Análisis de disparos usando R

Paquete StatsbombR

El paquete StatsbombR de R ofrece data de libre acceso de una gran cantidad de competiciones, con dicha información es posible realizar una gran cantidad de análisis ya que la base cuenta con una gran cantidad de eventos.

En esta oportunidad se presenta un análisis sobre los disparos realizados por los jugadores que participaron en la Eurocopa 2020, específicamente en los siete juegos que participó Italia. En los gráficos a generar se muestra información sobre el xG ó goles esperados y el resultado del disparo, es decir, si el disparo terminó en gol, en parada del portero ó si fué desviado.

Un análisis más completo incluiría el detalle de los minutos jugados por cada jugador con el fin de garantizar y tener una comparación justa, en este caso este análisis no considera esa restricción.

```
#CARGO LIBRERIAS NECESARIAS
library(StatsBombR)
library(lubridate)
library(dplyr)
library(purrr)
library(plotly)
library(png)
library(magick)
```

Acceso a la data

Para esta opción es necesario tener instalado el paquete StatsbombR, para más detalles sobre el proceso de instalación hacer clik aquí.

La data libre disponible se accede con este comando,

```
#DATA DISPONIBLE
free_data <- FreeCompetitions()
```

De dicha data podemos ver las competiciones disponibles,

```
#COMPETICIONES DISPONIBLES
table(free_data$competition_name)
```

```
##
##        Champions League FA Women's Super League          FIFA World Cup
##                      15                       3                       1
##                 La Liga                    NWSL          Premier League
##                      17                       1                       1
##               UEFA Euro       UEFA Women's Euro       Women's World Cup
##                       1                       1                       1
```

Destacan competiciones recientes como la Eurocopa del 2020 y algunos partidos de la Liga durante 17 temporadas. Se tiene información del mundial de Rusia 2018, mundial femenino del 2019 y Eurocopa del 2022.

Luego de elegir la competición se usa la función "FreeMatches" para acceder a los 51 partidos de la Eurocopa, luego se hace el filtro de los partidos de Italia. Una vez hecho esto se procede a emplear la función "StatsBombFreeEvents" para acceder a todos los eventos de cada juego, finalmente la función "allclean" limpia y agrega columnas extra.

```
#SELECCIONO COMPETICION
#PARTIDOS DISPONIBLES DE LA EURO
Matches <- FreeMatches(free_data[39,])

#PARTIDOS EN ESPECIFICO
Matches1 <- Matches[which(Matches$home_team.home_team_name=="Italy" |
                            Matches$away_team.away_team_name=="Italy"),]

#HAGO CONSULTA DE LOS PARTIDOS DE LA EURO
StatsBombData <- StatsBombFreeEvents(MatchesDF = Matches1, Parallel = T)

#REALIZO LIMPIEZA DE LAS COLUMNAS RELACIONADAS CON LAS COORDENADAS
events = allclean(StatsBombData)
```

Los eventos disponibles son,

```
#EVENTOS DISPONIBLES
table(events$type.name)
```

```
##
##             50/50     Bad Behaviour     Ball Receipt*     Ball Recovery
##                 6                 3              7911               634
##             Block             Carry         Clearance      Dispossessed
##               260              6626               283               160
##           Dribble     Dribbled Past              Duel             Error
##               230               145               441                 3
##    Foul Committed          Foul Won       Goal Keeper          Half End
##               229               220               250                44
##        Half Start   Injury Stoppage      Interception        Miscontrol
##                44                35               227               189
##           Offside  Own Goal Against      Own Goal For              Pass
##                 2                 1                 1              8192
##        Player Off         Player On          Pressure Referee Ball-Drop
##                 3                 3              2418                10
##            Shield              Shot       Starting XI      Substitution
##                11               205                14                71
##    Tactical Shift
##                21
```

Una descripción detallada del significado de cada evento se muestra aquí.

En cuanto a las categorías asociadas a los resultados del disparo se tienen,

```
#EVENTOS DISPONIBLES
table(events$shot.outcome.name)
```

```
##
##          Blocked             Goal            Off T             Post
##               54               27               66                4
##            Saved Saved Off Target    Saved to Post          Wayward
##               39                2                2               11
```

El siguiente código filtra los eventos a los disparos y se hace un enfoque en los disparos que no sean desde el punto de penal. Para mostrar un ejemplo se selecciona a Lorenzo Insigne.

```
#BASE DE DISPAROS SIN CONTABILIZAR TIROS DE PENALTY
shots = events %>%
  filter(type.name=="Shot" & (shot.type.name!="Penalty" | is.na(shot.type.name))
         & player.name=="Lorenzo Insigne") #1
```

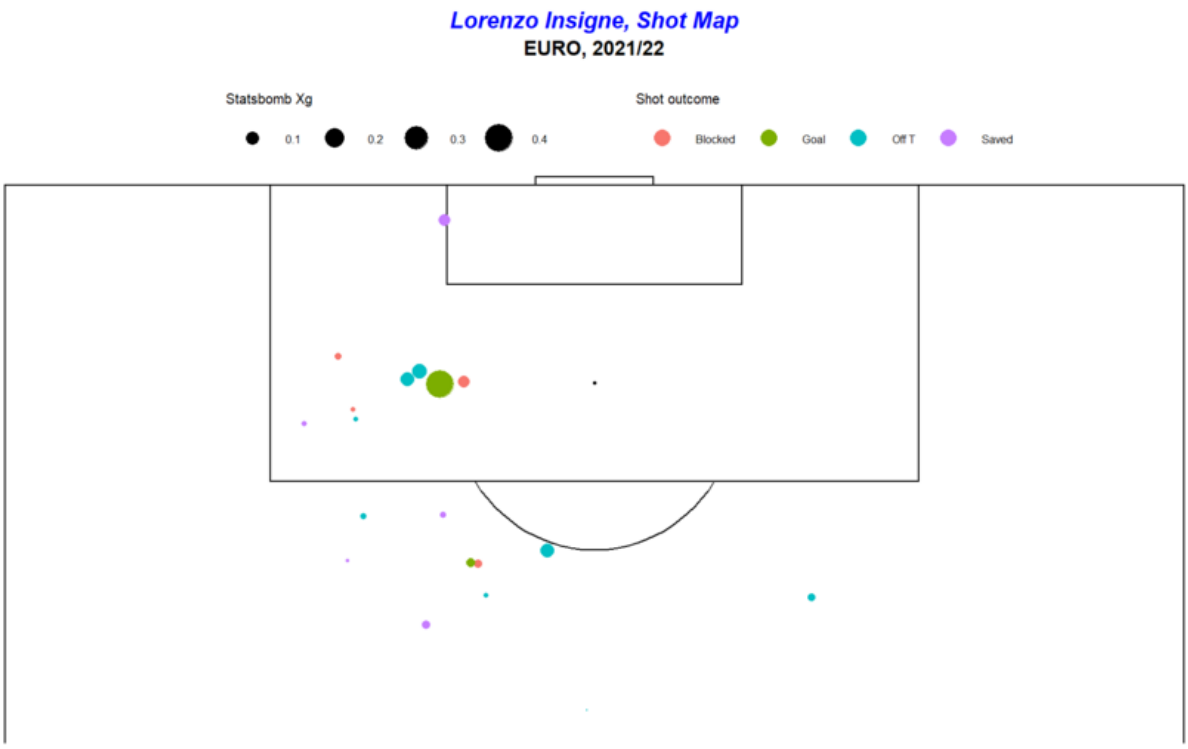
El siguiente código permite graficar un campo de futbol y los eventos relacionados con disparos de Lorenzo Insigne en los 7 juegos disputados por Italia,

```
#GRAFICO - LORENZO INSIGNE
insig <- ggplot() +
  annotate("rect",xmin = 0, xmax = 120, ymin = 0, ymax = 80, fill = NA, colour = "black",
           size = 0.6) +
  annotate("rect",xmin = 0, xmax = 60, ymin = 0, ymax = 80, fill = NA, colour = "black",
           size = 0.6) +
  annotate("rect",xmin = 18, xmax = 0, ymin = 18, ymax = 62, fill = NA, colour = "black",
           size = 0.6) +
  annotate("rect",xmin = 102, xmax = 120, ymin = 18, ymax = 62, fill = NA,
        colour = "black", size = 0.6) +
  annotate("rect",xmin = 0, xmax = 6, ymin = 30, ymax = 50, fill = NA,
        colour = "black", size = 0.6) +
  annotate("rect",xmin = 120, xmax = 114, ymin = 30, ymax = 50, fill = NA,
        colour = "black", size = 0.6) +
  annotate("rect",xmin = 120, xmax = 120.5, ymin =36, ymax = 44, fill = NA,
        colour = "black", size = 0.6) +
  annotate("rect",xmin = 0, xmax = -0.5, ymin =36, ymax = 44, fill = NA,
        colour = "black", size = 0.6) +
  annotate("segment", x = 60, xend = 60, y = -0.5, yend = 80.5, colour = "black",
        size = 0.6)+
  annotate("segment", x = 0, xend = 0, y = 0, yend = 80, colour = "black", size = 0.6)+
  annotate("segment", x = 120, xend = 120, y = 0, yend = 80, colour = "black",
        size = 0.6)+
  theme(rect = element_blank(),
        line = element_blank()) +
  # add penalty spot right
  annotate("point", x = 108 , y = 40, colour = "black", size = 1.05) +
  annotate("path", colour = "black", size = 0.6,
           x=60+10*cos(seq(0,2*pi,length.out=2000)),
           y=40+10*sin(seq(0,2*pi,length.out=2000)))+
  # add centre spot
  annotate("point", x = 60 , y = 40, colour = "black", size = 1.05) +
  annotate("path", x=12+10*cos(seq(-0.3*pi,0.3*pi,length.out=30)), size = 0.6,
           y=40+10*sin(seq(-0.3*pi,0.3*pi,length.out=30)), col="black") +
  annotate("path", x=107.84-10*cos(seq(-0.3*pi,0.3*pi,length.out=30)), size = 0.6,
```

```
          y=40-10*sin(seq(-0.3*pi,0.3*pi,length.out=30)), col="black") +
  geom_point(data = shots, aes(x = location.x, y = location.y, size = shot.statsbomb_xg
                               ,colour = factor(shot.outcome.name)))+
  scale_size(range = c(0, 10))+
  labs(title = "Lorenzo Insigne, Shot Map", subtitle = "EURO, 2021/22",
       size = "Statsbomb Xg", colour = "Shot outcome")+

  theme(
    plot.title = element_text(color="blue", size=18, face="bold.italic",hjust = 0.5),
    plot.subtitle = element_text(color="black", size=16, face="bold",hjust = 0.5),
    axis.title.x = element_blank(),
    axis.title.y = element_blank(),
    axis.text.y=element_blank(),
    axis.text.x=element_blank(),
    legend.direction = "horizontal",
    legend.position = "top",
    legend.margin = margin(c(20, 10, -85, 50)),
    legend.key.size = unit(1.5, "cm")  ) +
  guides(size = guide_legend(title.position = "top"),
         colour = guide_legend(title.position = "top",
        override.aes = list(size = 6, fill = "black"))) +
  coord_flip(xlim = c(88, 125))
```

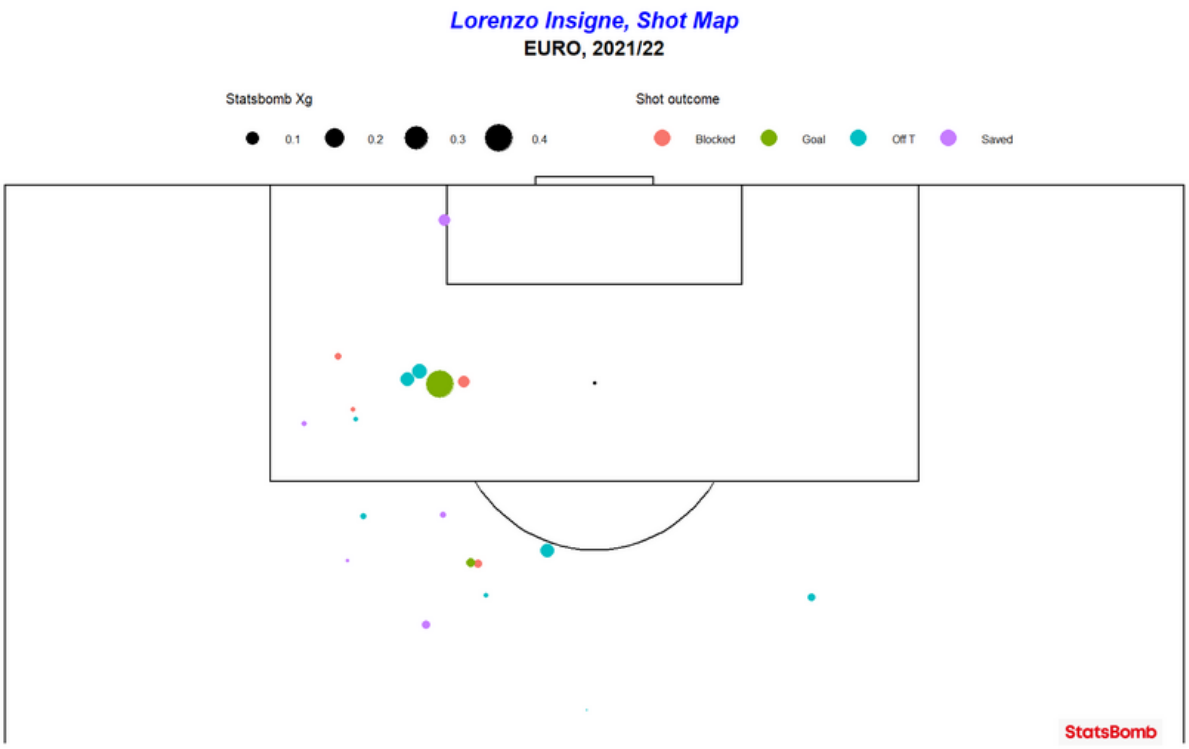
El resultado se presenta a continuación, el código que sigue guarda el gráfico en la carpeta images.

```
#PROCESO PARA GUARDAR PLOT
#ESTABLEZCO ANCHO Y LARGO DE LA PANTALLA
screen_width <- 1920
screen_height <- 1080

#GUARDO
ggsave("images/Insig.png", insig, width = screen_width/130, height = screen_height/130,
       dpi = 130)

#PROCESO PARA AGREGAR LOGO STATSBOMB
#LEO IMAGENES
plot <- image_read("images/Insig.png")
plot1 <- image_read("images/Insig.png")%>%  image_resize(900)

plot1
```



Con el fin de agregar el logo de la fuente de datos, se realiza el siguiente proceso,

```
#CARGO IMAGEN DEL LOGO DE STATSBOMB
logo <- image_read("images/logo.png") %>%  image_resize(150)

#OBTENGO DIMENSIONES DEL PLOT
plot_height <- magick::image_info(plot)$height
plot_width <- magick::image_info(plot)$width

#OBTENGO DIMENSIONES DEL LOGO
logo_width <- magick::image_info(logo)$width
logo_height <- magick::image_info(logo)$height

#CALCULO COORDENADA Y PARA UBICAR LOGO
y <- plot_height - logo_height - plot_height * 0.01

#CALCULO COORDENADA X PARA UBICAR LOGO
x <- plot_width * 0.85

coord <- paste0("+",x,"+",y)

#GRAFICO MAS LOGO
new <- plot %>%
  image_composite(logo, offset = coord)

new %>%  image_resize(900)
```




```
#GUARDO PLOT FINAL
image_write(new, path = "final_plots/Insig_logo.png", format = "png")
```

En el gráfico anterior se puede apreciar que cada punto es un disparo, el tamaño del punto está relacionado con el xG (expected goal) la cual es una métrica que indica que tan probable es que el jugador remantando desde esa posición consiga anotar. En color se indica el resultado del remate, específicamente para el caso anterior vemos que Lorenzo Insigne tuvo 19 disparos, los cuales se dividen en,

```
#DISTRIBUCION DE OUTCOME DE DISPAROS
table(shots$shot.outcome.name)
```

```
##
## Blocked    Goal   Off T   Saved
##       4       2       8       5
```

- 2 remates que terminaron en gol de los cuales uno de ellos tenía un xG muy bajo,
- 4 remates bloqueados ya sea por un defensa o un jugador del propio equipo.
- 8 remates fuera de la arquería.
- 5 remates en los que el portero evitó la anotación.

Función que genera los gráficos

A continuación se presenta una función que engloba todo el proceso realizado anteriormente, la idea de la misma es replicar el procedimiento para cualquier competición y para cualquier jugador que tenga información sobre el evento de disparo.

```
#FUNCION PARA GENERAR GRAFICO DE UN JUGADOR
#ARGUMENTOS:
#data: df a emplear, debe tener infromación sobre los disparos
#name: nombre del jugador a consultar
#max: valor maximo a mostrar luego del cambio de ejes
#titulo: string con el titulo del plot
```

```r
goles_plot <- function(data,name,max = 125,titulo){
  #BASE DE DISPAROS
  shots = data %>%
    filter(type.name=="Shot" & player.name == name) #1

  #GRAFICO INICIAL
  a <- ggplot() +
    annotate("rect",xmin = 0, xmax = 120, ymin = 0, ymax = 80, fill = NA,
             colour = "black", size = 0.6) +
    annotate("rect",xmin = 0, xmax = 60, ymin = 0, ymax = 80, fill = NA,
             colour = "black", size = 0.6) +
    annotate("rect",xmin = 18, xmax = 0, ymin = 18, ymax = 62, fill = NA,
             colour = "black", size = 0.6) +
    annotate("rect",xmin = 102, xmax = 120, ymin = 18, ymax = 62, fill = NA,
             colour = "black", size = 0.6) +
    annotate("rect",xmin = 0, xmax = 6, ymin = 30, ymax = 50, fill = NA,
             colour = "black", size = 0.6) +
    annotate("rect",xmin = 120, xmax = 114, ymin = 30, ymax = 50, fill = NA,
             colour = "black", size = 0.6) +
    annotate("rect",xmin = 120, xmax = 120.5, ymin =36, ymax = 44, fill = NA,
             colour = "black", size = 0.6) +
    annotate("rect",xmin = 0, xmax = -0.5, ymin =36, ymax = 44, fill = NA,
             colour = "black", size = 0.6) +
    annotate("segment", x = 60, xend = 60, y = -0.5, yend = 80.5, colour = "black",
             size = 0.6)+
    annotate("segment", x = 0, xend = 0, y = 0, yend = 80, colour = "black", size = 0.6)+
    annotate("segment", x = 120, xend = 120, y = 0, yend = 80, colour = "black",
             size = 0.6)+
    theme(rect = element_blank(),
          line = element_blank()) +
    # add penalty spot right
    annotate("point", x = 108 , y = 40, colour = "black", size = 1.05) +
    annotate("path", colour = "black", size = 0.6,
             x=60+10*cos(seq(0,2*pi,length.out=2000)),
             y=40+10*sin(seq(0,2*pi,length.out=2000)))+
    # add centre spot
    annotate("point", x = 60 , y = 40, colour = "black", size = 1.05) +
    annotate("path", x=12+10*cos(seq(-0.3*pi,0.3*pi,length.out=30)), size = 0.6,
             y=40+10*sin(seq(-0.3*pi,0.3*pi,length.out=30)), col="black") +
    annotate("path", x=107.84-10*cos(seq(-0.3*pi,0.3*pi,length.out=30)), size = 0.6,
             y=40-10*sin(seq(-0.3*pi,0.3*pi,length.out=30)), col="black") +
    geom_point(data = shots, aes(x = location.x, y = location.y, size = shot.statsbomb_xg
                                 ,colour = factor(shot.outcome.name)),alpha = 0.5)+
    scale_size(range = c(0, 10))+
    labs(title = paste0(name," , Shot Map"), subtitle = titulo,
         size = "Statsbomb Xg", colour = "Shot outcome")+

    theme(
      plot.title = element_text(color="blue", size=18, face="bold.italic",hjust = 0.5),
      plot.subtitle = element_text(color="black", size=16, face="bold",hjust = 0.5),
      axis.title.x = element_blank(),
      axis.title.y = element_blank(),
      axis.text.y=element_blank(),
```

```
      axis.text.x=element_blank(),
      legend.direction = "horizontal",
      legend.position = "top",
      legend.margin = margin(c(20, 10, -85, 50)),
      legend.key.size = unit(1.5, "cm")
    ) +
    guides(size = guide_legend(title.position = "top"),
           colour = guide_legend(title.position = "top",
                    override.aes = list(size = 6, fill = "black"))) +
    coord_flip(xlim = c(range(shots$location.x)[1], max))

  #PROCESO PARA AGREGAR LOGO
  #GUARDO
  n <- paste0("images/",name,".png")
  ggsave(n, a, width = screen_width/130, height = screen_height/130,  dpi = 130)

  #PROCESO PARA AGREGAR LOGO STATSBOMB
  #LEO IMAGENES
  plot <- image_read(n)

  logo <- image_read("images/logo.png") %>%  image_resize(150)

  #OBTENGO DIMENSIONES DEL PLOT
  plot_height <- magick::image_info(plot)$height
  plot_width <- magick::image_info(plot)$width

  #OBTENGO DIMENSIONES DEL LOGO
  logo_width <- magick::image_info(logo)$width
  logo_height <- magick::image_info(logo)$height

  #CALCULO COORDENADA Y PARA UBICAR LOGO
  y <- plot_height - logo_height - plot_height * 0.01

  #CALCULO COORDENADA X PARA UBICAR LOGO
  x <- plot_width * 0.85

  coord <- paste0("+",x,"+",y)

  #GRAFICO MAS LOGO
  new <- plot %>%
    image_composite(logo, offset = coord)

  #GUARDO PLOT FINAL
  image_write(new, path = paste0("final_plots/",name,"_logo.png"), format = "png")

  #MENSAJE
  print("Gráfico generado y guarda en la ruta: ")
  print(paste0("final_plots/"))

  #IMPRIMO GRAFICO
  return(new)
}#FINAL FUNCTION
```

Para probar la función se genera una nueva base en donde obtenemos a los máximos goleadores de la Eurocopa

sin contar los goles de penal.

```
#BUSCO GOLEADORES EURO
StatsBombData <- StatsBombFreeEvents(MatchesDF = Matches, Parallel = T)

events1 <- allclean(StatsBombData)

goles = events1 %>%
  filter(type.name=="Shot" & (shot.outcome.name == "Goal") ) #1

top <- (as.data.frame(table(goles$player.name)))
top <- top[order(top$Freq,decreasing = T),]
top1 <- top[c(1:6),]
```

El top 6 de goleadores es el siguiente,

```
#TOP GOLEADORES DE LA EURO SIN CONTAR GOLES DE PENALTI
top1
```

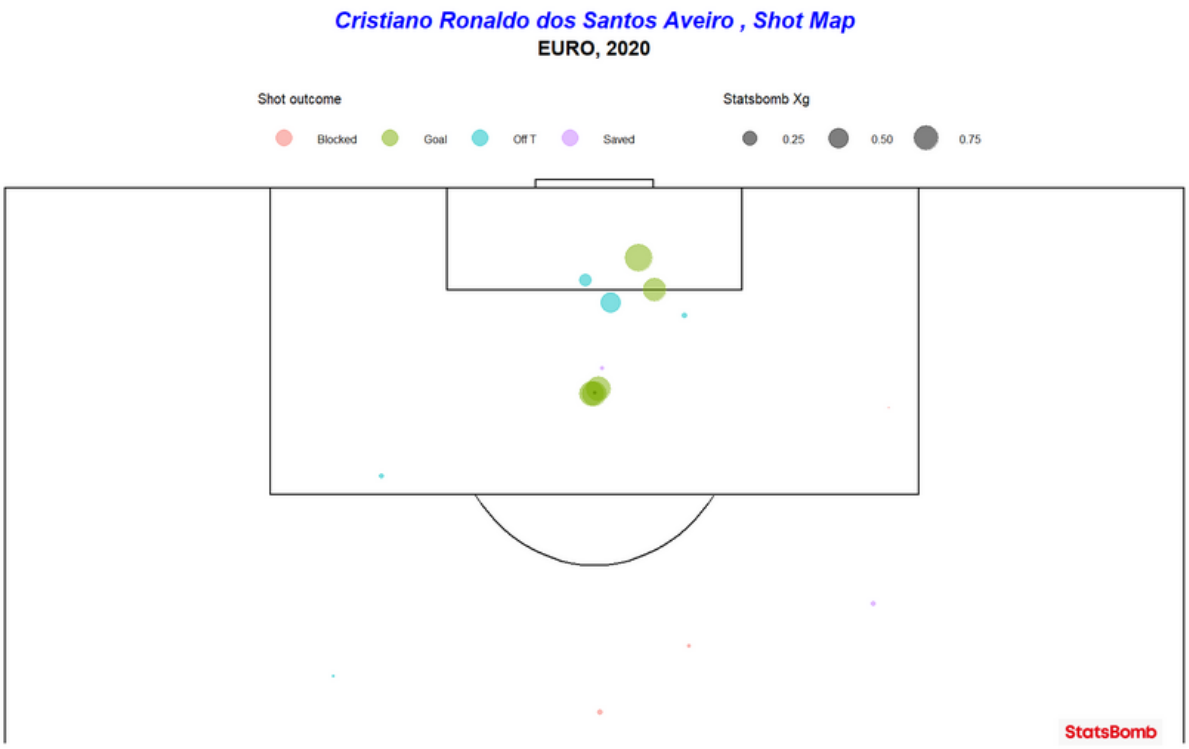
```
##                                  Var1 Freq
## 21 Cristiano Ronaldo dos Santos Aveiro    5
## 36                         Harry Kane    5
## 74                       Patrik Schick    5
## 26                Emil Peter Forsberg    4
## 45                      Karim Benzema    4
## 82               Romelu Lukaku Menama    4
```

Mapa de disparos de los máximos goleadores de la Eurocopa

```
a <- goles_plot(events1,name = "Cristiano Ronaldo dos Santos Aveiro",titulo = "EURO, 2020")
```

```
## [1] "Gráfico generado y guarda en la ruta: "
## [1] "final_plots/"
```

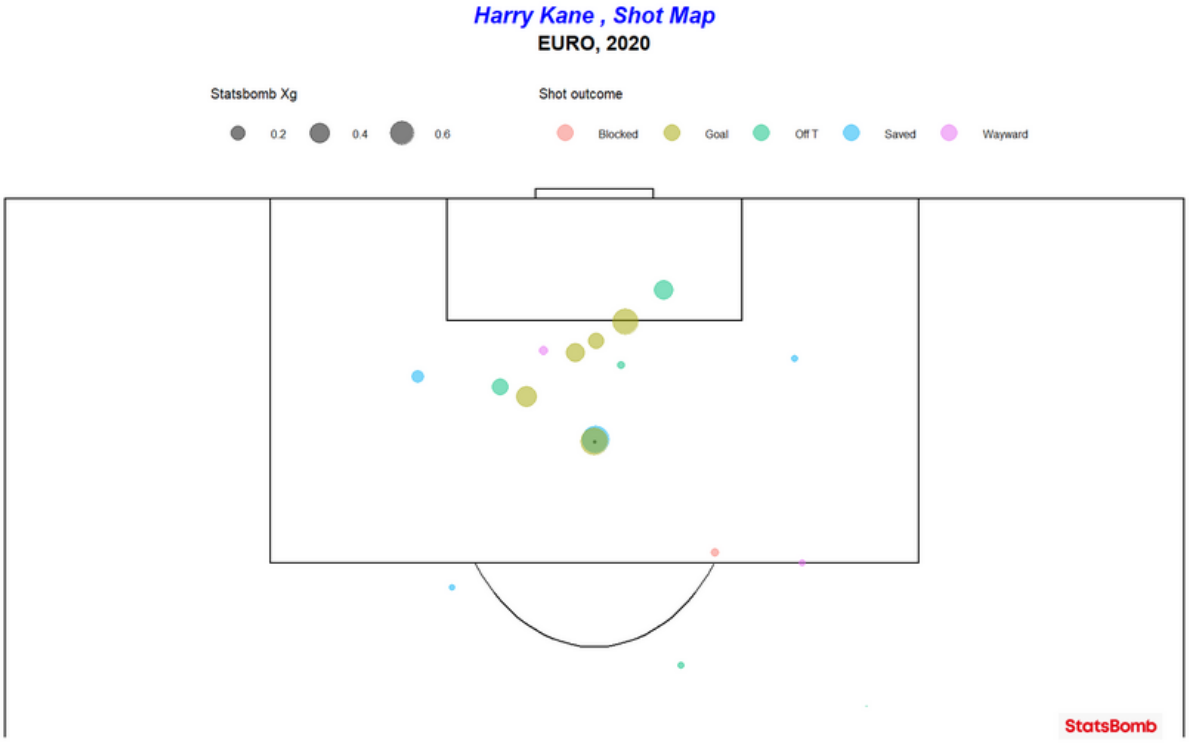
```
a %>% image_resize(900)
```




```
a <- goles_plot(events1,name = "Harry Kane",titulo = "EURO, 2020")
```

```
## [1] "Gráfico generado y guarda en la ruta: "
## [1] "final_plots/"
```

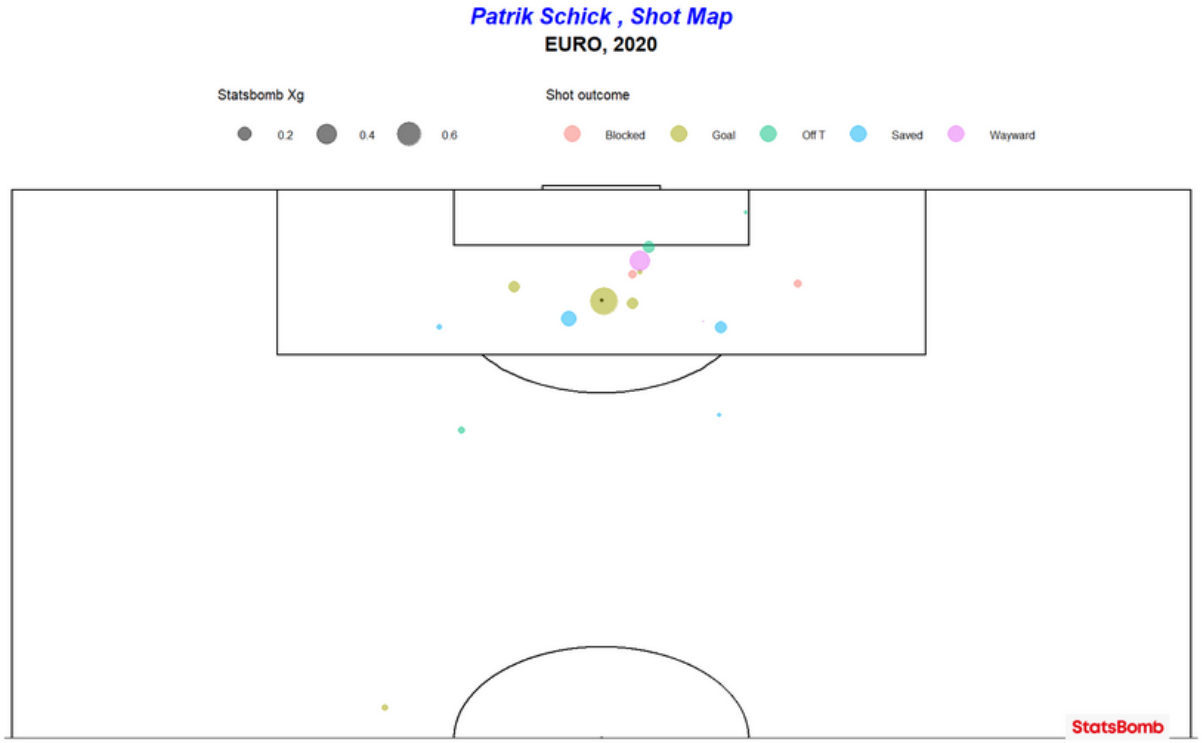
```
a %>% image_resize(900)
```



```
a <- goles_plot(events1,name = "Patrik Schick",max = 130,titulo = "EURO, 2020")
```

```
## [1] "Gráfico generado y guarda en la ruta: "
## [1] "final_plots/"
```

```
a %>% image_resize(900)
```




```
a <- goles_plot(events1,name = "Emil Peter Forsberg",titulo = "EURO, 2020")
```

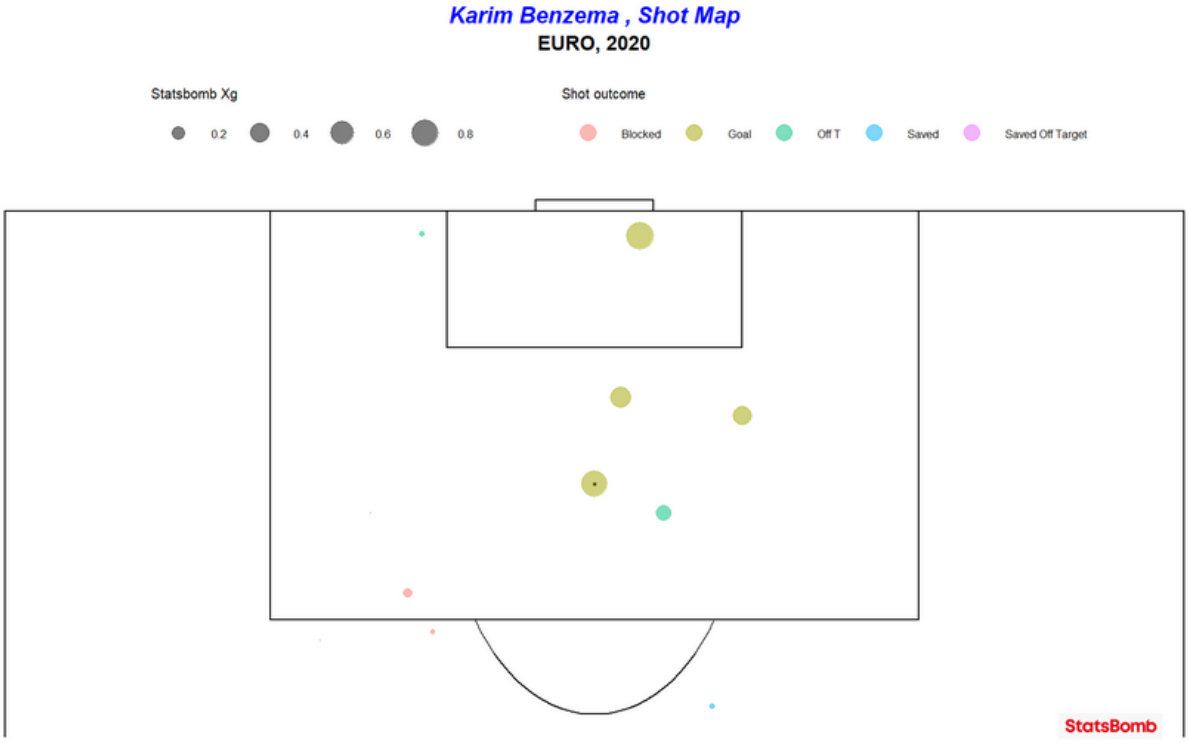
```
## [1] "Gráfico generado y guarda en la ruta: "
## [1] "final_plots/"
```

```
a %>% image_resize(900)
```


```
a <- goles_plot(events1,name = "Karim Benzema",titulo = "EURO, 2020")
```

```
## [1] "Gráfico generado y guarda en la ruta: "
## [1] "final_plots/"
```

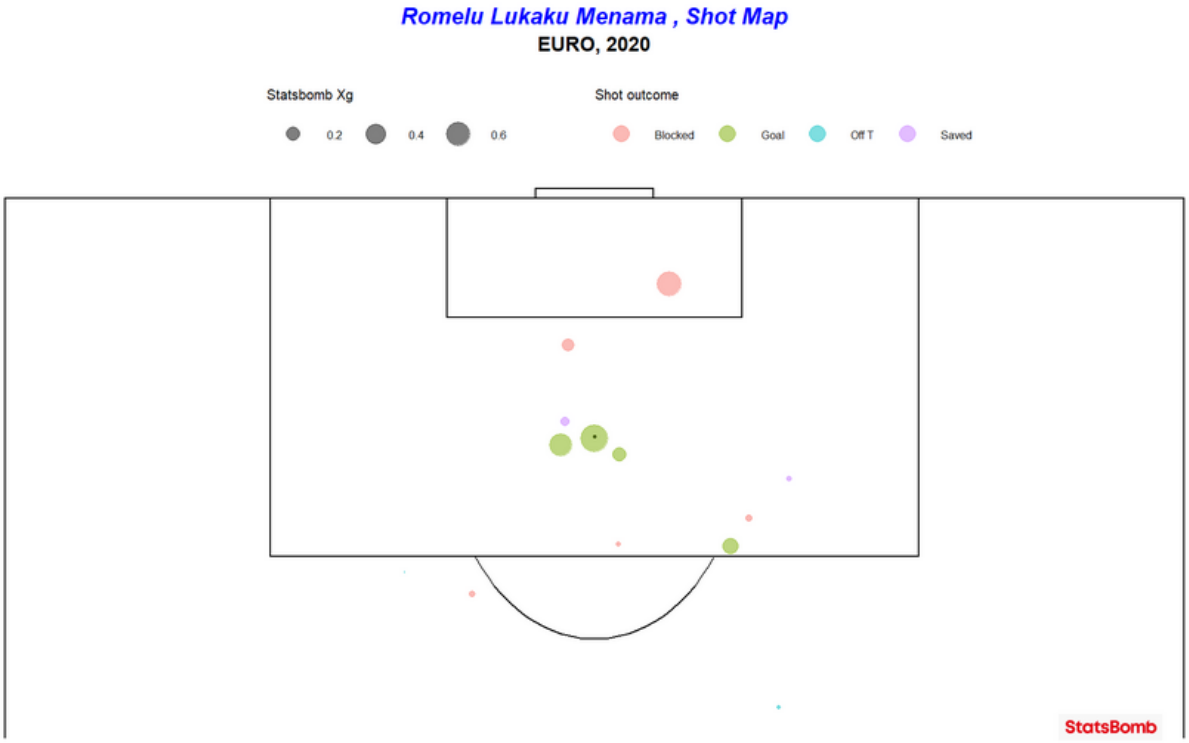
```
a %>% image_resize(900)
```



```
a <- goles_plot(events1,name = "Romelu Lukaku Menama",titulo = "EURO, 2020")
```

```
## [1] "Gráfico generado y guarda en la ruta: "
## [1] "final_plots/"
```

```
a %>% image_resize(900)
```



Las imagenes anteriormente mostradas se encuentran en la carpeta “final_plots”. De esta manera se ha realizado un mapa de disparos de los máximos goleadores de la Eurocopa 2020. Para mayor comodidad puede revisar mi repositorio en el siguiente enlace.