Preprocesado de datos del Fifa 2017

Jesús González

8/3/2021

Contents

Carga del archivo

```
fifa.df <- read.csv(file="fifa_raw.csv", encoding = "UTF-8")
paste0("Número de observaciones: ", nrow(fifa.df),". Número de variables: ", ncol(fifa.df))
```

```
## [1] "Número de observaciones: 17588. Número de variables: 54"
```

Realizamos un subset con las variables escogidas para el estudio.

```
col_names <- c("ID", "Name", "Nationality", "Club_Joining", "Contract_Expiry", "Rating", "Height", "Weig
fifa.df.sub <- fifa.df[, col_names]
paste0("Número de observaciones: ", nrow(fifa.df.sub),". Número de variables: ", ncol(fifa.df.sub))
```

```
## [1] "Número de observaciones: 17588. Número de variables: 12"
```

Se ha pasado de 54 variables a 11, manteniendo el número de observaciones. A continuación se muestran las 5 primeras observaciones de la muestra.

```
head(fifa.df.sub,5)
```

```
##   ID              Name Nationality Club_Joining Contract_Expiry Rating Height
## 1  1 Cristiano Ronaldo    Portugal   07/01/2009            2021     94 1,85 m
## 2  2      Lionel Messi   Argentina   07/01/2004            2018     93   <NA>
## 3  3            Neymar      Brazil   07/01/2013            2021     92 1,74 m
## 4  4       Luis Suárez     Uruguay   07/11/2014            2021     92 1,82 m
## 5  5      Manuel Neuer     Germany   07/01/2011            2021     92 1,93 m
##     Weight Preffered_Foot Birth_Date Age       Work_Rate
## 1     <NA>              2 02/05/1985  32      High / Low
## 2 72475 gr              1 06/24/1987  29 Medium / Medium
## 3 68884 gr              2 02/05/1992  25   High / Medium
## 4 85511 gr              2 01/24/1987  30   High / Medium
## 5     <NA>              2 03/27/1986  31 Medium / Medium
```

Vamos a examinar cada una de las variables.

```
summary(fifa.df.sub)
```

```
##        ID            Name           Nationality        Club_Joining
##  Min.   :    1   Length:17588       Length:17588       Length:17588
##  1st Qu.: 4398   Class :character   Class :character   Class :character
##  Median : 8794   Mode  :character   Mode  :character   Mode  :character
##  Mean   : 8794
##  3rd Qu.:13191
##  Max.   :17588
##
##  Contract_Expiry     Rating         Height             Weight
##  Min.   :2017    Min.   :45.00   Length:17588       Length:17588
##  1st Qu.:2017    1st Qu.:62.00   Class :character   Class :character
##  Median :2019    Median :66.00   Mode  :character   Mode  :character
##  Mean   :2019    Mean   :66.17
##  3rd Qu.:2020    3rd Qu.:71.00
##  Max.   :2023    Max.   :94.00
##  NA's   :1
##  Preffered_Foot   Birth_Date             Age         Work_Rate
##  Min.   :1.000   Length:17588       Min.   :17.00   Length:17588
##  1st Qu.:2.000   Class :character   1st Qu.:22.00   Class :character
##  Median :2.000   Mode  :character   Median :25.00   Mode  :character
##  Mean   :1.767                      Mean   :25.46
##  3rd Qu.:2.000                      3rd Qu.:29.00
##  Max.   :2.000                      Max.   :47.00
##
```

De la observación de las características de las variables y de las cinco primera observaciones podemos ver los siguiente:

1. Contract_Expiry: Presenta como mínimo un valor NA.

2. Preffered_Foot: Contiene valores 1,2 que se tendrán que traducir en Left y Right respectivamente.
3. En las variables numéricas, a excepción de la anterior en la que carece de sentido, la media y la mediana tienen valores similares, que indica poca presencia de Outliers.
4. Las variables Height and Weight presentan valores NA que tendrán que ser tratados.

Verificar la duplicación de registros

En esta sección se realizará la comprobación de existencia de observaciones duplicadas en la muestra. Para ello, primeramente vamos a comprobar si el campo "ID" se encuentra duplicado mediante la comparación de la longitud del vector antes y después de aplicar la función de eliminación de duplicados.

```
if (length(fifa.df.sub$ID) == length(unique(fifa.df.sub$ID))) {
  print("No hay diferencias en el ID")} else {print("Existen diferencias en el ID")}
```

```
## [1] "No hay diferencias en el ID"
```

No obstante, esto no nos garantiza que no existan jugadores duplicados, dado que el ID puede ser generado con un autonumérico en Base de Datos. Como doble comprobación, utilizaremos la variable con el nombre del jugador, como criterio extra en la búsqueda de duplicados.

```
if (length(unique(fifa.df.sub$Name)) != nrow(fifa.df.sub)){
  paste("Tenemos", (nrow(fifa.df.sub) - length(unique(fifa.df.sub$Name))), "posibles registros duplicado
}
```

```
## [1] "Tenemos 228 posibles registros duplicados."
```

Vamos a mostrar algunos de los duplicados.

```
head(fifa.df.sub[duplicated(fifa.df.sub$Name),], 10)
```

```
##        ID           Name Nationality Club_Joining Contract_Expiry Rating Height
## 643   643       Fernando      Brazil   06/27/2014            2019     79 183 cm
## 671   671         Danilo      Brazil   07/01/2011            2020     79 185 cm
## 698   698 Lisandro López   Argentina   01/04/2016            2019     79 174 cm
## 768   768          Bruno       Spain   07/08/2014            2019     78 185 cm
## 862   862        Rafinha      Brazil   07/01/2011            2018     78 1,72 m
## 983   983        Gabriel      Brazil   07/01/2016            2019     77 184 cm
## 999   999        Marcelo      Brazil   07/01/2012            2019     77 182 cm
## 1013 1013          Naldo      Brazil   08/28/2016            2020     77 188 cm
## 1090 1090       Juanfran       Spain   07/01/2014            2017     77 179 cm
## 1115 1115 Carlos Sánchez    COLOMBIA   08/15/2014            2018     77 182 cm
##         Weight Preffered_Foot Birth_Date Age       Work_Rate
## 643  76.603 kg              2 07/25/1987  29   Medium / High
## 671  75.327 kg              2 05/10/1984  32    Low / Medium
## 698    74.8 kg              2 03/02/1983  34     High / High
## 768  86.665 kg              2 05/24/1990  26 Medium / Medium
## 862   68498 gr              2 09/07/1985  31 Medium / Medium
## 983  80.426 kg              1 09/18/1993  23 Medium / Medium
## 999  74.769 kg              2 07/27/1989  27   Medium / High
## 1013 84.653 kg              2 08/25/1988  28 Medium / Medium
## 1090 72.425 kg              2 09/11/1988  28   High / Medium
## 1115 82.453 kg              1 02/06/1986  31       Med / Med
```

Si escogemos uno aleatóriamente, por ejemplo el jugador "Fernando".

```
fifa.df.sub[fifa.df.sub$Name == "Fernando",]
```

```
##        ID     Name Nationality Club_Joining Contract_Expiry Rating Height
## 578   578 Fernando      Brazil   07/18/2016            2021     79 175 cm
## 643   643 Fernando      Brazil   06/27/2014            2019     79 183 cm
## 9188 9188 Fernando       Spain   07/05/2016            2017     66 185 cm
##         Weight Preffered_Foot Birth_Date Age       Work_Rate
## 578  80.652 kg              2 03/03/1992  25      Low / High
## 643  76.603 kg              2 07/25/1987  29   Medium / High
## 9188 79.368 kg              2 06/10/1990  26 Medium / Medium
```

Podemos ver comparando el resto de variables que muestra a tres jugadores diferentes. Como es lógico pensar que Fernando es un nombre común, vamos a probar con otro como "Lisandro López".

```
fifa.df.sub[fifa.df.sub$Name == "Lisandro López",]
```

```
##      ID           Name Nationality Club_Joining Contract_Expiry Rating Height
## 542 542 Lisandro López   Argentina   09/03/2013            2021     79 187 cm
## 698 698 Lisandro López   Argentina   01/04/2016            2019     79 174 cm
##        Weight Preffered_Foot Birth_Date Age     Work_Rate
## 542 80.387 kg              2 09/01/1989  27 Medium / High
## 698   74.8 kg              2 03/02/1983  34   High / High
```

Viendo por el resultado, que también estamos hablando de dos jugadores con variables diferentes.

Tras lo visto, y viendo que la calidad del dato es bastante pobre, no podemos confirmar que existan observaciones duplicadas en la muestra.

Normalización de los datos cuantitativos

En este apartado se va a realizar la normalización de la variables cuantitativas con objeto de uniformizar su formato.

Rating

La variable numérica *rating*, incluye la valoración del jugador dentro de un rango entre 0 y 100. Un vistazo a sus valores máximos y mínimo nos puede mostrar si están dentro del rango esperado.

```
str(fifa.df.sub$Rating)
```

```
##  int [1:17588] 94 93 92 92 92 90 90 90 90 89 ...
```

```
summary(fifa.df.sub$Rating)[c(1,6)]
```

```
## Min. Max.
##   45   94
```

Viendo que todos sus valores se encuentran dentro del intervalo correcto.

Height

La variable que muestra la altura de los jugadores, height, debe de estar en cm con un formato de 3 dígitos sin decimales.

```
head(fifa.df.sub$Height,20)
```

```
##  [1] "1,85 m" NA       "1,74 m" "1,82 m" "1,93 m" NA       "1,85 m" "1,83 m"
##  [9] NA       "1,99 m" "1,92 m" "1,73 m" "1,74 m" "180 cm" "184 cm" "183 cm"
## [17] "1,83 m" "173 cm" "191 cm" "176 cm"
```

Además, tenemos que tener especial cuidado debido a que las medidas en metros, no cumplen con una homegeneidad de formato, como se puede ver en el siguiente registro

```
fifa.df[29,c("Name", "Height")]
```

```
##            Name Height
## 29 Philipp Lahm  1,7 m
```

Vemos que hay observaciones con medidas en centímetros, y otras en metros. Eliminaremos la unidad que aparece al final

```
fifa.df.sub$Height <- str_replace_all(fifa.df.sub$Height, "[cm]", "")
head(fifa.df.sub$Height, 20)
```

```
##  [1] "1,85 " NA      "1,74 " "1,82 " "1,93 " NA      "1,85 " "1,83 " NA
## [10] "1,99 " "1,92 " "1,73 " "1,74 " "180 "  "184 "  "183 "  "1,83 " "173 "
## [19] "191 "  "176 "
```

Conseguimos trasformar las medidas de metros a centímetros eliminando la coma y aplicando una función que elimina los espacios en blanco iniciales y finales.

```
fifa.df.sub$Height <- trimws(str_replace(fifa.df.sub$Height, "[,]", ""))
head(fifa.df.sub$Height, 20)
```

```
##  [1] "185" NA    "174" "182" "193" NA    "185" "183" NA    "199" "192" "173"
## [13] "174" "180" "184" "183" "183" "173" "191" "176"
```

Como paso de comprobación, vamos a ver si existe alguna medida cuya longitud sea superior a 3 dígitos.

```
tail(fifa.df.sub[nchar(fifa.df.sub$Height)>3, c("Height")])
```

```
## [1] NA NA NA
```

Apareciendo los NA que trataremos más adelante.

El siguiente paso será transformar la medida a numérico.

```
fifa.df.sub$Height <- as.numeric(fifa.df.sub$Height)
str(fifa.df.sub$Height)
```

```
##  num [1:17588] 185 NA 174 182 193 NA 185 183 NA 199 ...
```

Por último, acabaremos de ajustar aquellas observaciones que originalmente carecían de los dos decimales.

```
fifa.df.sub$Height <- ifelse(nchar(fifa.df.sub$Height) == 1, fifa.df.sub$Height*100, ifelse(nchar(fifa.
```

```
fifa.df.sub[29, c("Name", "Height")]
```

```
##            Name Height
## 29 Philipp Lahm    170
```

Weight

La última variable numérica muestra el peso de los jugadores y debe de estar **expresada en kg, sin decimales**. Los pasos son similares a la variable Height. Esta vez, vamos a realizar la transformación haciendo uso de pipes para concatenar pasos.

```
str(fifa.df.sub$Weight)
```

```
##  chr [1:17588] NA "72475 gr" "68884 gr" "85511 gr" NA "82.671 kg" NA ...
```

De la observación de los primeros registro, podemos ver que la variable se encuentra en formato de string, y que contiene la unidad de medida kg o gramos. Los valores están expresados con tres decimales que deberemos

de reducir a dos. Además, en las observaciones dadas en kg, podemos encontrar que el separador decimal es el punto o bien la coma.

Primeramente vamos a diferenciar las medidas de gramos del resto, para poder operar con ellas posteriormente.

```
  fifa.df.sub <-  fifa.df.sub %>%
    mutate(Weight_Unit = ifelse(str_detect(Weight, "gr"),"gr",""))
  fifa.df.sub[1:10, c("Weight", "Weight_Unit")]
```

```
##       Weight Weight_Unit
## 1       <NA>        <NA>
## 2   72475 gr          gr
## 3   68884 gr          gr
## 4   85511 gr          gr
## 5       <NA>        <NA>
## 6  82.671 kg
## 7       <NA>        <NA>
## 8   74683 gr          gr
## 9  95.429 kg
## 10  91394 gr          gr
```

A continuación vamos a realizar los siguientes pasos encadenados:

1. Eliminación de la unidad en la variable Weight.
2. Eliminación de los espacios en blanco en cada observación de la variable.
3. Sustitución de las comas por los puntos como separadores decimales.
4. Conversión de la variable en Numérica.
5. Añadir el resultado en una nueva columna, "Weight2", que nos permitirá poder comparar con la columna original.

```
  fifa.df.sub$Weight2 <-
    fifa.df.sub$Weight %>%
    str_replace_all(c(","="."",  "kg"="", "gr"="")) %>%
    trimws() %>%
    as.numeric()
```

```
fifa.df.sub[1:10, c("Weight", "Weight_Unit", "Weight2")]
```

```
##       Weight Weight_Unit   Weight2
## 1       <NA>        <NA>        NA
## 2   72475 gr          gr 72475.000
## 3   68884 gr          gr 68884.000
## 4   85511 gr          gr 85511.000
## 5       <NA>        <NA>        NA
## 6  82.671 kg                82.671
## 7       <NA>        <NA>        NA
## 8   74683 gr          gr 74683.000
## 9  95.429 kg                95.429
## 10  91394 gr          gr 91394.000
```

El siguiente paso es utilizar la columna Weight_Unit para convertir los valores en gramos a kilos.

```
gr_to_kg <- function(value, unit){
  if (is.na(unit)==FALSE) {
    if (unit == "gr") {
      return(trunc(value/1000))
```

```
    }
    else{
      return(trunc(value))
    }
  }
  else
  {return(NA)}
}
for(i in 1:nrow(fifa.df.sub)){
  fifa.df.sub[i,"Weight2"] <- gr_to_kg(fifa.df.sub[i,"Weight2"], fifa.df.sub[i,"Weight_Unit"])
}
```

Para finalizar, borramos las columnas que ya no nos sirven.

A modo de comprobación, nos fijamos en los valores extremos de la serie y en el formato que nos devuelve.

```
fifa.df.sub <- subset(fifa.df.sub, select = -c(Weight, Weight_Unit))
names(fifa.df.sub)[names(fifa.df.sub) == 'Weight2'] <- 'Weight'
#fifa.df.sub <- fifa.df.sub %>% rename(Weight = Weight2)
summary(fifa.df.sub$Weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   48.00   70.00   75.00   75.25   80.00  110.00       3
```

Normalización de los datos caulitativos

Name y Nationality

Si examinamos la variable Name, podemos comprobar que algunos de sus valores presentan espacios en blanco. Además de no tener una uniformidad con respecto al uso de mayúsculas, tal y como podemos ver a continuación.

```
filas <- c(314, 973, 998, 794)
fifa.df.sub[filas, "Name"]
```

```
## [1] "   Lucas pérez"    " Juan fernando"    "luís Hernández   "
## [4] "   Lamine sané"
```

Aplicamos la limpieza y normalización al campo Name y al Nacionality.

```
fifa.df.sub$Name <- fifa.df.sub$Name %>% trimws() %>% str_to_title()
fifa.df.sub$Nationality <- fifa.df.sub$Nationality %>% trimws() %>% str_to_title()
fifa.df.sub[filas, c("Name", "Nationality")]
```

```
##               Name Nationality
## 314    Lucas Pérez       Spain
## 973  Juan Fernando    Colombia
## 998 Luís Hernández       Spain
## 794    Lamine Sané     Senegal
```

Preffered_Foot

La variable preffered_foot contiene dos identificadores numéricos que tienen la siguiente correspondencia:

- 1 -> Left
- 2 -> Right

Vamos a realizar una transformación de la variable a tipo factor con los atributos descritos.

```
str(fifa.df.sub$Preffered_Foot)
```

```
##  int [1:17588] 2 1 2 2 2 2 2 1 2 1 ...
```

```
fifa.df.sub$Preffered_Foot <- factor(fifa.df.sub$Preffered_Foot)
levels(fifa.df.sub$Preffered_Foot) <- c("Left", "Right")
str(fifa.df.sub$Preffered_Foot)
```

```
##  Factor w/ 2 levels "Left","Right": 2 1 2 2 2 2 2 1 2 1 ...
```

Work_Rate

Examinamos la variable Work_Rate. Como se puede observar a continuación, muestra dos valores de tipo string, el primer valor como la valoración cualitativa en ataque y el segundo en defensa.

```
unique(fifa.df.sub$Work_Rate)
```

```
##  [1] "High / Low"      "Medium / Medium" "High / Medium"   "Medium / Low"
##  [5] "High / High"     "Med / Med"       "Medium / High"   "Low / High"
##  [9] "Low / Medium"    "Hig / Med"       "Low / Low"
```

De la observación de los valores que presenta la variable, podemos ver que no hay uniformidad entre ellos, presentando abreviaturas, como por ejemplo Hig y Med para High y Medium.

```
fifa.df.sub[fifa.df.sub$Work_Rate == 'Med / Med', 'Work_Rate'] = 'Medium / Medium'
fifa.df.sub[fifa.df.sub$Work_Rate == 'Hig / Med', 'Work_Rate'] = 'High / Medium'
summary(fifa.df.sub$Work_Rate)
```

```
##    Length     Class      Mode
##     17588 character character
```

Una vez obtenidas las nueve combinaciones válidas, podemos transformar la variable a factor.

```
fifa.df.sub$Work_Rate = factor(fifa.df.sub$Work_Rate)
levels(fifa.df.sub$Work_Rate)
```

```
## [1] "High / High"     "High / Low"      "High / Medium"   "Low / High"
## [5] "Low / Low"       "Low / Medium"    "Medium / High"   "Medium / Low"
## [9] "Medium / Medium"
```

Encontrando las 9 posibles combinaciones para esta variable.

Posibles inconsistencias y variables tipo fecha

Club_Joining

La variable Club_Joining tiene que estar dentro del rango: 1990 a 2017. Al examinarla, vemos que el tipo de datos es literal.

```
str(fifa.df.sub$Club_Joining)
```

```
##  chr [1:17588] "07/01/2009" "07/01/2004" "07/01/2013" "07/11/2014" ...
```

Así que el siguiente paso será transformar su tipo a fecha, tal y como le corresponde.

```
str(fifa.df.sub[, "Club_Joining"])
```

```
##  chr [1:17588] "07/01/2009" "07/01/2004" "07/01/2013" "07/11/2014" ...
```

```
fifa.df.sub$Club_Joining = dmy( fifa.df.sub[, "Club_Joining"])
```

```
## Warning: 6256 failed to parse.
```

```
str(fifa.df.sub[, "Club_Joining"])
```

```
##  Date[1:17588], format: "2009-01-07" "2004-01-07" "2013-01-07" "2014-11-07" "2011-01-07" ...
```

Una vez convertida la variable, vamos a comprobar si existen fechas fuera de rango

```
summary(fifa.df.sub[year(fifa.df.sub$Club_Joining) <= 1990 || year(fifa.df.sub$Club_Joining) > 2017, "C
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      NA      NA      NA      NA      NA      NA
```

Viendo que todas las fechas se encuentran dentro del rango.

Contract_Expiry >= Club_Joining?

En este apartado, vamos a mirar si se cumple **la regla de integridad** de que no exista ningún jugador cuyo año de expiración de contrato sea inferior al año de inicio del contrato. Lo ideal hubiera sido poder comparar a nivel de fechas, pero en nuestro caso, la variable *Contract_Expiry* **sólo recoge el año**. De todas maneras, teniendo en cuenta que los jugadores son renovados por temporadas, la afectación de no bajar a máxima granularidad no tiene que ser importante.

```
nrow(na.omit(fifa.df.sub[fifa.df.sub$Contract_Expiry < year(fifa.df.sub$Club_Joining), c("Club_Joining"
```

```
## [1] 0
```

No encontrándose ningún caso. Lo que si que encontramos es un registro con valor desconocido que sería interesante tratar.

```
fifa.df.sub[is.na(fifa.df.sub$Contract_Expiry)== TRUE, ]
```

```
##      ID          Name Nationality Club_Joining Contract_Expiry Rating Height
## 384 384 Didier Drogba Ivory Coast         <NA>              NA     81    189
##     Preffered_Foot Birth_Date Age    Work_Rate Weight
## 384          Right 03/11/1978  39 Medium / Low     80
```

Revisar si la edad corresponde a la fecha de nacimiento

Vamos a verificar que la variable Age, a fecha 01/01/2017 tiene su correspondencia con la que incorpora el registro en la variable Birth_Date. Primero de todo, si examinamos esta última variable, encontramos que tenemos que realizar una transformación de tipo de datos de litaral a fecha (casting).

```
str(fifa.df.sub$Birth_Date)
```

```
##  chr [1:17588] "02/05/1985" "06/24/1987" "02/05/1992" "01/24/1987" ...
```

```
fifa.df.sub$Birth_Date <- dmy(fifa.df.sub$Birth_Date)
```

```
## Warning: 10502 failed to parse.
```

```
str(fifa.df.sub$Birth_Date)
```

```
##  Date[1:17588], format: "1985-05-02" NA "1992-05-02" NA NA "1990-07-11" NA NA "1981-03-10" ...
```

Ahora que ya tenemos la variable en su formato correcto, podemos calcular la edad en el primer día del año 2017 y compararla con la almacenada.

```
current_date <- ymd(20170101)
fifa.df.sub<- fifa.df.sub %>%
  mutate(Calculated_Age = trunc(as.numeric(as.period(interval(fifa.df.sub$Birth_Date, current_date)), u
fifa.df.sub[1:10, c("Name", "Birth_Date", "Age", "Calculated_Age")]
```

```
##                  Name Birth_Date Age Calculated_Age
## 1   Cristiano Ronaldo 1985-05-02  32             31
## 2        Lionel Messi       <NA>  29             NA
## 3              Neymar 1992-05-02  25             24
## 4         Luis Suárez       <NA>  30             NA
## 5        Manuel Neuer       <NA>  31             NA
## 6             De Gea 1990-07-11  26             26
## 7  Robert Lewandowski       <NA>  28             NA
## 8         Gareth Bale       <NA>  27             NA
## 9  Zlatan Ibrahimovic 1981-03-10  35             35
## 10   Thibaut Courtois 1992-11-05  24             24
```

En los casos en los que no tenemos año de nacimiento, la fórmula no ha podido calcular la diferencia de tiempo. En otros casos, como por ejemplo el primer registro, vemos que la edad calculada difiere de la edad almacenada en 1 dígito. Este resultado podría ser debido a si emplearon redondeo en el cálculo de la edad (nosotros hemos truncado la cifra a su parte entrena), o si utilizaron una fecha diferente al primero de enero del 2017. En todo caso, vamos a revisar cuantas observaciones se ven afectadas por la diferencia entre la edad calculada y la almacenada.

```
head(na.omit(fifa.df.sub[fifa.df.sub$Calculated_Age!=fifa.df.sub$Age, c("Name", "Birth_Date", "Age", "Ca
```

```
##                  Name Birth_Date Age Calculated_Age
## 1   Cristiano Ronaldo 1985-05-02  32             31
## 3              Neymar 1992-05-02  25             24
## 12       Eden Hazard 1991-07-01  26             25
## 24        Toni Kroos 1990-04-01  27             26
## 34         Jan Oblak 1993-07-01  24             23
## 39      Ivan Rakitic 1988-10-03  29             28
```

```
paste("Número de registros con Edad diferente: ", nrow(na.omit(fifa.df.sub[fifa.df.sub$Calculated_Age!=
```

```
## [1] "Número de registros con Edad diferente:  2077"
```

Vamos a actualizar la variable Age con los valores calculados.

```
fifa.df.sub[1:10, c("Name", "Birth_Date", "Age", "Calculated_Age")]
```

```
##                  Name Birth_Date Age Calculated_Age
## 1   Cristiano Ronaldo 1985-05-02  32             31
## 2        Lionel Messi       <NA>  29             NA
## 3              Neymar 1992-05-02  25             24
## 4         Luis Suárez       <NA>  30             NA
## 5        Manuel Neuer       <NA>  31             NA
## 6             De Gea 1990-07-11  26             26
## 7  Robert Lewandowski       <NA>  28             NA
## 8         Gareth Bale       <NA>  27             NA
## 9  Zlatan Ibrahimovic 1981-03-10  35             35
## 10   Thibaut Courtois 1992-11-05  24             24
```

```
#fifa.df.sub$Age2 <-fifa.df.sub$Age
#fifa.df.sub[fifa.df.sub$Calculated_Age!=fifa.df.sub$Age & !is.na(fifa.df.sub$Calculated_Age), "Age2"] 
real_age <- function(age1, age2){
```

```
  if (age1 != age2 & !is.na(age2)) {
    return(age2)
  } else {
    return(age1)
  }
}
for(i in 1:nrow(fifa.df.sub)){
  fifa.df.sub[i,"Age"] <- real_age(fifa.df.sub[i,"Age"], fifa.df.sub[i,"Calculated_Age"])
}
fifa.df.sub[1:10, c("Name", "Birth_Date", "Age", "Calculated_Age")]
```

```
##                  Name Birth_Date Age Calculated_Age
## 1   Cristiano Ronaldo 1985-05-02  31             31
## 2         Lionel Messi       <NA>  29             NA
## 3               Neymar 1992-05-02  24             24
## 4          Luis Suárez       <NA>  30             NA
## 5         Manuel Neuer       <NA>  31             NA
## 6               De Gea 1990-07-11  26             26
## 7  Robert Lewandowski       <NA>  28             NA
## 8          Gareth Bale       <NA>  27             NA
## 9  Zlatan Ibrahimovic 1981-03-10  35             35
## 10   Thibaut Courtois 1992-11-05  24             24
```

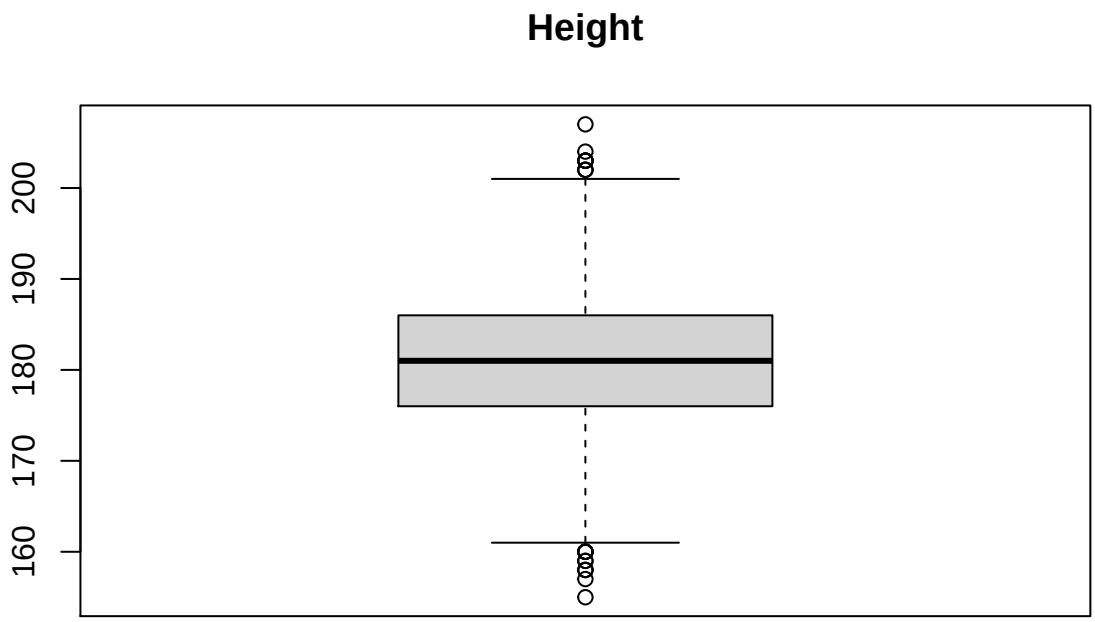
Ya hemos conseguido actualizar la variable Age con los valores calculados para aquellos casos en los que eran diferentes y existían valores. Es el momento de borrar la columna temporal Calculated_Age que nos ha servido para el cálculo.

```
fifa.df.sub <- select(fifa.df.sub, -Calculated_Age)
```

Valores atípicos

Vamos a revisar si existen valores atípicos para la variables Height.

```
boxplot(fifa.df.sub$Height, main = "Height", color= "gray")
```



Tanto el bloxplot de la variable Height, como de la Weight nos marcan valores fuera del 1.5 veces el rango intercuartílico, que podríamos considerar outliers (!). En concreto, si nos fijamos en la variable Height, podemos ver que corresponden a la siguiente lista.

```
boxplot.stats(fifa.df.sub$Height)$out %>% unique() %>% sort()
```

```
## [1] 155 157 158 159 160 202 203 204 207
```

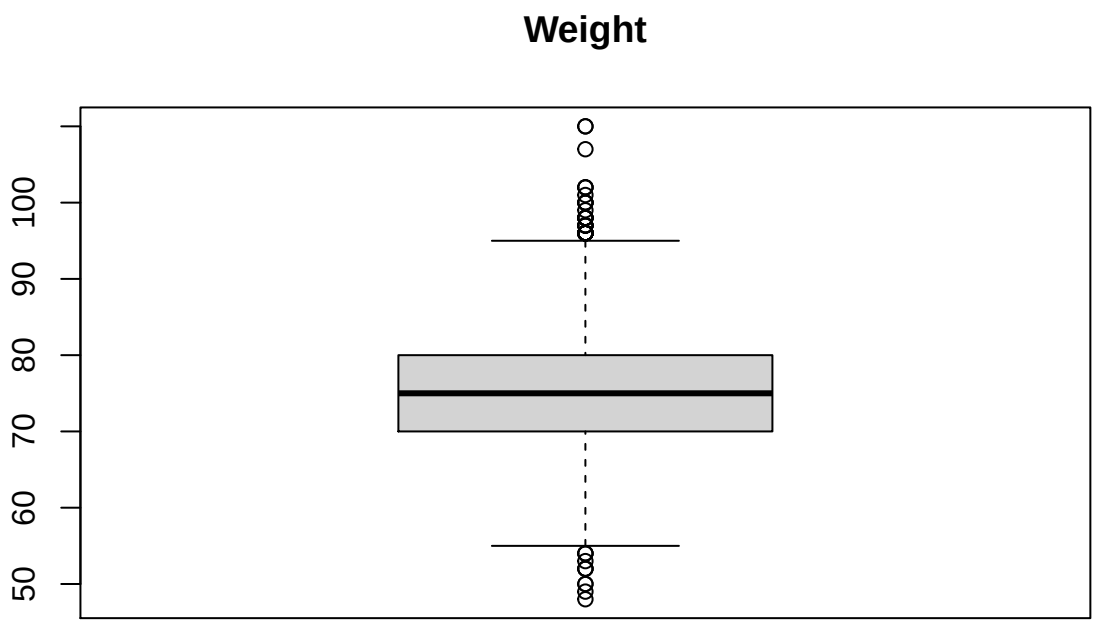
```
summary(fifa.df.sub$Height)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   155.0   176.0   181.0   181.1   186.0   207.0       3
```

Aunque la media es una medida de tendencia central que se ve fuertemente influenciada por los valores extremos, en este caso, vemos que media y mediana prácticamente coinciden. Por esa razón, no parece indicado la eliminación de los valores extremos encontrados por el bloxplot.

El paso siguiente será repetir la búsqueda de outliers para la variable Weight

```
boxplot(fifa.df.sub$Weight, main = "Weight", color= "gray")
```




```
boxplot.stats(fifa.df.sub$Weight)$out %>% unique() %>% sort()
```

```
##  [1]  48  49  50  52  53  54  96  97  98  99 100 101 102 107 110
```

```
summary(fifa.df.sub$Weight)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   48.00   70.00   75.00   75.25   80.00  110.00       3
```

En este caso, la desviación entre media y mediana es también pequeño (menos de un 0.4 %), **no justificándose la eliminación** de los valores extremos de la serie numérica.

Imputación de valores

En este capítulo vamos a inferir los valores desconocidos de las variables Height y Weight, a partir del valor conocido de una de ellas utilizando la regresión lineal.

Inferir Height a partir de Weight

Examinemos primero los valores desconocidos de la serie.

```
gamer_na_height <- fifa.df.sub[is.na(fifa.df.sub$Height) == TRUE, 'ID']
fifa.df.sub[gamer_na_height, c("Name", "Height")]
```

```
##                 Name Height
## 2       Lionel Messi     NA
## 6             De Gea     NA
## 9 Zlatan Ibrahimovic     NA
```

Para inferir los valores desconocidos, vamos a utilizar un modelo de regresión lineal, utilizando la variable peso como variable conocida.

```
fmla <- fifa.df.sub$Height ~ fifa.df.sub$Weight
lineal.model <- lm(fmla, data = fifa.df.sub)
lineal.model
```

```
##
## Call:
## lm(formula = fmla, data = fifa.df.sub)
##
## Coefficients:
##        (Intercept)  fifa.df.sub$Weight
##           125.8960              0.7336
```

A modo de prueba podemos observar que el coeficiente es positivo (0.73), lo que indicaría que peso y altura se relacionan directamente, a más peso, más altura, aunque lo lógica nos dice que esto no siempre se cumple.

La métrica de R^2, cuyo rango va de 0 a 1, nos determinará cuanto de bien se ajusta nuestro modelo a los datos. La bonanza del modelo nos lo determina la proximidad a 1 del resultado. En este caso es ligeramente superior a 0.5, lo cual nos indica un pobre ajuste.

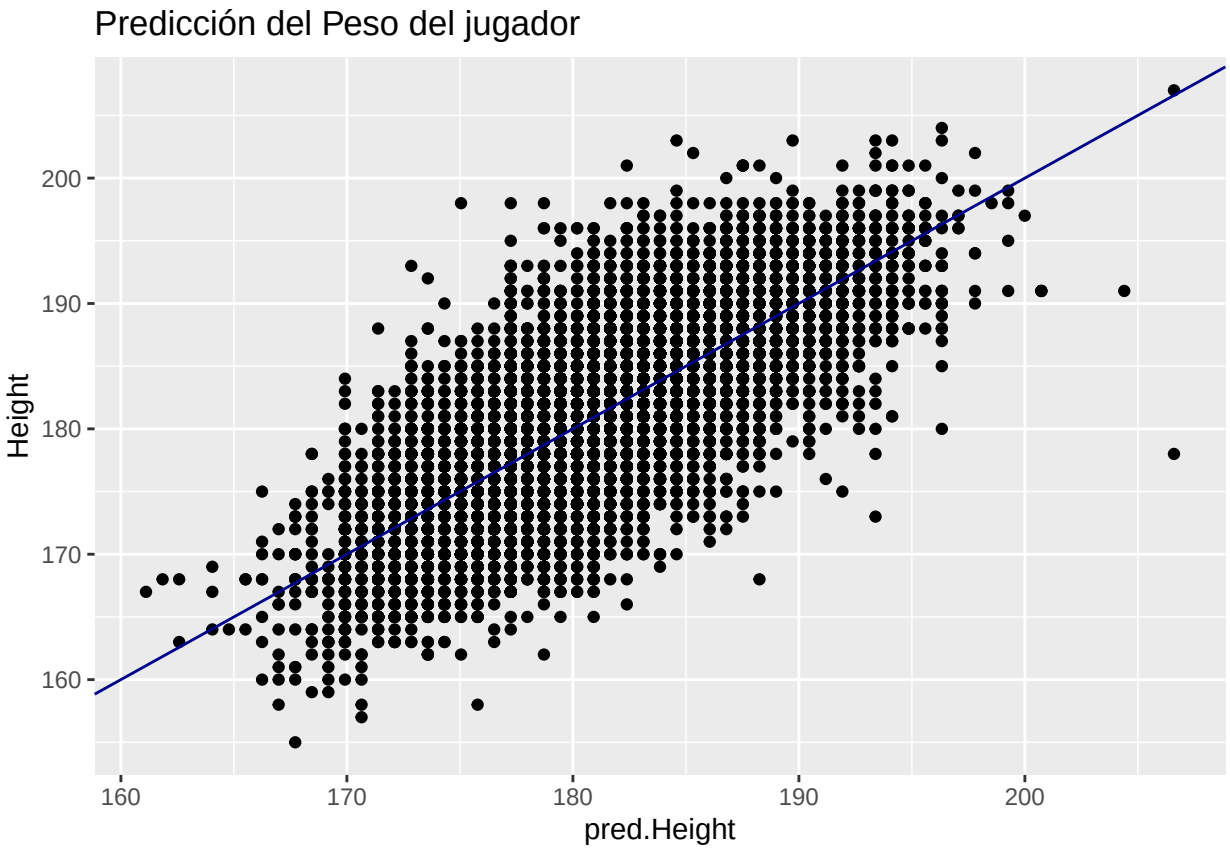
```
summary(lineal.model)$r.squared
```

```
## [1] 0.574707
```

Realizamos la predicción, y examinaremos el modelo gráficamente para ver si la predicción se ajusta a los datos que ya tenemos. Cuanto menos disperso estén los puntos respecto de la recta, mejor será el ajuste de nuestra regresión.

```
fifa.df.sub$pred.Height <- predict(lineal.model, fifa.df.sub)

ggplot(fifa.df.sub, aes(x = pred.Height, y = Height)) +
  geom_point() +
  geom_abline(color= "darkblue") +
  ggtitle("Predicción del Peso del jugador")
```

```
## Warning: Removed 6 rows containing missing values (geom_point).
```



Examinamos el resultado de la predicción para los jugadores con valores de altura desconocidos. Hemos de aplicar el mismo formato numérico, sin decimales, que ya aplicamos en su momento a los valores originales.

```
fifa.df.sub[is.na(fifa.df.sub$Height), c("Name", "Height", "pred.Height")]
```

```
##                 Name Height pred.Height
## 2       Lionel Messi     NA    178.7187
## 6             De Gea     NA    186.0552
## 9 Zlatan Ibrahimovic     NA    195.5927
```

Por último, vamos a actualizar la variable de altura con los nuevos valores inferidos.

```
fifa.df.sub[is.na(fifa.df.sub$Height), "Height"] <- trunc(fifa.df.sub[is.na(fifa.df.sub$Height), "pred.
paste("Número de filas con altura desconocida: ", nrow(fifa.df.sub[is.na(fifa.df.sub$Height),]))
```

```
## [1] "Número de filas con altura desconocida:  0"
```

Mostramos los valores resultantes.

Inferir Weight a partir de Height

A continuación, realizaremos la predicción del peso a partir de la altura del jugador, siguiendo el esquema utilizado en el apartado anterior. Previamente, vamos a registrar las observaciones afectadas.

```
gamer_na_weight <- fifa.df.sub[is.na(fifa.df.sub$Weight) == TRUE, 'ID']
fifa.df.sub[gamer_na_weight, c("Name", "Weight")]
```

```
##                 Name Weight
## 1  Cristiano Ronaldo     NA
```

```
## 5       Manuel Neuer     NA
## 7 Robert Lewandowski     NA
```

```
fmla <- fifa.df.sub$Weight ~ fifa.df.sub$Height
lineal.model <- lm(fmla, data = fifa.df.sub)
summary(lineal.model)$r.squared
```

```
## [1] 0.574829
```

Se mantiene el R cuadrado. A continuación, vamos a predecir el peso a partir de la altura.

```
fifa.df.sub$pred.Weight <- predict(lineal.model, fifa.df.sub)
fifa.df.sub[is.na(fifa.df.sub$Weight), c("Name", "Weight", "pred.Weight")]
```

```
##                 Name Weight pred.Weight
## 1  Cristiano Ronaldo     NA    78.30424
## 5       Manuel Neuer     NA    84.57250
## 7 Robert Lewandowski     NA    78.30424
```

El siguiente paso que daremos será actualizar la variable del peso del jugador que presenta valores desconocidos con las predicciones encontradas. Estas predicciones se tienen que adaptar al formato de los datos de origen. Para ello, truncaremos el valor quedándonos con la parte entera.

```
fifa.df.sub[is.na(fifa.df.sub$Weight), "Weight"] <- trunc(fifa.df.sub[is.na(fifa.df.sub$Weight), "pred.
fifa.df.sub[gamer_na_weight, c("Name", "Weight", "pred.Weight")]
```

```
##                 Name Weight pred.Weight
## 1  Cristiano Ronaldo     78    78.30424
## 5       Manuel Neuer     84    84.57250
## 7 Robert Lewandowski     78    78.30424
```

Como última comprobación, miraremos que las dos variables tratadas, Height y Weight, están libres de valores desconocidos, para finalmente, eliminar del modelo de datos las dos variables de predicción utilizadas y que ya no aportan nada al modelo.

```
paste("Número de filas con peso desconocido: ", nrow(fifa.df.sub[is.na(fifa.df.sub$Weight),]))
```

```
## [1] "Número de filas con peso desconocido:  0"
```

```
paste("Número de filas con altura desconocida: ", nrow(fifa.df.sub[is.na(fifa.df.sub$Height),]))
```

```
## [1] "Número de filas con altura desconocida:  0"
```

```
fifa.df.sub[, c("pred.Weight", "pred.Height")] <- NULL
```

Estudio descriptivo de las variables cuantitativas

En este capítulo, vamos a realizar el estudio de las variables cuantitativas visualizando sus medidas de tendencia centrales.

```
col_names <- c('Contract_Expiry', 'Rating', 'Height', 'Age', 'Weight')
col_names
```

```
## [1] "Contract_Expiry" "Rating"          "Height"          "Age"
## [5] "Weight"
```

La variable **Contract_Expiry**, aunque se encuentra en formato numérico, no debemos olvidar que corresponde a un año y por tanto, su naturaleza es ser parte de una fecha. Por otro lado, es una variable que cuenta con un valor desconocido, que nos obliga a tratarlo para poder realizar el estudio. Podríamos optar

por eliminar el registro, o no tenerlo en cuenta para este estudio dado que corresponde a una sola observación entre 17588, pero estaríamos perdiendo información relevante de este jugador. Por ello, considero más práctico inferir la media de la población, asumiendo el riesgo de introducir un dato por criterios estadísticos, que no se corresponderá con la realidad.

```
fifa.df.sub[is.na(fifa.df.sub$Contract_Expiry)== TRUE, "Contract_Expiry"] =
round(mean(fifa.df.sub[is.na(fifa.df.sub$Contract_Expiry)== FALSE, "Contract_Expiry"] ),0)
```

```
quantitative_study <- function(df, col_names) {
  media <- as.data.frame(lapply(df[, col_names], mean)) %>% mutate(estimator=c("media"))
  mediana <- as.data.frame(lapply(df[, col_names], median)) %>% mutate(estimator=c("mediana"))
  winsor_media <- as.data.frame(lapply(df[, col_names], winsor.mean)) %>% mutate(estimator=c("winsor_me
  df_out <- merge(media, mediana, all=TRUE)
  df_out <- merge(df_out, winsor_media, all=TRUE)
  return(df_out)
}

print(quantitative_study(fifa.df.sub, col_names))
```

```
##   Contract_Expiry   Rating   Height      Age   Weight    estimator
## 1        2018.602 66.16267 181.0776 25.00250 75.21282 winsor_media
## 2        2018.899 66.16619 181.1055 25.34222 75.25273        media
## 3        2019.000 66.00000 181.0000 25.00000 75.00000      mediana
```

Análisis de Componentes Principales (ACP)

En este capítulo vamos a realizar un análisis de componentes principales (PCA en inglés) sobre las variables "Rating", "Height", "Weight" y "Age". Aunque primeramente vamos a mirar la matriz de correlaciones, dado que para que la ACP sea efectiva, tiene que existir un alto grado de correlación entre las variables.

```
cor(fifa.df.sub[, col_names])
```

```
##                 Contract_Expiry     Rating      Height         Age      Weight
## Contract_Expiry      1.00000000 0.04743154 -0.08068535 -0.11842924 -0.05316544
## Rating               0.04743154 1.00000000  0.04713470  0.45693974  0.13945142
## Height              -0.08068535 0.04713470  1.00000000  0.07661418  0.75820508
## Age                 -0.11842924 0.45693974  0.07661418  1.00000000  0.22296343
## Weight              -0.05316544 0.13945142  0.75820508  0.22296343  1.00000000
```

Las dos únicas variables correlacionadas directamente son Altura y Peso.

Antes de lanzar el análisis de PCA hemos de pensar que la matriz de datos está formada por variables con diferentes magnitudes y rangos. Por ello, vamos a utilizar la matriz de correlaciones que transformará las variables en estandarizadas de media cero y desviación típica uno.

Lancemos el análisis y estudiemos su resultado.

```
col_names <- c( "Rating", "Height", "Weight", "Age")
fifa.pca <- prcomp(na.omit(fifa.df.sub[, col_names]), center = TRUE, scale. = TRUE)
summary(fifa.pca)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4
## Standard deviation     1.3775 1.1543 0.7368 0.47682
## Proportion of Variance 0.4744 0.3331 0.1357 0.05684
## Cumulative Proportion  0.4744 0.8075 0.9432 1.00000
```

Vemos que la proporción de varianza no proporciona buenos resultados. Como hemos visto anteriormente, la correlación entre las variables era escasa. No obstante, fijándonos en la varianza acumulada podemos observar que con los dos primeros componentes podemos representar el 80,8% del modelo.

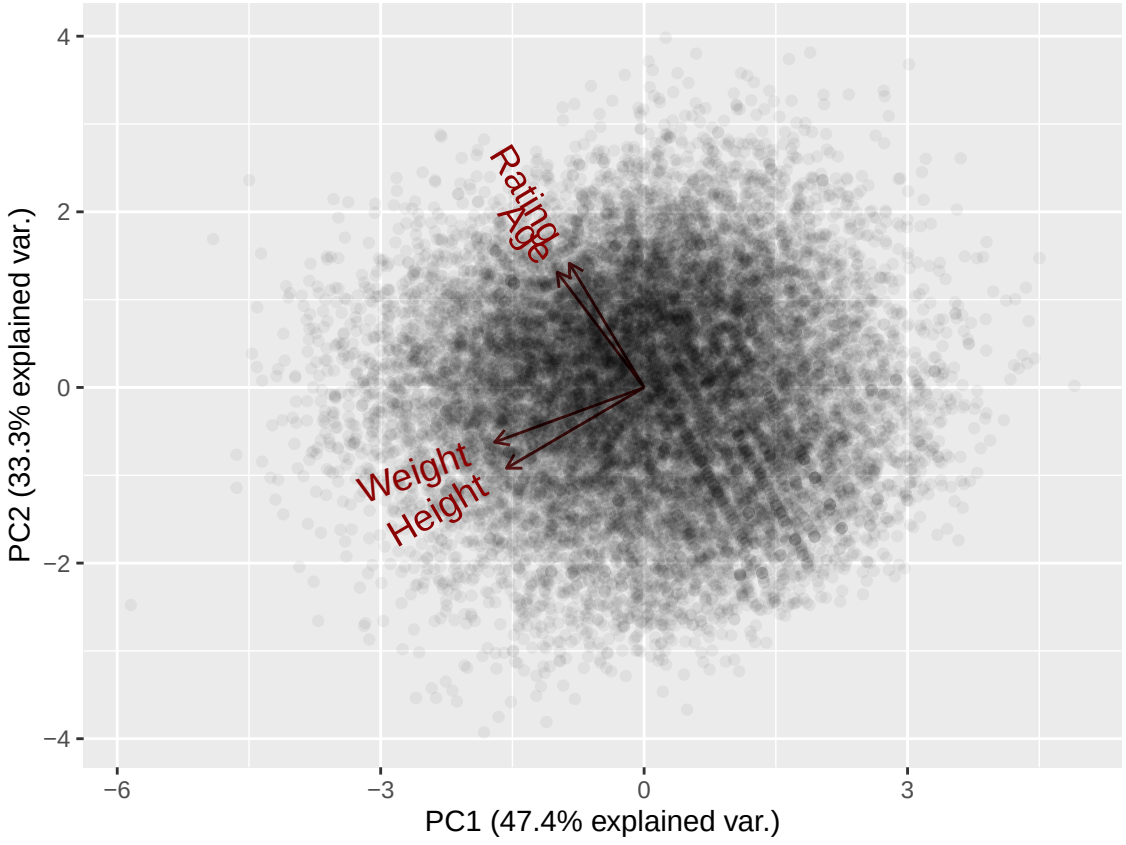
Veamos los vectores propios.

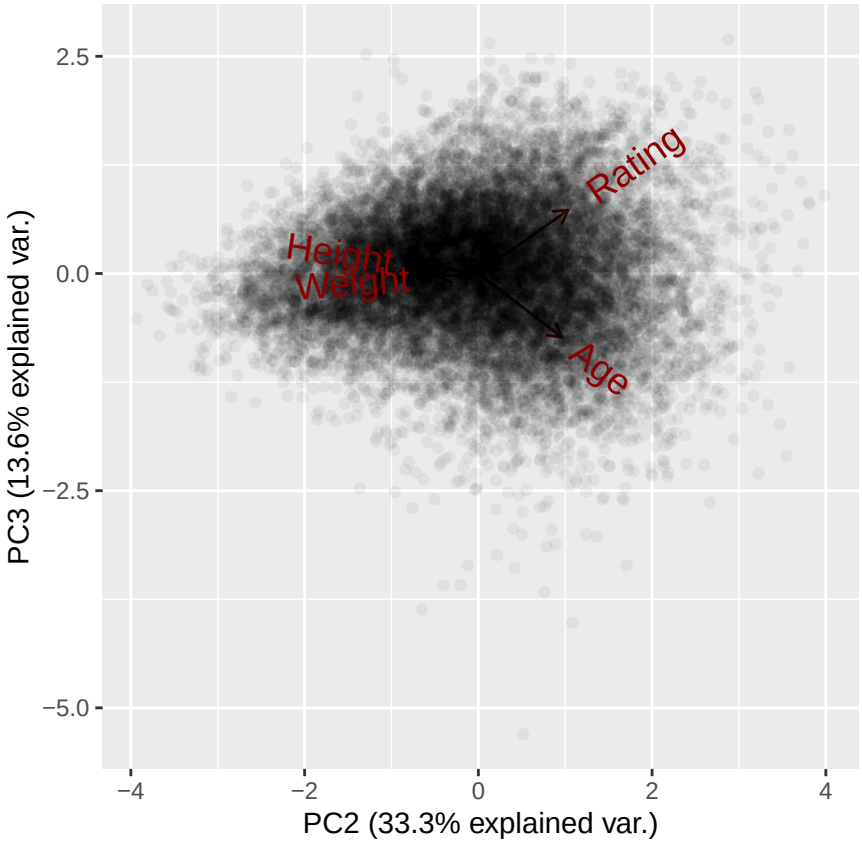
```
fifa.pca$rotation
```

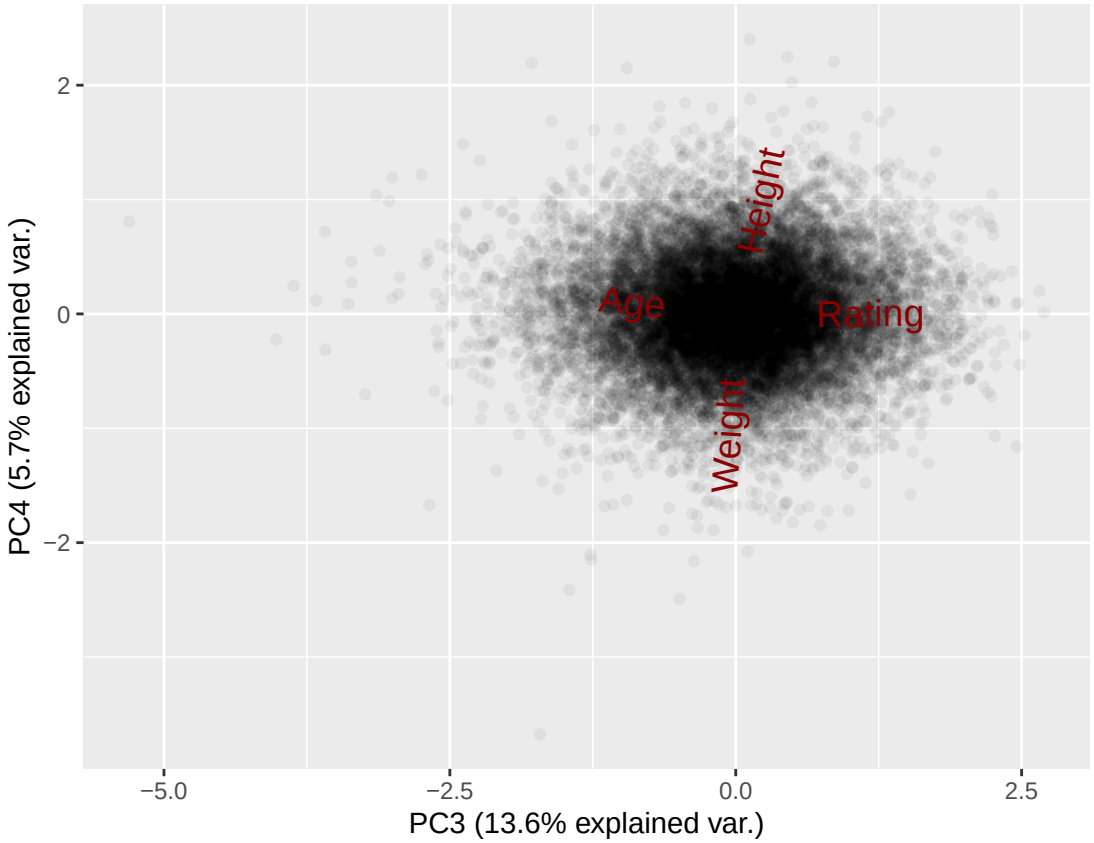
```
##                PC1        PC2         PC3          PC4
## Rating -0.3202585  0.6348940  0.70304710  0.008295396
## Height -0.5899523 -0.4129870  0.09610398  0.687140488
## Weight -0.6407452 -0.2800861 -0.03051664 -0.714189113
## Age    -0.3726075  0.5898331 -0.70395848  0.133052797
```

Para saber que componentes utilizar, vamos a verlo visualmente.

```
fifa_biplot_chart <- function(x, y){
  ggbiplot(fifa.pca, choices = x:y, obs.scale = 1, var.scale = 1,
           pc.biplot =TRUE, labels.size = 3, var.axes = TRUE,
           varname.size=5, alpha = 0.05)

}

par(mfrow= c(2,2))

for(i in 1:3) {
  print(fifa_biplot_chart(i,i+1))
  }
```




2.5
0.0
−2.5
−5.0
−4
−2
0
2
4
PC3 (13.6% explained var.)
PC2 (33.3% explained var.)
Rating
Height
Weight
Age



Del examen del primer gráfico, que representa a las dos primeras componentes, podemos observar lo siguiente: - 1. Con la primera componente se explicaría el 47,4% de la varianza, siendo la que más contribuye al modelo. - 2. Podemos observar también, una proximidad entre el par de variables Weight y Height, por un lado, y Rating y Age por el otro. Una mayor proximidad indica un alto grado de correlación entre las variables, algo que ya habíamos visto al examinar la matriz de correlación. - 3. La longitud de las flechas indican la importancia que tienen las variables originales en las componentes.

Por último, encontramos un punto residual fuera de la nube de puntos, que se repite en los tres gráficos de componentes. Este punto, podría ser clasificado como outlier y sacado fuera del modelo de datos.

Veamos los valores de las variables para este punto. Primero encontramos la proyección para este punto, fijándonos que es el valor mínimo que toma PC1.

```
projection <- fifa.pca$x[fifa.pca$x[, 1] == min((fifa.pca$x)[, 1]), ]
print(projection)
```

```
##        PC1        PC2        PC3        PC4
## -5.8452177 -2.4787299 -0.2470206 -0.8274027
```

Encontramos la posición del valor mínimo encontrado.

```
pos <-min(which(fifa.pca$x[, 1] == projection[1]))
```

A continuación vamos a calcular la matriz original de valores a partir de la proyección y de los vectores propios, teniendo en cuenta que los valores están centrados y reescalados.

```
reverse_pca <- t(t(fifa.pca$x %*% t(fifa.pca$rotation))* fifa.pca$scale + fifa.pca$center)
reverse_pca[pos,]
```

```
## Rating Height Weight    Age
```

```
##      67     207     110      29
```

Podemos ver que es un punto en el que tanto las variables peso como la altura muestran valores anormalmente altos.

Archivo final