TRABAJO FIN DE GRADO

MODELO DE PROBABILIDAD PARA EL ANÁLISIS DE RESULTADOS DE LA LIGA ESPAÑOLA

TRABAJO FIN DE GRADO PARA LA OBTENCIÓN DEL TÍTULO DE GRADUADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

FEBRERO 2024

Pablo Legerén Somolinos

DIRECTOR DEL TRABAJO FIN DE GRADO:

Jesús Juan Ruiz

INDUSTRIALES
ETSII | UPM

POLITÉCNICA

El heroísmo del trabajo está en

"acabar" cada tarea.

AGRADECIMIENTOS.

La finalización de este Trabajo de Fin de Grado pone fin a mis estudios en el Grado de Ingeniería en Tecnologías Industriales. En los cuatro años estudiando esta carrera, he aprendido mucho sobre la resiliencia, el compañerismo y la constancia. Sin lugar a dudas, ha sido un trayecto duro pero bonito en el que me han acompañado un montón de personas que han sido una inspiración y que ha sido una suerte tener a mi lado.

En primer lugar, quiero expresar mi más sincero agradecimiento a mi tutor, Jesús Juan, por su valiosa ayuda y orientación durante la elaboración de este trabajo. Sus consejos y comentarios han sido de gran utilidad para mí, y me han permitido mejorar significativamente mi trabajo.

También quiero agradecer a mis amigos y familiares por su apoyo durante todo este proceso. Sus palabras de ánimo y consejos me han ayudado a superar los retos y a seguir adelante.

En especial, quiero dar las gracias a mis padres y a mis hermanos, por su apoyo incondicional. Siempre han estado ahí para mí, y me han ayudado a alcanzar todas las metas que me propuesto.

RESUMEN

Este proyecto nace del interés del alumno por el análisis de los datos combinado con la oportunidad de una gran fuente de datos como es el deporte rey, el fútbol. Actualmente en el mundo de este deporte, está tomando cada vez más importancia el análisis de las estadísticas tanto del propio equipo como del rival, aunque la cantidad de datos todavía está a años luz de otros deportes como el hockey o el fútbol americano, y más aún de otros sectores como pueden ser la banca o las finanzas.

En este Trabajo de Fin de Grado se trata de conseguir, con la ayuda del programa RStudio, un modelo con la eficacia, simplicidad, sensibilidad, estabilidad y objetividad como para poder predecir al principio de una temporada de fútbol el campeón de la misma y otros objetivos que los equipos implicados se podrían marcar como conseguir la permanencia o la clasificación para las competiciones europeas.

El modelo usado toma ideas de modelos propuestos más antiguos como el modelo de Maher o el de Dixon-Coles. En nuestro caso se basa en el cálculo de unos valores propios de cada uno de los enfrentamientos que suceden a lo largo de la temporada, teniendo en cuenta tanto los equipos que se enfrentan como cuál actúa de local y cuál de visitante.

Estos valores calculados, las λ, serán el parámetro utilizado para mediante una distribución de Poisson, poder estimar el número de goles que cada uno de los equipos implicados en el enfrentamiento conseguirá anotar. Estas λ tendrán una componente que dependerá del propio equipo, otra componente que dependerá del rival y otra componente que dependerá del resto de equipos de la temporada; a su vez, estas tres variables dependen también de la localidad o no de cada uno de los equipos, es decir, para dos supuestos equipos ‘i’ y ‘j’, el valor de λ no será el mismo para *i contra j* que para *j contra i.*

Para poder realizar los cálculos necesarios se tomarán los datos de todos los enfrentamientos de la Liga Española desde la temporada 1929-1930; estos datos procederán de distintas fuentes como pueden ser un paquete del propio RStudio, una tabla de un usuario de una página web como la obtención manual de los mismos. Estos datos serán utilizados después de un proceso de recogida, limpieza, preparación, unificación de los distintos formatos y organización de estos.

Una vez calculados los valores necesarios para poder aplicar la distribución de Poisson necesaria para simular los enfrentamientos, se iterará mil veces consiguiendo así un alto número de escenarios que acercarán los resultados un poco más a la realidad. Obtenidos los resultados se valorará si estos, como se ha supuesto durante este resumen, podrían considerarse como que reflejan la realidad o no.

Finalmente se comentarán posibles utilidades y futuras líneas de aplicación siguiendo el mismo modelo.

Palabras clave:

Local, visitante, modelo probabilístico, distribución de Poisson, Liga Española, goles, temporada, parámetro λ, datos, probabilidad, clasificación.

Códigos UNESCO:

120302 – Lenguajes Algorítmicos.

120323 – Lenguajes de programación.

120901 - Estadística analítica.

120902 - Cálculo en Estadística.

120903 - Análisis de datos.

120907 – Teoría de la Distribución y Probabilidad.

ÍNDICE

1. INTRODUCCIÓN Y OBJETIVOS.

El fútbol, como fenómeno social y deporte global, ha capturado la atención de millones de personas en todo el mundo. Dentro de este fascinante universo, el análisis de resultados se ha convertido en un componente crucial para comprender los factores subyacentes que influyen en el rendimiento de los equipos. En este contexto, el presente Trabajo de Fin de Grado se sumerge en el emocionante campo de los modelos de probabilidad aplicados al análisis de resultados de la Liga Española, una de las competiciones de fútbol más seguidas y competitivas del planeta.

La elección de este tema nace sobre todo de la creciente necesidad de desarrollar herramientas analíticas que no solo describan los resultados de los partidos, sino que también permitan anticiparlos con un grado significativo de precisión. La Liga Española, reconocida por su intensidad y calidad técnica, proporciona un escenario ideal para explorar la eficacia de modelos de probabilidad en el ámbito deportivo. ¿Es posible prever los resultados de los enfrentamientos basándonos en variables específicas?

El objetivo primordial de esta investigación es desarrollar un modelo de probabilidad que no solo arroje luz sobre esta pregunta, sino que también proporcione una herramienta valiosa para analistas, entrenadores y aficionados interesados en el rendimiento de los equipos de la Liga Española. Al abordar esta tarea, nos sumergimos en un terreno que fusiona la emoción del deporte con la precisión de los métodos estadísticos y probabilísticos.

Este estudio se presenta en un momento en el que la analítica deportiva está en auge, y la capacidad de prever los resultados de los partidos se ha convertido en un aspecto estratégico para diversos actores en el ámbito del fútbol. Al avanzar en esta dirección, no solo se contribuye al cuerpo de conocimiento existente en el análisis deportivo, sino que también se explora el potencial de aplicación de la teoría de la probabilidad en un contexto dinámico y altamente impredecible.

A lo largo de las páginas que siguen, exploraremos la teoría detrás de los modelos de probabilidad, analizaremos datos relevantes de la Liga Española y desarrollaremos un marco metodológico sólido para la construcción y validación de nuestro modelo. Este trabajo busca, en última instancia, ofrecer una perspectiva innovadora y aplicable a uno de los deportes más apasionantes del mundo.

2. CONTEXTO FUTBOLÍSTICO.

2.1. FÚTBOL EN ESPAÑA.

El fútbol es uno de los deportes más jugados en todo el mundo. Según una encuesta realizada por la FIFA, Big Count 2006, existen más de 265 millones de personas que estén federadas. Esto representa casi el 5% de la población mundial y no incluye a aquellas personas que lo practican sin estar inscritos en ninguna organización, por mera diversión o por hacer deporte.

La FIFA (Federación Internacional de Fútbol Asociación) cuenta con 211 federaciones en las seis confederaciones que existen en el mundo, estas federaciones agrupan más de 220 ligas por todo el mundo

Una de estas ligas prestigiosas y seguidas del mundo está en España. Nuestra liga está considerada una de las cinco grandes ligas junto con la Premier League inglesa, la Bundesliga alemana, la Ligue 1 francesa y la Serie A italiana. Es el segundo campeonato nacional con más títulos en competiciones internacionales oficiales a nivel mundial y el primero en Europa al sumar un total de 83 títulos. España tiene un sistema de ligas unidas por descensos y ascensos, con una liga en lo más alto: La Liga EA Sports. La segunda división se conoce como La Liga Hypermotion. Estas ligas, como ya he mencionado están unidas por un sistema de ascensos y descensos; en el caso de la primera división, La Liga EA Sports, al final de cada temporada los tres equipos que menos puntos hayan conseguido descienden de categoría a la Hypermotion y los tres primeros de esta, ascienden a Primera.

Algunos de los equipos más destacados de La Liga son el Real Madrid, el Barcelona, el Atlético de Madrid o alguno histórico como el Deportivo de La Coruña.

El Real Madrid y el Barcelona son dos de los clubes más exitosos de la competición durante los últimos años, tanto es así, que desde el año 2000 solo ha habido otros tres equipos que hayan sido campeones de Liga, siendo nueve en total desde el comienzo de La Liga. El clásico español, el partido que enfrenta a estos dos rivales es uno de los partidos más seguidos y emocionantes a nivel mundial.

(A partir de ahora, para referirnos a la primera división, La Liga EA Sports, utilizaremos el término La Liga). La temporada liguera empieza a mediados de agosto y termina a finales de mayo. Cada uno de los veinte equipos que la componen disputan un total de 38 jornadas enfrentándose así a todos los equipos dos veces, una de local y otra de visitante.

2.2. CLASIFICACIÓN DE LA LIGA.

Para poder entender los resultados obtenidos en este trabajo es fundamental comprender cómo funciona la clasificación de La Liga. Los tres posibles resultados de un partido son victoria, empate o derrota. Se le llama victoria a la situación en la que los goles anotados son más que los encajados, empate cuando estas dos cantidades son iguales y derrota cuando encajas más goles de los que anotas.

El reparto de puntos está relacionado con el resultado final del partido, no depende de la diferencia de goles entre equipo local y visitante, un 1-0 y un 6-1 valen lo mismo. Con una victoria el equipo ganador suma 3 puntos, con un empate los dos equipos suman 1 punto y con una derrota no se suma ningún punto.

Al final de La Liga se ordenan los equipos en función de sus puntos en orden descendente, en caso de empate a puntos, el siguiente parámetro que se tiene en cuenta para ordenar los equipos es *gol-average* particular de los equipos empatados, es decir, el saldo de los dos enfrentamientos.

Al final de la temporada el equipo con más puntos se corona campeón de La Liga y los tres equipos con menos puntos descienden a Segunda División. Esto hace que, teniendo en cuenta el sistema de puntuación, cada partido sea igual de importante que todos los demás, de ahí la frase de: "Las Ligas se ganan contra los equipos pequeños". [1]

		P.J.	P.G.	P.E.	P.P.	G.F.	G.C.	Ptos
1	Deportivo	38	21	21	11	66	44	**69**
2	Barcelona	38	19	19	12	70	46	**64**
3	Valencia	38	18	18	10	59	39	**64**
4	Zaragoza	38	16	16	7	60	40	**63**
5	Real Madrid	38	16	16	8	58	48	**62**
6	Alavés	38	17	17	11	41	37	**61**
7	Celta	38	15	15	15	45	43	**53**
8	Valladolid	38	14	14	13	36	44	**53**
9	Rayo Vallecano	38	15	15	16	51	53	**52**
10	Mallorca	38	14	14	15	52	45	**51**
11	Athletic de Bilbao	38	12	12	12	47	57	**50**
12	Málaga	38	11	11	12	55	50	**48**
13	Real Sociedad	38	11	11	13	42	49	**47**
14	Espanyol	38	12	12	15	51	48	**47**
15	Racing de Santander	38	10	10	12	52	50	**46**
16	Oviedo	38	11	11	15	44	60	**45**
17	Numancia	38	11	11	15	47	59	**45**
18	Betis	38	11	11	18	33	56	**42**
19	Atlético de Madrid	38	9	9	18	48	64	**38**
20	Sevilla	38	5	5	20	42	67	**28**

Ilustración 1 - Ejemplo de clasificación de la Liga Española de la temporada 1999-2000.

2.3. ESTADÍSTICAS EN EL FÚTBOL.

Hace unos años poca relevancia tenían las estadísticas en el fútbol, de hecho, Steve McLaren, quien fue un entrenador de la selección inglesa, llegó a decir que "las estadísticas no significan nada para mí".

Durante estos últimos años el discurso ha cambiado radicalmente, hoy en día muchas son las referencias que indican la importancia de las estadísticas y del tratamiento de datos como dijo Cristiano Ronaldo en una gala de premios: "Las estadísticas no engañan" [5].

Solo hay que ver como ahora en cada equipo de La Liga se han implementado equipos de Data Science encargados de recopilar, filtrar y tratar cada vez cantidades más grandes de datos tanto de los rivales, como de su propio equipo, para poder estudiar a sus rivales y potenciar al máximo a sus jugadores.

El Big Data en el fútbol ha revolucionado la preparación de los equipos, tanto física como tácticamente. Aunque el fútbol sea uno de los deportes más complicados de parametrizar por

la multitud de interacciones y de factores implicados y sobre todo por la dependencia de cada jugada con todas las anteriores y la infinidad de resultados posibles de cada una de ellas. En otros deportes como puede ser el béisbol es más fácil de predecir ya que cada bateo es independiente de los anteriores. El problema no es tanto conseguir la información sino filtrarla, se recogen una cantidad enorme de datos de los que se puede obtener muchísima información relevante la pregunta a responder es "¿Cuáles son los factores que ganan partidos?".

Aunque la cantidad de información recopilada esté creciendo exponencialmente, aún no iguala la cantidad de información disponible en otros deportes como el béisbol o el fútbol americano, lo que nos hace pensar que todavía hay mucho camino por recorrer y mucho margen de mejora.

2.4. GOLES.

Como ya hemos mencionado el objetivo del fútbol es simple, anotar más goles que tu rival. Cada equipo, formado por diez jugadores de campo y un portero, compite contra otro durante dos partes de cuarenta y cinco minutos por conseguirlo. Se considera gol cuando la pelota sobrepasa la línea de fondo dentro del arco.

Conseguir anotar goles no es tarea sencilla, los principales encargados son los delanteros, que son los jugadores que ocupan las posiciones más atacantes. Al máximo goleador de la temporada se le concede un premio individual, el premio Pichichi, en memoria de un histórico jugador del Athletic Club de Bilbao, Rafael Moreno "Pichichi". Algunos de los jugadores más famosos que han ganado este trofeo son Leo Messi, Cristiano Ronaldo o Karim Benzema.

Y al igual que existe un "encargado" de anotar goles, también existe un responsable de evitar que esto ocurra, este es el portero. Es el único de los 11 jugadores que no es considerado jugador de campo y es el único que puede jugar con las manos. El análogo al premio Pichichi para los porteros es el premio Zamora, en memoria también de un histórico portero Ricardo Zamora. Algunos de los galardonados con este premio más famosos son porteros como Iker Casillas, Víctor Valdés o Jan Oblak

2.5. MODELIZAR RESULTADOS DE FÚTBOL.

Se podría decir que existen dos tipos de modelos de modelización de resultados de fútbol. El primero, utilizado por la mayoría de los estadísticos, trata de modelar el número de goles tanto anotados como encajados pudiendo calcular, indirectamente, la probabilidad de cada resultado del partido. El segundo enfoque, desarrollado por econometristas aplicados, modela directamente los resultados de ganar-perder-empatar mediante modelos de regresión de elección discreta como el probit ordenado o logit.

Podría decirse que los objetivos del segundo tipo se alcanzan dentro de los del primero, pero no funcionaría al revés; partiendo desde el resultado del partido no podemos determinar el

número de goles que se marcan. También habría que tener en cuenta que, como ya se ha mencionado antes, los puntos obtenidos después del partido no dependen de los goles que hayas anotado sino de si has ganado (+3), perdido o empatado (+1).

En este caso adoptaremos el primer enfoque tratando, para cada partido, de poder determinar el resultado final, y así poder adjudicar los puntos correspondientes y construir la consecuente clasificación.

3. PREDICCIÓN DE RESULTADOS.

Durante mucho tiempo se ha intentado construir un modelo que sea capaz de predecir el resultado de un partido de fútbol, pero, como hemos comentado anteriormente, la infinidad de posibilidades y la dependencia entre las acciones hace que esto no sea nada fácil. En un partido de fútbol son infinidad los factores que influyen en el resultado: forma física jugadores, estado de ánimo, cambio de entrenador, climatología… En este trabajo se ha intentado reducir este número de variables hasta un mínimo que nos permite conseguir cierto parecido con la realidad.

El primer paso que habría que dar es comprobar si los goles se reparten de una forma lógica o son sucesos aleatorios, si siguen algún tipo de probabilidad. Observando la distribución de estos se podría decir que se asemeja bastante a una distribución de Poisson. [2]

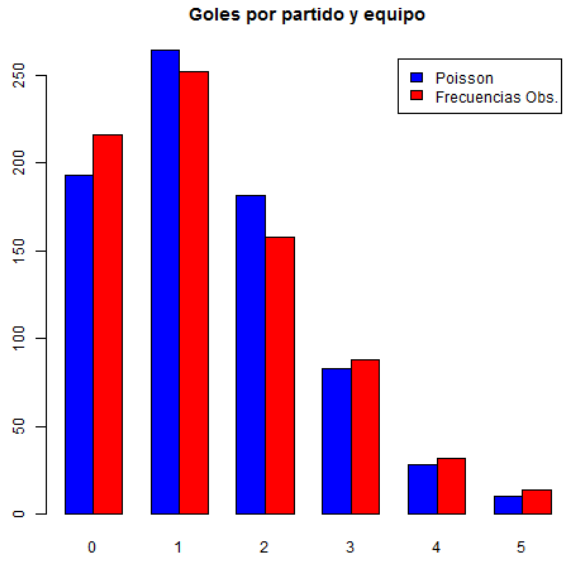

Ilustración 2 - Comparación de los goles obtenidos en una temporada de fútbol con los calculados mediante una distribución de Poisson.

Observando el gráfico superior podemos decir que existe cierto parecido entre ambas distribuciones, la teórica (azul) con la λ calculada como la media de goles anotados, y la real (roja).

Por ahora supondremos que esto se cumple, que la distribución se puede tratar como una distribución de Poisson, no obstante, más adelante comprobaremos si realmente se puede o no.

3.1. DISTRIBUCIÓN DE POISSON.

La distribución de Poisson es un modelo estadístico discreto que describe la probabilidad de que ocurra un número específico de eventos en un intervalo fijo de tiempo o espacio, bajo la premisa de que estos eventos son independientes y ocurren a una tasa constante. Fue propuesta por el matemático francés Siméon-Denis Poisson en el siglo XIX.

Esta distribución es aplicable cuando los eventos son raros y aleatorios, pero la tasa de ocurrencia es conocida. Su función de masa de probabilidad está dada por la fórmula:

$$P(X = k) = \frac{\lambda^k \cdot e^{-\lambda}}{k!} \; para \; k = 0,1,2 \ldots$$

Ecuación 1 – Función de probabilidad de una distribución de Poisson.

donde λ representa la tasa promedio de ocurrencia y k es el número de eventos que se desea evaluar.

Uno de los aspectos destacados de la distribución de Poisson es su capacidad para modelar fenómenos en los cuales la ocurrencia de eventos es infrecuente. Además, se utiliza comúnmente como una aproximación de la distribución binomial cuando el número de ensayos es grande y la probabilidad de éxito es baja.

En una distribución de Poisson la media es igual a la varianza, es decir, la desviación estándar es igual a la raíz cuadrada de la media:

$$E[x] = \lambda$$

$$[x] = \lambda \qquad \sigma^2 = \sqrt{\lambda}$$

Ecuación 2 - Propiedades estadísticas de una distribución de Poisson.

Esta distribución describe la probabilidad de que suceda un suceso en un intervalo de tiempo determinado (90 minutos) a través de una cota promedia de ocurrencia. Esta cota será el promedio de goles anotados tanto en campo local como en campo visitante. Finalmente, como el número de eventos es independiente del tiempo calculamos las probabilidades de que tanto el equipo local como el equipo visitante anoten exactamente k goles en un partido.

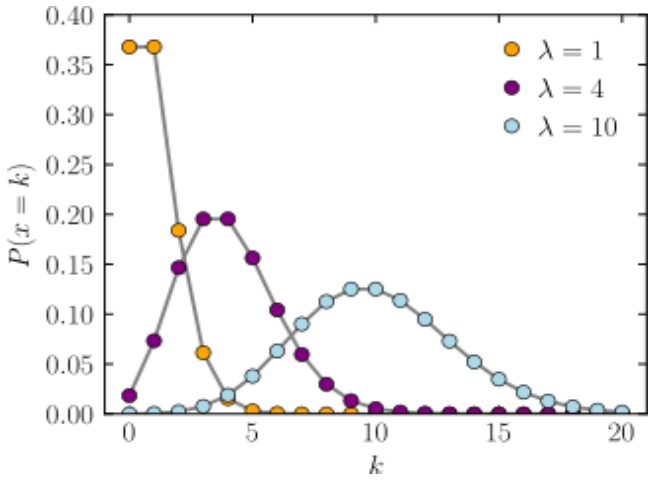

Ilustración 3 - Ejemplo de distribución de Poisson para diferentes valores de Lambda.

3.2. MODELOS ESTADÍSITICOS PREVIOS.

3.2.1. MODELO DE MAHER.

En 1982 Maher, publicó un artículo llamado "Modelling association football scores". En este artículo, propuso un modelo que utilizaba dos variables de Poisson independientes para predecir el número de goles anotados por cada equipo en un partido de fútbol.

Lo innovador de este modelo fue el cambio de lo que hasta entonces se había usado, una binomial negativa, por una distribución de Poisson, haciendo que la media de la Poisson varíe en función de variables explicativas. Las razones que presentó fueron las siguientes:

- La posesión importa en el fútbol, un equipo que tiene más el balón tiene más probabilidades de anotar gol.
- La probabilidad de que un ataque termine en gol es pequeña, pero el número de veces que un equipo tiene posesión del balón durante un partido es muy grande. Si esta probabilidad es constante y los ataques son independientes, el número de goles sigue una distribución Binomial y, en estas condiciones, la aproximación que mejor se ajusta es la Poisson.
- La media de esta Poisson variará en función de la calidad del equipo y, si se considera la distribución de todos los goles marcados por todos los equipos, se debería considerar la distribución Poisson con media variable.

3.2.2. MODELO DE DIXON Y COLES.

En 1992 Dixon y Coles proponen unas modificaciones al modelo que había propuesto Maher diez años antes. En esta nueva versión se plantean modificaciones para mejorar la precisión

en los partidos con un número bajo de goles y también para que los parámetros de ataque y de defensa sean dinámicos y tengan su base en el rendimiento reciente. En concreto propusieron dos mejoras específicas con respecto a un modelo básico de Poisson:

1. Introdujeron un término de interacción para corregir la subestimación de la frecuencia de los partidos de baja puntuación. Los autores afirmaban y se quejaban de que los partidos con resultados bajos (0-0, 1-0, 0-1 o 1-1) son inherentemente infradeclarados por un modelo básico de Poisson.
2. Aplicar un componente de decaimiento temporal para que las jornadas más recientes tengan más peso.

La función de masa de probabilidad conjunta de resultante de este modelo es:

$$P(X_1 = x_1, X_2 = x_2) = \tau_{\lambda_1,\lambda_2}(x_1, x_2)\frac{\lambda_1^{x_1} \cdot exp(-\lambda_1)}{x_1!}\frac{\lambda_2^{x_2} \cdot exp(-\lambda_2)}{x_2!}$$

Ecuación 3 - Función de masa de probabilidad conjunta del modelo de Dixon - Coles

siendo:

$$\tau_{\lambda_1,\lambda_2}(x_1, x_2) = \begin{cases} 1 - \lambda_1\lambda_2\widetilde{\omega} & si\ x_1 = x_2 = 0 \\ 1 + \lambda_1\widetilde{\omega} & s\ i\ x_1 = 0, x_2 = 1 \\ 1 + \lambda_2\widetilde{\omega} & si\ x_1 = 1,\ x_2 = 0 \\ 1 - \widetilde{\omega} & si\ x_1 = x_2 = 1 \\ 1 & otras\ opciones \end{cases}$$

3.2.3. OTROS MODELOS.

Muchos otros estudios han examinado la modelización de los resultados del fútbol, tomando como variable dependiente el número de goles (por ejemplo, Rue y Salvensen, 1998; Skinner y Freeman, 2009; Volf, 2009 o Baio y Blangiardo, 2010). Otros han considerado modelos con la diferencia de goles como variable respuesta (Karlis y Ntzoufras, 2009, Heuer y Rubner, 2009), el resultado del partido en términos de ganar, perder o empatar (por ejemplo, Dobson y Goddard, 2000; Goddard y Asimkopoulos, 2004 y Brillinger, 2006) o la puntuación de los equipos (Harville, 1997, Naim et al., 2007 y Ben-Naim y Hengartner, 2007). Muchos de estos trabajos utilizan modelos donde el número de goles sigue una distribución de Poisson o una distribución binomial negativa. También se aplican modelos mixtos y diferentes tipos de procesos estocásticos que permiten considerar efectos temporales. Además, un número significativo de artículos utiliza un enfoque bayesiano.

3.3. NUESTRO MODELO DE ESTUDIO.

A la hora de elegir un modelo de estudio para este trabajo se han tenido que estudiar estos tres modelos siguientes, cada cual más relacional que el anterior. El principal y único objetivo, y por tanto el único motivo por el que elegir uno u otro, era poder aumentar la probabilidad de acierto de cada resultado. La diferencia entre los tres modelos propuestos, al haber demostrado que la distribución más fiable a la hora de representar la cantidad de goles es la de Poisson, es la manera de encontrar el valor de λ para cada uno de ellos.

3.3.1. MODELO DE DISTRIBUCIÓN DE POISSON SIMPLE CON EQUIPOS INDEPENDIENTES.

En este primer modelo la relación entre equipos era nula, daba igual cuál era el rival, el equipo se comporta igual independientemente del equipo al que se enfrente. Como hemos comentado al principio del trabajo no podemos asumir que cada equipo es independiente cuando los dos compiten en el mismo campo. Por tanto, buscamos mejorar nuestro modelo para representar mejor nuestro conjunto de datos.

$$\lambda_A = Promedio\ de\ goles\ equipo\ A$$

$$\lambda_B = Promedio\ de\ goles\ equipo\ B$$

3.3.2. MODELO SUPONIENDO DEPENDENCIA ENTRE EQUIPOS.

El segundo modelo se basa en el primero y añade una puntuación defensiva para cada equipo. En este caso el valor de λ para cada equipo se calcula como la media entre la media de goles anotados por el equipo y la media de goles encajados por el equipo rival.

$$\lambda_A = \frac{Media\ golesF\ de\ A + Media\ golesC\ de\ B}{2}$$

$$\lambda_B = \frac{Media\ golesF\ de\ B + Media\ golesC\ de\ A}{2}$$

3.3.3. MODELO DE DISTRIBUCIÓN DE POISSON MULTIVARIABLE CON FACTOR LOCAL/VISITANTE.

En este último modelo hemos tenido en cuenta, partiendo de la idea del segundo, el factor de jugar de local o de visitante. Este último es el modelo elegido para el resto del trabajo.

Solo hay que ver los resultados que se han obtenido a lo largo de la historia para comprobar que la probabilidad de ganar en casa no es la misma que la de ganar fuera. De hecho para corroborar esto, se han calculado el porcentaje de victorias en las que gana el equipo local (H), frente a las que gana el equipo visitante (A) frente a las que quedaron en empate en los resultados ligueros desde la temporada 1995-96 ya que así se abarcan casi 30 años de competición con más de 10800 partidos, un número que se ha considerado significativo.

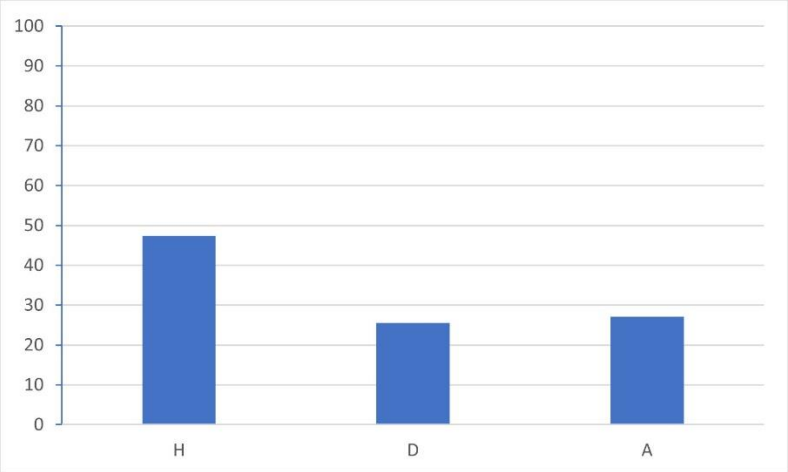

Ilustración 4 - Comparación de los porcentajes de los posibles resultados en la liga española desde la temporada 1995-96

Este gráfico se ha realizado con una base de datos que más adelante se podrá ver. En el gráfico se puede observar como la probabilidad de ganar en casa es casi dos veces más grande que la de ganar fuera de ella, un 47.36% frente a un 27.08%, quedando solamente un 25.56% de empatar. Frente a estos resultados solo queda la opción de tener en cuenta el factor local/visitante a la hora de predecir los resultados.

Trataremos de definir las variables explicativas calculándolas a partir de las estadísticas previas de cada equipo. Estas variables tendrán en cuenta el rendimiento de un equipo tanto en campo rival como en campo visitante, definiendo así para cada equipo cuatro variables significativas:

- CAC: Capacidad Atacante en Casa (de local).
- CDC: Capacidad Defensiva en Casa (de local).
- CAF: Capacidad Atacante Fuera (de visitante).
- CDF: Capacidad Defensiva Fuera (de visitante).

Además de estas cuatro variables propias de cada equipo necesitaremos también conocer el valor de la media de goles anotados tanto en casa (GMC) como fuera (GMF), los cuales serán únicos para todos los equipos.

1. **Capacidad Atacante (CA):** la Capacidad Atacante es la encargada de medir el rendimiento ofensivo de cada equipo. Esta se ha definido como la relación entre la fuerza atacante de cada equipo como local (CAC) en relación con el resto de los equipos. Del mismo modo se calculará el rendimiento fuera (CAF). Se define la fuerza atacante como la relación entre los goles que mete un equipo, dentro o fuera de casa, con los que mete en promedio el resto de los equipos de la liga, dentro o fuera de casa respectivamente. La relación entre la eficacia del ataque y el valor de la CA es proporcional, esto es, que un buen ataque implica un alto valor de CA; y al contrario, un equipo que mete pocos goles con respecto a la media de los equipos tendrá un CA bajo.

$$CAC_i = \frac{Promedio\ de\ goles\ metidos\ en\ casa\ por\ el\ equipo\ i}{Promedio\ de\ goles\ metidos\ por\ los\ equipos\ en\ casa}$$

$$CAF_i = \frac{Promedio\ de\ goles\ metidos\ fuera\ por\ el\ equipo\ i}{Promedio\ de\ goles\ metidos\ por\ los\ equipos\ fuera}$$

2. **Capacidad Defensiva (CD):** la Capacidad Defensiva es la encargada de medir el rendimiento defensivo de cada equipo. Este se ha calculado de la misma forma que la CA, pero en lugar de tener en cuenta los goles metidos se tienen en cuenta los goles encajados. Una CD alta implicaría que tu equipo encaja más goles que la media de los demás equipos, por tanto, una defensa débil supone un alto valor de CD.

$$CDC_i = \frac{Promedio\ de\ goles\ encajados\ en\ casa\ por\ el\ equipo\ i}{Promedio\ de\ goles\ encajados\ por\ los\ equipos\ en\ casa}$$

$$CDF_i = \frac{Promedio\ de\ goles\ encajados\ fuera\ por\ el\ equipo\ i}{Promedio\ de\ goles\ encajados\ por\ los\ equipos\ fuera}$$

Como ya hemos mencionado la cantidad de goles anotados se puede asemejar con una distribución de Poisson; ahora el siguiente paso sería definir esa λ del modelo.

Conociendo los valores de CAC y CDC para el equipo local; CAF y CDF para el visitante; y los valores de GMC y GMF para la temporada podemos calcular un valor de λ con el que trabajaremos de aquí en adelante.

Este valor de λ se calcula con una multiplicación de estos tres valores (CA,CD,GM) quedando las expresiones para cada equipo (equipo L, equipo V):

$$\lambda_L = CAC_L \cdot CDF_V \cdot GMC$$

$$\lambda_V = CAF_V \cdot CDC_L \cdot GMF$$

Ecuación 4 - Cálculo de los valores de lambda, local y visitante, para cada equipo, local y visitante respectivamente

De esta forma, el valor de la λ dependerá de los dos equipos que se enfrentan en cada partido, como del resto de equipos de la temporada (GMC, GMF); a la vez que también depende de cúal de los dos equipos juega de local o de visitante.

4. APLICACIÓN DEL MODELO.

4.1. ENTORNO.

Una vez tenemos el modelo definido, el modelo de distribución de Poisson multivariable con factor local/visitante, pasamos a probar su eficacia y a comprobar si realmente representa de forma fidedigna la realidad.

Todo el desarrollo se realizará en un entorno virtual del RStudio, ya que conocemos la forma de trabajar y para este tipo de proyectos en una de las opciones más interesantes, por sus servicios de cálculo y tratamiento de datos.

Para poder alcanzar todas las funcionalidades que se creen necesarias ha sido necesaria la descarga de algunos paquetes auxiliares, estos son: *tidyverse, vcd,readxl, ggplot2, dplyr, DT* y *engsoccerdata*.

Con el paquete *tidyverse, vcd, dplyr* simplificaremos el tratamiento y manejo de los *dataframes.* El paquete *DT* nos ofrece opciones de interfaz para las tablas de datos. El paquete *readxl* nos permitirá poder trabajar con archivos .xlsx. Y del paquete *engsoccerdata* es de donde sacaremos todo el registro de partidos y resultados utilizados en el estudio.

4.2. DATOS.

4.2.1. RECOGIDA DE LOS DATOS.

Como ya hemos mencionado, la mayor parte de los datos provienen del paquete *engsoccerdata*. En este paquete se encuentran los resultados de partidos de Gran Bretaña y de otros países europeos, como España, Portugal o Bélgica. Se encuentran resultados tanto de torneos ligueros como de otros torneos; y además de esto, más datos como pueden ser el número de enfrentamientos entre dos equipos, la clasificación histórica de equipos…

El problema de este paquete es que los datos solo llegan a la temporada 2020-2021, por tanto, para poder trabajar con datos más recientes hemos tenido que conseguir los datos de otras fuentes. Hasta la temporada pasada se ha usado una base de datos de un usuario de Kaggle y para esta última temporada se ha tenido que introducir a mano en una tabla del mismo formato que las anteriores.

Uno de los problemas que se han tenido que solucionar es la diferencia de formatos entre las distintas fuentes, los distintos nombres con los que se refieren a los mismos equipos, la falta de datos de algún partido o un formato incorrecto de algún dato. Por ejemplo, a la hora de referirse a equipos como el CD Alavés, alguna vez se decía CD Alavés pero otras simplemente Alavés o incluso *Alaves.*

El formato con el que vienen los datos en el paquete de *engsoccerdata* es una tabla con 12 columnas:

a. Date: día del partido.
b. Season: año de la temporada de cada jornada.
c. home: nombre del equipo local.
d. visitor: nombre del equipo visitante.
e. HT: resultado en el descanso (*Half Time*).
f. FT: resultado final (*Full Time).*
g. hgoal: goles metidos por el equipo local (*Home goal).*
h. vgoal: goles metidos por el equipo visitante (*Visitor goal).*
i. tier: nivel de la división.
j. round: distingue entre partidos de liga (league) o de playoff. (En nuestro caso al trabajar con resultados solamente ligueros, esta columna solo tendrá el valor "*league*".
k. group: grupo liguero.
l. notes: posibles notas accesorias.

	Date	Season	home	visitor	HT	FT	hgoal	vgoal	tier	round	group	notes
1	1929-02-10	1928	Arenas de Getxo	Atletico Madrid	0-2	2-3	2	3	1	league	NA	NA
2	1929-02-10	1928	Espanyol Barcelona	Real Union	1-0	3-2	3	2	1	league	NA	NA
3	1929-02-10	1928	Real Madrid	CE Europa	0-0	5-0	5	0	1	league	NA	NA
4	1929-02-10	1928	Real Sociedad	Athletic Bilbao	1-1	1-1	1	1	1	league	NA	NA
5	1929-02-12	1928	Racing Santander	FC Barcelona	0-0	0-2	0	2	1	league	NA	NA
6	1929-02-17	1928	FC Barcelona	Real Madrid	0-1	1-2	1	2	1	league	NA	NA
7	1929-02-17	1928	Athletic Bilbao	Espanyol Barcelona	3-0	9-0	9	0	1	league	NA	NA
8	1929-02-17	1928	Atletico Madrid	Real Sociedad	0-3	0-3	0	3	1	league	NA	NA
9	1929-02-17	1928	Real Union	Racing Santander	0-1	3-1	3	1	1	league	NA	NA
10	1929-02-17	1928	CE Europa	Arenas de Getxo	2-0	5-2	5	2	1	league	NA	NA
11	1929-02-24	1928	Real Sociedad	FC Barcelona	2-0	3-0	3	0	1	league	NA	NA
12	1929-02-24	1928	Espanyol Barcelona	CE Europa	3-0	3-1	3	1	1	league	NA	NA
13	1929-02-24	1928	Arenas de Getxo	Real Union	1-0	1-1	1	1	1	league	NA	NA
14	1929-02-24	1928	Racing Santander	Athletic Bilbao	0-2	0-4	0	4	1	league	NA	NA
15	1929-02-24	1928	Real Madrid	Atletico Madrid	1-1	2-1	2	1	1	league	NA	NA
16	1929-03-03	1928	Real Union	Atletico Madrid	1-1	1-2	1	2	1	league	NA	NA

Ilustración 5 - Fragmento del dataframe obtenido desde el paquete de RStudio engsoccerdata.

De este dataset se tienen 26195 registros, es decir, partidos desde el 10 de febrero de 1929, con un primer registro de un Arenas de Getxo – Atlético de Madrid, con resultado final 2-3; hasta el 22 de mayo de 2022 con un registro de un Elche CF – Getafe CF, con un resultado final de 3-1.

Para la temporada 2021-2022 en adelante, se ha utilizado otra fuente como ya se ha mencionado, una base de datos de un usuario de la página web Kaggle El formato de estos datos era distinto, tenía 10 columnas con diferente información que lo anterior. [3]

Season	Date	HomeTeam	AwayTeam	FTHG	FTAG	FTR	HTHG	HTAG	HTR
2021-22	13/08/2021	Valencia	Getafe	1	0	H	1	0	H
2021-22	14/08/2021	Cadiz	Levante	1	1	D	0	1	A
2021-22	14/08/2021	Mallorca	Betis	1	1	D	1	0	H
2021-22	14/08/2021	Alaves	Real Madrid	1	4	A	0	0	D
2021-22	14/08/2021	Osasuna	Espanol	0	0	D	0	0	D
2021-22	15/08/2021	Celta	Ath Madrid	1	2	A	0	1	A
2021-22	15/08/2021	Barcelona	Sociedad	4	2	H	2	0	H
2021-22	15/08/2021	Sevilla	Vallecano	3	0	H	1	0	H
2021-22	16/08/2021	Villarreal	Granada	0	0	D	0	0	D
2021-22	16/08/2021	Elche	Ath Bilbao	0	0	D	0	0	D
2021-22	20/08/2021	Betis	Cadiz	1	1	D	1	1	D
2021-22	21/08/2021	Alaves	Mallorca	0	1	A	0	0	D
2021-22	21/08/2021	Espanol	Villarreal	0	0	D	0	0	D
2021-22	21/08/2021	Granada	Valencia	1	1	D	1	0	H
2021-22	21/08/2021	Ath Bilbao	Barcelona	1	1	D	0	0	D
2021-22	22/08/2021	Sociedad	Vallecano	1	0	H	0	0	D

Ilustración 6 - Fragmento del dataframe obtenido desde la web Kaggle.

En este caso ofrece información similar a la anterior como pueden ser la temporada (Season), la fecha (Date), el equipo visitante y el local (HomeTeam y AwayTeam), y también, tanto para el descanso como para el final del partido ofrece los goles de cada equipo: FTHG, FTAG, FTR para el final del partido; y HTHG, HTAG, HTR para el descanso, teniendo la columnas HTR y FTR tres posibles valores: H, gana equipo local, V, gana equipo visitante, o D, empate.

Estos datos, en formato Excel, se combinan manualmente con los resultados de la actual temporada en un archivo que se llama “ultimas2temporadas.xlsx”.

4.2.2 LIMPIEZA Y PREPARACIÓN DE LOS DATOS.

Como ya hemos mencionado, el dato con el que trabajamos tenía algún problema de formato así que unificarlo todo es el primer paso para poder trabajar con él.

Los pasos que se han seguido para poder trabajar con ellos han sido:

1. Eliminar la columna de ‘tier’ del dataset Spain, ya que al trabajar solo con los datos ligueros la información que aportaba era innecesaria.
2. Asignar un índice a los datos de Spain.
3. Importar el documento .xlsx que contiene los resultados de las dos últimas temporadas con el documento “ultimas2temporadas.xlsx”, después de un trabajo en local para reordenar las columnas, crear nuevas y unificar los nombres de los equipos para poder combinarla posteriormente con los datos de la otra fuente.
4. Convertir los datos de la columna “Date” de este archivo en datos de tipo fecha.
5. Reemplaza los valores de las columnas “HT”, “group”, “notes” del dataset por “NA”.

```
predata = spain[,-9]
rownames_to_column(predata,var = 'index')

prenueva_data = read_xlsx("ultimas2temporadas.xlsx", sheet = 'Hoja4')

prenueva_data$Date = as.Date(prenueva_data$Date)
prenueva_data$HT <- replace(prenueva_data$HT, is.na(prenueva_data$HT), NA)
prenueva_data$group <- replace(prenueva_data$group,
is.na(prenueva_data$group), NA)
prenueva_data$HT <- replace(prenueva_data$notes,
is.na(prenueva_data$notes), NA)
```

Una vez solucionado el problema de unificar los datos provenientes de diferentes fuentes el siguiente paso fue unificar las dos tablas para trabajar a partir de ahora con un único dataset y quedarnos únicamente con las columnas que aportan información relevante, facilitando así los posteriores comandos, como se muestra a continuación.

```
data1 = predata
nueva_data = prenueva_data
data2 = rbind(data1, nueva_data)

data = data2[,-c(5,9,10,11)]
```

El comando rbind permite combinar dos datasets añadiendo uno a continuación del otro, siempre que las columnas de ambos sean las mismas. En este caso nos quedamos con las columnas: Date, Season, home, visitor, FT, hgoal, vgoal; de donde podremos sacar toda la información para calcular nuestras variables

Con estos comandos lo que conseguimos es una tabla uniforme con un único formato, siendo estas las primeras filas:

	Date	Season	home	visitor	FT	hgoal	vgoal
1	1929-02-10	1928	Arenas de Getxo	Atletico Madrid	2-3	2	3
2	1929-02-10	1928	Espanyol Barcelona	Real Union	3-2	3	2
3	1929-02-10	1928	Real Madrid	CE Europa	5-0	5	0
4	1929-02-10	1928	Real Sociedad	Athletic Bilbao	1-1	1	1
5	1929-02-12	1928	Racing Santander	FC Barcelona	0-2	0	2
6	1929-02-17	1928	FC Barcelona	Real Madrid	1-2	1	2

Y estas siguientes las últimas:

	Date	Season	home	visitor	FT	hgoal	vgoal
27330	2024-03-01	2023	UD Las Palmas	CD Alaves	*NA*	*NA*	*NA*
27331	2024-03-01	2023	CA Osasuna	Villarreal CF	*NA*	*NA*	*NA*
27332	2024-03-01	2023	Rayo Vallecano	Athletic Bilbao	*NA*	*NA*	*NA*
27333	2024-03-01	2023	Real Madrid	Real Betis	*NA*	*NA*	*NA*
27334	2024-03-01	2023	Real Sociedad	Atletico Madrid	*NA*	*NA*	*NA*
27335	2024-03-01	2023	Sevilla FC	FC Barcelona	*NA*	*NA*	*NA*

Ilustración 7 - Fragmento del dataframe llamado data con el que se trabajará a partir de ahora.

En este caso se puede ver que el valor de las columnas FT, hgoal y vgoal es NA ya que estas jornadas todavía no se han disputado, sino que son las últimas de este campeonato liguero.

Con esta tabla, que llamamos *data*, será la que tomaremos como referencia a partir de ahora.

A la hora de trabajar no trabajaremos con el conjunto completo de los datos ya que a medida que pasa el tiempo el pasado va perdiendo peso. Como anunciaron Dixon y coles en su modelo, cuánto más cercana sea la información más peso tendrá a la hora de calcular las variables, por tanto habrá que reducir las observaciones.

Para ello lo que haremos será filtrar el dataframe *data* imponiendo la condición de que el valor de *data$Season* sea menor que la temporada que se quiere predecir y se añade la condición adicional de que *data$Season* también sea más pequeño que 2023 ya que para esta temporada, la actual, no se tienen todos los datos.

Esto en R quedaría así:

```
tmp = 2022
datos=data[(data$Season > tmp-9) & (data$Season < 2023),]
#datos=data[data$Season == tmp,]
datos
```

En este caso se ha elegido que la temporada que se quiere estudiar es la temporada 2022-23. Se seleccionan los datos de las nueve temporadas anteriores y se guardan en un nuevo dataframe llamado *datos*.

4.3.3. ORGANIZACIÓN DE LOS DATOS.

Como ya hemos mencionado las variables con las que trabajaremos son diferentes en el caso de que el equipo sea local o visitante, por tanto, lo que habría que hacer sería duplicar las observaciones del anterior dataframe consiguiendo así que el mismo enfrentamiento se pueda estudiar para el equipo local y para el equipo visitante.

Para ello se ejecuta el siguiente código en RStudio:

```
temp=rbind(
  datos %>% select(Season, Equipo=home, opp=visitor, GF=hgoal, GC=vgoal),
```

```
  datos %>% select(Season, Equipo=visitor, opp=home, GF=vgoal, GC=hgoal)
    )
#Duplica las observaciones, una para equipo de cada enfrentamiento.

temp$GF=as.numeric(temp$GF)
temp$GC=as.numeric(temp$GC)
temp=temp %>% mutate(dG=GF-GC)
temp
```

Además de lo que se ha comentado en el párrafo anterior, añadimos una nueva columna con la diferencia de los goles de cada partido. El dataframe *temp* sería algo así:

Season	Equipo	opp	GF	GC	dG
2014	Malaga CF	Athletic Bilbao	1	0	1
2014	Sevilla FC	Valencia CF	1	1	0
2014	Granada CF	Deportivo La Coruna	2	1	1
2014	UD Almeria	Espanyol Barcelona	1	1	0
2014	SD Eibar	Real Sociedad	1	0	1
2014	Celta Vigo	Getafe CF	3	1	2
2014	FC Barcelona	Elche CF	3	0	3
2014	Levante UD	Villarreal CF	0	2	-2
2014	Real Madrid	Cordoba CF	2	0	2
2014	Rayo Vallecano	Atletico Madrid	0	0	0
2014	Getafe CF	UD Almeria	1	0	1
2014	Valencia CF	Malaga CF	3	0	3

Ilustración 8 - Fragmento del dataframe llamado temp.

De esta forma ahora sería mucho más sencillo tener un dataframe con las observaciones de cada equipo jugando como local y las que actúa de visitante, esto se haría de la misma forma en la que se ha duplicado el dataframe *datos* pero imponiendo en cada caso la condición de que actúe o de local o de visitante.

Una vez separadas las observaciones se construye una clasificación con el formato y las reglas de la Liga Española (sumando tres puntos por victoria y uno por empate). Cada uno de estos dataframe las columnas: equipo, partidos jugados (PJ), goles a favor (GF), goles en contra (GC), diferencia de goles (DG), partidos ganados (PG), partidos empatados (PE), partidos perdidos (PP), puntos (calculados como se ha mencionado anteriormente) y dos columnas que nos ayudarán a calcular las posteriores variables GMC y GMF.

Todo esto se hará con el siguiente fragmento código:

```
#Para los partidos que se juegan en casa, solo toma las observaciones que
se tienen como local

tempL=rbind(datos %>% select(Season, Equipo=home, opp=visitor, GF=hgoal,
                             GC=vgoal))
```

```r
tempL=tempL %>% mutate(dG=GF-GC)

#En este caso selecciona solamente los partidos que cada equipo juega como
local.

LOCAL1=tempL %>%
  group_by(Equipo) %>%
      summarize(PJ=n(),
      GF=sum(GF),
      GC=sum(GC),
      DG=sum(dG),
      PG=sum(dG>0),
      PE=sum(dG==0),
      PP=sum(dG<0),
) %>%
  mutate(Puntos = (PG*3) + PE) %>%
  mutate(GMC=GF/PJ) %>%
  mutate(GRC=GC/PJ)
```

La variable *tempL* es la que almacena las observaciones de local, es un dataframe con el mismo formato que el anterior *temp*. La variable *LOCAL1* también es un dataframe, en este caso es el que contiene una clasificación de los equipos jugando como local.

Equipo	PJ	GF	GC	DG	PG	PE	PP	Puntos	GMC	GRC
FC Barcelona	190	506	148	358	144	25	21	457	2.6631579	0.7789474
Real Madrid	190	474	152	322	138	34	18	448	2.4947368	0.8000000
Atletico Madrid	190	353	119	234	134	39	17	441	1.8578947	0.6263158
Sevilla FC	190	334	176	158	119	43	28	400	1.7578947	0.9263158
Villarreal CF	190	335	197	138	100	46	44	346	1.7631579	1.0368421
Athletic Bilbao	190	274	174	100	92	51	47	327	1.4421053	0.9157895
Valencia CF	190	301	199	102	88	62	40	326	1.5842105	1.0473684
Real Sociedad	190	276	189	87	92	49	49	325	1.4526316	0.9947368
Celta Vigo	190	284	244	40	77	51	62	282	1.4947368	1.2842105
Real Betis	171	254	214	40	78	40	53	274	1.4853801	1.2514620
Espanyol Barcelona	171	213	213	0	68	50	53	254	1.2456140	1.2456140
Getafe CF	171	196	151	45	67	51	53	252	1.1461988	0.8830409

Ilustración 9 - Fragmento de la tabla construida a modo de clasificación de las últimas nueve temporadas.

En esta tabla se pueden observar los equipos ordenados por puntos. Las dos últimas columnas, correspondientes a GMC y a GRC (Goles Metidos en Casa y Goles Recibidos en

Casa) se han calculado, como se puede observar en el código, como la división de los goles metidos/recibidos en casa entre los partidos jugados.

Para el FC Barcelona serían 506 goles anotados y 148 encajados en 190 partidos quedando unos valores de 2.663 y de 0.778 para GMC y GRC, respectivamente.

Esto mismo habría que repetirlo para las observaciones de cada equipo visitante, esto es:

```
tempV=rbind(datos %>% select(Season, Equipo=visitor, opp=home, GF=vgoal,
                             GC=hgoal))
tempV=tempV %>% mutate(dG=GF-GC)

VISITANTE1=tempV %>%
     group_by(Equipo) %>%
     summarize(PJ=n(),
     GF=sum(GF),
     GC=sum(GC),
     DG=sum(dG),
     PG=sum(dG>0),
     PE=sum(dG==0),
     PP=sum(dG<0),
 ) %>%
  mutate(Puntos = (PG*3) + PE) %>%
  mutate(GMF=GF/PJ) %>%
  mutate(GRF=GC/PJ)
```

En este caso sería exactamente el mismo código, lo que podría llevar a una confusión ya que hemos visto en el dataframe temp que la diferencia de goles se calculaba como los goles de la primera columna menos los de la segunda, eso era los del local menos los del visitante.

Lo que hemos conseguido separando las observaciones en local y visitante es que ahora la primera columna corresponda al equipo visitante a pesar de que DG se calcule como DG=GF-GC.

Con todo esto el dataframe *VISITANTE1* quedaría con el mismo formato que el anterior *LOCAL1*:

Equipo	PJ	GF	GC	DG	PG	PE	PP	Puntos	GMF	GRF
FC Barcelona	190	398	176	222	116	45	29	393	2.0947368	0.9263158
Real Madrid	190	389	202	187	119	35	36	392	2.0473684	1.0631579
Atletico Madrid	190	278	177	101	94	51	45	333	1.4631579	0.9315789
Sevilla FC	190	228	254	-26	65	58	67	253	1.2000000	1.3368421
Villarreal CF	190	227	217	10	65	55	70	250	1.1947368	1.1421053
Real Sociedad	190	229	263	-34	66	47	77	245	1.2052632	1.3842105
Real Betis	171	193	246	-53	57	41	73	212	1.1286550	1.4385965
Valencia CF	190	221	276	-55	52	53	85	209	1.1631579	1.4526316
Athletic Bilbao	190	181	244	-63	49	60	81	207	0.9526316	1.2842105
Celta Vigo	190	200	295	-95	45	54	91	189	1.0526316	1.5526316
Getafe CF	171	135	255	-120	31	54	86	147	0.7894737	1.4912281
Espanyol Barcelona	171	166	287	-121	31	49	91	142	0.9707602	1.6783626

Ilustración 10 - Fragmento de la tabla construida a modo de clasficación con los resultados de los equipos como local de las últimas nueve temporadas.

4.3 COMPROBACIÓN DISTRIBUCIÓN DE POISSON.

Ya se ha comentado en el capítulo 3 (Predicción de resultados) que los goles anotados siguen una distribución de Poisson.

Ahora que tenemos los datos podremos comprobarlo y así confirmar la hipótesis que se ha hecho al principio de este trabajo y teniendo tanto los datos totales como las observaciones para los dos escenarios lo comprobaremos en las tres opciones.

Para ello habrá que calcular el valor de λ para cada uno de los tres casos. Para obtener este valor lo que se hará será calcular la media de los goles a favor de cada dataset. Esto en R se traduce en:

-Para las observaciones totales:

```
lambdaT=mean(temp$GF) #Calcula la media de los goles metidos y lo asigna a
la lambda de la distribucion de Poisson
fptemp = rpois(length(temp$GF),lambda=lambdaT) #Calcula 760 valores
aleatroios de la
#distribución de Poisson con la lambda calculada
par(mfrow = c(1,2))

barplot(table(fptemp),col = "blue",ylab = "Frecuencia", xlab = "Goles
durante la
     temporada",main = "Distribución teórica",ylim=c(0,2500)) #Saca el
gráfico de
#los goles esperados
grid(nx = NA, ny = NULL,lty = 1, col = "gray", lwd = 1)
```

```
barplot(table(temp$GF),col = "red",ylab = "Frecuencia",xlab = "Goles
durante la temporada"
     ,main="Distribución real",ylim=c(0,2500)) #Saca el gráfico de los
#goles reales
grid(nx = NA, ny = NULL,lty = 1, col = "gray", lwd = 1)
```

En este código, similar a los dos siguientes lo que se está haciendo es calcular el valor de λ y con este valor obtener el mismo número de valores aleatorios que de goles anotados para posteriormente hacer dos histogramas, uno para los valores aleatorios con la λ calculada y otro con los valores de los goles a favor obtenidos.

En esta gráfica se puede ver la gran semejanza entre los dos conjuntos de datos algo que anima a no rechazar la hipótesis de distribución de Poisson.

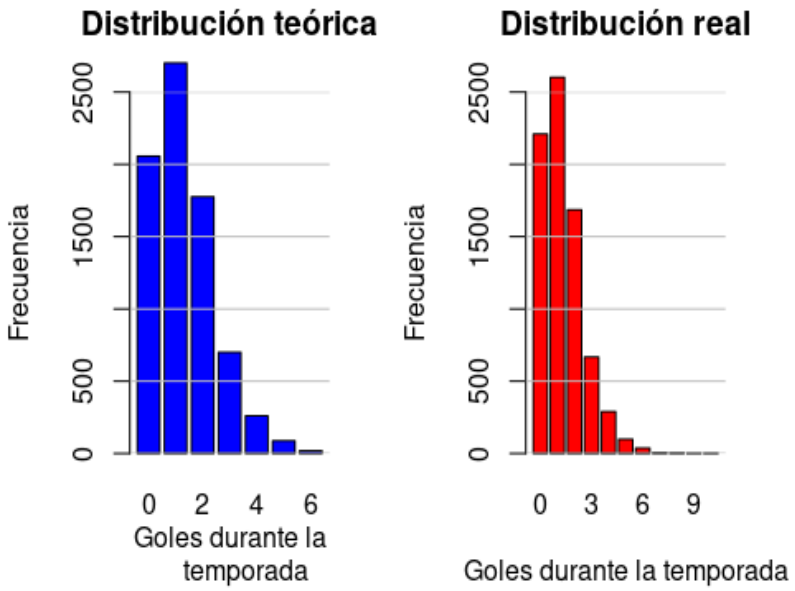

Ilustración 11 - Comparación de la distribución teórica de los goles con la distribución real de las últimas nueve temporadas.

Lo siguiente sería realizar lo mismo tanto para las observaciones en casa como fuera de casa:

- Local:

```
lambdaL=mean(tempL$GF) #Calcula la lambda de la ditrib. de Poisson
calculando
#la media de los goles a favor

fptempL=rpois(length(tempL$GF),lambda=lambdaL) #Calcula 380 valores
aleatroios de la
#distribución de Poisson con la lambda calculada
```

```
par(mfrow = c(1,2))

barplot(table(fptempL),col = "blue",ylab = "Frecuencia",xlab = "Goles en
casa",
     main="Distribución teórica",ylim=c(0,1250))

# Saca el gráfico de los goles esperados
grid(nx = NA, ny = NULL,lty = 1, col = "gray", lwd = 1)

barplot(table(tempL$GF),col = "red",ylab = "Frecuencia",xlab = "Goles en
casa",
     main = "Distribución real",ylim=c(0,1250))

#Saca el gráfico de los goles reales
grid(nx = NA, ny = NULL,lty = 1, col = "gray", lwd = 1)
```

En este caso los gráficos también muestran gran parecido:

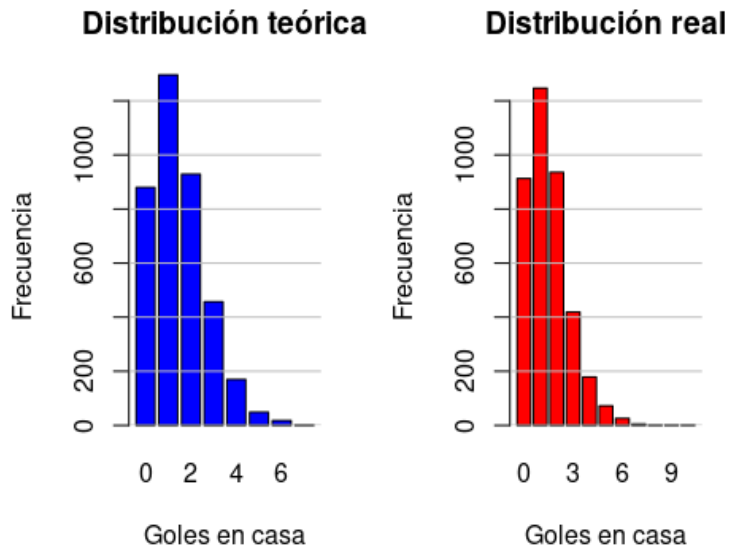

Ilustración 12 - Comparación de la distribución teórica de los goles con la real de los goles anotados de los equipos como local.

- Visitante:

```
lambdaV=mean(tempV$GF) #Hacer el histograma y compararlo con el histograma
de debajo
fptempV=rpois(length(tempV$GF),lambda=lambdaV)

par(mfrow = c(1,2))
```

```
barplot(table(fptempV),col = "blue",ylab = "Frecuencia",
        xlab = "Goles fuera de casa",main="Distribución
teórica",ylim=c(0,1250))
grid(nx = NA, ny = NULL,lty = 1, col = "gray", lwd = 1)

barplot(table(tempV$GF), col ="red", ylab = "Frecuencia",
        xlab = "Goles fuera de casa",main = "Distribución
real",ylim=c(0,1250))
grid(nx = NA, ny = NULL,lty = 1, col = "gray", lwd = 1)
```

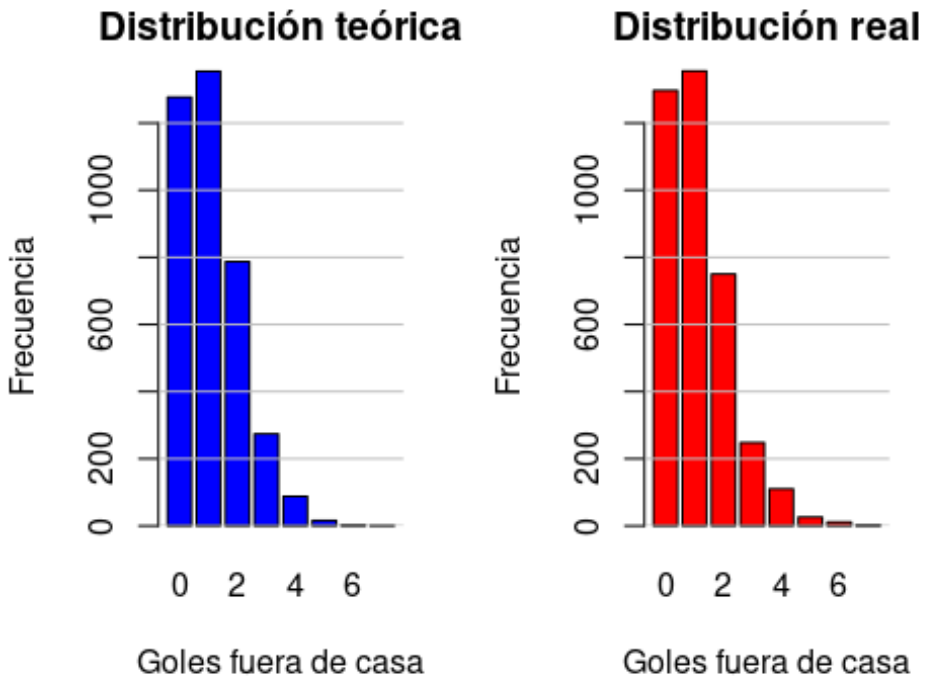

Ilustración 13 -Comparación de la distribución teórica de los goles con la real de los goles anotados de los equipos como visitante.

Todos estos gráficos reflejan la gran similitud que estamos suponiendo, pero una simple comparación visual no valdría para afirmar que sigue una distribución de Poisson. Para comprobarlo firmemente lo más sensato es realizar un contraste de hipótesis.

Pero para poder afirmar que realmente la distribución real se podría asemejar a una distribución de Poisson realizar un contraste de hipótesis:

$$H_0: la\ distribución\ de\ goles\ sigue\ una\ distribución\ de\ Poisson$$

$$H_1: la\ distribución\ de\ goles\ NO\ sigue\ una\ distribución\ de\ Poisson$$

En el siguiente espacio de código podemos ver cómo se ha realizado el contraste en RStudio.

```
#Para realizar el contraste y ver si se puede considerar Poisson.
a=rep(0,max(temp$GF))
for (i in (1:max(temp$GF+1))) {
  a[i]=sum(temp$GF==i-1)
  i=i+1
}

lambdaT=mean(temp$GF)
frec_esperada=dpois(0:max(temp$GF),lambdaT)
frecuencia_esperada=frec_esperada*length(temp$GF)
frecuencia_esperada

w=rbind(frecuencia_esperada,a)

chisq.test(w,simulate.p.value = TRUE)

##
##  Pearson's Chi-squared test with simulated p-value (based on 2000
##  replicates)
##
## data:  w
## X-squared = 39.897, df = NA, p-value = 0.08746
```

Respecto a la función *chisq.value*, el atributo '`simulate.p.value = TRUE`' se añade ya que el número de observaciones no es tan elevado como para realizar un contraste normal. Esto lo que consigue es que los p-valores se calculen mediante una simulación Monte Carlo.

En este contraste lo que se está haciendo es comparar los resultados obtenidos a lo largo de las temporadas de estudio, con las frecuencias esperadas teóricamente. Recordemos que en estos contrastes un p-valor menor que el valor de nivel de significación nos lleva a poder concluir que la hipótesis nula queda rechazada y aceptada, por tanto, la alternativa. En este caso la hipótesis nula es la que nos asegura que los datos siguen una distribución de Poisson frente a la alternativa que propone lo contrario; como podemos ver el pvalor = 0.08046, mayor que $\alpha = 0.05$ podemos concluir que se acepta la hipótesis nula confirmando que se sigue una distribución de Poisson.

Este contraste de Poisson se podría extender a un número deseado de temporadas y podríamos suponer que siempre se aceptará la hipótesis nula, esto nos permite extender el estudio al número de temporadas que deseamos haciéndolo lo más parecido a la realidad posible.

4.4. CÁLCULO DE LAS VARIABLES.

Como ya hemos mencionado en numerosas ocasiones, nuestro modelo se basa en el cálculo de unos valores de λ con valores propios de cada equipo y otros que afectan a todos, estos son la Capacidad Atacante en Casa, la Capacidad Defensiva en Casa, la Capacidad Atacante Fuera, la Capacidad Defensiva Fuera, los Goles Metidos en Casa y los Goles Metidos Fuera.

Como cada una de estas variables hay que calcularla para cada equipo lo más natural es recibir un bucle pasando por todos los equipos donde vaya calculando cada una de las variables. Las expresiones con las que se calcularán cada una de las variables ya se han mostrado anteriormente, pero se recordarán ahora para que, una vez vistos los distintos *dataframe* con las distintas observaciones, se pueda entender mejor lo que se va a hacer.

Con cada una de las tablas se han calculado ya los valores de GMC y de GMF. Para el primero, GMC, se ha hecho la media de cada uno de estos valores de cada equipo, otra opción sería calcular el número total de goles anotados por todos los equipos y dividirlo entre todas las jornadas que se han jugado. Para GMF es trivial que sería lo mismo, pero con la otra tabla. Lo mismo sería para las variables intermedias GRC y GRF.

```
GMC=colMeans(LOCAL1[-1])[9]
GRC=colMeans(LOCAL1[-1])[10]

GMF=colMeans(VISITANTE1[-1])[9]
GRF=colMeans(VISITANTE1[-1])[10]
```

Una vez tenemos estos dos valores, el cálculo de las demás variables se facilita. El valor de CAC se calculará como el GMC de cada equipo dividido entre el total; el valor de CAF será lo mismo pero con los valores de GMF; el valor de CDC se calculará como análogamente a CAC pero con los valores de GRC; y, como es de esperar, el valor de CDF será con GRF.

Todos estos valores que se irán calculando se irán guardando en un dataframe con tantas filas como equipos se tengan para estudiar, en este caso son 31. Estas filas tendrán como nombre los nombres de los 31 equipos ordenados alfabéticamente, este orden facilitará cálculos posteriores al establecer una regla general.

```
dimT=dim(temp1)
equip=rep(0,dimT[1])
for (t in 1:dimT[1]){
  equip[t]=temp1$Equipo[t]
}
equipos=sort(equip,decreasing=FALSE)
equipos
```

```
##  [1] "Athletic Bilbao"     "Atletico Madrid"     "CA Osasuna"
##  [4] "Cadiz CF"            "CD Alaves"           "CD Leganes"
##  [7] "Celta Vigo"          "Cordoba CF"          "Deportivo La Coruna"
## [10] "Elche CF"            "Espanyol Barcelona"  "FC Barcelona"
## [13] "Getafe CF"           "Girona"              "Granada CF"
## [16] "Levante UD"          "Malaga CF"           "Rayo Vallecano"
## [19] "RCD Mallorca"        "Real Betis"          "Real Madrid"
## [22] "Real Sociedad"       "Real Valladolid"     "SD Eibar"
## [25] "SD Huesca"           "Sevilla FC"          "Sporting Gijon"
## [28] "UD Almeria"          "UD Las Palmas"       "Valencia CF"
## [31] "Villarreal CF"
```

```r
#Calcular los valores que después nos calcularan las landas.

CAC=rep(0,dimT[1]) #Capacidad Atacante en Casa
CDC=rep(0,dimT[1]) #Capacidad Defensiva en Casa
CAF=rep(0,dimT[1]) #Capacidad Atacante Fuera
CDF=rep(0,dimT[1]) #Capacidad Defensiva Fuera
gmf=rep(0,dimT[1]) #Media de goles fuera
gmc=rep(0,dimT[1]) #Media de goles en casa

#Bucle que va iterando por cada fila calculando cada variable.
for (k in 1:dimT[1]){
  CAC[k]=LOCAL1$GMC[k]/GMC
  CDC[k]=LOCAL1$GRC[k]/GRC
  CAF[k]=VISITANTE1$GMF[k]/GMF
  CDF[k]=VISITANTE1$GRF[k]/GRF
}

PRElandas=data.frame(equipos=temp1$Equipo,CAC,CDC,CAF,CDF)
```

No se ha comentado, pero *temp1* es un dataframe similar a *LOCAL1* y a *VISITANTE1* pero para todas las observaciones. Se calculan las dimensiones de este dataframe y tomando el primer resultado de aplicar la función *dim*, el que corresponde al número de filas, y se construye un vector de tantos elementos como filas donde se van introduciendo los nombres de los equipos.

Para calcular las variables se inicializan a cero y se va iterando para cada valor del vector anterior (cada equipo) calculando cada variable. Posteriormente se construye el dataframe con una columna donde se indica el equipo y, además, una columna para cada una de las variables quedando finalmente como se ve en la imagen inferior.

equipos	CAC	CDC	CAF	CDF
Athletic Bilbao	1.0510332	0.7584800	0.9302319	0.8091625
Atletico Madrid	1.3540683	0.5187306	1.4287539	0.5869745
CA Osasuna	0.8183227	1.1914820	0.9079611	0.9727637
CD Alaves	0.7342985	0.9278615	0.8663565	1.1369965
CD Leganes	0.7384084	0.9154069	0.7709104	0.9617096
Cadiz CF	0.7096392	1.1006679	0.7966074	0.9285472
Celta Vigo	1.0893921	1.0636157	1.0278805	0.9782908
Cordoba CF	0.4603065	1.4384966	0.5139403	1.1606840
Deportivo La Coruna	0.9206130	1.3077242	0.8351529	1.1441028
Elche CF	0.7901929	1.0897702	0.7811892	1.0810942
Espanyol Barcelona	0.9078267	1.0316491	0.9479343	1.0575121
FC Barcelona	1.9409591	0.6451439	2.0454823	0.5836582
Getafe CF	0.8353711	0.7313569	0.7709104	0.9396013
Girona	0.9973308	1.1043004	1.1477999	1.0059261
Granada CF	0.8274558	1.1146792	1.0352225	1.2696461

Ilustración 14 - Tabla con los valores de las variables de estudio calculadas para los equipos.

En esta tabla están los datos de todos los equipos del intervalo de estudio, para poder predecir una temporada en concreto necesitaremos únicamente los valores de los equipos que compiten en dicha temporada. Para conseguir esto se hace algo similar a lo realizado anteriormente: se obtienen los equipos de la temporada que se quiera, se meten en un vector y se filtra la tabla superior quedándonos solo con los equipos que están en este último vector. Esto es lo que el código siguiente pretende realizar.

```
equipos_estudio=sort(unique(temp[temp$Season==tmp,]$Equipo),decreasing=FAL
SE)
equipos_estudio
```

```
##  [1] "Athletic Bilbao"    "Atletico Madrid"    "CA Osasuna"
##  [4] "Cadiz CF"           "Celta Vigo"         "Elche CF"
##  [7] "Espanyol Barcelona" "FC Barcelona"       "Getafe CF"
## [10] "Girona"             "Rayo Vallecano"     "RCD Mallorca"
## [13] "Real Betis"         "Real Madrid"        "Real Sociedad"
## [16] "Real Valladolid"    "Sevilla FC"         "UD Almeria"
## [19] "Valencia CF"        "Villarreal CF"
```

```
PRElandasF=PRElandas[PRElandas$equipos %in% equipos_estudio,]
dim(PRElandasF)
```

```
## [1] 20  5
```

```
rownames(PRElandasF)=1:20
```

Como se puede observar los valores dentro del vector *equipos_estudio* corresponden con los nombres de los equipos que disputaron la temporada 2022-23 de la primera división de la Liga Española, siempre manteniendo el mismo orden alfabético de antes.

También podemos ver las dimensiones del dataframe *PRElandasF* son de 20 filas, correspondiéndose con los 20 equipos y las 5 columnas manteniendo el formato anterior; como se puede observar en la siguiente imagen.

	equipos	CAC	CDC	CAF	CDF
1	Athletic Bilbao	1.0510332	0.7584800	0.9302319	0.8091625
2	Atletico Madrid	1.3540683	0.5187306	1.4287539	0.5869745
3	CA Osasuna	0.8183227	1.1914820	0.9079611	0.9727637
4	Cadiz CF	0.7096392	1.1006679	0.7966074	0.9285472
5	Celta Vigo	1.0893921	1.0636157	1.0278805	0.9782908
6	Elche CF	0.7901929	1.0897702	0.7811892	1.0810942
7	Espanyol Barcelona	0.9078267	1.0316491	0.9479343	1.0575121
8	FC Barcelona	1.9409591	0.6451439	2.0454823	0.5836582
9	Getafe CF	0.8353711	0.7313569	0.7709104	0.9396013
10	Girona	0.9973308	1.1043004	1.1477999	1.0059261
11	RCD Mallorca	0.8343056	0.9154069	0.7966074	1.2435900
12	Rayo Vallecano	1.0229034	1.1115656	0.8736985	1.1662110
13	Real Betis	1.0825727	1.0364925	1.1021164	0.9064389
14	Real Madrid	1.8182107	0.6625803	1.9992276	0.6698805
15	Real Sociedad	1.0587050	0.8238662	1.1769232	0.8721711
16	Real Valladolid	0.7096392	1.0461794	0.7323649	0.9782908
17	Sevilla FC	1.2811865	0.7671982	1.1717838	0.8423249
18	UD Almeria	0.9397925	1.0897702	0.8993955	1.3099148
19	Valencia CF	1.1546022	0.8674571	1.1358080	0.9152822
20	Villarreal CF	1.2850224	0.8587389	1.1666444	0.7196241

Ilustración 15 - Tabla con los valores de las variables de estudio para los equipos de la temporada de estudio.

4.5. CÁLCULO DE λ.

La base de nuestro modelo es el cálculo de los diferentes valores de λ para cada uno de los enfrentamientos, y recordemos que estos valores son el resultado de la multiplicación de las variables que acabamos de calcular; por tanto, el próximo paso sería el cálculo de estas λ.

Para ello nos basaremos en la tabla superior y a partir de ahí con distintas iteraciones por filas iremos calculando los valores que necesitemos.

Como se sabe, una temporada está compuesta por 38 jornadas en la que todos los equipos se cruzan dos veces, cada una de las veces en el estadio de cada uno de los dos equipos; es decir, un equipo 'i' se enfrente a un equipo ficticio 'j' dos veces, una en casa de 'i' y otra en casa de 'j'.

Sabiendo esto, sabremos que para cada pareja de equipos existirán cuatro valores de λ, dos por equipo y dos por cada enfrentamiento. Por tanto, para agilizar el proceso de selección de las variables precisas y la multiplicación de las mismas se han creado varias funciones que facilitarán este proceso haciéndolo más automático.

En concreto se han creado dos funciones: *landaL*y *landaV*. Para un enfrentamiento de un ficticio equipo 'i' contra un ficticio equipo 'j'; *landaL* calculará el valor de λ cuando cada uno de los dos equipos actúe de local.

4.5.1 FUNCIÓN *landaL*.

Esta función viene definida por el siguiente código:

```
landaL <- function(l,v) {

  landaL = PRElandasF[PRElandasF$equipos == l,2]*
    PRElandasF[PRElandasF$equipos == v,5]*GMC

  return(landaL)
}
```

Esta función recibe dos argumentos, que pueden ser dos valores numéricos o dos cadenas de texto, cada uno de estos argumentos identifica a cada uno de los equipos que participan al enfrentamiento. El primero de los argumentos es el correspondiente al equipo local, y el otro input corresponde al visitante.

La funcionalidad de esta función, valga la redundancia, se explica de la siguiente forma:

1. La función toma los argumentos l y v.
2. Se utilizan estos argumentos para filtrar las filas del dataframe *PRElandasF* donde la columna en la que están los nombres de los equipos es igual a 'l' y a 'v', respectivamente.
3. Luego realiza operaciones de multiplicación con los valores de la segunda columna de la fila del equipo 'l' en *PRElandasF*, los correspondientes a la variable CAC; los valores de la quinta columna de la fila del equipo 'v' en *PRElandasF*, los correspondientes a CDF; y la variable llamada GMC.
4. El resultado de estas operaciones se asigna a una variable llamada *landaL*.
5. Finalmente, la función devuelve el valor de *landaL*.

4.5.2. FUNCIÓN *landaV*.

Esta función viene definida por el siguiente código:

```
landaV <- function(v,l) {

  landaV = PRElandasF[PRElandasF$equipos== v,4]*
    PRElandasF[PRElandasF$equipos == l,3]*GMF

  return(landaV)
}
```

Esta función, muy similar a la anterior, recibe dos argumentos, que pueden ser dos valores numéricos o dos cadenas de texto, cada uno de estos argumentos identifica a cada uno de los equipos que participan al enfrentamiento. El primero de los argumentos es el correspondiente al equipo visitante, y el otro input corresponde al local.

La funcionalidad de esta función, valga la redundancia, se explica de la siguiente forma:

1. La función toma los argumentos v y l.
2. Se utilizan estos argumentos para filtrar las filas del dataframe *PRElandasF* donde la columna en la que están los nombres de los equipos es igual a v y a l, respectivamente.
3. Luego realiza operaciones de multiplicación con los valores de la cuarta columna de la fila del equipo v en *PRElandasF*, los correspondientes a la variable CAF; los valores de la tercera columna de la fila del equipo l en *PRElandasF*, los correspondientes a CDC; y la variable llamada GMF.
4. El resultado de estas operaciones se asigna a una variable llamada *landaV*.
5. Finalmente, la función devuelve el valor de *landaV*.

Una vez declaradas estas funciones pasaremos directamente al cálculo directo de las λ para cada enfrentamiento.

Lo primero de todo es crear dos tablas, teniendo una fila y una columna para cada equipo, es decir, dos tablas de 20x20 creando así una casilla para cada cruce. El motivo de crear dos tablas y no solo una es trivial, una para cada enfrentamiento que hay entre dos equipos.

```
n_equipos=20
landasL=matrix(1,nrow=n_equipos,ncol=n_equipos)
landasV=matrix(1,nrow=n_equipos,ncol=n_equipos)

colnames(landasL)=equipos_estudio
rownames(landasL)=equipos_estudio
colnames(landasV)=equipos_estudio
rownames(landasV)=equipos_estudio
```

```
for (j in 1:n_equipos){
  for (i in 1:n_equipos){
    if (i==j) {
      landasL[i,j]=0
      landasV[i,j]=0
    } else {
      landasL[i,j]=landaL(equipos_estudio[i],equipos_estudio[j])
      #Cuando la fila es local y la columna visitante.
      landasV[i,j]=landaV(equipos_estudio[i],equipos_estudio[j])
      #Cuando la fila es visitante y la columna local.

    }
  }
}
```

En el anterior código se ha realizado lo que se ha comentado. Lo primero de todo es crear e inicializar dos matrices con los nombres de filas y columnas correspondientes a los equipos en orden alfabético.

Posteriormente se utiliza un bucle anidado para recorrer todas las combinaciones de equipos. En caso de que 'i' sea igual a 'j', es decir la diagonal principal, el valor que se establece es nulo, ya que un equipo nunca jugará contra sí mismo. Para los casos en los que esto no ocurre, es decir, todo lo que no sea la diagonal principal, lo que se hace es utilizar las funciones creadas anteriormente para calcular los valores de λ.

El resultado de este bucle son dos tablas, *landasL* y *landasV*, llenas de valores que se usarán posteriormente.

	Athletic Bilbao	Atletico Madrid	CA Osasuna	Cadiz CF	Celta Vigo	Elche CF	Espanyol Barcelona	FC Barcelona	Getafe CF	Girona	Rayo Vallecano	RCD Mallorca
Athletic Bilbao	0.0000000	0.8464790	1.4028277	1.3390628	1.4107983	1.559052	1.525044	0.8416966	1.355004	1.4506513	1.681799	1.793388
Atletico Madrid	1.5033388	0.0000000	1.8072926	1.7251429	1.8175613	2.008559	1.964746	1.0843755	1.745680	1.8689048	2.166697	2.310459
CA Osasuna	0.9085334	0.6590590	0.0000000	1.0425793	1.0984317	1.213860	1.187382	0.6553355	1.054991	1.1294609	1.309430	1.396312
Cadiz CF	0.7878688	0.5715278	0.9471647	0.0000000	0.9525463	1.052644	1.029683	0.5682988	0.914875	0.9794544	1.135521	1.210864
Celta Vigo	1.2094851	0.8773724	1.4540257	1.3879337	0.0000000	1.615951	1.580702	0.8724154	1.404457	1.5035948	1.743179	1.858840
Elche CF	0.8773025	0.6364039	1.0546806	1.0067406	1.0606732	0.000000	1.146566	0.6328084	1.018726	1.0906357	1.264418	1.348313
Espanyol Barcelona	1.0079042	0.7311436	1.2116881	1.1566114	1.2185727	1.346626	0.000000	0.7270129	1.170381	1.2529957	1.452649	1.549033
FC Barcelona	2.1549276	1.5632057	2.5906233	2.4728677	2.6053428	2.879125	2.816322	0.0000000	2.502307	2.6789400	3.105804	3.311876
Getafe CF	0.9274612	0.6727894	1.1149806	1.0642997	1.1213157	1.239149	1.212119	0.6689884	0.000000	1.1529913	1.336710	1.425401
Girona	1.1072750	0.8032282	1.3311503	1.2706435	1.3387137	1.4	1.239149	1.447122	0.7986902	1.285770	0.0000000	1.595868	1.701755
Rayo Vallecano	1.1356667	0.8238238	1.3652824	1.3032241	1.3730397	1.517325	1.484227	0.8191694	1.318739	1.4118261	0.000000	1.745389
RCD Mallorca	0.9262782	0.6719313	1.1135584	1.0629422	1.1198855	1.237568	1.210573	0.6681351	1.075596	1.1515207	1.335005	0.000000
Real Betis	1.2019139	0.8718802	1.4449239	1.3792455	1.4531337	1.605836	1.570807	0.8669543	1.395665	1.4941826	1.732267	1.847204
Real Madrid	2.0186476	1.4643468	2.4267894	2.3164808	2.4405780	2.697046	2.638214	1.4560737	2.344058	2.5095209	2.909390	3.102430
Real Sociedad	1.1754151	0.8526576	1.4130673	1.3488369	1.4210961	1.570432	1.536175	0.8478404	1.364895	1.4612400	1.694075	1.806478
Real Valladolid	0.7878688	0.5715278	0.9471647	0.9041117	0.9525463	1.052644	1.029683	0.5682988	0.914875	0.9794544	1.135521	1.210864
Sevilla FC	1.4224226	1.0318393	1.7100162	1.6322882	1.7197322	1.900450	1.858995	1.0260097	1.651720	1.7683122	2.050076	2.186100
UD Almeria	1.0433938	0.7568881	1.2543532	1.1973371	1.2614802	1.394043	1.363634	0.7526119	1.211591	1.2971152	1.503798	1.603577
Valencia CF	1.2818838	0.9298911	1.5410625	1.4710142	1.5498185	1.712681	1.675322	0.9246375	1.488526	1.5935987	1.847524	1.970108
Villarreal CF	1.4266813	1.0349287	1.7151360	1.6371753	1.7248811	1.906140	1.864561	1.0290816	1.656665	1.7736065	2.056214	2.192645

Ilustración 16 - Tabla landasL donde están todos los valores de lambda para los enfrentamientos donde el equipo de la fila actúa de local.

En el caso de *landasL*, el equipo que está en el eje horizontal es el que actúa de local. Por poner un ejemplo, la celda (1,2) donde la fila corresponde a Athletic Bilbao y la columna

corresponde al Atlético Madrid lo que está representado es el partido donde el club vasco juega de local y el madrileño de visitante.

En el caso de *landasV*, el equipo que está en el eje horizontal es el equipo que actúa como visitante. En el ejemplo que se ha puesto anteriormente, el enfrentamiento sería el contrario, es decir, se jugaría en Madrid y el club vasco sería el visitante, ya que en la función que se utiliza para calcular los valores de esta tabla (*landaV*) el pimer argumento corresponde al equipo visitante y en el bucle se introduce el término 'i' correspondiente a las filas.

	Athletic Bilbao	Atletico Madrid	CA Osasuna	Cadiz CF	Celta Vigo	Elche CF	Espanyol Barcelona	FC Barcelona	Getafe CF	Girona	Rayo Vallecano	RCD Mallorca
Athletic Bilbao	0.0000000	0.4941591	1.1350434	1.0485310	1.0132339	1.0381495	0.9827815	0.6145845	0.6967136	1.0519915	1.0589125	0.8720456
Atletico Madrid	1.1097761	0.0000000	1.7433264	1.6104509	1.5562377	1.5945058	1.5094655	0.9439474	1.0700906	1.6157659	1.6263959	1.3393849
CA Osasuna	0.7052534	0.4823285	0.0000000	1.0234280	0.9889760	1.0132951	0.9592527	0.5998707	0.6800336	1.0268057	1.0335610	0.8511679
Cadiz CF	0.6187600	0.4231750	0.9719985	0.0000000	0.8676865	0.8890230	0.8416085	0.5263016	0.5966332	0.9008767	0.9068035	0.7467793
Celta Vigo	0.7984000	0.5460322	1.2541916	1.1585978	0.0000000	1.1471265	1.0859464	0.6790989	0.7698493	1.1624215	1.1700690	0.9635863
Elche CF	0.6067840	0.4149845	0.9531856	0.8805343	0.8508925	0.0000000	0.8253193	0.5161152	0.5850855	0.8834403	0.8892525	0.7323256
Espanyol Barcelona	0.7363023	0.5035630	1.1566434	1.0684846	1.0325158	1.0579055	0.0000000	0.6262801	0.7099722	1.0720109	1.0790637	0.8886407
FC Barcelona	1.5888161	1.0866041	2.4958413	2.3056095	2.2279950	2.2827817	2.1610334	0.0000000	1.5320002	2.3132188	2.3284374	1.9175366
Getafe CF	0.5988000	0.4095242	0.9406437	0.8689483	0.8396966	0.8603449	0.8144598	0.5093242	0.0000000	0.8718161	0.8775518	0.7226897
Girona	0.8915467	0.6097360	1.4005140	1.2937675	1.2502149	1.2809579	1.2126402	0.7583271	0.8596651	0.0000000	1.3065771	1.0760046
Rayo Vallecano	0.6786400	0.4641274	1.0660629	0.9848081	0.9516561	0.9750575	0.9230544	0.5772341	0.6543719	0.9880583	0.0000000	0.8190483
RCD Mallorca	0.6187600	0.4231750	0.9719985	0.8979133	0.8676865	0.8890230	0.8416085	0.5263016	0.5966332	0.9008767	0.9068035	0.0000000
Real Betis	0.8560623	0.5854679	1.3447721	1.2422743	1.2004551	1.2299745	1.1643759	0.7281449	0.8254496	1.2463742	1.2545740	1.0331786
Real Madrid	1.5528881	1.0620326	2.4394027	2.2534726	2.1776132	2.2311610	2.1121658	1.3208473	1.4973570	2.2609098	2.2757842	1.8741753
Real Sociedad	0.9141680	0.6252069	1.4360494	1.3265944	1.2819368	1.3134598	1.2434086	0.7775682	0.8814775	1.3309726	1.3397290	1.1033063
Real Valladolid	0.5688600	0.3890479	0.8936115	0.8255009	0.7977118	0.8173276	0.7737368	0.4838580	0.5485177	0.8282253	0.8336742	0.6865552
Sevilla FC	0.9101760	0.6224767	1.4297785	1.3208014	1.2763388	1.3077242	1.2379789	0.7741727	0.8776282	1.3251605	1.3338787	1.0984883
UD Almeria	0.6986000	0.4777782	1.0974177	1.0137730	0.9796460	1.0037357	0.9502031	0.5942115	0.6736182	1.0171188	1.0238104	0.8431380
Valencia CF	0.8822320	0.6033656	1.3858818	1.2802505	1.2371530	1.2675748	1.1999708	0.7504043	0.8506835	1.2844758	1.2929263	1.0647628
Villarreal CF	0.9061840	0.6197466	1.4235075	1.3150085	1.2707408	1.3019886	1.2325492	0.7707772	0.8737790	1.3193484	1.3280283	1.0936704

Ilustración 17 - Tabla landasV donde están todos los valores de lambda para los enfrentamientos donde el equipo de la fila actúa de visitante.

4.6. RECREACIÓN DE TEMPORADA.

Una vez que se han calculado los cuatro valores de cada enfrentamiento el siguiente paso ya sería simular estos enfrentamientos. Para esto también, con la idea de agilizar procesos e intentar evitar el uso excesivo de bucles se ha creado una función que nos ayudará.

La función en cuestión es una función que se ha llamado *match*. Esta viene definida por el siguiente código:

```
match <- function(r1,r2){

  gL=rpois(1,r1)
  gV=rpois(1,r2)

  result=c(gL,gV)

  return(result)
}
```

Esta función toma dos argumentos 'r1' y 'r2', estos valores son los valores de λ para cada equipo. Con estos valores, se utiliza la función 'rpois' para calcular un número aleatorio siguiendo la distribución de Poisson con los argumentos r1 y r2. Estos dos valores calculados,

gL y gV, se corresponderían con los goles anotados por cada uno de los equipos en el enfrentamiento y se guardarían en un vector llamado *result* que es lo que finalmente devuleve esta función.

Lo que viene a continuación es la parte técnicamente más exigente del trabajo. Lo que se quiere realizar es un bucle que pase por cada uno de los enfrentamientos tomando los valores de las dos tablas con los valores de λ. Además de esto, se cree que una sola replicación del proceso no reflejaría realmente las posibilidades de cada resultado por tanto se realizan 1000 replicaciones.

Para cada partido interesaría también, a la hora de construir una tabla con el formato de clasificación los goles que anota y que encaja cada equipo, además de ir teniendo en cuenta los partidos que ganan, los que pierden, los que empatan y los puntos que se consiguen.

Lo primero que se hace en este código es inicializar varias matrices: tabla_puntos, tabla_GF, tabla_GC, tabla_PG, tabla_PE, tabla_PP y tabla_posiciones. Estas matrices servirán posteriormente para ir almacenando los resultados de cada una de las replicaciones de la temporada. Estas matrices tienen 20 filas (correspondientes a los equipos de estudio) y 1000 columnas, una por cada replicación. A todas estas matrices se le asignan nombres de filas con los equipos en orden alfabético excepto a tabla_posiciones que se asignan números del 1 al 20.

Lo primero que se hace en este código es inicializar varias matrices: tabla_puntos, tabla_GF, tabla_GC, tabla_PG, tabla_PE, tabla_PP y tabla_posiciones. Estas matrices servirán posteriormente para ir almacenando los resultados de cada una de las replicaciones de la temporada. Estas matrices tienen 20 filas (correspondientes a los equipos de estudio) y 1000 columnas, una por cada replicación. A todas estas matrices se le asignan nombres de filas con los equipos en orden alfabético excepto a tabla_posiciones que se asignan números del 1 al 20.

```
tabla_puntos=matrix(0,20,1000)
tabla_GF=matrix(0,20,1000)
tabla_GC=matrix(0,20,1000)
tabla_PG=matrix(0,20,1000)
tabla_PE=matrix(0,20,1000)
tabla_PP=matrix(0,20,1000)
tabla_posiciones=matrix(0,20,1000)
rownames(tabla_puntos)=equipos_estudio
rownames(tabla_GF)=equipos_estudio
rownames(tabla_GC)=equipos_estudio
rownames(tabla_PG)=equipos_estudio
rownames(tabla_PE)=equipos_estudio
rownames(tabla_PP)=equipos_estudio
rownames(tabla_posiciones)=paste("Posición",1:20)
```

Posteriormente se define una matriz tridimensional o array que se llama resultado2. En esta matriz se almacenarán los resultados de los partidos simulados. Esta matriz tendrá unas

dimensiones de 20x20x2, es decir serán dos “tablas” de 20x20, se incializa a 0 y se asignan los nombres de los equipos a las filas y las columnas.

```
dimensiones=c(20,20,2)
resultado2=array(0,dim=dimensiones)
rownames(resultado2)=equipos_estudio
colnames(resultado2)=equipos_estudio
parti=c(0,0)
replicaciones = 1000
```

En el bucle se itera sobre cada una de las 1000 replicaciones y dentro de cada replicación se itera sobre cada posible enfrentamiento, para ello se utilizan tres variables ‘k’, ‘i’, y ‘j’.

```
for (k in 1:replicaciones){
  for (i in 1:20){
    for( j in 1:20){
```

Para simular el partido se utiliza la función *match* que se había creado introduciendo como argumentos las λ respectivas a cada equipo.

Como esta función devuelve dos valores, para poder obtener los valores de goles anotados y encajados por cada equipo se guarda el primero de los valores devueltos, correspondiente a los goles del equipo local en una de las tablas del array *resultado2* y el otro valor en la otra tabla.

Al igual que en el anterior bucle se vuelve a contemplar la posibilidad de que los valores de ‘i’ y ‘j’ coincidan, en este caso se le asigna un valor NA. Para el resto de los casos se realiza lo que se ha comentado.

Como esta función devuelve dos valores, para poder obtener los valores de goles anotados y encajados por cada equipo se guarda el primero de los valores devueltos, correspondiente a los goles del equipo local en una de las tablas del array *resultado2* y el otro valor en la otra tabla.

```
      if (i==j){
        resultado2[i,j,1]=NA
        resultado2[i,j,2]=NA
      } else {
        parti=match(landasL[i,j],landasV[j,i])
        resultado2[i,j,1]=parti[1]
        resultado2[i,j,2]=parti[2]
      }
    }
    }
```

Se inicializan los valores de las siguientes variables a cero una vez terminada una iteración para no arrastrar los valores calculados.

```
  gol_C=0
  gol_F=0
  part_empata2=0
  part_gana2=0
  part_perdi2=0
  puntos=0
```

Se crea la matriz *dif* para poder facilitar el cálculo de goles a favor, en contra y poder sacar las demás estadísticas. En caso de no trabajar con una sola tabla lo que habría que hacer sería volver a recurrir a un bucle, algo que sería un proceso más lento y costoso. Esta matriz se define como la diferencia entre las dos tablas correspondientes a *resultado2*. A esta matriz se le asignan los nombres de los equipos a las filas y las columnas.

También dentro del bucle aparece la necesidad de definir una variable temporal, en forma de matriz, para facilitar posteriormente el cálculo del número de partidos ganados, empatados y perdidos.

```
  diag(resultado2[,,1])=NA
  diag(resultado2[,,2])=NA
  dif = resultado2[,,1]-resultado2[,,2]
  colnames(dif)=equipos_estudio
  rownames(dif)=equipos_estudio
  dift=t(dif)
```

Los goles a favor para cada equipo se calcularán como la suma de los valores de la fila dentro la tabla donde actúa de local más los valores de las columnas (ese el motivo de la matriz traspuesta) de la tabla donde actúa de visitante. Para los goles en contra sería lo contrario, las filas de visitante y las columnas de local.

```
  gol_F=0
  gol_F=rowSums(resultado2[,,1], na.rm = TRUE) +
rowSums(t(resultado2[,,2]), na.rm = TRUE)

  gol_C=rowSums(resultado2[,,2], na.rm = TRUE) +
rowSums(t(resultado2[,,1]), na.rm = TRUE)
```

Para los partidos ganados, empatados y perdidos la idea es la misma, operar con la tabla *dif* y su traspuesta *dift*. Los ganados será la suma de las filas donde la en *dif* el valor sea posotivo y en la traspuesta sea negativo, para los empatados será la suma de filas donde los valores sean nulos y para los perdidos se ha optado por hacer una resta.

En ambos casos, al hacer uso de la función *rowSums* se ha tenido que hacer uso del atributo `na.rm=TRUE`, esto lo que consigue es que los valores NA de la diagonal principal que se han definido como NA no se tengan en cuenta ya que sino aparecería un mensaje de error al intentar operar dos variables de distinta categoría (numérica y NA).

```
  part_gana2=rowSums((dif>0)+(dift<0), na.rm=TRUE)
  part_empata2=rowSums((dif==0)+(dift==0), na.rm=TRUE)
  part_perdi2=38-part_gana2-part_empata2
```

Posteriormente, para el cálculo de los puntos es algo mas trivial, podríamos recurrir simplemente a multiplicar por tres el número de partidos ganados y sumarle el de partidos empatados, pero se ha optado por desarrollar el código ya que ofrece una mayor concisión y permite realizar estos cálculos en una sola línea de código. En caso de la otra opción correríamos el riesgo de arrastrar un mal cálculo que nos pueda hacer cometer errores en el cálculo.

```
  puntos = rowSums((dif>0)*3 + (dif==0)*1 + (dift<0)*3 + (dift==0)*1,
                   na.rm = TRUE)
```

Por último, lo que queda es ordenar todos estos datos en una matriz, a la que llamamos temporada. En esta matriz se incluirán los puntos, los goles a favor, los goles en contra y el número de partidos tanto empatados, como perdidos, como ganados.

```
temporada=data.frame(puntos,GF=gol_F,GC=gol_C,PG=part_gana2,PE=part_empata
2,
                     PP=part_perdi2)

  tabla_puntos[,k]=temporada[,1]
  tabla_GF[,k]=temporada[,2]
  tabla_GC[,k]=temporada[,3]
  tabla_PG[,k]=temporada[,4]
  tabla_PE[,k]=temporada[,5]
  tabla_PP[,k]=temporada[,6]
  tabla_posiciones[,k]=order(-temporada$puntos)

  k=k+1
}
```

El resultado de todo esto es una temporada como se puede ver en la imagen de debajo.

	puntos	GF	GC	PG	PE	PP
Athletic Bilbao	54	48	39	13	15	10
Atletico Madrid	71	56	22	20	11	7
CA Osasuna	46	30	48	12	10	16
Cadiz CF	45	35	52	12	9	17
Celta Vigo	54	52	46	15	9	14
Elche CF	25	30	66	4	13	21
Espanyol Barcelona	44	34	49	10	14	14
FC Barcelona	80	87	41	24	8	6
Getafe CF	44	30	34	10	14	14
Girona	32	32	49	7	11	20
Rayo Vallecano	34	36	61	8	10	20
RCD Mallorca	26	29	75	6	8	24
Real Betis	53	58	52	14	11	13
Real Madrid	89	90	30	28	5	5
Real Sociedad	72	58	36	20	12	6
Real Valladolid	46	27	41	12	10	16
Sevilla FC	57	48	43	17	6	15
UD Almeria	46	46	57	13	7	18
Valencia CF	51	40	37	12	15	11
Villarreal CF	67	48	36	19	10	9

Ilustración 18 - Ejemplo de temporada con las columnas calculadas en la iteración.

Como lo que se está intentando es comprobar si un modelo refleja la realidad o no, está claro que una de las mil replicaciones no tiene porqué reflejar la realidad al completo. Para ello lo que se ha hecho es realizar las medias de los valores que se han guardado en las matrices creadas al principio de la simulación, con estos valores si que podremos comprobar si realmente el modelo es útil o no.

Sabiendo que la temporada que se está intentando replicar es la 2022-23, es natural tener que enseñar cómo acabo aquella temporada para poder tener los valores con los que compararlo. En la imagen de debajo se puede ver cómo termino esa temporada la clasificación, se pueden ver al igual que en nuestra tabla, el número de puntos, goles a favor, goles en contra, y los partidos empatados, ganados y perdidos. [4]

Equipos	Puntos	J.	G.	E.	P.	F.	C.
Barcelona	88	38	28	4	6	70	20
Real Madrid	78	38	24	6	8	75	36
Atlético	77	38	23	8	7	70	33
Real Sociedad	71	38	21	8	9	51	35
Villarreal	64	38	19	7	12	59	40
Real Betis	60	38	17	9	12	46	41
Osasuna	53	38	15	8	15	37	42
Athletic	51	38	14	9	15	47	43
Mallorca	50	38	14	8	16	37	43
Girona	49	38	13	10	15	58	55
Rayo Vallecano	49	38	13	10	15	45	53
Sevilla	49	38	13	10	15	47	54
Celta	43	38	11	10	17	43	53
Cádiz	42	38	10	12	16	30	53
Getafe	42	38	10	12	16	34	45
Valencia	42	38	11	9	18	42	45
Almería	41	38	11	8	19	49	65
Real Valladolid	40	38	11	7	20	33	63
Espanyol	37	38	8	13	17	52	69
Elche	25	38	5	10	23	30	67

Ilustración 19 - Clasificación del final de la temporada 2022-23 de la primera división de la Liga Española.

Para poder comparar estos valores se ha construido una tabla con los valores medios para cada equipo.

```
resultados_df <- data.frame(
  Puntos = rowMeans(tabla_puntos),
  G = rowMeans(tabla_PG),
  E = rowMeans(tabla_PE),
  P = rowMeans(tabla_PP),
  GF = rowMeans(tabla_GF),
  GC = rowMeans(tabla_GC)
)
```

El resultado de este código se muestra a continuación.

	Puntos	G	E	P	GF	GC
FC Barcelona	86.180	26.476	6.752	4.772	85.143	28.932
Real Madrid	82.527	25.147	7.086	5.767	81.411	31.855
Atletico Madrid	74.452	21.714	9.310	6.976	59.412	27.380
Villarreal CF	61.466	17.245	9.731	11.024	52.449	38.705
Sevilla FC	61.243	17.244	9.511	11.245	52.921	39.878
Real Sociedad	55.919	15.294	10.037	12.669	47.211	42.047
Valencia CF	55.174	15.142	9.748	13.110	48.117	44.330
Athletic Bilbao	54.231	14.629	10.344	13.027	42.453	39.387
Real Betis	51.181	13.822	9.715	14.463	45.895	47.730
Celta Vigo	49.058	13.180	9.518	15.302	44.873	50.654
Girona	47.398	12.599	9.601	15.800	44.257	52.190
Getafe CF	46.562	11.899	10.865	15.236	34.293	42.424
Espanyol Barcelona	43.781	11.309	9.854	16.837	38.647	52.330
Rayo Vallecano	42.349	11.008	9.325	17.667	40.281	56.471
CA Osasuna	40.224	10.118	9.870	18.012	35.630	53.753
UD Almeria	39.393	10.106	9.075	18.819	38.414	60.624
RCD Mallorca	39.205	9.832	9.709	18.459	34.061	55.610
Cadiz CF	38.633	9.356	10.565	18.079	31.211	50.156
Real Valladolid	37.935	9.221	10.272	18.507	30.485	50.962
Elche CF	37.659	9.229	9.972	18.799	32.628	54.374

Ilustración 20 - Tabla con las medias de las variables calculadas para cada uno de los equipos.

Por lo que ha simple vista se puede observar es que el orden de los tres primeros es similar, y de los cinco primeros clasificados se han acertado cuatro de ellos, quedando el no acertado, la Real Sociedad, en sexta posición. Aunque estos dos datos aislados puedan mostrar que el modelo se asemeja con la realidad lo mejor sería realizar un análisis y comparación de los resultados obtenidos.

4.7. ANÁLISIS Y COMPARACIÓN DE LOS RESULTADOS.

Como se acaba de mencionar en el párrafo anterior, dos datos aislados no son suficiente como para dar por probada la validez de un modelo. Para ello lo que haremos será comparar los resultados obtenidos con los reales en cuanto a puntos conseguidos, goles a favor y goles en contra se refiere.

4.7.1. PUNTOS.

Para comparar los valores obtenidos para los puntos de cada equipo lo que se hará será comparar la primera columna de la tabla anterior con los puntos reales.

Equipo	PT reales	P calculados	ErrorAbsoluto	ErrorRelativo %
Athletic	51	54,231	3,231	6,335294118
Atlético	77	74,452	2,548	3,309090909
CA Osasuna	53	40	12,776	24,10566038
Cádiz	42	38,633	3,367	8,016666667
Celta	43	49,058	6,058	14,08837209
Elche	25	37,659	12,659	50,636
Espanyol	37	43,781	6,781	18,32702703
FC Barcelona	88	86,18	1,82	2,068181818
Getafe	42	46,562	4,562	10,86190476
Girona	49	47,398	1,602	3,269387755
Rayo	49	42,349	6,651	13,57346939
RCD Mallorca	50	39,205	10,795	21,59
Real Betis	60	51,181	8,819	14,69833333
Real Madrid	78	82,527	4,527	5,803846154
Real Sociedad	71	55,919	15,081	21,24084507
Real Valladolid	40	37,935	2,065	5,1625
Sevilla	49	61,243	12,243	24,98571429
UD Almería	41	39,393	1,607	3,919512195
Valencia	42	55,174	13,174	31,36666667
Villarreal	64	61,466	2,534	3,959375

Ilustración 21 - Tabla con los puntos calculados, los errores absolutos y los errores relativos para los 20 equipos.

En la tabla superior se pueden ver los valores tanto de los puntos reales y calculados, como de los errores tanto relativos como absolutos. Para poder llegar a comprender mejor estos datos, realizar gráficas será una forma más visual de llegar a comprenderlos.

En la siguiente gráfica se representará el error absoluto cometido a la hora de pronosticar los puntos, esto es, la diferencia en valor absoluto entre los puntos reales obtenidos y los puntos calculados. Con un valor medio de 6,645 puntos, o lo que equivale a dos victorias y un empate, se considera que este error es asumible sabiendo que el campeonato consta de 38 jornadas y en cada enfrentamiento existen tres posibles resultados. Se puede observar que la mayor diferencia se produce con la Real Sociedad.

Para poder llegar a comprender este error podemos echar un ojo en las sensaciones de los propios vascos con respecto a la temporada de su equipo. En diversos periódicos tanto de Guipúzcoa como a nivel nacional se alaba la temporada 2022-23 del conjunto blanquiazul con titulares como: “El balance de la Real Sociedad en 2023: un año para enmarcar” o “Imanol Alguacil, la cara visible del éxito de una Real Sociedad que asombra a Europa”.

Por tanto podemos concluir que este valor del error tan elevado es algo que no se puede explicar ya que nuestro modelo o tiene todos los factores en cuenta, ya que es imposible.

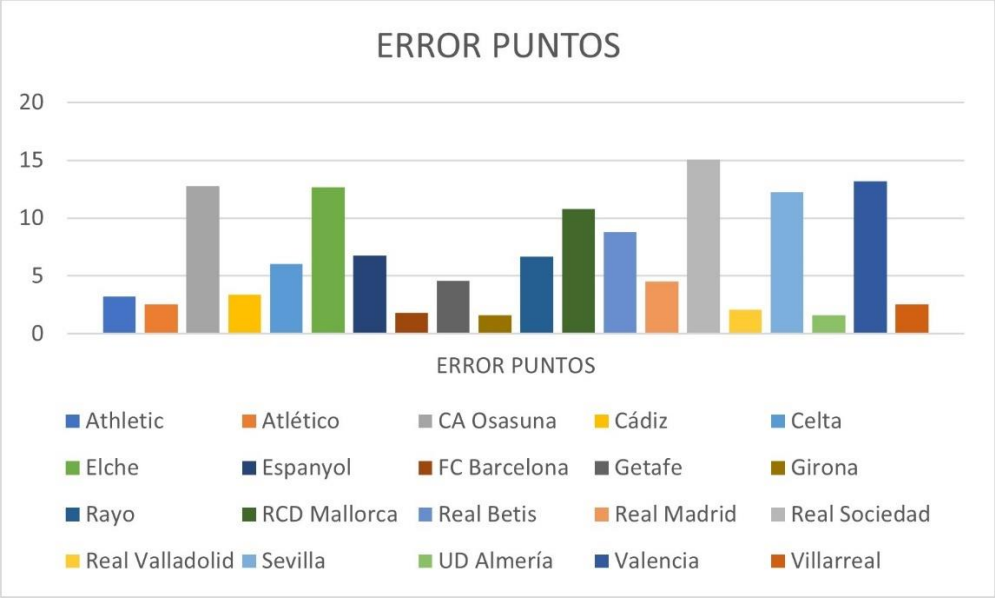

Ilustración 22 - Histograma donde se representan los errores absolutos del cálculo de los puntos.

Pero hablar de diferencia de puntos no sería del todo exacto ya que la cantidad de puntos conseguida por cada equipo no es la misma, por tanto, lo más razonable es trabajar con el error relativo donde esta diferencia se divide entre el total de los puntos obtenidos.

Con un error medio del 14.36%, podemos afirmar que nuestro modelo tiene una fiabilidad del 85.64% a la hora de predecir la cantidad de puntos obtenidos por cada uno de los equipos, algo que teniendo en cuenta la simplicidad del modelo es bastante aceptable.

Para poder tener una forma más visual de entender los valores de la tabla, se realizará una gráfica de barras con los valores para todos los equipos.

En este caso el muy notable como existe un error bastante alto, es el relativo al Elche C.F. En este caso el Elche consiguió 12 puntos menos de los que se han pronosticado. Pero, al igual que en el caso anterior de la Real Sociedad, podemos recurrir a la prensa para entender que esta diferencia no es explicable desde un punto de vista matemático, sino que hay que recurrir a otros factores.

Para este equipo existen titulares como: “El Elche de la temporada 22-23 tiene el peor puntaje del siglo XXI” o “El Elche CF despide un 2023 para olvidar”. Estos titulares dejan bastante claro que estos resultados eran imprevisibles a principio de temporada.

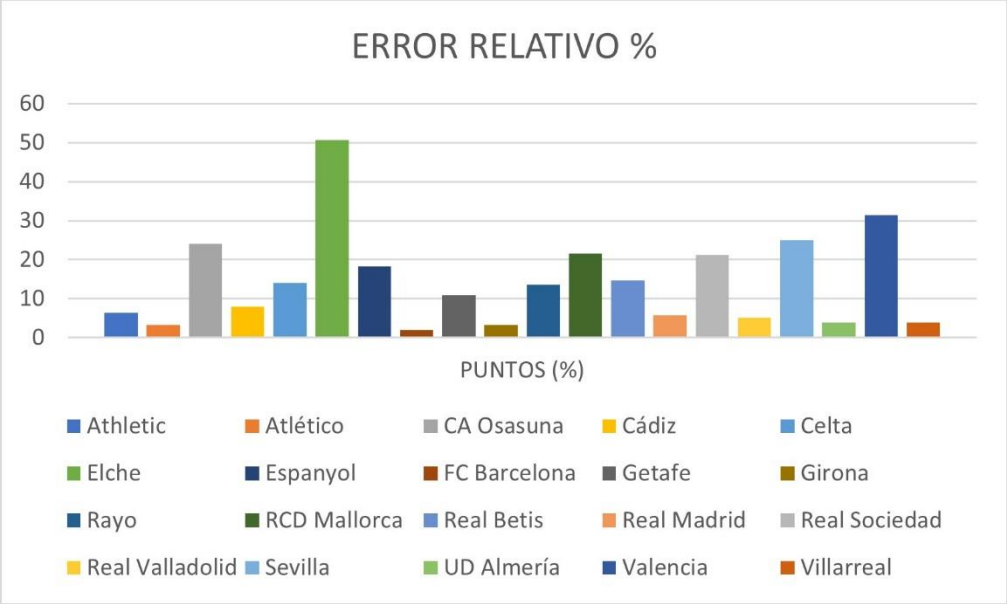

Ilustración 23 - Histograma donde se representan los errores relativos del cálculo de los puntos

4.7.2. GOLES.

Al igual que con los puntos también podemos valorar el pronóstico goleados de la temporada para cada equipo. En este caso es de esperar que los errores sean relativamente mayores ya que es un evento mucho más aleatorio. Aunque se haya intentado modelizar siempre mantendrá un componente de aleatoriedad impredecible.

Equipo	GF reales	GF calculados	ErrorRelativo %	GC reales	GC calculados	ErrorRelativo %
Athletic	47	42,453	9,674468085	43	39,387	8,402325581
Atlético	70	59,412	15,12571429	33	27,38	17,03030303
CA Osasuna	37	35,63	3,702702703	42	53,753	27,98333333
Cádiz	30	31,211	4,036666667	53	50,156	5,366037736
Celta	43	44,873	4,355813953	53	50,654	4,426415094
Elche	30	32,628	8,76	67	54,374	18,84477612
Espanyol	52	38,647	25,67884615	69	52,33	24,15942029
FC Barcelona	70	85,143	21,63285714	20	28,932	44,66
Getafe	34	34,293	0,861764706	45	42,424	5,724444444
Girona	58	44,257	23,69482759	55	52,19	5,109090909
Rayo	45	40,281	10,48666667	53	56,471	6,549056604
RCD Mallorca	37	34,061	7,943243243	43	55,61	29,3255814
Real Betis	46	45,895	0,22826087	41	47,73	16,41463415
Real Madrid	75	81,411	8,548	36	31,855	11,51388889
Real Sociedad	51	47,211	7,429411765	35	42,047	20,13428571
Real Valladol	33	30,485	7,621212121	63	50,962	19,10793651
Sevilla	47	52,921	12,59787234	54	39,878	26,15185185
UD Almería	49	38,414	21,60408163	65	60,624	6,732307692
Valencia	42	48,117	14,56428571	45	44,33	1,488888889
Villarreal	59	52,449	11,10338983	40	38,705	3,2375

Ilustración 24 - Tabla donde aparecen los goles calculados con sus errores relativos y absolutos, tanto a favor como en contra.

A simple vista se puede observar que el error de los goles recibidos, GC, es mayor que el de los goles a favor, GF. A continuación, se pueden observar las dos gráficas como se ha realizado en el apartado de los puntos, para obtener una perspectiva más visual de los datos.

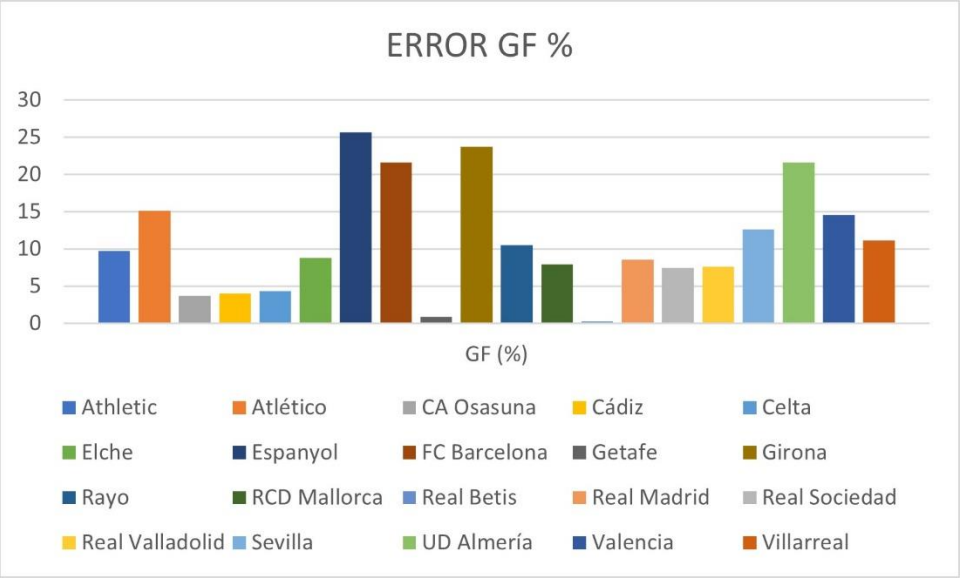

Ilustración 25 - Histograma donde se representan los errores relativos del cálculo de los goles a favor.

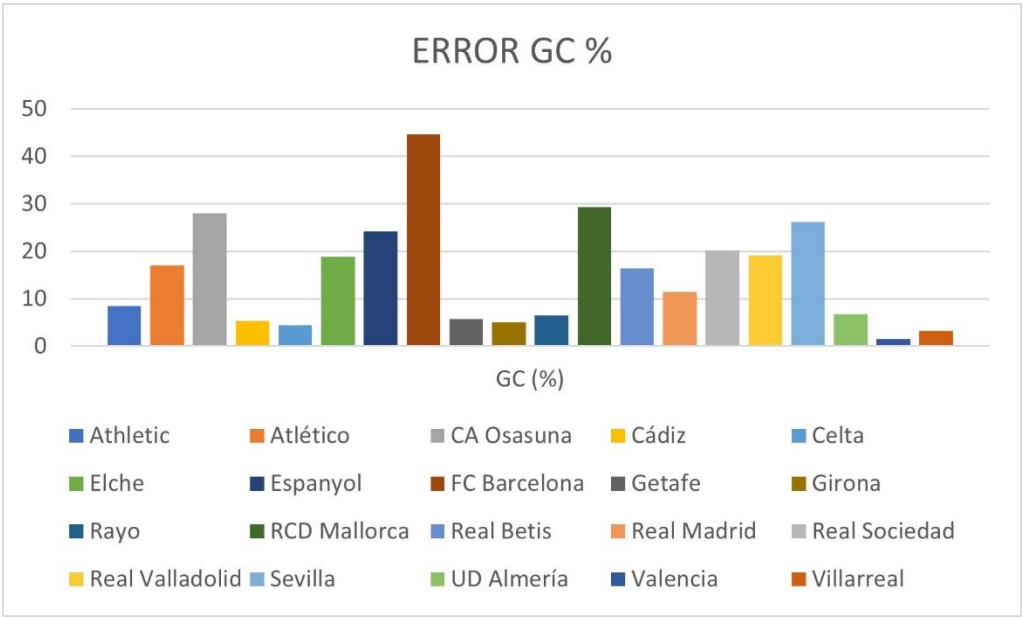

Ilustración 26 - Histograma donde se representan los errores relativos del cálculo de los goles en contra.

En este caso, como se ha mencionado al principio del capítulo, los errores son de esperar que sean más altos que los obtenidos al predecir los puntos, pero esto no es del todo así. En concreto los errores relativos medios para los goles a favor y en contra son 10.98% y 15.12%,

respectivamente, esto habla bien de la robustez del modelo en cuanto a predecir de forma similar todas las variables.

En este caso, los errores absolutos, es decir las diferencias entre reales y calculados serían de 5.72 goles a favor, y de 6.81 goles en contra. Estos valores, teniendo en cuenta la larga duración de la temporada se consideran buenas estimaciones ya que equivaldrían a un error de 0.15 y 0.18 goles por partido, algo prácticamente insignificante.

4.7.3. PREDICCIONES DE CLASIFICACIÓN.

Otra de las funcionalidades que ofrece este modelo es, al realizar 1000 iteraciones, tratar de estimar la probabilidad de cada equipo de acabar la temporada en cada una de las 20 posiciones del campeonato. Para esto, dentro del bucle se creo una matriz llamada *tabla_posiciones* donde se fueron almacenando, para cada una de las iteraciones la posición en la que terminaba cada equipo.

```
frecuencias = apply(tabla_posiciones, 1, table)
```

Se aplica la función table a cada fila de *tabla_posiciones*. Esto genera una lista de tablas de frecuencia para cada fila, donde las tablas representan la cantidad de veces que aparece cada valor en esa fila.

```
tabla_frecuencias=as.data.frame(bind_rows(frecuencias))
tabla_frecuencias[is.na(tabla_frecuencias)]<-0
nom_col=as.integer(colnames(tabla_frecuencias))
for (e in 1:20){
  a=as.integer(nom_col[e])
  nom_columna=equipos_estudio[a]
  colnames(tabla_frecuencias)[e]=nom_columna
}
```

Se crea un dataframe a partir de las tablas de la variable frecuencias, se reemplazan los valores NA por 0 para que no de error, posteriormente convierte los nombres de las columnas a enteros e itera sobre las 20 primeras columnas asignando a cada una el nombre de uno de los equipos.

```
posiciones=paste("Posición",c(1:20))
rownames(tabla_frecuencias)=posiciones
```

A las filas se les asigna un nombre combinando “Posición” con un número del 1 al 20

```
probabilidades=100*(tabla_frecuencias/replicaciones)[,seq(2,40,2)]
colnames(probabilidades)=colnames(tabla_frecuencias)
rownames(probabilidades)=rownames(tabla_frecuencias)
```

Para calcular la probabilidad no nos vale con la frecuencia, sino que hay que dividir cada frecuencia entre el número de iteraciones, es decir 1000, para después multiplicar por 100 para llevarlo a un porcentaje. El resultado de este fragmento de código es una tabla que traspuesta y combinada con un mapa de calor nos da como resultado la imagen que se muestra a continuación.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
FC Barcelona	62,20	26,80	9,80	0,90	0,20		0,10													
Real Madrid	30,20	45,30	20,00	3,20	1,10	0,10	0,10													
Atletico Madrid	7,00	24,50	49,30	12,60	4,00	1,50	0,50	0,50			0,10									
Real Sociedad	0,20	0,40	1,70	9,30	13,20	12,70	13,00	12,00	9,70	9,40	5,00	3,00	3,00	2,90	2,00	0,70	1,20	0,30	0,20	0,10
Real Betis	0,10		0,40	3,80	6,50	8,70	9,50	10,60	9,20	9,50	9,30	8,00	5,90	5,40	3,40	3,50	3,10	1,30	1,10	0,70
Sevilla FC	0,10	1,30	6,80	24,60	19,20	13,90	10,10	6,50	6,20	3,80	3,30	1,70	1,20	0,80	0,20	0,20	0,10			
Villarreal CF	0,20	1,50	8,50	23,80	19,80	13,80	11,30	6,60	4,90	3,30	2,30	1,90	0,70	0,90	0,20	0,10	0,10		0,10	
Athletic Bilbao		0,10	1,40	7,10	11,80	13,90	12,80	13,30	10,00	7,30	7,00	5,30	3,10	2,40	1,90	0,70	1,00	0,60	0,20	0,10
Valencia CF		0,10	1,60	8,80	10,90	15,10	13,30	10,60	11,40	7,70	4,50	5,50	3,30	1,20	1,60	1,80	1,30	0,80	0,10	0,40
Celta Vigo			0,20	2,30	5,20	6,50	8,70	10,30	9,80	10,20	7,40	7,60	7,90	5,60	4,70	4,10	3,10	2,60	2,70	1,10
Espanyol Barcelona			0,10	0,50	1,10	1,50	2,40	4,40	5,30	7,10	9,10	7,20	9,30	9,10	9,00	7,20	8,80	7,30	6,00	4,60
Girona			0,20	1,90	3,10	5,80	6,50	8,30	8,80	8,50	9,20	10,10	6,60	7,90	7,40	5,60	3,60	3,10	2,10	1,30
CA Osasuna				0,10	0,50	0,70	1,60	2,10	3,20	3,90	5,70	6,90	7,90	9,60	10,60	11,00	10,10	9,70	8,50	7,90
Getafe CF				0,80	2,20	3,00	3,80	6,20	7,50	8,30	10,00	9,90	10,40	8,20	8,70	5,10	4,40	6,40	3,50	1,60
Rayo Vallecano				0,10	0,50	0,70	1,80	2,10	3,70	6,10	5,70	7,40	8,40	9,80	7,50	11,70	9,50	7,60	10,20	7,20
RCD Mallorca				0,10	0,20	0,30	1,40	1,50	2,30	3,20	4,90	4,60	6,80	8,50	9,80	10,10	10,00	13,50	11,20	11,60
UD Almeria				0,10		0,80	1,30	1,70	2,00	3,00	4,40	4,80	7,20	7,30	9,10	11,10	11,70	11,90	10,80	12,80
Cadiz CF					0,10	0,70	0,50	1,30	2,10	3,10	5,50	5,90	8,00	8,70	8,10	8,80	8,50	10,40	13,60	14,70
Elche CF					0,40	0,10	0,50	1,30	2,20	3,10	3,30	4,40	6,30	6,00	7,80	9,10	12,80	12,40	13,70	16,60
Real Valladolid						0,20	0,80	0,70	1,70	2,50	3,30	5,80	4,00	5,70	8,00	9,20	10,70	12,10	16,00	19,30

Ilustración 27 - Mapa de calor donde se ven las probabilidades de cada equipo de quedar en cada una de las 20 posiciones de la clasificación.

En este caso se ha creado un mapa de calor, con un gradiente de color con los colores rojo, naranja y blanco para los valores máximos, medios y mínimos respectivamente, con las probabilidades para poder entenderlo con mayor claridad. Se puede observar como el FC Barcelona era el equipo con mayor probabilidad de ganar la liga con un 62.2% seguido del Real Madrid con un 30% y del Atlético de Madrid con un 7%. Respecto a la parte baja de la tabla podemos ver que hay mayor igualdad en los datos, el más probable de quedar en última posición en este caso es el Real Valladolid con un 19.3% seguido del Elche CF con un 16.6%.

Aunque, como en todas las predicciones que hemos hecho siempre hay algún dato que no entra dentro de lo esperado, se puede decir que las aproximaciones son bastante acertadas.

Para seguir desarrollando estas predicciones se desglosarán algunas de los objetivos que se podrían marcar antes de la temporada algunos de los equipos.

- <u>Ganador de la competición:</u>

  Se ha predicho que la temporada iba a tener un claro ganador que sería el FC Barcelona (62.20%), el conjunto madridista (30.20%) también se convertiría en un candidato al título, aunque con la mitad de probabilidad que el FC Barcelona. Existen también probabilidades más remotas como una victoria del Real Betis (0.10%), del

Sevilla Club de Fútbol (0.10%) o del Villareal (0.20%), aunque como nos dicen los datos es algo bastante complicado.

- Clasificación para la Liga de Campeones:

La Liga de Campeones es una competición europea en la que se enfrentan los mejores equipos de las mejores ligas europeas como son la española, la inglesa, la francesa, la italiana o la alemana. En el caso de la competición española, los 4 primeros clasificados son los que consiguen el pase europeo.

En este caso el Real Madrid tendría asegurada la clasificación con más de un 90% de probabilidad, al igual que el FC Barcelona y Atlético de Madrid. Realmente la disputa estaría para la cuarta y última plaza donde Sevilla (32.80%), Villareal CF (34.00%) o Real Sociedad (11.60%) son los mejor posicionados para conseguirlo seguidos de otros clubes como el Athletic Club de Bilbao (8.60%), Valencia (10.50%) o Real Betis (4.30%); aunque también existen opciones remotas como la clasificación del Girona (2.10%) o del Celta de Vigo (2.50%).

- Descenso a la Segunda división:

Como se comentó al principio del trabajo, los últimos tres clasificados al final de la temporada descienden a la Segunda División. En esta parte de la tabla suele estar la cosa más reñida, en este caso existen siete equipos con más de un 25% de probabilidad de caer a la Segunda División. Estos equipos son el Real Valladolid (47.40%), el Elche CF (42.70%), el Cádiz CF (38.70%), el UD Almería (35.50%), el RCD Mallorca (36.30%), el CA Osasuna (26.10%) y el Rayo Vallecano (25.00%). También hay otras opciones como que descienda el Celta (6.40%) o el Espanyol (17.9%).

5. POSIBLES APLICACIONES.

Tras haber visto los resultados que nos ofrece el modelo no estaría de más comentar algunas de las aplicaciones que estos resultados podrían ofrecer tanto a los aficionados como a los propios clubes.

- Predicciones y pronósticos deportivos: El uso más evidente de obtener una predicción sobre los resultados de una temporada es este. Estas predicciones pueden ser de interés para apostadores, aficionados, casas de apuestas incluso para medios de comunicación deportivos. La capacidad de proporcionar predicciones precisas puede generar un interés adicional en el torneo.

- Gestión de riesgos y estrategia: Esta aplicación no está tan enfocada en cuanto a los resultados a final de temporada sino en cuanto a jornadas en específico. Equipos, entrenadores y directores técnicos podrían utilizar el modelo para evaluar riesgos y diseñar estrategias para la temporada. Conocer las probabilidades de ganar, perder o empatar en diferentes escenarios puede ayudar en la toma de decisiones tácticas y estratégicas.

- Análisis de desempeño de equipos y jugadores: El modelo podría utilizarse para evaluar el rendimiento esperado de equipos y jugadores a lo largo de la temporada. Esto podría ser valioso para los equipos al planificar estrategias de juego, realizar transferencias o ajustar tácticas según las probabilidades de éxito.

- Optimización de recursos financieros: Clubes y organizaciones deportivas podrían utilizar el modelo para gestionar sus recursos financieros de manera más eficiente. Con predicciones precisas, podrían asignar presupuestos de manera más inteligente en función de las expectativas de rendimiento.

- Participación del público y engagement: La publicación de predicciones al principio de la temporada puede aumentar la participación del público y el engagement en las plataformas digitales y sociales. Los aficionados podrían participar en concursos de pronósticos o seguir de cerca las actualizaciones basadas en el rendimiento predicho.

- Extensión a otras ligas o competiciones: El mismo modelo se podría aplicar a otras competiciones como puede ser la Segunda División o a competiciones ligueras de otros países siempre y cuando el formato de los datos con los que se trabaje sea el mismo que se ha empleado durante este trabajo.

Es importante destacar que la efectividad y la relevancia de estas aplicaciones dependerán de la calidad y la precisión de modelo. Además, la ética y la transparencia en la comunicación de las predicciones son aspectos fundamentales, especialmente cuando se trata de apuestas y pronósticos deportivos.

6. POSIBLES LÍNEAS FUTURAS.

La principal línea de mejora del modelo es la mejora de la efectividad y precisión de los resultados que brinda. Para ello se podrían proponer opciones como:

- Consideración de factores dinámicos: Explorar la inclusión de factores dinámicos que puedan cambiar a lo largo de la temporada como lesiones de jugadores clave, cambios en la dirección técnica o el rendimiento en competiciones paralelas podría brindar al modelo una mayor capacidad de adaptación ante cambios en tiempo real que podría mejorar la precisión.

- Incorporación de nuevas variables: Se podría analizar la inclusión de datos contextuales más ricos como condiciones meterológicas, desplazamientos de los equipos o estadísticas de posesión de balón, entre otros.

- Uso de Modelos Bayesianos e incertidumbre: Explorar enfoques basados en modelos bayesianos que permitan cuantificar y comunicar la incertidumbre asociada con las predicciones. Esto puede resultar útil para brindar una perspectiva más completa a los usuarios finales sobre la confiabilidad de las estimaciones.

Otra línea de investigación, relacionada en gran parte con las posibilidades que se acaban de mencionar, es poder realizar la predicción en un punto de la temporada. Esto podría interesar en caso de que el comienzo no haya sido como estaba previsto y los resultados que se han ido obteniendo no eran los esperados.

Habiendo comprobado que este modelo se puede extender a la temporada que se quiera podemos aplicarla a esta temporada en curso, a la temporada 2023-24. En este caso, los datos de esta temporada no se han podido extraer de ninguna base de datos, sino que se han tenido que ir actualizando tras cada jornada disputada.

El último registro que se tiene es de la jornada 21, la disputada el fin de semana del 21 y de enero. Existe un pequeño problema ya que por la disputa de la Supercopa de España hay tres partidos aplazados que se jugarán más adelante, por tanto, estos seis equipos tendrán un partido jugado menos.

El procedimiento a seguir es similar al que se ha seguido durante todo el trabajo, primero se realizaría una recolección, tratamiento, limpieza y organización de los datos para facilitar los posteriores cálculos.

Para esta parte los valores de λ se calcularán en parte con los datos de la misma temporada, combinados con los datos de temporadas anteriores. Para ello se separarán los datos de esta temporada en dos tablas: *temp_actual,* con los partidos ya disputados, y *temp_pred*, con las jornadas que todavía no se han disputado y se pretenden predecir.

```
dia_estudio = as.Date("2024/01/13")
temp_actual = data[(data$Season == 2023 & data$Date<dia_estudio),]
temp_actual

temp_pred = data[(data$Season == 2023 & data$Date>dia_estudio),]
```

El procedimiento es elegir un día en el que se separarán los datos y a partir de ahí coger las observaciones de la taba inicial *data* con un valor de fecha mayor o menos que el día escogido y separarlo en temporada para predecir y actual, respectivamente.

A partir de aquí se procede a realizar lo mismo que antes: separar las observaciones dependiendo de si se actúa de local o de visitante y organizar una tabla con formato de clasificación.

Con estas tablas, se procede a calcular las variables necesarias para el cálculo de las λ de cada uno de los enfrentamientos para guardarlos posteriormente en una tabla que llamaremos *PRElandas_actual1*. El resultado de esta tabla quedaría como se puede ver en la siguiente imagen.

	equipos	CAC	CDC	CAF	CDF
1	Athletic Bilbao	1.6453382	0.9107007	0.9024613	0.6014581
2	Atletico Madrid	1.5417428	0.9182899	1.3947129	0.8019441
3	CA Osasuna	0.7312614	1.0624842	1.4585232	1.3365735
4	Cadiz CF	0.7373553	0.9182899	0.5966686	0.9720535
5	CD Alaves	0.6703230	1.3356944	0.4475015	1.0935601
6	Celta Vigo	0.4875076	1.0624842	1.0665451	1.1360875
7	FC Barcelona	1.4076782	0.9182899	1.5587967	0.8687728
8	Getafe CF	0.6703230	0.5565393	1.2679208	1.3365735
9	Girona	1.8281536	1.0624842	1.7228806	0.7351154
10	Granada CF	0.8531383	1.5178346	0.6563355	1.4702309
11	Rayo Vallecano	0.6032907	0.6678472	0.7458358	1.0328068
12	RCD Mallorca	0.6032907	1.5026562	0.7383774	0.5346294
13	Real Betis	0.9140768	0.6071338	0.7383774	1.0692588
14	Real Madrid	1.6758074	0.5008854	1.4767548	0.4678007
15	Real Sociedad	1.2065814	0.9182899	1.0441701	0.6075334
16	Sevilla FC	0.9140768	1.3660511	0.9024613	1.1360875
17	UD Almeria	0.6703230	1.5026562	0.8204193	1.7010936
18	UD Las Palmas	0.8714199	0.5843663	0.5966686	0.6075334
19	Valencia CF	0.8714199	0.4174045	1.0441701	1.1543135
20	Villarreal CF	1.0968921	1.6696180	0.8204193	1.3365735

Ilustración 28 - Tabla con los valores de λ para la temporada actual.

Una novedad en esta aplicación del modelo es la aplicación de pesos, entendiéndose pesos como los coeficientes que permiten dar más importancia, o menos a los valores más recientes.

$$\lambda = \theta_1 \cdot \lambda_{\text{actual}} + \theta_2 \cdot \lambda_{\text{antiguo}}$$

Ecuación 5 - Ecuación donde se muestran los coeficientes de la ecuación de las nuevas landas.

Pero es en este momento cuando llegaría el problema ya que usar el mismo modelo con menos de 20 observaciones no proporcionará unos resultados que se puedan tomar como aceptables, es por ello por lo que sería necesaria la confección de otro modelo con nuevas variables para poder calcular unos nuevos valores de λ_{actual} que nos permitan obtener resultados más fiables. Es por esta razón no se ha aplicado a esta temporada, sino que se ha valorado como una posible línea de investigación, ya que la obtención y aplicación del nuevo modelo podría tener la misma, o una mayor, complejidad que el que se ha utilizado.

7. PLANIFICACIÓN Y PRESUPUESTO.

7.1. PLANIFICACIÓN.

La planificación temporal para el Trabajo de Fin de Grado se ha reajustado en varias ocasiones, esto se debe a complicaciones que han ido surgiendo tanto a la hora de abordar este tema, como por razones externas como el suspenso de una asignatura, lo cual me impedía presentar el trabajo en la convocatoria extraordinaria de noviembre.

Además, se suma la propia carga de trabajo del curso de estudios y la participación en actividades extrauniversitarias que sumaban carga horaria. No obstante, a continuación, se muestran un reflejo temporal, en algunos casos más fiel que en otros, de los distintos periodos temporales del desarrollo del trabajo.

A continuación, se muestra un Diagrama de Gantt donde se puede ver el desarrollo del trabajo. Posteriormente se desarrollará cada una de las tareas.

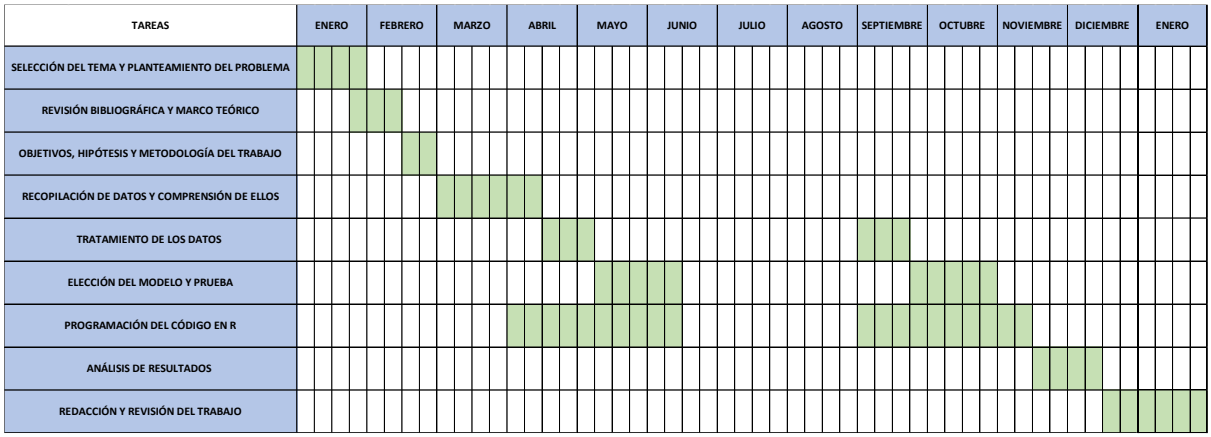

Ilustración 29 - Diagrama de Gantt donde se muestra el desarrollo del trabajo.

- TAREA 1 - Selección del tema y planteamiento del problema: El trabajo como se ha comentado en el resumen nace de una idea de poder analizar los datos que nos deja el fútbol. Al principio se planteó la idea de realizar un modelo para estudiar no tanto los resultados sino los eventos de un partido de fútbol, temas como xG (Goles esperados) o medir la calidad de los pases en función de la posición inicial y final. Finalmente, por la escasez de los datos necesarios para los proyectos anteriores, se decidió empezar con este tema. Durante este proceso se tuvieron varias tutorías donde se debatieron estos temas y se vieron las alternativas.

- TAREA 2 – Revisión bibliográfica y marco teórico: Una vez elegido el tema quedaba buscar en la red cuál era el estado de esta investigación. Se encontraron varios trabajos y modelos que abordaban el tema.

- TAREA 3 – Objetivos, hipótesis y metodología del trabajo: Una vez visto el alcance del problema se definieron los objetivos que en un principio se querían alcanzar, aunque finalmente estos hayan sido dinámicos ya que iban cambiando conforme avanzaba el trabajo; las hipótesis y la metodología con las que se trabajaría.

- TAREA 4 – Recopilación y comprensión de los datos: Como se ha dejado entender en el capítulo sobre la obtención de datos, estos han sido extraídos de varias fuentes. Uno de los problemas iniciales fue comprender qué servicios ofrecía el paquete *engsoccerdata* y cuáles eran realmente valiosos para nuestro trabajo. Una vez elegidos los datos elegidos, había que entender cuáles de las columnas nos interesaban al igual que entender qué información nos aportaba cada una. En la última parte de esta tarea ya se empezó a trabajar con RStudio para poder ir "trasteando" con los datos.

- TAREA 5 – Tratamiento de los datos: Una vez recopilados los datos, el siguiente paso sería convertir los datos al formato con el que se trabajará posteriormente. Esta tarea fue realizada completamente en RStudio.

- TAREA 6 – Elección del modelo y prueba: Cuando ya se tenían los datos preparados y habiendo revisado la bibliografía existente sobre el tema el siguiente paso sería empezar a construir el modelo. Tras varias tutorías con el tutor y bastantes intentos fallidos, se llegó al modelo final.

- TAREA 7 – Programación del código en R: Aunque esta tarea sea algo transversal ya que la mayor parte del desarrollo del trabajo se realiza en este programa se ha decidido incluir una tarea única ya que la extensión de esta es grande y que ha habido etapas únicamente de revisión y optimización del código.

- TAREA 8 – Análisis de resultados: Después de haber conseguido el modelo y finalizado el código de RStudio, se procedió a realizar un análisis de los resultados obtenidos para comprobar si eran válidos o no.

- TAREA 9 – Redacción y revisión del trabajo: Con todas las tareas anteriores completadas con éxito, el siguiente paso era comenzar a redactar lo que sería este documento.

Como se puede observar en el Diagrama de Gantt hay un espacio de dos meses vacíos, esto se debe a que para entregar el Trabajo de Fin de Grado había que tener todas las asignaturas aprobadas y después de suspender una asignatura en la recuperación, la próxima fecha disponible para poder entregarlo pasaba a ser el 4 de febrero por tanto se decidió retomar el trabajo en septiembre. A partir de septiembre se compatibilizó el desarrollo del trabajo con el estudio de la asignatura restante, un máster fuera de la UPM y de unas prácticas extracurriculares, por lo que el ritmo de trabajo se ralentizó.

Durante todo el trabajo ha habido un seguimiento por parte del tutor con una tutoría cuando ha sido necesario o por medio de correos electrónicos.

7.2. PRESUPUESTO.

El presupuesto para la realización del Trabajo de Fin de Grado puede variar dependiendo de varios factores, como el salario del tutor y el alumno, el coste del uso de programas o de otros recursos.

El TFG consta de 12 ECTS, suponiendo que cada crédito equivale a 30 horas de trabajo, el número de horas dedicadas sería de 360 horas. Suponiendo que no han sido exactas, se valorará una variación del 10% de las horas, es decir, se supondrán alrededor de 396 horas.

Respecto al tutor, se han tenido un total de 5 tutorías, estimando cada una de ellas con una duración de dos horas y media, se calculan un total de 12 horas en tutorías a las que habría que sumarles el trabajo realizado por el tutor y la revisión del documento, sumando un total de 40 horas, lo equivalente al 10% del trabajo.

Respecto al uso del programa de uso principal RStudio como otros como Microsoft Word o Microsoft Excel su coste es nulo, ya que la licencia que se ha usado es gratuita, por tanto, esto no suma ningún coste. Tampoco añade ningún gasto añadido el conseguir los datos ya que la base de datos es de uso público.

Habría que tener en cuenta el costo de la matrícula del TFG que serían un total de 233€. El equipo informático sería un ordenador portátil HP con un precio inicial de 499.00€ al que se le supondrá una vida útil de 4 años funcionando 8 horas por día los 250 días laborales, es decir, una vida útil de 800 horas; respecto al uso se supone un 60% del desarrollo, lo equivalente a horas. Se supondrá que la potencia requerida por el ordenador es de 190 W. Para el precio de la electricidad se asumirá un coste medio de 0.1823€/kWh.[5]

A continuación, se mostrarán los valores con los aspectos detallados del coste total, siendo **P** la potencia del ordenador, **Pi** el precio inicial del ordenador y **Vu** la vida útil estimada del ordenador.

PRESUPUESTO FINAL					
Concepto		Nº horas (h)	Coste Unitario (c)	Fórmula	Total
Personal	Alumno	396	15 €/h	c*h	5.940 €
	Tutor	40	25 €/h	c*h	1.000 €
Material	Matrícula	-	-	-	233 €
	Portátil	237,6	0,1823 €/kWh.	P*c*h	8,22 €
Amortización	Portátil	-	124,75 €/año	Pi*h/Vu	14,85 €
COSTE TOTAL SIN IVA					7.196 €
COSTE TOTAL CON IVA (21%)					8.707 €

Ilustración 30 - Tabla que muestra los costes detallados del presupuesto final total.

Por tanto, observando la tabla concluimos que el precio final del Trabajo de Fin de Grado sería de **8707 €.**

8. CONTRIBUCIÓN A LOS ODS.

En septiembre del año 2015 líderes mundiales se reunieron en la Asamblea General de las Naciones Unidad con el objetivo de implantar en sus respectivos países la Agenda 2030. Esta agenda incluye 17 objetivos llamados “Objetivos de Desarrollo Sostenible” o por sus siglas “ODS”.

Estos objetivos engloban una amplia gama de desafíos globales como erradicar la pobreza o luchar contra el cambio climático, pasando por conseguir una educación de calidad y agua limpia a disposición de toda la población mundial. Los ODS no buscan únicamente mejorar la calidad de vida de las personas, sino también proteger el plantea y asegurar la prosperidad para las generaciones presentes y futuras. [6]

Ilustración 31 - Objetivos de Desarrollo Sostenible.

Aunque aparentemente la contribución con los ODS sea escasa, explorando cada uno de los objetivos se podría decir que el desarrollo de un modelo de probabilidad para el análisis de resultados de la Liga de Fútbol Española puede contribuir a varios Objetivos de Desarrollo Sostenible (ODS), dependiendo de cómo se enfoque y utilice la información generada por dicho modelo.

En concreto tiene efecto sobre estos objetivos:

- ODS 9: Industria, innovación e infraestructuras: El desarrollo de un modelo de probabilidad para el análisis deportivo puede considerarse como una contribución a la innovación en el ámbito deportivo, mejorando la comprensión y la toma de decisiones.

- ODS 4: Educación de calidad: En caso de que este modelo pueda ser utilizado para educar a los aficionados, entrenadores o analistas sobre los factores que

influyen en los resultados de los partidos, se podría argumentar que también se está haciendo una aportación a este objetivo.

También podríamos decir que contribuye con otros objetivos, aunque de forma más rebuscada; como pueden ser el ODS 16: Paz, justicia e instituciones sólidas o con el ODS 3: Salud y bienestar.

BIBLIOGRAFÍA.

[1]: https://www.libertaddigital.com/deportes/liga/1999-2000/

[2]: https://rpubs.com/joser/goles11_12

[3]: https://www.kaggle.com/datasets/kishan305/la-liga-results-19952020

[4]: https://www.resultados-futbol.com/primera2023

[5]: https://datosmacro.expansion.com/energia-y-medio-ambiente/electricidad-precio-hogares/espana#:~:text=El%20precio%20medio%20de%20la,que%20en%20el%20periodo%20previo

[6]: https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/

Autores, V. (2023). Distribución de Poisson. *Wikipedia*.

Castro Montaño, J. A. (29 de 07 de 2022). ¿Cuántas ligas de fútbol hay en el mundo y cuáles son las mejores? *ANTENA 2*.

FIFA Communications Division, I. S. (31 de 05 de 2007). FIFA Big Count 2006: 270 million people active in football. pág. 12.

Giovio, E. (18 de Diciembre de 2016). Cristiano Ronaldo: “Las estadísticas no engañan...”. *El País*.

Goddard, J. (2004). Regression models for forecasting goals and match results in association football. *Science Direct*.

Gómez, V. (22 de Febrero de 2023). El Elche de la temporada 22-23 tiene el peor puntaje del siglo XXI. *AS*.

González, J. A. (30 de Diciembre de 2023). El Elche CF despide un 2023 para olvidar. *Tele Elx*.

Ibas, J. C. (2023). Estimación de resultados deportivos mediante modelos lineales generalizados. *Universitat de Barcelona*.

Maher, M. J. (1982). Modelling association football scores . *90 minut*, 10.

Marugán, H. (07 de Enero de 2024). Imanol alguacil, la cara visible del éxito de una Real Sociedad que asombra a Europa. *El Debate*.

Rodrigo, M. (23 de Diciembre de 2023). El balance de la Real Sociedad en 2023: un año para enmarcar. *Noticias de Gipuzkoa*.

Sheehan, D. (2018). Predicting Football Results With Statistical Modelling: Dixon-Coles and Time-Weighting.

Varios, A. (2023). Red Bayesiana. *Wikipedia*.

ÍNDICE DE FIGURAS.

ÍNDICE DE ECUACIONES.

ESCUELA TÉCNICA SUPERIOR DE INGENIEROS INDUSTRIALES
UNIVERSIDAD POLITÉCNICA DE MADRID

José Gutiérrez Abascal, 2. 28006 Madrid
Tel.: 91 336 3060
info.industriales@upm.es

www.industriales.upm.es