Exploring data #1

Plots

Plots to explore data

Plots can be invaluable in exploring your data.

Today, we will focus on **useful**, rather than **attractive** graphs, since we are focusing on exploring rather than presenting data.

Next lecture, we will talk more about customization, to help you make more attractive plots that would go into final reports.

`ggplot` conventions

Here, we'll be using functions from the `ggplot2` library, so you'll need to install that package:

```r
library("ggplot2")
```

The basic steps behind creating a plot with `ggplot2` are:

1. Create an object of the `ggplot` class, typically specifying the **data** to be shown in the plot;
2. Add on (using +) one or more **geoms**, specifying the **aesthetics** for each; and
3. Add on (using +) other elements to create and customize the plot (e.g., add layers to customize scales or themes or to add facets).

Note: To avoid errors, end lines with +, don't start lines with it.

Plot data

The `ggplot` function requires you to input a dataframe with the data you will plot. All the columns in that dataframe can be mapped to specific aesthetics within the plot.

```
beijing_pm %>%
  slice(1:3)
```

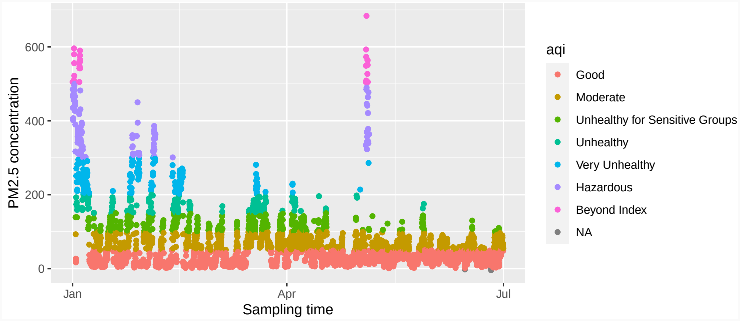
```
## # A tibble: 3 x 5
##   sample_time         value qc    aqi       heating
##   <dttm>              <dbl> <chr> <fct>     <lgl>
## 1 2017-01-01 00:00:00   505 Valid Beyond ~ TRUE
## 2 2017-01-01 01:00:00   485 Valid Hazardo~ TRUE
## 3 2017-01-01 02:00:00   466 Valid Hazardo~ TRUE
```

For example, if we input the `beijing_pm` dataframe, we would be able to create a plot that shows each sample's sampling time on the x-axis, $PM_{2.5}$ concentration on the y-axis, and AQI by the color of the point.

Plot aesthetics

Aesthetics are plotting elements that can show certain elements of the data.

For example, you may want to create a scatterplot where color shows AQI, x-position shows sampling time, and y-position shows $PM_{2.5}$ concentration.



Plot aesthetics

In the previous graph, the mapped aesthetics are color, x, and y. In the ggplot code, all of these aesthetic mappings will be specified within an aes call, which will be nested in another call in the ggplot pipeline.

Aesthetic	ggplot abbreviation	beijing_pm column
x-axis position	x =	sample_time
y-axis position	y =	value
color	color =	aqi

This is how these mappings will be specified in an aes call:

```
# Note: This code should not be run by itself.
# It will eventually be nested in a ggplot call.
aes(x = sample_time, y = value, color = aqi)
```

Plot aesthetics

Here are some common plot aesthetics you might want to specify:

Code	Description
`x`	Position on x-axis
`y`	Position on y-axis
`shape`	Shape
`color`	Color of border of elements
`fill`	Color of inside of elements
`size`	Size
`alpha`	Transparency (1: opaque; 0: transparent)
`linetype`	Type of line (e.g., solid, dashed)

Geoms

You will add **geoms** that create the actual geometric objects on the plot. For example, a scatterplot has “points” geoms, since each observation is displayed as a point.

There are `geom_*` functions that can be used to add a variety of geoms. The function to add a “points” geom is `geom_point`.

We just covered three plotting elements:

- Data
- Aesthetics
- Geoms

These are three elements that you will almost always specify when using `ggplot`, and they are sufficient to create a number of basic plots.

Creating a ggplot object

You can create a scatterplot using ggplot using the following code format:

```
## Generic code
ggplot(data = dataframe) +
  geom_point(mapping = aes(x = column_1, y = column_2,
                           color = column_3))
```

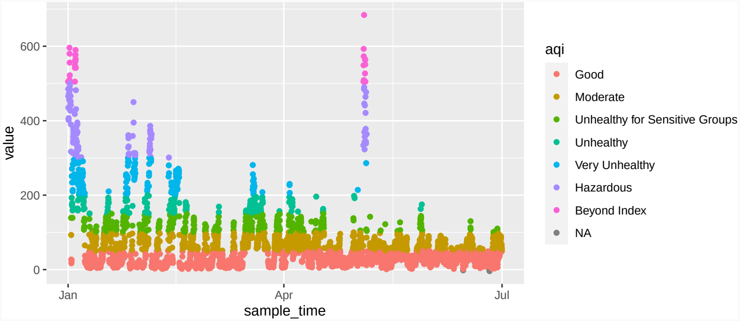
Notice that:

1. The ggplot call specifies the **dataframe** with the data you want to plot
2. A **geom** is added using the appropriate geom_* function for a scatterplot (geom_point).
3. The mappings between columns in the dataframe and **aesthetics** of the geom is specified within an aes call in the mapping argument of the geom_* function call.
4. The aes call includes mappings to two aesthetics that are required from the geom_point geom (x and y) and one that is optional (color).

Creating a ggplot object

Let's put these ideas together to write the code to create a plot for our example data:

```r
ggplot(data = beijing_pm) +
geom_point(mapping = aes(x = sample_time, y = value,
                         color = aqi))
```



Adding geoms

There are a number of different `geom_*` functions you can use to add geoms to a plot. They are divided between geoms that directly map the data to an aesthetic and those that show some summary or statistic of the data.

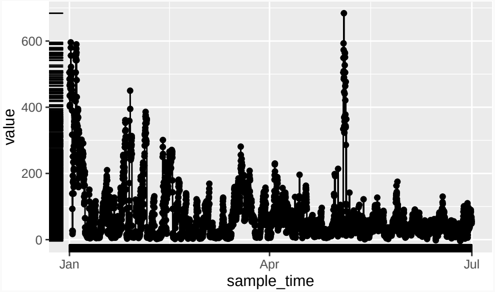
Some of the most common direct-mapping geoms are:

Geom(s)	Description
`geom_point`	Points in 2-D (e.g. scatterplot)
`geom_line`, `geom_path`	Connect observations with a line
`geom_abline`	A line with a certain intercept and slope
`geom_hline`, `geom_vline`	A horizontal or vertical line
`geom_rug`	A rug plot
`geom_label`, `geom_text`	Text labels

Creating a ggplot object

You can add several geoms to the same plot as layers:

```
ggplot(data = beijing_pm) +
geom_point(mapping = aes(x = sample_time, y = value)) +
geom_line(mapping = aes(x = sample_time, y = value)) +
geom_rug(mapping = aes(x = sample_time, y = value))
```



Creating a ggplot object

You may have noticed that all of these geoms use the same aesthetic mappings (height to x-axis position, weight to y-axis position, and sex to color). To save time, you can specify the aesthetic mappings in the first `ggplot` call. These mappings will then be the default for any of the added geoms.

```r
ggplot(data = beijing_pm,
       mapping = aes(x = sample_time, y = value)) +
  geom_point() +
  geom_line() +
  geom_rug()
```

Creating a ggplot object

Because the first argument of the ggplot call is a dataframe, you can also "pipe into" a ggplot call:

```
beijing_pm %>%
  ggplot(aes(x = sample_time, y = value)) +
  geom_point() +
  geom_line() +
  geom_rug()
```