Preprocesado de datos del Fifa 2017

Jesús González

25/3/2021

Contents

Introducción

Esta actividad se centra en la etapa de preprocesado necesario para preparar los datos para su posterior análisis. Para ello, se detectan y se corrigen posibles errores, inconsistencias y valores perdidos. Se presenta, además, una breve estadística descriptiva y se realizará un análisis de componentes principales (PCA) con algunas de las variables cuantitativas.

El fichero de datos utilizado contiene el estilo de juego del videojuego de consola Fifa 2017, así como estadísticas reales de los jugadores de fútbol. El conjunto de datos está formado por algo más de 17500 registros y 53 variables, de las cuales se han escogido una selección para este trabajo.

1. 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 12, 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 lo 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. Ésto indica que los valores extremos de la serie no tienen un gran impacto sobre la misma, al no mover la media de su posición central.
4. Las variables **Height** y **Weight** presentan valores NA que tendrán que ser tratados con posterioridad. Además el tipo de estas variables es literal, por lo que tendrán que ser tratadas para su conversión a numéricas enteras.

2. 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 existen valores duplicados en el campo *"ID"*, 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 duplicados.")
}
```

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

Vamos a mostrar algunos de los duplicados.

```
head(fifa.df.sub[duplicated(fifa.df.sub$Name),], 5)
```

```
##      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
##        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
```

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

3. Normalización de los datos cuantitativos

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

3.1 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.

3.2 Height

La variable que muestra la altura de los jugadores, *height*, debe de estar en cm **con un formato de 3 dígitos sin decimales**. Vamos a explorar como viene el dato en esta variable.

```
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 la siguiente selección de registros.

```
bad_height <- c(29, 76, 102, 172, 419 )
fifa.df[bad_height,c("Name", "Height")]
```

```
##                 Name Height
## 29      Philipp Lahm  1,7 m
## 76     Javi Martinez  1,9 m
## 102    Douglas costa  1,7 m
## 172    Asmir Begovic    2 m
## 419       Kurt Zouma  1,9 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.

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

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

Apareciendo los NA que trataremos más adelante, verificando que no existe ninguna observación mayor de 3 dígitos después de la transformación de la variable.

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.
```

Como paso de verificación, vamos a revisar las observaciones con formato problemático detectadas anteriormente.

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

```
##                 Name Height
## 29      Philipp Lahm    170
## 76     Javi Martinez    190
## 102    Douglas costa    170
## 172    Asmir Begovic    200
## 419       Kurt Zouma    190
```

Vemos que todas ellas se encuentran ya corregidas.

3.3 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. Algunos valores están expresados con tres decimales que deberemos eliminar, truncando la expresión para quedarnos con la parte entera.

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 y de los separadores decimales en la variable *Weight*.
2. Eliminación de los espacios en blanco en cada observación de la variable.
3. Conversión de la variable en Numérica.
4. Añadir el resultado como paso temporal 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()
```

El siguiente paso es utilizar la columna *Weight_Unit* para convertir los valores de gramos a kilos. Adicionalmente, aprovechamos para quedarnos con la parte entera del resultado.

```
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.

```
fifa.df.sub <- subset(fifa.df.sub, select = -c(Weight, Weight_Unit))
names(fifa.df.sub)[names(fifa.df.sub) == 'Weight2'] <- 'Weight'
```

Y revisamos el resultado.

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

```
##                   Name Weight
## 1    Cristiano Ronaldo     NA
## 2         Lionel Messi     72
## 3               Neymar     68
## 4          Luis Suárez     85
## 5         Manuel Neuer     NA
## 6               De Gea     82
## 7  robert Lewandowski      NA
## 8          Gareth Bale     74
## 9   Zlatan Ibrahimovic     95
## 10    Thibaut Courtois     91
```

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

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

```
## Min. Max.
##   48  110
```

4. Normalización de los datos caulitativos

En este apartado vamos a inspeccionar y normalizar los datos cualitativos.

4.1 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, c("Name", "Nationality")]
```

```
##                  Name Nationality
## 314       Lucas pérez       Spain
## 973     Juan fernando    Colombia
## 998 luís Hernández          Spain
## 794       Lamine sané     Senegal
```

Aplicamos la limpieza y normalización al campo *Name* y al *Nacionality*, verificando el resultado.

```
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
```

4.2 Preffered_Foot

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

- 1 -> Left
- 2 -> Right

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

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

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

```
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 ...
```

4.3 Work_Rate

Siguiendo la línea de este estudio, primeramente vamos a examinar 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'
```

El siguiente paso será 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 válidas para esta variable.

5 Posibles inconsistencias y variables tipo fecha

En el quinto apartado vamos a tratar las variables de tipo fecha, transformándolas a este tipo cuando no lo estén. Es importante notar que las fechas vienen configuradas con el formato mes/día/año.

5.1 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.

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

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

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.

5.2 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 al 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
```

En este caso concreto, y viendo que tampoco tenemos la fecha de contrato del club, se puede eliminar el registro o inferir la media de la serie.

5.3 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 literal a fecha (un casting de la variable).

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

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

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

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

```
##  Date[1:17588], format: "1985-02-05" "1987-06-24" "1992-02-05" "1987-01-24" "1986-03-27" ...
```

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 1985-02-05  32             31
## 2         Lionel Messi 1987-06-24  29             29
## 3               Neymar 1992-02-05  25             24
## 4          Luis Suárez 1987-01-24  30             29
## 5         Manuel Neuer 1986-03-27  31             30
## 6               De Gea 1990-11-07  26             26
## 7  Robert Lewandowski 1988-08-21  28             28
## 8          Gareth Bale 1989-07-16  27             27
## 9  Zlatan Ibrahimovic 1981-10-03  35             35
## 10   Thibaut Courtois 1992-05-11  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 1985-02-05  32             31
## 3              Neymar 1992-02-05  25             24
```

```
## 4        Luis Suárez 1987-01-24  30             29
## 5       Manuel Neuer 1986-03-27  31             30
## 12       Eden Hazard 1991-01-07  26             25
## 17      Sergio Ramos 1986-03-30  31             30
```

```
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:  5561"
```

Vamos a actualizar la variable *Age* con los valores calculados.

```
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 1985-02-05  31             31
## 2         Lionel Messi 1987-06-24  29             29
## 3               Neymar 1992-02-05  24             24
## 4          Luis Suárez 1987-01-24  29             29
## 5         Manuel Neuer 1986-03-27  30             30
## 6               De Gea 1990-11-07  26             26
## 7  Robert Lewandowski 1988-08-21  28             28
## 8          Gareth Bale 1989-07-16  27             27
## 9  Zlatan Ibrahimovic 1981-10-03  35             35
## 10   Thibaut Courtois 1992-05-11  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.

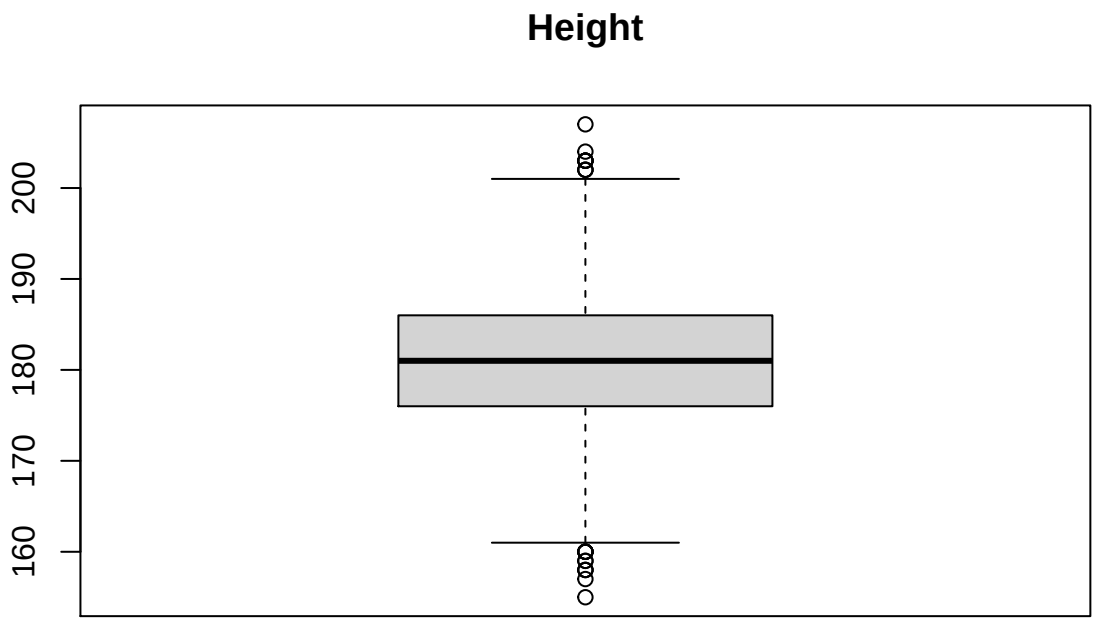
```
fifa.df.sub <- select(fifa.df.sub, -Calculated_Age)
```

6 Estudio de valores atípicos

En este capítulo del análisis exploratorio, vamos a revisar si existen valores atípicos para la variables *Height* y *Weight*.

6.1 Valores atípicos de *Height*

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



El bloxplot de la variable *Height*, nos marca valores fuera del 1.5 veces el rango intercuartílico. La pregunta que nos tendríamos que realizar, es si estos datos extremos los podemos considerar *outliers* o no. Examinemos la lista de valores.

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

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

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

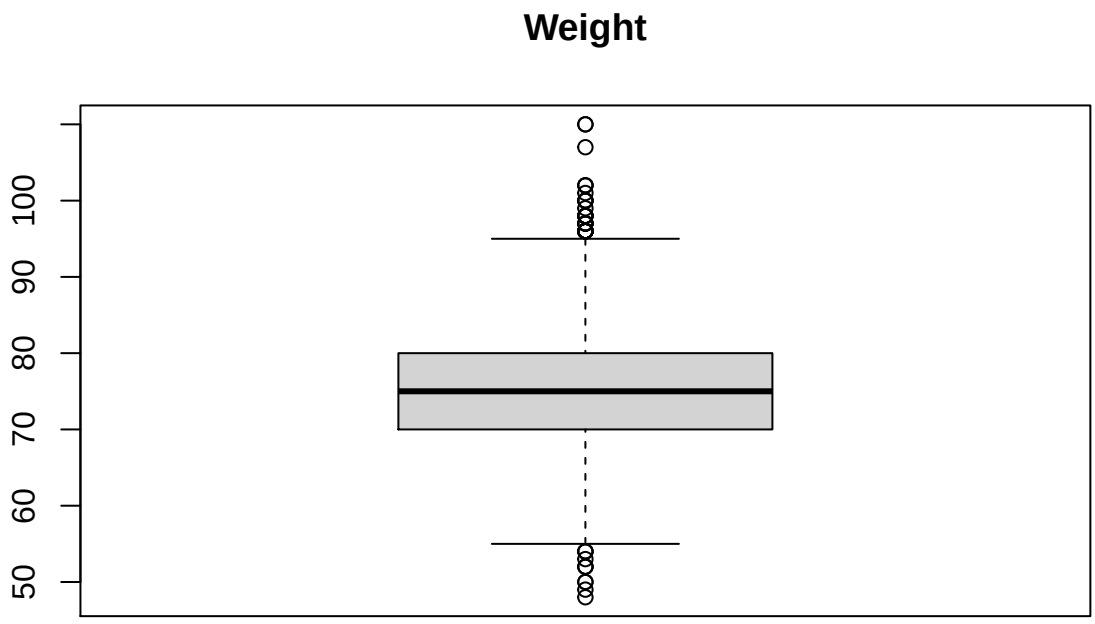
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
##   155.0   176.0   181.0   181.1   186.0   207.0       3
```

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.

6.2 Valores atípicos de *Weight*

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

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



Como pasaba en la variable anterior, media y mediana se encuentran centradas, por lo que podemos afirmar que los valores extremos no influencian en exceso desvirtuando la tendencia de la serie.

Vamos a ver los valores extremos en formato numérico.

```
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, los valores extremos son perfectamente válidos y acordes con pesos de la vida real, y 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.

7. 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 para ello una regresión lineal.

7.1. 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
```

Generaremos el modelo de regresión lineal utilizando la variable peso como variable conocida para inferir la altura.

```
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 en la vida real encontramos casos en los que esta relación directa no siempre se cumple, son casos particulares, y como regla general esta relación es perfectamente válida.

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 de nuestro modelo.

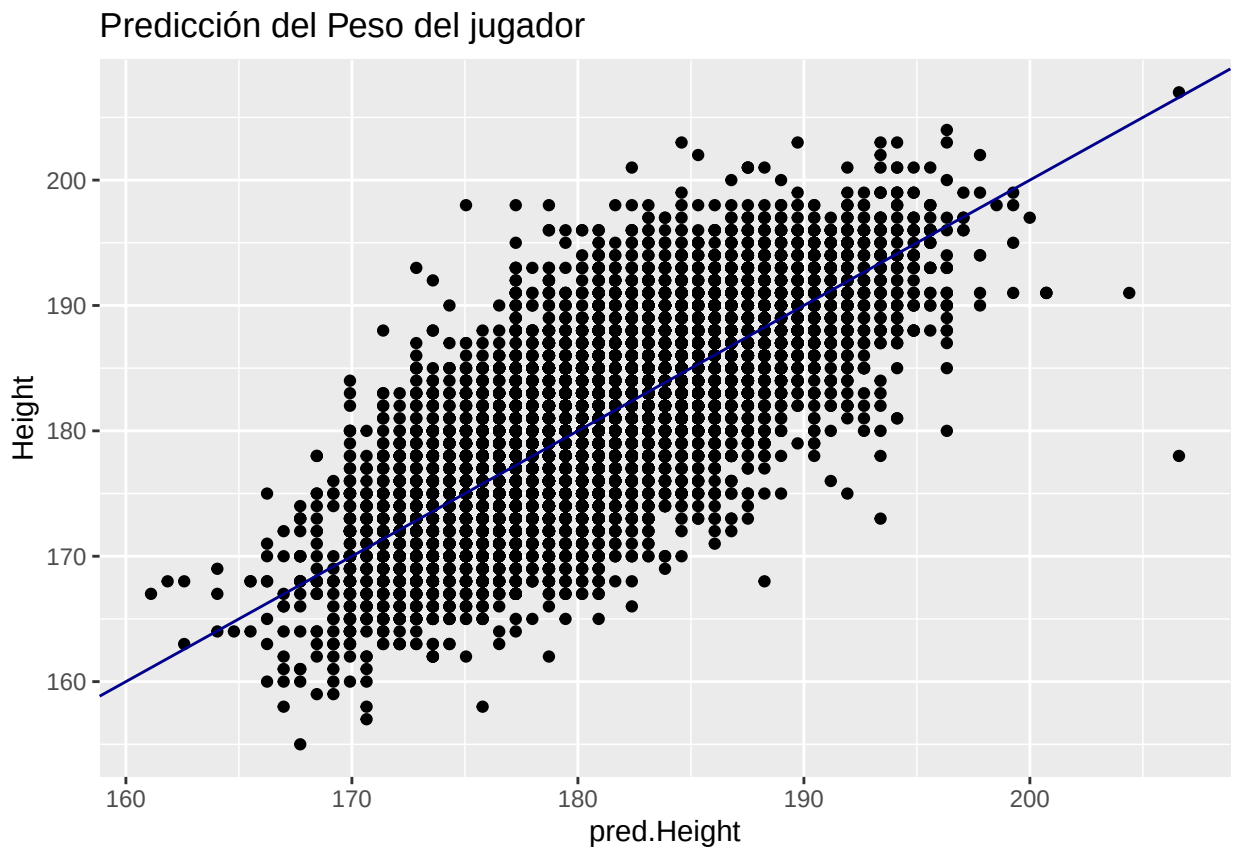
```
summary(lineal.model)$r.squared
```

```
## [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] {.ul}, 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 tras finalizar el proceso.

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

```
##                 Name Height
## 2       Lionel Messi    178
## 6             De Gea    186
## 9 Zlatan Ibrahimovic    195
```

7.2. 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")]
```

```
##                 Name Weight
## 1  Cristiano Ronaldo     78
## 5       Manuel Neuer     84
## 7 Robert Lewandowski     78
```

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
```

7.3 Contract_Expiry

La variable **Contract_Expiry** cuenta con un valor desconocido, que nos obliga a tratarlo para poder realizar el estudio. Tenemos varias opciones para ello: - 1. 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. - 2. Podríamos imputar el valor más repetido de la serie, pero podríamos incurrir en un sesgo.

- 3. 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 (sesgo).

```
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)
```

8. 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"
```

Montaremos una tabla con las medidas de tendencia central y de dispersión.

```
quantitative_study <- function(df, col_names) {
  media <- as.data.frame(lapply(df[, col_names], mean)) %>%
    round(digits = 2)  %>% mutate(estimator=c("media"))

  mediana <- as.data.frame(lapply(df[, col_names], median)) %>%
    round(digits = 2)  %>% mutate(estimator=c("mediana"))

  media_rec <- as.data.frame(lapply(df[, col_names], mean, trim=0.05)) %>%
    round(digits = 2)  %>% mutate(estimator=c("media recortada"))

  winsor_media <- as.data.frame(lapply(df[, col_names], winsor.mean)) %>%
    round(digits = 2)  %>% mutate(estimator=c("winsor_media"))

  desv_est <- as.data.frame(lapply(df[, col_names], sd)) %>%
    round(digits = 2)  %>% mutate(estimator=c("desviación estándar"))

  RIC <- as.data.frame(lapply(df[, col_names], IQR)) %>%
    round(digits = 2)  %>% mutate(estimator=c("RIC"))

  DAM <- as.data.frame(lapply(df[, col_names], mad)) %>%
    round(digits = 2)  %>%  mutate(estimator=c("DAM"))

  df_out <- merge(media, mediana, all=TRUE)
  df_out <- merge(df_out, media_rec, all=TRUE)
  df_out <- merge(df_out, winsor_media, all=TRUE)
  df_out <- merge(df_out, desv_est, all=TRUE)
  df_out <- merge(df_out, RIC, all=TRUE)
  df_out <- merge(df_out, DAM, all=TRUE)
  return(df_out)
}

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

```
##   Contract_Expiry Rating Height   Age Weight           estimator
```

```
## 1            1.48   7.41   7.41  4.45   7.41                 DAM
## 2            1.70   7.08   6.67  4.68   6.90 desviación estándar
## 3            3.00   9.00  10.00  7.00  10.00                 RIC
## 4         2018.60  66.16 181.08 24.89  75.21        winsor_media
## 5         2018.78  66.16 181.11 24.99  75.16     media recortada
## 6         2018.90  66.17 181.11 25.14  75.25              media
## 7         2019.00  66.00 181.00 25.00  75.00            mediana
```

Si tomamos como "baseline" **la mediana**, al ser una medida de tendencia central robusta, podemos compararla con el resto de medidas de tendencia. A priori, podríamos pensar que como **la media** es una medida de tendencia central que se ve influida por los extremos, éstos están escorando ligeramente la tendencia alejándola del centro de la serie. No obstante, si comparamos con **la media recortada** (también llamada media truncada), en la que se han eliminado **un 5% de los valores extremos**, vemos que efectivamente los la serie tiene un ligero desvío hacia la derecha (hacia un mayor valor). Esto se puede acabar de confirmar con **la media winsorizada**, medida robusta que nos muestra unos valores similares a los anteriores.

En cuanto a **la desviación**, se obtiene una menor dispersión de los valores utilizando como base para su cálculo la media que no la mediana. Al haber visto que los extremos no tienen una gran influencia en la serie, podríamos tomar como buena la medida de **la desviación estándar**, aunque la **DAM** sea una medida de dispersión más robusta.

9. 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 -0.1205747 -0.05316544
## Rating               0.04743154 1.00000000  0.04713470  0.4560377  0.13945142
## Height              -0.08068535 0.04713470  1.00000000  0.0765333  0.75820508
## Age                 -0.12057466 0.45603766  0.07653330  1.0000000  0.22219008
## Weight              -0.05316544 0.13945142  0.75820508  0.2221901  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     1.3773 1.1541 0.7374 0.47695
## Proportion of Variance 0.4742 0.3330 0.1359 0.05687
## Cumulative Proportion  0.4742 0.8072 0.9431 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.

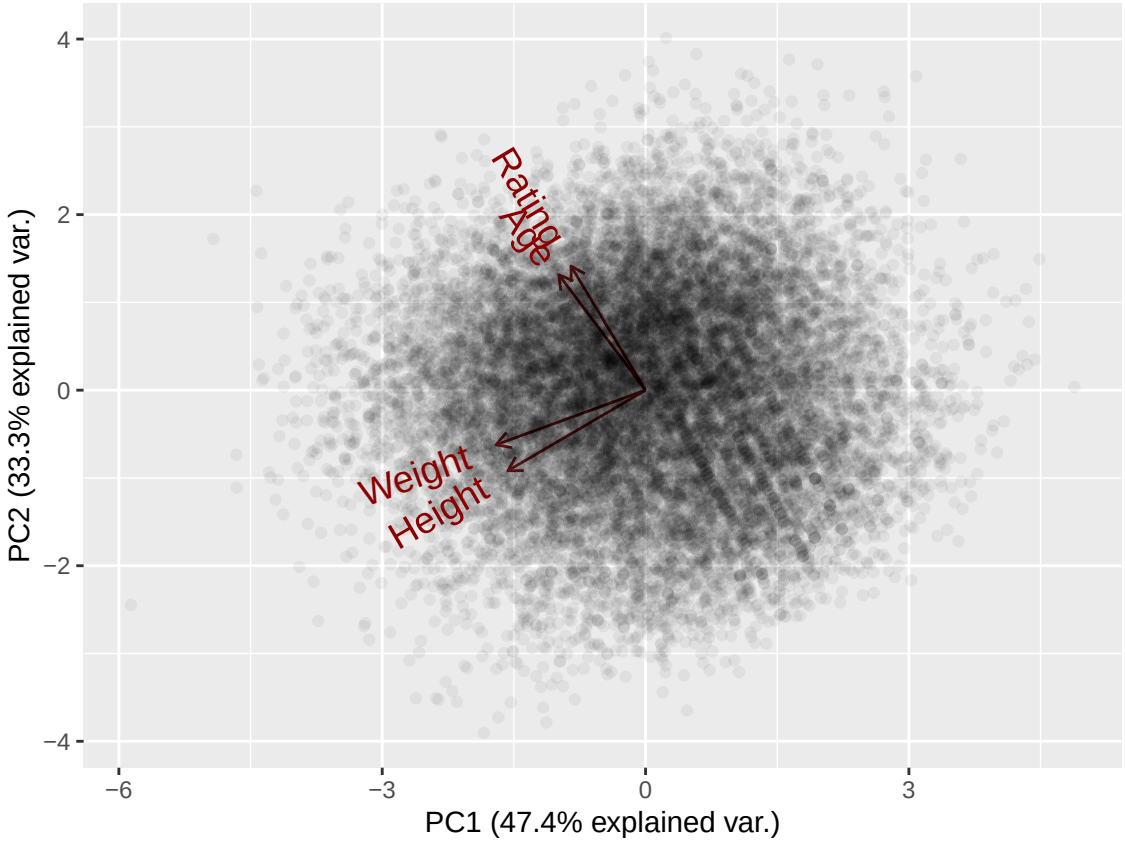
Veamos los vectores propios.

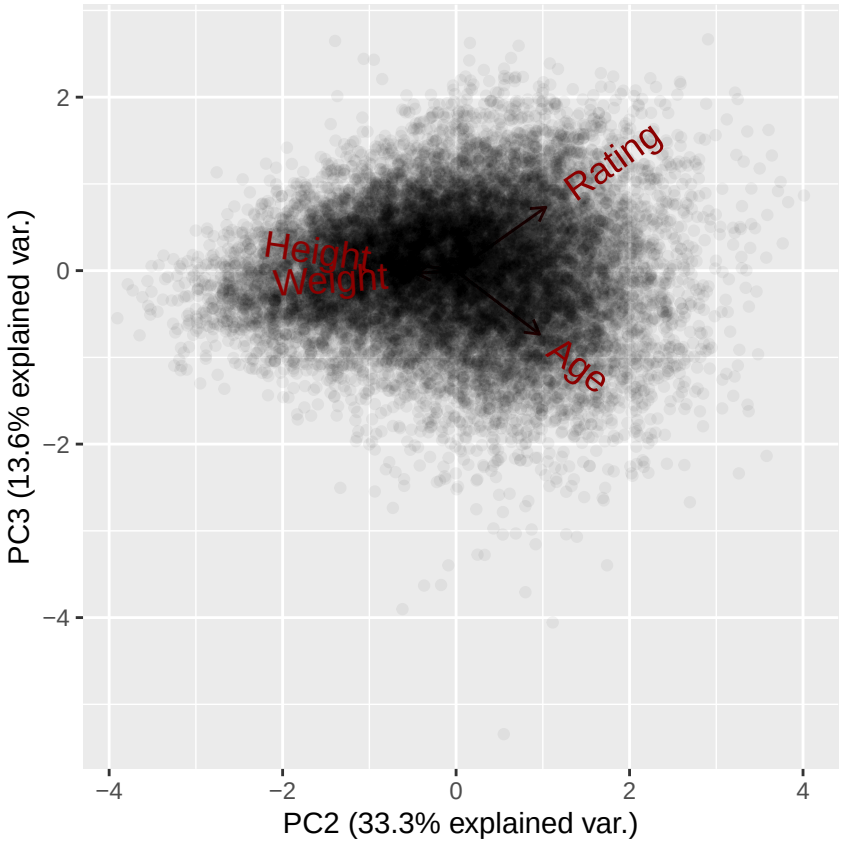
```
fifa.pca$rotation
```

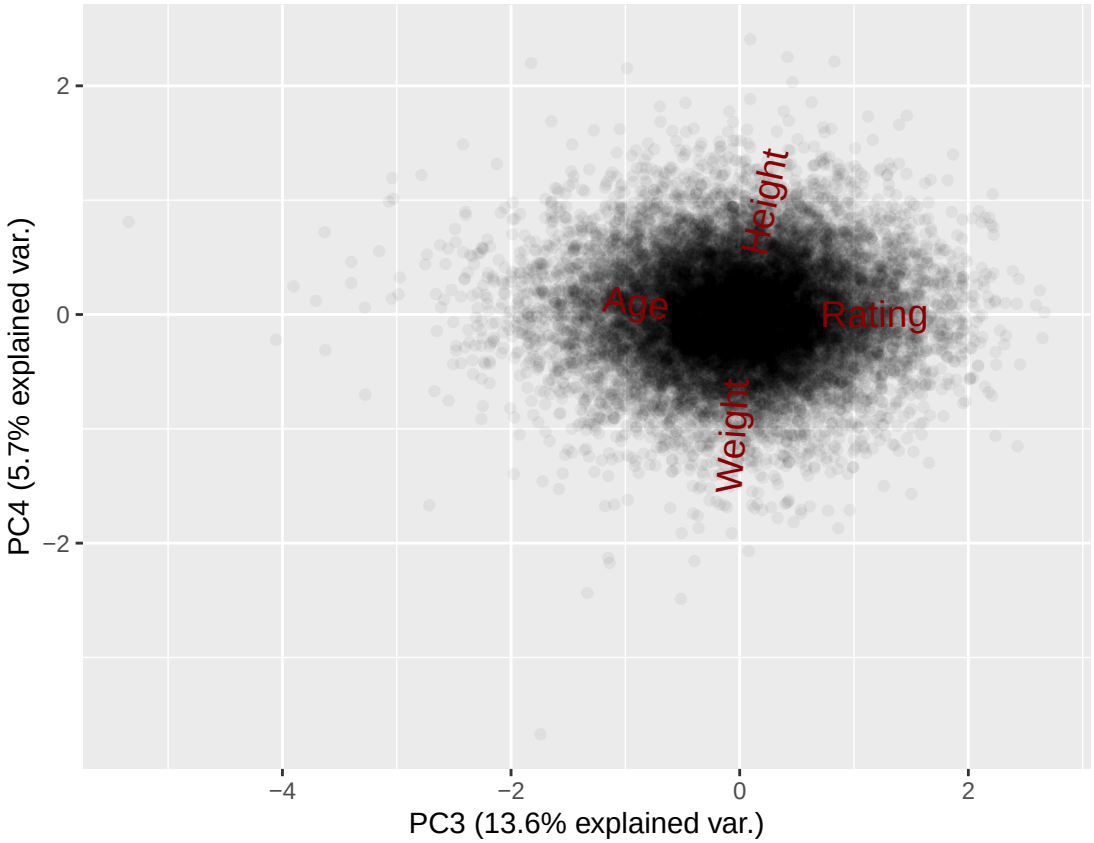
```
##                PC1        PC2         PC3          PC4
## Rating -0.3197566  0.6351195  0.70306211  0.009088975
## Height -0.5904533 -0.4122123  0.09494980  0.687335786
## Weight -0.6410093 -0.2795464 -0.02977018 -0.714195077
## Age    -0.3717899  0.5903880 -0.70413205  0.131955677
```

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))
  }
```




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



Del examen del primer gráfico, que representa a las dos primeras componentes, podemos observar lo siguiente:

- Con la primera componente se explicaría el 47,4% de la varianza, siendo la que más contribuye al modelo.
- La segunda componente contribuye con el 33,3 %. Así entre las dos componentes podemos explicar el 80,7% de la varianza.
- El resto de componentes explican el 19,3% de la varianza acumulada. Si el objetivo es reducir la dimensionalidad del modelo, podríamos plantear el suprimir estos dos últimos componentes, teniendo un modelo con un grado aceptable de similitud al original. En todo caso, más adelante lo podremos ver visualmente.
- 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. En efecto, si comprobamos en la matriz de correlaciones, vemos que existe algo menos del 47 % de correlación entre estas dos últimas variables.
- La longitud de las flechas indican la importancia que tienen las variables originales en las componentes.En todos los gráficos vemos que la longitud es similar, no destacando ninguna variable sobre el resto.

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, aunque para ello se tendría que valorar su valor respecto al contexto del análisis, cosa que veremos a continuación.

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
## -5.8644612 -2.4460813 -0.2794348 -0.8215221
```

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, pero perfectamente posibles. Así que podemos considerar que aunque este punto corresponde a un extremo de la serie, no podemos considerarlo como outlier.

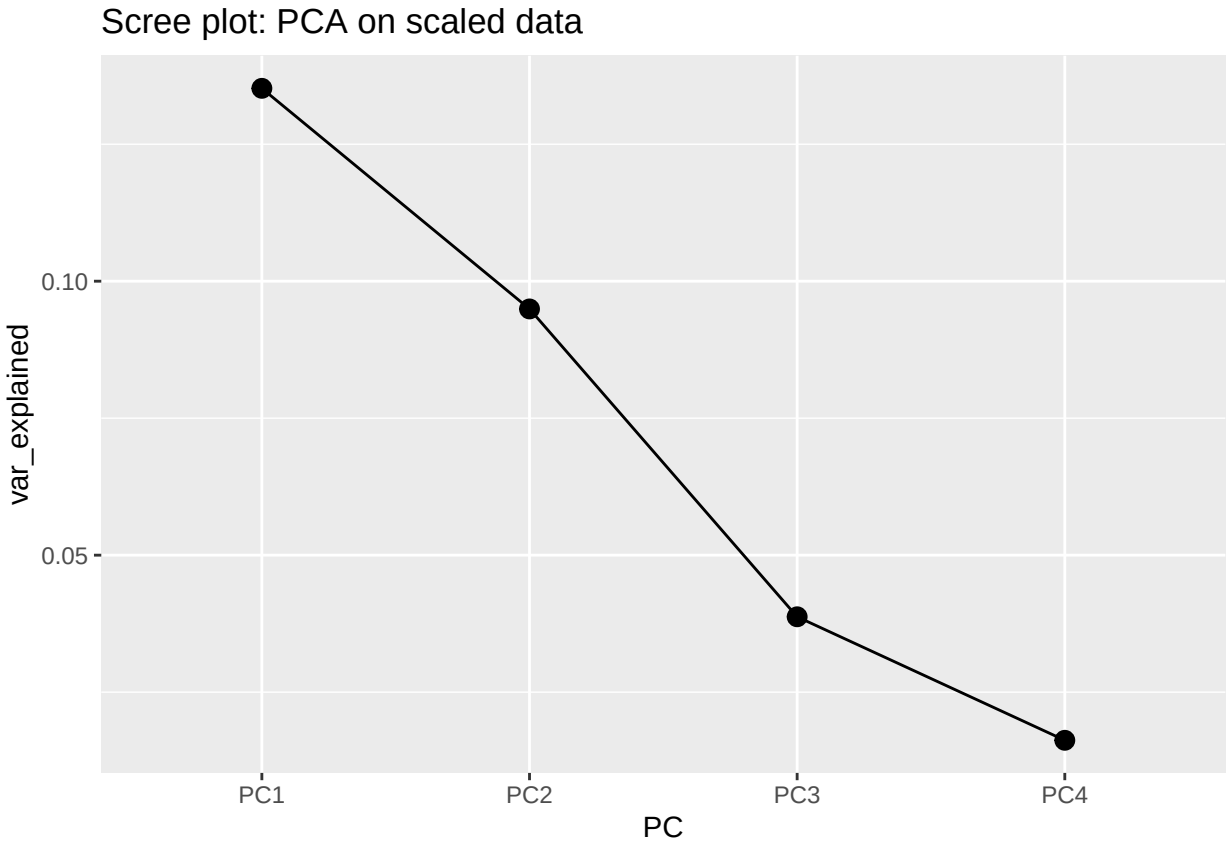
Como trabajo adicional, vamos a comprobar visualmente como se reparte la varianza entre los principales componentes. Calcularemos la varianza a partir de la desviación estándar calculada en los resultados de la PCA.

```
var_fifa <- data.frame(PC = paste0("PC", 1:4), var_explained = (fifa.pca$sdev)^2 / sum(fifa.pca$sdev)^2)
print(var_fifa)
```

```
##    PC var_explained
## 1 PC1    0.13519704
## 2 PC2    0.09493439
## 3 PC3    0.03875421
## 4 PC4    0.01621378
```

Una vez visto los valores numéricos, vamos a visualizarlos en un Scree plot.

```
var_fifa %>%
  ggplot(aes(x=PC, y= var_explained, group=1)) +
  geom_point(size=3) +
  geom_line() +
  labs(title = "Scree plot: PCA on scaled data")
```



Visualmente podemo ver como la pendiente de la recta se hace más acusada a partir de la segunda componente, quedando PC3 y PC4 como mínimos de la curva.

Agradecimientos a: https://datavizpyr.com/how-to-make-scree-plot-in-r-with-ggplot2/ por esta última parte.

Archivo final

En este trabajo hemos trabajado con un subset de datos del archivo original. En este punto, podríamos incluir a nuestro subset el resto de variables no tratadas. Al quedar fuera del ámbito de estudio y para mayor claridad en caso de examen, se considera que es mejor dejarlo reducido.

Este trabajo está compuesto de los siguientes ficheros:

- Fifa2017.Rmd con éste trabajo en Markdown y R.
- Fifa2017.html, traducción del trabajo en formato html.
- Fifa2017.pdf, fichero de texto pdf.

Conclusión

A lo largo de esta actividad se ha puesto en práctica las técnicas de verificación y normalización de variables. Además, se ha buscado inconsistencia en los datos y se ha enriquecido la información tratando los valores perdido.

Se ha dedicado un capítulo a la observación de los valores extremos, detectándolos y eligiendo la mejor opción según el contexto de los datos.

Se han calculado algunas medidas de tendencia central y dispersión, enfrentándolas entre ellas para obtener un avance del análisis descriptivo.

Por último, se ha realizado un análisis de componentes principales con algunas de las variables cuantitativas, con el resultado de poder escoger la eliminación de los dos últimos componentes para quedarnos con un modelo equivalente con una varianza suficientemente alta.