QTM 150

Week 12 – ggplot2 (cont'd)

Umberto Mignozzetti
Apr 16

Recap

You now know:

- The main objects in R.
- How to do basic operations with datasets.
- How to create graphs and plots.
- Data manipulation with `dplyr`

Great job!!

Do you have any questions?

Today we are going to develop even further our **ggplot** skills!

This week

We will have a **quiz** posted today after 4:00 PM. Due by **Monday** (because of the holidays this week).

We will have a **problem set** posted tomorrow, due by the next lab.

We will have to post a one-pager plan of the analysis that you plan to do with your group, by **Tuesday**.

Our GitHub page is: https://github.com/umbertomig/qtm150

Today's Agenda

`ggplot2` graphs:

- Graphs for numeric variables
- Graphs for discrete variables
- Graphs for discrete x discrete variables
- Graphs for discrete x numeric variables
- Graphs for numeric x numeric variables

Getting Started

Getting Started: loading packages

```
# Loading tidyverse
library(tidyverse)
```

```
## ── Attaching packages ─────────────────────────────────────── tidyv

## ✓ ggplot2 3.3.3     ✓ purrr   0.3.4
## ✓ tibble  3.1.0     ✓ dplyr   1.0.5
## ✓ tidyr   1.1.3     ✓ stringr 1.4.0
## ✓ readr   1.4.0     ✓ forcats 0.5.0

## ── Conflicts ────────────────────────────────────────── tidyverse_c
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

Loading datasets

```r
# Loading tips dataset
tips ← read.csv('https://raw.githubusercontent.com/umbertomig/qtm
head(tips, 2)
```

```
##   obs totbill  tip sex smoker day  time size
## 1   1   16.99 1.01   F     No Sun Night    2
## 2   2   10.34 1.66   M     No Sun Night    3
```

```r
# Loading PErisk dataset
PErisk ← read.csv('https://raw.githubusercontent.com/umbertomig/c
head(PErisk, 2)
```

```
##     country courts       barb2 prsexp2 prscorr2    gdpw2
## 1 Argentina      0 -0.7207754       1         3  9.69017
## 2 Australia      1 -6.9077550       5         4 10.30484
```

```r
data(USArrests)
```

ggplot underlying logic

ggplot underlying logic

`ggplot` is based on the *grammar of graphs* idea. This idea emphasizes that all graphs are composed of three elements:

- A dataset
- A coordinated system (**mapping** and **aes**)
- Geometric figures (**geom**s)

From these elementary building blocks, we can build any graph we want.

Graphs for numeric variables

geom_histogram

A histogram is a great graph for a numeric variable.

In the dataset USArrests, we can have the histogram for the murder variable:

```
ggplot(data = USArrests) +
  geom_histogram(mapping = aes(x = Murder), bins=15)
```

Your turn: Make a histogram of the variable *Assault* in the dataset **USArrests**.

geom_density

Density plots are great graphs to have an idea about how the data is distributed.

This is the code for a density plot of the *Murder* variable, in the dataset **USArrests**:

```
ggplot(data = USArrests) +
  geom_density(mapping = aes(x = Murder),
               kernel = 'gaussian')
```

geom_density

Box-plots are very useful to check how a variable is distributed.

Here is a box-plot of the *Murder* variable, in the dataset **USArrests**:

```
ggplot(data = USArrests) +
  geom_boxplot(mapping = aes(x = 1, y = Murder), alpha=0.3)
```

Your turn: Make a box-plot of the variable *Assault* in the dataset **USArrests**.

geom_violin

And a smoothed version of a density plot is the `violin plot`. When there are multiple numeric variables, it is a useful graph.

Here is a violin-plot of the *Murder* variable, in the dataset **USArrests**:

```
ggplot(data = USArrests) +
  geom_violin(mapping = aes(x = 'Violin', y = Murder), lwd = 2)
```

Your turn: Make a violin-plot of the variable *Assault* in the dataset **USArrests**.

Customizing plots

We can customize several characteristics of the plot:

```
ggplot(data = USArrests) +
  geom_histogram(mapping = aes(x = Murder),
                 bins = 8, color = 'gray20',
                 fill = 'gray80', alpha = 0.5) +
  labs(x = 'Murder rate',
       y = 'Frequency',
       title = '1970 US States Murder Rates')
```

Graph for discrete variables

geom_bar

A bar plot for the *courts* variable in the dataset **PErisk** would tell us how many countries had independent courts in 1992:

```
ggplot(data = PErisk) +
  geom_bar(mapping = aes(x = courts))
```

```
PErisk$courts = as.factor(PErisk$courts)
levels(PErisk$courts) ← c('Non-independent', 'Independent')
```

```
ggplot(data = PErisk) +
  geom_bar(mapping = aes(x = courts))
```

```
ggplot(data = PErisk) +
  geom_bar(mapping = aes(x = factor(prsexp2)),
           fill = rainbow(6), color = 'black')
```

Your turn: Make a barplot of the variable *day* in the dataset **tips**.

Graphs discrete x discrete variables

Mosaic-plots

A bar-plot with two variables, one against the other, gives us a good idea whether they are related with each other.

This code plots the variable *corruption* against the variable *courts*:

```
ggplot(data = PErisk) +
  geom_bar(mapping = aes(x = factor(prscorr2),
                         fill = as.factor(courts)),
           position = 'fill')
```

```
PErisk %>%
  mutate(corruption = ifelse(prscorr2>2, 'Low', 'High')) %>%
  ggplot() +
  geom_bar(mapping = aes(x = corruption, fill = as.factor(courts)),
           position = 'fill')
```

Your turn: Make a mosaic-plot of the variables *sex* and *smoke*, in the dataset **tips**.

Graphs discrete x numeric variables

Box-plots for multple categories

To visualize the variation in a numeric variable, conditioning by a discrete variable, we can use the `box-plot` (or violin-plots) in a clever way:

```
PErisk %>%
  mutate(corruption = ifelse(prscorr2>2, 'Low', 'High')) %>%
  ggplot() +
  geom_boxplot(mapping = aes(x = as.factor(corruption), y = barb2))
```

```
PErisk %>%
  mutate(corruption = ifelse(prscorr2>2, 'Low', 'High')) %>%
  ggplot() +
  geom_violin(mapping = aes(x = as.factor(corruption), y = barb2))
```

Your turn: Plot the variable *tip* against the *time* of the day, in the dataset **tips**.

Graphs numeric x numeric variables

Scatter-plots

Two numeric variables, one against the other, are better visualized using a scatter-plot:

```
ggplot(data = PErisk) +
  geom_point(mapping = aes(x = barb2, y = gdpw2))
```

```
ggplot(data = PErisk) +
  geom_point(mapping = aes(x = barb2, y = gdpw2)) +
  geom_smooth(mapping = aes(x = barb2, y = gdpw2), method = 'lm')
```

Scatter-plots

Two numeric variables, one against the other, are better visualized using a scatter-plot:

```
ggplot(data = PErisk) +
  geom_point(mapping = aes(x = barb2, y = gdpw2, color = courts)) +
  geom_smooth(mapping = aes(x = barb2, y = gdpw2), method = 'lm')
```

```
ggplot(data = PErisk) +
  geom_point(mapping = aes(x = barb2, y = gdpw2, color = courts)) +
  geom_smooth(mapping = aes(x = barb2, y = gdpw2, color = courts),
              method = 'lm')
```

Your turn: In the dataset **tips**, plot the variable *tip* against the variable *totbill*, with and without differentiating by *smoker*.

Questions?

Have a great weekend!