



**Universidad de Valladolid**  
**FACULTAD DE CIENCIAS**

**TRABAJO FIN DE GRADO**

Grado en Estadística

Nuevo paquete para el análisis de  
datos de juego y jugadores de fútbol:  
**GASB**

Autor:

Ramiro Gómez Nuño

Tutora:

Teresa González Arteaga



# Índice

1	Introducción.....	5
2	Herramienta R y RStudio. Crear paquete R. ....	7
2.1	Crear el proyecto.....	7
2.2	Introducir las funciones. ....	10
2.3	Construir la librería o paquete .....	11
3	Datos StatsBomb.....	13
3.1	Formato JSon.....	13
3.2	Eventos y variables. ....	14
3.3	Definiciones relevantes.....	16
3.4	Datos de partidos de fútbol utilizados.....	20
4	Métricas en el análisis de partidos de fútbol.....	23
4.1	Expected Goal.....	23
4.2	Ajuste de eventos por 90 minutos .....	29
4.3	Ajuste por Posesión para acciones defensivas.....	31
5	Paquete GASB.....	34
5.1	Archivos de datos .....	35
5.2	Campo de fútbol.....	37
5.3	Funciones auxiliares.....	40
5.4	Pases y combinaciones. ....	49
5.5	Análisis defensivo .....	58
5.6	Análisis ofensivo. ....	66
5.7	Análisis de rendimiento de jugadores. ....	77
6	Conclusiones.....	92
7	Referencias .....	93

## Resumen

El análisis de datos se presenta como una realidad en el mundo del fútbol y como una herramienta a la que acuden los profesionales del sector. Se sabe que lo más valioso y, por tanto, difícil de conseguir en Estadística, es la información. Gracias a la tecnología se pueden recoger los datos de todos los partidos de fútbol a nivel profesional cada semana obteniendo un gran volumen de información de calidad. Una vez se tienen los datos la única barrera es la imaginación, se pueden realizar análisis de todo tipo.

En este trabajo se presenta un paquete de R compatible con el formato de datos de *StatsBomb* en el que se incluyen funciones con las que realizar análisis del juego y jugadores. El paquete GASB, creado en la realización de este TFG, cuenta con multitud de funciones gráficas para realizar representaciones sobre el terreno de juego con las que resumir ciertas acciones de juego en fútbol.

El paquete GASB mezcla análisis ya utilizados en el mundo del fútbol y otros propios introducidos por el autor. Contiene funciones que permiten al usuario elegir libremente el propio el tipo de análisis que quiere realizar. Además se explica con detalle el formato de datos de *StatsBomb*, algunas métricas específicas del mundo del fútbol como el *expected goal* y posibles utilidades reales de las funciones del paquete.

### Palabras Clave

R, Paquete, Estadística, Fútbol, Expected Goal, GASB

## Abstract

Data analysis is presented as a reality in the world of soccer and as a tool used by professionals in the sector. It is known that the most valuable and, therefore, difficult to obtain in Statistics, is information. Thanks to technology, data from all soccer matches can be collected at a professional level every week, obtaining a large volume of quality information. Once you have the data, you can perform all kinds of analyzes, the only barrier is your imagination.

In this work, an R package compatible with the StatsBomb data format is presented, which includes functions to perform analysis of the game and players. The GASB package, created in the completion of this TFG, has a multitude of graphical functions to make representations on the field of play to summarize certain actions of the game in soccer.

The GASB package mixes analyzes already used in the world of soccer and others of its own introduced by the author. It contains functions that allow the user to freely choose the type of analysis he wants to carry out. In addition, the StatsBomb data format is explained in detail, as well as some specific metrics from the world of soccer such as the expected goal and possible real utilities of the package's functions.

### Palabras Clave

R, Package, Statistics, Football, Expected Goal, GASB

# 1 Introducción

El análisis de datos se ha convertido en una verdadera revelación en el sector deportivo. Gracias al tratamiento de grandes volúmenes de información que ofrecen los distintos eventos deportivos, los expertos son capaces de mejorar la toma de decisiones basándose en los datos.

El primer deporte en usarlo fue el béisbol en los años 70, cuando Bill James empezó a analizar los registros históricos de los jugadores de béisbol de las grandes ligas americanas, como recoge la web *Three Points* en su artículo *Big Data aplicado a los deportes*. Bill James inició el análisis del béisbol a través de evidencia objetiva, específicamente a través de estadísticas, con el fin de medir de manera eficaz las actividades que suceden dentro del campo de juego. De hecho, gracias a esta estrategia basada en datos, Billy Beane, general manager de los Oakland Athletics, consiguió reflotar económica y deportivamente a su equipo.

Desde entonces, son muchos los deportes que ya están empleando las posibilidades del análisis de datos para mejorar los análisis predictivos que ayudan en múltiples áreas, desde prevenir lesiones, cambiar las tácticas de juego, mejorar el rendimiento o encontrar errores, como sucede en los deportes de motor.

En el caso del baloncesto, por ejemplo, la NBA está transformando el sistema tradicional de juego. Mientras antes se apostaba por jugadores más altos que protegieran el aro y marcaran puntos fácilmente, ahora la tendencia se basa en el “small ball”, con jugadores más bajos y versátiles. Los datos han demostrado que es más productivo anotar triples que canastas de dos puntos aunque el propio aro no esté tan bien defendido.

Durante un partido de fútbol se pueden llegar a capturar alrededor de ocho millones de datos. Sin embargo, el ojo humano solo es capaz de retener el 30% de esa información. Gracias a esos datos, el Big Data aporta una gran valía en muchos clubes del mundo, que ya han empezado a usar esta tecnología para mejorar sus estrategias de juego, fichajes de nuevos jugadores, etc. La historia del análisis de datos en el mundo del deporte se explica muy bien en el libro *“Soccermaths: Mathematical adventures in the beautiful game”*, autor: *David Sumpter*, editorial: *Bloomsbury* (2016).

Lo cierto es que la ciencia de datos ya se utiliza en equipos en España, en concreto en La Liga, mediante un sistema que registra los partidos con 8 cámaras y ordenadores ubicados en diferentes lugares de los estadios para poder captar todos los movimientos y posteriormente los envía a una central donde se tratan los datos. Este tipo de cámaras es capaz de recoger 25 registros por segundo a partir de las cuales se obtienen datos técnicos y estadísticas.

Por análisis de datos entendemos los procedimientos usados a la hora de convertir los datos en información que facilite la toma de decisiones. En el mundo del fútbol todo esto puede aplicarse en diferentes situaciones: mercado de fichajes, análisis de juego, rendimiento de jugadores, etc. En este trabajo se tratarán estos dos últimos.

En la memoria de este trabajo se diferencian tres partes:

- Descripción de R y Rstudio y creación de un paquete R. *Apartado 2: Herramienta R y RStudio. Crear paquete R.*
- Descripción de los datos que se recogen en los partidos de fútbol y métricas necesarias para el uso de los mismos. *Apartado 3: Datos StatsBomb y Apartado 4: Métricas.*
- Descripción del paquete R original GASB, construido por el autor de este trabajo, y sus posibles usos en el análisis de juego y análisis de jugadores. *Apartado 5: Paquete GASB.*

El objetivo del paquete GASB, protagonista de este trabajo, es apoyar el análisis del juego para analistas y entrenadores. Hay muchos profesionales en el sector del fútbol que se ciñen más a su criterio e intuición. Un motivo para ello puede ser porque se ven invadidos por porcentajes, tablas de frecuencia, intervalos de confianza... conceptos que desconocen y en los que no confían para realizar su trabajo. Este paquete entre otras cosas realiza análisis sobre el campo de fútbol facilitando al usuario su comprensión mediante multitud de representaciones gráficas muy relacionadas con situaciones del juego real invitándole a querer saber más sobre el análisis de datos.

El paquete GASB, construido por el autor de este TFG, realiza análisis para cualquier base de datos con el formato *StatsBomb*. Además da libertad al usuario para poder elegir qué análisis realizar según su propio criterio.

El trabajo incluye el propio paquete GASB para instalarlo en una sesión R y poder trabajar con él. El paquete GASB contiene dos bases de datos con la información de las temporadas 2018/2019 y 2019/2020 de la liga *FA Women's Superleague*: "eventsLF" y "rendimiento", con las que se muestra al usuario como puede realizar sus propios análisis de juego o análisis de jugadores y obtener sus propias conclusiones.

## 2 Herramienta R y RStudio. Crear paquete R.

*Rstudio* es un entorno de desarrollo integrado para el lenguaje de programación R, dedicado a la computación estadística, gráficos y análisis de datos. Existe una red de servidores *ftp* y *web* en todo el mundo que almacena versiones idénticas y actualizadas de código y documentación para R denominado *CRAN*.

<https://ftp.cixug.es/CRAN/>

Oficina de software libre (CIXUG)

<https://cran.rediris.es/>

Spanish National Research Network, Madrid

En este trabajo se han empleado varios paquetes incluidos en el CRAN:

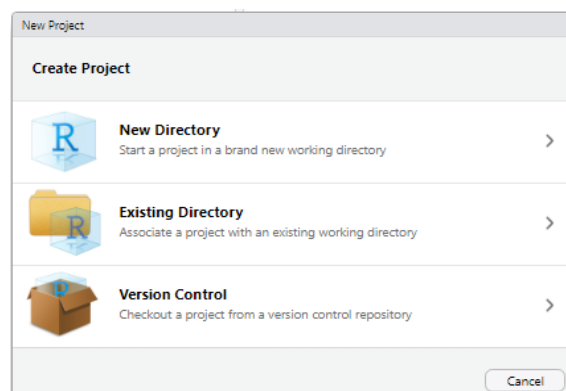
- *tidyverse*
- *ggplot2*
- *rgl*
- *fmsb*
- *plotly*
- *devtools*
- *roxygen2*

En este trabajo se presenta un paquete de R creado por el autor de este TFG para analizar datos de partidos de fútbol descargados de la web de la empresa StatsBomb.

Para la creación de un paquete en R debemos tener instalados los paquetes Roxygen2 (para generar documentación de ayuda en nuestro paquete) y devtools (herramientas de desarrollo de paquetes).

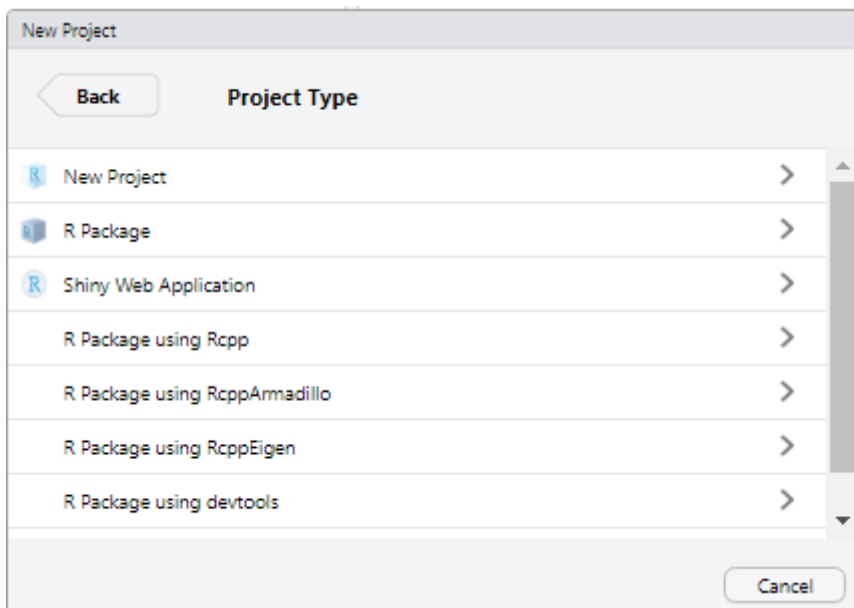
### 2.1 Crear el proyecto.

En Rstudio seleccionamos *File* -> *New Project*. Se abrirá la siguiente ventana. Ver *ilustración 2-1*



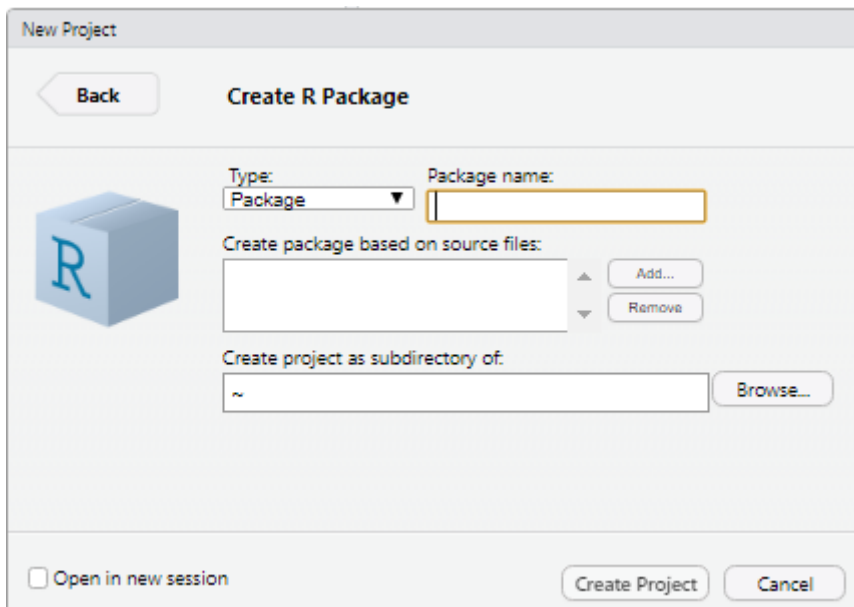
*Ilustración 2-1. Nuevo proyecto.*

Seleccionamos *New Directory*. Rstudio creará un directorio donde almacenaremos los archivos relacionados con nuestro proyecto. Se abrirá una ventana para que decidamos que tipo de proyecto queremos realizar. Ver *ilustración 2-2*.



*Ilustración 2-2. Tipo de proyecto.*

Seleccionamos *R Package* y llegamos a la siguiente ventana. Ver *ilustración 2-3*.

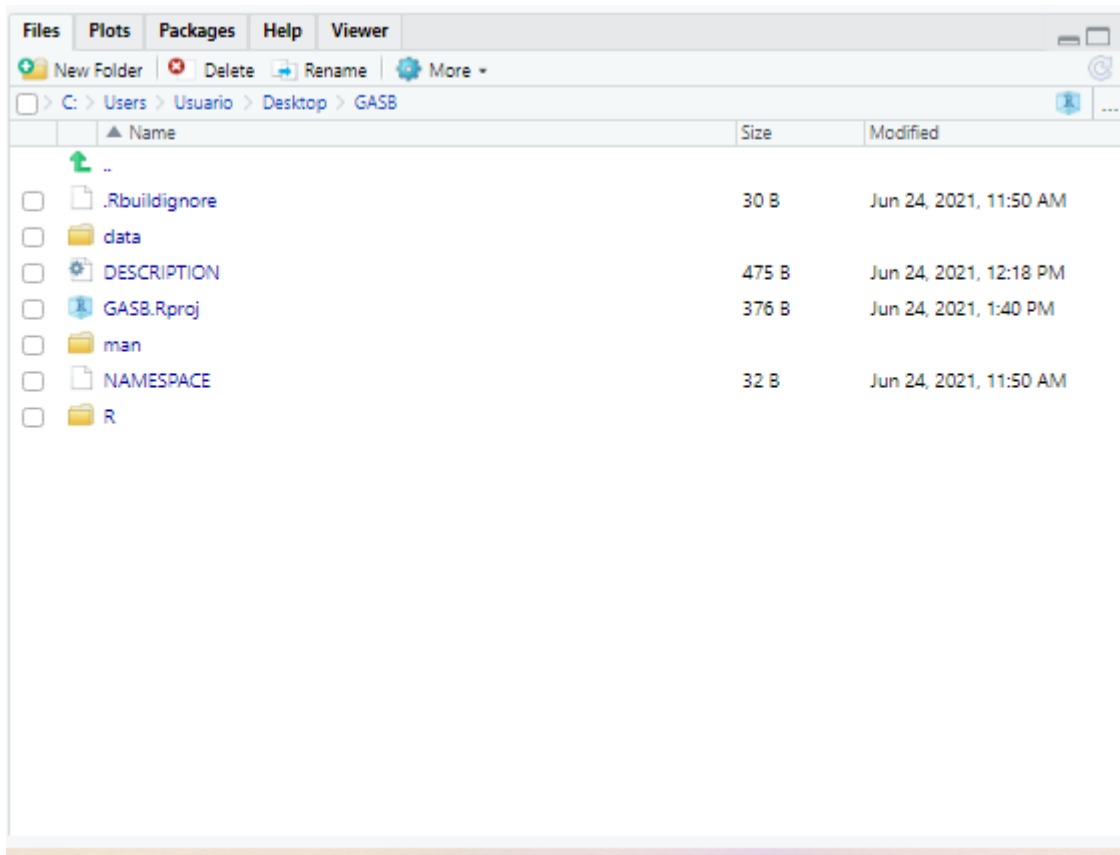


*Ilustración 2-3. Crear paquete.*

En la casilla *Package name*: especificamos el nombre de nuestro proyecto, por ejemplo *GASB* . Indicamos el directorio raíz del que colgará el subdirectorio con nuestro paquete R.

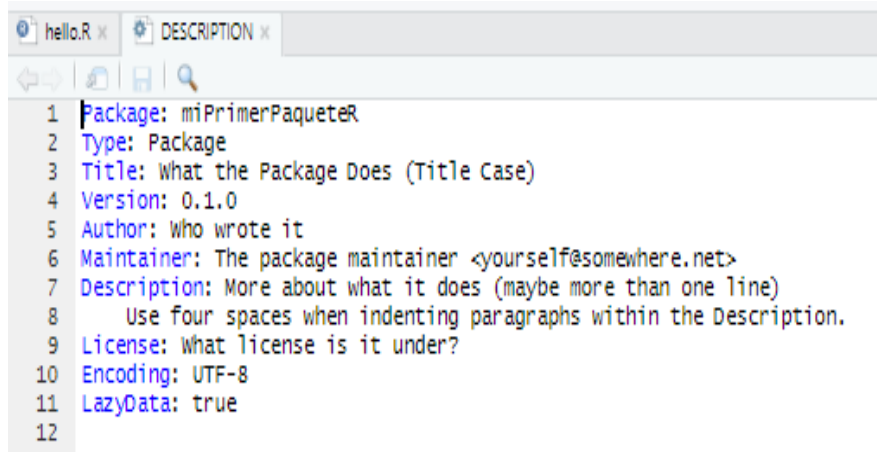


Al pinchar en *Create Project* se creará un directorio en la ubicación indicada con la estructura necesaria para que R pueda construir el paquete (ver *ilustración 2-4*):



*Ilustración 2-4. Estructura.*

En primer lugar picamos en el archivo DESCRIPTION (en el panel inferior derecho, *ilustración 2-4*), y completamos la información relativa al nombre del paquete, autor, descripción, etc: (*ilustración 2-5*)



*Ilustración 2-5. Descripción.*

## 2.2 Introducir las funciones.

Una vez creado el proyecto pinchamos en el icono de la carpeta denominada R en el panel inferior derecho, ilustración 2-4. Esta es la carpeta donde deben ir los archivos de script que contienen el código de nuestras funciones. Por defecto, salvo que hayamos indicado algún archivo en el paso anterior, en esta carpeta aparecerá un único archivo (vacío) con el nombre del paquete. Podemos abrir ese archivo y escribir directamente en él nuestras funciones, o podemos ir añadiendo archivos nuevos a esta carpeta. Suele ser buena práctica tener un único archivo por función ya que ello facilita las tareas de mantenimiento (actualización de la función, documentación, etc).

Para crear un archivo con una función vamos a File → New File → R Script y creamos un nuevo archivo. A modo de ejemplo, vamos a crear una función llamada `seleccionEventos()` que dibuje un histograma y le superponga una curva normal. El código de la función está en la ilustración 2-6.

```
1
2
3 seleccionEventos<-function(eventos, idJugador=0, idPartido=0, idEquipo=0, rival=F){
4   if(idEquipo!=0 & rival==T){
5     partidos <- unique(eventos$match_id[which(eventos$team.id==idEquipo)])
6     eventos<-eventos[partidos]
7     filter(match_id%in%partidos)
8   }
9   if(idEquipo!=0 & rival==F){
10    eventos<-eventos[team.id==idEquipo]
11  }
12 }
13 if(idPartido!=0){
14   eventos<-eventos[match_id==idPartido]
15 }
16 }
17 if(idJugador!=0){
18   eventos<-eventos[player.id==idJugador]
19 }
20 }
21 return(eventos)
22 }
```

Ilustración 2-6. Código de la función.

Para que R pueda generar automáticamente el archivo de ayuda para esta función incluimos las siguientes líneas que se muestran en la *ilustración 2-7* antes del comienzo de la función. En estas líneas describimos qué argumentos recibe la función, qué valor o valores devuelve, algunos ejemplos .... El paquete `roxygen2` se encarga de construir los archivos de ayuda en formato html a partir de esta información.

```
##' @title Seleccion de eventos
##' @description Funcion que selecciona eventos determinados por el usuario de una base de datos
##' @param eventos dataframe: Base de datos de la que queremos extraer los eventos
##' @param idJugador int: Identificador del jugador que queremos seleccionar
##' @param idPartido int: Identificador del partido que queremos seleccionar
##' @param idEquipo int: Identificador del equipo que queremos seleccionar
##' @param rival boolean: Selección de los eventos de equipo rival
##' @return dataframe con los eventos
##' @examples
##' datos<-seleccion(events, idJugador=3476)
##'
```

Ilustración 2-7. Documentación.

El archivo con nuestra función debe quedar, por tanto, como se presenta en la *ilustración 2-8*:

```

1 #' @title Seleccion de eventos
2 #' @description Funcion que selecciona eventos determinados por el usuario de una base de datos
3 #' @param eventos dataframe: Base de datos de la que queremos extraer los eventos
4 #' @param idJugador int: Identificador del jugador que queremos seleccionar
5 #' @param idPartido int: Identificador del partido que queremos seleccionar
6 #' @param idEquipo int: Identificador del equipo que queremos seleccionar
7 #' @param rival boolean: Selección de los eventos de equipo rival
8 #' @return dataframe con los eventos
9 #' @examples
10 #' datos<-seleccion(events, idJugador=3476)
11 #'
12
13
14 seleccionEventos<-function(eventos, idJugador=0, idPartido=0, idEquipo=0, rival=F){
15   if(idEquipo!=0 & rival==T){
16     partidos <- unique(eventos$match_id[which(eventos$team.id==idEquipo)])
17     eventos<-eventos%>%
18       filter(match_id%in%partidos)
19   }
20   if(idEquipo!=0 & rival==F){
21     eventos<-eventos%>%
22       filter(team.id==idEquipo)
23   }
24   if(idPartido!=0){
25     eventos<-eventos%>%
26       filter(match_id==idPartido)
27   }
28 }

```

Ilustración 2-8. Función.

Guardamos el script de la función (*File → Save as*) en un archivo R, preferentemente con el mismo nombre de la función. No es estrictamente necesario, pero nos facilitará posteriormente poder localizar la función si la queremos modificar. Por defecto, la función se guarda en la carpeta *R* del directorio de nuestro paquete.

### 2.3 Construir la librería o paquete

Para construir nuestro paquete vamos a la pestaña *Build* (panel superior derecho). Antes que nada, comprobamos nuestra configuración, y la ajustamos en caso necesario. Para ello picamos en el icono *More* de dicho panel (*ilustración 2-9*). A continuación en *Configure Build Tools*. Se abre una ventana como la *ilustración 2-10*. Pinchamos en la casilla *Generate documentation with Roxygen*. Además picamos el botón *Configure...*, y se nos abre el menú de la *ilustración 2-11*.

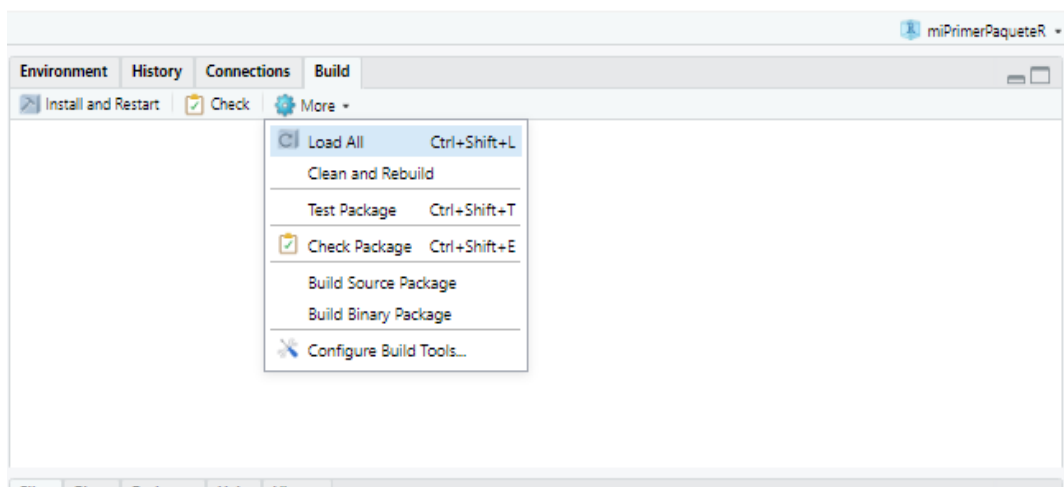


Ilustración 2-9. Configurar documentación.

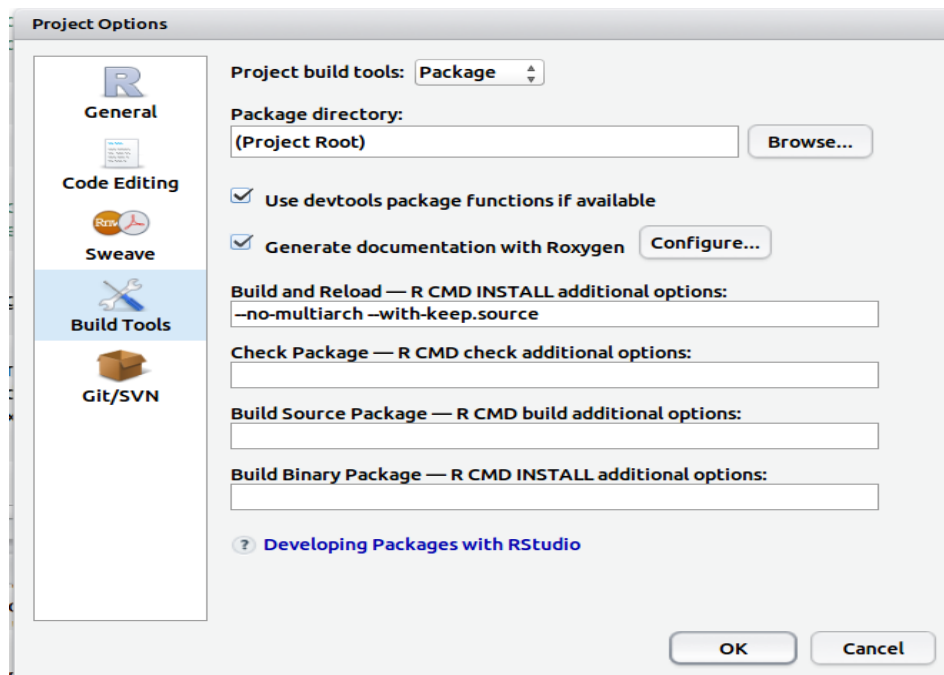


Ilustración 2-10. Configurar roxygen.

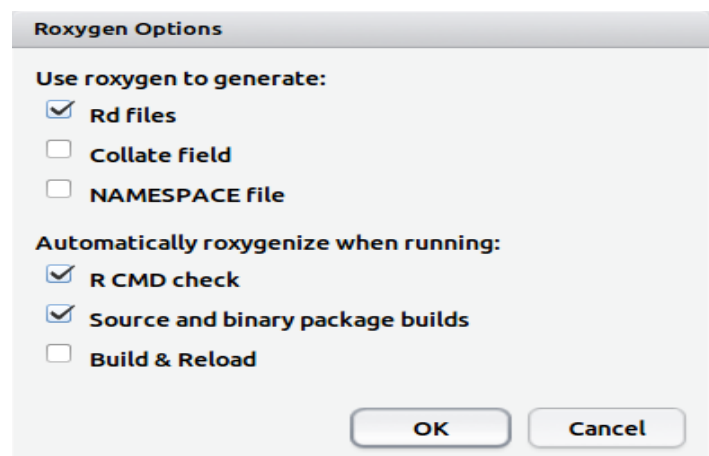


Ilustración 2-11. Configurar Rox.

Seleccionamos las casillas *NAMESPACE file* y *Build & Reload*, además de las opciones marcadas por defecto (*Rd files*, *R CMD check* y *Source and binary packages build*).

Una vez ajustada la configuración, pinchamos en *Build & Reload*.

Con esto tenemos el paquete creado con una única función y el documento de ayuda correspondiente. Después hemos de añadir todas las funciones y su documentación que compongan el paquete final. Además se pueden añadir ficheros de datos junto a su documentación.

## 3 Datos StatsBomb

En este trabajo se utilizan los datos con el formato de *StatsBomb*. StatsBomb es una empresa dedicada a la recogida y análisis de datos de fútbol. Su formato no es el único posible. Pero hay que reconocer que es muy amigable con el usuario. He escogido estos datos porque StatsBomb tiene una cantidad considerable de datos públicos y gratuitos.

Los datos utilizados para llevar a cabo el análisis de partidos de fútbol son los denominados eventos. Cada evento es una observación. La manera usual de utilizar esta información es agrupar los eventos en partidos. En cada partido se recogen una media de entre 3.000 y 4.000 eventos.

### 3.1 Formato JSON

Los datos de los partidos de fútbol con sus respectivos eventos habitualmente tienen formato JSON. Un archivo JSON es un archivo que almacena objetos y estructuras de datos simples en formato "JavaScript Object Notation". Este es un formato estándar de intercambio de datos. Los archivos JSON son ligeros, están basados en texto, son legibles y pueden editarse usando un editor de texto básico. La *ilustración 3-1* muestra un ejemplo.

```
{
  "menu": {
    "id": "file",
    "value": "File",
    "popup": {
      "menuitems": [
        {
          "value": "New", "onclick": "CreateNewDoc()"
        },{
          "value": "Open", "onclick": "OpenDoc()"
        },{
          "value": "Close", "onclick": "CloseDoc()"
        }
      ]
    }
  }
}
```

*Ilustración 3-1. Formato JSON. Fuente Wikipedia.*

Para trabajar con los datos en R leeremos los ficheros en un objeto de clase dataframe. Este dataframe contiene los eventos recogidos para ambos equipos que participan en ese partido. Algunos elementos tienen elementos secundarios, normalmente un par de nombre/identificación, o matrices secundarias, las cuales se detallarán más adelante.

### 3.2 Eventos y variables.

StatsBomb distingue 22 tipos de eventos distintos. Todos los eventos tienen 20 variables en común como el identificador (cada evento tiene un identificador único) o el minuto en el que se ha realizado el evento. Y cada tipo de evento tiene sus propias variables. Por ejemplo, el evento de tipo *disparo* tiene una variable *outcome* que puede tomar los valores: Gol, fuera, bloqueado, parado, al palo.

#### 3.2.1 Variables comunes:

La *Tabla 1* se recogen algunas variables comunes con una breve descripción de cada una. La tabla completa se encuentra en el *Apéndice*.

Columna	Tipo	Secundario	Tipo Secundario	Descripción	Valores	Descripción de los valores
Id	Uuid			Valor único para el evento.	Ej. "57726e09-9ba1-4388-8c40-faf57e5b8d1d"	
index	integer			Secuencia de los eventos por orden en el partido.	Ej, 1 #de eventos	
period	integer			Parte del partido.	1	1er tiempo
					2	2do tiempo
					3	1er tiempo prórroga
					4	2do tiempo prórroga
					5	Tanda Penaltis
timestamp	timestamp			Tiempo del partido en el que el evento tuvo lugar.	Ej. 00:00:02.78	
minute	Integer			Minuto del partido en el que ocurrió el evento.	Ej. 43	
second	Integer			Segunda parte del timestamp	Ej. 59	

*Tabla 1. Variables en común. Continúa Apéndice.*

### 3.2.2 Tipos de eventos:

La *Tabla 2* recoge algunos tipos de eventos con sus variables propias y sus respectivas descripciones. Todos los tipos de eventos se incluyen en una tabla en el apéndice.

Tipo de evento (Data Frame anidado)	Columna	Tipo	Tipo secundario	Secundario/s	Descripción	Valores	Descripción de valores
50_50	outcome	object	Id/name	Integer/text	Id/nombre del resultado del duelo 50/50	108/"Won"	El jugador gana el duelo.
						109/"Lost"	El jugador pierde el duelo.
						147/"Success To Team"	El balón lo recogió su equipo
						148/"Success To Opposition"	El balón lo recogió el rival.
	counterpress	Boolean			Evento realizado 5 segundos después de perder el balón	TRUE	
Bad Behaviour	card	object	Id/name	Integer/text	Id/nombre de la tarjeta mostrada.	65/"Yellow Card"	
						66/"Second Yellow"	
						67/"Red Card"	
Ball Receipt	outcome	object	Id/name	Integer/text	Id/nombre resultado de recibir el balón	9/"Incomplete"	
Ball Recovery	offensive	boolean			Recuperación de un balón por acierto propio	TRUE	El rival intentaba regatear.

*Tabla 2. Sumario tipos de eventos. Continúa apéndice.*

### 3.3 Definiciones relevantes

Tanto en la *Tabla 1* como *Tabla 2* se describen los distintos tipos de eventos con sus respectivas variables. Pero la información proporcionada por las tablas no es suficiente para entender completamente los datos. En esta sección completaremos esa información con algunas definiciones y explicaciones.

#### 3.3.1 Posiciones y *roll* en el campo:

La *tabla 3* recoge las posiciones en el campo con la numeración que simboliza las mismas y que se representa en la *ilustración 3-2*.

Numero de posición	Abreviacion Posición	Nombre de Posicion
1	GK	Portero
2	RB	Lateral Derecho
3	RCB	Central Derecho
4	CB	Defensa Central
5	LCB	Central Izquierdo
6	LB	Lateral Izquierdo
7	RWB	Carrilero Derecho
8	LWB	Carrilero Izquierdo
9	RDM	Medio defensivo Derecho
10	CDM	Medio Defensivo
11	LDM	Medio Defensivo Izquierdo
12	RM	Medio Derecho
13	RCM	Medio Centro Derecho
14	CM	Medio Centro
15	LCM	Medio Centro Izquierdo
16	LM	Medio Izquierdo
17	RW	Extremo Derecho
18	RAM	Interior Derecho
19	CAM	Medio Centro Ofensivo
20	LAM	Interior Izquierdo
21	LW	Extremo Izquierdo
22	RCF	Delantero Derecho
23	ST	Delantero
24	LCF	Delantero Izquierdo
25	SS	Media Punta



*Ilustración 3-2. Posiciones. Fuente Statsbomb.*

*Tabla 3. Posiciones.*



### 3.3.2 Coordenadas del campo

No todos los campos de fútbol tienen las mismas dimensiones exactamente. Estas tienen cierta flexibilidad: el ancho debe tener un mínimo de 45 metros y un máximo de 90 y el largo un mínimo de 90 metros y un máximo de 120 metros. Aunque las medidas pueden variar el campo debe ser siempre rectangular. No se especifica la proporción entre ancho y largo, pero por ejemplo un campo de 90 x 90 no sería reglamentario.

Todo esto lo señalamos para explicar que los datos que se recogen en los distintos campos y se estandarizan en un campo de 73 metros por 110 metros. Toda la información referida a distancias se recoge en yardas. Las dimensiones de los campos serán 80 x 120.

Tanto el tamaño de las áreas como de las porterías y del círculo central sí que son las mismas en todos los campos siendo estas las siguientes:

Círculo central: Radio de 9.15 metros.

Área grande: 40.3 metros de ancho por 16.5 metros de largo. (44 x 18 yardas)

Área pequeña: 18.32 metros de ancho por 5.5 metros de largo. (20 x 6 yardas)

Punto de penalti: 11 metros a la línea de meta. (12 yardas)

Portería: 7.32 metros de ancho por 2.44 metros de alto. (8 x 2.67 yardas)

Todos los eventos de un equipo se recogen “de izquierda a derecha”. Esto nos facilita tareas a la hora de trabajar con ello. No hay que cambiar las coordenadas a mitad del partido. La zona del campo respecto a su portería donde ocurre un evento en todo momento.

Los datos presuponen que el eje Y está invertido. Esto es conveniente para representar acciones de ataque o de defensa en un campo vertical.

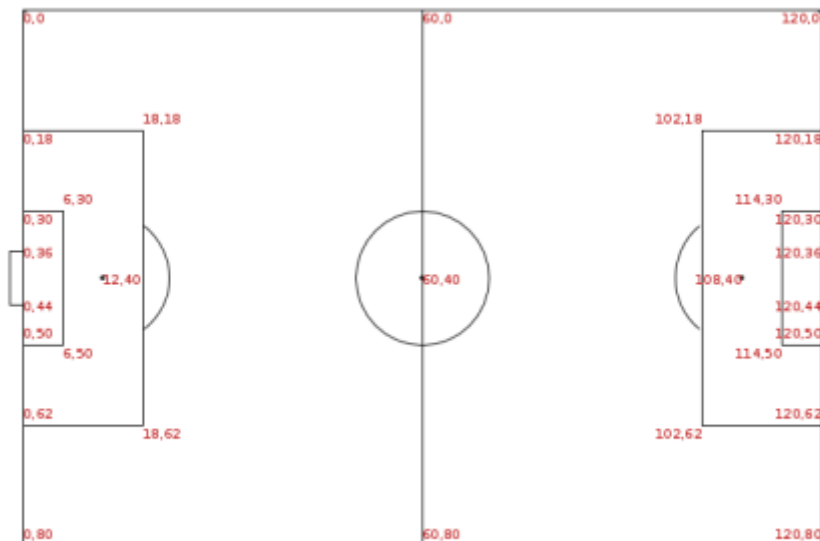
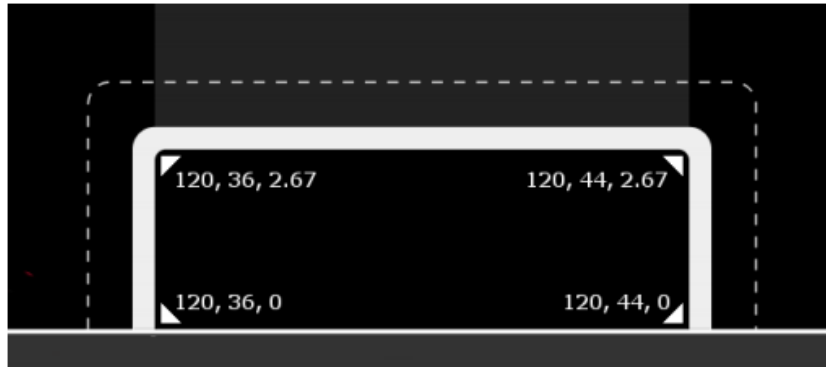


Ilustración 3-3. Coordenadas campo de fútbol. Fuente StatsBomb.

### 3.3.3 Coordenadas de la portería

La *ilustración 3-4* recoge las coordenadas de la portería rival.



*Ilustración 3-4. Coordenadas portería. Fuente StatsBomb..*

### 3.3.4 Freeze frame.

El evento de tipo *Shot* incluye una entrada la cual es un dataframe que incluye la situación de los jugadores cercanos al disparo. Dicho dataframe tiene el formato de la *tabla 4*.

Columna	Tipo	Secundario	Tipo Secundario
Location	Array(x, y)		
Player	Object	Id/name	integer/text
Position	Object	Id/name	Integer/text
teammate	boolean		

*Tabla 4. Freeze frame disparo.*

### 3.3.5 Lineup / alineación

Los eventos tipo *Starting XI* incluyen un data frame llamado *lineup* indicando los jugadores de inicio y su posición con la estructura de la *tabla 5*:

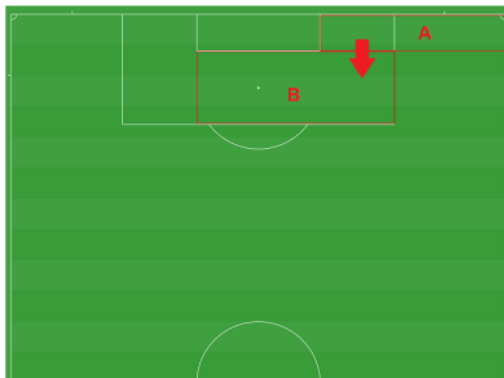
Columna	Tipo	Secundario	Tipo Secundario
Jersey_number	integer		
Player	Object	Id/name	integer/text
Position	Object	Id/name	Integer/text

*Tabla 5. Once inicial.*

### 3.3.6 Cutback. Pase atrás.

También encontramos una variable secundaria en los eventos de tipo Pass (pases) denominada *cutback*, en castellano su traducción sería “pase de la muerte”. Este es un pase que se realiza desde la línea de fondo para encontrar un compañero mejor posicionado para el disparo.

El pase se realiza desde la zona A (de cualquier lado de la portería) a la zona B. Mostrado en la *ilustración 3-5*.

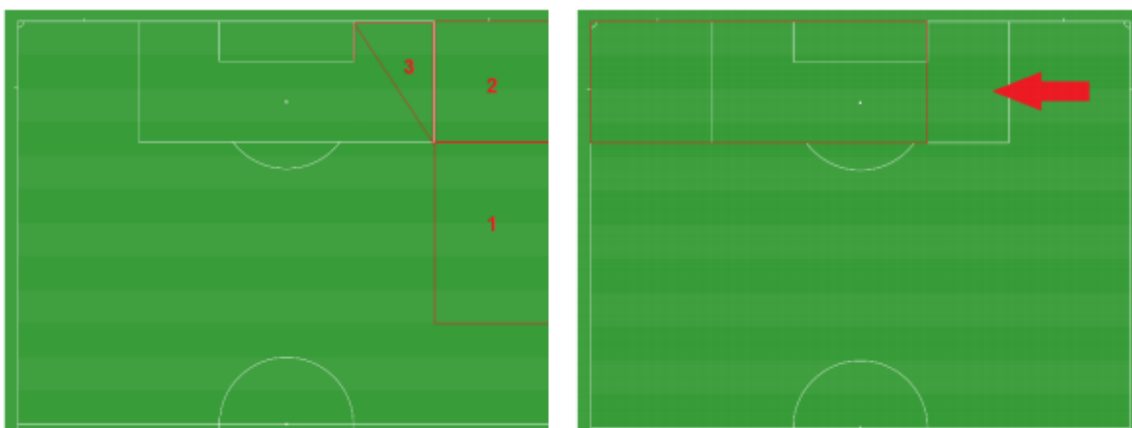


*Ilustración 3-5. Pase cutback. Fuente StatsBomb.*

### 3.3.7 Throws. Centros.

También se catalogan los pases como centros o en inglés *Throws*. Las características de este tipo de pases son la situación desde donde se realiza y hacia dónde se dirige el pase.

El pase se inicia desde las zonas 1,2 o 3 hacia la siguiente zona. Ver *ilustración 3-6*.



*Ilustración 3-6. Centros. Fuente StatsBomb.*

### 3.4 Datos de partidos de fútbol utilizados

La empresa *StatsBomb* facilita de forma gratuita todos los eventos de muchos partidos de una serie de competiciones. Las competiciones no están al completo exceptuando *FIFA World Cup* de 2018 y *Women's World Cup* de 2019, los mundiales masculino y femenino respectivamente. Aunque las competiciones no estén completas, los partidos sí que lo están. En un partido están todos los eventos sucedidos en él.

Estos datos se pueden obtener mediante el paquete libre de R *StatsBombR*. En la *Tabla 6* se ven las distintas competiciones y el volumen de partidos que ofrece este paquete.

Competición	Temporada	Partidos Registrados	Partidos Totales
Champions League	2003/2004	1	125
	2004/2005	1	125
	2006/2007	1	125
	2008/2009	1	125
	2009/2010	1	125
	2010/2011	1	125
	2011/2012	1	125
	2012/2013	1	125
	2013/2014	1	125
	2014/2015	1	125
	2015/2016	1	125
	2016/2017	1	125
	2017/2018	1	125
	2018/2019	1	125
FA Women's Super League	2018/2019	107	110
	2019/2020	87	110
FIFA World Cup	2018	64	64
La Liga	2004/2005	7	380
	2005/2006	17	380
	2006/2007	26	380
	2007/2008	28	380
	2008/2009	31	380
	2009/2010	35	380
	2010/2011	33	380
	2011/2012	33	380
	2012/2013	32	380
	2013/2014	31	380
	2014/2015	38	380
	2015/2016	33	380
	2016/2017	34	380
	2017/2018	34	380
2018/2019	34	380	
2019/2020	33	380	
NWLS	2018	36	111
Premier League	2003/2004	33	380
Women's World Cup	2019	52	52

Tabla 6. Datos públicos StatsBomb.

En el caso de los partidos de *LaLiga* incluidos en el paquete StatsBombR todos los partidos registrados son del *Fútbol Club Barcelona*. Aunque sea mínimo el volumen de partidos de esta competición, lo que respecta a estudiar a dicho club se dispone de una buena cantidad de datos.

Si bien es cierto que las dos copas del mundo están completas he decidido no usar los datos por las circunstancias tan especiales que se dan en este tipo de torneos.

- Poco volumen de partidos.
- Todos los equipos no se enfrentan entre sí.
- Hay una gran diferencia en el número de partidos jugados del equipo que llega a la final (7 partidos) frente al que es eliminado en fase de grupos (3 partidos).

En este TFG utilizaremos los datos de la liga *FA Women's Super League*. Aunque no esté completa 194 partidos de las temporadas 2018/2019 y 2019/2020 de 220 partidos totales es un buen tamaño para este proyecto. Sí es cierto que tenemos menos volumen de equipos y jugadores que si escogiésemos todos los partidos que ofrece el paquete StatsBombR, este tipo de competición presenta datos más estables y reales por las siguientes razones.

- Más volumen de partidos que cualquier otra competición del paquete.
- Todos los equipos juegan entre sí.
- Los equipos juegan el mismo número de partidos.
- No hay prórrogas, los partidos duran más o menos lo mismo, dependiendo del tiempo añadido.

Para conseguir los datos de los partidos que nos interesan primero instalamos el paquete "devtools" y "StatsBombR".

```
>install.packages("devtools")
>devtools::install_github("statsbomb/StatsBombR")
> library(StatsBombR)
```

Obtenemos información de las distintas competiciones incluidas en el paquete StatsBombR (ver ilustración 3-7) con la siguiente función del paquete StatsbombR:

```
comp<-FreeCompetitions()
```

	competition_id	season_id	country_name	competition_name	competition_gender	season_name	match_updated	match_available
1	16	4	Europe	Champions League	male	2018/2019	2020-07-29T05:00	2020-07-29T05:00
2	16	1	Europe	Champions League	male	2017/2018	2020-07-29T05:00	2020-07-29T05:00
3	16	2	Europe	Champions League	male	2016/2017	2020-08-26T12:33:15.869622	2020-07-29T05:00
4	16	27	Europe	Champions League	male	2015/2016	2020-08-26T12:33:15.869622	2020-07-29T05:00
5	16	26	Europe	Champions League	male	2014/2015	2020-08-26T12:33:15.869622	2020-07-29T05:00
6	16	25	Europe	Champions League	male	2013/2014	2020-08-26T12:33:15.869622	2020-07-29T05:00

Ilustración 3-7. Aspecto dataframe competiciones.

Con este data frame tenemos toda la información necesaria para conseguir los partidos de todas las competiciones disponibles en el paquete: 37 observaciones (competiciones) con 8 variables.

Como en este trabajo se utilizan los datos de FA Women's Super League 2018/2019 y 2019/2020. Se aplica la función *FreeMatches()* con el filtro de la competición y temporadas requeridas. El

resultado es un dataframe con 194 observaciones (partidos) y 40 variables (ver ilustración 3-8). La variable más importante *match\_id*, identificador de los partidos que utilizaremos para acceder a los eventos correspondientes (ver ilustración 3-9).

```
Comp <- FreeCompetitions() %>%
  filter(competition_id==37 & season_id%in%c(42,4))

Matches <- FreeMatches(Comp)
```

	match_id	match_date	kick_off	home_score	away_score	match_status	last_updated	match_week
1	2275054	2020-01-05	15:00:00.000	1	0	available	2020-07-29T05:00	11
2	2275072	2020-01-05	13:30:00.000	3	1	available	2020-07-29T05:00	11
3	2275085	2020-01-05	15:00:00.000	1	4	available	2020-07-29T05:00	11
4	2275113	2020-01-19	16:00:00.000	2	1	available	2020-07-29T05:00	13
5	2275142	2020-01-05	13:00:00.000	0	1	available	2020-07-29T05:00	11
6	2275099	2019-09-08	15:00:00.000	0	1	available	2020-07-29T05:00	1

Ilustración 3-8. Aspecto dataframe partidos.

```
StatsBombData <- StatsBombFreeEvents(MatchesDF = Matches, Parallel = T)
StatsBombData = allclean(StatsBombData)
eventosLF = StatsBombData
```

	id	index	period	timestamp	minute	second	possession	duration
1	caa84999-ca96-408d-93b5-c8f74d832237	1	1	00:00:00.000	0	0	1	0.000000
2	9fffa92c-6397-4dd4-9ff4-5c560bd84e53	2	1	00:00:00.000	0	0	1	0.000000
3	906917ac-7840-41cf-9e76-17df27d3edb3	3	1	00:00:00.000	0	0	1	0.000000
4	430a8774-f8e7-4691-a0fd-4eeb0491d5d0	4	1	00:00:00.000	0	0	1	0.000000
5	883510b1-40a0-49be-a590-ab2f1dcc6bdf	5	1	00:00:00.903	0	0	2	0.612044
6	fc4294cf-4059-4518-8806-7046c9aebffd	6	1	00:00:01.515	0	1	2	NA

Ilustración 3-9. Aspecto dataframe eventos.

Recordamos que los archivos de los partidos tienen formato JSON. Este formato de datos a menudo resulta incómodo por los dataframes anidados que se encuentran en él.

La función *allclean()* realiza la limpieza necesaria de los dataframes anidados *location end\_location...* y agrega las variables auxiliares para obtener la información de los eventos que la requieren como *Shot*, *Pass* o *Goalkeeper*.

El resultado de aplicar *allclean()* a los partidos de la FA Women's Superleague 2018/2019 y 2019/2020 es un dataframe con 648877 observaciones, de aquí en adelante denominados *eventos*, y 187 variables. Por lo tanto, tenemos 648877 eventos de 194 partidos, lo que significa que hay una media de 3344.727 eventos por partido.

## 4 Métricas en el análisis de partidos de fútbol

El análisis de datos de partidos de fútbol, como en cualquier otro ámbito de estudio, requiere de métricas y ajustes de datos propios. En esta sección se describen los más utilizados: expected goal, ajuste por 90 minutos y ajuste de acciones defensivas por posesión.

### 4.1 Expected Goal

Los goles son los sucesos más importantes en el fútbol ya que determinan el ganador de un partido pero no proporcionan información suficiente para analizar a los equipos. El promedio de goles está entre 2-3 goles por partido en el fútbol actual.

Por poner un ejemplo en un partido de la NBA lo más normal es que ambos equipos anoten más de 90 puntos, cantidad razonable para analizar el rendimiento a través de los puntos. Sin embargo no podemos hacer lo mismo en el fútbol.

El expected goal es una métrica de rendimiento. La misión del Expected Goal no es predecir nada, si no dar un valor a las ocasiones de gol generadas.

Para explicar el concepto de expected goal vamos a tomar como ejemplo un tiro a puerta con distintas características: lugar del campo, pierna dominante o débil, posición del portero rival, tras controlar el balón o al primer toque, bajo presión etc. ¿Qué probabilidad tiene ese tiro de acabar en gol?

#### 4.1.1 Total shots Ratio

Los goles vienen precedidos de tiros (ocasiones), por tanto podemos dar un paso atrás y medir el rendimiento de los equipos por los tiros realizados. Esto ya nos da un volumen de información considerable por partido, entre 20 y 30 tiros por partido, diez veces más que goles.

El 10-11% de los tiros acaban en gol, el equipo que tire más que el rival ganará el partido. Partiendo de esa base gente como Gabriel Desjardins, James Grayson, y Benjaminy Danny Pugsley implementaron estas ideas desde el análisis del hockey y crearon la métrica **Total shots Ratio**.

La métrica **Total Shots Ratio** es el ratio entre los tiros de un equipo y el total de los tiros en un partido o una serie de partidos.

$$\text{Total shots Ratio (Equipo A)} = \text{Tiros a favor} / (\text{Tiros a favor} + \text{Tiros recibidos})$$

Un ejemplo concreto:

En el partido Atlético de Madrid – Granada de La Liga 20/21, el equipo local remató 16 veces y el visitante 11. El TSR del Atlético de Madrid sería  $16 / (16+11) = 0.60$ . El TSR del Granada al tratarse de un ratio esta acotado entre 0 y 1. En consecuencia el TSR del Granada se calcularía:  $1 - \text{TSR}(\text{Atlético de Madrid}) = 0.40$ .

Con el transcurso de los partidos se comprobó que el Total Shots Ratio era un mejor predictor del ratio de goles de un equipo que el propio ratio de goles de ese mismo equipo. Sin embargo surgieron varios problemas intrínsecos en el planteamiento: no todos los tiros son iguales, unos

tiros tienen más valor que otros. Para superar estos inconvenientes surgieron los modelos de Goles Esperados (xG como acrónimo de Expected Goals).

#### 4.1.2 Modelo Expected Goal

Los modelos de Expected Goals suponen un paso adelante. Como espectadores sabemos que no es lo mismo un tiro desde 30 metros de la portería a otro en la línea de gol. Por tanto no pueden tener el mismo valor para analizar el rendimiento de un equipo.

No existe un modelo único de *Expected Goal*. Cada modelo tiene sus particularidades y unos utilizan unas y otras variables, aunque tradicionalmente las variables más relevantes y que están incluidas en prácticamente todos los modelos son:

- Distancia a portería.
- Ángulo respecto a portería.
- Parte del cuerpo con la que se realiza el remate o tiro. (Distinguiendo pierna hábil de pierna débil.)
- Tipo de asistencia o acción previa (pase en profundidad, pase atrás, centro al área...).

Con esta información sobre un histórico suficientemente grande podemos asignar una probabilidad de que termine en gol a cada tiro mediante modelos logísticos.

Los datos de StatsBomb nos facilitan información suficiente para representar la situación del disparo (ver *ilustración 4-1*). Con esto se puede mejorar el modelo de expected goal.

- Posición del portero en cada remate.
- Colocación del portero: tumbado, movimiento o fijo.
- Posición de los defensores y atacantes en cada remate.



Ilustración 4-1. Freeze frame de un disparo. Fuente StatsBomb.



Esta información es muy útil a la hora de valorar las distintas ocasiones ya que no es lo mismo rematar un balón “a portería vacía” que hacerlo con muchos oponentes en el camino entre el balón y la portería.

Otra posible variable para incluir en los modelos es una tercera coordenada z. La altura del balón en el momento del remate. No es lo mismo rematar un balón a ras de suelo que rematar un balón a 1 metro de altura.

#### 4.1.3 Mapa de tiro. Shot Map.

La mejor manera de representar el concepto *Expected Goal* gráficamente son los *Mapas de Tiros* o *Shot Map*.

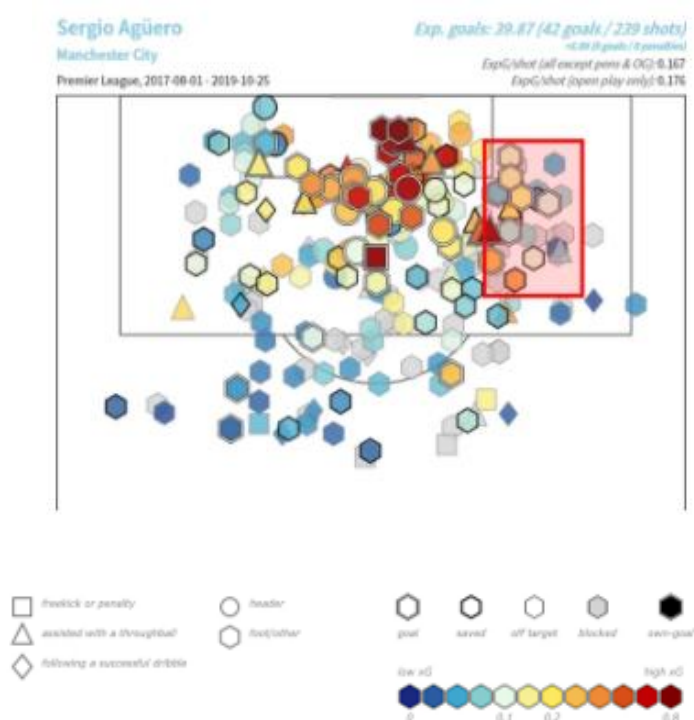


Ilustración 4-2. Mapa de tiros. Fuente StatsBomb

En la *ilustración 4-2* podemos ver los tiros realizados por Sergio Agüero de 2017 a 2019. Con distintas características mencionadas anteriormente:

- La parte del cuerpo con la que se realiza el remate y el tipo de pase previo se muestran en las figuras geométricas.
- El ángulo a portería y la distancia lo muestra la situación del propio mapa.
- Por último el color representa el expected Goal asignado por el modelo StatsBomb a cada tiro. El borde de las figuras indican el resultado del tiro: Gol, fuera, bloqueado o parado.

Las características de la ocasión son más importantes que el jugador que realiza el lanzamiento. Bien es cierto que algunos jugadores destacan por sobresalirse y marcar muchos más goles que lo estimado por los modelos, como es el caso de Leo Messi. También al contrario, hay jugadores que consiguen marcar menos goles que lo estimado, como Jesús Navas. Las estimaciones se amoldan al grueso de jugadores.

#### 4.1.4 Otras métricas basadas en el Expected Goal.

Los datos de StatsBomb incluyen en cada tiro la variable  $\text{shot.xG}$ , la cual indica el valor de ese tiro para el modelo de StatsBomb.

Cuando hablamos de **Expected Goal generado**, nos referimos a la suma de todos los expected goal de las ocasiones generadas. Por ejemplo si un equipo ha generado 3 ocasiones con un expected goal de 0.12, 0.6 y 0.4, este equipo ha generado un expected goal de 1.12. Decimos que este equipo ha anotado por debajo de su rendimiento si este equipo no ha conseguido hacer gol. De igual manera decimos que este equipo ha anotado por encima de su rendimiento si ha conseguido hacer 2 o 3 goles.

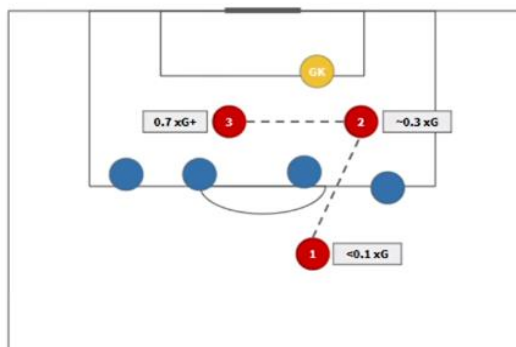
También se puede aprovechar esta métrica para valorar asistencias. Una asistencia de tiro se define como el pase que precede al disparo. El **expected goal asistido** es el valor que le damos a una asistencia, siendo este valor el expected goal del disparo que asiste. No es lo mismo dar una asistencia a un jugador en una situación en la que tiene un expected goal de 0.9 como puede ser dentro del área pequeña que una asistencia a un jugador en peor situación de tiro.

Las aplicaciones que tiene esta métrica son todas las que se nos puedan ocurrir. A continuación pondremos tres breves ejemplos: toma de decisiones en el juego, rendimiento de un jugador y rendimiento de un equipo.

- Toma de decisiones:

El expected goal es una herramienta de la que no sólo se pueden aprovechar los analistas y cuerpo técnico, si no que también le pueden dar utilidad los jugadores a la hora de tomar decisiones en el juego.

Se ve claramente en la situación del “pase extra”. Hay momentos en el que es mejor dar un pase a un jugador que está en mejor situación que realizar el disparo directamente como se muestra en la *ilustración 4-3*.



*Ilustración 4-3. Pase extra.*

- Evaluar el rendimiento real de un jugador:

Un ejemplo muy ilustrativo llevado por *StatsBomb* es el caso de *Enis Bardhi*, jugador del Levante UD. Este jugador consiguió anotar 9 goles la temporada 2017/2018. Esto llama la atención para un centrocampista. (Ver ilustración 4-4)

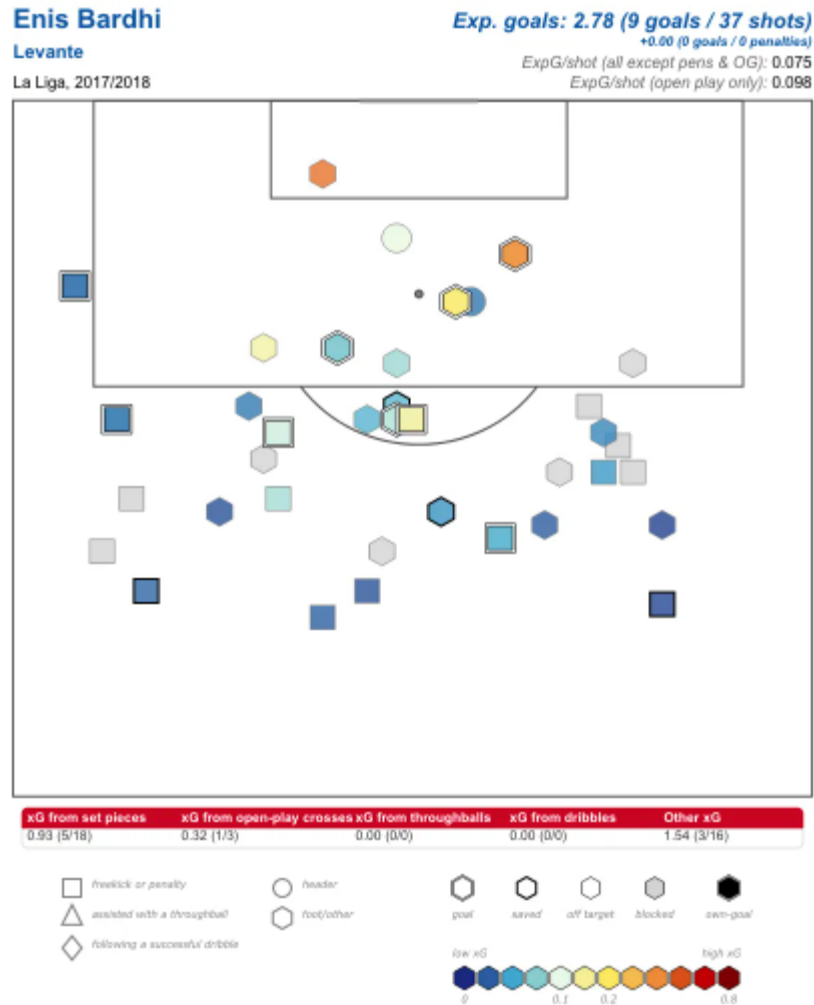


Ilustración 4-4. Mapa de tiros. Fuente StatsBomb

La explicación es que el jugador estaba rindiendo muy por encima de su Expected Goal: 9 goles frente a 2.78 xG generado. Si un club hubiese fichado a ese jugador esperando que marque 9-10 goles por temporada habría cometido un error. La temporada siguiente el mismo jugador generó un Expected Goal incluso mayor pero sólo consiguió anotar 3 goles.

- Evaluar rendimiento de un equipo:

En este caso nos fijaremos en la temporada 2018/2019 del Deportivo Alavés. Este equipo llegó a mitad de temporada en los primeros puestos de la clasificación de La Liga, 5º puesto. Lo que sucedía es que el equipo estaba rindiendo muy por encima del Expected Goal estimado. En la segunda parte de la temporada el rendimiento del equipo se amoldó al Expected Goal generado incluso bajó por debajo y acabó la temporada en el puesto 11º.

En la *ilustración 4-5* el área verde es la diferencia de goles anotados respecto al Expected Goal generado y el área morada es la diferencia entre el expected Goal generado y los goles anotados. Esta temporada 2020/2021 le está ocurriendo lo mismo al Elche CF. Está rindiendo muy por encima del Expected Goal generado. Su puesto ahora mismo es el 10º.

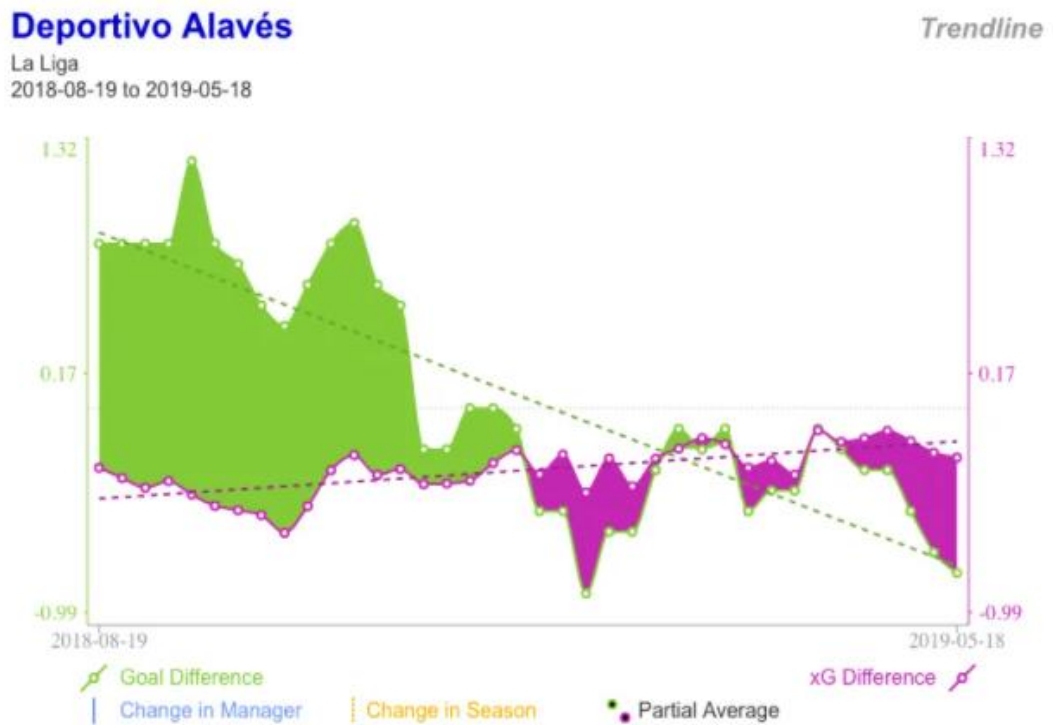


Ilustración 4-5. Diferencia expected Goal.

## 4.2 Ajuste de eventos por 90 minutos

Para poder ilustrar la aportación de un jugador de manera objetiva el primer paso es estandarizar sus acciones, así podremos compararle con otros jugadores. Una primera aproximación es utilizar el ajuste por 90 minutos

El primer motivo por lo que se realiza este ajuste es que un jugador tendrá más o menos posibilidades de realizar cualquier tipo de acción en función del tiempo que esté en el campo.

En cierta medida el tiempo que juega cada jugador es independiente a él. Un jugador juega los minutos en los que su entrenador cree oportuna su participación. Un jugador juega más o menos minutos en función de su estado físico, este motivo si que depende de él. Pero un jugador puede disputar menos minutos por decisiones tácticas que toma su entrenador. Por eso decimos que el tiempo que juega cada jugador no depende de él al 100%.

Estandarizar las acciones por partido disputado está fuertemente sesgado ya que hay una gran diferencia entre jugadores que disputan todo el partido y jugadores que sólo juegan una pequeña parte del partido ya que no han salido de inicio. Para corregir este inconveniente la solución que se toma es ajustar las medidas de rendimiento por cada 90 minutos de juego.

Por ejemplo para ajustar los tiros realizados:

$$\text{Tiros cada 90 minutos} = (90/\text{minutos jugados}) * \text{total tiros}$$

La necesidad de este ajuste se muestra con un ejemplo en la *tabla 7*. La jugadora Lucy Quinn, con 22 tiros, y Anghard James, con 23 disparos, han tirado prácticamente las mismas veces, incluso una vez más Anghard. Pero el volumen de tiros por cada 90 minutos es bien distinto. Lprimera promedia 3.08 tiros por 1.33 tiros cada 90 minutos de la segunda.

Jugador	Equipo	Minutos jugados	Tiros Totales	Tiros/90
Lucy Quinn	Birmingham	643	22	3.08
Angharad James	Everton	1559	23	1.33

Tabla 7. Tiros ajustados.

También se puede apreciar la función de esta estandarización en otras situaciones. En la *tabla 8* Daniele del Arsenal ha realizado el doble de disparos que Jessica Anne, sin embargo promedian prácticamente el mismo volumen de tiros por cada 90 minutos.

Jugador	Equipo	Minutos jugados	Tiros Totales	Tiros/90
Daniele Van de Donk	Arsenal	2865	59	1.85
Jessica Anne Clarke	Liverpool	1378	28	1.83

Tabla 8. Tiros ajustados.

Para poder utilizar esta estandarización de manera más precisa es necesario comprobar el tamaño de la muestra del jugador. No es raro que un jugador que ha jugado un número muy reducido de minutos haya realizado los suficientes disparos para que su volumen de tiros por

90 minutos sean altos. Si este jugador jugase más minutos sería muy difícil que esa media se mantuviese tan elevada. Un caso extremo es el ejemplo de Olivia Chance (ver *tabla 9*), que promedia 28.4 tiros cada 90 minutos. Esto es debido a que tan solo ha disputado 38 minutos realizando 12 disparos.

Jugador	Equipo	Minutos jugados	Tiros Totales	Tiros/90
Olivia	Everton	38	12	28.42

*Tabla 9. Tiros de Olivia.*

Para evitar este tipo de problema se suele exigir una cantidad mínima de minutos jugados. Lo más utilizado es 900 minutos, abarcando 10 partidos completos, pero esta cantidad puede variar en función de la competición. Por ejemplo, no es lo mismo una liga de 20 equipos en los que cada equipo disputa 38 partidos, que una liga como es el caso de *FA Women's Super League* (datos de este TFG) formada por 11 equipos dónde cada equipo disputa 20 partidos.

Por último, hay que tener en cuenta que las métricas cada 90 minutos no están exentas de sesgos. El más significativo es el relacionado a las sustituciones. Se ha comprobado que los jugadores atacantes que entran al campo desde el banquillo tienen mejor promedio de tiros, goles o asistencias cada 90 minutos. Las razones que explican esto pueden ser varias. Estos jugadores entran al campo con mayor capacidad física que sus defensores los cuales ya tienen una fatiga de minutos acumulada en el partido. Tienen más facilidad de generar acciones respecto a sus compañeros que se enfrentaron a estos defensores en igualdad de condiciones con respecto a la fatiga.

A nivel táctico es muy frecuente que un entrenador introduzca a un jugador atacante en el campo debido a que tiene un resultado adverso en el marcador. El equipo busca generar más acciones de ataque aunque descuide el apartado defensivo. Esto provoca que aumente el volumen de acciones de ataque en los últimos minutos de los partidos.

### 4.3 Ajuste por Posesión para acciones defensivas

Cuando se introdujo el mundo del análisis de datos en el fútbol unos de los grandes problemas fueron la evaluación de las métricas defensivas. Las métricas no aportaban información estable tanto a nivel de equipo como a nivel de jugadores. Los resultados que se obtenían respecto a las acciones defensiva con los tiros y goles concedidos eran incoherentes. Los equipos que más acciones defensivas, incluso los que más porcentaje de acierto en los intentos de acciones defensivas tenían no guardaban relación con la cantidad de ocasiones y goles que concedían.

Un paso adelante para utilizar estos datos fue tener en cuenta la posesión del balón. Un equipo cuando está en posesión del balón, ni realiza acciones defensivas, ni concede ocasiones al rival. Si no se tiene en cuenta la posesión del balón, los datos indican que es contraproducente realizar acciones defensivas si quieres evitar conceder acciones de ataque del rival.

Con el fin de entender esto estudiaremos un ejemplo con los equipos de “FA Women’s Super League”.

En la *tabla 10* se presentan los tiros que conceden los distintos equipos, la suma de las intercepciones y las entradas exitosas de cada uno de ellos. Todas estas variables están estandarizadas por partido. Las intercepciones y las entradas exitosas no son el las únicas acciones defensivas que se pueden valorar pero sí que son las más representativas. También se adjunta la posesión para comprobar su repercusión en la representación de este tipo de variables. Hay que recordar que la variable posesión representa el porcentaje del tiempo que ha tenido el balón un equipo durante el partido.

Equipo	Tiros concedidos	Acciones Defensivas	Posesión
Chelsea FCW	7.59	16.71	60.09
Manchester City WFC	11.03	18.5	56.93
Brighton & Hove Albion WFC	13.46	23.08	40.69
Bristol City WFC	17.62	17.8	36.1
Reading WFC	11.56	17.56	56.26
Yeovil Town LFC	24.21	19.89	30.22
Arsenal WFC	7.49	17.14	64.86
Liverpool WFC	14.48	22.42	40.42
Birmingham City WFC	10.97	20.75	57.09
West Ham United LFC	14.82	18.21	50.26
Tottenham Hotspur Women	17.13	20.34	47.19
Everton LFC	13.91	19.73	45.27
Equipo	Tiros concedidos	Acciones Defensivas	Posesión
Manchester United	9.57	15.28	52.8

*Tabla 10. Tiros concedidos, acciones defensivas y Posesión FA Women’s Superleague.*

En la *tabla 10* se han destacado los casos más representativos de este fenómeno. La posesión del *Arsenal WFC* es la más alta de la liga, la del del *Yeovil Town LFC* la más baja y el *West Ham*

*United LFC* tiene cerca del 50% de posesión. Cuando comparamos los valores de estos equipos para estas variables concluimos que:

- Las medias de las acciones defensivas y de los tiros concedidos son 18.87 y 13.4 respectivamente.
- El *Arsenal WFC* realizando menos acciones defensivas que la media, concede menos tiros que la media. Mientras que el *Yeovil Town LFC* realiza más acciones defensivas y encaja más tiros que la media. Este es un claro ejemplo de este fenómeno. El *Arsenal WFC* tiene mucho tiempo el balón luego no realiza tantas acciones como el *Yeovil Town LFC* que apenas tiene el balón en su poder.
- En el caso del *West Ham United LFC* siendo un equipo con posesión neutra como sus valores están muy cerca de la media, teniendo más sentido.

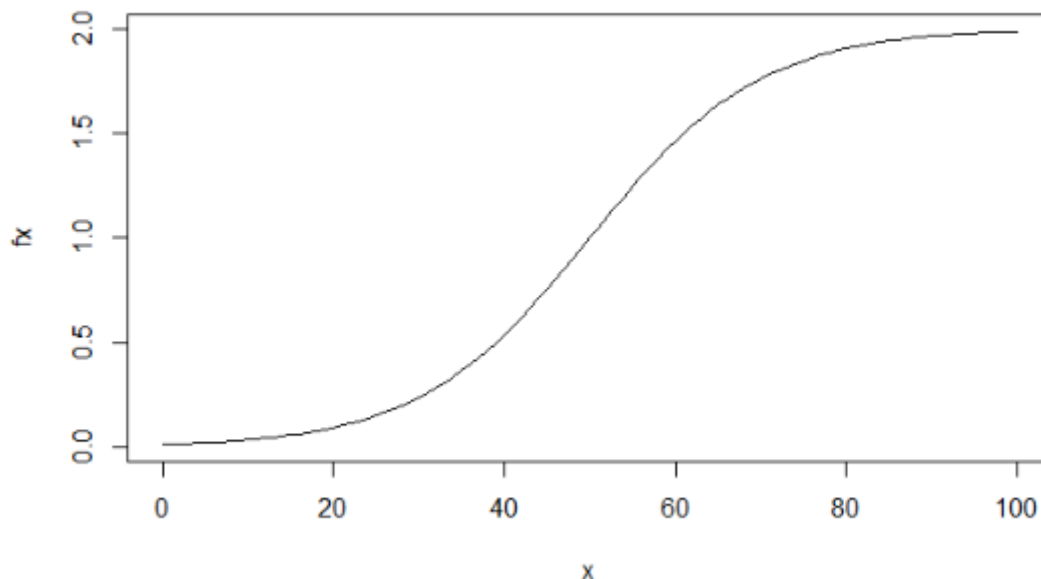
Estos ejemplos muestran con claridad la necesidad de un ajuste para estas variables. Además la posesión ha de tenerse en cuenta.

*StatsBomb* ajusta las acciones mediante una fórmula sigmoidea (ver ilustración 4-6). Mediante este ajuste consiguen independizar las acciones defensivas con respecto a la posesión. Se da más importancia a las acciones defensivas de los equipos que más posesión tienen y menos a los equipos que tienen menos posesión. En este trabajo ajustaremos los datos defensivos en función de la posesión de la misma manera de la que lo hace *StatsBomb*.

La fórmula en cuestión es la siguiente:

$$\text{Acciones} * \frac{2}{1 + e^{-0.1*(x-50)}}$$

$$x \in [0,100]$$



ilustracion 4-6. Función de ajuste de posesión.

Donde x es la posesión del balón del equipo, siendo 0 no tener nunca y 100 tener el balón todo el tiempo.



En la *tabla11* actualizamos los datos de la *tabla10* ajustando esta vez los datos defensivos en función de la posesión. Podemos ver información con más sentido que antes.

Equipo	Tiros concedidos	Acciones Defensivas	Posesión %
Chelsea FCW	7.59	24.48	60.09
Manchester City WFC	11.03	24.67	56.93
Brighton & Hove Albion WFC	13.46	13.06	40.69
Bristol City WFC	17.62	7.1	36.1
Reading WFC	11.56	22.8	56.26
Yeovil Town LFC	24.21	4.83	30.22
Arsenal WFC	7.49	27.95	64.86
Liverpool WFC	14.48	12.44	40.42
Birmingham City WFC	10.97	27.82	57.09
West Ham United LFC	14.82	16.41	50.26
Tottenham Hotspur Women	17.13	17.49	47.19
Everton LFC	13.91	15.15	45.27
Manchester United	9.57	17.41	52.8

*Tabla 11. Tiros concedidos, acciones defensivas ajustadas por posesión, posesión FA Women`s Superleague.*

## 5 Paquete GASB

En el paquete *GASB*, creado en la realización de este TFG, se han implementado funciones con las que realizar análisis estadísticos que emplean las empresas y equipos a nivel mundial. El paquete es compatible con cualquier base de datos en formato *StatsBomb*.

Este apartado tiene dos objetivos:

- Explicar todas las funciones del paquete a modo de documentación del mismo.
- Explicar el funcionamiento y la finalidad de estas funciones en el análisis del juego. Así se muestra cómo se usan los datos en el mundo del fútbol.

Elementos que componen el paquete GASB:

- Archivos de datos:
  - o eventsLF
  - o rendimiento
- Funciones:
  - o crear\_StatsBombpitch()
  - o shotMapSB()
  - o seleccionEventos()
  - o calculoPosesion()
  - o minutosJugados()
  - o calculoXGAssist()
  - o calculoXGbuild()
  - o rendimientoJugadores()
  - o SBMAPpases()
  - o matrizPases()
  - o redPases()
  - o sonarJugador()
  - o mapaSonar()
  - o analisisDefensivoEquipo()
  - o analisisDefensivoJugador()
  - o XGMap()
  - o assistMap()
  - o freezeFrame()
  - o freezeFrame3d()

- radarJugador()
- jugadorSimilar()

A continuación se describirán con detalle todos los elementos que forman el paquete GASB.

## 5.1 Archivos de datos

El paquete GASB cuenta con dos archivos de datos para realizar distintos análisis que se utilizan para los ejemplos que se muestran a lo largo de este apartado: “eventsLF” y “rendimiento”.

Se puede utilizar cualquier base de datos con formato StatsBomb.

### 5.1.1 eventsLF

El archivo de datos **eventsLF** contiene los datos descritos en el *Apartado 3.4: Datos de partidos de fútbol utilizados*. Este archivo recoge 648877 eventos pertenecientes a partidos de fútbol disputados en la *FA Women’s Superleague* en las temporadas 2018/2019 y 2019/2020. Cada evento tiene 187 variables descritas detalladamente en el apéndice.

### 5.1.2 rendimiento

El archivo de datos **rendimiento** recoge los datos necesarios para hacer análisis de jugadores según su rendimiento. El archivo *rendimiento* se compone de 313 observaciones (313 jugadores que participaron en la *FA Women’s Superleague* en las temporadas 2018/2019 y 2019/2020) y 20 variables descritas a continuación.

- player.id: Identificador del jugador.
- posicion: Número que identifica su posición en el campo. (Ver *ilustración 3-2*)
- passing: Porcentaje de acierto en el pase.
- pressures: Acciones de presión.
- fouls: Faltas cometidas.
- tackDribbled La proporción de entradas exitosas a ocasiones en que el oponente le/la regateó.
- PAdjTackles: Entradas exitosas ajustadas en función de la posesión.
- PAdjInterceptions: Interceptaciones ajustadas en función de la posesión.
- AerialWins: Duelos aéreos ganados.
- AerialWinsPor: Porcentaje de duelos aéreos ganados.
- PressuredLongBalls: Balones largos bajo presión.
- UnPressuredlongBalls: Balones largos sin presión.

- Xgbuild: La suma de los goles esperados de todos los ataques del equipo que terminaron en tiro en los que el jugador tocó el balón pero no realizó el tiro o la asistencia del tiro.
- deepProgressions: Incursiones en el último tercio a través de pases, regates o conducciones.
- SuccesDribbles: Regates exitosos.
- Turnovers: Perdidas de posesión a través de fallos en el control del balón o regates fallidos.
- XGAssisted: Goles esperados asistidos.
- pressureRegains: Recuperaciones del equipo del jugador dentro de cinco segundos de una presión realizada por parte del jugador.
- foulsWon: Faltas recibidas.
- XgContra: Expected Goal generado por los rivales con el jugador en el campo.

Todas las acciones descritas están ajustadas por 90 minutos y por posesión en las acciones que lo requieren como es el caso de las intercepciones y las entradas al rival.

## 5.2 Campo de fútbol.

Para gran parte del análisis es necesario ilustrar los datos y las acciones mediante una representación sobre el campo de fútbol. Estas funciones del paquete nos ofrecen los análisis más importantes.

### 5.2.1 crear\_StatsBombpitch()

La primera función que se presenta en este paquete es una función que representa un campo de fútbol. Esta función resulta muy útil para recrear determinadas acciones del juego o resúmenes de un partido completo.

La función **crear\_statsBombPitch()** crea un gráfico que simula un terreno de juego de 73 metros de ancho por 110 metros de largo. (80 yardas de ancho por 120 yardas de largo.)

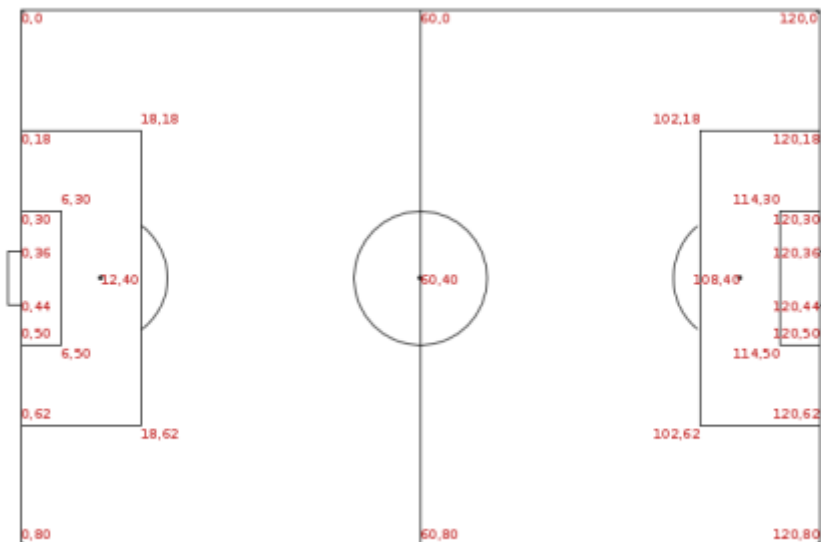


Ilustración 5-1. Coordenadas del campo. Fuente StatsBomb.

Argumentos:

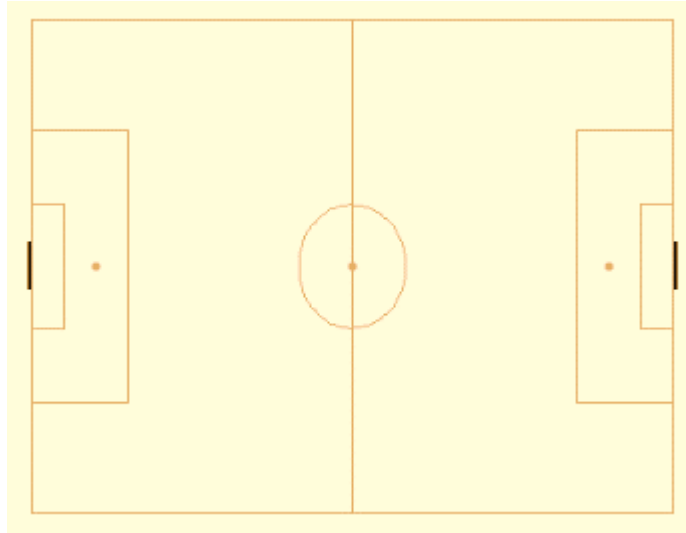
- *colorCésped*: Color del césped. Por defecto blanco.
- *colorLinea*: color de las líneas. Por defecto gris.
- *colorFondo*: color del fondo fuera del campo. Por defecto blanco.
- *colorGol*: Color de las porterías. Por defecto negro.
- *BasicFeatures*: Lógico. Dibujar sólo las líneas esenciales o también el círculo central y los balcones de los áreas. Por defecto *true*.

- *horiz*: dibujar el campo de manera horizontal o vertical. Por defecto *TRUE* (horizontal)

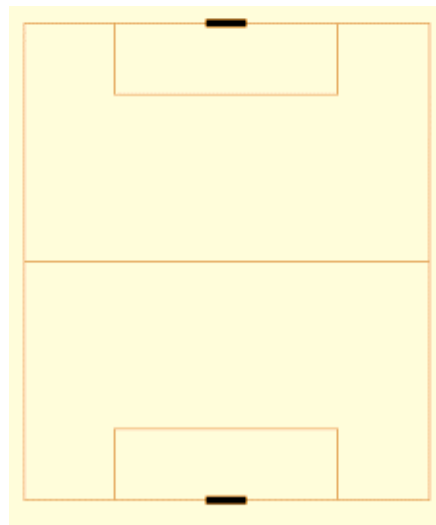
Ejemplos:

- `crear_StatsBombPitch()`
- `crear_StatsBombPitch(BasicFeatures = T, horiz = F)`

Los resultados se muestran en las *ilustraciones 5-2 y 5-3* respectivamente.



*Ilustración 5-2. Salida crear\_StatsBombPitch()*



*Ilustración 5-3.- crear\_StatsBombPitch(BasicFeatures = T, horiz = F)*

En muchas ocasiones se analizan aspectos del juego en el que se fijan con más detalle su situación con respecto a la portería. Como pueden ser disparos, centros al área o distintas jugadas de ataque en campo contrario o a balón parado. Para ello se crea un mapa vertical en el que solo se incluye una mitad del campo.

### 5.2.2 ShotMapSB()

La función *ShotMapSB()* crea un mapa recreando el campo contrario.

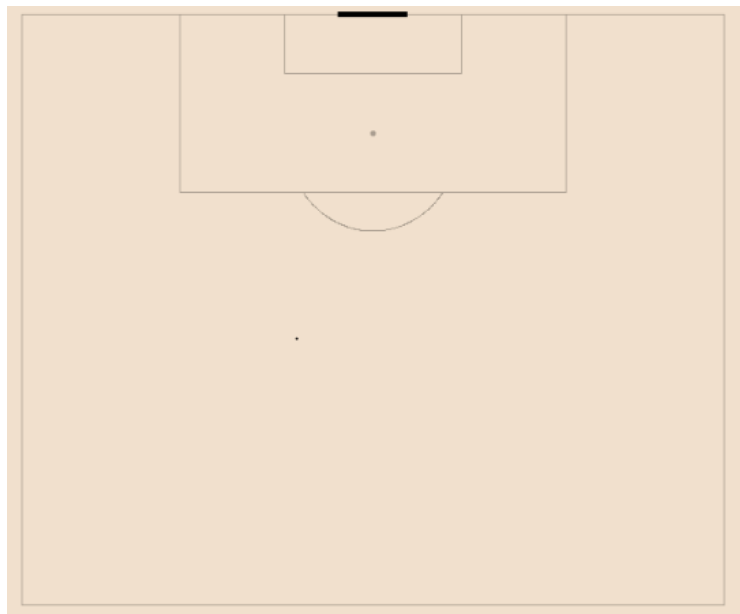
Argumentos:

- *colorCésped*: Color del césped. Por defecto blanco.
- *colorLinea*: color de las líneas. Por defecto gris.
- *colorFondo*: color del fondo fuera del campo. Por defecto blanco.
- *colorGol*: Color de las porterías. Por defecto negro.

Ejemplo:

ShotMapSB()

Ver resultado en la *ilustración 5-4*



*Ilustración 5-4.ShotMapSB()*

## 5.3 Funciones auxiliares

En esta sección se incluyen funciones esenciales para realizar los análisis que son requeridas por otras funciones del paquete GASB dedicado al análisis del juego.

### 5.3.1 `seleccionEventos()`

la función **`seleccionEventos()`** permite seleccionar los eventos de interés que queremos analizar a partir de datos de una o varias competiciones. A veces no necesitamos todos estos datos sino una parte como los referidos a un único equipo, un único partido o un jugador determinado.

Para realizar los distintos análisis primero debemos seleccionar los eventos de interés. Los datos con los que trabajamos en este TFG como se explica en el *Apartado 3.4: Datos de partidos de fútbol utilizados* de este trabajo son los correspondientes a las temporadas 2018/2019 y 2019/2020 de *FA Women's Super League*.

Con la función `seleccionEventos()` se puede hacer esa selección de los eventos objetivo del para un determinado análisis.

Argumentos:

- *eventos*: El argumento *eventos* es un data frame con los eventos de nuestra base de datos en formato StatsBomb. Este argumento es obligatorio.
- *idEquipo*: Argumento de tipo integer. Sirve para seleccionar los eventos de un equipo en concreto mediante su identificador (*team.id*), este identificador es único para cada equipo. Este argumento es opcional, si se omite no se seleccionarán los datos por equipo.
- *idPartido*: Argumento de tipo integer. Sirve para seleccionar los datos de un partido en concreto mediante su identificador (*match\_id*). Este argumento es opcional.
- *idJugador*: Argumento de tipo integer. Selecciona los eventos realizados por un determinado jugador mediante su identificador (*player.id*). Este argumento es opcional.
- *rival*: Argumento de tipo lógico. Este argumento se usa cuando queremos hacer un análisis de un equipo y necesitamos los datos de sus rivales. *FALSE* : sin datos del rival, *TRUE* con datos del rival. Por defecto *FALSE*.

Los argumentos se pueden combinar en función del objetivo del análisis.

Salida:

- La función `seleccionEventos()` devuelve un dataframe con los eventos requeridos.

Ejemplo:

Partiendo de `eventsLF`, datos descargados de StatsBomb web e incluidos en este paquete, queremos obtener los referidos al *Everton LFC* junto al de sus rivales.

Hay que comprobar el *team.id* del *Everton LFC*. En este caso es "967".

```
seleccionEventos(eventsLF, idEquipo = 967, rival = T)
```



### 5.3.2 calculoPosesion()

El objetivo de la función **calculoPosesion()** es conseguir el porcentaje de posesión que tiene un equipo en particular en uno o más partidos.

La posesión se calcula a partir del tiempo que ha tenido el balón un equipo con respecto al rival. Mediante las variables *duration* y *possession\_team.id*.

La variable *duration* indica el tiempo que ha durado una acción de un equipo y con *possession\_team.id* identificamos que equipo es el responsable de esa acción. Mediante la suma de las duraciones de los eventos realizados por el equipo y la suma de la duración de los eventos de sus rivales obtenemos los datos necesarios para el cálculo.

Hay eventos de tipo *Substitution*, *StartingXI...* que no representan posesión del balón pero todos estos tienen duración 0. Los eventos de tipo *Pressure*, estos eventos si que tienen duración y no representan posesión del balón, por lo que los tenemos que eliminar del dataframe de estudio.

Con la siguiente ecuación obtenemos la posesión del equipo objetivo:

$$\text{Posesión} = \frac{\text{Tiempo Equipo}}{\text{Tiempo Equipo} + \text{Tiempo Rival}} * 100$$

Argumentos:

- *eventos*: Dataframe con los eventos de los partidos que se quiere calcular. Obligatorio.
- *idEquipo*: Argumento tipo integer que indica el identificador del equipo objetivo de estudio. Obligatorio.

Salida:

- Numérico entre 0 y 100 indicando el porcentaje del tiempo que ha tenido el equipo el balón.

Ejemplo:

Posesión del *Everton LFC* en todos los partidos de la *FA Women's Super League* .

Entrada:

```
calculoPosesion(eventsLF, idEquipo = 967)
```

Salida:

45.21037

### 5.3.3 minutosJugados().

Se necesita saber los minutos que juega un determinado jugador en un marco de partidos concreto, ya que no todos juegan la misma cantidad. Con la función **minutosJugados()** calculamos los minutos que juega un jugador particular.

Para calcular el tiempo que un jugador ha estado en el campo se considera partido por partido si el jugador sale de inicio, es sustituido o sale en otro momento del juego

Argumentos:

- *eventos*: Dataframe. Marco de eventos que queremos estudiar. Obligatorio.
- *idJugador*: Integer. Identificador del jugador objetivo. Obligatorio.

Salida:

- Integer con la cantidad de minutos que ha disputado un jugador.

Ejemplo:

Minutos que ha jugado *Jessica Carter*, jugadora del *Chelsea FCW* durante todos los partidos de nuestra muestra. Su *player.id* es 19422.

Entrada:

```
minutosJugados(eventsLF, idJugador = 19422)
```

Salida:

848

#### 5.3.4 calculoXGAssist().

El expected goal asistido consiste en dar el valor de expected goal de un tiro al pase que le precede, es decir, a su asistencia. Como ocurre en la falta de goles para valorar de manera consistente el rendimiento de un jugador o de un equipo, ocurre en la falta de asistencias, por este motivo le damos a cada asistencia de un tiro el valor de expected goal del tiro al que asiste.

Gracias a que los datos de StatsBomb facilitan el valor de Expected Goal de cada tiro podemos fácilmente identificando el pase que le precede dar a este pase su valor correspondiente.

La función ***calculoXGAssist()*** calcula el total del Expected Goal asistido de un jugador en concreto en un marco de datos específico (un partido, una temporada...).

Argumentos:

- *eventos*: Dataframe. Marco de eventos que se quiere estudiar. Argumento obligatorio.
- *idJugador*: Integer. Identificador del jugador objetivo. Argumento obligatorio.

Salida:

- *numérico* con el total de expected goal asistido por el jugador.

Ejemplo:

Calcularemos el expected Goal asistido de *Jessica Carter*, jugadora del ejemplo anterior.

Entrada:

```
calculoXGAssist(eventsLF, idJugador = 19422)
```

Salida:

0.4307297

### 5.3.5 calculoXGbuild().

El expected Goal construido (Expected Goal Build) de un jugador es la suma de todos los expected goals de todas las jugadas en la que el jugador participó en la posesión del balón.

Argumentos:

- *eventos*: Dataframe. Marco de eventos que se quiere estudiar. Argumento obligatorio.
- *idJugador*: Integer. Identificador del jugador objetivo. Argumento obligatorio.

Salida:

- Numérico con el total de expected goal build del jugador.

Ejemplo:

Calcularemos el expected Goal construido de *Jessica Carter*..

Entrada:

```
calculoXGbuild(eventsLF, idJugador = 19422)
```

Salida:

7.420143

### 5.3.6 rendimientoJugadores()

El rendimiento de los jugadores se valora según el distinto volumen de acciones que hayan realizado. En este trabajo se analizan las acciones que tiene en cuenta StatsBomb.

En el paquete *GASB* se implementa el cálculo para las variables características de los defensas y los centrocampistas. Se calculan los valores de todos los jugadores de un marco de eventos para así tener el contexto del mismo.

A continuación se detallan las variables más importantes que se tienen en cuenta para valorar el rendimiento de un jugador dependiendo de su posición.

Defensa central:

- Passing%: Porcentaje de acierto en el pase.
- Pressures: Acciones de presión.
- Fouls: Faltas cometidas.
- Tack/Dribbled Past%: La proporción de entradas exitosas a ocasiones en que el oponente le/la regateó.
- PAdj Tackles: Entradas exitosas ajustadas en función de la posesión.
- PAdj interceptions: Interceptaciones ajustadas en función de la posesión.
- Aerial Wins: Duelos aéreos ganados.
- Aerial Win%: Porcentaje de duelos aéreos ganados.
- Pressured Long Balls: Balones largos bajo presión.
- Unpressured long Balls: Balones largos sin presión.
- XGBuildup: La suma de los goles esperados de todos los ataques del equipo que terminaron en tiro en los que el jugador tocó el balón pero no realizó el tiro o la asistencia del tiro.

Defensa lateral:

- Passing%: % Acierto en el pase.
- Pressures: Acciones de presión.
- Fouls: Faltas cometidas.
- Tack/Dribbled Past%: La proporción de entradas exitosas a ocasiones en que el oponente le/la regateó.
- PAdj Tackles: Entradas exitosas ajustadas en función de la posesión.
- PAdj interceptions: Interceptaciones ajustadas en función de la posesión.
- Aerial Wins: Duelos aéreos ganados.
- Aerial Wins: Duelos aéreos ganados.

- XGBuildup: La suma de los goles esperados de todos los ataques del equipo que terminaron en tiro en los que el jugador tocó el balón pero no realizó el tiro o la asistencia del tiro.
- Deep Progressions: Incursiones en el último tercio a través de pases, regates o conducciones.
- Successful Dribbles: Regates exitosos.
- Turnovers: Perdidas de posesión a través de fallos en el control del balón o regates fallidos.

#### Centrocampistas:

- Passing%: Porcentaje de acierto en el pase.
- Pressures: Acciones de presión.
- Turnovers: Perdidas de posesión a través de fallos en el control del balón o regates fallidos.
- Deep Progressions: Incursiones en el último tercio a través de pases, regates o conducciones.
- XGBuildup: La suma de los goles esperados de todos los ataques del equipo que terminaron en tiro en los que el jugador tocó el balón pero no realizó el tiro o la asistencia del tiro.
- PAdj Tackles: Entradas exitosas ajustadas en función de la posesión.
- PAdj interceptions: Interceptaciones ajustadas en función de la posesión.
- XG Assisted: Goles esperados asistidos.
- Pressure Regains: Recuperaciones del equipo del jugador dentro de cinco segundos de una presión realizada por parte del jugador.
- Successful Dribbles: Regates exitosos.
- Fouls Wons: Faltas recibidas.

#### Extremos y mediapuntas:

- XG: Goles esperados.
- Shots: tiros.
- Touches in box: Toques del balón dentro del área de penalti.
- Successful Box Cross%: El porcentaje de los pases completados al área que fueron centros.
- Open Play XG Assisted: Goles esperados asistidos en juego normal (es decir, sin contar los que provienen a balón parado)
- XG/Shot: Goles esperados por tiro. Es decir, la calidad media de sus tiros.
- Fouls Wons: Faltas recibidas.

- Turnovers: Perdidas de posesión a través de fallos en el control del balón o regates fallidos.
- Pressure Regains: Recuperaciones del equipo del jugador dentro de cinco segundos de una presión realizada por parte del jugador.
- Successful Dribbles: Regates exitosos.

#### Delanteros:

- XG: Goles esperados.
- Shots: tiros.
- Touches in box: Toques del balón dentro del área de penalti.
- XG Assisted: Goles esperados asistidos.
- Pressure Regains: Recuperaciones del equipo del jugador dentro de cinco segundos de una presión realizada por parte del jugador.
- Pressures: Acciones de presión.
- Turnovers: Perdidas de posesión a través de fallos en el control del balón o regates fallidos.
- Successful Dribbles: Regates exitosos.
- XG/Shot: Goles esperados por tiro. Es decir, la calidad media de sus tiros.
- Aerial Wins: Duelos aéreos ganados.
- Shot Touch%: El porcentaje de sus toques del balón que fueron tiros.

#### Argumentos:

- *Events*: Dataframe con el marco de datos a estudiar.

#### Salida:

- Dataframe con las variables objetivo de estudio para defensas y centrocampistas además de su posición su identificador y el expected goal generado por sus rivales mientras ellos están en el campo.

#### Ejemplo:

El rendimiento de los 313 jugadores de la liga *FA Women`s League* estudiada en este trabajo se obtiene con el siguiente comando y una fracción del resultado se puede ver en la *ilustración 5-5*.

```
rendimientoJugadores(eventsLF)
```

	position	passing	deepProgresions	XgAssisted	succesDribbles	foulsWon	Turnovers
15555	16	74.47307	3.7823834	0.105700518	1.96891192	1.19170984	4.61139896
4643	19	79.62675	4.9907579	0.151871253	2.41219963	1.58040665	4.28373383
10185	3	82.44156	7.7222222	0.050819955	0.361111111	0.16666667	0.611111111
15554	5	88.17204	3.5332606	0.022149612	0.52344602	0.26172301	0.71973828
17524	5	88.67805	7.6470588	0.030539068	0.41176471	0.23529412	0.52941176
10170	1	77.53623	1.0000000	0.000000000	0.00000000	0.20000000	0.20000000
4659	17	71.75573	4.9080882	0.125298177	3.03308824	1.59926471	4.02573529
15553	21	68.40149	5.5792683	0.324353781	1.55487805	1.09756098	3.20121951
10395	2	85.75458	4.5544554	0.154928845	0.79207921	0.54455446	2.27722772
15550	23	66.58163	2.1808741	0.117942003	1.05149286	0.50627434	3.07659022
10172	13	81.27813	5.7232704	0.079584611	1.32075472	0.91194969	2.29559748
10222	6	74.20349	5.5459459	0.108408819	0.58378378	0.35675676	1.29729730
15549	10	78.73786	7.0652174	0.043451237	0.68840580	0.72463768	0.57971014
4658	15	83.71810	10.1212121	0.085569084	1.39393939	0.69696970	1.69696970
4642	3	78.08418	7.3754513	0.080830901	0.25992780	0.42238267	0.71480144
4640	1	77.49077	0.7777778	0.000000000	0.111111111	0.00000000	0.00000000
4647	19	85.42857	7.0000000	0.215052372	2.00000000	1.22222222	2.66666667

pressureRegains	pressure	PAAdjTackles	PAAdjInterceptions	fouls	tackDribbled	Aerial Wins	Aerial WinsPor
5.1295337	25.12953368	2.6726056	1.33630279	2.3316062	71.26437	1.13989637	51.162791
4.7412200	21.91774492	1.5807088	0.84680828	5.8225508	67.56757	0.45748614	45.833333
1.1666667	6.25000000	0.7918180	0.64099553	1.3333333	65.51724	1.97222222	71.717172
1.5376227	7.03380589	0.9325665	0.75493476	1.4721919	65.07937	1.43947655	65.671642
1.3529412	7.41176471	0.2618068	0.15708411	0.0000000	73.17073	4.70588235	79.207921
0.0000000	0.00000000	0.0000000	0.00000000	0.0000000	0.00000	0.00000000	0.000000
3.0882353	15.82720588	2.1999381	0.40739594	2.6470588	63.82979	0.00000000	0.000000
3.7500000	18.01829268	0.6756811	0.27027243	1.9207317	27.77778	0.09146341	16.666667
2.5742574	9.85148515	0.8777164	0.80457336	0.0000000	56.81818	0.99009901	54.054054
4.3228040	22.23712678	1.1507922	0.40277725	0.0000000	60.78431	2.10298572	43.902439
5.3773585	24.46540881	2.6038220	1.32325382	0.0000000	60.99291	2.92452830	57.763975
2.8864865	12.12972973	1.6292314	0.86253427	0.7783784	68.42105	0.77837838	42.105263
2.2826087	10.03623188	1.6059666	0.96357995	1.7391304	59.21053	2.17391304	65.217391
4.7575758	18.03030303	1.6864695	1.15173529	0.7272727	53.92157	0.60606061	58.823529
1.7220217	6.04332130	1.7281824	0.96010132	0.0000000	80.28169	2.59927798	80.808081
0.0000000	0.00000000	0.1641655	0.00000000	0.0000000	33.33333	0.00000000	0.000000
3.9629630	15.66666667	2.1341512	0.60194007	4.3333333	57.77778	0.88888889	61.538462

PressuredLongBalls	UnPressuredlongBalls	Xgbuild	XgContra
0.41450777	0.9326425	0.366273613	1.0733691
0.29112754	1.7467652	0.513076502	1.0520101
0.80555556	6.0555556	0.730501117	0.9911216
0.13086150	2.3882225	0.630291580	0.9454445
0.41176471	4.7647059	0.668951510	1.0933633
0.00000000	4.4000000	0.140187327	1.0482214
0.11029412	0.6617647	0.453463746	0.6409582
0.18292683	2.0121951	0.663681179	0.8614773
0.04950495	1.8316832	0.736750903	0.6369631
0.19472090	0.2336651	0.378291714	0.7329183
0.15723270	0.7861635	0.534838611	1.1012365
0.19459459	1.8162162	0.897090597	0.7136904
0.39855072	2.2826087	0.823234581	0.6959777
0.54545455	3.9393939	0.735609895	1.0064031
0.64981949	5.9458484	0.674524575	0.6915568
0.66666667	4.3333333	0.307476382	0.8558507
0.14814815	1.1481481	0.790979904	0.7906549

Ilustración 5-5. Data frame. Rendimiento.



## 5.4 Pases y combinaciones.

Los pases entre dos jugadores de un mismo equipo es la acción más repetida en todos y cada uno de los partidos. Es la manera más segura de desplazar el balón por el campo obteniendo mejores situaciones de éste y evitando que el rival consiga la posesión. Por eso es fundamental poder analizar estas acciones.

En este paquete incluimos cinco funciones con las que analizar visualmente sobre el campo los patrones de pase de los equipos y jugadores: `SBMAPpases()`, `matrizPases()`, `redPases()`, `sonarJugador()` y `mapaSonar()`.

### 5.4.1 SBMAPpases()

La función `SBMAPpases()` realiza un mapa de pases mediante flechas donde el inicio de estas es el origen del pase y el final el destino. Distingue entre pases **completados**, pases **incompletados**, **asistencias de tiro** y **asistencias de gol**. También calcula la cantidad de pases completados y el porcentaje de acierto.

La función `SBMAPpases()` puede procesar los eventos que desee el usuario. La función procesa los eventos de tipo pase de la información que recibe, sin distinguir equipo. Es conveniente seleccionar los eventos que se quieren estudiar antes de usar esta función.

Argumentos:

- `Events`: Data frame con los datos objetivo de estudio.
- `idJugador`: Integer. Identificador del jugador.
- `campoRival`: Lógico. `TRUE` se filtra los pases en campo rival.

Ejemplo:

```
manCity<-seleccionEventos(eventsLF, idPartido = 2275037, idEquipo = 746, rival = F)
```

```
SBMAPpases(manCity)
```

La *ilustración 5-6* representa el mapa de pases del Manchester City WFC frente al Chelsea WFC. Se han completado un total de 241 pases con un acierto del 82%. La mayoría de los pases se realizan en campo propio.

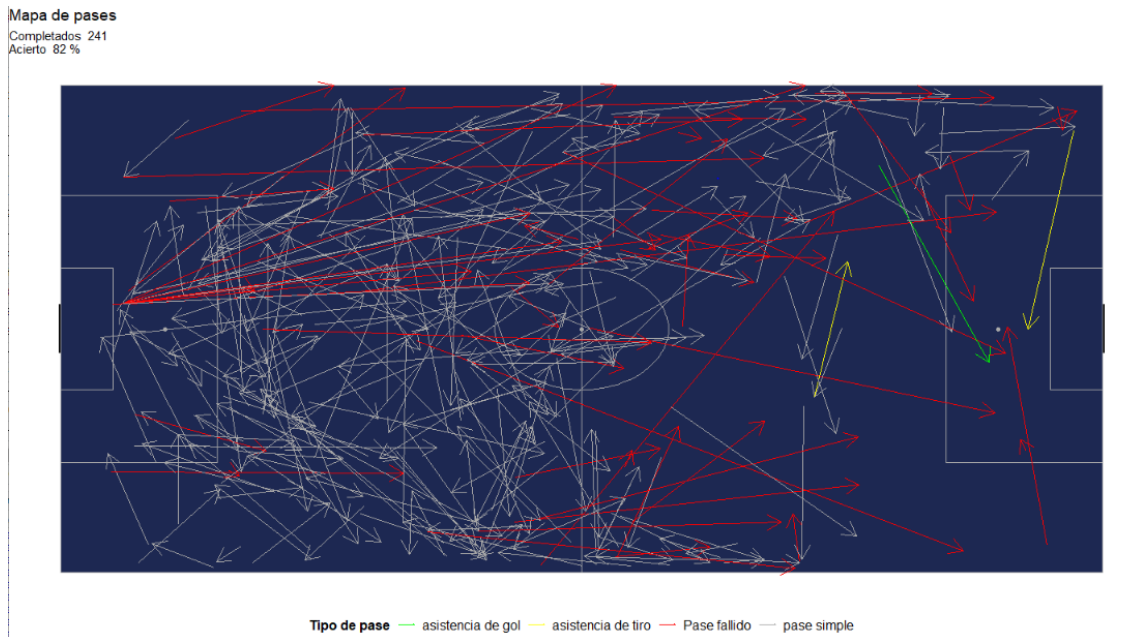


Ilustración 5-6. Mapa de pases Manchester City.

Este tipo de gráficos puede ir acompañado de distintos filtros para facilitar la comprensión del juego.

- Pases en campo rival:

SBMAPpases(manCity, camporival=T)

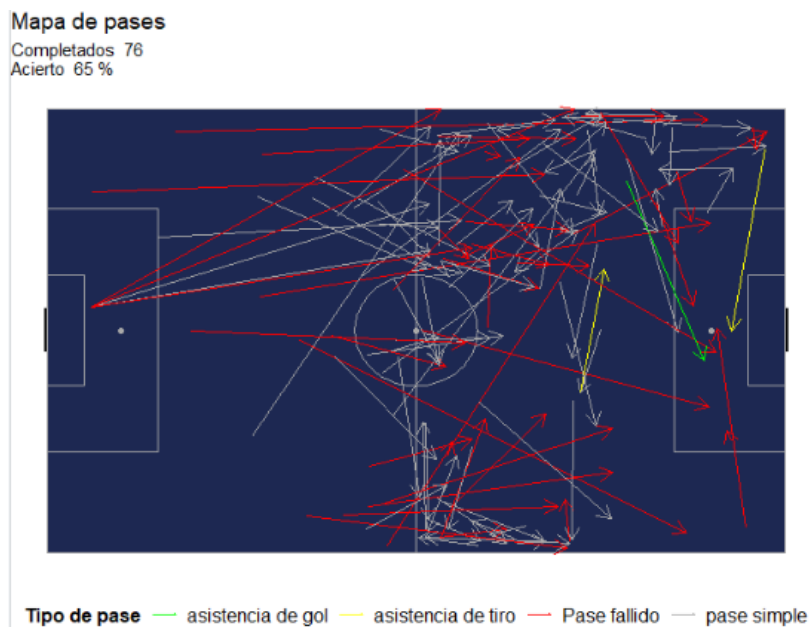
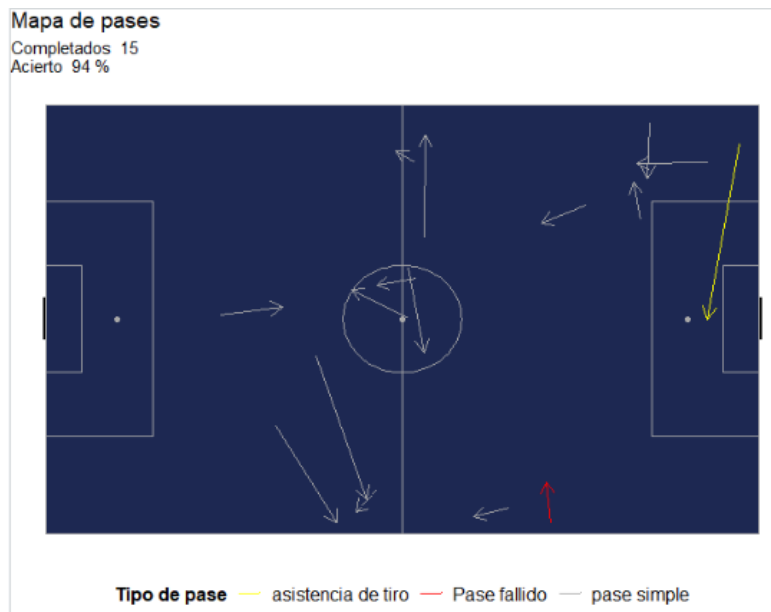


Ilustración 5-7. Pases en campo Rival manchester City.

La *ilustración 5-7* muestra los pases que se han realizado en campo rival. Se han completado 76 de los 241 totales y el acierto ha disminuido hasta el 65%. La banda izquierda tiene mayor circulación de balón que la derecha. Desde esa banda se ha dado un pase de gol y una asistencia de tiro.

- Mapa de pases de un determinado jugador (ver *ilustración 5-8*):

SBMAPpases(manCity, idJugador = 10185)



*Ilustración 5-8. Pases Stephanie Houghton.*

## 5.4.2 matrizPases()

Una parte muy importante en el análisis de pases es la relación entre los jugadores. La frecuencia con la que dos jugadores conectan mediante pases es una métrica básica para analizar el juego de un equipo. Por ejemplo, si un jugador X tiene una frecuencia acumulada de más de 300 pases con el jugador Y y una frecuencia acumulada de pases de 14 con el jugador Z. No implica que sea más efectivo prestar atención al jugador Y que al Z, cuando el jugador X posea el balón.

La función `matrizPases()` crea una matriz simétrica de puntos. El color y el tamaño de los puntos representan las frecuencias absolutas de pases entre todos los jugadores de los eventos objetivo de estudio. Gracias a la librería `plotly()` conseguimos interactividad sobre este gráfico. Se muestran con más claridad los valores de la matriz situando el cursor del ratón encima de cada punto. La matriz no diferencia entre ejecutor y destinatario. Si el jugador X realiza 12 pases al jugador Y y el jugador Y realiza 30 pases al jugador X, tanto el valor XY como el valor YX de la matriz será 42.

Argumentos:

- `events`: Dataframe con los datos.

Ejemplo:

Matriz de pases del Chelsea WFC las temporadas 2018/2019 y 2019/2020 en la FA Women's Superleague. (Ver *ilustración 5-9*)

```
eventos<-seleccionEventos(eventsLF, idEquipo = 967,rival = F)
```

```
matrizPases(eventos)
```

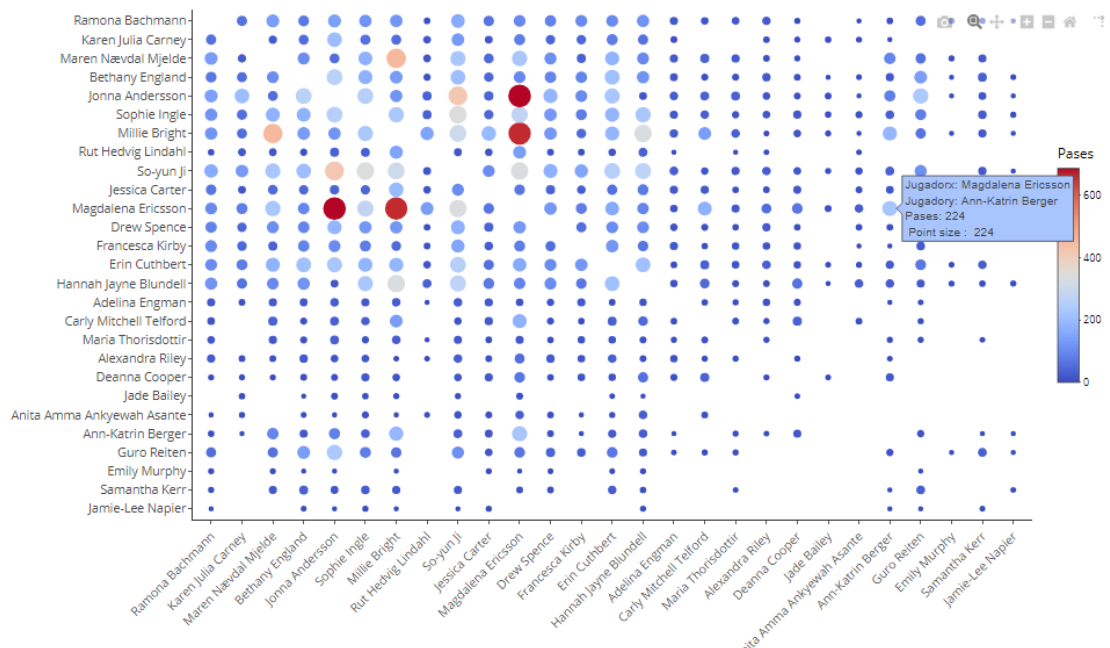


Ilustración 5-9. Matriz de pases. Chelsea.

### 5.4.3 redPases()

El análisis táctico de un equipo en ataque reside en tres aspectos: cómo se asocian sus jugadores, la importancia de cada jugador durante la posesión del balón y la situación de cada jugador en el campo durante el ataque del equipo. Estos tres elementos se resumen mediante un gráfico **red de pases**. Una *red de pases* es un **grafo**. Los jugadores son los vértices del grafo y los pases entre los jugadores las aristas.

Una *red de pases* se puede realizar de muchas formas. Su fin es representar la relación de pases entre los jugadores. También se puede incluir más información.

La función **redPases()** del paquete GASB devuelve una red de pases representada en un campo de fútbol. La posición de los distintos jugadores (vértices) es la posición media de todas las posiciones donde han recibido o han realizado un pase durante el partido. Los pases (las aristas) son representadas mediante un segmento entre los jugadores. Se diferencia entre ejecutor y destinatario, es un **grafo dirigido**.

Información del grafo:

- Una arista XY comienza con un punto al lado de vértice X, señalando que esa arista es XY y no YX.
- El **tamaño** de los vértices y el de los puntos al comienzo de las aristas indican la **cantidad de pases** que representan.
- Si el vértice X es más grande que el vértice Y porque X ha completado más pases que Y.
- Si el punto de la arista XY es más grande que el punto de la arista XZ porque X ha realizado más pases hacia Y que hacia Z.
- El **color** de las aristas y los vértices representa el **expected Goal**.
- El color de un **vértice** X representa el **expected Goal generado** por el jugador X. Si X ha tirado a portería generando un expected goal de 0.9, su color es cálido por su alto expected Goal generado.
- El color de la **arista** YX representa el **expected Goal asistido** de Y a X.
- Si el jugador Y asiste un disparo con 0.4 de expected goal al jugador X. La arista YX es de un color más cálido.

El grafo tiene interactividad gracias al paquete *plotly()* situando el cursor encima de los vértices y los puntos de las aristas visualizamos la cantidad exacta de pases que representa y su expected Goal.

La función **redPases()** tiene el atributo **minPases** con el que indicamos la cantidad de pases mínima que debe tener una arista para que sea representada. Así se consigue un grafo más limpio representando sólo las aristas relevantes. Aunque una arista sea más pequeña que la cantidad mínima de pases requerida pero representa un expected Goal mayor que 0, esta arista también se representa.

Argumentos:

- *Temp*: Dataframe con los datos de un partido.
- *Home*: lógico. *TRUE* si seleccionamos el equipo de casa.
- *Minpases*: Integer. Pases mínimos para las aristas.
- 

Ejemplo:

Grafo de Pases del West Ham United LFC

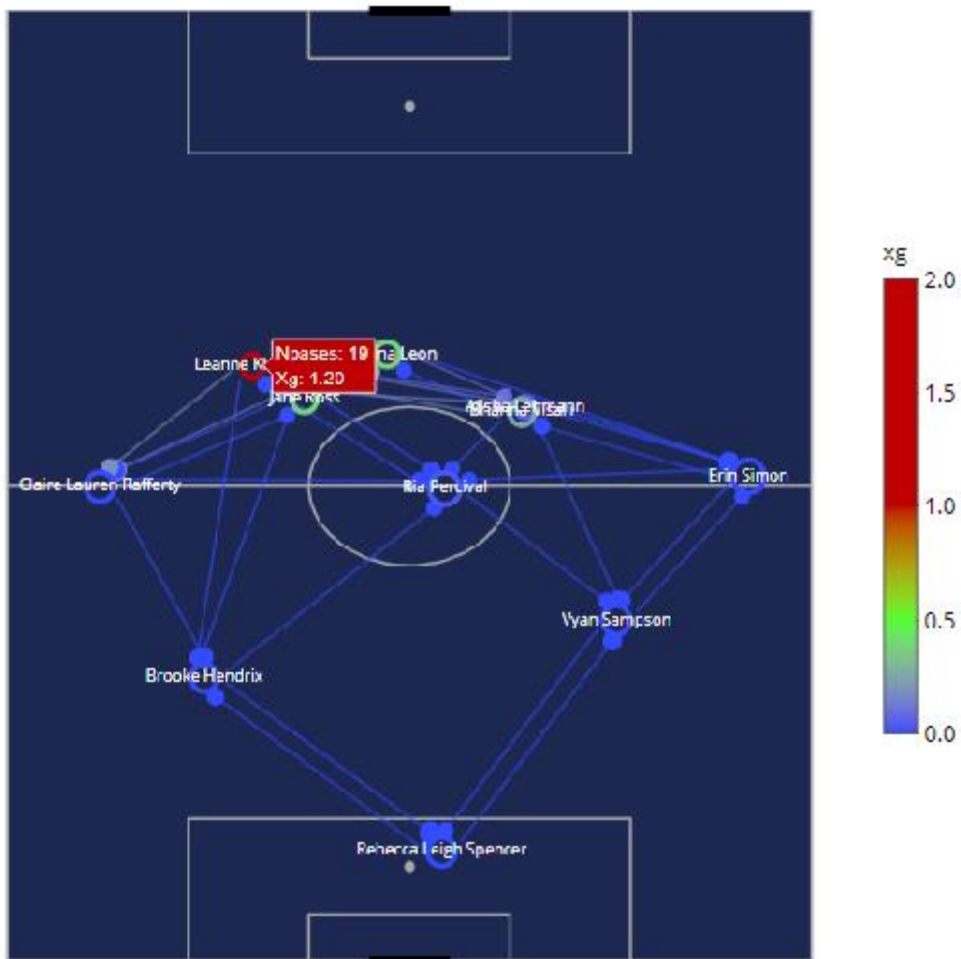


Ilustración 5-10. Red de Pases West Ham United.

La *ilustración 5-10* representa la red de pases de un partido del *West Ham United LFC*. Se aprecia que la jugadora *Leanne Kiernan* ha realizado 19 pases y ha generado un expected goal de 1.2.

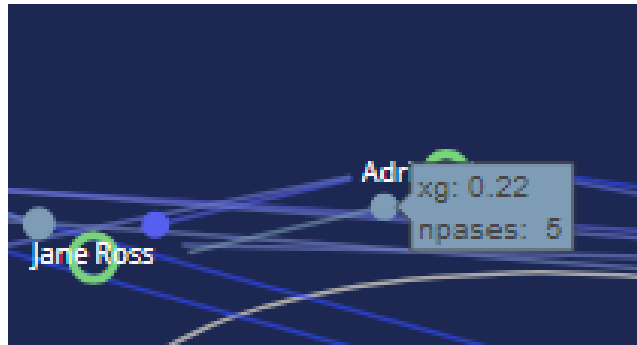


Ilustración 5-11. Zoom ilustración 5-10.

Siendo  $\text{minPases} = 6$  en la *ilustración 5-11* se representa la arista Adriana Leon-Jane Ross por tener un expected goal asistido de 0.22.

#### 5.4.4 sonarJugador()

Los gráficos **sonar de pases**. Estos gráficos sirven para detallar las tendencias individuales de los pases de cada jugador. En un gráfico sonar se resume la **frecuencia absoluta** de los pases en las distintas direcciones y la **longitud media** de los pases en cada dirección.

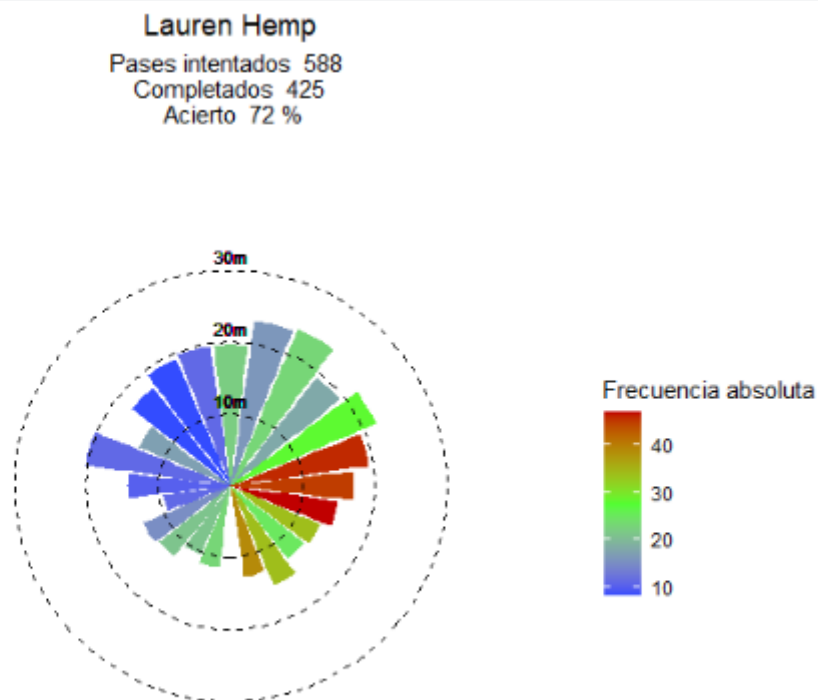
La función **sonarJugador()** realiza un gráfico sonar de pases. Distingue entre 24 direcciones. Cada dirección abarca un ángulo de 15 grados. La **frecuencia absoluta** de pases se representa mediante el **color** de las barras. Cuánto más cálido sea el color, mayor frecuencia de pases tendrá esa dirección. La **longitud media** de los pases se representa mediante las **longitudes de las barras**.

Ejemplo:

Sonar de pases de Lauren Hemp las temporadas 2018/2019 y 2019/2020 en la FA Women's Superleague. (Ver *ilustración 5-12*)

```
temp<-seleccionEventos(eventsLF,idJugador = 15555)
```

```
sonarJugador(temp)
```



*Ilustración 5-12. Sonar de pases de Lauren Hemp.*



#### 5.4.5 mapaSonar()

La función **mapaSonar()** crea un mapa de los distintos gráficos sonar de pases de los jugadores de un equipo en un partido. La posición donde se sitúa cada gráfico es la posición media donde han realizado o recibido los pases los jugadores.

Argumentos:

- *events*: Dataframe con los datos de un partido.
- *Home*: Lógico. *TRUE* si seleccionamos el equipo que juega en casa. Por defecto *TRUE*.

Ejemplo:

Mapa de los sonar de pases de los distintos jugadores de un partido del Reading WFC.(Ver ilustración 5-13)

```
partido<-seleccionEventos(eventsLF, idPartido = 19753)
```

```
mapaSonar(partido)
```

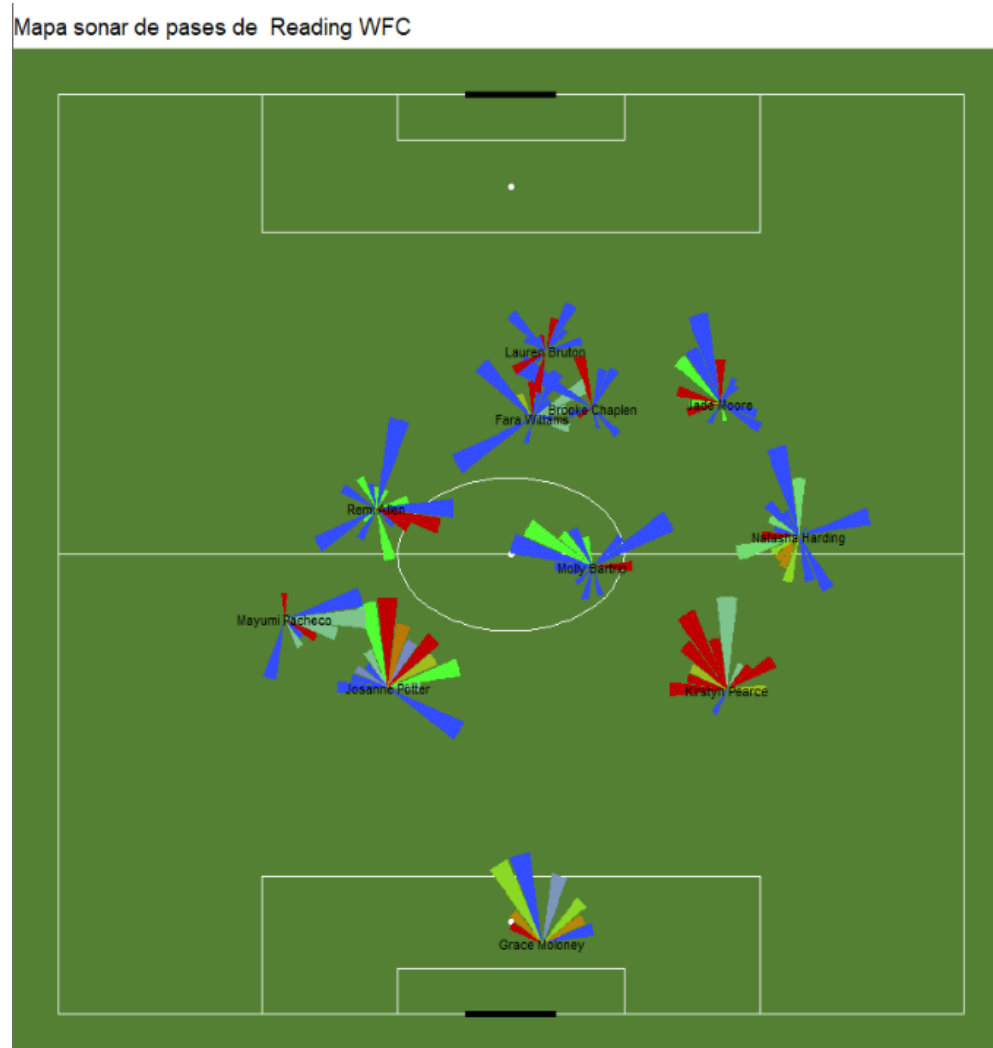


Ilustración 5-13. Mapa sonar. Reading WFC.

## 5.5 Análisis defensivo

Este apartado está dedicado al análisis de las acciones defensivas. El análisis defensivo en el fútbol está marcado por dos variables: Dónde se realizan las acciones defensivas y con qué frecuencia.

Las distintas acciones que son consideradas de carácter defensivo son las siguientes:

- **Recuperación:** Un jugador roba la pelota a un jugador rival.
- **Duelo:** Un jugador consigue la posesión del balón en una disputa con un rival en una situación en la que el balón no se encuentra en la posesión de ningún equipo.
- **Bloqueo:** Un jugador impide que un pase o un disparo llegue a su destino sin conseguir la posesión del balón interponiéndose en la trayectoria.
- **Despeje:** Un jugador manda el balón a situaciones del campo menos peligrosas en una jugada de ataque del equipo rival.
- **Intercepción:** Un jugador roba el balón interponiéndose en la trayectoria de un pase del equipo rival.
- **Falta:** Un jugador detiene la posesión del equipo rival mediante un infracción.
- **Presión:** Un jugador provoca que un jugador rival de un pase forzado al acercarse a él.

El paquete *GASB* incluye dos funciones relacionadas con el análisis defensivo:  *analisisDefensivoEquipo()* y  *analisisDefensivoJugador()*. Las dos funciones se basan en el mismo concepto. Se diferencian en detalles por lo que se describirán de forma conjunta.

### 5.5.1 analisisDefensivoEquipo() / analisisDefensivoJugador()

La mejor forma de realizar un análisis defensivo es mediante un **mapa de calor**. El dicho mapa de calor indica en que lugares del campo se realizan las acciones defensivas y con qué frecuencia. Se distinguen **30 zonas** del campo formadas gracias a la división del campo en 5 carriles verticales y 6 sectores horizontales.

Se usan distintos ajustes de las acciones defensivas para los equipos y para los jugadores.

$$\text{Acciones defensivas equipos} = \frac{\text{Acciones Defensivas}}{\text{pases que realiza el rival}}$$

Se distinguen los sectores tanto en las acciones defensivas como en los pases del rival. Así se mide la agresividad de forma más ajustada.

El ajuste realizado para los jugadores es el ya mencionado en el *Apartado 4.3: Ajuste por posesión para acciones defensivas* de este trabajo.

$$\text{Acciones defensivas jugador} = \text{Acciones defensivas} * \frac{2}{1 + e^{-0.1 * (\text{posesion} - 50)}}$$

Para medir la frecuencia de las acciones en cada sector se necesita un **contexto**. Los equipos y los jugadores se comparan con otros equipos y otros jugadores. Se toma de referencia el **mínimo**, la **media** y el **máximo** de acciones defensivas realizadas en cada sector por el contexto. En el mapa de calor los colores oscuros indican que se realizan menos acciones defensivas que la media siendo negro el mínimo. Los colores cálidos indican que se realizan más acciones defensivas que la media siendo rojo intenso el máximo.

Las acciones se reescalan a partir del mínimo, la media y el máximo en cada sector de la siguiente manera:

Para *acciones defensivas ajustadas mayores que la media*

$$\text{acciones defensivas ajustadas} = \frac{\text{acciones defensivas ajustadas} - \text{media}}{\text{Maximo} - \text{media}}$$

Para *acciones defensivas ajustadas menores que la media*

$$\text{acciones defensivas ajustadas} = \frac{\text{acciones defensivas ajustadas} - \text{media}}{\text{media} - \text{Minimo}}$$

Los valores de referencia tras estos cálculos quedan de la siguiente manera:

- mínimo = -1
- media = 0
- máximo = 1.

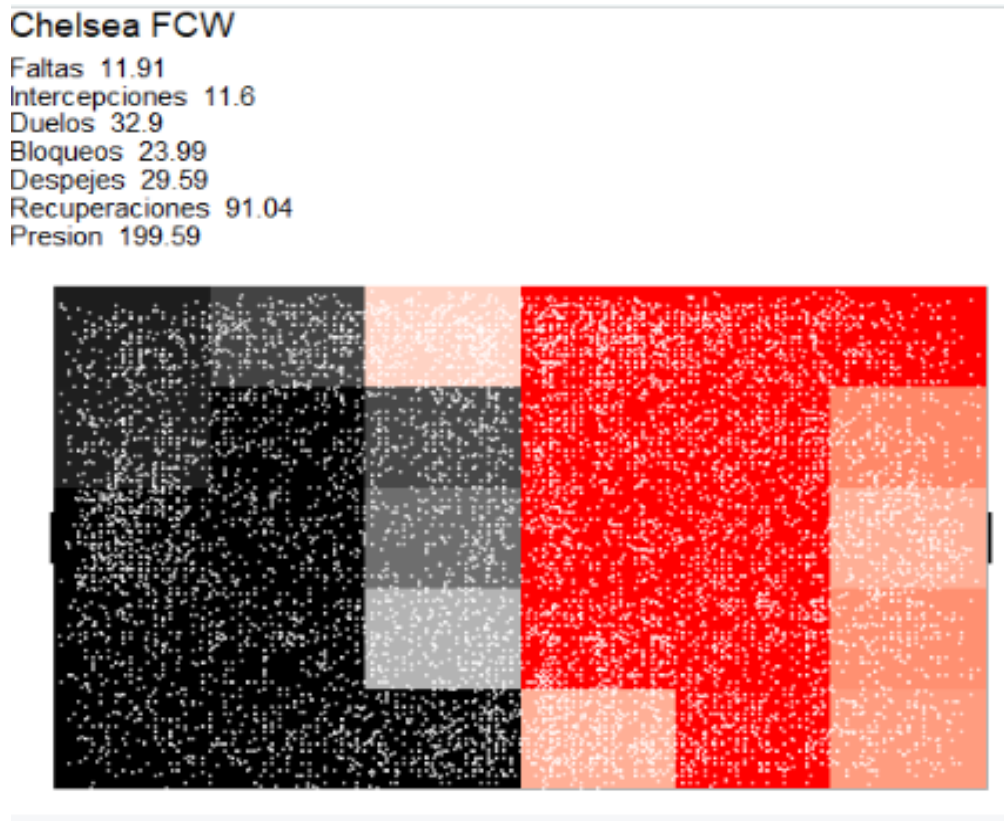
Argumentos de la función ***análisisDefensivoEquipo()***:

- *eventos*: Dataframe. Marco en el que se incluyen los eventos del equipo objetivo y los de su contexto.
- *idEquipo*: Integer. Identificador del equipo objetivo.
- *Presion*: Lógico. *FALSE* si se tienen en cuenta todas las acciones defensivas. *TRUE* si se tienen en cuenta solo las acciones de presión.

Ejemplo:

Mapa de calor de análisis defensivo del Chelsea WFC con identificador 971.(Ver *ilustración 5-14*)

analisisDefensivoEquipo(eventsLF, idEquipo = 971)



*Ilustración 5-14. Mapa defensivo ChelseaWFC*

Con la función *análisisDefensivoEquipo()* se ve de manera muy clara los distintos perfiles defensivos. Realizando el análisis defensivo que nos proporciona la función *análisisDefensivo()* en los distintos equipos de la FA women's Superleague apreciamos distintos perfiles defensivo a nivel de equipo.

- Bloque alto:

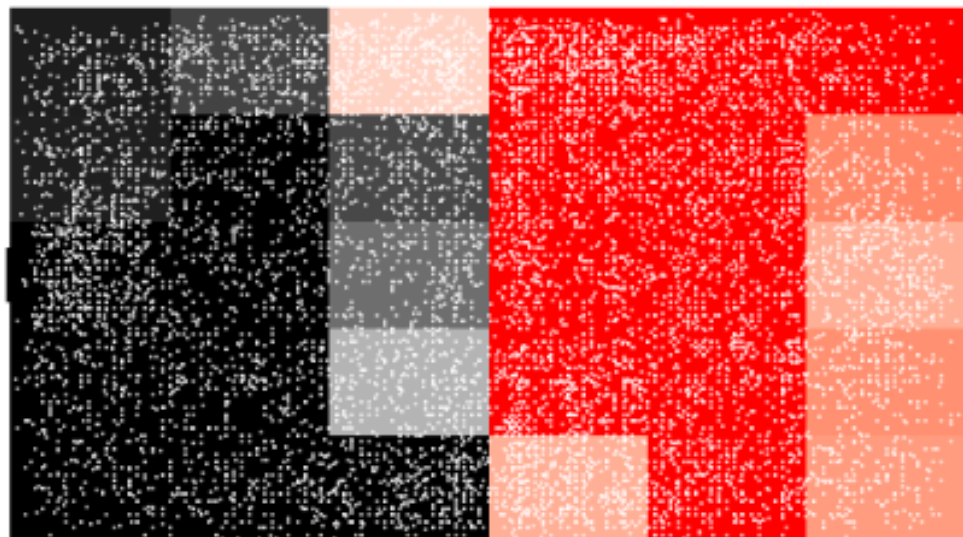
El Chelsea realiza acciones defensivas por encima de la media en campo contrario. En campo propio su intensidad defensiva es menor que la media de su liga.(ver *ilustración 5-15*)

`análisisDefensivoEquipo(eventsLF, idEquipo = 971)`

---

### Chelsea FCW

Faltas 11.91  
Intercepciones 11.6  
Duelos 32.9  
Bloqueos 23.99  
Despejes 29.59  
Recuperaciones 91.04  
Presion 199.59



*Ilustración 5-15. Ejemplo bloque alto.*

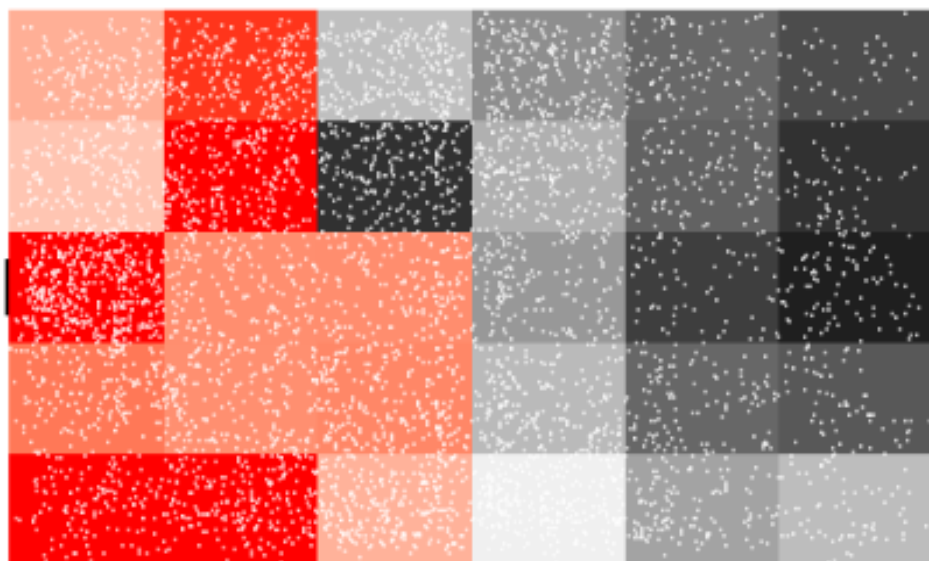
- Bloque bajo:

El Tottenham Hotspur Women (identificador 749) realiza más acciones defensivas que la media en su propio campo. En campo contrario su intensidad defensiva disminuye. (ver *ilustración 5-16*)

analisisDefensivoEquipo(eventsLF, idEquipo = 749)

### Tottenham Hotspur Women

Faltas 11.49  
Intercepciones 8.78  
Duelos 25.32  
Bloqueos 18.55  
Despejes 30.61  
Recuperaciones 50.98  
Presion 143.44



*Ilustración 5-16. Ejemplo bloque bajo.*

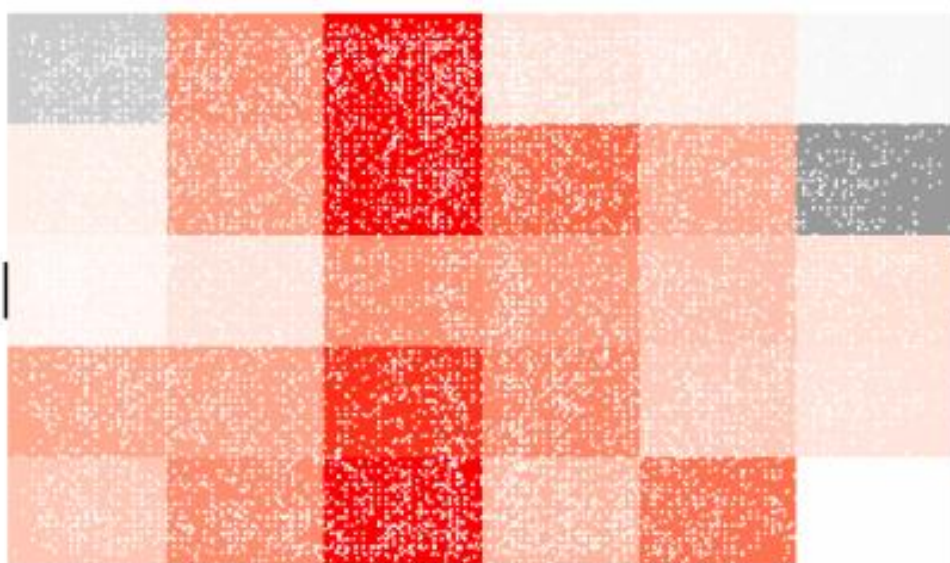
- Alta intensidad defensiva/ bloque medio:

El Reading WFC (identificador 974) realiza más acciones defensivas que la media en casi la totalidad del campo, destacando en el medio campo.(ver *ilustración 5-17*)

analisisDefensivoEquipo(eventsLF, idEquipo = 974)

### Reading WFC

Faltas 14.65  
Intercepciones 10.25  
Duelos 38.2  
Bloqueos 23.94  
Despejes 37.66  
Recuperaciones 92.04  
Presion 183.39



*Ilustración 5-17. Ejemplo bloque medio.*

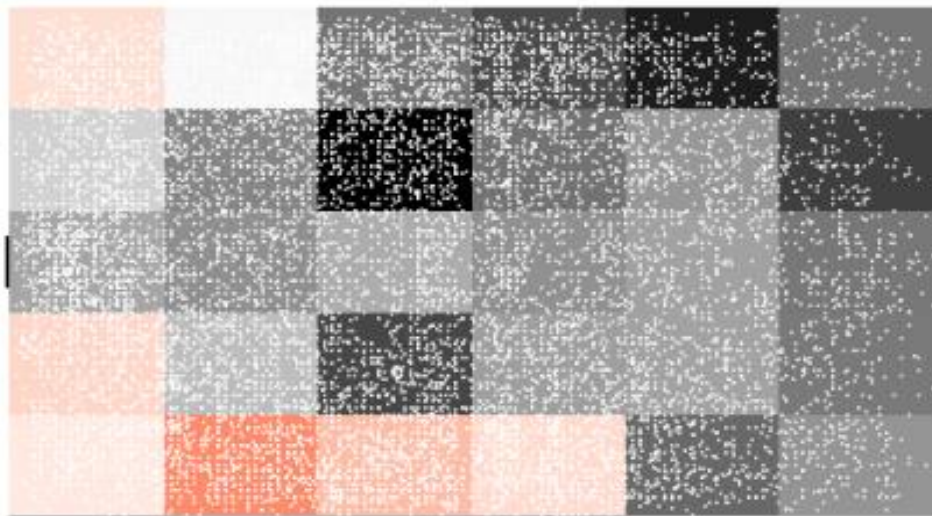
- Baja intensidad defensiva:

El Brighton & Hove Albion WFC (identificador 965) realiza menos acciones defensivas que la media en todo el campo. ( Ver *ilustración 5-18*)

`analisisDefensivoEquipo(eventsLF, IdEquipo = 965)`

### Brighton & Hove Albion WFC

Faltas 6.18  
Intercepciones 7.13  
Duelos 14.16  
Bloqueos 11  
Despejes 14.54  
Recuperaciones 33.11  
Presion 101.2



*Ilustración 5-18. Ejemplo baja intensidad defensiva.*



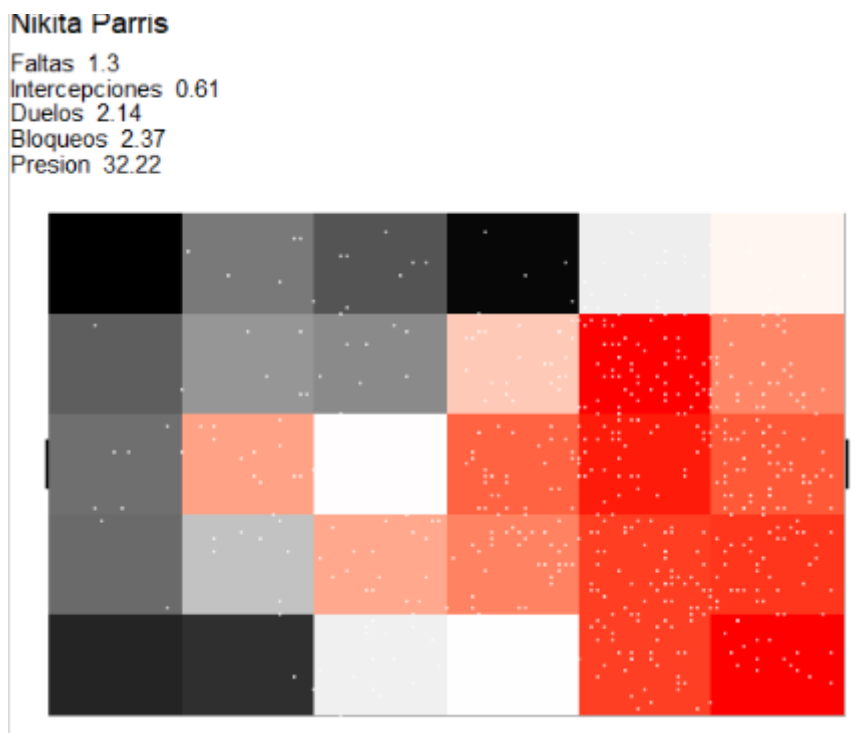
Argumentos de la función **analisisDefensivoJugador()**:

- *eventos*: Data frame. Marco en el que se incluyen los eventos del jugador objetivo y los de su contexto.
- *idJugador*: Integer. Identificador del jugador objetivo.
- *Posiciones*: Vector. Posiciones con las que se quiere comparar a nuestro jugador. Ver *ilustración 3-2*.
- *Presion*: Lógico. *FALSE* si se tienen en cuenta todas las acciones defensivas. *TRUE* si se tienen en cuenta solo las acciones de presión.

Ejemplo:

Mapa de calor de la jugadora Nikita Parris, con identificador 4654, comparada con las jugadoras que juegan de delanteras (posición 23) al igual que ella. (Ver *ilustración 5-19*)

`analisisDefensivoJugador(eventsLF, idJugador = 4654, posiciones = c(23))`



*Ilustración 5-19. Mapa defensivo Nikita Parris.*

## 5.6 Análisis ofensivo.

El análisis ofensivo de un equipo o un jugador se resume en: Disparos y asistencias. Gracias a mapas se puede ver las tendencias de disparos y asistencias de un equipo o un jugador. Este apartado tiene cuatro funciones: mapa de tiro, mapa de asistencias y dos mapas que representan la situación del juego en el momento del disparo.

### 5.6.1 XGMap()

La función **XGMap()** crea un mapa de disparos. En este mapa se muestra la distribución de los disparos en el campo, el tipo de disparo, su resultado y su expected goal.

La posición de cada disparo se indica con la posición del punto en el campo.

El resultado de un disparo se refleja con tres posibilidades:

- Borde negro: el disparo fue gol.
- Color nítido: el disparo fue dentro de la portería.
- Color poco nítido: el disparo fue fuera de la portería.

Hay cuatro tipos de disparo:

- Cabeza.
- Lanzamiento de falta.
- Precedido de un pase filtrado.
- Pie.

Gracias a la función *plotly()* se puede filtrar los disparos del mapa de cada tipo directamente sobre el gráfico haciendo doble click en el punto de la legenda obtenemos un mapa en el que se incluyen solo los disparos del tipo que se desea.

El color de los puntos indica el expected goal de los disparos. Cuanto más frío sea el color, menor expected goal tendrá el disparo. Si el color es cálido el disparo tiene un alto expected goal. Como es una escala continua el expected goal exacto de cada disparo se obtiene poniendo el cursor encima de cada punto gracias a la interactividad introducida con *plotly()*.

Se pueden obtener mapas de tiro de un único equipo o de un jugador en particular.

La función **XGMap()** además hace un recuento del expected goal generado, los goles conseguidos y la diferencia entre ellos u lo muestra en el gráfico.

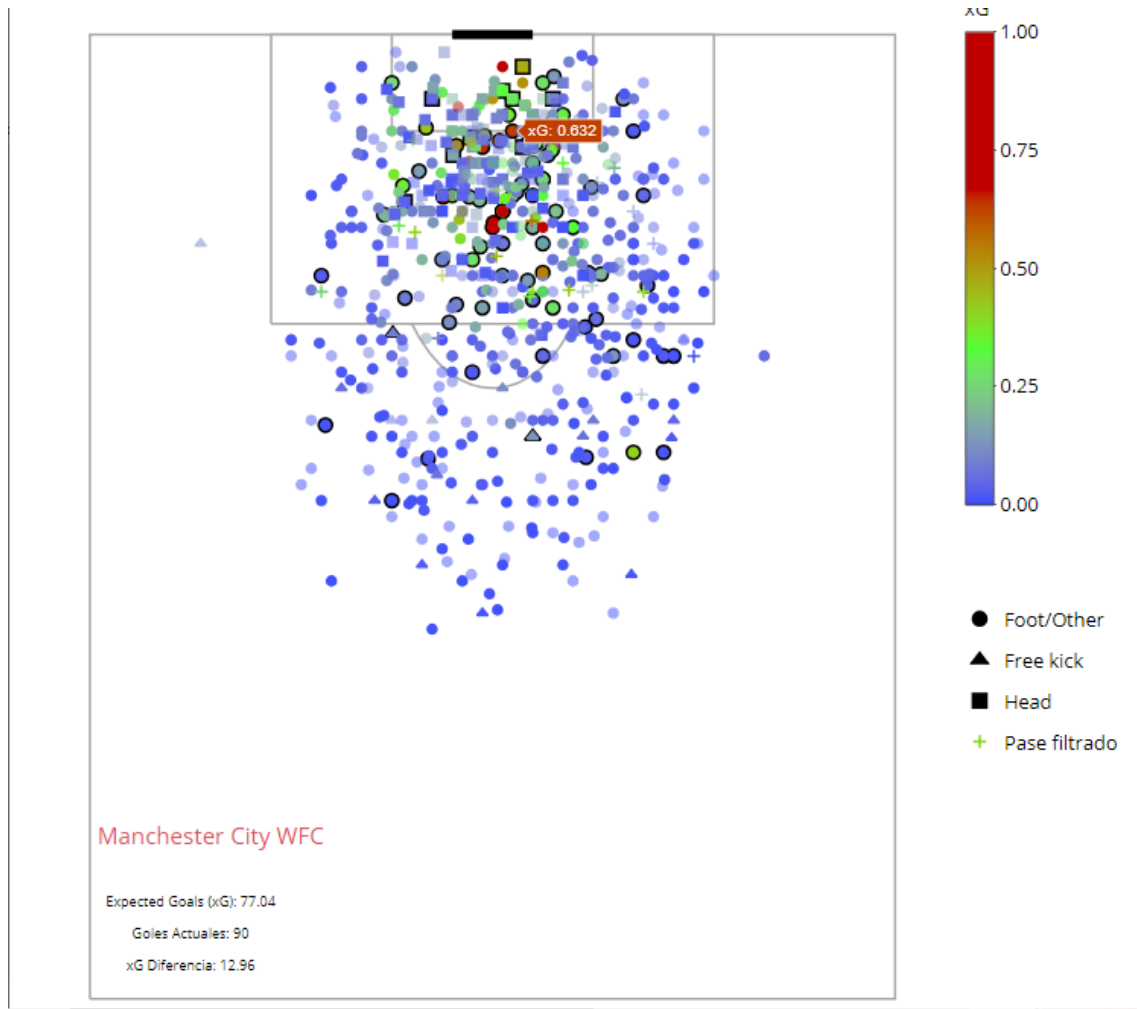
Argumentos:

- *Events*: Datos objetivo de estudio.
- *idEquipo*: Identificador del equipo a estudiar.
- *idJugador*: Identificador del jugador a estudiar.

Ejemplo:

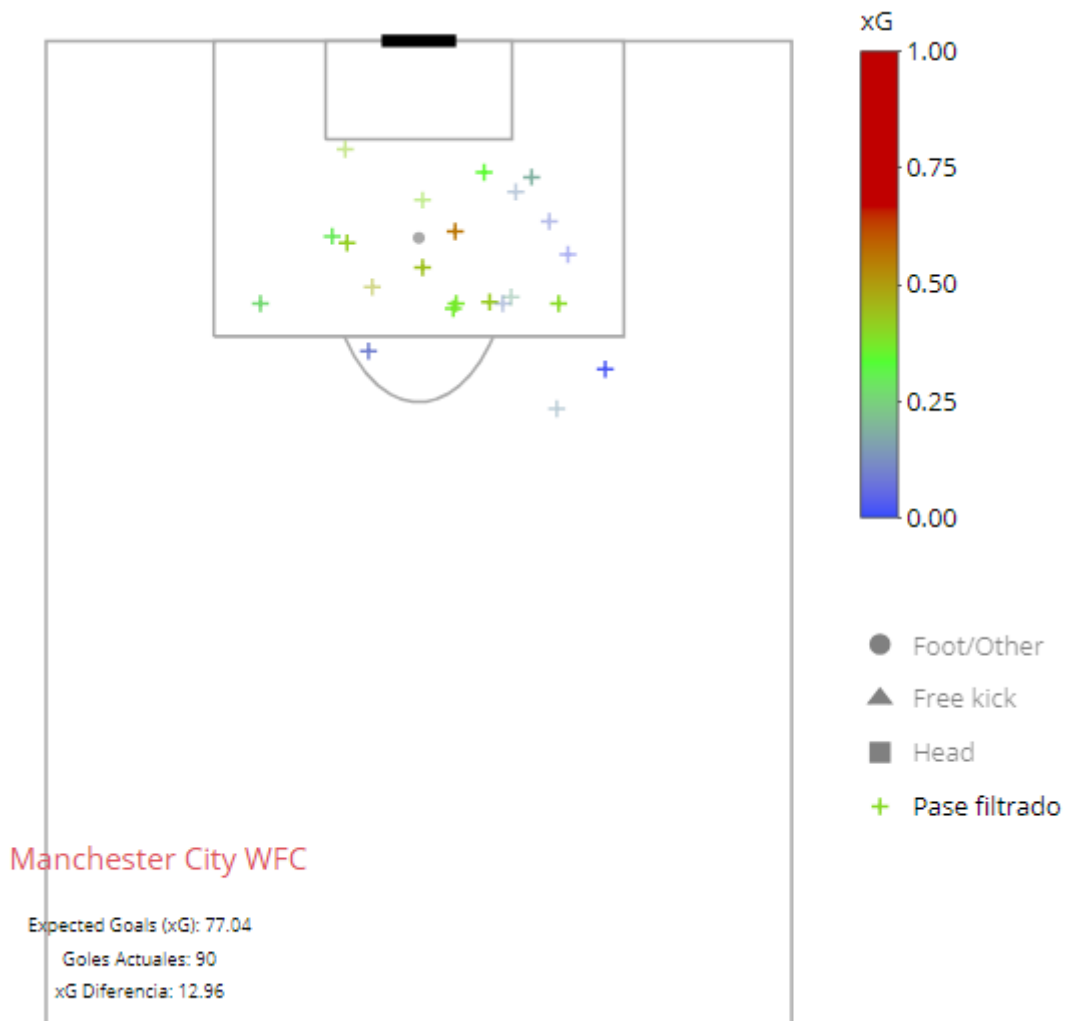
Mapa de tiros del Manchester City (identificador 746). El expected goal de los tiros aumenta según se acerca a la portería. A primera vista, el ángulo respecto a la portería también afecta al expected goal. (Ver *ilustración 5-20*)

`xGMap(eventsLF, idEquipo=746)`



*Ilustración 5-20. Mapa de tiros del Manchester City.*

En la *ilustración 5-21* se representan únicamente los disparos precedidos de un pase filtrado. Aunque ninguno haya resultado gol, su expected goal es más elevado que la media de disparos. (Ver *ilustración 5-21*)



*Ilustración 5-21. Mapa de tiros de pase filtrado de Manchester City.*

## 5.6.2 assistMap()

La función **assistMap()** dibuja un mapa de asistencias de tiro y asistencias de gol. Puede usarse con equipos enteros o jugadores particulares. Los pases se representan mediante flechas. El inicio es el origen del pase y el final, indicado con una “x”, el destino del pase.

Se distinguen 4 tipos de pase:

- Centros.
- Pase filtrado.
- Pase de la muerte (cut back).
- Pase simple.

El color de las flechas representa el **expected goal asistido**. El expected goal asistido de un pase es el expected goal del disparo al que asiste.

Gracias a la función *plotly()* al colocar el cursor en el inicio de cada flecha nos da la siguiente información:

- Coordenadas origen.
- Coordenadas destino.
- Tipo de pase.
- Expected goal asistido.
- Resultado del disparo. Gol/No Gol.

Haciendo doble click en los distintos segmentos de la leyenda se filtra el tipo de pase que se quiere visualizar y su resultado (Gol/No Gol).

En la función *assistMap()* se le da la opción al usuario de acotar las zonas de estudio indicado del inicio del pase mediante ymin, ymax, xmin y xmax. Es decir si el usuario solo quiere trabajar con las asistencias producidas en la banda izquierda deberá introducir “ymax = 18”.

Además esta función realiza y representa un resumen con la siguiente información: Expected Goal generado, Goles, diferencia entre expected goal y goles, asistencias totales en la zona y porcentaje “asistencias zona/asistencias totales”.

Gracias a esta función se puede realizar un análisis concluyente de las tendencias de ataque de un equipo, en este caso Manchester City WFC. En la *tabla 7* se plasman los datos de las *ilustraciones 5-22 y 5-23*. Se ataca el doble por la banda derecha que por la izquierda, 15.42% frente a un 28.54%. Una de cada tres asistencias de tiro se realizan desde la banda derecha.

	Banda izquierda (ymax=18)	Banda derecha (ymin=62)
Expected Goal	5.96	14.6
Goles	6	16
Diferencia	0.04	1.4
Asistencias zona	74	137
Porcentaje	15.42%	28.54%

Tabla 12. Análisis de asistencias. Manchester City WFC.

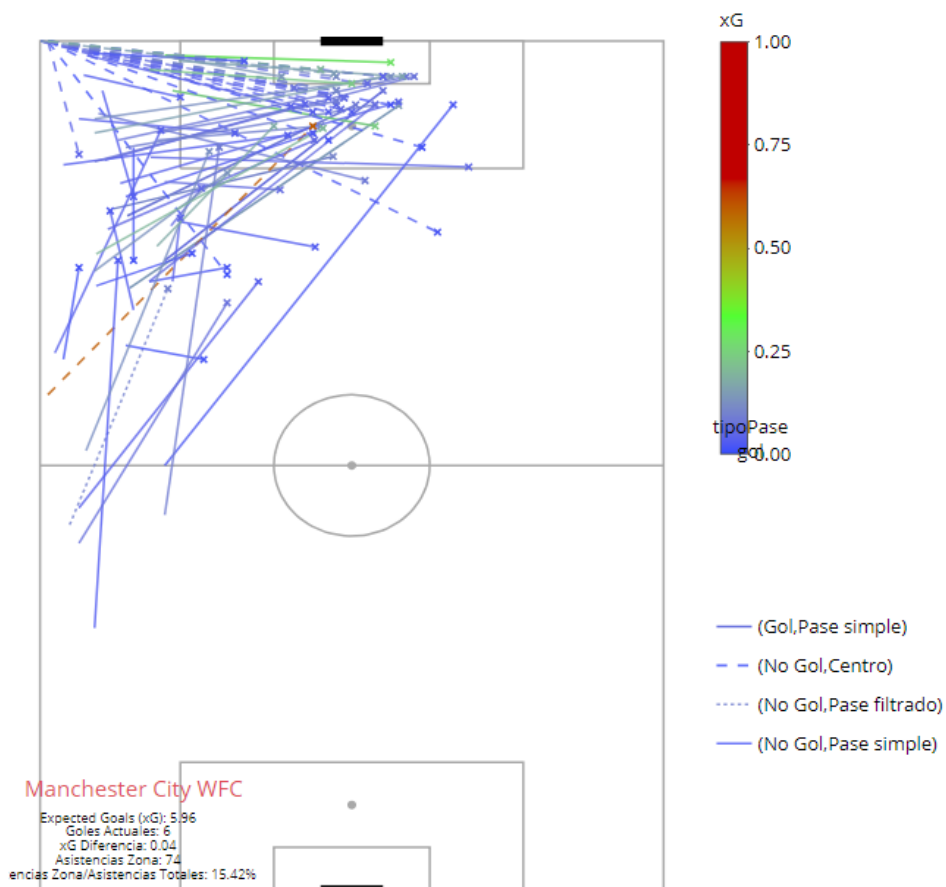
Argumentos:

- Events: Datos objetivo de estudio.
- idEquipo: Integer. Identificador del equipo.
- idJugador: Integer. Identificador del jugador.
- Ymin: Coordenada mínima, ancho del campo. -10 por defecto.
- Ymax: Coordenada máxima, ancho del campo. 90 por defecto.
- Xmin: Coordenada mínima, largo del campo. -10 por defecto.
- Xmax: Coordenada máxima, largo del campo. 130 por defecto.

Ejemplo:

La gráfica de la *ilustración 5-22* se corresponde con las asistencias realizadas por el Manchester City desde la banda izquierda.

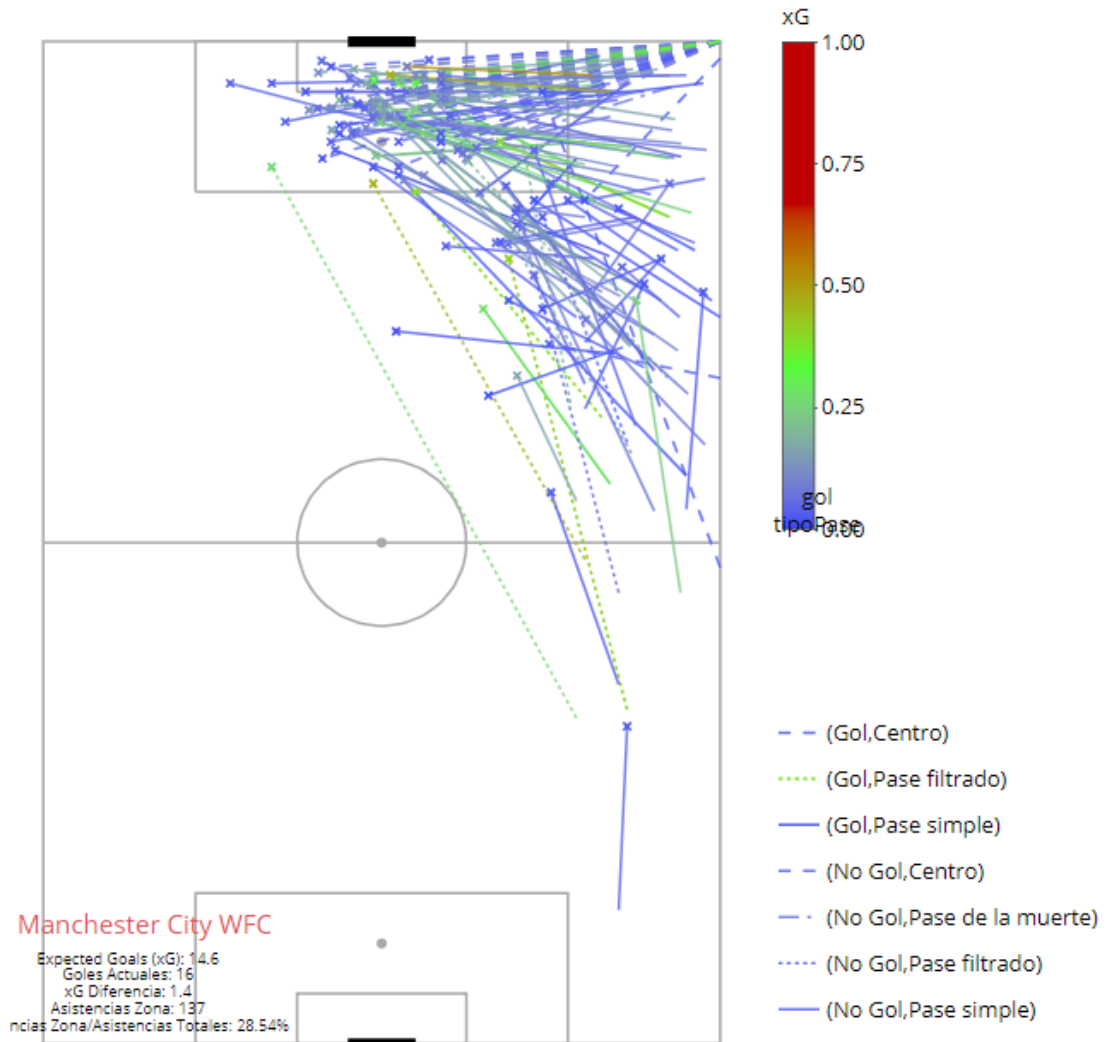
`assistMap(eventsLF, idEquipo = 746, ymax=18)`



*Ilustración 5-22. Mapa asistencias banda izquierda Manchester City.*

De la misma manera la *ilustración 5-23* representa las asistencias realizadas por el Manchester City desde la banda derecha.

`assistMap(eventsLF, idEquipo=746, ymin=62)`



*Ilustración 5-23. Mapa asistencias banda derecha Manchester City.*

### 5.6.3 freezeFrame()

La función **freezeFrame()** utiliza la variable **shot.freezeFrame** para crear un mapa con la posición de los jugadores cercanos al balón en el momento que se realiza un disparo.

En el mapa se distinguen cuatro tipos de jugadores:

- Defensas.
- Atacantes.
- Portero.
- Lanzador.

freezeFrame() es una herramienta muy útil. Podemos ver si el portero tiene buena visión, si hay defensores incomodando al lanzador, si hay atacantes en mejor posición que el lanzador, etc. También en *freezeFrame()* se incluye la situación del portero en el momento del disparo.

- Tumbado.
- En movimiento.
- Parado.

Y la técnica usada para atajar el disparo.

- Vuelo.
- En el sitio.

Respecto al lanzamiento, se incluye:

- Expected Goal.
- Parte del cuerpo.
- Resultado del disparo.

Argumentos de la función **freezeFrame()**:

- *Events*: Datos con la información.
- *idTiro*: Identificador del disparo.
- *idGoalKeeper*: Identificador del evento con la información del portero.



Ejemplo:

La *ilustración 5-24* es la situación de un disparo de Kim Little. El disparo se realiza con la pierna izquierda. El portero emplea la técnica de vuelo para evitar el gol. La jugadora que realiza el disparo no tiene rivales delante de ella. El disparo no acaba en gol.

```
idGoalKeeper1<-"5c540134-e4dc-4986-bedb-9600756d17df"
```

```
idTiro1<-events[which(events$id==idGoalKeeper)-1,]
```

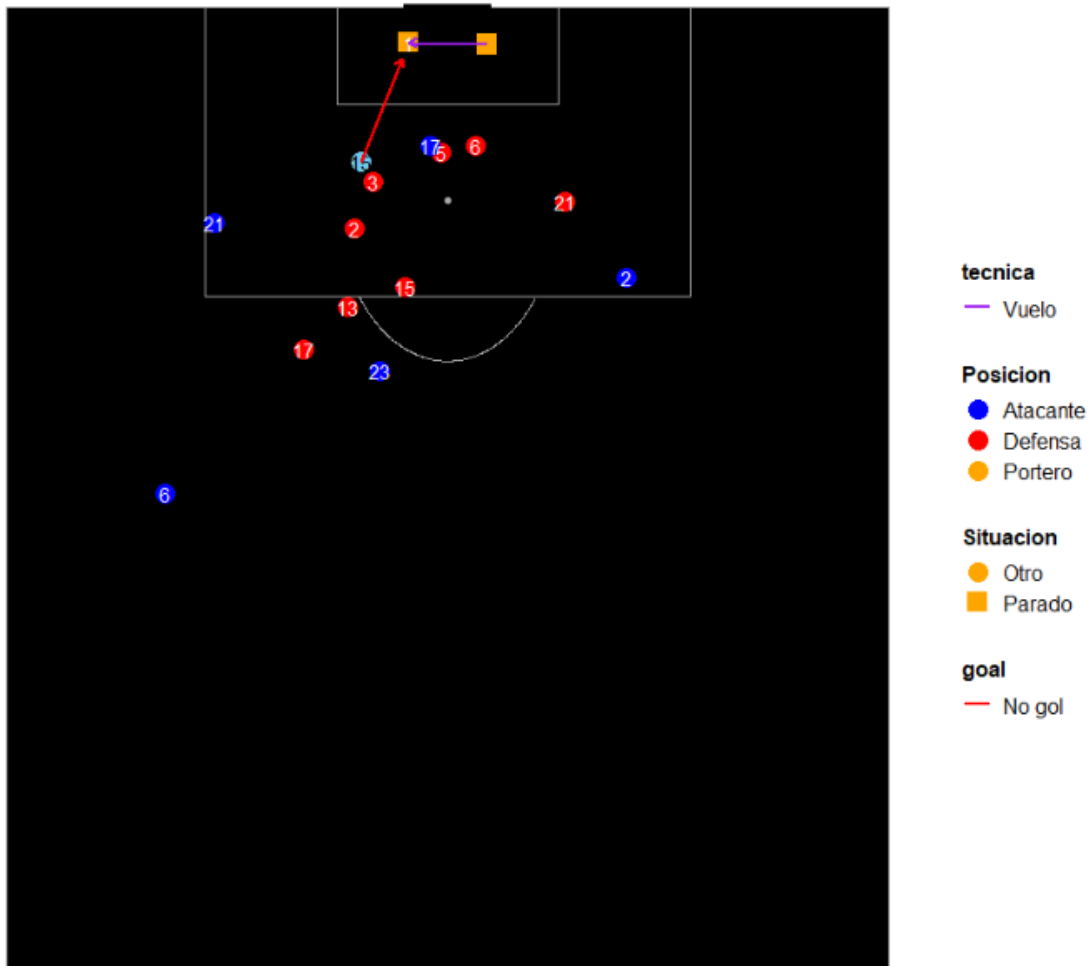
```
idTiro<-idTiro$id
```

```
freezeFrame(eventsLF,idTiro,idGoalKeeper)
```

Kim Little

Expected goal 0.21

Parte del cuerpo: Left Foot



*Ilustración 5-24. Freeze frame de un disparo de Kim Little.*

#### 5.6.4 freezeFrame3d()

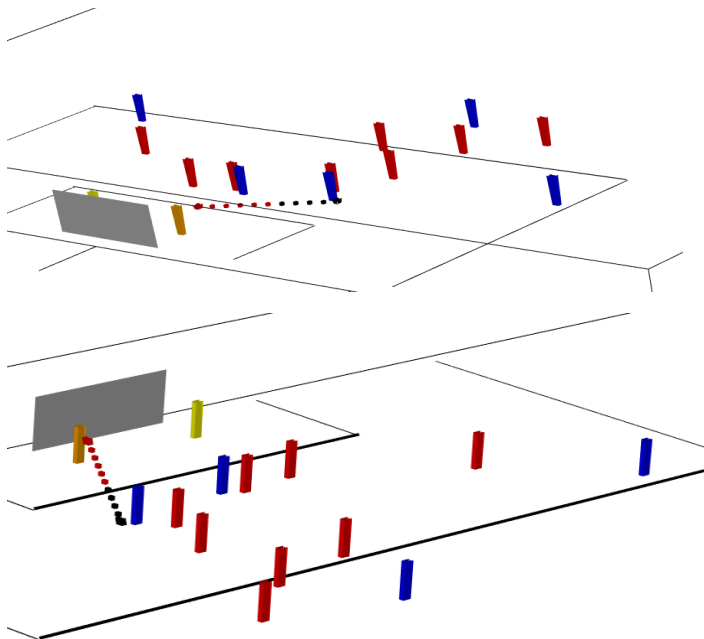
La función **freezeFrame3d()** utiliza los datos `shot.freezeFrame` para crear un mapa en 3 dimensiones de un disparo. Este mapa sirve de apoyo para la salida de la función `freezeFrame()`. La función `freezeFrame3d()` sirve para ver la situación del disparo con más perspectiva. Incluye la coordenada z del destino del disparo, esto es la altura del balón. Se distingue entre pie izquierdo, pie derecho y cabeza. El color de la trayectoria del balón indica el resultado del disparo: Gol, verde y No gol, rojo. La situación inicial del portero se representa con el color amarillo y la final con el color naranja. El gráfico de salida se puede girar al gusto del usuario con el ratón.

Argumentos:

- *Tiro*: Evento del disparo.
- *Portero*: Evento con la información del portero.

Ejemplo: La *ilustración 5-25* es la recreación 3D de la *ilustración 5-24*. Únicamente se incluyen dos capturas de distintos ángulos.

```
 tiro<-eventsLF%>%  
  filter(id==idTiro1)  
 portero<-eventsLF%>%  
  filter(id==idGoalKeeper1)  
 freezeFrame3d(tiro, portero)
```



*Ilustración 5-25. Distintas perspectivas de la ilustración 5-24 con freezeFrame3d().*

Ejemplo:

Otro ejemplo de freeze frame es el representado en la *ilustración 5-26*. Disparo realizado por Magdalena Ericsson. Expected goal = 0.56. Disparo realizado con la cabeza.

```
idTiro2<-"18f98073-7e8b-4158-818a-f246119d1903"
```

```
idGoalKeeper2<-events[which(events$id==idTiro2)+1,]
```

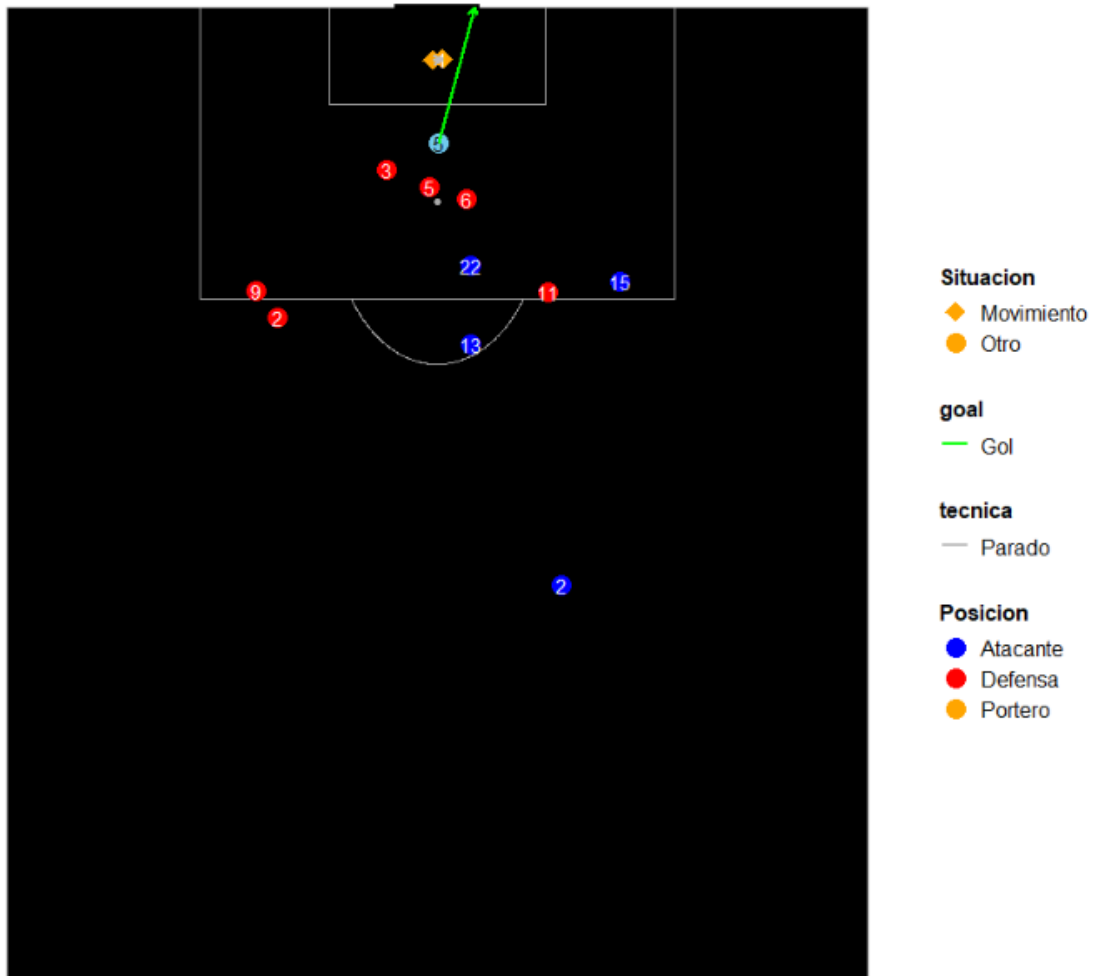
```
idGoalKeeper2<-idGoalKeeper2$id
```

```
freezeFrame(eventsLF,idTiro2,idGoalKeeper2)
```

**Magdalena Ericsson**

Expected goal 0.56

Parte del cuerpo: Head



*Ilustración 5-26. Freeze frame de un disparo de Magdalena Ericsson*

Ejemplo:

*Ilustración 5-26* incluyendo el eje Z con la función `freezeFrame3d()`. (Ver *ilustración 5-27*)

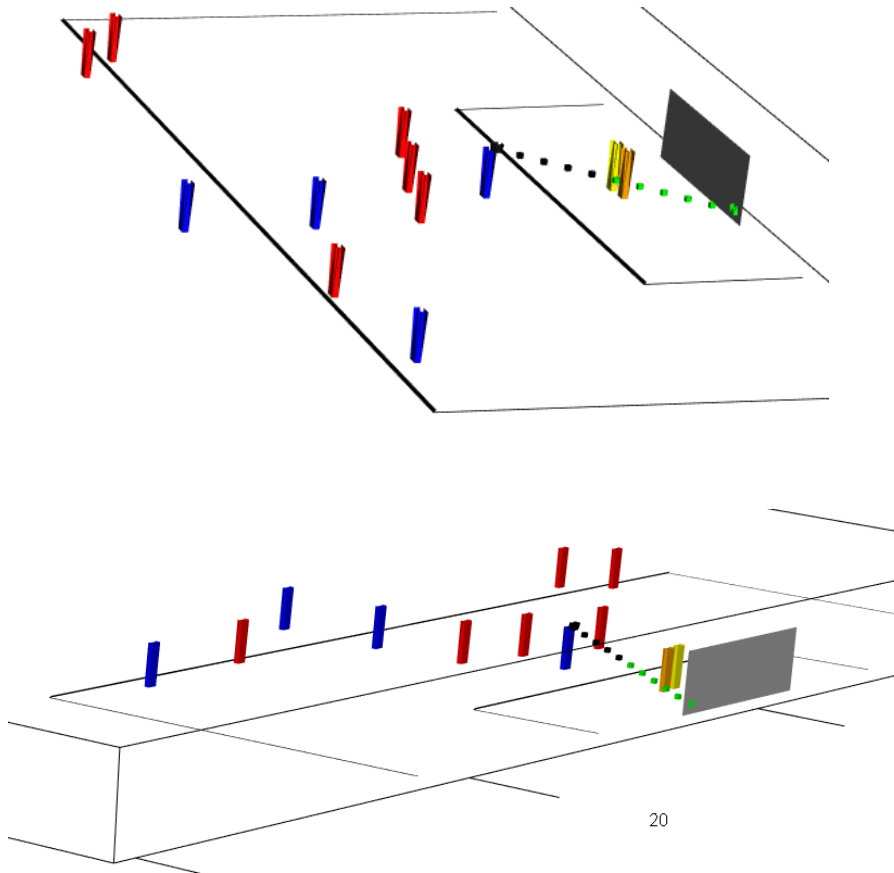
```
 tiro<-tiros%>%
```

```
  filter(id==idTiro2)
```

```
 portero<-eventsLF%>%
```

```
  filter(id==idGoalKeeper2)
```

```
 freezeFrame3d(tiro, portero)
```



*Ilustración 5-27. Distintas perspectivas de la ilustración 5-25 con `freezeFrame3d()`.*

## 5.7 Análisis de rendimiento de jugadores.

El análisis de rendimiento de jugadores consiste en medir el rendimiento de cada jugador en base a las acciones que realiza. Se valoran unas acciones u otras dependiendo la posición del jugador.

*StatsBomb* trabaja con distintas acciones por posición descritas en el *Apartado 5.3.6: rendimientoJugadores()*.

El paquete *GASB* incluye dos funciones para realizar análisis de jugadores: *radarJugador()* y *jugadorSimilar()*.

### 5.7.1 radarJugador().

La función ***radarJugador()*** realiza gráficos radar en los que se representan el valor de las acciones realizadas por un jugador. La función realiza gráficos para las posiciones *Defensa Central*, *Defensa Lateral*, y *Medio Centro*. Para cada posición se analizan 11 tipos de acciones. Los valores del radar para cada acción se escalan a partir del mínimo y el máximo de cada acción. El mínimo y el máximo se toma del percentil 5 y 95, respectivamente, de los datos históricos que posee la empresa *StatsBomb*.

La función *radarJugador()* tiene la opción de comparar al jugador con la media de las acciones realizadas por los jugadores en el contexto en su posición.

Además la función *radarJugador()* tiene como salida una tabla con los valores exactos de cada acción y el percentil que ocupa en el contexto que se le quiere dar.

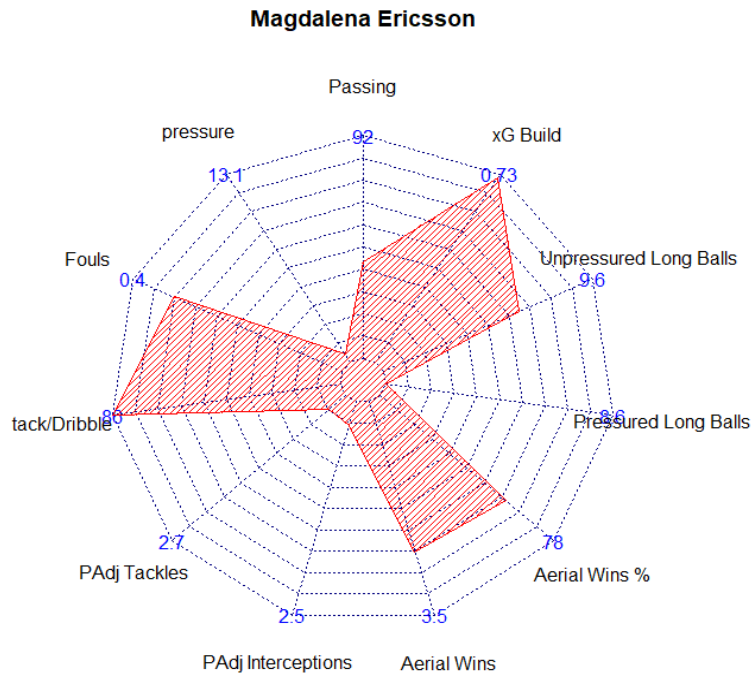
Argumentos:

- *events*: Data frame con los eventos que se quieren analizar.
- *idJugador*: Integer. Identificador del jugador objetivo de estudio.
- *Datos*: Dataframe con el rendimiento del contexto con el que se quiere comparar al jugador.
- *comparacion*: Lógico. *TRUE* si se quiere comparar al jugador con la media.
- *Posición*: String que indica la posición del jugador. "CB" para defensa central. "FB" para defensa lateral. "MD" para medio centro.

Ejemplo:

Gráfico radar de Magdalena Ericsson en la posición de defensa central. (Ver *ilustración 5-28 y tabla 13*)

radarJugador(eventsLF,4633,rendimiento,F,"CB")



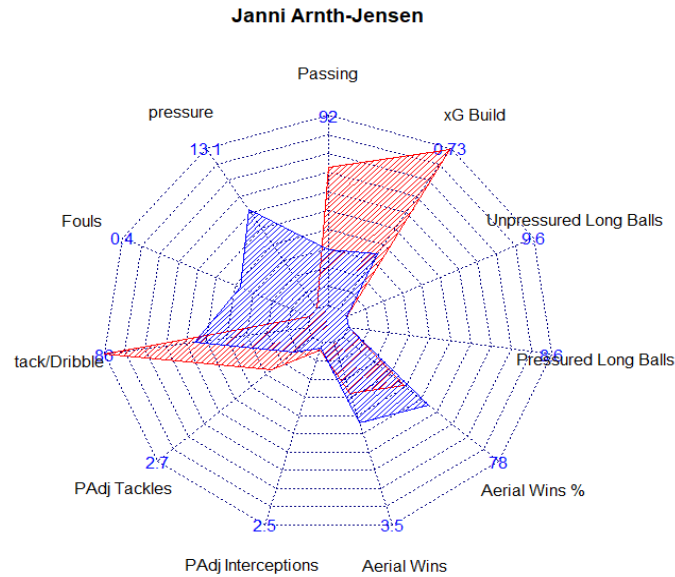
*Ilustración 5-28. Radar de Magdalena Ericsson.*

Variable	Valor	Percentil
Passing	81.2413154	64
Pressure	4.6363636	3
Fouls	0.6363636	60
Tack/dribble	86.8421053	90
PAdj Tackles	0.8059032	58
PAdj Interceptions	0.7163584	73
Aerial Wins	2.6363636	86
Aerial Wins %	68.5039370	64
Pressured Long Balls	0.6969697	88
Unpressured Long Balls	7.3939394	99
xG Build	0.7186329	86

*Tabla 13. Rendimiento Magdalena Ericsson*

Gráfico radar de Janni Arnth-Jensen en la posición de defensa central junto a la media de las jugadoras que juegan en posición de defensa central en la FA women`s Super League. (Ver *ilustración 5-29* y *tabla 14*)

radarJugador(eventsLF,idJgador = 10091,rendimiento,T,"CB")



*Ilustración 5-29. Radar de Jann Arnth-Jensen(rojo) comparado con la media de su liga(azul).*

Variable	Valor	Percentil
Passing	86.8421053	81
Pressure	4.2187500	1
Fouls	8.0859375	98
Tack/dribble	100.0000000	100
PAdj Tackles	1.1575682	79
PAdj Interceptions	0.5787841	60
Aerial Wins	1.4062500	35
Aerial Wins %	57.1428571	32
Pressured Long Balls	0.0000000	13
Unpressured Long Balls	1.4062500	18
xG Build	0.7250134	88

*Tabla 14. Rendimiento Jann Arnth-Jensen*

Gráfico radar de Jessica Marie Samuelsson en la posición de defensa lateral junto a la media de las jugadoras que juegan en posición de defensa lateral en la FA women`s Super League. (Ver ilustración 5-30 y tabla 15)

radarJugador(eventsLF, idJugador = 15612, rendimiento, T, "FB")

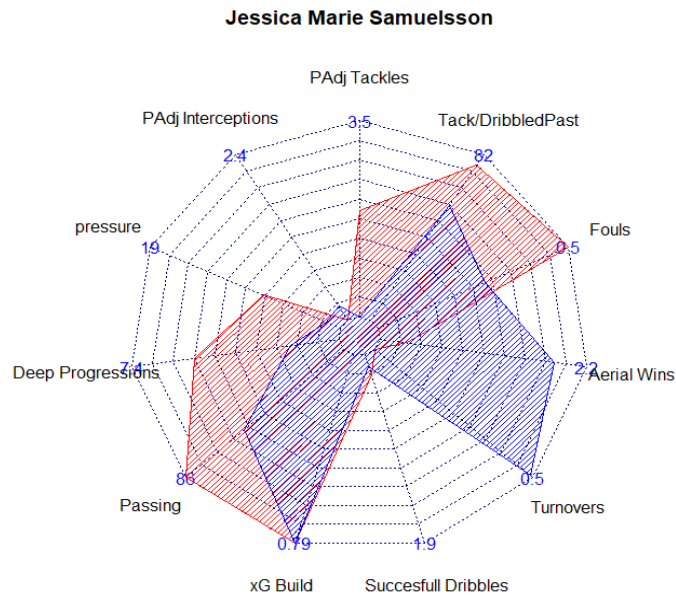


Ilustración 5-30. Radar de Jessica Marie Samuelsson(rojo) comparado con la media de su liga(azul).

Variable	Valor	Percentil
PAdj Tackles	2.3031409	100
PAdj Interceptions	0.0000000	0
Pressure	13.0569948	86
Deep Progressions	6.0621762	82
Passing	88.0281690	83
xG Build	1.5637099	100
Succesfull Dribbles	0.4663212	72
Turnovers	3.2642487	100
Aerial Wins	0.4663212	4
Fouls	0.0000000	58
Yack/DribbledPast	80.0000000	81

Tabla 15. Rendimiento Jessica Marie Samuelsson



Gráfico radar de Lauren Hemp en la posición de medio centro junto a la media de las jugadoras que juegan en posición de medio centro en la FA women`s Super League. (Ver ilustración 5-31 y tabla 16)

radarJugador(eventsLF,15555,rendimiento,T,"MD")

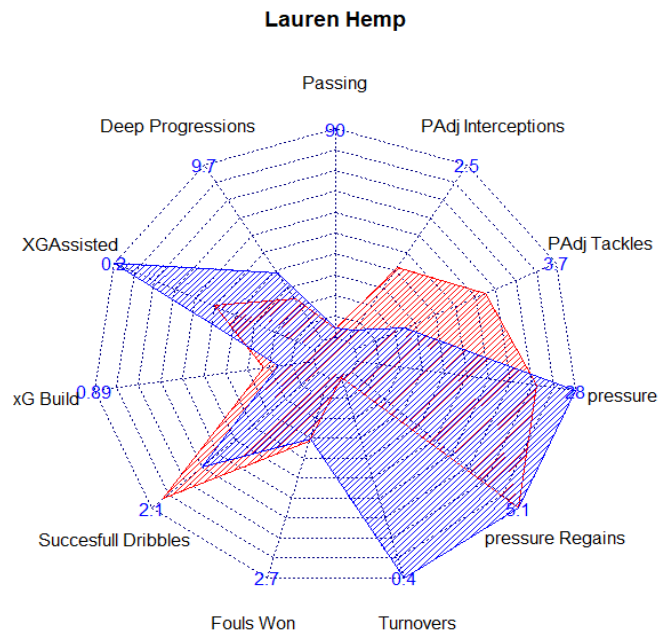


Ilustración 5-31. Radar de Lauren Hemp(rojo) comparado con la media de su liga(azul).

Variable	Valor	Percentil
Passing	72.6775956	47
Deep Progressions	4.3523316	51
XG Assisted	0.1057005	89
xG Build	0.3662736	71
Succesfull Dribbles	1.9689119	82
Fouls Won	1.1917098	64
Turnovers	4.6113990	89
Pressure Regains	5.1295337	63
Pressure	25.1295337	69
PAdj Tackles	2.6726056	91
PAdj Interceptions	1.3363028	89

Tabla 16. Rendimiento Lauren Hemp

### 5.7.2 jugadorSimilar()

La función **jugadorSimilar()** compara un jugador con otros y obtiene una lista de los jugadores “más similares” a un jugador objetivo en un determinado contexto.

La función *jugadorSimilar()* se basa en distintas acciones dependiendo de la posición (defensa central, defensa lateral, medio centro) del jugador al igual que en el *Apartado 5.7.1: radarJugador()* de este trabajo.

La función requiere la información sobre el rendimiento de los jugadores que se quieren comparar, en él debe estar incluido el jugador objetivo de estudio en la posición en la que se quiere comparar el jugador (Defensa central, defensa lateral o medio centro). Por ejemplo: Si se pretende buscar centrales similares a *Magdalena Ericsson*, esta jugadora debe estar en la información del contexto con el valor 3, 4, o 5 en la variable posición.

Para entender el funcionamiento de la función *jugadorSimilar()* debemos conocer dos conceptos: análisis de componentes principales y similitud coseno. La función mide la similitud coseno de las componentes principales de las variables del rendimiento de los jugadores. A continuación se explica con más detalle.

#### Similitud coseno:

$$\cos(\theta) = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

La similitud coseno mide la similitud entre dos vectores A y B en un espacio que posee un producto interior en el que se evalúa el valor del coseno del ángulo comprendido entre ellos. Esta función trigonométrica proporciona un valor igual a 1 si el ángulo comprendido es cero, es decir si ambos vectores apuntan a un mismo lugar. Cualquier ángulo existente entre los vectores, el coseno arrojaría un valor inferior a uno. Si los vectores fuesen ortogonales el coseno se anularía, y si apuntasen en sentido contrario su valor sería -1. De esta forma, el valor de esta métrica se encuentra entre -1 y 1. La similitud coseno se suele emplear como indicador de cohesión de clústeres y nunca como una distancia ya que no cumple la desigualdad triangular.

La base de esta métrica es que compara la proyección de los jugadores midiendo su ángulo respecto al origen. En la *ilustración 5-32* tenemos el siguiente ejemplo. Suponiendo los ejes x e y dos características de los puntos X, Y y Z. Si empleamos la distancia euclídea, el punto Z está más cerca del punto Y que del X, pero si nos fijamos en el ángulo respecto al origen, el punto X y Z crean un ángulo menor que el punto X e Y, es decir, Z es más similar a X que a Y. La proyección de X se acerca más a Z que la proyección de Y.

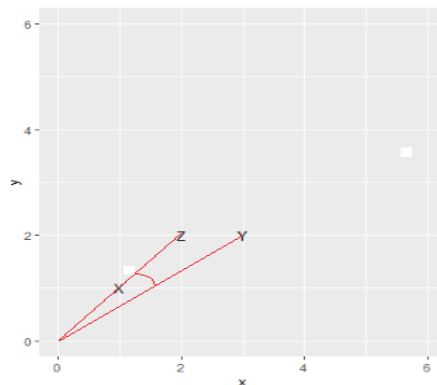


Ilustración 5-32. Ejemplo similitud coseno

### **Análisis de componentes principales y similitud coseno:**

La función `jugadorSimilar()` emplea en particular un **análisis de componentes principales (ACP) escalado y centrado**. Muchas variables están correlacionadas, Aerial Wins-Aerial Wins % y passing-xG build son dos ejemplos. La función principal de un ACP es reducir el número de dimensiones concentrando la variabilidad de los datos en componentes principales las cuales explican la variabilidad de los datos de más a menos.

Con el ACP obtenemos vectores con sus variables totalmente independientes unas de otras, requisito para utilizar la similitud coseno.

Al hacer ACP escalado y centrado conseguimos que las variables tengan el mismo peso para el análisis. Además en un ACP los valores para las componentes principales toman valores positivos y negativos lo cual le da más consistencia a aplicar la similitud coseno.

La similitud coseno y el ACP tienen un punto en común: la varianza da protagonismo. En el ACP los valores de una componente principal tienen la misma varianza que la variabilidad que explica esa misma componente principal, una componente principal es más importante cuanto más variabilidad explique. El ACP cuenta con la ventaja de que esta variabilidad es a su vez la varianza de los valores para esa componente principal. Por otro lado, en la similitud coseno una vez escalados los datos, las variables que más intervienen en el resultado son las que más variabilidad tienen, por ejemplo, en un espacio vectorial de dos dimensiones (X,Y), si la variable X tiene una varianza de 10 e Y una varianza de 1, dos vectores con una diferencia de 0.5 en la variable X son más similares que dos vectores con una diferencia de 0.5 en la variable Y. Por lo tanto aplicar la similitud coseno a las componentes principales es una herramienta muy potente

### **Función `jugadorSimilar()`:**

La función `jugadorSimilar()` filtra del contexto los jugadores en la posición objetivo de estudio (defensa central, defensa lateral o medio centro) y sus respectivas acciones de interés vistas en el *Apartado 5.7.1: radarJugador()*.

Una vez filtrado el contexto realiza un ACP escalado y centrado.

Por último calcula la similitud coseno de los valores de las componentes principales de los distintos jugadores.

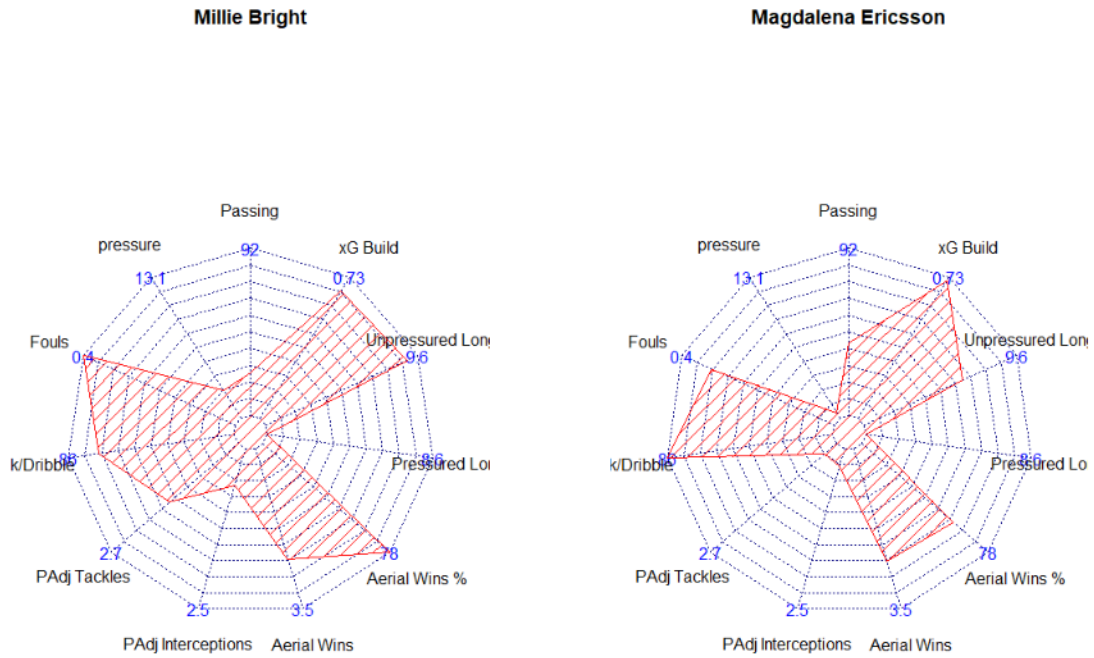
Argumentos de la función `jugadorSimilar()`:

- *events*: Dataframe con los eventos del jugador.
- *idJugador*: Integer. Identificador jugador objetivo de estudio.
- *Datos*. Integer. Información de rendimiento de los jugadores a comparar.
- *posicion*: String que indica la posición del jugador. "CB": Defensa Central. "FB": Defensa Lateral. "MD": Medio Centro

Ejemplo:

La jugadora más similar a Millie Bright es Magdalena Ericsson. Su similitud es 0.7118. Ver *ilustración 5-33 y tabla 17*.

`jugadorSimilar(eventsLF, idJugador = 4642, rendimiento, posicion="CB")`



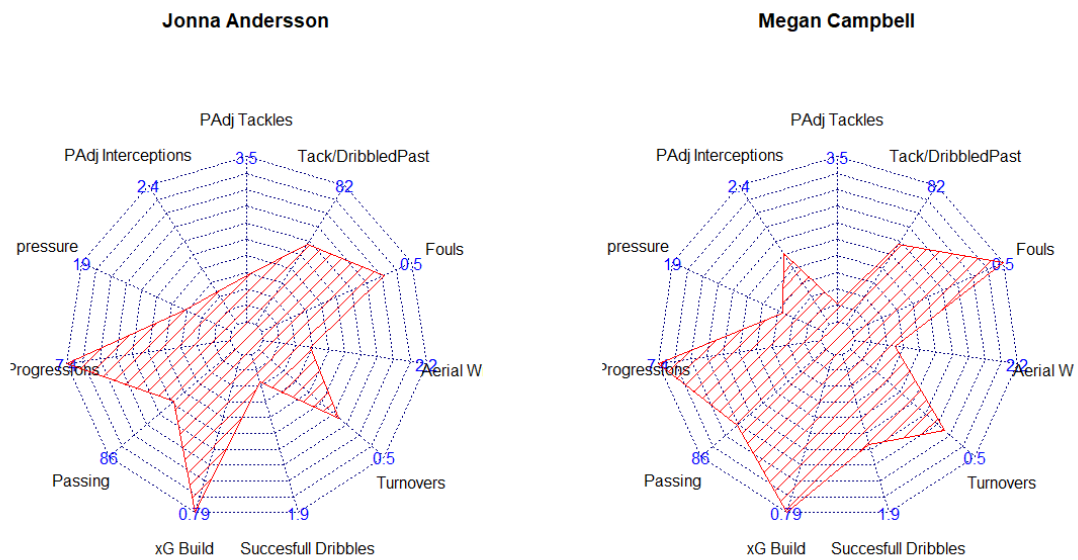
*Ilustración 5-33. Jugador similar a Millie Bright. Magdalena Ericsson.*

Jugador	Similitud
4633	0.7118659
10185	0.6101139
18999	0.5963737
20061	0.5755479

*Tabla 17. Jugadoras similares a Millie Bright.*

Jonna Andersson tiene una similitud coseno de 0.858 con Megan Cambell. Ver *ilustración 5-34* y *tabla 18*.

`jugadorSimilar(eventsLF, 10222, rendimiento, posicion="FB")`



*Ilustración 5-34. Jugadora similar a Jonna Andersson, Megan Campbell.*

Jugador	Similitud
23465	0.8581406
17275	0.7029530
15620	0.6343435
10179	0.6324435
19422	0.5967617
10395	0.5510430
15612	0.5205945

*Tabla 18. Jugadoras similares a Jonna Andersson.*

En el caso de Dominique Johanna Anna Bloodworth encontramos a Keira Waish como jugadora más similar con una similitud coseno de 0.959 (ilustración 5-35). Para esta jugadora también se encuentran otras dos jugadoras con una similitud por encima de 0.9, Drew Spence (0.928) y Katie Zelem (0.92241), ver tabla 19. Ver ilustración 5-36 e ilustración 5-37

jugadorSimilar(eventsLF, 10666,rendimiento,posicion="MD")

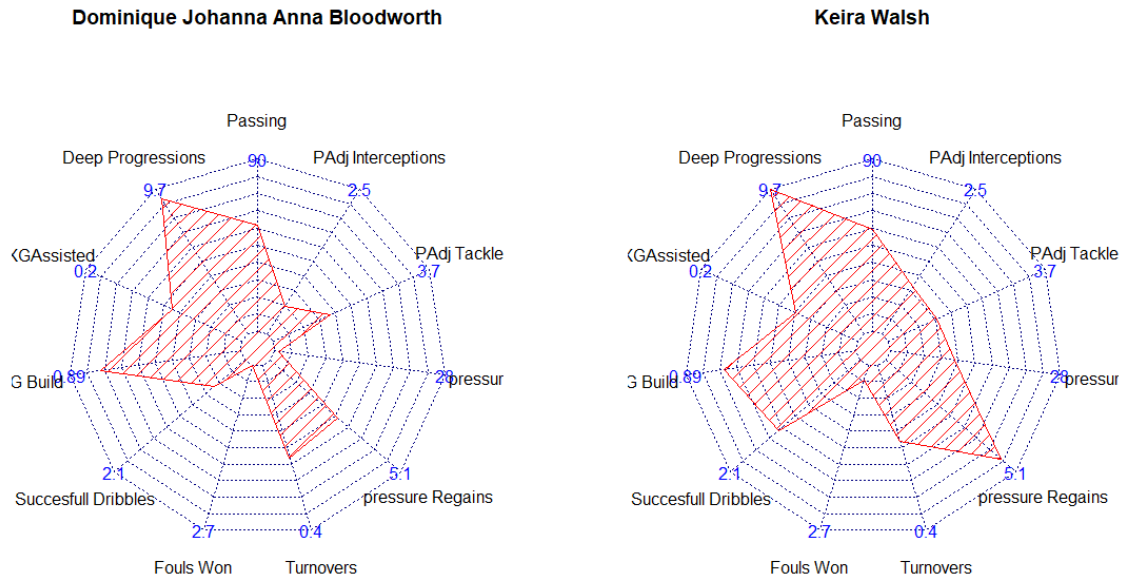
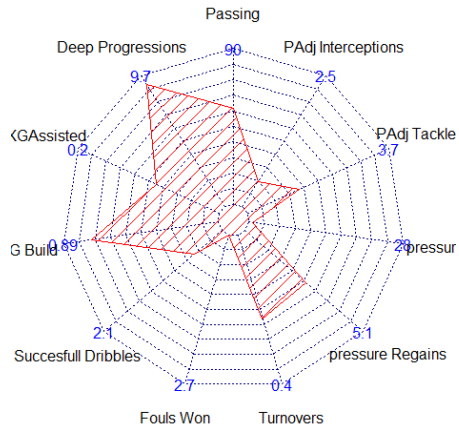


Ilustración 5-35. Jugadora similar a Dominique Jonna, Keira Waish.

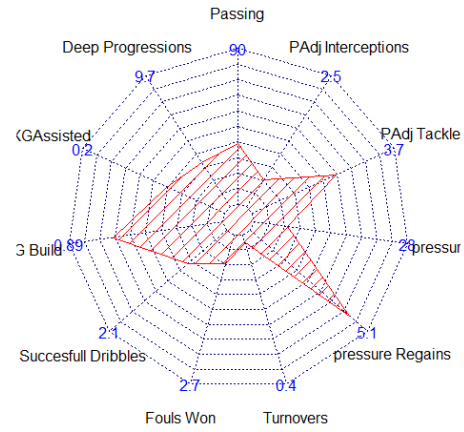
Jugador	Similitud
4658	0.9590912
4658	0.9282471
31540	0.9224108
15549	0.8909476
15616	0.8397158
10172	0.8270301
10650	0.7885379
10188	0.7657362
10192	0.7333380

Tabla 19. Jugadoras similares a Dominique Jonna.

**Dominique Johanna Anna Bloodworth**

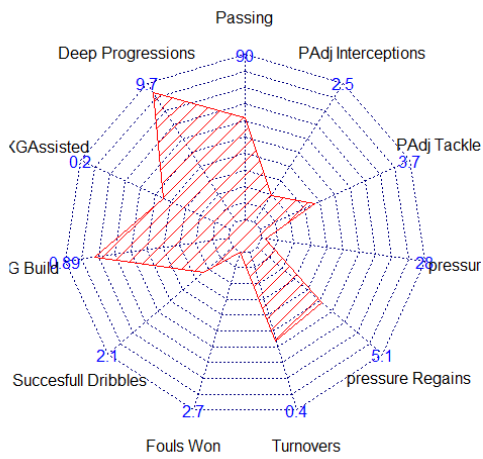


**Drew Spence**

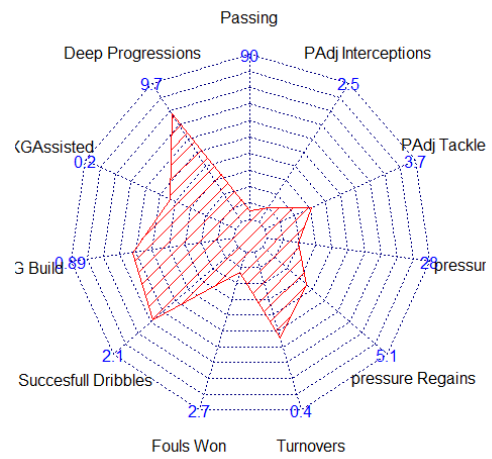


*Ilustración 5-36. Gráfico radar Dominique Jonna y Drew Spence. Realizado con radarJugador().*

**Dominique Johanna Anna Bloodworth**



**Katie Zelem**



*Ilustración 5 37. Gráfico radar Dominique Jonna y Katie Zelem. Realizado con radarJugador().*

### Código jugadorSimilar():

```
jugadorSimilar<-function(events,idJugador,datos,posicion){
  if(posicion=="CB"){
    datosaux<-datos[which(datos$posicion%in%c(3,4,5)),]
    datosaux<-datosaux%>%
      select(passing,pressure,fouls,tackDribbled,PAdjTackles,PAdjInterceptio
ns,AerialWins,AerialWinsPor,PressuredLongBalls,UnPressuredlongBalls,
Xgbuild)
    jugadores<-as.numeric(rownames(datosaux))
    pca <- prcomp(datosaux, scale = TRUE, center=T)
    similitudes<-
data.frame(matrix(rep(Inf,length(datosaux$passing)*length(datosaux$passing
)),ncol = length(datosaux$passing)))
    names(similitudes)<-rownames(datosaux)
    rownames(similitudes)<-rownames(datosaux)

    for(i in 1:length(datosaux$passing)){
      for(j in 1:length(datosaux$passing)){
        if(i==j){
          similitudes[i,j]<--Inf
        }else{
          similitudes[i,j]<-
sum(pca$x[i,1:11]*pca$x[j,1:11])/sqrt(sum(pca$x[i,1:1
1]^2)*sum(pca$x[j,1:11]^2))
        }
      }
    }
    for(i in 1:length(datosaux$passing)){
      if(jugadores[i]==idJugador){
        jugador<-i
      }
    }
    minimos<-data.frame(Inf,0.5)
    names(minimos)<-c("jugador","distancia")
    for (j in 1:length(jugadores)){
```



```

        if(min(minimos$distancia)<similitudes[jugador,j]){
            minimos<-rbind(minimos,c(jugadores[j],similitudes[jugador,j]))
            minimos<-minimos[order(minimos$distancia,decreasing = T),]
        }
    }
    minimos<-minimos[1:(length(minimos$jugador)-1),]
    par(mfrow=c(1,2))
    radarJugador(events,idJugador,datos,posicion = "CB")
    radarJugador(events,minimos[1,1],datos,posicion = "CB")
    return(minimos)
}

if(posicion=="FB"){
    datosaux<-datos[which(datos$posicion%in%c(2,6,7,8)),]
    datosaux<-datosaux%>%
        select(PAdjTackles,PAdjInterceptions,pressure,deepProgresions,passing
            ,Xgbuild,succesDribbles,Turnovers,AerialWins,fouls,tackDribbled)
    jugadores<-as.numeric(rownames(datosaux))
    pca <- prcomp(datosaux, scale = TRUE,center = T)
    similitudes<-
    data.frame(matrix(rep(Inf,length(datosaux$passing)*length(datosaux$passing
    )),ncol = length(datosaux$passing)))
    names(similitudes)<-rownames(datosaux)
    rownames(similitudes)<-rownames(datosaux)
    for(i in 1:length(datosaux$passing)){
        for(j in 1:length(datosaux$passing)){
            if(i==j){
                similitudes[i,j]<--Inf
            }else{
                similitudes[i,j]<-
                sum(pca$x[i,1:11]*pca$x[j,1:11])/sqrt(sum(pca$x[i,1:1
                1]^2)*sum(pca$x[j,1:11]^2))
            }
        }
    }
}

```

```

for(i in 1:length(datosaux$passing)){
  if(jugadores[i]==idJugador){
    jugador<-i
  }
}
minimos<-data.frame(Inf,0.5)
names(minimos)<-c("jugador","distancia")
for(j in 1:length(jugadores)){
  if(min(minimos$distancia)<similitudes[jugador,j]){
    minimos<-rbind(minimos,c(jugadores[j],similitudes[jugador,j]))
    minimos<-minimos[order(minimos$distancia,decreasing = T),]
  }
}
minimos<-minimos[1:(length(minimos$jugador)-1),]
par(mfrow=c(1,2))
radarJugador(events,idJugador,datos,posicion = "FB")
radarJugador(events,minimos[1,1],datos,posicion = "FB")
return(minimos)
}
if(posicion=="MD"){
  datosaux<-datos[which(datos$posicion%in%c(9:16)),]
  datosaux<-datosaux%>%
  select(passing,deepProgresions,XgAssisted,Xgbuild,succesDribbles,foulsWon,Tu
rners,pressureRegains,pressure,PAdjTackles,PAdjInterceptions)
  pca <- prcomp(datosaux, scale = TRUE)
  jugadores<-as.numeric(rownames(datosaux))
  similitudes<-
  data.frame(matrix(rep(Inf,length(datosaux$passing)*length(datosaux$passing
)),ncol = length(datosaux$passing)))
  names(similitudes)<-rownames(datosaux)
  rownames(similitudes)<-rownames(datosaux)
  for(i in 1:length(datosaux$passing)){
    for(j in 1:length(datosaux$passing)){
      if(i==j){

```

```

        similitudes[i,j]<--Inf
      }else{
        similitudes[i,j]<-
        sum(pca$x[i,1:11]*pca$x[j,1:11])/sqrt(sum(pca$x[i,1:1
1]^2)*sum(pca$x[j,1:11]^2))
      }
    }
  }
  for(i in 1:length(datosaux$passing)){
    if(jugadores[i]==idJugador){
      jugador<-i
    }
  }
  minimos<-data.frame(Inf,0.7)
  names(minimos)<-c("jugador","distancia")
  for (j in 1:length(jugadores)){
    if(min(minimos$distancia)<similitudes[jugador,j]){
      minimos<-rbind(minimos,c(jugadores[j],similitudes[jugador,j]))
      minimos<-minimos[order(minimos$distancia,decreasing = T),]
    }
  }
  minimos<-minimos[1:(length(minimos$jugador)-1),]
  par(mfrow=c(1,2))
  radarJugador(events,idJugador,datos,posicion = "MD")
  radarJugador(events,minimos[1,1],datos,posicion="MD")
  return(minimos)
}
}

```

## 6 Conclusiones

En este TFG se ha visto la utilidad del análisis de datos de partidos de fútbol. El paquete GASB realiza análisis que concentran mucha información y muy útil sin dejar de lado el juego, de manera que el usuario se siente relacionado con los momentos del juego. El paquete GASB es flexible con los intereses del usuario permitiendo elegir en muchas funciones el tipo de análisis a realizar.

Elegir la posición de los jugadores con los que se compara el jugador objetivo en la función `analisisDefensivoJugador()` es idea original del autor. El resto de análisis defensivos por zonas comparan al jugador objetivo con jugadores de todas las posiciones, esto produce que siempre destaque en su posición ya que está la mayor parte del tiempo en esas zonas. Sin embargo al comparar a los jugadores solamente con jugadores de su posición sale a la luz el volumen de trabajo defensivo tanto en su zona como en otras zonas más alejadas.

La función `jugadorSimilar()` realiza un método de similitud totalmente original. En muchas funciones del paquete me he inspirado en análisis de datos de empresas y equipos, pero el código y el método de todas ellas es original. En el caso de la función `jugadorSimilar()` el método no es tan intuitivo, hay muchas maneras de medir la similitud. La elegida por el autor consiste en realizar un análisis de componentes principales a las variables de rendimiento de los jugadores y calcular la similitud coseno a los valores de los jugadores en las componentes principales

El análisis de datos en el juego es el menos desarrollado en fútbol, ya que los análisis de captación de jugadores es donde se saca el mayor rendimiento económico. En este trabajo se ha visto como los datos pueden ser muy útiles para apoyar los análisis de los profesionales.

Como trabajo futuro se plantea idear y realizar análisis de otros aspectos del juego como acciones a balón parado y análisis de porteros. También me gustaría estudiar las acciones defensivas con más detalle. Se valora igual un acto de presión y un robo de balón, con datos sería posible ver si está bien o si se puede realizar un modelo que ajuste el valor de cada acción, es decir, ponderar cada acción por categoría y zona. Dar un valor a un robo de balón en el área contraria distinto a un acto de presión en el medio campo.

## 7 Referencias

Murdoch, D. and Adler, D. (2021). rgl: 3D Visualization Using OpenGL. R package version 0.106.8. <https://CRAN.R-project.org/package=rgl>

Nakazawa, M. (2019). fmsb: Functions for Medical Statistics Book with some Demographic Data . R package version 0.7.0. <https://CRAN.R-project.org/package=fmsb>

*StatsBomb* [en línea]. Descripción de datos y métricas [consulta: Febrero 2021 – mayo 2021]. Disponible en: <https://statsbomb.com/es/>.

Sumpter, D. “*Soccermatics: Mathematical adventures in the beautiful game*”. editorial: *Bloomsbury* (2016).

Three Points [en línea]. Big Data en el deporte [en línea]. Introducción. [consulta 20 de Junio de 2021]. Disponible en <https://www.threepoints.com/int/big-data-aplicado-a-los-deportes>

Wickham, H. , Hester, J. and Chang, W. (2021). devtools: Tools to Make Developing R Packages Easier. R package version 2.4.1. <https://CRAN.R-project.org/package=devtools>

Wickham, H. , Danenberg, P. , Csárdi, G. and Eugster, M. (2020). roxygen2: In-Line Documentation for R. <https://roxygen2.r-lib.org/>, <https://github.com/r-lib/roxygen2>.

Wickham, H. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag New York, 2016.

Wickham H. et al., (2019). Welcome to the tidyverse. *Journal of Open Source Software*, 4(43), 1 686, <https://doi.org/10.21105/joss.01686>

Wikipedia [en línea]. Similitud coseno [consulta: 10 de Mayo de 2021]. Disponible en [https://es.wikipedia.org/wiki/Similitud\\_coseno](https://es.wikipedia.org/wiki/Similitud_coseno)

Yam, D. (2020). StatsBombR: Cleans and pulls StatsBomb data from the API. R package version 0.1.0.

## Apéndice

Tabla 1 se recogen algunas variables comunes de los eventos con una breve descripción de cada una.

Columna	Tipo	Secundario	Tipo Secundario	Descripción	Valores	Descripción de los valores
Id	Uuid			Valor único para el evento.	Ej. "57726e09-9ba1-4388-8c40-faf57e5b8d1d"	
index	integer			Secuencia de los eventos por orden en el partido.	Ej, 1 #de eventos	
period	integer			Parte del partido.	1	1er tiempo
					2	2do tiempo
					3	1er tiempo prórroga
					4	2do tiempo prórroga
					5	Tanda Penaltis
timestamp	timestamp			Tiempo del partido en el que el evento tuvo lugar.	Ej. 00:00:02.78	
minute	Integer			Minuto del partido en el que ocurrió el evento.	Ej. 43	
second	Integer			Segunda parte del timestamp	Ej. 59	
type	object	Id/name	Integer/text	Id/nombre del evento	4/"Ball Receipt"	Receptor de un pase.
					2/"Ball recovery"	Intento de balón
					3/"Dispossessed"	Jugador pierde el balón ante la entrada de un rival sin intentar regatear.
					4/"Duel"	Duelo entre dos jugadores al 50/50.
					5/"Camera On"	La cámara dejó de grabar.
					6/"Block"	Bloquear un balón

						interponiendos e en el camino.
					8/"Offside"	Infracción del fuera de juego.
					9/"Clearance"	Despeje de balón para evitar peligro.
					10/"Interception"	Interceptar el balón en un pase del rival.
					14/"Dribble"	Regatear un rival.
					16/"Shot"	Tiro a portería..
					17/"Pressure"	Presionar a rival poseedor del balón.
					18/"Half Start"	Comienzo de un periodo.
					19/"Substitution"	Sustitución.
					20/"Own Goal Against"	Más de un gol en propia de un equipo en un partido.
					21/"Foul Won"	Falta recibida.
					22/"Foul Committed"	Falta cometida por un el equipo.
					23/"Goal Keeper"	Acciones que puede hacer un portero.
					24/"Goal Keeper"	Acción que puede realizar un portero.
					25/"Own Goal For"	Gol en propia.
					26/"Player On"	Vuelta de un jugador al campo que se ha perdido un evento.
					27/"Player Off"	Jugador sale del campo y no es sustitución ni expulsión.
					28/"Shield"	Proteger y dejar salir el balón para obtener la posesión.
					30/"Pass"	Pase a un compañero.
					33/"50/50"	Dos jugadores compiten por un balón suelto.
					34/"Half End"	Fin de un periodo.
					35/"Startring XI"	Alineación inicial.
					36/"Tactical Shift"	Táctica inicial del equipo.

					37/ "Error"	Error de un jugador que provoca un tiro en contra.
					38/ "Miscontrol"	Perder el balón por un mal control.
					39/ "Dribbled Past"	Jugador es superado por el regate de un rival.
					40/ "Injury Stoppage"	Paro del juego por una lesión.
					41/ "Refree Ball-Drop"	Balón a tierra.
					43/ "Carry"	Conducción del balón.
posesion	Integer			Indicador único de una posesión es un partido.	Ej. 1	No comienza una nueva posesión hasta que el equipo ha estabilizado la posesión del balón.
posesión_team	Object	Id/name	integer/text	Id y nombre del equipo con la posesión del balón.	Ej, 1/"Arsenal"	
play_pattern	Object	Id/name	Integer/text	Situación de juego.	1/Regular Play	El evento no transcurre en una situación de las siguientes
					2/"From Corner"	Eventos sucesivos a un corner.
					3/"From Free Kick"	Eventos sucesivos a una falta.
					4/"From Throw In"	Eventos sucesivos a un centro al área.
					5/"Other"	
					6/"From Counter"	Los eventos son parte de un contrataque.
					7/"From Goal Kick"	Eventos sucesivos a un saque de puerta.
					8/"From Keeper"	Eventos después de una acción del portero.



					9/ "From Kick Off"	Eventos siguientes a un saque inicial.
team	object	Id/name	Integer/text	Id/Nombre del equipo que realiza el evento.	Ej.1/"Arsenal"	
player	object	Id/name	Integer/text	Id/nombre del futbolista que realiza el evento.	Ej. 5079/"Zlatan Ibrahimovic"	
position	object	Id/name	Integer/text	Id/nombre de la posición de un futbolista que realiza el evento.	Ej. 1/"Goal Keeper"	
location	array(x, y)			Array con dos enteros que determinan las coordenadas del campo de ese evento.	Ej. (60, 40)	Las dimensiones del campo se explican a continuación.
duration	decimal			El tiempo en segundos de un evento.	Tiempo en segundos	
under_pressure	Boolean			La acción fue realizada bajo presión del rival.	True	Información más adelante.
off_camera	Boolean			El evento fue filmado por la cámara.	FALSE, TRUE	
Out	Boolean			Añadido si el evento se realizó con la pelota fuera del campo.	TRUE	
related_events	array(uuid,uuid....)			Array de ids de eventos relacionados. Ejemplo: Disparo, parada.	e.g., "2b7d06c7-9bcb-4bbfa6e5-08e54e1303ac" or ("2b7d06c7-9bcb-4bbf-a6e5-08e54e1303ac", "03b001b6-bf44-4691-ae47-e279f5a9b38c")	
tactics	object	formation	text	Para evento de tipo "Starting XI"	Ej. 343	Ej. Tres defensas 4 medios tres delanteros
		lineup	array	Describir las posiciones de los jugadores en la alineación.	Ver más adelante.	

Tabla 2 recoge algunos tipos de eventos con sus variables propias y sus respectivas descripciones

Tipo de evento (Data Frame anidado)	Columna	Tipo	Tipo secundario	Secundario/s	Descripción	Valores	Descripción de valores
50_50	outcome	object	Id/name	Integer/text	Id/nombre del resultado del duelo 50/50	108/"Won"	El jugador gana el duelo.
						109/"Lost"	El jugador pierde el duelo.
						147/"Success To Team"	El balón lo recogió su equipo
						148/"Success To Opposition"	El balón lo recogió el rival.
	counterpre ss	Boolean			Evento realizado 5 segundos después de perder el balón	TRUE	
Bad Behaviour	card	object	Id/name	Integer/text	Id/nombre nombre de la tarjeta mostrada.	65/"Yellow Card"	
						66/"Second Yellow"	
						67/"Red Card"	
Ball Receipt	outcome	object	Id/name	Integer/text	Id/nombre resultado de recibir el balón	9/"Incomplete"	
Ball Recovery	offensive	boolean			Recuperación de un balón por acierto propio	TRUE	El rival intentaba regatear.
	Recovery_f ailure	boolean			La recuperación fue por fallo del rival	TRUE	Se recupera un pelota que intenta recuperar el rival.
Block	deflection	boolean			El balón es desviado.	TRUE	El balón sigue su dirección pero desviada.
	offensive	boolean			El bloque fue en ataque.	TRUE	
	save_block	boolean			El bloque salva un disparo	TRUE	
	counterpre ss	boolean				TRUE	
Carry	end_locatio n	Array(x,y)			Coordenadas donde acaba la conducción del balón	Ej.(70,90)	
Clearance	aerial_won	boolean			Si fue en un duelo aéreo	TRUE	
	body_part	object	Id/name	Integer/text		37/"Head"	

					Id/nombre de la parte del cuerpo con la que se realiza.	38/"Left Foot" 70/"Other" 40/"Right Foot"	
Dribble	overrun	boolean					
	nutmeg	boolean			Cuando el regate se hizo entre las piernas del defensor		
	No Touch	boolean			Si el jugador regateó sin tocar el balón	TRUE	
	outcome	object	Id/name	Integer/text	Resultado del regate	8/"Complete" 9/"Incomplete"	
Dribbled Past	counterpress	boolean				TRUE	
Duel	counterpress	boolean				True	
	type	object	Id/name	Integer/text	Id/nombre del tipo de duelo	10/"Aerial Lost" 11/"Tackle"	Duelo aéreo perdido Oponerse a un rival.
	outcome	object	Id/name	Integer/text	Id/nombre del resultado del duelo	1/"Lost" 4/"Won" 13/"Lost in Play" 14/"Lost Out" 15/"Success" 16/"Success in Play" 17/"Success Out"	Entrada exitosa El balón cayó en un rival El balón salió para posesión del rival.  El balón cayó en un rival El balón salió para posesión del equipo.
Foul Committed	counterpress	boolean				TRUE	
	offensive	boolean			Falta cometida con posesión del balón	TRUE	
	type	object	Id/name	Integer/text	Id/nombre del atributo de la falta cometida	19/"6 seconds" 20/"Backpass Pick" 21/"Dangerous Play" 22/"Dive" 23/"Foul Out" 24/"Handball"	El portero retuvo el balón 6 segundos con la mano. El portero cogió un pase del compañero realizado con el pie. Juego peligroso Simular una falta. Falta normal. Mano.

	advantage	boolean			La falta se cometió pero el juego siguió por tener ventaja.	TRUE	
	penalty	boolean			La falta es penalti	TRUE	
	card	object	Id/name	Integer/text	Id/nombre del tipo de la tarjeta	5/"Yellow Card"	
						6/"Second Yellow"	
						7/"Red Card"	
Foul Won	defensive	boolean			Añadir si la falta recibida fue con la posesión de la pelota	TRUE	
	advantage	boolean			Añadir si se aplicó la ley de la ventaja	TRUE	
	penalty	boolean			Si la falta recibida fue penalti	TRUE	
GoalKeeper	position	object	Id/name	Integer/text	Id/nombre del atributo de la situación de un portero antes de un disparo	42/"Moving"	El portero se estaba moviendo.
						43/"Prone"	Estaba en el suelo
						44/"Set"	Estaba parado.
	technique	object	Id/name	Integer/text	Id/nombre de la técnica usada por el portero	45/"Diving"	Lanzarse al suelo.
						46/"Standing"	Paró el balón en la posición.
	body_part	object	Id/name	Integer/text	Id/nombre de la parte del cuerpo con la que realizó la parada.	35/"Both Hands"	Ambas manos.
						36/"Chest"	Pecho
						37/"Head"	
						38/"Left Foot"	
						39/"Left hand"	
						40/"Right Foot"	
	41/"Right Hand"						
	type	object	Id/name	Integer/text	Id/nombre de toda situación que puede realizar un portero. Conceder gol, parar, bloquear, despejar...	25/"Collected"	Atrapar el balón.
26/"Goal Conceded"							
28/"Penalty Conceded"							
29/"Penalty Saved"							
30/"Punch"						Despeja de puños.	
32/"Shot Faced"						No lo paró ni le metieron gol	
33/"Shot Saved"							

					34/"Smother"	Equivalente a entrada/tackle , portero saliendo a desposeer el balón de un rival.
					113/"Saved Off T"	Parada a un disparo que iba fuera de portería.
					114/"Shot Saved To Post"	Parada a un disparo que rechaza en el poste.
					110/"Saved To Post"	Parada que acaba eb el palo.
					109/"Penalty Saved To Post"	
					31/"Saved"	Portero salva algo que no es un disparo
					27/"Keeper Sweeper"	El portero sale del área pequeña a despejar.
	outcome	object	ld/name	Integer/text	47/"Claim"	Keeper Sweeper que atrapó el portero
					48/"Clear"	Keeper Sweeper que despeja
					49/"Collected Twice"	Recoge el balón después de varios intentos
					50/"Fail"	Accion desafortunada
					51/"In play"	Parada con el balón en juego
					52/"In Play Danger"	Parada a un tiro del oponente
					53/"In Play Safe"	Portero para el trio de un compañero
					55/"No touch"	Gol concedido que el portero no tocó.
					56/"Saved Twice"	Parar disparos consecutivos
					157/"Succes"	Rechace a un compañero
					58/"Touched In"	Parada que acabó en gol.
					59/"Touched Out"	Portero para el balón y sale fuera
					4/"Won"	Retener la posesión
					16/"Succes In Play"	Rechace que recoge un compañero
					17/"Succes out"	Rechace que sale fuera pero en

						posesión de su equipo.
						13/"Lost in Play"
						14/"Lost out"
						117/"Punvhed Out"
Half End	Early Video End	boolean			El video no llega al final del periodo	TRUE
	Match suspended	boolean			El árbitro decide posponer o suspender el partido	TRUE
Half Start	Late Video Start	boolean			No se grabó el inicio del periodo	TRUE
Injury Stoppage	In_vhain	boolean			Añadir se la pelota la tenía el equipo del lesionado	TRUE
Interception	outcome	object	Id/name	Integer/text	Id/nombre del resultado de la intercepción	1/"Lost"
						13/"Lost in Play"
						14/"Lost Out"
						15/"Succes"
						16/"Succes In Play"
						17/"Succes Out"
4/"Won"						
Miscontrol	aerial_won	boolean		Añadir si el control fue en el aire	TRUE	
Pass	recipient	object	Id/name	Integer/text	Id/nombre del jugador que recibió el balón.	Ej. 2934/"Erik Lamela"
	length	decimal			La longitud del pase en yardas	Ej. 16.9
	angle	decimal			El ángulo del pase donde 0 mira hacia delante, valores positivos de	Ej. 2.49

					0 a pi en sentido horario, y 0 - pi en sentido antihorario		
Height	Object	Id/name	Integer/text	Id/nombre de la altura del pase.	17/"Ground Pass"	Pase raso	
					2/"Low Pass"	Pase bajo	
					3/"High Pass"	Pase alto	
end_location	Array(x, y)			Array con las coordenadas donde acabó el pase	Ej. (20,80)		
assisted_shot_id	uuid			Id del disparo al que asiste	Ej. "57726e09-9ba1-4388-8c40-faf57e5b8d1d"		
backheel	boolean			Añadir si el pase fue de tacón	TRUE		
deflected	boolean			Añadir si el pase es desviado	TRUE	Puede ser completo o incompleto	
miscommunication	boolean			Añadir si el pase fue una falta de comunicación	TRUE	El pase era preciso pero el jugador que tenía que recibir se equivocó.	
cross	boolean			Añadir si el pase fue un cruzado	TRUE	Se describe más adelante	
cut_back	boolean			Añadir si el pase fue un "pase de la muerte"	TRUE	Información más adelante	
switch	boolean			Añadir si el pase superó el medio campo	TRUE	Estos pases se valoran si son más largos de 40 yardas.	
Shot_assist	boolean			Añadir si el pase fue una asistencia de tiro	TRUE		
goal_assist	boolean			Añadir si el pase fue una asistencia de gol.	TRUE		
body_part	object	Id/name	Integer/text	Id/nombre de la parte del cuerpo con la que se realiza el pase	68/"Drop Kick"	Saque de bolea del portero	
					37/"Head"		
					69/"Keeper Arm"	Pase con la mano del portero	
					38/"Left Foot"		
					70/"Other"		
					40/"Right Foot"		
106/"No Touch"	El jugador destino dejó pasar el balón para que lo recogiese un compañero.						
type	object	Id/name	Integer/text		37/"Corner"		
					62/"Free Kick"	Tiro Libre	

					Id/nombre atributo del tipo de pase	63/"Goal Kick"	Saque de meta
						64/"Interception"	Intercepcion que a su vez es un pase
						65/"Kick Off"	Pase de inicio de periodo o después de un gol.
						66/"Recovery"	Recuoeración que a su vez es un pase
	outcome	object	Id/name	Integer/text	Id/nombre del resultado del pase	9/"Incomplete"	La pelota no llega a un compañero y sigue en juego
						74/"Injury Clearence"	Tirar el balón fuera por una lesión
						75/"Out"	Pelota sale fuera del campo
						76/"Pass OFFside"	El balón lo recoge un jugador en fuera de juego
						77/"Unknown"	Se paró el juego en medio del pase.
	technique	object	Id/name	Integer/text	Id/nombre de la técnica del pase	104/"Inswinging"	Efecto hacia portería en los corners
						105/"Outswinging"	Efecto lejos de portería en los corners
						107/"Straight"	Corner sin efecto
						108/"Through Ball"	Pase cruza la línea de defensa
Player Off	permanent	boolean			Añadir si el jugador abandona el campo por lesión u no entra y no hay cambios.	TRUE	Pressure
Pressure	counterpress	boolean			Si la acción se realizó antes de 5 segundos que se perdió el balón	TRUE	
Shot	Key_pass_id	uuid				Ej. "57726e09-9ba1-4388-8c40-faf57e5b8d1d"	
	End_location	Array(x,y) or (x,y,z)				Ej. (120,50) or (120,32,2)	Información más adelante
	Aerial_won	boolean			Añadir si el disparo es	TRUE	



					en un duelo aéreo		
Follows_dribble	boolean				Añadir si el disparo es tras regate	TRUE	
First_time	boolean				Si es al primer toque	TRUE	
Freeze_array	array				Información sobre los jugadores cercanos al disparo	Data from JSON	
Open_goal	boolean				Añadir si fue a portería vacía	TRUE	
Statsbomb_xg	numeric				Valor de xG del disparo según el modelo statsBomb	Ej. 0.09	
deflected	boolean				Añadir si el disparo fue desviado	TRUE	
technique	object	Id/name	Integer/text	Id/nombre de la técnica usada	89/"Backheel"	Tacón	
					90/"Diving Header"	Remate de cabeza en plancha	
					91/"Half Volley"	Remate desde el suelo después de un rebote	
					92/"Lob"	Disparo alto para evitar la oposición de un jugador o portero	
					93/"Normal"		
					94/"Overhead Kick"	El jugador está de espaldas a la portería cuando remata el balón	
					95/"Volley"	El balón no toca el suelo antes de rematar.	
Body_part	Object	Id/name	Id/nombre de la parte del cuerpo con la que se remata		37/"Head"		
					38/"Left Foot"		
					70/"Other"		
					40/"Right Foot"		
Type	Object	Id/name	Integer/text	Id/nombre del tipo de disparo que fue	61/"Corner"	Corner directo	
					62/"Free Kick"	Tiro libre	
					87/"Open Play"	En juego normal	
					88/"Penalty"		
					65/"Kicks Off"	Tiro directo de saque inicial.	
outcome	object	Id/name	Integer/text	Id/nombre del resultado del disparo	96/"Blocked"	Disparo parado por un defensor	
					97/"Goal"		
					98/"Off T"	Disparo fuera	
					99/"Post"	Poste	

						100/"Saved"	Parada del portero
						101/"Wayward"	Disparo infoensivo que no llega a portería o el jugador no imacta con el balón
						115/"Saved Off T"	Parado por el portero pero no iba a portería.
						116/"Saved To Post"	Portero detiene el disparo pero rebota en el marco de la portería.
Substitución	Replacement	object	Id/name	Integer/text	Id/nombre del jugador que sale al campo	Ej. 3049/"Aaron Ramsey"	
	Outcome	object	Id//name	Integer/text	Id/nombre atributo de la razón del cambio	102/"Injury" 103/"Tactical"	