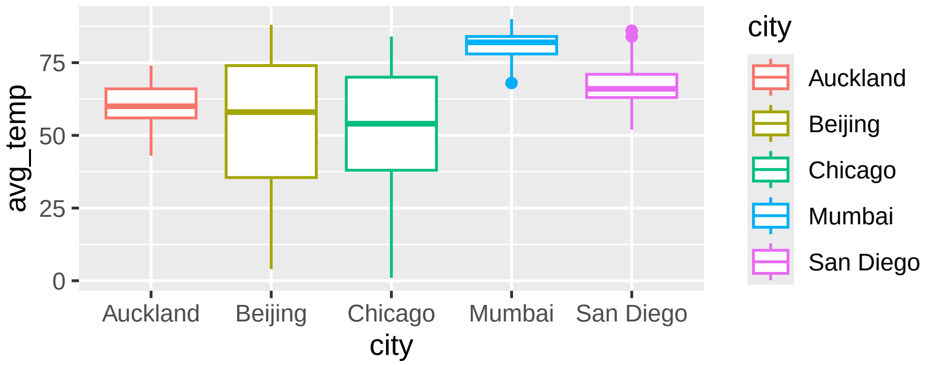
(b). Next, we want to study the distribution of average temperature across different cities.

How would we do this?

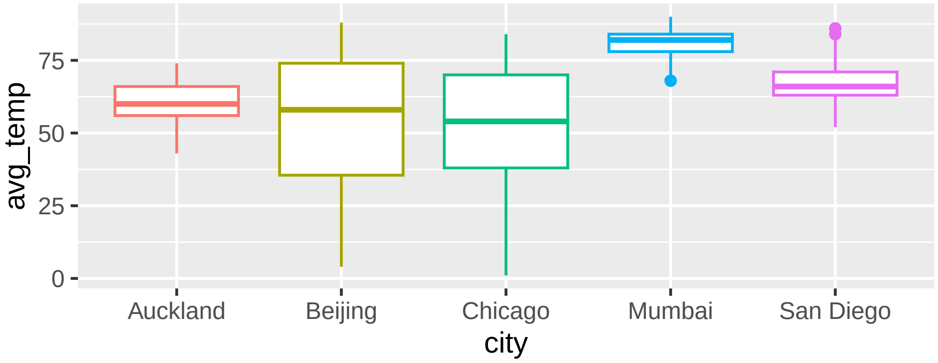
```
ggplot(Weather, aes(y=avg_temp, group=city)) +
  geom_boxplot(aes(color=city))
```



```
ggplot(Weather, aes(y=avg_temp, x = city, group=city)) +
  geom_boxplot(aes(color=city))
```

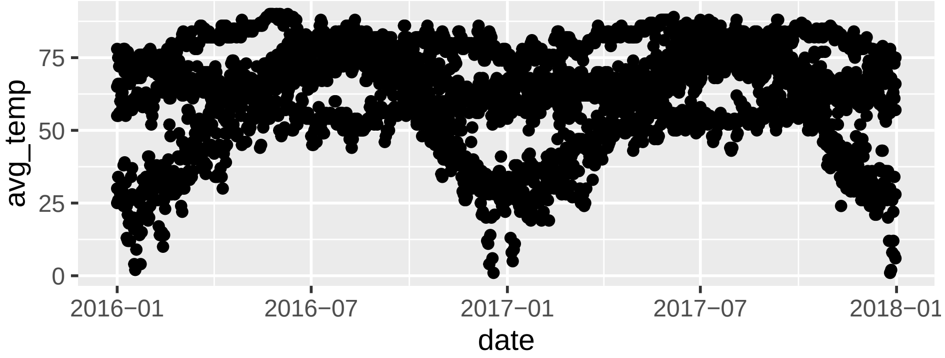


```
ggplot(Weather, aes(y=avg_temp, x = city, group=city)) +
  geom_boxplot(aes(color=city)) + theme(legend.position = "none")
```



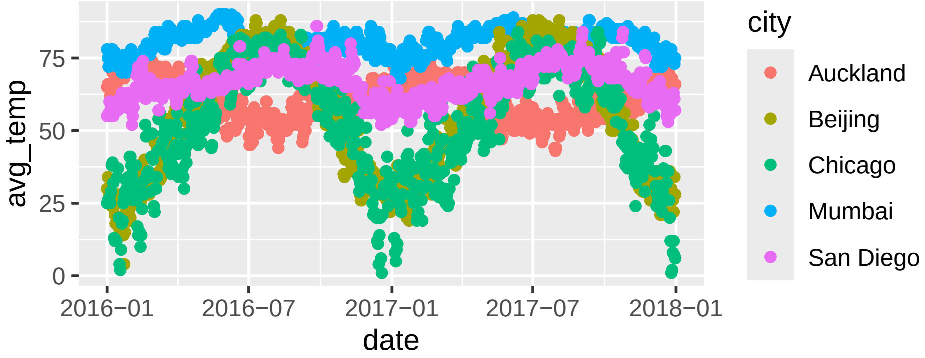
(c). The average temperature may be related to date. How to show the relation between temperature and date?

```
ggplot(Weather, aes(x=date, y=avg_temp)) +
  geom_point()
```



(d). Maybe different city has totally different trend on average temperature (ie. relationship between temperature and date grouped by city).

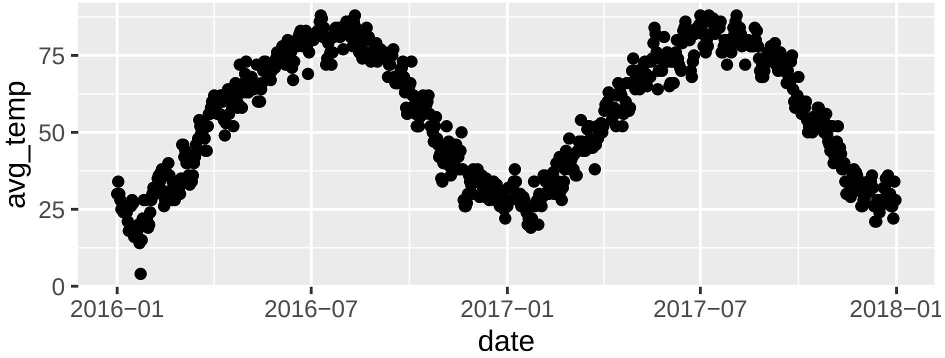
```
ggplot(Weather, aes(x=date, y=avg_temp)) +
  geom_point(aes(color=city))
```



(e). What if we only care about one city? Show the relationship between temperature and date for Beijing only.

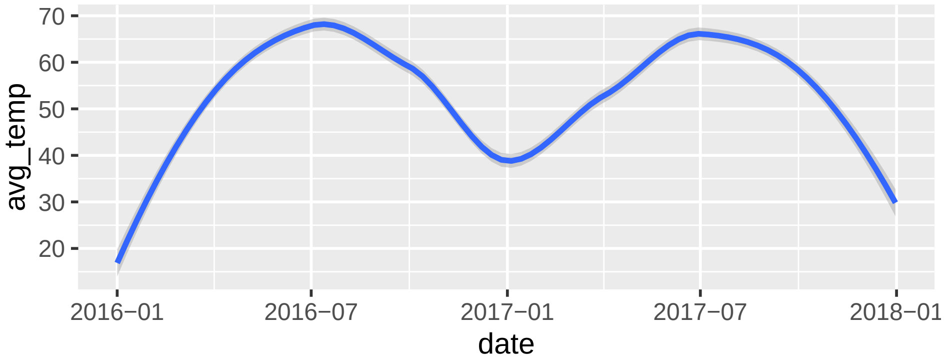
```r
Beijing <- Weather %>% filter(city=='Beijing')

ggplot(Beijing, aes(x=date, y=avg_temp)) +
  geom_point()
```



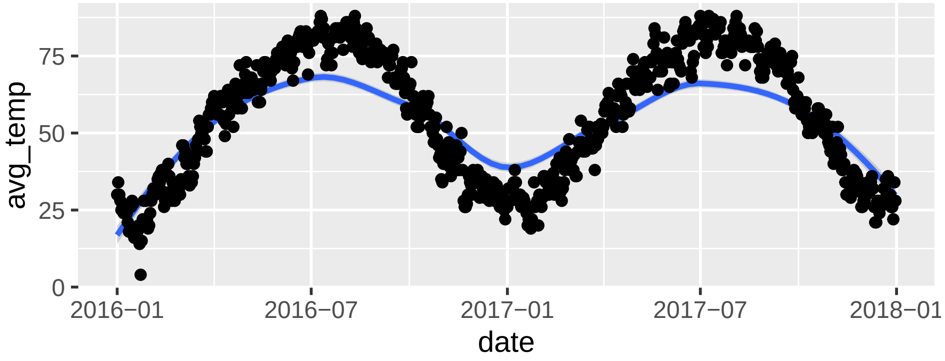
```r
ggplot(Beijing, aes(x=date, y=avg_temp)) +
  geom_smooth()
```

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



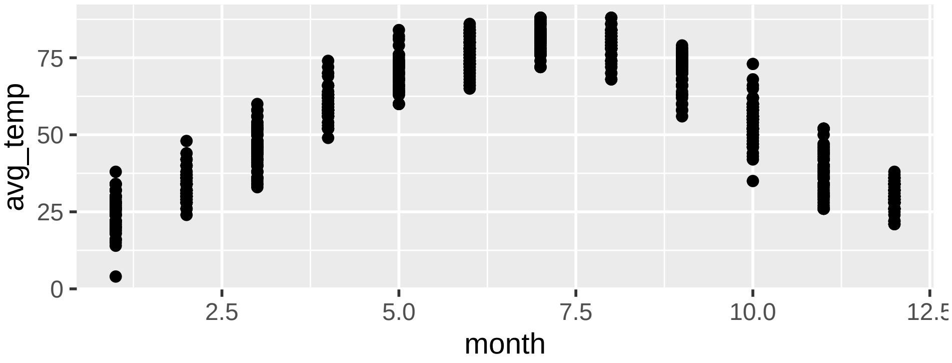
```r
ggplot(Beijing, aes(x=date, y=avg_temp)) +
  geom_smooth() +
  geom_point()
```

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

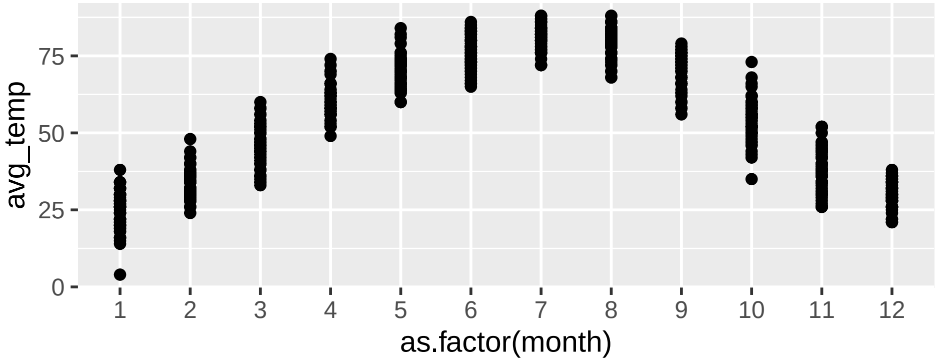


(f). Instead of the date, we can also use the month. Show the relationship between month and temperature (for Beijing).

```r
ggplot(Beijing, aes(y=avg_temp, x = month)) +
  geom_point()
```

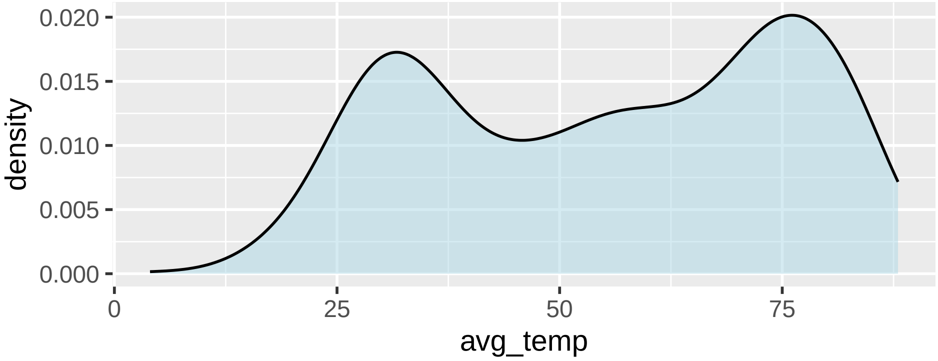


```
ggplot(Beijing, aes(y=avg_temp, x = as.factor(month))) +
  geom_point()
```

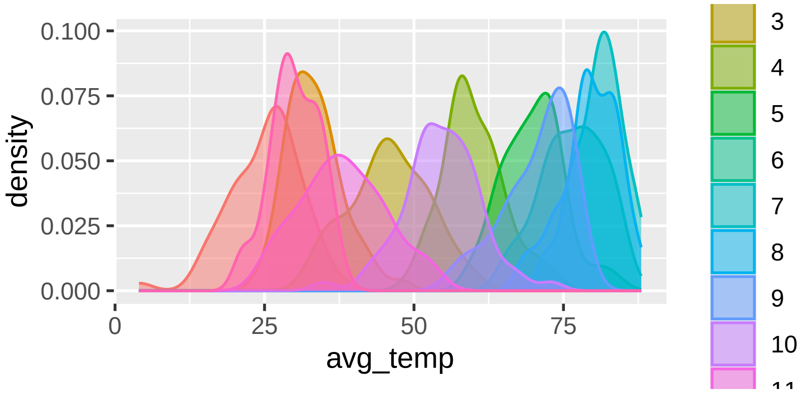


(g). Show the distribution of average temperature for the Beijing data set by month.

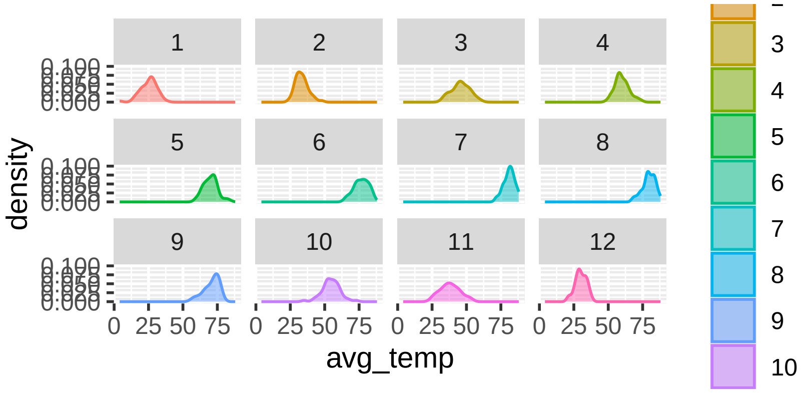
```
ggplot(Beijing, aes(x=avg_temp)) +
  geom_density(fill='lightblue', alpha=0.5)
```



```
ggplot(Beijing, aes(x=avg_temp, group=month)) +
  geom_density(aes(color=as.factor(month), fill=as.factor(month)), alpha=0.5)
```

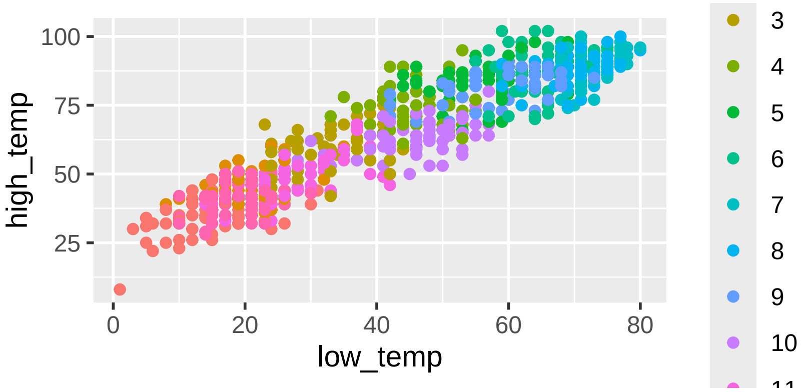


```
ggplot(Beijing, aes(x=avg_temp, group=month)) +
  geom_density(aes(color=as.factor(month), fill=as.factor(month)), alpha=0.5) +
  facet_wrap(~month, nrow=3)
```



(h). Show the relationship between the low temperature and the high temperature colored by month

```
ggplot(Beijing, aes(x=low_temp, y=high_temp)) +
  geom_point(aes(color=as.factor(month)))
```


high_temp
100
75
50
25
low_temp
0
20
40
60
80
3
4
5
6
7
8
9
10
11