

Organización óptima de fases de grupo en competiciones deportivas



Sergio Anaut Aineto
Trabajo de fin de grado en Matemáticas
Universidad de Zaragoza

Director del trabajo: Javier Lopez Lorente
28 de junio de 2021

Abstract

In this work we focus on the optimal organization of group stages on football competitions, more specifically on the UEFA Champions League and the UEFA Europa League.

There are many models that are used to predict the results of a football match, in our case, we will use Poisson regression to predict the number of goals that a team will score when facing another team, based on the characteristics such as the team's value at the beginning of the season. Our model will use the following equation:

$$\lambda = e^{\mathbf{x}\beta} = e^{\beta_0 + \beta_1 X_1 + \dots + \beta_k X_k} \quad (1)$$

where λ is the Poisson parameter, the β_i 's will be the regression coefficients, extracted by maximum likelihood, corresponding to our predictor variables X_j . We will use this model to predict the goals scored by the home team, the away team and we have added a case to analyze single matches.

Then we will check that our results can be considered acceptable by comparing the number of actual home wins, away wins and draws estimated with our prediction.

Once the estimates have been obtained, we will proceed to the simulation of the group stages using program made in Pascal, which will yield the number of collision and inconsequential matches and the number of times that each team would qualify depending on the type of draw, the order in which the matches are played, the number of qualified teams per group and the size of the group. We will analyse the results and look for the combination that allows us to obtain a greater number of interesting matches and a competition as fair as possible. Moreover, we will purpose a different method to run the teams which will analyse the fairness of the tournaments.

For learning Poisson regression techniques, I have used the books [2], [11], [8]. For programming in Pascal, I have used the book [10].

Índice general

Abstract	III
1. Introducción	1
2. Modelo de regresión de Poisson	3
2.1. Introducción	3
2.2. Presentación del modelo	4
2.3. Obtención de coeficientes del modelo de goles del equipo local	5
2.4. Obtención de coeficientes del modelo de goles del equipo visitante	7
2.5. Validación de los modelos	9
2.6. Obtención de coeficientes para partido único	12
3. Análisis de formación de grupos	13
3.1. Explicación de la simulación	13
3.2. Análisis de los partidos intrascendentes y de colisión	15
3.3. Análisis de la justicia del sistema	21
3.3.1. Mejoras para obtener un modelo más justo	23
3.4. Conclusiones	24
A. PASCAL	27
B. TABLAS	85
C. R-STUDIO	113
Bibliografía	121

Capítulo 1

Introducción

Hoy en día todos sabemos que el fútbol es el deporte más popular y seguido del mundo, solo hace falta ver cómo miles de personas se reunían, al menos antes de la pandemia, en bares y casas cuando se iba a disputar un partido importante. Por hacernos una idea de la popularidad de este deporte, basta con fijarnos en la expectación que tuvo la final del Mundial de Rusia en 2018 con alrededor de 1120 millones de espectadores o el “Clásico” entre Real Madrid y FC Barcelona en 2017 con cerca de 650 millones, comparada con la Superbowl 2020 con 100 millones de espectadores o con las finales de los play-offs de la NBA con 15 millones de espectadores por partido. Esto hace que sea muy difícil encontrar gente que no tenga las ideas básicas sobre cómo se practica este deporte, cuyo principal objetivo es marcar un número de goles mayor que el equipo rival en un periodo de tiempo de 90 minutos.

Los orígenes del fútbol moderno comenzaron en Gran Bretaña a partir de 1863 con la Asociación Inglesa de Fútbol y gracias a su facilidad para practicarlo y al desarrollo de las colonias inglesas por todo el mundo, consiguieron que el fútbol se convirtiera en el deporte más jugado del planeta. Actualmente existen grandes cantidades de información como los resultados de otras temporadas, jugadores que estaban lesionados en ese partido, el número de goles que marcó y recibió un equipo, el valor monetario de la plantilla... Todos estos datos que están al alcance de todo el mundo se pueden aprovechar para crear modelos matemáticos que nos permitan predecir resultados de los próximos partidos y que nos dan probabilidades de ocurrencia muy cercanas a la realidad. Es por ello, que cada vez más equipos de fútbol buscan expertos en análisis de datos, con el fin de obtener unos resultados más positivos en temas de fichajes, reducir lesiones o sacar patrones sobre las características de juego de un equipo.

El análisis de datos en el fútbol se está utilizando con diversos objetivos. Entre ellos destacan la predicción de resultados para apuestas deportivas como se observa en el trabajo de Sabroso [12] o en el de Iturralde [5], la optimización en la organización de torneos como hicieron P. Laliena y F.J. López en [6] o los formatos óptimos para mantener la competitividad [1]. En este trabajo nos centraremos en esta última aplicación y usaremos herramientas matemáticas para analizar la mejor manera de organizar las fases de grupos en torneos de fútbol, de forma que se cumplan simultáneamente los requisitos de justicia (los mejores equipos tienen más posibilidad de pasar a la siguiente ronda) e interés para el público (se minimizan el número de partidos en los que no hay nada en juego o en los que un resultado pactado de antemano beneficia a los dos contrincantes). En los últimos años han aparecido en la literatura algunos trabajos en esta línea. Así, Guyon [3] estudia las posibles situaciones en las que pueden existir partidos de colisión, analiza la probabilidad de que existan estos dando unas probabilidades predeterminadas de victoria a cada equipo según su ‘fuerza’ y propone otros sistemas de puntuación analizando el riesgo de colisión para cada uno de estos casos.

El trabajo lo dividiremos en dos partes. La primera consistirá en crear un modelo matemático que nos permita estimar el número de goles que marcará un equipo A al enfrentarse a un equipo B. Dicho estudio lo realizaremos mediante el modelo de regresión de Poisson y para el cual utilizaremos variables

predictoras sencillas.

El modelo para el número de goles lo usaremos en la segunda parte del trabajo, en la que evaluaremos distintas configuraciones de torneos. Para ello, dada una configuración, realizaremos una simulación de Monte Carlo para obtener indicadores de 'justicia' e 'interés'. Analizaremos varias organizaciones de fases de grupos de torneo en función de sus características, como el número de equipos que participan, tamaño de los grupos, número de equipos que pasan de ronda, formas de sortear los grupos, así como si la competición es con partidos de ida y vuelta o de un solo partido por enfrentamiento.

Una vez definido el tipo de sorteo, número de equipos por grupo y si queremos que sea un evento de ida y vuelta o partido único, simularemos los partidos de la fase de grupos a excepción del último partido y así poder analizar si éste es un partido intrascendente o de colisión. Para entender mejor esta parte del trabajo, incluiremos la definición de estos conceptos relacionados con el mundo del fútbol.

- **Partidos intrascendentes:** Son aquellos partidos en los que los dos equipos que lo disputan no se juegan nada, esto ocurre cuando los dos equipos están clasificados, los dos están eliminados o uno está clasificado y el otro eliminado, independientemente del resultado del partido.
- **Partidos de colisión:** Son aquellos partidos en los que existe un resultado que permite clasificarse a los dos equipos que juegan ese partido siempre que no estén clasificados ya.
- **Sistema justo:** Es aquel sistema que permite que cuanto mejor sea un equipo, más probabilidades de clasificarse tenga.

Para evaluar una organización de torneo determinado, calcularemos el porcentaje de partidos intrascendentes y de colisión, así como la probabilidad que tiene cada equipo de clasificarse en comparación con el ranking del equipo, entre los participantes del torneo.

Capítulo 2

Modelo de regresión de Poisson

2.1. Introducción

El objetivo de esta parte del trabajo es encontrar un modelo que prediga el número de goles que marcará un equipo A al enfrentarse a un equipo B en función de las características de cada uno, las cuáles serán utilizadas como variables predictoras. Existen trabajos de algunos compañeros relacionados con este ámbito como *Predicción de resultados de partidos de fútbol* [12] en el que utiliza la regresión logística para predecir las victorias o *Predicción de eventos deportivos* [5] donde se usa el modelo de Maher para predecir el número de goles que marcará un equipo. En nuestro caso, utilizaremos el modelo de regresión de Poisson.

El modelo de regresión de Poisson se utiliza cuando la variable respuesta es una cantidad discreta que se ajusta bien a una distribución de Poisson y se pretende estudiar la influencia de las variables explicativas en su media. Este modelo es apropiado cuando la variable dependiente es un conteo como podría ser por ejemplo el número de llamadas a un teléfono, que dependen de otras variables como podría ser el día o la hora a la que se haga. La distribución de Poisson ha sido usada para modelizar el número de goles en los partidos de fútbol, a partir del trabajo [9]. En nuestro caso, intentaremos predecir el número de goles que marcará un equipo en un partido a partir de las características de los equipos que disputen ese partido. En este caso, nuestro objetivo es analizar torneos de clubes europeos y por eso hemos creado una tabla con diferentes partidos de la UEFA Champions League de las 3 temporadas anteriores con un total de 204 partidos.

Sean Y_1, \dots, Y_n variables aleatorias independientes, con Y_i la variable respuesta del modelo, sean X_1, \dots, X_k las variables explicativas que utilizaremos para definir nuestro modelo y sean los β_1, \dots, β_k los pesos que le damos a cada variable predictora. Por lo tanto, queremos un modelo que tenga la siguiente forma:

$$\lambda(\mathbf{x}) = E(Y|X = \mathbf{x}) = \mathbf{x}'\beta \quad (2.1)$$

Como se ha mencionado antes, la variable respuesta Y en una regresión de Poisson son recuentos, por lo que su valor medio tomará valores en $[0, \infty)$, por tanto al no poder tomar valores negativos no se podrá usar el modelo lineal típico $\lambda_i = \beta_0 + \beta_1 X_i$ ya que para ciertos valores de X_i , nuestra variable respuesta podría tomar valores negativos. Una forma de solucionar este problema sería modelar $\log(\lambda_i)$ en lugar de λ_i de manera que consideraríamos el modelo de Poisson como $\log(\lambda_i) = \mathbf{x}'\beta$ siendo este el producto escalar del vector de variables explicativas por el vector de parámetros. Podríamos concluir que la función de regresión del modelo de Poisson se puede definir como:

$$\lambda = e^{\mathbf{x}'\beta} = e^{\beta_0 + \beta_1 X_1 + \dots + \beta_k X_k} \quad (2.2)$$

Por tanto, estamos frente a un modelo que se podría denominar como modelo log-lineal y la interpretación de los coeficientes de las variables explicativas se podrían representar como un incremento de la

respuesta esperada al aumentar en una unidad si la variable es numérica o al cambiar de categoría si la variable es cualitativa, es decir, el modelo sufre efectos multiplicativos, donde la media para la variable de Poisson se multiplica por la potencia de e^{β_j}

$$\frac{\lambda((x_1, \dots, x_j + 1, \dots, x_k), \beta)}{\lambda((x_1, \dots, x_j, \dots, x_k), \beta)} = \frac{e^{\beta_0 + x_1\beta_1 + \dots + (x_j+1)\beta_j + \dots + x_k\beta_k}}{e^{\beta_0 + x_1\beta_1 + \dots + x_j\beta_j + \dots + x_k\beta_k}} = e^{\beta_j} \quad (2.3)$$

Respecto a la obtención de los estimadores, utilizaremos el método de máxima verosimilitud. Como las ecuaciones de verosimilitud no son lineales en los parámetros, las estimaciones se calcularán a través de procedimientos iterativos.

Para realizar nuestro estudio, utilizaremos el programa R. A diferencia del modelo lineal que usábamos en las asignaturas del grado para predecir la variable respuesta, usaremos un modelo lineal generalizado (*glm*) lo que nos permitirá que nuestra variable dependiente tenga una distribución no normal.

2.2. Presentación del modelo

Una vez introducido el modelo de regresión de Poisson, queremos aplicarlo a nuestro problema, donde buscamos predecir el número de goles que marcará un equipo en un partido a través de unas variables predictoras, es decir, queremos diseñar un modelo que nos permita predecir los goles que marcará un equipo cuando juegue contra otro independientemente de sus 'nombres' basándonos exclusivamente en las características específicas de cada uno de ellos como podría ser el valor de equipo, la calidad de la liga en la que juegan, el promedio de goles marcados, la posición en el ranking... A continuación, explicamos de una manera más detallada cada una de estas variables:

- PROMEDIO_GOLES_MARCADOS_LOCAL, PROMEDIO_GOLES_MARCADOS_VISITANTE representan el promedio de goles que marcó el equipo local/visitante la temporada pasada en su liga.
- PROMEDIO_GOLES_RECIBIDOS_LOCAL, PROMEDIO_GOLES_RECIBIDOS_VISITANTE representan el promedio de goles recibidos por el equipo local/visitante la temporada pasada en su liga.
- RANKING_LOCAL, RANKING_VISITANTE representan la posición que ocupa el equipo local/visitante antes de empezar la temporada en el Ranking FIFA, cuanto mejor es un equipo, menor posición ocupa en el ranking.
- VALOR_EQUIPO_LOCAL, VALOR_EQUIPO_VISITANTE representa el valor del equipo local/visitante en miles de millones, al inicio de la temporada.
- NIVEL_LIGA_EQUIPO_LOCAL, NIVEL_LIGA_EQUIPO_VISITANTE variable factor que nos indica la calidad de la liga del equipo local/visitante siendo 3 si el equipo juega en una liga 'buena', 2 si juega en una liga 'regular' o 1 si juega en una liga 'mala'.

Una vez definidas las variables predictoras, procedemos a estimar los coeficientes. Para ello dividiremos por un lado un modelo que prediga los goles del equipo local y por otro un modelo que estime los goles del equipo visitante en el encuentro. Para cada uno de los modelos realizaremos un estudio de los valores que obtengamos y posteriormente validaremos los modelos tanto de una manera interna como externa. De manera análoga, crearemos un modelo de partido único donde no habrá diferencia entre equipo local y visitante.

2.3. Obtención de coeficientes del modelo de goles del equipo local

Para estudiar el modelo, hemos creado una tabla en Excel con 204 partidos de la UEFA Champions League de la temporada 2017-2018 a la 2019-2020. Para los equipos de cada uno de estos partidos, hemos buscado el promedio de goles que marcó y el promedio de goles que recibió la temporada anterior, así como el valor monetario del equipo, la clasificación en el ranking FIFA antes de empezar la temporada o el nivel de su liga, es decir, si un partido es de la temporada 2019-2020, los promedios los obtenemos de la temporada 2018-2019.

Para este modelo sobre goles del equipo local, seguiremos procedimientos similares a [12], [1] y consideramos que algunas de las variables predictoras antes mencionadas como el promedio de goles recibidos por el equipo local, así como el promedio de goles marcados por el equipo visitante no influirán, aun así tenemos un número elevado de variables predictoras, por ello, realizamos una selección de modelo paso a paso con el criterio AIC. El criterio de AIC es una medida estadística que proporciona una comparación entre modelos y da un modelo con las variables más significativas y con el cuál se pierde la menor cantidad de información, por tanto, a menor AIC, mejor modelo. Para el modelo de goles del equipo local, el resultado que obtenemos es el siguiente (Aparece stepwise completo en el Apéndice C):

```

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    0.789501   0.170945   4.618 0.0000387 ***
RANKING_UEFA_LOCAL -0.006200   0.002983  -2.078   0.0377 *
VALOR_EQUIPO_LOCAL  0.430472   0.185309   2.323   0.0202 *
VALOR_EQUIPO_VISITANTE -0.717655   0.164748  -4.356 0.00001324 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 290.99  on 203  degrees of freedom
Residual deviance: 234.75  on 200  degrees of freedom
AIC: 653.9

Number of Fisher Scoring iterations: 5

```

Figura 2.1: Resumen del modelo de goles del equipo local.

Por lo tanto, el criterio AIC nos reduce el modelo de goles local a las variables predictoras RANKING_LOCAL, VALOR_EQUIPO_LOCAL y VALOR_EQUIPO_VISITANTE y la estimación de los parámetros sería:

- $\hat{\beta}_0 = 0,7895$
- $\hat{\beta}_3 = -0,0062$
- $\hat{\beta}_4 = 0,4304$
- $\hat{\beta}_{11} = -0,7176$

Los valores de estos coeficientes nos indican, en el caso del término independiente, el valor esperado de la respuesta cuando las variables explicativas valen 0, mientras que el resto de los coeficientes expresan que si aumentamos en una unidad el RANKING_LOCAL, VALOR_EQUIPO_LOCAL y VALOR_EQUIPO_VISITANTE, la media de la variable Poisson se multiplica por $e^{-0,0062}$, $e^{0,4304}$ y $e^{-0,7176}$ respectivamente.

Por lo tanto, queda que la media de nuestra Poisson para predecir el número de goles que marcará el equipo A contra el equipo B, con el equipo A como local, vendrá dado por:

$$\hat{\lambda}_i = e^{0,7895 - 0,0062X_{3i} + 0,4304X_{4i} - 0,7176X_{11i}} \quad (2.4)$$

Respecto al *summary* obtenido, cabe destacar que la variable más significativa y que más influye a la hora de predecir el parámetro λ sería VALOR_EQUIPO_VISITANTE ya que tiene el p-valor más

pequeño. Otro factor que debemos de tener en cuenta para poder validar nuestro modelo es la sobre-dispersión. Bajo un modelo de Poisson, esperamos que las medias y las varianzas de la respuesta sean aproximadamente iguales. Cuando la estimación del parámetro de sobredispersión sea mayor que 1, diremos que nuestro modelo tendrá sobredispersión.

Para calcular dicho parámetro, usaremos los residuos tipo Deviance que vienen dados por la siguiente expresión:

$$d_i = \text{sgn}(Y_i - \hat{\lambda}_i) \sqrt{2Y_i \log\left(\frac{Y_i}{\hat{\lambda}_i}\right) - 2(n - Y_i) \log\left(\frac{n - Y_i}{n - \hat{\lambda}_i}\right)} \tag{2.5}$$

Estos residuos tienen que presentar una distribución asintóticamente normal y podemos apreciar en la Figura 2.2 que nuestro modelo se parece bastante a la normal.

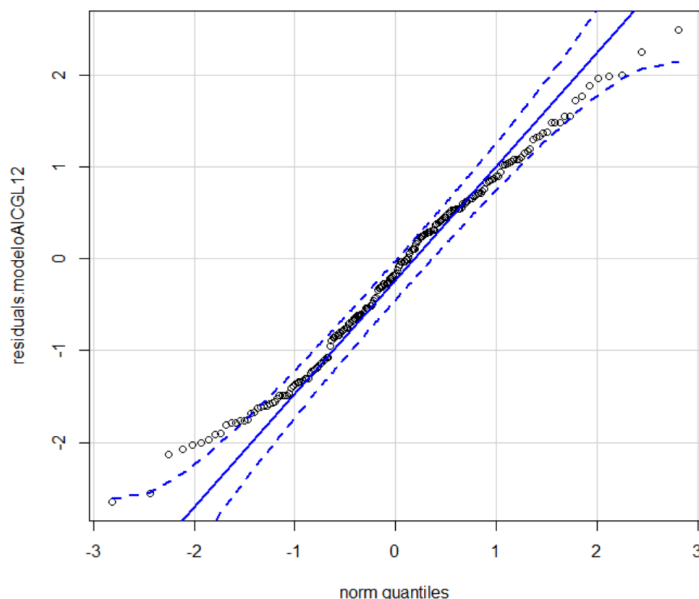


Figura 2.2: Qqplot de los residuos del modelo de goles del equipo local.

De esta forma, si queremos calcular la sobredispersión, bastará con calcular el cociente entre la suma de los residuos al cuadrado entre el número de grados de libertad de los residuos.

$$\phi_{\text{sobredispersion}} = \frac{\sum d_i^2}{df.residual} = \frac{234,75}{200} = 1,17375 \tag{2.6}$$

Podemos apreciar que nos sale una sobredispersión bastante cercana a 1 por lo que lo podemos considerar como aceptable, para asegurarnos utilizamos el *dispersiontest* de la *library AEC* y apreciamos en la Figura 2.3 como efectivamente, no rechazamos la hipótesis de equidispersión.

```
Overdispersion test

data: modeloAICGL12
z = -0.10349, p-value = 0.5412
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
0.9909628
```

Figura 2.3: Test de sobredispersión para el modelo de goles del equipo local.

Podemos analizar también la gráfica de los residuos frente a los valores ajustados con la función *scatterplot* de R, fijándonos en la Figura 2.4.

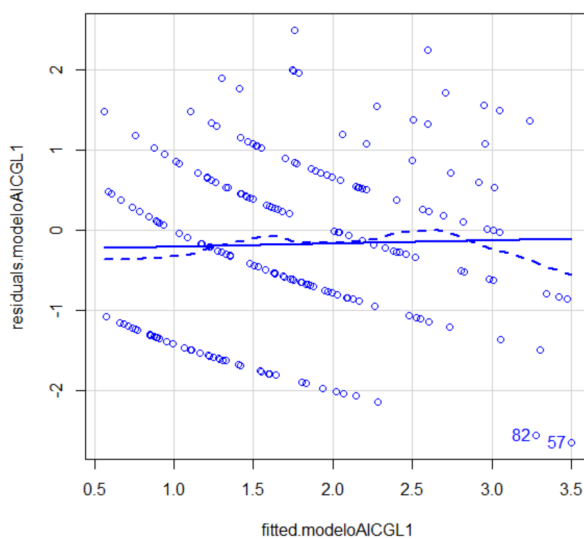


Figura 2.4: Diagrama de dispersión de los valores ajustados frente a los residuos Deviance del modelo de goles del equipo local.

Tal y como se puede apreciar vemos que los datos parecen seguir una pauta parecida, se puede ver que aparecen curvas, las cuales hacen referencia al número de goles, es decir, empezando desde la izquierda sería la curva de 0 goles, 1 gol... Como el número de goles que se pueden marcar pertenece a un intervalo bastante pequeño, nos permite ver con mayor facilidad como se ajustan nuestras estimaciones.

Como era de esperar, vemos que la mayor concentración de puntos se encuentra en torno a la media de goles que es 1,75. Tal y como se puede ver en la gráfica, los valores más destacados son el 57 y 82 que se corresponden con los partidos de FC Barcelona-Slavia Praga que terminó 0-0 y el Real Madrid-CSKA Moscú que acabaron 0-3 y se esperaban valores de 3,49 y 3,27 para el equipo local.

2.4. Obtención de coeficientes del modelo de goles del equipo visitante

Para este modelo, procederemos de manera análoga al modelo anterior. En este caso queremos predecir los goles que marcara el equipo visitante. Al igual que en el modelo de goles del equipo local, hay variables predictoras que consideramos que no influirán en el resultado como son el promedio de goles marcados por el equipo local o el promedio de goles recibidos por el equipo visitante. Al seguir teniendo gran cantidad de variables predictoras, utilizaremos el criterio de Akaike para hacer la selección del modelo más adecuado y que menos información pierda. De esta manera, obtenemos los siguientes resultados (Aparece completo el stepwise en el Apéndice C):

```

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)   -0.359226   0.394113  -0.911 0.362042
RANKING_UEFA_VISITANTE -0.006222   0.003450  -1.803 0.071337 .
VALOR_EQUIPO_VISITANTE  0.354195   0.209112   1.694 0.090302 .
PROMEDIO_GOLES_MARCADOS_VISITANTE 0.418574   0.145856   2.870 0.004108 **
VALOR_EQUIPO_LOCAL    -0.683930   0.186359  -3.670 0.000243 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 300.29 on 203 degrees of freedom
Residual deviance: 248.66 on 199 degrees of freedom
AIC: 606.29

Number of Fisher Scoring iterations: 5

```

Figura 2.5: Resumen del modelo de goles del equipo visitante

Vemos que para predecir el número de goles del equipo visitante se necesitan muchas más variables predictoras que en el modelo anterior quedándonos el siguiente modelo:

$$\hat{\lambda}_i = e^{-0,359226 - 0,006222X_{10i} + 0,354195X_{11i} + 0,418574X_{8i} - 0,68393X_{4i}} \quad (2.7)$$

Por lo tanto, el criterio AIC nos reduce el modelo de goles local a las variables predictoras RANKING_VISITANTE, VALOR_EQUIPO_LOCAL, VALOR_EQUIPO_VISITANTE y PROMEDIO_GOLES_MARCADOS_VISITANTE y la estimación de los parámetros sería:

$$\begin{aligned} \hat{\beta}_0 &= -0,359226 & \hat{\beta}_4 &= -0,68393 & \hat{\beta}_8 &= 0,418574 \\ \hat{\beta}_{10} &= -0,006222 & \hat{\beta}_{11} &= 0,354195 \end{aligned}$$

Fijándonos en los p-valores obtenidos, vemos que la variable más influyente sería la variable VALOR_EQUIPO_LOCAL. Nos fijamos ahora en el diagrama de dispersión de la Figura 2.6.

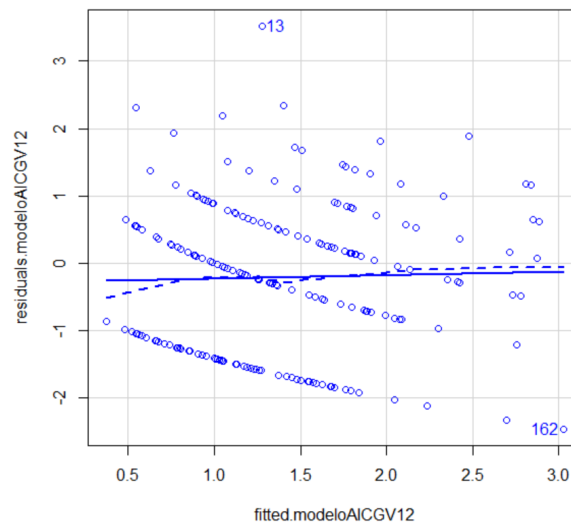


Figura 2.6: Diagrama de dispersión de los valores ajustados frente a los residuos Deviance del modelo de goles del equipo visitante.

Vemos que ahora la mayoría de los puntos están concentrados en 0,1,2 goles y vemos como se comprimen todos a la zona de la media que en este caso es de 1,37.

Por otro lado, estudiamos la sobredispersión de este modelo. De la misma forma que en el modelo anterior, fijándonos en el *summary*, se observa que la suma de los residuos *Deviance* al cuadrado es 248,66 y los grados de libertad son 197 por lo que si hacemos la división obtenemos:

$$\phi_{sobredispersión} = \frac{\sum d_i^2}{df.residual} = \frac{248,66}{197} = 1,2622 \quad (2.8)$$

Vemos que es un valor bastante cercano a 1, si realizamos el *dispersiontest* obtenemos lo siguiente:

```
Overdispersion test

data: modeloAICGV12
z = 0.67005, p-value = 0.2514
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
1.088723
```

Figura 2.7: Test de sobredispersión para el modelo de goles del equipo visitante.

Efectivamente nos sale un p-valor de 0,25 y por lo tanto podemos aceptar que no hay sobredispersión.

Si queremos calcular las probabilidades de un resultado, nos tendremos que fijar en nuestro modelo de predicción de goles local y goles visitante que son nuestras probabilidades marginales. Para calcular la probabilidad de un resultado, necesitamos la probabilidad conjunta. Para calcularla, necesitaremos independencia entre variables y así poder calcular el producto. Para comprobar si esta hipótesis de independencia entre los goles que marca un equipo y el contrincante es razonable, usaremos los residuos deviance de las regresiones anteriores. Estos residuos tienen distribución asintóticamente normal de media 0 y varianza 1. Si las respuestas (goles de cada equipo) son independientes, entonces también lo serán los residuos. Para comprobar su independencia, usaremos la función *cor.test* de R.

```
Pearson's product-moment correlation

data: DATOSBUENO11$residuals.modeloAICGL12 and DATOSBUENO11$residuals.modeloAICGV12
t = -1.2496, df = 202, p-value = 0.2129
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 -0.22227843  0.05039598
sample estimates:
      cor
-0.08758152
```

Figura 2.8: Test de correlación de Pearson entre variables

Se puede apreciar que nos sale una correlación muy pequeña, del -0.087 además, cabe destacar que nos sale un p-valor mayor que 0.05 por lo que podemos aceptar la hipótesis de independencia entre variables.

2.5. Validación de los modelos

Una vez que hemos visto que nuestras variables respuesta pueden suponerse independientes, podemos calcular la probabilidad de que se dé un resultado en un partido mediante el producto de dos variables aleatorias Poisson. A partir de estas probabilidades podremos comprobar como se ajusta nuestro modelo a lo que de verdad sucedió. En primer lugar, a partir del conjunto de partidos seleccionados, contaremos la cantidad de veces que se dió cada resultado.

Local\Visitante	0 goles	1 goles	2 goles	3 goles o más
0 goles	12	7	11	14
1 goles	15	17	17	8
2 goles	14	15	12	5
3 goles	14	14	3	3
4 goles o más	7	2	4	2

Tabla 2.1: Tabla que representa el número de resultados de cada marcador en nuestro conjunto de datos.

Para comparar estos resultados con nuestro modelo, una vez calculadas las probabilidades de cada resultado para cada partido, calcularemos el valor esperado del número de veces que aparece cada marcador. Como el número de veces que nos sale un resultado es la suma de los indicadores de que se dé ese resultado, obtenemos lo siguiente:

Local\Visitante	0 goles	1 goles	2 goles	3 goles o más
0 goles	9,824	13,232	10,291	10,744
1 goles	15,798	18,852	13,144	11,875
2 goles	14,550	15,496	9,689	7,548
3 goles	10,004	9,618	5,420	3,656
4 goles o más	10,074	8,471	4,132	1,571

Tabla 2.2: Tabla que representa el número esperado de resultados de cada marcador según nuestro modelo.

Podemos apreciar que la Tabla 2.2 , normalizada para que sumen la misma cantidad de partidos, sigue una distribución bastante similar a la Tabla 2.1. Para comprobar la similitud de las tablas, hemos realizado un test de bondad de ajuste de chi-cuadrado, que bajo la hipótesis nula de que los datos se ajustan bien y con una chi cuadrado con 19 grados de libertad, se ha obtenido un p-valor de 0.649, por lo que podemos afirmar que se ajusta correctamente.

Además, al poder obtener las probabilidades de cada resultado para cada partido, podemos calcular la probabilidad que tiene de ganar, perder o empatar cada equipo. De la misma forma que antes, calculamos estas 3 probabilidades para cada partido y el número esperado de victorias locales, visitantes o empate será la suma de las probabilidades de victoria local, visitante o empate de todos los partidos.

	Local	Visitante	Empate
Observado	95	64	45
Predicción	94,99	67,08	41,08

Tabla 2.3: Tabla que representa el número de predicciones de victoria local, visitante o empate frente a los resultados que se observaron.

Vemos que los resultados que obtenemos se acercan mucho a lo que realmente sucedió en los partidos que observamos. A primera vista, parece que nuestro modelo proporciona buenos resultados.

Para comprobar si estos lo son, haremos un análisis a partir de las curvas de ROC. Las curvas ROC son un método matemático que permite analizar la bondad de un modelo cuya variable respuesta es de tipo dicotómico. Respecto a la representación gráfica, las curvas de ROC muestran la sensibilidad frente a la 1-especificidad. Veamos qué significan estos conceptos aplicados a los partidos de fútbol.

Definición. La **sensibilidad** es la probabilidad de que, si en un partido gana el equipo A, el modelo prediga que gana el equipo A. A esto se le conoce como Razón de Verdaderos Positivos

Definición. La **especificidad** es la probabilidad de que, si en un partido no gana el equipo A, el modelo prediga que no gana el equipo A. A esto se le conoce como Razón de Verdaderos Negativos

Para comprobar si un modelo es aceptable a partir de la curva de ROC se calcula el área situada bajo la curva a lo que denominamos AUC. Esta área tiene un valor entre 0,5 y 1, dónde 1 representaría el valor de diagnóstico perfecto y 0,5 una prueba sin valor diagnóstico. Cuando tenemos un $AUC > 0.7$, diremos que nuestro modelo tiene un valor de diagnóstico aceptable, ya que es un valor comúnmente usado en la literatura.

A partir de los modelos de *goles del local* y *goles del visitante* hemos sacado las probabilidades de victoria local, victoria visitante y empate. Veamos qué curvas ROC obtenemos a partir de estos modelos.

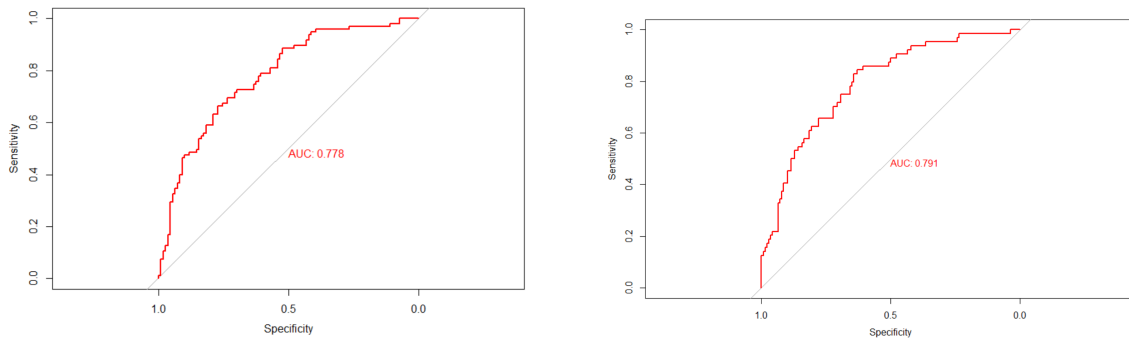


Figura 2.9: Curvas ROC, a la izquierda está representado el éxito de victoria local y a la derecha el éxito de victoria visitante.

Podemos ver que los valores del AUC están en torno a 0,78 por lo que podemos considerarlo aceptable. Nos fijamos ahora en el AUC para los empates.

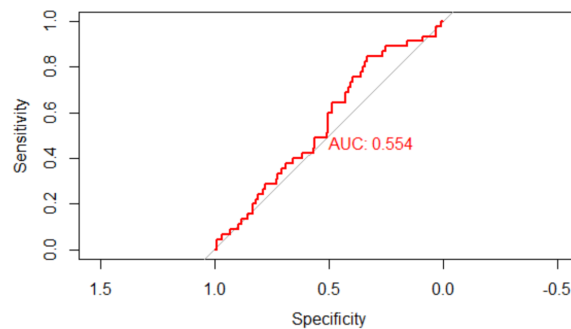


Figura 2.10: Curva ROC que representa el éxito de empate.

Como se puede apreciar, nos sale un valor bastante malo, esto se debe a que predecir un empate es muy difícil, ya que es el resultado menos frecuente de los 3 posibles, dado que una victoria local o visitante puede ser fácil predecirla debido a la superioridad de uno de los dos equipos en sus estadísticas, mientras que para el empate es más complicado que las estadísticas de ambos equipos estén igualadas tal y como podemos ver en [13]. Para los casos de victoria local y victoria visitante es normal que se obtengan valores más aceptables, ya que las curvas ROC dan victoria local cuando la probabilidad de victoria local sea mayor que un cierto valor y de manera análoga para la victoria visitante. En el caso del empate, es mucho más complicado predecirlo y por tanto es entendible que el valor de diagnóstico sea peor.

2.6. Obtención de coeficientes para partido único

A pesar de que los partidos de nuestro conjunto de datos son de ida y vuelta, vamos a hacer con estos mismos partidos, un análisis como si fueran partidos únicos. De la misma forma que se han estudiado los goles del equipo local y el equipo visitante, vamos a crear un modelo para predecir los goles que marcaría un equipo sin tener en cuenta si juega en casa o fuera. A partir de nuestra tabla de partidos, hemos creado una nueva columna en la que hemos juntado los goles del local y los goles del visitante, por lo que hemos pasado de tener 204 registros a 408. Estos nuevos coeficientes los calculamos para poder hacer un análisis más amplio sobre cómo puede afectar que las fases de grupos no sean partidos de ida y vuelta, sino que sean a partido único como podría ser la fase de grupos del Mundial de fútbol. Realizamos un análisis análogo a los modelos anteriores.

```

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept)    0.209377   0.260250   0.805   0.42109
RANKING_UEFA_EQ1 -0.006084   0.002272  -2.678   0.00742 **
VALOR_EQUIPO_EQ1  0.398972   0.138609   2.878   0.00400 **
PROMEDIO_GOLES_MARCADOS_EQ1  0.214042   0.097094   2.204   0.02749 *
VALOR_EQUIPO_EQ2 -0.702090   0.123459  -5.687  0.0000000129 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for poisson family taken to be 1)

Null deviance: 600.36 on 407 degrees of freedom
Residual deviance: 495.98 on 403 degrees of freedom
AIC: 1264.8

Number of Fisher Scoring iterations: 5

```

Figura 2.11: Resumen del modelo de goles para partido único.

Vemos que las variables predictoras más influyentes son RANKING_UEFA_EQ1, VALOR_EQUIPO_EQ1, PROMEDIO_GOLES_MARCADOS_EQ1 y VALOR_EQUIPO_EQ2, siendo esta la variable más influyente a la hora de predecir el número de goles.

De la misma forma que para los modelos anteriores, para comprobar si nuestro modelo es válido, comprobamos si tiene o no sobredispersión.

$$\phi_{sobredispersion} = \frac{\sum d_i^2}{df.residual} = \frac{495,98}{403} = 1,2307 \quad (2.9)$$

Realizamos el test de dispersión en la Figura 2.12 para confirmar que no existe dicho problema.

```

Overdispersion test

data: modeloAICGV12
z = 0.67005, p-value = 0.2514
alternative hypothesis: true dispersion is greater than 1
sample estimates:
dispersion
1.088723

```

Figura 2.12: Test de sobredispersión para el modelo de goles para partido único.

Vemos que efectivamente, obtenemos un p-valor de 0,2514 por lo que podemos afirmar que no existirá sobredispersión.

Capítulo 3

Análisis de formación de grupos

Una vez obtenidas las estimaciones de los parámetros, procedemos a estudiar las diferentes propuestas de formación de grupos para así analizar en qué casos se consigue una reducción de partidos irrelevantes o con riesgo de colisión. Utilizaremos la simulación de Monte Carlo a partir de las estimaciones obtenidas para sacar una estimación de cuántos partidos intrascendentes y de colisión se obtendrán en función del tipo de torneo, la formación de grupos y el orden de los partidos.

Para ello, estudiaremos 2 formaciones distintas, una en la que los grupos serán de 4 equipos y otra en la que los grupos serán de 3. Para el caso de 4 equipos, haremos referencia a la UEFA Champions League 2020-2021 donde participan 32 equipos, por lo que se crearán 8 grupos de 4 equipos y donde se clasificarán los 2 mejores de cada grupo. Para el caso de grupos de 3, nos centraremos en la UEFA Europa League 2020-2021, donde participan 48 equipos. Para ello haremos 16 grupos de 3 equipos y analizaremos, por un lado, el caso en el que solamente se clasifique el primero y por otro en el que se clasifiquen los dos mejores de cada grupo. Cabe mencionar que en nuestro modelo de sorteo hemos obviado las reglas geográficas impuestas por la UEFA, por la que equipos del mismo país no pueden estar en un mismo grupo. Además, hemos formado los bombos por el puesto del ranking, lo cual sigue el método de la UEFA, aunque en el caso de la UEFA Champions League, el primer bombo está formado por los campeones de la temporada anterior en ciertas ligas nacionales.

3.1. Explicación de la simulación

Una vez explicados los casos que vamos a analizar, comentamos cuál será la estructura de nuestro programa de simulación. En primer lugar, realizaremos un sorteo para dividir los equipos en grupos, para ello, hemos considerado interesante realizar dos sorteos distintos, por un lado, el sorteo clásico de la FIFA y por otro, el sorteo que planteó Julien Guyon [4] debido a que consideraba que el sorteo de la FIFA tenía ciertas deficiencias. Para esta explicación, nos centraremos en el caso de 32 equipos. Primero, hemos creado un ranking relativo basado en el ranking original FIFA para así dar un valor entre 1 y 32 a cada uno de los equipos que forman esta competición. Explicamos ahora en qué consiste cada sorteo aplicado a nuestro programa:

Definición. Llamaremos **Sorteo FIFA** a aquel en el que dividimos los equipos en 4 urnas diferentes dependiendo del ranking, es decir, si nuestra competición la forman 32 equipos, los componentes del primer bombo serán los 8 equipos que tengan mejor ranking. Una vez formadas las urnas, extraemos equipos de manera aleatoria del primer bombo y los ponemos en diferentes grupos, una vez hecho esto, se hace lo mismo con el bombo 2, bombo 3 y bombo 4. De esta manera conseguimos formar los grupos de manera totalmente aleatoria.

Definición. Llamaremos **Sorteo Guyon** a aquel en el que el procedimiento de extraer los equipos de las urnas es el mismo que el Sorteo FIFA, pero en este caso cada bombo lo dividimos en 2 subgrupos

ordenados por el ranking relativo, de manera que un sorteo se haría entre los 4 mejores del bombo 1 y 3 y los 4 peores de los bombos 2 y 4 y en el otro sus opuestos, es decir, seguiría una curva en forma de S. De esta forma, se pretende obtener un sorteo más equitativo. El objetivo de Guyon con este tipo de sorteo es obtener grupos donde la suma de los ranking de los 3 equipos menos pesados sea lo más pareja posible, evitando así los denominados "Grupos de la muerte".

En segundo lugar, estableceremos los órdenes en los que se jugarán los partidos. Notemos que los partidos intrascendentes o de colisión solo pueden darse (salvo situaciones excepcionales) en la última jornada. Por ello, nos centraremos en determinar cuáles deben ser los enfrentamientos en esta última jornada. En el caso de grupos de 4, habrá 12 combinaciones distintas de jugarse la última jornada puesto que son partidos de ida y vuelta. Los posibles enfrentamientos en la última jornada serán los siguientes: 14–23, 12–34, 13–24, 32–41, 43–21, 42–31, 41–23, 21–34, 31–24, 23–41, 34–21 y 24–31. Como los resultados que obtenemos de los 6 primeros enfrentamientos son muy similares a los 6 últimos, nos centraremos exclusivamente en los primeros 6 casos, donde por ejemplo la combinación 14–23, significaría que la última jornada la jugarían el equipo del bombo 1 contra el equipo del bombo 4 y el equipo del bombo 2 contra el del bombo 3, teniendo a los equipos del bombo 1 y 2 como equipos locales. Para el caso de grupos de 3, también habría 6 formas distintas, pero en este caso solo se jugaría un partido en la última jornada, los partidos podrían ser: 12,13,23,21,31,32. De la misma forma si quisiéramos fijarnos en que los partidos fueran únicos, tendríamos 3 posibilidades.

Una vez establecidos los grupos y fijada la ordenación de las jornadas, realizaremos la simulación de los partidos de cada grupo, a excepción de la última jornada para así poder estudiar los partidos de colisión e intrascendentes. Para cada equipo guardaremos el número total de goles que ha marcado, los que ha recibido, los puntos obtenidos, así como estos mismos valores correspondientes a cada enfrentamiento directo para decidir, en caso de empate a puntos, qué equipo queda mejor clasificado. El método de puntuación y de desempate que seguiremos será el indicado por la UEFA, donde la victoria se valorará con 3 puntos, el empate con 1 y la derrota con 0. En el caso de empate a puntos, cuando se produce entre 2 equipos, miraremos los puntos obtenidos entre ellos en los enfrentamientos directos, si hay alguno que ha obtenido mejor puntuación, ese quedará por delante, en caso de igualdad, miraremos la diferencia de goles entre los anotados y los recibidos en sus enfrentamientos, si todavía sigue esa igualdad, lo realizaremos por sorteo. Si el empate se produce entre 3 equipos realizaremos el mismo proceso que antes, primero nos fijaremos en los puntos obtenidos en sus enfrentamientos y si todavía sigue esa igualdad, nos fijaremos en su diferencia de goles y si también hay empate, realizaremos sorteo. Por último, en el hipotético caso que los 4 equipos estuvieran empatados a puntos, hecho bastante improbable, directamente realizaríamos un sorteo para obtener sus posiciones.

Una vez realizada la simulación de los partidos a excepción de la última jornada, estudiaremos la cantidad de partidos que serán intrascendentes o con riesgo de colisión. Definamos los siguientes conceptos:

- **Intrascendentes:** Son aquellos partidos en los que los dos equipos que lo disputan no se juegan nada, esto ocurre cuando los dos equipos están clasificados, los dos están eliminados o uno está clasificado y el otro eliminado, independientemente del resultado del partido.
- **Colisión:** Son aquellos partidos en los que existe un resultado que permite clasificarse a los dos equipos que juegan ese partido siempre que no estén clasificados ya. Tanto en los grupos de 4 como en los grupos de 3 con 2 clasificados, esto sólo puede ocurrir cuando en la última jornada se enfrentan el primer y segundo clasificado. En el caso de grupos de 3 y 1 clasificado, llamaremos colisión al partido que enfrente al segundo contra el tercer clasificado, pudiendo el segundo pasar a la siguiente ronda y el tercero estar ya eliminado.

Por último, se simulará la última jornada y veremos qué equipos se clasifican. Esto nos servirá para analizar la justicia del sistema, es decir, en qué método los mejores equipos tienen probabilidad más alta de clasificarse.

3.2. Análisis de los partidos intrascendentes y de colisión

Hemos hecho 1000 simulaciones en cada una de las situaciones descritas anteriormente. Recordemos que a la hora de simular un partido usamos los modelos ajustados en el capítulo anterior. Vamos a comenzar explicando el caso de 32 equipos con 8 grupos de 4. En este caso, hemos considerado 12 situaciones distintas en función del sorteo que usemos, sorteo FIFA o Guyon y cuáles son los partidos de la última jornada. Hay que tener en cuenta que en la última jornada se jugarán 16 partidos entre todos los grupos, de los cuales podrán ser intrascendentes todos, mientras que de colisión solamente puede ser uno por grupo, puesto que es imposible que en los dos partidos todos los equipos puedan salir beneficiados, por lo que como mucho podrán existir 8. Realizamos las 1000 simulaciones y obtenemos los resultados en porcentajes de la Tabla 3.1.

	14-23	12-34	13-24	32-41	42-31	43-21	14-23 G	12-34 G	13-24 G	32-41 G	42-31 G	43-21 G
INTRASCENDENTE	46,75	53,53	46,81	50,59	49,18	53,21	46,21	52,95	48,16	50,15	50,73	52,30
COLISION	12,26	19,65	16,42	10,41	15,08	20,27	11,38	20,78	16,18	10,15	14,77	20,48

Tabla 3.1: Porcentaje de partidos de colisión e intrascendentes para los grupos de 4 en una competición de ida y vuelta.

Como podemos apreciar en el sorteo de la FIFA, cuando más casos de colisión existen es cuando en el último partido juegan los equipos del bombo 1 y 2, algo que tiene sentido ya que lo habitual es que estos dos equipos, al ser en teoría más fuertes, hayan conseguido mayor puntuación y pueda existir algún resultado que permita pasar a ambos. Por el mismo motivo, podríamos explicar también que existan un mayor número de partidos intrascendentes ya que los dos mejores puede que ya estén clasificados y por tanto, el partido que enfrenta a los equipos del tercer y cuarto bombo, como el que enfrenta a los miembros del primer y segundo bombo serían irrelevantes.

A partir de los datos obtenidos, si queremos que la competición sea lo más atractiva posible, se podría concluir que si en la última jornada juegan el equipo del bombo 1 contra el del bombo 4, obtenemos tanto un menor número de partidos intrascendentes como de colisión. Si nos fijamos en los resultados obtenidos con el sorteo de Guyon, se puede apreciar que no hay casi diferencia con el sorteo de la FIFA, se obtienen valores muy similares por lo que podríamos decir que no tiene mucha influencia realizar el sorteo de esta manera. Además, resaltar también que si el último partido se juega en casa de un equipo o de otro, los porcentajes obtenidos son similares.

Para ver mejor esta similitud entre el sorteo tradicional y el presentado por Guyon, fijémonos en el diagrama de barras apiladas de la Figura 3.1 y en los estadísticos de la Tabla 3.2 para los partidos de colisión y en el diagrama de cajas de la Figura 3.2 y los estadísticos de la Tabla 3.3 para los partidos intrascendentes de los grupos de 4 con partidos de ida y vuelta. Como los resultados obtenidos son parecidos para un mismo partido cuando juegan en casa de un equipo o de otro, extraeremos los estadísticos solamente de un enfrentamiento entre los mismos equipos como veremos en la Tabla 3.2.

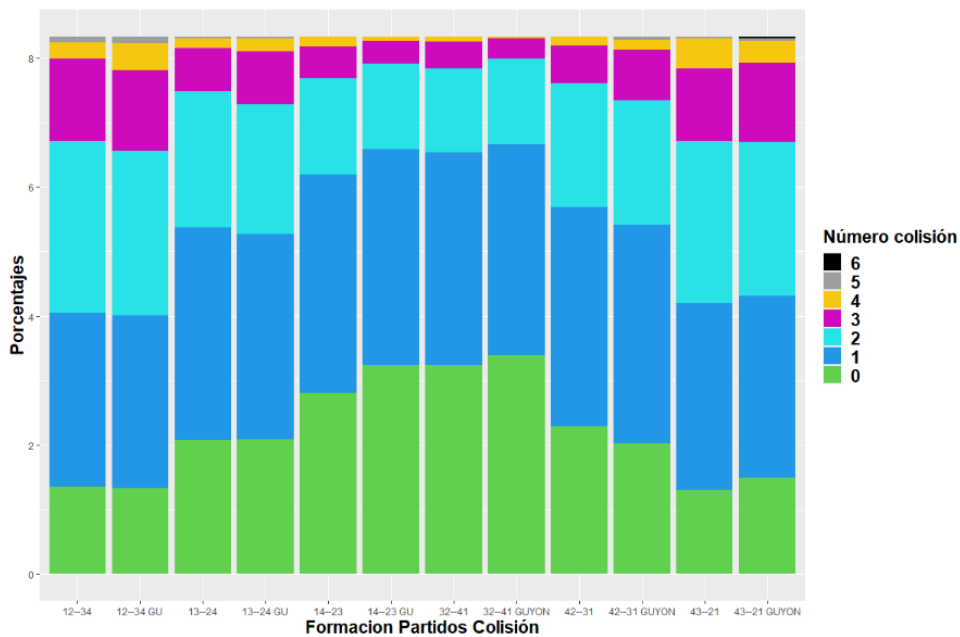


Figura 3.1: Diagrama de barras apiladas para el número de partidos de colisión por torneo dependiendo de tipo de sorteo y los enfrentamientos de la última jornada en los grupos de 4 a doble partido.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
COLISION 14-23	0	0	1	2	4	0.96	0.98
COLISION 14-23 GU	0	0	1	1	5	0.88	0.91
COLISION 12-34	0	1	2	2	5	1.09	1.57
COLISION 12-34 GU	0	1	2	2	6	1.14	1.66
COLISION 13-24	0	1	1	2	5	0.99	1.31
COLISION 13-24 GU	0	0	1	2	5	1.04	1.29

Tabla 3.2: Principales estadísticos para los partidos de colisión en grupos de 4 a doble partido.

En base a la Figura 3.1, podemos ver las similitudes que existen entre ambos sorteos para los partidos de colisión. Para confirmar está relación entre ambos tipos de sorteos, podemos realizar un test de comparación de medias entre el sorteo FIFA y el de Guyon. A pesar de que nuestras variables no son continuas ni normales, son independientes y al tener una gran cantidad de datos, sus medias seguirán una distribución aproximadamente normal, por lo que realizaremos la función *t.test* de R y veremos que efectivamente podemos aceptar que sus medias son iguales (Ver Apéndice B Grupos de 4 ordenados por el Ranking UEFA). Fijémonos ahora en los partidos intrascendentes.

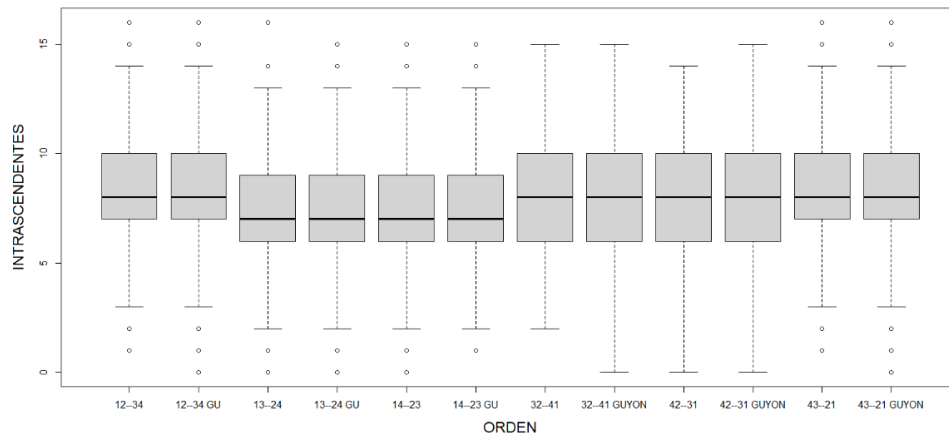


Figura 3.2: Diagrama de cajas para los partidos intrascendentes dependiendo de tipo de sorteo y los enfrentamientos de la última jornada para grupos de 4 a doble partido.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
INTRASCENDENTE 14-23	0	6	7	9	15	2.42	7.48
INTRASCENDENTE 14-23 GU	1	6	7	9	15	2.42	7.39
INTRASCENDENTE 12-34	1	7	8	10	16	2.57	8.56
INTRASCENDENTE 12-34 GU	0	7	8	10	16	2.61	8.47
INTRASCENDENTE 13-24	0	6	7	9	16	2.53	7.49
INTRASCENDENTE 13-24 GU	1	7	8	10	16	2.65	7.7

Tabla 3.3: Principales estadísticos para los partidos intrascendentes en grupos de 4 a doble partido.

En cuanto a los partidos intrascendentes, se puede apreciar que se obtienen resultados similares entre ambos sorteos como se puede apreciar tanto en el diagrama de cajas de la Figura 3.2 como observando los estadísticos de la Tabla 3.3.

Fijémonos ahora si las fases de grupos fueran a partido único. De primeras, se puede pensar que, si el último partido lo juegan entre los 2 mejores del grupo, será más fácil que ocurra colisión en ese partido ya que a priori, habrán conseguido la mayoría de puntos posibles. Veamos los resultados en la Tabla 3.4:

	13-24	12-34	14-23	13-24G	12-34G	14-23G
INTRASCENDENTE	11,79	19,32	10,60	12,81	18,71	10,86
COLISION	17,47	26,01	14,03	18,56	26,21	13,68

Tabla 3.4: Porcentaje de partidos de colisión e intrascendentes para los grupos de 4 en una competición a partido único.

Vemos que efectivamente, con un 26% de los partidos que enfrenta a los equipos 12-34 hay una diferencia considerable frente al resto de los casos. Respecto a los partidos intrascendentes vemos que también hay un mayor número de partidos cuando los partidos se juegan en este orden. Por lo que podemos decir que ésta sería la peor disposición para jugar la última jornada. Respecto a las otras dos opciones, podríamos considerar mejor aquella en la que se enfrentan el equipo 1 contra el 4 y el equipo 2 contra el 3, ya que a pesar de tener valores parecidos en lo que a partidos intrascendentes se refiere, su riesgo de colisión es de un 14% frente a un 17%. Apreciamos también que el sorteo de Guyon tampoco influye en este caso.

De la misma forma que hemos hecho para los partidos de ida y vuelta, sacamos los diagramas de cajas y barras apiladas para partido único.

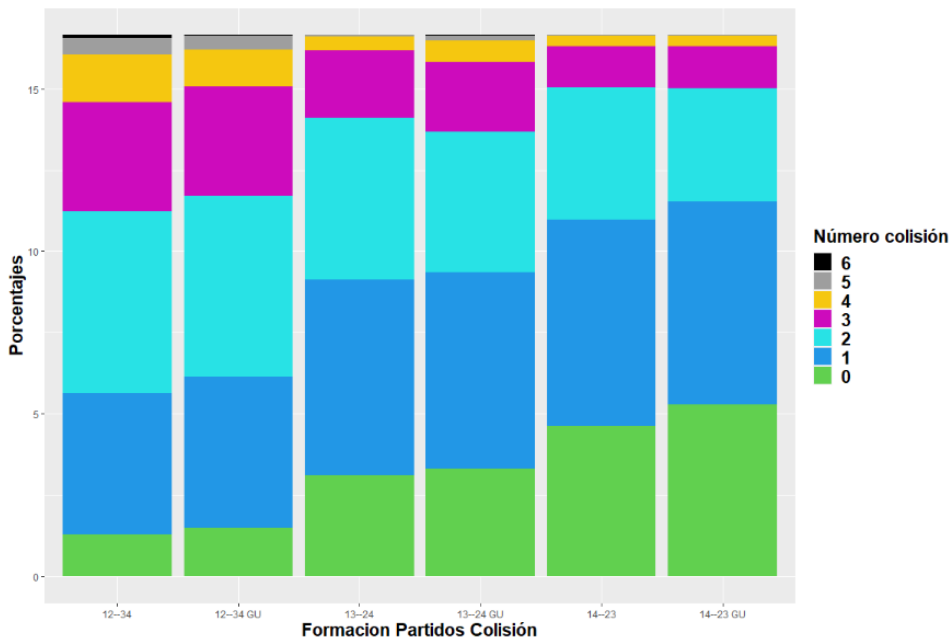


Figura 3.3: Diagrama de barras apiladas para los partidos colisión dependiendo de tipo de sorteo y los enfrentamientos de la última jornada con partidos únicos para los grupos de 4.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
COLISION 13-24	0	1	1	2	5	1.03	1.39
COLISION GU 13-24	0	1	1	2	6	1.13	1.48
COLISION 12-34	0	1	2	3	6	1.22	2.08
COLISION GU 12-34	0	1	2	3	6	1.17	2.09
COLISION 14-23	0	0	1	2	6	1.00	1.12
COLISION GU 14-23	0	0	1	2	5	1.01	1.09

Tabla 3.5: Principales estadísticos para los partidos de colisión en partido único para los grupos de 4.

Vemos que al igual que para partidos de ida y vuelta, no hay diferencia entre el sorteo de Guyon y el establecido por la FIFA, en la mayoría de los casos los estadísticos son bastante similares.

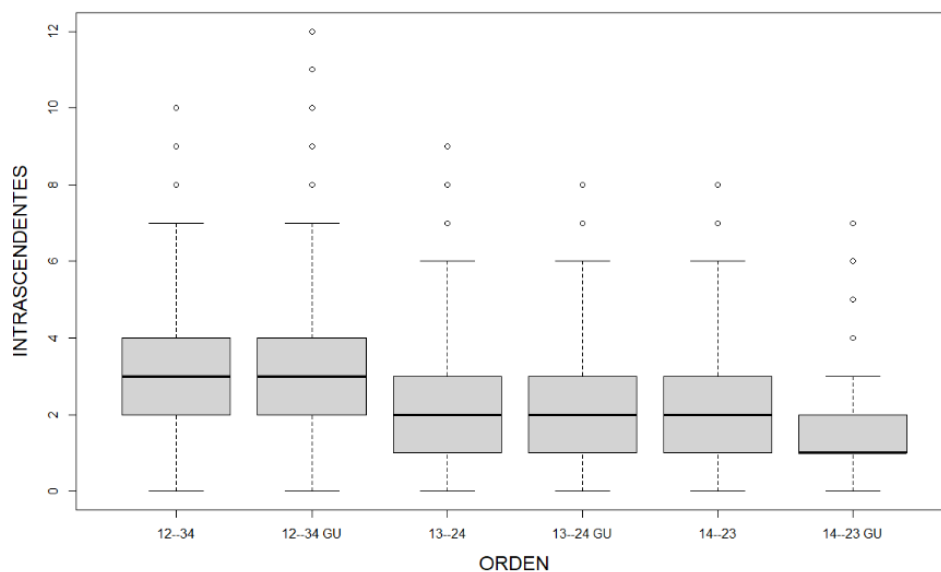


Figura 3.4: Diagrama de cajas para los partidos intrascendentes dependiendo de tipo de sorteo y los enfrentamientos de la última jornada con partidos únicos para los grupos de 4.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
INTRASCENDENTE 13-24	0	1	2	3	9	1.55	1.88
INTRASCENDENTE GU 13-24	0	1	2	3	8	1.55	2.05
INTRASCENDENTE 12-34	0	2	3	4	10	2.07	3.09
INTRASCENDENTE GU 12-34	0	2	3	4	12	2.10	2.99
INTRASCENDENTE 14-23	0	1	2	3	8	1.37	1.69
INTRASCENDENTE GU 14-23	0	1	1	2	7	1.33	1.73

Tabla 3.6: Principales estadísticos para los partidos intrascendentes con partido único para los grupos de 4.

Al igual que con los partidos de colisión, los partidos intrascendentes con el sistema de Guyon obtienen valores parecidos a los del sorteo tradicional, sin embargo, hay un caso, el 14-23 GU que parece ser mejor, pero comprobándolo con el gráfico de barras apiladas, nos sale prácticamente igual.

Veamos qué ocurre ahora con los grupos de 3 equipos. Como hemos comentado antes, utilizaremos los 48 equipos de la UEFA Europa League y analizaremos tanto que haya un clasificado como dos. Empezaremos estudiando los grupos de 3 en los que solamente se clasifica un equipo por grupo.

	32	31	21	23	13	12	32 G	31 G	21 G	23 G	13 G	12 G
INTRASCENDENTE	63,1	54,33	40,92	62,27	49,33	35,86	63,54	53,61	39,93	62,95	49	35,25
COLISION	15,66	14,73	6,44	14,72	13,65	6,325	15,65	14,56	5,9375	14,76	14,05	5,99

Tabla 3.7: Porcentaje de partidos de colisión e intrascendentes para los grupos de 3 con 1 clasificado en una competición de ida y vuelta.

Vemos que se obtiene en torno a un 63 % de partidos intrascendentes cuando la última jornada la disputan el equipo 2 contra el equipo 3, siendo el equipo 2 y el equipo 3 los extraídos del bombo 2 y bombo 3 respectivamente. Esto se puede considerar como normal ya que en la mayoría de los casos el equipo más fuerte se habrá clasificado ya y por tanto este partido no tendrá importancia. Respecto a este mismo asunto, si el último enfrentamiento fuese entre los dos mejores, considerando mejores a los equipos de los dos primeros bombos, vemos que se disminuye en más de un 20 % el riesgo de que el partido sea intrascendente. En cuanto a los partidos de colisión, vemos que tanto si se enfrenta el tercero contra

el primero o contra el segundo, el riesgo de colisión es muy similar con un valor cercano al 14%. Lo que sí cabe destacar es el caso en el que jueguen el equipo 1 contra el equipo 2, este riesgo se reduce a un 6% de los partidos. Esta cifra nos indica que es muy complicado que ninguno de los 2 no tenga opción de clasificarse. De la misma forma que para el caso de grupos de 4, el sorteo de Guyon no tendría ninguna repercusión (se pueden ver las gráficas y tablas en el Apéndice B Grupos de 3 con 1 clasificado ordenados por Ranking UEFA). Analicemos ahora el caso de partido único.

	31	21	23	31 G	21 G	23 G
INTRASCENDENTE	12,23	4,18	29,73	11,83	3,75	30,31
COLISION	5,56	3,89	5,96	5,52	3,68	5,97

Tabla 3.8: Porcentaje de partidos de colisión e intrascendentes para los grupos de 3 con 1 clasificado en una competición a partido único.

Respecto a los partidos únicos, obtenemos unos porcentajes muy bajos de partidos de colisión, algo que es bastante comprensible puesto que, al jugarse solamente 3 partidos, es muy difícil que un equipo se haya quedado descolgado y no pueda clasificarse. En cambio, si nos fijamos en los partidos intrascendentes, sí que hay bastante variabilidad. Vemos que el caso con mayor cantidad de estos partidos es cuando se enfrentan el equipo 2 contra el equipo 3, algo que puede verse normal por el hecho de que sería habitual que el equipo 1 estuviera ya clasificado. De la misma forma se entiende que el menor valor ocurra cuando se enfrentan los dos mejores, bajando así a un 4%, por lo que es una reducción muy a tener en cuenta. Por lo tanto, cuando los grupos son de 3 con 1 clasificado, tanto si la fase de grupos fuese ida y vuelta como a partido único, la mejor opción sería la de que jugaran el último partido los 2 mejores.

Por último, analizaremos el caso de grupos de 3 en el que se clasifiquen los 2 mejores.

	32	31	21	23	13	12	32 G	31 G	21 G	23 G	13 G	12 G
INTRASCENDENTE	35,68	51,18	62,7	32,3	46,50	62,64	35,78	51,51	62,2	32,52	46,13	62,33
COLISION	10,19	19,32	19,88	10,71	19,71	19,33	9,74	19,44	20,95	9,99	20,25	20,12

Tabla 3.9: Porcentaje de partidos de colisión e intrascendentes para los grupos de 3 con 2 clasificados en una competición de ida y vuelta.

Como se puede apreciar en la Tabla 3.9, vemos que hay menor número de partidos con riesgo de colisión cuando el último partido se juega entre los 2 miembros más flojos con prácticamente un 10% de los casos. Respecto a los otros posibles enfrentamientos observamos que se obtiene cerca de un 20%, por lo que interesaría que el último partido se jugase entre los equipos teóricamente más flojos. Además, observamos que es en estos casos cuando el número de partidos intrascendentes es menor, algo que se puede entender como que en ese último partido, ambos van a llegar 'jugándose' un puesto para clasificarse a la siguiente ronda. Por el contrario, tendríamos que evitar que la última jornada la jugasen los 2 equipos más fuertes. De la misma forma que en los casos anteriores, se coincide que con el sorteo de Guyon no dan resultados distintos.

Fijémonos en el caso de partido único ahora. Este hecho ha sido estudiado por Julien Guyon [3], donde las probabilidades de ganar de cada equipo dependen de si es considerado fuerte, medio o débil, dando siempre las mismas probabilidades. En él, decía que, para minimizar el riesgo de colisión, el equipo fuerte debía jugar sus partidos el primero, ya que se supone que obtendrá una mejor puntuación que si cualquiera de los otros dos equipos jugaran sus correspondientes partidos antes. A diferencia de Guyon, nuestro modelo da unas estimaciones más precisas. Veamos qué resultados obtenemos con nuestra simulación.

	31	21	23	31 G	21 G	23 G
INTRASCENDENTE	11,05	28,48	4,52	11,55	28,09	4,23
COLISION	27,55	24,23	18,6	27,53	24,58	19,25

Tabla 3.10: Porcentaje de partidos de colisión e intrascendentes para los grupos de 3 con 2 clasificados en una competición a partido único.

Como se puede apreciar en la Tabla 3.10, confirmamos lo que decía Guyon, en nuestro caso obtenemos que, si el último enfrentamiento se produce entre el segundo y tercer miembro, el porcentaje de partidos de colisión se reduce. Observamos, por otro lado, que el mayor porcentaje de estos partidos se darían si se enfrentasen el primer equipo y el tercero, algo que se puede deber principalmente a que el equipo 1 (ya clasificado) dejase ganar al equipo 3. Respecto a los partidos intrascendentes, vemos que, si juegan el último partido entre el equipo medio y el flojo, solamente un 4 % de los partidos serán intrascendentes, un porcentaje bastante bajo comparado con el resto de valores, ya que será habitual que en esa última jornada, ambos equipos se jueguen el pase de ronda. Por tanto, interesaría que el último enfrentamiento se quedase sin jugar el equipo más fuerte.

3.3. Análisis de la justicia del sistema

Para decidir si el sistema es justo, tendremos que mirar la cantidad de veces que pasa de ronda cada equipo y ver que efectivamente los mejores equipos tienen mayor probabilidad de clasificarse. Podemos ver las veces que se clasifica cada equipo en el Apéndice B. Cabe destacar que para el análisis de la justicia no va a tener importancia el orden en el que se juegan los partidos, sino que nos centraremos en el tipo de sorteo, el número de equipos por grupos o el tipo competición si es de ida y vuelta o si es a partido único.

Comenzamos por los grupos de 4. Para saber quiénes son los 'mejores equipos', hemos decidido crear un programa en el que todos los equipos jueguen contra todos y una vez finalizado seleccionamos a los 16 mejores. Tras 10000 simulaciones, extraemos el número de ocasiones en que ha quedado cada equipo entre esos 16 y calculamos la correlación lineal que hay entre este valor y el número de veces que se habría clasificado con nuestro modelo. Vemos que en todos los casos obtenemos unas correlaciones del 95 %, sin embargo, para asegurarnos de que la relación es buena, nos fijaremos también en el coeficiente de correlación de concordancia [7].

Definición. Llamaremos **coeficiente de correlación de concordancia** a aquel coeficiente que mide la correlación entre 2 variables para poder evaluar la cercanía entre dos puntuaciones. La fórmula para su cálculo vendrá dada por

$$\rho_c = \frac{2\rho\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2 + (\mu_x - \mu_y)^2} \quad (3.1)$$

donde μ_x y μ_y son las medias para las dos variables, σ_x^2 y σ_y^2 las varianzas correspondientes y ρ el coeficiente de correlación entre ambas.

Al observar los resultados obtenidos (ver Apéndice B), se puede apreciar que un sistema es más justo cuando se disputa a doble partido en vez de a partido único, algo que es entendible ya que es más fácil que un equipo bueno se clasifique cuando juegue un mayor número de partidos.

Cabe destacar que a diferencia de los grupos de 4, que se obtienen valores similares, en los grupos de 3 si que se presenta una reducción de la concordancia cuando se utiliza en método de Guyon (Se pueden ver en Apéndice B). Es por ello, que obtener el coeficiente de correlación de concordancia nos va a permitir distinguir mejor que esquema es más justo.

A continuación, se muestra en la Tabla 3.11 la diferencia de los coeficientes de concordancia según las características de la competición. Además, representaremos gráficamente los grupos de 3 con 1 clasificado con los grupos de 4 para poder comparar como afecta la variación de esta correlación dependiendo

del número de equipos por grupo, el método de sorteo y el tipo de competición. Las consecuencias de estas diferencias se pueden ver en los diagramas de dispersión mostrados en la Figura 3.5 y en la Figura 3.6.

	IDA/VUELTA FIFA	IDA/VUELTA GUYON	P. ÚNICO FIFA	P. ÚNICO GUYÓN
C.CONCORDANCIA GR3 1CL	83,76	79,73	74,29	69,75
C.CONCORDANCIA GR3 2 CL	84,08	81,13	72,64	68,88
C.CONCORDANCIA GR4	91,48	92,29	86,02	86,91

Tabla 3.11: Coeficiente de concordancia para la formación 12 según sorteo y tipo de competición para grupos de 3 con 1 clasificado.

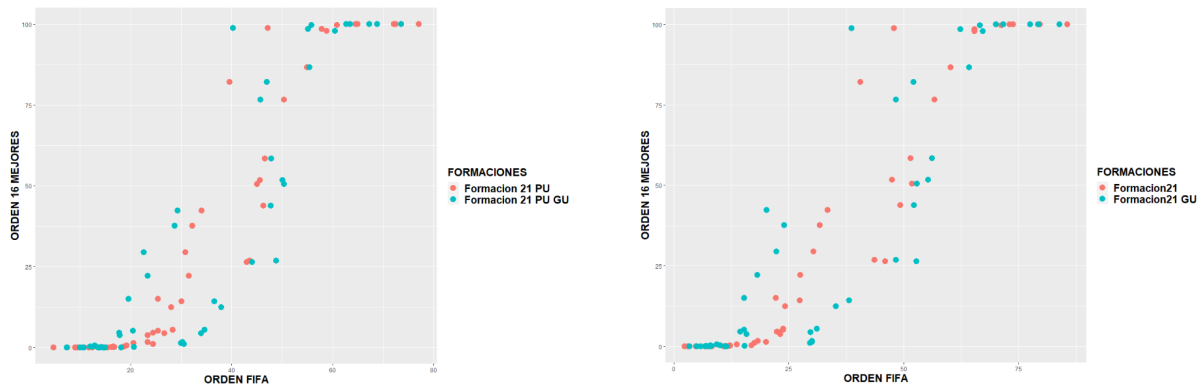


Figura 3.5: Diagrama de dispersión para la formación 12 según sorteo y tipo de competición para grupos de 3 con 1 clasificado a la izquierda con partido único y a la derecha con ida y vuelta.

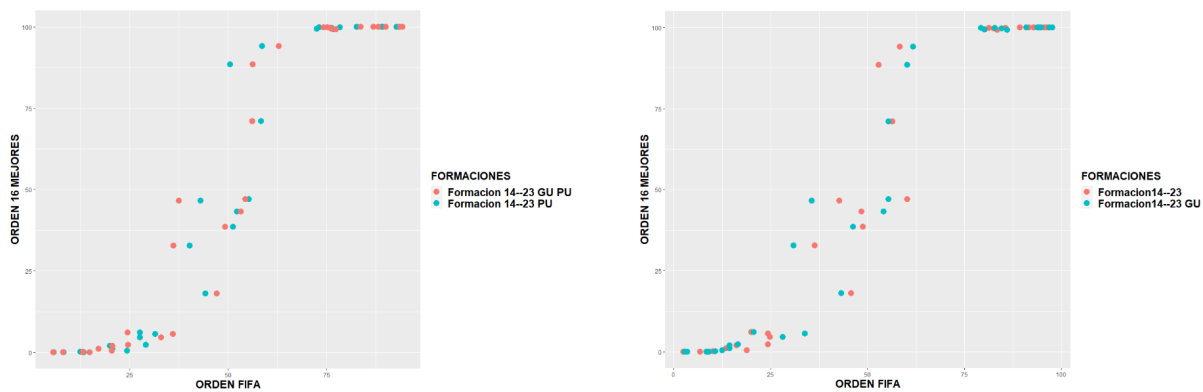


Figura 3.6: Diagrama de dispersión para la formación 14-32 según sorteo y tipo de competición para grupos de 4 a la izquierda con partido único y a la derecha con ida y vuelta.

Vemos que los 4 casos presentan una forma bastante similar que se asemeja a una curva con forma de S, algo que tiene bastante sentido ya que cuanto mejor es un equipo, más normal es que se clasifique, lo que vendría representado por la parte superior de la S, lo mismo pero al revés pasaría con los equipos malos. Como apreciamos en la Tabla 3.11 se puede ver la diferencia de concordancias en los grupos de 3 y la casi inexistente en los grupos de 4. Esto se ve reflejado en la Figura 3.5 para el grupo de 3, en la que los casos de Guyon se presentan mucho más dispersos que con el método FIFA, mientras que para los grupos de 4 en la Figura 3.6 se ve que siguen una distribución muy similar. Podemos apreciar que en el diagrama de dispersión hay ciertas deficiencias, por ejemplo en los grupos de 4, equipos como el Leipzig, Inter de Milán o Atalanta cuyo ranking relativo es mayor, se clasifican mayor número de veces

que algunos, como el Shakhtar Donetsk, cuyo ranking es menor. Esto se debe a que el ranking FIFA no se corresponde con las características que hemos considerado para nuestro modelo.

3.3.1. Mejoras para obtener un modelo más justo

A pesar de que se han obtenido unos resultados bastante aceptables con unas correlaciones buenas, vamos a intentar buscar alguna mejora que nos permita tener una correlación de concordancia mejor entre el número de veces que se clasifica cada equipo. Para ello vamos a crear otro programa donde se juegue una liga entre todos los miembros de la competición con 10000 simulaciones y en lugar de formar los bombos dependiendo del ranking FIFA correspondiente a cada equipo, lo formaremos según su puesto en la simulación realizada, cuántos más puntos obtenga, mejor clasificación. De esta manera se pretende obtener una distribución de los bombos más orientada a la 'fuerza' de cada equipo basada en las características que hemos considerado y no por el Ranking FIFA.

Una vez ordenados los equipos de esta nueva forma, hemos repetido el estudio anterior para observar el número de partidos de colisión e intrascendentes y vemos que se obtienen valores prácticamente iguales, por tanto, el mejor orden para jugar la última jornada coincide con lo analizado cuando ordenábamos por el ranking FIFA. Debido a esta igualdad, continuaremos el estudio fijándonos en el mejor orden en cada caso como podemos ver en la Tabla 3.12, dejando sin analizar el sorteo de Guyon ya que habíamos visto que no afectaba al número de partidos de colisión e intrascendentes.

	Grupos 4 IYV INTR	Grupos 4 IYV COLISION	Grupos 3 1 CL IYV INTR	Grupos 3 1 CL IYV COLISION	Grupos 3 2 CL IYV INTR	Grupos 3 2 CL IYV COLISION
FIFA	46,75	12,26	35,86	6,32	32,30	10,71
SIMULACION	47,70	10,90	35,66	5,72	31,20	9,25
	Grupos 4 PU INTR	Grupos 4 PU COLISION	Grupos 3 1 CL PU INTR	Grupos 3 1 CL PU COLISION	Grupos 3 2 CL PU INTR	Grupos 3 2 CL PU COLISION
FIFA	10,60	14,03	4,18	3,89	4,52	18,60
SIMULACION	10,82	12,13	3,58	3,47	3,81	17,96

Tabla 3.12: Comparación porcentajes partidos intrascendentes y de colisión entre los enfrentamientos óptimos según el tipo de competición.

Respecto a la justicia de este nuevo sistema, podemos apreciar en la Tabla 3.13 la diferencia en el coeficiente de concordancia entre el orden propuesto por la FIFA y el que acabamos de estudiar. Vemos que, tanto en grupos de 4 como de 3, obtenemos una concordancia mayor con nuestra simulación. Esto quiere decir que hemos encontrado un sistema más justo que el anterior a pesar de que este ya era bueno.

	Grupos 4 IYV FIFA	Grupos 4 IYV GUYON	Grupos 3 1 CL IYV FIFA	Grupos 3 1 CL IYV GUYON	Grupos 3 2 CL IYV FIFA	Grupos 3 2 CL IYV GUYON
FIFA	91,48	92,29	83,70	79,71	84,69	81,52
SIMULACION	94,72	95,63	88,89	85,67	85,20	81,50
	Grupos 4 PU FIFA	Grupos 4 PU GUYON	Grupos 3 1 CL PU FIFA	Grupos 3 1 CL PU GUYON	Grupos 3 2 CL PU FIFA	Grupos 3 2 CL PU GUYON
FIFA	86,56	87,33	75,07	71,00	73,77	68,88
SIMULACION	89,82	89,63	78,83	75,79	74,81	69,03

Tabla 3.13: Comparación coeficientes de concordancia según tipo de sorteo y tipo de competición

A continuación en la Figura 3.7 hemos escogido el enfrentamiento óptimo para los grupos de 4.

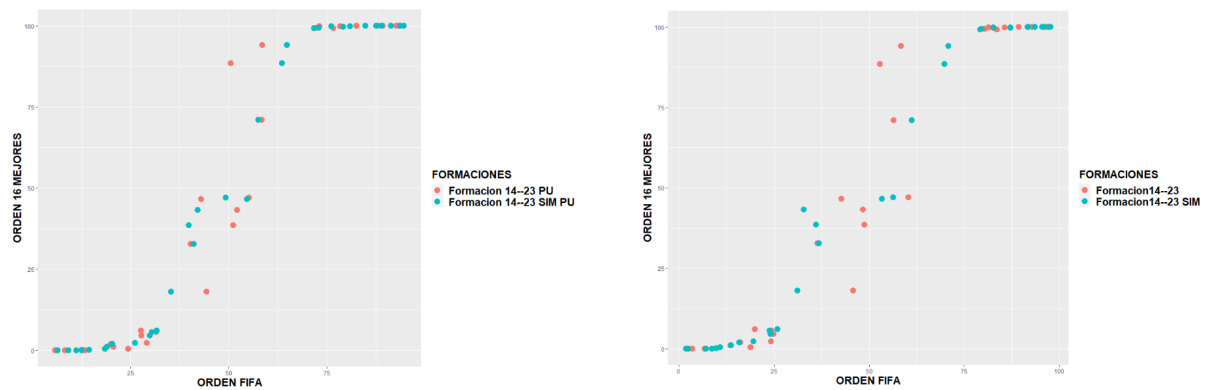


Figura 3.7: Diagrama de dispersión para la formación 14–23 comparando orden FIFA con nuestra simulación grupos de 4 a la izquierda con partido único y a la derecha con ida y vuelta.

Como podemos apreciar en la Figura 3.7, a pesar de que ambos siguen la curva de tipo S, se ve que en el caso de tener los equipos ordenados por el ranking FIFA, la parte central está más dispersa que en el caso en el que ordenamos los equipos después de haber jugado una liga.

Por lo tanto, si ordenamos los equipos por su potencial en lugar de por su ranking FIFA, obtenemos unos valores similares en cuanto a partidos intrascendentes y de colisión, pero además hemos podido conseguir que la correlación haya aumentado; este nuevo orden nos ha permitido obtener un sistema similar al de la FIFA pero que sería todavía más justo.

3.4. Conclusiones

En este ámbito, ya existen artículos relacionados como el ya mencionado [3] u otros que han aparecido recientemente como [1] en el que analizaban los partidos tanto de colisión como intrascendentes basándose en el ranking ELO de las selecciones nacionales utilizando, al igual que nosotros, la regresión de Poisson y proporcionando al final porcentajes de la cantidad de partidos que había de cada tipo. A diferencia de ellos, nosotros hemos trabajado con los equipos de la UEFA Champions League dando una estimación mucho más fina obtenida a partir de la regresión de Poisson y haciendo uso de variables predictoras que no pueden usarse en selecciones nacionales. De la misma forma que en el artículo antes mencionado, hemos obtenido los porcentajes de partidos de colisión e intrascendentes pero además, hemos querido analizar la justicia en las fases de grupos, algo que hasta el momento no ha aparecido en la literatura.

A la vista de los resultados que hemos obtenido, podemos sacar una serie de conclusiones generales que afectan a las 3 formaciones que hemos estudiado. Respecto al formato de sorteo, hemos podido ver que, tanto para el sorteo tradicional de la FIFA como para el propuesto Guyon, la cantidad de partidos intrascendentes como los de colisión son muy parecidos por lo que se puede afirmar que a pesar de que el sorteo de Guyon pretenda hacer formaciones de grupos más equitativas, vemos que este hecho no afecta a la hora de que haya mayor o menor número de este tipo de partidos. Además, cuando las fases de grupos son de ida y vuelta, si el último enfrentamiento es entre 2 mismos equipos, da igual que se juegue en 'casa' de uno o de otro, se obtienen porcentajes similares de los partidos estudiados. De la misma forma, se ha observado que si formamos los bombos como la FIFA o lo hacemos por nuestro método, la cantidad de estos tipos de partidos es similar.

Analizando la colisión e intrascendencia de los grupos según el número de integrantes en cada uno, vemos que para los grupos de 4 el mejor enfrentamiento en la última jornada sería que se disputara entre el equipo del bombo 1 y el del 4 y el del bombo 2 contra el del 3, mientras que el peor sería que se enfrentaran entre sí los 2 mejores. En cuanto a los grupos de 3 con 1 clasificado, el orden óptimo sería que jugasen los 2 mejores equipos en la última jornada y se trataría de evitar dejar para el último

partido el enfrentamiento entre los 2 peores. Ya por último, para los grupos de 3 con 2 clasificados, nos interesaría que la última jornada la disputasen los equipos de los bombos 2 y 3. Por el otro lado, trataríamos de evitar que el equipo que se queda sin jugar el último partido fuese el equipo del bombo 3.

Respecto a la justicia, queda claro que una competición será más justa cuando sean partidos de ida y vuelta en lugar de a partido único ya que si un equipo es favorito a clasificarse, le beneficiará jugar más partidos para conseguir el objetivo. Cabe destacar que, a diferencia de los grupos de 4, tanto en los grupos de 3 con 1 clasificado como con 2, existe una diferencia tanto en partido único como en ida y vuelta en la que el método de Guyon hace que sea menos justo en comparación con el sorteo tradicional, por tanto en estos casos habría que descartar este método. Además, podemos afirmar que con el sistema de la FIFA, obtenemos una correlación de concordancia muy buena entre el número de veces que se clasifica cada equipo y el número de ocasiones que se clasificarían si se jugara una hipotética liga entre ellos, sin embargo, en todos los casos vemos que si en lugar de utilizar el orden establecido por la posición en el ranking, lo hacemos en función de la posición en la que quedan al jugarse varias veces una liga entre los equipos, se obtiene un sistema todavía mejor que el anterior y que además mantiene la cantidad de partidos intrascendentes y de colisión que se obtenían por el método FIFA.

En general, observamos que las conclusiones obtenidas son lógicas y, en ese sentido, este trabajo confirma lo que otros autores como en [3] ya habían obtenido. El valor añadido del trabajo es que cuantifica, con un modelo muy cercano a la realidad, el porcentaje de ocurrencia de cada evento en cada tipo de organización de torneo y nos muestra como la programación de los partidos, en particular los enfrentamientos de la última jornada, es determinante para obtener partidos emocionantes. Está claro que el sistema de juego óptimo dependerá del formato de la competición que se vaya a jugar, pero lo que es seguro es que programar el orden de los partidos con antelación no tiene ningún efecto negativo sino por el contrario reduce el riesgo de ponerse de acuerdo y aumentar a su vez el atractivo de la última ronda.

Ya para terminar, en relación con la justicia de los torneos, queda claro que el actual método de ranking FIFA no es el mejor como ha quedado demostrado este año con la clasificación de algún equipo para octavos de final de la UEFA Champions League y que si se utiliza un procedimiento como el propuesto en este TFG, con una simulación de una liga entre todos los participantes, muy probablemente dará lugar a torneos más justos.

Apéndice A

PASCAL

Simulaciones con grupos de 4 equipos

Programa principal grupos de 4

En este apartado se muestra el programa que se ha utilizado para hacer las correspondientes simulaciones de los grupos 4, sacando así los partidos de colisión, partidos intrascendentes y la cantidad de veces que se clasifica cada equipo después de realizar una simulación de Monte Carlo. En este programa se añaden ya los datos tanto si las fases de grupo fuesen de ida y vuelta o como de partido único. Además están añadidos tantos los datos con las tablas ordenando los equipos por ranking FIFA, como las tablas en las que se ordenan los equipos por la simulación de jugar una liga entre todos los equipos (Está a continuación)

```
program ProgramaGrupos4;
uses crt,math;
type
  vector=array[1..32]of integer;
  Tmatriz=array[1..8,1..4] of integer;
  Tmatriz32=array[1..32,1..32] of integer;
  vector4=array[1..4] of integer;
  jornada=array[1..8,1..6,1..2] of integer;
var
  i,j,m,k,r,numero,sorteo,numero1,partidos,orden:integer;
  r1,r2,r3,r4,r5,r6,r7,r8:integer;
  rr1,rr2,rr3,rr4,rr5,rr6,rr7,rr8:integer;
  t1,t2,t3,t4,t5,t6,t7,t8:integer;
  lista1,lista2,lista3,lista4,lista5,lista6,lista7,lista8:array[1..8] of integer;
  g,puntosequipo,golesafavorequipo,golesencontraequipo:array[1..32] of integer;
  equipo:array[1..8,1..4]of integer;
  casa,fuera:jornada;
  a:real;
  golesenfrentamientos,puntosenfrentamientos:Tmatriz32;
  misdatos,misdatos1,misdatosu:array[1..1024] of real;      {Los tengo que definir como reales ya que coge valores reales}
  matriz,matriz1,matrizuni:array[1..32,1..32] of real;
  posicion,posicion1,posicionu:integer;
  f,q,u,Resul:text;
  numpint,numpcol:integer;
  intentos,intentos1:integer;
  p,s:integer;
  vclas:vector;
  vect4,vect41:vector4;
  {Definimos la funcion densidad de Poisson y para ello defino antes el factorial}
  function factorial( num : integer) : integer;
  begin
    if num = 1 then factorial := 1
    else if num = 0 then factorial := 1
    else factorial := num * factorial( num-1 );
  end;
function Poisson(landa:real; x:integer):real;
  begin
    Poisson:=Exp(-landa)*power(landa,x)/factorial(x);
  end;
{Simulo una Poisson y que me diga cuantos goles marca el equipo local tomando un valor aleatorio y y mirando donde cae en la probabilidad acumulada}
function simularPoissonL(landa:real):integer;
  var i:integer;
  y:real;
  valor:array[1..10] of real;
  pacum:array[1..10] of real;
  glocal:integer;
begin
for i:=0 to 9 do
  begin
```

```

    valor[i+1]:=Poisson(landa,i);
    if i=0 then pacum[i+1]:=valor[i+1]
    else pacum[i+1]:=pacum[i]+valor[i+1];
    y:=random();
    glocal:=0;
    while (pacum[glocal+1]<y) and (glocal<=10) do glocal:=glocal+1;
    end;
simularPoissonL:=glocal;
end;
{Simulo una Poisson y que me diga cuantos goles marca el equipo visitante tomando un valor aleatorio y y mirando donde
cae en la probabilidad acumulada}
function simularPoissonV(landa:real):integer;
    var
        i:integer;
        y:real;
        valor:array[1..10] of real;
        pacum:array[1..10] of real;
        gvisitante:integer;
begin
for i:=0 to 9 do
    begin
        valor[i+1]:=Poisson(landa,i);
        if i=0 then pacum[i+1]:=valor[i+1]
        else pacum[i+1]:=pacum[i]+valor[i+1];
        y:=random();
        gvisitante:=0;
        while (pacum[gvisitante+1]<y) and (gvisitante<=10) do gvisitante:=gvisitante+1;
        end;
simularPoissonV:=gvisitante;
end;

{Hago un procedimiento que meto a los equipos que juegan y el valor del landa correspondiente, y se me van añadiendo los valores
de puntos, goles a favor
y goles en contra que tengo de la jornada anterior}
procedure simularpartido(a:integer;b:integer;c:real;d:real;var gfavoreq:vector;var gcontrareq: vector;
var ptoseqpo:vector;var golesentreequipos:Tmatriz32;var puntosentreequipos:Tmatriz32);

var
    golesfavora:integer;
    golescontraa:integer;
begin
    golesfavora:=simularPoissonL(c);
    golescontraa:=simularPoissonV(d);
    gfavoreq[a]:=gfavoreq[a]+golesfavora;
    gcontrareq[a]:=gcontrareq[a]+golescontraa;
    gfavoreq[b]:=gfavoreq[b]+golescontraa;
    gcontrareq[b]:=gcontrareq[b]+golesfavora;
    if (golesfavora>golescontraa) then
        begin
            ptoseqpo[a]:= ptoseqpo[a]+3;
            puntosentreequipos[a,b]:=puntosentreequipos[a,b]+3;
        end;

    if (golesfavora<golescontraa) then
        begin
            ptoseqpo[b]:= ptoseqpo[b]+3;
            puntosentreequipos[b,a]:=puntosentreequipos[b,a]+3;
        end;
    if(golesfavora=golescontraa) then
        begin
            ptoseqpo[a]:= ptoseqpo[a]+1;
            ptoseqpo[b]:= ptoseqpo[b]+1;
            puntosentreequipos[a,b]:=puntosentreequipos[a,b]+1;
            puntosentreequipos[b,a]:=puntosentreequipos[b,a]+1;
        end;
    golesentreequipos[a,b]:=golesfavora+golesentreequipos[a,b];
    golesentreequipos[b,a]:=golescontraa+golesentreequipos[b,a];
end;

{Esto de Primerclasificado,Segundoclasificado,Ultimoclasificado y Tercerclasificado, lo hago para establecer un orden,
para el caso del primero y segundo empezare mirando por el primer miembro del grupo, para el caso del tercero
y cuarto, empezare mirando por el último, con esto consigo que en caso de empate no se repita ninguno y de esta
manera tengo un orden establecido, lo siguiente será ordenarlos en caso de empate.}

function Primerclasificado(i:integer;puntuacion:vector;EQ:Tmatriz):integer;
var j,POSI,s:integer;
    MAYOR:array[1..8]of integer;
    matrizpuntos,matrizordenada:array[1..8,1..4] of integer;
    equipo:array[1..8,1..4]of integer;
begin
    for j:=1 to 4 do
        begin
            equipo[i,j]:=EQ[i,j];
            matrizpuntos[i,j]:=puntuacion[equipo[i,j]];
            {writeln(matrizpuntos[i,j]);}
            MAYOR[i]:=matrizpuntos[i,1];
            POSI:=1;
            for s:=2 to 4 do
                if matrizpuntos[i,s]>MAYOR[i] then
                    begin
                        MAYOR[i]:=matrizpuntos[i,s];
                        POSI:=s;
                    end;
            matrizordenada[i,1]:=equipo[i,POSI];
        end;
    Primerclasificado:=matrizordenada[i,1];
end;

function Segundoclasificado(i:integer;puntuacion:vector;EQ:Tmatriz):integer;

```

```

var j,POSI,POSI2,s:integer;
MAYOR,MAYOR2:array[1..8]of integer;
matrizpuntos,matrizordenada:array[1..8,1..4] of integer;
equipo:array[1..8,1..4]of integer;

begin
  for j:=1 to 4 do
    begin
      equipo[i,j]:=EQ[i,j];
      matrizpuntos[i,j]:=puntuacion[equipo[i,j]];
      MAYOR[i]:=matrizpuntos[i,1];
      POSI:=1;
      for s:=2 to 4 do
        if matrizpuntos[i,s]>MAYOR[i] then
          begin
            MAYOR[i]:=matrizpuntos[i,s];
            POSI:=s;
          end;
          matrizordenada[i,1]:=equipo[i,POSI];
        end;
      {Aquí acabaria el primer clasificado y analizamos los casos del segundo que mas}
      if POSI=1 then
        begin
          for j := POSI to 3 do
            begin
              matrizpuntos[i,j] := matrizpuntos[i,j+1];
              matrizpuntos[i,j+1]:=puntuacion[equipo[i,j+1]];
              MAYOR2[i]:=matrizpuntos[i,1];
              POSI2:=1;
              for s:=2 to 3 do
                if matrizpuntos[i,s]>MAYOR2[i] then
                  begin
                    MAYOR2[i]:=matrizpuntos[i,s];
                    POSI2:=s;
                  end;
                  matrizordenada[i,2]:=equipo[i,POSI2+1];
                end;
              end
            else if POSI=2 then
              begin
                for j := POSI to 3 do
                  begin
                    matrizpuntos[i,1] := matrizpuntos[i,1];
                    matrizpuntos[i,j] := matrizpuntos[i,j+1];
                    matrizpuntos[i,j+1]:=puntuacion[equipo[i,j+1]];
                    MAYOR2[i]:=matrizpuntos[i,1];
                    POSI2:=1;
                    for s:=2 to 3 do
                      if matrizpuntos[i,s]>MAYOR2[i] then
                        begin
                          MAYOR2[i]:=matrizpuntos[i,s];
                          POSI2:=s;
                        end;
                      end;
                    if POSI2=1 then matrizordenada[i,2]:=equipo[i,POSI2]
                    else matrizordenada[i,2]:=equipo[i,POSI2+1];
                  end
                else if POSI=3 then
                  begin
                    j:=3;
                    matrizpuntos[i,1] := matrizpuntos[i,1];
                    matrizpuntos[i,2] := matrizpuntos[i,2];
                    matrizpuntos[i,j] := matrizpuntos[i,j+1];
                    matrizpuntos[i,j+1]:=puntuacion[equipo[i,j+1]];
                    MAYOR2[i]:=matrizpuntos[i,1];
                    POSI2:=1;
                    for s:=2 to 3 do
                      if matrizpuntos[i,s]>MAYOR2[i] then
                        begin
                          MAYOR2[i]:=matrizpuntos[i,s];
                          POSI2:=s;
                        end;
                      if POSI2=1 then matrizordenada[i,2]:=equipo[i,POSI2]
                      else if POSI2=2 then matrizordenada[i,2]:=equipo[i,POSI2]
                      else matrizordenada[i,2]:=equipo[i,POSI2+1];
                    end
                  end
                else if POSI=4 then
                  begin
                    for j := 1 to 3 do
                      begin
                        matrizpuntos[i,j] := matrizpuntos[i,j];
                        matrizpuntos[i,j]:=puntuacion[equipo[i,j]];
                        MAYOR2[i]:=matrizpuntos[i,1];
                        POSI2:=1;
                        for s:=2 to 3 do
                          begin
                            if matrizpuntos[i,s]>MAYOR2[i] then
                              begin
                                MAYOR2[i]:=matrizpuntos[i,s];
                                POSI2:=s;
                              end;
                            end;
                          matrizordenada[i,2]:=equipo[i,POSI2];
                        end;
                      end;
                    Segundoclasificado:=matrizordenada[i,2];
                  end;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;
function Ultimoclasificado(i:integer;puntuacion:vector;EQ:Tmatriz):integer;

```

```

var j, POSI4, s: integer;
MENOR: array[1..8] of integer;
matrizpuntos, matrizordenada: array[1..8, 1..4] of integer;
equipo: array[1..8, 1..4] of integer;
begin
  for j:=1 to 4 do
    begin
      equipo[i, j] := EQ[i, j];
      matrizpuntos[i, j] := puntuacion[equipo[i, j]];
      MENOR[i] := matrizpuntos[i, 4];
      POSI4:=4;
      for s:=3 downto 1 do
        begin
          if matrizpuntos[i, s] < MENOR[i] then
            begin
              MENOR[i] := matrizpuntos[i, s];
              POSI4:=s;
            end;
          end;
          matrizordenada[i, 4] := equipo[i, POSI4];
        end;
      Ultimoclasificado:=matrizordenada[i, 4];
    end;
  function Tercerclasificado(i: integer; puntuacion: vector; EQ: Tmatriz): integer;
  var j, POSI3, POSI4, s: integer;
  MENOR, MENOR2: array[1..8] of integer;
  matrizpuntos, matrizordenada: array[1..8, 1..4] of integer;
  equipo: array[1..8, 1..4] of integer;
  begin
    for j:=1 to 4 do
      begin
        equipo[i, j] := EQ[i, j];
        matrizpuntos[i, j] := puntuacion[equipo[i, j]];
        MENOR[i] := matrizpuntos[i, 4];
        POSI4:=4;
        for s:=3 downto 1 do
          begin
            if matrizpuntos[i, s] < MENOR[i] then
              begin
                MENOR[i] := matrizpuntos[i, s];
                POSI4:=s;
              end;
            end;
            matrizordenada[i, 4] := equipo[i, POSI4];
          end;
        {Queremos sacar ahora el tercer mejor del grupo, igual que antes empezabamos por el primer
        elemento e ibamos viendo uno a uno, ahora haremos lo mismo pero empezando desde el ultimo
        y yendo hacia abajo}

        if POSI4=1 then
          begin
            for j := POSI4 to 3 do
              begin
                matrizpuntos[i, j+1] := puntuacion[equipo[i, j+1]];
                matrizpuntos[i, j] := matrizpuntos[i, j+1];
                MENOR2[i] := matrizpuntos[i, 3];
                POSI3:=3;
                for s:=2 downto 1 do
                  begin
                    if matrizpuntos[i, s] < MENOR2[i] then
                      begin
                        MENOR2[i] := matrizpuntos[i, s];
                        POSI3:=s;
                      end;
                    end;
                    matrizordenada[i, 3] := equipo[i, POSI3+1];
                  end;
                end;
              end;
            else if POSI4=2 then
              begin
                for j := POSI4 to 3 do
                  begin
                    matrizpuntos[i, 1] := matrizpuntos[i, 1];
                    matrizpuntos[i, j+1] := puntuacion[equipo[i, j+1]];
                    matrizpuntos[i, j] := matrizpuntos[i, j+1];
                    MENOR2[i] := matrizpuntos[i, 3];
                    POSI3:=3;
                    for s:=2 downto 1 do
                      begin
                        if matrizpuntos[i, s] < MENOR2[i] then
                          begin
                            MENOR2[i] := matrizpuntos[i, s];
                            POSI3:=s;
                          end;
                        end;
                      end;
                    if POSI3=1 then matrizordenada[i, 3] := equipo[i, POSI3]
                    else matrizordenada[i, 3] := equipo[i, POSI3+1];
                  end;
                end;
              end;
            else if POSI4=3 then
              begin
                j:=3;
                matrizpuntos[i, 1] := matrizpuntos[i, 1];
                matrizpuntos[i, 2] := matrizpuntos[i, 2];
                matrizpuntos[i, j+1] := puntuacion[equipo[i, j+1]];
                matrizpuntos[i, j] := matrizpuntos[i, j+1];
                MENOR2[i] := matrizpuntos[i, 3];
                POSI3:=3;
                for s:=2 downto 1 do

```

```

begin
  if matrizpuntos[i,s]<MENOR2[i] then
    begin
      MENOR2[i]:=matrizpuntos[i,s];
      POSI3:=s;
    end;
  end;
  if POSI3=1 then matrizordenada[i,3]:=equipo[i,POSI3]
  else if POSI3=2 then matrizordenada[i,3]:=equipo[i,POSI3]
  else matrizordenada[i,3]:=equipo[i,POSI3+1];
end
else if POSI4=4 then
begin
  for j := 1 to 3 do
    begin
      matrizpuntos[i,j]:=puntuacion[equipo[i,j]];
      MENOR2[i]:=matrizpuntos[i,3];
      POSI3:=3;
      for s:=2 downto 1 do
        begin
          if matrizpuntos[i,s]<MENOR2[i] then
            begin
              MENOR2[i]:=matrizpuntos[i,s];
              POSI3:=s;
            end;
          end;
          matrizordenada[i,3]:=equipo[i,POSI3];
        end;
      end;
      Tercerclasificado:=matrizordenada[i,3];
    end;
  {Sacaremos ahora la clasificacion buena definitiva, para ello, nos fijamos en los empates
  entre 2 equipos, miraremos su gol average particular y en caso de que coincidan,
  miraremos la diferencia de goles general, si coinciden, lo hacemos a sorteo. En caso
  de que haya triple empate, miraremos los puntos que han obtenido entre sí los equipos que
  esten empatados, el que más puntos haya conseguido, mejor clasificado. Si coinciden, pasaremos
  a mirar la diferencia de goles y si vuelven a coincidir ya hacemos un sorteo.}
function Clasificacion(i:integer;puntosequipo:vector;golesafavorequipo:vector;golesencontraequipo:vector;
golesenfrentamientos:Tmatriz32;puntosenfrentamientos:Tmatriz32;EQ:Tmatriz):vector4;
var
  matrizordenada:Tmatriz;
  auxiliar,auxiliar2,alea,alear,tp,ts,tu,pp,dd,taux:integer;
  aux:vector4;
begin
  matrizordenada[i,1]:=Primerclasificado(i,puntosequipo,EQ);
  matrizordenada[i,2]:=Segundoclasificado(i,puntosequipo,EQ);
  matrizordenada[i,3]:=Tercerclasificado(i,puntosequipo,EQ);
  matrizordenada[i,4]:=Ultimoclasificado(i,puntosequipo,EQ);
  {ANALIZAMOS TODOS LOS POSIBLES CASOS 1=2, 1=3, 1=4, 2=3,2=4;3=4 y miramos su gol average particular
  en el caso de que haya empate entre dos equipos, miramos los goles que se han marcado el uno al otro y
  vemos que si la diferencia entre los goles anotados es mayor en uno que en otro, ese equipo ira por
  delante, en caso de que el gol average fuera igual, nos fijaríamos en la diferencia de goles general}
  for pp:=1 to 3 do
    begin
      dd:=pp+1;
      if puntosequipo[matrizordenada[i,pp]]=puntosequipo[matrizordenada[i,dd]] then
        begin
          if golesenfrentamientos[matrizordenada[i,pp],matrizordenada[i,dd]]<
          golesenfrentamientos[matrizordenada[i,dd],matrizordenada[i,pp]] then
            begin
              auxiliar:=matrizordenada[i,pp];
              matrizordenada[i,pp]:=matrizordenada[i,dd] ;
              matrizordenada[i,dd]:=auxiliar;
            end;
          if golesenfrentamientos[matrizordenada[i,pp],matrizordenada[i,dd]]=
          golesenfrentamientos[matrizordenada[i,dd],matrizordenada[i,pp]] then
            begin
              if golesafavorequipo[matrizordenada[i,pp]]-golesencontraequipo[matrizordenada[i,pp]]<
              golesafavorequipo[matrizordenada[i,dd]]-golesencontraequipo[matrizordenada[i,dd]] then
                begin
                  auxiliar:=matrizordenada[i,pp];
                  matrizordenada[i,pp]:=matrizordenada[i,dd] ;
                  matrizordenada[i,dd]:=auxiliar;
                end;
              if golesafavorequipo[matrizordenada[i,pp]]-
              golesencontraequipo[matrizordenada[i,pp]]=golesafavorequipo[matrizordenada[i,dd]]-
              golesencontraequipo[matrizordenada[i,dd]] then
                begin
                  {randomize();}
                  alea:=random(2)+1;
                  if alea=2 then
                    begin
                      auxiliar:=matrizordenada[i,pp];
                      matrizordenada[i,pp]:=matrizordenada[i,dd] ;
                      matrizordenada[i,dd]:=auxiliar;
                    end;
                end;
            end;
          end;
        end;
      end;
    end;
  end;
  {+++++}
  {En este caso, analizaremos el caso de que hubiera triple empate, dado que ya es largo de por si, para
  desempatar nos fijaremos solamente en la diferencia de goles general(hacemos todas las posibles combinaciones)}
  {En caso de que haya triple empate, miraremos los puntos que han obtenido entre sí los equipos que esten
  empatados, el que más puntos haya conseguido, mejor clasificado. Si coinciden, pasaremos a mirar la
  diferencia de goles y si vuelven a coincidir ya hacemos un sorteo.}
  for tp:=1 to 2 do
    begin

```



```

puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]] and
(puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]>
puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]] and
(puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]<
puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) then
begin

auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,tu] ;
matrizordenada[i,ts]:=matrizordenada[i,ts];
matrizordenada[i,tu]:=auxiliar;
end;
if (puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]>
puntosenfrentamientos[matrizordenada[i,2],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,2],matrizordenada[i,tu]] and
(puntosenfrentamientos[matrizordenada[i,2],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,2],matrizordenada[i,tu]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,2]]) then
begin
matrizordenada[i,tp]:=matrizordenada[i,tp];
begin
if golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]>
golesafavorequipo[matrizordenada[i,tu]]-golesencontraequipo[matrizordenada[i,tu]]
then
begin
matrizordenada[i,ts]:=matrizordenada[i,ts];
matrizordenada[i,tu]:=matrizordenada[i,tu];
end;
if golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]<
golesafavorequipo[matrizordenada[i,tu]]-golesencontraequipo[matrizordenada[i,tu]]
then
begin
auxiliar:=matrizordenada[i,ts];
matrizordenada[i,ts]:=matrizordenada[i,tu] ;
matrizordenada[i,tu]:=auxiliar;
end;
if golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]=
golesafavorequipo[matrizordenada[i,tu]]-golesencontraequipo[matrizordenada[i,tu]]
then
begin
alea:=random(2)+1;
if alea=1 then
begin
matrizordenada[i,ts]:=matrizordenada[i,ts];
matrizordenada[i,tu]:=matrizordenada[i,tu];
end;
if alea=2 then
begin
auxiliar:=matrizordenada[i,ts];
matrizordenada[i,ts]:=matrizordenada[i,tu] ;
matrizordenada[i,tu]:=auxiliar;
end;
end;
end;
end;
if (puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]]>
puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]] and
(puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) then
begin
auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,ts];
matrizordenada[i,ts]:=auxiliar;
if golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]>golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]] then
begin
matrizordenada[i,ts]:=matrizordenada[i,ts];
matrizordenada[i,tu]:=matrizordenada[i,tu];
end;
if golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]<golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]] then
begin
auxiliar:=matrizordenada[i,ts];
matrizordenada[i,ts]:=matrizordenada[i,tu] ;
matrizordenada[i,tu]:=auxiliar;
end;
if golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]=
golesafavorequipo[matrizordenada[i,tu]]-golesencontraequipo[matrizordenada[i,tu]]
then
begin
alea:=random(2)+1;

```

```

if alea=1 then
begin
matrizordenada[i,ts]:=matrizordenada[i,ts];
matrizordenada[i,tu]:=matrizordenada[i,tu];
end;
if alea=2 then
begin
auxiliar:=matrizordenada[i,ts];
matrizordenada[i,ts]:=matrizordenada[i,tu] ;
matrizordenada[i,tu]:=auxiliar;
end;
end;
end;
if (puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]>
puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]) and
(puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]] =
puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]]) then
begin
auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,tu];
matrizordenada[i,tu]:=auxiliar;
if golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]>
golesafavorequipo[matrizordenada[i,tu]]-golesencontraequipo[matrizordenada[i,tu]]
then
begin
matrizordenada[i,ts]:=matrizordenada[i,ts];
matrizordenada[i,tu]:=matrizordenada[i,tu];
end;
if golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]<
golesafavorequipo[matrizordenada[i,tu]]-golesencontraequipo[matrizordenada[i,tu]]
then
begin
auxiliar:=matrizordenada[i,ts];
matrizordenada[i,ts]:=matrizordenada[i,tu] ;
matrizordenada[i,tu]:=auxiliar;
end;
if golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]=
golesafavorequipo[matrizordenada[i,tu]]-golesencontraequipo[matrizordenada[i,tu]]
then
begin
alea:=random(2)+1;
if alea=1 then
begin
matrizordenada[i,ts]:=matrizordenada[i,ts];
matrizordenada[i,tu]:=matrizordenada[i,tu];
end;
if alea=2 then
begin
auxiliar:=matrizordenada[i,ts];
matrizordenada[i,ts]:=matrizordenada[i,tu] ;
matrizordenada[i,tu]:=auxiliar;
end;
end;
end;
end;
if (puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]] =
puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]]) and
(puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]]>
puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) then
begin
matrizordenada[i,3]:=matrizordenada[i,tu];
if golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]]>golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]] then
begin
matrizordenada[i,tp]:=matrizordenada[i,tp];
matrizordenada[i,ts]:=matrizordenada[i,ts];
end;
if golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]]<
golesafavorequipo[matrizordenada[i,ts]]-golesencontraequipo[matrizordenada[i,ts]]
then
begin
auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,ts] ;
matrizordenada[i,ts]:=auxiliar;
end;
if golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]]=
golesafavorequipo[matrizordenada[i,ts]]-golesencontraequipo[matrizordenada[i,ts]]
then
begin
alea:=random(2)+1;
if alea=1 then
begin
matrizordenada[i,tp]:=matrizordenada[i,tp];
matrizordenada[i,ts]:=matrizordenada[i,ts];
end;
if alea=2 then

```



```

begin
    auxiliar:=matrizordenada[i,tp];
    matrizordenada[i,tp]:=matrizordenada[i,ts] ;
    matrizordenada[i,ts]:=auxiliar;
end;
end;
end;
if (puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]=
puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) and
(puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]]<
puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) then
begin
    auxiliar:=matrizordenada[i,tu];
    matrizordenada[i,tu]:=matrizordenada[i,ts];
    matrizordenada[i,ts]:=auxiliar;
    if golesafavorequipo[matrizordenada[i,tp]]-
    golesencontraequipo[matrizordenada[i,tp]]>
    golesafavorequipo[matrizordenada[i,ts]]-golesencontraequipo[matrizordenada[i,ts]]
    then
        begin
            matrizordenada[i,tp]:=matrizordenada[i,tp];
            matrizordenada[i,ts]:=matrizordenada[i,ts];
        end;
    if golesafavorequipo[matrizordenada[i,tp]]-
    golesencontraequipo[matrizordenada[i,tp]]<
    golesafavorequipo[matrizordenada[i,ts]]-golesencontraequipo[matrizordenada[i,ts]]
    then
        begin
            auxiliar:=matrizordenada[i,tp];
            matrizordenada[i,tp]:=matrizordenada[i,ts] ;
            matrizordenada[i,ts]:=auxiliar;
        end;
    if golesafavorequipo[matrizordenada[i,tp]]-
    golesencontraequipo[matrizordenada[i,tp]]=
    golesafavorequipo[matrizordenada[i,ts]]-golesencontraequipo[matrizordenada[i,ts]]
    then
        begin
            alea:=random(2)+1;
            if alea=1 then
                begin
                    matrizordenada[i,tp]:=matrizordenada[i,tp];
                    matrizordenada[i,ts]:=matrizordenada[i,ts];
                end;
            if alea=2 then
                begin
                    auxiliar:=matrizordenada[i,tp];
                    matrizordenada[i,tp]:=matrizordenada[i,ts] ;
                    matrizordenada[i,ts]:=auxiliar;
                end;
            end;
        end;
end;
end;
if (puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]]=
puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) and
(puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]<
puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) then
begin
    auxiliar:=matrizordenada[i,tu];
    matrizordenada[i,tu]:=matrizordenada[i,tp];
    matrizordenada[i,tp]:=auxiliar;
    if golesafavorequipo[matrizordenada[i,tp]]-
    golesencontraequipo[matrizordenada[i,tp]]>golesafavorequipo[matrizordenada[i,ts]]-
    golesencontraequipo[matrizordenada[i,ts]] then
        begin
            matrizordenada[i,tp]:=matrizordenada[i,tp];
            matrizordenada[i,ts]:=matrizordenada[i,ts];
        end;
    if golesafavorequipo[matrizordenada[i,tp]]-
    golesencontraequipo[matrizordenada[i,tp]]<golesafavorequipo[matrizordenada[i,ts]]-
    golesencontraequipo[matrizordenada[i,ts]] then
        begin
            auxiliar:=matrizordenada[i,tp];
            matrizordenada[i,tp]:=matrizordenada[i,ts] ;
            matrizordenada[i,ts]:=auxiliar;
        end;
    if golesafavorequipo[matrizordenada[i,tp]]-
    golesencontraequipo[matrizordenada[i,tp]]=golesafavorequipo[matrizordenada[i,ts]]-
    golesencontraequipo[matrizordenada[i,ts]] then
        begin
            alea:=random(2)+1;
            if alea=1 then
                begin
                    matrizordenada[i,tp]:=matrizordenada[i,tp];
                    matrizordenada[i,ts]:=matrizordenada[i,ts];
                end;
            if alea=2 then
                begin
                    auxiliar:=matrizordenada[i,tp];
                    matrizordenada[i,tp]:=matrizordenada[i,ts] ;
                    matrizordenada[i,ts]:=auxiliar;
                end;
            end;
        end;
end;
end;
end;

```

```

end;
if (puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,2],
matrizordenada[i,tu]] = puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) and
(puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]] =
puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) then
begin
if (golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]] = golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]]) and (golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]] = golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]) then
begin
alea:=random(6)+1;
if alea=1 then
begin
matrizordenada[i,tp]:=matrizordenada[i,tp];
matrizordenada[i,ts]:=matrizordenada[i,ts];
matrizordenada[i,tu]:=matrizordenada[i,tu];
end;
if alea=2 then
begin
matrizordenada[i,tp]:=matrizordenada[i,tp];
auxiliar:=matrizordenada[i,ts];
matrizordenada[i,ts]:=matrizordenada[i,tu] ;
matrizordenada[i,tu]:=auxiliar;
end;
if alea=3 then
begin
matrizordenada[i,ts]:=matrizordenada[i,ts];
auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,tu] ;
matrizordenada[i,tu]:=auxiliar;
end;
if alea=4 then
begin
matrizordenada[i,tu]:=matrizordenada[i,tu];
auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,ts] ;
matrizordenada[i,ts]:=auxiliar;
end;
if alea=5 then
begin
auxiliar:=matrizordenada[i,tp];
auxiliar2:=matrizordenada[i,ts];
matrizordenada[i,tp]:=matrizordenada[i,tu];
matrizordenada[i,ts]:=auxiliar ;
matrizordenada[i,tu]:=auxiliar2;
end;
if alea=6 then
begin
auxiliar:=matrizordenada[i,tp];
auxiliar2:=matrizordenada[i,tu];
matrizordenada[i,tp]:=matrizordenada[i,ts];
matrizordenada[i,ts]:=auxiliar2 ;
matrizordenada[i,tu]:=auxiliar;
end;
end;
if (golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]] > golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]) and (golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]] > golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]]) and (golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]] > golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]]) then
begin
matrizordenada[i,tp]:=matrizordenada[i,tp];
matrizordenada[i,ts]:=matrizordenada[i,ts];
matrizordenada[i,tu]:=matrizordenada[i,tu];
end;
if (golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]] > golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]) and (golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]] > golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]]) and (golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]] > golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]) then
begin
matrizordenada[i,tp]:=matrizordenada[i,tp];
auxiliar:=matrizordenada[i,ts];
matrizordenada[i,ts]:=matrizordenada[i,tu] ;
matrizordenada[i,tu]:=auxiliar;
end;
if (golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]] > golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]]) and (golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]] > golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]]) and (golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]] > golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]]) then
begin
auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,ts] ;
matrizordenada[i,ts]:=auxiliar;
matrizordenada[i,tu]:=matrizordenada[i,tu];
end;
end;

```

```

if (golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]>golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]]) and (golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]] > golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]]) and (golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]] > golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]]) then
begin
auxiliar:=matrizordenada[i,tp];
auxiliar2:=matrizordenada[i,tu];
matrizordenada[i,tp]:=matrizordenada[i,ts] ;
matrizordenada[i,ts]:=auxiliar2;
matrizordenada[i,tu]:=auxiliar;
end;
if (golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]]>golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]]) and (golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]] > golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]) and (golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]] > golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]) then
begin
auxiliar:=matrizordenada[i,tp];
auxiliar2:=matrizordenada[i,ts];
matrizordenada[i,tp]:=matrizordenada[i,tu] ;
matrizordenada[i,ts]:=auxiliar;
matrizordenada[i,tu]:=auxiliar2;
end;
if (golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]]>golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]]) and (golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]] > golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]) and (golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]] > golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]]) then
begin
auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,tu] ;
matrizordenada[i,ts]:=matrizordenada[i,ts];
matrizordenada[i,tu]:=auxiliar;
end;
if (golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]]>golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]) and (golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]] = golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]]) then
begin
matrizordenada[i,tp]:=matrizordenada[i,tp];
alea:=random(2)+1;
if alea=1 then
begin
matrizordenada[i,ts]:=matrizordenada[i,ts];
matrizordenada[i,tu]:=matrizordenada[i,tu];
end;
if alea=2 then
begin
auxiliar:=matrizordenada[i,ts];
matrizordenada[i,ts]:=matrizordenada[i,tu] ;
matrizordenada[i,tu]:=auxiliar;
end;
end;
if (golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]>golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]]) and (golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]] = golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]]) then
begin
auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,ts];
alea:=random(2)+1;
if alea=1 then
begin
matrizordenada[i,ts]:=auxiliar;
matrizordenada[i,tu]:=matrizordenada[i,tu];
end;
if alea=2 then
begin
auxiliar2:=matrizordenada[i,tu];
matrizordenada[i,ts]:=auxiliar2 ;
matrizordenada[i,tu]:=auxiliar;
end;
end;
if (golesafavorequipo[matrizordenada[i,tu]]-
golesencontraequipo[matrizordenada[i,tu]]>golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]]) and (golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]] = golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]]) then
begin
auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,tu];
alea:=random(2)+1;
if alea=1 then
begin
matrizordenada[i,tu]:=auxiliar;
matrizordenada[i,ts]:=matrizordenada[i,ts];
end;
if alea=2 then
begin
auxiliar2:=matrizordenada[i,ts];

```

```

                matrizordenada[i,ts]:=auxiliar ;
                matrizordenada[i,tu]:=auxiliar2;
            end;
        end;

    if (golesafavorequipo[matrizordenada[i,tp]]-
        golesencontraequipo[matrizordenada[i,tp]])=golesafavorequipo[matrizordenada[i,ts]]-
        golesencontraequipo[matrizordenada[i,ts]]) and (golesafavorequipo[matrizordenada[i,ts]]-
        golesencontraequipo[matrizordenada[i,ts]] > golesafavorequipo[matrizordenada[i,tu]]-
        golesencontraequipo[matrizordenada[i,tu]]) then
    begin
        matrizordenada[i,tu]:=matrizordenada[i,tu];
        alea:=random(2)+1;
        if alea=1 then
            begin
                matrizordenada[i,tp]:=matrizordenada[i,tp];
                matrizordenada[i,ts]:=matrizordenada[i,ts];
            end;
        if alea=2 then
            begin
                auxiliar:=matrizordenada[i,tp];
                matrizordenada[i,tp]:=matrizordenada[i,ts] ;
                matrizordenada[i,ts]:=auxiliar;
            end;
        end;
    end;

    if (golesafavorequipo[matrizordenada[i,ts]]-
        golesencontraequipo[matrizordenada[i,ts]])=golesafavorequipo[matrizordenada[i,tu]]-
        golesencontraequipo[matrizordenada[i,tu]]) and (golesafavorequipo[matrizordenada[i,tp]]-
        golesencontraequipo[matrizordenada[i,tp]] < golesafavorequipo[matrizordenada[i,tu]]-
        golesencontraequipo[matrizordenada[i,tu]]) then
    begin
        auxiliar:=matrizordenada[i,tu];
        matrizordenada[i,tu]:=matrizordenada[i,tp];
        alea:=random(2)+1;
        if alea=1 then
            begin
                matrizordenada[i,ts]:=matrizordenada[i,ts];
                matrizordenada[i,tp]:=auxiliar;
            end;
        if alea=2 then
            begin
                auxiliar2:=matrizordenada[i,ts];
                matrizordenada[i,ts]:=auxiliar ;
                matrizordenada[i,tp]:=auxiliar2;
            end;
        end;
    end;

    if (golesafavorequipo[matrizordenada[i,tu]]-
        golesencontraequipo[matrizordenada[i,tu]])=golesafavorequipo[matrizordenada[i,tp]]-
        golesencontraequipo[matrizordenada[i,tp]]) and (golesafavorequipo[matrizordenada[i,tp]]-
        golesencontraequipo[matrizordenada[i,tp]] > golesafavorequipo[matrizordenada[i,ts]]-
        golesencontraequipo[matrizordenada[i,ts]]) then
    begin
        auxiliar:=matrizordenada[i,tu];
        matrizordenada[i,tu]:=matrizordenada[i,ts];
        alea:=random(2)+1;
        if alea=1 then
            begin
                matrizordenada[i,tp]:=matrizordenada[i,tp];
                matrizordenada[i,ts]:=auxiliar;
            end;
        if alea=2 then
            begin
                auxiliar2:=matrizordenada[i,tp];
                matrizordenada[i,tp]:=auxiliar ;
                matrizordenada[i,ts]:=auxiliar2;
            end;
        end;
    end;
end;
end;
end;
end;
end;

{Por ultimo analizaremos el caso en el que hubiera cuadruple empate(algo muy improbable), en este caso, lo realizaremos con un sorteo}
if (puntosequipo[matrizordenada[i,2]]=puntosequipo[matrizordenada[i,3]]) and
(puntosequipo[matrizordenada[i,2]]=puntosequipo[matrizordenada[i,4]]) and
(puntosequipo[matrizordenada[i,2]]=puntosequipo[matrizordenada[i,1]]) then
begin
    aux[1]:=matrizordenada[i,1];
    aux[2]:=matrizordenada[i,2];
    aux[3]:=matrizordenada[i,3];
    aux[4]:=matrizordenada[i,4];
    for taux:=1 to 4 do
    begin
        alea:=random(5-taux)+1;
        alear:=aux[alea];
        matrizordenada[i,taux]:=alear;
        aux[alea]:=aux[5-taux];
    end;
end;
end;
Clasificacion[1]:=matrizordenada[i,1];
Clasificacion[2]:=matrizordenada[i,2];
Clasificacion[3]:=matrizordenada[i,3];
Clasificacion[4]:=matrizordenada[i,4];
end;

{Este proceso nos permitira recorrer los posibles resultados de un partido e ir guardando los datos.}
procedure simpartidocol(a:integer;b:integer;cc:integer;vv:integer;var gfavoreq:vector;var gcontrareq:vector;var ptoseqpo:vector;var golesentreequipos:Tmatriz32;var puntosentreequipos:Tmatriz32);

```

```

{Lo defino como procedimiento, si quisiera que me sacara un valor por pantalla

var
  golesfavora:integer;
  golescontraa:integer;

begin
  golesfavora:=cc;
  golescontraa:=vv;
  gfavoreq[a]:=gfavoreq[a]+golesfavora;
  gcontrareq[a]:=gcontrareq[a]+golescontraa;
  gfavoreq[b]:=gfavoreq[b]+golescontraa;
  gcontrareq[b]:=gcontrareq[b]+golesfavora;

  if (golesfavora>golescontraa) then
    begin
      ptoseqpo[a]:= ptoseqpo[a]+3;
      puntosentreequipos[a,b]:=puntosentreequipos[a,b]+3;
    end;

  if (golesfavora<golescontraa) then
    begin
      ptoseqpo[b]:= ptoseqpo[b]+3;
      puntosentreequipos[b,a]:=puntosentreequipos[b,a]+3;
    end;

  if(golesfavora=golescontraa) then
    begin
      ptoseqpo[a]:= ptoseqpo[a]+1;
      ptoseqpo[b]:= ptoseqpo[b]+1;
      puntosentreequipos[a,b]:=puntosentreequipos[a,b]+1;
      puntosentreequipos[b,a]:=puntosentreequipos[b,a]+1;
    end;
  golesentreequipos[a,b]:=golesfavora+golesentreequipos[a,b];
  golesentreequipos[b,a]:=golescontraa+golesentreequipos[b,a];

end;
{Nos saca si en el grupo hay un partido de colisión o no}
function colision(ir:integer;Home:jornada;Away:jornada;Eq:Tmatriz;var gfavoreq:vector;var gcontrareq:
vector;var ptoseqpo:vector;var golesentreequipos:Tmatriz32;var puntosentreequipos:Tmatriz32;var
clasif1:integer;var clasif2:integer;var clasif3:integer;var clasif4:integer):integer;
var
  ii,jj,k,l,kk,ll,x,xx,escol,ps,ss,ps1,ss1:integer;
  gfavoreqss,gfavoreqss1,gfavoreqssaux:vector;
  gcontrareqss,gcontrareqss1,gcontrareqssaux:vector;
  ptoseqposs,ptoseqposs1,ptoseqpossaux:vector;
  golesentreequiposss,golesentreequiposss1,golesentreequiposssaux:Tmatriz32;
  puntosentreequiposss,puntosentreequiposss1,puntosentreequiposssaux:Tmatriz32;
  vecl,vecl1:vector4;
begin
  if ((clasif3=Away[ir,6,2]) and (clasif4=Home[ir,6,2])) or ((clasif3=Home[ir,6,2]) and (clasif4=Away[ir,6,2])) then
    begin
      xx:=0;
{Suponemos que juegan 1 contra 2 y 3 contra 4, lo primero que hace es que habra colision si juegan el tercero contra el cuarto}
  x:=0;
{Recorremos los resultados de este partido para ver si tendrían opciones, inicializamos con xx=0, si tienen, empezamos el programa
, si no pueden pasar ninguno de los 2, no habrá colision}
  k:=0;
  while(k<10) and (xx=0) do
    begin
      l:=0;
      while(l<10) and (xx=0) do
        begin
          gfavoreqss1:=gfavoreq;
          gcontrareqss1:=gcontrareq;
          ptoseqposs1:=ptoseqpo;
          golesentreequiposss1:=golesentreequipos;
          puntosentreequiposss1:=puntosentreequipos;
          simpartidocol(Home[ir,6,2],Away[ir,6,2],k,l,gfavoreqss1,gcontrareqss1,ptoseqposs1,
          golesentreequiposss1,puntosentreequiposss1);
          vecl:=Clasificacion(ir,ptoseqposs1,gfavoreqss1,gcontrareqss1,golesentreequiposss1,
          puntosentreequiposss1,Eq);
          ps:=vecl[1];
          ss:=vecl[2];
          if (clasif3=ps) or (clasif3=ss) or (clasif4=ps) or (clasif4=ss) then
            {Si existe algun resultado que hace que el que iba tercero o cuarto puede quedar
            entre los 2 primeros, empezamos}
            begin
              xx:=1;
              x:=0;
              ii:=0;
              while (ii<10) and (x=0) do
                begin
                  jj:=0;
                  while(jj<10) and (x=0) do
                    begin
                      gfavoreqss:=gfavoreq;
                      gcontrareqss:=gcontrareq;
                      ptoseqposs:=ptoseqpo;
                      golesentreequiposss:=golesentreequipos;
                      puntosentreequiposss:=puntosentreequipos;
                      simpartidocol(Home[ir,6,1],Away[ir,6,1],ii,jj,gfavoreqss,gcontrareqss,ptoseqposs,
                      golesentreequiposss,puntosentreequiposss);
{Lo uso pa comporbar} vecl1:=Clasificacion(ir,ptoseqposs,gfavoreqss,gcontrareqss,golesentreequiposss,
puntosentreequiposss,Eq);
{Simulamos el segundo}
                      ps1:=vecl1[1];
                    end;
                  end;
                end;
              end;
            end;
          end;
        end;
      end;
    end;
  end;

```

```

ss1:=vecl1[2];
gfavoreqssaux:=gfavoreqss;
gcontrareqssaux:=gcontrareqss;
ptoseqposaux:=ptoseqpos;
golesentreequiposaux:=golesentreequipos;
puntosentreequiposaux:=puntosentreequipos;
escol:=1;
kk:=0;
{Recorremos el segundo}
while(kk<10) and (escol=1) do
begin
ll:=0;
while(ll<10) and (escol=1) do
begin
gfavoreqss:=gfavoreqssaux;
gcontrareqss:=gcontrareqssaux;
ptoseqpos:=ptoseqposaux;
golesentreequipos:=golesentreequiposaux;
puntosentreequipos:=puntosentreequiposaux;
simpartidocol(Home[ir,6,2],Away[ir,6,2],kk,ll,gfavoreqss,gcontrareqss,ptoseqpos,
golesentreequipos,puntosentreequipos);
vecl1:=Clasificacion(ir,ptoseqpos,gfavoreqss,gcontrareqss,golesentreequipos,
puntosentreequipos,Eq);
{Simulamos el segundo}
ps1:=vecl1[1];
ss1:=vecl1[2];
if ((clasif1=ps1) and (clasif2=ss1)) then escol:=1
else if (clasif1=ss1) and (clasif2=ps1) then escol:=1
else escol:=0;
ll:=ll+1;
end;
kk:=kk+1;
end;
if escol=1 then x:=1;
jj:=jj+1;
end;
ii:=ii+1;
end;
end
else if (ptoseqpo[Clasif1]=ptoseqpo[Clasif2]) and (ptoseqpo[Clasif1]=ptoseqpo[Clasif3])
and (ptoseqpo[Clasif1]=ptoseqpo[Clasif4]) then
begin
xx:=0; {Esto lo hacemos para decir que si van empatados todos a puntos, no habra colision}
x:=0;
end
else
begin
xx:=0;
x:=0;
end;
l:=l+1;
end;
k:=k+1;
end;
colision:=x;
end
{También se podría dar el caso que fueran tercero y cuarto los otros dos equipos, por tanto lo tenemos que mirar tambien}
else if ((clasif3=Away[ir,6,1]) and (clasif4=Home[ir,6,1])) or ((clasif3=Home[ir,6,1]) and (clasif4=Away[ir,6,1])) then
begin
xx:=0;
{Suponemos que juegan 1 contra 2 y 3 contra 4, lo primero que hace es que habra colision si juegan el tercero contra el cuarto}
x:=0;
{Recorremos este partido para ver si tendrían opciones, inicializamos con xx=0, si tienen, empezamos el programa}
k:=0;
while(k<10) and (xx=0) do
begin
l:=0;
while(l<10) and (xx=0) do
begin
gfavoreqss1:=gfavoreq;
gcontrareqss1:=gcontrareq;
ptoseqpos1:=ptoseqpo;
golesentreequipos1:=golesentreequipos;
puntosentreequipos1:=puntosentreequipos;
simpartidocol(Home[ir,6,1],Away[ir,6,1],k,l,gfavoreqss1,gcontrareqss1,ptoseqpos1,
golesentreequipos1,puntosentreequipos1);
vecl1:=Clasificacion(ir,ptoseqpos1,gfavoreqss1,gcontrareqss1,golesentreequipos1,
puntosentreequipos1,Eq);
ps:=vecl1[1];
ss:=vecl1[2];
if (clasif3=ps) or (clasif3=ss) or (clasif4=ps) or (clasif4=ss) then
begin
xx:=1;
x:=0;
ii:=0;
while (ii<10) and (x=0) do
begin
jj:=0;
while(jj<10) and (x=0) do
begin
gfavoreqss:=gfavoreq; {Recorremos el primer partido}
gcontrareqss:=gcontrareq;
ptoseqpos:=ptoseqpo;
golesentreequipos:=golesentreequipos;
puntosentreequipos:=puntosentreequipos;
simpartidocol(Home[ir,6,2],Away[ir,6,2],ii,jj,gfavoreqss,gcontrareqss,ptoseqpos,
golesentreequipos,puntosentreequipos);
vecl1:=Clasificacion(ir,ptoseqpos,gfavoreqss,gcontrareqss,golesentreequipos,
puntosentreequipos,Eq);

```

```

{Simulamos el segundo}
    ps1:=vecl1[1];
    ss1:=vecl1[2];
    gfavoreqssaux:=gfavoreqss;
    gcontrareqssaux:=gcontrareqss;
    ptoseqpossaux:=ptoseqposs;
    golesentreequipossaux:=golesentreequiposs;
    puntosentreequipossaux:=puntosentreequiposs;
    escol:=1;
    kk:=0;
    while(kk<10) and (escol=1) do
        begin
            ll:=0;
            while(ll<10) and (escol=1) do
                begin
                    gfavoreqss:=gfavoreqssaux;
                    gcontrareqss:=gcontrareqssaux;
                    ptoseqposs:=ptoseqpossaux;
                    golesentreequiposs:=golesentreequipossaux;
                    puntosentreequiposs:=puntosentreequipossaux;
                    simpartidocol(Home[ir,6,1],Away[ir,6,1],kk,ll,gfavoreqss,gcontrareqss,ptoseqposs,
                    golesentreequiposs,puntosentreequiposs);
                    vecl1:=Clasificacion(ir,ptoseqposs,gfavoreqss,gcontrareqss,golesentreequiposs,
                    puntosentreequiposs,Eq);
                    {Simulamos el segundo}
                    ps1:=vecl1[1];
                    ss1:=vecl1[2];
                    if ((clasif1=ps1) and (clasif2=ss1)) then escol:=1
                    else if (clasif1=ss1) and (clasif2=ps1) then escol:=1
                    else escol:=0;
                    ll:=ll+1;
                end;
                kk:=kk+1;
            end;
            if escol=1 then x:=1;
            jj:=jj+1;
        end;
        ii:=ii+1;
    end;
end;
else if (ptoseqpo[Clasif1]=ptoseqpo[Clasif2]) and (ptoseqpo[Clasif1]=ptoseqpo[Clasif3]) and (ptoseqpo[Clasif1]=
ptoseqpo[Clasif4]) then
    begin
        xx:=0; {Esto lo hacemos para decir que si van empatados todos a puntos, no habra colision}
        x:=0;
    end
    else
        begin
            xx:=0;
            x:=0;
        end;
        l:=l+1;
    end;
    k:=k+1;
end;
colision:=x;
end
else
    begin
        x:=0;
        colision:=x;
    end;
end;
end;

function intrascendente1(ir:integer;Home:jornada;Away:jornada;Eq:Tmatriz;var gfavoreq:vector;var
gcontrareq: vector;var ptoseqpo:vector;var golesentreequipos:Tmatriz32;var
puntosentreequipos:Tmatriz32;var clasif1:integer;var clasif2:integer;var clasif3:integer;var
clasif4:integer):integer;
var
    ii,jj,k,l,x,esint,ps1,ss1,ts1,cs1:integer;
    gfavoreqss,gfavoreqssaux:vector;
    gcontrareqss,gcontrareqssaux:vector;
    ptoseqposs,ptoseqpossaux:vector;
    golesentreequiposs,golesentreequipossaux:Tmatriz32;
    puntosentreequiposs,puntosentreequipossaux:Tmatriz32;
    vecl1:vector4;
begin
    if (ptoseqpo[Clasif1]=ptoseqpo[Clasif2]) and (ptoseqpo[Clasif1]=ptoseqpo[Clasif3]) and (ptoseqpo[Clasif1]=ptoseqpo[Clasif4])
    then
        begin
            {Esto lo hacemos para decir que si van empatados todos a puntos, no habra colision}
            x:=0;
            intrascendente1:=x;
        end
    else
        begin
            x:=1;
            ii:=0;
            while (ii<10) and (x=1) do
                begin
                    jj:=0;
                    while(jj<10) and (x=1) do
                        begin
                            gfavoreqss:=gfavoreq;
                            gcontrareqss:=gcontrareq;
                            ptoseqposs:=ptoseqpo;
                            golesentreequiposs:=golesentreequipos;
                            puntosentreequiposs:=puntosentreequipos;
                            simpartidocol(Home[ir,6,2],Away[ir,6,2],ii,jj,gfavoreqss,gcontrareqss,ptoseqposs,golesentreequiposs,

```

```

puntosentreequipos);
vecl1:=Clasificacion(ir,ptoseqpos,gfavoreqss,gcontrareqss,golesentreequipos,
puntosentreequipos,Eq);
{Simulamos el segundo}
ps1:=vecl1[1];
ss1:=vecl1[2];
gfavoreqssaux:=gfavoreqss;
gcontrareqssaux:=gcontrareqss;
ptoseqposaux:=ptoseqpos;
golesentreequiposaux:=golesentreequipos;
puntosentreequiposaux:=puntosentreequipos;
esint:=1;
k:=0;
while(k<10) and (esint=1) do
begin
l:=0;
while(l<10) and (esint=1) do
begin
gfavoreqss:=gfavoreqssaux;
gcontrareqss:=gcontrareqssaux;
ptoseqpos:=ptoseqposaux;
golesentreequipos:=golesentreequiposaux;
puntosentreequipos:=puntosentreequiposaux;
simpartidocol(Home[ir,6,1],Away[ir,6,1],k,l,gfavoreqss,gcontrareqss,
ptoseqpos,
golesentreequipos,puntosentreequipos);
vecl1:=Clasificacion(ir,ptoseqpos,gfavoreqss,gcontrareqss,
golesentreequipos,puntosentreequipos,Eq);
{Simulamos el segundo}
ps1:=vecl1[1];
ss1:=vecl1[2];
ts1:=vecl1[3];
cs1:=vecl1[4];
{Vemos si el partido del equipo 2 contra el equipo 3 es intrascendente, para ello analizamos varios casos,
según como vaya clasificados antes de empezar la última jornada}
if ((Home[ir,6,2]=clasif1) and (Away[ir,6,2]=clasif2)) or ((Home[ir,6,2]=clasif2)
and (Away[ir,6,2]=clasif1)) then
begin
if (ts1=Home[ir,6,2]) or (ts1=Away[ir,6,2]) or (cs1=Home[ir,6,2]) or (cs1=Away[ir,6,2])
then esint:=0
else esint:=1;
end
else if (clasif1=Home[ir,6,2]) and (clasif3=Away[ir,6,2]) then {1-3}
begin
if (Home[ir,6,2]=ts1) or (Home[ir,6,2]=cs1) or (Away[ir,6,2]=ss1) or (Away[ir,6,2]=ps1)
then esint:=0
else esint:=1;
end
else if (clasif1=Away[ir,6,2]) and (clasif3=Home[ir,6,2]) then
begin
if (Home[ir,6,2]=ss1) or (Home[ir,6,2]=ps1) or (Away[ir,6,2]=ts1) or (Away[ir,6,2]=cs1)
then esint:=0
else esint:=1;
end
else if (clasif1=Home[ir,6,2]) and (clasif4=Away[ir,6,2]) then {1-4}
begin
if (Home[ir,6,2]=ts1) or (Home[ir,6,2]=cs1) or (Away[ir,6,2]=ss1) or (Away[ir,6,2]=ps1)
then esint:=0
else esint:=1;
end
else if (clasif1=Away[ir,6,2]) and (clasif4=Home[ir,6,2]) then
begin
if (Home[ir,6,2]=ss1) or (Home[ir,6,2]=ps1) or (Away[ir,6,2]=ts1) or (Away[ir,6,2]=cs1)
then esint:=0
else esint:=1;
end
else if (clasif2=Home[ir,6,2]) and (clasif3=Away[ir,6,2]) then {2-3}
begin
if (Home[ir,6,2]=ts1) or (Home[ir,6,2]=cs1) or (Away[ir,6,2]=ss1) or (Away[ir,6,2]=ps1)
then esint:=0
else esint:=1;
end
else if (clasif2=Away[ir,6,2]) and (clasif3=Home[ir,6,2]) then
begin
if (Home[ir,6,2]=ss1) or (Home[ir,6,2]=ps1) or (Away[ir,6,2]=ts1) or (Away[ir,6,2]=cs1)
then esint:=0
else esint:=1;
end
else if (clasif2=Home[ir,6,2]) and (clasif4=Away[ir,6,2]) then {2-4}
begin
if (Home[ir,6,2]=ts1) or (Home[ir,6,2]=cs1) or (Away[ir,6,2]=ss1) or (Away[ir,6,2]=ps1)
then esint:=0
else esint:=1;
end
else if (clasif2=Away[ir,6,2]) and (clasif4=Home[ir,6,2]) then
begin
if (Home[ir,6,2]=ss1) or (Home[ir,6,2]=ps1) or (Away[ir,6,2]=ts1) or (Away[ir,6,2]=cs1)
then esint:=0
else esint:=1;
end
else if ((clasif3=Home[ir,6,2]) and (clasif4=Away[ir,6,2])) or
((clasif3=Away[ir,6,2]) and (clasif4=Home[ir,6,2])) then
{3-4}
begin
if (Home[ir,6,2]=ps1) or (Home[ir,6,2]=ss1) or (Away[ir,6,2]=ss1) or (Away[ir,6,2]=ps1)
then esint:=0
else esint:=1;
end
else esint:=1;

```



```

                l:=l+1;
            end;
            k:=k+1;
        end;
        if esint=0 then x:=0;
            jj:=jj+1;
        end;
        ii:=ii+1;
    end;
    intrascendente1:=x;
end;

end;
function intrascendente2(ir:integer;Home:jornada;Away:jornada;Eq:Tmatriz;var gfavoreq:vector;var
gcontrareq: vector;var ptoseqpo:vector;var golesentreequipos:Tmatriz32;var
puntosentreequipos:Tmatriz32;var clasif1:integer;var clasif2:integer;var clasif3:integer;var
clasif4:integer):integer;
var
    ii,jj,k,l,x,esint,ps1,ss1,ts1,cs1:integer;
    gfavoreqss,gfavoreqssaux:vector;
    gcontrareqss,gcontrareqssaux:vector;
    ptoseqposss,ptoseqposssaux:vector;
    golesentreequiposss,golesentreequiposssaux:Tmatriz32;
    puntosentreequiposss,puntosentreequiposssaux:Tmatriz32;
    vec11:vector4;
begin
    if (ptoseqpo[Clasif1]=ptoseqpo[Clasif2]) and (ptoseqpo[Clasif1]=ptoseqpo[Clasif3]) and (ptoseqpo[Clasif1]=ptoseqpo[Clasif4]) then
        begin
            {Esto lo hacemos para decir que si van empatados todos a puntos, no habra colision}
            x:=0;
            intrascendente2:=x;
        end
    else
        begin
            x:=1;
            ii:=0;
            while (ii<10) and (x=1) do
                begin
                    jj:=0;
                    while(jj<10) and (x=1) do
                        begin
                            gfavoreqss:=gfavoreq;
                            gcontrareqss:=gcontrareq;
                            ptoseqposss:=ptoseqpo;
                            golesentreequiposss:=golesentreequipos;
                            puntosentreequiposss:=puntosentreequipos;
                            simpartidocol(Home[ir,6,1],Away[ir,6,1],ii,jj,gfavoreqss,gcontrareqss,
                            ptoseqposss,golesentreequiposss,puntosentreequiposss);
                            vec11:=Clasificacion(ir,ptoseqposss,gfavoreqss,gcontrareqss,golesentreequiposss,puntosentreequiposss,Eq);
                            {Simulamos el segundo}
                            ps1:=vec11[1];
                            ss1:=vec11[2];
                            gfavoreqssaux:=gfavoreqss;
                            gcontrareqssaux:=gcontrareqss;
                            ptoseqposssaux:=ptoseqposss;
                            golesentreequiposssaux:=golesentreequiposss;
                            puntosentreequiposssaux:=puntosentreequiposss;
                            esint:=1;
                            k:=0;
                            while(k<10) and (esint=1) do
                                begin
                                    l:=0;
                                    while(l<10) and (esint=1) do
                                        begin
                                            gfavoreqss:=gfavoreqssaux;
                                            gcontrareqss:=gcontrareqssaux;
                                            ptoseqposss:=ptoseqposssaux;
                                            golesentreequiposss:=golesentreequiposssaux;
                                            puntosentreequiposss:=puntosentreequiposssaux;
                                            simpartidocol(Home[ir,6,2],Away[ir,6,2],k,l,gfavoreqss,gcontrareqss,
                                            ptoseqposss,golesentreequiposss,puntosentreequiposss);
                                            vec11:=Clasificacion(ir,ptoseqposss,gfavoreqss,gcontrareqss,golesentreequiposss,puntosentreequiposss,Eq);
                                            {Simulamos el segundo}
                                            ps1:=vec11[1];
                                            ss1:=vec11[2];
                                            ts1:=vec11[3];
                                            cs1:=vec11[4];

```

{Vemos si el partido del equipo 1 contra el equipo 4 es intrascendete, para ello analizamos varios casos, según como vaya claficados antes de empezar la ultima jornada}

```

                            if ((Home[ir,6,1]=clasif1) and (Away[ir,6,1]=clasif2)) or ((Home[ir,6,1]=clasif2) and
                            (Away[ir,6,1]=clasif1)) then {1-2}
                                begin
                                    if (ts1=Home[ir,6,1]) or (ts1=Away[ir,6,1]) or (cs1=Home[ir,6,1]) or (cs1=Away[ir,6,1])
                                    then esint:=0
                                    else esint:=1;
                                    end
                                else if (clasif1=Home[ir,6,1]) and (clasif3=Away[ir,6,1]) then {1-3}
                                    begin
                                        if (Home[ir,6,1]=ts1) or (Home[ir,6,1]=cs1) or (Away[ir,6,1]=ss1) or (Away[ir,6,1]=ps1)
                                        then esint:=0
                                        else esint:=1;
                                        end
                                    else if (clasif1=Away[ir,6,1]) and (clasif3=Home[ir,6,1]) then
                                        begin
                                            if (Home[ir,6,1]=ss1) or (Home[ir,6,1]=ps1) or (Away[ir,6,1]=ts1) or (Away[ir,6,1]=cs1)
                                            then esint:=0
                                            else esint:=1;
                                            end
                                        else if (clasif1=Home[ir,6,1]) and (clasif4=Away[ir,6,1]) then {1-4}
                                            begin

```

```

        if (Home[ir,6,1]=ts1) or (Home[ir,6,1]=cs1) or (Away[ir,6,1]=ss1) or (Away[ir,6,1]=ps1)
        then esint:=0
        else esint:=1;
    end
else if (clasif1=Away[ir,6,1]) and (clasif4=Home[ir,6,1]) then
    begin
        if (Home[ir,6,1]=ss1) or (Home[ir,6,1]=ps1) or (Away[ir,6,1]=ts1) or (Away[ir,6,1]=cs1)
        then esint:=0
        else esint:=1;
    end
else if (clasif2=Home[ir,6,1]) and (clasif3=Away[ir,6,1]) then          {2-3}
    begin
        if (Home[ir,6,1]=ts1) or (Home[ir,6,1]=cs1) or (Away[ir,6,1]=ss1) or (Away[ir,6,1]=ps1)
        then esint:=0
        else esint:=1;
    end
else if (clasif2=Away[ir,6,1]) and (clasif3=Home[ir,6,1]) then
    begin
        if (Home[ir,6,1]=ss1) or (Home[ir,6,1]=ps1) or (Away[ir,6,1]=ts1) or (Away[ir,6,1]=cs1)
        then esint:=0
        else esint:=1;
    end
else if (clasif2=Home[ir,6,1]) and (clasif4=Away[ir,6,1]) then          {2-4}
    begin
        if (Home[ir,6,1]=ts1) or (Home[ir,6,1]=cs1) or (Away[ir,6,1]=ss1) or (Away[ir,6,1]=ps1)
        then esint:=0
        else esint:=1;
    end
else if (clasif2=Away[ir,6,1]) and (clasif4=Home[ir,6,1]) then
    begin
        if (Home[ir,6,1]=ss1) or (Home[ir,6,1]=ps1) or (Away[ir,6,1]=ts1) or (Away[ir,6,1]=cs1)
        then esint:=0
        else esint:=1;
    end
else if (clasif3=Home[ir,6,1]) and (clasif4=Away[ir,6,1]) or (clasif3=Away[ir,6,1])
and (clasif4=Home[ir,6,1]) then          {3-4}
    begin
        if (Home[ir,6,1]=ps1) or (Home[ir,6,1]=ss1) or (Away[ir,6,1]=ss1) or (Away[ir,6,1]=ps1)
        then esint:=0
        else esint:=1;
    end
else esint:=1;
    l:=l+1;
end;
k:=k+1;
end;
if esint=0 then x:=0;
    jj:=jj+1;
end;
ii:=ii+1;
end;
intrascendente2:=x;
end;
end;
function partidosintrascendentes(ir:integer;Home:jornada;Away:jornada;Eq:Tmatriz;var gfavoreq:vector;var
gcontrareq: vector;var ptoseqpo:vector;var golesentreequipos:Tmatriz32;var
puntosentreequipos:Tmatriz32;var clasif1:integer;var clasif2:integer;var clasif3:integer;var
clasif4:integer):integer;
var parint1,parint2,parinttotales:integer;
begin
    parint1:=intrascendente1(ir,Home,Away,Eq,gfavoreq,gcontrareq,ptoseqpo,golesentreequipos,puntosentreequipos,
    clasif1,clasif2,clasif3,clasif4);
    parint2:=intrascendente2(ir,Home,Away,Eq,gfavoreq,gcontrareq,ptoseqpo,golesentreequipos,puntosentreequipos,
    clasif1,clasif2,clasif3,clasif4);
    parinttotales:=parint1+parint2;
    partidosintrascendentes:=parinttotales;
end;
{#####PROGRAMA PRINCIPAL}
begin
    randomize();
    write('Introduce el tipo de Sorteo, Fifa[1] Guyon[2]: ');
    read(sorteo);
    while (sorteo<>1) and (sorteo<>2) do
        begin
            write('Introduzca bien el numero: ');
            read(sorteo);
        end ;
    write('Introduce , IdaVuelta[1] PartidoUnico[2]: ');
    read(partidos);
    while (partidos<>1) and (partidos<>2) do
        begin
            write('Introduzca bien el numero: ');
            read(partidos);
        end ;
    if partidos=1 then
        begin
            write('Introduce un orden de la jornada que este entre 1 y 6: ');
            read(numero);
            while (numero<>1) and (numero<>2) and (numero<>3) and (numero<>4) and (numero<>5) and (numero<>6) do
                begin
                    write('Introduzca bien el numero: ');
                    read(numero);
                end ;
            end;
        if partidos=2 then
            begin
                write('Introduce un orden de la jornada que este entre 1 y 3: ');
                read(numero1);
                while (numero1<>1) and (numero1<>2) and (numero1<>3) do

```

```

begin
write('Introduzca bien el numero: ');
read(numero1);
end ;
end;
write('Introduce el tipo de orden de los equipos, Ranking[1] OrdenSimulacion[2]: ');
read(orden);
while (orden<>1) and (orden<>2) do
begin
write('Introduzca bien el numero: ');
read(orden);
end ;
if orden =1 then
begin
assign(q,'DEFINITIVOLandasA132PoissonFiltrado.txt');
assign(f,'DEFINITIVOLandasA32PoissonFiltrado.txt');
assign(u,'DEFINITIVOLandasA32UNICOPoissonFiltrado.txt');
assign(Resul,'Resultados.txt');
end;
if orden =2 then
begin
assign(q,'DEFINITIVOSIMULACIONLandasA132PoissonFiltrado.txt'); {Este documento cojo las columnas y
voy poniendo una debajo de otra}
assign(f,'DEFINITIVOSIMULACIONLandasA32PoissonFiltrado.txt'); {Este documento lo hice
cogiendo filas y pasandolas a columnas y poniendo una debajo de otra}
assign(u,'DEFINITIVOSIMULACIONLandasA32UNICOPoissonFiltrado.txt');
assign(Resul,'Resultados.txt');
end;
reset(q);
reset(f);
reset(u);
reset(Resul);
rewrite(Resul);
posicion1:=1;
posicion:=1;
posicionu:=1;
while not Eof(q) do {Lectura de los landas, sacamos la matriz de R, los pongo todos en una columna en un bloc de notas y
digo que me lea todo el fichero y me los guarde en un array Misdatos y posteriormente,
ese array lo convertimos en una matriz}
begin
readLn(q,a);
misdatos1[posicion1]:=a;
posicion1:=posicion1+1;
end;
while not Eof(f) do
begin
readLn(f,a);
misdatos[posicion]:=a;
posicion:=posicion+1;
end;
while not Eof(u) do
begin
readLn(u,a);
misdatosu[posicionu]:=a;
posicionu:=posicionu+1;
end;
for i:=1 to 32 do
begin
for j:=1 to 32 do matriz1[i,j]:=misdatos1[32*(i-1)+j];
end;
for i:=1 to 32 do
begin
for j:=1 to 32 do matriz[i,j]:=misdatos[32*(i-1)+j];
end;
for i:=1 to 32 do
begin
for j:=1 to 32 do matrizuni[i,j]:=misdatosu[32*(i-1)+j];
end;
{SORTEO-----}
for p:=1 to 32 do
begin
vclas[p]:=0;
end;
intentos:=0;
intentos1:=0;
for s:=1 to 1000 do
begin
if sorteo=1 then
begin
for t1:=1 to 8 do lista1[t1]:=t1;
for t2:=1 to 8 do lista2[t2]:=t2+8;
for t3:=1 to 8 do lista3[t3]:=t3+16;
for t4:=1 to 8 do lista4[t4]:=t4+24;
for t1:=1 to 8 do
begin
r1:=random(8-t1)+1;
rr1:=lista1[r1];
g[t1]:=rr1;
lista1[r1]:=lista1[9-t1];
end;
for t2:=1 to 8 do
begin
r2:=random(9-t2)+1;
rr2:=lista2[r2];
g[t2+8]:=rr2;
lista2[r2]:=lista2[9-t2];
end;
for t3:=1 to 8 do
begin

```

```

        r3:=random(9-t3)+1;
        rr3:=lista3[r3];
        g[t3+16]:=rr3;
        lista3[r3]:=lista3[9-t3];
    end;
for t4:=1 to 8 do
    begin
        r4:=random(9-t4)+1;
        rr4:=lista4[r4];
        g[t4+24]:=rr4;
        lista4[r4]:=lista4[9-t4];
    end;
end;
if sorteo=2 then
    begin
        for t1:=1 to 4 do lista1[t1]:=t1;
for t2:=1 to 4 do lista2[t2]:=t2+12;
for t3:=1 to 4 do lista3[t3]:=t3+16;
for t4:=1 to 4 do lista4[t4]:=t4+28;
        for t1:=1 to 4 do
            begin
                r1:=random(5-t1)+1;
                rr1:=lista1[r1];
                g[t1]:=rr1;
                lista1[r1]:=lista1[5-t1];
            end;
for t2:=1 to 4 do
            begin
                r2:=random(5-t2)+1;
                rr2:=lista2[r2];
                g[t2+8]:=rr2;
                lista2[r2]:=lista2[5-t2];
            end;
for t3:=1 to 4 do
            begin
                r3:=random(5-t3)+1;
                rr3:=lista3[r3];
                g[t3+16]:=rr3;
                lista3[r3]:=lista3[5-t3];
            end;
for t4:=1 to 4 do
            begin
                r4:=random(5-t4)+1;
                rr4:=lista4[r4];
                g[t4+24]:=rr4;
                lista4[r4]:=lista4[5-t4];
            end;
for t5:=1 to 4 do lista5[t5]:=t5+4;
for t6:=1 to 4 do lista6[t6]:=t6+8;
for t7:=1 to 4 do lista7[t7]:=t7+20;
for t8:=1 to 4 do lista8[t8]:=t8+24;
for t5:=1 to 4 do
            begin
                r5:=random(5-t5)+1;
                rr5:=lista5[r5];
                g[t5+4]:=rr5;
                lista5[r5]:=lista5[5-t5];
            end;
for t6:=1 to 4 do
            begin
                r6:=random(5-t6)+1;
                rr6:=lista6[r6];
                g[t6+12]:=rr6;
                lista6[r6]:=lista6[5-t6];
            end;
for t7:=1 to 4 do
            begin
                r7:=random(5-t7)+1;
                rr7:=lista7[r7];
                g[t7+20]:=rr7;
                lista7[r7]:=lista7[5-t7];
            end;
for t8:=1 to 4 do
            begin
                r8:=random(5-t8)+1;
                rr8:=lista8[r8];
                g[t8+28]:=rr8;
                lista8[r8]:=lista8[5-t8];
            end;
end;
{Hasta aqui el sorteo-----}
{Defino a los equipos como equipo[i,j] con i el grupo y j el bombo al que pertenecen}
for i:=1 to 8 do
    for j:=1 to 4 do equipo[i,j]:=g[8*(j-1)+i];
{Planteamos jugar las jornadas}
{ inicializar puntos de cada equipo asi como goles a favor y en contra}
for r:=1 to 32 do
    begin
        puntosequipo[r]:=0;
        golesafavorequipo[r]:=0;
        golesencontraequipo[r]:=0;
    end;
for i:=1 to 32 do
    begin
        for j:=1 to 32 do golesenfrentamientos[i,j]:=0;
        end;
for i:=1 to 32 do
    begin
        for j:=1 to 32 do puntosenfrentamientos[i,j]:=0;
    end;
end;

```



```

end
else if numero=4 then
begin
casa[i,k,m]:=equipo[i,m];
fuera[i,k,m]:=equipo[i,4-(m-1)];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz[equipo[i,m],equipo[i,4-(m-1)]]
,matriz[equipo[i,4-(m-1)],equipo[i,m]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero=5 then
begin
casa[i,k,m]:=equipo[i,1+(m-1)];
fuera[i,k,m]:=equipo[i,2+m];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz[equipo[i,1+(m-1)],equipo[i,
2+m]] ,matriz[equipo[i,2+m],equipo[i,1+
(m-1)]] ,golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero=6 then
begin
casa[i,k,m]:=equipo[i,1+2*(m-1)];
fuera[i,k,m]:=equipo[i,2+m];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz[equipo[i,1+2*(m-1)],equipo[i,
2+m]] ,matriz[equipo[i,2+m],equipo[i,
1+2*(m-1)]] ,golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end;
end;
end;
{JORNADA3}
3 : begin
for i:=1 to 8 do
for m:=1 to 2 do
begin
if numero=1 then          {14-23}
begin
casa[i,k,m]:=equipo[i,1+(m-1)];
fuera[i,k,m]:=equipo[i,2+m];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz[equipo[i,1+(m-1)],equipo[i,
2+m]] ,matriz[equipo[i,2+m],equipo[i,1+
(m-1)]] ,golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero=2 then
begin
casa[i,k,m]:=equipo[i,1+(m-1)];
fuera[i,k,m]:=equipo[i,2+m];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz[equipo[i,1+(m-1)],equipo[i,
2+m]] ,matriz[equipo[i,2+m],equipo[i,1+
(m-1)]] ,golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero=3 then
begin
casa[i,k,m]:=equipo[i,m];
fuera[i,k,m]:=equipo[i,4-(m-1)];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz[equipo[i,m],equipo[i,4-(m-1)]]
,matriz[equipo[i,4-(m-1)],equipo[i,m]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero=4 then
begin
casa[i,k,m]:=equipo[i,4-(m-1)];
fuera[i,k,m]:=equipo[i,2-(m-1)];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz1[equipo[i,2-(m-1)],equipo[i,4-
(m-1)]] ,matriz1[equipo[i,4-(m-1)],equipo[i,2-
(m-1)]] ,golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero=5 then
begin
casa[i,k,m]:=equipo[i,2+m];
fuera[i,k,m]:=equipo[i,2-(m-1)];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz1[equipo[i,2-(m-1)],equipo[i,
2+m]] ,matriz1[equipo[i,2+m],equipo[i,2-
(m-1)]] ,golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
{Aquí ponemos la matriz1, juegan 3-2;4-1, cuando señalo la matriz1 que pongo primero el visitante, es
porque la matriz 1 creada, la casilla[1,2], dice los goles del equipo 2 como local al 1 como visitante en
cambio con la matriz[] la casilla [1,2], nos indica los goles del equipo 1 como local al 2 como visitante}
end
else if numero=6 then
begin
casa[i,k,m]:=equipo[i,4-(m-1)];
fuera[i,k,m]:=equipo[i,2-(m-1)];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz1[equipo[i,2-(m-1)],equipo[i,4
m-1]] ,matriz1[equipo[i,4-(m-1)],equipo[i,2-
(m-1)]] ,golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end;
end;
end;
{JORNADA4}
4 : begin
for i:=1 to 8 do
for m:=1 to 2 do
begin
if numero=1 then          {14-23}
begin

```

```

casa[i,k,m]:=equipo[i,4-(m-1)];
fuera[i,k,m]:=equipo[i,2-(m-1)];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz1[equipo[i,2-(m-1)],equipo[i,4-
(m-1)]],matriz1[equipo[i,4-(m-1)],equipo[i,2-
(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero=2 then
begin
casa[i,k,m]:=equipo[i,4-(m-1)];
fuera[i,k,m]:=equipo[i,2-(m-1)];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz1[equipo[i,2-(m-1)],equipo[i,4-
(m-1)]],matriz1[equipo[i,4-(m-1)],equipo[i,2-
(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero=3 then
begin
casa[i,k,m]:=equipo[i,2+m];
fuera[i,k,m]:=equipo[i,2-(m-1)];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz1[equipo[i,2-(m-1)],equipo[i,
2+m]],matriz1[equipo[i,2+m],equipo[i,2-
(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
{Aquí ponemos la matriz1, juegan 3-2;4-1, cuando señalo la matriz1 que pongo primero el visitante, es
porque la matriz 1 creada, la casilla[1,2], dice los goles del equipo 2 como local al 1 como visitante en
cambio con la matriz[] la casilla [1,2], nos indica los goles del equipo 1 como local al 2 como visitante}
end
else if numero=4 then
begin
casa[i,k,m]:=equipo[i,1+(m-1)];
fuera[i,k,m]:=equipo[i,2+m];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz[equipo[i,1+(m-1)],equipo[i,
2+m]],matriz[equipo[i,2+m],equipo[i,1+
(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero=5 then
begin
casa[i,k,m]:=equipo[i,m];
fuera[i,k,m]:=equipo[i,4-(m-1)];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz[equipo[i,m],equipo[i,4-(m-1)]],
matriz[equipo[i,4-(m-1)],equipo[i,m]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero=6 then
begin
casa[i,k,m]:=equipo[i,1+(m-1)];
fuera[i,k,m]:=equipo[i,2+m];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz[equipo[i,1+(m-1)],equipo[i,
2+m]],matriz[equipo[i,2+m],equipo[i,1+
(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end;
end;
end;
{JORNADA5}
5 : begin
for i:=1 to 8 do
for m:=1 to 2 do
begin
if numero=1 then {14-23}
begin
casa[i,k,m]:=equipo[i,4-2*(m-1)];
fuera[i,k,m]:=equipo[i,3-2*(m-1)];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz1[equipo[i,3-2*(m-1)],equipo[i,
4-2*(m-1)]],matriz1[equipo[i,4-2*(m-1)],equipo[i,
3-2*(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero=2 then
begin
casa[i,k,m]:=equipo[i,2+m];
fuera[i,k,m]:=equipo[i,2-(m-1)];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz1[equipo[i,2-(m-1)],equipo[i,
2+m]],matriz1[equipo[i,2+m],equipo[i,2-
(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
{Aquí ponemos la matriz1, juegan 3-2;4-1, cuando señalo la matriz1 que pongo primero el visitante, es
porque la matriz 1 creada, la casilla[1,2], dice los goles del equipo 2 como local al 1 como visitante en
cambio con la matriz[] la casilla [1,2], nos indica los goles del equipo 1 como local al 2 como visitante}
end
else if numero=3 then
begin
casa[i,k,m]:=equipo[i,4-2*(m-1)];
fuera[i,k,m]:=equipo[i,3-2*(m-1)];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz1[equipo[i,3-2*(m-1)],equipo[i,
4-2*(m-1)]],matriz1[equipo[i,4-2*(m-1)],equipo[i,
3-2*(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero=4 then
begin
casa[i,k,m]:=equipo[i,4-2*(m-1)];
fuera[i,k,m]:=equipo[i,3-2*(m-1)];
simularpartido(casa[i,k,m],fuera[i,k,m],matriz1[equipo[i,3-2*(m-1)],equipo[i,
4-2*(m-1)]],matriz1[equipo[i,4-2*(m-1)],equipo[i,
3-2*(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);

```



```

{JORNADA2}
2 : begin
  for i:=1 to 8 do
    for m:=1 to 2 do
      begin
        if numero1=1 then          {14-23}
          begin
            casa[i,k,m]:=equipo[i,4-2*(m-1)];
            fuera[i,k,m]:=equipo[i,3-2*(m-1)];
            simularpartido(casa[i,k,m],fuera[i,k,m],matrizuni[equipo[i,
            4-2*(m-1)],equipo[i,3-2*(m-1)]],matrizuni[equipo[i,3-2*(m-1)],equipo[i,
            4-2*(m-1)]],golesafavorequipo,golesencontraequipo,
            puntosequipo,golesenfrentamientos,puntosenfrentamientos);
            {Aquí ponemos la matriz1, juegan 3-2;4-1, cuando señalo la matriz1 que pongo primero el visitante, es
            porque la matriz 1 creada, la casilla[1,2], dice los goles del equipo 2 cmo local al 1 como visitante en
            cambio con la matriz[] la casilla [1,2], nos indica los goles del equipo 1 como local al 2 como visitante}
          end
          else if numero1=2 then
            begin
              casa[i,k,m]:=equipo[i,2+m];
              fuera[i,k,m]:=equipo[i,2-(m-1)];
              simularpartido(casa[i,k,m],fuera[i,k,m],matrizuni[equipo[i,2+m],equipo[i,2-
              (m-1)]],matrizuni[equipo[i,2-(m-1)],equipo[i,
              2+m]],golesafavorequipo,golesencontraequipo,
              puntosequipo,golesenfrentamientos,puntosenfrentamientos);
            end
          else if numero1=3 then
            begin
              casa[i,k,m]:=equipo[i,4-(m-1)];
              fuera[i,k,m]:=equipo[i,2-(m-1)];
              simularpartido(casa[i,k,m],fuera[i,k,m],matrizuni[equipo[i,4-(m-1)],equipo[i,
              2-(m-1)]],matrizuni[equipo[i,2-(m-1)],equipo[i,4-
              (m-1)]],golesafavorequipo,golesencontraequipo,
              puntosequipo,golesenfrentamientos,puntosenfrentamientos);
            end
          end;
        end;
      end;
    end;
  end;
  for i:=1 to 8 do
    for m:=1 to 2 do
      begin
        if numero1=1 then
          begin
            casa[i,6,m]:=equipo[i,1+(m-1)];
            fuera[i,6,m]:=equipo[i,2+m];
          end
        else if numero1=2 then
          begin
            casa[i,6,m]:=equipo[i,1+2*(m-1)];
            fuera[i,6,m]:=equipo[i,2*m];
          end
        else if numero1=3 then
          begin
            casa[i,6,m]:=equipo[i,m];
            fuera[i,6,m]:=equipo[i,4-(m-1)];
          end
        end;
      end;
    end;
  end;
  numpint:=0;
  numpcol:=0;
  for i:=1 to 8 do
    begin
      vect4:=Clasificacion(i,puntosequipo,golesafavorequipo,golesencontraequipo,golesenfrentamientos,puntosenfrentamientos,equipo);
      numpcol:=numpcol+colision(i,casa,fuera,equipo,golesafavorequipo,golesencontraequipo,
      puntosequipo,golesenfrentamientos,puntosenfrentamientos,vect4[1],vect4[2],vect4[3],vect4[4]);
      numpint:= numpint
      +partidosintrascendentes(i,casa,fuera,equipo,golesafavorequipo,golesencontraequipo,
      puntosequipo,golesenfrentamientos,puntosenfrentamientos,vect4[1],vect4[2],vect4[3],vect4[4]);
    end;
  end;
  intentos:=intentos+numpint;
  intentos1:=intentos1+numpcol;
  if partidos=1 then
    begin
      {JORNADA 6}
      {Simulamos la ultima jornada y con esto veremos que equipos son los que se han clasificado.}
      for i:=1 to 8 do
        for m:=1 to 2 do
          begin
            if numero=1 then
              begin
                casa[i,6,m]:=equipo[i,m];
                fuera[i,6,m]:=equipo[i,4-(m-1)];
                simularpartido(casa[i,6,m],fuera[i,6,m],matriz[equipo[i,m],equipo[i,4-
                (m-1)]],matriz[equipo[i,4-
                (m-1)],equipo[i,m]],golesafavorequipo,golesencontraequipo,
                puntosequipo,golesenfrentamientos,puntosenfrentamientos);
              end
            else if numero=2 then
              begin
                casa[i,6,m]:=equipo[i,1+2*(m-1)];
                fuera[i,6,m]:=equipo[i,2*m];
                simularpartido(casa[i,6,m],fuera[i,6,m],matriz[equipo[i,1+2*(m-1)],equipo[i,
                2*m]],matriz[equipo[i,2*m],equipo[i,
                1+2*(m-1)]],golesafavorequipo,golesencontraequipo,
                puntosequipo,golesenfrentamientos,puntosenfrentamientos);
              end
            else if numero=3 then
              begin

```

```

casa[i,6,m]:=equipo[i,1+(m-1)];
fuera[i,6,m]:=equipo[i,2+m];
simularpartido(casa[i,6,m],fuera[i,6,m],matriz[equipo[i,1+(m-1)],equipo[i,
2+m]] ,matriz[equipo[i,2+m],equipo[i,1+
(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero=4 then
begin
casa[i,6,m]:=equipo[i,2+m];
fuera[i,6,m]:=equipo[i,2-(m-1)];
simularpartido(casa[i,6,m],fuera[i,6,m],matriz[equipo[i,2-(m-1)],equipo[i,
2+m]] ,matriz[equipo[i,2+m],equipo[i,2-
(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
{Aquí ponemos la matriz1, juegan 3-2;4-1, cuando señalo la matriz1 que pongo primero el visitante, es
porque la matriz 1 creada, la casilla[1,2], dice los goles del equipo 2 como local al 1 como visitante en
cambio con la matriz[] la casilla [1,2], nos indica los goles del equipo 1 como local al 2 como visitante}
end
else if numero=5 then
begin
casa[i,6,m]:=equipo[i,4-(m-1)];
fuera[i,6,m]:=equipo[i,2-(m-1)];
simularpartido(casa[i,6,m],fuera[i,6,m],matriz1[equipo[i,2-(m-1)],equipo[i,4-
(m-1)]],matriz1[equipo[i,4-(m-1)],equipo[i,2-
(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero=6 then
begin
casa[i,6,m]:=equipo[i,4-2*(m-1)];
fuera[i,6,m]:=equipo[i,3-2*(m-1)];
simularpartido(casa[i,6,m],fuera[i,6,m],matriz1[equipo[i,
3-2*(m-1)],equipo[i,4-2*(m-1)]],matriz1[equipo[i,4-2*(m-1)],equipo[i,
3-2*(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
end;
end;
if partidos=2 then
begin
{JORNADA 3}
{Simulamos la ultima jornada y con esto veremos que equipos son los que se han clasificado.}
for i:=1 to 8 do
for m:=1 to 2 do
begin
if numero1=1 then
begin
casa[i,6,m]:=equipo[i,1+(m-1)];
fuera[i,6,m]:=equipo[i,2+m];
simularpartido(casa[i,6,m],fuera[i,6,m],matrizuni[equipo[i,1+
(m-1)],equipo[i,2+m]] ,matrizuni[equipo[i,2+m],equipo[i,1+
(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero1=2 then
begin
casa[i,6,m]:=equipo[i,1+2*(m-1)];
fuera[i,6,m]:=equipo[i,2*m];
simularpartido(casa[i,6,m],fuera[i,6,m],matrizuni[equipo[i,
1+2*(m-1)],equipo[i,2*m]] ,matrizuni[equipo[i,2*m],equipo[i,
1+2*(m-1)]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
else if numero1=3 then
begin
casa[i,6,m]:=equipo[i,m];
fuera[i,6,m]:=equipo[i,4-(m-1)];
simularpartido(casa[i,6,m],fuera[i,6,m],matrizuni[equipo[i,m],equipo[i,4-
(m-1)]],matrizuni[equipo[i,4-
(m-1)],equipo[i,m]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end
end;
end;
for i:=1 to 8 do
begin
vect41:=Clasificacion(i,puntosequipo,golesafavorequipo,golesencontraequipo,golesenfrentamientos,
puntosenfrentamientos,equipo);
vclas[vect41[1]]:=vclas[vect41[1]]+1;
vclas[vect41[2]]:=vclas[vect41[2]]+1;
end;
end;
{Sacamos el numero de veces que pasa cada equipo.}
write('[' );
for p:=1 to 32 do
begin
write(vclas[p],', ');
end;
writeln(')');
writeln(intentos);
writeln(intentos1);
write(Resul,intentos,' ',intentos1,' ');
for p:=1 to 32 do
begin
write(Resul,vclas[p],', ');
end;
close(q);
close(f);

```

```

close(u);
close(Resul);
ReadKey;
end.

```

Liga entre todos los equipos

En este apartado, realizamos un programa en el que se simula una liga entre todos los equipos que participan en la competición. Se busca hacer lo siguiente, hacer una liga en la que todos jueguen contra todos con el sistema de puntos tradicional y a partir de 10000 simulaciones de esta liga, obtendremos cuántos puntos ha obtenido cada equipo y en función de esto se sacara el orden con el que meteremos a los equipos en los bombos.

```

program Ligatodosgrupos4;
uses crt,math;
type
  vector=array[1..32]of integer;
  Tmatriz32=array[1..32,1..32] of integer;
var
  i,j,r,aux:integer;
  puntosequipo,golesafavorequipo,golesencontraequipo:array[1..32] of integer;
  a:real;
  vectPuntos:vector;
  golesenfrentamientos,puntosenfrentamientos:Tmatriz32;
  misdatos,misdatos1:array[1..1024] of real; {Los tengo que definir como reales ya que coge valores reales}
  matriz,matriz1:array[1..32,1..32] of real;
  posicion,posicion1:integer;
  f,q,Resul:text;
  p,s:integer;
  equipos:vector;
  {Definimos la funcion densidad de Poisson y para ello defino antes el factorial}
  function factorial( num : integer) : integer;
  begin
    if num = 1 then
      factorial := 1
    else if num = 0 then
      factorial := 1
    else
      factorial := num * factorial( num-1 );
    end;
  end;
  function Poisson(landa:real; x:integer):real;
  begin
    Poisson:=Exp(-landa)*power(landa,x)/factorial(x);
  end;
#Simulo una Poisson y que me diga cuantos goles marca el equipo local
tomando un valor aleatorio y y mirando donde cae en la probabilidad acumulada

function simularPoissonL(landa:real):integer;
var i:integer;
y:real;
valor:array[1..10] of real;
pacum:array[1..10] of real;
glocal:integer;
begin
for i:=0 to 9 do
begin
valor[i+1]:=Poisson(landa,i);
if i=0 then
begin
pacum[i+1]:=valor[i+1];
end
else
begin
pacum[i+1]:=pacum[i]+valor[i+1];
end;
y:=random();
glocal:=0;
while (pacum[glocal+1]<y) and (glocal<=10) do
begin
glocal:=glocal+1;
end;
end;
simularPoissonL:=glocal;
end;
{Simulo una Poisson y que me diga cuantos goles marca el equipo visitante tomando un valor aleatorio y y mirando donde
cae en la probabilidad acumulada}
function simularPoissonV(landa:real):integer;
var
i:integer;
y:real;
valor:array[1..10] of real;
pacum:array[1..10] of real;
gvisitante:integer;
begin
for i:=0 to 9 do
begin
valor[i+1]:=Poisson(landa,i);
if i=0 then pacum[i+1]:=valor[i+1]
else pacum[i+1]:=pacum[i]+valor[i+1];
y:=random();

```

```

    gvisitante:=0;
    while (pacum[gvisitante+1]<y) and (gvisitante<=10) do gvisitante:=gvisitante+1;
    end;
simularPoissonV:=gvisitante;
end;

{Hago un procedimiento que meto a los equipos que juegan y el valor del landa correspondiente, y se me van añadiendo los valores de puntos,
goles a favor y goles en contra que tengo de la jornada anterior}
procedure simularpartido(a:integer;b:integer;c:real;d:real;var gfavoreq:vector;var gcontrareq: vector;var
ptoseqpo:vector;var golesentreequipos:Tmatriz32;var puntosenreequipos:Tmatriz32);
var
    golesfavora:integer;
    golescontraa:integer;
begin
    golesfavora:=simularPoissonL(c);
    golescontraa:=simularPoissonV(d);
    gfavoreq[a]:=gfavoreq[a]+golesfavora;
    gcontrareq[a]:=gcontrareq[a]+golescontraa;
    gfavoreq[b]:=gfavoreq[b]+golescontraa;
    gcontrareq[b]:=gcontrareq[b]+golesfavora;
    if (golesfavora>golescontraa) then
    begin
        ptoseqpo[a]:= ptoseqpo[a]+3;
        puntosenreequipos[a,b]:=puntosenreequipos[a,b]+3;
    end;
    if (golesfavora<golescontraa) then
    begin
        ptoseqpo[b]:= ptoseqpo[b]+3;
        puntosenreequipos[b,a]:=puntosenreequipos[b,a]+3;
    end;
    if(golesfavora=golescontraa) then
    begin
        ptoseqpo[a]:= ptoseqpo[a]+1;
        ptoseqpo[b]:= ptoseqpo[b]+1;
        puntosenreequipos[a,b]:=puntosenreequipos[a,b]+1;
        puntosenreequipos[b,a]:=puntosenreequipos[b,a]+1;
    end;
    golesentreequipos[a,b]:=golesfavora+golesentreequipos[a,b];
    golesentreequipos[b,a]:=golescontraa+golesentreequipos[b,a];
end;
{#####PROGRAMA PRINCIPAL}
begin
    randomize();
    assign(q,'DEFINITIVO0LandasA132PoissonFiltrado.txt'); {Este documento cojo las columnas y voy poniendo una debajo de otra}
    assign(f,'DEFINITIVO0LandasA32PoissonFiltrado.txt'); {Este documento lo hice cogiendo filas y pasandolas a columnas y
poniendo una debajo de otra}
    assign(Resul,'Resultados.txt');
    reset(q);
    reset(f);
    reset(Resul);
    rewrite(Resul);
    posicion1:=1;
    posicion:=1;
    while not Eof(q) do {Lectura de los landas, sacamos la matriz de R, los pongo todos en una columna en un bloc de notas
y digo que me lea todo el fichero y me los guarde en un array Misdatos y
posteriormente, ese array lo convertimos en una amtriz}
    begin
        readLn(q,a);
        misdatos1[posicion1]:=a;
        posicion1:=posicion1+1;
    end;
    while not Eof(f) do
    begin
        readLn(f,a);
        misdatos[posicion]:=a;
        posicion:=posicion+1;
    end;
    for i:=1 to 32 do for j:=1 to 32 do matriz1[i,j]:=misdatos1[32*(i-1)+j];
    for i:=1 to 32 do
    begin
        for j:=1 to 32 do matriz[i,j]:=misdatos[32*(i-1)+j]
        end;
    end;
    for r:=1 to 32 do
    begin
        puntosequipo[r]:=0;
        golesafavorequipo[r]:=0;
        golesencontraequipo[r]:=0;
    end;
    for i:=1 to 32 do
    begin
        for j:=1 to 32 do golesenfrentamientos[i,j]:=0;
        end;
    end;
    for i:=1 to 32 do
    begin
        for j:=1 to 32 do puntosenfrentamientos[i,j]:=0;
        end;
    end;
    for s:=1 to 10000 do
    begin
    for r:=1 to 32 do equipos[r]:=r;
    for i:=1 to 32 do
        for j:=1 to 32 do
            begin
                if i<j then
                begin
                    simularpartido(equipos[i],equipos[j],matriz[equipos[i],equipos[j]]
                    ,matriz[equipos[j],equipos[i]],golesafavorequipo,golesencontraequipo,
                    puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                end;
                if i>j then

```

```

begin
  simularpartido(equipos[j],equipos[i],matriz1[equipos[i],equipos[j]]
    ,matriz1[equipos[j],equipos[i]],golesafavorequipo,golesencontraequipo,
    puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end;
end;
end;
for p:=1 to 32 do vectPuntos[p]:=puntosequipo[p];
{Sacamos el numero de veces que pasa cada equipo.}
write('[' );
for p:=1 to 32 do write(Resul,puntosequipo[p],',');
writeln(')');
write('[' );
for p:=1 to 32 do write(vectPuntos[p], ' ');
writeln(')');
for p := 1 to 32 do begin
  for j := p+1 to 32 do begin
    if (puntosequipo[j] > puntosequipo[p]) then begin
      aux := puntosequipo[p];
      puntosequipo[p] := puntosequipo[j];
      puntosequipo[j] := aux;
    end;
  end;
end;
end;
for p := 1 to 32 do Write(puntosequipo[p], ' ');
close(q);
close(f);
close(Resul);
ReadKey;
end.

```

Número de ocasiones entre los 16 mejores

En este último apartado sobre los grupos de 4, con el fin de ver si nuestro sistema es justo, pretendemos saber cuántas veces queda entre los 16 mejores cada uno de los participantes de esta competición. A partir de esto, pretendemos sacar una correlación con la cantidad de veces que se clasifica cada equipo en el programa principal.

```

program Grupos4mejores16;
uses crt,math;
type
  vector=array[1..32]of integer;
  Tmatriz32=array[1..32,1..32] of integer;
var
  i,j,r,hh,x,aux:integer;
  puntosequipo,golesafavorequipo,golesencontraequipo:array[1..32] of integer;
  a:real;
  vectPuntos,vectPuntos1:vector;
  golesenfrentamientos,puntosenfrentamientos:Tmatriz32;
  misdatos,misdatos1:array[1..1024] of real; {Los tengo que definir como reales ya que coge valores reales}
  matriz,matriz1:array[1..32,1..32] of real;
  posicion,posicion1:integer;
  f,q,Resul:text;
  p,s:integer;
  vclas:vector;
  equipos:vector;
  {Definimos la funcion densidad de Poisson y para ello defino antes el factorial}
  function factorial( num : integer) : integer;
  begin
    if num = 1 then factorial := 1
    else if num = 0 then factorial := 1
    else factorial := num * factorial( num-1 );
  end;
  function Poisson(landa:real; x:integer):real;
  begin
    Poisson:=Exp(-landa)*power(landa,x)/factorial(x);
  end;
  {Simulo una Poisson y que me diga cuantos goles marca el equipo local tomando un valor aleatorio y y mirando donde cae en la probabilidad acumulada}
  function simularPoissonL(landa:real):integer;
  var i:integer;
  y:real;
  valor:array[1..10] of real;
  pacum:array[1..10] of real;
  glocal:integer;
  begin
    for i:=0 to 9 do
      begin
        valor[i+1]:=Poisson(landa,i);
        if i=0 then pacum[i+1]:=valor[i+1]
        else pacum[i+1]:=pacum[i]+valor[i+1];
        y:=random();
        glocal:=0;
        while (pacum[glocal+1]<y) and (glocal<=10) do glocal:=glocal+1;
      end;
    simularPoissonL:=glocal;
  end;
  {Simulo una Poisson y que me diga cuantos goles marca el equipo visitante

```

```

tomando un valor aleatorio y y mirando donde cae en la probabilidad acumulada}
function simularPoissonV(landa:real):integer;
var
  i:integer;
  y:real;
  valor:array[1..10] of real;
  pacum:array[1..10] of real;
  gvisitante:integer;
begin
for i:=0 to 9 do
begin
  valor[i+1]:=Poisson(landa,i);
  if i=0 then pacum[i+1]:=valor[i+1]
  else pacum[i+1]:=pacum[i]+valor[i+1];
  y:=random();
  gvisitante:=0;
  while (pacum[gvisitante+1]<y) and (gvisitante<=10) do gvisitante:=gvisitante+1;
end;
simularPoissonV:=gvisitante;
end;

{Hago un procedimiento que meto a los equipos que juegan y el valor del landa correspondiente, y se me van añadiendo
los valores de puntos, goles a favor y goles en contra que tengo de la jornada anterior}
procedure simularpartido(a:integer;b:integer;c:real;d:real;var gfavoreq:vector;var gcontrareq: vector;var
ptoseqpo:vector;var golesentreequipos:Tmatriz32;var puntosentreequipos:Tmatriz32);
var
  golesfavora:integer;
  golescontraa:integer;
begin
  golesfavora:=simularPoissonL(c);
  golescontraa:=simularPoissonV(d);
  gfavoreq[a]:=gfavoreq[a]+golesfavora;
  gcontrareq[a]:=gcontrareq[a]+golescontraa;
  gfavoreq[b]:=gfavoreq[b]+golescontraa;
  gcontrareq[b]:=gcontrareq[b]+golesfavora;
  if (golesfavora>golescontraa) then
  begin
    ptoseqpo[a]:= ptoseqpo[a]+3;
    puntosentreequipos[a,b]:=puntosentreequipos[a,b]+3;
  end;
  if (golesfavora<golescontraa) then
  begin
    ptoseqpo[b]:= ptoseqpo[b]+3;
    puntosentreequipos[b,a]:=puntosentreequipos[b,a]+3;
  end;
  if(golesfavora=golescontraa) then
  begin
    ptoseqpo[a]:= ptoseqpo[a]+1;
    ptoseqpo[b]:= ptoseqpo[b]+1;
    puntosentreequipos[a,b]:=puntosentreequipos[a,b]+1;
    puntosentreequipos[b,a]:=puntosentreequipos[b,a]+1;
  end;
  golesentreequipos[a,b]:=golesfavora+golesentreequipos[a,b];
  golesentreequipos[b,a]:=golescontraa+golesentreequipos[b,a];
end;
#####PROGRAMA PRINCIPAL
begin
  randomize();
  assign(q,'DEFINITIVOLandasA132PoissonFiltrado.txt'); {Este documento cojo las columnas y voy poniendo una
  debajo de otra}
  assign(f,'DEFINITIVOLandasA32PoissonFiltrado.txt'); {Este documento lo hice cogiendo filas y pasandolas a columnas y
  poniendo una debajo de otra}
  assign(Resul,'Resultados.txt');
  reset(q);
  reset(f);
  reset(Resul);
  rewrite(Resul);
  posicion1:=1;
  posicion:=1;
  while not Eof(q) do {Lectura de los landas, sacamos la matriz de R, los pongo todos en una columna en un bloc de notas y
  digo que me lea todo el fichero y me los guarde en un array Misdatos y posteriormente, ese array lo
  convertimos en una matriz}
  begin
    readLn(q,a);
    misdatos1[posicion1]:=a;
    posicion1:=posicion1+1;
  end;
  while not Eof(f) do
  begin
    readLn(f,a);
    misdatos[posicion]:=a;
    posicion:=posicion+1;
  end;
  for i:=1 to 32 do
  begin
    for j:=1 to 32 do matriz1[i,j]:=misdatos1[32*(i-1)+j];
    end;
  for i:=1 to 32 do
  begin
    for j:=1 to 32 do matriz[i,j]:=misdatos[32*(i-1)+j];
    end;
  end;
  for p:=1 to 32 do
  begin
    vclas[p]:=0;
  end;
  for s:=1 to 10000 do
  begin
    for r:=1 to 32 do
    begin

```

```

        puntosequipo[r]:=0;
        golesafavorequipo[r]:=0;
        golesencontraequipo[r]:=0;
    end;
for i:=1 to 32 do
    begin
        for j:=1 to 32 do golesenfrentamientos[i,j]:=0;
        end;
    for i:=1 to 32 do
        begin
            for j:=1 to 32 do puntosenfrentamientos[i,j]:=0;
            end;
        for r:=1 to 32 do
            begin
                equipos[r]:=r;
            end;
        for i:=1 to 32 do
            for j:=1 to 32 do
                begin
                    if i<j then
                        begin
                            similarpartido(equipos[i],equipos[j],matriz[equipos[i],equipos[j]]
                                ,matriz[equipos[j],equipos[i]],golesafavorequipo,golesencontraequipo,
                                puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                        end;
                    if i>j then
                        begin
                            similarpartido(equipos[j],equipos[i],matriz1[equipos[i],equipos[j]]
                                ,matriz1[equipos[j],equipos[i]],golesafavorequipo,golesencontraequipo,
                                puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                        end;
                    end;
                end;
            end;
        for p:=1 to 32 do
            begin
                vectPuntos[p]:=puntosequipo[p];
            end;
        for p := 1 to 32 do begin
            for j := p+1 to 32 do begin

                if (puntosequipo[j] > puntosequipo[p]) then begin
                    aux := puntosequipo[p];
                    puntosequipo[p] := puntosequipo[j];
                    puntosequipo[j] := aux;
                end;
            end;
        end;
        for p:=1 to 32 do
            begin
                vectPuntos1[p]:=puntosequipo[p];
            end;
            p:=1;
            while (p<33) do
                begin
                    hh:=1;
                    x:=0;
                    while (hh<17) and (x=0) do
                        begin
                            if vectPuntos[p]=vectPuntos1[hh] then
                                begin
                                    vclas[equipos[p]]:=vclas[equipos[p]]+1;
                                    vectPuntos1[hh]:=0;
                                    x:=1;
                                end
                            else
                                begin
                                    x:=0;
                                    vclas[equipos[p]]:=vclas[equipos[p]];
                                end;
                            hh:=hh+1;
                        end;
                    p:=p+1;
                end;
            end;
        for p:=1 to 32 do
            begin
                write(Resul,vclas[equipos[p]],',');
            end;
            close(q);
            close(f);
            close(Resul);
            ReadKey;
        end.

```

Simulaciones con grupos de 3 equipos

Programa principal

En este apartado se muestra el programa que se ha utilizado para hacer las correspondientes simulaciones de los grupos 3 tanto si solamente se clasifica 1 equipo como si se clasifican 2, sacando así los

partidos de colisión, partidos intrascendentes y la cantidad de veces que se clasifica cada equipo en cada uno de los casos después de realizar una simulación de Monte Carlo. En este programa se añaden ya los datos tanto si las fases de grupo fuesen de ida y vuelta o bien de partido único. Además, están añadidos tantos los datos con las tablas ordenando los equipos por ranking FIFA, como las tablas en las que se ordenan los equipos por la simulación de jugar una liga entre todos los equipos (Está a continuación)

```

program ProgramaGrupos3;
uses crt,math;
type
  vector=array[1..48]of integer;
  Tmatriz=array[1..16,1..3] of integer;
  Tmatriz32=array[1..48,1..48] of integer;
  vector4=array[1..3] of integer;
  jornada=array[1..16,1..6] of integer;
var
  i,j,k,r,numero,sorteo,partidos,numero1,clasificados,orden:integer;
  r1,r2,r3,r4,r5,r6:integer;
  rr1,rr2,rr3,rr4,rr5,rr6:integer;
  t1,t2,t3,t4,t5,t6:integer;
  lista1,lista2,lista3,lista4,lista5,lista6:array[1..16] of integer;
  g,puntosequipo,golesafavorequipo,golesencontraequipo:array[1..48] of integer;
  equipo:array[1..16,1..3]of integer;
  a:real;
  golesenfrentamientos,puntosenfrentamientos:Tmatriz32;
  casa,fuera,nojuega:jornada;
  misdatos,misdatos1,misdatosu:array[1..2304] of real;}
  matriz,matriz1,matrizuni48:array[1..48,1..48] of real;
  posicion,posicion1,posicionu:integer;
  f,q,u,resul:text;
  numpint,numpcol,numpint1,numpcol1:integer;
  intentos,intentos1:integer;
  p,s:integer;
  vclas:vector;
  vect4,vect41:vector4;

{Definimos la funcion densidad de Poisson y para ello defino antes el factorial}
function factorial( num : integer) : integer;
begin
  if num = 1 then factorial := 1
  else if num = 0 then factorial := 1
  else factorial := num * factorial( num-1 );
end;
function Poisson(landa:real; x:integer):real;
begin
  Poisson:=Exp(-landa)*power(landa,x)/factorial(x);
end;

{Simulo una Poisson y que me diga cuantos goles marca el equipo local tomando un valor
aleatorio y mirando donde cae en la probabilidad acumulada}
function simularPoissonL(landa:real):integer;
  var i:integer;
  y:real;
  valor:array[1..10] of real;
  pacum:array[1..10] of real;
  glocal:integer;
begin
for i:=0 to 9 do
  begin
  valor[i+1]:=Poisson(landa,i);
  if i=0 then pacum[i+1]:=valor[i+1]
  else pacum[i+1]:=pacum[i]+valor[i+1];
  y:=random();
  glocal:=0;
  while (pacum[glocal+1]<y) and (glocal<=10) do glocal:=glocal+1;
  end;
simularPoissonL:=glocal;
end;
{Simulo una Poisson y que me diga cuantos goles marca el equipo visitante tomando un valor
aleatorio y mirando donde cae en la probabilidad acumulada}
function simularPoissonV(landa:real):integer;
  var
  i:integer;
  y:real;
  valor:array[1..10] of real;
  pacum:array[1..10] of real;
  gvisitante:integer;
begin
for i:=0 to 9 do
  begin
  valor[i+1]:=Poisson(landa,i);
  if i=0 then pacum[i+1]:=valor[i+1]
  else pacum[i+1]:=pacum[i]+valor[i+1];
  y:=random();
  gvisitante:=0;
  while (pacum[gvisitante+1]<y) and (gvisitante<=10) do gvisitante:=gvisitante+1;
  end;
simularPoissonV:=gvisitante;
end;

{Hago un procedimiento que meto a los equipos que juegan y el valor del landa
correspondiente, y se me van añadiendo los valores de puntos, goles a favor
y goles en contra que tengo de la jornada anterior}

procedure simularpartido(a:integer;b:integer;c:real;d:real;var gfavoreq:vector;

```



```

                var gcontrareq: vector; var ptoseqpo: vector; var golesentreequipos: Tmatriz32;
                var puntosentreequipos: Tmatriz32;
var
  golesfavora: integer;
  golescontraa: integer;
begin
  golesfavora:=simularPoissonL(c);
  golescontraa:=simularPoissonV(d);
  gfavoreq[a]:=gfavoreq[a]+golesfavora;
  gcontrareq[a]:=gcontrareq[a]+golescontraa;
  gfavoreq[b]:=gfavoreq[b]+golescontraa;
  gcontrareq[b]:=gcontrareq[b]+golesfavora;
  if (golesfavora>golescontraa) then
    begin
      ptoseqpo[a]:= ptoseqpo[a]+3;
      puntosentreequipos[a,b]:=puntosentreequipos[a,b]+3;
    end;
  if (golesfavora<golescontraa) then
    begin
      ptoseqpo[b]:= ptoseqpo[b]+3;
      puntosentreequipos[b,a]:=puntosentreequipos[b,a]+3;
    end;
  if(golesfavora=golescontraa) then
    begin
      ptoseqpo[a]:= ptoseqpo[a]+1;
      ptoseqpo[b]:= ptoseqpo[b]+1;
      puntosentreequipos[a,b]:=puntosentreequipos[a,b]+1;
      puntosentreequipos[b,a]:=puntosentreequipos[b,a]+1;
    end;
  golesentreequipos[a,b]:=golesfavora+golesentreequipos[a,b];
  golesentreequipos[b,a]:=golescontraa+golesentreequipos[b,a];
end;

{Esto de Primerclasificado,Segundoclasificado,Ultimoclasificado y Tercerclasificado,
lo hago para establecer un orden, para el caso del primero y segundo empezare mirando
por el primer miembro del grupo, para el caso del tercero y cuarto, empezare mirando
por el último, con esto consigo que en caso de empate no se repita ninguno y de esta manera
tengo un orden establecido, lo siguiente será ordenarlos en caso de empate.}

function Primerclasificado(i: integer; puntuacion: vector; EQ: Tmatriz): integer;
var j, POSI, s: integer;
  MAYOR: array[1..16] of integer;
  matrizpuntos, matrizordenada: array[1..16, 1..3] of integer;
  equipo: array[1..16, 1..3] of integer;
begin
  for j:=1 to 3 do
    begin
      equipo[i,j]:=EQ[i,j];
      matrizpuntos[i,j]:=puntuacion[equipo[i,j]];
      MAYOR[i]:=matrizpuntos[i,1];
      POSI:=1;
      for s:=2 to 3 do
        if matrizpuntos[i,s]>MAYOR[i] then
          begin
            MAYOR[i]:=matrizpuntos[i,s];
            POSI:=s;
          end;
      matrizordenada[i,1]:=equipo[i,POSI];
    end;
  Primerclasificado:=matrizordenada[i,1];
end;

function Segundoclasificado(i: integer; puntuacion: vector; EQ: Tmatriz): integer;
var j, POSI, POSI2, s: integer;
  MAYOR, MAYOR2: array[1..16] of integer;
  matrizpuntos, matrizordenada: array[1..16, 1..3] of integer;
  equipo: array[1..16, 1..3] of integer;
begin
  for j:=1 to 3 do
    begin
      equipo[i,j]:=EQ[i,j];
      matrizpuntos[i,j]:=puntuacion[equipo[i,j]];
      MAYOR[i]:=matrizpuntos[i,1];
      POSI:=1;
      for s:=2 to 3 do
        if matrizpuntos[i,s]>MAYOR[i] then
          begin
            MAYOR[i]:=matrizpuntos[i,s];
            POSI:=s;
          end;
      matrizordenada[i,1]:=equipo[i,POSI];
    end;
  {Aquí acabaría el primer clasificado y analizamos los casos del segundo que mas}
  if POSI=1 then
    begin
      for j := POSI to (3-1) do
        begin
          matrizpuntos[i,j] := matrizpuntos[i,j+1];
          matrizpuntos[i,j+1]:=puntuacion[equipo[i,j+1]];
          MAYOR2[i]:=matrizpuntos[i,1];
          POSI2:=1;
          for s:=2 to (3-1) do
            if matrizpuntos[i,s]>MAYOR2[i] then
              begin
                MAYOR2[i]:=matrizpuntos[i,s];
                POSI2:=s;
              end;
          matrizordenada[i,2]:=equipo[i,POSI2+1];
        end;
    end;
end;

```

```

end
else if POSI=2 then
begin
for j := POSI to 2 do
begin
matrizpuntos[i,1] := matrizpuntos[i,1];
matrizpuntos[i,j] := matrizpuntos[i,j+1];
matrizpuntos[i,j+1]:=puntuacion[equipo[i,j+1]];
MAYOR2[i]:=matrizpuntos[i,1];
POSI2:=1;
for s:=2 to (3-1) do
if matrizpuntos[i,s]>MAYOR2[i] then
begin
MAYOR2[i]:=matrizpuntos[i,s];
POSI2:=s;
end;
end;
if POSI2=1 then matrizordenada[i,2]:=equipo[i,POSI2]
else matrizordenada[i,2]:=equipo[i,POSI2+1];
end
end
else if POSI=3 then
begin
j:=3;
matrizpuntos[i,1] := matrizpuntos[i,1];
matrizpuntos[i,2] := matrizpuntos[i,2];
matrizpuntos[i,j] := matrizpuntos[i,j+1];
MAYOR2[i]:=matrizpuntos[i,1];
POSI2:=1;
for s:=2 to (3-1) do
if matrizpuntos[i,s]>MAYOR2[i] then
begin
MAYOR2[i]:=matrizpuntos[i,s];
POSI2:=s;
end;
if POSI2=1 then matrizordenada[i,2]:=equipo[i,POSI2]
else if POSI2=2 then matrizordenada[i,2]:=equipo[i,POSI2]
else matrizordenada[i,2]:=equipo[i,POSI2+1];
end;
Segundoclasificado:=matrizordenada[i,2];
end;
end;

function Ultimoclasificado(i:integer;puntuacion:vector;EQ:Tmatriz):integer;
var j,POSI4,s:integer;
MENOR:array[1..16]of integer;
matrizpuntos,matrizordenada:array[1..16,1..3] of integer;
equipo:array[1..16,1..3]of integer;
begin
for j:=1 to 3 do
begin
equipo[i,j]:=EQ[i,j];
matrizpuntos[i,j]:=puntuacion[equipo[i,j]];
MENOR[i]:=matrizpuntos[i,3];
POSI4:=3;
for s:=2 downto 1 do
begin
if matrizpuntos[i,s]<MENOR[i] then
begin
MENOR[i]:=matrizpuntos[i,s];
POSI4:=s;
end;
end;
matrizordenada[i,3]:=equipo[i,POSI4];
end;
end;
Ultimoclasificado:=matrizordenada[i,3];
end;
{Sacaremos ahora la clasificacion buena definitiva, para ello, nos fijamos en los empates
entre 2 equipos, miraremos su gol average particular y en caso de que coincidan,
miraremos la diferencia de goles general, si coinciden, lo hacemos a sorteo.
En caso de que haya triple empate, miraremos los puntos que han obtenido entre sí los equipos
que esten empatados, el que más puntos haya conseguido, mejor clasificado.Si coinciden,
pasaremos a mirar la diferencia de goles y si vuelven a coincidir ya hacemos un sorteo.}

function Clasificacion(i:integer;puntosequipo:vector;golesafavorequipo:vector;
golesencontraequipo:vector;golesenfrentamientos:Tmatriz32;
puntosenfrentamientos:Tmatriz32;EQ:Tmatriz):vector4;
var
matrizordenada:Tmatriz;
auxiliar,auxiliar2,alea,tp,ts,tu,pp,dd:integer;
begin
matrizordenada[i,1]:=Primerclasificado(i,puntosequipo,EQ);
matrizordenada[i,2]:=Segundoclasificado(i,puntosequipo,EQ);
matrizordenada[i,3]:=Ultimoclasificado(i,puntosequipo,EQ);
{ANALIZAMOS TODOS LOS POSIBLES CASOS 1=2, 1=3, 1=4, 2=3,2=4;3=4 y miramos su gol
average particular en el caso de que haya empate entre dos equipos, miramos los goles que
se han marcado el uno al otro y vemos que si la diferencia entre los goles anotados es mayor
en uno que en otro, ese equipo ira por delante, en caso de que el gol average fuera igual,
nos fijariamos en la diferencia de goles general}
for pp:=1 to 2 do
begin
dd:=pp+1;
if puntosequipo[matrizordenada[i,pp]]=puntosequipo[matrizordenada[i,dd]] then
begin
if golesenfrentamientos[matrizordenada[i,pp],matrizordenada[i,dd]]<
golesenfrentamientos[matrizordenada[i,dd],matrizordenada[i,pp]] then
begin
auxiliar:=matrizordenada[i,pp];
matrizordenada[i,pp]:=matrizordenada[i,dd] ;
matrizordenada[i,dd]:=auxiliar;
end;
end;
end;
end;

```

```

if golesenfrentamientos[matrizordenada[i,pp],matrizordenada[i,dd]]=
golesenfrentamientos[matrizordenada[i,dd],matrizordenada[i,pp]] then
begin
    if golesafavorequipo[matrizordenada[i,pp]]-
golesencontraequipo[matrizordenada[i,pp]]<
golesafavorequipo[matrizordenada[i,dd]]-golesencontraequipo[matrizordenada[i,dd]] then
begin
    auxiliar:=matrizordenada[i,pp];
matrizordenada[i,pp]:=matrizordenada[i,dd] ;
matrizordenada[i,dd]:=auxiliar;
end;
    if golesafavorequipo[matrizordenada[i,pp]]-
golesencontraequipo[matrizordenada[i,pp]]=golesafavorequipo[matrizordenada[i,dd]]-
golesencontraequipo[matrizordenada[i,dd]] then
begin
    alea:=random(2)+1;
    if alea=1 then
begin
matrizordenada[i,pp]:=matrizordenada[i,pp] ;
matrizordenada[i,dd]:=matrizordenada[i,dd];
end;
    if alea=2 then
begin
    auxiliar:=matrizordenada[i,pp];
matrizordenada[i,pp]:=matrizordenada[i,dd] ;
matrizordenada[i,dd]:=auxiliar;
end;
end;
end;
end;
end;
{+++++}

```

{En este caso, analizaremos el caso de que hubiera triple empate, dado que ya es largo de por si, para desempatar nos fijaremos solamente en la diferencia de goles general (hacemos todas las posibles combianciones)}

{En caso de que haya triple empate, miraremos los puntos que han obtenido entre si los equipos que esten empatados, el que más puntos haya conseguido, mejor clasificado. Si coinciden, pasaremos a mirar la diferencia de goles y si vuelven a coincidir ya hacemos un sorteo.}

```

for tp:=1 to 1 do
begin
    ts:=tp+1;
    tu:=tp+2;
    if (puntosequipo[matrizordenada[i,tp]]=puntosequipo[matrizordenada[i,ts]]) and
(puntosequipo[matrizordenada[i,tp]]=puntosequipo[matrizordenada[i,tu]]) then
begin
    if (puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]>
puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]>
puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]]) and
(puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]>puntosenfrentamientos
[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) and
(puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]]>
puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) then
begin
matrizordenada[i,tp]:=matrizordenada[i,tp];
matrizordenada[i,ts]:=matrizordenada[i,ts];
matrizordenada[i,tu]:=matrizordenada[i,tu];
end;
    if (puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]>
puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]]) and
(puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]>
puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) and
(puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]>
puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]) then
begin
matrizordenada[i,tp]:=matrizordenada[i,tp];
auxiliar:=matrizordenada[i,ts];
matrizordenada[i,ts]:=matrizordenada[i,tu] ;
matrizordenada[i,tu]:=auxiliar;
end;
    if (puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]]>
puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]) and
(puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]]>
puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) and
(puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,3]]>
puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) then
begin
auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,ts] ;

```



```

                matrizordenada[i,tu]:=auxiliar;
            end;
        end;
    end;
end;

if (puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]]>
puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]) and
(puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]) then
begin
    auxiliar:=matrizordenada[i,tp];
    matrizordenada[i,tp]:=matrizordenada[i,ts];
    matrizordenada[i,ts]:=auxiliar;
    if golesafavorequipo[matrizordenada[i,ts]]-
    golesencontraequipo[matrizordenada[i,ts]]>golesafavorequipo[matrizordenada[i,tu]]-
    golesencontraequipo[matrizordenada[i,tu]] then
        begin
            matrizordenada[i,ts]:=matrizordenada[i,ts];
            matrizordenada[i,tu]:=matrizordenada[i,tu];
        end;
    if golesafavorequipo[matrizordenada[i,ts]]-
    golesencontraequipo[matrizordenada[i,ts]]<golesafavorequipo[matrizordenada[i,tu]]-
    golesencontraequipo[matrizordenada[i,tu]] then
        begin
            auxiliar:=matrizordenada[i,ts];
            matrizordenada[i,ts]:=matrizordenada[i,tu] ;
            matrizordenada[i,tu]:=auxiliar;
        end;
    if golesafavorequipo[matrizordenada[i,ts]]-
    golesencontraequipo[matrizordenada[i,ts]]=golesafavorequipo[matrizordenada[i,tu]]-
    golesencontraequipo[matrizordenada[i,tu]] then
        begin
            alea:=random(2)+1;
            if alea=1 then
                begin
                    matrizordenada[i,ts]:=matrizordenada[i,ts];
                    matrizordenada[i,tu]:=matrizordenada[i,tu];
                end;
            if alea=2 then
                begin
                    auxiliar:=matrizordenada[i,ts];
                    matrizordenada[i,ts]:=matrizordenada[i,tu] ;
                    matrizordenada[i,tu]:=auxiliar;
                end;
            end;
        end;
    end;
end;
if (puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,tu],matrizordenada[i,ts]]>
puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]) and
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]]) then
begin
    auxiliar:=matrizordenada[i,tp];
    matrizordenada[i,tp]:=matrizordenada[i,tu];
    matrizordenada[i,tu]:=auxiliar;
    if golesafavorequipo[matrizordenada[i,ts]]-
    golesencontraequipo[matrizordenada[i,ts]]>golesafavorequipo[matrizordenada[i,tu]]-
    golesencontraequipo[matrizordenada[i,tu]] then
        begin
            matrizordenada[i,ts]:=matrizordenada[i,ts];
            matrizordenada[i,tu]:=matrizordenada[i,tu];
        end;
    if golesafavorequipo[matrizordenada[i,ts]]-
    golesencontraequipo[matrizordenada[i,ts]]<golesafavorequipo[matrizordenada[i,tu]]-
    golesencontraequipo[matrizordenada[i,tu]] then
        begin
            auxiliar:=matrizordenada[i,ts];
            matrizordenada[i,ts]:=matrizordenada[i,tu] ;
            matrizordenada[i,tu]:=auxiliar;
        end;
    if golesafavorequipo[matrizordenada[i,ts]]-
    golesencontraequipo[matrizordenada[i,ts]]=golesafavorequipo[matrizordenada[i,tu]]-
    golesencontraequipo[matrizordenada[i,tu]] then
        begin
            alea:=random(2)+1;
            if alea=1 then
                begin
                    matrizordenada[i,ts]:=matrizordenada[i,ts];
                    matrizordenada[i,tu]:=matrizordenada[i,tu];
                end;
            if alea=2 then
                begin
                    auxiliar:=matrizordenada[i,ts];
                    matrizordenada[i,ts]:=matrizordenada[i,tu] ;
                    matrizordenada[i,tu]:=auxiliar;
                end;
            end;
        end;
    end;
end;
if (puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos[matrizordenada[i,tp],matrizordenada[i,tu]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tu]])

```



```

golesencontraequipo[matrizordenada[i,ts]] then
begin
matrizordenada[i,tp]:=matrizordenada[i,tp];
matrizordenada[i,ts]:=matrizordenada[i,ts];
end;
if golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]]<golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]] then
begin
auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,ts] ;
matrizordenada[i,ts]:=auxiliar;
end;
if golesafavorequipo[matrizordenada[i,tp]]-
golesencontraequipo[matrizordenada[i,tp]] = golesafavorequipo[matrizordenada[i,ts]]-
golesencontraequipo[matrizordenada[i,ts]] then
begin
alea:=random(2)+1;
if alea=1 then
begin
matrizordenada[i,tp]:=matrizordenada[i,tp];
matrizordenada[i,ts]:=matrizordenada[i,ts];
end;
if alea=2 then
begin
auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,ts] ;
matrizordenada[i,ts]:=auxiliar;
end;
end;
end;
if (puntosenfrentamientos[matrizordenada[i,ts],matrizordenada[i,tp]]
+puntosenfrentamientos[matrizordenada[i,2],
matrizordenada[i,tu]] = puntosenfrentamientos [matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos [matrizordenada[i,tu],matrizordenada[i,ts]]) and
(puntosenfrentamientos [matrizordenada[i,tp],matrizordenada[i,ts]]
+puntosenfrentamientos [matrizordenada[i,tp],matrizordenada[i,tu]] =
puntosenfrentamientos [matrizordenada[i,tu],matrizordenada[i,tp]]
+puntosenfrentamientos [matrizordenada[i,tu],matrizordenada[i,ts]]) then
begin
if (golesafavorequipo [matrizordenada[i,tu]]-
golesencontraequipo [matrizordenada[i,tu]] = golesafavorequipo [matrizordenada[i,tp]]-
golesencontraequipo [matrizordenada[i,tp]]) and (golesafavorequipo [matrizordenada[i,tp]]-
golesencontraequipo [matrizordenada[i,tp]] = golesafavorequipo [matrizordenada[i,ts]]-
golesencontraequipo [matrizordenada[i,ts]]) then
begin
alea:=random(6)+1;
if alea=1 then
begin
matrizordenada[i,tp]:=matrizordenada[i,tp];
matrizordenada[i,ts]:=matrizordenada[i,ts];
matrizordenada[i,tu]:=matrizordenada[i,tu];
end;
if alea=2 then
begin
matrizordenada[i,tp]:=matrizordenada[i,tp];
auxiliar:=matrizordenada[i,ts];
matrizordenada[i,ts]:=matrizordenada[i,tu] ;
matrizordenada[i,tu]:=auxiliar;
end;
if alea=3 then
begin
matrizordenada[i,ts]:=matrizordenada[i,ts];
auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,tu] ;
matrizordenada[i,tu]:=auxiliar;
end;
if alea=4 then
begin
matrizordenada[i,tu]:=matrizordenada[i,tu];
auxiliar:=matrizordenada[i,tp];
matrizordenada[i,tp]:=matrizordenada[i,ts] ;
matrizordenada[i,ts]:=auxiliar;
end;
if alea=5 then
begin
auxiliar:=matrizordenada[i,tp];
auxiliar2:=matrizordenada[i,ts];
matrizordenada[i,tp]:=matrizordenada[i,tu];
matrizordenada[i,ts]:=auxiliar ;
matrizordenada[i,tu]:=auxiliar2;
end;
if alea=6 then
begin
auxiliar:=matrizordenada[i,tp];
auxiliar2:=matrizordenada[i,tu];
matrizordenada[i,tp]:=matrizordenada[i,ts];
matrizordenada[i,ts]:=auxiliar2 ;
matrizordenada[i,tu]:=auxiliar;
end;
end;
if (golesafavorequipo [matrizordenada[i,tp]]-
golesencontraequipo [matrizordenada[i,tp]] > golesafavorequipo [matrizordenada[i,ts]]-
golesencontraequipo [matrizordenada[i,ts]]) and (golesafavorequipo [matrizordenada[i,tp]]-
golesencontraequipo [matrizordenada[i,tp]] > golesafavorequipo [matrizordenada[i,tu]]-
golesencontraequipo [matrizordenada[i,tu]]) and (golesafavorequipo [matrizordenada[i,ts]]-
golesencontraequipo [matrizordenada[i,ts]] > golesafavorequipo [matrizordenada[i,tu]]-
golesencontraequipo [matrizordenada[i,tu]]) then
begin

```



```

Clasificacion[1]:=matrizordenada[i,1];
Clasificacion[2]:=matrizordenada[i,2];
Clasificacion[3]:=matrizordenada[i,3];
end;

{Este proceso nos permitira recorrer los posibles resultados de un partido e ir guardando los datos.}

procedure simpartidocol(a:integer;b:integer;cc:integer;vv:integer;var gfavoreq:vector;
var gcontrareq:vector;var ptoseqpo:vector;var golesentreequipos:Tmatriz32;
var puntosentreequipos:Tmatriz32);

var
golesfavora:integer;
golescontraa:integer;
begin
golesfavora:=cc;
golescontraa:=vv;
{writeln('goles a favor: ',golesfavora); }
{writeln('goles en contra: ',golescontraa);}
gfavoreq[a]:=gfavoreq[a]+golesfavora;
gcontrareq[a]:=gcontrareq[a]+golescontraa;
gfavoreq[b]:=gfavoreq[b]+golescontraa;
gcontrareq[b]:=gcontrareq[b]+golesfavora;
if (golesfavora>golescontraa) then
begin
ptoseqpo[a]:= ptoseqpo[a]+3;
puntosentreequipos[a,b]:=puntosentreequipos[a,b]+3;
end;

if (golesfavora<golescontraa) then
begin
ptoseqpo[b]:= ptoseqpo[b]+3;
puntosentreequipos[b,a]:=puntosentreequipos[b,a]+3;
end;

if(golesfavora=golescontraa) then
begin
ptoseqpo[a]:= ptoseqpo[a]+1;
ptoseqpo[b]:= ptoseqpo[b]+1;
puntosentreequipos[a,b]:=puntosentreequipos[a,b]+1;
puntosentreequipos[b,a]:=puntosentreequipos[b,a]+1;
end;
golesentreequipos[a,b]:=golesfavora+golesentreequipos[a,b];
golesentreequipos[b,a]:=golescontraa+golesentreequipos[b,a];
end;
{Nos saca si en el grupo hay un partido de colisión o no}
function colision(ir:integer;Home:jornada;Away:jornada;Noplay:jornada;Eq:Tmatriz;var gfavoreq:vector;
var gcontrareq:vector;var ptoseqpo:vector;var golesentreequipos:Tmatriz32;
var puntosentreequipos:Tmatriz32;var clasif1:integer;var clasif2:integer;var clasif3:integer):integer;
var
kk,ll,x,escol,ts1:integer;
gfavoreqss:vector;
gcontrareqss:vector;
ptoseqpo:vector;
golesentreequiposs:Tmatriz32;
puntosentreequiposs:Tmatriz32;
vecl1:vector4;
begin
if ((clasif2=Noplay[i,6]) or (clasif1=Noplay[i,6])) then
begin
escol:=1;
kk:=0;
while(kk<10) and (escol=1) do
begin
ll:=0;
while(ll<10) and (escol=1) do
begin
gfavoreqss:=gfavoreq;
gcontrareqss:=gcontrareq;
ptoseqpo:=ptoseqpo;
golesentreequiposs:=golesentreequipos;
puntosentreequiposs:=puntosentreequipos;
simpartidocol(Home[ir,6],Away[ir,6],kk,ll,gfavoreqss,gcontrareqss,ptoseqpo,golesentreequiposs,
puntosentreequiposs);
vecl1:=Clasificacion(ir,ptoseqpo,gfavoreqss,gcontrareqss,
golesentreequiposs,puntosentreequiposs,Eq); {Simulamos el segundo}

ts1:=vecl1[3];
if (Noplay[i,6]=ts1) then
begin
{Si el primer puesto no ha cambiado, entonces se cumplirá que hay colision}
escol:=0;
x:=1;
end
else
begin
escol:=1;
x:=0;
end;
ll:=ll+1;
end;
kk:=kk+1;
end;
end
end
colision:=x;
end;

```

```

function intrascendente2(ir:integer;Home:jornada;Away:jornada;Eq:Tmatriz;var gfavoreq:vector;
var gcontrareq: vector;var ptoseqpo:vector;var golesentreequipos:Tmatriz32;
var puntosentreequipos:Tmatriz32;var clasif1:integer;var clasif2:integer;var clasif3:integer):integer;
var
  k,l,x,esint,ps1,ss1,ts1:integer;
  gfavoreqss:vector;
  gcontrareqss:vector;
  ptoseqpos:vector;
  golesentreequiposss:Tmatriz32;
  puntosentreequiposss:Tmatriz32;
  vecl1:vector4;

begin
  if (ptoseqpo[Clasif1]=ptoseqpo[Clasif2]) and (ptoseqpo[Clasif1]=ptoseqpo[Clasif3]) then
    begin
      {Esto lo hacemos para decir que si van empatados todos a puntos, no habra colision}
      x:=0;
      intrascendente2:=x;
    end
  else
    begin
      esint:=1;
      k:=0;
      while(k<10) and (esint=1) do
        begin
          l:=0;
          while(l<10) and (esint=1) do
            begin
              gfavoreqss:=gfavoreq;
              gcontrareqss:=gcontrareq;
              ptoseqpos:=ptoseqpo;
              golesentreequiposss:=golesentreequipos;
              puntosentreequiposss:=puntosentreequipos;
              simpartidocol(Home[ir,6],Away[ir,6],k,l,gfavoreqss,gcontrareqss,ptoseqpos,golesentreequiposss,
              puntosentreequiposss);
              vecl1:=Clasificacion(ir,ptoseqpos,gfavoreqss,gcontrareqss,
              golesentreequiposss,puntosentreequiposss,Eq); {Simulamos el segundo}
              ps1:=vecl1[1];
              ss1:=vecl1[2];
              ts1:=vecl1[3];
              if ((Away[ir,6]=clasif1) and (Home[ir,6]=clasif2)) or ((Away[ir,6]=clasif2)
              and (Home[ir,6]=clasif1))
              then
                {1-2}
                begin
                  if (ts1=Away[ir,6]) or (ts1=Home[ir,6]) then
                    begin
                      esint:=0;
                      x:=0;
                    end
                  else esint:=1;
                    end
                else if (clasif1=Home[ir,6]) and (clasif3=Away[ir,6]) then {1-3}
                  begin
                    if (Away[ir,6]=ss1) or (Away[ir,6]=ps1) or (Home[ir,6]=ts1) then
                      begin
                        esint:=0;
                        x:=0;
                      end
                    else esint:=1;
                      end
                else if (clasif1=Away[ir,6]) and (clasif3=Home[ir,6]) then
                  begin
                    if (Home[ir,6]=ss1) or (Home[ir,6]=ps1) or (Away[ir,6]=ts1) then
                      begin
                        esint:=0;
                        x:=0;
                      end
                    else esint:=1;
                      end
                else if (clasif2=Home[ir,6]) and (clasif3=Away[ir,6]) then {2-3}
                  begin
                    if ((Away[ir,6]=ss1) and (Home[ir,6]=ts1)) or ((Away[ir,6]=ps1) and
                    (Home[ir,6]=ts1)) then
                      begin
                        esint:=0;
                        x:=0;
                      end
                    else esint:=1;
                      end
                else if (clasif2=Away[ir,6]) and (clasif3=Home[ir,6]) then
                  begin
                    if ((Home[ir,6]=ss1) and (Away[ir,6]=ts1)) or ((Home[ir,6]=ps1) and
                    (Away[ir,6]=ts1)) then
                      begin
                        esint:=0;
                        x:=0;
                      end
                    else esint:=1;
                      end
                else esint:=1;
                  l:=l+1;
                end;
                k:=k+1;
              end;
              if esint=1 then x:=1;
            end
          end
        end
      end
    end
  end
end

```

```

        intrascendente2:=x;
    end;

end;

function intrascendente1(ir:integer;Home:jornada;Away:jornada;Eq:Tmatriz;var gfavoreq:vector;
var gcontrareq: vector;var ptoseqpo:vector;var golesentreequipos:Tmatriz32;
var puntosentreequipos:Tmatriz32;var clasif1:integer;var clasif2:integer;var clasif3:integer):integer;
var
    k,l,x,ps1:integer;
    gfavoreqss:vector;
    gcontrareqss:vector;
    ptoseqposs:vector;
    golesentreequiposs:Tmatriz32;
    puntosentreequiposs:Tmatriz32;
    vecl1:vector4;
begin
    if (ptoseqpo[Clasif1]=ptoseqpo[Clasif2]) and (ptoseqpo[Clasif1]=ptoseqpo[Clasif3]) then
        begin
            {Esto lo hacemos para decir que si van empatados todos a puntos, no habra colision}
            x:=0;
            intrascendente1:=x;
        end
    else
        begin
            x:=1;
            k:=0;
            while(k<10) and (x=1) do
                begin
                    l:=0;
                    while(l<10) and (x=1) do
                        begin
                            gfavoreqss:=gfavoreq;
                            gcontrareqss:=gcontrareq;
                            ptoseqposs:=ptoseqpo;
                            golesentreequiposs:=golesentreequipos;
                            puntosentreequiposs:=puntosentreequipos;
                            simpartidocol(Home[ir,6],Away[ir,6],k,l,gfavoreqss,gcontrareqss,ptoseqposs,golesentreequiposs,
                            puntosentreequiposs);
                            vecl1:=Clasificacion(ir,ptoseqposs,gfavoreqss,gcontrareqss,
                            golesentreequiposs,puntosentreequiposs,Eq); {Simulamos el segundo}
                            ps1:=vecl1[1];
                            if ((Away[ir,6]=clasif1) and (Home[ir,6]=clasif2)) or ((Away[ir,6]=clasif2)
                            and (Home[ir,6]=clasif1)) then {1-2}
                                begin
                                    if ps1<>clasif1 then x:=0
                                    else x:=1;
                                    end
                                else if (clasif1=Home[ir,6]) and (clasif3=Away[ir,6]) then {1-3}
                                    begin
                                        if ps1<>clasif1 then x:=0
                                        else x:=1;
                                        end
                                    else if (clasif1=Away[ir,6]) and (clasif3=Home[ir,6]) then
                                        begin
                                            if ps1<>clasif1 then x:=0
                                            else x:=1;
                                            end
                                        else if (clasif2=Home[ir,6]) and (clasif3=Away[ir,6]) then {2-3}
                                        begin
                                            if ps1<>clasif1 then x:=0
                                            else x:=1;
                                            end
                                        else if (clasif2=Away[ir,6]) and (clasif3=Home[ir,6]) then
                                        begin
                                            if ps1<>clasif1 then x:=0
                                            else x:=1;
                                            end;
                                        l:=l+1;
                                        end;
                                    k:=k+1;
                                end;
                            intrascendente1:=x;
                        end;
                    end;
                end;
            end;
        end;
    end;

end;

function colision1(ir:integer;Home:jornada;Away:jornada{;Noplay:jornada};Eq:Tmatriz;
var gfavoreq:vector;var gcontrareq: vector;var ptoseqpo:vector;var golesentreequipos:Tmatriz32;
var puntosentreequipos:Tmatriz32;var clasif1:integer;var clasif2:integer;var clasif3:integer):integer;
var
    ii,jj,k,l,x,xx,escol1,ss,ts,ps1:integer;
    gfavoreqss,gfavoreqss1:vector;
    gcontrareqss,gcontrareqss1:vector;
    ptoseqposs,ptoseqposs1:vector;
    golesentreequiposs,golesentreequiposs1:Tmatriz32;
    puntosentreequiposs,puntosentreequiposs1:Tmatriz32;
    vecl,vecl1:vector4;
begin
    if ((clasif3=Away[ir,6]) and (clasif2=Home[ir,6])) or ((clasif3=Home[ir,6]) and (clasif2=Away[ir,6])) then
        begin
            {Suponemos que juegan 1 contra 2 y 3 contra 4, lo primero que hace es que habra colision si juegan el tercero contra el cuarto}
            {Recorremos los resultados de este partido para ver si tendrían opciones, inicializamos con xx=0, si
            tienen, empezamos el programa, si no pueden pasar ninguno de los 2, no habrá colision}
            k:=0;
            escol1:=1;
            while(k<2) and (escol1=1) do
                begin
                    l:=0;
                    while(l<2) and (escol1=1) do
                        begin

```

```

                xx:=1;
                gfavoreqss1:=gfavoreq;
                gcontrareqss1:=gcontrareq;
                ptoseqposs1:=ptoseqpo;
                golesentreequiposs1:=golesentreequipos;
                puntosentreequiposs1:=puntosentreequipos;

                simpartidocol(Home[ir,6],Away[ir,6],k,l,gfavoreqss1,gcontrareqss1,ptoseqposs1,golesentreequiposs1,
                puntosentreequiposs1);
                vecl:=Clasificacion(ir,ptoseqposs1,gfavoreqss1,gcontrareqss1,
                golesentreequiposs1,puntosentreequiposs1,Eq);
                ss:=vecl[2];
                ts:=vecl[3];
                if (clasif3=ss) or (clasif3=ts) then
{Si existe algun resultado que hace que el que iba tercero o cuarto puede quedar entre los 2 primeros, empezamos}
                begin
                    escol1:=1;
                    xx:=1;
                    end
                else
                begin
                    escol1:=0;
                    xx:=0;
                    end;
                l:=l+1;
                end;
                k:=k+1;
            end;
            if xx=1 then
                begin
                    x:=0;
                    ii:=0;
                    while (ii<2) and (x=0) do
                        begin
                            jj:=0;
                            while(jj<2) and (x=0) do
                                begin
                                    gfavoreqss:=gfavoreq;                                {Recorremos el primer partido}
                                    gcontrareqss:=gcontrareq;
                                    ptoseqposs:=ptoseqpo;
                                    golesentreequiposs:=golesentreequipos;
                                    puntosentreequiposs:=puntosentreequipos;
                                    simpartidocol(Home[ir,6],Away[ir,6],ii,jj,gfavoreqss,gcontrareqss,
                                    ptoseqposs,golesentreequiposs,puntosentreequiposs);
{Lo uso pa comprobar} vecl1:=Clasificacion(ir,ptoseqposs,gfavoreqss,gcontrareqss,golesentreequiposs,puntosentreequiposs,Eq);
                                end;
                            end;
                            ii:=ii+1;
                        end;
                    end;
                    else x:=0;
                    colision1:=x;
                end
            else
                begin
                    x:=0;
                    colision1:=x;
                end;
            end;
        end;
end;

{#####PROGRAMA PRINCIPAL}

begin
    randomize();
    write('Introduce tipo de sorteo Fifa[1], Guyon [2]: ');
    read(sorteo);
    while (sorteo<>1) and (sorteo<>2) do
        begin
            write('Introduzca bien el numero: ');
            read(sorteo);
        end;
    write('Introduce , IdaVuelta[1] PartidoUnico[2]: ');
    read(partidos);
    while (partidos<>1) and (partidos<>2) do
        begin
            write('Introduzca bien el numero: ');
            read(partidos);
        end ;
    if partidos=1 then
        begin
            write('Introduce un orden de la jornada que este entre 1 y 6: ');
            read(numero);
            while (numero<>1) and (numero<>2) and (numero<>3) and (numero<>4) and (numero<>5) and (numero<>6) do
                begin
                    write('Introduzca bien el numero: ');
                    read(numero);
                end ;
            end;
        if partidos=2 then
            begin
                write('Introduce un orden de la jornada que este entre 1 y 3: ');
                read(numero1);
                while (numero1<>1) and (numero1<>2) and (numero1<>3) do

```

```

begin
write('Introduzca bien el numero: ');
read(numero1);
end;
end;
write('Numero de clasificados , 1Clasificado[1] 2Clasificados[2]: ');
read(clasificados);
while (clasificados<>1) and (clasificados<>2) do
begin
write('Introduzca bien el numero: ');
read(clasificados);
end;
write('Introduce el tipo de orden de los equipos, Ranking[1] OrdenSimulacion[2]: ');
read(orden);
while (orden<>1) and (orden<>2) do
begin
write('Introduzca bien el numero: ');
read(orden);
end;
if orden =1 then
begin
assign(q,'DEFINITIVOLandasA148PoissonFiltrado.txt');
assign(f,'DEFINITIVOLandasA48PoissonFiltrado.txt');
assign(u,'DEFINITIVOLandasA48UNICOPoissonFiltrado.txt');
assign(resul,'resultados.txt');
end;
if orden =2 then
begin
assign(q,'DEFINITIVOSIMULACIONLandasA148PoissonFiltrado.txt');
assign(f,'DEFINITIVOSIMULACIONLandasA48PoissonFiltrado.txt');
assign(u,'DEFINITIVOSIMULACIONLandasA48UNICOPoissonFiltrado.txt');
assign(resul,'resultados.txt');
end;
end;

reset(q);
reset(f);
reset(u);
reset(resul);
rewrite(resul);
posicion:=1;
posicion:=1;
posicionu:=1;
while not Eof(q) do
begin
readLn(q,a);
misdatos1[posicion]:=a;
posicion:=posicion+1;

end;
while not Eof(f) do
begin
readLn(f,a);
misdatos[posicion]:=a;
posicion:=posicion+1;

end;
while not Eof(u) do
begin
readLn(u,a);
misdatosu[posicionu]:=a;
posicionu:=posicionu+1;

end;
for i:=1 to 48 do
begin
for j:=1 to 48 do matriz1[i,j]:=misdatos1[48*(i-1)+j];
end;

for i:=1 to 48 do
begin
for j:=1 to 48 do matriz[i,j]:=misdatos[48*(i-1)+j];
end;
end;
for i:=1 to 48 do
begin
for j:=1 to 48 do matrizuni48[i,j]:=misdatosu[48*(i-1)+j];
end;
end;
{SORTEO-----}
for p:=1 to 48 do
begin
vclas[p]:=0;
end;
intentos:=0;
intentos1:=0;
for s:=1 to 1000 do
begin
if sorteo=1 then
begin
for t1:=1 to 16 do lista1[t1]:=t1;
for t2:=1 to 16 do lista2[t2]:=t2+16;
for t3:=1 to 16 do lista3[t3]:=t3+32;
for t1:=1 to 16 do
begin
r1:=random(17-t1)+1;
rr1:=lista1[r1];
g[t1]:=rr1;
lista1[r1]:=lista1[17-t1];
end;
for t2:=1 to 16 do

```

```

begin
r2:=random(17-t2)+1;
rr2:=lista2[r2];
g[t2+16]:=rr2;
lista2[r2]:=lista2[17-t2];
end;
for t3:=1 to 16 do
begin
r3:=random(17-t3)+1;
rr3:=lista3[r3];
g[t3+32]:=rr3;
lista3[r3]:=lista3[17-t3];
end;
end;
if sorteo=2 then
begin
for t1:=1 to 8 do lista1[t1]:=t1;
for t2:=1 to 8 do lista2[t2]:=t2+24;
for t3:=1 to 8 do lista3[t3]:=t3+32;
for t1:=1 to 8 do
begin
r1:=random(9-t1)+1;
rr1:=lista1[r1];
g[t1]:=rr1;
lista1[r1]:=lista1[9-t1];
end;
for t2:=1 to 8 do
begin
r2:=random(9-t2)+1;
rr2:=lista2[r2];
g[t2+24]:=rr2;
lista2[r2]:=lista2[9-t2];
end;
for t3:=1 to 8 do
begin
r3:=random(9-t3)+1;
rr3:=lista3[r3];
g[t3+32]:=rr3;
lista3[r3]:=lista3[9-t3];
end;
for t4:=1 to 8 do lista4[t4]:=t4+8;
for t5:=1 to 8 do lista5[t5]:=t5+16;
for t6:=1 to 8 do lista6[t6]:=t6+40;
for t4:=1 to 8 do
begin
r4:=random(9-t4)+1;
rr4:=lista4[r4];
g[t4+8]:=rr4;
lista4[r4]:=lista4[9-t4];
end;
for t5:=1 to 8 do
begin
r5:=random(9-t5)+1;
rr5:=lista5[r5];
g[t5+16]:=rr5;
lista5[r5]:=lista5[9-t5];
end;
for t6:=1 to 8 do
begin
r6:=random(9-t6)+1;
rr6:=lista6[r6];
g[t6+40]:=rr6;
lista6[r6]:=lista6[9-t6];
end;
end;
{Hasta aqui el sorteo-----}
{Defino a los equipos como equipo[i,j] con i el grupo y j el bombo al que pertenecen}
for i:=1 to 16 do
for j:=1 to 4 do equipo[i,j]:=g[16*(j-1)+i];
for r:=1 to 48 do
begin
puntosequipo[r]:=0;
golesafavorequipo[r]:=0;
golesencontraequipo[r]:=0;
end;
for i:=1 to 48 do
begin
for j:=1 to 48 do golesenfrentamientos[i,j]:=0;
end;
for i:=1 to 48 do
begin
for j:=1 to 48 do puntosenfrentamientos[i,j]:=0;
end;
if partidos=1 then
begin
for k:=1 to 5 do
begin
case k of
{JORNADA1}
1 : begin
for i:=1 to 16 do
begin
if numero=1 then
begin
casa[i,k]:=equipo[i,1];
fuera[i,k]:=equipo[i,2];

```

```

simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,1],equipo[i,2]]
,matriz[equipo[i,2],equipo[i,1]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end;
if numero=2 then
begin
casa[i,k]:=equipo[i,1];
fuera[i,k]:=equipo[i,2];
simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,1],equipo[i,2]]
,matriz[equipo[i,2],equipo[i,1]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end;
if numero=3 then
begin
casa[i,k]:=equipo[i,1];
fuera[i,k]:=equipo[i,3];
simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,1],equipo[i,3]]
,matriz[equipo[i,3],equipo[i,1]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end;
if numero=4 then
begin
casa[i,k]:=equipo[i,1];
fuera[i,k]:=equipo[i,2];
simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,1],equipo[i,2]]
,matriz[equipo[i,2],equipo[i,1]],golesafavorequipo,
golesencontraequipo,puntosequipo,
golesenfrentamientos,puntosenfrentamientos);
end;
if numero=5 then
begin
casa[i,k]:=equipo[i,1];
fuera[i,k]:=equipo[i,2];
simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,1],equipo[i,2]]
,matriz[equipo[i,2],equipo[i,1]],golesafavorequipo,
golesencontraequipo, puntosequipo,
golesenfrentamientos,puntosenfrentamientos);
end;
if numero=6 then
begin
casa[i,k]:=equipo[i,1];
fuera[i,k]:=equipo[i,3];
simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,1],equipo[i,3]]
,matriz[equipo[i,3],equipo[i,1]],golesafavorequipo,
golesencontraequipo, puntosequipo,
golesenfrentamientos,puntosenfrentamientos);
end;
end;
end;
{JORNADA2}
2 : begin
for i:=1 to 16 do
begin
if numero=1 then
begin
casa[i,k]:=equipo[i,2];
fuera[i,k]:=equipo[i,3];
simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,2],equipo[i,3]]
,matriz[equipo[i,3],equipo[i,2]],golesafavorequipo,
golesencontraequipo, puntosequipo,
golesenfrentamientos,puntosenfrentamientos);
end;
if numero=2 then
begin
casa[i,k]:=equipo[i,1];
fuera[i,k]:=equipo[i,3];
simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,1],equipo[i,3]]
,matriz[equipo[i,3],equipo[i,1]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end;
if numero=3 then
begin
casa[i,k]:=equipo[i,1];
fuera[i,k]:=equipo[i,2];
simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,1],equipo[i,2]]
,matriz[equipo[i,2],equipo[i,1]],golesafavorequipo,
golesencontraequipo, puntosequipo,
golesenfrentamientos,puntosenfrentamientos);
end;
if numero=4 then
begin
casa[i,k]:=equipo[i,3];
fuera[i,k]:=equipo[i,2];
simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,2],equipo[i,3]]
,matriz1[equipo[i,3],equipo[i,2]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
end;
if numero=5 then
begin
casa[i,k]:=equipo[i,3];
fuera[i,k]:=equipo[i,1];
simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,1],equipo[i,3]]
,matriz1[equipo[i,3],equipo[i,1]],golesafavorequipo,
golesencontraequipo,puntosequipo,
golesenfrentamientos,puntosenfrentamientos);
end;
if numero=6 then
begin
casa[i,k]:=equipo[i,2];

```



```

        fuera[i,k]:=equipo[i,1];
        simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,1],equipo[i,2]],
            matriz1[equipo[i,2],equipo[i,1]],golesafavorequipo,golesencontraequipo,
            puntosequipo,golesenfrentamientos,puntosenfrentamientos);
        end;
    end;
end;

{JORNADA3}
3 : begin
    for i:=1 to 16 do
        begin
            if numero=1 then
                begin
                    casa[i,k]:=equipo[i,3];
                    fuera[i,k]:=equipo[i,1];
                    simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,1],equipo[i,3]],
                        matriz1[equipo[i,3],equipo[i,1]],golesafavorequipo,golesencontraequipo,
                        puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                    end;
                if numero=2 then
                    begin
                        casa[i,k]:=equipo[i,3];
                        fuera[i,k]:=equipo[i,2];
                        simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,2],equipo[i,3]],
                            matriz1[equipo[i,3],equipo[i,2]],golesafavorequipo,golesencontraequipo,
                            puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                        end;
                    if numero=3 then
                        begin
                            casa[i,k]:=equipo[i,3];
                            fuera[i,k]:=equipo[i,2];
                            simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,2],equipo[i,3]],
                                matriz1[equipo[i,3],equipo[i,2]],golesafavorequipo,golesencontraequipo,
                                puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                            end;
                    if numero=4 then
                        begin
                            casa[i,k]:=equipo[i,3];
                            fuera[i,k]:=equipo[i,1];
                            simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,1],equipo[i,3]],
                                matriz1[equipo[i,3],equipo[i,1]],golesafavorequipo,golesencontraequipo,
                                puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                            end;
                    if numero=5 then
                        begin
                            casa[i,k]:=equipo[i,3];
                            fuera[i,k]:=equipo[i,2];
                            simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,2],equipo[i,3]],
                                matriz1[equipo[i,3],equipo[i,2]],golesafavorequipo,golesencontraequipo,
                                puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                            end;
                    if numero=6 then
                        begin
                            casa[i,k]:=equipo[i,3];
                            fuera[i,k]:=equipo[i,2];
                            simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,2],equipo[i,3]],
                                matriz1[equipo[i,3],equipo[i,2]],golesafavorequipo,golesencontraequipo,
                                puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                            end;
                    end;
                end;
            end;
        end;
    end;

{JORNADA4}
4 : begin
    for i:=1 to 16 do
        begin
            if numero=1 then
                begin
                    casa[i,k]:=equipo[i,1];
                    fuera[i,k]:=equipo[i,3];
                    simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,1],equipo[i,3]]
                        ,matriz[equipo[i,3],equipo[i,1]],golesafavorequipo,golesencontraequipo,
                        puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                    end;
                if numero=2 then
                    begin
                        casa[i,k]:=equipo[i,2];
                        fuera[i,k]:=equipo[i,3];
                        simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,2],equipo[i,3]]
                            ,matriz[equipo[i,3],equipo[i,2]],golesafavorequipo,golesencontraequipo,
                            puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                        end;
                    if numero=3 then
                        begin
                            casa[i,k]:=equipo[i,2];
                            fuera[i,k]:=equipo[i,3];
                            simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,2],equipo[i,3]]
                                ,matriz[equipo[i,3],equipo[i,2]],golesafavorequipo,golesencontraequipo,
                                puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                            end;
                    if numero=4 then
                        begin
                            casa[i,k]:=equipo[i,1];
                            fuera[i,k]:=equipo[i,3];
                            simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,1],equipo[i,3]]
                                ,matriz[equipo[i,3],equipo[i,1]],golesafavorequipo,golesencontraequipo,
                                puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                            end;
                    end;
                end;
            end;
        end;
    end;
end;

```

```

        puntosequipo,golesenfrentamientos,puntosenfrentamientos);
    end;
if numero=5 then
    begin
        casa[i,k]:=equipo[i,2];
        fuera[i,k]:=equipo[i,3];
        simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,2],equipo[i,3]]
            ,matriz[equipo[i,3],equipo[i,2]],golesafavorequipo,golesencontraequipo,
            puntosequipo,golesenfrentamientos,puntosenfrentamientos);
    end;
if numero=6 then
    begin
        casa[i,k]:=equipo[i,2];
        fuera[i,k]:=equipo[i,3];
        simularpartido(casa[i,k],fuera[i,k],matriz[equipo[i,2],equipo[i,3]]
            ,matriz[equipo[i,3],equipo[i,2]],golesafavorequipo,golesencontraequipo,
            puntosequipo,golesenfrentamientos,puntosenfrentamientos);
    end;
end;

end;

{JORNADA5}
5 : begin
    for i:=1 to 16 do

        begin
            if numero=1 then
                begin
                    casa[i,k]:=equipo[i,2];
                    fuera[i,k]:=equipo[i,1];
                    simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,1],equipo[i,2]]
                        ,matriz1[equipo[i,2],equipo[i,1]],golesafavorequipo,golesencontraequipo,
                        puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                end;
            if numero=2 then
                begin
                    casa[i,k]:=equipo[i,2];
                    fuera[i,k]:=equipo[i,1];
                    simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,1],equipo[i,2]]
                        ,matriz1[equipo[i,2],equipo[i,1]],golesafavorequipo,golesencontraequipo,
                        puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                end;
            if numero=3 then
                begin
                    casa[i,k]:=equipo[i,3];
                    fuera[i,k]:=equipo[i,1];
                    simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,1],equipo[i,3]]
                        ,matriz1[equipo[i,3],equipo[i,1]],golesafavorequipo,golesencontraequipo,
                        puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                end;
            if numero=4 then
                begin
                    casa[i,k]:=equipo[i,2];
                    fuera[i,k]:=equipo[i,1];
                    simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,1],equipo[i,2]]
                        ,matriz1[equipo[i,2],equipo[i,1]],golesafavorequipo,golesencontraequipo,
                        puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                end;
            if numero=5 then
                begin
                    casa[i,k]:=equipo[i,2];
                    fuera[i,k]:=equipo[i,1];
                    simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,1],equipo[i,2]]
                        ,matriz1[equipo[i,2],equipo[i,1]],golesafavorequipo,golesencontraequipo,
                        puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                end;
            if numero=6 then
                begin
                    casa[i,k]:=equipo[i,3];
                    fuera[i,k]:=equipo[i,1];
                    simularpartido(casa[i,k],fuera[i,k],matriz1[equipo[i,1],equipo[i,3]]
                        ,matriz1[equipo[i,3],equipo[i,1]],golesafavorequipo,golesencontraequipo,
                        puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                end;
        end;

    end;
end;
end;
for i:=1 to 16 do

    begin
        if numero=1 then
            begin
                casa[i,6]:=equipo[i,3];
                fuera[i,6]:=equipo[i,2];
                nojuega[i,6]:=equipo[i,1];
            end
        else if numero=2 then
            begin
                casa[i,6]:=equipo[i,3];
                fuera[i,6]:=equipo[i,1];
                nojuega[i,6]:=equipo[i,2];
            end
        else if numero=3 then
            begin
                casa[i,6]:=equipo[i,2];
                fuera[i,6]:=equipo[i,1];
                nojuega[i,6]:=equipo[i,3];
            end
        end;
    end;
end;

```

```

else if numero=4 then
begin
casa[i,6]:=equipo[i,2];
fuera[i,6]:=equipo[i,3];
nojuega[i,6]:=equipo[i,1];
end
else if numero=5 then
begin
casa[i,6]:=equipo[i,1];
fuera[i,6]:=equipo[i,3];
nojuega[i,6]:=equipo[i,2];
end
else if numero=6 then
begin
casa[i,6]:=equipo[i,1];
fuera[i,6]:=equipo[i,2];
nojuega[i,6]:=equipo[i,3];
end
end;

if partidos=2 then
begin
for k:=1 to 2 do
begin
case k of
{JORNADA1}
1 : begin
for i:=1 to 16 do
begin
if numero1=1 then {14-23}
begin
casa[i,k]:=equipo[i,1];
fuera[i,k]:=equipo[i,2];
simularpartido(casa[i,k],fuera[i,k],matrizuni48[equipo[i,1],equipo[i,2]]
,matrizuni48[equipo[i,2],equipo[i,1]],golesafavorequipo,
golesencontraequipo,puntosequipo,golesenfrentamientos,
puntosenfrentamientos);
end
else if numero1=2 then
begin
casa[i,k]:=equipo[i,1];
fuera[i,k]:=equipo[i,3];
simularpartido(casa[i,k],fuera[i,k],matrizuni48[equipo[i,1],equipo[i,3]]
,matrizuni48[equipo[i,3],equipo[i,1]],golesafavorequipo,
golesencontraequipo, puntosequipo,golesenfrentamientos,
puntosenfrentamientos);
end
else if numero1=3 then
begin
casa[i,k]:=equipo[i,1];
fuera[i,k]:=equipo[i,2];
simularpartido(casa[i,k],fuera[i,k],matrizuni48[equipo[i,1],equipo[i,2]]
,matrizuni48[equipo[i,2],equipo[i,1]],golesafavorequipo,
golesencontraequipo, puntosequipo,golesenfrentamientos,
puntosenfrentamientos);
end
end;
end;
{JORNADA2}
2 : begin
for i:=1 to 16 do
begin
if numero1=1 then {14-23}
begin
casa[i,k]:=equipo[i,2];
fuera[i,k]:=equipo[i,3];
simularpartido(casa[i,k],fuera[i,k],matrizuni48[equipo[i,2],equipo[i,3]]
,matrizuni48[equipo[i,3],equipo[i,2]],golesafavorequipo,
golesencontraequipo, puntosequipo,golesenfrentamientos,
puntosenfrentamientos);
end
else if numero1=2 then
begin
casa[i,k]:=equipo[i,2];
fuera[i,k]:=equipo[i,3];
simularpartido(casa[i,k],fuera[i,k],matrizuni48[equipo[i,2],equipo[i,3]]
,matrizuni48[equipo[i,3],equipo[i,2]],golesafavorequipo,
golesencontraequipo, puntosequipo,golesenfrentamientos,
puntosenfrentamientos);
end
else if numero1=3 then
begin
casa[i,k]:=equipo[i,1];
fuera[i,k]:=equipo[i,3];
simularpartido(casa[i,k],fuera[i,k],matrizuni48[equipo[i,1],equipo[i,3]]
,matrizuni48[equipo[i,3],equipo[i,1]],golesafavorequipo,
golesencontraequipo, puntosequipo,golesenfrentamientos,
puntosenfrentamientos);
end
end;
end;
end;
end;
end;
end;
end;

```

```

for i:=1 to 16 do
    begin
    if numero1=1 then
        begin
            casa[i,6]:=equipo[i,3];
            fuera[i,6]:=equipo[i,1];
            nojuega[i,6]:=equipo[i,2];
        end
    else if numero1=2 then
        begin
            casa[i,6]:=equipo[i,2];
            fuera[i,6]:=equipo[i,1];
            nojuega[i,6]:=equipo[i,3];
        end
    else if numero1=3 then
        begin
            casa[i,6]:=equipo[i,2];
            fuera[i,6]:=equipo[i,3];
            nojuega[i,6]:=equipo[i,1];
        end
    end;

    end;

{Una vez jugadas las 5 primeras jornadas, queremos ver cuantos partidos intrascendentes y
de colision hay para ello empezamos inicializando estos partidos en 0}
numpint:=0;
numpcol:=0;
numpint1:=0;
numpcol1:=0;

if clasificados=2 then
    begin
    for i:=1 to 16 do
        begin
        vect4:=Clasificacion(i,puntosequipo,golesafavorequipo,golesencontraequipo,golesenfrentamientos,
            puntosenfrentamientos,equipo);
        numpcol:=numpcol+colision(i,casa,fuera,nojuega,equipo,golesafavorequipo,golesencontraequipo,
            puntosenfrentamientos,puntosenfrentamientos,vect4[1],vect4[2],vect4[3]);
        numpint:= numpint+intrascendente2(i,casa,fuera,equipo,golesafavorequipo,golesencontraequipo,
            puntosequipo,golesenfrentamientos,puntosenfrentamientos,vect4[1],vect4[2],vect4[3]);
        end;
        intentos:=intentos+numpint;
        intentos1:=intentos1+numpcol;
    end;

if clasificados=1 then
    begin
    for i:=1 to 16 do
        begin
        vect4:=Clasificacion(i,puntosequipo,golesafavorequipo,golesencontraequipo,golesenfrentamientos,
            puntosenfrentamientos,equipo);
        numpcol1:=numpcol1+colision1(i,casa,fuera,equipo,golesafavorequipo,golesencontraequipo,
            puntosequipo,golesenfrentamientos,puntosenfrentamientos,vect4[1],vect4[2],vect4[3]);
        numpint1:= numpint1+intrascendente1(i,casa,fuera,equipo,golesafavorequipo,golesencontraequipo,
            puntosequipo,golesenfrentamientos,puntosenfrentamientos,vect4[1],vect4[2],vect4[3]);
        end;
        intentos:=intentos+numpint1;
        intentos1:=intentos1+numpcol1;
    end;
}JORNADA 6}

{Simulamos la ultima jornada y con esto veremos que equipos son los que se han clasificado.}
if partidos=1 then
    begin
    for i:=1 to 16 do
        begin
        if numero=1 then
            begin
                casa[i,6]:=equipo[i,3];
                fuera[i,6]:=equipo[i,2];
                simularpartido(casa[i,6],fuera[i,6],matriz1[equipo[i,2],equipo[i,3]]
                    ,matriz1[equipo[i,3],equipo[i,2]],golesafavorequipo,golesencontraequipo,
                    puntosequipo,golesenfrentamientos,puntosenfrentamientos);
            end;
        if numero=2 then
            begin
                casa[i,6]:=equipo[i,3];
                fuera[i,6]:=equipo[i,1];
                simularpartido(casa[i,6],fuera[i,6],matriz1[equipo[i,1],equipo[i,3]]
                    ,matriz1[equipo[i,3],equipo[i,1]],golesafavorequipo,golesencontraequipo,
                    puntosequipo,golesenfrentamientos,puntosenfrentamientos);
            end;
        if numero=3 then
            begin
                casa[i,6]:=equipo[i,2];
                fuera[i,6]:=equipo[i,1];
                simularpartido(casa[i,6],fuera[i,6],matriz1[equipo[i,1],equipo[i,2]]
                    ,matriz1[equipo[i,2],equipo[i,1]],golesafavorequipo,golesencontraequipo,
                    puntosequipo,golesenfrentamientos,puntosenfrentamientos);
            end;
        if numero=4 then
            begin
                casa[i,6]:=equipo[i,2];
                fuera[i,6]:=equipo[i,3];
                simularpartido(casa[i,6],fuera[i,6],matriz[equipo[i,2],equipo[i,3]]
                    ,matriz[equipo[i,3],equipo[i,2]],golesafavorequipo,golesencontraequipo,
                    puntosequipo,golesenfrentamientos,puntosenfrentamientos);
            end;
        end;
    end;
end;

```


Liga entre todos los equipos

De la misma forma que hemos hecho con los grupos de 4, lo que se pretende es establecer un nuevo orden de la fuerza de los equipos, basándonos en los puntos que obtenga cada uno en una liga en la que se enfrenten todos contra todos

```

program ligatodos48equipos;
uses crt,math;
type
  vector=array[1.. 48]of integer;
  Tmatriz32=array[1..48,1..48] of integer;
var
  i,j,r:integer;
  puntosequipo,golesafavorequipo,golesencontraequipo:array[1..48] of integer;
  a:real;
  golesenfrentamientos,puntosenfrentamientos:Tmatriz32;
  misdatos,misdatos1:array[1..2304] of real;   {Los tengo que definir como reales ya que coge valores reales}
  matriz,matriz1:array[1..48,1..48] of real;
  posicion,posicion1:integer;
  f,q,Resul:text;
  p,s:integer;
  equipos:vector;
  {Definimos la funcion densidad de Poisson y para ello defino antes el factorial}
  function factorial( num : integer) : integer;
  begin
    if num = 1 then factorial := 1
    else if num = 0 then factorial := 1
    else factorial := num * factorial( num-1 );
  end;
  function Poisson(landa:real; x:integer):real;
  begin
    Poisson:=Exp(-landa)*power(landa,x)/factorial(x);
  end;
  {Simulo una Poisson y que me diga cuantos goles marca el equipo local tomando un valor aleatorio y y mirando donde
  cae en la probabilidad acumulada}
  function simularPoissonL(landa:real):integer;
  var i:integer;
      y:real;
      valor:array[1..10] of real;
      pacum:array[1..10] of real;
      glocal:integer;
  begin
    for i:=0 to 9 do
      begin
        valor[i+1]:=Poisson(landa,i);
        if i=0 then pacum[i+1]:=valor[i+1]
        else pacum[i+1]:=pacum[i]+valor[i+1];
        y:=random();
        glocal:=0;
        while (pacum[glocal+1]<y) and (glocal<=10) do glocal:=glocal+1;
      end;
    simularPoissonL:=glocal;
  end;
  {Simulo una Poisson y que me diga cuantos goles marca el equipo visitante tomando un valor aleatorio y y mirando
  donde cae en la probabilidad acumulada}
  function simularPoissonV(landa:real):integer;
  var
    i:integer;
    y:real;
    valor:array[1..10] of real;
    pacum:array[1..10] of real;
    gvisitante:integer;
  begin
    for i:=0 to 9 do
      begin
        valor[i+1]:=Poisson(landa,i);
        if i=0 then pacum[i+1]:=valor[i+1]
        else pacum[i+1]:=pacum[i]+valor[i+1];
        y:=random();
        gvisitante:=0;
        while (pacum[gvisitante+1]<y) and (gvisitante<=10) do gvisitante:=gvisitante+1;
      end;
    simularPoissonV:=gvisitante;
  end;

  {Hago un procedimiento que meto a los equipos que juegan y el valor del landa correspondiente, y se me van añadiendo
  los valores de puntos, goles a favor y goles en contra que tengo de la jornada anterior}
  procedure simularpartido(a:integer;b:integer;c:real;d:real;var gfavoreq:vector;var gcontrareq: vector;
  var pposeqo:vector;var golesentreequipos:Tmatriz32;var puntosentreequipos:Tmatriz32);
  var
    golesfavora:integer;
    golescontraa:integer;
  begin
    golesfavora:=simularPoissonL(c);
    golescontraa:=simularPoissonV(d);
    gfavoreq[a]:=gfavoreq[a]+golesfavora;
    gcontrareq[a]:=gcontrareq[a]+golescontraa;
    gfavoreq[b]:=gfavoreq[b]+golescontraa;
    gcontrareq[b]:=gcontrareq[b]+golesfavora;
    if (golesfavora>golescontraa) then
      begin

```

```

    ptoseqpo[a]:= ptoseqpo[a]+3;
    puntosentreequipos[a,b]:=puntosentreequipos[a,b]+3;
end;
if (golesfavora<golescontraa) then
begin
    ptoseqpo[b]:= ptoseqpo[b]+3;
    puntosentreequipos[b,a]:=puntosentreequipos[b,a]+3;
end;
if(golesfavora=golescontraa) then
begin
    ptoseqpo[a]:= ptoseqpo[a]+1;
    ptoseqpo[b]:= ptoseqpo[b]+1;
    puntosentreequipos[a,b]:=puntosentreequipos[a,b]+1;
    puntosentreequipos[b,a]:=puntosentreequipos[b,a]+1;
end;
golesentreequipos[a,b]:=golesfavora+golesentreequipos[a,b];
golesentreequipos[b,a]:=golescontraa+golesentreequipos[b,a];
end;

#####PROGRAMA PRINCIPAL
begin
    randomize();
    assign(q,'DEFINITIVOLandasA148PoissonFiltrado.txt');
    assign(f,'DEFINITIVOLandasA48PoissonFiltrado.txt');
    assign(Resul,'Resultados.txt');
    reset(q);
    reset(f);
    reset(Resul);
    rewrite(Resul);
    posicion1:=1;
    posicion:=1;
    while not Eof(q) do
    begin
        readLn(q,a);
        misdatos1[posicion1]:=a;
        posicion1:=posicion1+1;
    end;
    while not Eof(f) do
    begin
        readLn(f,a);
        misdatos[posicion]:=a;
        posicion:=posicion+1;
    end;
    for i:=1 to 48 do
    begin
        for j:=1 to 48 do matriz1[i,j]:=misdatos1[48*(i-1)+j];
        end;

    for i:=1 to 48 do
    begin
        for j:=1 to 48 do matriz[i,j]:=misdatos[48*(i-1)+j] ;
        end;

    for r:=1 to 48 do
    begin
        puntosequipo[r]:=0;
        golesafavorequipo[r]:=0;
        golesencontraequipo[r]:=0;
    end;

    for i:=1 to 48 do
    begin
        for j:=1 to 48 do golesenfrentamientos[i,j]:=0;
        end;

    for i:=1 to 48 do
    begin
        for j:=1 to 48 do puntosenfrentamientos[i,j]:=0;
        end;

    for s:=1 to 10000 do
    begin
        for r:=1 to 48 do equipos[r]:=r;
        for i:=1 to 48 do
            for j:=1 to 48 do
                begin
                    if i<j then similarpartido(equipos[i],equipos[j],matriz[equipos[i],equipos[j]]
                    ,matriz[equipos[j],equipos[i]],golesafavorequipo,golesencontraequipo,
                    puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                    if i>j then similarpartido(equipos[j],equipos[i],matriz1[equipos[i],equipos[j]]
                    ,matriz1[equipos[j],equipos[i]],golesafavorequipo,golesencontraequipo,
                    puntosequipo,golesenfrentamientos,puntosenfrentamientos);
                end;
            end;
        end;
    end;
    {Sacamos el numero de veces que pasa cada equipo.}
    write('[' );
    for p:=1 to 48 do write(puntosequipo[p],',');
    writeln(')');
    for p:=1 to 48 do write(Resul,puntosequipo[p],', ');
    close(q);
    close(f);
    close(Resul);
    ReadKey;
end.

```

Número de ocasiones entre los 16 o 32 mejores

Como en el caso de los grupos de 4, vamos a hacer lo mismo para los de 3. Como en este caso estudiaremos tanto que haya 1 clasificado como 2, buscaremos obtener cuantas veces quedan por un lado entre los 16 mejores y otra entre los 32 y después sacar la correspondiente correlación con el número de veces que se clasifica cada equipo en el programa principal.

```

program mejores16y32grupos3;
uses crt,math;
type
  vector=array[1.. 48]of integer;
  Tmatriz32=array[1..48,1..48] of integer;
var
  i,j,r,aux,hh,x,clasificados:integer;
  puntosequipo,golesafavorequipo,golescontraequipo:array[1..48] of integer;
  a:real;
  golesenfrentamientos,puntosenfrentamientos:Tmatriz32;
  misdatos,misdatos1:array[1..2304] of real;
  matriz,matriz1:array[1..48,1..48] of real;
  posicion,posicion1:integer;
  f,q,u,Resul:text;
  p,s:integer;
  vclas,vectPuntos,vectPuntos1:vector;
  equipos:vector;

{Definimos la función densidad de Poisson y para ello defino antes el factorial}
function factorial( num : integer) : integer;
begin
  if num = 1 then factorial := 1
  else if num = 0 then factorial := 1
  else factorial := num * factorial( num-1 );
end;
function Poisson(landa:real; x:integer):real;
begin
  Poisson:=Exp(-landa)*power(landa,x)/factorial(x);
end;
{Simulo una Poisson, que me diga cuantos goles marca el equipo local tomando
un valor aleatorio y y mirando donde cae en la probabilidad acumulada}

function simularPoissonL(landa:real):integer;
var i:integer;
    y:real;
    valor:array[1..10] of real;
    pacum:array[1..10] of real;
    glocal:integer;
begin
for i:=0 to 9 do
  begin
  valor[i+1]:=Poisson(landa,i);
  if i=0 then pacum[i+1]:=valor[i+1]
  else pacum[i+1]:=pacum[i]+valor[i+1];
  y:=random();
  glocal:=0;
  while (pacum[glocal+1]<y) and (glocal<=10) do glocal:=glocal+1;
  end;
simularPoissonL:=glocal;
end;
{Simulo una Poisson y que me diga cuantos goles marca el equipo visitante
tomando un valor aleatorio y y mirando donde cae en la probabilidad acumulada}

function simularPoissonV(landa:real):integer;
var
  i:integer;
  y:real;
  valor:array[1..10] of real;
  pacum:array[1..10] of real;
  gvisitante:integer;
begin
for i:=0 to 9 do
  begin
  valor[i+1]:=Poisson(landa,i);
  if i=0 then pacum[i+1]:=valor[i+1]
  else pacum[i+1]:=pacum[i]+valor[i+1];
  y:=random();
  gvisitante:=0;
  while (pacum[gvisitante+1]<y) and (gvisitante<=10) do gvisitante:=gvisitante+1;
  end;
simularPoissonV:=gvisitante;
end;

{Hago un procedimiento que meto a los equipos que juegan y el valor del landa
correspondiente, y se me van añadiendo los valores de puntos, goles a favor
y goles en contra que tengo de la jornada anterior}

procedure simularpartido(a:integer;b:integer;c:real;d:real;var gfavoreq:vector;
var gcontrareq: vector;var ptoseqpo:vector;
var golesentreequipos:Tmatriz32;var puntosentreequipos:Tmatriz32);
var
  golesfavora:integer;
  golescontraa:integer;
begin
  golesfavora:=simularPoissonL(c);
  golescontraa:=simularPoissonV(d);
  gfavoreq[a]:=gfavoreq[a]+golesfavora;
  gcontrareq[a]:=gcontrareq[a]+golescontraa;

```



```

gfavoreq[b]:=gfavoreq[b]+golescontraa;
gcontrareq[b]:=gcontrareq[b]+golesfavora;
if (golesfavora>golescontraa) then
begin
ptoseqpo[a]:= ptoseqpo[a]+3;
puntosentreequipos[a,b]:=puntosentreequipos[a,b]+3;
end;
if (golesfavora<golescontraa) then
begin
ptoseqpo[b]:= ptoseqpo[b]+3;
puntosentreequipos[b,a]:=puntosentreequipos[b,a]+3;
end;
if(golesfavora=golescontraa) then
begin
ptoseqpo[a]:= ptoseqpo[a]+1;
ptoseqpo[b]:= ptoseqpo[b]+1;
puntosentreequipos[a,b]:=puntosentreequipos[a,b]+1;
puntosentreequipos[b,a]:=puntosentreequipos[b,a]+1;
end;
golesentreequipos[a,b]:=golesfavora+golesentreequipos[a,b];
golesentreequipos[b,a]:=golescontraa+golesentreequipos[b,a];
end;

{#####PROGRAMA PRINCIPAL}
begin
randomize();
write('Numero de clasificados , 1Clasificado[1] 2Clasificados[2]: ');
read(clasificados);
while (clasificados<>1) and (clasificados<>2) do
begin
write('Introduzca bien el numero: ');
read(clasificados);
end ;
assign(q,'DEFINITIVOLandasA148PoissonFiltrado.txt');
assign(f,'DEFINITIVOLandasA48PoissonFiltrado.txt');
assign(u,'DEFINITIVOLandasA48UNICOPoissonFiltrado.txt');
assign(Resul,'Resultados.txt');
reset(q);
reset(f);
reset(u);
reset(Resul);
rewrite(Resul);
posicion1:=1;
posicion:=1;
{Lectura de los landas, sacamos la matriz de R, los pongo todos en una columna
en un bloc de notas y digo que me lea todo el fichero y me los guarde en un
array Misdatos y posteriormente, ese array lo convertimos en una amtriz}

while not Eof(q) do
begin
readLn(q,a);
misdatos1[posicion1]:=a;
posicion1:=posicion1+1;
end;
while not Eof(f) do
begin
readLn(f,a);
misdatos[posicion]:=a;
posicion:=posicion+1;
end;
for i:=1 to 48 do
begin
for j:=1 to 48 do matriz1[i,j]:=misdatos1[48*(i-1)+j];
end;

for i:=1 to 48 do
begin
for j:=1 to 48 do matriz[i,j]:=misdatos[48*(i-1)+j];
end;

for p:=1 to 48 do vclas[p]:=0;
for s:=1 to 10000 do
begin
for r:=1 to 48 do
begin
puntosequipo[r]:=0;
golesafavorequipo[r]:=0;
golesencontraequipo[r]:=0;
end;
for i:=1 to 48 do
begin
for j:=1 to 48 do golesenfrentamientos[i,j]:=0;
end;

for i:=1 to 48 do
begin
for j:=1 to 48 do puntosenfrentamientos[i,j]:=0;
end;
for r:=1 to 48 do equipos[r]:=r;
for i:=1 to 48 do
for j:=1 to 48 do
begin
if i<j then simularpartido(equipos[i],equipos[j],matriz[equipos[i],equipos[j]]
,matriz[equipos[j],equipos[i]],golesafavorequipo,golesencontraequipo,
puntosequipo,golesenfrentamientos,puntosenfrentamientos);
if i>j then simularpartido(equipos[j],equipos[i],matriz1[equipos[i],equipos[j]],
matriz1[equipos[j],equipos[i]],golesafavorequipo,golesencontraequipo,

```

```

        puntosequipo,golesenfrentamientos,puntosenfrentamientos);
    end;

for p:=1 to 48 do vectPuntos[p]:=puntosequipo[p];
for p := 1 to 48 do begin
    for j := p+1 to 48 do
        begin
            if (puntosequipo[j] > puntosequipo[p]) then
                begin
                    aux := puntosequipo[p];
                    puntosequipo[p] := puntosequipo[j];
                    puntosequipo[j] := aux;
                end;
            end;
        end;
    end;
for p:=1 to 48 do vectPuntos1[p]:=puntosequipo[p];
if clasificados=2 then
    begin
        p:=1;
        while (p<49) do
            begin
                hh:=1;
                x:=0;
                while (hh<33) and (x=0) do
                    begin
                        if vectPuntos[p]=vectPuntos1[hh] then
                            begin
                                x:=1;
                                vclas[equipos[p]]:=vclas[equipos[p]]+1;
                                vectPuntos1[hh]:=0;
                            end
                        else
                            begin
                                x:=0;
                                vclas[equipos[p]]:=vclas[equipos[p]];
                                end;
                            hh:=hh+1;
                            end;
                            p:=p+1;
                        end;
                    end;
                if clasificados=1 then
                    begin
                        p:=1;
                        while (p<49) do
                            begin
                                hh:=1;
                                x:=0;
                                while (hh<17) and (x=0) do
                                    begin
                                        if vectPuntos[p]=vectPuntos1[hh] then
                                            begin
                                                x:=1;
                                                vclas[equipos[p]]:=vclas[equipos[p]]+1;
                                                vectPuntos1[hh]:=0;
                                            end
                                        else
                                            begin
                                                x:=0;
                                                vclas[equipos[p]]:=vclas[equipos[p]];
                                                end;
                                            hh:=hh+1;
                                            end;
                                            p:=p+1;
                                        end;
                                    end;
                                end;
                            end;
                            for p:=1 to 48 do write(Resul,vclas[equipos[p]],',');
                                close(q);
                                close(f);
                                close(u);
                                close(Resul);
                                ReadKey;
                                end.

```

Apéndice B

TABLAS

Porcentaje de veces que se clasifica cada equipo

En las siguientes tablas se puede observar el porcentaje de ocasiones que se clasifica cada equipo, así como las ocasiones que han quedado entre los 16 o 32 mejores (según las simulaciones) y la correlación lineal y de concordancia entre ambos valores (valor en azul y naranja respectivamente)

Grupos de 4 ordenados por el Ranking UEFA

%	14-23	12-24	13-24	32-41	42-31	43-21	14-23 GUYON	12-34 GUYON	13-24 GUYON	32-41 GUYON
BAYERN MUNICH	96,8	97,5	96	96,7	97,1	96,1	96,9	97	97,2	97
REAL MADRID	89,3	91,5	90,8	89,2	88,1	90,3	91	92	91,2	91,2
FC BARCELONA	91,6	93,3	93,6	92,4	90,9	93,6	93,9	93,6	94	92,4
ATLETICO MADRID	83,5	80,6	82,9	84,1	82,3	83,5	86	81,7	83,4	84,6
JUVENTUS	85,6	85,5	85,6	85,1	85,9	85,3	82,8	84,9	83,4	84,5
MAN. CITY	96,2	97,8	97,7	97,5	97,4	98,1	97,6	96,6	97,3	97
PSG	95,5	94,7	95,3	94,1	93,7	95,1	94,4	93,4	93,6	95,5
SEVILLA	60,3	58,8	57,6	59,3	60,6	59,8	55,4	59,1	55,9	54,9
MAN. UNITED	82,6	80,7	80,7	81,7	81,5	83,9	84,6	85,3	82,4	83,4
LIVERPOOL	92,8	95,6	94	93,9	93,3	92,5	94,8	94,8	94,4	93,8
SHAKHTAR DONETSK	48,4	53,3	52,8	51,6	51,4	50,2	54,1	54,3	54,1	53,2
B.DORTMUND	80,3	78,9	79,8	79,2	81,1	79,4	80,1	82,2	82,4	80,3
CHELSEA	81,3	82,1	80,2	83,4	82,1	81,6	79,3	79,7	75,6	81,1
OPORTO	48,8	50,6	52	51,3	49,5	50,5	46,3	49,5	50,5	46,5
AJAX	56,5	61	58,4	58,6	59	57,4	55,4	58,6	58,2	57,4
ZENIT	45,8	45,1	45,3	45,4	44,5	48,4	43,2	42,8	40,9	42,8
D.KIEV	24,9	22,5	23,7	23,2	22,3	23,8	28,1	29,1	31	28,9
SALZBURGO	24,4	26,5	25,7	22,8	25,1	22,8	33,9	30,5	30	29,7
LEIPZIG	52,9	52,8	52,5	54,9	55,7	54,6	60,2	58,5	61	60,5
INTER DE MILAN	58,4	57,6	59,9	56,8	57,5	56,7	61,8	65,4	63,3	65,1
OLYMPIACOS	24,3	21,3	21,8	21,5	22,7	19,6	16,6	19,3	15,7	18,1
LAZIO	36,4	35,7	35,3	37	36,9	35,7	30,9	26,1	31,1	28,6
KRASNODAR	18,9	16	17,2	19,2	18,8	18,1	12,5	13,3	14,2	15,7
ATALANTA	42,7	40,1	40,9	40,4	38,5	39,5	35,7	31,7	36,7	35,1
LOK.MOSCU	10,1	9	7,2	9,7	8,8	9,1	10,8	9,4	10,5	9,5
O.MARSELLA	13,6	13,6	14,6	15,7	16	15,2	14,5	15,1	14,4	16
BRUJAS	16,2	17,6	14,3	14,3	15,6	17	14,5	16,5	15	15,2
B.MONCHENG	20,1	20,2	21,4	19,9	22,4	21,7	20,7	18	18,9	19,2
I.BASAKSEHIR	8,8	8,6	9,2	8,2	7,5	8,9	9,1	7,8	9,4	8,7
MIDTJYLLAND	3,6	3,9	3,8	2,7	4	3,7	3,5	4,3	3,5	4,1
STADE RENNAIS	6,8	5,8	8	7,7	7,9	6,3	8,5	7,4	7,8	8,1
FERENCVAROS	2,6	1,8	1,8	2,5	1,9	1,6	2,9	2,1	3	1,9
Correlación Pearson	95,6654963	95,4807854	95,6465454	95,7615069	95,81025275	95,50620152	96,04980318	95,97315198	96,2762124	96,47093908
Corr. Concordancia	91,4888571	91,7005527	91,7268134	91,8135826	91,65080145	91,64042564	92,29006833	92,42772396	92,4114431	92,71916888

%	42-31 GUYON	43-21 GUYON	13-24	12-34	14-23	13-24 GUYON	12-34 GUYON	14-23 GUYON	16 MEJORES
BAYERN MUNICH	97,1	96,1	92	92,4	92,7	93,8	92,9	94,3	10000
REAL MADRID	90,8	91,1	82,5	82,5	82,6	84,9	85,6	83,6	9998
FC BARCELONA	93,4	93,9	88,2	88,9	88,1	87	88,8	86,9	10000
ATLETICO MADRID	84,9	83,8	75,3	75,1	76,7	76,1	78,9	77,4	9927
JUVENTUS	86	83,8	79,5	78,9	78,4	78,2	77,5	74,2	9979
MAN. CITY	97,2	97,2	92,4	93,4	93,4	91,5	92,8	93,6	10000
PSG	94,6	93,7	88	88,1	89,2	88,1	86,6	90	10000
SEVILLA	56	56,7	55,8	53,2	55,2	55,9	52,7	54,4	4706
MAN. UNITED	82,9	84,5	75,1	75,6	76,3	76	76,3	75,9	9973
LIVERPOOL	95,4	94,5	88,9	86,8	89	87,8	88,5	88,1	10000
SHAKHTAR DONETSK	52,3	56,4	51,9	52,9	52,2	51,3	53,4	53,2	4316
B.DORTMUND	80,1	78,8	75,2	72,5	72,5	74	76,5	76,2	9939
CHELSEA	81	80	76,9	75,9	73,1	75	73,1	75,3	9985
OPORTO	49,4	46,2	51,2	50	51,2	50,5	49	49,3	3860
AJAX	56,8	54,8	57,4	55,7	58,4	56	54,9	56,1	7106
ZENIT	42,3	44,7	47,1	44,1	44,3	45,2	44,9	47,1	1805
D.KIEV	28,7	32,1	29	30,5	27,7	35,1	35,2	33	458
SALZBURGO	28,9	31,2	31,6	33,7	31,5	34,2	35,6	36	558
LEIPZIG	59,2	59,6	52,3	53,3	50,5	61,5	58,1	56,3	8845
INTER DE MILAN	65,4	63,7	58,3	58,5	58,6	62,4	62,6	62,9	9412
OLYMPIACOS	16	14,1	25,4	31,1	29,2	21	25	24,7	232
LAZIO	36,2	32,1	40,3	40,6	40,3	37,2	37,5	36,2	3269
KRASNODAR	12,8	15	24,1	22,9	24,4	21,9	20	20,5	52
ATALANTA	34,2	35,4	44,1	45,2	43	38,2	38,4	37,5	4654
LOK.MOSCU	7,4	9,8	13,7	13	12,5	11,9	14,9	13,2	11
O.MARSELLA	14,5	14,2	20,5	20,6	20,6	20	18,5	17,2	106
BRUJAS	14,6	14,3	19,6	21,9	20	21,9	19,2	20,6	192
B.MONCHENG	19,8	19,5	23,1	24,4	27,6	25,1	22,2	24,5	611
I.BASAKSEHIR	8,7	8,3	13,5	12,2	13,4	13,3	16	13,2	5
MIDTJYLLAND	4,5	4,8	9,3	8,7	8,3	8,4	8,3	8,1	0
STADE RENNAIS	7	7,4	12,8	10,8	13,3	11,3	10,4	14,9	1
FERENCVAROS	1,9	2,3	5	6,6	5,8	5,3	5,7	5,6	0
Corr.Lineal	96,48	95,93	95,07	94,84	94,69	95,56	95,45	95,08	
Corr. Concordancia	92,98	92,14	86,56	85,97	86,02	87,33	87,30	86,91	

Figura B.1: Porcentaje de veces que se clasifica cada equipo en los grupos de 4 ordenados como FIFA según la última jornada y sus correlaciones

Grupos de 4 ordenados por la simulación

%	14-23	12-34	13-24	32-41	42-31	43-21	14-23 GUYON	12-34 GUYON	13-24 GUYON	32-41 GUYON
MAN CITY	96,6	98,1	96,3	95,8	97,4	98,2	97,1	97,8	98,1	96,2
BAYERN MUNICH	89,9	93,2	94,3	88,1	94,6	95	91,5	95,5	94,8	90,5
LIVERPOOL	92,8	98,4	98,4	93,4	98,8	98,4	94,7	98,3	98,9	93
PSG	81,8	96	96,5	83,5	95	94,2	84,9	96,8	96,4	84,7
FC BARCELONA	85,8	96,5	96,8	84,6	97,2	96,7	82,5	96,1	96,2	81,7
REAL MADRID	97,5	89,7	92,4	97	91,4	90,2	97,6	90,9	89,8	97,4
CHELSEA	94,4	86,5	87,7	94	86,3	87,6	94,7	86,8	86,1	94,6
MAN UNITED	59	86,9	89	55,9	87,6	88,9	54,7	89	87,8	56,4
JUVENTUS	80,5	83,7	83,6	80	84,5	81,7	84,8	85,8	84,8	84,7
B.DORTMUND	93	80,2	80,7	94,4	81,4	78,8	94,3	82,2	83,8	95,1
ATLETICO MADRID	50,3	82	80,2	51,2	81,1	81,2	51,8	83,1	82,4	51,7
INTER DE MILAN	79,7	73	74	79,5	72,4	70,2	80,8	75,1	76,8	81,1
LEIPZIG	80,9	67,8	69,2	82,2	68,2	70,3	81,2	66,3	64	80,5
AJAX	49,3	63,1	63,6	51,4	61,9	61,1	46,7	58,4	57,9	47
ATALANTA	60,6	55	56	62,6	55,8	55,7	54,7	53,7	52,5	56,6
SEVILLA	44,8	56	57,8	45,9	56,2	52,7	42,7	51,3	51,5	42,8
SHAKHTAR DONETSK	24,4	36,9	33,2	24,4	34,7	35,2	29,7	42,7	44	27,9
OPORTO	24,9	35,2	33,5	25,3	36,9	34,6	30,9	41,5	42,8	33,1
LAZIO	55,4	32,2	34,3	54,3	34,9	35,5	59	40,7	40	61,9
ZENIT	57,4	29,5	28,1	59,3	28,3	32,6	64,4	34,1	36,4	63,9
B.MONCHENG	21,2	26,8	24	22,2	25	24,5	17	21,1	20,2	17,6
SALZBURGO	38,2	24	21,2	36,3	21,2	24,3	31,1	16,3	19	29
D.KIEV	16,5	21,7	21,2	16,6	21,8	24,3	13,5	17,7	15,9	12,9
OLYMPIACOS	41,1	17,6	18,2	40,1	18	20	38,8	15	14,6	36,7
BRUJAS	8,8	16,2	15,9	9,4	16,6	14,1	7,9	13	13,3	10,4
O.MARSELLA	15,5	13,2	13,6	13,8	13,3	12,4	13,8	13,4	12,4	15,7
KRASNODAR	14,6	10,7	10,5	16,4	11,3	12,2	16,3	7,6	9,2	14
LOK.MOSCU	23,1	8,9	9,3	20,8	7,9	9,3	20,4	6,9	7,7	21
I.BASAKSEHIR	8,6	7,7	7,9	8,2	9	8	8,3	8,7	8,2	8,5
STADE RENNAIS	4,2	7,5	5,9	4	6,5	7,2	3,9	7,7	7,3	4,4
MIDTJYLLAND	6,2	3,5	3,8	7	2,7	2,9	7,2	4,8	4,3	6,2
FERENCVAROS	3	2,3	2,9	2,4	2,1	2	3,1	1,7	2,9	2,8
Correlación Pearson	90,2463579	98,0575049	98,1253556	90,2304411	98,10654972	97,85058921	90,41090486	98,55866822	98,4749943	90,60357411
Corr. Concordancia	94,7250767	95,3495004	95,2831263	95,4637589	95,07056765	95,40759721	95,63943829	96,13865648	96,03964266	95,74134872

%	42-31 GUYON	43-21 GUYON	13-24	12-34	14-23	12-24 GUYON	12-34 GUYON	14-23 GUYON	16 MEJORES
MAN CITY	97,5	97,5	93,1	94,3	92,5	94,2	93	93,8	10000
BAYERN MUNICH	94,5	95,8	87,7	90,3	88,4	91,1	90,2	90,4	10000
LIVERPOOL	98,5	97,9	92,9	94,1	95,2	96,3	95,2	95,2	10000
PSG	96,6	95,8	88,6	88,4	90,1	91,7	90,9	90,5	10000
FC BARCELONA	95,5	96,6	91,4	90,2	92,6	90,3	92,4	91,8	10000
REAL MADRID	89,7	91,7	86,7	82,7	86	83,2	84,1	83	9998
CHELSEA	87,2	86,3	80,9	79,6	79,2	78,9	78,9	79	9985
MAN UNITED	88,1	87,7	79,1	79,8	79,3	77,9	80,2	81	9973
JUVENTUS	86,6	85,8	75,6	76,9	77,4	77,3	77,1	80	9979
B.DORTMUND	82,9	83,4	71,5	75,6	73,4	75,5	74,6	74,1	9939
ATLETICO MADRID	83,2	82,7	72,2	74,3	74	74,2	77,2	80,1	9927
INTER DE MILAN	75,8	74	67,8	65,7	66,9	72,4	68,4	70	9412
LEIPZIG	66,6	67,6	62,2	65	61,9	62,6	62,2	62	8845
AJAX	58,1	55,9	57,7	58,9	59,8	56,2	54,6	53,3	7106
ATALANTA	52,5	51,8	51	51,7	53,4	50,1	46,7	49,3	4654
SEVILLA	52,8	49,4	52,9	51,7	52,1	50,4	51,9	51,2	4706
SHAKHTAR DONETSK	42,4	42,3	39,7	37,1	42,6	43,8	47	45,8	4316
OPORTO	41,3	42,4	40	42,3	39	44,2	43,2	44,6	3860
LAZIO	40,5	42,9	40,5	38,6	41,3	42,2	44,7	45,1	3269
ZENIT	36,3	39,1	35,7	33,1	32,4	37,4	40,8	40,5	1805
B.MONCHENG	20,2	17,8	31,5	31,1	29,7	29	25,6	24,9	611
SALZBURGO	17,1	19	31,7	29,2	30,2	26,2	26	26,1	558
D.KIEV	16,8	16,6	30,2	31,5	29	22	24,5	23,8	458
OLYMPIACOS	17,2	15	25,2	27,1	26,2	24,9	19,9	20,6	232
BRUJAS	10,7	14,5	21,5	19,7	21	21,2	20,9	19,1	192
O.MARSELLA	11,5	13	19,5	20,9	18,8	19,7	19,3	18,2	106
KRASNODAR	11,1	8,9	16,8	14,8	16,6	14	14,9	14,7	52
LOK.MOSCU	6,4	7	15,3	12,4	14,3	13,3	16	13,6	11
I.BASAKSEHIR	8	9,2	15,7	16,5	15,2	15,6	14	14,3	5
STADE RENNAIS	7,5	6,1	11,1	11,8	9,6	10,8	11,8	10,8	1
MIDTJYLLAND	4,3	4,2	8,3	9,4	6,5	7,9	8,3	6,5	0
FERENCVAROS	2,6	2,1	6	5,3	5,4	5,5	5,5	6,7	0
Corr.Lineal	98,58	98,39	96,71	96,95	96,87	97,32	97,28	97,43	
Corr. Concordancia	95,84	95,90	89,83	89,65	88,96	89,64	89,64	89,55	

Figura B.2: Porcentaje de veces que se clasifica cada equipo en los grupos de 4 ordenados por simulación según la última jornada y sus correlaciones

```
> t.test(Diagrama_cajas_grupos_45`INTRASCENDENTE 13-24 GU`,Diagrama_cajas_grupos_45`INTRASCENDENTE 13-24`)

Welch Two Sample t-test

data: Diagrama_cajas_grupos_45`INTRASCENDENTE 13-24 GU` and Diagrama_cajas_grupos_45`INTRASCENDENTE 13-24`
t = 0.53139, df = 1997.7, p-value = 0.5952
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.1614376  0.2814376
sample estimates:
mean of x mean of y
  7.476    7.416

> t.test(Diagrama_cajas_grupos_45`COLISION 13-24 GU`,Diagrama_cajas_grupos_45`COLISION 13-24`)

Welch Two Sample t-test

data: Diagrama_cajas_grupos_45`COLISION 13-24 GU` and Diagrama_cajas_grupos_45`COLISION 13-24`
t = 0.9427, df = 1992.9, p-value = 0.3459
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.04645567  0.13245567
sample estimates:
mean of x mean of y
  1.276    1.233
```

Figura B.3: Test de comparación de medias entre para el enfrentamiento 13–24.

```
> t.test(Diagrama_cajas_grupos_45`INTRASCENDENTE 32--41`,Diagrama_cajas_grupos_45`INTRASCENDENTE 32--41 GU`)

Welch Two Sample t-test

data: Diagrama_cajas_grupos_45`INTRASCENDENTE 32--41` and Diagrama_cajas_grupos_45`INTRASCENDENTE 32--41 GU`
t = 0.55956, df = 1996.5, p-value = 0.5758
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.1502885  0.2702885
sample estimates:
mean of x mean of y
  8.036    7.976

> t.test(Diagrama_cajas_grupos_45`COLISION 32--41 GU`,Diagrama_cajas_grupos_45`COLISION 32--41`)

Welch Two Sample t-test

data: Diagrama_cajas_grupos_45`COLISION 32--41 GU` and Diagrama_cajas_grupos_45`COLISION 32--41`
t = -1.423, df = 1990.8, p-value = 0.1549
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.13317599  0.02117599
sample estimates:
mean of x mean of y
  0.839    0.895
```

Figura B.4: Test de comparación de medias entre para el enfrentamiento 14–23.

```
> t.test(Diagrama_cajas_grupos_45`INTRASCENDENTE 12--34 GU`,Diagrama_cajas_grupos_45`INTRASCENDENTE 12--34`)

Welch Two Sample t-test

data: Diagrama_cajas_grupos_45`INTRASCENDENTE 12--34 GU` and Diagrama_cajas_grupos_45`INTRASCENDENTE 12--34`
t = 0.017238, df = 1997.6, p-value = 0.9862
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.225533  0.229533
sample estimates:
mean of x mean of y
  8.452    8.450

> t.test(Diagrama_cajas_grupos_45`COLISION 12--34 GU`,Diagrama_cajas_grupos_45`COLISION 12--34`)

Welch Two Sample t-test

data: Diagrama_cajas_grupos_45`COLISION 12--34 GU` and Diagrama_cajas_grupos_45`COLISION 12--34`
t = 0.97955, df = 1993.1, p-value = 0.3274
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
-0.04910287  0.14710287
sample estimates:
mean of x mean of y
  1.649    1.600
```

Figura B.5: Test de comparación de medias entre para el enfrentamiento 12–34.

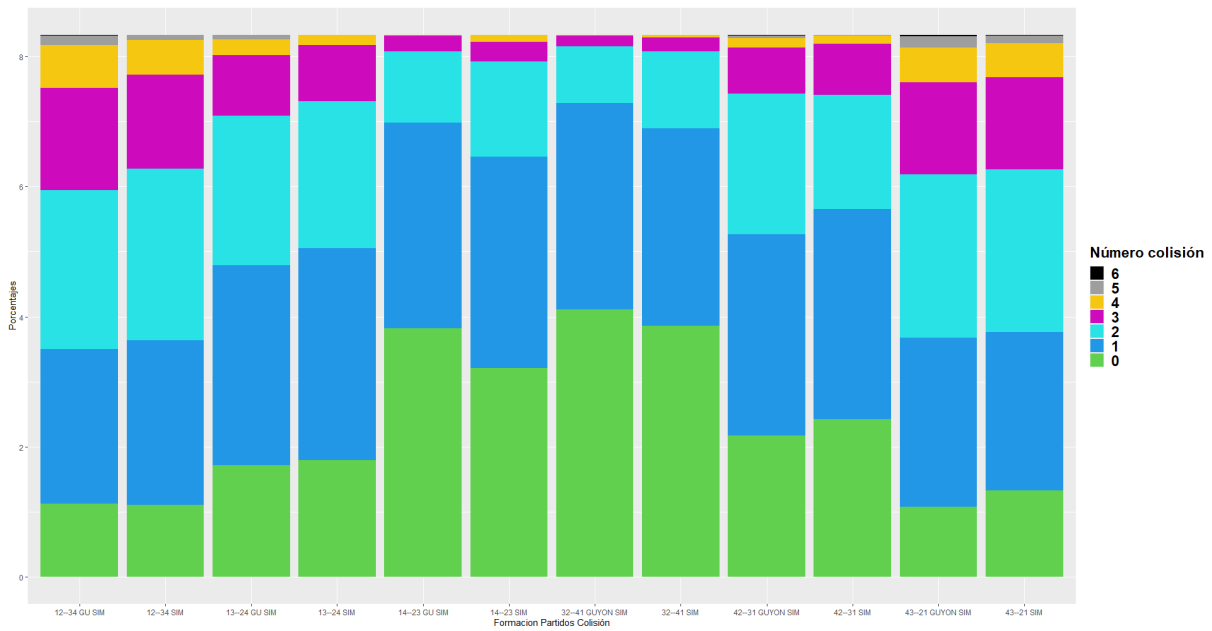


Figura B.6: Diagrama de barras apiladas de los partidos de colisión de los distintos órdenes para grupos de 4.

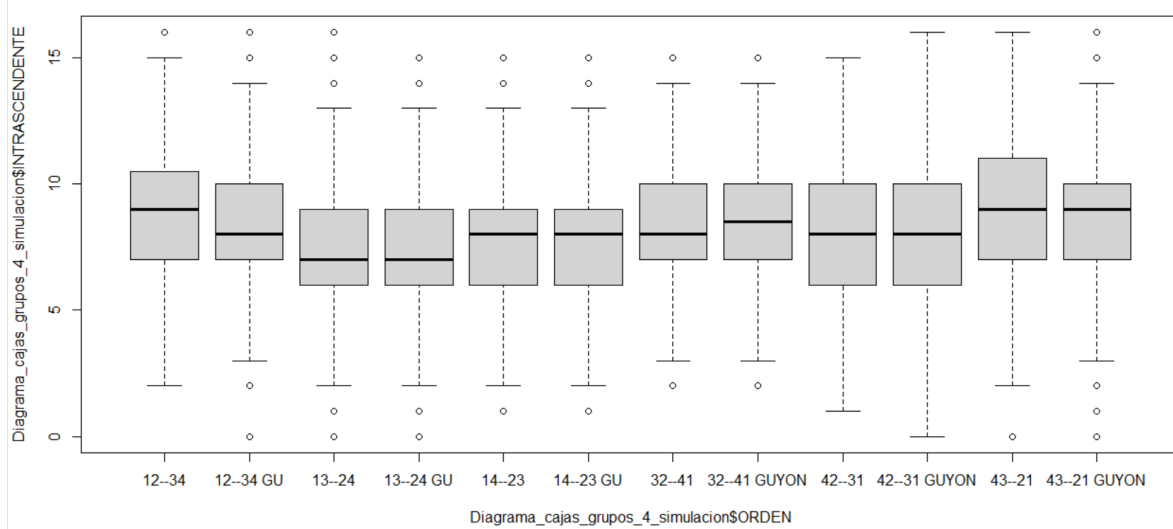


Figura B.7: Diagrama de cajas de los partidos intrascendentes de los distintos órdenes para grupos de 4.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
COLISION 14-23	0	0	1	1	4	0.90	0.87
COLISION 14-23 GU	0	0	1	1	4	0.80	0.79
COLISION 12-34	0	1	2	2	5	1.13	1.81
COLISION 12-34 GU	0	1	2	3	6	1.22	1.80
COLISION 13-24	0	1	1	2	4	0.98	1.30
COLISION 13-24 GU	0	1	1	2	5	1.07	1.34

Tabla B.1: Estadísticos de los partidos de colisión para grupos de 4 ordenados por simulación

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
INTRASCENDENTE 14-23	1	6	8	9	15	2.35	7.63
INTRASCENDENTE 14-23 GU	1	6	8	9	15	2.41	7.67
INTRASCENDENTE 12-34	2	7	9	10	16	2.67	8.71
INTRASCENDENTE 12-34 GU	0	7	8	10	16	2.74	8.76
INTRASCENDENTE 13-24	0	6	7	9	16	2.47	7.39
INTRASCENDENTE 13-24 GU	0	6	7	9	15	2.59	7.42

Tabla B.2: Estadísticos de los partidos intrascendentes para grupos de 4 ordenados por simulación

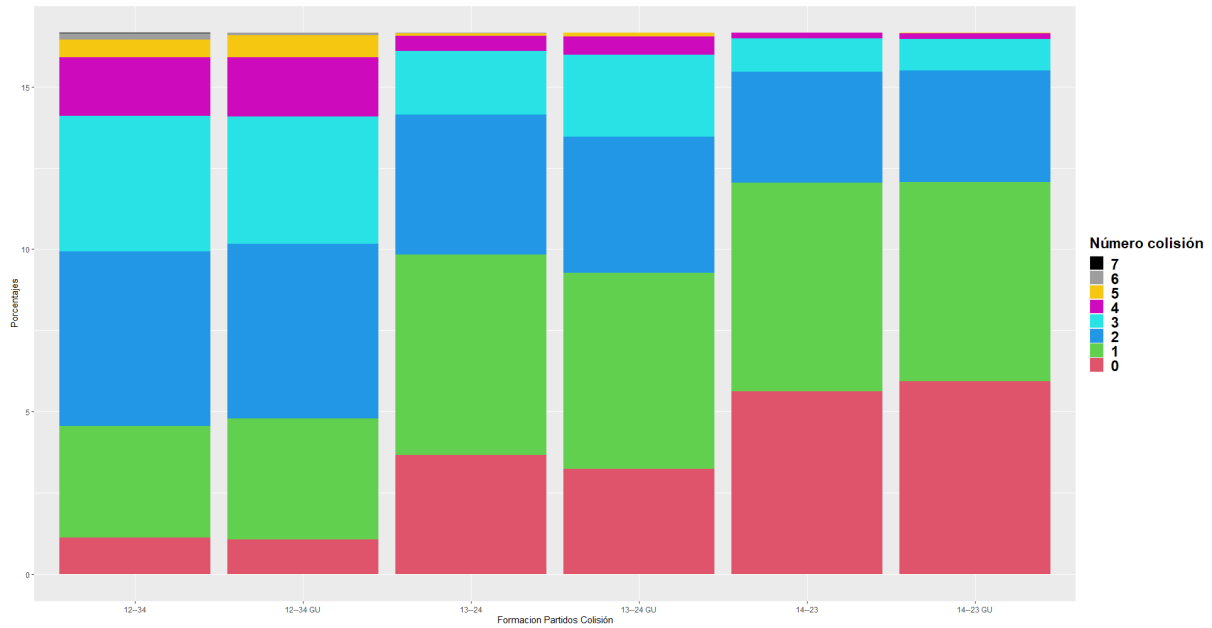


Figura B.8: Diagrama de cajas de los partidos de colisión de los distintos órdenes con partido único en grupos de 4.

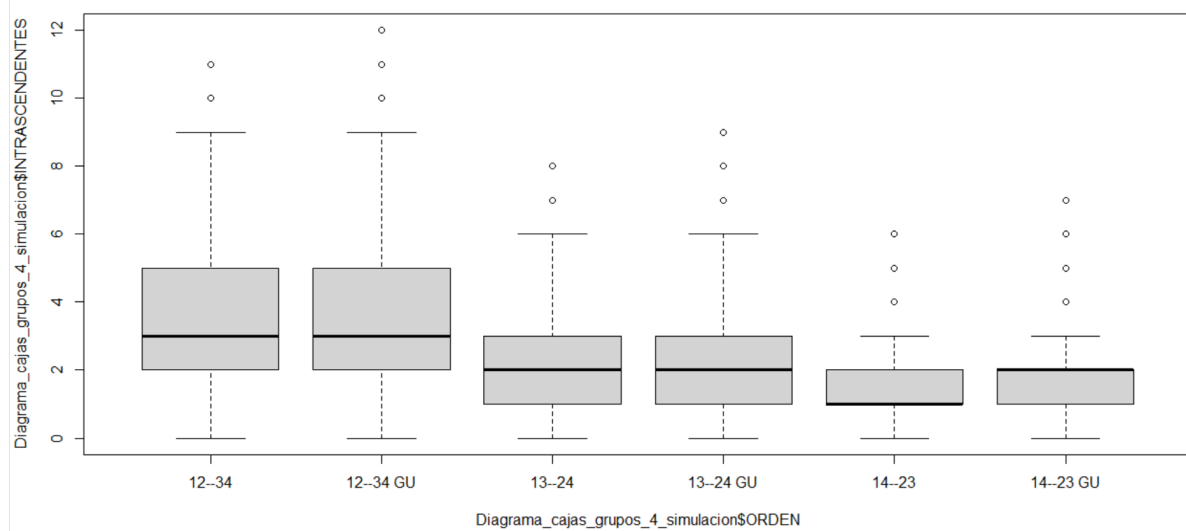


Figura B.9: Diagrama de cajas de los partidos intrascendentes de los distintos órdenes con partido único en grupos de 4.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
COLISION 13-24	0	1	1	2	6	1.07	1.37
COLISION 13-24 GU	0	1	1	2	5	1.10	1.49
COLISION 12-34	0	1	2	3	7	1.26	2.23
COLISION 12-34 GU	0	1	2	3	6	1.24	2.23
COLISION 14-23	0	0	1	2	4	0.94	0.97
COLISION 14-23 GU	0	0	1	2	5	0.95	0.93

Tabla B.3: Estadísticos de los partidos de colisión para grupos de 4 ordenados por simulación con partido único

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
INTRASCENDENTE 13-24	0	1	2	3	8	1.60	1.86
INTRASCENDENTE GU 13-242	0	1	2	3	9	1.67	1.96
INTRASCENDENTE 12-34	0	2	3	5	11	2.21	3.37
INTRASCENDENTE GU 12-34	0	2	3	5	12	2.22	3.35
INTRASCENDENTE 14-23	0	1	1	2	6	1.31	1.73
INTRASCENDENTE GU 14-23	0	1	2	2	7	1.28	1.61

Tabla B.4: Estadísticos de los partidos intrascendentes para grupos de 4 ordenados por simulación con partido único

Grupos de 3 con 1 clasificado ordenados por Ranking UEFA

%	32	31	21	23	13	12	31GUYON	31GUYON	21GUYON	23GUYON
ARSENAL	79,5	81,2	79,7	81,7	81,9	80,4	77,9	78	79,4	78,1
TOTTENHAM	85,6	85,8	85,6	85,7	86,7	87,6	86,1	84,9	83,9	82,5
ROMA	75,8	72,9	73,1	76,1	75,4	75,9	70,6	73,2	71,7	70,8
NAPOLIS	82,7	82,1	79,3	81,8	80,3	79,5	79,2	76,7	77,6	79,6
BENFICA	69,5	70,7	71,3	69,4	69,3	69,5	67,8	63,9	66,6	66,5
B.LEVERKUSEN	73,1	75,4	73,8	72,8	73,3	72,7	68,3	72,4	70,1	69,8
VILLARREAL	64,1	65,5	65,5	62,7	62,6	65,1	59,4	59,3	62,4	60,6
CSKA MOSCU	54,2	53	56,7	54,5	57,2	53,3	46,6	49,5	48,4	51,6
SC BRAGA	52,3	48,8	51,5	51,6	50,4	49,6	55,6	55,2	56,2	56,2
GANTE	51,4	49,2	47,5	47,2	51,6	49	52,7	56,4	55,4	54,3
PSV	58,6	56,8	60,2	56,7	59,2	58,4	63,1	61,1	64,3	64,6
CELTIC	64,8	63,2	65,5	62,4	63,3	62,8	69,6	69,8	67,3	72,1
DINAMO ZAGREB	49,7	49,3	49,3	45,1	47,9	52,4	53,4	54,1	52,3	53,5
S.PRAGA	44,9	42,5	43,7	44,2	43,5	46,3	49,9	48,5	48,4	46,9
SLAVIA PRAGA	46,8	42,4	46	45,8	44,8	45,8	51,5	51,7	52,8	49,4
LUDOGORETS	46,1	48,7	51,9	49,7	51,7	49,1	55,7	58,1	52,9	54,7
YOUNG BOYS	30,5	31,3	33,5	31,3	31,2	32	24	22,3	20,2	21,1
ESTRELLA ROJA	31,5	31,4	30,4	29,9	31,3	30,2	20,1	19,7	22,3	20,2
RAPID VIENA	22	24,2	23,8	25,1	22,9	22,7	16,1	15,4	15,3	12,6
LEICESTER	50,7	48,9	47,9	55,1	50,5	53,2	39,8	39,6	38,7	40,5
QARABAG	20,7	22,4	22,4	21,1	22,6	22,1	14,9	15,3	14,5	16,1
PAOK	26,5	29	27,5	28,4	29,7	26,2	17,2	19,9	18,2	18
ST.LIEJA	20,3	23,3	23,2	21,5	22,7	21,3	15,6	15,1	15,9	13,4
REAL SOCIEDAD	32,5	31,9	31,8	36	32,7	31,7	22,5	21,6	24	23,7
MILAN	40,7	41	40,6	40,8	39,6	40,6	52,8	50,7	52,2	50,7
AZ ALKMAR	24,4	26	24,3	25,5	24	24,4	36,6	34,5	35,3	35,1
FEYERNOORD	23,2	24	23,8	23,9	24,5	22,8	32,1	31,3	31,2	33,2
GRANADA	20,4	18,3	20,1	19,7	18,1	20,3	27,1	28,6	30,1	28,6
AEK	22,3	20,5	18,3	19,7	19,3	21,4	29,8	25,5	30,1	30,5
MACCABI	20,9	21,6	17,6	23,1	19,7	21,7	28,7	29,9	29,7	28,3
G.RANGERS	23,6	24,7	23,1	21	25,9	22,3	32,4	32,3	29,8	30,5
MOLDE	25,1	26,1	27,4	27,3	24,7	26	35,4	37,5	38,2	36
HOFFENHEIM	23,3	26,2	22,2	21,3	23,1	21,9	15,9	13,7	15,3	15
LASK	12,9	11,1	11,3	12,3	12,9	13,3	8,9	8,1	8,2	9
APOEL.B.SHEVA	10	10,7	11,1	10,2	9,8	10,8	7,2	6,7	6,8	6,8
SIVASSPOR	13,7	13,7	12,2	14	12	15,4	10,3	9,1	8	9,1
CLUJ	13,7	12,5	13,7	14,2	15	12,9	8,2	10,2	9,4	9,7
ZORYA	9,2	11,2	10,9	10,2	10,9	11,8	7,7	6,6	5,9	6
LILLE	14,2	17,5	16,9	15,8	11,8	14	8,9	8,7	10,1	9,9
NIZA	13	13,4	12,3	13,1	15,1	11,3	6,8	10,1	7,1	9,4
RIJKA	12,9	11,2	12,1	12,4	11,5	11	15,5	16,3	15,4	16,6
DUNDALK	6,4	5,7	7,3	6,6	6,7	6,4	7,3	8,6	8	10,4
S.LIBEREC	8,7	6,8	8,3	7,4	8,5	7,6	11,1	11,4	11,5	11,1
AMBERES	8,4	8,8	7,5	6,7	6	6,4	11	9,8	11,2	10,1
LECH POZNAN	8,8	8,7	7,6	8,4	7,7	8,5	11,9	13	11,5	11,8
WOLFSBERGER	4,3	5,2	4,8	5,6	4,6	6,6	7,4	8	7,8	8,2
OMONIA	3,5	3,7	3,1	2,8	3,5	3,1	5,5	4,6	5	3,8
CSKA SOFIA	2,6	1,5	2,4	2,2	2,4	2,7	3,9	3,1	3,4	3,4
Corr. Lineal	94,30	94,16	94,40	94,53	94,16	94,30	89,25	89,08	89,21	89,87
Corr. Concordancia	83,71	83,23	83,76	83,74	83,71	83,69	79,71	79,60	79,74	80,33

Figura B.10: Porcentaje de veces que se clasifica cada equipo en los grupos de 3 con 1 clasificado ordenados como FIFA según la última jornada y sus correlaciones

%	13GUYON	12GUYON	31	21	23	31GUYON	21GUYON
ARSENAL	76,4	79	71,8	72,2	71,1	70,6	67,3
TOTTENHAM	84,1	84,2	78,2	77,1	76,1	72,3	73,6
ROMA	70,5	71,1	66,1	64,6	66,1	63,6	62,8
NAPOLES	76	79,4	73,4	72,6	70,6	70	68,9
BENFICA	64,5	68,3	62,6	60,9	61,5	57,7	55,9
B. LEVERKUSEN	71	69,1	65,5	65	65	63,9	63,5
VILLARREAL	61	59,9	57,8	57,9	59	57,2	55,2
CSKA MOSCU	51,6	50,4	51,7	50,4	50	44,3	45,8
SC BRAGA	59,4	56,1	46,6	46,6	45,1	48,6	47,9
GANTE	55,5	55,3	47,4	45,7	44	50,9	50,1
PSV	62	63,3	52,1	55	54,1	58,3	55,5
CELTIC	71,2	68,5	61,3	58,9	58,2	61,8	60,5
DINAMO ZAGREB	53,7	55,4	43,5	46,3	44,4	46,8	47,8
S.PRAGA	47,8	48,4	41,4	43,6	38,7	46,9	48,9
SLAVIA PRAGA	52	48,4	37,5	43	41,1	48,1	44,1
LUDOGORETS	56,4	53,4	45,4	45,1	46,2	54,2	50,4
YOUNG BOYS	20,1	21,6	35,4	34,1	33,8	26,2	29,3
ESTRELLA ROJA	21	19,9	29,4	30,9	32,8	22,3	22,6
RAPID VIENA	15,7	16,8	24	25,5	26	20	20,5
LEICESTER	40,1	36,7	47,3	47,2	48,5	41,4	40,3
QARABAG	14,7	15,6	25	24,5	27,8	20,2	17,8
PAOK	19,7	19,6	30,2	31,6	30,6	20,8	23,4
ST.LIEJA	13,3	14,2	25,4	23,4	25,8	19,9	17,9
REAL SOCIEDAD	22,7	22,3	33,4	32,3	31,8	24,3	28,8
MILAN	51,6	53,5	40,3	39,6	37,8	46	47
AZ ALKMAR	34,1	35,7	28,8	28,1	30,3	39,1	38
FEYERNOORD	32,6	32,6	25,6	28,4	27,1	32,4	34,7
GRANADA	28,1	30,9	25,8	20,6	25	29,6	30
AEK	25,6	31,2	23,6	23,4	28,1	31,1	30,3
MACCABI	26,2	27,5	23,7	24,5	21,4	30,3	30,6
G.RANGERS	33,5	33,3	28,3	26,7	27,4	33,9	34
MOLDE	38,4	39,3	31,6	30,1	31,2	36,6	36,6
HOFFENHEIM	16,1	16,1	24,5	25,5	25,7	18,7	19,6
LASK	8,2	7,6	17,9	16,8	17,1	12,5	14,3
APOEL.B.SHEVA	7,2	6,2	14,1	13,6	16,3	11,1	10,8
SIVASSPOR	10,5	9,2	17,6	18,8	18,1	13,3	12,6
CLUJ	11	9,6	18,4	19,2	16,3	12,4	12,9
ZORYA	6,2	7,1	15,9	15,5	15,3	13,2	10,7
LILLE	8,9	7,9	16,4	16,6	15,8	12,3	12,1
NIZA	9,5	8,2	14,4	16,2	17,4	11,8	13,4
RIJEKA	15,9	13,3	13,6	16,5	16,3	20,3	20,7
DUNDALK	8,9	7,6	8,3	9	9,4	12,9	13,7
S.LIBEREC	9,9	12,2	12,4	10,5	12,2	15,1	14,6
AMBERES	11,1	9,6	12,1	11,8	11,7	14,1	18,2
LECH POZNAN	12,4	10,7	13,3	12,4	11	16,5	15
WOLFSBERGER	7,5	6,1	8,6	9,5	9,9	12,2	14
OMONIA	4,1	4,5	7	7,5	5,3	6,9	10
CSKA SOFIA	2,1	3,2	5,4	4,8	5,6	7,4	7,4
Corr.Lineal	89,43	88,78	93,93	93,46	93,40	88,66	89,00
Corr. Concordancia	79,88	79,59	75,07	74,30	73,48	71,01	69,76

Figura B.11: Porcentaje de veces que se clasifica cada equipo en los grupos de 3 con 1 clasificado ordenados como FIFA según la última jornada y sus correlaciones

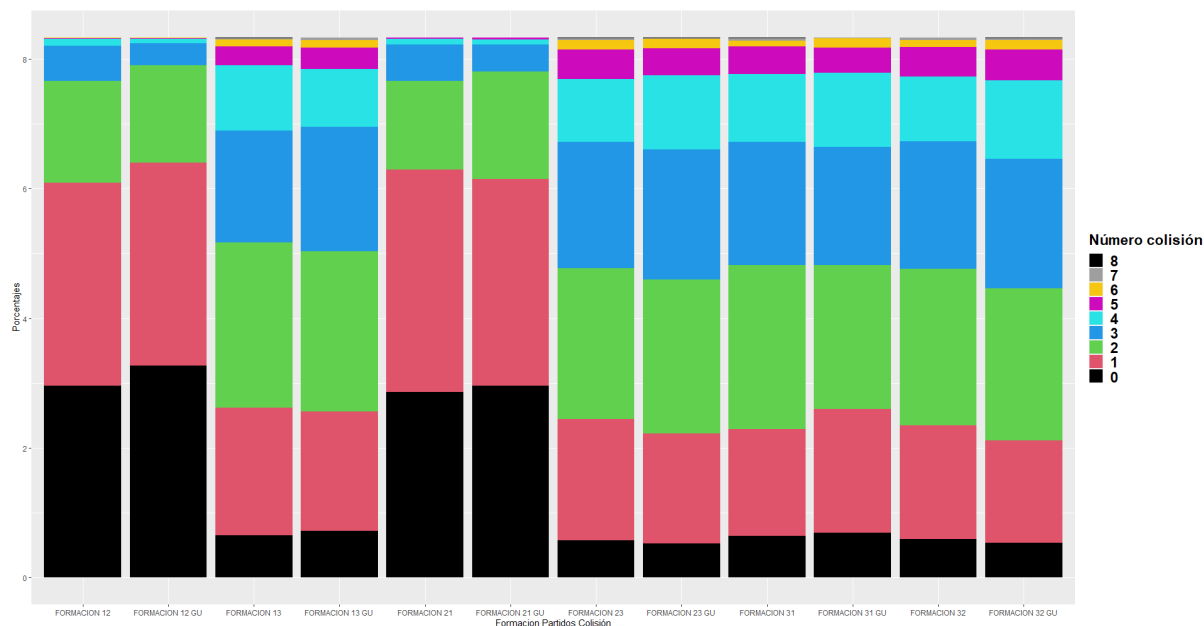


Figura B.12: Diagrama de barras apiladas de los partidos de colisión de los distintos órdenes para grupos de 3 con 1 clasificado.

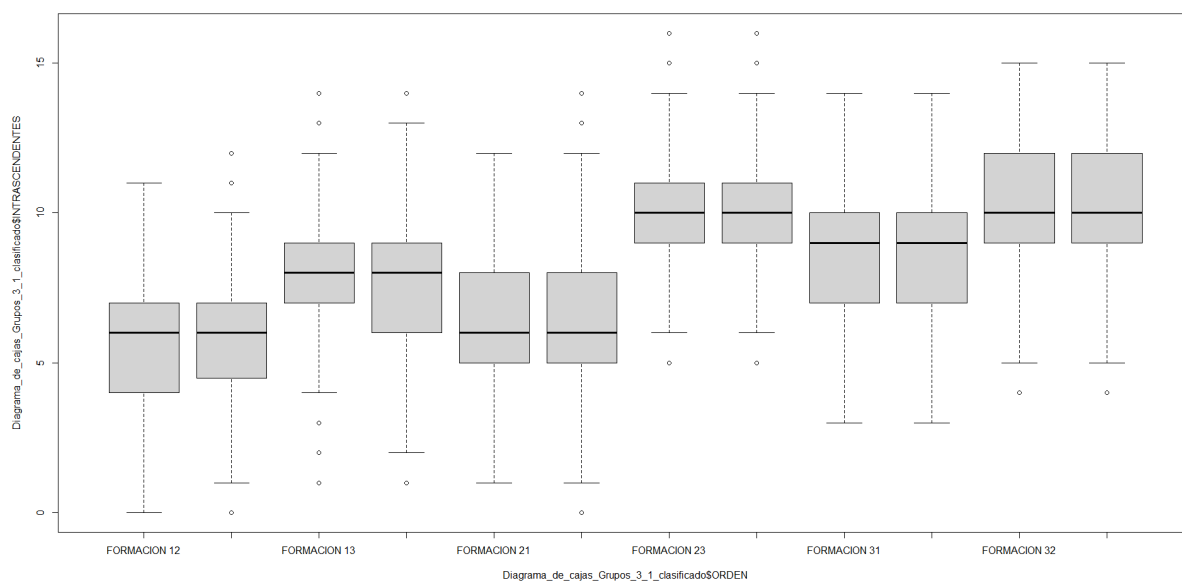


Figura B.13: Diagrama de cajas de los partidos intrascendentes de los distintos órdenes para grupos de 3 con 1 clasificado.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
COLISION 32	0	1	2	3	7	1.38	2.50
COLISION 32 GU	0	1	2	3	8	1.40	2.50
COLISION 31	0	1	2	3	8	1.38	2.35
COLISION 31 GU	0	1	2	3	7	1.40	2.33
COLISION 21	0	0	1	1	5	0.95	1.03
COLISION 21 GU	0	0	1	2	5	0.94	0.95

Tabla B.5: Estadísticos de los partidos de colisión para grupos de 3 con 1 clasificado

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
INTRASCENDENTES 32	4	9	10	12	15	1.85	10.09
INTRASCENDENTES 32 GU	4	9	10	12	15	1.89	10.16
INTRASCENDENTES 31	3	7	9	10	14	1.95	8.69
INTRASCENDENTES 31 GU	3	7	9	10	14	1.94	8.57
INTRASCENDENTES 21	1	5	6	8	12	1.94	6.54
INTRASCENDENTES 21 GU	0	5	6	8	14	1.99	6.38

Tabla B.6: Estadísticos de los partidos intrascendentes para grupos de 3 con 1 clasificado

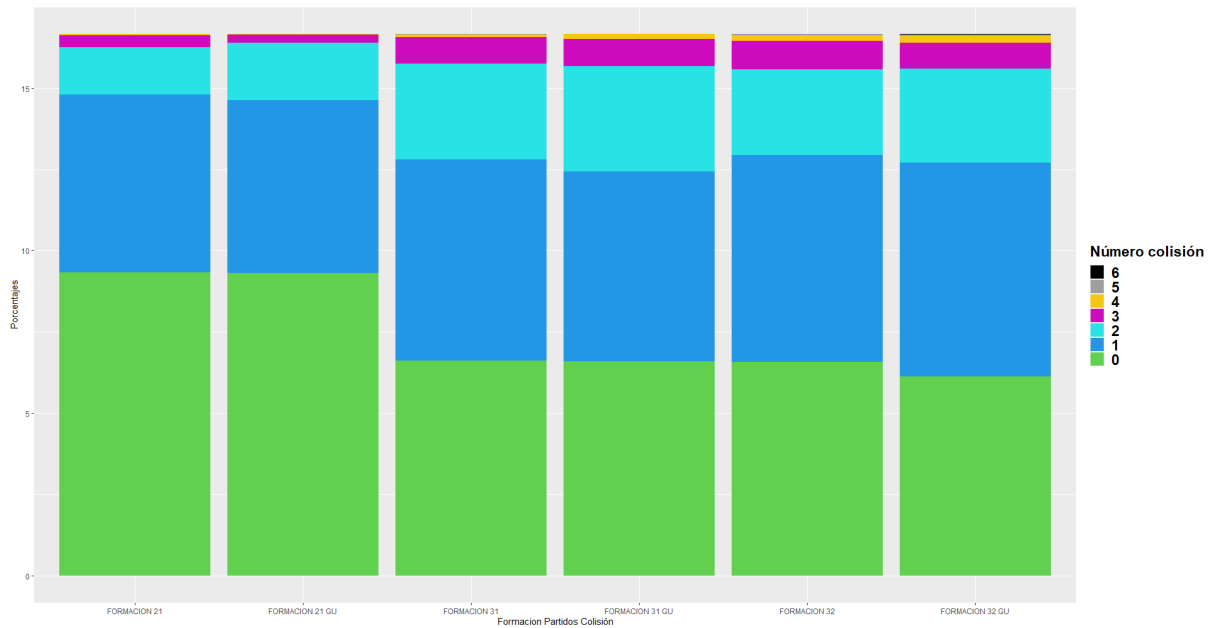


Figura B.14: Diagrama de barras apiladas de los partidos de colisión de los distintos órdenes con partido único para grupos de 3 con 1 clasificado.

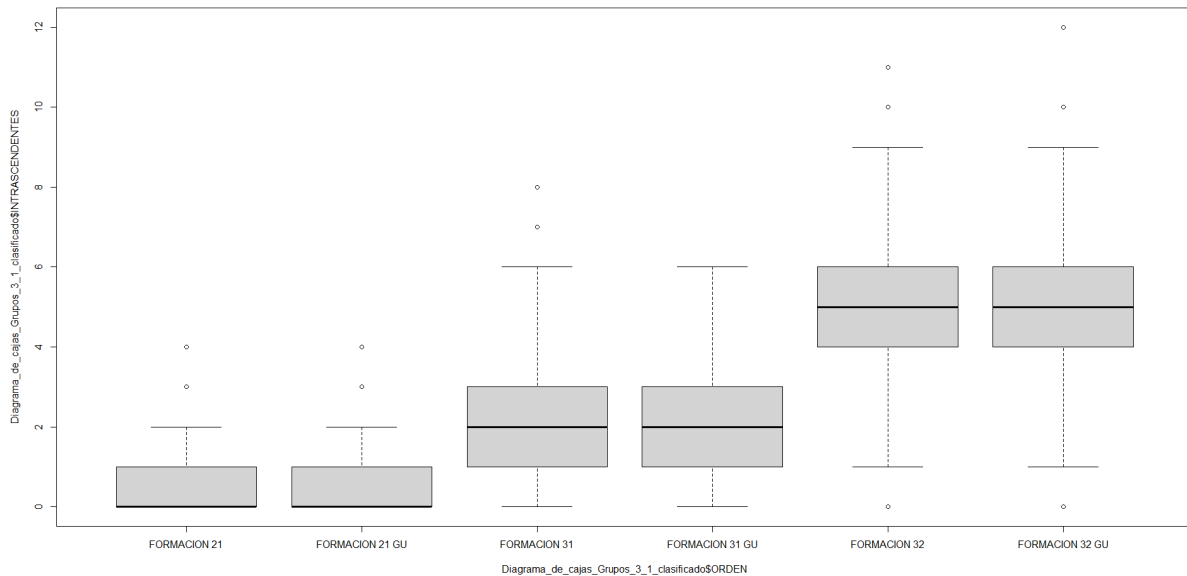


Figura B.15: Diagrama de cajas de los partidos intrascendentes de los distintos órdenes con partido único para grupos de 3 con 1 clasificado.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
COLISION 32	0	0	1	1	5	0.94	0.95
COLISION 32 GU	0	0	1	1	6	0.95	0.95
COLISION 31	0	0	1	1	5	0.91	0.89
COLISION 31 GU	0	0	1	2	4	0.93	0.88
COLISION 21	0	0	0	1	4	0.76	0.62
COLISION 21 GU	0	0	0	1	4	0.75	0.58

Tabla B.7: Estadísticos de los partidos de colisión para grupos de 3 con 1 clasificado con partido único

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
INTRASCENDENTES 32	0	4	5	6	11	1.77	4.75
INTRASCENDENTES 32 GU	0	4	5	6	12	1.73	4.85
INTRASCENDENTES 31	0	1	2	3	8	1.31	1.95
INTRASCENDENTES 31 GU	0	1	2	3	6	1.25	1.89
INTRASCENDENTES 21	0	0	0	1	4	0.75	0.66
INTRASCENDENTES 21 GU	0	0	0	1	4	0.73	0.60

Tabla B.8: Estadísticos de los partidos intrascendentes para grupos de 3 con 1 clasificado con partido único

Grupos de 3 con 1 clasificado ordenados por la simulación

%	32	31	21	23	13	12	32GUYON	31GUYON	21GUYON	23GUYON
TOTTENHAM	87,7	89,5	87,6	89,6	89,7	88,1	86,6	85,6	84,7	86,9
NAPOLES	83,5	85,8	81,6	84,1	81,9	83,8	79,1	81,3	76,8	78,2
ARSENAL	83,2	81,7	83,6	83,6	82,3	83,3	80,4	80,7	79,8	79,5
ROMA	75,7	75,8	76,7	75,8	77,9	74,9	74,6	72,2	72,5	70,2
B.LEVERKUSEN	76,9	72,7	75,5	77,4	78,8	75,9	75,1	71,9	73,6	69,6
BENFICA	72,7	69,9	72,2	69,1	71,1	72,8	67,4	68	67,2	69,2
LEICESTER	68,8	65,7	68,3	67,6	68,1	69,2	63,9	62,5	64,4	67,4
VILLARREAL	68,2	70,8	69,2	67,9	63,8	65,9	63,3	63	64,2	63,6
CELTIC	66,3	63,4	66,1	66,6	67,9	67	71,1	72,2	70,1	70,6
PSV	58,2	61,9	59,5	56	60,5	57	65,7	64,4	65,3	65,2
MILAN	57,1	58,7	60,7	54,6	60,8	59,2	62,2	63,3	63,5	62,5
CSKA MOSCU	56,7	57,9	57,8	54,3	60,1	57,1	61,3	63,3	60,2	60,5
SC BRAGA	52,4	55,1	52,8	54,9	49,7	52,2	58,3	58,9	59,1	59,8
GANTE	53	50,1	50,1	50,7	49,6	50,8	55,3	59,4	58,6	59,4
LUDOGORETS	49,5	50,7	50,1	52,5	46,9	51,9	57,9	57,8	54,3	58
DINAMO ZAGREB	50,1	52,8	49,8	52,5	52,7	47	57,6	56,4	53,8	55,5
YOUNG BOYS	30,7	30,3	27,4	31,2	29,4	31,4	21,5	22,3	21,2	21,8
REAL SOCIEDAD	32,8	30,8	30,9	30,6	31	29,8	21,8	22,4	22,3	23,4
ESTRELLA ROJA	25,4	28,5	29,3	26,4	29,9	29,6	18,6	17,7	18,4	18,6
SLAVIA PRAGA	27	27,1	27,9	28	27,7	28,8	18,9	16,6	19,4	18,2
S.PRAGA	24	30,4	26,9	24,6	28,3	29,2	18,6	19,4	20,4	17,6
PAOK	27,6	26,3	28,6	28	27	28,5	17,9	17,9	16,5	17,9
HOFFENHEIM	27,6	27,4	24,6	29,2	23,7	26,6	16,1	19	18,2	18,7
MOLDE	25,1	24,4	25,9	24,8	24,3	24,1	15,1	17,4	17	14
AZ ALKMAR	23,4	24,2	25,3	25,6	24,4	27,3	34,8	35,8	34,6	35,3
FEYERNOORD	19,8	22,7	23,1	23	19,4	22,3	28	30,9	31,5	29,1
RAPID VIENA	22,8	21,1	22,2	22,4	20,3	21,3	33,5	27,4	31,7	29,3
QARABAG	21,3	20,1	20,1	21,4	20,8	20,9	30,4	30,1	29,8	31,9
G.RANGERS	22,8	20	21,3	21,9	23,1	21	30,5	29,4	32,3	27,3
ST.LIEJA	20,9	19,5	19,2	19,4	22,6	21,4	29,6	28,6	29,2	28,4
AEK	16,4	19,5	19,9	19,2	21,2	17,5	27,5	27,1	28	28,3
GRANADA	20	19,8	17,2	18,3	20,9	21,2	28,4	25,6	27,7	26
MACCABI	15,8	13,6	16,8	15,4	13,8	12,3	8,3	7,1	11,4	9
CLUJ	14,7	13,3	13,4	12,7	14,8	12,3	8,2	9	8,2	6,7
LILLE	14,1	14,5	15,4	14,8	14,8	14,1	8,8	9,4	9,2	10,4
SIVASSPOR	13,4	12,6	12	12,9	12,7	12,2	7,8	8,1	8,5	8,7
LASK	11,8	12,3	12,2	12,2	11,7	13,5	6,5	7,9	6,7	7,6
NIZA	12,9	11,1	12,4	12,1	12,8	11,4	7,7	7,7	7,4	8,3
RIJEKA	11,4	11,3	9,8	10,7	10,4	10,4	6,7	6,6	6,9	7,7
APOEL B.SHEVA	11,1	12,1	10,3	8,8	8,7	7,8	7,1	6,3	5,1	6,8
ZORYA	9,8	8,9	10	10,5	9,5	10,4	13,5	12	14	13,3
LECH POZNAN	8	6,8	8,2	7,9	5,2	7	9,8	12	11,4	11,2
AMBERES	7,3	6,9	8,1	6,8	7	8,7	10	10	9,9	10,9
S.LIBEREC	7,3	6,3	5,3	8,5	7,2	7,8	12,3	11,3	10,8	11,7
DUNDALK	5,3	5,9	6	4,8	6	5,2	7,1	8,9	9,5	8,5
WOLFSBERGER	4,3	4,8	4,1	5,2	4,3	5,3	7,5	7,4	7,7	8,1
OMONIA	2,7	2,5	2,9	3,3	3,4	2,4	4,7	3,2	4	5,7
CSKA SOFIA	2,5	2,5	1,7	2,2	1,9	2,2	3	4,6	3	3,5
Corr.Lineal	97,33	97,17	97,47	96,75	97,30	97,38	92,94	93,18	93,04	93,26
Corr. Concordancia	88,89	88,88	89,11	88,31	89,14	88,93	85,68	85,78	85,08	85,50

Figura B.16: Porcentaje de veces que se clasifica cada equipo en los grupos de 3 con 1 clasificado ordenados por simulación según la última jornada y sus correlaciones

%	13GUYON	12GUYON	31	21	23	31GUYON	21GUYON	23GUYON	16 MEJOR
TOTTENHAM	83,9	85,4	78,2	78	76,5	74,8	74,9	73,5	10000
NAPOLES	78,5	80,6	74,9	72,9	71,9	66,8	68,4	72,4	10000
ARSENAL	79,5	80,2	72,6	73,2	71,3	69,5	68,4	68,9	9999
ROMA	73,5	71,3	67,2	67,3	69,1	64,1	61,7	64,1	9999
B.LEVERKUSEN	71,7	72	69,5	66,8	67,2	64,7	62,3	63,9	9994
BENFICA	67,6	65	63,5	63,2	62	59,4	59,7	61,7	9969
LEICESTER	63,4	62,1	58,6	60,4	58,6	60,1	58,9	57,5	9881
VILLARREAL	60,2	63,1	58,4	58,7	58,7	55,7	55,5	58,3	9838
CELTIC	73	73,6	59,4	59,9	58,1	61,6	62,5	64,8	9780
PSV	68,5	65,3	53,8	54,6	52,5	60,7	57,2	56,3	8668
MILAN	61	63,1	52,6	51,9	52,5	53,2	57,4	60,2	8204
CSKA MOSCU	61,6	62,9	48,9	52,4	51,4	55,1	54,2	54,8	7653
SC BRAGA	58,5	55,8	47,5	46,6	46,4	53,9	50,9	51,9	5842
GANTE	56,5	56,5	46,8	46,6	46,5	52,8	51,4	49,2	5177
LUDOGORETS	56,7	57	47,1	48,5	48,2	55,2	51,5	54,3	5050
DINAMO ZAGREB	54,6	55,5	49,1	46,6	44,5	51,3	48,7	48,1	4388
YOUNG BOYS	22,1	18,5	31,9	30,4	33,6	23,1	25,9	24,8	4235
REAL SOCIEDAD	20,5	21,8	32,7	32,5	32,8	28,4	26,4	23,7	3759
ESTRELLA ROJA	19	21,5	30,5	31	31,5	22,8	22,8	21,5	2942
SLAVIA PRAGA	19,6	17,8	28,7	31,9	27,8	23,8	22,8	25,1	2633
S.PRAGA	19,3	20,1	29,1	29,6	29,8	24,5	25,7	21,4	2689
PAOK	18,8	18,4	28,5	27,8	31	21,5	20,7	24,5	2219
HOFFENHEIM	17,5	20,1	29	26,8	30,8	20,5	23,3	20,9	1509
MOLDE	16	17,3	29,2	27,3	30,1	22,8	23,7	20,3	1432
AZ ALKMAR	35,8	35,4	26,9	29,4	28,2	37,5	37	34,3	1243
FEYERNOORD	32,5	29,7	26,4	29,4	24,3	33	34,1	32,1	541
RAPID VIENA	31,1	30,9	25,4	24,4	26,7	32	31,9	31,9	523
QARABAG	29,3	32,4	26,7	25,5	27,3	29,5	32	33,9	454
G.RANGERS	30,3	27,5	24,9	24,5	26,5	31,9	31,6	36,3	444
ST.LIEJA	28,8	30,8	22,4	22,4	23,6	29,6	31	29,7	374
AEK	25	27,8	24,1	24,7	23,3	32,7	30,7	30,5	164
GRANADA	27,2	28	22,8	24	22,8	26,9	29,7	28,1	131
MACCABI	9,9	9,3	17,7	18,2	18,1	13	15,1	14,9	106
CLUJ	9,9	8,5	19,2	20,7	18,8	14,6	13,1	13,5	54
LILLE	10,2	11,1	17,2	15,5	16,8	14,7	12	10,8	35
SIVASSPOR	8,4	7,7	17,6	16,2	18,7	11,8	12,4	14,6	33
LASK	8,6	7,7	15,3	15,8	18,1	12	12	10,3	13
NIZA	7,9	7,9	16,2	13,4	17	13,5	12,6	11,8	11
RIJEKA	7,3	6,9	15,5	15,3	15,1	9,1	11,7	10,9	10
APOEL.B.SHEVA	6,7	5,7	14,6	12,9	14,7	8,8	10	10,7	2
ZORYA	13,8	14,4	14,6	13,9	11,5	18,7	20,8	17,7	1
LECH POZNAN	12,4	9,1	11,9	13,4	11,9	14,2	16,9	15,6	0
AMBERES	9,1	11,8	12	13,4	12,6	14,2	15,6	16,3	0
S.LIBEREC	10,2	10,3	11,3	11,7	11,9	15,9	16,7	14,6	1
DUNDALK	9,9	7,1	8,9	9,2	9,4	10,8	12,8	11,3	0
WOLFSBERGER	7	7,8	9,3	9,3	8,8	14	12,5	12,3	0
OMONIA	3,7	5	6,6	6,3	6,8	8,2	7,2	8,2	0
CSKA SOFIA	3,5	2,3	4,8	5,6	4,3	7,1	5,7	7,6	0
Corr.Lineal	93,15	92,82	96,15	96,36	96,15	92,26	92,28	92,53	
Corr. Concordancia	85,12	85,00	78,84	78,89	77,80	75,80	74,66	76,32	

Figura B.17: Porcentaje de veces que se clasifica cada equipo en los grupos de 3 con 1 clasificado ordenados por simulación según la última jornada y sus correlaciones

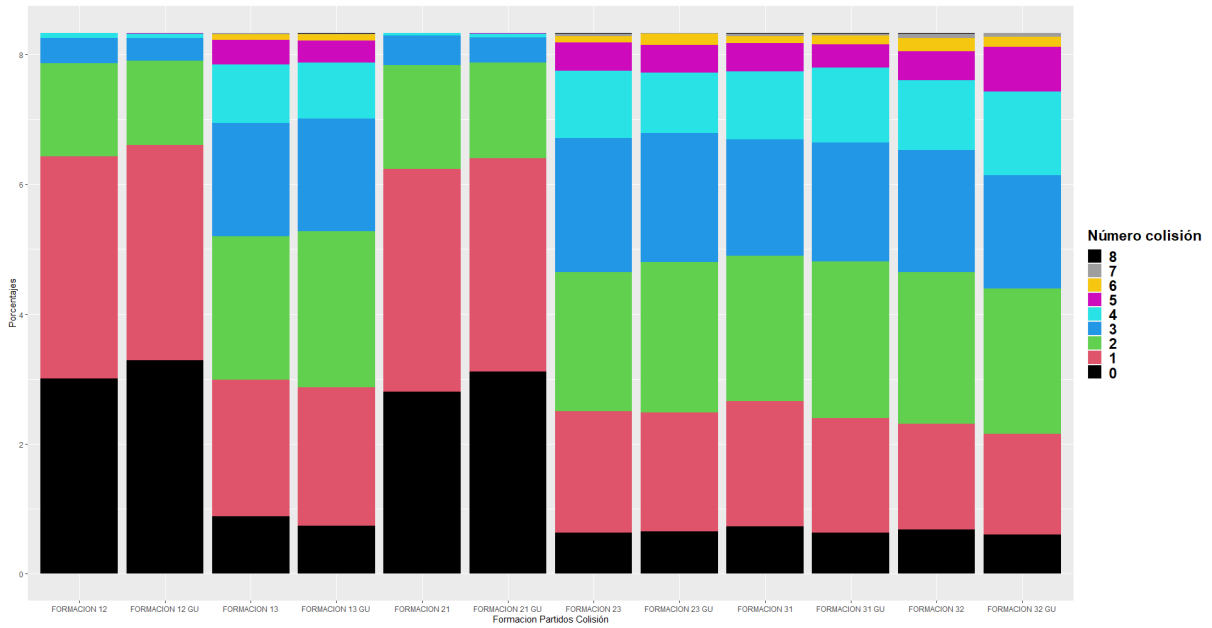


Figura B.18: Diagrama de barras apiladas de los partidos de colisión de los distintos órdenes ordenando los equipos por simulación.

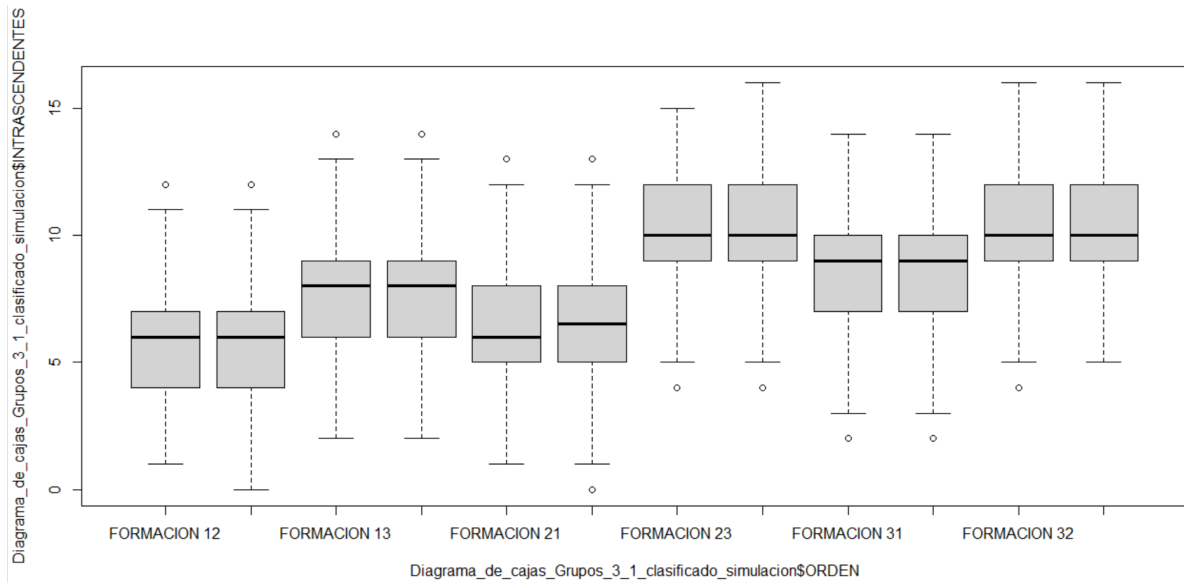


Figura B.19: Diagrama de cajas de los partidos intrascendentes de los distintos órdenes ordenando los equipos por simulación.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
COLISION 32	0	1	2	3	8	1.48	2.49
COLISION 32 GU	0	1	2	4	7	1.49	2.60
COLISION 31	0	1	2	3	8	1.43	2.37
COLISION 31 GU	0	1	2	3	8	1.39	2.38
COLISION 21	0	0	1	2	4	0.89	0.978
COLISION 21 GU	0	0	1	1	5	0.90	0.90

Tabla B.9: Estadísticos de los partidos de colisión para grupos de 3 con 1 clasificado ordenando los equipos por simulación.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
INTRASCENDENTES 32	4	9	10	12	16	1.92	10.33
INTRASCENDENTES 32 GU	5	9	10	12	16	1.85	10.46
INTRASCENDENTES 31	2	7	9	10	14	2.04	8.64
INTRASCENDENTES 31 GU	2	7	9	10	14	1.94	8.71
INTRASCENDENTES 21	1	5	6	8	13	1.99	6.46
INTRASCENDENTES 21 GU	0	5	6	8	13	1.99	6.49

Tabla B.10: Estadísticos de los partidos intrascendentes para grupos de 3 con 1 clasificado ordenando los equipos por simulación.

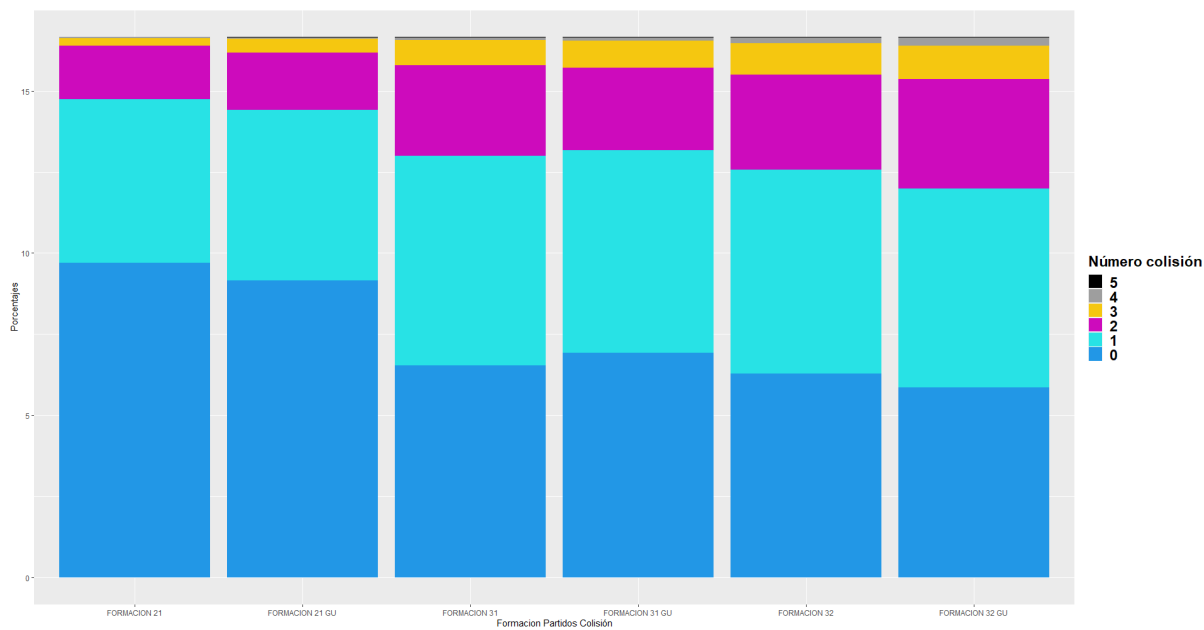


Figura B.20: Diagrama de barras apiladas de los partidos de colisión de los distintos órdenes con partido único ordenando los equipos por simulación.

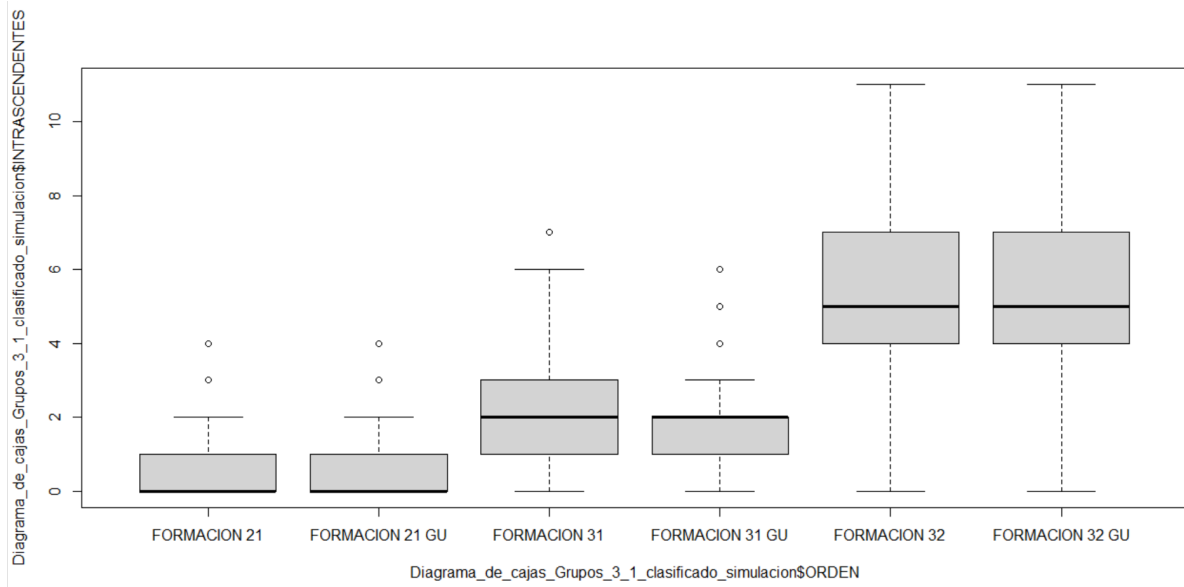


Figura B.21: Diagrama de cajas de los partidos intrascendentes de los distintos órdenes con partido único ordenando los equipos por simulación.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
COLISION 31	0	0	1	1	5	0.89	0.872
COLISION 31 GU	0	0	1	1	5	0.90	0.92
COLISION 21	0	0	0	1	4	0.74	0.55
COLISION 21 GU	0	0	0	1	5	0.80	0.58
COLISION 32	0	0	1	1	5	0.95	0.96
COLISION 32 GU	0	0	1	2	5	0.97	1.01

Tabla B.11: Estadísticos de los partidos de colisión para grupos de 3 con 1 clasificado con partido único ordenados por simulación

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
INTRASCENDENTES 31	0	1	2	3	7	1.26	1.72
INTRASCENDENTES 31 GU	0	1	2	2	6	1.24	1.69
INTRASCENDENTES 21	0	0	0	1	4	0.72	0.57
INTRASCENDENTES 21 GU	0	0	0	1	4	0.71	0.53
INTRASCENDENTES 32	0	4	5	7	11	1.88	5.18
INTRASCENDENTES 32 GU	0	4	5	7	11	1.84	5.29

Tabla B.12: Estadísticos de los partidos intrascendentes para grupos de 3 con 1 clasificado con partido único ordenados por simulación

Grupos de 3 con 2 clasificados ordenados por Ranking UEFA

%	32	31	21	23	13	12	31GUYON	31GUYON	21GUYON	21GUYON
ARSENAL	97	96,8	97,1	97,4	97,5	96,8	95,7	95,8	95,5	94,8
TOTTENHAM	98,3	99	98,4	97,8	97,7	98,5	97,7	97,6	98,3	97,3
ROMA	95,3	95,3	94,9	96	95,3	94,9	93,2	94	92,8	94,2
NAPOLIS	97	97,9	97,4	96,5	97	98	95,5	95,7	95,9	97,3
BENFICA	95,6	94,8	93,7	93,8	92,7	94,5	91,2	92,6	90,1	91,6
B.LEVERKUSEN	95,6	94,3	94,8	94,9	95,7	95,3	91,1	93,2	93,1	92,4
VILLARREAL	91,7	91,6	90,4	92,6	93,4	91,8	89,2	90,5	87,3	89,4
CSKA MOSCU	85,7	85,8	85,8	86	84,9	84,9	81,3	79,4	81,4	81,3
SC BRAGA	85,3	84,7	83,9	83,6	82,8	82	87,4	88	89,2	88,4
GANTE	83,2	81,7	81,3	81,7	83,8	85,9	89	86,8	86,1	86,5
PSV	87,3	87,6	89,3	87,5	86,8	88,6	91,4	92,4	90,4	90,5
CELTIC	91,9	88,6	89,4	93,6	91,2	90,4	94,1	94,5	94	93,8
DINAMO ZAGREB	84,5	81,8	83,9	80,7	80,7	83,3	86,8	86	87,1	85,8
S.PRAGA	80,6	79,2	80,6	77,4	79	80,9	85,7	83,2	84,6	85,5
SLAVIA PRAGA	79,7	80,6	81	79,9	78	79,5	82,9	83,2	83,8	83,5
LUDOGORETS	84,1	84,1	82,2	83,1	83,5	84,7	88,2	88,6	87,4	87,2
YOUNG BOYS	77,2	76,4	75,5	77,8	77,9	78,1	70,8	66,9	70	68,2
ESTRELLA ROJA	73,8	76	75,7	76,4	78,6	78,2	68,1	70,1	67,2	68,9
RAPID VIENA	69	70	68,2	66,6	71,9	71	60,3	59,7	58,8	61
LEICESTER	88,1	90,2	89,5	89,8	89,1	90	83,1	84,6	84	82,6
QARABAG	69,3	66,4	67,6	66,7	67,6	68,8	57,3	58,1	59,6	59,1
PAOK	71,8	75,6	76,4	77,4	74,7	71,4	66,1	64,3	66,7	68,5
ST.LIEJA	69,6	65,3	69,5	65,5	67,2	68	62,6	59,6	56,7	60,1
REAL SOCIEDAD	79,3	76,6	76,5	78,7	79	77	70,5	72	71,6	71,7
MILAN	84,4	84,1	83,9	83,7	83,2	83,1	89,1	89,4	91,8	89,8
AZ ALKMAR	76	72,1	72,9	76,8	72,4	72,8	81,1	79,5	78,9	79,7
FEYERNOORD	67,2	72	68	70,7	69,5	70,4	74,9	76,7	77,2	76,8
GRANADA	66,5	65,7	65,5	66	62,5	64,9	72,8	72,7	72,5	72,3
AEK	66,3	67,3	66,8	64,3	65,9	65,9	72,2	75,3	72,8	73,1
MACCABI	67	64,6	64,4	64,6	62,3	64,4	71,7	72,8	73,9	73
G.RANGERS	67,7	69,1	70,6	67,8	69,8	67,8	78,2	74,5	77,8	77,1
MOLDE	73,1	72,6	73,3	70,3	72,7	72,7	80,3	80,4	81,1	81,5
HOFFENHEIM	58,3	59,7	59,5	59,8	57,4	60,4	54,9	54,4	53,8	53,2
LASK	43,2	44,1	43,2	45,4	44	42,5	38,4	39,3	38,9	39,9
APOEL.B.SHEVA	39,1	41,8	38,8	40,8	41,2	39,9	31,9	34,6	36	34,6
SIVASSPOR	43,2	46,1	44,6	44,1	47,9	42,4	39,2	41,9	40,8	38,6
CLUJ	47,7	47,5	48,9	49,4	44,5	45,2	41,8	42,1	42,1	37,2
ZORYA	39,9	39,3	40,1	39,1	36,6	39,7	36,5	33,4	34,7	33,3
LILLE	47,2	46,5	50,6	46,1	47,2	48,6	43,9	43,2	43,2	43,5
NIZA	46,6	44,3	44,1	44,3	47,6	46,7	39,7	37	41,5	41,3
RIJEKA	40	42,6	40,9	42	45	42,2	44,5	48,1	44,8	46,2
DUNDALK	28,6	28,5	29,5	29,3	31	32,5	33,7	33,2	36	32,3
S.LIBEREC	35,2	36,1	34,7	34,7	34	34,2	37,3	38,7	37,5	40,1
AMBERES	34,1	34,8	35,2	33,6	34,6	33,5	38,4	40,9	39,2	39,9
LECH POZNAN	34,5	35,1	33,3	34,2	35,8	32,7	42,4	41,5	38,1	42,1
WOLFSBERGER	25,1	29	28,8	29,3	28,1	28,7	31,9	33,1	33,7	30,8
OMONIA	21	20,2	20,8	22,5	21,5	19,8	25,1	23,2	23,8	24,2
CSKA SOFIA	17,2	16,6	18,6	19,8	19,3	16,5	20,9	17,3	18,3	19,9
Corr.Lineal	95,63	95,57	95,93	95,89	95,64	95,82	92,33	91,41	91,85	91,84
Corr. Concordancia	84,70	83,95	84,17	84,09	83,67	84,65	81,53	80,99	81,14	81,29

Figura B.22: Porcentaje de veces que se clasifica cada equipo en los grupos de 3 con 2 clasificados ordenados por FIFA según la última jornada y sus correlaciones

%	13GUYON	12GUYON	31	21	23	31GUYON	21GUYON	23GUYON	32 MEJOR
ARSENAL	96,7	94,5	93,3	92	93,8	93,5	93,2	91,3	10000
TOTTENHAM	97	98,2	94,9	94,9	95	95	94,3	94,9	10000
ROMA	92,5	92,6	92,5	91,6	91	90,4	87,5	88,4	10000
NAPOLES	95,8	96	94,4	94,8	93,8	91,7	92,1	90,7	10000
BENFICA	91,4	90,9	88,6	89	88,3	87,5	87,3	86,5	10000
B.LEVERKUSEN	92,9	93,7	89,8	88,8	91,4	89,2	87,7	89,2	10000
VILLARREAL	88,8	89,3	85,1	86,9	88,3	86,5	83,6	83,6	10000
CSKA MOSCU	82,9	82	79,3	82	82	79,9	78,4	76,6	9982
SC BRAGA	89,3	88,6	78,1	78,8	77,1	82,1	82,9	81,9	9935
GANTE	87	86,3	76,8	79,9	80	84,3	81,3	82	9904
PSV	91,4	90,7	85,9	83,7	83,5	84,7	84,6	87,2	9990
CELTIC	93,9	93,6	85,1	87,7	86	89,2	88,7	87,7	10000
DINAMO ZAGREB	87,3	87,3	78,4	77,1	76,9	80,9	83,1	79,2	9843
S.PRAGA	83,2	84,7	77,4	73,8	75	81,7	80,9	79,3	9583
SLAVIA PRAGA	85,1	84	77	76,2	74,3	78,4	78,8	78,6	9577
LUDOGORETS	86,6	89	80	78,6	78,7	82	80,8	82,7	9893
YOUNG BOYS	69,2	69,9	75,7	76	73,1	68,9	66,5	66,4	9818
ESTRELLA ROJA	68,3	67,8	75,9	74,1	72,9	63,7	64,9	67,7	9609
RAPID VIENA	60,5	60,3	67,4	67	68,3	58,5	57,6	60,2	7760
LEICESTER	83,5	85	86,1	83,8	85,3	79,9	79,6	80,2	10000
QARABAG	60	58	69	67,1	68,3	60	60	57,7	7502
PAOK	66,2	67	72,9	72,4	71,7	66	67,4	67,1	9438
ST.LIEJA	55,9	58,4	65,8	65,9	66,8	57,2	60,1	59,1	7165
REAL SOCIEDAD	70,8	70,1	72,6	73,4	73,1	66,7	67,6	68,2	9810
MILAN	88,1	89,7	79,9	81,8	79,4	83,8	84,6	84,6	9994
AZ ALKMAR	79,5	78,9	72,4	68,9	68,7	74,2	77,9	74,7	8973
FEYERNOORD	77,8	79,8	69,5	65,9	66,9	73,2	73,6	73,3	7874
GRANADA	73,5	71,8	64,6	64,4	64,1	70,8	69,1	70,6	5939
AEK	73,5	73,1	64,1	64,2	64,6	71,7	71,5	71	5914
MACCABI	73	72	66,1	64,1	63,6	69,7	69,7	69,7	5140
G.RANGERS	76	77,4	66,9	68,4	69,5	71,8	72,6	75,6	7377
MOLDE	81,5	80,6	70	69,8	71,1	75,4	76,8	73,3	8903
HOFFENHEIM	55,3	53,8	60,2	61	61,5	53,6	59,8	57,3	9082
LASK	38,1	39,4	47,8	49,2	50,3	44,2	46,8	47,8	2733
APOEL.B.SHEVA	34,8	34,5	42,9	43,4	44,7	41,6	38,7	42	1164
SIVASSPOR	40,7	38,4	49	53,3	49,6	45,5	49	46,7	3292
CLUJ	40,4	40	52	49,5	52,4	44,4	47,9	47,5	3895
ZORYA	34,1	35,4	45,3	47,4	46,9	42,1	39,6	40,5	1147
LILLE	44,1	42,2	51,2	52,8	50,6	48,5	47	48,4	3803
NIZA	40,1	42,6	49,5	51,1	50,8	45,5	43,4	42	2608
RIJEKA	46,8	45,5	47,4	48,5	51,1	53,8	54,4	53,5	1621
DUNDALK	34,5	35,2	38	35,3	37,2	43,7	41,2	41,9	41
S.LIBEREC	40,1	38,5	44,3	42,1	42,9	47,3	46	46,9	224
AMBERES	39,3	37,5	41,5	42,8	41,6	46,3	47,1	45,9	193
LECH POZNAN	37,5	38,9	39,9	43,9	41,6	44,7	44,6	47,1	250
WOLFSBERGER	30,6	31,4	36,5	35,8	35,9	40,8	41,5	42,3	24
OMONIA	24,9	26,8	30,6	31,6	32,7	37,8	35,9	36,8	0
CSKA SOFIA	19,6	18,7	28,4	29,3	27,7	31,7	32,4	34,2	0
Corr.Lineal	92,05	92,01	94,73	94,31	94,03	89,04	90,72	90,16	
Corr. Concordancia	81,41	81,44	73,78	72,65	72,37	68,88	69,75	68,38	

Figura B.23: Porcentaje de veces que se clasifica cada equipo en los grupos de 3 con 2 clasificados ordenados por FIFA según la última jornada y sus correlaciones

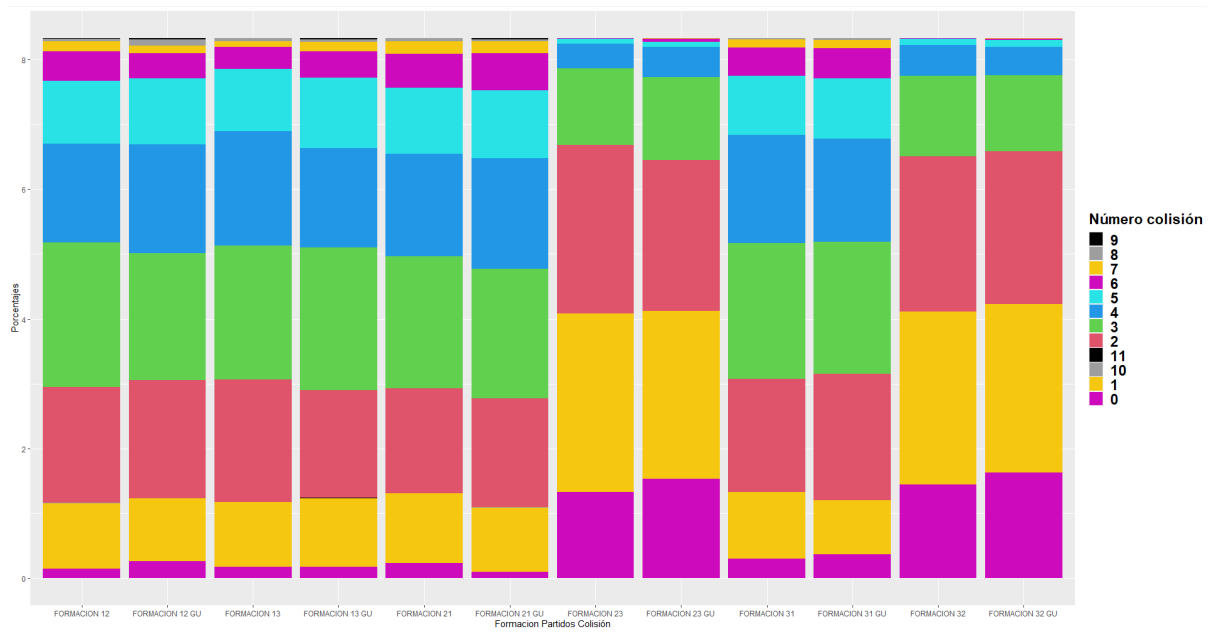


Figura B.24: Diagrama de barras apiladas de los partidos de colisión de los distintos órdenes para grupos de 3 con 2 clasificados.

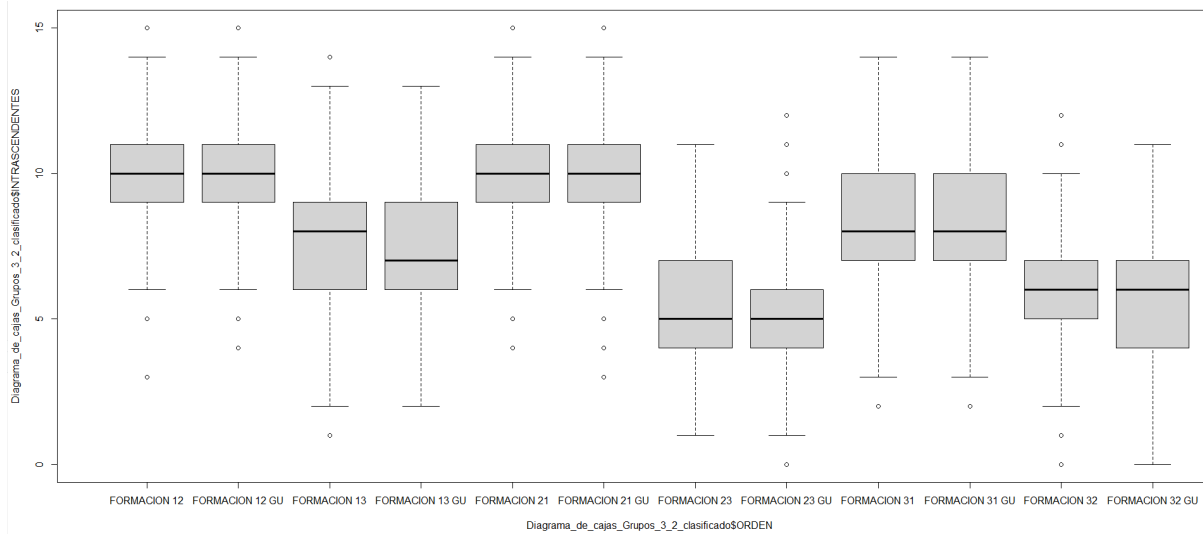


Figura B.25: Diagrama de cajas de los partidos intrascendentes de los distintos órdenes para grupos de 3 con 2 clasificados.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
COLISION 32	0	1	2	2	6	1.17	1.63
COLISION 32 GU	0	1	1	2	7	1.21	1.55
COLISION 31	0	2	3	4	8	1.56	3.09
COLISION 31 GU	0	2	3	4	8	1.58	3.11
COLISION 21	0	2	3	4	8	1.63	3.18
COLISION 21 GU	0	2	3	4	10	1.62	3.35

Tabla B.13: Estadísticos de los partidos de colisión para grupos de 3 con 2 clasificados

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
INTRASCENDENTES 32	0	5	6	7	12	1.88	5.71
INTRASCENDENTES 32 GU	0	4	6	7	11	1.90	5.72
INTRASCENDENTES 31	2	7	8	10	14	1.96	8.19
INTRASCENDENTES 31 GU	2	7	8	10	14	1.97	8.24
INTRASCENDENTES 21	4	9	10	11	15	1.91	10.03
INTRASCENDENTES 21 GU	3	9	10	11	15	1.90	9.95

Tabla B.14: Estadísticos de los partidos intrascendentes para grupos de 3 con 2 clasificados

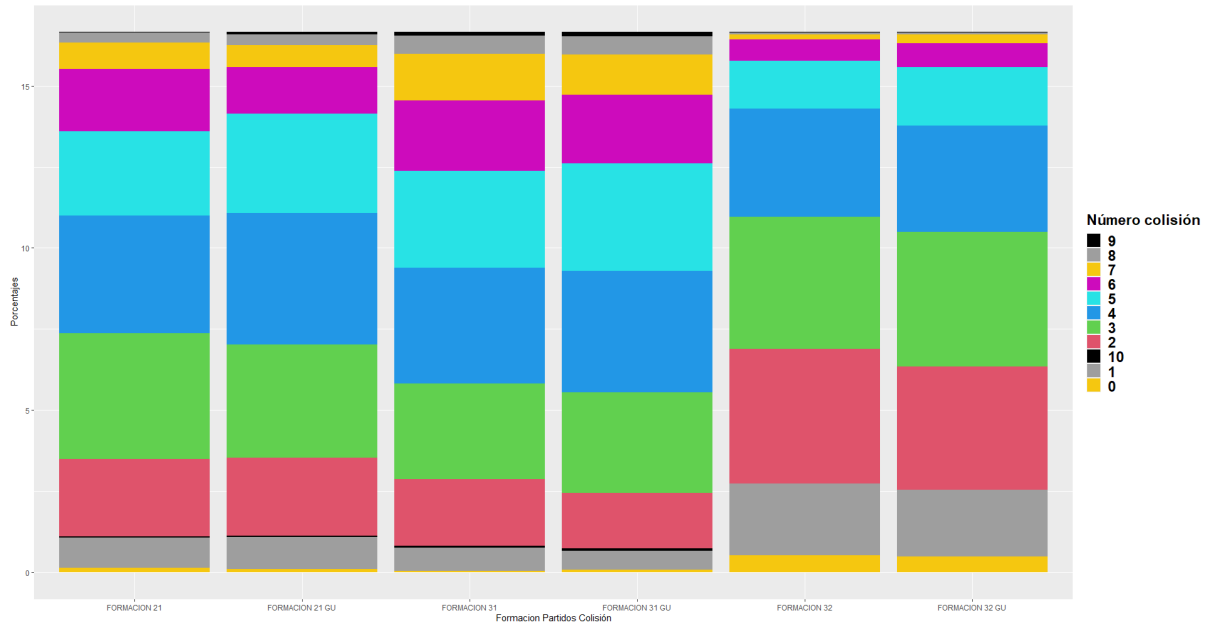


Figura B.26: Diagrama de barras apiladas de los partidos de colisión de los distintos órdenes con partido único para grupos de 3 con 2 clasificados.

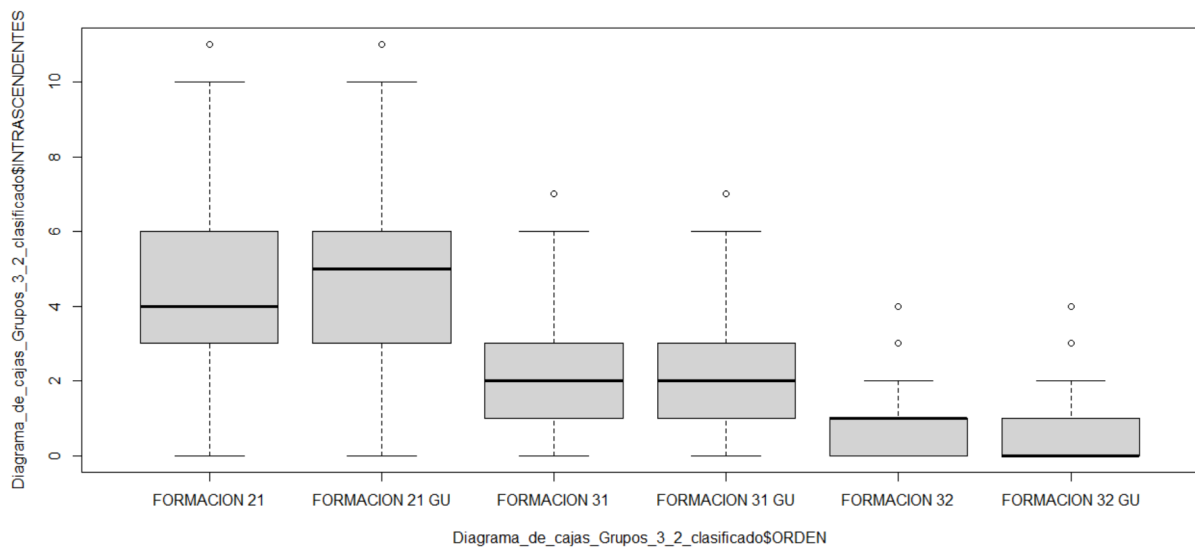


Figura B.27: Diagrama de cajas de los partidos intrascendentes de los distintos órdenes con partido único para grupos de 3 con 2 clasificados.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
COLISION 31	0	3	4	6	10	1.82	4.40
COLISION 31 GU	0	3	4	5	10	1.7	4.40
COLISION 21	0	3	4	5	10	0.91	3.87
COLISION 21 GU	0	3	4	5	10	1.67	3.93
COLISION 32	0	2	3	4	9	1.5	2.97
COLISION 32 GU	0	2	3	4	9	1.55	3.08

Tabla B.15: Estadísticos de los partidos de colisión para grupos de 3 con 2 clasificados con partido único

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
INTRASCENDENTES 31	0	1	2	3	7	1.26	1.76
INTRASCENDENTES 31 GU	0	1	2	3	7	1.30	1.84
INTRASCENDENTES 21	0	3	4	6	11	1.76	4.55
INTRASCENDENTES 21 GU	0	3	5	6	11	1.79	4.49
INTRASCENDENTES 32	0	0	1	1	4	0.78	0.72
INTRASCENDENTES 32 GU	0	0	0	1	4	0.8	0.67

Tabla B.16: Estadísticos de los partidos intrascendentes para grupos de 3 con 2 clasificados con partido único

Grupos de 3 con 2 clasificados ordenados por la simulación

%	32	31	21	23	13	12	32GUYON	31GUYON	21GUYON	23GUYON
TOTTENHAM	98,2	98,3	98,7	98,4	98,8	99,1	97,5	98	98	97
NAPOLES	98,2	96,9	97,1	98	97	98,2	97,1	97,4	96,3	95,2
ARSENAL	97,7	97,5	97,5	97,6	96	98	96,8	96,3	96,4	96,5
ROMA	95,9	94,6	95,5	95,2	96,3	95,5	93,6	92,9	94,5	94,3
B.LEVERKUSEN	95,3	95,2	95,6	94	94,4	95,6	93,7	93,6	94	92,4
BENFICA	94,3	92,7	95	93	92,5	94,3	93,6	92	91,6	92,8
LEICESTER	92,1	92,3	91	91,9	91,8	93,1	89	92	89,3	90,6
VILLARREAL	92,2	92,2	91,4	91,9	92,4	90,8	89,8	88,7	90,2	89,7
CELTIC	90,9	91	91,6	92	91	92,3	94,1	94,8	95,6	94,1
PSV	86,7	89,1	88,6	87,6	88,1	89,9	92,3	91,3	92,6	91,5
MILAN	88,1	86,4	87,7	88,9	88	88,8	91,8	90,5	90,1	90,4
CSKA MOSCU	86,4	86,4	86,7	86,7	89	88,2	92,1	91,6	91,5	91,6
SC BRAGA	84,4	83,9	84,4	84,7	85,5	84,8	90,4	89,6	88,6	89,5
GANTE	83,7	84,8	84,6	84,6	84,9	84	87	87,6	88	88,1
LUDOGORETS	80,7	82,4	83	83,6	84,2	85,6	86,5	89	88,6	87,8
DINAMO ZAGREB	82	82,6	83,5	80,7	82,1	81	87,3	87,3	86,5	86,6
YOUNG BOYS	77,7	79,8	78,8	77,8	76,2	78,8	69,9	68,6	68,3	68,6
REAL SOCIEDAD	78,3	79	76,1	77,7	77,4	77,9	68,4	72,8	74,5	70,8
ESTRELLA ROJA	78,4	76,1	74,2	74,9	75,8	74,7	68,3	67	68,6	69,3
SLAVIA PRAGA	73,5	73,5	75,8	74,3	76,6	75,1	65,8	67,2	67,9	65,9
S.PRAGA	76,4	74,8	74,7	71,8	74,7	75,9	67,6	69,7	65,4	69
PAOK	76,1	73,6	74,9	72,4	73,7	74,2	64,7	64,4	68	68,4
HOFFENHEIM	73,1	70,4	72,6	72,6	73,5	74	65,1	68,2	63,6	67,4
MOLDE	71,4	71,6	74,4	72,5	71,5	73,4	63,8	65,2	64	64,3
AZ ALKMAR	73,5	71	73,5	73,4	73,3	73,3	79,7	81,2	79,4	79,8
FEYERNOORD	68,4	68,4	68,8	72	68,2	65,4	75,3	74,3	76,2	75,1
RAPID VIENA	68,3	70,7	67,5	71,3	71,3	67,5	77,4	77,3	73,6	76,3
QARABAG	70,3	70,7	69,4	68,1	68,3	67,3	75,4	76,2	73,4	76,5
G.RANGERS	68,7	67,6	67	67,4	67	70,4	75,9	77,8	76,8	76
ST.LIEJA	66,7	70,1	67,7	67,2	67,1	70	73,1	73,6	75,8	74,6
AEK	65,2	63,1	65,9	65,6	67,4	67,5	73,3	72,2	73,2	73,6
GRANADA	67	65,1	66,3	65,6	64,4	65,7	72,7	74	74,5	72,8
MACCABI	49,7	49,1	50,2	48,6	47,4	49,1	44	43,5	40,3	41,1
CLUJ	48,7	48,8	48,5	46,8	46,5	49,4	40,8	37,1	40,3	38,6
LILLE	46,2	48,2	45,7	48,4	47,9	45,3	40,9	42,5	45,4	43,2
SIVASSPOR	47,2	46,2	45,7	46,4	45,9	48	38,8	39,5	38,6	37,1
LASK	46,5	47	43	44,6	45,7	42,2	39,6	35,8	38,5	38,7
NIZA	44,5	46	43	45,4	48,9	45,5	42,5	37,6	37,7	39,2
RIJEKA	42,4	42,5	42	42,8	42,2	40,1	35,1	37,5	33,6	33,5
APOEL B.SHEVA	37,4	39,6	41	40,6	38,7	39	33,6	32,5	35	36,4
ZORYA	37,2	42,3	39,1	41,6	38,7	38,1	45,7	45	47,6	41,9
LECH POZNAN	32,2	34,6	35,6	32,8	32,5	33,1	39	38,7	36,1	40,6
AMBERES	37,1	32,3	33,7	33,3	36,2	32,6	37,3	38	39,9	41,4
S.LIBEREC	34,3	32,4	34,1	35,1	35,6	32,9	37,5	36,4	38,9	39,7
DUNDALK	28	30,8	28,7	30,6	29,6	29,2	33,8	35,3	35	35,4
WOLFSBERGER	31,3	30,3	29,6	28,8	27,3	27,8	33,6	33,1	32,5	32,3
OMONIA	20,5	22	24,3	22,3	21,5	20,2	25,9	25,8	26,3	25
CSKA SOFIA	17	16,1	16,3	18,5	17	17,2	22,9	19,4	19,3	19,4
Corr.Lineal	96,34	96,17	96,31	96,07	96,21	96,38	91,78	92,18	96,69	91,98
Corr.Concordancia	85,20	84,58	85,07	84,45	84,99	86,08	81,51	82,54	81,81	81,89

Figura B.28: Porcentaje de veces que se clasifica cada equipo en los grupos de 3 con 2 clasificados ordenados por simulación según la última jornada y sus correlaciones

%	13GUYON	12GUYON	31	21	23	31GUYON	21GUYON	23GUYON	32 MEJOR
TOTTENHAM	96,6	97,7	95,1	95,4	95,5	95	93,7	94,2	10000
NAPOLES	96,7	96,7	93,1	94,2	94,2	91,7	93	92,1	10000
ARSENAL	95,5	96,6	94,2	93,1	93,7	91,9	92,2	92,2	10000
ROMA	94,6	94,8	91,2	90,3	90,3	87,9	88,6	89,3	10000
B.LEVERKUSEN	94	92,5	91,3	92,1	92,6	87,7	89,4	89,3	10000
BENFICA	90,8	92,7	88,3	88,8	88,7	86,8	88,3	85,8	10000
LEICESTER	91,7	88,4	86,3	90	87,9	84,2	84,4	87,6	10000
VILLARREAL	91	89,3	88,2	89,2	86,9	85,7	85,2	83,6	10000
CELTIC	93,5	93,6	87,4	89	87,4	90,3	90,6	89,3	10000
PSV	92,4	91	82,3	84,2	84,7	86,7	87,6	88,3	9990
MILAN	91,8	91,3	80,7	84,1	82,5	86,1	85,9	86,9	9994
CSKA MOSCU	90,5	89,3	83,3	81,6	84,3	84,6	86,1	85,5	9982
SC BRAGA	89,5	89,3	78,5	78,7	79,8	83,3	83	83,9	9935
GANTE	86,8	88,6	80,9	78,9	81,7	81,9	83,8	82,6	9904
LUDOGORETS	90,3	88	82,4	79	79,1	81	82,8	86,7	9893
DINAMO ZAGREB	86,8	85,5	77,5	77,8	77,4	83,8	82,8	82,5	9843
YOUNG BOYS	70,6	69,6	73,6	75,3	75,8	68	69,9	66,2	9818
REAL SOCIEDAD	69,9	71,5	75	73,7	72,3	70,3	69,5	68,1	9810
ESTRELLA ROJA	67,2	67,8	73,1	70,4	75,1	65,2	66	67,1	9609
SLAVIA PRAGA	67,3	68,8	72,2	72,7	73,2	67	65,7	67,1	9577
S.PRAGA	67,2	65,6	70,4	72,7	71,8	63,9	66,3	66,5	9583
PAOK	66,3	65,6	71	70	69,9	65,3	65,7	65,2	9438
HOFFENHEIM	65,1	65,2	73,9	70,1	70,8	64,6	62,2	66,6	9082
MOLDE	63,6	64	69,2	70,4	70,2	64,6	63,3	62,4	8903
AZ ALKMAR	78,3	80,8	69,9	70,7	70,3	75,9	77,2	74,8	8973
FEYERNOORD	77,5	74,5	66,1	70	67,8	73,7	73,2	73,3	7874
RAPID VIENA	76,6	77,1	65,9	67,2	64,9	72,7	70,8	69,7	7760
QARABAG	74,5	76	67,7	69,4	67,4	73,4	73	70,9	7502
G.RANGERS	73,6	76,1	65,6	65,5	66,7	71,1	70,9	73,7	7377
ST.LIEJA	76,8	74,5	65,4	67,9	66,8	69,5	69,3	71,8	7165
AEK	71,1	73,2	65,3	65,3	68,1	67,7	69,5	70	5914
GRANADA	76,3	73,4	63,5	65,7	63,3	68,3	69,7	69,5	5939
MACCABI	46,4	42	53,3	56,5	50,6	47,2	45	46,9	5140
CLUJ	39,3	41	53,4	50,3	54,3	49,7	45,3	49,5	3895
LILLE	42,1	43,1	51,4	50,2	53	46,9	47,1	47,3	3803
SIVASSPOR	37,7	40,7	50,2	50,1	49,7	43,6	45,7	47,2	3292
LASK	39,3	37,8	50,8	46,7	50,9	43,9	44,2	43,3	2733
NIZA	37,5	41	50,7	50,9	49,7	44	48	45,5	2608
RIJEKA	34,8	34,5	48,4	46,6	45,6	41,5	41,7	39,5	1621
APOEL.B.SHEVA	34,8	33,1	46,8	46,5	46,3	43,4	39,6	37,5	1164
ZORYA	43,8	43,8	43,6	45	43,2	50,9	52,8	49,3	1147
LECH POZNAN	37,9	39,1	44,1	40,3	42,2	51,1	47,6	46,6	250
AMBERES	40	40,3	43,8	40	43	48,7	48	45	193
S.LIBEREC	38,7	38,4	39,4	39,9	39,4	47,8	49,4	49,2	224
DUNDALK	34,5	35,2	38,3	37,6	34,6	42,3	39,3	40,5	41
WOLFSBERGER	33,6	33,5	36,9	36,9	38,5	40,3	39,8	42,8	24
OMONIA	24,9	27,4	31,5	32	30,2	36,3	34,2	33	0
CSKA SOFIA	20,3	20,1	28,9	27,1	27,7	32,6	32,7	34,2	0
Corr.Lineal	91,94	91,89	94,53	94,82	94,68	89,86	89,85	90,49	
Corr. Concordanc	81,93	81,41	73,20	74,81	74,60	69,03	70,22	70,91	

Figura B.29: Porcentaje de veces que se clasifica cada equipo en los grupos de 3 con 2 clasificados ordenados por simulación según la última jornada y sus correlaciones

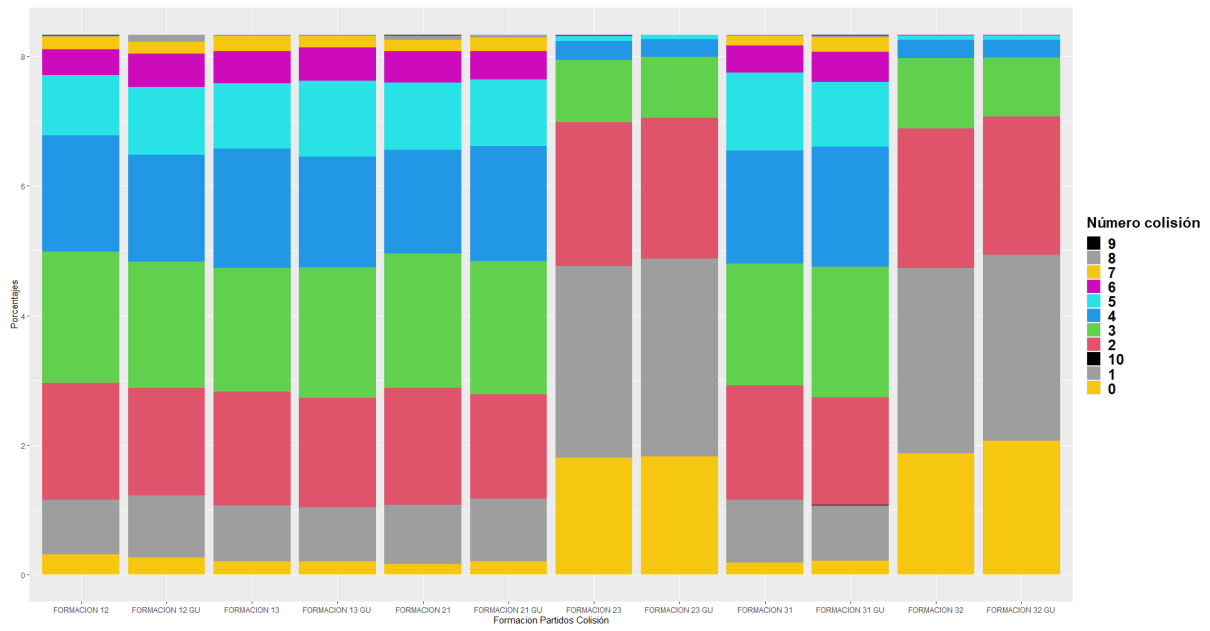


Figura B.30: Diagrama de barras apiladas de los partidos de colisión de los distintos órdenes ordenados por simulación para grupos de 3 con 2 clasificados ordenados por simulación.

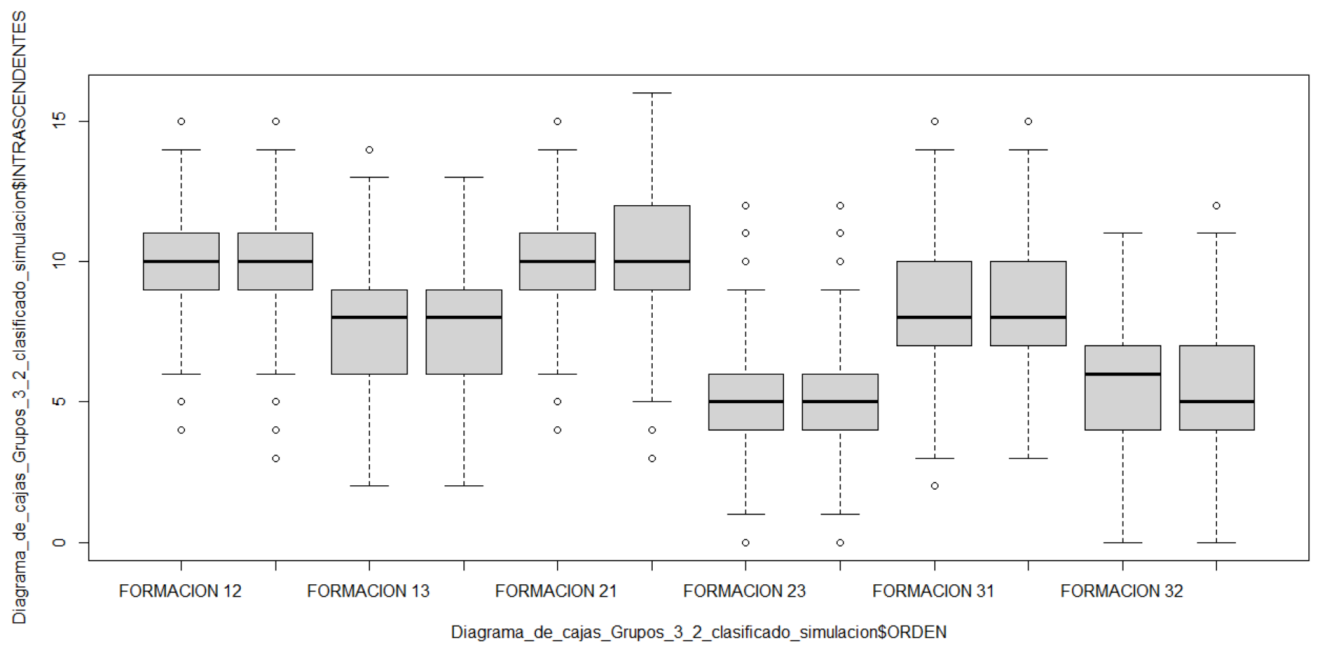


Figura B.31: Diagrama de cajas de los partidos intrasession de los distintos órdenes ordenados por simulación para grupos de 3 con 2 clasificados.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
COLISION 32	0	1	1	2	6	1.13	1.44
COLISION 32 GU	0	1	1	2	6	1.12	1.41
COLISION 31	0	2	3	4	8	1.55	3.17
COLISION 31 GU	0	2	3	4	10	1.60	3.26
COLISION 21	0	2	3	4	9	1.60	3.17
COLISION 21 GU	0	2	3	4	8	1.59	3.31

Tabla B.17: Estadísticos de los partidos de colisión para grupos de 3 con 2 clasificados ordenados por simulación.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
INTRASCENDENTES 32	0	4	6	7	11	1.89	5.54
INTRASCENDENTES 32 GU	0	4	5	7	12	1.88	5.41
INTRASCENDENTES 31	2	7	8	10	15	1.95	8.50
INTRASCENDENTES 31 GU	3	7	8	10	15	1.95	8.42
INTRASCENDENTES 21	4	9	10	11	15	1.95	10.16
INTRASCENDENTES 21 GU	3	9	10	12	16	1.94	10.07

Tabla B.18: Estadísticos de los partidos intrascendentes para grupos de 3 con 2 clasificados ordenados por simulación.

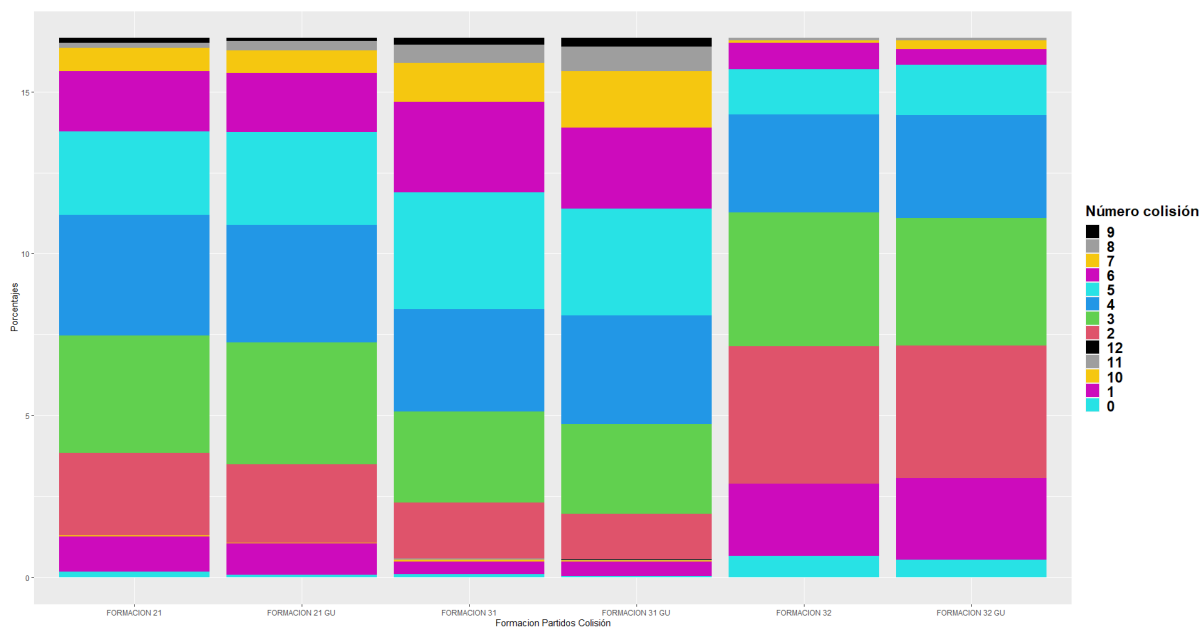


Figura B.32: Diagrama de barras apiladas de los partidos de colisión de los distintos órdenes con partido único ordenados por simulación para grupos de 3 con 2 clasificados.

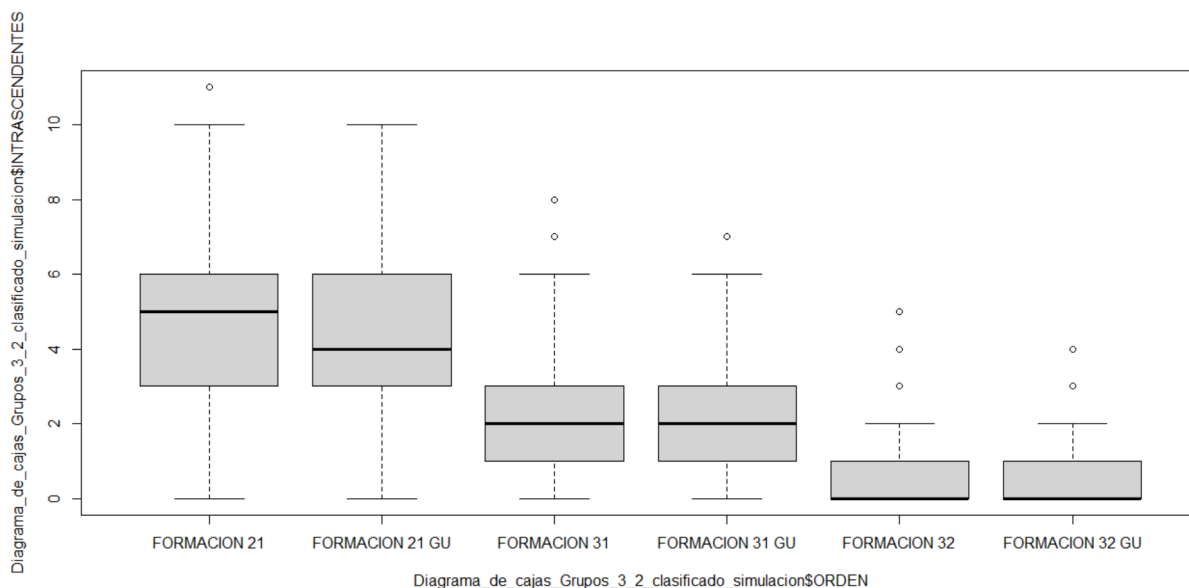


Figura B.33: Diagrama de cajas de los partidos intrascendentes de los distintos órdenes con partido único ordenados por simulación para grupos de 3 con 2 clasificados.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
COLISION 31	0	3	5	6	11	1.80	4.64
COLISION 31 GU	0	3	5	6	12	1.84	4.59
COLISION 21	0	3	4	5	10	1.73	3.82
COLISION 21 GU	0	3	4	5	10	1.69	3.97
COLISION 32	0	2	3	4	8	1.51	2.87
COLISION 32 GU	0	2	3	4	8	1.53	2.87

Tabla B.19: Estadísticos de los partidos de colisión para grupos de 3 con 2 clasificados con partido único ordenados por simulación.

	Mínimo	Q1	Median	Q3	Máximo	Desv.Típica	Media
INTRASCENDENTES 31	0	1	2	3	8	1.29	1.95
INTRASCENDENTES 31 GU	0	1	2	3	7	1.32	1.93
INTRASCENDENTES 21	0	3	5	6	11	1.81	4.62
INTRASCENDENTES 21 GU	0	3	4	6	10	1.80	4.54
INTRASCENDENTES 32	0	0	0	1	5	0.77	0.61
INTRASCENDENTES 32 GU	0	0	0	1	4	0.71	0.514

Tabla B.20: Estadísticos de los partidos intrascendentes para grupos de 3 con 2 clasificados con partido único ordenados por simulación.

Apéndice C

R-STUDIO

Regresión de Poisson en R ida y vuelta

Aquí se presenta el programa que hemos utilizado para sacar los coeficientes aplicados en nuestro modelo. En este caso lo hemos hecho con los equipos de la Champions League pero para los de la UEFA Europa League se haría exactamente igual, cambiando solamente las variables predictoras que nos saliesen más relevantes. De la misma forma, para los coeficientes de partido único, cambiaríamos solamente las variables predictoras resultantes del stepwise.

```
DATOSBUENO11<- DATOSBUENO1
library(Rcmdr)

DATOSBUENO11 <- within(DATOSBUENO11, {
  NIVEL_LIGA_EQUIPO_LOCAL<-as.factor(NIVEL_LIGA_EQUIPO_LOCAL)
  NIVEL_LIGA_EQUIPO_VISITANTE<-as.factor(NIVEL_LIGA_EQUIPO_VISITANTE)
})

#Definimos el modelo lineal generalizado para los goles del equipo local.
GolesLocal12<-glm( GOLES_DEL_LOCAL~ RANKING_UEFA_LOCAL+ VALOR_EQUIPO_LOCAL+ PROMEDIO_GOLES_MARCADOS_LOCAL*NIVEL_LIGA_EQUIPO_LOCAL
+ RANKING_UEFA_VISITANTE+ VALOR_EQUIPO_VISITANTE
+ PROMEDIO_GOLES_RECIBIDOS_VISITANTE*NIVEL_LIGA_EQUIPO_VISITANTE, family = poisson, data=DATOSBUENO11)

#Obtenemos el modelo con menor AIC

#####MODELO GOLES EQUIPO LOCAL
modeloAICGL12<-stepwise(GolesLocal12, direction= 'backward/forward', criterion='AIC')

Direction: backward/forward
Criterion: AIC

Start: AIC=669.65
GOLES_DEL_LOCAL ~ RANKING_UEFA_LOCAL + VALOR_EQUIPO_LOCAL + PROMEDIO_GOLES_MARCADOS_LOCAL *
  NIVEL_LIGA_EQUIPO_LOCAL + RANKING_UEFA_VISITANTE + VALOR_EQUIPO_VISITANTE +
  PROMEDIO_GOLES_RECIBIDOS_VISITANTE * NIVEL_LIGA_EQUIPO_VISITANTE

- PROMEDIO_GOLES_MARCADOS_LOCAL:NIVEL_LIGA_EQUIPO_LOCAL          Df Deviance  AIC
- RANKING_UEFA_VISITANTE                                         1  228.59 667.74
- PROMEDIO_GOLES_RECIBIDOS_VISITANTE:NIVEL_LIGA_EQUIPO_VISITANTE 2  232.40 669.55
<none>                                                           228.50 669.65
- VALOR_EQUIPO_VISITANTE                                         1  231.55 670.70
- VALOR_EQUIPO_LOCAL                                             1  232.61 671.76
- RANKING_UEFA_LOCAL                                             1  233.03 672.18

Step: AIC=665.83
GOLES_DEL_LOCAL ~ RANKING_UEFA_LOCAL + VALOR_EQUIPO_LOCAL + PROMEDIO_GOLES_MARCADOS_LOCAL +
  NIVEL_LIGA_EQUIPO_LOCAL + RANKING_UEFA_VISITANTE + VALOR_EQUIPO_VISITANTE +
  PROMEDIO_GOLES_RECIBIDOS_VISITANTE + NIVEL_LIGA_EQUIPO_VISITANTE +
  PROMEDIO_GOLES_RECIBIDOS_VISITANTE:NIVEL_LIGA_EQUIPO_VISITANTE

- NIVEL_LIGA_EQUIPO_LOCAL          Df Deviance  AIC
- PROMEDIO_GOLES_MARCADOS_LOCAL    1  228.71 663.86
- RANKING_UEFA_VISITANTE           1  228.79 663.94
<none>                             228.68 665.83
- PROMEDIO_GOLES_RECIBIDOS_VISITANTE:NIVEL_LIGA_EQUIPO_VISITANTE 2  232.72 665.87
- VALOR_EQUIPO_VISITANTE          1  231.60 666.75
- VALOR_EQUIPO_LOCAL              1  232.69 667.84
- RANKING_UEFA_LOCAL              1  233.44 668.59
```

```

+ PROMEDIO_GOLES_MARCADOS_LOCAL:NIVEL_LIGA_EQUIPO_LOCAL      2  228.50 669.65

Step: AIC=662.95
GOLES_DEL_LOCAL ~ RANKING_UEFA_LOCAL + VALOR_EQUIPO_LOCAL + PROMEDIO_GOLES_MARCADOS_LOCAL +
  RANKING_UEFA_VISITANTE + VALOR_EQUIPO_VISITANTE + PROMEDIO_GOLES_RECIBIDOS_VISITANTE +
  NIVEL_LIGA_EQUIPO_VISITANTE + PROMEDIO_GOLES_RECIBIDOS_VISITANTE:NIVEL_LIGA_EQUIPO_VISITANTE

      Df Deviance  AIC
- PROMEDIO_GOLES_MARCADOS_LOCAL      1  229.85 661.01
- RANKING_UEFA_VISITANTE              1  229.92 661.07
- PROMEDIO_GOLES_RECIBIDOS_VISITANTE:NIVEL_LIGA_EQUIPO_VISITANTE  2  233.72 662.87
<none>                                1  229.80 662.95
- VALOR_EQUIPO_VISITANTE              1  232.85 664.00
- RANKING_UEFA_LOCAL                  1  233.87 665.02
+ NIVEL_LIGA_EQUIPO_LOCAL              2  228.68 665.83
- VALOR_EQUIPO_LOCAL                  1  235.67 666.82

Step: AIC=661.01
GOLES_DEL_LOCAL ~ RANKING_UEFA_LOCAL + VALOR_EQUIPO_LOCAL + RANKING_UEFA_VISITANTE +
  VALOR_EQUIPO_VISITANTE + PROMEDIO_GOLES_RECIBIDOS_VISITANTE +
  NIVEL_LIGA_EQUIPO_VISITANTE + PROMEDIO_GOLES_RECIBIDOS_VISITANTE:NIVEL_LIGA_EQUIPO_VISITANTE

      Df Deviance  AIC
- RANKING_UEFA_VISITANTE              1  229.97 659.12
- PROMEDIO_GOLES_RECIBIDOS_VISITANTE:NIVEL_LIGA_EQUIPO_VISITANTE  2  233.81 660.97
<none>                                1  229.85 661.01
- VALOR_EQUIPO_VISITANTE              1  232.91 662.06
+ PROMEDIO_GOLES_MARCADOS_LOCAL        1  229.80 662.95
- RANKING_UEFA_LOCAL                  1  234.21 663.36
+ NIVEL_LIGA_EQUIPO_LOCAL              2  228.71 663.86
- VALOR_EQUIPO_LOCAL                  1  235.72 664.87

Step: AIC=659.12
GOLES_DEL_LOCAL ~ RANKING_UEFA_LOCAL + VALOR_EQUIPO_LOCAL + VALOR_EQUIPO_VISITANTE +
  PROMEDIO_GOLES_RECIBIDOS_VISITANTE + NIVEL_LIGA_EQUIPO_VISITANTE +
  PROMEDIO_GOLES_RECIBIDOS_VISITANTE:NIVEL_LIGA_EQUIPO_VISITANTE

      Df Deviance  AIC
- PROMEDIO_GOLES_RECIBIDOS_VISITANTE:NIVEL_LIGA_EQUIPO_VISITANTE  2  233.82 658.97
<none>                                1  229.97 659.12
+ RANKING_UEFA_VISITANTE              1  229.85 661.01
+ PROMEDIO_GOLES_MARCADOS_LOCAL        1  229.92 661.07
- VALOR_EQUIPO_VISITANTE              1  234.08 661.23
- RANKING_UEFA_LOCAL                  1  234.41 661.56
+ NIVEL_LIGA_EQUIPO_LOCAL              2  228.82 661.97
- VALOR_EQUIPO_LOCAL                  1  235.90 663.05

Step: AIC=658.97
GOLES_DEL_LOCAL ~ RANKING_UEFA_LOCAL + VALOR_EQUIPO_LOCAL + VALOR_EQUIPO_VISITANTE +
  PROMEDIO_GOLES_RECIBIDOS_VISITANTE + NIVEL_LIGA_EQUIPO_VISITANTE

      Df Deviance  AIC
- NIVEL_LIGA_EQUIPO_VISITANTE          2  234.53 655.69
- PROMEDIO_GOLES_RECIBIDOS_VISITANTE    1  233.83 656.98
<none>                                  1  233.82 658.97
+ PROMEDIO_GOLES_RECIBIDOS_VISITANTE:NIVEL_LIGA_EQUIPO_VISITANTE  2  229.97 659.12
+ PROMEDIO_GOLES_MARCADOS_LOCAL        1  233.72 660.87
+ RANKING_UEFA_VISITANTE              1  233.81 660.97
- VALOR_EQUIPO_LOCAL                  1  238.53 661.68
- VALOR_EQUIPO_VISITANTE              1  238.57 661.72
- RANKING_UEFA_LOCAL                  1  238.62 661.78
+ NIVEL_LIGA_EQUIPO_LOCAL              2  232.78 661.94

Step: AIC=655.69
GOLES_DEL_LOCAL ~ RANKING_UEFA_LOCAL + VALOR_EQUIPO_LOCAL + VALOR_EQUIPO_VISITANTE +
  PROMEDIO_GOLES_RECIBIDOS_VISITANTE

      Df Deviance  AIC
- PROMEDIO_GOLES_RECIBIDOS_VISITANTE    1  234.75 653.90
<none>                                  1  234.53 655.69
+ PROMEDIO_GOLES_MARCADOS_LOCAL        1  234.43 657.58
+ RANKING_UEFA_VISITANTE              1  234.51 657.66
- RANKING_UEFA_LOCAL                  1  239.10 658.25
+ NIVEL_LIGA_EQUIPO_LOCAL              2  233.72 658.87
+ NIVEL_LIGA_EQUIPO_VISITANTE          2  233.82 658.97
- VALOR_EQUIPO_LOCAL                  1  240.06 659.21
- VALOR_EQUIPO_VISITANTE              1  252.26 671.42

Step: AIC=653.9
GOLES_DEL_LOCAL ~ RANKING_UEFA_LOCAL + VALOR_EQUIPO_LOCAL + VALOR_EQUIPO_VISITANTE

      Df Deviance  AIC
<none>                                  1  234.75 653.90
+ PROMEDIO_GOLES_RECIBIDOS_VISITANTE    1  234.53 655.69
+ PROMEDIO_GOLES_MARCADOS_LOCAL        1  234.59 655.74
+ RANKING_UEFA_VISITANTE              1  234.69 655.84
- RANKING_UEFA_LOCAL                  1  239.36 656.51
+ NIVEL_LIGA_EQUIPO_VISITANTE          2  233.83 656.98
+ NIVEL_LIGA_EQUIPO_LOCAL              2  233.89 657.04
- VALOR_EQUIPO_LOCAL                  1  240.12 657.27
- VALOR_EQUIPO_VISITANTE              1  254.91 672.06
>

```

```

summary(modeloAICGL12)
coeficientesLocAIC12<-coefficients(summary(modeloAICGL12))

```



```
#Obtenemos los valores ajustados y los residuos deviance
DATOSBUENO11<-within(DATOSBUENO11,{
  fitted.modeloAICGL12<-fitted(modeloAICGL12)
  residuals.modeloAICGL12<-residuals(modeloAICGL12,type="deviance")
})
scatterplot(residuals.modeloAICGL12~fitted.modeloAICGL12, regLine=TRUE, smooth=list(span=0.5, spread=FALSE),
  boxplots=FALSE, data=DATOSBUENO11,id=list(method="mahal",n=2))
#Definimos el modelo para los goles visitantes
GolesVisitante12<-glm( GOLES_DEL_VISITANTE~RANKING_UEFA_VISITANTE+VALOR_EQUIPO_VISITANTE
  +PROMEDIO_GOLES_MARCADOS_VISITANTE*NIVEL_LIGA_EQUIPO_VISITANTE+RANKING_UEFA_LOCAL+VALOR_EQUIPO_LOCAL
  +PROMEDIO_GOLES_RECIBIDOS_LOCAL*NIVEL_LIGA_EQUIPO_LOCAL, family = poisson, data=DATOSBUENO11)

summary(GolesVisitante12)

##### MODELO GOLES EQUIPO VISITANTE

modeloAICGV12<-stepwise(GolesVisitante12, direction= 'backward/forward', criterion='AIC')

Direction: backward/forward
Criterion: AIC

Start: AIC=615.45
GOLES_DEL_VISITANTE ~ RANKING_UEFA_VISITANTE + VALOR_EQUIPO_VISITANTE +
  PROMEDIO_GOLES_MARCADOS_VISITANTE * NIVEL_LIGA_EQUIPO_VISITANTE +
  RANKING_UEFA_LOCAL + VALOR_EQUIPO_LOCAL + PROMEDIO_GOLES_RECIBIDOS_LOCAL *
  NIVEL_LIGA_EQUIPO_LOCAL

Df Deviance AIC
- PROMEDIO_GOLES_MARCADOS_VISITANTE:NIVEL_LIGA_EQUIPO_VISITANTE 2 239.46 613.09
- RANKING_UEFA_LOCAL 1 237.84 613.47
- PROMEDIO_GOLES_RECIBIDOS_LOCAL:NIVEL_LIGA_EQUIPO_LOCAL 2 240.30 613.93
- VALOR_EQUIPO_LOCAL 1 239.02 614.65
<none> 237.81 615.45
- RANKING_UEFA_VISITANTE 1 240.37 616.00
- VALOR_EQUIPO_VISITANTE 1 240.90 616.53

Step: AIC=613.09
GOLES_DEL_VISITANTE ~ RANKING_UEFA_VISITANTE + VALOR_EQUIPO_VISITANTE +
  PROMEDIO_GOLES_MARCADOS_VISITANTE + NIVEL_LIGA_EQUIPO_VISITANTE +
  RANKING_UEFA_LOCAL + VALOR_EQUIPO_LOCAL + PROMEDIO_GOLES_RECIBIDOS_LOCAL +
  NIVEL_LIGA_EQUIPO_LOCAL + PROMEDIO_GOLES_RECIBIDOS_LOCAL:NIVEL_LIGA_EQUIPO_LOCAL

Df Deviance AIC
- RANKING_UEFA_LOCAL 1 239.46 611.09
- PROMEDIO_GOLES_RECIBIDOS_LOCAL:NIVEL_LIGA_EQUIPO_LOCAL 2 242.13 611.76
- VALOR_EQUIPO_LOCAL 1 240.37 612.00
- NIVEL_LIGA_EQUIPO_VISITANTE 2 242.86 612.49
<none> 239.46 613.09
- RANKING_UEFA_VISITANTE 1 242.23 613.86
- VALOR_EQUIPO_VISITANTE 1 242.28 613.91
+ PROMEDIO_GOLES_MARCADOS_VISITANTE:NIVEL_LIGA_EQUIPO_VISITANTE 2 237.81 615.45
- PROMEDIO_GOLES_MARCADOS_VISITANTE 1 244.59 616.22

Step: AIC=611.09
GOLES_DEL_VISITANTE ~ RANKING_UEFA_VISITANTE + VALOR_EQUIPO_VISITANTE +
  PROMEDIO_GOLES_MARCADOS_VISITANTE + NIVEL_LIGA_EQUIPO_VISITANTE +
  VALOR_EQUIPO_LOCAL + PROMEDIO_GOLES_RECIBIDOS_LOCAL + NIVEL_LIGA_EQUIPO_LOCAL +
  PROMEDIO_GOLES_RECIBIDOS_LOCAL:NIVEL_LIGA_EQUIPO_LOCAL

Df Deviance AIC
- PROMEDIO_GOLES_RECIBIDOS_LOCAL:NIVEL_LIGA_EQUIPO_LOCAL 2 242.19 609.82
- VALOR_EQUIPO_LOCAL 1 240.50 610.13
- NIVEL_LIGA_EQUIPO_VISITANTE 2 242.87 610.50
<none> 239.46 611.09
- RANKING_UEFA_VISITANTE 1 242.24 611.87
- VALOR_EQUIPO_VISITANTE 1 242.29 611.92
+ RANKING_UEFA_LOCAL 1 239.46 613.09
+ PROMEDIO_GOLES_MARCADOS_VISITANTE:NIVEL_LIGA_EQUIPO_VISITANTE 2 237.84 613.47
- PROMEDIO_GOLES_MARCADOS_VISITANTE 1 244.59 614.22

Step: AIC=609.82
GOLES_DEL_VISITANTE ~ RANKING_UEFA_VISITANTE + VALOR_EQUIPO_VISITANTE +
  PROMEDIO_GOLES_MARCADOS_VISITANTE + NIVEL_LIGA_EQUIPO_VISITANTE +
  VALOR_EQUIPO_LOCAL + PROMEDIO_GOLES_RECIBIDOS_LOCAL + NIVEL_LIGA_EQUIPO_LOCAL

Df Deviance AIC
- PROMEDIO_GOLES_RECIBIDOS_LOCAL 1 242.22 607.85
- NIVEL_LIGA_EQUIPO_LOCAL 2 244.94 608.58
- NIVEL_LIGA_EQUIPO_VISITANTE 2 245.44 609.07
- VALOR_EQUIPO_LOCAL 1 243.54 609.17
<none> 242.19 609.82
- RANKING_UEFA_VISITANTE 1 245.02 610.65
- VALOR_EQUIPO_VISITANTE 1 245.29 610.92
+ PROMEDIO_GOLES_RECIBIDOS_LOCAL:NIVEL_LIGA_EQUIPO_LOCAL 2 239.46 611.09
+ RANKING_UEFA_LOCAL 1 242.13 611.76
+ PROMEDIO_GOLES_MARCADOS_VISITANTE:NIVEL_LIGA_EQUIPO_VISITANTE 2 240.31 611.94
- PROMEDIO_GOLES_MARCADOS_VISITANTE 1 248.11 613.74

Step: AIC=607.85
GOLES_DEL_VISITANTE ~ RANKING_UEFA_VISITANTE + VALOR_EQUIPO_VISITANTE +
  PROMEDIO_GOLES_MARCADOS_VISITANTE + NIVEL_LIGA_EQUIPO_VISITANTE +
  VALOR_EQUIPO_LOCAL + NIVEL_LIGA_EQUIPO_LOCAL

Df Deviance AIC
- NIVEL_LIGA_EQUIPO_LOCAL 2 245.05 606.68
- NIVEL_LIGA_EQUIPO_VISITANTE 2 245.44 607.07
- VALOR_EQUIPO_LOCAL 1 243.55 607.18
<none> 242.22 607.85
```

```
- RANKING_UEFA_VISITANTE          1  245.08 608.71
- VALOR_EQUIPO_VISITANTE          1  245.31 608.94
+ RANKING_UEFA_LOCAL              1  242.15 609.78
+ PROMEDIO_GOLES_RECIBIDOS_LOCAL  1  242.19 609.82
+ PROMEDIO_GOLES_MARCADOS_VISITANTE: NIVEL_LIGA_EQUIPO_VISITANTE 2  240.38 610.01
- PROMEDIO_GOLES_MARCADOS_VISITANTE 1  248.45 612.08
```

```
Step: AIC=606.68
GOLES_DEL_VISITANTE ~ RANKING_UEFA_VISITANTE + VALOR_EQUIPO_VISITANTE +
  PROMEDIO_GOLES_MARCADOS_VISITANTE + NIVEL_LIGA_EQUIPO_VISITANTE +
  VALOR_EQUIPO_LOCAL
```

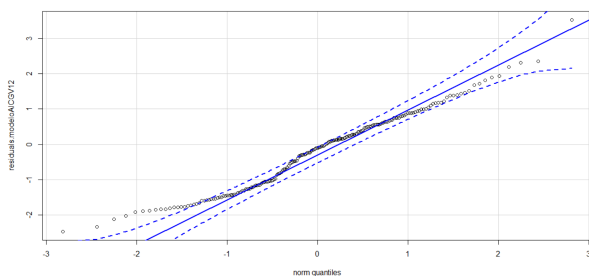
	Df	Deviance	AIC
- NIVEL_LIGA_EQUIPO_VISITANTE	2	248.66	606.29
<none>		245.05	606.68
- RANKING_UEFA_VISITANTE	1	247.76	607.39
+ NIVEL_LIGA_EQUIPO_LOCAL	2	242.22	607.85
+ PROMEDIO_GOLES_MARCADOS_VISITANTE: NIVEL_LIGA_EQUIPO_VISITANTE	2	242.72	608.35
- VALOR_EQUIPO_VISITANTE	1	248.79	608.42
+ PROMEDIO_GOLES_RECIBIDOS_LOCAL	1	244.94	608.58
+ RANKING_UEFA_LOCAL	1	245.03	608.66
- PROMEDIO_GOLES_MARCADOS_VISITANTE	1	250.49	610.12
- VALOR_EQUIPO_LOCAL	1	259.41	619.04

```
Step: AIC=606.29
GOLES_DEL_VISITANTE ~ RANKING_UEFA_VISITANTE + VALOR_EQUIPO_VISITANTE +
  PROMEDIO_GOLES_MARCADOS_VISITANTE + VALOR_EQUIPO_LOCAL
```

	Df	Deviance	AIC
<none>		248.66	606.29
+ NIVEL_LIGA_EQUIPO_VISITANTE	2	245.05	606.68
+ NIVEL_LIGA_EQUIPO_LOCAL	2	245.44	607.07
- VALOR_EQUIPO_VISITANTE	1	251.51	607.14
- RANKING_UEFA_VISITANTE	1	252.12	607.75
+ RANKING_UEFA_LOCAL	1	248.60	608.23
+ PROMEDIO_GOLES_RECIBIDOS_LOCAL	1	248.64	608.27
- PROMEDIO_GOLES_MARCADOS_VISITANTE	1	256.85	612.48
- VALOR_EQUIPO_LOCAL	1	262.90	618.53

```
summary(modeloAICGV12)
coeficientesVISAIC12<-coefficients(summary(modeloAICGV12))
DATOSBUENO11<-within(DATOSBUENO11,{
  fitted.modeloAICGV12<-fitted(modeloAICGV12)
  residuals.modeloAICGV12<-residuals(modeloAICGV12,type="deviance")
})

scatterplot(residuals.modeloAICGV12~fitted.modeloAICGV12, regLine=TRUE, smooth=list(span=0.5, spread=FALSE),
  boxplots=FALSE, data=DATOSBUENO11,id=list(method="mahal",n=2))
#Se estudia la sobredispersión, obtenemos parametro y realizamos test de equidispersión.
sum((residuals(modeloAICGV12, type="deviance"))^2)/modeloAICGV12$df.residual#SOBREDISPERSION LOCAL
library(AER)
dispersiontest(modeloAICGV12)
#Comprobamos que los goles de local sean independientes de los visitantes, nos fijamos en los residuos.
cor.test(DATOSBUENO11$residuals.modeloAICGL12,DATOSBUENO11$residuals.modeloAICGV12)
with(DATOSBUENO11, qqPlot(residuals.modeloAICGL12, dist = "norm",
  id = list(method = "y", n = 0)))
with(DATOSBUENO11, qqPlot(residuals.modeloAICGV12, dist = "norm",
  id = list(method = "y", n = 0)))
```



```
#Calculamos los landas para los equipos de la Champions League
attach(EQUIPOS_2020_2021)
landaLOC<-function(j,i){
  return(exp( coeficientesLocAIC12[1]+coeficientesLocAIC12[2]*RANKING_UEFA[i] +coeficientesLocAIC12[3]*VALOR_EQUIPO[i]+
  coeficientesLocAIC12[4]*VALOR_EQUIPO[j]))}
landaVIS<-function(j,i){
  return(exp( coeficientesVISAIC12[1]+ coeficientesVISAIC12[3]*VALOR_EQUIPO[j]
  + coeficientesVISAIC12[4] *PROMEDIO_GOLES_MARCADOS[j]
  + coeficientesVISAIC12[5]*VALOR_EQUIPO[i]
  + coeficientesVISAIC12[2]*RANKING_UEFA[j]
  ))}
N<-32
A123<-matrix(N*N,N,N)
i<-1
j<-1
while(i<=N){
  A123[i,i]<-0
  i<-i+1
```

```

}
i<-1
j<-i+1
while(i<=N-1){
  while(j<=N){
    A123[i,j]<-landaLOC(j,i)
    j<-j+1
  }
  i<-i+1
  j<-i+1
}
j<-1
i<-j+1

while(j<=N-1){
  while(i<=N){
    A123[i,j]<-landaVIS(i,j)
    i<-i+1
  }
  j<-j+1
  i<-j+1
}

#Sacamos las probabilidades de cada resultado en cada partido
PROBPARTIDOS1<-function(j,i,k){
  return(dpois(i,LANDASDATOSBUENO1[j,i])*dpois(k,LANDASDATOSBUENO1[j,2]))}
TABLAPROBPARTIDOS1<-matrix(204*36,204,36)
j<-1
i<-0
k<-0
h<-1
while (j<=204){
  while(i<=5){
    while(k<=5){
      TABLAPROBPARTIDOS1[j,h]<-PROBPARTIDOS1(j,i,k)
      k<-k+1
      h<-h+1
    }
    i<-i+1
    k<-0
  }
  j<-j+1
  h<-1
  i<-0
  k<-0
}
TABLAPROBPARTIDOS1<-as.data.frame(TABLAPROBPARTIDOS1)
#Sumamos cuantos resultados de cada tipo hemos predicho.
sumapartidos<-colSums(TABLAPROBPARTIDOS1)

#CURVAS ROC
library(PROC)
rocVL1<-roc(DATOSBUENO1$HOME,DATOSBUENO1$ProbVicL1)
auc(rocVL1)
plot(rocVL1,col="red",print.auc=TRUE)

rocVV1<-roc(DATOSBUENO1$AWAY,DATOSBUENO1$ProbVicV1)
auc(rocVV1)
plot(rocVV1,col="red",print.auc=TRUE)

rocEE1<-roc(DATOSBUENO1$TIE,DATOSBUENO1$ProbEmp1)
auc(rocEE1)
plot(rocEE1,col="red",print.auc=TRUE)

##### SEGUIMOS EL MISMO PROCEDIMIENTO PARA PARTIDO ÚNICO

DATOSBUENO1PARTIDOUNICO11<-DATOSBUENO1PARTIDOUNICO
library(Rcmdr)
DATOSBUENO1PARTIDOUNICO11<- within(DATOSBUENO1PARTIDOUNICO11, {
  NIVEL_LIGA_EQUIPO_EQ1<-as.factor(NIVEL_LIGA_EQUIPO_EQ1)
  NIVEL_LIGA_EQUIPO_EQ2<-as.factor(NIVEL_LIGA_EQUIPO_EQ2)
})

GolesL12<-glm( GOLES_EQ1~ RANKING_UEFA_EQ1 + VALOR_EQUIPO_EQ1+ PROMEDIO_GOLES_MARCADOS_EQ1
+PROMEDIO_GOLES_RECIBIDOS_EQ1*NIVEL_LIGA_EQUIPO_EQ1+ RANKING_UEFA_EQ2+ VALOR_EQUIPO_EQ2+
PROMEDIO_GOLES_MARCADOS_EQ2
+ PROMEDIO_GOLES_RECIBIDOS_EQ2*NIVEL_LIGA_EQUIPO_EQ2, family = poisson, data=DATOSBUENO1PARTIDOUNICO11)
mean(DATOSBUENO1PARTIDOUNICO11$GOLES_EQ1)
modeloAICGLPU12<-stepwise(GolesL12, direction= 'backward/forward', criterion='AIC')

#####MODELO GOLES PARTIDO ÚNICO

Direction: backward/forward
Criterion: AIC

Start: AIC=1281.05
GOLES_EQ1 ~ RANKING_UEFA_EQ1 + VALOR_EQUIPO_EQ1 + PROMEDIO_GOLES_MARCADOS_EQ1 +
PROMEDIO_GOLES_RECIBIDOS_EQ1 * NIVEL_LIGA_EQUIPO_EQ1 + RANKING_UEFA_EQ2 +
VALOR_EQUIPO_EQ2 + PROMEDIO_GOLES_MARCADOS_EQ2 + PROMEDIO_GOLES_RECIBIDOS_EQ2 *
NIVEL_LIGA_EQUIPO_EQ2

Df Deviance AIC
- PROMEDIO_GOLES_RECIBIDOS_EQ1:NIVEL_LIGA_EQUIPO_EQ1 2 489.34 1278.1
- PROMEDIO_GOLES_MARCADOS_EQ2 1 488.27 1279.0
- RANKING_UEFA_EQ2 1 488.31 1279.1
- PROMEDIO_GOLES_RECIBIDOS_EQ2:NIVEL_LIGA_EQUIPO_EQ2 2 491.52 1280.3

```

```

<none>                                488.27 1281.0
- PROMEDIO_GOLES_MARCADOS_EQ1          1  490.75 1281.5
- VALOR_EQUIPO_EQ2                     1  491.84 1282.6
- RANKING_UEFA_EQ1                     1  494.45 1285.2
- VALOR_EQUIPO_EQ1                     1  496.03 1286.8

Step: AIC=1278.12
GOLES_EQ1 ~ RANKING_UEFA_EQ1 + VALOR_EQUIPO_EQ1 + PROMEDIO_GOLES_MARCADOS_EQ1 +
  PROMEDIO_GOLES_RECIBIDOS_EQ1 + NIVEL_LIGA_EQUIPO_EQ1 + RANKING_UEFA_EQ2 +
  VALOR_EQUIPO_EQ2 + PROMEDIO_GOLES_MARCADOS_EQ2 + PROMEDIO_GOLES_RECIBIDOS_EQ2 +
  NIVEL_LIGA_EQUIPO_EQ2 + PROMEDIO_GOLES_RECIBIDOS_EQ2:NIVEL_LIGA_EQUIPO_EQ2

                                Df Deviance  AIC
- NIVEL_LIGA_EQUIPO_EQ1          2  490.27 1275.0
- PROMEDIO_GOLES_MARCADOS_EQ2    1  489.34 1276.1
- RANKING_UEFA_EQ2               1  489.39 1276.2
- PROMEDIO_GOLES_RECIBIDOS_EQ1   1  489.81 1276.6
- PROMEDIO_GOLES_RECIBIDOS_EQ2:NIVEL_LIGA_EQUIPO_EQ2  2  492.88 1277.7
<none>                            489.34 1278.1
- PROMEDIO_GOLES_MARCADOS_EQ1    1  491.69 1278.5
- VALOR_EQUIPO_EQ2              1  493.16 1279.9
+ PROMEDIO_GOLES_RECIBIDOS_EQ1:NIVEL_LIGA_EQUIPO_EQ1  2  488.27 1281.0
- VALOR_EQUIPO_EQ1              1  496.53 1283.3
- RANKING_UEFA_EQ1              1  496.68 1283.5

Step: AIC=1275.05
GOLES_EQ1 ~ RANKING_UEFA_EQ1 + VALOR_EQUIPO_EQ1 + PROMEDIO_GOLES_MARCADOS_EQ1 +
  PROMEDIO_GOLES_RECIBIDOS_EQ1 + RANKING_UEFA_EQ2 + VALOR_EQUIPO_EQ2 +
  PROMEDIO_GOLES_MARCADOS_EQ2 + PROMEDIO_GOLES_RECIBIDOS_EQ2 +
  NIVEL_LIGA_EQUIPO_EQ2 + PROMEDIO_GOLES_RECIBIDOS_EQ2:NIVEL_LIGA_EQUIPO_EQ2

                                Df Deviance  AIC
- PROMEDIO_GOLES_MARCADOS_EQ2    1  490.28 1273.1
- RANKING_UEFA_EQ2               1  490.37 1273.2
- PROMEDIO_GOLES_RECIBIDOS_EQ1   1  490.50 1273.3
- PROMEDIO_GOLES_RECIBIDOS_EQ2:NIVEL_LIGA_EQUIPO_EQ2  2  493.65 1274.4
<none>                            490.27 1275.0
- VALOR_EQUIPO_EQ2              1  493.61 1276.4
- PROMEDIO_GOLES_MARCADOS_EQ1    1  494.36 1277.1
+ NIVEL_LIGA_EQUIPO_EQ1          2  489.34 1278.1
- VALOR_EQUIPO_EQ1              1  497.67 1280.5
- RANKING_UEFA_EQ1              1  497.67 1280.5

Step: AIC=1273.06
GOLES_EQ1 ~ RANKING_UEFA_EQ1 + VALOR_EQUIPO_EQ1 + PROMEDIO_GOLES_MARCADOS_EQ1 +
  PROMEDIO_GOLES_RECIBIDOS_EQ1 + RANKING_UEFA_EQ2 + VALOR_EQUIPO_EQ2 +
  PROMEDIO_GOLES_RECIBIDOS_EQ2 + NIVEL_LIGA_EQUIPO_EQ2 + PROMEDIO_GOLES_RECIBIDOS_EQ2:NIVEL_LIGA_EQUIPO_EQ2

                                Df Deviance  AIC
- RANKING_UEFA_EQ2               1  490.39 1271.2
- PROMEDIO_GOLES_RECIBIDOS_EQ1   1  490.51 1271.3
- PROMEDIO_GOLES_RECIBIDOS_EQ2:NIVEL_LIGA_EQUIPO_EQ2  2  493.65 1272.4
<none>                            490.28 1273.1
- VALOR_EQUIPO_EQ2              1  493.81 1274.6
+ PROMEDIO_GOLES_MARCADOS_EQ2    1  490.27 1275.0
- PROMEDIO_GOLES_MARCADOS_EQ1    1  494.39 1275.2
+ NIVEL_LIGA_EQUIPO_EQ1          2  489.34 1276.1
- VALOR_EQUIPO_EQ1              1  497.69 1278.5
- RANKING_UEFA_EQ1              1  497.69 1278.5

Step: AIC=1271.17
GOLES_EQ1 ~ RANKING_UEFA_EQ1 + VALOR_EQUIPO_EQ1 + PROMEDIO_GOLES_MARCADOS_EQ1 +
  PROMEDIO_GOLES_RECIBIDOS_EQ1 + VALOR_EQUIPO_EQ2 + PROMEDIO_GOLES_RECIBIDOS_EQ2 +
  NIVEL_LIGA_EQUIPO_EQ2 + PROMEDIO_GOLES_RECIBIDOS_EQ2:NIVEL_LIGA_EQUIPO_EQ2

                                Df Deviance  AIC
- PROMEDIO_GOLES_RECIBIDOS_EQ1   1  490.63 1269.4
- PROMEDIO_GOLES_RECIBIDOS_EQ2:NIVEL_LIGA_EQUIPO_EQ2  2  493.71 1270.5
<none>                            490.39 1271.2
+ RANKING_UEFA_EQ2               1  490.28 1273.1
+ PROMEDIO_GOLES_MARCADOS_EQ2    1  490.37 1273.2
- PROMEDIO_GOLES_MARCADOS_EQ1    1  494.50 1273.3
- VALOR_EQUIPO_EQ2              1  495.10 1273.9
+ NIVEL_LIGA_EQUIPO_EQ1          2  489.39 1274.2
- VALOR_EQUIPO_EQ1              1  497.84 1276.6
- RANKING_UEFA_EQ1              1  497.92 1276.7

Step: AIC=1269.41
GOLES_EQ1 ~ RANKING_UEFA_EQ1 + VALOR_EQUIPO_EQ1 + PROMEDIO_GOLES_MARCADOS_EQ1 +
  VALOR_EQUIPO_EQ2 + PROMEDIO_GOLES_RECIBIDOS_EQ2 + NIVEL_LIGA_EQUIPO_EQ2 +
  PROMEDIO_GOLES_RECIBIDOS_EQ2:NIVEL_LIGA_EQUIPO_EQ2

                                Df Deviance  AIC
- PROMEDIO_GOLES_RECIBIDOS_EQ2:NIVEL_LIGA_EQUIPO_EQ2  2  493.80 1268.6
<none>                            490.63 1269.4
+ PROMEDIO_GOLES_RECIBIDOS_EQ1   1  490.39 1271.2
+ RANKING_UEFA_EQ2               1  490.51 1271.3
+ PROMEDIO_GOLES_MARCADOS_EQ2    1  490.61 1271.4
- PROMEDIO_GOLES_MARCADOS_EQ1    1  494.83 1271.6
- VALOR_EQUIPO_EQ2              1  495.18 1272.0
+ NIVEL_LIGA_EQUIPO_EQ1          2  489.89 1272.7
- VALOR_EQUIPO_EQ1              1  498.41 1275.2
- RANKING_UEFA_EQ1              1  498.61 1275.4

Step: AIC=1268.59
GOLES_EQ1 ~ RANKING_UEFA_EQ1 + VALOR_EQUIPO_EQ1 + PROMEDIO_GOLES_MARCADOS_EQ1 +
  VALOR_EQUIPO_EQ2 + PROMEDIO_GOLES_RECIBIDOS_EQ2 + NIVEL_LIGA_EQUIPO_EQ2

                                Df Deviance  AIC

```

```
- PROMEDIO_GOLES_RECIBIDOS_EQ2          1  493.81 1266.6
- NIVEL_LIGA_EQUIPO_EQ2                  2  495.84 1266.6
<none>                                   493.80 1268.6
+ PROMEDIO_GOLES_RECIBIDOS_EQ2:NIVEL_LIGA_EQUIPO_EQ2  2  490.63 1269.4
+ PROMEDIO_GOLES_RECIBIDOS_EQ1          1  493.71 1270.5
+ RANKING_UEFA_EQ2                       1  493.74 1270.5
+ PROMEDIO_GOLES_MARCADOS_EQ2           1  493.80 1270.6
- PROMEDIO_GOLES_MARCADOS_EQ1           1  498.64 1271.4
+ NIVEL_LIGA_EQUIPO_EQ1                  2  493.15 1271.9
- VALOR_EQUIPO_EQ2                       1  499.23 1272.0
- VALOR_EQUIPO_EQ1                       1  500.83 1273.6
- RANKING_UEFA_EQ1                       1  502.09 1274.9
```

```
Step: AIC=1266.59
GOLES_EQ1 ~ RANKING_UEFA_EQ1 + VALOR_EQUIPO_EQ1 + PROMEDIO_GOLES_MARCADOS_EQ1 +
  VALOR_EQUIPO_EQ2 + NIVEL_LIGA_EQUIPO_EQ2
```

	Df	Deviance	AIC
- NIVEL_LIGA_EQUIPO_EQ2	2	495.98	1264.8
<none>		493.81	1266.6
+ PROMEDIO_GOLES_RECIBIDOS_EQ1	1	493.72	1268.5
+ RANKING_UEFA_EQ2	1	493.74	1268.5
+ PROMEDIO_GOLES_MARCADOS_EQ2	1	493.80	1268.6
+ PROMEDIO_GOLES_RECIBIDOS_EQ2	1	493.80	1268.6
- PROMEDIO_GOLES_MARCADOS_EQ1	1	498.76	1269.5
+ NIVEL_LIGA_EQUIPO_EQ1	2	493.16	1269.9
- VALOR_EQUIPO_EQ2	1	499.25	1270.0
- VALOR_EQUIPO_EQ1	1	501.01	1271.8
- RANKING_UEFA_EQ1	1	502.09	1272.9

```
Step: AIC=1264.76
GOLES_EQ1 ~ RANKING_UEFA_EQ1 + VALOR_EQUIPO_EQ1 + PROMEDIO_GOLES_MARCADOS_EQ1 +
  VALOR_EQUIPO_EQ2
```

	Df	Deviance	AIC
<none>		495.98	1264.8
+ NIVEL_LIGA_EQUIPO_EQ2	2	493.81	1266.6
+ PROMEDIO_GOLES_MARCADOS_EQ2	1	495.84	1266.6
+ PROMEDIO_GOLES_RECIBIDOS_EQ2	1	495.84	1266.6
+ PROMEDIO_GOLES_RECIBIDOS_EQ1	1	495.86	1266.6
+ RANKING_UEFA_EQ2	1	495.88	1266.7
- PROMEDIO_GOLES_MARCADOS_EQ1	1	500.83	1267.6
+ NIVEL_LIGA_EQUIPO_EQ1	2	495.11	1267.9
- RANKING_UEFA_EQ1	1	503.61	1270.4
- VALOR_EQUIPO_EQ1	1	504.22	1271.0
- VALOR_EQUIPO_EQ2	1	530.27	1297.0

```
summary(modeloAICGLPU12)
coeficientesLocAICUNI1<-coefficients(summary(modeloAICGLPU12))

sum((residuals(modeloAICGLPU12, type="deviance"))^2)/modeloAICGLPU12$df.residual

DATOSBUENO1PARTIDOUNICO11<-within(DATOSBUENO1PARTIDOUNICO11,{
  fitted.modeloAICGLPU1<-fitted(modeloAICGLPU12)
  residuals.modeloAICGLPU1<-residuals(modeloAICGLPU12,type="deviance")
})

scatterplot(residuals.modeloAICGLPU1~fitted.modeloAICGLPU1, regLine=TRUE, smooth=list(span=0.5, spread=FALSE),
  boxplots=FALSE, data=DATOSBUENO1PARTIDOUNICO11,id=list(method="mahal",n=4))
library(AER)
dispersiontest(modeloAICGV12)

attach(EQUIPOS_2020_2021_ORDENADOS_SIMULACION)

landaLOC<-function(j,i){
  return(exp( coeficientesLocAICUNI1[1]+ coeficientesLocAICUNI1[2]*RANKING_UEFA[i]+coeficientesLocAICUNI1[3]*VALOR_EQUIPO[i] +
  coeficientesLocAICUNI1[4]*PROMEDIO_GOLES_MARCADOS[i]+ coeficientesLocAICUNI1[5]*VALOR_EQUIPO[j]))}

landaVIS<-function(j,i){
  return(exp( coeficientesLocAICUNI1[1]+ coeficientesLocAICUNI1[2]*RANKING_UEFA[j]+coeficientesLocAICUNI1[3]*VALOR_EQUIPO[j] +
  coeficientesLocAICUNI1[4]*PROMEDIO_GOLES_MARCADOS[j]+ coeficientesLocAICUNI1[5]*VALOR_EQUIPO[i]))}

N<-32
A<-matrix(N*N,N,N)
i<-1
j<-1
while(i<=N){
  A[i,i]<-0
  i<-i+1
}

i<-1
j<-i+1
while(i<=N-1){
  while(j<=N){
    A[i,j]<-landaLOC(j,i)
    j<-j+1
  }
  i<-i+1
  j<-i+1
}

j<-1
i<-j+1
```

```
while(j<=N-1){  
  while(i<=N){  
    A[i,j]<-landaVIS(i,j)  
    i<-i+1  
  }  
  j<-j+1  
  i<-j+1  
}
```

Bibliografía

- [1] CHATER, M., ARRONDEL, L., GAYANT, J.P. Y LASLIER J.F., (2021). *Fixing match-fixing: Optimal schedules to promote competitiveness*, European Journal of Operational Research, 26 (2), 439-457.
- [2] DOBSON, A.J. & BARNETT, A.G., (2008). *An Introduction to Generalized Linear Models*, CRC Press
- [3] GUYON, J., (2020). *Risk of collusion: Will groups of 3 ruin the FIFA World Cup?*, Journal of Sports Analytics, 6 (4), 259-279.
- [4] GUYON, J., (2015). *Rethinking the FIFA World Cup final draw*, Journal of Quantitative Analysis in Sports, 11 (3), 169-182.
- [5] ITURRALDE, A., (2017). *Predicción de eventos deportivos*, <https://zaguan.unizar.es/record/64244?ln=es>
- [6] LALIENA, P. Y LÓPEZ, F. J., (2019). *Fair draws for group rounds in sport tournaments*, International Transactions in Operational Research, 26 (2), 439-457.
- [7] LAWRENCE, I., Y LIN, K., (1989). *A concordance correlation coefficient to evaluate reproducibility*, Biometrics, 45 (1), 255-268
- [8] LÓPEZ, E. Y RUIZ, M., (2011). *Análisis de datos con el Modelo Lineal Generalizado. Una aplicación con R*, Revista española de pedagogía, 248, 59-80.
- [9] MAHER, M. J., (1982). *Modelling association football scores*, Statistica Neerlandica, 36(3), 109-118.
- [10] PAREJA, C., OJEDA, M., ANDEYRO, Á. Y ROSSI, C., (1997). *Algoritmos y Programación en Pascal*, <http://antares.sip.ucm.es/cpareja/libroAlgoritmos/docs/libro-completo.pdf>
- [11] ROBACK, P. & LEGLER, J., (2021). *Beyond Multiple Linear Regression: Applied Generalized Linear Models and Multilevel Models in R*, CRC Press
- [12] SABROSO, S., (2018). *Predicción de resultados de partidos de fútbol*, <https://zaguan.unizar.es/record/77769?ln=es>
- [13] VALERA, F., (2013). *Sistema de predicción de resultados en eventos deportivos y su aplicación en las apuestas*, <https://e-archivo.uc3m.es/bitstream/handle/10016/17882/PFC%20-%20Fernando%20Valera%20Guardiola.pdf?sequence=1&isAllowed=y>