ESTUDIO SOBRE LA BASE DE DATOS FIFA 19

Alumnos: Juan Pablo Bolta Ballester,

Guillermo Ferrando Muñoz,

Evgeny Grachev

Cristina Portilla Alique

Grupo: 16
Asignatura: MDP I

Curso: 2021/2022

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Índice

1 Introducción

Esta base de datos proviene del repositorio de datos Kaggle y contiene información de todos los jugadores pertenecientes al videojuego FIFA 19, un videojuego de fútbol que se puede jugar en cualquier dispositivo electrónico. En este trabajo, estudiaremos los futbolistas y sus variables con dos objetivos principales: encontrar agrupamientos verosímiles/útiles/legítimos de dichos jugadores y predecir el valor (precio) o cualquier otra variable que resuma la calidad de un jugador de fútbol. Los agrupamientos serían de ayuda en caso de que, por ejemplo, un jugador haya resultado lesionado, poder encontrar sustitutos. Respecto al otro objetivo, si se quiere formar un nuevo equipo de fútbol, se podría estimar su calidad en general.

1.1 Descripción de las variables

En un principio, usaremos 56 variables y 17918 observaciones que se corresponden a los diferentes jugadores de fútbol del videojuego. El gran número de variables hace que la descripción de cada una de ellas ocupe mucho en la memoria, por ello, vamos a incluir las descripciones en el **Anexo 0**.

2 Pre-procesado de los datos

Hemos eliminado algunas variables que no hemos considerado útiles para el análisis, como enlaces a fotos o número de la camiseta. Para el tratamiento de datos faltantes, realizamos una primera exploración de los datos. Obtenemos el porcentaje de datos faltantes existente en cada columna y observamos una baja cantidad de datos faltantes, en concreto hay 241 jugadores con valor faltante en la variable Club y 60 jugadores con valor faltante en la variable Posición. Como vemos en los porcentajes (**Anexo 0**), el 0.26% de los jugadores de la base de datos no tienen valores sobre las variables de habilidades. Al tratarse de un grupo tan reducido y de jugadores poco importantes, decidimos eliminarlos directamente.

2.1 Limpieza y asignación de los tipos de variables adecuados

Hay variables que no están en su clase correspondiente, y algunas que hay que modificar para facilitar su tratamiento. Eliminamos el símbolo € de las variables Wage y Value, y las transformamos a tipo numérico. Por otro lado, también transformamos el peso expresándose en kilos en vez de libras y la altura a centímetros en vez de pies. Pasamos ambas a numérico.

La variable Preferred.Foot la convertimos a dicotómica y la variable Position se pasa a dummy (Preferred.Foot sería dummy también), ya que tiene 27 valores diferentes y nos interesa convertirlos en ocho (portero, lateral, central, medio, medio ofensivo, medio defensivo, extremo y delantero), agrupándolos por las categorías más importantes para facilitar la interpretación de las variables.

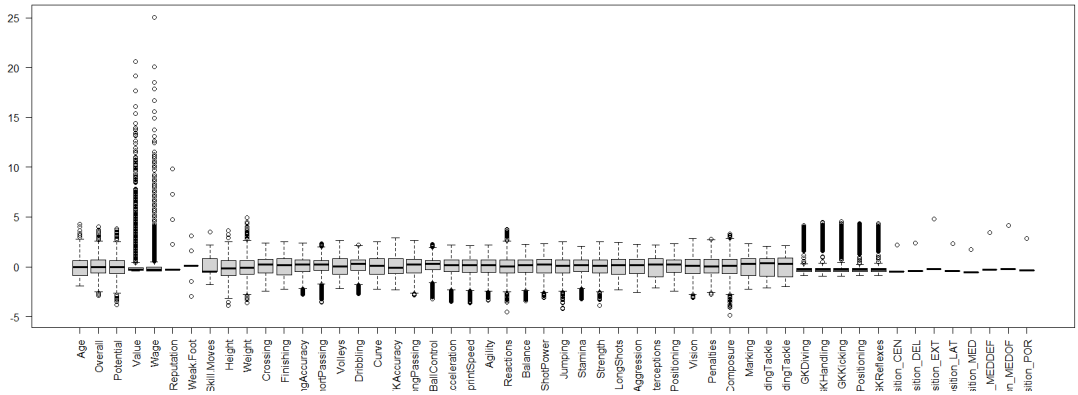
3 Análisis exploratorio

Hemos realizado un conciso análisis exploratorio de nuestras variables para familiarizarnos con ellas, para verlo en su totalidad, consultar el **Anexo 0**. Ahí estudiamos las distribuciones de cada variable para encontrar posibles datos anómalos y para decidir si será necesario estandarizar todas las variables numéricas de cara a la implementación del PCA más adelante.

3.1 Variables de habilidades de jugador

Sí remarcaremos la importancia de las variables de las habilidades de jugador, ya que componen la mayoría de variables de la base de datos. En el gráfico de **Anexo 0 - Variables de habilidades de jugador**, tenemos una serie de boxplots de las variables de habilidades de los jugadores, todas ellas medidas en una misma escala de 0 a 100. Podemos ver una clara diferenciación entre todas las variables menos las cinco últimas de la derecha. La diferencia de este pequeño grupo de variables se debe a que son medidas pensadas para los porteros, por lo que todos los demás jugadores que no lo sean tendrán valores bajos en dichas variables. En general, suponemos (sería lógico) que la gran mayoría de los datos atípicos que aparecen en cada boxplot son los porteros.

Tras el análisis, decidimos centrar y escalar todas nuestras variables numéricas mediante varianza unitaria. A continuación, podemos observar un gráfico de *boxplots* con todas ellas.



Cabe mencionar que las variables Value y Wage tienen una gran cantidad de valores fuertemente anómalos, además de haber otras variables como International.Reputation con valores atípicos. Esto nos lleva a la conclusión de que hay que estar atentos a ellas por si tenemos que considerarlas como auxiliares al modelo. Lo veremos más adelante en el primer método aplicado: PCA.

4 Primer método - PCA

4.1 Objetivo

Escogemos realizar PCA porque la mayoría de nuestras variables son numéricas, y son las que más se enfocan en nuestro estudio. Con este estudio trataremos de encontrar, en caso de que las haya, relaciones entre las distintas variables consideradas, lo que a su vez derivará en encontrar relaciones entre las observaciones.

4.2 Implementación del método

Hemos aplicado un PCA inicial con todas nuestras variables numéricas tipificadas, incluyendo a las categóricas como auxiliares en el modelo. En este primer modelo elegimos el número de componentes, estudiamos el comportamiento de datos atípicos y anómalos y hacemos gráficos de contribución de T^2 -Hotelling y suma de cuadrados residual para los mismos. Para consultar esto, ver el **Anexo 1 - PCA inicial**. Tras este modelo inicial, decidimos considerar como auxiliares las variables Wage, Value e International.Reputation.

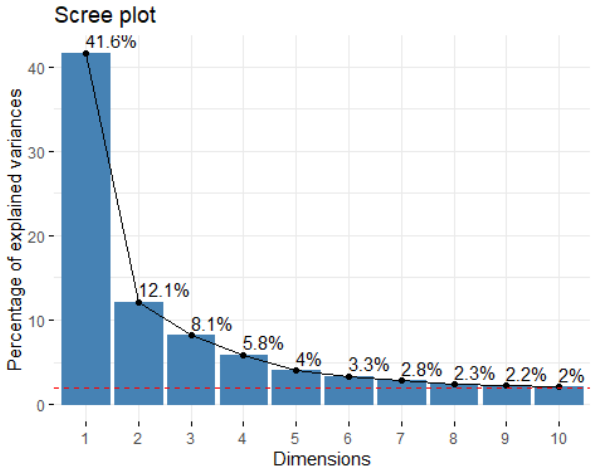
4.2.1 Elección del número de componentes

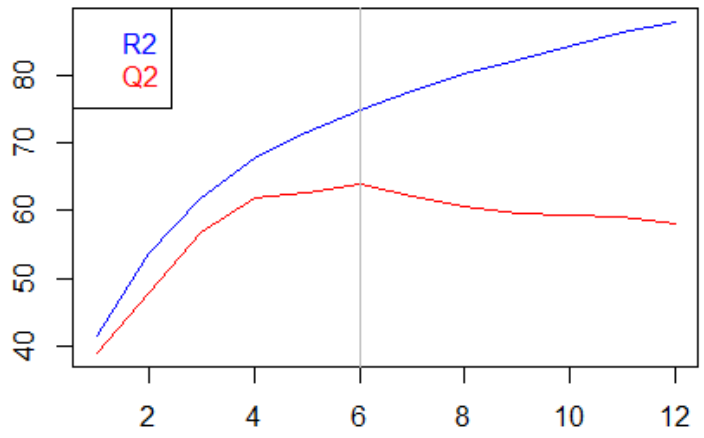
Tras el modelo inicial, ya podemos hacer un nuevo PCA marcando las variables mencionadas como auxiliares numéricas (además de las categóricas consideradas anteriormente). Primero, reconfirmamos el número de componentes óptimo para nuestros datos, de la misma forma que habíamos hecho antes.

Consultando el scree plot, con 6 componentes principales explicaríamos el 74.92% de la variabilidad de nuestros datos. Por la posible presencia de sesgo en las variables de habilidades de los jugadores, parece una cantidad razonable de inercia explicada. Si escogemos más componentes podríamos estar explicando ruido, o por lo menos más aún del que podrían estar explicando las 6 componentes.

Decimos que las variables de habilidades pueden estar sesgadas ya que sus valores están en una escala de 0 a 100, pero al parecer no se basan en un cálculo matemático objetivo.

Para ayudarnos a confirmar el número de componentes principales que nos conviene, recurriremos a la validación cruzada para obtener los estadísticos R^2 y Q^2: El primer estadístico mide la bondad de ajuste del modelo, a medida que obtengamos más variables, esperaremos que R^2 crezca, pero usar únicamente este estadístico no es una buena idea por lo que comentábamos antes sobre el ruido. Con el segundo estadístico obtendremos una visualización de cuál es el rango efectivo de nuestros datos, en otras palabras, Q^2 estima la bondad de predicción del modelo.



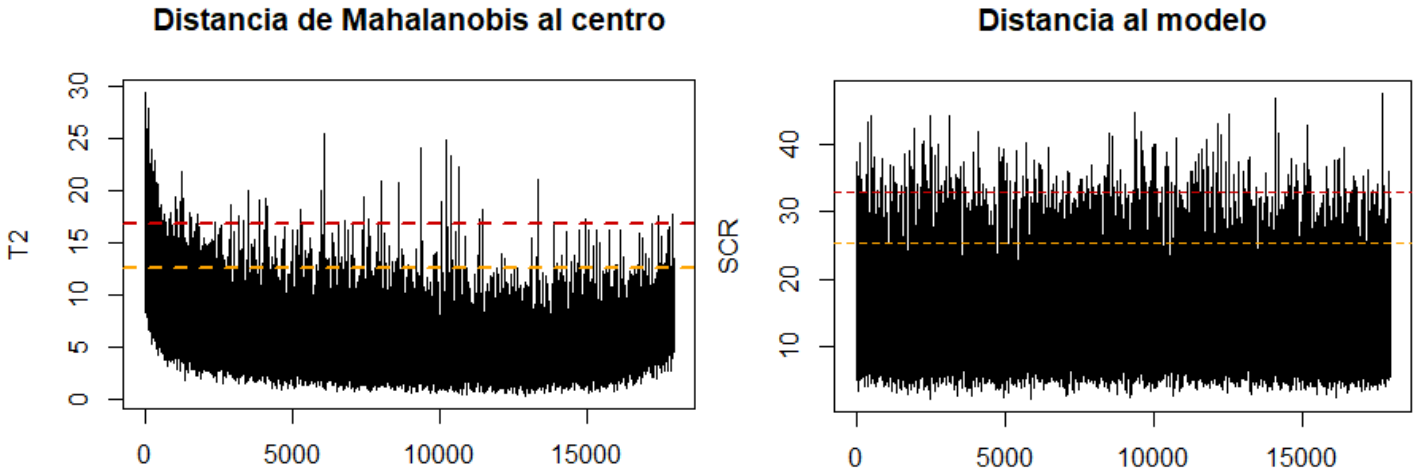


A la vista del gráfico anterior, podemos observar que Q^2 llega a un máximo con 6 componentes, a partir de ahí, las predicciones serán peores. Por tanto, podemos confirmar que tenemos una situación similar a la del PCA inicial y que usar un número de 6 componentes principales sigue siendo una buena elección, así que concluimos en usar esa cantidad.

4.2.2 Validación del modelo PCA

4.2.2.1 Detección de anómalos con T2-Hotelling y atípicos con SCR

Como siempre, antes de sacar conclusiones de nuestro modelo PCA, primero tenemos que validarlo. Comenzamos buscando outliers extremos con un gráfico de valores de T^2-Hotelling. Las líneas discontinuas se corresponden con los límites de confianza del 95% y 99%, respectivamente en naranja y rojo. Siempre y cuando la cantidad no sea excesiva, **utilizaremos como criterio de aceptación como falsas alarmas que los valores estén por debajo de tres veces el límite de confianza del 99%**, ya que queremos mantener toda la información posible.

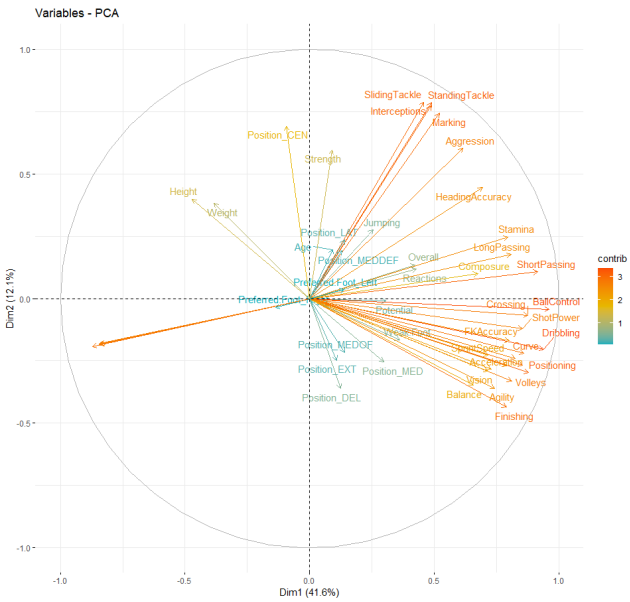


Podemos ver que, en este modelo reajustado, los valores anómalos y atípicos no superarían ni dos veces los límites de 99% en ningún caso, aunque el número de alarmas aumenta ligeramente. Respecto a las alarmas de T^2-Hotelling para el 95% y el 99% se nos quedarían, respectivamente, en 0.0359% y 0.0087%. Para la SCR, se quedaría en 0.069% y 0.0148%. Ahora podemos dar el modelo como validado.

4.2.3 Interpretación del modelo

4.2.3.1 Gráficos de loadings

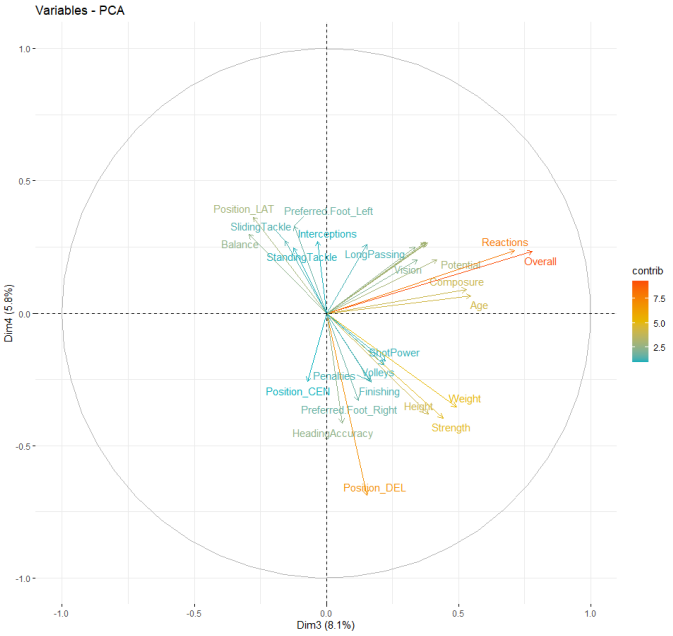
Trataremos de interpretar los distintos procesos latentes que se obtienen de nuestras variables mediante gráficos de *loadings*. Para ver una tabla resumen con las correlaciones entre variables iniciales y componentes principales, ver **Anexo 1** - Correlaciones entre variables originales y latentes.

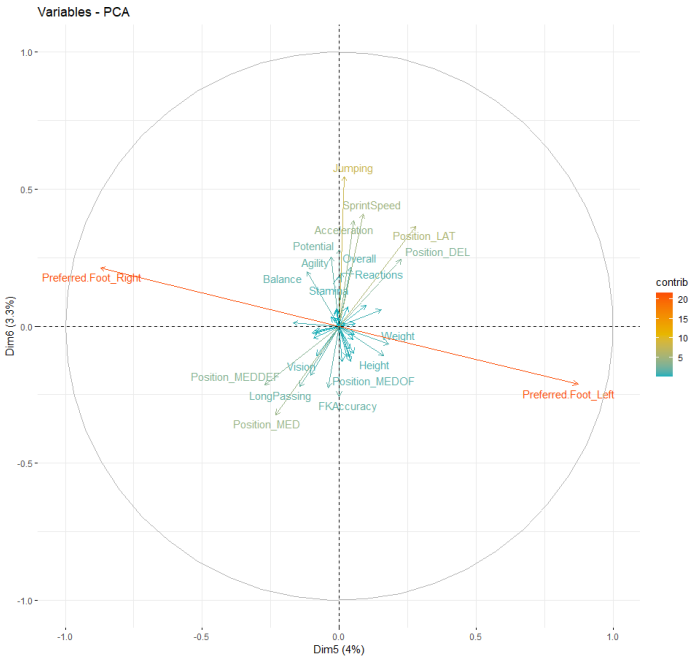


El primer gráfico de *loadings* nos muestra cómo la **primera componente** es la que más inercia explica con diferencia. Esto es porque separa las variables de habilidades de porteros (no se ven los nombres, son las flechas naranjas en sentido negativo de la primera componente) y de jugadores no porteros, estas variables tienen una fuerte contribución a dicha componente. La contribución de las variables Weight y Height en la primera componente es considerable también. Por otra parte, al parecer, la primera componente indica que apenas hay relación entre tener buenas puntuaciones en las variables de habilidades y tener fuerza, es decir, no hay correlación entre ser fuerte y ser un buen futbolista, tengas la posición en el campo que tengas. No es hasta que

nos fijamos en **la segunda componente** que podemos identificar mejor otras estructuras latentes: dicha componente sugiere, por ejemplo, que ser bueno metiendo goles, ágil con el balón y/o mantener bien el equilibrio **no** tiene correlación con ser un jugador agresivo en general, pero sí la tiene negativa con pesar y medir más. Por otro lado, la segunda componente ayuda a separar las variables Height, Weight, Position_CEN y Strengh de las variables Position_EXT, Position_MED, Position_MEDOF y Position_DEL de tal manera que se establecen relaciones negativas entre estos dos conjuntos de variables, que significa que aquellos con posición centro tienden a ser más corpulentos, a diferencia de los extremos, mediocentros, mediocentro ofensivos y delanteros, siendo lógicas dichas relaciones. Se pueden mencionar algunos aspectos interesantes, y es que, aunque sean pequeñas (tenidas en cuenta al haber eigenvalues altos), las relaciones positivas con la preferencia del uso de pie derecho y las variables de porteros, o la relación positiva entre la preferencia del pie izquierdo y la fortaleza física, pases largos y cortos, la compostura ante la presión defensora y la velocidad moviendo el balón, o que los futbolistas más mayores tiendan a ser laterales, mediocentro defensivos y que salten más. Preferred.foot_Right y Preferred.foot_Left están negativamente correlacionadas con la misma cantidad de loading, ocurre de forma paralela para el resto de componentes.

Si observamos el gráfico de *loadings* de la **tercera y cuarta componente**, donde se muestran las 30 variables con más contribución, de las conclusiones que podemos sacar y con respecto a lo mencionado en las anteriores, algunas son similares y otras contradictorias. Se puede observar que la tercera componente separa aceptablemente las variables Balance y Position_LAT de Height, Weight, Strength, Vision, Potential, Composure, Age, Reactions, Overall, etc. La cuarta componente relaciona positivamente la posición de delantero con la precisión de golpear con la cabeza y la preferencia del pie derecho, relaciona de la misma forma el pie izquierdo con el equilibrio, posición lateral y la capacidad de quitar la pelota tanto agresiva como relajadamente. Estos dos últimos conjuntos mencionados se relacionan negativamente entre ellos y están incorrelacionados con el grupo de las variables positivamente correlacionadas con Reactions y Overall.

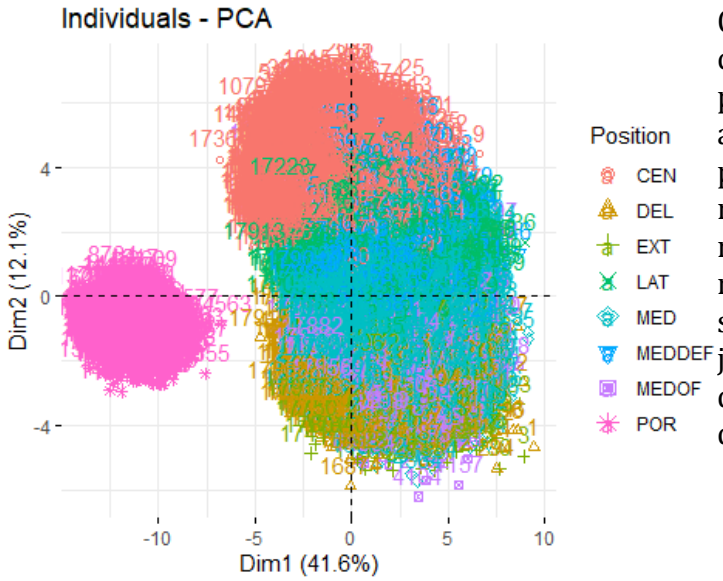




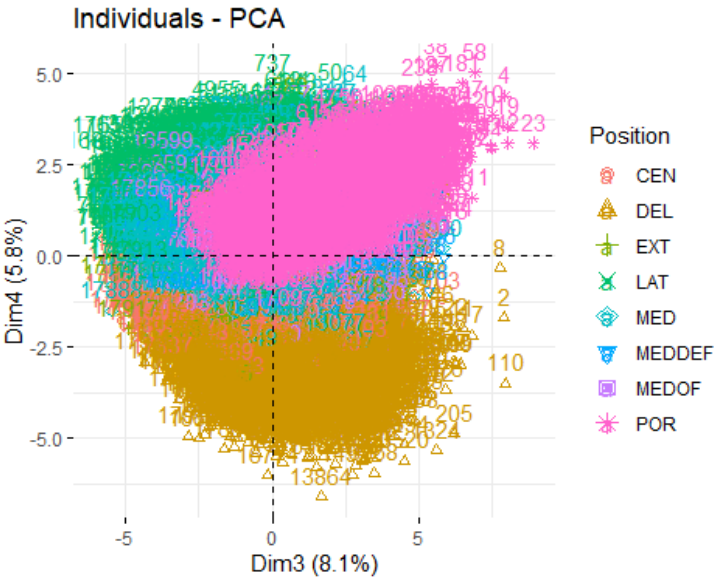
Finalmente, observando el gráfico de *loadings* de la **quinta y sexta componente**, destacamos que las variables más explicadas por la quinta componente son Preferred.Foot_Left y Preferred.Foot_Right, con una lógica correlación negativa entre ellas, como mencionábamos antes. Otras relaciones entre variables a destacar, para ambas componentes son por ejemplo los delanteros o laterales, que tienen relación positiva con potencia de salto, velocidad y aceleración; o los mediocentros, tanto defensivos como ofensivos, para los que encontramos relaciones positivas con las variables relacionadas con pases, visión de juego o control de balón. Relaciones bastante acordes con la realidad. Destaca que la altura y el peso no están relacionados con ninguna de las posiciones citadas previamente.

4.2.3.2 Gráficos de scores

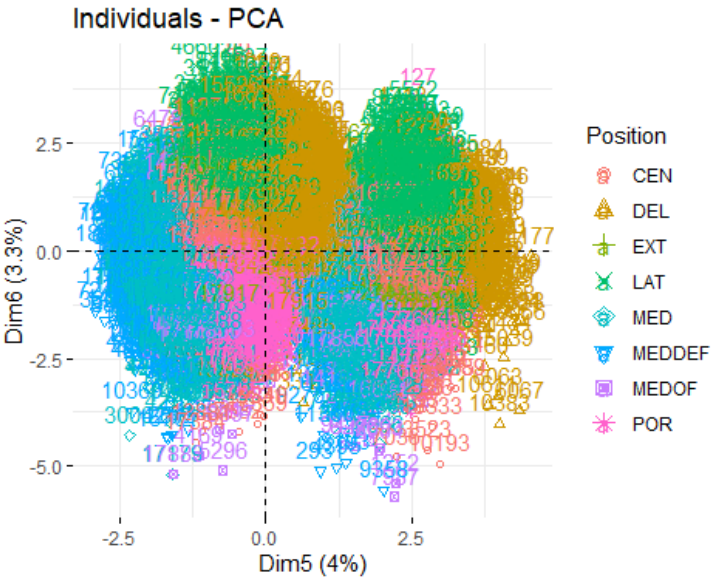
Una vez estudiados los comportamientos latentes, podemos consultar los gráficos de *scores*. Dichos *scores* los hemos coloreado según la posición del jugador.



Consultando el primer gráfico, queda clara la capacidad de explicación de variabilidad de la primera componente, ya que quedan claramente alejados dos conjuntos de individuos, siendo el más pequeño (coloreado de rosa) el de los porteros, y el más grande el del resto de los futbolistas. En el mayor conjunto podemos llegar a discernir distintas nubes de puntos, con el inconveniente de estar muy solapadas. Sí que se podría disgregar la nube de jugadores de posición centro con parte de la de delanteros, pero para esta segunda nube es difícil diferenciar el resto de posiciones.



En el segundo gráfico se nos presenta una situación similar a la anterior; las nubes fáciles de separar, si se toman por separado, son las de porteros de y de centros. Si no, estas dos componentes no contribuyen al distanciamiento entre *scores* según la posición.



Curiosamente, en el tercer gráfico, donde las componentes explican menor variabilidad, sí se muestra una mejor separación entre las distintas nubes de posiciones. Es cierto que tenemos casi dos mismas agrupaciones generales de individuos, pero no están tan solapadas como antes. Esto se debe a la fuerte contribución de las variables Preferred.Foot_Left y Preferred.Foot_Right a la quinta componente, como veíamos antes en los gráficos de loadings. Por tanto, en este tercer gráfico de *scores*, tenemos dos conjuntos similares: el de la izquierda se corresponde con aquellos jugadores que prefieren el pie derecho, y el de la derecha con los que prefieren el pie izquierdo.

Con estos resultados, podemos pasar al siguiente método para completar el primer objetivo del trabajo: *clustering*.

5 Segundo método - Clustering

5.1 Objetivo

Mediante el análisis de clustering, buscamos encontrar grupos de jugadores con características parecidas. Con este análisis se trata de agrupar a los jugadores según las habilidades de juego en las cuales destacan. El estudio de clustering puede ser de utilidad para encontrar tipos de jugadores específicos o para encontrar sustitutos de algún jugador para el equipo del usuario.

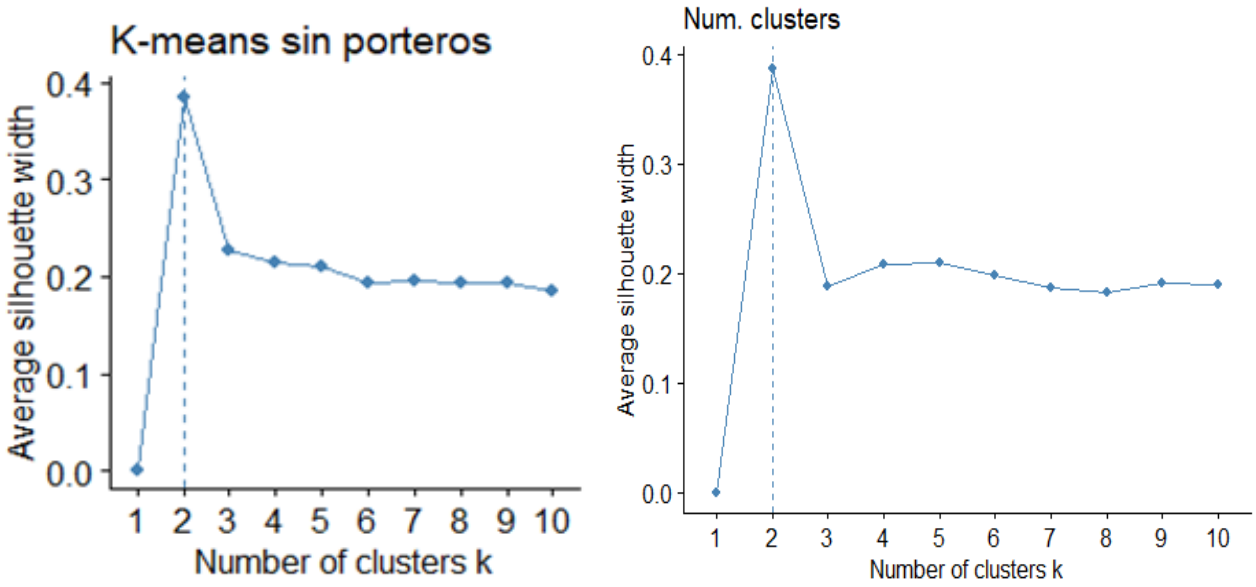
5.2 Implementación del método

Para realizar el procedimiento de clustering, utilizamos como base las coordenadas resultantes del PCA, ya que como hemos visto anteriormente las variables de nuestros datos se encuentran muy correlacionadas. A partir de los resultados del PCA, hemos visto ya una tendencia de agrupamiento muy clara que separa a los porteros de los jugadores en dos grupos bien diferenciados. En este análisis nos centraremos principalmente en si existen tendencias de agrupamiento en los jugadores, por lo que en primer lugar, separaremos a los jugadores de los porteros y realizaremos un PCA únicamente sobre los jugadores, ya que los porteros forman un cluster propio.. Siguiendo la línea de antes, escogeremos 6 componentes principales y añadiremos la suma de cuadrados residual como una séptima variable latente, para no perder la inercia no explicada de los datos originales. Para consultar la elección de componentes y validación de este PCA de solo jugadores, consultar **Anexo 2 - PCA con solo jugadores**.

5.3 Número óptimo de clusters

A continuación, estimaremos el número óptimo de clusters con el algoritmo de k-medias y utilizando como criterio el coeficiente de Silhouette. A pesar de que lo óptimo sería probar con más criterios como el criterio del codo (wss) o el estadístico de gap para contrastar los resultados, utilizamos únicamente el de Silhouette ya que estos dos últimos métodos no funcionan debido a las dimensiones de los datos.

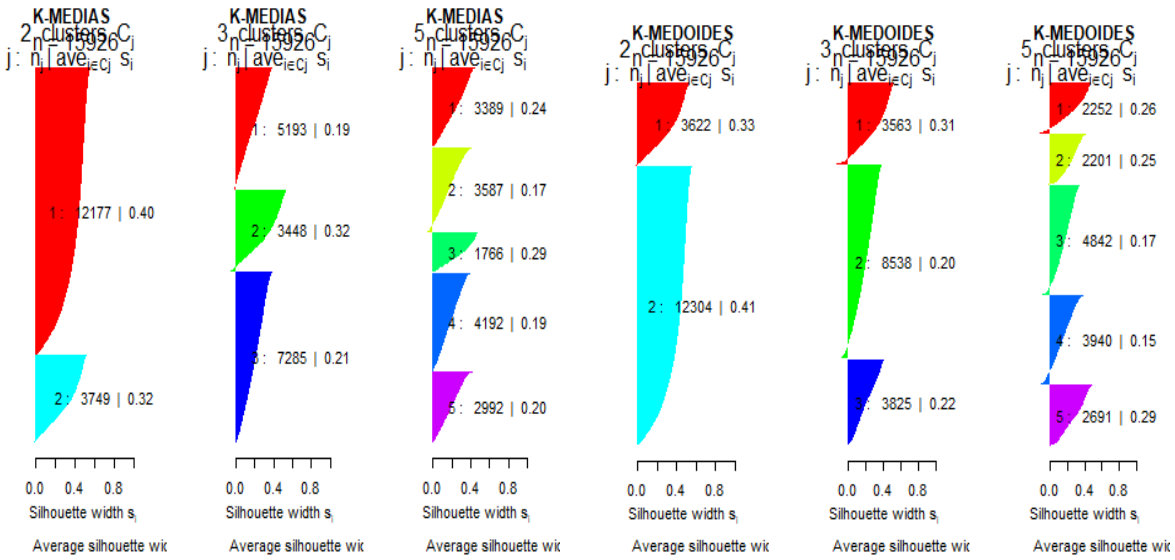
Se aplica a la base de datos sin los porteros, ya que al tener valores para las variables muy distintas, realiza la separación en dos clusters directamente, uno de porteros y otro para los jugadores. Queremos estudiar si hay tendencias de agrupamiento según las variables de juego y las posiciones de los jugadores. El objetivo principal es detectar grupos de jugadores parecidos que sean de utilidad si un usuario quiere encontrar un jugador con determinadas características para mejorar su equipo o sustituir a algún jugador. A continuación calculamos el coeficiente de silhouette para la base de datos eliminando los porteros con los métodos k-means y k-medoides.



Como se esperaba, el número de clusters que maximiza el coeficiente de Silhouette es 2, si obviamos los porteros, este coeficiente baja significativamente. El número óptimo de clusters para los jugadores sigue siendo 2, y baja para 3, 4 y 5 clusters manteniéndose en un valor similar para estos tres. Probamos a continuación con el método de k-medoides.

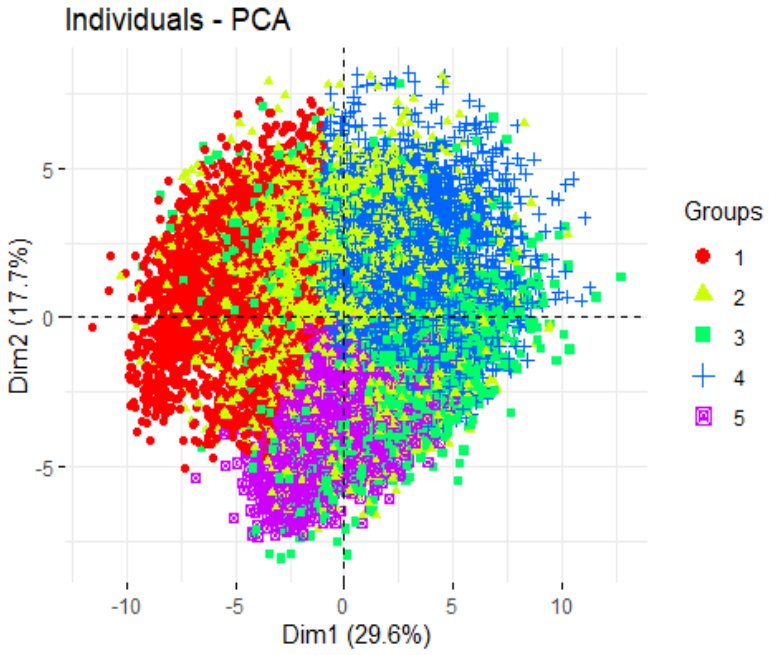
Observamos que tanto para el método de k-medias como para el de k-medoides, el número de clusters que maximiza el coeficiente de Silhouette es 2, a pesar de esto, si seleccionamos dos clusters es posible que se agrupen en un mismo cluster jugadores que no tengan demasiada relación entre sí al tratarse de clusters tan grandes. En el método de k-medias podemos ver como para 3, 4 y 5 clusters, tenemos un coeficiente de Silhouette similar, y algo más elevado que para el resto de clusters.

Debido a que los resultados son similares, es difícil decantarse por uno de los métodos. Comparamos las clasificaciones de k-medias y k-medoides para 2, 3 y 5 clusters con el coeficiente de Silhouette. Devolvemos el coeficiente de Silhouette para cada individuo por cluster en cada método.



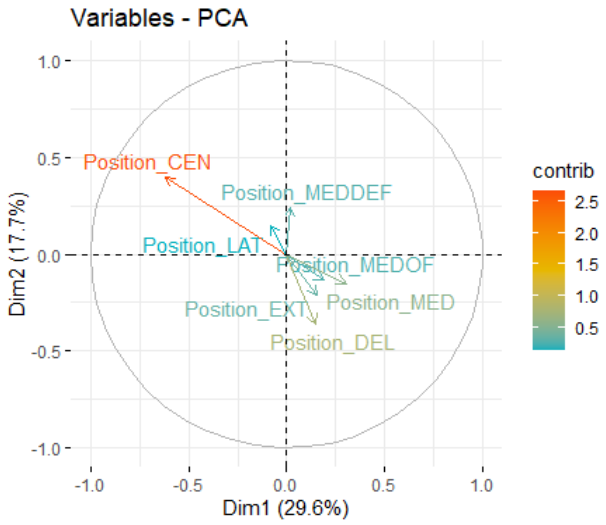
Los resultados obtenidos respectivos al valor medio del coeficiente de Silhouette son similares para los diferentes números de clusters que hemos determinado. Es cierto que para dos clusters, es mejor el agrupamiento según el coeficiente, pero al agrupar tantos individuos en dos grupos tan grandes, es posible que se estén añadiendo al mismo grupo jugadores muy diferentes entre sí. Por esto, descartamos 2 como el número óptimo de clusters y decidimos entre 3 y 5. Se observa que en el método de *kmeans*, hay menos jugadores mal clasificados que en *kmedoides*, por lo que nos fijaremos en los resultados de *kmeans*. Para 3 y 5 clusters, el coeficiente de Silhouette es prácticamente igual, seleccionaremos en principio 5 clusters, al ser un valor más próximo al número de posiciones reales en las que se dividen los jugadores. Como se observa, el coeficiente de Silhouette obtenido para 5 clusters en *kmeans* es de 0.21, un valor relativamente bajo, por lo que podemos esperar que no existan unas variables que separen los grupos claramente.

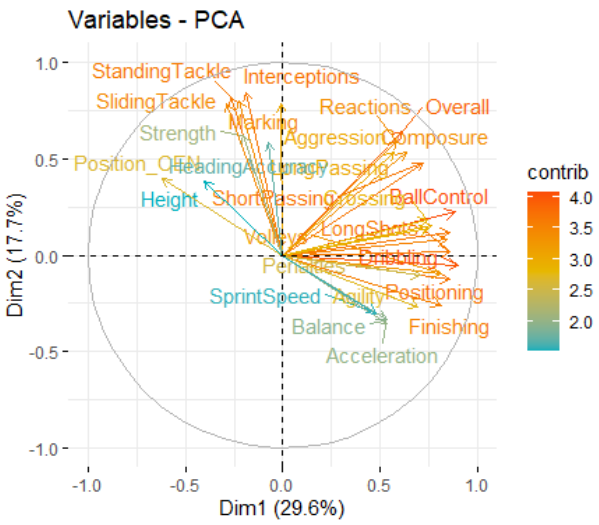
A continuación, obtenemos los gráficos de *scores* y *loadings* para ver qué variables influyen de mayor manera a la división en clusters. Queremos observar qué características de juego predominan en cada cluster. Para relacionar los clusters obtenidos con las posiciones de los jugadores y sus variables, comparamos el gráfico de *scores* con el gráfico de *loadings* para las variables. Se obtienen dos gráficos de *loadings*, uno que incluye todas las variables y su contribución, y otro más resumido que muestra las posiciones de los jugadores.



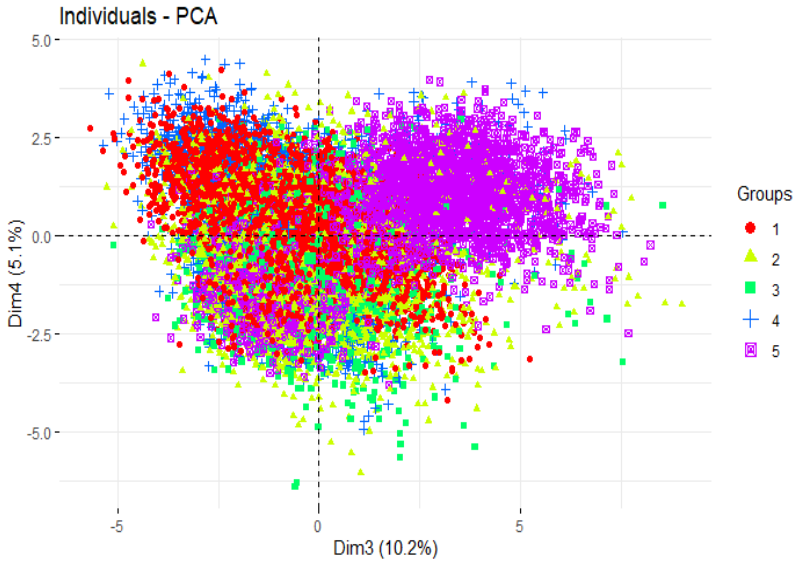
En el gráfico de *scores* (coloreados por clusters), observamos que los individuos no están muy diferenciados por lo general, algo que esperábamos debido al bajo valor del coeficiente de Silhouette. A pesar de ello, podemos diferenciar ciertos agrupamientos. La segunda componente explica un 17,7% de inercia total y marca diferencia entre el cluster 4 y el cluster 5, que a su vez se encuentran bastante mezclados en las primeras dos componentes con el cluster 2. La primera dimensión, que explica un 29,6% de la variabilidad de los datos, separa algo al cluster 1 de los clusters 3 y 4.

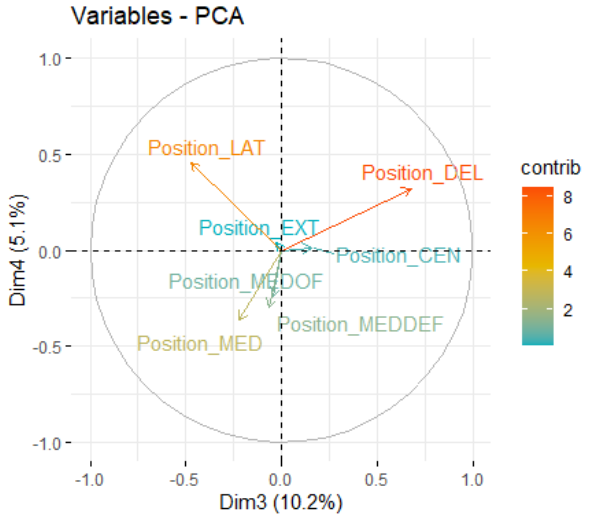
Con los gráficos de *loadings*, representamos las posiciones de los jugadores para ver si los clusters obtenidos se separan adecuadamente según la posición del jugador. También realizamos la representación para las variables respectivas a las características de cada jugador.

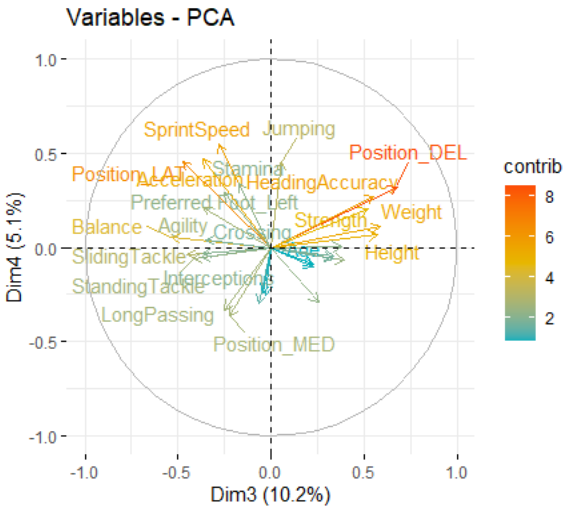




Como se puede observar, en las primeras dos dimensiones no conseguimos separar ninguno de los grupos claramente, pero se obtiene una idea general de las variables y las posiciones relacionadas con cada cluster. Antes de comenzar a interpretar, obtenemos los gráficos para las sucesivas componentes.







En la tercera y cuarta dimensión, vemos que se consigue separar el cluster 5, mientras que el resto de agrupaciones se encuentran solapadas. Con los gráficos de *loadings*, representamos las posiciones de los jugadores para ver si los clusters obtenidos se separan adecuadamente según la posición del jugador. También realizamos la representación para las variables respectivas a las características de cada jugador. Respecto a las componentes 5 y 6, no han sido incluidas en el análisis porque no aportan información relevante, en el **Anexo 2** se puede ver la separación en estas componentes y la contribución de los clusters a cada componente.

A la vista de los gráficos de *scores* y *loadings*, podemos obtener algunas conclusiones: El cluster 1 está integrado mayoritariamente por jugadores defensivos, que tienen la posición de central. Podemos ver en el gráfico de *loadings*, la dirección de la variable "Position_CEN", coincidente con el primer cluster y con una contribución razonablemente alta.

El cluster 2, también estará formado por jugadores que destacan en las habilidades defensivas, pero no necesariamente son centrales. La zona de intersección del primer cluster con el segundo, se encuentra en la dirección de variables como "Standing Tackle", "Sliding Tackle", "Interceptions", "Marking" o "Strength". Todas son características predominantes en los jugadores que destacan en el físico y en la defensa. La posición Medio Defensivo, también se observa en la dirección del cluster 2, con lo que afirmamos la suposición.

Por otro lado, vemos que el cluster 3, está formado por aquellos jugadores que destacan en características relacionadas con habilidades ofensivas como "Dribbling", "Finishing" o "Shot Power" por lo que es el cluster indicado si tratamos de buscar un jugador con características de centrocampista ofensivo, delantero y con valores altos en las características por lo general.

El cluster 4, se encuentra en una zona que no está relacionada con ninguna posición en específico, pero está correlacionado con muchas variables más generales y no tan específicas para alguna posición en concreto. Variables como "Overall", "Reactions", "Short Passing" y "Ball Controll, son algunas de las que más contribución tienen en este cluster. Este cluster estará integrado por jugadores con valores relativamente altos para el conjunto de variables en general y más ofensivos que defensivos.

Por último, en la tercera dimensión el cluster 5 se separa del resto y usando el gráfico de loadings podemos ver que está integrado principalmente por delanteros con características físicas destacadas, como "Strength", "Weight", o "Height". Por lo que puede ser el cluster indicado para buscar un delantero con estas características.

En conclusión, y a pesar de no obtener una clara separación entre los diferentes tipos de jugadores, el análisis clustering puede ser útil para la búsqueda de jugadores con determinadas características. Un usuario del juego, conociendo las variables que mayor contribución tienen a cada cluster, puede reducir la búsqueda a uno de los grupos para encontrar el jugador óptimo con las características que busca.

6 Tercer método - PLS

6.1 Objetivo PLS

Por medio de este estudio buscamos predecir los valores de las variables Value, Wage e International.Reputation. Como vimos en el método PCA, había grandes diferencias entre los jugadores que ocupan la posición de portero y los del resto de posiciones, ya que son posiciones con características bastante diferentes. Por ello, hemos decidido realizar dos análisis PLS diferentes, uno para los porteros y otro para el resto de jugadores de campo. Debido al gran tamaño de los análisis y a que tanto el de

jugadores como porteros son parecidos, hemos decidido centrarnos en explicar el de jugadores, ya que son la mayoría de las filas de nuestra base de datos, y analizar a los porteros en el **Anexo 3 - Implementación del método PLS para porteros.**

6.2 Implementación del método PLS para jugadores

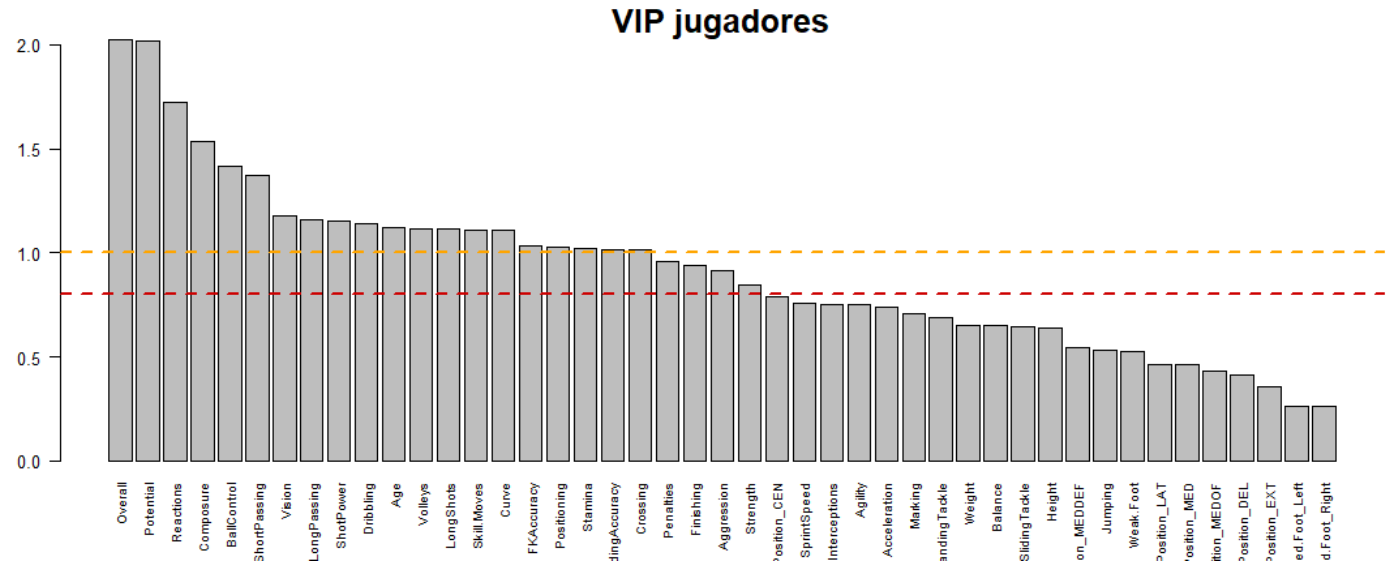
Para poder realizar el análisis sobre los jugadores, seleccionamos únicamente aquellos cuya posición sea diferente a "portero", y realizamos una partición del 75% de los datos para el entrenamiento y 25% para evaluar los resultados.

Además, seleccionamos las variables a predecir, en nuestro caso Value, Wage e International.Reputation, y las variables predictoras, que son las mismas que usamos para el modelo PCA.

6.2.1 Selección de variables PLS jugadores

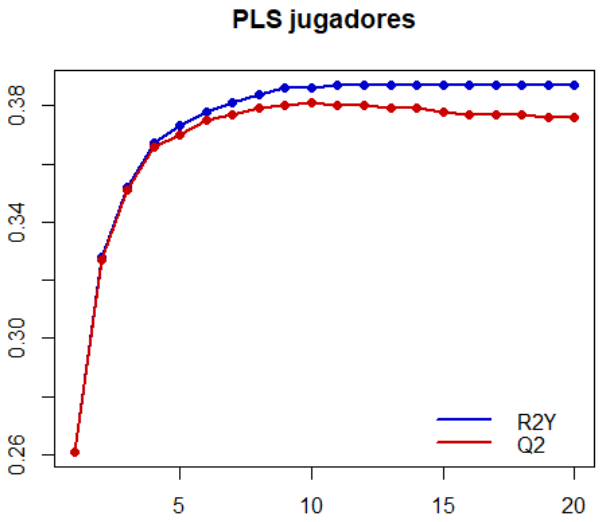
Primero, vamos a seleccionar aquellas variables que sean significativas en el modelo PLS. A lo que esto se refiere es que a pesar de que contemos con 42 regresores, no necesariamente necesitaremos usar todos para conseguir un buen modelo, además que tener menos variables facilitará la interpretación. Por ello, realizamos un modelo PLS **inicial** con los datos de entrenamiento centrados y escalados y usando validación cruzada 10-fold. Escogeremos un número de componentes que aporte unos buenos valores de R^2 y Q^2 (en este caso diez componentes).

Una vez obtenido este modelo inicial, vamos a estudiar los valores de VIP de cada variable para saber la importancia relativa que tiene cada una en el modelo. Como criterio de aceptación, conservaremos sólo aquellas que sean mayores o iguales al valor de VIP de 0.8. Si tienen un valor VIP menor al mencionado, podremos considerar que son irrelevantes y/o estadísticamente no significativas.



En este caso, hay 21 variables que no superan el valor de VIP de 0.8. Tras esto, el número de variables explicativas útiles se queda en 24. Cabe mencionar que para confirmar la no significancia estadística de estas variables eliminadas, lo óptimo habría sido consultar los gráficos de coeficientes PLS de cada variable y generar intervalos de confianza (por ejemplo, de 95%). Estos intervalos se generarían, mediante permutaciones, para los coeficientes de cada variable explicativa, y si dichos intervalos incluyeran el valor 0, podríamos confirmar la no significancia de las variables. No hemos conseguido hacer el código para generar los intervalos de confianza, pero para ver los gráficos de coeficientes, mírese el **Anexo 3 - Coeficientes PLS modelo jugadores**.

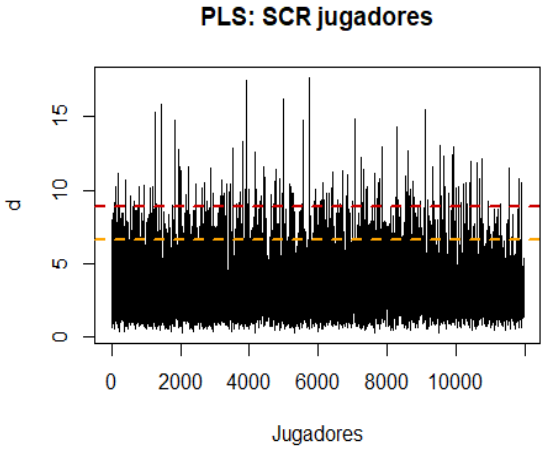
6.2.2 Elección del número de componentes Jugadores

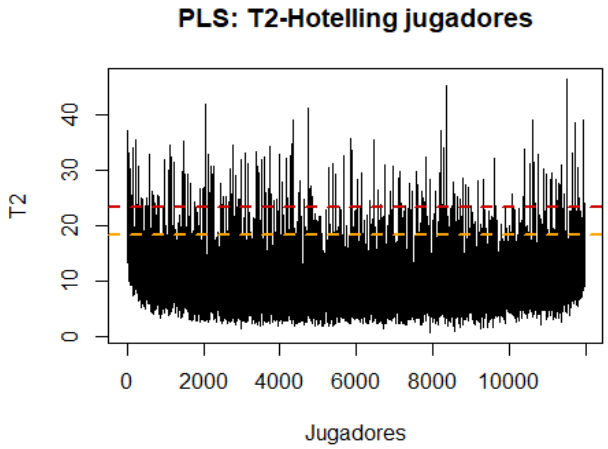


Reajustamos el modelo sin estas variables que no superan el criterio de VIP, de nuevo con validación cruzada 10-fold. Recurriremos de nuevo a R^2 y Q^2 para elegir el número de componentes PLS. Tenemos una situación parecida a la inicial, pero se puede ver ligeramente que con 10 componentes se alcanza el máximo de Q^2, por lo que con ese número de componentes PLS las predicciones deberían ser mejores.

6.2.3 Validación T2 de Hotelling y SCR Jugadores

Antes de interpretar el modelo, vamos a validarlo. Consideraremos por válidos aquellos valores que no superen 2 veces los límites de confianza del 99%. A diferencia del anterior objetivo realizado con PCA y clustering, ahora sí podríamos considerar eliminar algún individuo muy anómalo o atípico, ya que el objetivo es predecir las 3 variables antes mencionadas, y no es necesario mantener todos los jugadores.





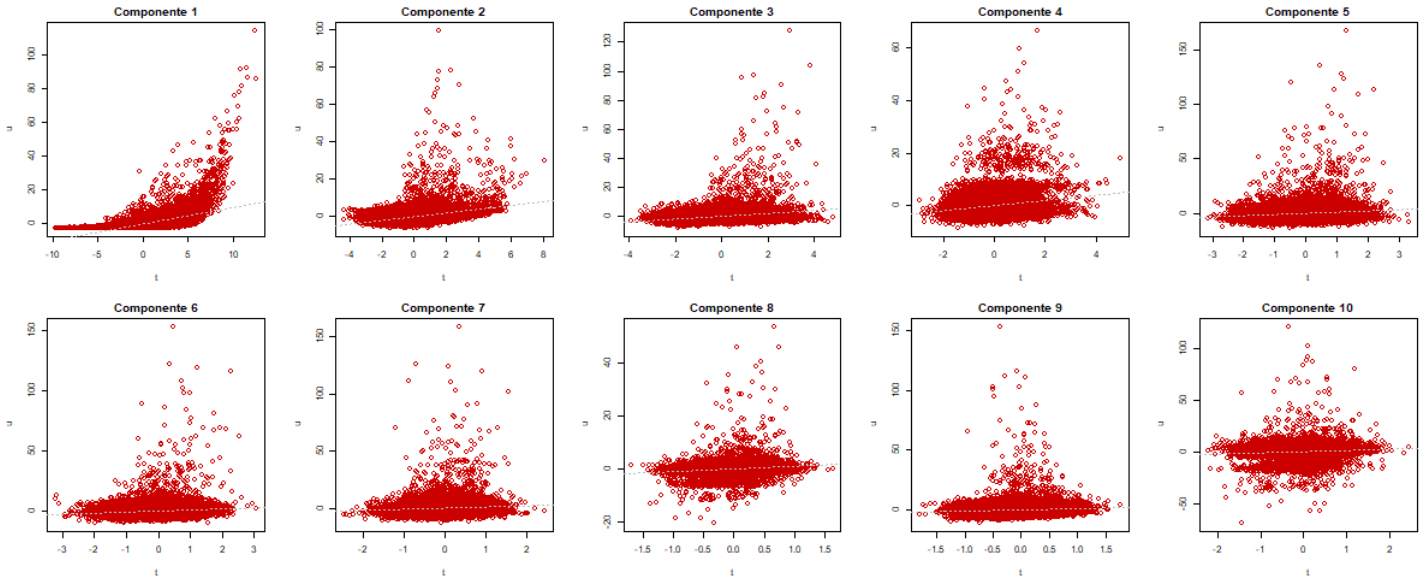
Tras haber eliminado 4 jugadores en dos iteraciones de reajustes del modelo, (ver en el **Anexo 3 - Iteraciones T2 y SCR jugadores**) vemos que hay pocos jugadores que pasen el límite del 95%-99%. Por encima del límite de confianza de 95% tenemos 742 observaciones y por encima del límite de 99% tenemos 234 observaciones, que respectivamente son el 0.062% y el 0.02%. Respecto a la suma de cuadrados residual, el 0.052% pasan el primer límite y el 0.014% pasan el segundo. Por ello, consideramos que el modelo está validado.

6.2.4 Interpretación del modelo PLS Jugadores

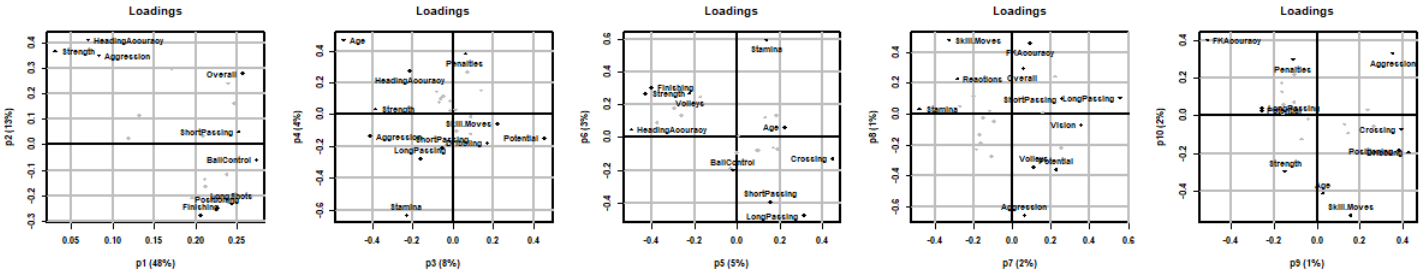
Si observamos la relación entre los *scores* de los jugadores, en la primera componente vemos que hay una tendencia lineal donde se encuentran la mayoría de jugadores, pero se acaba convirtiendo en exponencial, donde se concentran menos valores. Esta exponencialidad puede ser debida a los jugadores anómalos, que por lo que hemos estado estudiando en PCA, parece que tienden a ser los jugadores más importantes

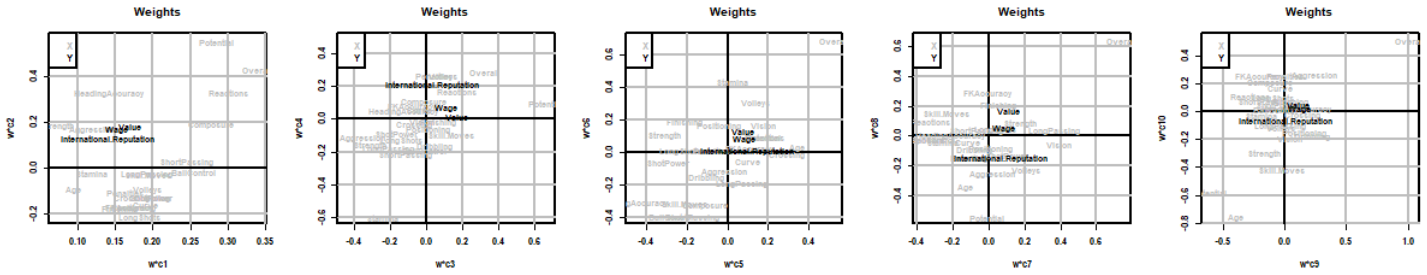
o que mejor juegan en comparación a la gran mayoría de los demás. Como es de esperar, la primera componente es la que más relación lineal manifiesta, en este caso de 0.565. Las otras 9 componentes, por orden, tienen correlaciones lineales de 0.3457, 0.2455, 0.2085, 0.1094 0.1202, 0.079, 0.1557, 0.0744 y 0.0843, en la figura de abajo se pueden visualizar. Llama la atención que el gráfico anterior de R^2 y Q^2 marcara este número de componentes como el mejor, ya que las correlaciones de las últimas componentes son prácticamente nulas. De todas formas, decidimos continuar así.

Cabe destacar que seguramente estos resultados tan poco precisos se deban a la relación exponencial mencionada. Para arreglarlo, podríamos recurrir al uso de transformaciones en los datos, repitiéndo el análisis pero en vez de hacer el modelo con unas variables de respuesta 'Y', aplicaríamos logaritmos para que quedasen de la forma 'ln(Y)'. Es una tarea compleja en la que no indagaremos, pero que podrá ser desarrollada en el futuro.



Si generamos los gráficos de *loadings* y de *weightings* (y hacemos zoom), podemos interpretar las relaciones entre los regresores y las variables de respuesta:

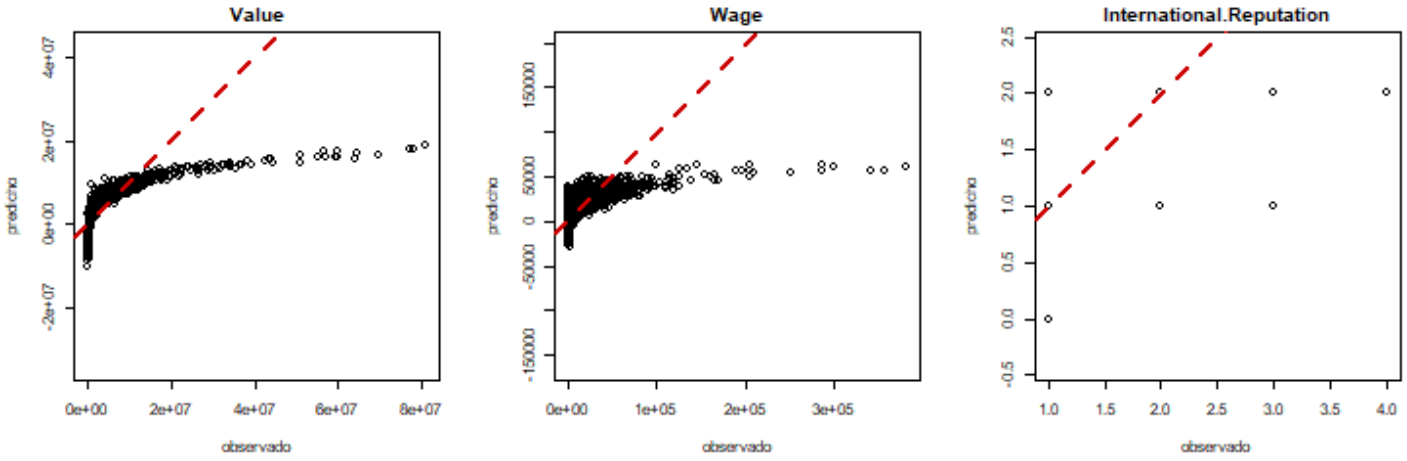




Como se podía esperar sabiendo las correlaciones de los *scores* de antes, los únicos gráficos de weightings que nos pueden dar una información mínimamente valiosa son el de las componentes 1 y 2 y el de las componentes 3 y 4. Lo que se puede observar es que las tres variables de respuesta están siempre juntas, lo que significa que están correlacionadas positivamente entre ellas, pero tampoco mucho, ya que no se encuentran en la periferia. Como hemos mencionado antes, si aplicáramos la transformación logarítmica al modelo, es posible que los resultados y la interpretabilidad mejorasen.

6.2.5 Predicciones para test jugadores

Finalmente, realizamos las predicciones con nuestros datos de test. Como habíamos observado antes, el modelo consigue ajustar en un principio a los datos, pero luego se desvía bastante respecto a los jugadores con valores altos. Como decíamos antes, puede ser debido a que el modelo se ajusta bastante a la mayoría, que son los jugadores mediocres, no consiguiendo bien predecir el valor de los jugadores importantes, que son menos.



7 Conclusión

7.1 Comparación entre los métodos aplicados

La implementación de PCA nos ha ayudado a encontrar y analizar relaciones entre las variables y entre los jugadores de nuestra base de datos. Nos ha permitido detectar las variables que contribuyen a una anomalía desmesurada de algunos jugadores con respecto a la gran mayoría del resto. El descubrimiento de la fuerte relación entre las variables (sobre todo entre las de habilidades de jugadores) y la identificación de las variables Wage, Value e International.Reputation como los principales causantes de anomalías ha sido clave para el resto del trabajo.

Por otro lado, el clustering nos ha permitido encontrar agrupaciones de jugadores parecidos, independientemente de la posición de campo a la que pertenecen en un principio. Enlazando con PCA, hemos sacado agrupamientos mediante las variables latentes y la suma de cuadrados residual del modelo en vez de directamente con las variables originales. Contar con las componentes principales ha que la fuerte relación de las mismas dificultaba el sacar resultados medianamente decentes y nuestro objetivo era encontrar grupos mínimamente diferenciados de jugadores con características parecidas para poder ser reemplazados en caso de lesión.

Finalmente, PLS nos ha permitido relacionar las antes mencionadas variables Wage, Value e Importational.Reputation con el resto de las variables y predecirlas aprovechando dicha relación.

7.2 Discusión de los métodos no aplicados

En nuestro caso, en primer lugar, nos hemos decantado por usar PCA en vez de MCA debido a que nuestras variables son numéricas. Podríamos haber categorizado las variables para utilizar el método MCA pero es una tarea muy compleja que no vale la pena.

En segundo lugar, aunque podríamos haber utilizado reglas de asociación, hemos considerado que no nos iban a aportar mejores soluciones a nuestros objetivos que los métodos realizados, por lo que decidimos no implementarlos.

Finalmente, no hemos aplicado ni análisis discriminante ni PLS-DA porque nuestras variables de respuesta eran de tipo numérico, no categórico.

7.3 Conclusión final

Tras estos tres análisis, hemos podido obtener una serie de conclusiones con relación a nuestros objetivos iniciales: Mediante PCA y clustering hemos creado una referencia no solo para jugadores de FIFA ya experimentados, sino también para jugadores novatos que buscan crearse un equipo ya sea por poco conocimiento del juego o simplemente por la diversión de jugar con un equipo generado de forma aleatoria.

Implementar PLS nos ha permitido crear un modelo capaz de predecir el precio, el valor y la reputación internacional para todos los jugadores del videojuego a partir del resto de variables del estudio. Cabe destacar que la presencia de tantos jugadores mediocres hace que se infravaloren tanto el precio como el valor y la reputación de los jugadores.

Este proyecto lo consideramos útil para aquellas personas fanáticas del videojuego que buscan tanto crear los mejores equipos para competir, buscar jugadores sustitutos y/o estimar el valor de su propio equipo ya formado. También podría funcionar como complemento en los equipos de la vida real, siempre y cuando las variables de la base de datos sean verosímiles.

8 Anexo 0

Descripción de la base de datos:

- **IDENTIFICADOR:**
  - **Name:** nombre del jugador (lo usaremos para identificar a los jugadores).
- **NUMÉRICAS:**
  - **Height:** altura (cm).
  - **Weight:** peso (kg).
  - **Crossing:** precisión en los centros.
  - **Finishing:** capacidad de convertir (meter) un gol.
  - **HeadingAccuracy:** precisión y facilidad al golpear de cabeza.
  - **ShortPassing:** pases cortos.
  - **Volleys:** remates acrobáticos con los pies.
  - **Dribbling:** superar rivales regateando.
  - **Curve:** capacidad de que la pelota cambie de dirección durante la trayectoria.
  - **FKAccuracy:** precisión en los tiros libres.
  - **LongPassing:** pases largos.
  - **BallControl:** control de balón.
  - **Acceleration:** velocidad en las primeras zancadas.
  - **SprintSpeed:** velocidad máxima en carrera.
  - **Agility:** agilidad con la pelota. Giros y movimientos rápidos y precisos.
  - **Reactions:** velocidad de movimientos con el balón.
  - **Balance:** equilibrio aguantando el contacto de otros jugadores.
  - **ShotPower:** potencia imprimida al golpear la pelota.
  - **Jumping:** capacidad de salto para imponerse en balones aéreos.
  - **Stamina:** fortaleza física.
  - **Strength:** fuerza ante el contacto.
  - **LongShots:** disparos desde fuera del área.
  - **Aggression:** fuerza con la que va a por la pelota.
  - **Interceptions:** posicionamiento en defensa para cortar centros y pases.
  - **Positioning:** desmarques o colocación para recibir el balón.
  - **Vision:** capacidad de ver huecos, anticipar desmarques y aprovecharlos.
  - **Penalties:** acierto en los penaltis.
  - **Composure:** compostura, le afecta menos la presión del defensor.
  - **Marking:** capacidad de contener a un atacante.
  - **StandingTackle:** facilidad para quitar la pelota sin hacer una entrada agresiva.
  - **SlidingTackle:** facilidad para quitar la pelota haciendo entradas agresivas.
  - **GKDiving:** estirada, más posibilidad de interceptar tiros colocados.
  - **GKHandling:** habilidad para detener y retener la pelota con claridad y facilidad.
  - **GKKicking:** precisión y alcance de un saque de portería.
  - **GKPositioning:** posicionamiento de un portero.
  - **GKReflexes:** reflejos ante disparos a bocajarro.

NOTA: todas las variables numéricas excepto edad, valor, sueldo, altura y peso, son puntuaciones que se les otorgan a los jugadores con valores posibles entre 0-99. **NOTA 2:** las variables GKDiving, GKHandling, GKKicking, CKPositioning y GKReflexes son habilidades que corresponden únicamente a los jugadores con posición portero (POR).

- **ORDINALES:**
  - **International.Reputation:** reputación a nivel internacional.
  - **Weak.Foot:** nivel de su pie débil.
  - **Skill.Moves:** puntuación en cuanto a skills (habilidades). **NOTA:** son variables ordinales codificadas del 1-5, con nota 1 si tienen mala puntuación y nota 5 si tienen buena puntuación en cada variable.
- **CATEGÓRICAS:**
  - **Nationality:** nacionalidad del jugador.
  - **Club:** equipo del jugador.
  - **Preferred.Foot:** pie dominante del jugador, codificada como binaria: **Preferred.Foot_Left** (zurdo), **Preferred.Foot_Right** (diestro).
  - **Position:** posición del jugador. Demasiados valores posibles. Las ocho categorías en las que hemos decidido agrupar a los jugadores son las que hemos considerado más comunes en el fútbol actual, considerando centrales, delanteros, extremos, laterales, mediocentros, mediocentros defensivos, mediocentros ofensivos y porteros. Hemos tomado la semejanza entre la posición asignada en el fifa del jugador y una de las elegidas como criterio, por ejemplo hemos asignado a los carrileros como laterales, o a los segundos delanteros como delanteros, ya que es la posición a la que más se asemejan. Codificada como *dummy* (**Position.CEN**, **Position.DEL**, **Position.EXT**, **Position.LAT**, **Position.MED**, **Position.MEDDEF**, **Position.MEDOF**, **Position.POR**)

8.1 Pre-procesado de los datos

```
datos = read.csv("/Users/porti/OneDrive/Escritorio/MDP I/Trabajo/fifa_original.csv",
encoding = 'UTF-8', na.strings=c("","na"))
#Los datos faltantes estaban escritos como espacios en blanco, por lo que los sustituimos
por NA para tratarlos con mayor facilidad
```

8.2 Eliminación de variables inútiles

Eliminaremos las variables que no serán utilizadas en el análisis.

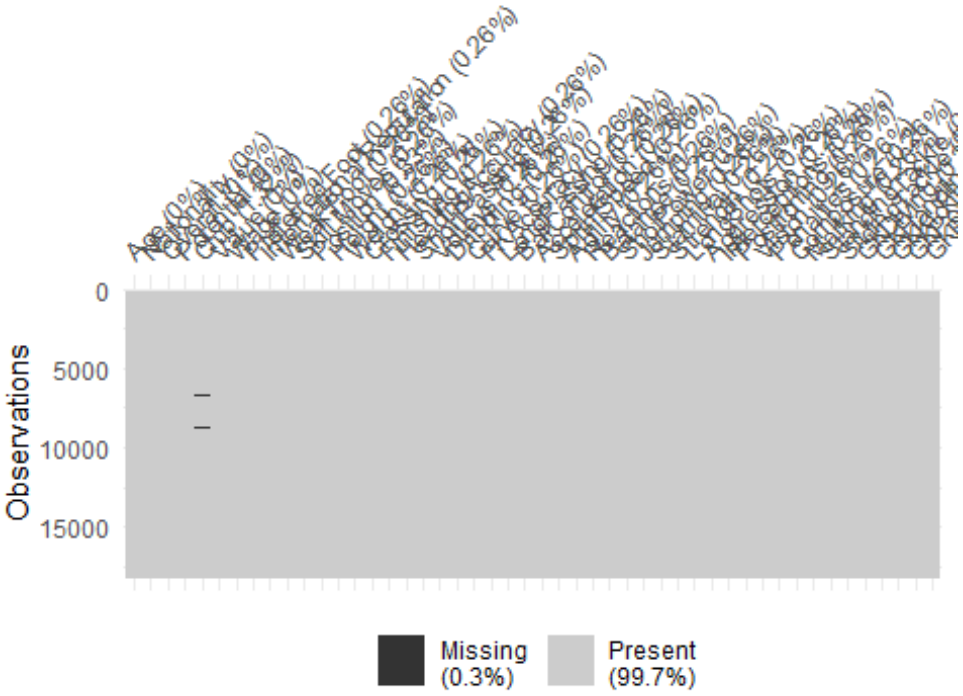
```
fifa = datos[,-c(1,2,3,5,7,11,14,19:21,23:26,29:54,89)]
fifa_paralelo = datos[,-c(1,2,5,7,11,14,19:21,23:26,29:54,89)]
```

8.3 Tratamiento de datos faltantes

Para el tratamiento de datos faltantes, realizamos una primera exploración de los datos. Obtenemos el porcentaje de datos faltates existente en cada columna y observamos una baja cantidad de datos faltantes. Respecto al numero de datos faltantes, hay 241 jugadores con valor faltante en la variable Club y 60 jugadores con valor faltante en la variable Posición. Como vemos en los porcentajes, el 0,26% de los jugadores de la base de datos no tienen valores sobre las variables de habilidades. Al tratarse de un grupo tan reducido, decidimos eliminarlos directamente.

Véase a continuación el porcentaje de valores faltantes para cada variable:

```
vis_miss(fifa, warn_large_data =  FALSE)
```




```
sum(is.na(fifa))

## [1] 2221

#gg_miss_upset(fifa)
#gg_miss_upset(fifa, nsets = n_var_miss(fifa))
#gg_miss_upset(fifa,
 #             nsets = n_var_miss(fifa),
  #            nintersects = NA)
#gg_miss_var(fifa)

fifa <- na.omit(fifa)  #eliminar los faltantes
```

8.4 Limpieza y asignación de los tipos de variables adecuados

Hay variables que no están en su clase correspondiente, y algunas que hay que modificar para facilitar su tratamiento. Eliminamos simbolo € de las variables Wage y Value, las transformamos a tipo numerico y pasamos a numerico el peso y la altura también.

```
#Eliminamos el simbolo de € de la variable value
fifa$Value = gsub("^\\€", "", fifa$Value)

#Eliminamos el simbolo de € de la variable wage
fifa$Wage = gsub("^\\€", "", fifa$Wage)

#Transformamos el texto en número de acuerdo a si es M€, K€ o €.
numeric_value = function(char){
      num = as.numeric(substring(char, 1, nchar(char)-1))
      k = substring(char,nchar(char),nchar(char))
      mult = 1
```

```
      if(k=='M'){mult = 1000000}
      if(k=='K'){mult = 1000}
      num*mult
}
fifa$Value = sapply(fifa$Value, numeric_value)
fifa$Wage = sapply(fifa$Wage, numeric_value)

#La variable de la altura del jugador está medida en pies más pulgadas. La transformamos
a valor numérico expresado en cms.

fifa$Height = as.numeric(gsub("\'\\w+$", "", fifa$Height))*30.48 +
      as.numeric(gsub("^\\w+'", "", fifa$Height))*2.54

#La variable Weight representa el peso del jugador en libras. La transformamos a valor
numérico expresado en kgs.

fifa$Weight = as.numeric(gsub("lbs$", "", fifa$Weight))*0.453592



#La variable Preferred.Foot se pasa a tipo binario.

library(fastDummies)
fifa = dummy_cols(fifa, select_columns = 'Preferred.Foot', remove_selected_columns =
TRUE)
```

La variable Position se recodifica. Tiene 27 valores diferentes y vamos a convertirlos en ocho (portero, lateral, central, medio, medio ofensivo, medio defensivo, extremo y delantero), agrupándolos por las categorías más importantes para facilitar la interpretación de las variables. Después se convierte a binario.

```
for (i in 1:nrow(fifa)){
  if(fifa$Position[i] %in% c('CF', 'ST', 'RF', 'LF', 'LS', 'RS')){fifa$Position[i]='DEL'}
  if(fifa$Position[i] %in% c('RW', 'LW')){fifa$Position[i]='EXT'}
  if(fifa$Position[i] %in% c('LM', 'CM', 'LCM', 'RM', 'RCM')){fifa$Position[i]='MED'}
  if(fifa$Position[i] %in% c('LAM', 'CAM', 'RAM')){fifa$Position[i]='MEDOF'}
  if(fifa$Position[i] %in% c('LDM', 'CDM', 'RDM')){fifa$Position[i]='MEDDEF'}
  if(fifa$Position[i] %in% c('LCB', 'CB', 'RCB')){fifa$Position[i]='CEN'}
  if(fifa$Position[i] %in% c('LWB', 'LB', 'RB', 'RWB')){fifa$Position[i]='LAT'}
  if(fifa$Position[i] %in% c('GK')){fifa$Position[i]='POR'}
}


fifa = dummy_cols(fifa, select_columns = 'Position', remove_selected_columns = FALSE)

#table(fifa$Position)
```

Creamos dataframe con el tipo de cada variable para facilitar la implementación de los métodos más adelante.

```
descFifa = data.frame("variable" = colnames(fifa),
                      "tipo" = c("numerical", "categorical",
                                 rep("numerical", 2), "categorical",
```

```
                                  rep("numerical", 5), "categorical",
                                  rep("numerical", 46)), stringsAsFactors = FALSE)
rownames(descFifa) = descFifa$variable
```

8.5 Análisis exploratorio

A continuación, haremos un conciso análisis exploratorio de nuestras variables para ir familiarizándonos con ellas. Estudiamos las distribuciones de cada variable para encontrar posibles datos anómalos y para decidir si será necesario estandarizar todas las variables numéricas de cara a la implementación del PCA más adelante.

```
summary(fifa[,descFifa$variable[descFifa$tipo == "numerical"]])
```

```
##       Age             Overall        Potential         Value
##  Min.   :16.00   Min.   :46.00   Min.   :48.00   Min.   :    10000
##  1st Qu.:21.00   1st Qu.:62.00   1st Qu.:67.00   1st Qu.:   325000
##  Median :25.00   Median :66.00   Median :71.00   Median :   700000
##  Mean   :25.11   Mean   :66.24   Mean   :71.33   Mean   :  2450133
##  3rd Qu.:28.00   3rd Qu.:71.00   3rd Qu.:75.00   3rd Qu.:  2100000
##  Max.   :45.00   Max.   :94.00   Max.   :95.00   Max.   :118500000
##                                                  NA's   :11
##       Wage        International.Reputation   Weak.Foot      Skill.Moves
##  Min.   :  1000   Min.   :1.000            Min.   :1.000   Min.   :1.000
##  1st Qu.:  1000   1st Qu.:1.000            1st Qu.:3.000   1st Qu.:2.000
##  Median :  3000   Median :1.000            Median :3.000   Median :2.000
##  Mean   :  9884   Mean   :1.114            Mean   :2.947   Mean   :2.363
##  3rd Qu.:  9000   3rd Qu.:1.000            3rd Qu.:3.000   3rd Qu.:3.000
##  Max.   :565000   Max.   :5.000            Max.   :5.000   Max.   :5.000
##
##      Height          Weight          Crossing       Finishing
##  Min.   :154.9   Min.   : 49.90   Min.   : 5.00   Min.   : 2.00
##  1st Qu.:175.3   1st Qu.: 69.85   1st Qu.:38.00   1st Qu.:30.00
##  Median :180.3   Median : 74.84   Median :54.00   Median :49.00
##  Mean   :181.3   Mean   : 75.28   Mean   :49.75   Mean   :45.58
##  3rd Qu.:185.4   3rd Qu.: 79.83   3rd Qu.:64.00   3rd Qu.:62.00
##  Max.   :205.7   Max.   :110.22   Max.   :93.00   Max.   :95.00
##
##  HeadingAccuracy  ShortPassing      Volleys        Dribbling
##  Min.   : 4.0    Min.   : 7.00   Min.   : 4.00   Min.   : 4.00
##  1st Qu.:44.0    1st Qu.:54.00   1st Qu.:30.00   1st Qu.:49.00
##  Median :56.0    Median :62.00   Median :44.00   Median :61.00
##  Mean   :52.3    Mean   :58.71   Mean   :42.93   Mean   :55.41
##  3rd Qu.:64.0    3rd Qu.:68.00   3rd Qu.:57.00   3rd Qu.:68.00
##  Max.   :94.0    Max.   :93.00   Max.   :90.00   Max.   :97.00
##
##      Curve         FKAccuracy     LongPassing     BallControl     Acceleration
##  Min.   : 6.00   Min.   : 3.00   Min.   : 9.00   Min.   : 5.00   Min.   :12.0
##  1st Qu.:34.00   1st Qu.:31.00   1st Qu.:43.00   1st Qu.:54.00   1st Qu.:57.0
##  Median :49.00   Median :41.00   Median :56.00   Median :63.00   Median :67.0
##  Mean   :47.22   Mean   :42.88   Mean   :52.72   Mean   :58.41   Mean   :64.6
##  3rd Qu.:62.00   3rd Qu.:57.00   3rd Qu.:64.00   3rd Qu.:69.00   3rd Qu.:75.0
##  Max.   :94.00   Max.   :94.00   Max.   :93.00   Max.   :96.00   Max.   :97.0
##
##   SprintSpeed       Agility        Reactions        Balance
```

```
##  Min.   :12.00   Min.   :14.00   Min.   :21.00   Min.   :16.00
##  1st Qu.:57.00   1st Qu.:55.00   1st Qu.:56.00   1st Qu.:56.00
##  Median :67.00   Median :66.00   Median :62.00   Median :66.00
##  Mean   :64.72   Mean   :63.52   Mean   :61.82   Mean   :63.96
##  3rd Qu.:75.00   3rd Qu.:74.00   3rd Qu.:68.00   3rd Qu.:74.00
##  Max.   :96.00   Max.   :96.00   Max.   :96.00   Max.   :96.00
##
##    ShotPower        Jumping         Stamina         Strength
##  Min.   : 2.00   Min.   :15.00   Min.   :12.00   Min.   :17.00
##  1st Qu.:45.00   1st Qu.:58.00   1st Qu.:56.00   1st Qu.:58.00
##  Median :59.00   Median :66.00   Median :66.00   Median :67.00
##  Mean   :55.49   Mean   :65.12   Mean   :63.21   Mean   :65.32
##  3rd Qu.:68.00   3rd Qu.:73.00   3rd Qu.:74.00   3rd Qu.:74.00
##  Max.   :95.00   Max.   :95.00   Max.   :96.00   Max.   :97.00
##
##    LongShots       Aggression    Interceptions    Positioning     Vision
##  Min.   : 3.00   Min.   :11.00   Min.   : 3.00   Min.   : 2   Min.   :10.00
##  1st Qu.:33.00   1st Qu.:44.00   1st Qu.:26.00   1st Qu.:39   1st Qu.:44.00
##  Median :51.00   Median :59.00   Median :52.00   Median :55   Median :55.00
##  Mean   :47.13   Mean   :55.88   Mean   :46.69   Mean   :50   Mean   :53.45
##  3rd Qu.:62.00   3rd Qu.:69.00   3rd Qu.:64.00   3rd Qu.:64   3rd Qu.:64.00
##  Max.   :94.00   Max.   :95.00   Max.   :92.00   Max.   :95   Max.   :94.00
##
##    Penalties       Composure        Marking      StandingTackle
##  Min.   : 5.00   Min.   : 3.00   Min.   : 3.00   Min.   : 2.00
##  1st Qu.:39.00   1st Qu.:51.00   1st Qu.:30.00   1st Qu.:27.00
##  Median :49.00   Median :60.00   Median :53.00   Median :55.00
##  Mean   :48.54   Mean   :58.66   Mean   :47.26   Mean   :47.68
##  3rd Qu.:60.00   3rd Qu.:67.00   3rd Qu.:64.00   3rd Qu.:66.00
##  Max.   :92.00   Max.   :96.00   Max.   :94.00   Max.   :93.00
##
##  SlidingTackle      GKDiving       GKHandling      GKKicking
##  Min.   : 3.00   Min.   : 1.00   Min.   : 1.00   Min.   : 1.00
##  1st Qu.:24.00   1st Qu.: 8.00   1st Qu.: 8.00   1st Qu.: 8.00
##  Median :52.00   Median :11.00   Median :11.00   Median :11.00
##  Mean   :45.64   Mean   :16.59   Mean   :16.37   Mean   :16.21
##  3rd Qu.:64.00   3rd Qu.:14.00   3rd Qu.:14.00   3rd Qu.:14.00
##  Max.   :91.00   Max.   :90.00   Max.   :92.00   Max.   :91.00
##
##  GKPositioning     GKReflexes    Preferred.Foot_Left Preferred.Foot_Right
##  Min.   : 1.00   Min.   : 1.00   Min.   :0.0000      Min.   :0.0000
##  1st Qu.: 8.00   1st Qu.: 8.00   1st Qu.:0.0000      1st Qu.:1.0000
##  Median :11.00   Median :11.00   Median :0.0000      Median :1.0000
##  Mean   :16.36   Mean   :16.68   Mean   :0.2323      Mean   :0.7677
##  3rd Qu.:14.00   3rd Qu.:14.00   3rd Qu.:0.0000      3rd Qu.:1.0000
##  Max.   :90.00   Max.   :94.00   Max.   :1.0000      Max.   :1.0000
##
##   Position_CEN     Position_DEL     Position_EXT      Position_LAT
##  Min.   :0.0000   Min.   :0.0000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.0000   Median :0.0000   Median :0.00000   Median :0.0000
##  Mean   :0.1698   Mean   :0.1474   Mean   :0.04124   Mean   :0.1528
##  3rd Qu.:0.0000   3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.0000   Max.   :1.00000   Max.   :1.0000
```

```
##
##   Position_MED    Position_MEDDEF   Position_MEDOF     Position_POR
##  Min.   :0.0000   Min.   :0.00000   Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.0000   1st Qu.:0.00000   1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.0000   Median :0.00000   Median :0.00000   Median :0.0000
##  Mean   :0.2429   Mean   :0.07931   Mean   :0.05525   Mean   :0.1112
##  3rd Qu.:0.0000   3rd Qu.:0.00000   3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.0000   Max.   :1.00000   Max.   :1.00000   Max.   :1.0000
##
```

```
table(fifa$Position)
```

```
##
##    CEN    DEL    EXT    LAT    MED MEDDEF  MEDOF    POR
##   3043   2642    739   2738   4353   1421    990   1992
```

```
table(fifa$International.Reputation)
```

```
##
##     1     2     3     4     5
## 16305  1248   308    51     6
```

```
table(fifa$Weak.Foot)
```

```
##
##     1     2     3     4     5
##   153  3715 11201  2622   227
```

```
table(fifa$Skill.Moves)
```

```
##
##    1    2    3    4    5
## 1992 8443 6522  911   50
```

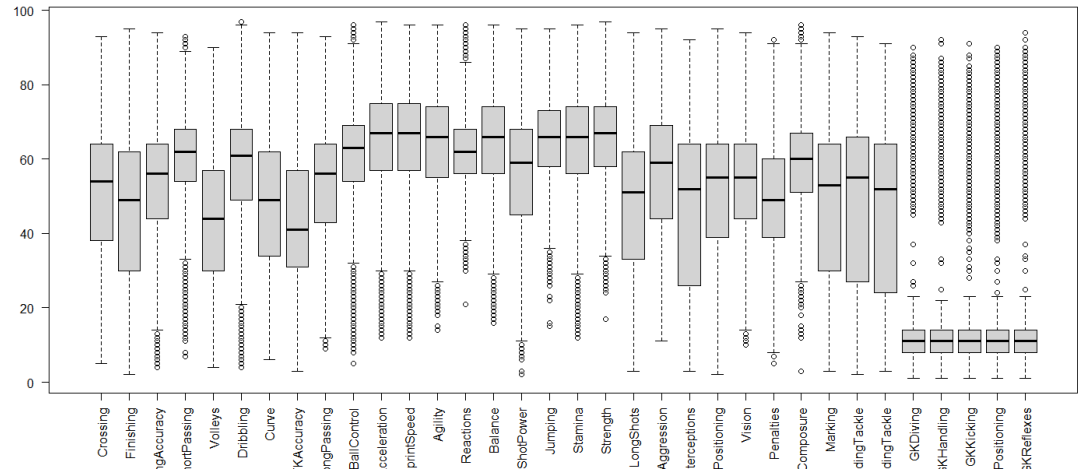
8.5.1 Variables de habilidades de jugador

En el siguiente gráfico tenemos una serie de boxplots de las variables de habilidades de los jugadores, todas ellas medidas en una misma escala de 0 a 100. Podemos ver una clara diferenciación entre todas la variables menos las cinco últimas de la derecha. La diferencia de este pequeño grupo de variables se debe a que son medidas pensadas para los porteros, por lo que todos los demás jugadores que no lo sean tendrán valores bajos en dichas variables.

En general, suponemos (sería lógico) que la gran mayoría de los datos atípicos que aparecen en cada boxplot son los porteros.

```
fifa_numericas = fifa[,setdiff(colnames(fifa), c("Nationality", "Club"))]

fifa_stats = fifa_numericas[,setdiff(colnames(fifa_numericas), c("Preferred.Foot_Right",
"Preferred.Foot_Left", "Age", "Wage", "Value",

"International.Reputation", "Skill.Moves", "Height", "Weight", "Weak.Foot",
                                                              "Overall",
"Potential","Position_DEL","Position_POR","Position_MED","Position_LAT",

"Position_MEDDEF","Position_MEDOF","Position_EXT","Position_CEN","Position"))]
```

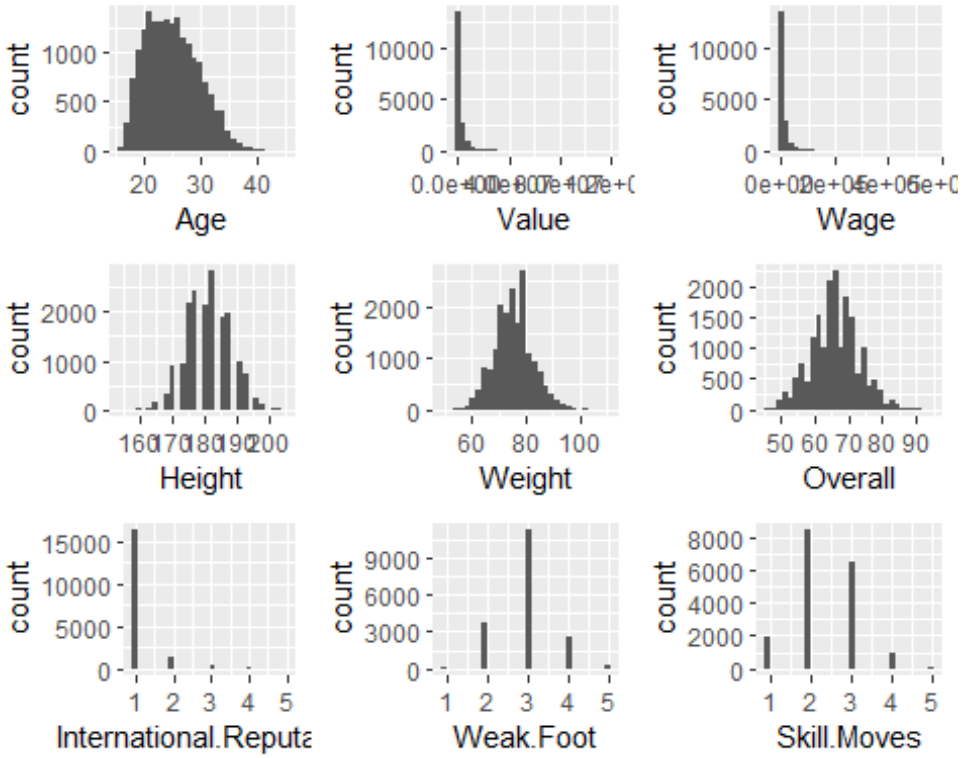
```
boxplot(fifa_stats, las = 2)
```



Pero, como sabemos, estas no son las únicas variables que se considerarán en el estudio. Tenemos otras más de tipo numérico, pero no están medidas de la misma forma que las mostradas en el gráfico anterior.

Realizamos unos histogramas del resto de variables que no corresponden a stats de los jugadores. Saltan a la vista las variables Value, Wage, International Reputation, cuentan con una fuerte asímetría positiva, es bastante seguro que tengamos que eliminar bastantes valores debido a esto. La variable Edad también es parecida en este sentido, pero no tanto como las anteriores. Por las demás, muestran una distribución normal.

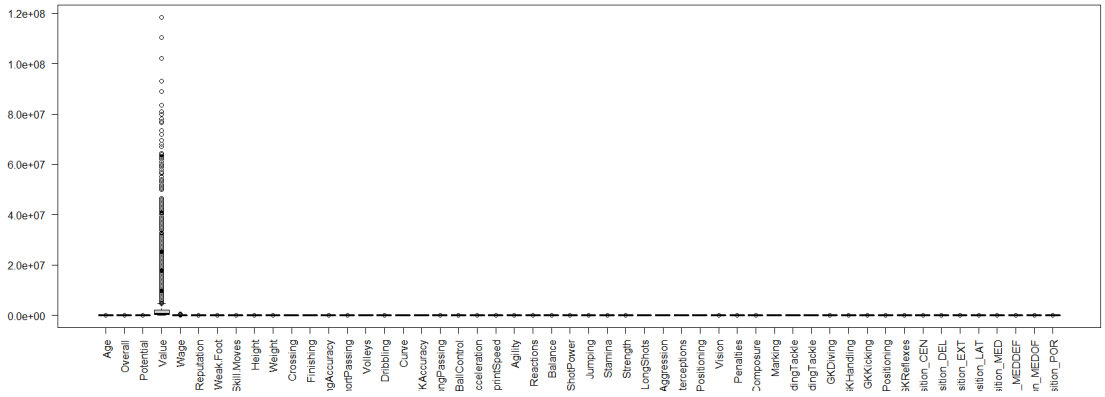
```
#par(mfrow = c(1, 3))

hist1 = pl <- ggplot(fifa, aes(x=Age))
hist1 = pl + geom_histogram(bins=30)
hist2 = pl <- ggplot(fifa, aes(x=Value))
hist2 = pl + geom_histogram(bins=30)
hist3 = pl <- ggplot(fifa, aes(x=Wage))
hist3 = pl + geom_histogram(bins=30)
hist4 = pl <- ggplot(fifa, aes(x=Height))
hist4 = pl + geom_histogram(bins=30)
hist5 = pl <- ggplot(fifa, aes(x=Weight))
hist5 = pl + geom_histogram(bins=30)
hist6 = pl <- ggplot(fifa, aes(x=Overall))
hist6 = pl + geom_histogram(bins=30)
hist7 = pl <- ggplot(fifa, aes(x=International.Reputation))
hist7 = pl + geom_histogram(bins=30)
hist8 = pl <- ggplot(fifa, aes(x=Weak.Foot))
hist8 = pl + geom_histogram(bins=30)
hist9 = pl <- ggplot(fifa, aes(x=Skill.Moves))
hist9 = pl + geom_histogram(bins=30)

ggarrange(hist1,hist2,hist3,hist4,hist5,hist6,hist7,hist8,hist9)
```




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

Véase ahora un gráfico de boxplots similar al primero, pero con todo el conjunto de variables. Claramente necesitaremos tipificar todas las variables para una posible comparación.

```
fifa_numericas = fifa[,setdiff(colnames(fifa), c("Nationality", "Club", "Position",
"Preferred.Foot_Right", "Preferred.Foot_Left"))]

boxplot(fifa_numericas, las = 2)
```

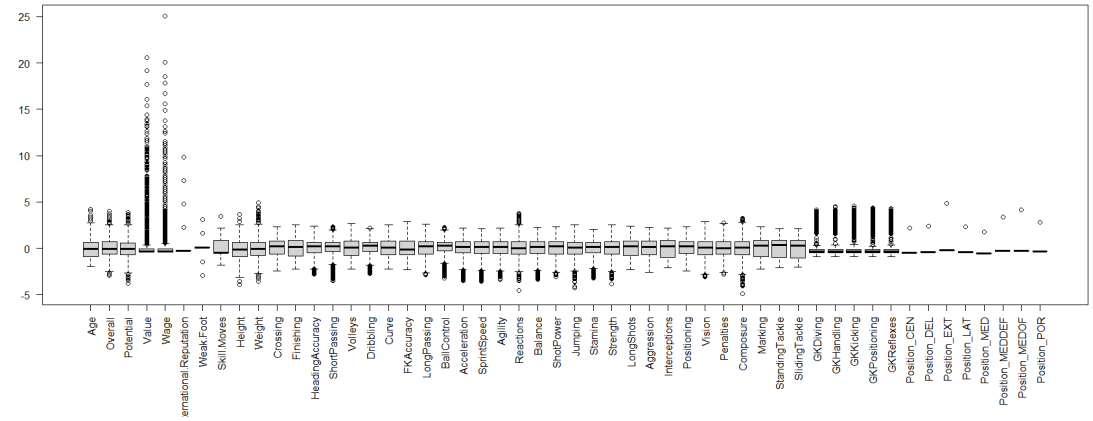


8.5.2 Conclusión - Centrado y escalado

Ya con los datos centrados y escalados mediante varianza unitaria, repetimos el gráfico. Tal y como sospechábamos antes, Value y Wage tienen una gran cantidad de valores fuertemente anómalos. Esto nos lleva a preguntarnos si deberíamos excluir estas variables para a la hora de hacer nuestros dos primeros

métodos en este trabajo: PCA y clustering. Intentaremos hacer el análisis PCA incluyéndolas e intentando tratar valores que salgan muy anómalos, si es que lo hacen.

```
fifa_numericas = fifa[,setdiff(colnames(fifa), c("Nationality", "Club", "Position",
"Preferred.Foot_Right", "Preferred.Foot_Left"))]

fifa_numericas <- scale(fifa_numericas, center = T, scale = T)
par(mar = c(9,4,2,2))
boxplot(fifa_numericas, las = 2)
```

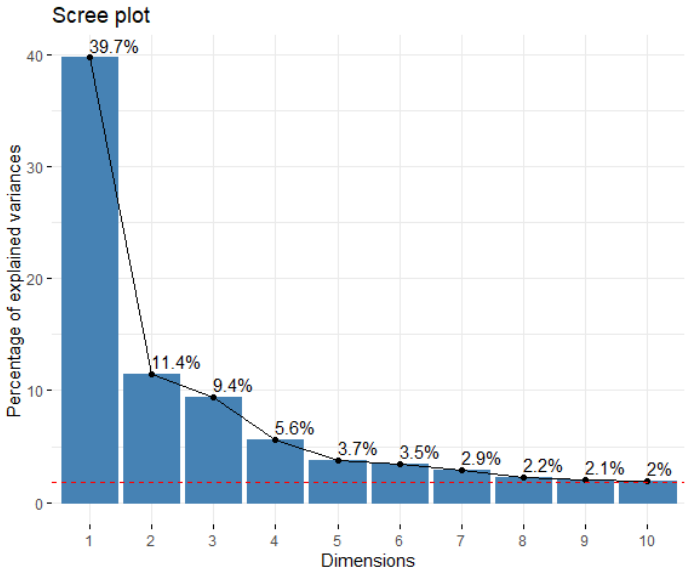


9 Anexo 1

9.1 PCA inicial

9.1.1 Elección del número de componentes

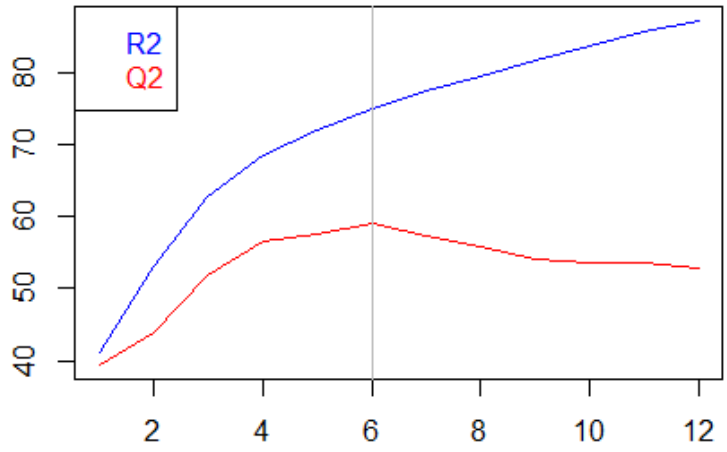
Realizamos un PCA con todas nuestras variables numéricas tipificadas, incluyendo a las categóricas como auxiliares en el modelo. Primero, elegiremos el número de componentes que queremos para explicar nuestros datos.



Consultando el scree plot, con 6 componentes principales explicaríamos el 73.3% de la variabilidad de nuestros datos. Por la alta cantidad de datos anómalos que tenemos, raíz del sospechoso comportamiento de variables como Value y Wage que hemos detectado en el análisis exploratorio anterior, y por la posible presencia de sesgo en las variables de habilidades de los jugadores, parece una cantidad razonable de inercia explicada si escogemos más componentes podríamos estar explicando ruido, o por lo menos más aún del que podrían estar explicando con 6.

Decimos que las variables de habilidades pueden estar sesgadas ya que sus valores están en una escala de 0 a 100, pero al parecer no se basan en un cálculo matemático objetivo.

Para ayudarnos a confirmar el número de componentes principales que nos conviene, recurriremos a la validación cruzada para obtener los estadísticos R^2 y Q^2: El primer estadístico mide la bondad de ajuste del modelo, a medida que obtengamos más variables, esperaremos que R^2 crezca, pero usar únicamente este estadístico no es una buena idea por lo que comentábamos antes sobre el ruido. Con el segundo estadístico obtendremos una visualización de cuál es el rango efectivo de nuestros datos, en otras palabras, Q^2 estima la bondad de predicción del modelo.



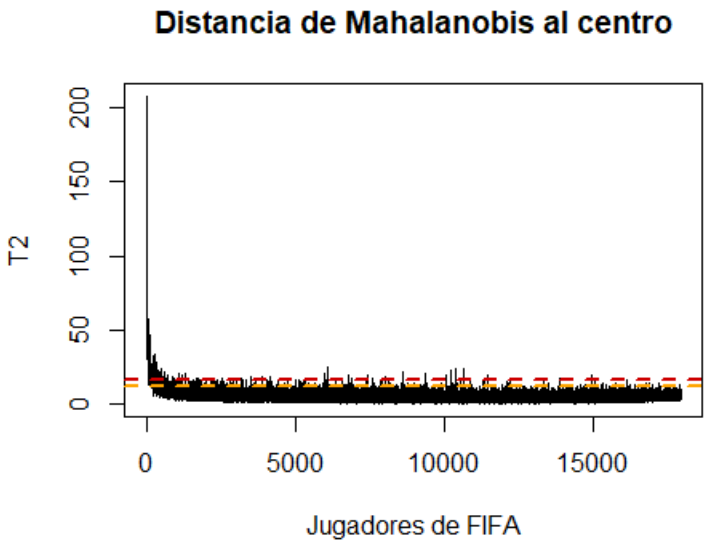
A la vista del gráfico anterior, podemos observar que Q^2 llega a un máximo con 6 componentes, a partir de ahí, las predicciones serán peores. Por tanto, parece que usar un número de 6 componentes principales sí es una buena elección, así que concluimos en usar esa cantidad.

9.1.2 Validación del modelo PCA

9.1.2.1 Detección de anómalos con T2-Hotelling

Como siempre, antes de sacar conclusiones de nuestro modelo PCA, primero tenemos que validarlo.

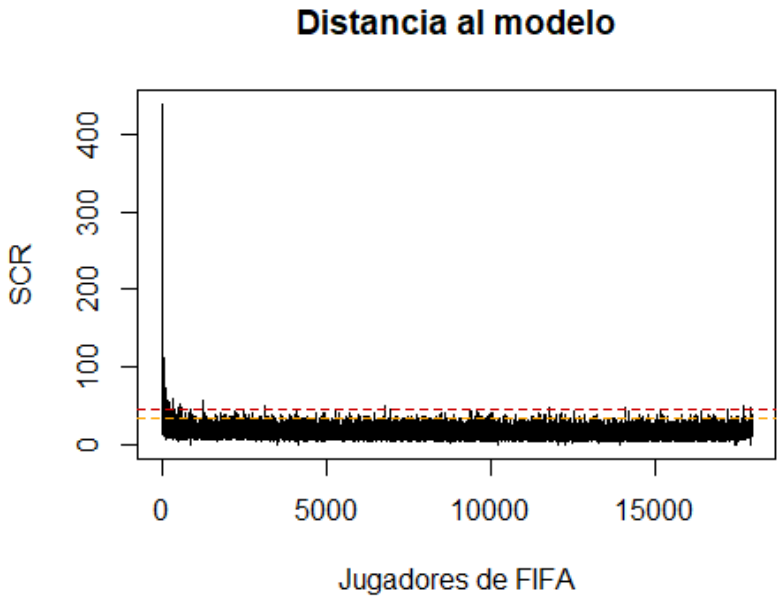
Comenzamos buscando outliers extremos con un gráfico de valores de T^2-Hotelling. Las líneas discontinuas se corresponden con los límites de confianza del 95% y 99%, respectivamente en naranja y rojo. Siempre y cuando la cantidad no sea excesiva, **utilizaremos como criterio de aceptación como falsas alarmas que los valores estén por debajo de tres veces el límite de confianza del 99%**, ya que queremos mantener toda la información posible.



Viendo el gráfico, podemos observar que hay ciertos jugadores con un valor de T2-Hotelling muy por encima de la línea roja, algunos de estos jugadores son ni más ni menos que Lionel Messi, Cristiano Ronaldo o Neymar. Si comprobamos el porcentaje de jugadores que supera cada límite, es mayor al que se esperaría para ser considerados falsas alarmas: por encima del límite de confianza de 95% tenemos 732 observaciones y por encima del límite de 99% tenemos 307 observaciones, que respectivamente son el 0.041% y el 0.017%.

9.1.2.2 Detección de atípicos con SCR

Comprobamos ahora los valores de la suma de cuadrados residual. Para el tratamiento de las alarmas, utilizaremos el mismo criterio que antes en T^2-Hotelling.

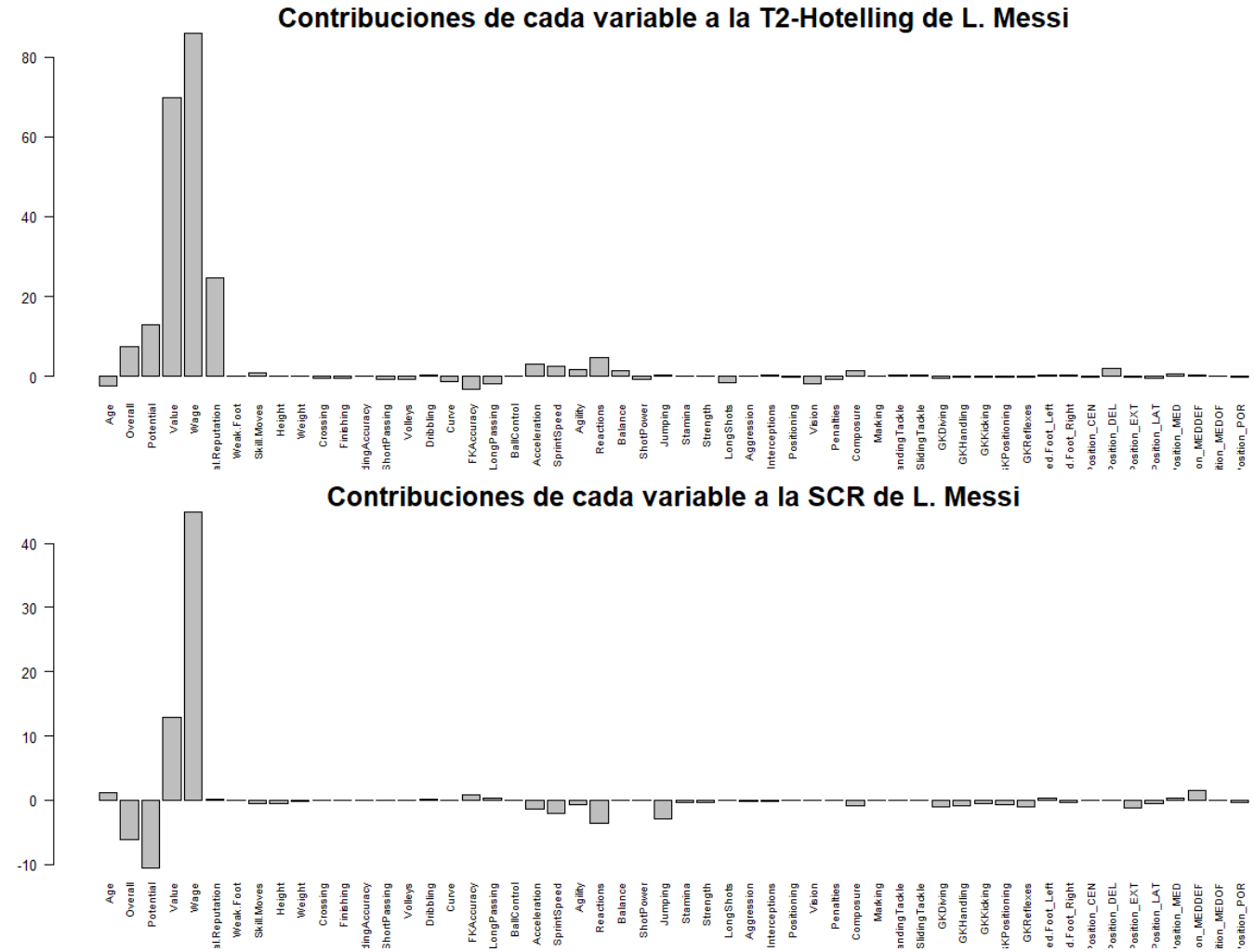


Al haber una serie de valores muy encima de la línea roja, nos encontramos en una situación similar a la de T2-Hotelling. Se diferencia en que el porcentaje de valores que supera los límites está dentro de los permitido: 0.027% (línea naranja), 0.005% (línea roja). Algunos de estos outliers moderados coinciden con algunos de los outliers severos mencionados antes.

9.1.2.3 Contribuciones T2-Hotelling y SCR al modelo

Los dos gráficos superiores nos han mostrado que esos jugadores tan anómalos y/o atípicos son jugadores muy reconocidos mundialmente, es más Cristiano Ronaldo y Neymar salen en las portadas de FIFA 19. Es lógico que aquellas personas que compren el videojuego quieran conseguir a estos jugadores y formar un equipo con ellos, por lo que no nos interesa deshacernos de estos jugadores tan admirados. En lugar de ir eliminando los valores anómalos y reajustar modelos, estudiaremos las contribuciones de cada variable a la T^2-Hotelling y a la SCR de estos outliers.

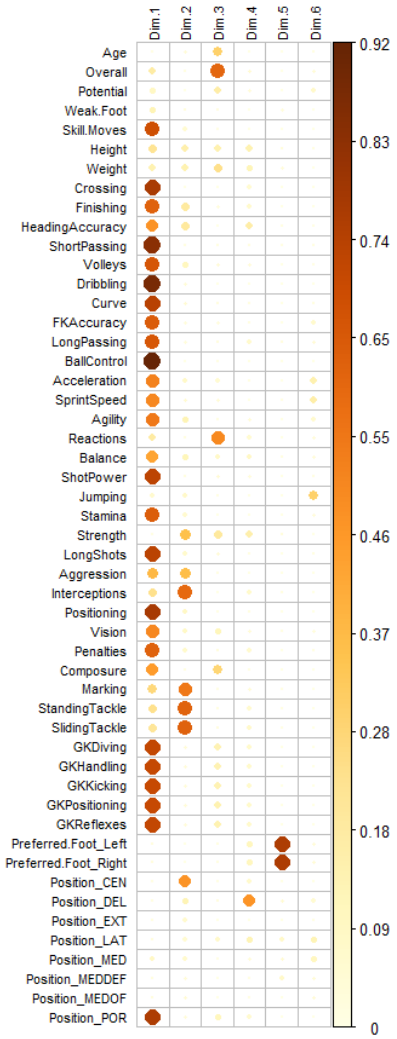
Lionel Messi tiene el valor más grande de T^2-Hotelling y de SCR (es la línea más alta de los gráficos anteriores), aprovecharemos esta coincidencia y haremos sus gráficos de contribución. De esta manera, identificaremos aquellas variables que influyen en mayor medida a generar estas anomalías.



Es fácil observar que las variables Value y Wage tienen una contribución enorme y desproporcionada con respecto a la mayoría de las otras variables, esto tiene sentido con respecto a lo que veíamos en el análisis exploratorio inicial. Por otra parte, estas variables no deberían ser muy influyentes en nuestro objetivo de agrupamientos de jugadores, pero sí podríamos considerarlas como variables de respuesta en nuestro

objetivo de predicción de calidad de jugadores. Es por estas dos razones que decidimos considerarlas como auxiliares al volver a ajustar un nuevo modelo. También decidimos considerar como auxiliar la variable International.Reputation.

9.2 Correlaciones entre variables originales y variables latentes

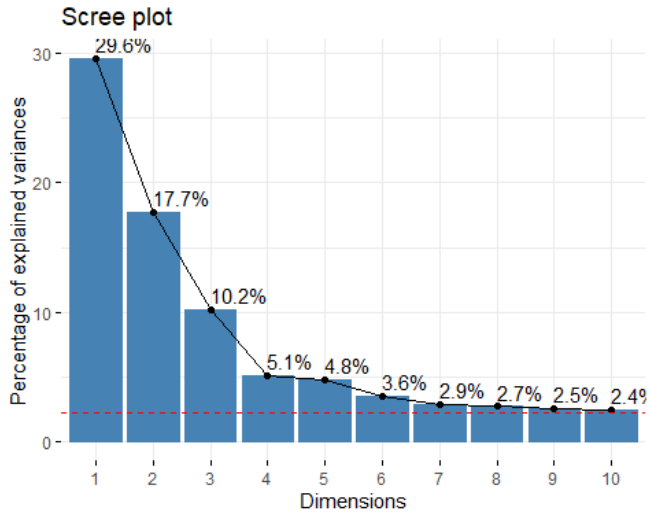


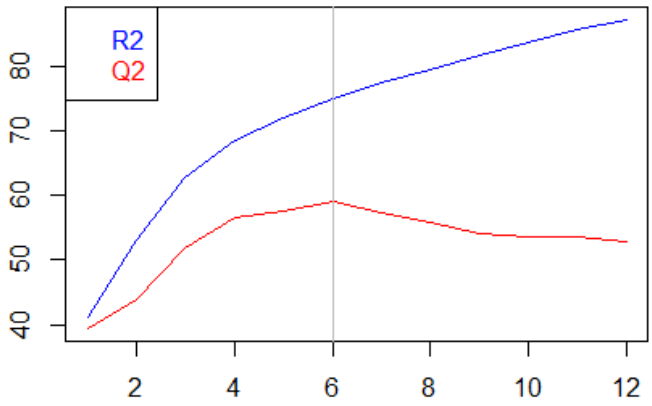
10 Anexo 2

10.1 PCA con solo jugadores

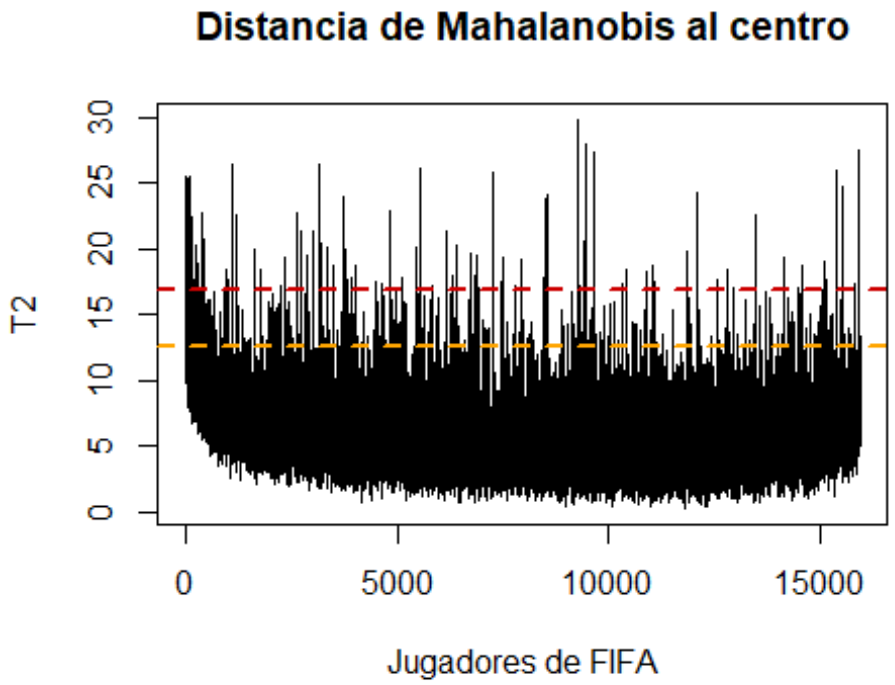
```
fifa_jugadores = subset(fifa, Position != "POR")
fifa_jugadores = fifa_jugadores[,setdiff(colnames(fifa_jugadores),
c("GKDiving","GKHandling","GKPositioning","GKKicking","GKReflexes"))]
fifa_numericas_jug = fifa_jugadores[,setdiff(colnames(fifa_jugadores), c("Nationality",
"Club", "Position"))]

descFifa_jugadores = data.frame("variable" = colnames(fifa_jugadores),
                      "tipo" = c("numerical", "categorical",
                                 rep("numerical", 2), "categorical",
                                 rep("numerical", 2), rep("categorical", 4),
```

```
                                  rep("numerical", 41)), stringsAsFactors = FALSE)
rownames(descFifa_jugadores) = descFifa_jugadores$variable

res.pca_jugadores = PCA(fifa_jugadores, scale.unit = TRUE, graph = FALSE, ncp = 6,
              quali.sup = which(descFifa_jugadores$tipo == "categorical"), quanti.sup =
6:8) #esas 3 como auxiliares

## Warning in PCA(fifa_jugadores, scale.unit = TRUE, graph = FALSE, ncp = 6, :
## Missing values are imputed by the mean of the variable: you should use the
## imputePCA function of the missMDA package

eig.val <- get_eigenvalue(res.pca_jugadores)
VPmedio = 100 * (1/nrow(eig.val))
fviz_eig(res.pca_jugadores, addlabels = TRUE) +
  geom_hline(yintercept=VPmedio, linetype=2, color="red")
```

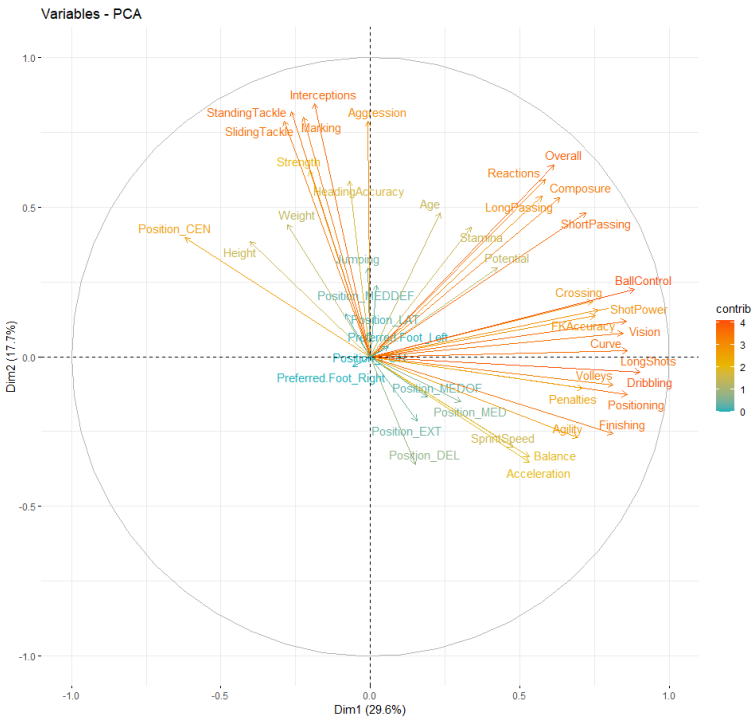




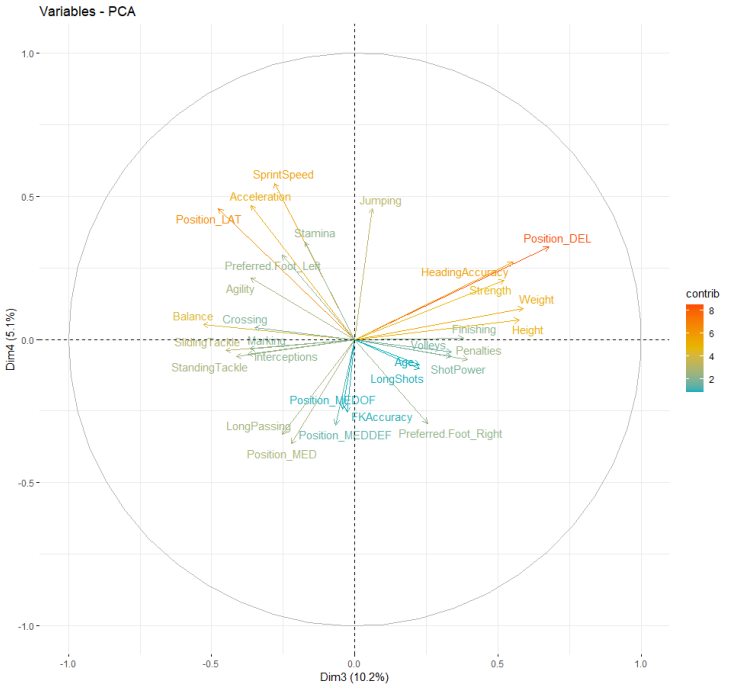

```
#{r T2, fig.width=5, fig.height=5}

K = 6
misScores = res.pca_jugadores$ind$coord[,1:K]
miT2 = colSums(t(misScores**2) / eig.val[1:K]) # max(miT2) # printeamos el valor máximo
I = nrow(fifa)
F95 = K*(I**2 - 1)/(I*(I - K)) * qf(0.95, K, I-K)
F99 = K*(I**2 - 1)/(I*(I - K)) * qf(0.99, K, I-K)
plot(1:length(miT2), miT2, type = "l", xlab = "Jugadores de FIFA", ylab = "T2", main =
"Distancia de Mahalanobis al centro")
abline(h = F95, col = "orange", lty = 2, lwd = 2)
abline(h = F99, col = "red3", lty = 2, lwd = 2)
```



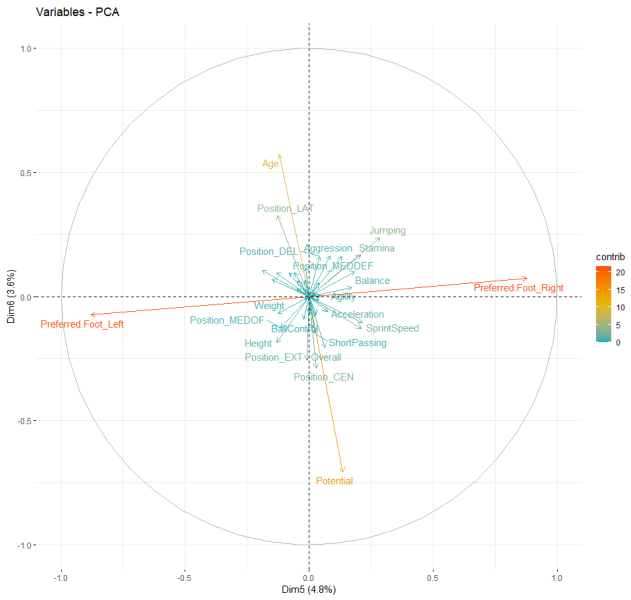

```
alarma95 = which(miT2 > F95)
alarma99 = which(miT2 > F99)
#dim(as.matrix(which(miT2 > F99)))
#334/17918

T2_eliminar <- which(miT2 > 3*F99)
#fifa <- fifa[-T2_eliminar,]
```



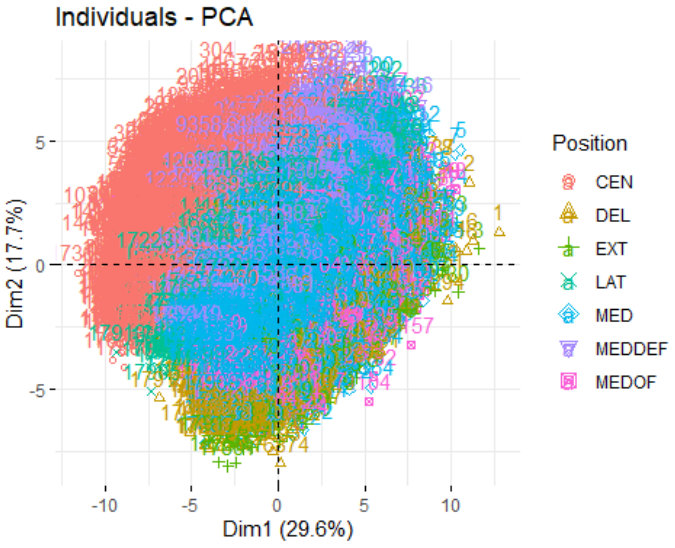

```
fviz_pca_var(res.pca_jugadores, axes = c(3,4), repel = TRUE, col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),invisible =
c("quanti.sup", "quali"), select.var = list(contrib = 30) )
```



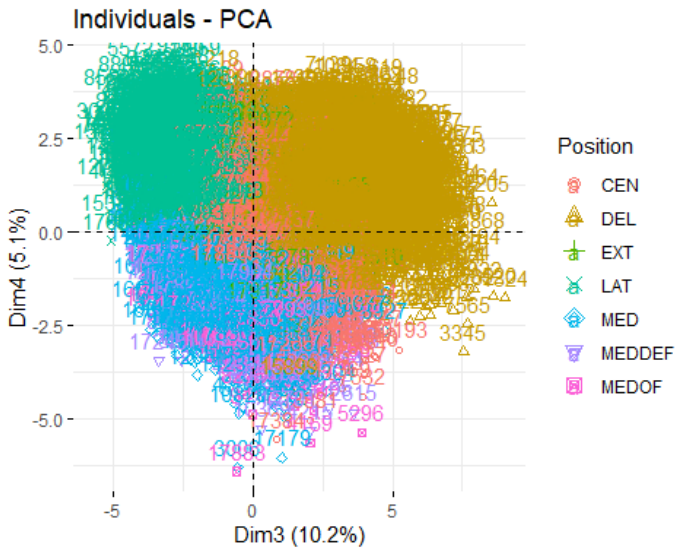

```
fviz_pca_var(res.pca_jugadores, axes = c(5,6), repel = TRUE, col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), invisible =
c("quanti.sup", "quali"))
```



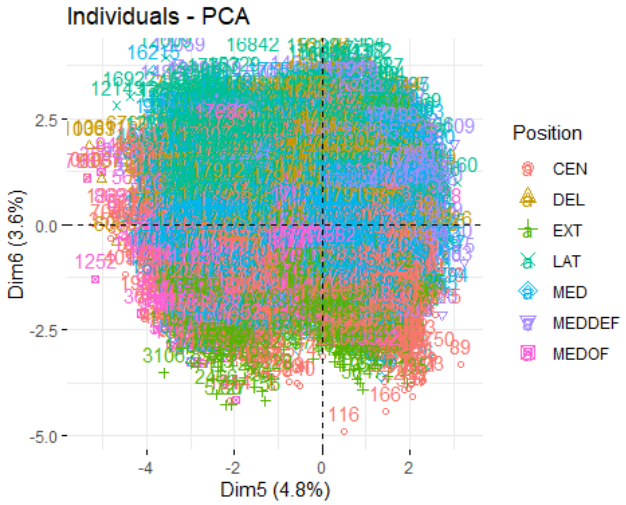

```
#corrplot(res.pca_2$var$contrib, is.corr=FALSE, tl.col = 1, win.asp = 0.5, cl.ratio = 0.3, cl.cex = 0.8)

fviz_pca_ind(res.pca_jugadores, axes = c(1,2), geom = c("point", "text"), addEllipses = F, habillage = "Position")
```




```
fviz_pca_ind(res.pca_jugadores, axes = c(3,4), geom = c("point", "text"), addEllipses = F, habillage = "Position")
```

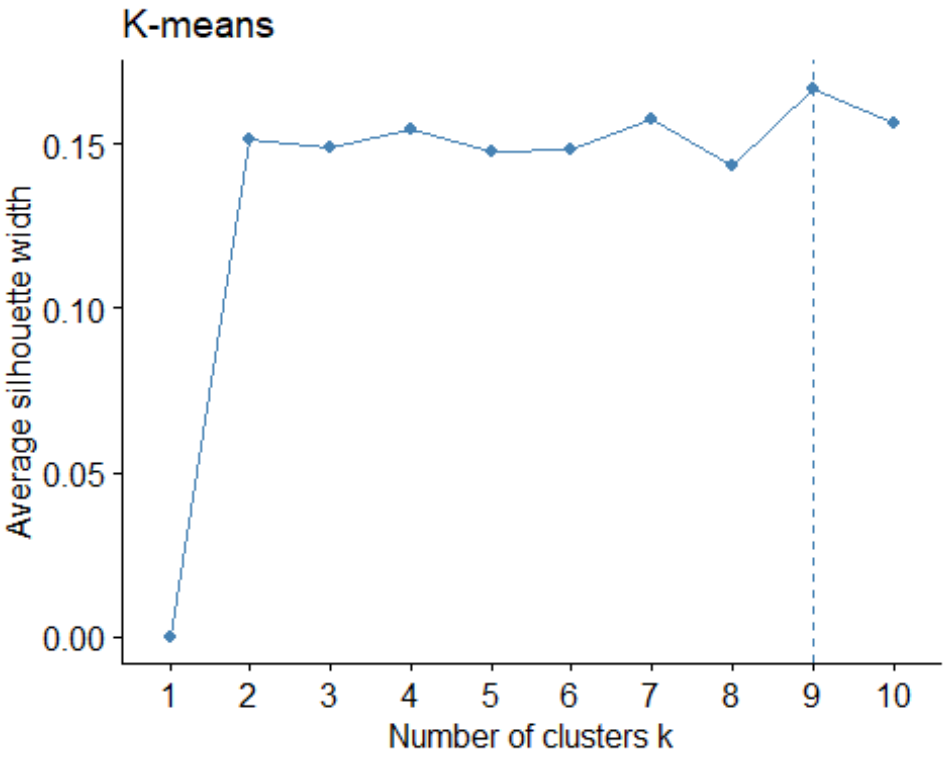



```
fviz_pca_ind(res.pca_jugadores, axes = c(5,6), geom = c("point", "text"), addEllipses =
F, habillage = "Position")
```



10.2 Clustering con los datos originales

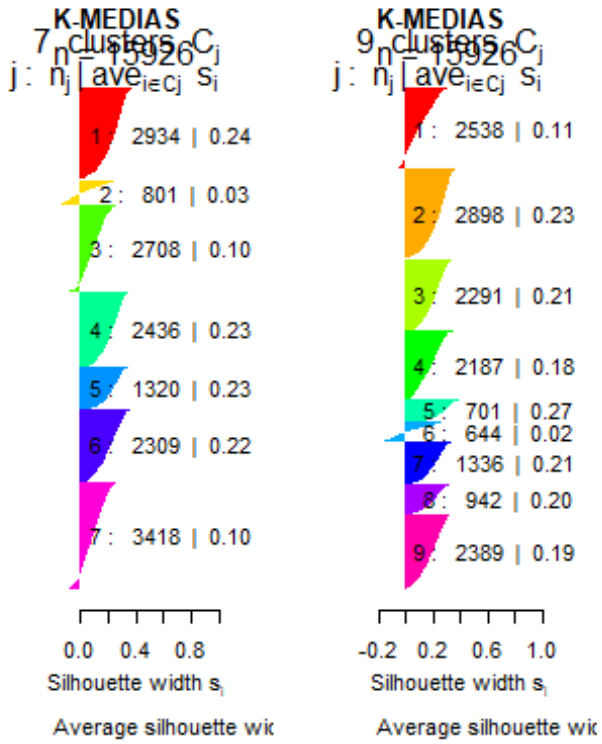
```
fifa_numericas_or = round(fifa_numericas, digits = 3)
fifa_numericas_or = as.data.frame(fifa_numericas_or)

fifa_numericas_jugadores = subset(fifa_numericas_or, Position_POR != 2.827)

midist_or = get_dist(fifa_numericas_jugadores, stand=FALSE, method = "euclidean")

fviz_nbclust(x = fifa_numericas_jugadores, FUNcluster = kmeans, method = "silhouette",
             k.max = 10, verbose = FALSE) +
  labs(title = "K-means")
```



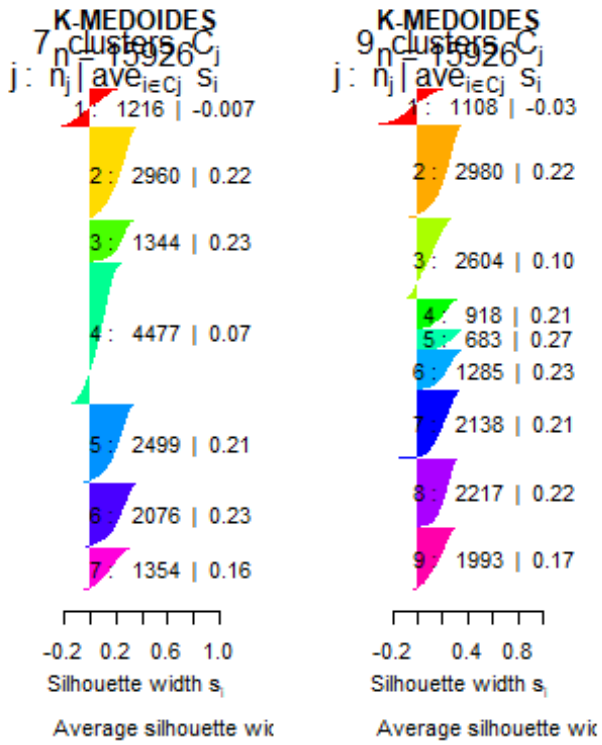

```
par(mfrow=c(1,1))
```

Se obtiene un coeficiente de silhouette bastante bajo, y indica que el óptimo es 9 clusters. Probamos el módelo con 7 clusters también, ya que tiene un coeficiente similar a 9.

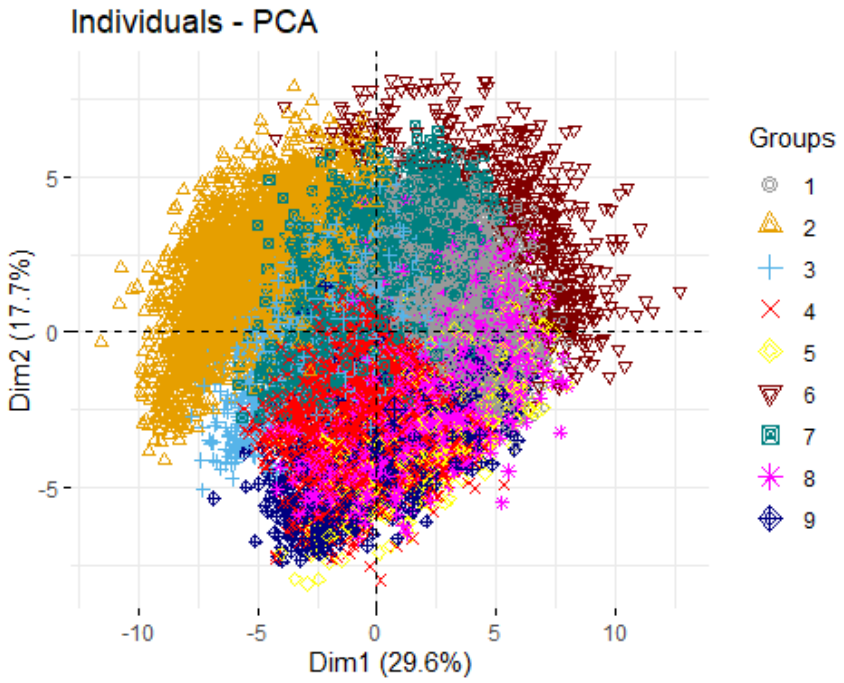
```
clust_means_1_or <- kmeans(fifa_numericas_jugadores, centers = 7, nstart = 20)
clust_means_2_or <- kmeans(fifa_numericas_jugadores, centers = 9, nstart = 20)
set.seed(100)

clust_medoides_1_or <- clara(fifa_numericas_jugadores, 7, metric = "euclidean", stand =
FALSE, samples = 50, pamLike = TRUE)
clust_medoides_2_or <- clara(fifa_numericas_jugadores, 9, metric = "euclidean", stand =
FALSE, samples = 50, pamLike = TRUE)

par(mfrow = c(1,3))
plot(silhouette(clust_means_1_or$cluster, midist_or), col=rainbow(7), border=NA, main =
"K-MEDIAS")
plot(silhouette(clust_means_2_or$cluster, midist_or), col=rainbow(9), border=NA, main =
"K-MEDIAS")
```



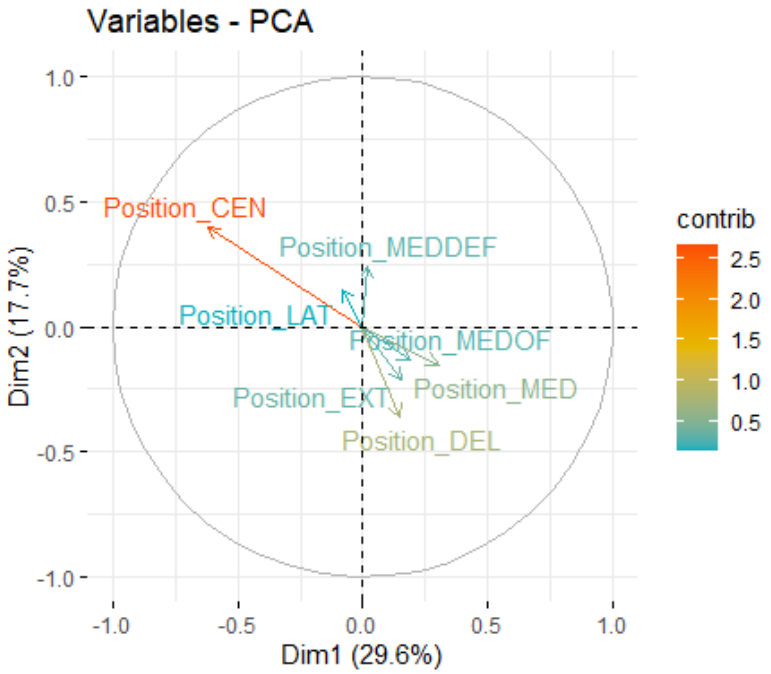

```
par(mfrow=c(1,3))
plot(silhouette(clust_medoides_1_or$clustering, midist_or), col=rainbow(7), border=NA,
main = "K-MEDOIDES")
plot(silhouette(clust_medoides_2_or$clustering, midist_or), col=rainbow(9), border=NA,
main = "K-MEDOIDES")
```



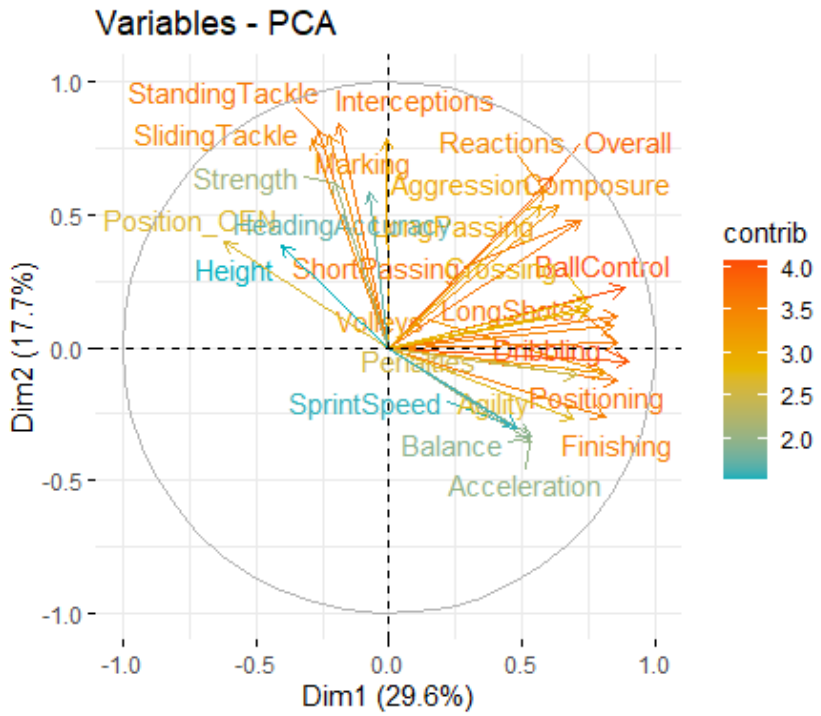
```
clust_or = factor(clust_means_2_or$cluster)
set.seed(100)

par(mfrow=c(1,3))

fviz_pca_ind(res.pca_jugadores, geom = "point", habillage = clust_or, addEllipses =
FALSE,
             palette = c("#999999", "#E69F00", "#56B4E9", "#FF0000", "#FFFF00",
"#800000", "#008080", "#FF00FF", "#000080"), axes = c(1,2))
```




```
fviz_pca_var(res.pca_jugadores, axes = c(1,2), repel = TRUE, col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"), invisible =
c("quanti.sup","quali"),select.var = list(name =
c("Position_CEN","Position_DEL","Position_MEDOF","Position_MEDDEF","Position_EXT","Positi
on_LAT","Position_MED"))) #, select.var = list(contrib = 70)
```



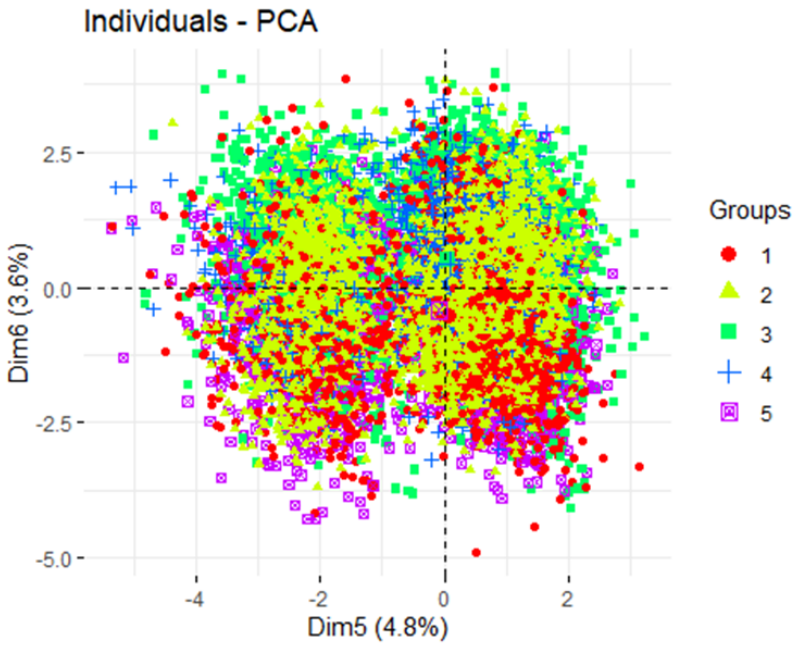

```
fviz_pca_var(res.pca_jugadores, axes = c(1,2), repel = TRUE, col.var = "contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),invisible =
c("quanti.sup", "quali"), select.var = list(contrib = 30) )
```



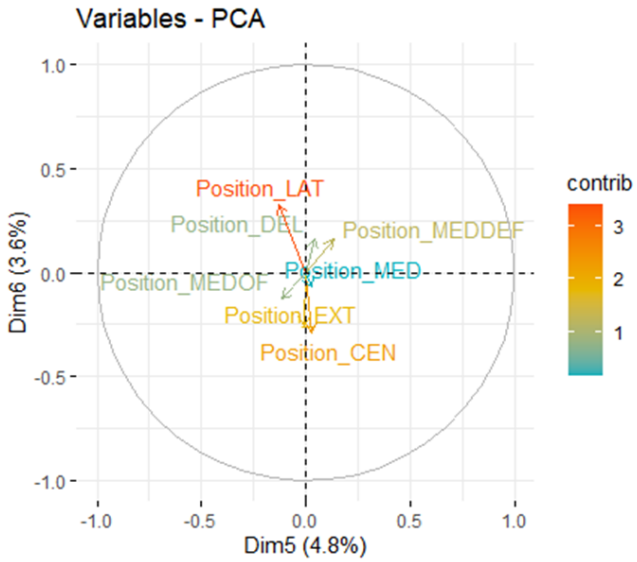
Seguiremos el análisis clustering con las componentes principales en lugar de los datos originales, debido a que no se pueden obtener conclusiones de estos, los clusters se solapan demasiado.

10.3 Perfil medio de los clusters.

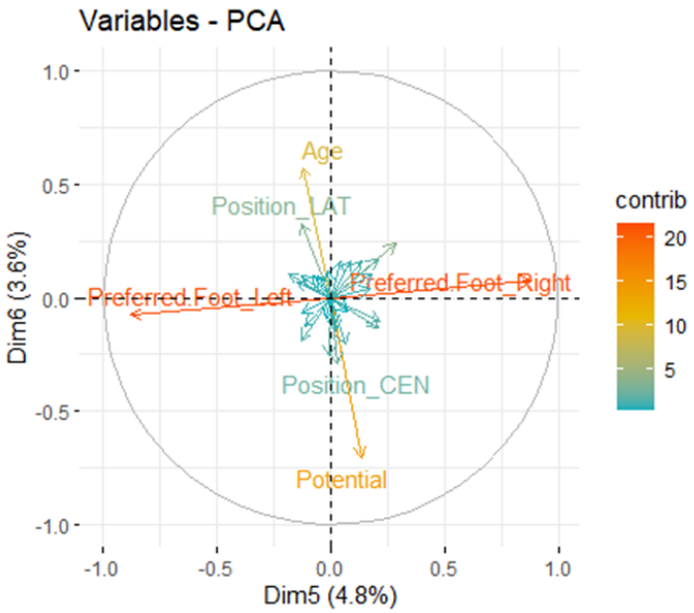
```
par(mfrow=c(1,3))

fviz_pca_ind(res.pca_jugadores, geom = "point", habillage = clust,
addEllipses = FALSE,
             palette = rainbow(5), axes = c(5,6))
```



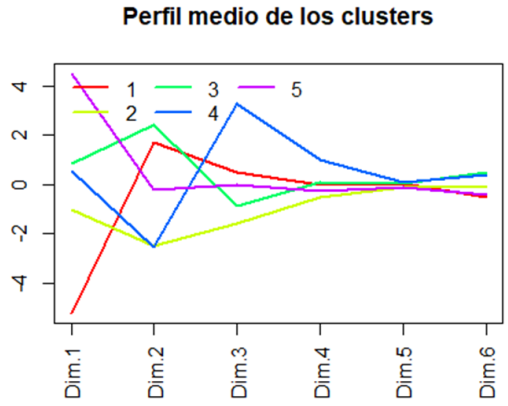

```
fviz_pca_var(res.pca_jugadores, axes = c(5,6), repel = TRUE, col.var =
"contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
invisible = c("quanti.sup","quali"),select.var = list(name =
c("Position_CEN","Position_DEL","Position_MEDOF","Position_MEDDEF","Posit
ion_EXT","Position_LAT","Position_MED"))) #, select.var = list(contrib =
70)
```



```
fviz_pca_var(res.pca_jugadores, axes = c(5,6), repel = TRUE, col.var =
"contrib",
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),invisible
= c("quanti.sup", "quali"), select.var = list(contrib = 30) )
```



En las dimensiones 5 y 6, la variabilidad esta explicada principalmente por las variables "Preferred.Foot.Left", y "Preferred.Foot.Right". Por tanto, al no tratarse de variables relacionadas con las habilidades de juego no nos interesan para el análisis.



Podemos observar como en las componentes 5 y 6 el valor de cada cluster es muy bajo, por lo que estas dimensiones no contribuyen demasiado a la separación entre grupos.

11 Anexo 3

11.1 Implementación del método PLS para porteros

Para poder realizar el análisis sobre los porteros, seleccionamos únicamente los jugadores pertenecientes a esta posición, y realizamos una partición del 75% de los datos para el entrenamiento y 25% para evaluar los resultados.

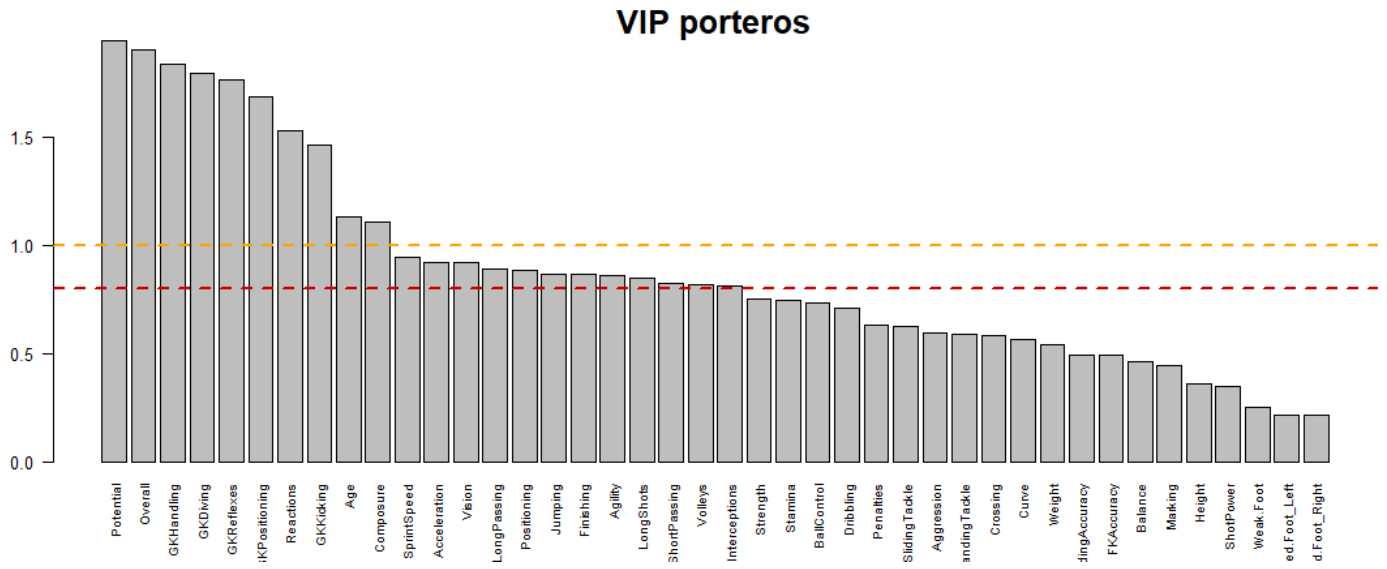
Además, seleccionamos las variables a predecir, en nuestro caso Value, Wage e International.Reputation, y las variables predictoras, que son las mismas que usamos para el modelo PCA.

11.1.1 Selección de variables

Primero, vamos a seleccionar aquellas variables que sean significativas en el modelo PLS. A lo que esto se refiere es que a pesar de que contemos con 42 regresores, no necesariamente necesitaremos usar todos para conseguir un buen modelo, además que tener menos variables facilitará la interpretación. Por ello, realizamos un modelo PLS **inicial** con los datos de entrenamiento centrados y escalados y usando validación cruzada 10-fold. Escogeremos un número de componentes que aporte unos buenos valores de R^2 y Q^2 (en este caso 3 componentes).

Una vez obtenido este modelo inicial, vamos a estudiar los valores de VIP de cada variable para saber la importancia relativa que tiene cada una en el modelo. Como criterio de aceptación, conservaremos sólo aquellas que sean mayores o iguales al valor de VIP de 0.8. Si tienen un valor VIP menor al mencionado, podremos considerar que son irrelevantes y/o estadísticamente no significativas.

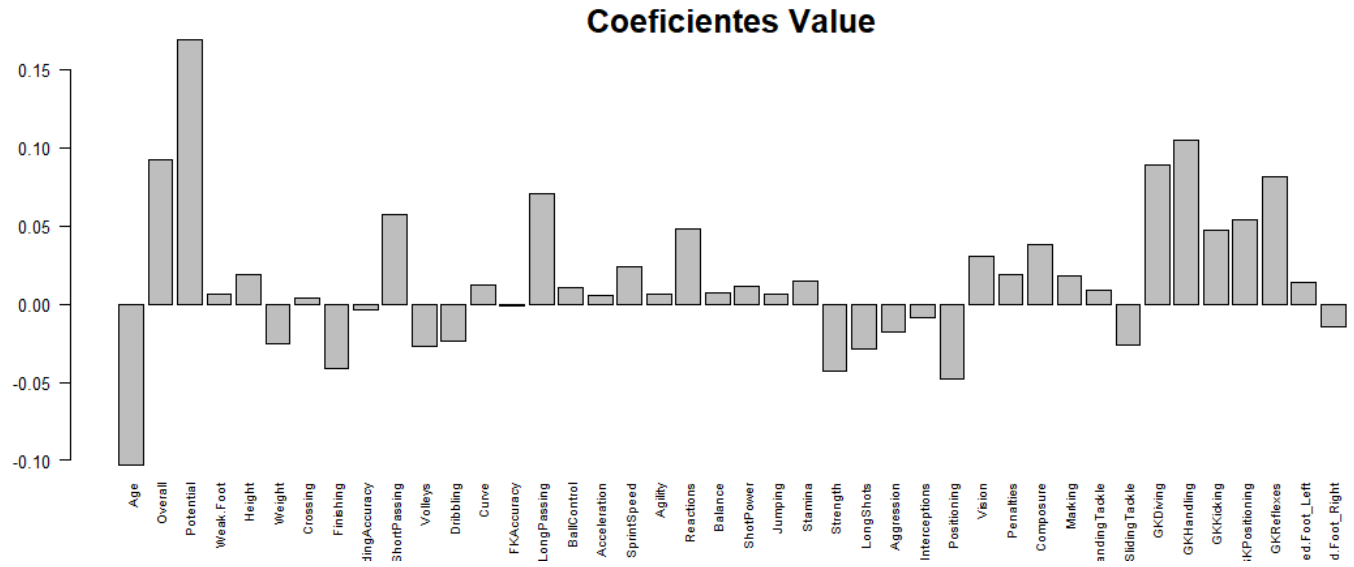
```
barplot(sort(mypls_p@vipVn, decreasing = TRUE), main = "VIP porteros", las = 2, cex.names
= 0.8, cex.main = 2)
abline(h = 1, col = "orange", lty = 2, lwd = 2)
abline(h = 0.8, col = "red3", lty = 2, lwd = 2)
```



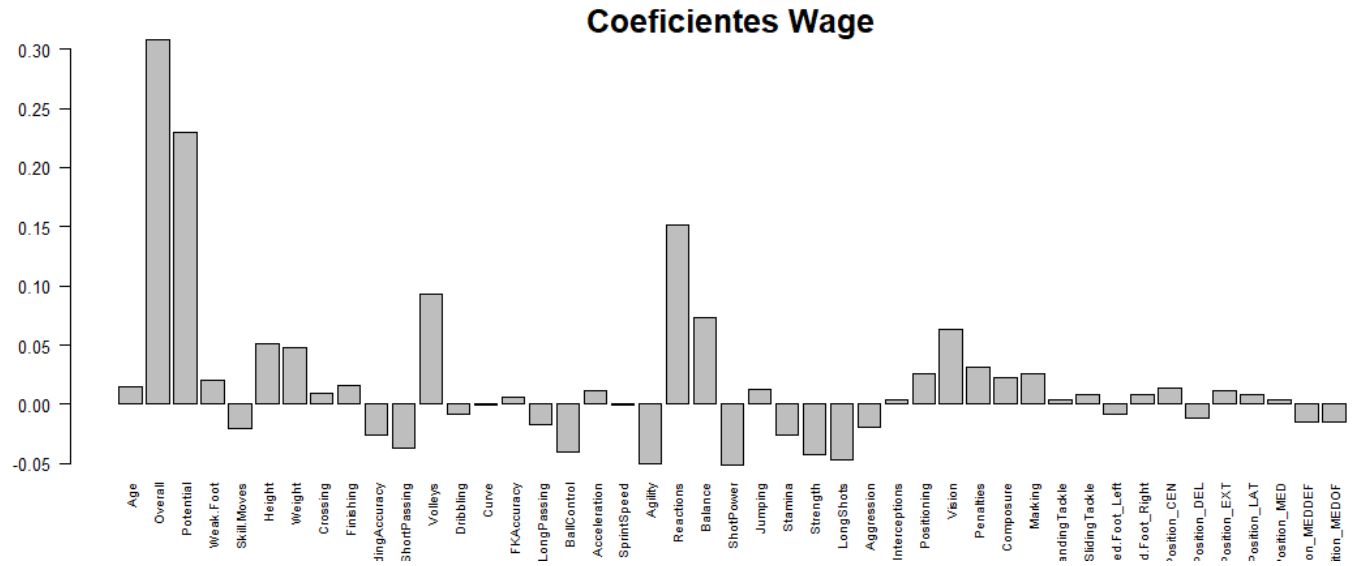
Son 20 las variables que no superan el valor de VIP de 0.8. Tras esto, el número de variables explicativas útiles se queda en 22.

Los coeficientes para cada variable de respuesta serían los siguientes:

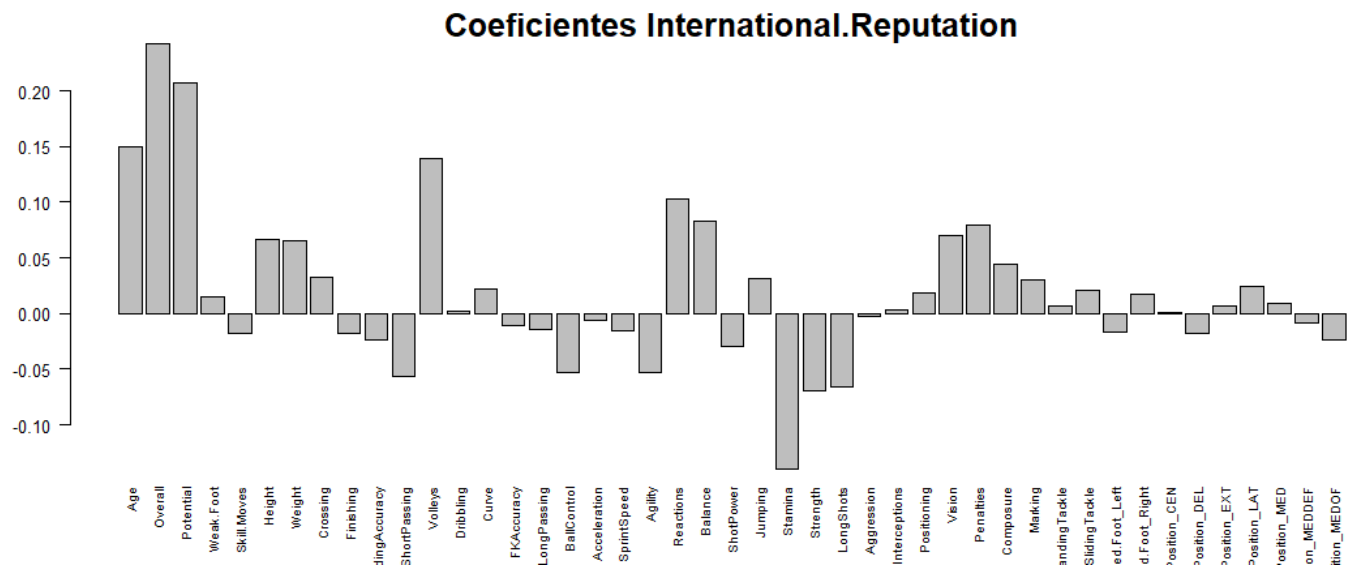
```
barplot(mypls_p@coefficientMN[,1],main = "Coeficientes Value", las =2, cex.names = 0.8,
cex.main = 2)
```




```
barplot(mypls_p@coefficientMN[,2],main = "Coeficientes Wage", las =2, cex.names = 0.8,
cex.main = 2)
```



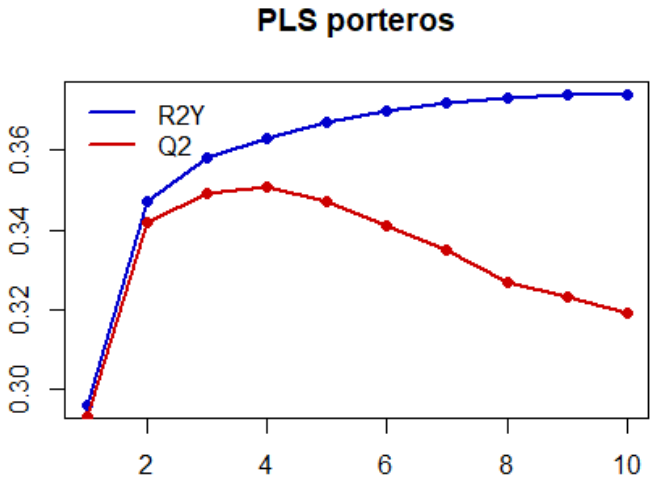

```
barplot(mypls_p@coefficientMN[,3],main = "Coeficientes International.Reputation", las =2,
cex.names = 0.8, cex.main = 2)
```



Salta a la vista el casi idéntico comportamiento de los coeficientes de Value y Wage. También se puede ver que las variables con las que más relación tienen son Overall y Potencial.

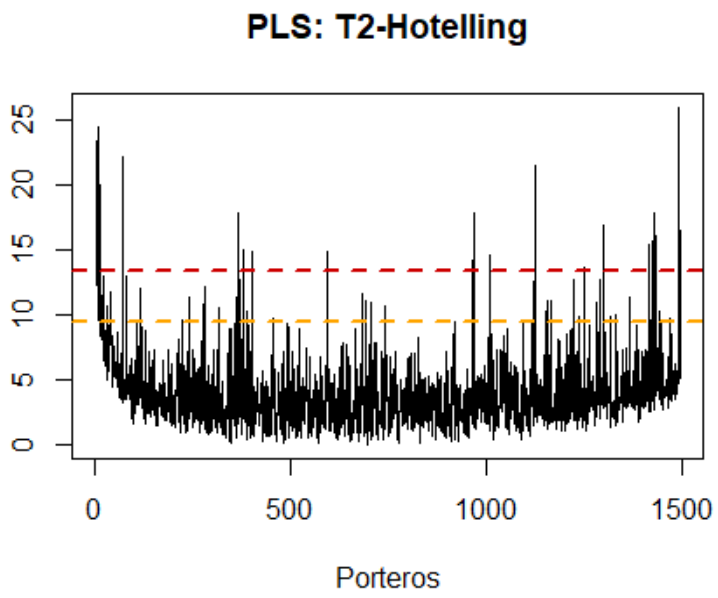
11.1.2 Elección del número de componentes

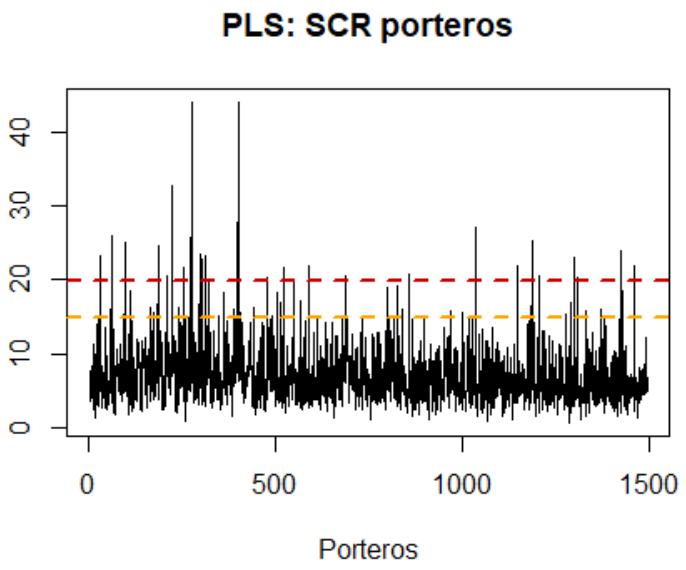
Reajustamos el modelo sin estas variables que no superan el criterio de VIP, de nuevo con validación cruzada 10-fold. Recurriremos de nuevo a R^2 y Q^2 para elegir el número de componentes PLS. En este caso nos quedaremos con 4 componentes.



11.1.3 Validación con T2 Hotelling y SCR

Antes de interpretar el modelo, vamos a validarlo. Consideraremos por válidos aquellos valores que no superen 2 veces los límites de confianza del 99%. A diferencia del anterior objetivo realizado con PCA y clustering, ahora sí podríamos considerar eliminar algún individuo muy anómalo o atípico, ya que el objetivo es predecir las 3 variables antes mencionadas, y no es necesario mantener todos los jugadores.





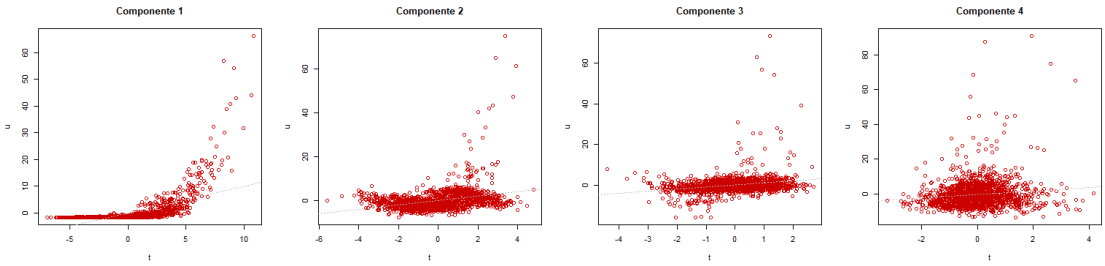
En el gráfico de T^2-Hotelling vemos que hay pocos porteros que pasen el límite del 95%-99%, y decidimos no eliminarlos ya que al igual que en PCA, los porteros anómalos son aquellos con mejores características y que se diferencian en gran medida de los porteros mediocres, que son la mayoría. Por encima del límite de confianza de 95% tenemos 69 observaciones y por encima del límite de 99% tenemos 22 observaciones, que respectivamente son el 0.046% y el 0.015%. Respecto a la suma de cuadrados residual, tampoco tenemos datos demasiado atípicos o anómalos, como podemos observar en el gráfico. Por tanto, damos por validado el modelo PLS de porteros.

11.1.4 Relación lineal entre scores porteros

Si observamos la relación entre los *scores* de los jugadores, en la primera componente vemos que hay una tendencia lineal donde se encuentran la mayoría de jugadores, pero se acaba convirtiendo en exponencial, donde se concentran menos valores. Esta exponencialidad puede ser debida a los jugadores anómalos, que por lo que hemos estado estudiando en PCA, parece que tienden a ser los jugadores más importantes o que mejor juegan en comparación a la gran mayoría de los demás. Como es de esperar, la primera componente es la que más relación lineal manifiesta, en este caso de 0.5987. Las otras 9 componentes, por orden, tienen correlaciones lineales de , 0.3113, 0.2079 y 0.1255, en la figura de abajo se pueden visualizar.

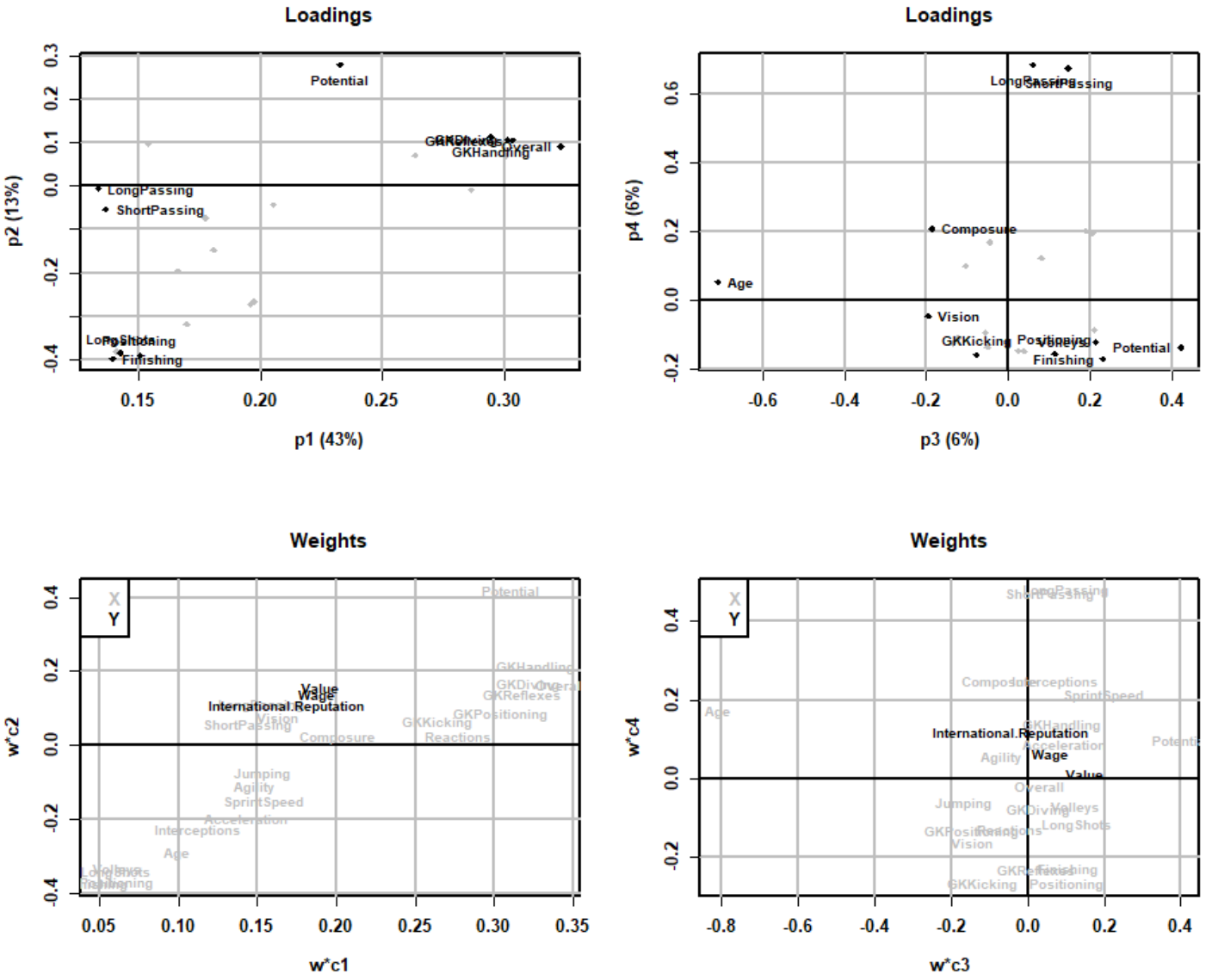
Cabe destacar que seguramente estos resultados tan poco precisos se deban a la relación exponencial mencionada. Para arreglarlo, podríamos recurrir al uso de transformaciones en los datos, repitiéndo el análisis pero en vez de hacer el modelo con unas variables de respuesta 'Y', aplicaríamos logaritmos para que quedasen de la forma 'ln(Y)'. Es una tarea compleja en la que no indagaremos, pero que podrá ser desarrollada en el futuro.

Las correlaciones de cada par de componentes t y u son, respetivamente, 0.5987, 0.3113, 0.2079 y 0.1255



11.1.5 Interpretación del modelo PLS porteros

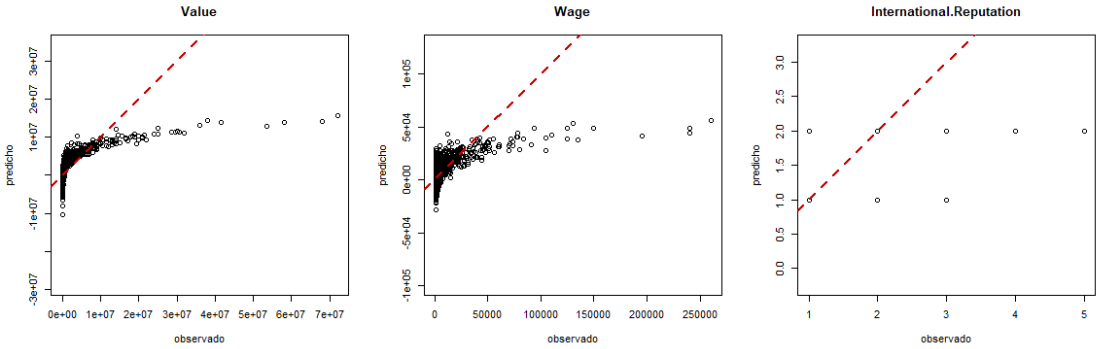
Los siguientes gráficos nos servirán para interpretar el modelo PLS de porteros.



Los resultados son similares al modelo de jugadores.

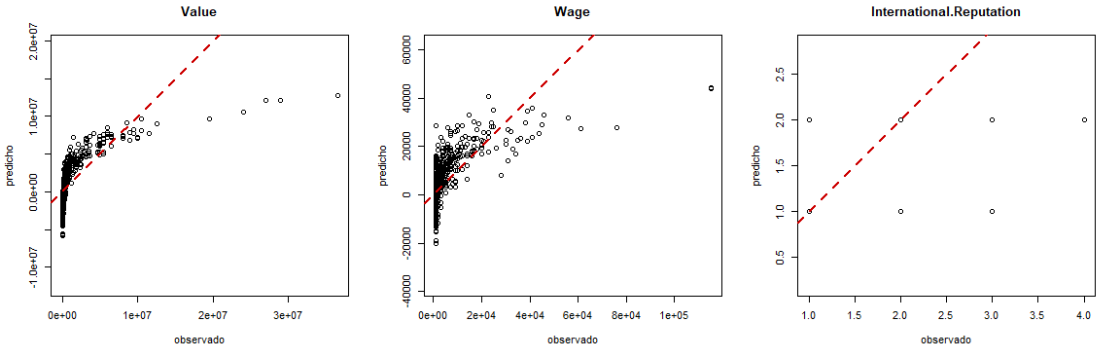
11.1.6 Predicciones con PLS porteros

Antes de predecir con la muestra de test final, realizamos una primera predicción para ver cómo funciona nuestro modelo. Observamos cómo el modelo se ajusta en un principio a los datos pero luego no lo consigue.



11.1.7 Predicciones para test porteros

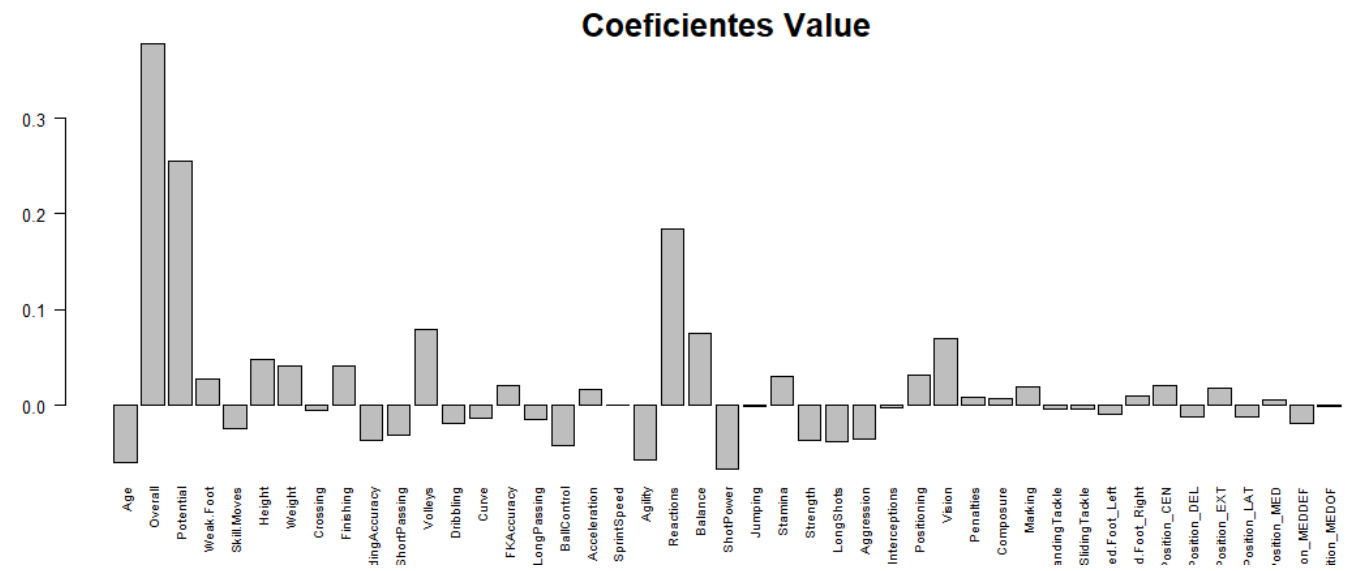
Finalmente, realizamos las predicciones con nuestros datos de test. Como habíamos observado antes, el modelo se consigue ajustar en un principio a los datos, pero luego se desvía infravalorando bastante a los porteros con valores altos. Esto puede ser debido a que el modelo se ajusta bastante a la mayoría, que son los porteros mediocres, no consiguiendo bien predecir el valor de los porteros importantes, que son menos.



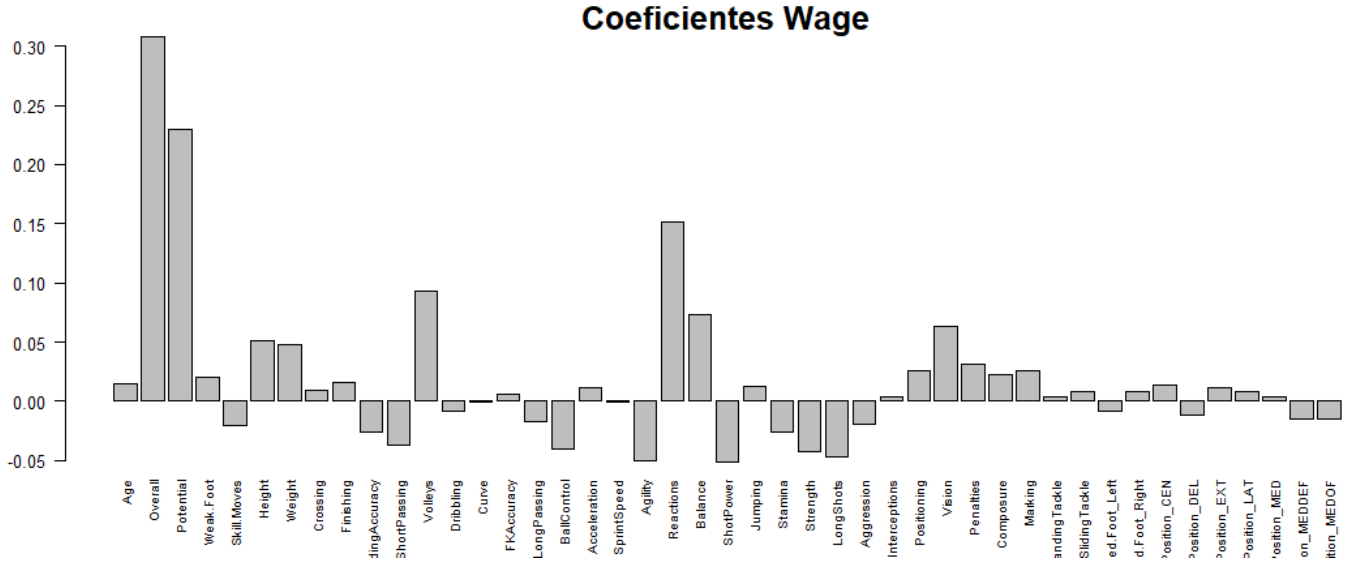
11.2 Coeficientes PLS modelo jugadores

Salta a la vista el casi idéntico comportamiento de los coeficientes de Value y Wage. También se puede ver que las variables con las que más relación tienen son Overall y Potencial.

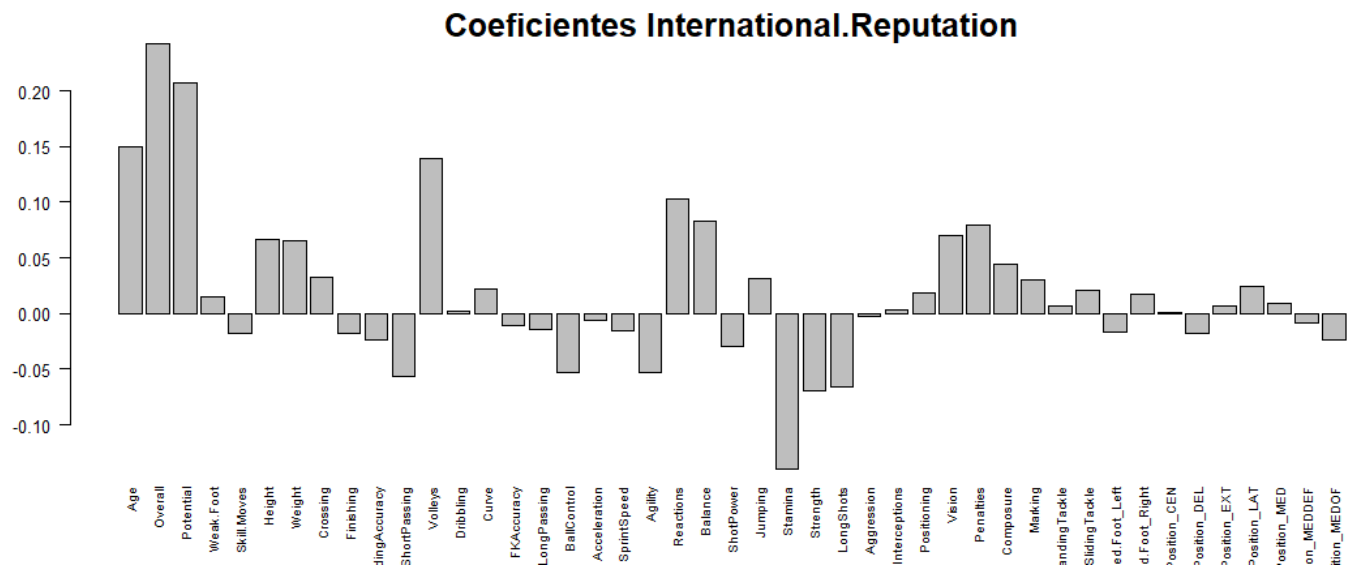
```
barplot(mypls_j@coefficientMN[,1],main = "Coeficientes Value", las =2, cex.names = 0.8,
cex.main = 2)
```




```
barplot(mypls_j@coefficientMN[,2],main = "Coeficientes Wage", las =2, cex.names = 0.8,
cex.main = 2)
```




```
barplot(mypls_j@coefficientMN[,3],main = "Coeficientes International.Reputation", las =2,
cex.names = 0.8, cex.main = 2)
```



11.3 Iteraciones T2 y SCR jugadores