Lab 9: Soccer data and professional looking plots

Michael Lopez, Skidmore College

Important note 1

First, we'll open RStudio by going to http://r.skidmore.edu/.

Important note 2

Open a new R Markdown file (File / New File / R Markdown...). You can create a basic name – Lab0, for example – and that'll set you up with a new file ready to go.

Overview

Today's goals

1. Introduction to soccer data
2. Summary shot-level statistics
3. Improving plots using ggplot

Data from the NWSL

Statsbomb is a soccer data science company located in England that has spent the last several years improving and disseminating soccer analytics. Over the last year, they have released much of their women's soccer data. This includes the National Women's Soccer League and other professional women's leagues. In this lab, we'll take a look at applying some of our tools to their soccer data, with a focus on how we can improve plots using `ggplot` add-ons.

You can read more about StatsBomb at http://statsbomb.com/.

And you can read about their data at https://github.com/statsbomb/open-data/tree/master/data.

One neat fact? One data scientist at StatsBomb's is Derrick Yam, Class of 2017 at Skidmore and a former men's soccer player.

First, we read in the data

```
library(tidyverse)
url.temp <- "https://raw.githubusercontent.com/statsbylopez/FYE_18/master/nwsl_shots.csv"
soccer.data <- read_csv(url.temp)
head(soccer.data)
dim(soccer.data)
```

Initial data steps

Recognizing categories

Often when we are working with new data sets, it helps to summarize each variable. There are seven variables, and without going through every row, it can be tedious to identify all responses to each variable.

```
soccer.data %>% count(shots.team)
soccer.data %>% count(shots.team, shots.outcome.name)
```

1. Use code similar to the above to identify the player with the most shots taken in this data set.
2. Use code similar to the above to identify the team with the most goals (`shots.outcome.name`).
3. Use code similar to the above to identify the team with the largest number of `Head` shots (`shots.body.part`).

Making a new variable.

The `shots.outcome.name` variable has several outcomes, only one of which is `Goal`. However, goals are likely the most interesting outcome for our purposes.

Recall: we can make a new variable using the `mutate()` command.

```
soccer.data <- soccer.data %>% mutate(is.goal = shots.outcome.name == "Goal")
soccer.data %>% count(is.goal)
```

4. Roughly what is the overall shot percentage (rate of `is.goal`) among all the shots in this data set?
5. Recall our notes on data summaries. Identify the type of shot (`shots.type.name`) with the highest rate of goals? Also identify the number of shots at each of those shot types.

Building a better plot

One possible question considers each team's scoring rate: that is, which teams have been best at converting their shots?

First, we start by using `group_by()` and `summarize()`, similar to what you should have done above.

```
team.sum <- soccer.data %>%
  group_by(shots.team) %>%
  summarise(n.shots = n(), shot.rate = mean(is.goal))
team.sum
```

6. Arsenal and Chelsea have the most shots – how do their shot rates compare?

We can plot the data above using a standard bar-plot.

```
ggplot(data = team.sum, aes(x = shots.team, y = shot.rate)) +
  geom_col()
```

Summary of plot improvements

There are several ways we can improve the chart above. Not all of these fixes are **always** needed, but all of them will be good to know and consider.

Here's a list for safe-keeping.

- Plot title and subtitle
- x and y axis labels
- Rotated x-axis
- y-axis labels as percents
- Order the graph
- Themes

Plot title and subtitle

We can add titles and labels using the `labs()` command. Here's an example:

```
ggplot(data = team.sum, aes(x = shots.team, y = shot.rate)) +
  geom_col() +
  labs (x = "Team", y = "Shot rate",
        title = "Shot rate among women's soccer teams",
        subtitle = "Sample data from StatsBomb")
```

Rotated x-axis

To see the full team names, we rotate the x-axis

```
ggplot(data = team.sum, aes(x = shots.team, y = shot.rate)) +
  geom_col() +
  labs (x = "Team", y = "Shot rate",
        title = "Shot rate among women's soccer teams",
        subtitle = "Sample data from StatsBomb") +
  theme(axis.text.x = element_text(angle = 45, vjust = .6))
```

Labels as percents

Our outcome is a percent: we can appropriately label the y-axis as follows:

```
ggplot(data = team.sum, aes(x = shots.team, y = shot.rate)) +
  geom_col() +
  labs (x = "Team", y = "Shot rate",
        title = "Shot rate among women's soccer teams",
        subtitle = "Sample data from StatsBomb") +
  theme(axis.text.x = element_text(angle = 45, vjust = .6)) +
  scale_y_continuous(labels = scales::percent)
```

Reorder the graph

Note: this uses the `forcats` package, so be sure to load it.

Sometimes, it helps to view the bar chart in a better order.

```
library(forcats)
ggplot(data = team.sum, aes(x = fct_reorder(shots.team, shot.rate), y = shot.rate)) +
  geom_col() +
  labs (x = "Team", y = "Shot rate",
        title = "Shot rate among women's soccer teams",
        subtitle = "Sample data from StatsBomb") +
```

```
  theme(axis.text.x = element_text(angle = 45, vjust = .6)) +
  scale_y_continuous(labels = scales::percent)
```

7. Take a close look at the code above – what is different? Where is R sorting the teams in order of shot rate?
8. Is a re-ordered axis appropriate for scatter plots?

Plotting themes

R comes with a base theme – however, we can expand on those themes fairly easily.

```
p.base <- ggplot(data = team.sum, aes(x = fct_reorder(shots.team, shot.rate), y = shot.rate)) +
  geom_col() +
  labs (x = "Team", y = "Shot rate",
        title = "Shot rate among women's soccer teams",
        subtitle = "Sample data from StatsBomb") +
  theme(axis.text.x = element_text(angle = 45, vjust = .6)) +
  scale_y_continuous(labels = scales::percent)
p.base + theme_classic() + theme(axis.text.x = element_text(angle = 45, vjust = .6))
p.base + theme_bw() + theme(axis.text.x = element_text(angle = 45, vjust = .6))
p.base + theme_minimal() + theme(axis.text.x = element_text(angle = 45, vjust = .6))
p.base + theme_dark() + theme(axis.text.x = element_text(angle = 45, vjust = .6))
```

A few notes here:

- The rotated x-axis has to be added in *after* changing the theme, as the base theme includes non-rotated labels.
- There's no right answer to which theme is best – shop around!

Alternative plot.

One aspect that's missing to our plot above is that we don't observe each team's shot counts: all we get is their shot conversion rates. We can add that information using a scatter plot.

```
ggplot(data = team.sum, aes(n.shots, shot.rate)) +
  geom_point()
```

Of course, now we are missing each team's name. We can add that in as a label

```
ggplot(data = team.sum, aes(n.shots, shot.rate, label = shots.team)) +
  geom_text()
```

This is nice and all, but we could do better.

First, we expand the x-axis to show all teams and the y-axis to start at 0.

```
ggplot(data = team.sum, aes(n.shots, shot.rate, label = shots.team)) +
  geom_text() +
  scale_x_continuous(lim = c(0, 210)) +
  scale_y_continuous(lim = c(0, 0.25))
```

9. Identify the team that, although they aren't taking many shots, tends to be converting at a high rate?
10. Why is it generally a good thing to have a y-axis that starts at 0?

Adding text

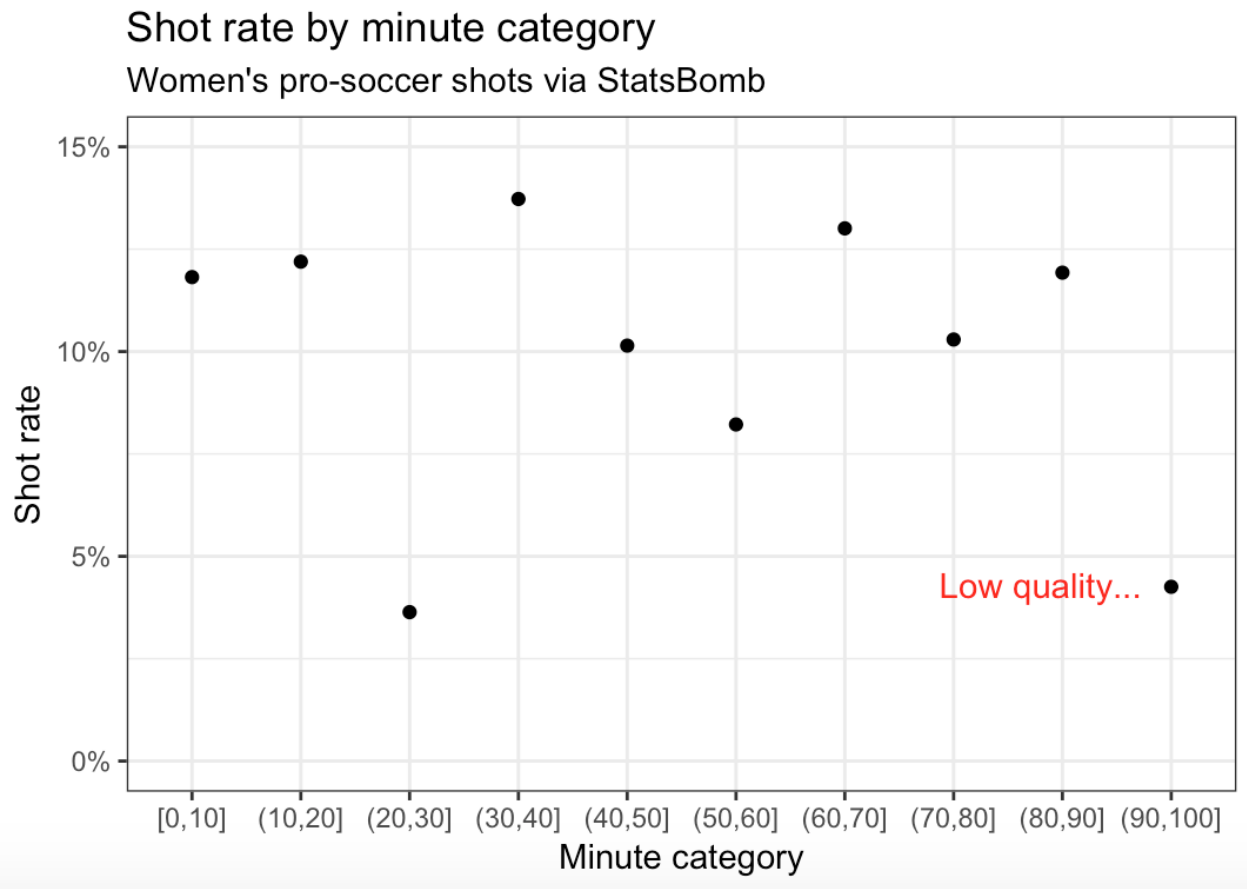
One last step which can be useful – especially in scatter plots – is to use text to identify what each quadrant refers to. For example:

```
ggplot(data = team.sum, aes(n.shots, shot.rate, label = shots.team)) +
  geom_text() +
  scale_x_continuous(lim = c(0, 210)) +
  scale_y_continuous(lim = c(0, 0.25)) +
  annotate("text", x = 175, y = 0.25, label = "Lots of quality shots",
           colour = "red")
```

Building it all together

11. Make the following plot. A bit of code will get you started

```
soccer.data <- soccer.data %>%
  mutate(minute.cat = cut(shots.minute, seq(0, 100, 10), include.lowest = TRUE))
```


Shot rate by minute category
Women's pro-soccer shots via StatsBomb
15%
10%
5%
0%
Shot rate
Low quality...
[0,10]
(10,20]
(20,30]
(30,40]
(40,50]
(50,60]
(60,70]
(70,80]
(80,90]
(90,100]
Minute category