TRABAJO DE FIN DE GRADO

Grado en Estadística

Título: Predicción del rendimiento futuro de los jugadores del circuito masculino de tenis

Autor: Luis Nuevo Mengual

Director: Luis Ortiz Gracia

Departamento: Econometria, Estadística i Economia Aplicada

Convocatoria: Junio 2022

UNIVERSITAT DE BARCELONA

UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH
Facultat de Matemàtiques i Estadística

A mi familia y amigos por apoyarme en todo lo que hago y confiar siempre en mí. A los que están y a los que se fueron.

Gracias Luis por darme la oportunidad de hacer realidad este trabajo y por toda la ayuda.

RESÚMEN

Este trabajo consiste en la creación de una base de datos con las variables de interés para su posterior modelización. Para hacerlo se hará uso de un modelo de regresión logística. El objetivo de este modelo es obtener una probabilidad de victoria entre dos jugadores para un partido de tenis.

A partir de este modelo se realizará una simulación de los torneos disputados en 2021 para a posteriori predecir el rendimiento futuro de los diferentes jugadores comparándolos con los resultados reales.

El objetivo es predecir y cuantificar la capacidad de predicción de los diferentes jugadores que componen el circuito masculino de tenis.

Palabras clave: Deporte, tenis, predicción, modelos lineales generalizados, regresión logística, probabilidad, simulación.

Clasifiación AMS:

- 62J12 Modelos lineales generalizados
- 62M20 Predicción

ABSTRACT

This report consists on creating a data base with the variables of interest to create a predicting model. To do it so, a logistic regression model will be used. The goal of tis model is to obtain a win probability between two players in a tennis match.

With the model a simulation will be made for all the tournaments played in 2021 to predict the future performance of the different players comparing to these results to the real ones.

The main goal is to predict and quantify the predictive capacity of the model for the different type of players that play at the man's tennis circuit.

Keywors: Sport, tennis, prediction, generalized linear models, logistic regression, probability, simulation.

AMS Classification:

- 62J12 Generalized linear models
- 62M20 Prediction

ÍNDICE

INTRODUCCIÓN

Durante el transcurso del tercer año de carrera se nos enseñaron diferentes modelos de predicción. Soy una persona amante y apasionada del deporte, por lo que pensé en aplicar por mi cuenta lo aprendido en este mundo, la duda estaba en cuál de ellos escoger.

En España hay tres deportes dominantes. El fútbol, el baloncesto y el tenis. Este último debe su alta popularidad en las nuevas generaciones como la mía a Rafael Nadal Parera, para muchos el mejor jugador de la historia del tenis y de los mejores deportistas españoles que se han visto. No es solo elevado su rendimiento, sino que su forma de ser y los valores que infunde tanto en espectadores como en tenistas es algo para tener en cuenta. Es por esto por lo que, hoy por hoy, soy seguidor de este deporte.

A todo esto, se le suma el hecho de que en internet hay una gran fuente de datos de los diferentes partidos disputados de manera abierta para todo el mundo. Es por estos motivos que empecé a trabajar en un modelo de predicción para partidos de tenis. Para hacerlo se tomaban en cuenta diferentes variables, las cuales dependían de cada jugador. Estas ponían contexto al partido y creaban dos perfiles únicos que se enfrentaban en un momento determinado en el tiempo para obtener una probabilidad de victoria para cada uno de ellos.

A partir de esta idea y el trabajo previo realizado decidí hacer este trabajo de fin de grado. La primera intención era predecir el resultado de un partido mediante probabilidades. En este caso quería ir más allá, predecir el rendimiento futuro de un jugador para el próximo año.

Con el objetivo de resolver este problema era necesario generar de 0 un nuevo modelo que mejorase el anterior con todo lo aprendido previamente. Para ello se generará una base de datos a partir de la información de los diferentes partidos, creando variables que puedan ser significativas para el modelo. El modelo al definir la probabilidad entre 0 y 1 de cada jugador será un modelo de respuesta binaria que, basándonos en pruebas previamente efectuadas, será el logístico, ya que fue el que mejor funcionaba para este caso.

Una vez procedido con todo eso, se hará una simulación del 2021 de la temporada de tenis, puesto que es el último año completo de datos que se tiene. Una vez terminada esta simulación se comparará los resultados observados en la realidad junto a los esperados obtenidos en la simulación para poder obtener una predicción individual de algunos jugadores en concreto.

CONCEPTOS PREVIOS

Este trabajo que trata sobre la predicción se centra en el deporte de raqueta llamado tenis. Para poder proceder durante el trabajo con un total entendimiento del proceso es necesario contextualizar este deporte, así mismo como algunos conceptos relacionados con este.

Una de las cosas necesarias es como se define el ganador. Ambos jugadores golpean la pelota con una raqueta dentro del límite del campo. En caso de que un jugador no devuelva la pelota dentro del campo rival, pierde el punto. Cada un número determinado de puntos se gana un juego. El primero en alcanzar los 6 juegos gana el set. Para ganarlo, no obstante, se debe ganar por una diferencia de dos. Es por ello por lo que en caso de empate a cinco juegos se disputa hasta 7. En caso de llegar a empate en seis juegos se disputa un último llamado tie-break. El primero en llegar a 7 juegos con una diferencia de dos, ganar el set. El ganador es el primero en llegar a 2 sets ganados en un mejor de 3 o a 3 sets ganados en un mejor de 5.

Además, el tenis se juega en diferentes superficies. Estas superficies pueden ser arcilla, hierba o cemento (también conocido como pista dura). Esto afecta en el juego, ya que el bote de la pelota es diferente dependiendo de en cuál se dispute el partido.

Este deporte se puede disputar tanto de forma individual como en la modalidad de dobles. En este trabajo nos centramos en partidos individuales disputados en el circuito ATP.

ATP son las siglas de Asociación de Tenistas Profesionales. Es la entidad que organiza los circuitos masculinos de tenis y que además elabora el *ranking* ATP. Estos torneos que organizan se dividen en tres categorías, ATP Tour, ATP *Challenger Tour* y ATP

Champions Tour. El *ranking* ATP, como comentábamos antes, es un orden que indica los jugadores que más puntos tienen al final del año. Estos puntos se obtienen en los diferentes torneos que se disputan a lo largo de la temporada, la cual es de un año y comienza en enero. No obstante, no todos los torneos a disputar tienen el mismo peso en el *ranking*. También participa en este circuito la federación internacional de tenis. Así pues, todos los torneos que hay son:

Categoría del evento	**Dinero en premios ($)**	**Puntos para el *Ranking***
Grand Slam	Por determinar según torneo	2000
ATP World Tour Finals	7 500 000	1100-1500
ATP Tour Masters 1000	de 3 748 925 a 5 452 985	1000
ATP Tour 500	de 1 333 085 a 2 249 215	500
ATP Tour 250	de 404 780 a 1 189 605	250
ATP Challenger Series	de 35 000+H a 125 000+H	80 a 125
ITF World Tennis Tour	de 15 000 a 25 000+H	10 a 20
Copa Davis	1 750 000	-

Nosotros nos centraremos en las cuatro primeras categorías, ya que son los que reparten más puntos. También, dada la magnitud de torneos tanto para los *Challenger Series* e ITF se ha prescindido de ellos. Otro motivo es que los puntos obtenidos de los jugadores con un *ranking* más alto provienen el 100% de los torneos que ya tenemos en cuenta y el objetivo es predecir el rendimiento de estos jugadores.

METODOLOGIA

En este capítulo se expondrá la metodología del estudio. El objetivo es mostrar el proceso realizado para alcanzar los resultados y conclusiones que veremos más adelante. La metodología del trabajo se divide en tres grandes apartados. Base de datos, donde veremos los datos usados, origen y estructura, junto a las variables utilizadas analizadas. Modelización, donde podremos ver qué tipo de modelo es el escogido como óptimo para la predicción, selección del mejor modelo y su validación. Por último, tendremos la simulación donde comentaremos cuál es el objetivo, planteamiento y estructura de esta.

BASE DE DATOS

Dado que el objetivo de este trabajo es la predicción del ganador de un partido determinado para poder realizar una simulación de una temporada entera del circuito ATP en torneos 250 o superior, era necesario un registro de los partidos previamente jugados. Es por ello que hice una búsqueda para obtener estos datos y encontré el repositorio en GitHub de Jeff Sackmann. Allí encontraremos todos los partidos disputados para todo tipo de torneo por año en el circuito masculino de tenis.

Los datos se reparten en diferentes csv, uno por cada año natural (temporada). También vemos como cada fila corresponde a un partido diferente. A continuación, vemos un par de registros para poder visualizar mejor los datos.

tourney_name	surface	draw_size	tourney_level	tourney_date	match_num	winner_id	winner_seed	winner_entry	winner_name
Atp Cup	Hard	16	A	20220103	300	200000			Felix Auger Aliassime
Atp Cup	Hard	16	A	20220103	299	133430			Denis Shapovalov
Atp Cup	Hard	16	A	20220103	298	105138			Roberto Bautista Agut
Atp Cup	Hard	16	A	20220103	297	105807			Pablo Carreno Busta

Podemos ver como son varios los atributos, 49 exactamente, asociados a esta base de datos, no obstante, se han eliminado los que no aportaban información necesaria. Estas son las variables utilizadas.

• Surface → En que superficie se ha jugado el partido. Puede ser arcilla, dura (cemento), hierba o moqueta.

• Tourney_level → Tipo de torneo en el que se juega. Puede ser Grand Slam, torneos de repartimiento de 1000, 500 y 250 puntos, Challenger, ITF, Master Finals y por último, Copa Davis.

• winner_name/loser_name → Nombre del jugador que ha ganado o perdido el partido

• winner_age/loser_age → Edad del jugador que ha ganado o perdido el partido

• winner_height/ loser_height → Altura del jugador que ha ganado o perdido el partido

• tourney_date → Fecha en la que se disputa el torneo

Además de esto, no se usarán todos los resultados. Se han eliminado los partidos donde el resultado sea un W/O (jugador se retira antes de jugar el partido), retiradas en medio del partido, partidos inacabados y otros resultados erróneos como datos faltantes. Es importante comentarlo, ya que en todas las estadísticas y en la base de datos de Jeff se contabilizan como victorias o derrotas estas situaciones.

Esto se hace porque no considero que un partido que ganas donde el jugador rival se ha lesionado o porque directamente no ha jugado no se le puede considerar una victoria y afectaría negativamente a la veracidad de los datos. También se eliminarán partidos con jugadores con un número muy reducido de partidos jugados para evitar *outliers*. Por ejemplo, un jugador que ha ganado sus dos primeros partidos tendrá un 100% de victoria, pero como imaginamos no es un dato significativo.

Una vez filtrados los datos como previamente comentado, estas observaciones se utilizan para calcular las métricas necesarias a la hora de modelizarlos. Los modelos tanto de regresión como de Machine Learning necesitan unos parámetros que expliquen el comportamiento de los datos. En este apartado explicamos dichos parámetros. Una vez hecha una investigación sobre que puede influir significativamente, se han planteado las siguientes variables.

Porcentaje de victoria

Es un hecho que cada jugador presenta diferentes rendimientos dependiendo del torneo en el que jueguen. Por ejemplo, Pablo Andújar, deportista español exnúmero 32 en el circuito, presenta un porcentaje de victorias muy bajo en dura y hierba, mientras que en arcilla rinde a un nivel muy alto. Con esto en mente se proponen como variables los porcentajes de victorias según:

- Tipo de superficie

- Tipo de torneo

- Tipo de superficie por tipo torneo (las dos anteriores cruzadas)

No obstante, muchos jugadores tienen diferentes etapas dónde pueden tener diferente rendimiento. Podría ser diferente el rendimiento de un jugador en sus dos últimos años comparado con los 5 previos o toda su carrera. Para solucionarlo se planteará de cada una de las tres variables previas tres alternativas para observar cuál es más importante. Se utilizarán los porcentajes de victorias en los últimos 2 años, 4 años y 6 años. Estos números se han obtenido a partir de los años de en promedio que los jugadores ejercen de profesionales, la cual es de 12. A partir de ahí se han seleccionado esos tres intervalos para cuantificar la importancia de cada uno. No se han cogido de más de 6 años, ya que a priori no creo que sea más significativo que el resto y la base de datos sería mucho más grande, lo que haría imposible ejecutarla para calcular todas las métricas. Por estos motivos, el rango máximo determinado de tiempo es de 6 años.

Edad

La edad puede ser un factor importante a la hora de predecir quién ganará el partido. Por una parte, los jugadores más jóvenes tienen mejores capacidades físicas y mejor resistencia, aunque jugadores mayores están más experimentados y asentados en el circuito. Si afectará o no y de qué forma en el modelo es algo que considero relevante.

Altura

En los últimos años, se está viendo un cambio en el perfil de los jugadores. Cada vez son más altos y producen la mayoría del peligro con sus servicios. Fabio Fognini, jugador profesional del circuito ATP desde el 2004, además de ser el italiano con más victorias de la historia, lo comentaba en una rueda de prensa este año en el torneo de Buenos Aires. Leyendo tales declaraciones nos hace pensar que la altura será un factor importante, por lo que se propondrá como variable también.

No obstante, la altura no necesariamente debe ser siempre un factor crucial. La altura lo que hace es ayudar a que el servicio del jugador sea más potente al golpear la bola desde una altura más elevada. Es común que los jugadores más altos sean los mejores sacadores del circuito. Esto se puede ver recompensado en pistas rápidas, pero tal vez no tanto en las lentas. Es por ello por lo que se usará la interacción junto a la superficie para ver el efecto que tiene según el tipo de pista.

Para el caso de esta variable me encontré con el problema de que había una gran cantidad de jugadores, los cuales no constaba su altura en la base de datos. Dado que es una variable necesaria y sería una gran pérdida de datos, se ha hecho lo siguiente. Mediante una consulta de SQL agrupada por nombre para poder obtener el número de partidos en los que participan, se ha puesto la altura de los jugadores con mucha presencia en los datos. Es decir, buscar y agregar de forma manual la altura, por ejemplo, Pedro Martínez, jugador español, los últimos años ha aparecido mucho en los cuadros principales, pero al ser un jugador “nuevo” en este nivel de torneo no tiene la edad. Como la cantidad de jugadores que hay es muy grande y sería poco convencional terminar de solucionar todos los NA, se ha puesto para el resto de los jugadores la media de las alturas de toda la muestra.

Head to Head

El *Head to head* (cara a cara) son los resultados previos de un enfrentamiento directo entre dos jugadores, es decir, entre todos los partidos disputados entre ambos jugadores, la cantidad de victorias de uno y de otro.

Una cosa muy importante a tener en cuenta es que un jugador puede tener más facilidad para ganar a otro por su forma de jugar. Esto queda plasmado en él cara a cara, que se propondrá como un porcentaje del total de victorias entre ambos jugadores, siendo 0 para ambos y sin tener ningún impacto en el modelo en el caso de que sea la primera vez que se enfrentan. Como comentábamos antes, cada jugador pasa por diferentes etapas de su juego y rinde diferente en otras superficies. Con esto en mente se sugieren los resultados del cara a cara con unos intervalos parecidos a los previamente vistos; 6,

3 y 1 años, teniendo en cuenta la superficie donde se juega y sin tenerla en cuenta. Con esto cubrimos la posible relevancia de que un jugador pueda tener facilidades para ganar a otro solo si se dan ciertas características en la pista.

Estado de forma

En el deporte el rendimiento de los jugadores va a rachas, unos meses juegas al máximo nivel y otros no eres capaz de jugar bien. Se plasma esta idea con la siguiente formula.

$$\log(1 - \%\ \text{victoria x años} + \%\ \text{victoria últimos z meses})$$

Se abarcan las opciones del estado de forma para el último año, últimos 6 meses, últimos 3 y último mes. Además, con motivo de lo comentado en el sub apartado de Porcentaje de victorias, se usarán también el porcentaje de victorias de los últimos 3 o 6 años en vez de históricamente. Esto crea 12 posibilidades

{Jugador 1, Jugador 2} x {% victoria z tiempo} x {% victoria últimos x meses}

Donde z = {6 años, 4 años, 2 años} y x = {12, 6, 3, 1} meses

Esta forma de medir el estado de forma reciente de un jugador es una fórmula utilizada en el artículo de Nicholas Devin donde se busca cuantificar el rendimiento en un tiempo específico comparando con el resto de su carrera penalizando en caso de obtener un rendimiento inferior.

Una vez expuestas las variables que se emplearán procedemos a elaborar la muestra que se usará para la modelización.

Para dichos datos a utilizar en los modelos se ha hecho una limpieza. Se han eliminado los datos faltantes en todas las variables, 5000 registros han sido eliminados. Esto ha eliminado prioritariamente todos los datos de jugadores que no tenían registros previos en el contexto que se analizaba. Un ejemplo sería un jugador que juega por primera vez en una pista de arcilla. Al no tener registros previos se genera un NA.

También se ha filtrado para que los registros de ambos jugadores para un período de tiempo en específico superen los 50 partidos y los 10 en las condiciones de pista y tipo de torneo de ese partido en específico. Con esto hacemos más verídico la modelización y prevenimos la aparición de *outliers*, ya que un jugador podría presentar un 100% de victoria cuando solo ha jugado dos partidos y eso a la hora de crear el modelo dará problemas de ajuste.

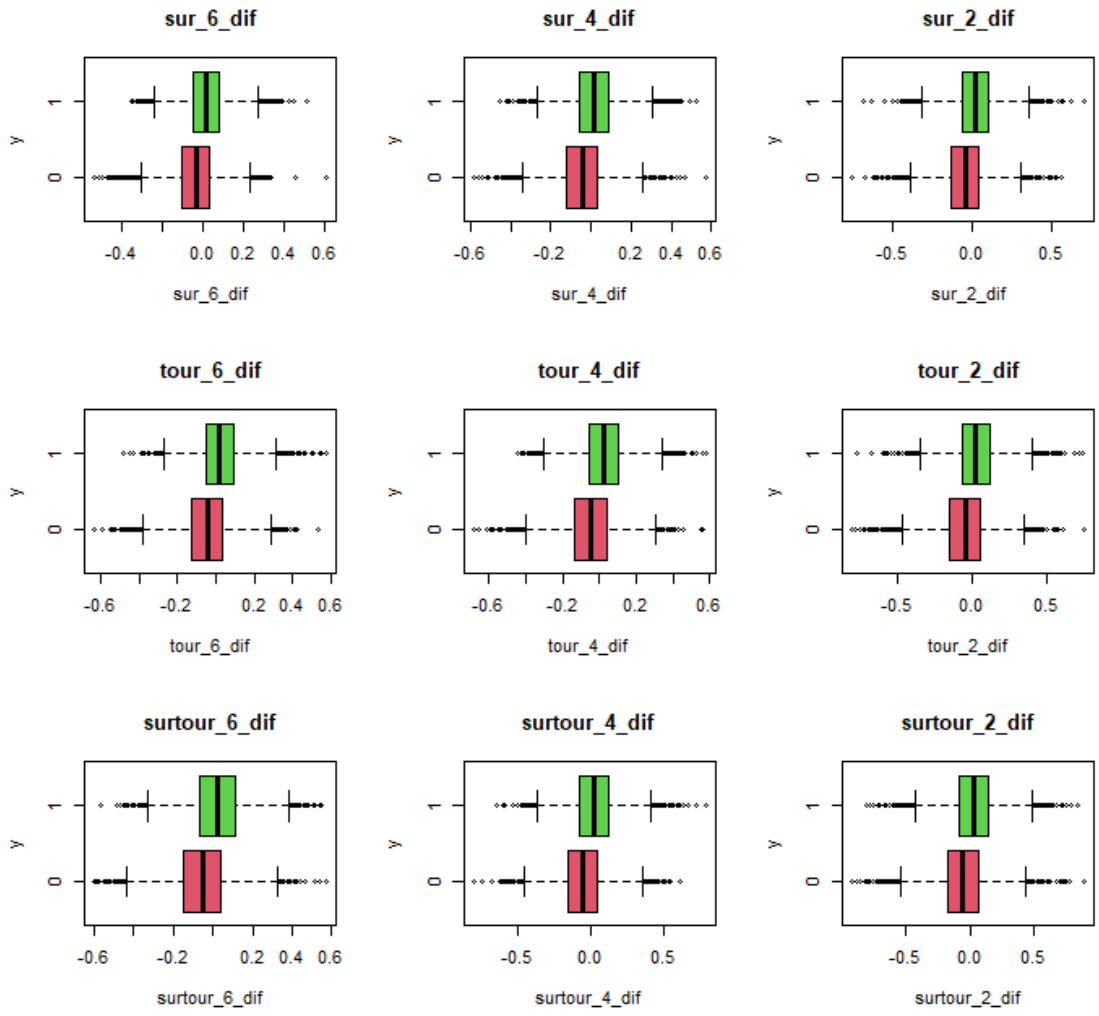
Antes de poder pasar a los modelos haremos un pequeño análisis bivariante. ¿Pero, por qué no univariante? El hecho de que haya una cantidad tan elevada de variables a analizar me hace querer descartar un análisis tan extenso. A esto hay que sumarle que muchas de ellas ya sabemos su comportamiento, por lo que no nos proporcionaría ningún tipo de información extra, por ejemplo, el *head to head* siempre serán valores entre 0 y 1 y sin ningún tipo de relación entre sí es el jugador 1 o 2, ya que estos están colocados así de forma aleatoria.

Dicho esto, lo interesante es ver el comportamiento de las variables en comparación con la variable respuesta, la cual es la siguiente.

$$y = \begin{cases} 1 \; si \; el \; ganador \; es \; el \; jugador \; 1 \\ 0 \; si \; el \; ganador \; es \; el \; jugador \; 2 \end{cases}$$

Antes de comentar los resultados, es importante explicar que la variable que se usará no será el estadístico en sí por jugador si no su diferencia. Por ejemplo, en un enfrentamiento entre un Jugador 1 con 20 años y un Jugador 2 con 26, no se emplearán el 20 y el 26 como variable explicativa sino su diferencia que en este caso sería de 6. Dicho esto, podemos proceder al análisis.

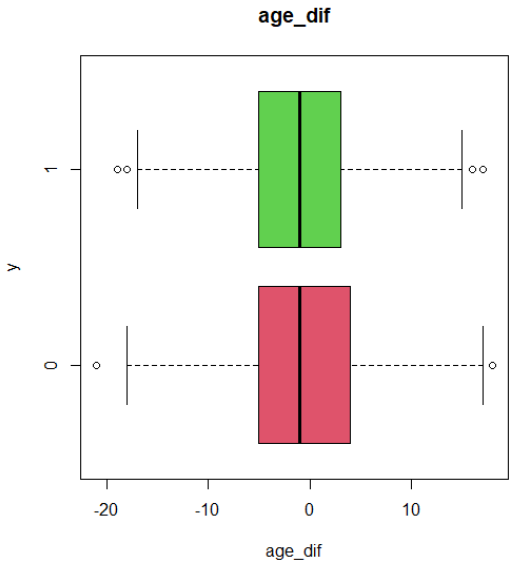
Porcentaje de victoria

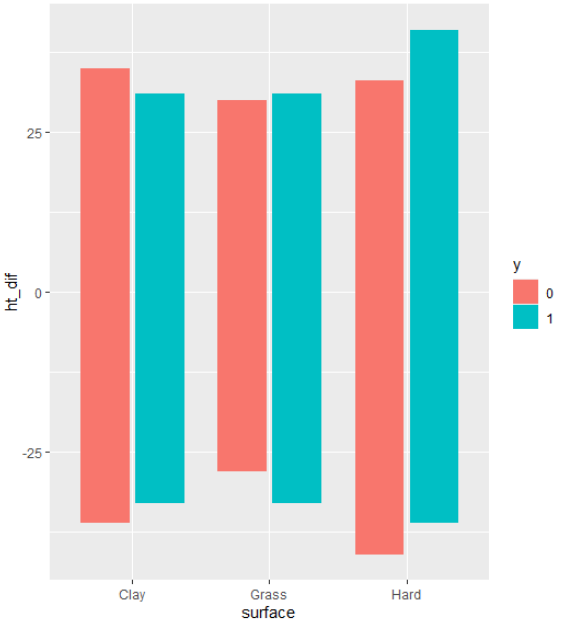


Primero nos encontramos con las variables de porcentaje de victorias tanto para tipo de pista, por tipo de torneo y ambas. Podemos observar como en todas ellas el resultado favorece al primer jugador al tener una diferencia positiva (jugador 1 tiene más

porcentaje de victorias) y al segundo al tenerla negativa. Además, las cajas son parecidas en todos los casos, por lo que la variabilidad es muy parecida para ambas variables. Esto nos sugiere que, a priori, estas variables podrían ser significativas en sus respectivos modelos.

Edad y altura

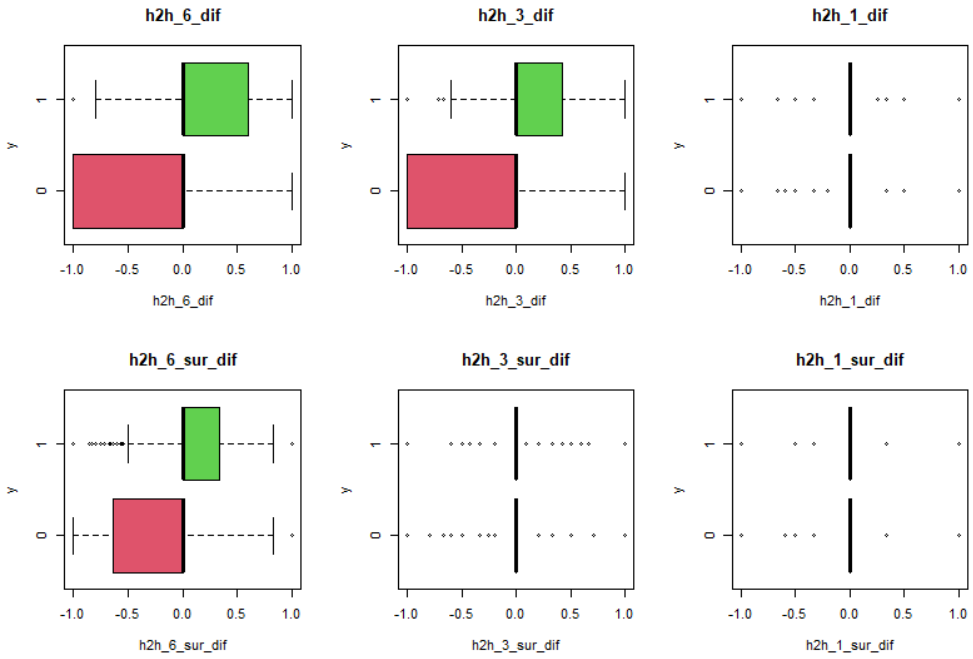




Para la variable edad no parece haber una tendencia para ninguno de los casos, ya que ambos encuentran su media muy cercana al 0.

En cuanto a la altura vemos dos casos distintos. El primero de ellos se sitúa en la altura para el tipo de pista “Dura”. Podemos ver como la barra de la victoria del primer jugador, es decir, cuando y=1, tiende más hacia diferencias positivas, mientras que para el jugador 2 totalmente lo contrario. Esto nos puede hacer pensar que a priori sí que afecta la altura en caso de disputar el partido en ese tipo de pista. Para las otras dos superficies no se pueden ver diferencias notables, por lo que a priori supondríamos que no es significativa su interacción.

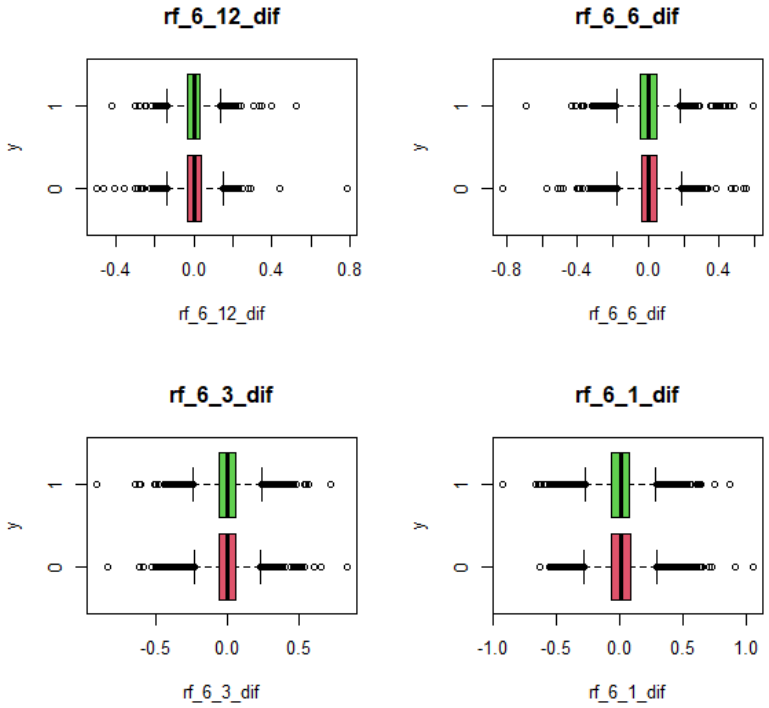
Head to Head

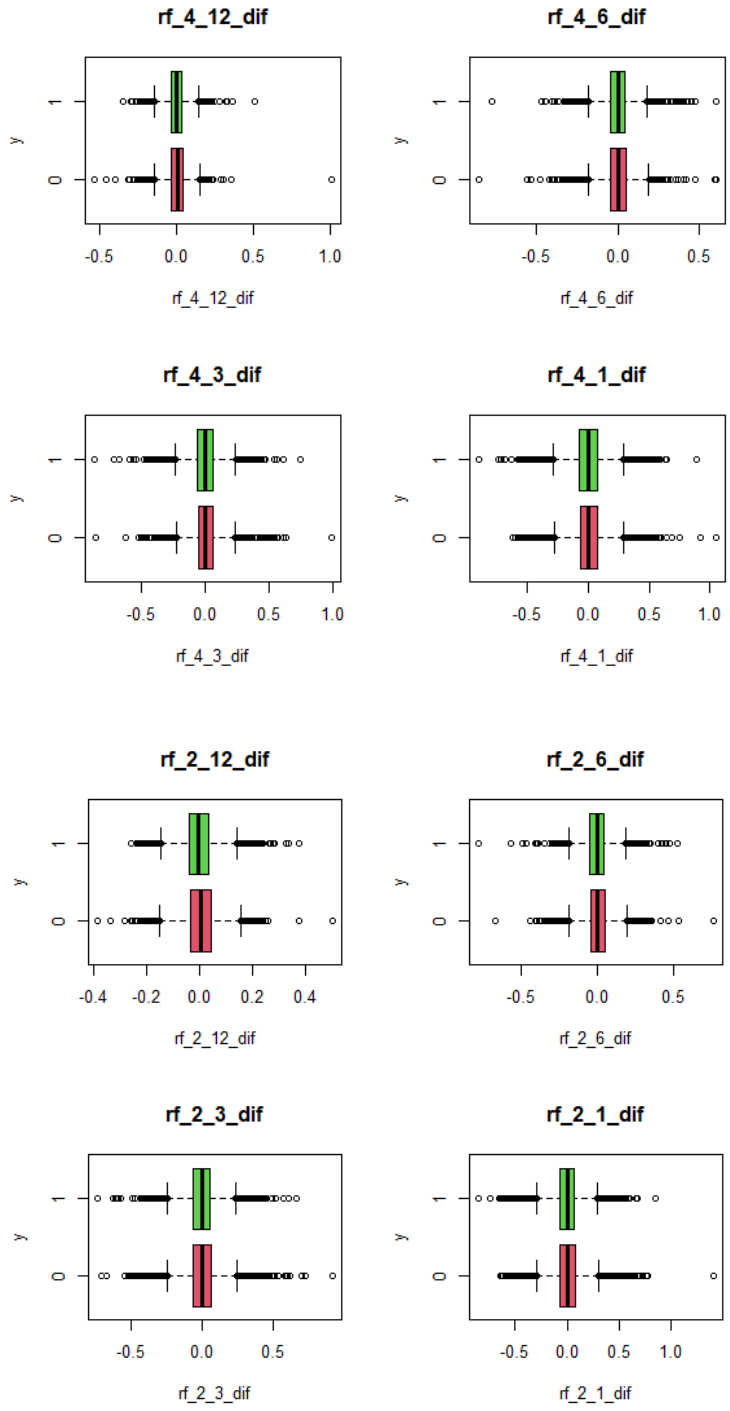


Para él cara a cara entre jugadores vemos dos tónicas totalmente distintas.

Para el h2h (head to head) teniendo en cuenta los últimos 6 años y para 3 años sin tener en cuenta el tipo de pista, vemos la misma tónica que para los porcentajes de victorias solo que más acentuada a diferencia del resto que no nos indican ningún tipo de significación de la variable en el modelo.

Estado de forma





Para finalizar encontramos las formas recientes. Aquí vemos que la tónica es la misma para todos. No obstante ser la misma, esta no indica grandes indicios de que haya significación ni de su inexistencia.

MODELIZACIÓN

Para poder realizar la simulación de la temporada ATP es necesario crear un modelo para poder predecir el resultado de un partido entre dos jugadores determinados. Se quiere predecir la probabilidad de que gane un jugador u otro y a partir de esa probabilidad seleccionar aleatoriamente el ganador.

Como previamente hemos comentado nuestra variable respuesta (y = qué jugador ha ganado) es una variable binaria. Con todo esto en mente se plantea usar un modelo de regresión logística. Este tipo de modelos es utilizado para predecir el resultado de una variable categórica en función de las variables independientes o predictores. Este modelo es útil para modelar la probabilidad de un evento ocurriendo en función de otros factores, lo cual es perfecto para la situación que es propuesta con la predicción de un partido de tenis. El modelo en sí quedaría de la siguiente forma.

$$logit(p_i) = \ln\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \beta_1 x_{1,i} + \cdots + \beta_k x_{k,i}$$

Por lo que a la hora de obtener la probabilidad de victoria para un conjunto de datos x aplicaríamos la siguiente formula.

$$p(x) = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} = \frac{e^{\beta_0+\beta_1 x}}{1+e^{\beta_0+\beta_1 x}}$$

Dado que hemos calculado tres tipos de porcentaje de victorias (por pista, por torneo y ambas cruzadas) se plantean dos modelos diferentes. El primero de ellos se tiene en cuenta los porcentajes de victorias tanto por tipo de pista como por nivel de torneo por separado para que calcule el peso que tiene cada una en el modelo. Por otra parte, el segundo planteamiento se descarta estas dos variables y se usa solo la variable que cruza ambas. Con esto queremos ver si es mejor para el modelo calcula que la importancia que tienen en las predicciones estos porcentajes de victorias son diferentes o si por contraparte es óptimo cruzarlas y dar un único coeficiente para la estimación de su parámetro.

También es importante comentar que estos modelos no tendrán intecept, es decir, será 0. Esto se debe a que en el caso de tener dos jugadores con exactamente las mismas características, la probabilidad de victoria tanto para uno como para otro debe ser de 0'5. Fijando nuestro β_0 en 0 hace que en encontrarnos con dicho caso la probabilidad de victoria quedaría como comentábamos, ya que sería $\frac{1}{1+e^0} = \frac{1}{1+1} = 0.5$. Esto ocurre

porque, como ya se ha comentado previamente, las variables son la diferencia de estadísticos, por lo que nos encontramos que en este caso todas las x_k serian 0. Por lo tanto, nuestras funciones lineales del modelo quedarían.

Modelo Superficie + Nivel Torneo

$$\beta_1 TourDif + \beta_2 SurfaceDif + \beta_3 AgeDif \ldots$$

Modelo variables cruzadas

$$\beta_1 SurfaceTourDif + \beta_2 AgeDif \ldots$$

No obstante, cada modelo contiene algunas variables las cuales tienen un período de tiempo de los últimos 6, 4 y 2 años, por lo que en total obtendremos 6 modelos distintos, pues el objetivo también es ver en qué intervalo de tiempo hemos de fijarnos para que el modelo sea el mejor posible.

Además, en cada modelo se añadirán las variables con su período de tiempo de referencia correspondiente para la variable de estado de forma. Todos los modelos tienen las variables de diferencia de edad, altura y los cara a cara, tanto teniendo en cuenta el tipo de superficie como si no. Con todo esto dicho, un ejemplo sería el modelo a 6 años de variables cruzadas que quedaría de la siguiente manera.

$$logit(p_i) =$$

$$\begin{aligned}\beta_1 SurfaceTourDif_6 + \beta_2 AgeDif + \beta_3 HeightDif{:}Surface + \beta_4 h2hDif_6 \\ + \beta_5 h2hDif_3 + \beta_6 h2hDif_1 + \beta_7 h2hSurDif_6 + \beta_8 h2hSurDif_3 \\ + \beta_9 h2hSurDif_1 + \beta_{10} rf_{(6,12)} + \beta_{11} rf_{(6,6)} + \beta_{12} rf_{(6,3)} + \beta_{13} rf_{(6,1)}\end{aligned}$$

Encontramos que por cada modelo de superficie + nivel de torneo tendremos 14 parámetros a estimar y 13 para variables cruzadas. Con tal de tener el mejor modelo posible, aplicamos la función *step* en R la cual calcula la mejor combinación de variables para explicar la variable a explicar. Una vez aplicado en cada modelo debemos seleccionar el mejor modelo de los 6. Para ello se usarán dos métodos de selección de modelo, el *Akaike Information Criterion* (AIC) y el *Bayesian Information Criterium* (BIC). Sus fórmulas son las siguientes.

$$AIC = 2k - 2ln(\hat{L})$$

$$BIC = kln(k) - 2ln(\hat{L})$$

Siendo k el número de parámetros estimados por el modelo y $\hat{L}$ el valor de la máxima verosimilitud de la función del modelo. Todo esto es calculado por las funciones *AIC( )* y *BIC( )* de R

Los valores para ambos estadísticos de todos los modelos son los siguientes

	AIC	BIC
m6a	9865.338	9941.877
m6b	10025.900	10088.522
m4a	9885.616	9955.196
m4b	8947.569	9009.106
m2a	11345.622	11402.279
m2b	11497.124	11567.946

Como podemos observar, tanto para el AIC como para el BIC, el mejor modelo es el 4b el cual corresponde al que tiene las variables superficie y tipo de torneo cruzadas y con teniendo en cuenta un intervalo de tiempo de 4 años. El modelo final seleccionado con la función *step* aplicada es:

$$logit(p_i) = \beta_1 SurfaceTourDif_4 + \beta_2 AgeDif + \beta_3 HeightDif{:}Surface + \beta_4 h2hDif_6 + \beta_5 h2hDif_3 + \beta_6 rf_{(4,12)} + \beta_7 rf_{(4,3)}$$

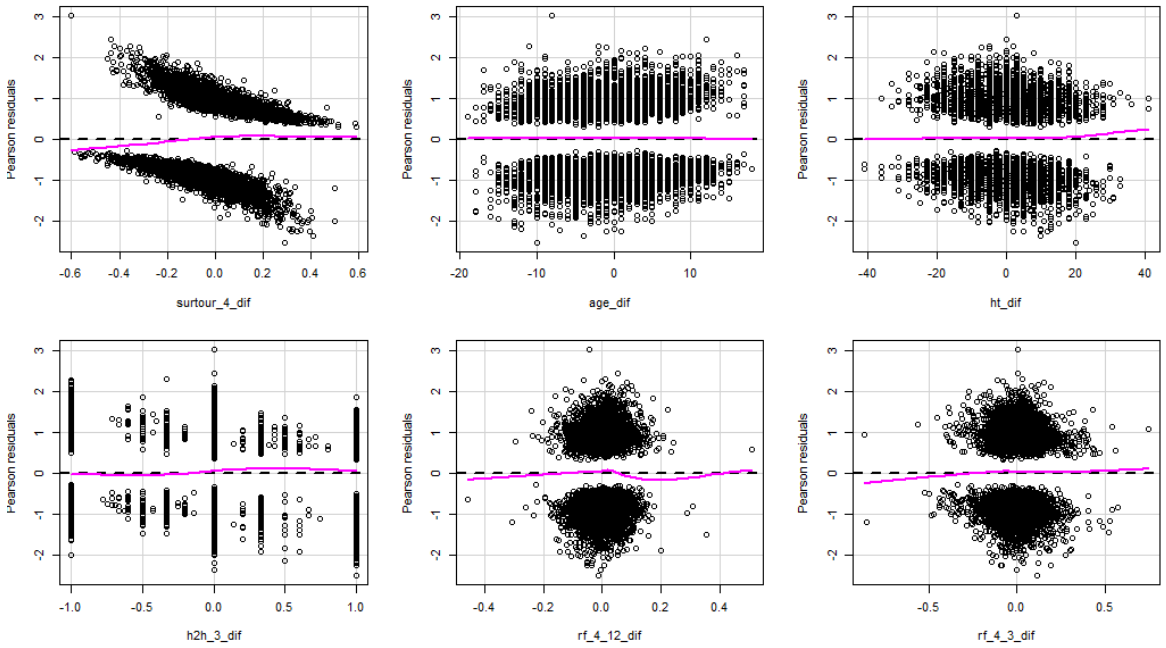
Finalmente, nos quedamos con la variable que identifica el modelo junto a las diferencias de edad, interacción, altura y superficie, cara a cara de los últimos 6 y 3 años y estado de forma del último año y de los últimos 3 meses, con los resultados de los últimos 4 años como referencia.

A continuación, encontramos los valores β finalmente estimados y su interpretación. De la interacción Height:Surface solo encontramos un valor porque es el único significativo, es decir, los otros dos estadísticamente no son diferentes de 0.

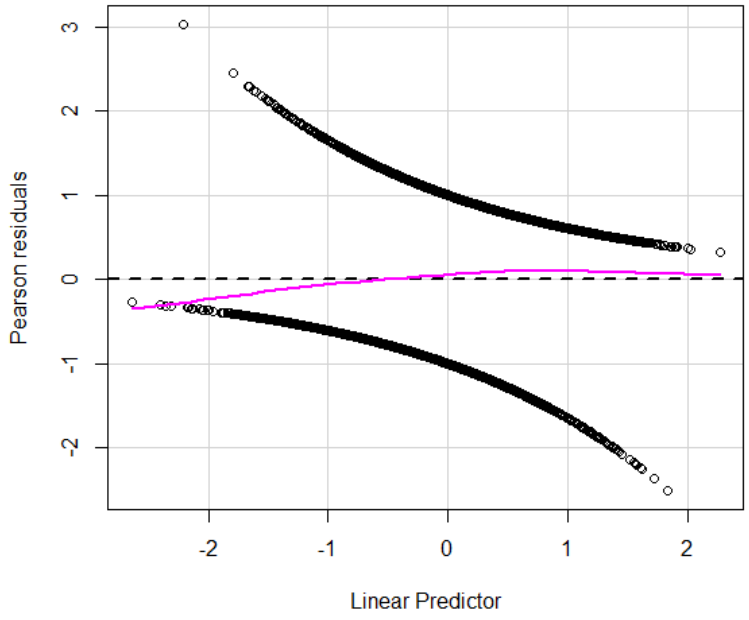
Variable	Parámetro	Interpretación
$SurfaceTour$	3.789859	Por cada diferencia de 0.1 en el porcentaje de victoria según tipo de pista y torneo la probabilidad de ganar dicho jugador aumenta en un 38'17%
Age	-0.021795	En este caso tenemos un parámetro negativo, por lo tanto, por cada año de más que tenga el jugador su probabilidad de victoria descenderá en un 2'06%
$Height:Hard$	0.011838	Por cada centímetro que sea más alto que el otro jugador, disputándose el partido en pista *Hard*, la probabilidad de ganar dicho jugador aumenta en un 1'18%
$h2hDif_6$	0.124109	En caso de que un jugador haya ganado todos los previos enfrentamientos entre ambos en los últimos 6 años, la probabilidad de victoria aumentará en un 12'41%
$h2hDif_3$	0.176542	En caso de que un jugador haya ganado todos los previos enfrentamientos entre ambos en los últimos 3 la probabilidad de victoria aumentará en un 17'65%
$rf_{(4,12)}$	2.471392	Por cada diferencia a favor de un jugador de 0.1 en el estado de forma en los últimos 12 meses respecto los últimos 4 años el porcentaje de victoria aumentará en un 24'71%
$rf_{(4,3)}$	0.861058	Por cada diferencia a favor de un jugador de 0.1 en el estado de forma en los últimos 3 meses respecto los últimos 4 años el porcentaje de victoria aumentará en un 8'61%

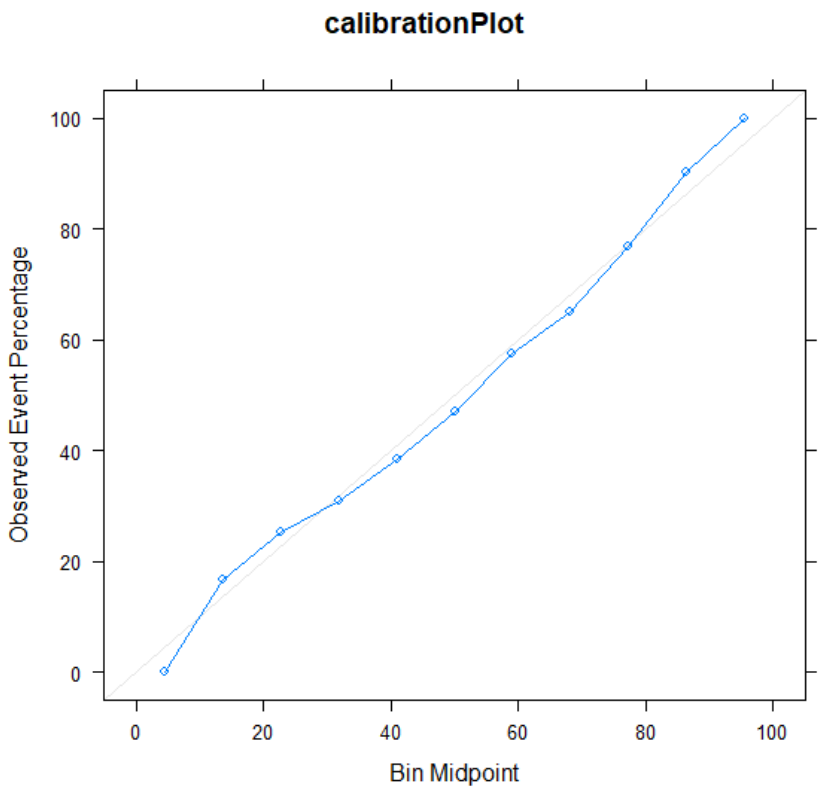
Como podemos observar, tan solo la edad tiene un efecto negativo en el porcentaje de victorias. Además, vemos como el modelo no se ha quedado con nada más una variable referente al estado de forma, pero si se observa que tiene más peso el estado de forma en el último año que en los últimos 3 meses.

Para finalizar debemos ver como de bueno es el modelo y validarlo antes de pasar a la simulación.

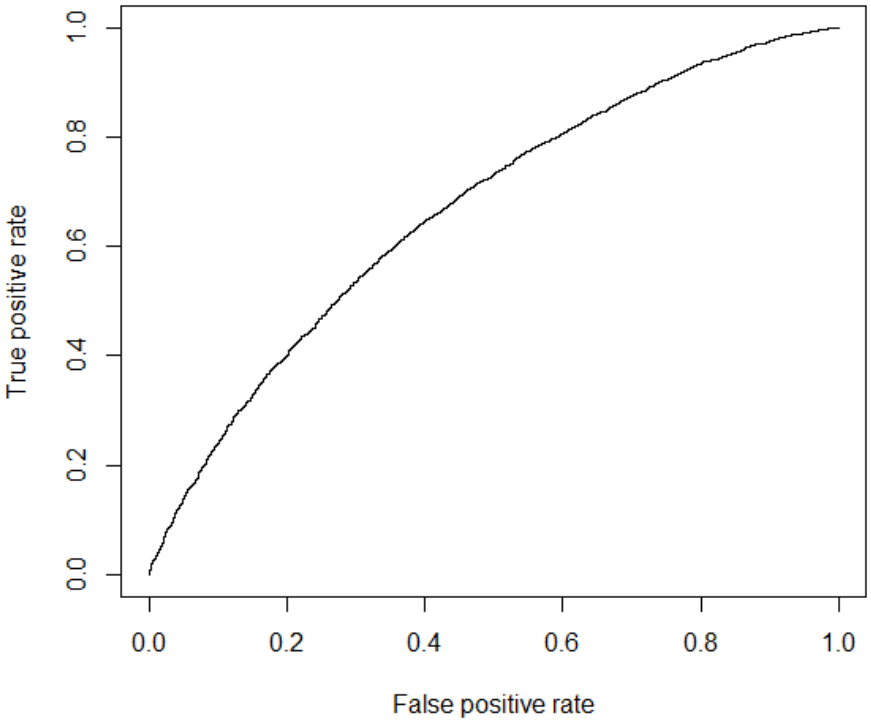


Encontramos arriba el ajuste de las predictoras. Podemos ver que el ajuste es muy bueno con la edad y él cara a cara. Se desvía un poco más en el resto de las variables. Vemos ahora el ajuste del modelo.



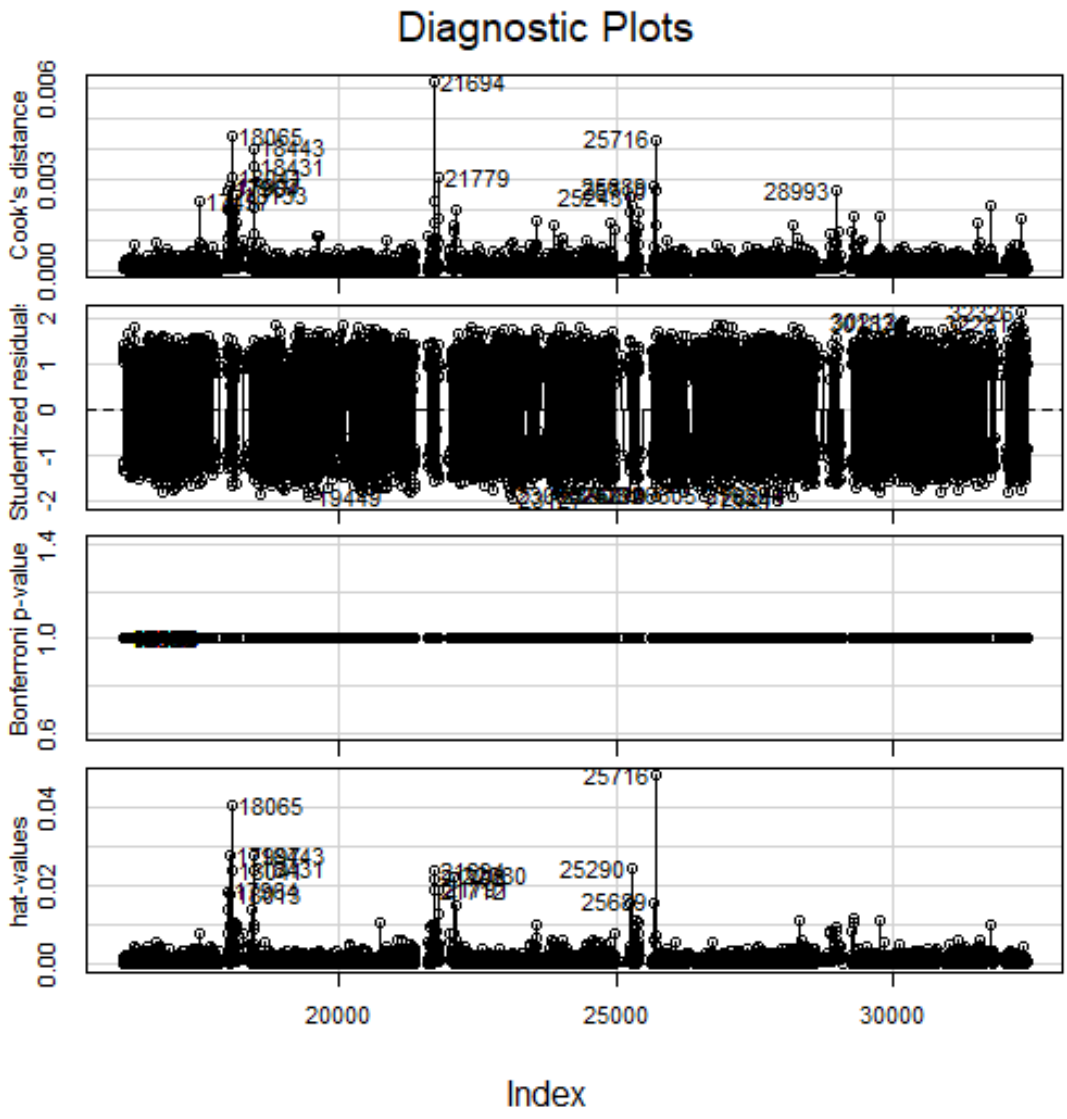


Podemos observar como el ajuste del modelo es bastante bueno, aunque todavía podría ser mejor. Para la capacidad predictiva del modelo tenemos el siguiente gráfico con la curva de *ROC*.

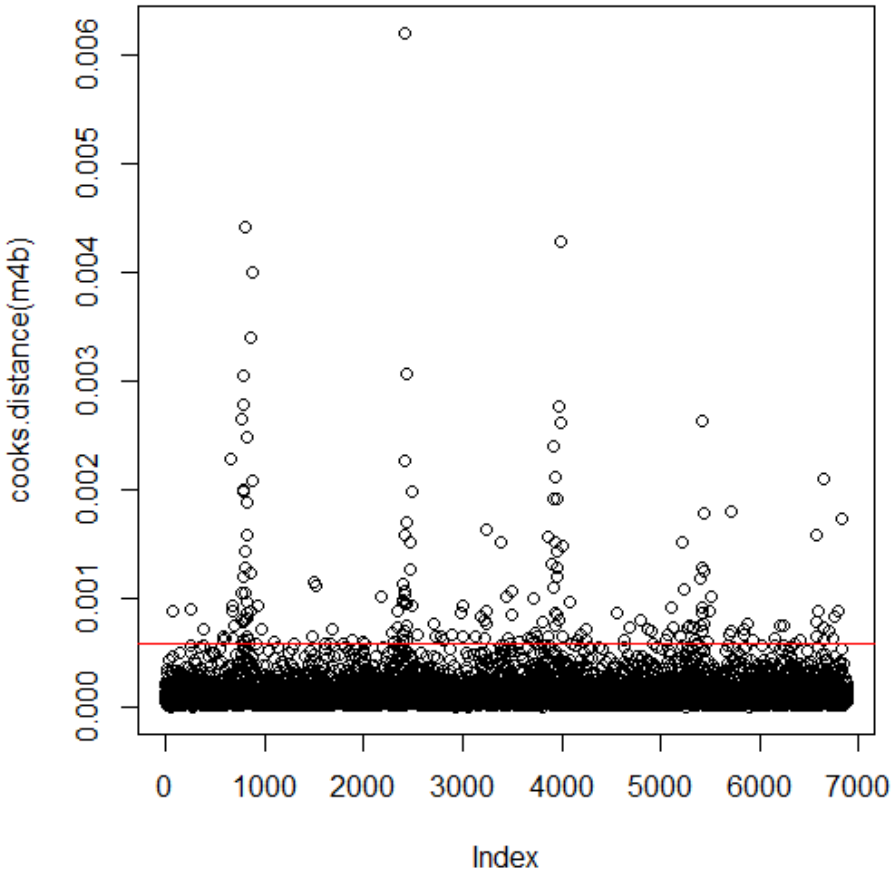


Siendo el estimador 0.67, lo cual nos indica que la capacidad predictiva del modelo no es mala pero que tampoco es muy elevada. Aun así, todavía podemos obtener una mejor capacidad predictiva.

Realizamos un análisis de *outliers* y observaciones influentes a posteriori, las cuales pueden estar afectando a las estimaciones y como consecuencia empeorando el modelo. Todo esto se obtiene a raíz de la función *influenceIndexPlot*( ).



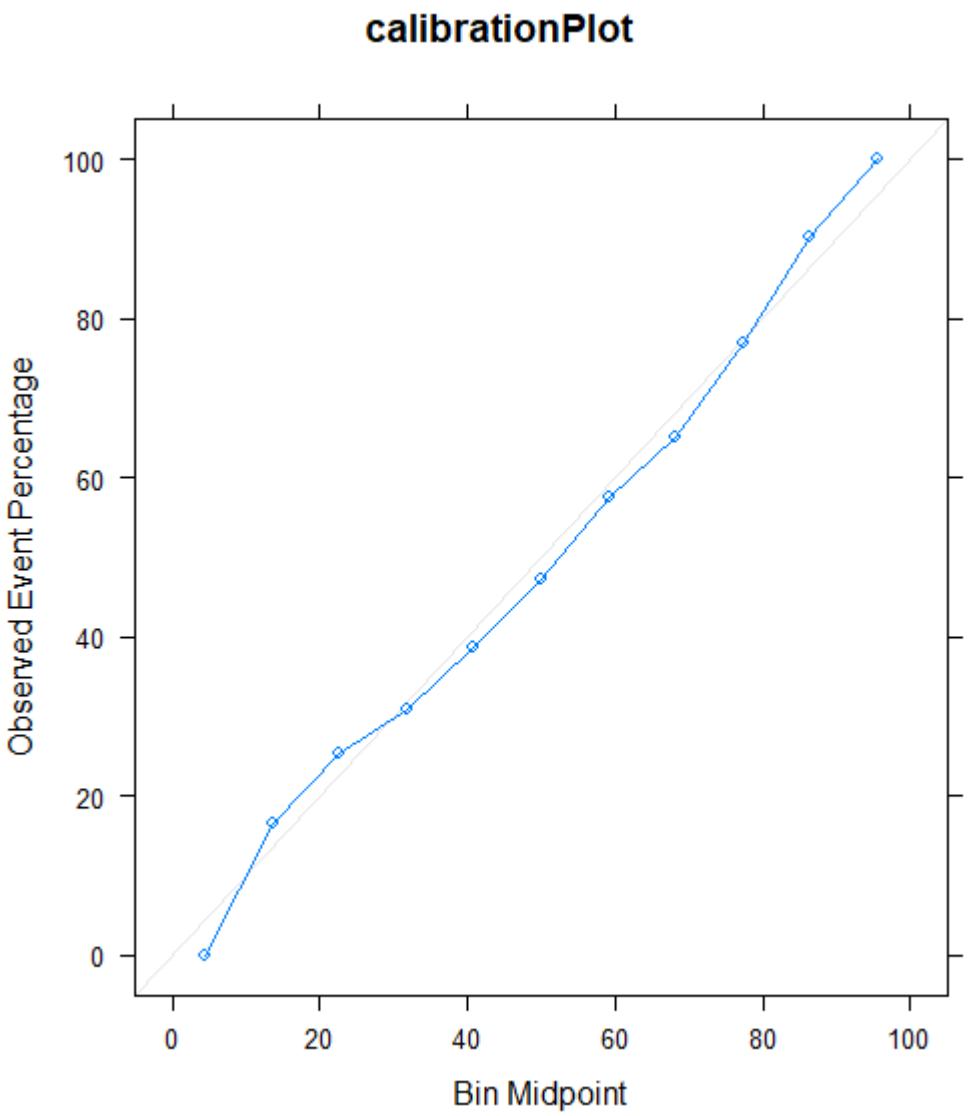
No obstante, de esta forma somos incapaces de ver realmente que está afectando al modelo. Al realizar un *OutlierTest* obtenemos que solo una observación se le considera *outlier*. En cuanto a las observaciones influentes encontramos el siguiente gráfico



Tenemos todas las distancias de Cook junto a una constante representada de color roja que indica el *cutoff*. Este *cutoff* el cual es igual a 4/n, nos indica que observaciones son influentes. Con tal de mejorar el modelo eliminamos estas observaciones que hemos encontrado. El *AIC* del modelo ahora es de 8496 frente al valor de 8950 que teníamos antes, por lo que vemos ya una mejora en el ajuste del modelo. Además, una vez hecho esto tenemos una mejora en el estimador del *AUC* de la curva de *ROC*, medida para analizar la capacidad predictiva. Ahora hemos pasado de tener 0.67 a un *AUC* de 0.7, por lo que ya podemos decir que el modelo es bueno prediciendo, no obstante, podría ser mejor ya que todavía tenemos un valor bastante reducido. Vemos las nuevas estimaciones de los parámetros del modelo.

Variable	Parámetro Antiguo	Nuevo Parámetro
$SurfaceTour$	3.789859	4.539629
Age	-0.021795	-0.025541
$Height{:}Hard$	0.011838	0.013363
$h2hDif_6$	0.124109	0.169776
$h2hDif_3$	0.176542	0.148620
$rf_{(4,12)}$	2.471392	2.850778
$rf_{(4,3)}$	0.861058	1.284958

Podemos observar como todas las estimaciones han cambiado. Algunas en menor cantidad, pero las que variables que ahora tienen un mayor peso en comparación con el que tenían previamente son *SurfaceTour* que ha aumentado en 1 la estimación y el estado de forma en los últimos 3 meses que ha aumentado en casi 0.7. Vemos finalmente de nuevo el *calibrationPlot* para ver el ajuste, donde observaremos que, aunque algunos puntos se alejen de la recta, el ajuste es bueno.



SIMULACIÓN

Como ya se ha comentado previamente, el objetivo del trabajo es poder predecir el rendimiento a futuro de los jugadores del circuito ATP a partir de una simulación del 2021 y comparándolo con su rendimiento real. Para ello se plantean dos opciones

- Simular los cuadros reales

- Simular cuadros realizando sorteo y colocando cabezas de serie

El cuadro de un torneo se genera de la siguiente forma. Dependiendo del número de participantes se colocan un tanto de jugadores en la segunda ronda del torneo, por lo que no tienen rival en la primera. Estos jugadores son decididos por el *ranking* en el que se encuentran, es decir, si son 8 jugadores cabeza de serie, los 8 mejores en *ranking* de los que se han apuntado no tienen rival. La segunda opción consistía en colocar a cabezas de serie y sortear el resto del cuadro, pero esto era muy costoso e inalcanzable. Primero de todo que hay varios torneos que se disputan a la vez, cada jugador tiene unas preferencias para decidir en cuál participar y algunos jugadores como por ejemplo Rafa Nadal no participan en torneos 250 y en pocos 500. Esto hace que se haga imposible seleccionar los jugadores que participan en un torneo u otro. Aun escogiendo dichos jugadores de forma aleatoria y agregando al algoritmo un mínimo de puntos a repartir para que x jugadores participen, encontramos otro problema. Es necesario tener en todo momento los puntos a repartir en cada torneo en otros años. Dado que esto sería muy costoso, no dispongo de ninguna base de datos creada donde esté esta información y además en consecuencia con la pandemia el funcionamiento de los *rankings* fue mucho más complejo, se ha escogido la primera opción, simular los cuadros reales. Es por este motivo también que no se ha añadido en el modelo una variable que sea *ranking*. En las primeras ideas que tenía si estaba, pero al no poder obtenerlo prácticamente se decidió prescindir de ella. De ahí que se plantearan diferentes períodos de tiempo para los porcentajes de victorias junto al estado de forma, ya que esto sustituye el *ranking*, que al final no deja de ser una métrica donde se indica el rendimiento en el último año.

Además de evitar estos problemas seleccionando la primera opción, tenemos dos puntos más a favor. Es mejor opción para comparar con los resultados reales, ya que se puede comparar los puntos que se esperaba obtener por cada jugador con unos cuadros específicos frente a los realmente obtenidos. También tenemos en cuenta las lesiones, sí que es verdad que esto es algo que se podría aleatorizar, si queremos comparar con los resultados reales veo necesario que los puntos a los que puede optar cada jugador sean los mismos.

Aun cubriéndonos de tantos problemas, hay algunos los cuales no se pueden solucionar. Algunos de ellos son cubiertos de cierto modo con la aleatoriedad a la hora de seleccionar un jugador. A continuación, encontramos un par de cosas que no se tienen en cuenta o posibles problemas.

- Jugadores que jueguen con molestias, enfermos, vuelvan después de una enfermedad, separación con un entrenador, condiciones meteorológicas adversas en el partido…

- Puntos y partidos ganados en el circuito *Challenger*

- Jugadores con pocos partidos pueden dar problemas en las métricas

Antes de poder realizar la simulación hace falta un registro de los cuadros. No he encontrado ningún repositorio gratuito que se pueda importar sobre esto, por lo que se ha creado a partir de los partidos disputados en 2021. Este ha sido el procedimiento.

Primero se importan los partidos, es necesario que la fecha esté en el formato correcto, por lo que nos aseguramos de ello. Después eliminamos los torneos que no reparten puntos y los *Master Finals*, más adelante vemos como simulamos esto.

Agrupamos y añadimos una etiqueta a cada torneo para poder saber cómo reparte los puntos para más adelante. Una vez hecho eso separamos los partidos por torneo y nos quedamos con el primero.

Guardamos todos los nombres junto a su edad y altura en el orden que aparece. Ahora leemos los partidos de segunda ronda y comprobamos si los nombres de los jugadores de dicho partido existen en el vector que hemos guardado con los partidos de primera ronda. En caso de que no existan ambos nombres significa que ninguno tuvo rival en la primera ronda, por lo que se generará (BYE, Nombre1, BYE, Nombre2) en el registro final del cuadro. BYE es el identificador de que el jugador no tiene rival y más adelante se usará para darle una probabilidad de 1 al jugador y que pase en todos los casos. En caso de que exista solo uno, se agregará los dos jugadores del partido registrado previamente junto al nuevo jugador junto a un BYE. En caso de que ambos existan se agregan los 4 jugadores que ya existían, los ganadores que se enfrentan en el partido de segunda ronda que estamos observando y los perdedores que se enfrentaron a estos en la ronda anterior. Una vez hecho eso obtenemos el siguiente output.

jugadores	edad	ht	torneo	surface	tourney_date	tourney_type
John Isner	35	206	1	Hard	2021-01-04	A
BYE	*NA*	186	1	Hard	2021-01-04	A
Thiago Monteiro	26	183	1	Hard	2021-01-04	A
Thomaz Bellucci	33	188	1	Hard	2021-01-04	A
Sebastian Korda	20	196	1	Hard	2021-01-04	A
Soon Woo Kwon	23	180	1	Hard	2021-01-04	A
Tommy Paul	23	185	1	Hard	2021-01-04	A
Ji Sung Nam	27	183	1	Hard	2021-01-04	A
Adrian Mannarino	32	183	1	Hard	2021-01-04	A
BYE	*NA*	186	1	Hard	2021-01-04	A
Cameron Norrie	25	188	1	Hard	2021-01-04	A
Jc Aragone	25	178	1	Hard	2021-01-04	A
Bjorn Fratangelo	27	183	1	Hard	2021-01-04	A
Kevin King	29	190	1	Hard	2021-01-04	A
Frances Tiafoe	22	188	1	Hard	2021-01-04	A

Aquí tenemos la información de jugadores que participaron junto a los datos de interés para las predicciones.

- Edad y altura → Se usan como predictoras en el modelo
- Tipo de pista, fecha y tipo de torneo → Se usan para calcular las métricas utilizadas para el resto de las variables explicativas
- Tipo de torneo detallado → Para poder actualizar el *ranking* y sumar los puntos de cada uno

Cada pareja de jugadores en el orden que se encuentran corresponde a un partido. También haciendo un unique por nombres en este *data frame* obtenemos otro junto a todos los jugadores para usar como repositorio de los puntos para el *ranking*. Como se ha comentado previamente, los jugadores que no tenían altura se les ha añadido manualmente en caso de que hayan disputado una gran cantidad de partidos en esta temporada, mientras que a los que no se ha rellenado con la media de los datos.

Para poder realizar la simulación, como hemos comentado, hace falta saber los puntos que se reparten. Ya sabemos que hay diferentes tipos de torneos los cuales reparten una cantidad de puntos mayor o menor a los ganadores. No obstante, también influye la cantidad de jugadores que participan, ya que en un mismo tipo de torneo puede haber más o menos competidores. Esto ocurre con los *Masters* 1000. El hecho de tener más jugadores inscritos hace que haya una ronda más en juego, puesto que se juega una ronda más. Dado que estos datos no son muchos, se ha introducido de forma manual.

Torneo	128	64	32	16	8	4	2	1
250	0	0	10	20	45	90	150	250
500	0	0	20	45	90	180	300	500
MP	0	10	45	90	180	360	600	1000
MG	10	25	45	90	180	360	600	1000
G	10	45	90	180	360	720	1200	2000

Como podemos observar tenemos el tipo de torneo, siendo MP un *Masters* 1000 con cuadro pequeño y MG uno con cuadro grande, además de tener los puntos que se reparten en cada ronda. Para identificar la ronda en la que se disputa el partido se ha puesto como nombre a la columna el número de jugadores restantes que hay, siendo por ejemplo 2 la final y 1 los puntos para el ganador.

Una vez recopilada toda esta información procedemos a la simulación. En el código anexado encontramos todo el procedimiento, el cual es el siguiente.

Primeramente, separamos el primer torneo que encontramos en los cuadros. Cada torneo tiene un identificador único, variable que se usa para separarlos. Previamente hemos visto como está almacenada la información de los torneos.

Una vez tenemos esto, calculamos las métricas, que previamente hemos visto que son las óptimas para el modelo, para dichos jugadores en el contexto que nos encontramos. Cada torneo tiene sus características, superficie en el que se juega, el tipo de torneo que se está disputando y la fecha en la que ocurre. Toda esta información es necesaria y se usa para calcular las variables explicativas del modelo.

Cuando ya tenemos todo calculado podemos emplear la función *predict*( ) junto al modelo antes creado para obtener las probabilidades de victoria para cada pareja de jugadores. Junto a esta probabilidad obtenemos un ganador de forma aleatoria utilizando la distribución de Bernoulli. Esta distribución obtiene un valor (n=1) entre 0 y 1 a partir de una probabilidad. De esta manera, en caso de ser 1 ganaría el primer jugador por orden en el que se encuentran en el *data frame* y 0 el segundo. Para cada jugador que haya perdido se le eliminará del cuadro y se le sumará en el *ranking* los puntos correspondientes a llegar a la ronda en la que se esté disputando el torneo. Además, se añadirá el partido disputado junto al ganador y perdedor a la base de datos de los partidos del circuito ATP de los últimos 4 años para poder calcular futuras métricas para otros partidos, teniendo en cuenta los datos obtenidos en la simulación. Una vez hecho esto, repetiremos el proceso hasta que se hayan eliminado tantos jugadores como para tener solo un registro. Este registro es el ganador final del torneo. Para finalizar con la simulación del torneo le sumamos los puntos correspondientes al tipo de torneo que se está disputando y pasamos al siguiente cuadro. Todo esto se realiza hasta terminar con los 60 torneos que se disputaron en el 2021. Una vez terminado todo esto procedemos a simular los *Masters Finals*, los cuales tienen un funcionamiento distinto.

El ATP *Finals*, o comúnmente conocido como copa de maestros, es un torneo que se disputa al final de la temporada donde participan los 8 mejores jugadores de la temporada. A diferencia de los demás torneos del año, el *Masters* no es un torneo de eliminación directa. Los ocho participantes se reparten en dos grupos de 4 jugadores, y cada tenista juega contra los otros tres de su grupo. Los dos jugadores mejor clasificados en esas dos liguillas juegan las semifinales ya eliminatorias, quedando solo dos que disputan la final. Los dos primeros jugadores se sitúan en diferentes grupos, mientras que el resto de los jugadores se sortean. La puntuación se reparte entre los jugadores así: En la primera ronda se concede por cada partido ganado 200 puntos, si se gana 3 partidos suman en total 600, si se gana un solo partido se otorgan 200 puntos. Al llegar a la final se suman 400 puntos a los que ya obtenidos; ganar el campeonato supone sumar otros 500 puntos.

Con esto en mente realizamos la simulación de la copa de maestros con los 8 mejores jugadores obtenidos en la predicción del resto de torneos. Se simula la fase de grupos y después la fase de eliminación desde semifinales de la misma forma que hacíamos previamente.

Una vez terminado esto tendremos un año entero simulado. Para finalizar del todo guardamos la información de tanto los puntos obtenidos como la posición del *ranking* en la n simulación en dos data frame para poder obtener los resultados por jugador más adelante. Con el objetivo de obtener unos resultados más exactos y no tan sujeto a la aleatoriedad se repite la simulación N veces.

RESULTADOS

A continuación, veremos los resultados obtenidos en el trabajo. Nuevamente, lo separamos en tres grandes apartados. Un primer apartado donde podremos ver el comportamiento del modelo al realizar predicciones con algunos partidos interesantes del 2022 para complementar la validación previamente vista. Un segundo apartado donde se expondrán los resultados obtenidos de forma general en las simulaciones del circuito ATP en el 2021. Para finalizar, a partir de unos jugadores de interés, se realiza un análisis más específico para, como se ha comentado antes en el trabajo, ver el rendimiento de los jugadores tanto en la realidad como en la simulación para poder realizar una predicción, además de poder comparar dicha predicción con los torneos ya disputados hasta junio de 2022.

PRUEBA MODELO

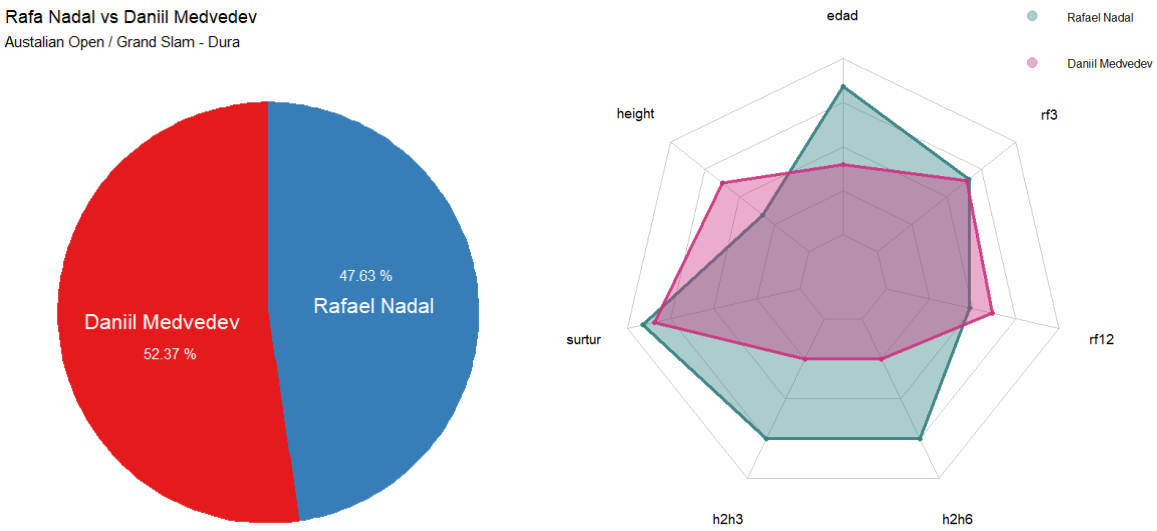
Como ya sabemos, el objetivo es poder realizar una simulación del año entero con el modelo previamente obtenido. Además de su validación, creo que es importante aplicar el modelo para algunos partidos de esta temporada para poder ver su comportamiento y resultados con algún caso más específico. Para ello se analizarán los siguientes partidos junto a los torneos en los que se disputaron.

- Rafael Nadal - Daniil Medvedev / Open Australia
- Daniel Evans - Andrey Rublev / Dubái Open
- Carlos Alcaraz - Novak Djokovic / Masters de Madrid
- Novak Djokovic - Rafael Nadal / Roland Garros

Estos partidos han sido escogidos para poder ver un ejemplo en cada tipo de pista y torneo que se recogía en la simulación además de ser partidos famosos de este 2022.

Nadal - Medvedev

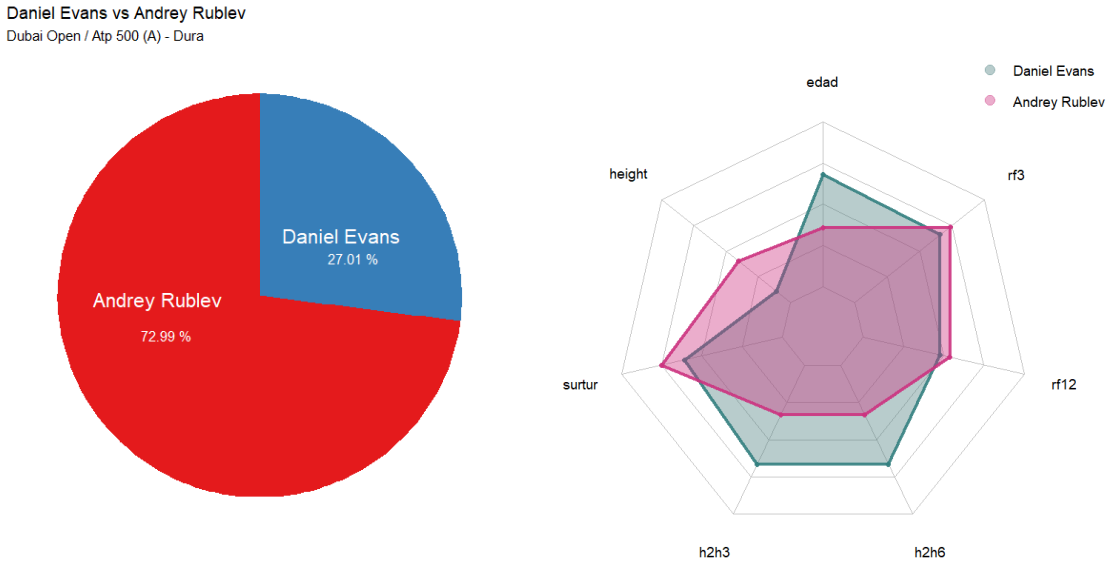
El primer partido con el que probaremos el modelo es el partido que enfrentó al ruso y al español en la primera final de un Grand Slam del 2022. En ese momento, Medvedev llegaba como el gran favorito para llevarse el título, mientras que Nadal llegaba de forma silenciosa tras un cuadro que se lo puso todo de cara.



Vemos como las probabilidades para este partido se acercan a ser 50-50 para ambos jugadores según el modelo. Esto, en un partido donde el ruso se planteaba el favorito indiscutible. En el gráfico de radar vemos las comparaciones entre ambos jugadores para las diferentes variables del modelo. Vemos como Rafa lidera en ambos cara a cara y en el porcentaje de victorias, no obstante, Daniil es más joven, más alto (cosa que influye, ya que se juega en pista dura) y venía en mejor forma. Las casas de apuestas le daban un 62% de ganar, un 10% más que el modelo. El partido acabó resultando en el vigésimo primer Grand Slam para el español tras una remontada en un partido muy reñido que terminó 2-6 / 6-7(5) / 6-4 / 6-4 / 7-5.

Evans - Rublev

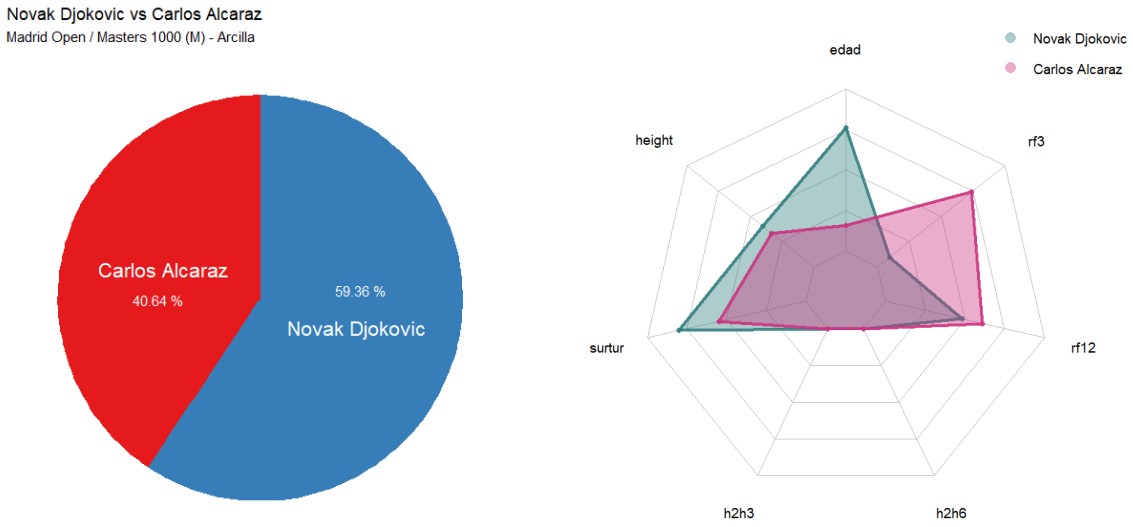
En este caso nos encontramos con el partido que se disputó en el ATP 500 de Dubái en dieciseisavos. Se enfrentaban por sexta vez en su carrera el británico Daniel Evans y el ruso Andrey Rublev.



Aquí vemos como Rublev domina en todos los aspectos, a excepción de los partidos previos entre ambos jugadores. Esto se plasma en la aplastante diferencia en la probabilidad de victoria del ruso, que en este caso es de un 72%. Las casas de apuestas le daban un 60% de victoria, pero en este caso el ruso nos daba la razón al modelo mostrando esa superioridad que se esperaba ganando en dos mangas.

Djokovic - Alcaraz

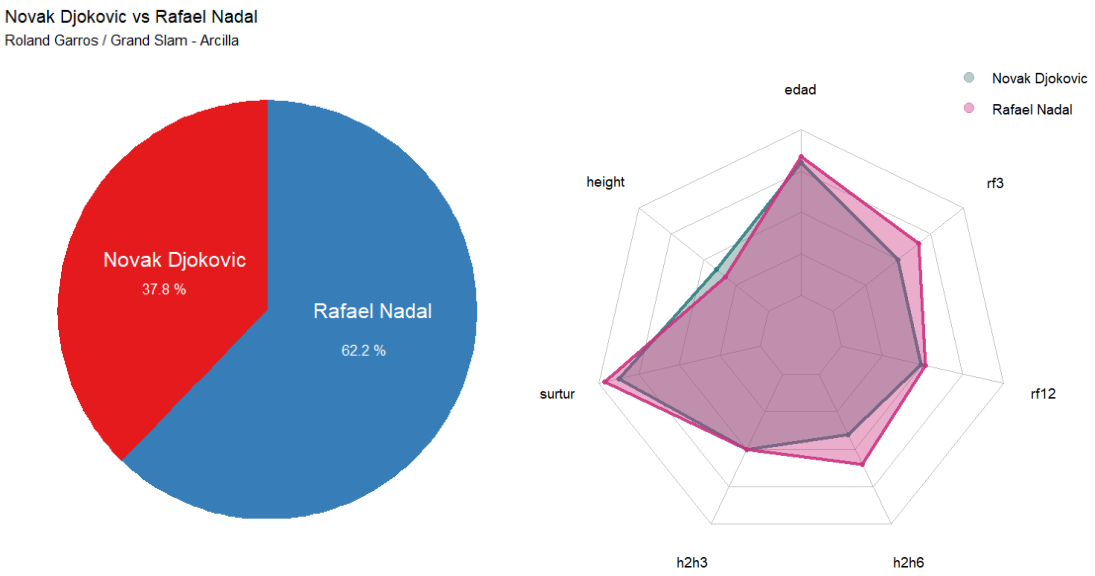
No siempre el modelo acierta, eso es obvio. Uno de esos partidos donde no pudo leer bien el resultado fue entre él tantas veces número 1, Novak Djokovic, con el llamado a tomar el relevo de Rafa, la joven promesa Carlos Alcaraz. El partido se sitúa en las semifinales del *Masters* 1000 de Madrid. Djokovic se plantaba como ligero favorito tanto para las casas de apuestas como en el modelo, como vemos a continuación.



Ligeramente favorito Djokovic con un 60% de victoria y un 55% según las casas de apuestas. Vemos como no hay valores para él cara a cara, ya que era el primer enfrentamiento de la historia entre estos dos jugadores. La altura, aún siendo parecida, no afecta en este partido, puesto que se disputó en arcilla, pista y tipo de torneo donde el serbio tiene mejor bagaje. No obstante, la juventud de Carlos y el estado de forma tan pletórico en el que venía le daba más oportunidades de las que se esperarían normalmente. El resultado fue una victoria muy ajustada en un tie-break en el tercer set para el murciano, sorprendiendo así a todo el mundo.

Nadal - Djokovic

El último partido donde probaremos el modelo será entre dos eternos rivales como son Nole y Rafa en los cuartos de final del Roland Garros, torneo de arcilla que se disputa en París.



Tenemos un partido con condiciones muy iguales entre ambos jugadores, pero con ventaja para el español dada su mejor forma en los últimos meses y su ligera superioridad en este torneo ante su rival. No obstante, en las casas de apuestas daban un 35% de probabilidades de ganar a Rafa. Esto se debe en parte a las posibles molestias que podía acarrear para el partido y que en la ronda anterior tuvo un enfrentamiento muy duro. Sorprendentemente, para algunos, aunque no para el modelo, el de Manacor se anteponía en cuatro sets por un 6-2 / 4-6 / 6-2 / 7-6(4).

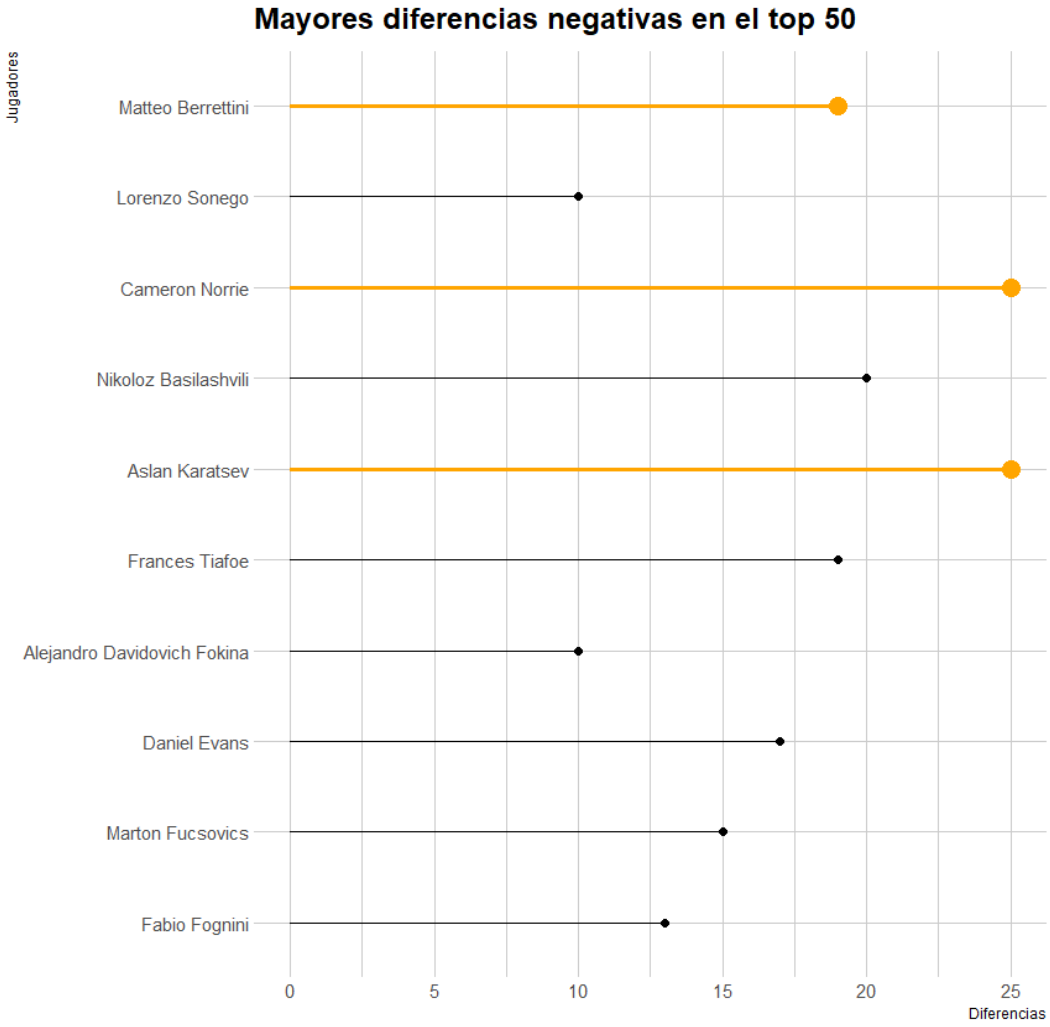
SIMULACIÓN

Como ya se ha comentado varias veces, el objetivo de la creación de este modelo es el de poder realizar una simulación del circuito ATP en el 2021. Para ello se han efectuado 500 simulaciones para poder encontrar el resultado final del *ranking* minimizando la aleatoriedad. Me hubiera gustado hacer más simulaciones, pero el coste computacional era enorme, por lo que no he podido hacerlo.

Ahora veremos los resultados generales de obtenidos. Por cada simulación obtenemos los puntos y el *ranking* de cada jugador en cada bucle. Con ello obtenemos los puntos medios obtenidos en el año para tener un *ranking* final. A continuación, tenemos el top 8 final obtenido y, por lo tanto, los jugadores que según la simulación clasificarían al ATP *Finals*.

Jugador	País	Edad
Alexander Zverev		25
Daniil Medvedev		26
Novak Djokovic		35
Stefanos Tsitsipas		23
Reilly Opelka		24
Andrey Rublev		24
Casper Ruud		23
Felix Auger Aliassime		21

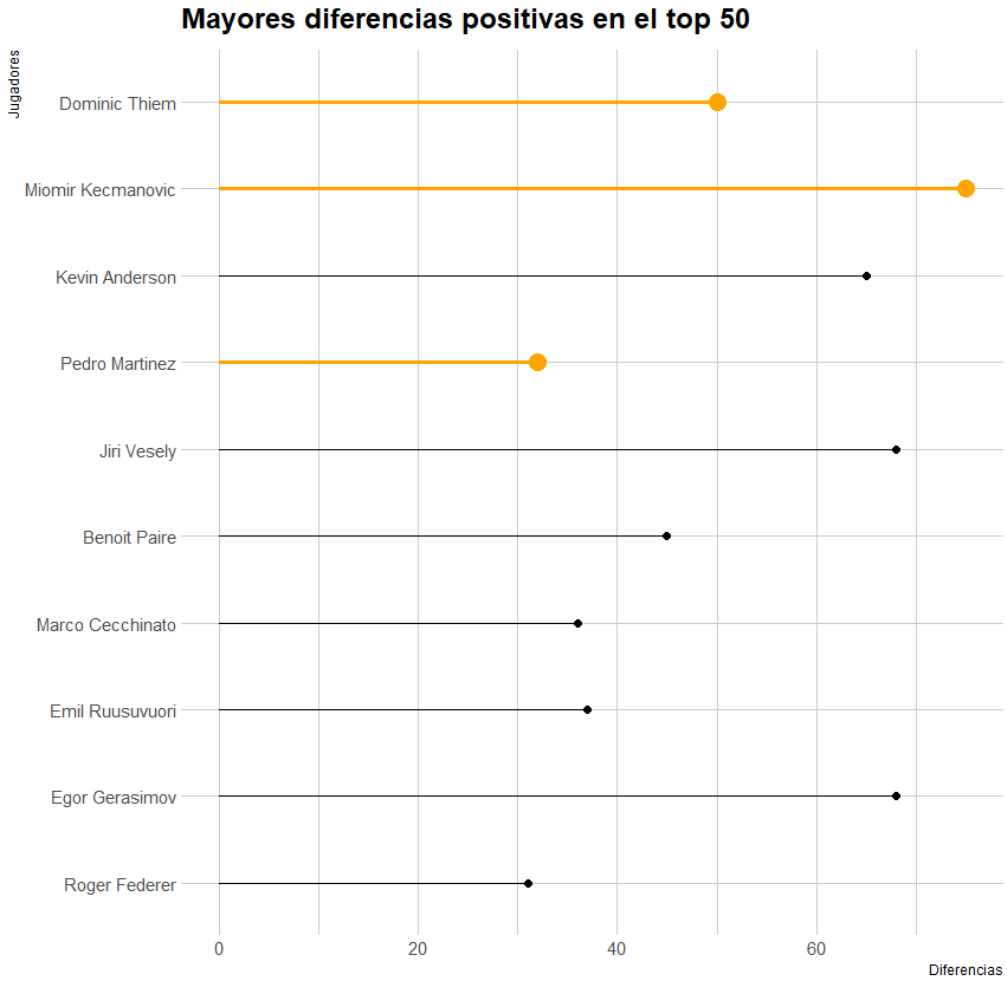
Ahora sí, lo interesante también es poder comparar los *rankings* obtenidos en la simulación con lo que realmente ocurrió. Es por ello por lo que para cuantificarlo restamos al *ranking* real el simulado. Así pues, las mayores diferencias negativas, es decir, donde se rindió más de lo esperado, son las siguientes.



Como el propio título del gráfico indica, nos hemos fijado solo en las diferencias negativas de jugadores que se encontraban en el top 50. Esto se debe a que solamente se han simulado partidos de nivel open hacia arriba y no *Challenger*. La mayoría de los jugadores que componen las posiciones entre la 100 y la 50 participan en bastantes fechas de este tipo de nivel de torneos. Para poder sacar conclusiones con un peso real nos tenemos que fijar únicamente en estos jugadores que sus puntos son todos de los torneos que hemos utilizado para la simulación, en este caso, los 50 primeros.

Nos fijamos en los jugadores de mayor *ranking*. Vemos que Berrettini ha quedado infravalorado por parte del modelo, en su caso terminó el año en la posición número 6 mientras que la simulación lo sitúa en la 25. Esto puede ser debido a cuadros difíciles a principio de año y que esto haya sido acarreado toda la temporada, ya que el italiano ya tenía buenos números a principio de año.

Por otra parte, encontramos jugadores como Karatsev y Norrie, ambos jugadores que ya llevaban un tiempo en el circuito, pero que no lograban resultados muy notorios. En este 2021 dieron un salto de nivel ganando ambos algunos títulos y demostrando un nivel enorme. Este salto de nivel no se ha sido capaz de cuantificar por el modelo, colocándolos en el 36 y 38, respectivamente, cuando terminaron 11 y 13 del circuito, 25 posiciones de diferencia.

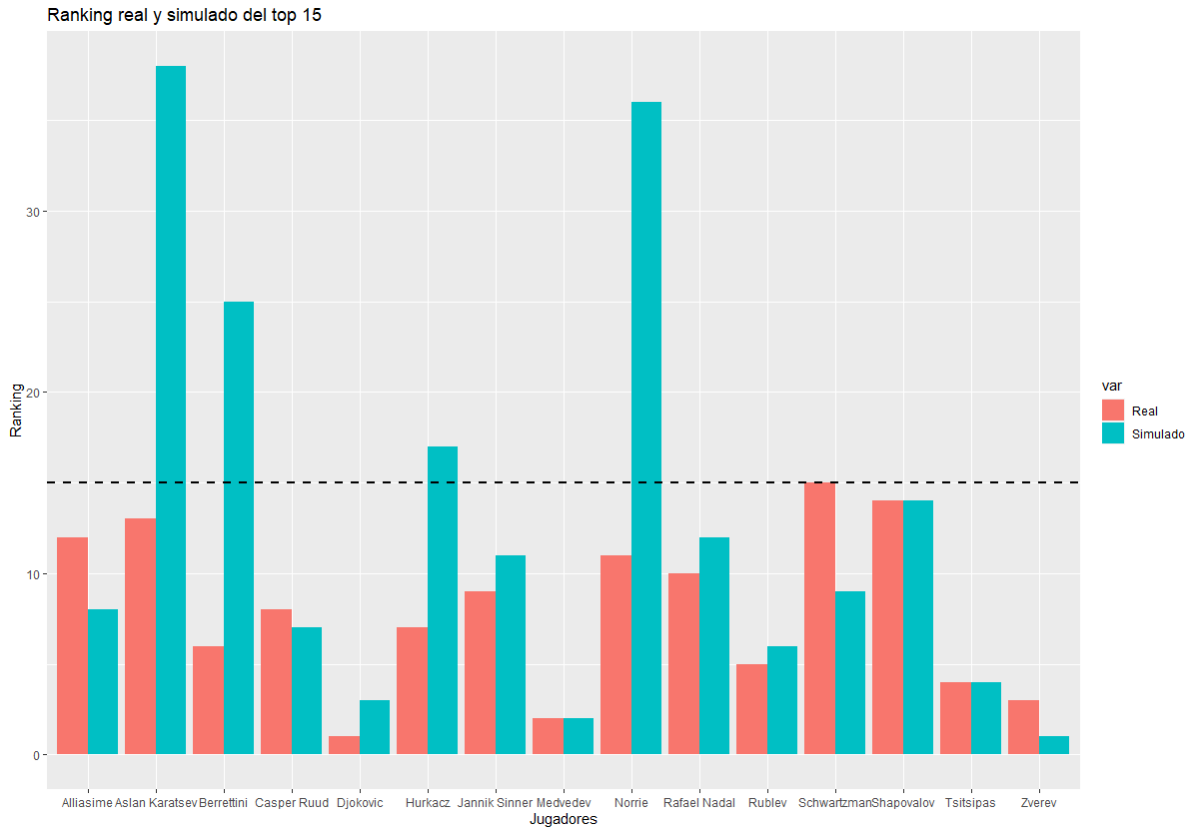


Volvemos a encontrar el mismo tipo de gráfico, pero con las diferencias positivas. En este caso tenemos a los jugadores los cuales, según la simulación, se esperaba que diesen un mejor resultado en el año. Marcados volvemos a tener a los jugadores de mayor *ranking* según la simulación. Podemos observar como las diferencias entre *rankings* son mayores que antes. La mayor diferencia la tenemos en Kecmanovic de 75 puestos mientras que antes era de -25 para Aslan y Cameron. La diferencia es tan grande que le dedicaremos un análisis más profundo más adelante.

El caso de Dominic Thiem es especial. El modelo lo sitúa en el puesto número 19 mientras que realmente terminó el 69 en la *race*. Esto se debe a las lesiones. El austriaco pasó un infierno con sus lesiones en la temporada anterior y esto, cómo ya comentábamos en los primeros apartados de la metodología, es muy complicado de poder medir. El ex número 3 del circuito tiene unos números fabulosos en torneos ATP por lo que probablemente, aunque haya disputado pocos torneos, se esperaba un gran rendimiento de él, mientras que en la realidad no llegó a estar al 100%.

El último caso que tenemos marcado es el de Pedro Martínez Portero. Pedro es un jugador bastante joven y un especialista en arcilla, es por ello que se esperaba una subida de nivel. Nada más lejos de la realidad, Tuvo dos lesiones durante la temporada, mismo caso que vimos antes, y sus mejores resultados fueron obtenidos en el circuito *Challenger* el cuál no se ha tenido en cuenta. Es por ello que se podía esperar una subida de nivel mayor por parte del español. Esta temporada empezó muy fuerte e incluso consiguió levantar su primer título ATP en Santiago, Chile. Por lo que la predicción no iba tan mal encaminada como nos indicaba la diferencia.

Una vez esto, nos centramos en las primeras posiciones. Al final lo más interesante es ver qué jugadores acabaran en las posiciones de arriba. A continuación, encontramos el top 15 del circuito junto a sus *rankings*.



Encontramos en azul el *ranking* simulado y en rojo el *ranking* real para los 15 jugadores con más puntos de la *race* del 2021. Vemos como se han acertado 11 de los 15 jugadores que se situarían en este intervalo, un 73% de los jugadores, un resultado muy positivo. Los 4 jugadores que se sitúan fuera son Hurkacz, que se queda en el límite en el puesto 17, y los tres jugadores que previamente vimos con las diferencias negativas. Podemos observar también como las diferencias entre los jugadores acertados en el top 15 no son muchas. Además, como último punto positivo, se han acertado a la perfección las posiciones de Denis Shapovalov, decimocuarto situado, Stefanos Tsitsipas, cuarto de la *race* y Daniil Medvedev, segundo del *ranking* y que por un corto período de tiempo fue capaz de robarle la primera plaza a Djokovic este año.

Una vez visto los resultados generales, vamos a fijarnos en los resultados de algunos jugadores de interés para hilar más fino en la simulación, estos jugadores son:

- Miomir Kecmanovic, por la gran diferencia que lo sitúa dentro del top 50
- Matteo Berretini, por ser un jugador de arriba y no haber podido predecirlo
- Carlos Alcaraz, jugador de interés personal
- Casper Rudd, jugador de interés por el año que realizó

JUGADORES

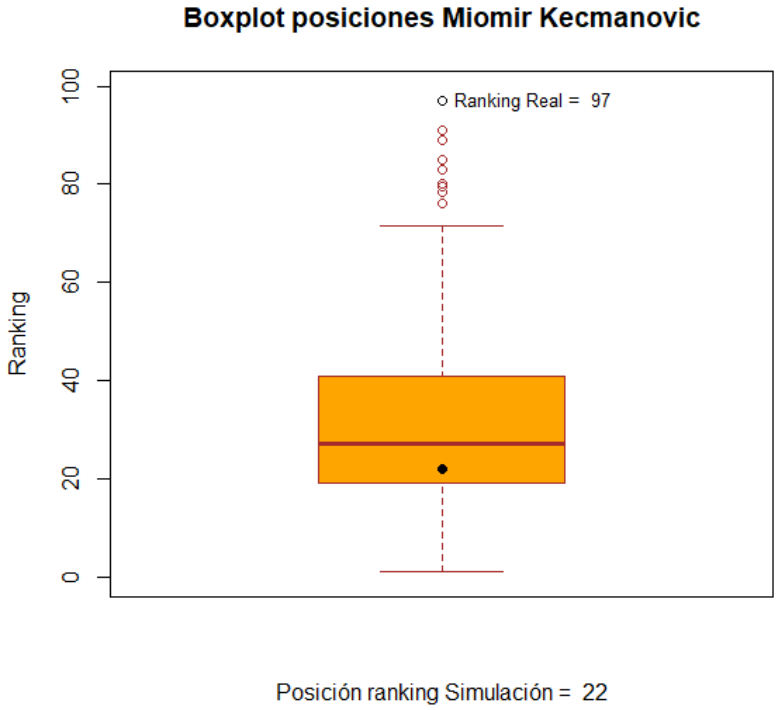
Una vez visto todo lo anterior, nos queda el último paso, predecir el rendimiento de un jugador a partir de los resultados esperados y observados. Para hacerlo debemos fijarnos con más detalle en los resultados simulados del jugador y contextualizarlos con el fin de poder explicarlos mejor. Para ello, de cada jugador veremos diferentes datos de interés. Uno de ellos es la posición anterior para la *race* del 2020, para ver la progresión del jugador.

Miomir Kecmanovic

Vemos un poco los datos del jugador

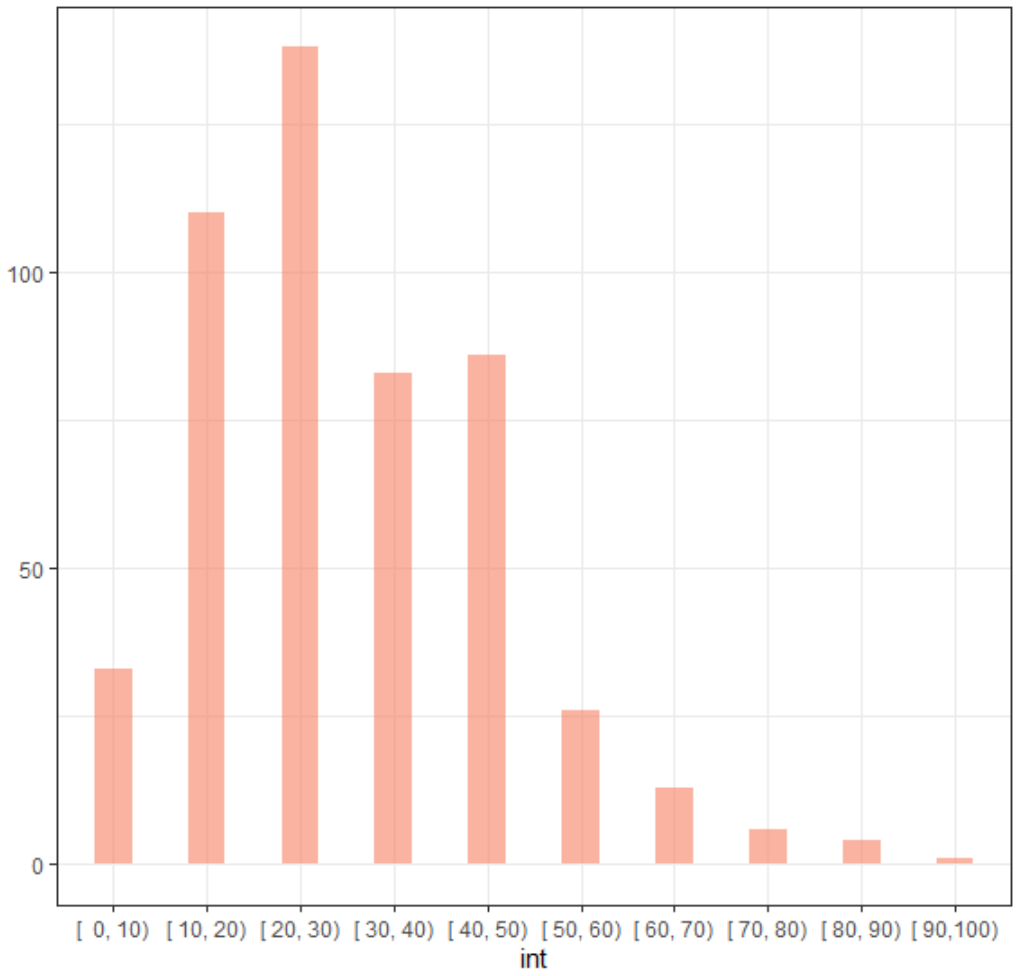
- Edad: 22 años
- Altura: 1.83 metros
- País: Serbia
- *Ranking race* 2019: 59
- Pista favorita: Dura, buenos resultados en arcilla también

Estamos frente a un jugador joven y polivalente. Veamos los resultados de la simulación.



Encontramos que el *ranking* real se sitúa muy lejos del esperado. Además, el 50% de los datos lo sitúan entre las posiciones 20 y 40, por lo que vemos que es constante en la mejora del *ranking* previo. Tenemos la existencia de varios *outliers*, siendo estos *rankings* de alrededor del puesto 85 de media, pero que ni en el peor de los casos alcanza el valor observado.

En la realidad, podemos ver como este jugador encajó bastantes derrotas seguidas sin tener realmente un buen resultado en ningún torneo, es por ello que terminó en el 97.



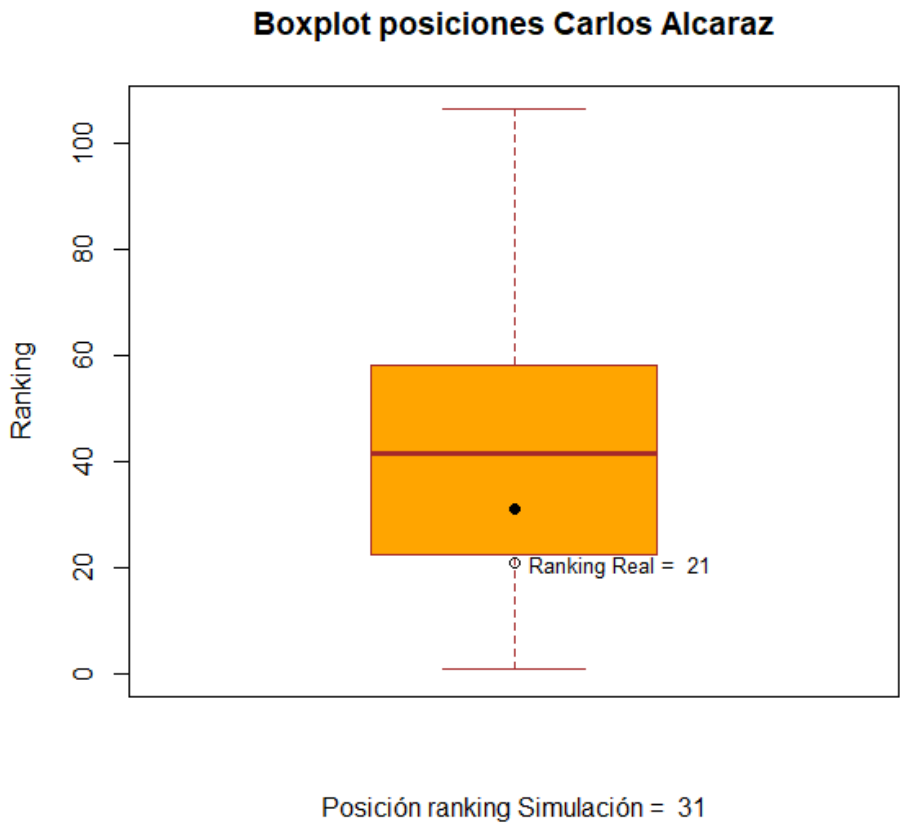
Aquí podemos ver la frecuencia de las posiciones logradas por intervalo. Vemos como los intervalos más repetidos son él [20,30) y el [10,20), resultados extremadamente buenos que supondrían un aumento notorio en su *ranking*. Incluso en bastantes ocasiones lo sitúa en el top 10. Esto nos hace indicar el gran recorrido de nivel que tenía y el nefasto año que realizó.

A partir de estos resultados, la predicción sería una clara subida del *ranking*. En este 2022 se podría esperar una subida de nivel para recuperar lo perdido y escalar puestos. Lo situaría mínimo por encima del puesto 50, mejorando su penúltimo resultado. Nada más lejos de la realidad, a mitad de esta temporada, el serbio ha mejorado y demostrado su nivel, ganando más partidos en 6 meses que todo el 2021 y actualmente, situándose en el puesto 30 de la race, por lo que estaríamos en un caso de acierto de la predicción hoy por hoy.

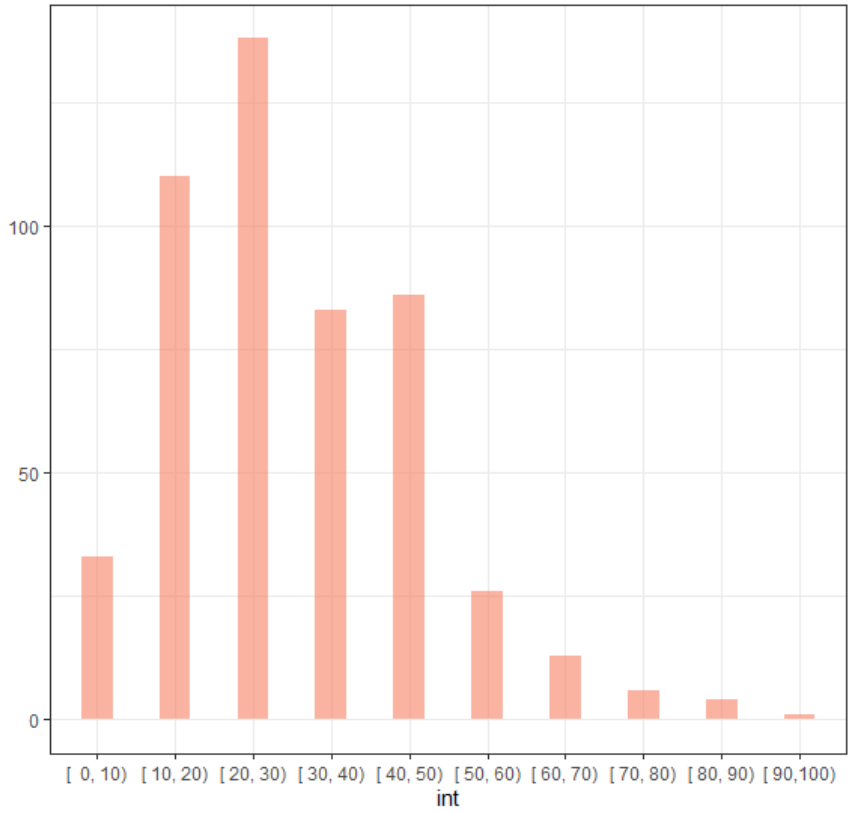
Carlos Alcaraz

- Edad: 19 años
- Altura: 1.83 metros
- País: España
- *Ranking race* 2020: 141
- Pista favorita: Por victorias arcilla, aunque personalmente él dijo que dura

En este caso tenemos a un jugador jovencísimo y con muchísimo potencial. No obstante, muy joven y con poca experiencia en el circuito, que previamente se sentaba fuera del top 100, por lo que sus puntos provenían del circuito Challenger. Veamos más de cerca sus resultados.



Vemos como no solo se espera que Carlos irrumpiera en el top 100, objetivo clásico de los jóvenes que disputan los circuitos menores, sino que parecía inevitable que rompiera la barrera de los 50 mejores. Ni más ni menos que una mejora de 110 puestos le daba la simulación. Aun así, el joven murciano, llegó a alcanzar el puesto 21.

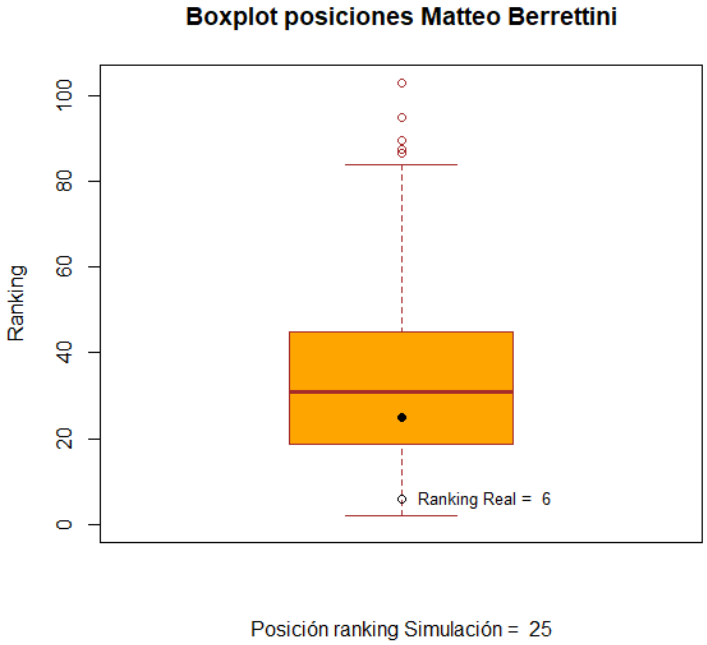


Vemos de forma aún más clara la entrada de Carlos entre los mejores, obteniendo la mayoría de los resultados entre el puesto 10 y el 30, por lo que el modelo fue capaz de predecir bien el rendimiento de Alcaraz en 2021, el cual tiene una mejora exponencial en su nivel tenístico. Viendo esta diferencia tan abismal se esperaría que siguiese así e incluso que tal vez no tenga techo, llegando a poder alcanzar el tan codiciado primer puesto en los próximos años. Sin ir más lejos, esta hipótesis queda confirmada, ya que el futuro y presente del tenis español se sitúa ni más ni menos que en el segundo puesto de la *race* del 2022, solo superado por la leyenda del tenis Rafael Nadal. Por lo que volvemos a encontrarnos en un caso bien predicho.

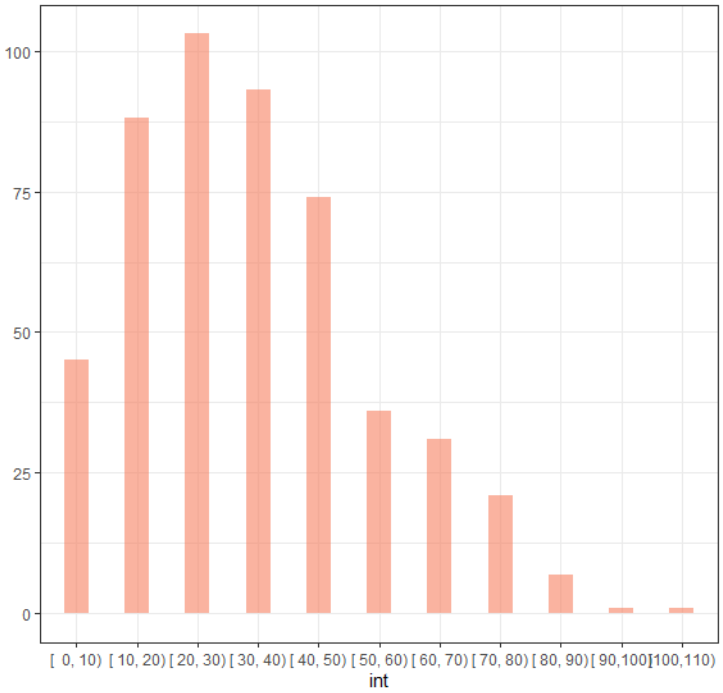
Matteo Berrettini

- Edad: 26 años
- Altura: 1.96 metros
- País: Italia
- *Ranking race* 2020: 10
- Pista favorita: Por victorias dura o hierba, aunque personalmente él dijo que arcilla

Esta vez nos encontramos con el italiano Matteo Berrettini. Un jugador con un saque potentísimo que entró por primera vez en el top 10 en el año 2019 y desde entonces se ha mantenido en esas posiciones. Este es un jugador el cual, como ya habíamos visto antes, quedó infravalorado por el modelo. Veamos más de cerca sus resultados.



Vemos como las simulaciones frecuentan el 50% entre el top 20 y 45 aproximadamente, lejos del resultado real como se puede observar.

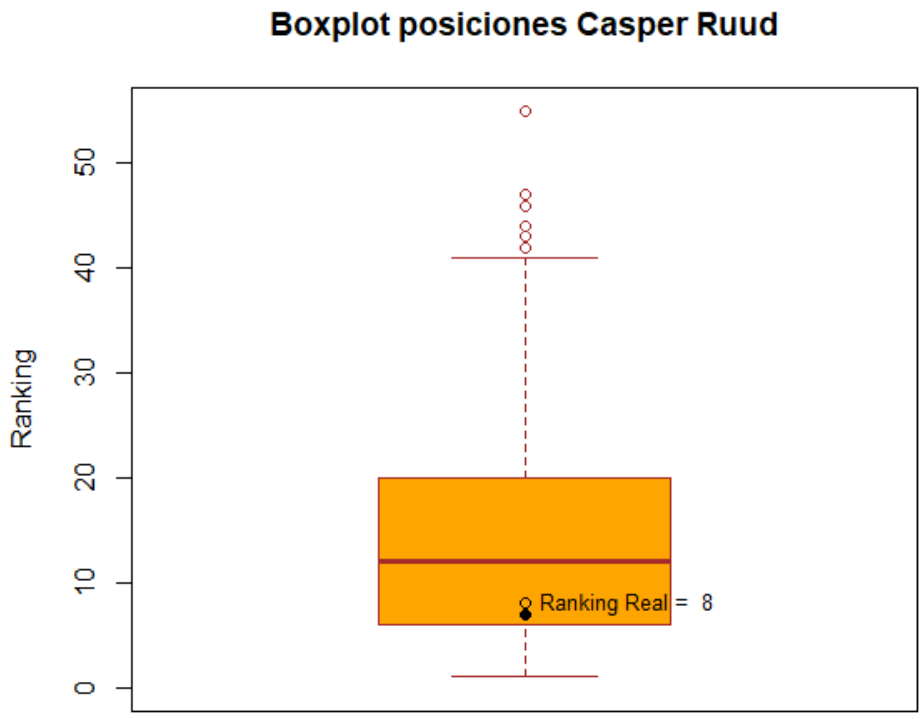


También vemos como el intervalo donde se sitúa el *ranking* real es el cuarto más frecuentado, lo cual nos indica que, en este caso particular, la simulación no se ha terminado de acercar a la realidad. De hecho, el más frecuente es donde se encuentra el resultado final, entre la posición 20 y 30. Viendo esto, mi predicción seria que entrase en el top 20 acercándose al 10. Actualmente, está el número 12 aún habiéndose perdido algún torneo lo que parece indicar que de nuevo podría volver a rendir a más nivel del esperado.

Casper Ruud

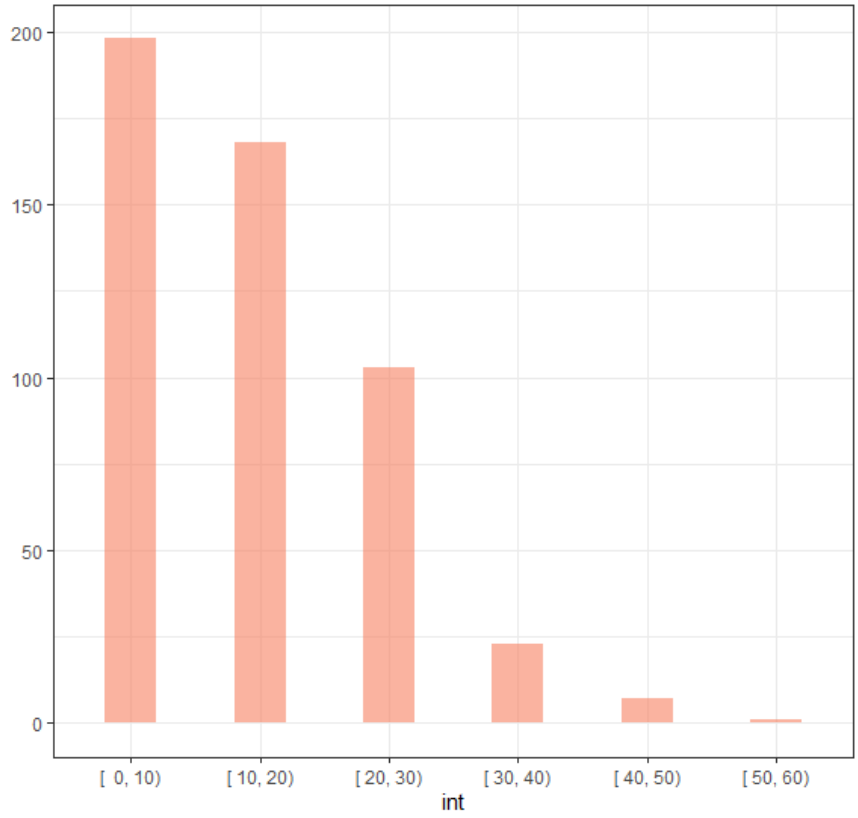
- Edad: 23 años
- Altura: 1.83 metros
- País: Noruega
- *Ranking race* 2020: 27
- Pista favorita: Arcilla

Encontramos aquí a un especialista en arcilla, el cual llegó a conseguir grandes resultados, incluso logrando un título en un ATP 250 y unas semifinales en los *Masters* 1000 de Roma. No obstante, un jugador el cual siempre se criticó por no saber jugar en el resto de las pistas.



Posición ranking Simulación = 7

Vemos como para la simulación, habiendo quedado en el puesto 27 la temporada anterior y con el bagaje que tiene en dura, lo sitúa como un fijo en el top 20 en el 75% de las ocasiones. Son pocos los casos que lo alejan de las posiciones de arriba y es por eso que encontramos un par de *outliers* situándolo fuera del top 40, por lo que se esperaría que subiese el *ranking*. En este caso particular, la exactitud de la predicción del modelo es altísima, equivocándose por tan solo una posición en la simulación que lo colocaba en el séptimo puesto.



Viendo la frecuencia de veces que aparece en cada intervalo vemos los datos de un jugador de arriba, de los grandes. Así es siendo el top 10 el lugar más frecuentado, seguido del 20 y el 30 en bastante menor cantidad. Acabó en el top 20 el 72% de las ocasiones y en el 10 el 40%, unos números muy buenos. Viendo que el modelo lo sitúa en los puestos de ATP *Finals*, la constancia en la que aparece en estos puestos y el contexto que tiene, mi predicción aquí sería que el noruego se mantendrá en el top 10, el top 15 cómo muy bajo. Es verdad que igual le falta algo todavía para entrar entre los cinco mejores, pero la subida de nivel que se esperaba y la que realmente ha logrado hace indicar que se mantendrá ahí al menos una temporada más.

Actualmente, se encuentra en el cuarto puesto del *ranking* habiendo alcanzado una final de un *Grand Slam* y de un *Masters* 1000 sumado de dos títulos en ATP 250. Aun así, todavía está por pasar la peor época para los especialistas en arcilla como él, ya que quedan los dos Grand Slams en dura, los cuales proporcionan muchos puntos al circuito, por lo que de momento va bien encaminada la predicción de este jugador.

También hemos visto a Opelka en el top 5, un jugador que recientemente alcanzó el top 20 por primera vez, y a Kevin Anderson, jugador que no está rindiendo y acabó muy mal la temporada pasada, en el 26. Esto se debe a la increíble altura que tienen ambos, más de dos metros para el sudafricano y 2’11 para el estadounidense. Parece ser que esto ha ayudado positivamente en el rendimiento esperado de estos jugadores y es que pocos en el circuito pueden acercarse a estas alturas.

CONCLUSIONES

Recordamos que el objetivo del trabajo era, realizar una simulación de un año del circuito ATP para así, comparándolo con los resultados observados, poder realizar predicciones de los diferentes jugadores que componen la asociación.

Con esto en mente y los resultados previamente expuestos sacamos las siguientes conclusiones:

- Es muy complicado conseguir predecir subidas de nivel de jugadores consolidados en el circuito. Estos serían jugadores que no destacan demasiado y que ya llevan algunos años obteniendo unos resultados que no resaltan mucho. Esto lo encontramos con los casos de Karatsev y Norrie previamente vistos.
- El modelo es bastante bueno prediciendo a las jóvenes promesas. Las promesas son eso, promesas, y no todas consiguen dar un salto de nivel en el circuito. Algunas de las marcadas en la simulación como Kecmanovic, Alcaraz o no tan jóvenes como Ruud ha podido predecir, o bien, la subida de nivel de ese año o, ayudado a predecir la de este.
- La simulación es capaz de acertar bien a los jugadores de arriba. Los llamados a dominar el circuito están bien colocados, 11 de 15 acertados (73%) en el top 15 y exactamente los cuatro mejores de la ATP que en la realida, resultados que abalan esta afirmación.

Hay cosas que fallan del modelo, una de ellas, como hemos comentado, es predecir el aumento de nivel de un jugador constante del circuito ya consolidado, ya que esto pasa de forma muy aleatoria o con variables que no se pueden cuantificar. Otra sería es la incapacidad de poder cuantificar mediante alguna variable las lesiones. Esto afecta de varias maneras, molestias en medio de un torneo, vuelta de una lesión muy fuerte de la cual todavía no se está recuperado, vuelta de una lesión no tan grave, etc. Poder recoger este dato sería muy interesante a la vez que clave para poder mejorar el modelo.

También se podría llegar a plantear un modelo donde se recojan variables como los *aces*, *winners*, errores no forzados... por punto jugado. No obstante, estos son datos que no están completos, ya que no en todos los partidos se recogen y además que la gran magnitud de variables explicativas que ya se tenían en cuenta hacen que sea imposible tantear esta posibilidad. También estaría interesante probar como funciona con modelos de Machine Learning

Por último, comentar, que es verdad que para poder predecir el rendimiento a futuro de un jugador no sirve con ver solo el resultado esperado y observado. Es estrictamente necesario encontrar el balance entre la estadística y su correcta contextualización, tanto del jugador como de los resultados obtenidos en la realidad, así mismo como los resultados más al detalle de la simulación.

BIBLIOGRAFIA

Barnett, O'Shaughnessy, & Bedford, (2011). Predicting a tennis match in progress for sports multimedia. https://www.researchgate.net/publication/220438547_Predicting_a_tennis_match_in_progress_for_sports_multimedia

Alexander De Seranno. (2020). Predicting Tennis Matches Using Machine Learning. https://libstore.ugent.be/fulltxt/RUG01/002/945/727/RUG01-002945727_2021_0001_AC.pdf

Gollub, Jacob. 2017. Producing Win Probabilities for Professional Tennis Matches from any Score. Bachelor's thesis, Harvard College. http://nrs.harvard.edu/urn-3:HUL.InstRepos:41024787

Lisi and Zanella (2017). Tennis betting: Can statistics beat bookmakers? https://www.researchgate.net/publication/310774506_Tennis_betting_Can_statistics_beat_bookmakers

Nicholas Devin. (2021). Predicting Winners of Professional Tennis Matches. https://nycdatascience.com/blog/student-works/predicting-winners-of-professional-tennis-matches/

Tristan J. Barnett and Stephen R. Clarke. (2002). Using Microsoft Excel to model a tennis match. https://www.researchgate.net/publication/228562992_Using_Microsoft_Excel_to_model_a_tennis_match

Webs consultadas

Canal tenis:

- Repartimiento de puntos. https://canaltenis.com/puntos-reparte-torneos-atp/

GitHub:

- Base de datos de partidos de Tenis de JeffSackmann. https://github.com/JeffSackmann/tennis_atp

Punto de Break:

- Noticia Fognini sobre la altura e influencia en el juego. https://www.puntodebreak.com/2022/02/11/fognini-retire-no-ver-tenis-no-gusta-viene

Research Gate

- Investigaciones https://www.researchgate.net

Web oficial ATP:

- *Ranking* de la *race* del 2021. https://www.atptour.com/en/rankings/singles

Wikipedia:

- Cuadros ATP 2021. https://es.wikipedia.org/wiki/Anexo:Torneos_ATP_en_2021
- Modelo de regresión logística. https://es.wikipedia.org/wiki/Regresi%C3%B3n_log%C3%ADstica
- Tenis https://es.wikipedia.org/wiki/Tenis#Saque
- Asociación de Tenistas Profesionales https://es.wikipedia.org/wiki/Asociación_de_Tenistas_Profesionales

También se ha consultado el material proporcionado en la carrera de estadística sobre la teoría de los conceptos aprendidos en diferentes asignaturas cursadas por parte de la UB y la UPC en el grado de Estadística.

ANEXO

Base de datos

```
load("E:/TENIS 12-2021/backfile1.RData")

dd$tourney_date <- as.Date(as.character(dd$tourney_date), "%Y%m%
d")
dd<-dd[order(dd$tourney_date, dd$match_num),]

dd$tourney_level[dd$tourney_level=="S"] <- "ITF"
dd$tourney_level[dd$tourney_level=="15"] <- "ITF"
dd$tourney_level[dd$tourney_level=="25"] <- "ITF"
dd <- dd%>% filter(tourney_date < as.Date("2021-01-01", "%Y-%m-%
d") & tourney_date >= as.Date("2010-01-01", "%Y-%m-%d"))
keep <- c("surface", "tourney_level", "tourney_date", "winner_na
me", "loser_name",
          "player_1_id",  "player_2_id", "y", "rank_p2", "rank_p
1", "age_p2", "age_p1", "winner_ht","loser_ht")
dd <- dd[,keep]
```

508597 partidos entre el 2000 y el 2020, 287333 a partir del 2010

Las funciones de abajo del bucle se incorporan desde un RData

Selección partidos ATP

```
j <- c()
start <- which(dd$tourney_date > as.Date("2016-01-01", "%Y-%m-%d
"))[1]
for(i in start:nrow(dd)){
  if(dd[i, "tourney_level"] == "A" | dd[i, "tourney_level"] == "
G" | dd[i, "tourney_level"] == "M"){
      j <- c(j,i)
  }
}
k = 1

while(k <= length(j)){
  i <- j[k]
  a <- dd[i,"player_1_id"]
  b <- dd[i,"player_2_id"]

  sur <- dd[1:(i-1),] %>% filter(surface == dd[i, "surface"])
  tour <- dd[1:(i-1),] %>% filter(tourney_level == dd[i, "tourne
y_level"])
  surtour <- sur %>% filter(tourney_level == dd[i, "tourney_leve
l"])

  head_to_head <- dd[(dd[1:(i-1),"player_1_id"] == a | dd[1:(i-1
),"player_1_id"] == b) &
```

```
                                  (dd[1:(i-1),"player_2_id"]== a |
dd[1:(i-1),"player_2_id"] == b),]

  ##############
  ### WIN % ###
  ##############

  # Tour
  dd[i,"tour_2_p1"] <- tourP(a,2);dd[i,"tour_2_p2"] <- tourP(b,2
)
  dd[i,"tour_4_p1"] <- tourP(a,4);dd[i,"tour_4_p2"] <- tourP(b,4
)
  dd[i,"tour_6_p1"] <- tourP(a,6);dd[i,"tour_6_p2"] <- tourP(b,6
)


  # Surface
  dd[i,"sur_2_p1"] <- surP(a,2);dd[i,"sur_2_p2"] <- surP(b,2)
  dd[i,"sur_4_p1"] <- surP(a,4);dd[i,"sur_4_p2"] <- surP(b,4)
  dd[i,"sur_6_p1"] <- surP(a,6);dd[i,"sur_6_p2"] <- surP(b,6)


  # Surface vs Tour
  dd[i,"surtour_2_p1"] <- surtourP(a,2);dd[i,"surtour_2_p2"] <-
surtourP(b,2)
  dd[i,"surtour_4_p1"] <- surtourP(a,4);dd[i,"surtour_4_p2"] <-
surtourP(b,4)
  dd[i,"surtour_6_p1"] <- surtourP(a,6);dd[i,"surtour_6_p2"] <-
surtourP(b,6)


  ###############
  ### HEIGHT ###
  ###############

  if(dd[i,"winner_name"] == dd[i,"player_1_id"]){
    dd[i,"ht_p1"] <- trunc(dd[i,"winner_ht"]);dd[i,"ht_p2"] <- t
runc(dd[i,"loser_ht"])
  } else {
    dd[i,"ht_p2"] <- trunc(dd[i,"winner_ht"]);dd[i,"ht_p1"] <- t
runc(dd[i,"loser_ht"])
  }

  ####################
  ### HEAD TO HEAD ###
  ####################
  tipopista<-dd[i,"surface"]
  dd[i, "h2h_6_p1"] <- h2hf(a,b,6,0)[1];dd[i, "h2h_6_p2"] <- h2h
f(a,b,6,0)[2]
  dd[i, "h2h_3_p1"] <- h2hf(a,b,3,0)[1];dd[i, "h2h_3_p2"] <- h2h
```

```
f(a,b,3,0)[2]
  dd[i, "h2h_1_p1"] <- h2hf(a,b,1,0)[1];dd[i, "h2h_1_p2"] <- h2h
f(a,b,1,0)[2]

  dd[i, "h2h_6_sur_p1"] <- h2hf(a,b,6,tipopista)[1];dd[i, "h2h_6
_sur_p2"] <- h2hf(a,6,0,tipopista)[2]
  dd[i, "h2h_3_sur_p1"] <- h2hf(a,b,3,tipopista)[1];dd[i, "h2h_3
_sur_p2"] <- h2hf(a,b,3,tipopista)[2]
  dd[i, "h2h_1_sur_p1"] <- h2hf(a,b,1,tipopista)[1];dd[i, "h2h_1
_sur_p2"] <- h2hf(a,b,1,tipopista)[2]


  ####################
  ### RECENT FORM ###
  ####################

  dd[i,"rf_6_12_p1"] <- rf(a,6,12);dd[i,"rf_6_12_p2"] <- rf(b,6,
12)
  dd[i,"rf_6_6_p1"] <- rf(a,6,6);dd[i,"rf_6_6_p2"] <- rf(b,6,6)
  dd[i,"rf_6_3_p1"] <- rf(a,6,3);dd[i,"rf_6_3_p2"] <- rf(b,6,3)
  dd[i,"rf_6_1_p1"] <- rf(a,6,1);dd[i,"rf_6_1_p2"] <- rf(b,6,1)

  dd[i,"rf_4_12_p1"] <- rf(a,4,12);dd[i,"rf_4_12_p2"] <- rf(b,4,
12)
  dd[i,"rf_4_6_p1"] <- rf(a,4,6);dd[i,"rf_4_6_p2"] <- rf(b,4,6)
  dd[i,"rf_4_3_p1"] <- rf(a,4,3);dd[i,"rf_4_3_p2"] <- rf(b,4,3)
  dd[i,"rf_4_1_p1"] <- rf(a,4,1);dd[i,"rf_4_1_p2"] <- rf(b,4,1)

  dd[i,"rf_2_12_p1"] <- rf(a,2,12);dd[i,"rf_2_12_p2"] <- rf(b,2,
12)
  dd[i,"rf_2_6_p1"] <- rf(a,2,6);dd[i,"rf_2_6_p2"] <- rf(b,2,6)
  dd[i,"rf_2_3_p1"] <- rf(a,2,3);dd[i,"rf_2_3_p2"] <- rf(b,2,3)
  dd[i,"rf_2_1_p1"] <- rf(a,2,1);dd[i,"rf_2_1_p2"] <- rf(b,2,1)


  ###################
  ### CONTADORES ###
  ###################

  # 6 años
  dd6 <- dd[1:(i-1),]%>%filter(tourney_date>= (dd[i,"tourney_dat
e"]-12*30*6))
  dd[i,"contador6_p1"] <- length(dd6$surface[(dd6$player_1_id==a
|dd6$player_2_id==a)
                                             & dd6$tourney_date <
dd[i,"tourney_date"]])
  dd[i,"contador6_p2"] <- length(dd6$surface[(dd6$player_1_id==b
|dd6$player_2_id==b)
                                             & dd6$tourney_date <
dd[i,"tourney_date"]])
```

```
  dd[i,"contador6_surtur_p1"] <- length(dd6$surface[(dd6$player_
1_id==a|dd6$player_2_id==a) & dd6$surface==dd[i,"surface"]
                                                    & dd6$tourney_
date < dd[i,"tourney_date"]
                                                    & dd6$tourney_
level==dd[i,"tourney_level"]])
  dd[i,"contador6_surtur_p2"] <- length(dd6$surface[(dd6$player_
1_id==b|dd6$player_2_id==b) & dd6$surface==dd[i,"surface"]
                                                    & dd6$tourney_
date < dd[i,"tourney_date"]
                                                    & dd6$tourney_
level==dd[i,"tourney_level"]])
  # 4 años
  dd3 <- dd[1:(i-1),]%>%filter(tourney_date>= (dd[i,"tourney_dat
e"]-12*30*4))
  dd[i,"contador4_p1"] <- length(dd3$surface[(dd3$player_1_id==a
|dd3$player_2_id==a)
                                             & dd3$tourney_date <
dd[i,"tourney_date"]])
  dd[i,"contador4_p2"] <- length(dd3$surface[(dd3$player_1_id==b
|dd3$player_2_id==b)
                                             & dd3$tourney_date <
dd[i,"tourney_date"]])

  dd[i,"contador4_surtur_p1"] <- length(dd3$surface[(dd3$player_
1_id==a|dd3$player_2_id==a) & dd3$surface==dd[i,"surface"]
                                                    & dd3$tourney_
date < dd[i,"tourney_date"]
                                                    & dd3$tourney_
level==dd[i,"tourney_level"]])
  dd[i,"contador4_surtur_p2"] <- length(dd3$surface[(dd3$player_
1_id==b|dd3$player_2_id==b) & dd3$surface==dd[i,"surface"]
                                                    & dd3$tourney_
date < dd[i,"tourney_date"]
                                                    & dd3$tourney_
level==dd[i,"tourney_level"]])

  # 2 años
  dd3 <- dd[1:(i-1),]%>%filter(tourney_date>= (dd[i,"tourney_dat
e"]-12*30*2))
  dd[i,"contador2_p1"] <- length(dd3$surface[(dd3$player_1_id==a
|dd3$player_2_id==a)
                                             & dd3$tourney_date <
dd[i,"tourney_date"]])
  dd[i,"contador2_p2"] <- length(dd3$surface[(dd3$player_1_id==b
|dd3$player_2_id==b)
                                             & dd3$tourney_date <
dd[i,"tourney_date"]])
```

```
  dd[i,"contador2_surtur_p1"] <- length(dd3$surface[(dd3$player_
1_id==a|dd3$player_2_id==a) & dd3$surface==dd[i,"surface"]
                                                    & dd3$tourney_
date < dd[i,"tourney_date"]
                                                    & dd3$tourney_
level==dd[i,"tourney_level"]])
  dd[i,"contador2_surtur_p2"] <- length(dd3$surface[(dd3$player_
1_id==b|dd3$player_2_id==b) & dd3$surface==dd[i,"surface"]
                                                    & dd3$tourney_
date < dd[i,"tourney_date"]
                                                    & dd3$tourney_
level==dd[i,"tourney_level"]])


  if(dd[i,"winner_name"] == dd[i,"player_1_id"]){
    dd[i,"y"] <- 1
  }else{ dd[i,"y"] <- 0 }

  k=k+1
}
```

Archivo partidos y modelo

```
write.csv(dd, file = "E:/TFG/partidos.csv")

train <- dd %>% drop_na(tour_2_p1)
write.csv(train, file = "E:/TFG/train.csv")
```

Completar Height

```
dd[dd$player_1_id == "J J Wolf","ht_p1"] <- 183; dd[dd$player_2_
id == "J J Wolf","ht_p2"] <- 183
dd[dd$player_1_id == "Alejandro Tabilo","ht_p1"] <- 188; dd[dd$p
layer_2_id == "Alejandro Tabilo","ht_p2"] <- 188
dd[dd$player_1_id == "Bjorn Fratangelo","ht_p1"] <- 183; dd[dd$p
layer_2_id == "Bjorn Fratangelo","ht_p2"] <- 183
dd[dd$player_1_id == "Ernesto Escobedo","ht_p1"] <- 185; dd[dd$p
layer_2_id == "Ernesto Escobedo","ht_p2"] <- 185
dd[dd$player_1_id == "Thomas Fabbiano","ht_p1"] <- 173; dd[dd$pl
ayer_2_id == "Thomas Fabbiano","ht_p2"] <- 173
dd[dd$player_1_id == "Maximilian Marterer","ht_p1"] <- 191; dd[d
d$player_2_id == "Maximilian Marterer","ht_p2"] <- 191
dd[dd$player_1_id == "Juan Ignacio Londero","ht_p1"] <- 180; dd[
dd$player_2_id == "Juan Ignacio Londero","ht_p2"] <- 180
dd[dd$player_1_id == "Jared Donaldson","ht_p1"] <- 188; dd[dd$pl
ayer_2_id == "Jared Donaldson","ht_p2"] <- 188
dd[dd$player_1_id == "Adrian Menendez Maceiras","ht_p1"] <- 183;
dd[dd$player_2_id == "Adrian Menendez Maceiras","ht_p2"] <- 183
dd[dd$player_1_id == "Jason Jung","ht_p1"] <- 178; dd[dd$player_
2_id == "Jason Jung","ht_p2"] <- 178
dd[dd$player_1_id == "Mackenzie Mcdonald","ht_p1"] <- 178; dd[dd
```

```
$player_2_id == "Mackenzie Mcdonald","ht_p2"] <- 178
dd[dd$player_1_id == "Nicolas Kicker","ht_p1"] <- 178; dd[dd$pla
yer_2_id == "Nicolas Kicker","ht_p2"] <- 178

dd[dd$player_1_id == "Marco Trungelliti","ht_p1"] <- 178; dd[dd$
player_2_id == "Marco Trungelliti","ht_p2"] <- 178
dd[dd$player_1_id == "Noah Rubin","ht_p1"] <- 175; dd[dd$player_
2_id == "Noah Rubin","ht_p2"] <- 175
dd[dd$player_1_id == "Pedro Martinez","ht_p1"] <- 185; dd[dd$pla
yer_2_id == "Pedro Martinez","ht_p2"] <- 185
dd[dd$player_1_id == "Alex Bolt","ht_p1"] <- 183; dd[dd$player_2
_id == "Alex Bolt","ht_p2"] <- 183
dd[dd$player_1_id == "Stefan Kozlov","ht_p1"] <- 183; dd[dd$play
er_2_id == "Stefan Kozlov","ht_p2"] <- 183
```

Detectar jugadores que vale la pena meter altura (código ejemplo, hacer con contadores mejor)

```
pr <-dd%>% filter(tourney_date >= as.Date("2016-01-01", "%Y-%m-%
d"))
pr <- sqldf::sqldf("SELECT winner_name, winner_ht, count(*) AS n
             FROM pr
             WHERE tourney_level = 'A'
             GROUP BY winner_name
             HAVING n > 10")
pr <- pr[is.na(pr$winner_ht),]
```

Modelos

```
set.seed(481078860)
train <- read.csv("E:/TFG/train2.csv");train$y <- as.factor(trai
n$y)

train <- train %>% drop_na(sur_2_p1,sur_2_p2)
train$surface <- as.factor(train$surface)
train$age_p1 <- as.numeric(train$age_p1);train$age_p2 <- as.nume
ric(train$age_p2)
train$ht_p1 <- as.numeric(train$ht_p1);train$ht_p2 <- as.numeric
(train$ht_p2)

#6 años
train6 <- train[train$contador6_p1 >= 50,];train6 <- train6[trai
n6$contador6_p2 >= 50,]
train6 <- train6[train6$contador6_surtur_p1 >= 10,];train6 <- tr
ain6[train6$contador6_surtur_p2 >= 10,]

#4 años
train4 <- train[train$contador4_p1 >= 50,];train4 <- train4[trai
n4$contador4_p2 >= 50,]
train4 <- train4[train4$contador4_surtur_p1 >= 10,];train4 <- tr
ain4[train4$contador4_surtur_p2 >= 10,]

#2 años
train2 <- train[train$contador2_p1 >= 50,];train2 <- train2[trai
n2$contador2_p2 >= 50,]
train2 <- train2[train2$contador2_surtur_p1 >= 10,];train2 <- tr
ain2[train2$contador2_surtur_p2 >= 10,]
```

Modelización

```
predictive_metrics <- function(dd){

dd[,"age_dif"] <- dd[,"age_p1"]-dd[,"age_p2"]
dd[,"ht_dif"] <- dd[,"ht_p1"]-dd[,"ht_p2"]

##################################################################
###############
dd[,"sur_2_dif"] <- dd[,"sur_2_p1"]-dd[,"sur_2_p2"]
dd[,"sur_4_dif"] <- dd[,"sur_4_p1"]-dd[,"sur_4_p2"]
dd[,"sur_6_dif"] <- dd[,"sur_6_p1"]-dd[,"sur_6_p2"]

dd[,"tour_2_dif"] <- dd[,"tour_2_p1"]-dd[,"tour_2_p2"]
dd[,"tour_4_dif"] <- dd[,"tour_4_p1"]-dd[,"tour_4_p2"]
dd[,"tour_6_dif"] <- dd[,"tour_6_p1"]-dd[,"tour_6_p2"]

dd[,"surtour_2_dif"] <- dd[,"surtour_2_p1"]-dd[,"surtour_2_p2"]
dd[,"surtour_4_dif"] <- dd[,"surtour_4_p1"]-dd[,"surtour_4_p2"]
```

```
dd[,"surtour_6_dif"] <- dd[,"surtour_6_p1"]-dd[,"surtour_6_p2"]

#################################################################
###############
dd[,"h2h_6_dif"] <- dd[,"h2h_6_p1"]-dd[,"h2h_6_p2"]
dd[,"h2h_3_dif"] <- dd[,"h2h_3_p1"]-dd[,"h2h_3_p2"]
dd[,"h2h_1_dif"] <- dd[,"h2h_1_p1"]-dd[,"h2h_1_p2"]

dd[,"h2h_6_sur_dif"] <- dd[,"h2h_6_sur_p1"]-dd[,"h2h_6_sur_p2"]
dd[,"h2h_3_sur_dif"] <- dd[,"h2h_3_sur_p1"]-dd[,"h2h_3_sur_p2"]
dd[,"h2h_1_sur_dif"] <- dd[,"h2h_1_sur_p1"]-dd[,"h2h_1_sur_p2"]

#################################################################
###############


dd[,"rf_6_12_dif"] <- dd[,"rf_6_12_p1"]-dd[,"rf_6_12_p2"]
dd[,"rf_6_6_dif"] <- dd[,"rf_6_6_p1"]-dd[,"rf_6_6_p2"]
dd[,"rf_6_3_dif"] <- dd[,"rf_6_3_p1"]-dd[,"rf_6_3_p2"]
dd[,"rf_6_1_dif"] <- dd[,"rf_6_1_p1"]-dd[,"rf_6_1_p2"]

dd[,"rf_4_12_dif"] <- dd[,"rf_4_12_p1"]-dd[,"rf_4_12_p2"]
dd[,"rf_4_6_dif"] <- dd[,"rf_4_6_p1"]-dd[,"rf_4_6_p2"]
dd[,"rf_4_3_dif"] <- dd[,"rf_4_3_p1"]-dd[,"rf_4_3_p2"]
dd[,"rf_4_1_dif"] <- dd[,"rf_4_1_p1"]-dd[,"rf_4_1_p2"]

dd[,"rf_2_12_dif"] <- dd[,"rf_2_12_p1"]-dd[,"rf_2_12_p2"]
dd[,"rf_2_6_dif"] <- dd[,"rf_2_6_p1"]-dd[,"rf_2_6_p2"]
dd[,"rf_2_3_dif"] <- dd[,"rf_2_3_p1"]-dd[,"rf_2_3_p2"]
dd[,"rf_2_1_dif"] <- dd[,"rf_2_1_p1"]-dd[,"rf_2_1_p2"]


keep2 <- c("y","age_dif","ht_dif","sur_6_dif","sur_4_dif","sur_2
_dif",
           "tour_6_dif","tour_4_dif","tour_2_dif", "surtour_6_di
f", "surtour_4_dif", "surtour_2_dif",
           "h2h_6_dif", "h2h_3_dif", "h2h_1_dif", "h2h_6_sur_dif
", "h2h_3_sur_dif", "h2h_1_sur_dif",
           "rf_6_12_dif", "rf_6_6_dif", "rf_6_3_dif","rf_6_1_dif
",
           "rf_4_12_dif", "rf_4_6_dif", "rf_4_3_dif","rf_4_1_dif
",
           "rf_2_12_dif", "rf_2_6_dif", "rf_2_3_dif","rf_2_1_dif
", "surface")

dd <- dd[,keep2]
return(dd)
}
```

Descriptiva (Bivariante)

```
pred6 <- predictive_metrics(train6)
train6 <- predictive_metrics(train6)
vars=colnames(train6)[c(4:12)]
# windows(10,10)
par(mfrow=c(3,3))
for (va in vars){
  if (!is.factor(train6[,va])){
    boxplot(as.formula(paste0(va,"~y")),train6,main=va,col=c(2,3
),horizontal=T)
  } else{
    barplot(prop.table(table(train6$y, train6[,va]),2),main=va,c
ol=c(2,3))
  }
}

vars=colnames(train6)[c(2)]
# windows(10,10)
par(mfrow=c(1,1))
for (va in vars){
  if (!is.factor(train6[,va])){
    boxplot(as.formula(paste0(va,"~y")),train6,main=va,col=c(2,3
),horizontal=T)
  } else{
    barplot(prop.table(table(train6$y, train6[,va]),2),main=va,c
ol=c(2,3))
  }
}
train6 <- train6 %>% drop_na(y,ht_dif)
ggplot(train6, aes(x = surface, y = ht_dif)) +
   geom_bar(
       aes(color = y, fill = y),
         stat = "identity", position = position_dodge(0.8),
         width = 0.7
     )

vars=colnames(train6)[c(13:18)]
# windows(10,10)
par(mfrow=c(2,3))
for (va in vars){
  if (!is.factor(train6[,va])){
    boxplot(as.formula(paste0(va,"~y")),train6,main=va,col=c(2,3
),horizontal=T)
  } else{
    barplot(prop.table(table(train6$y, train6[,va]),2),main=va,c
ol=c(2,3))
  }
}

vars=colnames(train6)[c(19:22)]
# windows(10,10)
```

```
par(mfrow=c(2,2))
for (va in vars){
  if (!is.factor(train6[,va])){
    boxplot(as.formula(paste0(va,"~y")),train6,main=va,col=c(2,3
),horizontal=T)
  } else{
    barplot(prop.table(table(train6$y, train6[,va]),2),main=va,c
ol=c(2,3))
  }
}
vars=colnames(train6)[c(23:26)]
# windows(10,10)
par(mfrow=c(2,2))
for (va in vars){
  if (!is.factor(train6[,va])){
    boxplot(as.formula(paste0(va,"~y")),train6,main=va,col=c(2,3
),horizontal=T)
  } else{
    barplot(prop.table(table(train6$y, train6[,va]),2),main=va,c
ol=c(2,3))
  }
}
vars=colnames(train6)[c(27:30)]
# windows(10,10)
par(mfrow=c(2,2))
for (va in vars){
  if (!is.factor(train6[,va])){
    boxplot(as.formula(paste0(va,"~y")),train6,main=va,col=c(2,3
),horizontal=T)
  } else{
    barplot(prop.table(table(train6$y, train6[,va]),2),main=va,c
ol=c(2,3))
  }
}
```

Modelos 6 años

```
#pred6 <- predictive_metrics(train6)
pred6 <- pred6 %>% drop_na(sur_6_dif,tour_6_dif,surtour_6_dif,ag
e_dif,
                           h2h_6_dif , h2h_3_dif , h2h_1_dif , h
2h_6_sur_dif , h2h_3_sur_dif , h2h_1_sur_dif ,
                           ht_dif,rf_6_12_dif , rf_6_6_dif , rf_
6_3_dif , rf_6_1_dif)
m6a <- glm(y~ 0 + sur_6_dif + tour_6_dif + age_dif + ht_dif:surf
ace +
               h2h_6_dif + h2h_3_dif + h2h_1_dif + h2h_6_sur_di
f + h2h_3_sur_dif + h2h_1_sur_dif +
               rf_6_12_dif + rf_6_6_dif + rf_6_3_dif + rf_6_1_d
if,
```

```
              data=pred6,family = binomial(link = "logit"))
m6a <- step(m6a)

m6b <- glm(y~ 0 + surtour_6_dif + age_dif + ht_dif:surface +
                 h2h_6_dif + h2h_3_dif + h2h_1_dif + h2h_6_sur_di
f + h2h_3_sur_dif + h2h_1_sur_dif +
                 rf_6_12_dif + rf_6_6_dif + rf_6_3_dif + rf_6_1_d
if,
              data=pred6,family = binomial(link = "logit"))
m6b <- step(m6b)
```

Modelos 4 años

```
pred4 <- predictive_metrics(train4)
pred4 <- pred4 %>% drop_na(sur_4_dif , tour_4_dif , age_dif , ht
_dif ,
                 h2h_6_dif , h2h_3_dif , h2h_1_dif , h2h_6_sur_di
f , h2h_3_sur_dif , h2h_1_sur_dif ,
                 rf_4_12_dif , rf_4_6_dif , rf_4_3_dif , rf_4_1_d
if)
m4a <- glm(y~ 0 + sur_4_dif + tour_4_dif + age_dif + ht_dif:surf
ace +
                 h2h_6_dif + h2h_3_dif + h2h_1_dif + h2h_6_sur_di
f + h2h_3_sur_dif + h2h_1_sur_dif +
                 rf_4_12_dif + rf_4_6_dif + rf_4_3_dif + rf_4_1_d
if,
              data=pred6,family = binomial(link = "logit"))
#m4a <- step(m4a, direction="forward",k=log(nrow(pred4)),trace=F
ALSE)
m4a <- step(m4a)

m4b <- glm(y~ 0 + surtour_4_dif + age_dif + ht_dif:surface +
                 h2h_6_dif + h2h_3_dif + h2h_1_dif + h2h_6_sur_di
f + h2h_3_sur_dif + h2h_1_sur_dif +
                 rf_4_12_dif + rf_4_6_dif + rf_4_3_dif + rf_4_1_d
if,
              data=pred4,family = binomial(link = "logit"))
m4b <- step(m4b)
```

Modelos 2 años

```
pred2 <- predictive_metrics(train2)
pred2 <- pred2 %>% drop_na(sur_2_dif , tour_2_dif , age_dif , ht
_dif ,
                 h2h_6_dif , h2h_3_dif , h2h_1_dif , h2h_6_sur_di
f , h2h_3_sur_dif , h2h_1_sur_dif ,
                 rf_2_12_dif , rf_2_6_dif , rf_2_3_dif , rf_2_1_d
if)
m2a <- glm(y~ 0 + sur_2_dif + tour_2_dif + age_dif + ht_dif:surf
ace +
                 h2h_6_dif + h2h_3_dif + h2h_1_dif + h2h_6_sur_di
```

```
f + h2h_3_sur_dif + h2h_1_sur_dif +
                rf_2_12_dif + rf_2_6_dif + rf_2_3_dif + rf_2_1_d
if,
             data=pred2,family = binomial(link = "logit"))
m2a <- step(m2a)

m2b <- glm(y~ 0 + surtour_2_dif + age_dif + ht_dif:surface +
                h2h_6_dif + h2h_3_dif + h2h_1_dif + h2h_6_sur_di
f + h2h_3_sur_dif + h2h_1_sur_dif +
                rf_2_12_dif + rf_2_6_dif + rf_2_3_dif + rf_2_1_d
if,
             data=pred2,family = binomial(link = "logit"))
m2b <- step(m2b)

AIC(m6a,m6b,m4a,m4b,m2a,m2b)
BIC(m6a,m6b,m4a,m4b,m2a,m2b)
summary(m4b)

residualPlot(m4b)

pred <- predict(m4b,type="response")
good <- 0;bad<-0
for(i in 1:nrow(pred4)){
  if(pred[i]<0.5 & pred4[i,"y"] == 0){
    good <- good+1
  } else if(pred[i]>=0.5 & pred4[i,"y"] == 1){
    good <- good+1
  } else {
    bad <- bad+1
  }
}
#good/(good+bad)

pr <- predict(m4b,type="response")
dadesroc <- ROCR::prediction(pr,pred4$y)
roc.perf <- ROCR::performance(dadesroc,"auc",fpr.stop=0.05)
plot(ROCR::performance(dadesroc,"tpr","fpr"))
roce <- round(AUC::auc(AUC::roc(pr,factor(pred4$y))),2)

#plot(allEffects(m2a),ask=F)
```

Eliminación Obs Ifluents

```
plot(cooks.distance(m4b))
abline(h=(4/nrow(pred4)), col="Red")
inf <- as.vector(which(cooks.distance(m4b) > (4/nrow(pred4))))
outlierTest(m4b)

pred4 <- pred4[-inf,]

m4b <- glm(y~ 0 + surtour_4_dif + age_dif+h2h_6_dif + h2h_3_dif+
```

```
rf_4_12_dif + rf_4_3_dif+ ht_dif:surface,
             data=pred4,family = binomial(link = "logit"))

influenceIndexPlot(m4b,id=list(lab=row.names(pred4),vars=c("Cook
", "Student","hat"), n=15))

pr <- predict(m4b,type="response")
dadesroc <- ROCR::prediction(pr,pred4$y)
roc.perf <- ROCR::performance(dadesroc,"auc",fpr.stop=0.05)
plot(ROCR::performance(dadesroc,"tpr","fpr"))
roce <- round(AUC::auc(AUC::roc(pr,factor(pred4$y))),2)

df <- data.frame(obs=factor(pred4$y),pre=1-fitted(m4b))
calPlotData <- caret::calibration(obs ~ pre, data = df)
library(caret)
xyplot(calPlotData, main = "calibrationPlot")
```

Cuadros ATP

```
library(tidyverse)
dd <- read.csv("https://raw.githubusercontent.com/JeffSackmann/tennis_atp/master/atp_matches_2021.csv")

dd$tourney_date <- as.Date(as.character(dd$tourney_date), "%Y%m%d")
dd<-dd[order(dd$tourney_date),]

eliminate <- c()
for(i in 1:nrow(dd)){
  if(dd[i,"tourney_name"]=="Atp Cup" | substr(dd[i,"tourney_name"],1,9) == "Davis Cup" | dd[i,"tourney_name"]=="Tokyo Olympics" |
     dd[i,"tourney_name"]=="NextGen Finals" | dd[i,"tourney_name"]=="Tour Finals") {
    eliminate <- c(eliminate,i)
  }
}
dd <- dd[-eliminate,]

partidos <- data.frame("jugadores" = c(), "age" = c(),"ht" = c(),"torneo" = c(), "surface" = c(), "tourney_date" = c(), "tourney_level" = c())
torneo = 0
tipo <- c(250, 250, 250, 250, "G", 250, 250, 250, 500, 250, 250, 250, 250, 500, 500, "MG", 250, 250, "MP", 500, 250, 250, 250, "MP", "MP", 250, 250, 250, 250, "G", 250, 500, 500, 250, 250, "G", 500, 250, 250, 250, 250, 250, 250, 250, 500, "MP", "MP", 250, "G", 250, 250, 250, 250, "MG", 250, 250, 500, 250, "MP", 250)

while(nrow(dd) > 0){
k=2
while(dd[1,"tourney_name"] == dd[k,"tourney_name"] & k <= nrow(dd)){
  end <- k
  k=k+1
}

d <- dd[1:end,]

r <- unique(d$round)
ce <- c();cf <- c()
edade <- c(); edadf <- c()
hte <- c(); htf <- c()
for(i in 1:nrow(d[d$round==r[1],])){
  ce <- c(ce,d[i,"winner_name"],d[i,"loser_name"])
  edade <- c(edade, trunc(d[i,"winner_age"]),trunc(d[i,"loser_age"]))
```

```
  hte <- c(hte, d[i,"winner_ht"],d[i,"loser_ht"])
}


for(k in (nrow(d[d$round==r[1],])+1) :(nrow(d[d$round==r[1],])+n
row(d[d$round==r[2],]))){
  if(is.na(match(d[k,"winner_name"],ce)) == T & is.na(match(d[k,
"loser_name"],ce)) == F){

      cf <- c(cf,d[k,"winner_name"],"BYE")
      cf <- c(cf,ce[1],ce[2])
      ce <- ce[-c(1,2)]


      edadf <- c(edadf,trunc(d[k,"winner_age"]),"BYE")
      edadf <- c(edadf,edade[1],edade[2])
      edade <- edade[-c(1,2)]

      htf <- c(htf,d[k,"winner_ht"],"BYE")
      htf <- c(htf,hte[1],hte[2])
      hte <- hte[-c(1,2)]

  } else if(is.na(match(d[k,"loser_name"],ce)) == T & is.na(matc
h(d[k,"winner_name"],ce)) == F){

      cf <- c(cf,d[k,"loser_name"],"BYE")
      cf <- c(cf,ce[c(1:2)])
      ce <- ce[-c(1:2)]

      edadf <- c(edadf,trunc(d[k,"loser_age"]),"BYE")
      edadf <- c(edadf,edade[1],edade[2])
      edade <- edade[-c(1,2)]

      htf <- c(htf,d[k,"loser_ht"],"BYE")
      htf <- c(htf,hte[1],hte[2])
      hte <- hte[-c(1,2)]


  } else if(is.na(match(d[k,"loser_name"],ce)) == T & is.na(matc
h(d[k,"winner_name"],ce)) == T){
    cf <- c(cf,d[k,"winner_name"],"BYE",d[k,"loser_name"],"BYE")
    edadf <- c(edadf,trunc(d[k,"winner_age"]),"BYE",trunc(d[k,"l
oser_age"]),"BYE")
    htf <- c(htf,d[k,"winner_ht"],"BYE",d[k,"loser_ht"],"BYE")

  } else {
    cf <- c(cf,ce[c(1:4)])
    ce <- ce[-c(1:4)]

    edadf <- c(edadf,edade[c(1:4)])
    edade <- edade[-c(1:4)]
```

```
    htf <- c(htf,hte[c(1:4)])
    hte <- hte[-c(1:4)]
  }

}
torneo = torneo + 1
partidos <- rbind(partidos,
                  data.frame(cf,
                             edadf,
                             htf,
                             rep(torneo,length(cf)),
                             rep(d[k,"surface"],length(cf)),
                             rep(d[k,"tourney_date"],length(cf))
,
                             rep(d[k,"tourney_level"],length(cf)
),
                             rep(tipo[torneo],length(cf))
                             ))

dd <- dd[-c(1:end),]

}

# COMPLETAR ALTURAS #
partidos$htf <- as.numeric(partidos$htf)
partidos[partidos$cf == "Ji Sung Nam","htf"] <- 183
partidos[partidos$cf == "Jc Aragone","htf"] <- 178
partidos[partidos$cf == "Bjorn Fratangelo","htf"] <- 183;partido
s[partidos$cf == "Kevin King","htf"] <- 190
partidos[partidos$cf == "Noah Rubin","htf"] <- 175;partidos[part
idos$cf == "Mackenzie Mcdonald","htf"] <- 178
partidos[partidos$cf == "Christian Harrison","htf"] <- 180;parti
dos[partidos$cf == "Tomas Martin Etcheverry","htf"] <- 196
partidos[partidos$cf == "Nicola Kuhn","htf"] <- 185;partidos[par
tidos$cf == "Adrian Andreev","htf"] <- 180
partidos[partidos$cf == "Altug Celikbilek","htf"] <- 185; partid
os[partidos$cf == "Kacper Zuk","htf"] <- 183
partidos[partidos$cf == "Andrea Arnaboldi","htf"] <- 175;partido
s[partidos$cf == "Pavel Kotov","htf"] <- 191
partidos[partidos$cf == "Hugo Grenier","htf"] <- 196;partidos[pa
rtidos$cf == "Alex Molcan","htf"] <- 178
partidos[partidos$cf == "Michael Vrbensky","htf"] <- 183;partido
s[partidos$cf == "Tristan Lamasine","htf"] <- 183
partidos[partidos$cf == "Ergi Kirkin","htf"] <- 182;partidos[par
tidos$cf == "Max Purcell","htf"] <- 185
partidos[partidos$cf == "Kamil Majchrzak","htf"] <- 180;partidos
[partidos$cf == "Botic Van De Zandschulp","htf"] <- 191
partidos[partidos$cf == "Tristan Schoolkate","htf"] <- 183;parti
dos[partidos$cf == "Roman Safiullin","htf"] <- 185
```

```
partidos[partidos$cf == "Aleksandar Vukic","htf"] <- 188;partido
s[partidos$cf == "Dane Sweeny","htf"] <- 170
partidos[partidos$cf == "John Patrick Smith","htf"] <- 188;parti
dos[partidos$cf == "Mikael Torpegaard","htf"] <- 193
partidos[partidos$cf == "Mario Vilella Martinez","htf"] <- 178;p
artidos[partidos$cf == "Thomas Fancutt","htf"] <- 188
partidos[partidos$cf == "Andrew Harris","htf"] <- 183;partidos[p
artidos$cf == "Borna Gojo","htf"] <- 196
partidos[partidos$cf == "Blake Mott","htf"] <- 180;partidos[part
idos$cf == "Tomas Machac","htf"] <- 183
partidos[partidos$cf == "Quentin Halys","htf"] <- 191;partidos[p
artidos$cf == "Jason Kubler","htf"] <- 178
partidos[partidos$cf == "Pedro Martinez","htf"] <- 185;partidos[
partidos$cf == "Sumit Nagal","htf"] <- 178
partidos[partidos$cf == "Frederico Ferreira Silva","htf"] <- 178
;partidos[partidos$cf == "Li Tu","htf"] <- 180
partidos[partidos$cf == "Alejandro Tabilo","htf"] <- 188;partido
s[partidos$cf == "Alex Bolt","htf"] <- 183
partidos[partidos$cf == "Arthur Rinderknech","htf"] <- 196;parti
dos[partidos$cf == "Benjamin Bonzi","htf"] <- 183
partidos[partidos$cf == "Bernabe Zapata Miralles","htf"] <- 183;
partidos[partidos$cf == "Brandon Nakashima","htf"] <- 188
partidos[partidos$cf == "Carlos Taberner","htf"] <- 183;partidos
[partidos$cf == "Daniel Altmaier","htf"] <- 188
partidos[partidos$cf == "Christopher Eubanks","htf"] <- 201;part
idos[partidos$cf == "Emilio Gomez","htf"] <- 185
partidos[partidos$cf == "Ernesto Escobedo","htf"] <- 185;partido
s[partidos$cf == "Federico Coria","htf"] <- 180
partidos[partidos$cf == "Francisco Cerundolo","htf"] <- 185;part
idos[partidos$cf == "Holger Rune","htf"] <- 188
partidos[partidos$cf == "Hugo Gaston","htf"] <- 173;partidos[par
tidos$cf == "Jenson Brooksby","htf"] <- 193
partidos[partidos$cf == "Juan Ignacio Londero","htf"] <- 180;par
tidos[partidos$cf == "Juan Manuel Cerundolo","htf"] <- 183
partidos[partidos$cf == "Liam Broady","htf"] <- 183;partidos[par
tidos$cf == "Marc Polmans","htf"] <- 188
partidos[partidos$cf == "Tallon Griekspoor","htf"] <- 188;partid
os[partidos$cf == "Thiago Seyboth Wild","htf"] <- 185
for(i in 1:nrow(partidos)){if(is.na(partidos[i,"htf"] == T)){par
tidos[i,"htf"] <- trunc(mean(partidos$htf),na.rm=T)}}

ranking = data.frame("jugador" = unique(partidos$cf), "puntos" =
rep(0,length( unique(partidos$cf))))

write.csv(partidos, "F:/TFG/Cuadros/cuadros.csv")
```

Simulación

```
library(tidyverse)
library(readxl)
library(Rlab)
library(zoo)
set.seed(22040619)

cuadros <- read.csv("E:/TFG/Cuadros/cuadros.csv");cuadros <- cua
dros[,-1]
ranking = data.frame("jugador" = unique(cuadros$cf), "puntos" =
rep(0,length(unique(cuadros$cf))));ranking <- ranking[-which(ran
king$jugador == "BYE"),];rtot <- ranking;post <- rep(0, nrow(ran
king))
colnames(cuadros) <- c("jugadores", "edad", "ht", "torneo", "sur
face", "tourney_date", "tourney_type", "tipo"); cuadros$tourney_
date <- as.Date(cuadros$tourney_date)
cuadros$ht <- as.numeric(cuadros$ht)
mht <- trunc(mean(cuadros$ht,na.rm=T))
for(i in 1:nrow(cuadros)){if(is.na(cuadros[i,"ht"]) == T){cuadro
s[i,"ht"] <- mht}}
cuadros$edad <- as.numeric(cuadros$edad)
partidos <- read.csv("E:/TFG/partidos.csv");  partidos$tourney_d
ate <- as.Date(partidos$tourney_date)
partidos <- partidos%>% filter(tourney_date < as.Date("2021-01-0
1", "%Y-%m-%d") & tourney_date >= as.Date("2017-01-01", "%Y-%m-%
d"))
partidos <- partidos[,c("surface", "tourney_date", "tourney_leve
l", "winner_name", "loser_name")]
partidosp <- partidos
puntos <- read_excel("E:/TFG/Cuadros/puntos.xlsx")
aht <- unique(cuadros[,c("jugadores","edad", "ht")])

######
rp <- as.data.frame(ranking[,1]);rpo <- as.data.frame(ranking[,1
])
#####

load("E:/TFG/modelf2.Rdata")
load("E:/TFG/dataT2.RData")
#nsim <- 10
#con=1
ini = Sys.time()
### SIMULACIÓN ###
while(con <= nsim){
cuadrop <- cuadros
partidos <- partidosp
while(nrow(cuadrop) > 0){
# SEPARAR TORNEOS #
k=2
```

```
while(cuadrop[1,"torneo"] == cuadrop[k,"torneo"] & k <= nrow(cua
drop)){
  end <- k
  k=k+1
}
d <- cuadrop[1:end,]


d <- dataT(d)


add <- data.frame("surface" = NA,"tourney_date" = NA, "tourney_l
evel" = NA,"winner_name" = NA,"loser_name" = NA)
while(nrow(d)>=2){
  i = 1
  prob <- c()
  while(i < nrow(d)){
    if(d[i,"jugadores"] == "BYE"){
      prob <- c(prob,0)
    } else if(d[i+1,"jugadores"] == "BYE"){
      prob <- c(prob,1)
    } else {
      h2h <- rbind(partidos %>% filter(winner_name == d[i,"jugad
ores"] & loser_name==d[i+1,"jugadores"]),
                   partidos %>% filter(winner_name == d[i+1,"jug
adores"] & loser_name==d[i,"jugadores"]))

      if(nrow(h2h != 0)){
        h2h1 <- length(h2h[h2h[i,"winner_name"] == d[i,"jugadore
s"] & h2h[,"tourney_date"] >= (d[i,"tourney_date"] - 3*30*12),1]
)/nrow(h2h)
        h2h2 <- length(h2h[h2h[i,"winner_name"] == d[i+1,"jugado
res"] & h2h[,"tourney_date"] >= (d[i,"tourney_date"] - 3*30*12),
1])/nrow(h2h)
      } else {
        h2h1 <- 0;h2h2 <-0
      }

      if(nrow(h2h != 0)){
        h2h1_6 <- length(h2h[h2h[i,"winner_name"] == d[i,"jugado
res"] & h2h[,"tourney_date"] >= (d[i,"tourney_date"] - 6*30*12),
1])/nrow(h2h)
        h2h2_6 <- length(h2h[h2h[i,"winner_name"] == d[i+1,"juga
dores"] & h2h[,"tourney_date"] >= (d[i,"tourney_date"] - 6*30*12
),1])/nrow(h2h)
      } else {
        h2h1_6 <- 0;h2h2_6 <-0
      }

      pr <- predict(m4b, newdata = data.frame("surtour_4_dif" =
```

```
(d[i,"surtour_4"]-d[i+1,"surtour_4"]),
                                          "age_dif" = (d[i,"edad"]
-d[i+1,"edad"]),
                                          "ht_dif" = (d[i,"ht"]-d[
i+1,"ht"]),
                                          "surface" = d[i,"surface
"],
                                          "h2h_6_dif" = h2h1_6-h2h
2_6,
                                          "h2h_3_dif" = h2h1-h2h2,
                                          "rf_4_12_dif" = (d[i,"rf
_4_12"]-d[i+1,"rf_4_12"]),
                                          "rf_4_3_dif" = (d[i,"rf_
4_3"]-d[i+1,"rf_4_3"])) ,type="response")
      prob <- c(prob,pr)
    }
    i=i+2
  }
  i = 1
  ronda <- as.character(nrow(d))
  while(i < nrow(d)){
    ganador <- rbern(1,prob[i])

    if(ganador == 1){
      ranking[ranking$jugador == d[i+1,"jugadores"], "puntos"] <
- ranking[ranking$jugador == d[i+1,"jugadores"], "puntos"] + as.
numeric(puntos[puntos$Torneo == d[i+1,"tipo"], ronda ==colnames(
puntos)])
      add <- rbind(add, data.frame("surface" = d[i,"surface"],"t
ourney_date" = as.Date(d[i,"tourney_date"]), "tourney_level" = d
[i,"tourney_type"],"winner_name" = d[i,"jugadores"],"loser_name"
= d[i+1,"jugadores"]))
      d <- d[-(i+1),]
    } else {
      ranking[ranking$jugador == d[i,"jugadores"], "puntos"] <-
ranking[ranking$jugador == d[i,"jugadores"], "puntos"] + as.nume
ric(puntos[puntos$Torneo == d[i,"tipo"], ronda ==colnames(puntos
)])
      add <- rbind(add, data.frame("surface" = d[i,"surface"],"t
ourney_date" = as.Date(d[i,"tourney_date"]), "tourney_level" = d
[i,"tourney_type"],"winner_name" = d[i+1,"jugadores"],"loser_nam
e" = d[i,"jugadores"]))
      d<- d[-i,]
    }

    i = i+1
  }
}
ranking[ranking$jugador == d[,"jugadores"], "puntos"] <- ranking
[ranking$jugador == d[,"jugadores"], "puntos"] + as.numeric(punt
```

```
os[puntos$Torneo == d[,"tipo"], 1 == colnames(puntos)])
add<-add[-1,];add$tourney_date <- add$tourney_date <- as.Date(ad
d$tourney_date)
partidos <- rbind(partidos,add)
cuadrop <- cuadrop[-c(1:end),]

}
pos <- rank(-ranking$puntos)
## ATP FINALS ##

ranking$pos <- pos
r8<-ranking[order(ranking$pos),]
delete_j <- c("Stefanos Tsitsipas", "Rafael Nadal", "Matteo Berr
ettini", "Dominic Thiem", "Cristian Garin")
for(v in delete_j){r8 <- r8[-which(r8$jugador == v),]}
r8 <- r8[1:8,]

for (i in 1:nrow(r8)) {
  r8[i,"edad"] <- aht[match(r8[i,"jugador"], aht$jugadores),"eda
d"]
  r8[i,"ht"] <- aht[match(r8[i,"jugador"], aht$jugadores),"ht"]
  r8[i,"surface"] <- "Hard"; r8[i,"tourney_type"] <- "F";r8[i,"t
ourney_date"] <- as.Date("2021-11-22")
}
colnames(r8)[1] <- "jugadores"

r8 <- dataT(r8)
s <- sample(3:8,3)
g1 <- r8[c(1,s),];g1$p <- 0;g1 <- fasegrupos(g1)
g2 <- r8[-c(1,s),];g2$p <- 0;g2 <- fasegrupos(g2)
for(i in 1:4){ranking[ranking$jugador == g1[i,"jugadores"], "pun
tos"] <- ranking[ranking$jugador == g1[i,"jugadores"], "puntos"]
+ 200*g1[i,"p"]
ranking[ranking$jugador == g2[i,"jugadores"], "puntos"] <- ranki
ng[ranking$jugador == g2[i,"jugadores"], "puntos"] + 200*g2[i,"p
"]}

sem <- semis(g1,g2)
d <- sem

while(nrow(d)>=2){
  i = 1
  prob <- c()
  while(i < nrow(d)){
    h2h <- rbind(partidos %>% filter(winner_name == d[i,"jugador
es"] & loser_name==d[i+1,"jugadores"]),
                 partidos %>% filter(winner_name == d[i+1,"jugad
ores"] & loser_name==d[i,"jugadores"]))

    if(nrow(h2h != 0)){
```

```
      h2h1 <- length(h2h[h2h[i,"winner_name"] == d[i,"jugadores"
] & h2h[,"tourney_date"] >= (d[i,"tourney_date"] - 3*30*12),1])/
nrow(h2h)
      h2h2 <- length(h2h[h2h[i,"winner_name"] == d[i+1,"jugadore
s"] & h2h[,"tourney_date"] >= (d[i,"tourney_date"] - 3*30*12),1]
)/nrow(h2h)
    } else {
      h2h1 <- 0;h2h2 <-0
    }

    if(nrow(h2h != 0)){
      h2h1_6 <- length(h2h[h2h[i,"winner_name"] == d[i,"jugadore
s"] & h2h[,"tourney_date"] >= (d[i,"tourney_date"] - 6*30*12),1]
)/nrow(h2h)
      h2h2_6 <- length(h2h[h2h[i,"winner_name"] == d[i+1,"jugado
res"] & h2h[,"tourney_date"] >= (d[i,"tourney_date"] - 6*30*12),
1])/nrow(h2h)
    } else {
      h2h1_6 <- 0;h2h2_6 <-0
    }

    pr <- predict(m4b, newdata = data.frame("surtour_4_dif" = (d
[i,"surtour_4"]-d[i+1,"surtour_4"]),
                                            "age_dif" = (d[i,"ed
ad"]-d[i+1,"edad"]),
                                            "ht_dif" = (d[i,"ht"
]-d[i+1,"ht"]),
                                            "surface" = d[i,"sur
face"],
                                            "h2h_6_dif" = h2h1_6
-h2h2_6,
                                            "h2h_3_dif" = h2h1-h
2h2,
                                            "rf_4_12_dif" = (d[i
,"rf_4_12"]-d[i+1,"rf_4_12"]),
                                            "rf_4_3_dif" = (d[i,
"rf_4_3"]-d[i+1,"rf_4_3"])) ,type="response")
    prob <- c(prob,pr)
    i=i+2
  }
  i = 1
  while(i < nrow(d)){
    ganador <- rbern(1,prob[i])
    if(ganador == 1){
      ranking[ranking$jugador == d[i+1,"jugadores"], "puntos"] <
- ranking[ranking$jugador == d[i+1,"jugadores"], "puntos"] + 400
      d <- d[-(i+1),]
    } else {
      ranking[ranking$jugador == d[i,"jugadores"], "puntos"] <-
ranking[ranking$jugador == d[i,"jugadores"], "puntos"] + 400
```

```
      d<- d[-i,]
    }

    i = i+1
  }
}
ranking[ranking$jugador == d[1,"jugadores"], "puntos"] <- rankin
g[ranking$jugador == d[1,"jugadores"], "puntos"] + 500

################
# rtot$puntos <- rtot$puntos + ranking$puntos
rp[,con+1] <- ranking$puntos
rpo[,con+1] <- rank(-ranking$puntos)
ranking$puntos <- 0; ranking <- ranking[,-3]
con = con +1
#post <- post + pos
}

rtot[,2] <- rowMeans(rp[,-1]);rtot[,3] <- apply(rp[,-1], 1, sd)
rtot[,4] <- rowMeans(rpo[,-1]);rtot[,5] <- apply(rpo[,-1], 1, sd
)
rtot[,6] <- rank(-rtot$puntos)

con=nsim+1
nsim <- 500
```

Resultados

```
library(ggplot2)
library(hrbrthemes)
library(viridis)
library(tidyverse)
library(fmsb)

####################
## PRUEBA MODELO ##
####################
(1/(1/1.9 + 1/1.9))/2.75

load("E:/TFG/modelf2.Rdata");load("E:/TFG/dataT.RData")

partidos <- rbind(read.csv("https://raw.githubusercontent.com/JeffSackmann/tennis_atp/master/atp_matches_2018.csv"),
                  read.csv("https://raw.githubusercontent.com/JeffSackmann/tennis_atp/master/atp_matches_2019.csv"),
                  read.csv("https://raw.githubusercontent.com/JeffSackmann/tennis_atp/master/atp_matches_2020.csv"),
                  read.csv("https://raw.githubusercontent.com/JeffSackmann/tennis_atp/master/atp_matches_2021.csv"),
                  read.csv("https://raw.githubusercontent.com/JeffSackmann/tennis_atp/master/atp_matches_2022.csv"))
eliminate <- c()
for(i in 1:nrow(partidos)){
  a <- partidos[i,"score"]
  if(substr(a,nchar(a)-2,nchar(a)) == "RET" | substr(a,nchar(a)-2,nchar(a)) == "W/O" | substr(a,1,nchar(a)) == ">"
     | substr(a,1,nchar(a)) == "0-0 0-0" | substr(a,1,nchar(a)) == "0-3" | substr(a,1,nchar(a)) == "0-3 Played and abandoned"
     | substr(a,1,nchar(a)) == "Walkover" | substr(a,1,nchar(a)) == "In Progress" | substr(a,1,nchar(a)) == "Def." | substr(a,1,nchar(a)) == "Apr-00"){
    eliminate <- c(eliminate,i)
  }
}
partidos <- partidos[-eliminate,]

partidos$tourney_date <- as.Date(as.Date(as.character(partidos$tourney_date), "%Y%m%d")); partidos <- partidos[,c("surface", "tourney_date", "tourney_level", "winner_name", "loser_name")]


# RAFA NADAL - DANIIL MEDVEDEV / Australian Open

date <- as.Date("2022-01-17")
dd <- data.frame("jugadores" = c("Rafael Nadal","Daniil Medvedev"),"tourney_date" = rep(date,2),
```

```
                  "surface" = rep("Hard",2),"tourney_type" = rep(
"G",2),"edad" = c(36,25),ht=c(185,198))

dd <- dataT(dd);h2h <- metricas_faltantes(dd,m4b)
p1 <- predict(m4b, newdata = data.frame("surtour_4_dif" = (dd[1,
"surtour_4"]-dd[2,"surtour_4"]),
                                        "age_dif" = (dd[1,"ed
ad"]-dd[2,"edad"]),
                                        "ht_dif" = (dd[1,"ht"
]-dd[2,"ht"]),
                                        "surface" = dd[1,"sur
face"],
                                        "h2h_6_dif" = h2h[1]-
h2h[2],
                                        "h2h_3_dif" = h2h[3]-
h2h[4],
                                        "rf_4_12_dif" = (dd[1
,"rf_4_12"]-dd[2,"rf_4_12"]),
                                        "rf_4_3_dif" = (dd[1,
"rf_4_3"]-dd[2,"rf_4_3"])) ,type="response");p2 <- 1-p1

ggplot(dd, aes(x="", y=c(p1,p2), fill=jugadores)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() +
  theme(legend.position="none") +
  geom_text(aes(y = c(p1-0.5*p1,(p1+p2)-0.5*(p2)), label = jugad
ores), color = "white", size=5.5) +
  geom_text(aes(y = c(0.90*p1-0.5*p1,0.95*(p1+p2)-0.5*(p2)), lab
el = c(paste(round(p1*100,2), "%"),paste(round(p2*100,2), "%")))
, color = "white", size=4) +
  scale_fill_brewer(palette="Set1")+ ggtitle(label = "Rafa Nadal
vs Daniil Medvedev",
                                             subtitle = "Austali
an Open / Grand Slam - Dura")

data <- data.frame("edad"=c(40,15), "height" = c(215,170),"surtu
r" = c(1,0), "h2h3" = c(1,0),"h2h6" = c(1,0),"rf12" = c(0.15,-0.
15),"rf3" = c(0.15,-0.15))
data <- rbind(data,
              data.frame("edad"=dd[1,"edad"], "height" = dd[1,"h
t"],"surtur" = dd[1,"surtour_4"],
                         "h2h3" = h2h[1],"h2h6" = h2h[3],"rf12"
= dd[1,"rf_4_12"],"rf3" = dd[1,"rf_4_3"]),
              data.frame("edad"=dd[2,"edad"], "height" = dd[2,"h
t"],"surtur" = dd[2,"surtour_4"],
                         "h2h3" = h2h[2],"h2h6" = h2h[4],"rf12"
= dd[2,"rf_4_12"],"rf3" = dd[2,"rf_4_3"]))

colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rg
```

```
b(0.7,0.5,0.1,0.9) )
colors_in=c( rgb(0.2,0.5,0.5,0.4), rgb(0.8,0.2,0.5,0.4) , rgb(0.
7,0.5,0.1,0.4) )

# plot with default options:
radarchart(data,
           #custom polygon
           pcol=colors_border , pfcol=colors_in , plwd=4 , plty=
1,
           cglcol="grey", cglty=1, axislabcol="grey", caxislabel
s=seq(0,20,5), cglwd=0.8,
           #custom labels
           vlcex=1.2 )

legend(x=0.7, y=1.4, legend = dd$jugadores, bty = "n", pch=20 ,
col=colors_in , text.col = "black", cex=1, pt.cex=3)

# Novak Djokovic - Carlos Alcaraz

date <- as.Date("2022-05-01")
dd <- data.frame("jugadores" = c("Novak Djokovic","Carlos Alcara
z"),"tourney_date" = rep(date,2),
                 "surface" = rep("Clay",2),"tourney_type" = rep(
"M",2),"edad" = c(34,19),ht=c(188,185))

dd <- dataT(dd);h2h <- metricas_faltantes(dd,m4b)
p1 <- predict(m4b, newdata = data.frame("surtour_4_dif" = (dd[1,
"surtour_4"]-dd[2,"surtour_4"]),
                                        "age_dif" = (dd[1,"edad"
]-dd[2,"edad"]),
                                        "ht_dif" = (dd[1,"ht"]-d
d[2,"ht"]),
                                        "surface" = dd[1,"surfac
e"],
                                        "h2h_6_dif" = h2h[1]-h2h
[2],
                                        "h2h_3_dif" = h2h[3]-h2h
[4],
                                        "rf_4_12_dif" = (dd[1,"r
f_4_12"]-dd[2,"rf_4_12"]),
                                        "rf_4_3_dif" = (dd[1,"rf
_4_3"]-dd[2,"rf_4_3"])) ,type="response");p2 <- 1-p1

ggplot(dd, aes(x="", y=c(p1,p2), fill=jugadores)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() +
  theme(legend.position="none") +
  geom_text(aes(y = c(p1-0.5*p1,(p1+p2)-0.5*(p2)), label = jugad
ores), color = "white", size=5.5) +
```

```
  geom_text(aes(y = c(0.90*p1-0.5*p1,0.95*(p1+p2)-0.5*(p2)), lab
el = c(paste(round(p1*100,2), "%"),paste(round(p2*100,2), "%")))
, color = "white", size=4) +
  scale_fill_brewer(palette="Set1")+ ggtitle(label = "Novak Djok
ovic vs Carlos Alcaraz",
                                             subtitle = "Madrid
Open / Masters 1000 (M) - Arcilla")

data <- data.frame("edad"=c(40,15), "height" = c(215,170),"surtu
r" = c(1,0), "h2h3" = c(1,0),"h2h6" = c(1,0),"rf12" = c(0.15,-0.
15),"rf3" = c(0.15,-0.15))
data <- rbind(data,
              data.frame("edad"=dd[1,"edad"], "height" = dd[1,"h
t"],"surtur" = dd[1,"surtour_4"],
                         "h2h3" = h2h[1],"h2h6" = h2h[3],"rf12"
= dd[1,"rf_4_12"],"rf3" = dd[1,"rf_4_3"]),
              data.frame("edad"=dd[2,"edad"], "height" = dd[2,"h
t"],"surtur" = dd[2,"surtour_4"],
                         "h2h3" = h2h[2],"h2h6" = h2h[4],"rf12"
= dd[2,"rf_4_12"],"rf3" = dd[2,"rf_4_3"]))

colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rg
b(0.7,0.5,0.1,0.9) )
colors_in=c( rgb(0.2,0.5,0.5,0.4), rgb(0.8,0.2,0.5,0.4) , rgb(0.
7,0.5,0.1,0.4) )

# plot with default options:
radarchart(data,
           #custom polygon
           pcol=colors_border , pfcol=colors_in , plwd=4 , plty=
1,
           cglcol="grey", cglty=1, axislabcol="grey", caxislabel
s=seq(0,20,5), cglwd=0.8,
           #custom labels
           vlcex=1.2 )

legend(x=0.7, y=1.4, legend = dd$jugadores, bty = "n", pch=20 ,
col=colors_in , text.col = "grey", cex=1.2, pt.cex=3)

# Daniel Evans - Rublev

date <- as.Date("2022-02-21")
dd <- data.frame("jugadores" = c("Daniel Evans","Andrey Rublev")
,"tourney_date" = rep(date,2),
                 "surface" = rep("Hard",2),"tourney_type" = rep(
"A",2),"edad" = c(32,24),ht=c(175,188))

dd <- dataT(dd);h2h <- metricas_faltantes(dd,m4b)
p1 <- predict(m4b, newdata = data.frame("surtour_4_dif" = (dd[1,
"surtour_4"]-dd[2,"surtour_4"]),
```

```
                                          "age_dif" = (dd[1,"edad"
]-dd[2,"edad"]),
                                          "ht_dif" = (dd[1,"ht"]-d
d[2,"ht"]),
                                          "surface" = dd[1,"surfac
e"],
                                          "h2h_6_dif" = h2h[1]-h2h
[2],
                                          "h2h_3_dif" = h2h[3]-h2h
[4],
                                          "rf_4_12_dif" = (dd[1,"r
f_4_12"]-dd[2,"rf_4_12"]),
                                          "rf_4_3_dif" = (dd[1,"rf
_4_3"]-dd[2,"rf_4_3"])) ,type="response");p2 <- 1-p1

ggplot(dd, aes(x="", y=c(p1,p2), fill=jugadores)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() +
  theme(legend.position="none") +
  geom_text(aes(y = c(p1-0.45*p1,(p1+p2)-0.35*(p2)), label = jug
adores), color = "white", size=5.5) +
  geom_text(aes(y = c(1.20*p1-0.5*p1,1.05*(p1+p2)-0.5*(p2)), lab
el = c(paste(round(p1*100,2), "%"),paste(round(p2*100,2), "%")))
, color = "white", size=4) +
  scale_fill_brewer(palette="Set1")+ ggtitle(label = "Daniel Eva
ns vs Andrey Rublev",
                                             subtitle = "Dubai O
pen / Atp 500 (A) - Dura")

data <- data.frame("edad"=c(40,15), "height" = c(215,170),"surtu
r" = c(1,0), "h2h3" = c(1,0),"h2h6" = c(1,0),"rf12" = c(0.15,-0.
15),"rf3" = c(0.15,-0.15))
data <- rbind(data,
              data.frame("edad"=dd[1,"edad"], "height" = dd[1,"h
t"],"surtur" = dd[1,"surtour_4"],
                         "h2h3" = h2h[1],"h2h6" = h2h[3],"rf12"
= dd[1,"rf_4_12"],"rf3" = dd[1,"rf_4_3"]),
              data.frame("edad"=dd[2,"edad"], "height" = dd[2,"h
t"],"surtur" = dd[2,"surtour_4"],
                         "h2h3" = h2h[2],"h2h6" = h2h[4],"rf12"
= dd[2,"rf_4_12"],"rf3" = dd[2,"rf_4_3"]))

colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rg
b(0.7,0.5,0.1,0.9) )
colors_in=c( rgb(0.3,0.5,0.5,0.4), rgb(0.8,0.2,0.5,0.4) , rgb(0.
7,0.5,0.1,0.4) )

# plot with default options:
radarchart(data,
```

```
            #custom polygon
            pcol=colors_border , pfcol=colors_in , plwd=4 , plty=
1,
            cglcol="grey", cglty=1, axislabcol="grey", caxislabel
s=seq(0,20,5), cglwd=0.8,
            #custom labels
            vlcex=1.2 )

legend(x=0.7, y=1.4, legend = dd$jugadores, bty = "n", pch=20 ,
col=colors_in , text.col = "grey", cex=1.2, pt.cex=3)

# Novak Djokovic - Rafael Nadal

date <- as.Date("2022-02-21")
dd <- data.frame("jugadores" = c("Novak Djokovic","Rafael Nadal"
),"tourney_date" = rep(date,2),
                 "surface" = rep("Clay",2),"tourney_type" = rep(
"G",2),"edad" = c(35,36),ht=c(188,185))

dd <- dataT(dd);h2h <- metricas_faltantes(dd,m4b)
p1 <- predict(m4b, newdata = data.frame("surtour_4_dif" = (dd[1,
"surtour_4"]-dd[2,"surtour_4"]),
                                        "age_dif" = (dd[1,"edad"
]-dd[2,"edad"]),
                                        "ht_dif" = (dd[1,"ht"]-d
d[2,"ht"]),
                                        "surface" = dd[1,"surfac
e"],
                                        "h2h_6_dif" = h2h[1]-h2h
[2],
                                        "h2h_3_dif" = h2h[3]-h2h
[4],
                                        "rf_4_12_dif" = (dd[1,"r
f_4_12"]-dd[2,"rf_4_12"]),
                                        "rf_4_3_dif" = (dd[1,"rf
_4_3"]-dd[2,"rf_4_3"])) ,type="response");p2 <- 1-p1

ggplot(dd, aes(x="", y=c(p1,p2), fill=jugadores)) +
  geom_bar(stat="identity", width=1) +
  coord_polar("y", start=0) +
  theme_void() +
  theme(legend.position="none") +
  geom_text(aes(y = c(p1+1.2*p1,(p2)-0.6*(p2)), label = jugadore
s), color = "white", size=5.5) +
  geom_text(aes(y = c(p1+1.075*p1,(p2)-0.5*(p2)), label = c(past
e(round(p1*100,2), "%"),paste(round(p2*100,2), "%"))), color = "
white", size=4) +
  scale_fill_brewer(palette="Set1")+ ggtitle(label = "Novak Djok
ovic vs Rafael Nadal",
                                             subtitle = "Roland
```

```
Garros / Grand Slam - Arcilla")

data <- data.frame("edad"=c(40,15), "height" = c(215,170),"surtu
r" = c(1,0), "h2h3" = c(1,0),"h2h6" = c(1,0),"rf12" = c(0.15,-0.
15),"rf3" = c(0.15,-0.15))
data <- rbind(data,
              data.frame("edad"=dd[1,"edad"], "height" = dd[1,"h
t"],"surtur" = dd[1,"surtour_4"],
                         "h2h3" = h2h[1],"h2h6" = h2h[3],"rf12"
= dd[1,"rf_4_12"],"rf3" = dd[1,"rf_4_3"]),
              data.frame("edad"=dd[2,"edad"], "height" = dd[2,"h
t"],"surtur" = dd[2,"surtour_4"],
                         "h2h3" = h2h[2],"h2h6" = h2h[4],"rf12"
= dd[2,"rf_4_12"],"rf3" = dd[2,"rf_4_3"]))

colors_border=c( rgb(0.2,0.5,0.5,0.9), rgb(0.8,0.2,0.5,0.9) , rg
b(0.7,0.5,0.1,0.9) )
colors_in=c( rgb(0.3,0.5,0.5,0.4), rgb(0.8,0.2,0.5,0.4) , rgb(0.
7,0.5,0.1,0.4) )

# plot with default options:
radarchart(data,
           #custom polygon
           pcol=colors_border , pfcol=colors_in , plwd=4 , plty=
1,
           cglcol="grey", cglty=1, axislabcol="grey", caxislabel
s=seq(0,20,5), cglwd=0.8,
           #custom labels
           vlcex=1.2 )

legend(x=0.7, y=1.4, legend = dd$jugadores, bty = "n", pch=20 ,
col=colors_in , text.col = "grey", cex=1.2, pt.cex=3)
#####################################
## RESULTADOS GENERALES SIMULACIÓN ##
#####################################
rtot <- read.csv("E:/TFG/sim.csv")[,-1]
rtot$dif <- rtot$real-rtot$V6
library(ggplot2)
library(dplyr)
library(hrbrthemes)
library(tidyverse)

# TOP 20
data <- rtot %>%
  arrange(puntos) %>%
  mutate(jugador=factor(jugador,jugador))
data <- data[c((nrow(data)-20):nrow(data)),];data$puntos <- as.n
umeric(data$puntos)
ggplot(data, aes(x=jugador, y=puntos)) +
  geom_segment(
```

```
    aes(x=jugador, xend=jugador, y=0, yend=puntos),
    color="black",
    size=0.7
  ) +
  geom_point(
    color="red", fill=alpha("orange", 0.75), shape = 21,
    size=5
  ) +
  theme_ipsum() +
  coord_flip() +
  theme(
    legend.position="none"
  ) +
  xlab("Jugadores") +
  ylab("Diferencias") +
  ggtitle("Top 20 ATP simulado")

# Mayores diferencias Negativas
data <- rtot %>%
  arrange(puntos) %>%
  mutate(jugador=factor(jugador,jugador))
data <- data[c((nrow(data)-50):nrow(data)),]
data <- data %>%
  arrange(dif) %>%
  mutate(dif = abs(dif))
data<-data[1:10,]

p <- ggplot(data, aes(x=jugador, y=dif)) +
  geom_segment(
    aes(x=jugador, xend=jugador, y=0, yend=dif),
    color=ifelse(data$jugador %in% c("Matteo Berrettini","Camero
n Norrie", "Aslan Karatsev"), "orange", "black"),
    size=ifelse(data$jugador %in% c("Matteo Berrettini","Cameron
Norrie", "Aslan Karatsev"), 1.3, 0.7)
  ) +
  geom_point(
    color=ifelse(data$jugador %in% c("Matteo Berrettini","Camero
n Norrie", "Aslan Karatsev"), "orange", "black"),
    size=ifelse(data$jugador %in% c("Matteo Berrettini","Cameron
Norrie", "Aslan Karatsev"), 5, 2)
  ) +
  theme_ipsum() +
  coord_flip() +
  theme(
    legend.position="none"
  ) +
  xlab("Jugadores") +
  ylab("Diferencias") +
  ggtitle("Mayores diferencias negativas en el top 50")
p
```

```r
# Mayores diferencias Positivas
data <- rtot %>%
  arrange(puntos) %>%
  mutate(jugador=factor(jugador,jugador))
data <- data[c((nrow(data)-50):nrow(data)),]
data <- data %>%
  arrange(-dif) %>%
  mutate(dif = abs(dif))
data<-data[1:10,]

p <- ggplot(data, aes(x=jugador, y=dif)) +
  geom_segment(
    aes(x=jugador, xend=jugador, y=0, yend=dif),
    color=ifelse(data$jugador %in% c("Pedro Martinez","Miomir Ke
cmanovic", "Dominic Thiem"), "orange", "black"),
    size=ifelse(data$jugador %in% c("Pedro Martinez","Miomir Kec
manovic", "Dominic Thiem"), 1.3, 0.7)
  ) +
  geom_point(
    color=ifelse(data$jugador %in% c("Pedro Martinez","Miomir Ke
cmanovic", "Dominic Thiem"), "orange", "black"),
    size=ifelse(data$jugador %in% c("Pedro Martinez","Miomir Kec
manovic", "Dominic Thiem"), 5, 2)
  ) +
  theme_ipsum() +
  coord_flip() +
  theme(
    legend.position="none"
  ) +
  xlab("Jugadores") +
  ylab("Diferencias") +
  ggtitle("Mayores diferencias positivas en el top 50")
p

# TOP 15
rtot <- rtot[order(rtot$real),]
data2 <- as.data.frame((rtot[1:15,c("jugador", "V6", "real")]))
data <- data.frame("jugador" = rep(0,nrow(data2)*2), "var" = rep
(0,nrow(data2)*2),"valor" = rep(0,nrow(data2)*2));i=1;j=1
while(i <= nrow(data2)*2){
  data[i,] <- list("jugador" = data2[j,1], "var" = "Simulado", "
valor" = data2[j,2])
  data[i+1,] <- list("jugador" = data2[j,1], "var" = "Real", "va
lor" = data2[j,3])
  i=i+2
  j <- j+1
}
data[c(11,12),1] <- "Berrettini";data[c(23,24),1] <- "Alliasime"
;data[c(29,30),1] <- "Schwartzman"
```

```
data[c(1,2),1] <- "Djokovic";data[c(21,22),1] <- "Norrie";data[c
(7,8),1] <- "Tsitsipas";data[c(3,4),1] <- "Medvedev"
data[c(5,6),1] <- "Zverev";data[c(9,10),1] <- "Rublev";data[c(13
,14),1] <- "Hurkacz";data[c(27,28),1] <- "Shapovalov"
ggplot(data, aes(fill=var, y=valor, x=jugador)) +
  geom_bar(position="dodge", stat="identity") + geom_hline(yinte
rcept=15,col="black", size = 1, linetype="dashed") +
  xlab("Jugadores") +
  ylab("Ranking") +
  ggtitle("Ranking real y simulado del top 15")
rm(list = ls())

################################################
## RESULTADOS JUGADORES DE INTERÉS SIMULACIÓN ##
################################################
library(ggplot2)
rtot <- read.csv("C:/Users/luis.nuevo/Documents/TFG/sim.csv")[,-
1]
rpo <- read.csv("C:/Users/luis.nuevo/Documents/TFG/rpo.csv")[,-1
]
rtot$dif <- rtot$real-rtot$V6

name = "Miomir Kecmanovic" # Ejemplo de jugador que se esperaba
más (comentar resultados 2022)
name = "Carlos Alcaraz" # No están los puntos de challenger (com
entarlo), "imposible" de preveer subida de nivel de Carlos
name = "Casper Ruud" # Ejemplo jugador bien predicho
name = "Matteo Berrettini" # Ejemplo de jugador mal predicho
caPo <- t(rpo[rpo$ranking...1. == name,]);caPo <- as.numeric(caP
o[-1,])

boxplot(caPo, main = paste("Boxplot posiciones",name),xlab=paste
("Posición ranking Simulación = ",rtot[rtot$jugador==name,6]),
        ylim=c(0,max(rtot[rtot$jugador==name,7]+2,max(caPo))), c
ol="orange", border = "brown",
        ylab = "Ranking", horizontal = F)
points(rtot[rtot$jugador==name,6], col = "black",pch=16)
points(rtot[rtot$jugador==name,7], col = "black",pch=1)
text(x = 1.14,y = (rtot[rtot$jugador==name,7]-0), labels = paste
("Ranking Real = ", rtot[rtot$jugador==name,7]), cex = 0.8)


Int <- Hmisc::cut2(caPo, seq(0,309,10))
pos <- data.frame("pos" = caPo,"int" = Int)
ggplot(pos, aes(x=pos, y=int)) +
  geom_bar(stat="identity", fill="#f68060", alpha=.6, width=.4)
+
  coord_flip() +
  xlab("") +
  theme_bw()
```