TRƯỜNG ĐẠI HỌC MỎ - ĐỊA CHẤT

KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO HỌC THUẬT

PHÂN TÍCH DỮ LIỆU STATSBOMB VỚI NGÔN NGỮ R

PHẠM AN CƯƠNG

12/2024

MỤC LỤC

1. GIỚI THIỆU

StatsBomb là một trong những nguồn dữ liệu phân tích thể thao hàng đầu, cung cấp thông tin chi tiết về các trận đấu bóng đá, bao gồm các chỉ số cầu thủ, đội bóng và các sự kiện trong trận. Việc sử dụng ngôn ngữ R để xử lý và phân tích dữ liệu StatsBomb mang lại nhiều lợi ích nhờ vào khả năng lập trình linh hoạt và mạnh mẽ trong thống kê.

2. MỤC TIÊU

Báo cáo này nhằm mục đích:

- Trình bày quy trình thu thập và chuẩn bị dữ liệu StatsBomb.
- Minh họa các kỹ thuật phân tích và trực quan hóa dữ liệu bằng R.
- Cung cấp ví dụ thực tế về việc áp dụng phân tích dữ liệu trong bóng đá.

3. PHƯƠNG PHÁP

3.1. Thu thập dữ liệu

Dữ liệu StatsBomb có thể được truy cập qua các định dạng khác nhau như JSON hoặc CSV. Người dùng có thể tải xuống trực tiếp từ trang web của StatsBomb hoặc qua API.
https://github.com/statsbomb/StatsBombR

3.2 Chuẩn bị dữ liệu

3.2.1 Truy cập dữ liệu

Gói này cung cấp quyền truy cập vào dữ liệu chi tiết theo từng sự kiện từ nhiều giải đấu và cuộc thi bóng đá. Bao gồm thông tin về đường chuyền, cú sút, pha vào bóng, lỗi và các sự kiện quan trọng khác của trận đấu. Dữ liệu được cấu trúc theo cách nắm bắt được sự phức tạp và sắc thái của các trận đấu bóng đá, cho phép người dùng thực hiện các phân tích chuyên sâu.
Để cài đặt gói dữ liệu trong R, cần cài đặt gói 'Devtools':
install.packages("devtools")
install.packages("remotes")
remotes::install_version("SDMTools", "1.1-221")
Sau đó, cài đặt StatsBombR:
devtools::install_github("statsbomb/StatsBombR")

3.2.2 Các chức năng thân thiện với người dùng

"StatsBombR" cung cấp một bộ chức năng cho phép người dùng dễ dàng truy vấn, lọc và thao tác dữ liệu. Các chức năng này được thiết kế trực quan, giúp người dùng dễ dàng tập trung vào phân tích của mình hơn là xử lý dữ liệu. Các chức năng chính bao gồm:

get_match_data(): Truy xuất dữ liệu chi tiết cho một trận đấu cụ thể.

get_team_data(): Cung cấp số liệu thống kê tổng hợp cho một đội cụ thể.

get_player_data(): Truy cập số liệu thống kê và số liệu hiệu suất của từng cầu thủ.

filter_events(): Lọc dữ liệu sự kiện dựa trên các tiêu chí đã chỉ định (ví dụ: loại sự kiện, cầu thủ, địa điểm).

3.2.3 Công cụ trực quan hóa

Gói này bao gồm các công cụ để tạo trực quan hóa sâu sắc. Người dùng có thể tạo nhiều loại bản đồ và biểu đồ khác nhau để hiểu rõ hơn về hành vi của cầu thủ và đội. Một số chức năng trực quan hóa bao gồm:

Sơ đồ chuyền bóng (Pass Maps): Biểu diễn trực quan các kiểu chuyền bóng và kết nối giữa các cầu thủ.

Sơ đồ cú sút (Shot Maps): Bản đồ chi tiết hiển thị vị trí và kết quả của các cú sút trong một trận đấu.

Sơ đồ nhiệt (Heat Maps): Hình ảnh trực quan làm nổi bật các khu vực trên sân mà cầu thủ hoặc đội bóng hoạt động tích cực nhất.

Sơ đồ sự kiện (Event Plots): Biểu đồ hiển thị các sự kiện cụ thể, chẳng hạn như bàn thắng, hỗ trợ và các hành động phòng thủ quan trọng.

Sử dụng các gói R như dplyr, tidyr để làm sạch và chuẩn hóa dữ liệu. Các bước chuẩn bị dữ liệu bao gồm:

- Xử lý các giá trị thiếu.
- Chuyển đổi định dạng thời gian.
- Tạo các biến mới cho phân tích (ví dụ: tỷ lệ ghi bàn, thời gian cầm bóng).

3.3. Phân tích Dữ liệu

Với dữ liệu và công cụ được cung cấp, người dùng có thể thực hiện các phân tích thống kê nâng cao. Bao gồm đánh giá hiệu suất của cầu thủ, chiến lược của đội và kết quả trận đấu bằng nhiều phương pháp thống kê và học máy khác nhau. Gói này hỗ trợ tích hợp với các thư viện R ggplot2, dplyr và caret để lập mô hình thống

kê, học máy và trực quan hóa dữ.Sử dụng các gói R stats cho các phân tích thống kê cơ bản:

- Phân tích mô tả: Tính toán các chỉ số thống kê cơ bản.
- Phân tích hồi quy: Xây dựng mô hình để dự đoán hiệu suất cầu thủ dựa trên các chỉ số.

Gói "StatsBombR" là một nguồn tài nguyên mạnh mẽ cho bất kỳ ai quan tâm đến phân tích bóng đá. Bằng cách cung cấp quyền truy cập vào dữ liệu bóng đá chi tiết và đáng tin cậy cùng với các công cụ thân thiện với người dùng để phân tích và trực quan hóa. Cho dù bạn là một nhà phân tích dày dạn kinh nghiệm hay một người hâm mộ bóng đá có sở thích về dữ liệu, "StatsBombR" cung cấp các khả năng bạn cần để khám phá bóng đá theo một góc nhìn mới. Với các tính năng mở rộng và chức năng mạnh mẽ, "StatsBombR" là một công cụ thiết yếu để phân tích bóng đá hiện đại, cho phép người dùng khai thác toàn bộ tiềm năng của dữ liệu bóng đá và đưa ra các quyết định sáng suốt dựa trên các phân tích sâu sắc và toàn diện.

3.3.1 Cài đặt gói dữ liệu

```
install.packages("devtools", repos="http://cran.us.r-project.org")
install.packages("remotes", repos="http://cran.us.r-project.org")
remotes::install_version("SDMTools", "1.1-221", repos="http://cran.us.r-project.org")
devtools::install_github("statsbomb/StatsBombR", repos="http://cran.us.r-project.org")
devtools::install_github("FCrSTATS/SBpitch", repos="http://cran.us.r-project.org")
```

3.3.2 Tải gói dữ liệu

```
library(tidyverse)
library(StatsBombR)
```

3.3.3 Lấy dữ liệu StatsBomb

```
Comps <- FreeCompetitions()
```

Lệnh này sẽ lấy tất cả các trận đấu miễn phí do StatsBomb cung cấp. Có 74 tập dữ liệu trận đấu miễn phí. Để lọc dữ liệu theo một trận đấu cụ thể, có thể lọc bằng cách chỉ định bất kỳ hai tên cột nào từ tập dữ liệu. Để chính xác hơn, có thể kiểm tra cấu trúc của dữ liệu bằng cách chạy str(df) trong đó df là tên bạn đã lưu khung dữ liệu. Trong tập dữ liệu StatsBomb này, competition_id và season_id là các cột số.

```
Comps = Comps %>%
```

```
filter(competition_id == "" & season_id == "")
```
Đối với ví dụ giải Copa America gần đây. competition_id cho Copa America là 223 và season_id là 282.
```
copa <- FreeCompetitions() %>%
filter(competition_id == 223 & season_id == 282)
copa_matches <- FreeMatches(copa)
#1 This pulls all the matches for the desired competition.
Copa_Stats_Bomb_Data <- free_allevents(MatchesDF = copa_matches, Parallel = T)
#2 This pulls all the event data for the matches that are chosen.
copa_data_clean = allclean(Copa_Stats_Bomb_Data)
#3 Extracts lots of relevant information such as x/y coordinates.
# More information can be found in the package info.

names(copa_data_clean)
```

3.3.4 Lọc theo nhóm cú sút và bàn thắng

Cách lọc dữ liệu để có được cú sút và bàn thắng của các đội và cầu thủ. Đối với ví dụ này, chúng ta sẽ tập trung vào các cột sau: team.name, type.name, shot.outcome.name và player.name. Để kiểm tra các giá trị có trong từng cột này, có thể chạy mã bên dưới. Điều này sẽ giúp hiểu rõ hơn về các biến và giúp phân tích dễ hiểu hơn.
```
unique(copa_data_clean$team.name)
unique(copa_data_clean$type.name)
unique(copa_data_clean$shot.outcome.name)
unique(copa_data_clean$player.name)
```

3.3.5 Lọc theo tổng số bàn thắng

```
shots_goals = copa_data_clean %>%
group_by(team.name) %>%
#1: This code groups the data by team, so that whatever operation we perform
# on it will be done on a team by team basis. I.e, we will find the shots and
# goals for every team one by one.
summarise(shots = sum(type.name=="Shot", na.rm = TRUE),
#2: Summarise takes whatever operation we give it and produces a
# new, separate table out of it. The vast majority of summarise
# uses come after group_by.
goals = sum(shot.outcome.name=="Goal", na.rm = TRUE))
#3: shots = sum(type.name=="Shot", na.rm = TRUE) is telling it to create a new
# column called shots that sums up all the rows under the type.name column
# that contain the word “Shot”. na.rm = TRUE tells it to ignore any NAs within
# that column. shot.outcome.name=="Goal", na.rm = TRUE)
```

```
# does the same but for goals.
```

3.3.6 Lọc theo từng trận

```
# Adding in the n_distinct(match_id) means we are dividing the number of
# shots/goals by each distinct (or unique) instance of a match, for every team.
# I.e, we are dividing the numbers per game.
shots_goals = copa_data_clean %>%
group_by(team.name) %>%
summarise(shots = sum(type.name=="Shot", na.rm =
TRUE)/n_distinct(match_id),
goals = sum(shot.outcome.name=="Goal", na.rm = TRUE)/n_distinct(match_id))
```

3.3.7 Lọc các cú sút của cầu thủ và đường chuyền quan trọng

Lọc dữ liệu để có được cú sút và đường chuyền quan trọng. Đối với điều này, chúng ta sẽ tập trung vào các cột sau: player.name, player.id, type.name và pass.shot_assist. Để kiểm tra các giá trị có trong từng cột này, bạn có thể chạy mã bên dưới

```
unique(copa_data_clean$player.name)
unique(copa_data_clean$player.id)
unique(copa_data_clean$type.name)
unique(copa_data_clean$pass.shot_assist)
```

3.3.8 Theo tổng số cú sút và đường chuyền

```
player_shots_keypasses = copa_data_clean %>%
group_by(player.name, player.id) %>%
#1: This code groups the data by player,
# so that whatever operation we perform on it will be done on a player by
# player basis. I.e, we will find the shots and goals for every player one by one.
summarise(shots = sum(type.name=="Shot", na.rm = TRUE),
keypasses = sum(pass.shot_assist==TRUE, na.rm = TRUE))
```

3.3.9 Nhận dữ liệu số phút đã thi đấu

```
player_minutes = get.minutesplayed(copa_data_clean)
#1: This function gives us the minutes played in each match by ever
# player in the dataset.
player_minutes = player_minutes %>%
group_by(player.id) %>%
summarise(minutes = sum(MinutesPlayed))
#2: Now we group that by player and sum it altogether to get
# their total minutes played.
```

3.3.10 Kết hợp dữ liệu phút thi đấu với khung dữ liệu cú sút của cầu thủ

```
player_shots_keypasses = left_join(player_shots_keypasses, player_minutes)
#1: left_join allows us to combine our shots and key passes table and our
# minutes table, with the the player.id acting as a reference point.
player_shots_keypasses = player_shots_keypasses %>%
mutate(nineties = minutes/90)
#2: `mutate` is a `dplyr` function that creates a new column. In this instance
# we are creating a column that divides the minutes totals by 90,
# giving us each players number of 90s played.
player_shots_keypasses = player_shots_keypasses %>%
mutate(shots_per90 = shots/nineties,
kp_per90 = keypasses/nineties,
shots_kp_per90 = shots_per90+kp_per90)
#3: Finally we divide our totals by our number of 90s to get our totals
# per 90s columns for shots and key passes.
# We also calculate the sum of these two columns.
```

3.3.11 Lọc phút cho những cầu thủ đã chơi ít nhất ba trận 90 phút (270 phút)

```
player_shots_keypasses = player_shots_keypasses %>%
filter(minutes>270)
#1: This code filters the data to only include players
# who have played at least 270 minutes for a fair comparison.
```

3.3.12 Số đường chuyền một phần ba cuối sân

```
passes = copa_data_clean %>%
filter(type.name=="Pass" & is.na(pass.outcome.name)) %>%
#1: This code filters the data to only include passes that have
# been completed. In this data, NA denotes a complete pass.
filter(location.x<80 & pass.end_location.x>=80) %>%
#2: This code filters the data to only include passes into the final third.
group_by(player.name) %>%
summarise(f3_passes = sum(type.name=="Pass"))
#3: This code groups the data by player and sums up the number of passes
# into the final third for each player.
```

3.3.13 Cầu thủ cụ thể chuyền bóng vào phần ba cuối sân

```
player_passes = copa_data_clean %>%
filter(type.name=="Pass" & is.na(pass.outcome.name) &
player.name=="Federico Santiago Valverde Dipetta") %>%
filter(location.x<80 & pass.end_location.x>=80)
```

3.4. Trực quan hóa Dữ liệu

Sử dụng ggplot2 để tạo ra các biểu đồ trực quan như:

- Biểu đồ phân phối các chỉ số cầu thủ.
- Biểu đồ tương quan giữa các biến (số cú sút và số bàn thắng).

3.4.1 Tải gói dữ liệu

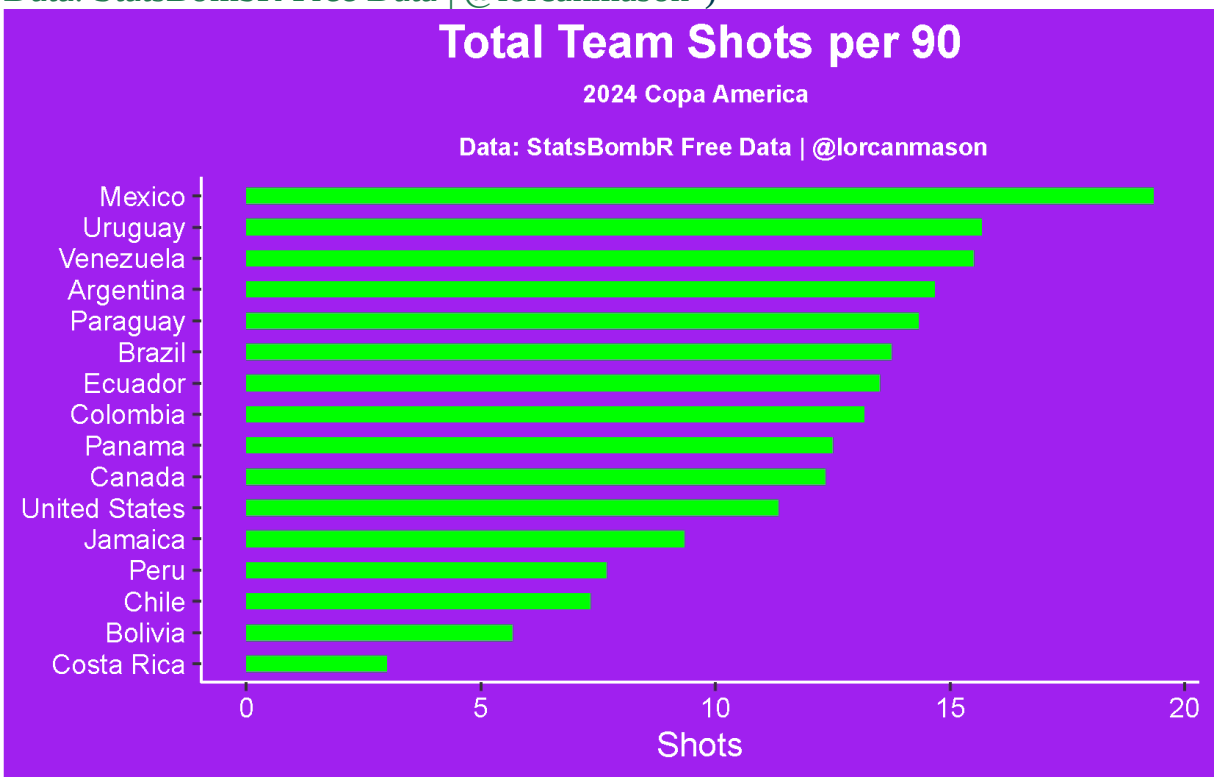
```
library(ggplot2)
#1: This is the package we use to create our plots.
library(ggrepel)
#2: This is the package we use to modify our plots.
library(SBpitch)
#3: This is the package we use to create our pitch.
library(scales)
#4: This is the package we use to modify our scales.
library(prismatic)

#5: This is the package we use to modify our colours
```

3.4.2 Sơ đồ các cú sút và bàn thắng

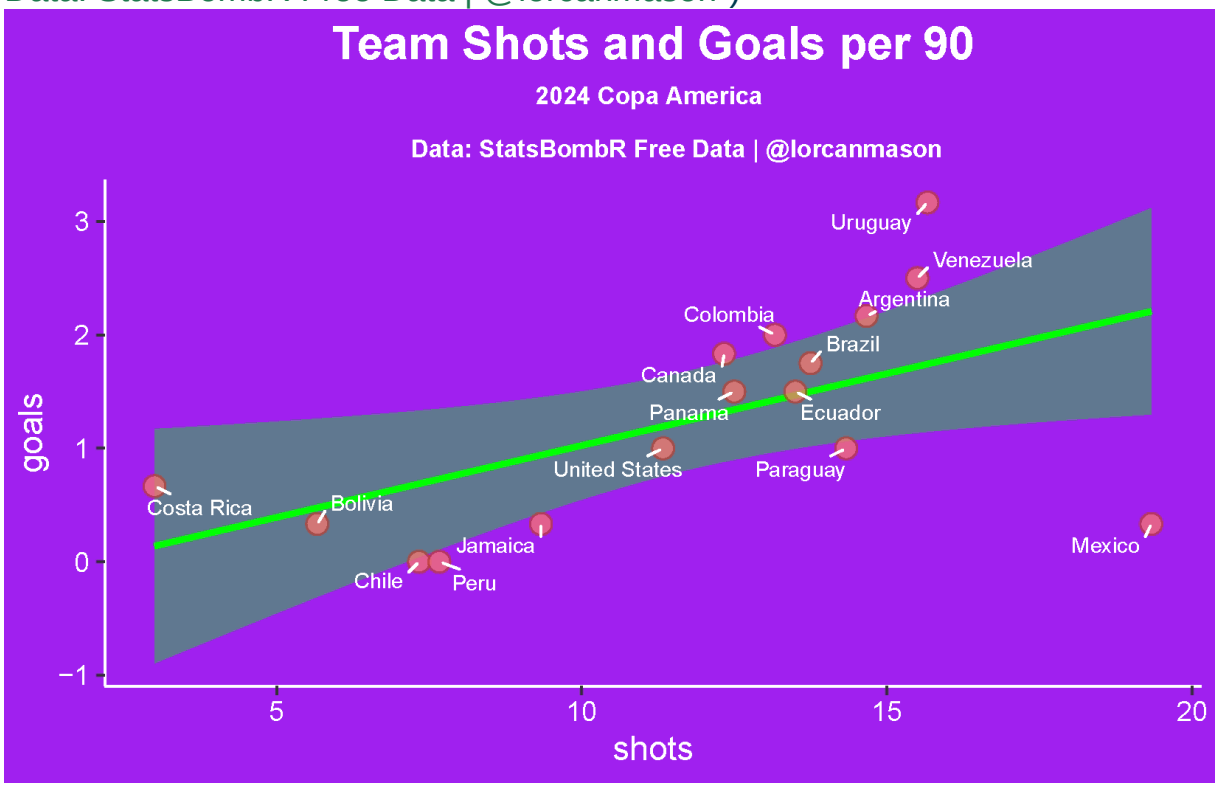
```
ggplot(data = shots_goals,
aes(x = reorder(team.name, shots), y = shots)) +
#1: This code sets up the plot and tells it what data to use and
# what columns to use for the x and y axis.
geom_bar(stat = "identity", width = 0.5, fill="green") +
#2: This code tells it to create a bar plot with the data we have given it.
# stat = "identity" tells it to use the data as it is, width = 0.5 sets the
# width of the bars, and fill = "green" sets the colour of the bars.
labs(y="Shots") +
#3: This code sets the label for the y axis.
coord_flip() +
#4: This code flips the x and y axis.
theme(
axis.title.y = element_blank(),
legend.position = "none",
plot.background = element_rect(fill = "purple", colour = "purple"),
panel.background = element_rect(fill = "purple", colour = "purple"),
panel.grid.major = element_line(colour = "purple"),
panel.grid.minor = element_blank(),
axis.line = element_line(colour = "white"),
axis.text = element_text(colour = "white"),
```

```
axis.title = element_text(colour = "white"),
plot.title = element_text(colour = "white", hjust=.5, face="bold", size = 15),
plot.subtitle = element_text(colour = "white", hjust=.5, face="bold", size = 8)) +
#5: This code sets the theme for the plot. It sets the background colour
# grid lines, axis lines, axis text, axis title, plot title, and plot subtitle.
labs(title = "Total Team Shots per 90",
subtitle = "2024 Copa America
Data: StatsBombR Free Data | @lorcanmason")
```




```
#6: This code sets the title and subtitle for the plot.
ggplot(shots_goals, aes(x = shots, y = goals, label = team.name)) +
#1: This code sets up the plot and tells it what data to use and what
# columns to use for the x and y axis.
geom_smooth(method = "lm", color = "green", fill = "green") +
#2: This code tells it to create a linear regression line on the plot.
# method = "lm" tells it to use a linear model, colour = "green" sets the
# colour of the line, and fill = "green" sets the fill colour of the line.
geom_point(aes(fill = "green", color = after_scale(clr_darken(fill, 0.3))),
shape = 21,
alpha = .75,
size = 3) +
#3: This code tells it to create points on the plot. aes(fill = "green",
9
# color = after_scale(clr_darken(fill, 0.3))) sets the fill and colour of
# the points, shape = 21 sets the shape of the points, alpha = .75 sets
# the transparency of the points, and size = 3 sets the size of the points.
geom_text_repel(size = 2.5, color = "white", min.segment.length = unit(0.1,
"lines")) +
```

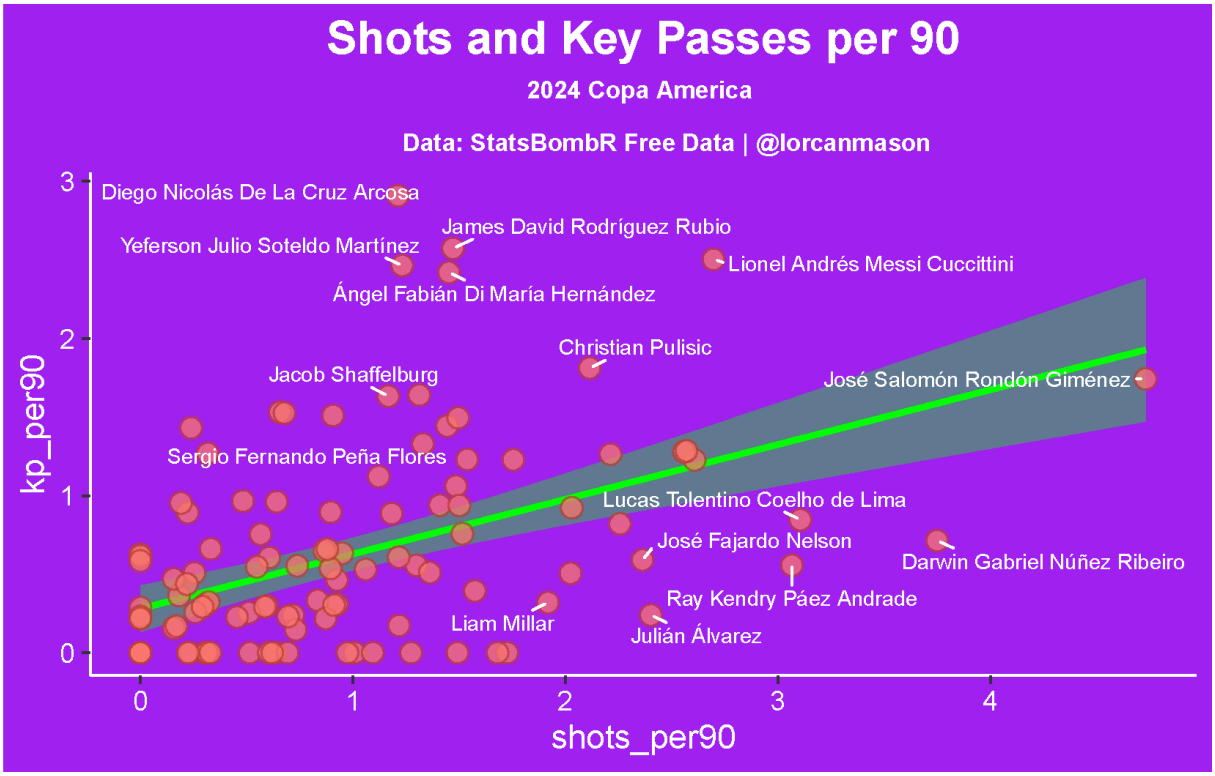
```
#4: This code tells it to create text labels on the plot. size = 2.5 sets
# the size of the text, colour = "white" sets the colour of the text,
# and min.segment.length = unit(0.1, "lines") sets the minimum length of the
segments.
theme(
legend.position = "none",
plot.background = element_rect(fill = "purple", colour = "purple"),
panel.background = element_rect(fill = "purple", colour = "purple"),
panel.grid.major = element_line(colour = "purple"),
panel.grid.minor = element_blank(),
axis.line = element_line(colour = "white"),
axis.text = element_text(colour = "white"),
axis.title = element_text(colour = "white"),
plot.title = element_text(colour = "white", hjust=.5, face="bold", size = 15),
plot.subtitle = element_text(colour = "white", hjust=.5, face="bold", size = 8)) +
labs(title = "Team Shots and Goals per 90",
subtitle = "2024 Copa America
Data: StatsBombR Free Data | @lorcanmason")
```



3.4.3 Sơ đồ các cú sút và đường chuyền chính của cầu thủ

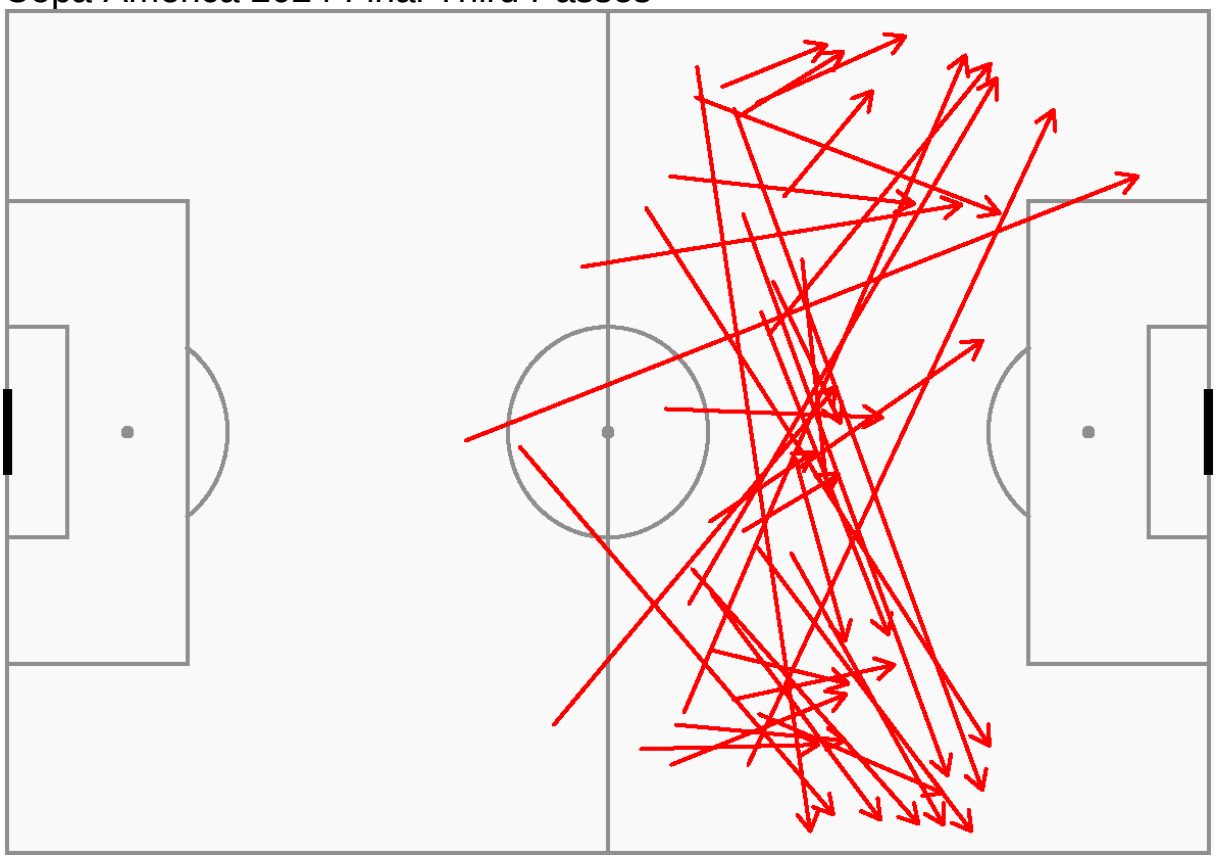
```
ggplot(player_shots_keypasses, aes(x = shots_per90, y = kp_per90,
label = player.name)) +
geom_smooth(method = "lm", color = "green", fill = "green") +
geom_point(aes(fill = "green", color = after_scale(clr_darken(fill, 0.3))),
```

```
shape = 21,
alpha = .75,
size = 3) +
geom_text_repel(size = 2.5, color = "white", min.segment.length =
unit(0.1, "lines")) +
theme(
legend.position = "none",
plot.background = element_rect(fill = "purple", colour = "purple"),
panel.background = element_rect(fill = "purple", colour = "purple"),
panel.grid.major = element_line(colour = "purple"),
panel.grid.minor = element_blank(),
axis.line = element_line(colour = "white"),
axis.text = element_text(colour = "white"),
axis.title = element_text(colour = "white"),
plot.title = element_text(colour = "white", hjust=.5, face="bold", size =
15),
plot.subtitle = element_text(colour = "white", hjust=.5, face="bold", size =
8)) +
labs(title = "Shots and Key Passes per 90",
subtitle = "2024 Copa America
Data: StatsBombR Free Data | @lorcanmason")
```

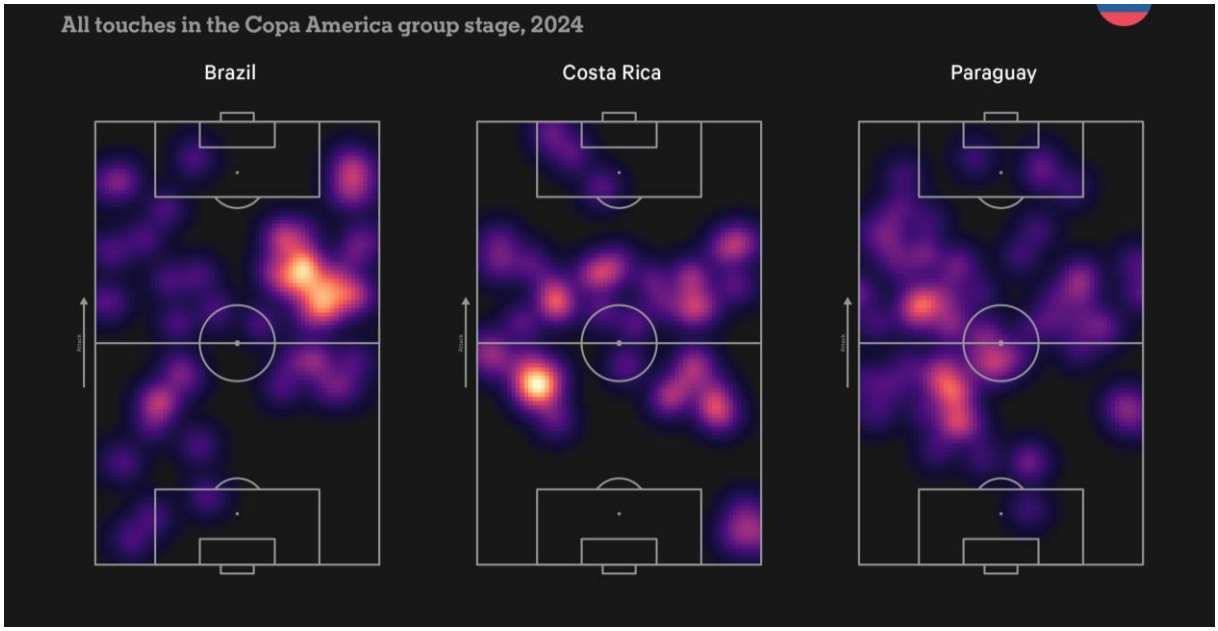


3.4.4 Sơ đồ cho các đường chuyền vào phần ba cuối sân

Copa America 2024 Final Third Passes



3.4.5 Sơ đồ nhiệt (Heat Maps)



4. KẾT QUẢ

Một số phát hiện từ việc phân tích dữ liệu StatsBomb bằng R:

- Sự khác biệt về hiệu suất giữa các cầu thủ trong các giải đấu khác nhau.
- Mối tương quan mạnh giữa tỷ lệ chuyền bóng và số bàn thắng ghi được.

5. THẢO LUẬN

Việc sử dụng R để phân tích dữ liệu StatsBomb giúp các nhà phân tích thể thao có cái nhìn sâu sắc hơn về hiệu suất cầu thủ và chiến thuật đội bóng. Hơn nữa, R cung cấp một nền tảng mạnh mẽ cho việc phát triển các mô hình dự đoán.

6. KẾT LUẬN

Ngôn ngữ R là một công cụ mạnh mẽ cho việc phân tích dữ liệu thể thao, và dữ liệu StatsBomb cung cấp một nguồn thông tin phong phú để phát triển các kỹ thuật phân tích hiện đại. Các nghiên cứu tiếp theo có thể mở rộng quy mô và độ phức tạp của phân tích để khám phá sâu hơn các yếu tố ảnh hưởng đến thành công trong bóng đá.

TÀI LIỆU THAM KHẢO

[1] StatsBomb. (2023). StatsBomb Data.

[2] Wickham, H. (2016). ggplot2: Elegant Graphics for Data Analysis. Springer.

[3] Grolemund, G., & Wickham, H. (2016). R for Data Science. O'Reilly Media.