R Notebook

Per prima cosa necessitiamo di installare i pacchetti necessari per il nostro progetto.

```
library(FactoMineR)
library(ggplot2)
library(cluster)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## v readr     2.1.5

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(rpart)
library(rpart.plot)
library(randomForest)
```

```
## randomForest 4.7-1.1
## Type rfNews() to see new features/changes/bug fixes.
##
```

```
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##     combine
##
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
library(caret)
```

```
## Loading required package: lattice
##
## Attaching package: 'caret'
##
## The following object is masked from 'package:purrr':
##
##     lift
```

Cominciamo leggendo il file per l'analisi.

```
df <- read.csv("C:/Users/filip/Desktop/attaccanti italiani/dataset scraped/Statistiche_Giocatori_all_Ti
```

```
df$Gol.per.Minuti <- as.numeric(gsub("'", "", df$Gol.per.Minuti))
df$Gol.per.Minuti.in.Nazionale <- as.numeric(gsub("'", "", df$Gol.per.Minuti.in.Nazionale))
df$Minuti.Giocati.in.Nazionale <- as.numeric(gsub("'", "", df$Minuti.Giocati.in.Nazionale))
df$Minuti.Giocati <- as.numeric(gsub("'", "", df$Minuti.Giocati))
```

```
str(df)
```

```
## 'data.frame':    187 obs. of  21 variables:
##  $ Player.Name                   : chr  "Giuseppe Meazza" "Silvio Piola" "Roberto Baggio" "Alessandr
##  $ Presenze.Totali               : int  492 612 604 777 276 624 643 476 534 566 ...
##  $ Gol                           : int  307 320 277 316 144 288 266 236 181 162 ...
##  $ Assist                        : int  3 7 152 184 10 46 10 34 9 13 ...
##  $ Sostituito.In                 : int  70 4 62 168 70 155 29 112 26 2 ...
##  $ Sostituito.Out                : int  92 1 144 257 92 189 58 102 49 13 ...
##  $ Cartellino.Giallo             : int  0 0 31 57 0 38 7 49 2 0 ...
##  $ Cartellino.Rosso              : int  6 6 2 0 2 1 1 6 5 0 ...
##  $ Gol.su.Rigore                 : int  16 26 93 70 9 15 41 25 5 17 ...
##  $ Gol.per.Minuti                : num  143 171 172 170 172 149 207 141 251 313 ...
##  $ Minuti.Giocati                : num  44 54.6 47.6 53.8 24.7 ...
##  $ Presenze.in.Nazionale         : int  53 34 56 91 47 57 61 49 64 70 ...
##  $ Gol.in.Nazionale              : int  33 30 27 27 25 25 25 23 23 22 ...
##  $ Assist.in.Nazionale           : int  0 0 14 11 0 4 2 4 0 3 ...
##  $ Sostituito.In..Nazionale.     : int  7 6 11 30 7 15 17 3 7 2 ...
##  $ Sostituito.Out..Nazionale.    : int  8 4 16 43 8 25 18 26 9 10 ...
##  $ Cartellino.Giallo.in.Nazionale: int  0 0 3 5 0 0 0 4 2 0 ...
##  $ Cartellino.Rosso.in.Nazionale : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Gol.su.Rigore.in.Nazionale    : int  3 1 7 6 1 2 2 0 1 1 ...
##  $ Gol.per.Minuti.in.Nazionale   : num  147 103 152 191 170 141 168 151 211 263 ...
##  $ Minuti.Giocati.in.Nazionale   : num  4.86 3.09 4.1 5.15 4.26 ...
```

#Esplorazione Dati

Cominciamo inizialmente eseguendo una piccola Esplorazione dei Dati

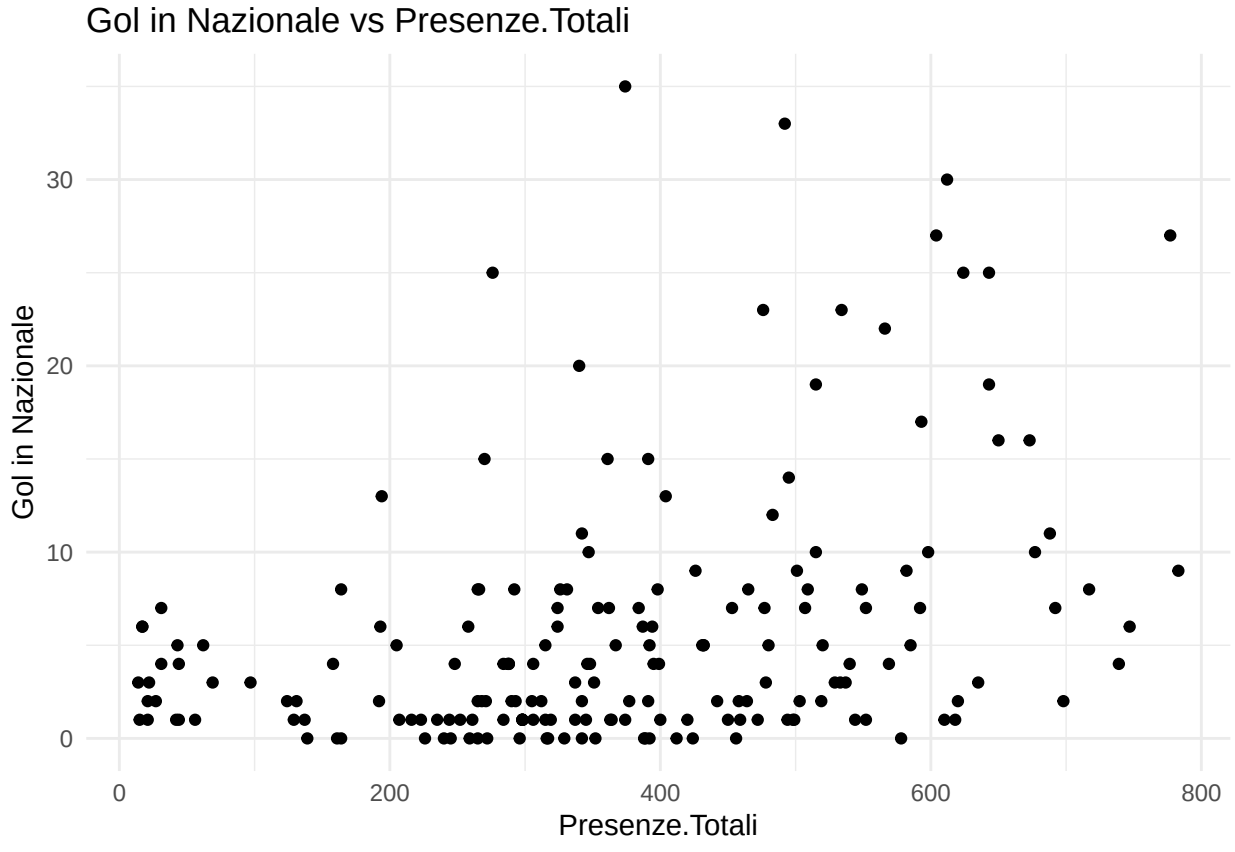
```
print(summary(df))
```

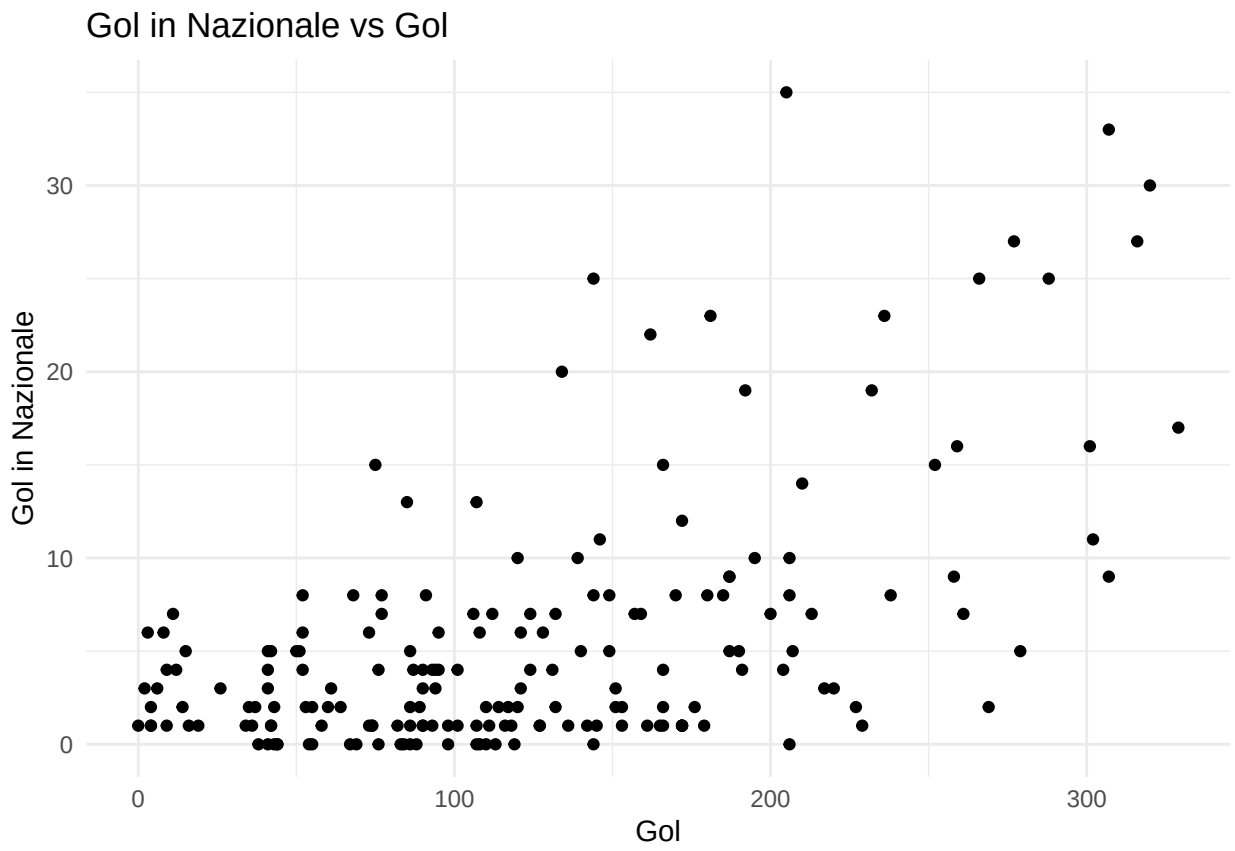
```
##  Player.Name        Presenze.Totali      Gol             Assist
##  Length:187         Min.   : 14.0   Min.   :  0.0   Min.   :  0.00
##  Class :character   1st Qu.:265.5   1st Qu.: 67.5   1st Qu.:  2.00
##  Mode  :character   Median :354.0   Median :112.0   Median : 10.00
##                     Mean   :367.9   Mean   :123.8   Mean   : 24.19
##                     3rd Qu.:496.5   3rd Qu.:172.0   3rd Qu.: 33.50
##                     Max.   :783.0   Max.   :329.0   Max.   :206.00
##  Sostituito.In    Sostituito.Out  Cartellino.Giallo Cartellino.Rosso
##  Min.   :  1.00   Min.   :  1.0   Min.   :  0.00    Min.   : 0.000
##  1st Qu.: 29.00   1st Qu.: 53.5   1st Qu.:  0.00    1st Qu.: 0.000
##  Median : 70.00   Median : 92.0   Median :  5.00    Median : 1.000
##  Mean   : 69.78   Mean   : 94.9   Mean   : 22.64    Mean   : 1.989
##  3rd Qu.: 97.50   3rd Qu.:123.0   3rd Qu.: 39.50    3rd Qu.: 3.000
##  Max.   :214.00   Max.   :268.0   Max.   :155.00    Max.   :11.000
##  Gol.su.Rigore   Gol.per.Minuti    Minuti.Giocati   Presenze.in.Nazionale
##  Min.   : 0.00   Min.   :  1.008   Min.   :  1.26   Min.   : 0.0
##  1st Qu.: 1.00   1st Qu.:181.000   1st Qu.: 19.13   1st Qu.: 4.0
##  Median : 6.00   Median :219.000   Median : 29.42   Median :12.0
##  Mean   :12.65   Mean   :264.322   Mean   : 35.40   Mean   :17.4
##  3rd Qu.:17.00   3rd Qu.:311.000   3rd Qu.: 39.01   3rd Qu.:24.5
##  Max.   :93.00   Max.   :989.000   Max.   :961.00   Max.   :91.0
##  Gol.in.Nazionale Assist.in.Nazionale Sostituito.In..Nazionale.
##  Min.   : 0.00    Min.   : 0.000      Min.   : 0.000
##  1st Qu.: 1.00    1st Qu.: 0.000      1st Qu.: 2.000
##  Median : 3.00    Median : 0.000      Median : 7.000
##  Mean   : 5.62    Mean   : 1.428      Mean   : 6.262
##  3rd Qu.: 7.00    3rd Qu.: 2.000      3rd Qu.: 7.000
##  Max.   :35.00    Max.   :25.000      Max.   :30.000
##  Sostituito.Out..Nazionale. Cartellino.Giallo.in.Nazionale
##  Min.   : 0.000             Min.   :0.0000
##  1st Qu.: 3.000             1st Qu.:0.0000
##  Median : 6.000             Median :0.0000
##  Mean   : 7.176             Mean   :0.6845
##  3rd Qu.: 8.000             3rd Qu.:1.0000
##  Max.   :43.000             Max.   :9.0000
##  Cartellino.Rosso.in.Nazionale Gol.su.Rigore.in.Nazionale
##  Min.   :0.00000               Min.   :0.0000
##  1st Qu.:0.00000               1st Qu.:0.0000
##  Median :0.00000               Median :0.0000
##  Mean   :0.04813               Mean   :0.3155
##  3rd Qu.:0.00000               3rd Qu.:0.0000
##  Max.   :2.00000               Max.   :7.0000
##  Gol.per.Minuti.in.Nazionale Minuti.Giocati.in.Nazionale
##  Min.   :  0.0               Min.   :  -1.000
##  1st Qu.:102.5               1st Qu.:   2.044
##  Median :192.0               Median :  89.000
##  Mean   :226.1               Mean   : 245.134
##  3rd Qu.:298.0               3rd Qu.: 448.000
```

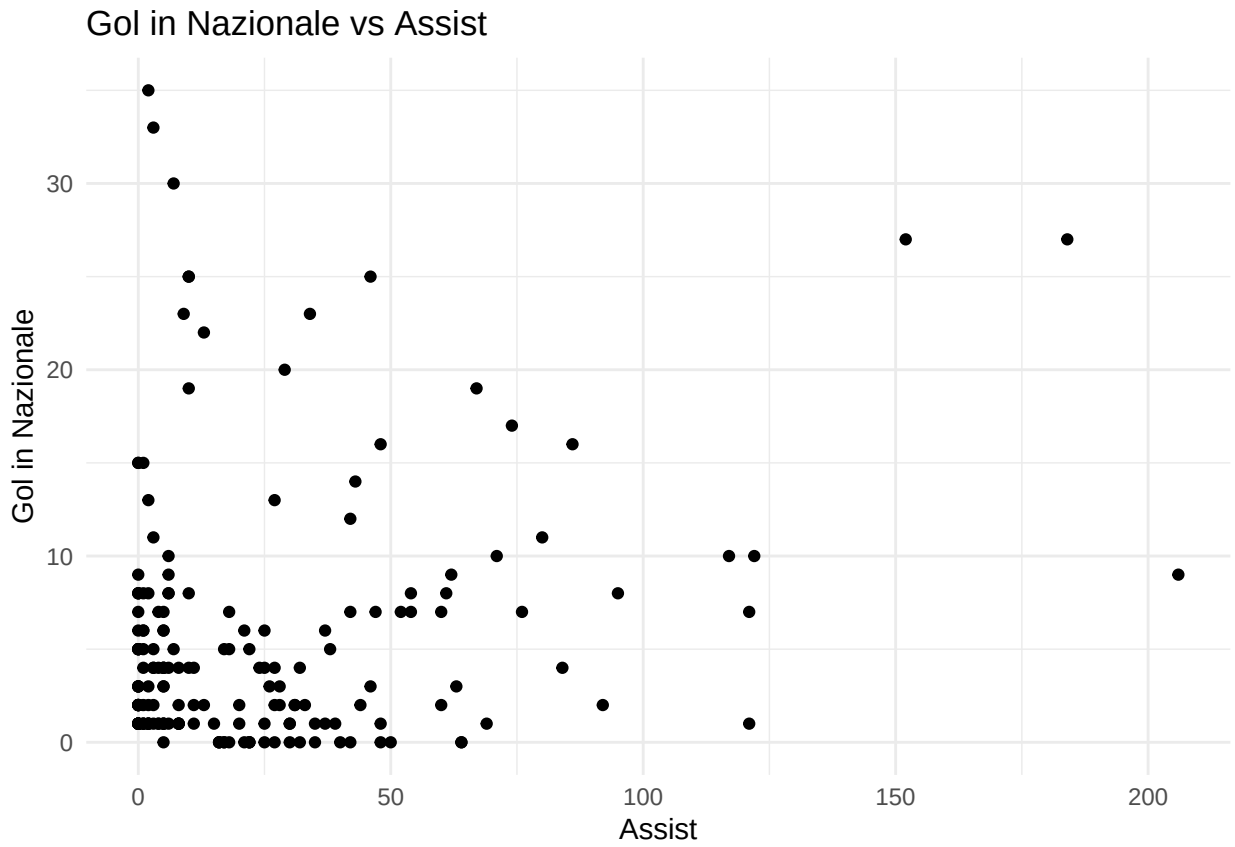
```
##  Max.   :934.0                Max.   :2840.000
```

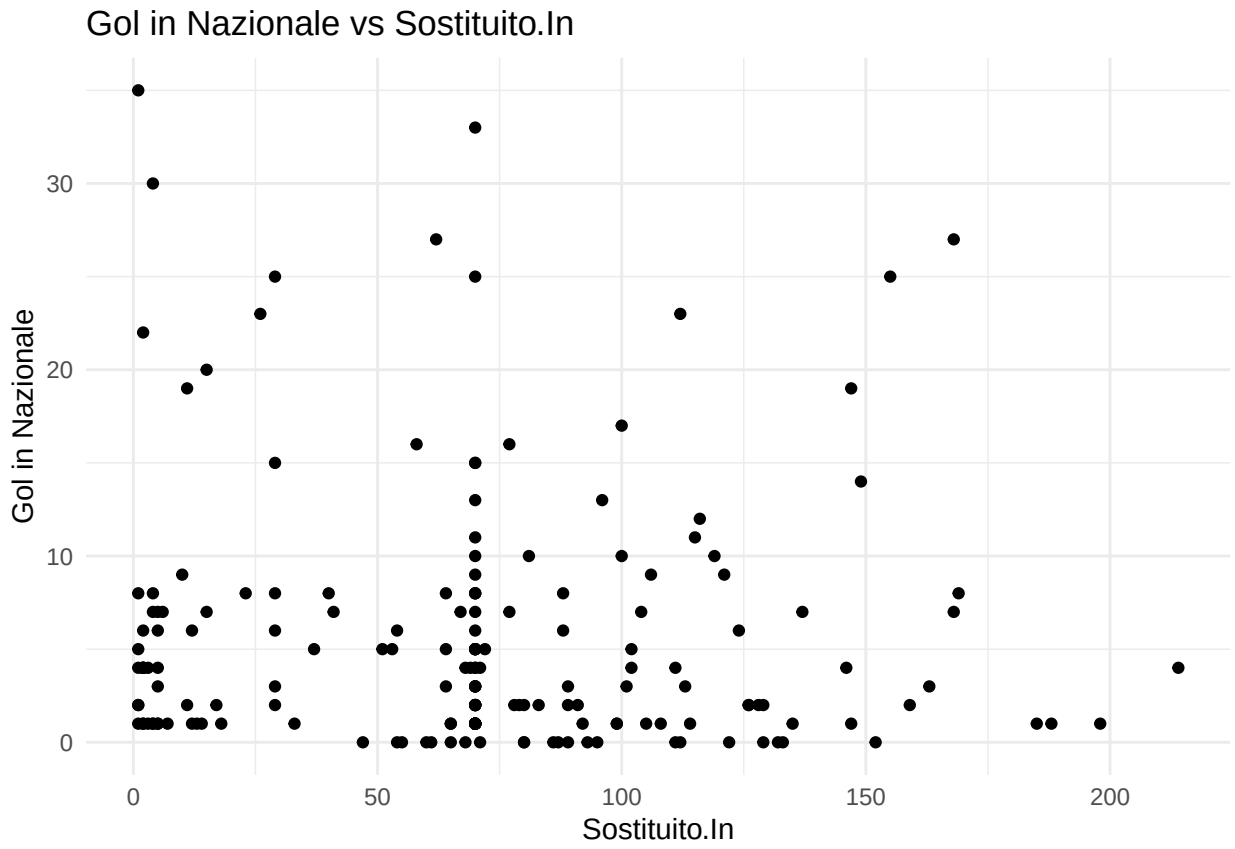
```r
numeric_vars <- names(df)[sapply(df, is.numeric)]
numeric_vars <- setdiff(numeric_vars, "Gol.in.Nazionale")

# Crea un grafico scatter per ogni variabile numerica contro "Gol.in.Nazionale"
for (var in numeric_vars) {
  p <- ggplot(df, aes_string(x = var, y = "Gol.in.Nazionale")) +
    geom_point() +
    labs(title = paste("Gol in Nazionale vs", var),
         x = var,
         y = "Gol in Nazionale") +
    theme_minimal()
  print(p)
}
```

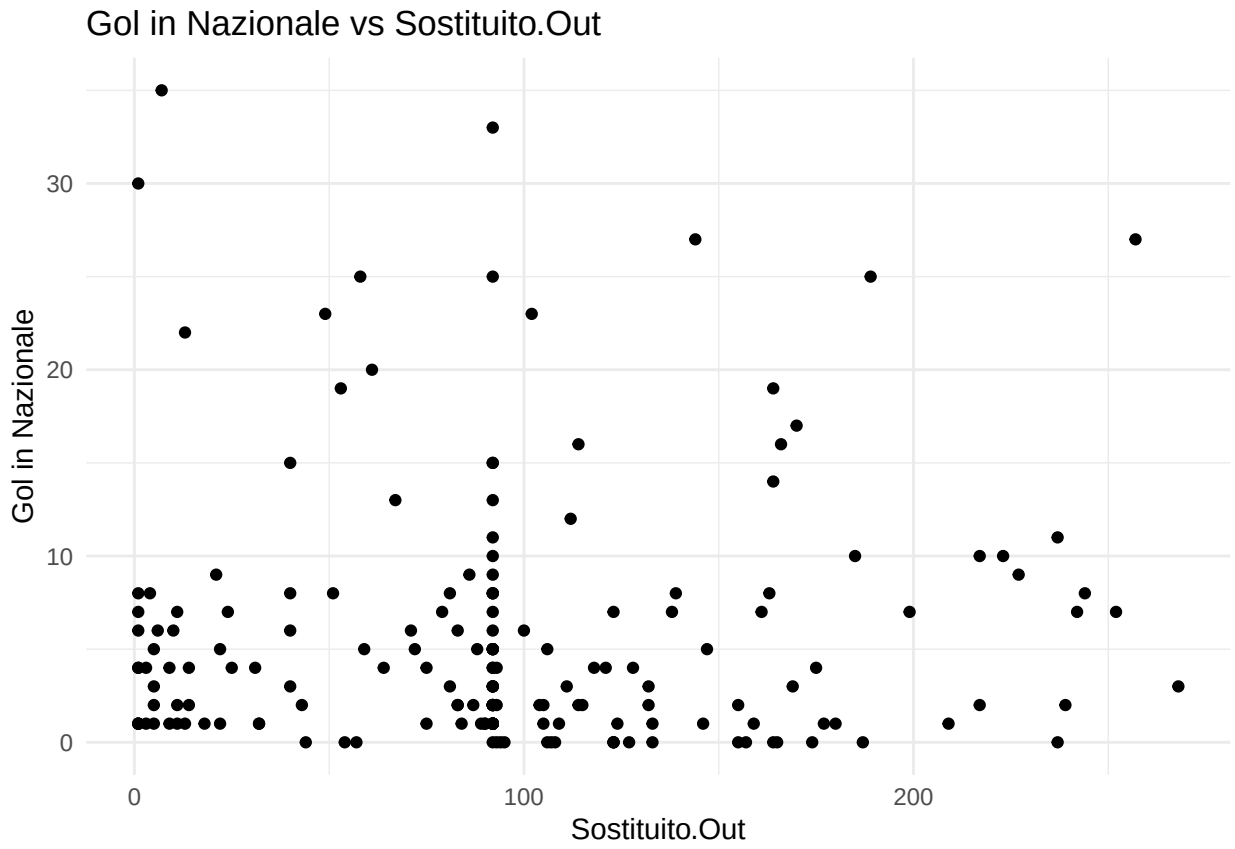
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

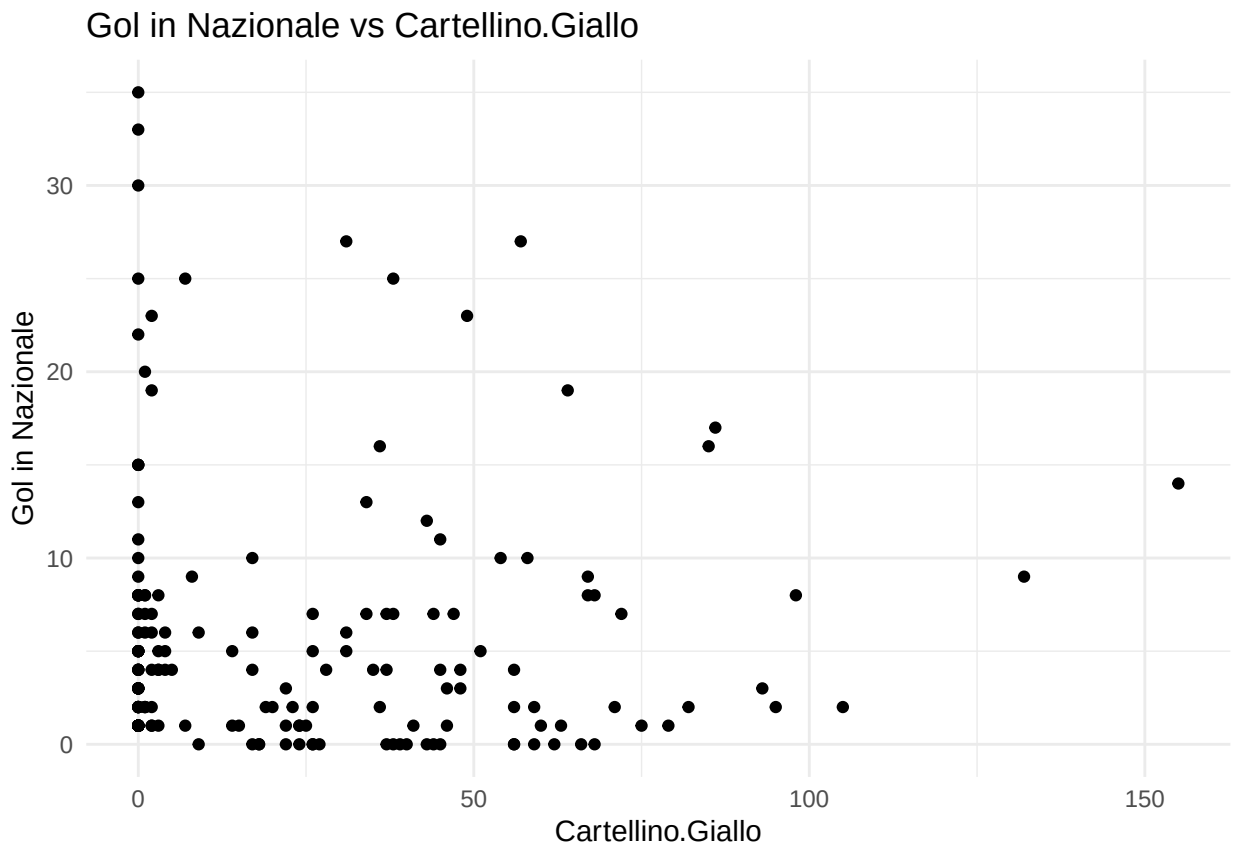


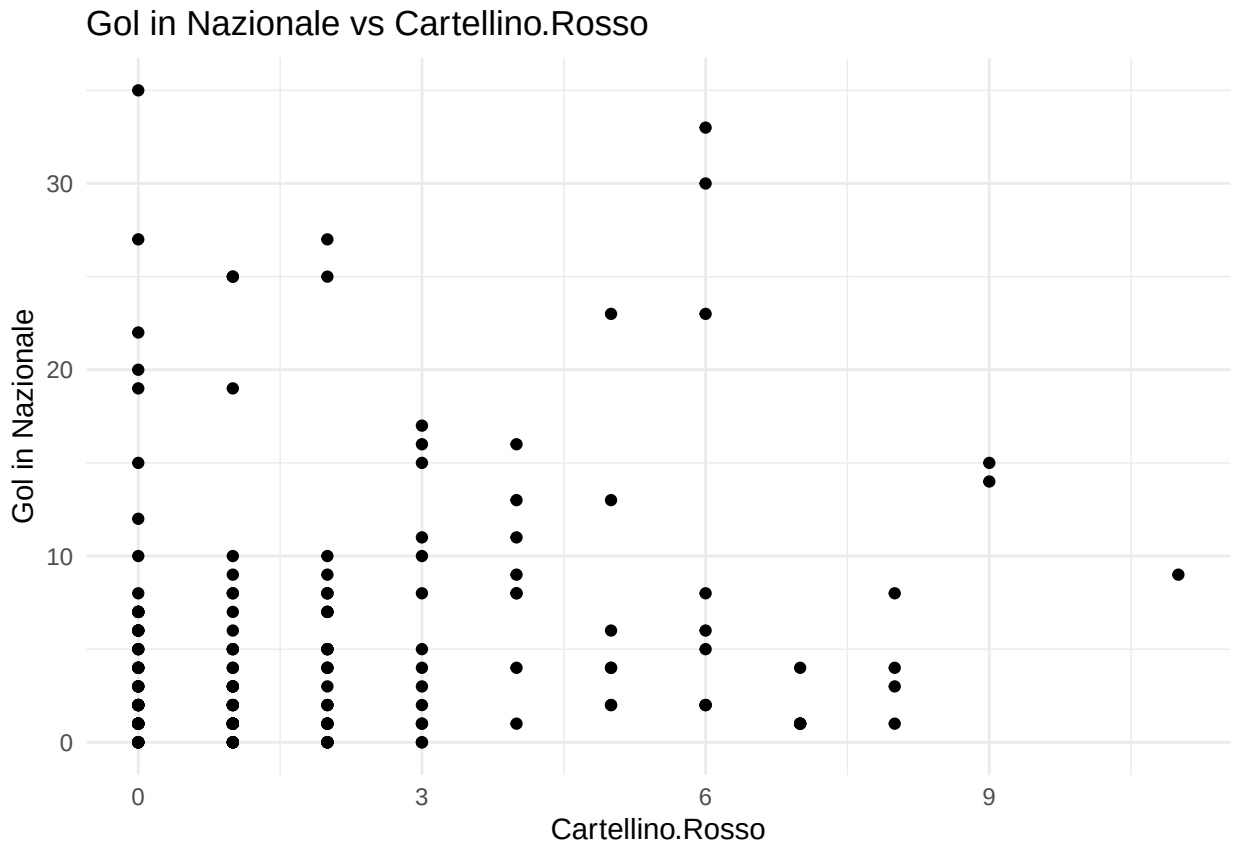

Gol in Nazionale vs Gol
Gol in Nazionale
Gol
0
10
20
30
0
100
200
300

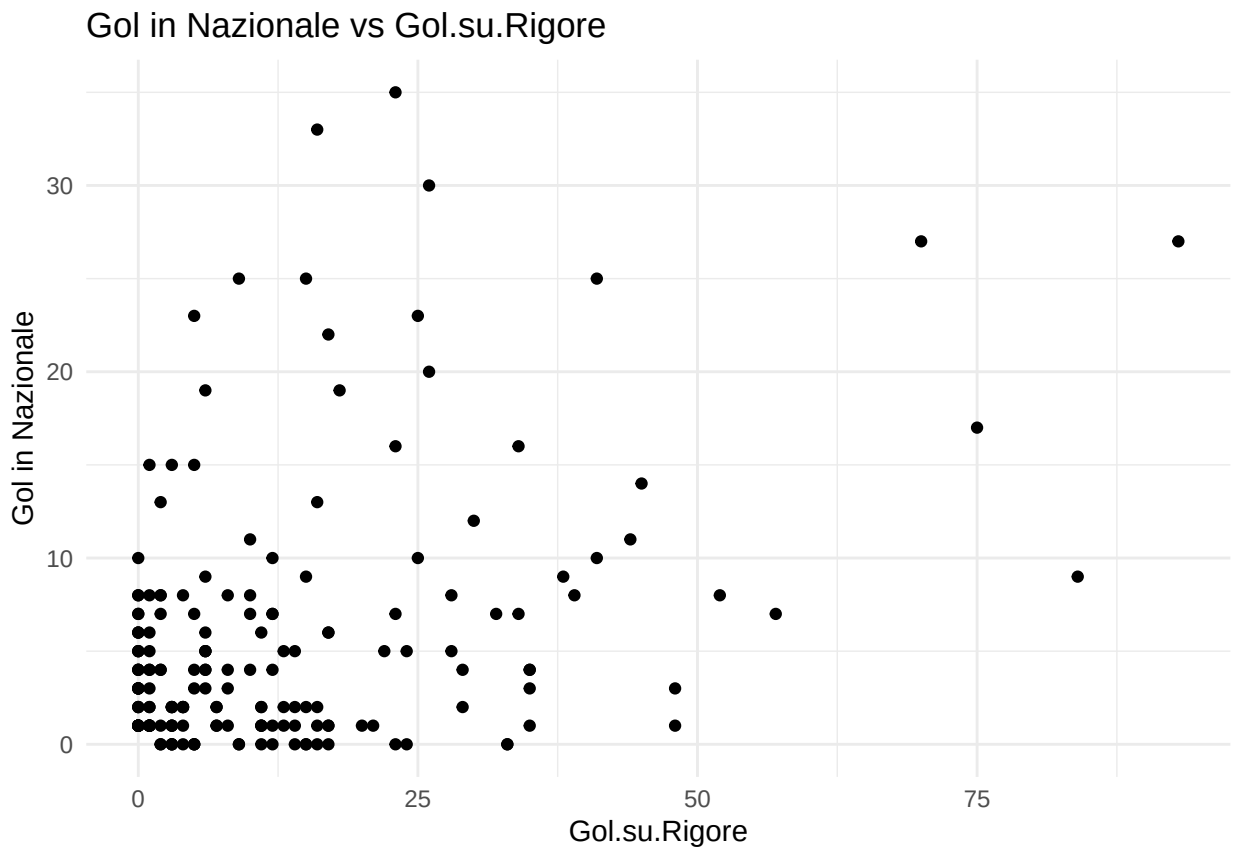

Gol in Nazionale vs Assist
Gol in Nazionale
30
20
10
0
0
50
100
150
200
Assist

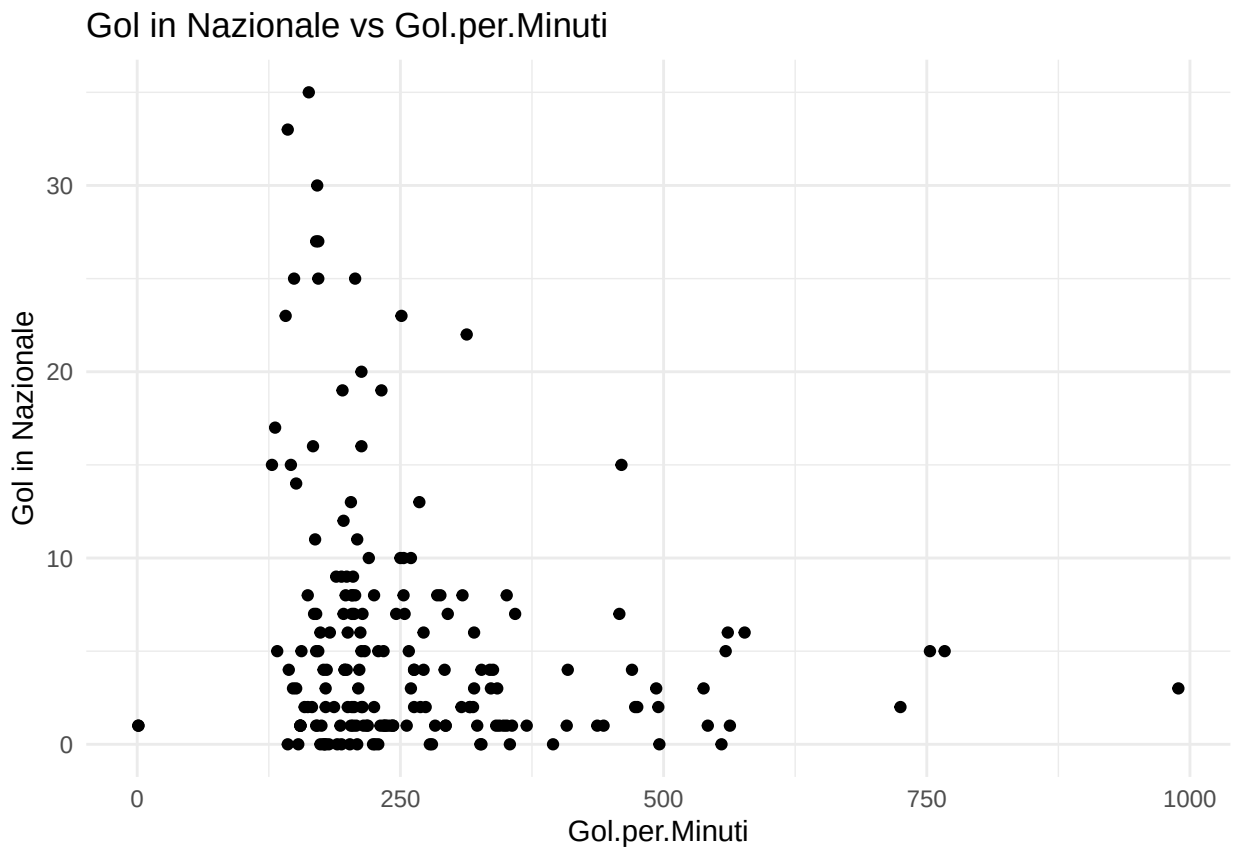

Gol in Nazionale vs Sostituito.In
Gol in Nazionale
30
20
10
0
0
50
100
150
200
Sostituito.In

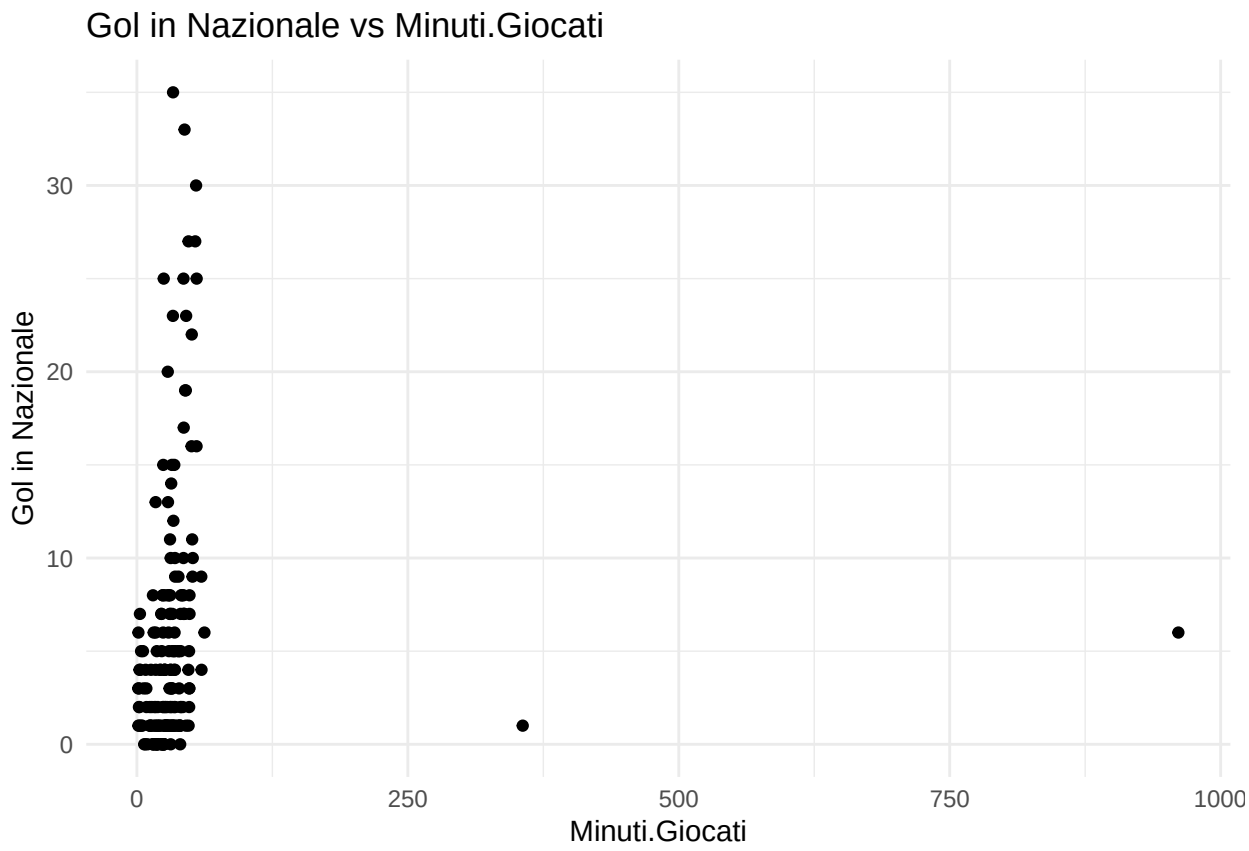

Gol in Nazionale vs Sostituito.Out
Gol in Nazionale
30
20
10
0
0
100
200
Sostituito.Out

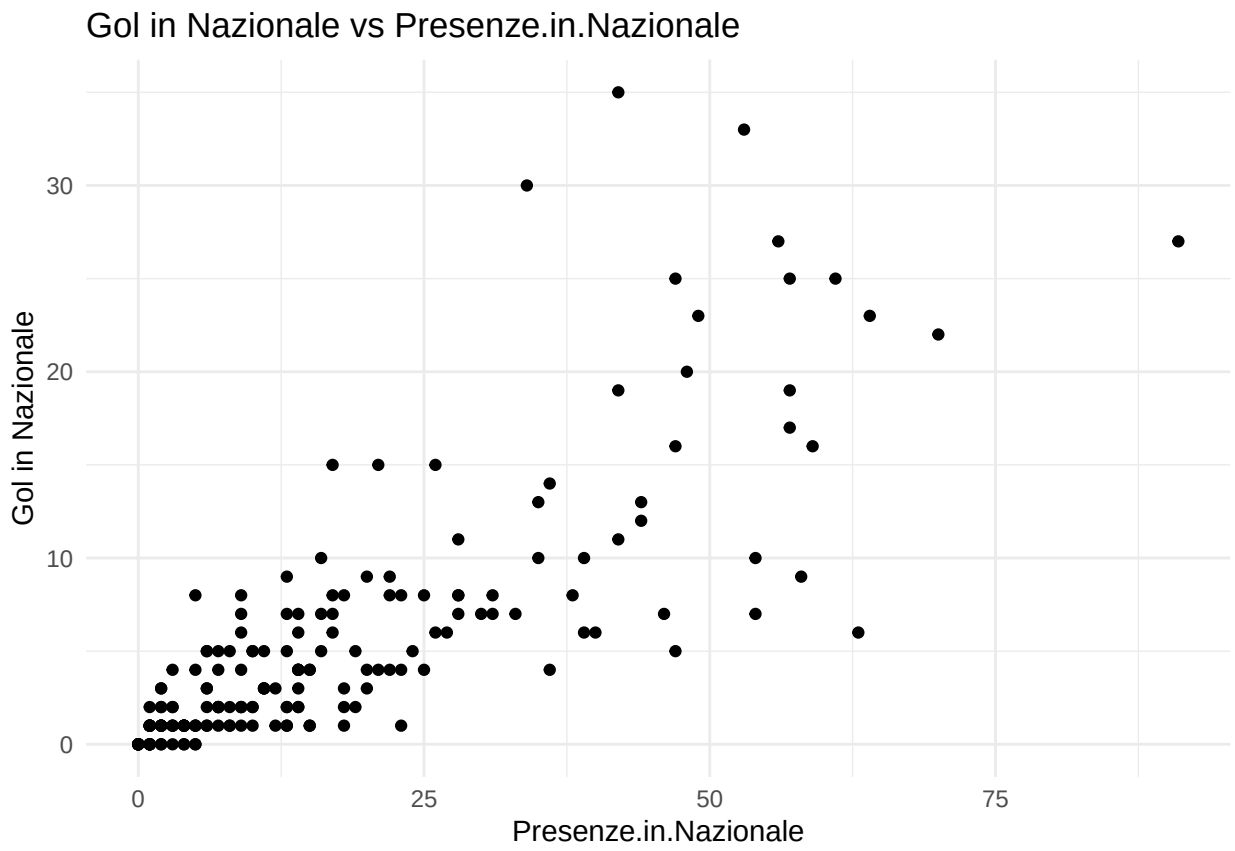

Gol in Nazionale vs Cartellino.Giallo
Gol in Nazionale
30
20
10
0
0
50
100
150
Cartellino.Giallo

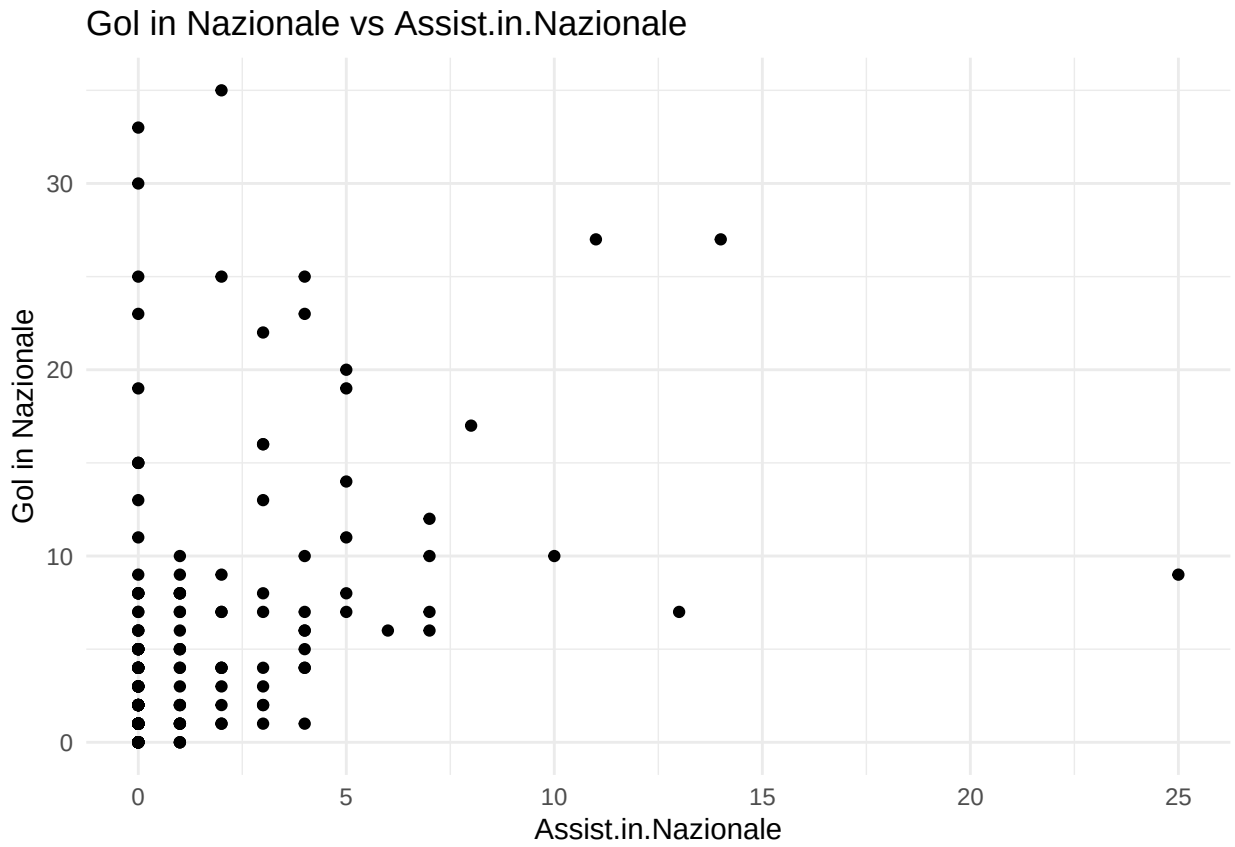

Gol in Nazionale vs Cartellino.Rosso
Gol in Nazionale
30
20
10
0
0
3
6
9
Cartellino.Rosso

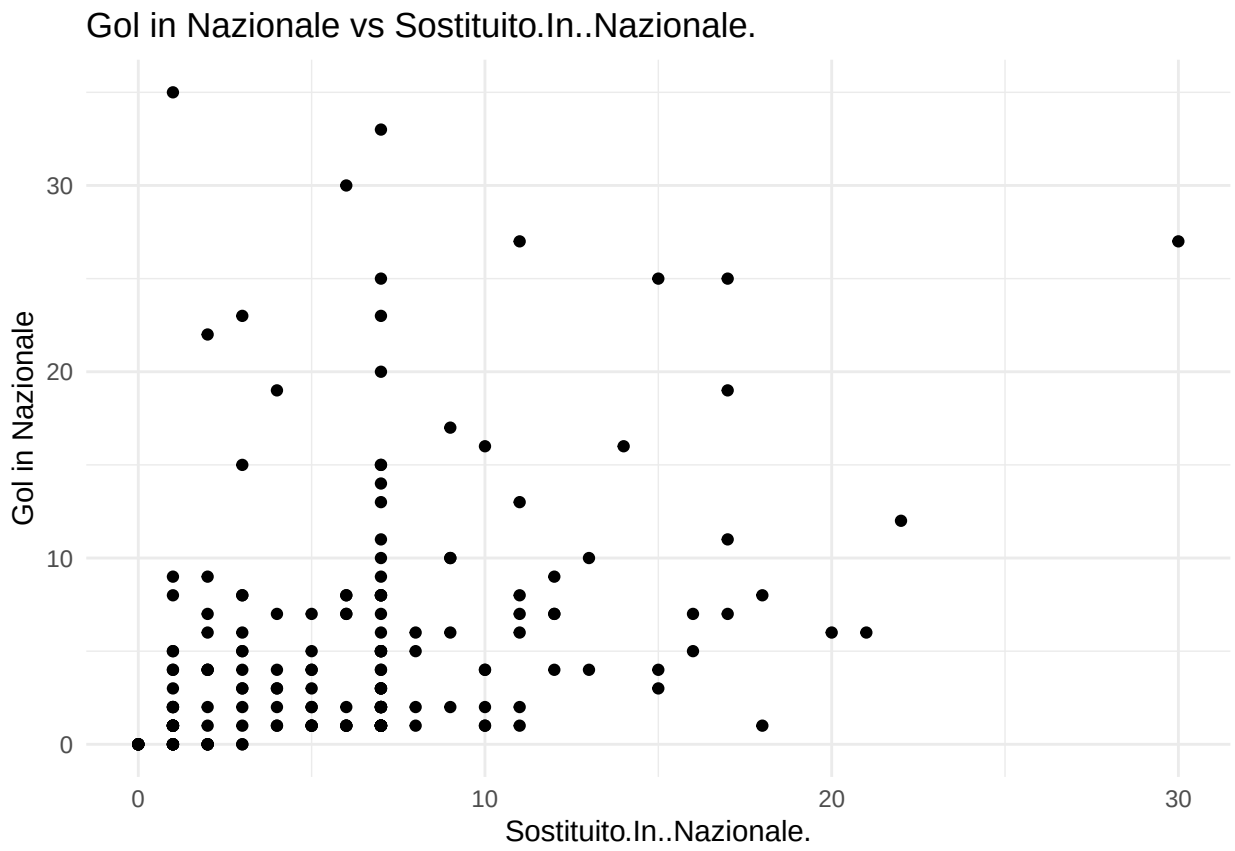

Gol in Nazionale vs Gol.su.Rigore
Gol in Nazionale
30
20
10
0
0
25
50
75
Gol.su.Rigore

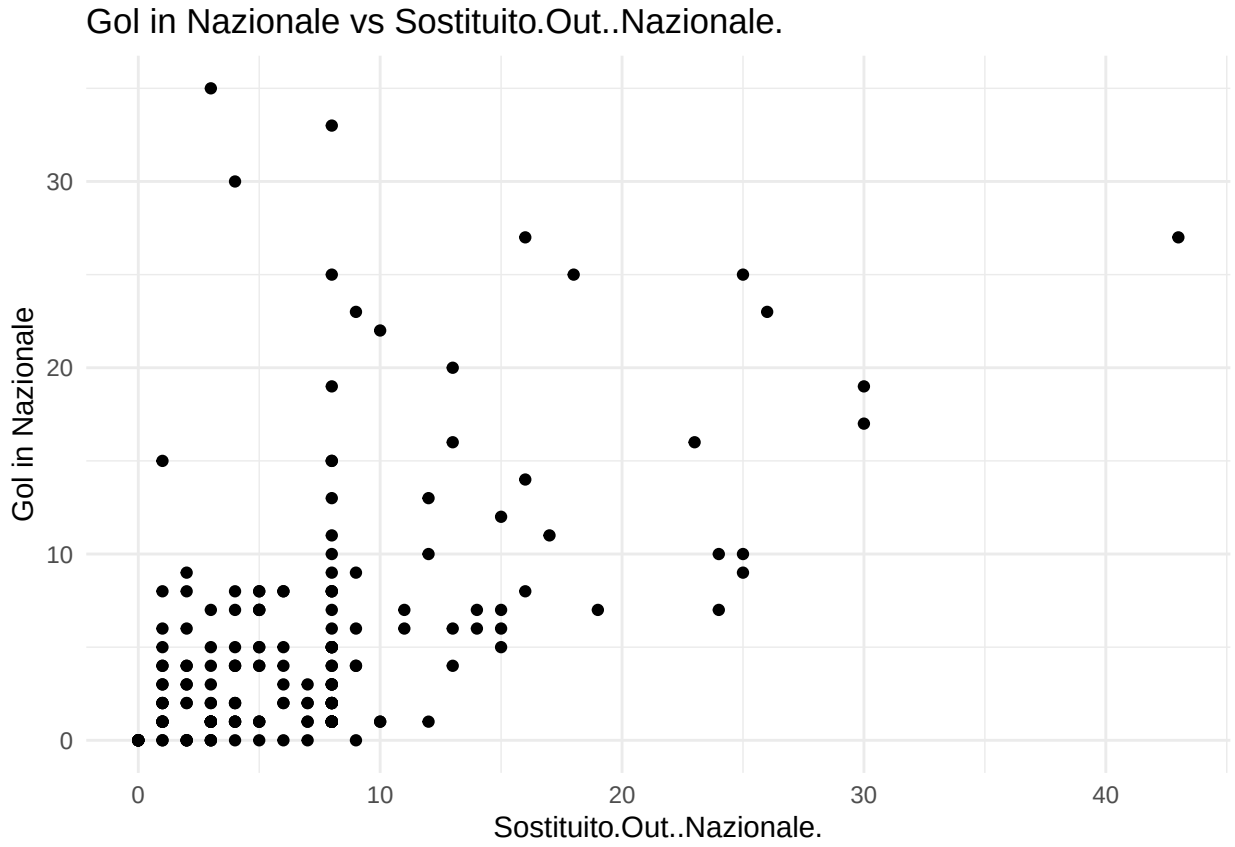

Gol in Nazionale vs Gol.per.Minuti
Gol in Nazionale
30
20
10
0
0
250
500
750
1000
Gol.per.Minuti

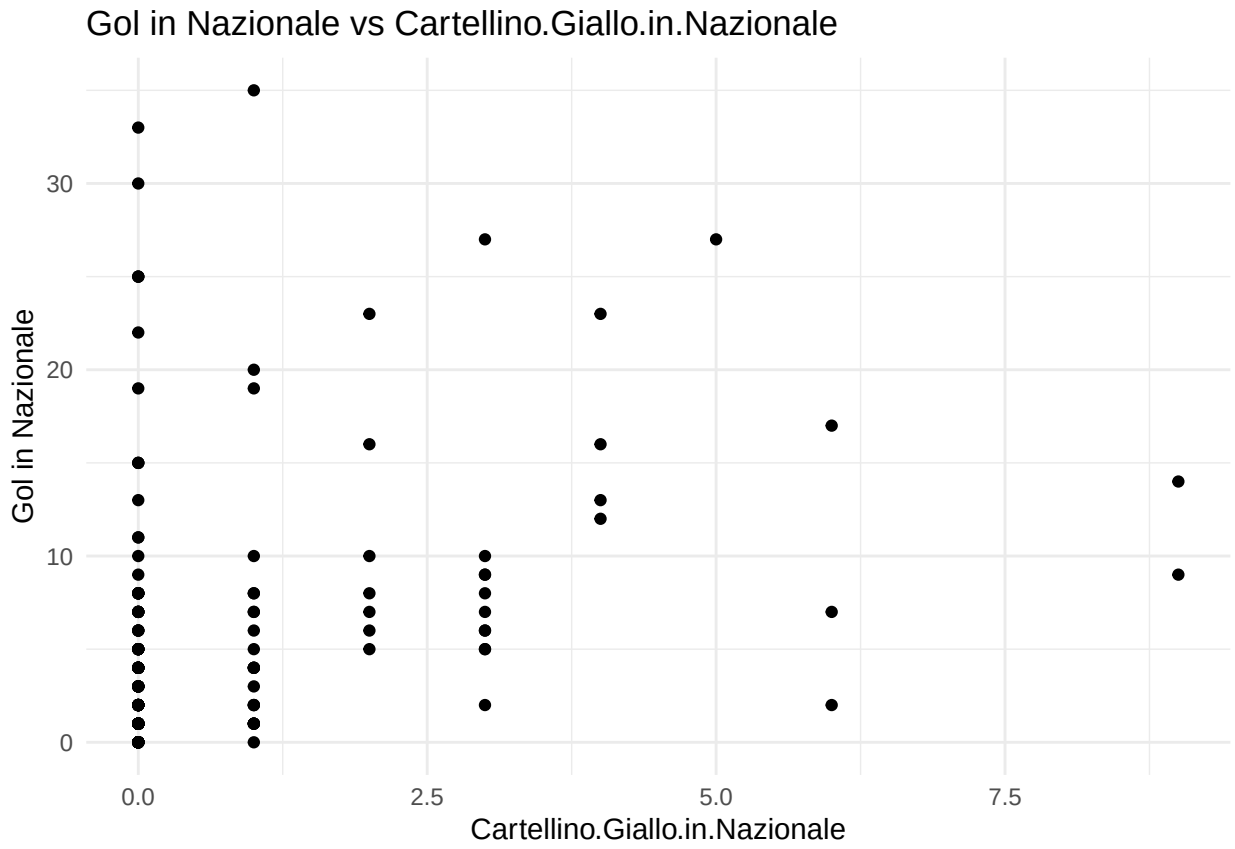

Gol in Nazionale vs Minuti.Giocati
Gol in Nazionale
30
20
10
0
0
250
500
750
1000
Minuti.Giocati

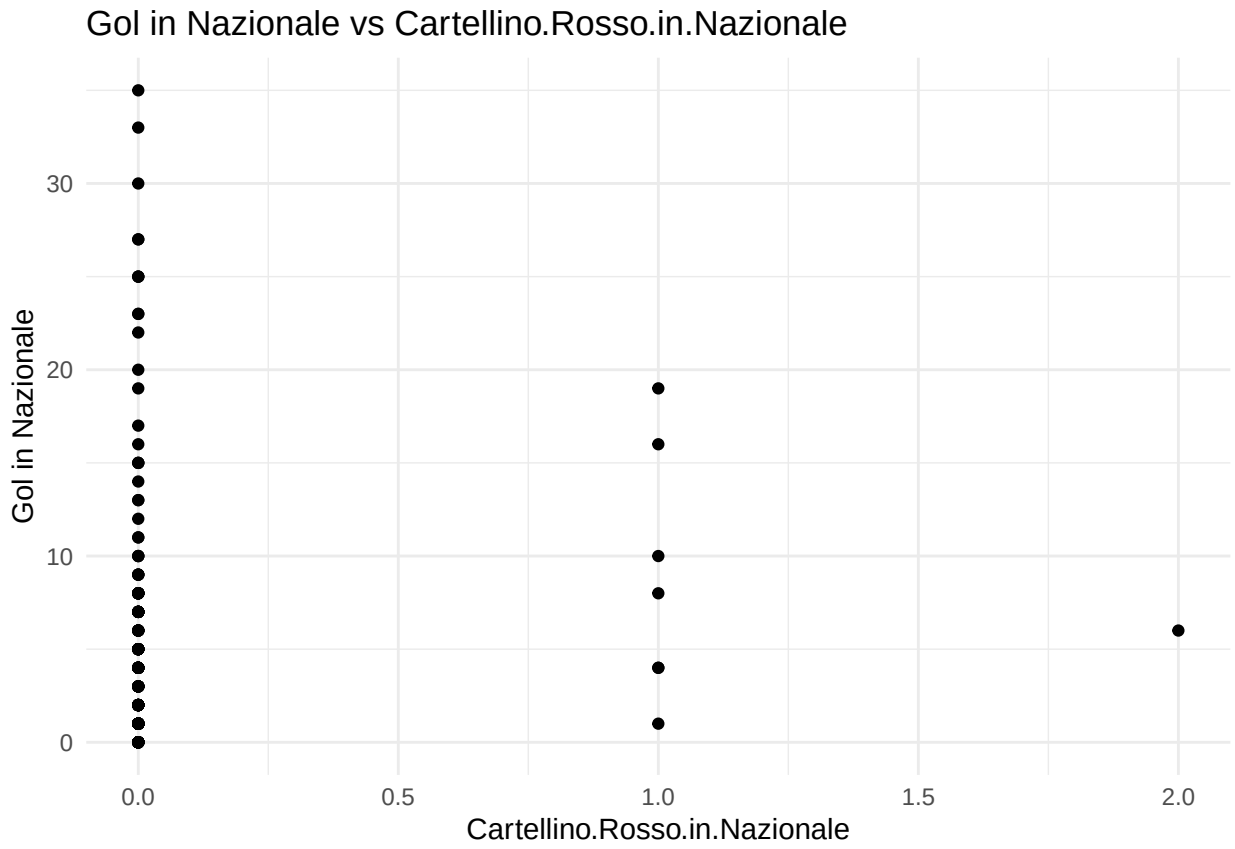

Gol in Nazionale vs Presenze.in.Nazionale
Gol in Nazionale
0
10
20
30
0
25
50
75
Presenze.in.Nazionale

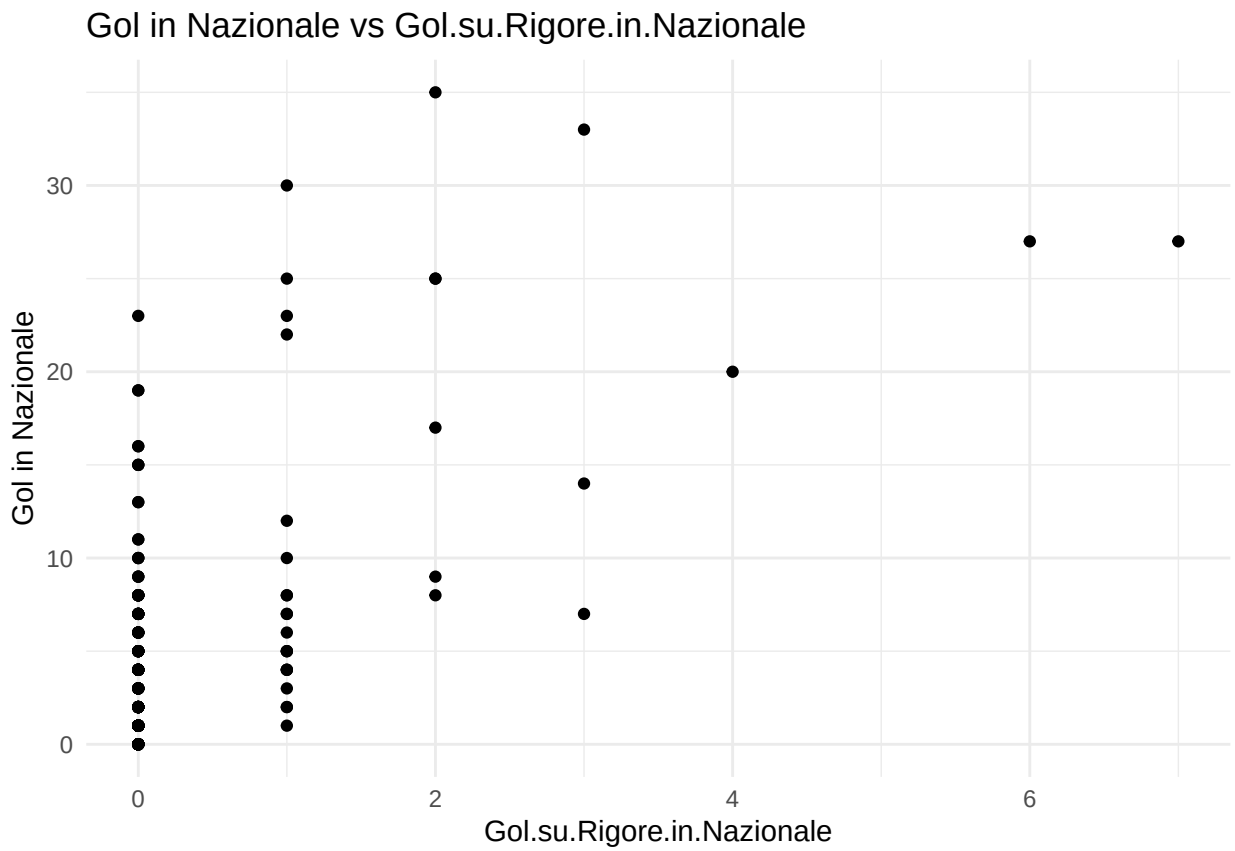

Gol in Nazionale vs Assist.in.Nazionale
Gol in Nazionale
0
10
20
30
0
5
10
15
20
25
Assist.in.Nazionale

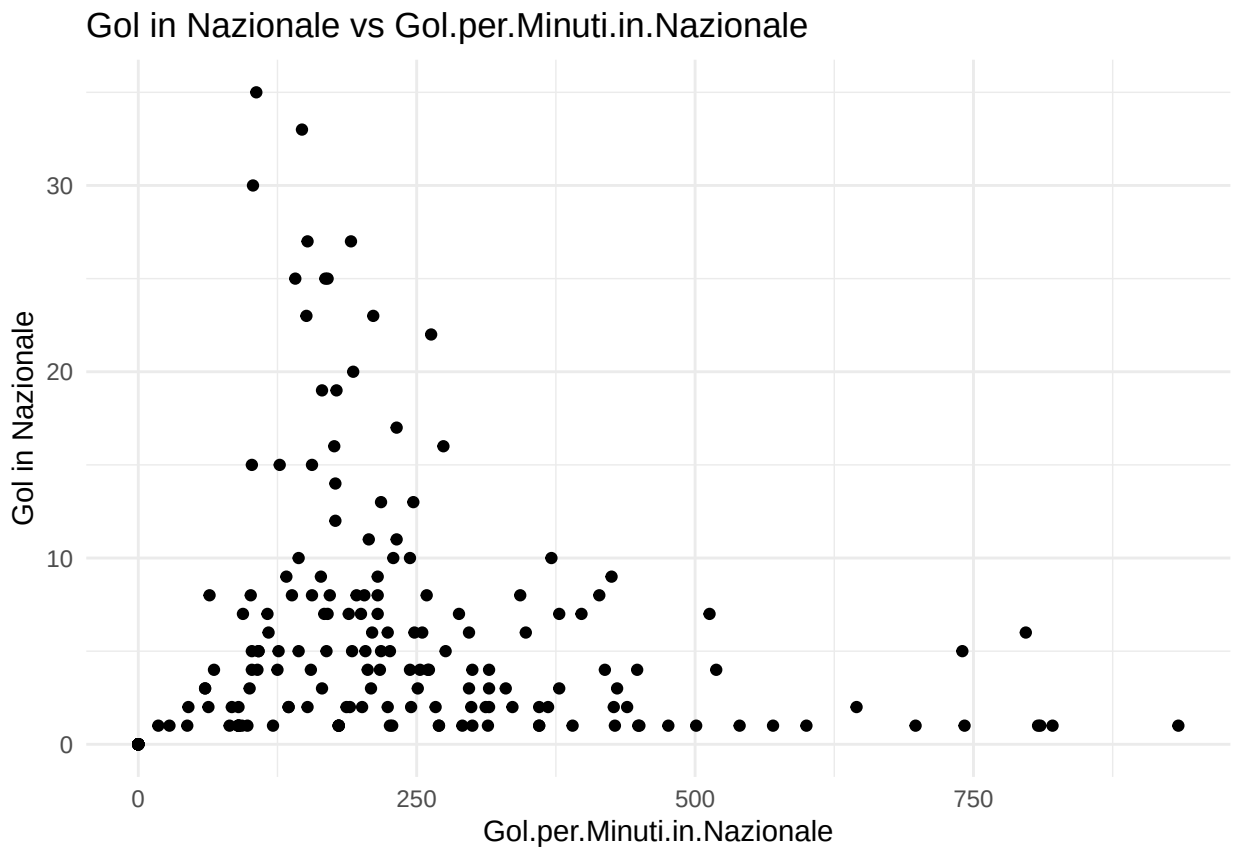

Gol in Nazionale vs Sostituito.In..Nazionale.
Gol in Nazionale
0
10
20
30
0
10
20
30
Sostituito.In..Nazionale.

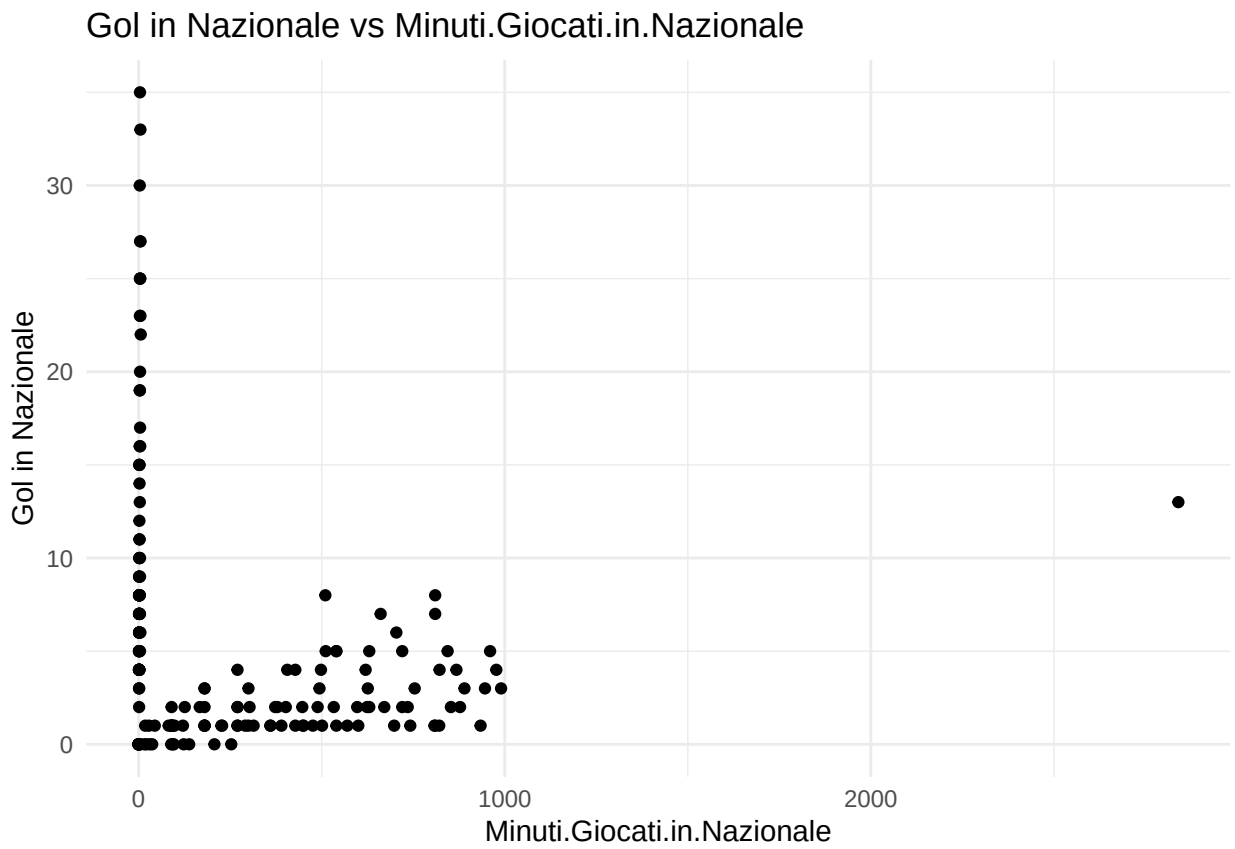

Gol in Nazionale vs Sostituito.Out..Nazionale.
Gol in Nazionale
0
10
20
30
0
10
20
30
40
Sostituito.Out..Nazionale.


Gol in Nazionale vs Cartellino.Giallo.in.Nazionale
Gol in Nazionale
30
20
10
0
0.0
2.5
5.0
7.5
Cartellino.Giallo.in.Nazionale


Gol in Nazionale vs Cartellino.Rosso.in.Nazionale
Gol in Nazionale
30
20
10
0
0.0
0.5
1.0
1.5
2.0
Cartellino.Rosso.in.Nazionale


Gol in Nazionale vs Gol.su.Rigore.in.Nazionale
Gol in Nazionale
30
20
10
0
0
2
4
6
Gol.su.Rigore.in.Nazionale


Gol in Nazionale vs Gol.per.Minuti.in.Nazionale
Gol in Nazionale
30
20
10
0
0
250
500
750
Gol.per.Minuti.in.Nazionale



Controlliamo la presenza di valori mancanti.

```
sum(is.na(df))
```

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

```
df_clean <- na.omit(df)
```

```
df_clean <- df_clean %>% mutate_if(~ is.character(.) && !all(. == df_clean$Player.Name), ~ as.numeric(as
```

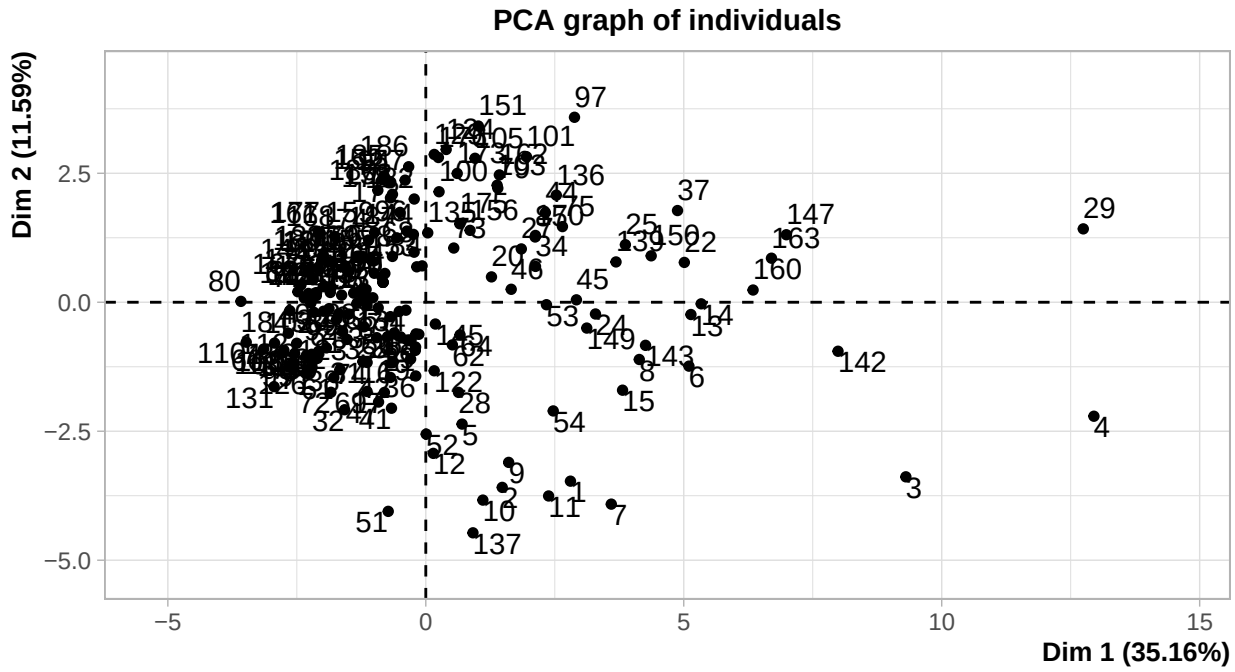
```
str(df_clean)
```

```
## 'data.frame':    187 obs. of  21 variables:
##  $ Player.Name                  : chr  "Giuseppe Meazza" "Silvio Piola" "Roberto Baggio" "Alessandro
##  $ Presenze.Totali              : int  492 612 604 777 276 624 643 476 534 566 ...
##  $ Gol                          : int  307 320 277 316 144 288 266 236 181 162 ...
##  $ Assist                       : int  3 7 152 184 10 46 10 34 9 13 ...
##  $ Sostituito.In                : int  70 4 62 168 70 155 29 112 26 2 ...
##  $ Sostituito.Out               : int  92 1 144 257 92 189 58 102 49 13 ...
##  $ Cartellino.Giallo            : int  0 0 31 57 0 38 7 49 2 0 ...
##  $ Cartellino.Rosso             : int  6 6 2 0 2 1 1 6 5 0 ...
##  $ Gol.su.Rigore                : int  16 26 93 70 9 15 41 25 5 17 ...
##  $ Gol.per.Minuti               : num  143 171 172 170 172 149 207 141 251 313 ...
##  $ Minuti.Giocati               : num  44 54.6 47.6 53.8 24.7 ...
##  $ Presenze.in.Nazionale        : int  53 34 56 91 47 57 61 49 64 70 ...
```

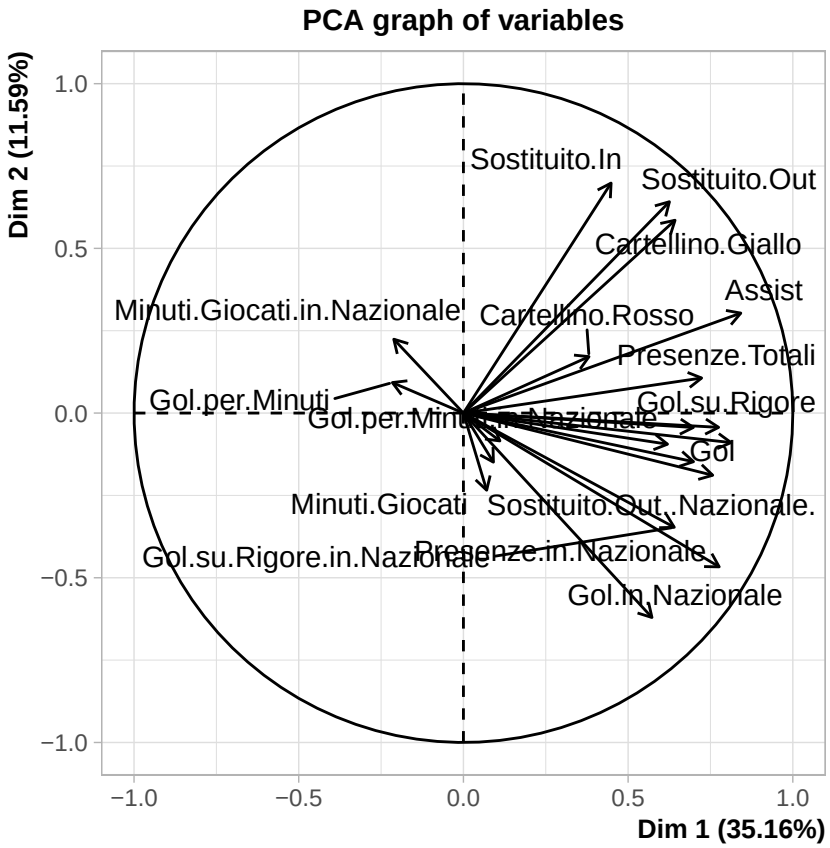
```
##  $ Gol.in.Nazionale              : int  33 30 27 27 25 25 25 23 23 22 ...
##  $ Assist.in.Nazionale           : int  0 0 14 11 0 4 2 4 0 3 ...
##  $ Sostituito.In..Nazionale.     : int  7 6 11 30 7 15 17 3 7 2 ...
##  $ Sostituito.Out..Nazionale.    : int  8 4 16 43 8 25 18 26 9 10 ...
##  $ Cartellino.Giallo.in.Nazionale: int  0 0 3 5 0 0 0 4 2 0 ...
##  $ Cartellino.Rosso.in.Nazionale : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Gol.su.Rigore.in.Nazionale    : int  3 1 7 6 1 2 2 0 1 1 ...
##  $ Gol.per.Minuti.in.Nazionale   : num  147 103 152 191 170 141 168 151 211 263 ...
##  $ Minuti.Giocati.in.Nazionale   : num  4.86 3.09 4.1 5.15 4.26 ...
```

Analisi delle Componenti Principali (ACP)

```
# Eseguiamo una PCA sul dataset pulito, selezionando solo le colonne numeriche
data_numeric <- df_clean %>% select_if(is.numeric)

# Eseguire PCA
pca_result <- PCA(data_numeric, scale.unit = TRUE)
```




```
## Warning: ggrepel: 4 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps
```




```
# Visualizzare i risultati della PCA
summary(pca_result)
```

```
##
## Call:
## PCA(X = data_numeric, scale.unit = TRUE)
##
##
## Eigenvalues
##                        Dim.1   Dim.2   Dim.3   Dim.4   Dim.5   Dim.6   Dim.7
## Variance               7.032   2.319   1.843   1.349   1.102   1.018   0.938
## % of var.             35.158  11.595   9.215   6.746   5.510   5.089   4.691
## Cumulative % of var.  35.158  46.753  55.968  62.713  68.223  73.312  78.003
##                        Dim.8   Dim.9  Dim.10  Dim.11  Dim.12  Dim.13  Dim.14
## Variance               0.835   0.813   0.609   0.474   0.377   0.303   0.257
## % of var.              4.175   4.063   3.044   2.372   1.886   1.514   1.284
## Cumulative % of var.  82.178  86.240  89.284  91.656  93.542  95.056  96.340
##                       Dim.15  Dim.16  Dim.17  Dim.18  Dim.19  Dim.20
## Variance               0.224   0.167   0.141   0.085   0.074   0.041
## % of var.              1.118   0.833   0.706   0.427   0.372   0.204
## Cumulative % of var.  97.458  98.291  98.998  99.424  99.796 100.000
##
## Individuals (the 10 first)
##                                 Dist    Dim.1    ctr   cos2    Dim.2    ctr
## 1                             |  6.410 |  2.804  0.598  0.191 | -3.468  2.773
## 2                             |  5.796 |  1.482  0.167  0.065 | -3.588  2.969
```

```
## 3                              | 11.811 |  9.305  6.585  0.621 | -3.384  2.640
## 4                              | 14.173 | 12.950 12.754  0.835 | -2.209  1.125
## 5                              |  3.764 |  0.703  0.038  0.035 | -2.362  1.286
## 6                              |  6.546 |  5.101  1.979  0.607 | -1.238  0.354
## 7                              |  6.074 |  3.596  0.983  0.350 | -3.913  3.531
## 8                              |  5.815 |  4.138  1.302  0.506 | -1.109  0.284
## 9                              |  4.593 |  1.606  0.196  0.122 | -3.104  2.222
## 10                             |  4.911 |  1.106  0.093  0.051 | -3.835  3.391
##                                  cos2    Dim.3    ctr   cos2
## 1                               0.293 | -2.425  1.707  0.143 |
## 2                               0.383 | -2.645  2.030  0.208 |
## 3                               0.082 | -1.474  0.630  0.016 |
## 4                               0.024 |  0.482  0.067  0.001 |
## 5                               0.394 | -1.098  0.350  0.085 |
## 6                               0.036 | -1.117  0.362  0.029 |
## 7                               0.415 | -1.108  0.356  0.033 |
## 8                               0.036 | -0.765  0.170  0.017 |
## 9                               0.457 | -0.413  0.049  0.008 |
## 10                              0.610 | -0.181  0.010  0.001 |
##
## Variables (the 10 first)
##                                   Dim.1    ctr   cos2    Dim.2    ctr   cos2
## Presenze.Totali                |  0.723  7.438  0.523 |  0.106  0.488  0.011 |
## Gol                            |  0.699  6.940  0.488 | -0.148  0.939  0.022 |
## Assist                         |  0.842 10.085  0.709 |  0.303  3.970  0.092 |
## Sostituito.In                  |  0.448  2.857  0.201 |  0.698 21.022  0.487 |
## Sostituito.Out                 |  0.625  5.562  0.391 |  0.642 17.746  0.412 |
## Cartellino.Giallo              |  0.642  5.870  0.413 |  0.585 14.775  0.343 |
## Cartellino.Rosso               |  0.381  2.067  0.145 |  0.171  1.265  0.029 |
## Gol.su.Rigore                  |  0.775  8.536  0.600 | -0.043  0.081  0.002 |
## Gol.per.Minuti                 | -0.216  0.661  0.046 |  0.093  0.369  0.009 |
## Minuti.Giocati                 |  0.071  0.071  0.005 | -0.234  2.365  0.055 |
##                                  Dim.3    ctr   cos2
## Presenze.Totali                -0.105  0.601  0.011 |
## Gol                            -0.438 10.413  0.192 |
## Assist                          0.049  0.129  0.002 |
## Sostituito.In                  -0.049  0.133  0.002 |
## Sostituito.Out                 -0.012  0.007  0.000 |
## Cartellino.Giallo              -0.017  0.016  0.000 |
## Cartellino.Rosso               -0.037  0.073  0.001 |
## Gol.su.Rigore                  -0.241  3.144  0.058 |
## Gol.per.Minuti                  0.619 20.758  0.383 |
## Minuti.Giocati                  0.204  2.259  0.042 |
```

```
print(pca_result)
```

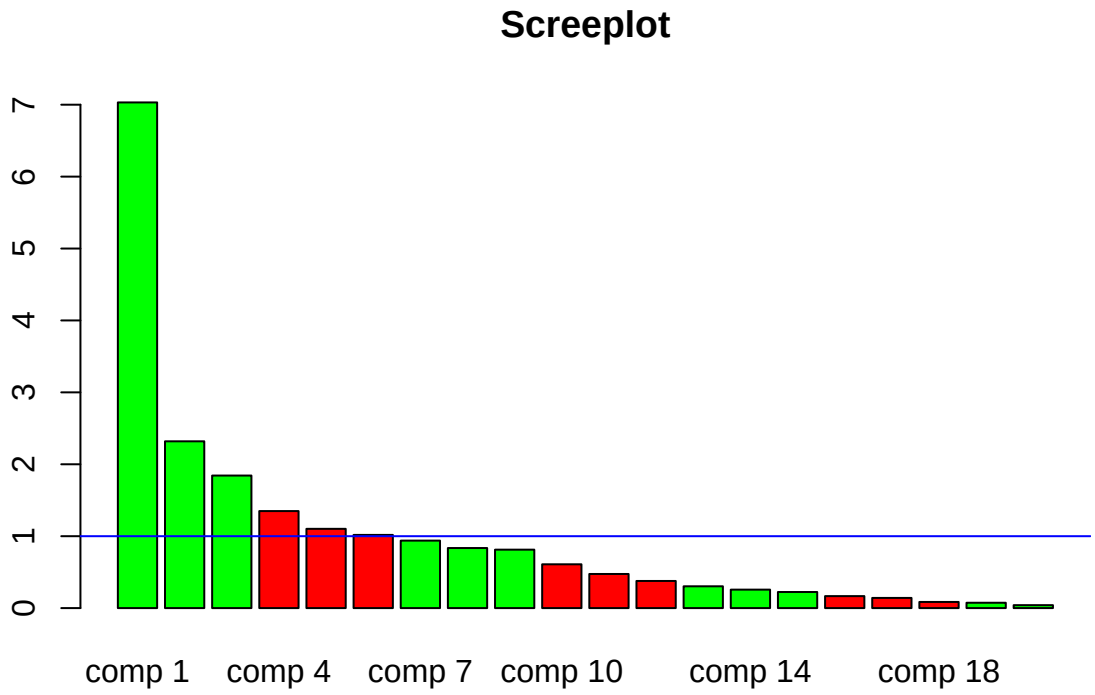
```
## **Results for the Principal Component Analysis (PCA)**
## The analysis was performed on 187 individuals, described by 20 variables
## *The results are available in the following objects:
##
##    name               description
## 1  "$eig"             "eigenvalues"
## 2  "$var"             "results for the variables"
## 3  "$var$coord"       "coord. for the variables"
```

```
## 4  "$var$cor"          "correlations variables - dimensions"
## 5  "$var$cos2"         "cos2 for the variables"
## 6  "$var$contrib"      "contributions of the variables"
## 7  "$ind"              "results for the individuals"
## 8  "$ind$coord"        "coord. for the individuals"
## 9  "$ind$cos2"         "cos2 for the individuals"
## 10 "$ind$contrib"      "contributions of the individuals"
## 11 "$call"             "summary statistics"
## 12 "$call$centre"      "mean of the variables"
## 13 "$call$ecart.type" "standard error of the variables"
## 14 "$call$row.w"       "weights for the individuals"
## 15 "$call$col.w"       "weights for the variables"
```

Grafico 1: PCA graph of individuals Il grafico PCA degli individui mostra la distribuzione dei giocatori in base alle prime due componenti principali. La prima componente (Dim 1) spiega il 35.16% della varianza totale, mentre la seconda componente (Dim 2) spiega l'11.59%. I punti nel grafico rappresentano i giocatori, con le loro coordinate determinate dalle componenti principali. I giocatori che sono vicini nel grafico hanno profili simili in termini di variabili misurate, mentre quelli distanti hanno profili diversi. Ad esempio, il giocatore numero 29 (Francesco Totti) si distingue chiaramente dagli altri, suggerendo caratteristiche eccezionali.

Grafico 2: PCA graph of variables Il grafico PCA delle variabili mostra le relazioni tra le variabili originali del dataset. Le frecce rappresentano le variabili, con la lunghezza della freccia che indica il contributo della variabile alla componente principale. Le variabili vicine tra loro sono altamente correlate. Ad esempio, Minuti.Giocati.in.Nazionale e Gol.per.Minuti sono fortemente correlate. Inoltre, Presenze.Totali, Assist, e Cartellino.Giallo mostrano correlazioni tra loro. Le variabili lungo la circonferenza dell'unità sono ben rappresentate dalle due componenti principali.

```r
# Barplot degli autovalori
barplot(pca_result$eig[,1], main = "Screeplot", col=c(rep("green",3), rep("red",3)))
abline(h=1, col="blue")
```




```
# Coordinate delle variabili
pca_result$var$coord
```

```
##                                      Dim.1       Dim.2       Dim.3
## Presenze.Totali                  0.72319734  0.10636031 -0.10524308
## Gol                              0.69859130 -0.14758522 -0.43807773
## Assist                           0.84208422  0.30343686  0.04868442
## Sostituito.In                    0.44823423  0.69820600 -0.04943472
## Sostituito.Out                   0.62537725  0.64150953 -0.01162161
## Cartellino.Giallo                0.64245039  0.58533882 -0.01721649
## Cartellino.Rosso                 0.38128346  0.17129562 -0.03679563
## Gol.su.Rigore                    0.77475759 -0.04334092 -0.24069523
## Gol.per.Minuti                  -0.21561996  0.09254868  0.61851475
## Minuti.Giocati                   0.07063976 -0.23416884  0.20402737
## Presenze.in.Nazionale            0.77648185 -0.46684942  0.19628536
## Gol.in.Nazionale                 0.57236330 -0.62063140 -0.17731668
## Assist.in.Nazionale              0.81243098 -0.08963680  0.25397241
## Sostituito.In..Nazionale.        0.61958166 -0.09448663  0.26494849
## Sostituito.Out..Nazionale.       0.75684914 -0.18873494  0.21374847
## Cartellino.Giallo.in.Nazionale   0.69812637 -0.04369027  0.26700128
## Cartellino.Rosso.in.Nazionale    0.09103887 -0.14900413  0.15989593
## Gol.su.Rigore.in.Nazionale       0.64010471 -0.34682785 -0.11404356
## Gol.per.Minuti.in.Nazionale      0.11074889 -0.08459298  0.76737480
## Minuti.Giocati.in.Nazionale     -0.21137706  0.22420768  0.44958738
##                                      Dim.4       Dim.5
## Presenze.Totali                 0.4735338213 -0.03393573
```

```
## Gol                              0.3210924466 -0.10117991
## Assist                          -0.1067826169  0.08216596
## Sostituito.In                   -0.0864524352 -0.23124260
## Sostituito.Out                  -0.0812772398 -0.11027552
## Cartellino.Giallo                0.0003163439  0.23646257
## Cartellino.Rosso                 0.4997131786  0.21570240
## Gol.su.Rigore                    0.0123703918  0.13981538
## Gol.per.Minuti                   0.1189035038  0.21314545
## Minuti.Giocati                  -0.1221258559  0.60382579
## Presenze.in.Nazionale            0.0679693722 -0.11620042
## Gol.in.Nazionale                 0.0194232146 -0.17085780
## Assist.in.Nazionale             -0.1292033869  0.16923469
## Sostituito.In..Nazionale.       -0.2109538061 -0.39980565
## Sostituito.Out..Nazionale.      -0.2309181719 -0.11554059
## Cartellino.Giallo.in.Nazionale -0.1585338546  0.31131469
## Cartellino.Rosso.in.Nazionale    0.6407441887 -0.05948848
## Gol.su.Rigore.in.Nazionale      -0.2963462495 -0.03414735
## Gol.per.Minuti.in.Nazionale      0.2716681502 -0.15368978
## Minuti.Giocati.in.Nazionale     -0.0244683762 -0.34117746
```

```
# Coordinate degli individui
pca_result$ind$coord
```

```
##            Dim.1       Dim.2       Dim.3        Dim.4        Dim.5
## 1    2.804151720 -3.46770588 -2.42545960  0.945995945 -1.160819987
## 2    1.481704000 -3.58842178 -2.64481688  2.071317650 -0.072740638
## 3    9.305197045 -3.38376786 -1.47402354 -2.364141798  0.639636423
## 4   12.950183981 -2.20921152  0.48164089 -3.770026352 -2.094816121
## 5    0.702970264 -2.36156704 -1.09805974 -0.305323749 -1.143734564
## 6    5.100961366 -1.23816948 -1.11735360 -0.821454652 -2.267045549
## 7    3.595508316 -3.91322586 -1.10797021 -0.100481015 -1.548300115
## 8    4.137849675 -1.10936124 -0.76469215  0.399132012  0.534941713
## 9    1.606367863 -3.10380903 -0.41251428  1.021845861 -0.139600917
## 10   1.106407972 -3.83483305 -0.18115968  0.508969222 -0.084068859
## 11   2.380672681 -3.75473941 -0.56276322 -1.720091220 -0.353084647
## 12   0.145433652 -2.92922580 -0.48166430  2.881850710 -0.648659794
## 13   5.139077624 -0.23785641  0.02913524 -0.739994296 -1.551357607
## 14   5.338189636 -0.03062318 -0.37747584  0.033555491  0.070713409
## 15   3.814041868 -1.70452744  0.05925301  3.476888220 -0.190911942
## 16  -0.637456945 -1.16413564 -1.37375230 -0.542059999 -0.947647473
## 17  -0.802467191 -1.74752406 -2.01137110  1.040938464 -0.118188884
## 18  -0.189585442 -0.61277026  0.17011431  1.429102572  0.430217056
## 19  -0.293708216 -1.10167224 -0.37200964  0.043931180 -0.511885278
## 20   1.273383252  0.49139772  3.45925651 -0.191873595 -2.253823121
## 21  -0.202619104 -0.93326649 -0.63031535  0.344420829 -0.534343828
## 22   5.011593663  0.77207878 -0.93506953  0.575778444 -0.969439151
## 23  -0.739233901 -0.66547031 -0.49891769 -0.282300380 -0.449864278
## 24   3.289397907 -0.22722819  0.38429991  2.211745467 -0.456088636
## 25   3.868729041  1.11522378  0.64283464 -0.603584332  0.022781671
## 26  -0.489347269 -0.67056404 -1.16425502  0.117254874 -0.543363166
## 27   1.851666845  1.03355081 -0.95824154  0.512970176  1.120227238
## 28   0.634284756 -1.74709509 -1.36473566  1.266018579  1.410229796
## 29  12.744321468  1.42254380  1.80084707 -0.032392443  3.922463414
## 30  -1.875574919 -0.12374368  0.03858648 -1.209430860 -0.870161173
```

```
## 31  -1.220148784 -1.15001400 -0.72080846  1.004303584 -0.170633169
## 32  -1.573442666 -2.08374686 -0.54866283  3.010504846  0.317350729
## 33  -1.196295192 -0.47645926  0.19702617 -0.252267340 -0.271161458
## 34   2.126866809  0.69453626 -1.29541982  0.160471142  0.357840061
## 35  -0.951285931 -0.68694084  0.07115360 -0.260151517 -0.524374728
## 36  -0.194426678 -1.42892605 -0.34476142  0.434549934  0.376631208
## 37   4.879491168  1.77740595 -1.13831805 -0.092170739 -0.581408017
## 38  -1.664272351 -1.30694241  0.26106805  0.439399566  0.189382435
## 39  -0.197826696 -0.86801633 -0.21641101  1.501694027 -0.070890669
## 40  -2.225835618 -0.01312770  0.63225154 -1.607524379 -1.257481865
## 41  -0.664795306 -2.05400730 -1.14449956  0.132528465  0.155348976
## 42  -0.684883207 -1.45709752 -0.78453682 -0.233296240  0.030262562
## 43  -0.308899864 -0.74676612 -1.13666050  0.171842095  0.499440342
## 44   2.317841125  1.73701769 -0.72955883  0.086453140 -0.725118965
## 45   2.919615510  0.04709978 -1.05711649  0.677468537  0.952924929
## 46   1.655674646  0.25139309 -0.40995524 -0.941838714 -0.882928521
## 47  -0.915435475 -1.93399204  0.83513821  0.493894594  0.668801268
## 48  -2.098266120 -1.03109721 -0.37391452  0.177720785 -0.061779616
## 49  -1.495058736 -0.23058741  1.02926830 -0.420690051  0.002166743
## 50  -2.100662213 -1.08753438 -0.57389513 -0.097701993  0.030433315
## 51  -0.728258100 -4.05330604  3.61226894 -3.194232495  8.008771891
## 52   0.007908557 -2.55588878  1.94314362 -2.301800342 -0.619328112
## 53   2.338926190 -0.04792423  0.65596413 -0.318901184 -1.020869863
## 54   2.470257879 -2.10658936  4.46208048  6.889137432  0.146623818
## 55  -0.922383118 -0.10879381 -0.84325016 -0.015521507 -1.027459502
## 56  -0.609877061 -0.59301370 -1.48624966  0.762169931 -0.421823392
## 57  -1.200176819  0.04406438 -0.24948699 -0.592587983 -0.742919600
## 58  -1.983068099 -0.17424553  0.56599658 -1.811841534 -1.171211050
## 59  -1.685630219 -0.22104903  0.34300452 -1.713111786 -1.610260715
## 60  -1.214697555  0.26757790  1.61912623  1.084483778  0.948013986
## 61  -1.782029756 -1.46014879 -0.73161265  0.687157859  0.712276947
## 62   0.515417469 -0.82611293 -0.76554810  0.244293630  0.577424325
## 63  -0.134770285 -0.62127163 -0.22940061  0.195994178 -0.663109687
## 64   0.664057662 -0.63160437  4.16398817  1.002989242  1.508939161
## 65  -1.199663528  0.05236070 -0.85304655 -0.360621728 -0.480920215
## 66  -3.144698740 -0.90848084  0.08004672 -1.115353790  0.154897464
## 67  -2.169927133 -1.11800465  0.30718765 -1.555854077  0.058392780
## 68  -0.226326598  0.97041095  0.75946974  2.998147538 -0.262177508
## 69  -1.138425465 -1.72877780 -1.09347417 -0.163797725  0.561376214
## 70  -0.513778040 -0.17852031 -0.35492334  0.936573004 -0.238046534
## 71  -2.206096753 -1.26930007  0.08449897  0.694940205  0.741155708
## 72  -1.839172373 -1.74540830  0.45693576  0.270918483  0.599568959
## 73   0.542322072  1.05077624  0.28622928  0.297710872 -0.330327087
## 74  -1.137499860 -1.16722082  1.07601577  1.557079567  1.034150073
## 75   2.648943116  1.46643392 -0.19340608  2.453073223 -1.185718518
## 76  -2.233880299  0.13182113  0.08718591 -1.414996701 -0.348766680
## 77  -2.354322095  0.08217354 -0.24265585 -1.763237957 -0.560916793
## 78  -2.931662260 -0.79179062  0.75340136  0.086876987  0.591360477
## 79   1.395975397  2.21126201 -2.05556186  1.865774275  1.252675309
## 80  -3.583212036  0.01707470  3.74070445 -0.170593796  0.473590147
## 81  -1.156977665  0.24831864  0.95501652  0.108636490 -1.233689705
## 82  -2.479686298  0.20377678  0.56663952 -0.744250509 -1.009589656
## 83  -0.686240943 -0.27923380  0.45682634  0.678293444 -0.403045618
## 84  -0.071414261  0.70500908  1.31425723  0.493860036  0.358991285
```

```
## 85   2.119418969  1.26495210 -0.25857252 -0.416759225 -1.308822406
## 86  -1.832593480  0.28606645  0.23127737 -0.616829621  0.169934169
## 87  -1.631299475  0.13827426 -0.21861882 -0.705714603 -0.318230569
## 88  -1.332860863 -0.01696248 -0.82342221 -0.535446979 -0.802337363
## 89  -2.151029167  0.17424586  0.31906732 -1.223619628 -0.496739594
## 90  -2.082542846  0.64258832  1.05414367 -0.786168994 -0.163869747
## 91  -1.610945025 -0.57064905  0.28121140  0.652277411  0.121117211
## 92  -1.726537955 -0.32799608 -0.55749379  0.386976781  0.025284687
## 93  -1.398496681  0.17726768  0.47066503 -0.127041306 -0.981617378
## 94  -0.684927682  2.33452750  2.28706156  1.219029609  1.325372638
## 95  -2.153669584 -1.06582659  0.13412840  0.619712104 -0.310162014
## 96  -0.364248166  1.37952411 -0.33363397  0.884340513  0.132218518
## 97   2.882142014  3.58534320 -0.89784587  1.222218493  0.669435284
## 98  -0.646427491  0.88803211  0.28429318  1.467888787 -0.005361757
## 99  -2.552761080 -1.20690626  0.22139576 -1.437267967 -0.550505360
## 100  0.257167214  2.14188004  0.37824455  0.314566407 -0.211566415
## 101  1.952139960  2.82030000 -0.47399104  0.984150453 -1.130069361
## 102 -2.170424591 -0.19443795  2.43492364 -0.752326373 -0.507079093
## 103  1.382921155  2.26774179  1.53117993  0.478863813  0.231678528
## 104 -2.768002632 -0.97732534 -1.07641912 -0.445003593  0.752541454
## 105  0.954139383  2.79034849 -1.60961321  0.192817111  0.209512058
## 106 -0.828876653  0.38597645 -1.24424945  0.731350642  0.064010606
## 107 -0.828876653  0.38597645 -1.24424945  0.731350642  0.064010606
## 108 -0.828876653  0.38597645 -1.24424945  0.731350642  0.064010606
## 109 -0.828876653  0.38597645 -1.24424945  0.731350642  0.064010606
## 110 -3.480284755 -0.77440451 -0.06670499 -1.029426264  0.851502791
## 111 -2.121790072  0.12742934 -0.57651039 -1.503439645 -0.808465291
## 112 -2.664151145 -0.59996128 -0.27630482 -0.001135181  0.806716759
## 113 -1.085807165  0.07212782 -0.69753626 -0.183032608 -0.440982973
## 114 -0.793444727  0.55855393 -0.09729626  1.241976585  0.494463305
## 115 -2.314310737 -1.02715277 -0.43423768 -0.076695570  0.101810767
## 116 -2.076925676 -0.94992244 -0.65294418  0.288403791  0.380776589
## 117 -1.026851120  0.08732936 -0.44868343  0.060212096 -0.591396382
## 118 -2.749966690 -0.93083655  0.15203398 -0.016926076  0.451735761
## 119 -2.275478192  0.17113731  1.07724583 -1.239388616 -0.938559729
## 120 -2.211689279 -1.00876161 -0.51208283  0.302563730  0.525554610
## 121 -1.813598391  0.77004819  0.08895424  0.629837528  0.117220021
## 122  0.162900963 -1.32843823 -0.60301778  0.535460055 -0.372736531
## 123 -1.523019379 -0.72042700  1.24471295  0.340988997  0.193786647
## 124  0.395307159  2.96107484 -0.81841988  0.152495715  0.164543739
## 125 -2.117614462  0.18717575  1.52166973 -0.833228469 -0.896305368
## 126 -2.307361907 -1.36542646  0.74559478  2.272909210 -0.642995328
## 127 -1.850644648  0.18957227  1.66225745 -0.508116619 -0.954319253
## 128 -1.986330410  0.30138030  2.81376290 -0.188576628 -0.751406998
## 129  0.164145056  2.86499477 -0.63610810  0.220237087 -0.666598408
## 130 -2.019524738  0.81329495  2.98224492  0.812316258 -0.412010196
## 131 -2.941037325 -1.63536555  1.90743889 -1.544290814  3.053342896
## 132 -1.926077037 -0.88452820  2.66628027  0.943496864 -0.629557536
## 133 -2.502040323 -0.79495090  1.74203442 -0.886542584 -0.644078204
## 134 -0.385507935 -0.15388489  0.98630804  1.319269268 -0.527407561
## 135  0.037836281  1.34564537  3.11211510  0.721772589 -1.360338392
## 136  2.530224504  2.07521924  2.51810035  0.111453513 -1.584629892
## 137  0.914088363 -4.46905158 -2.10877627 -0.167680325 -0.132769774
## 138 -2.541169829 -0.99180260 -0.40015445  0.066750166  0.553336088
```

```
## 139  3.687051617  0.78138232  0.92762261  1.879827913  0.376526979
## 140 -1.309706456  0.19225349 -0.75210056 -0.107000970 -0.199365400
## 141 -0.998808307  0.56183113  1.30613995 -0.238286144 -1.388669191
## 142  7.990424821 -0.94985814 -0.60800865 -0.844629318  0.963727385
## 143  4.259257172 -0.83505481  0.54529925 -1.702267126 -0.739239035
## 144 -0.239875914  1.31508952  0.01736291 -0.481006556 -0.927878660
## 145  0.185719852 -0.42205598  0.37152015 -1.209805659 -0.773190612
## 146 -2.262857474  0.63900840 -1.45870943 -0.269301545  1.026302466
## 147  6.988185783  1.30864230 -0.46245211 -0.420166839  2.634311009
## 148 -1.490032832  0.59510686  1.85087301 -0.363820810  0.035882746
## 149  3.120268227 -0.50031150  2.66270866 -1.693849337  0.290266827
## 150  4.368769229  0.89974740 -0.33622577  0.048798504  1.615575560
## 151  1.018594827  3.41582750  0.39370295  0.149097013 -0.489374692
## 152 -0.842487947  2.37845252  0.35690110 -0.243557663  1.261441834
## 153 -0.174758198  0.68703442 -0.60858208 -0.984398028  0.294957388
## 154 -1.571152035  0.66788386 -1.51284520 -0.288634248  0.816597637
## 155 -1.051153754  0.80809195 -0.28781440 -0.813053756 -0.592290199
## 156  0.857737533  1.39345236  2.51683101 -0.693026202  0.363107242
## 157 -0.400459563  2.36661558 -1.74478251 -0.015461805  0.551542364
## 158 -0.646057679  2.09249726 -1.27892000 -0.193474999  0.758005425
## 159 -0.993558149  1.35445624 -1.45307976 -0.608744041  0.153570212
## 160  6.343759710  0.23717477  1.10563681 -1.010481376  0.072318641
## 161 -2.083810782  1.33754974 -0.67537640 -0.608477349  0.906706100
## 162  1.426769101  2.46808749  4.03799492  0.704846010 -0.733605507
## 163  6.700345907  0.85382826  2.82768144 -1.422463320  0.580872814
## 164 -1.327569218  0.89275523  1.17823027 -0.545357117 -0.486213317
## 165 -0.851644723  0.49138229  1.88429752 -0.537187529  0.126415633
## 166 -1.218063974  0.88629805 -1.31009052 -0.389786518  0.457482697
## 167 -2.419645493  0.33535544 -1.27046832 -1.005547842  0.073645542
## 168 -1.763592445  1.27477528 -0.08629916 -0.316322540  1.463556382
## 169 -1.138409433 -0.06329175  1.35997222 -1.369361815 -1.366969948
## 170  2.124766608  1.29481253  0.38205274 -1.040953150 -1.614319042
## 171 -2.188786606  0.43535857 -1.17962965 -1.173356947  0.707071287
## 172 -0.498754705  1.74160499 -1.34086331 -0.069825933  0.530723475
## 173  0.608904336  2.49647295 -2.47495696  0.442275516  0.771108782
## 174 -1.109742192  1.20694864 -1.92175459 -0.179588817  1.000683405
## 175  0.659365875  1.52048284 -1.20056896  0.116184878 -0.125522236
## 176  0.244518531  2.80563086 -1.14676112 -0.412403610  0.978741235
## 177 -2.046878191  1.34541564 -0.92141799 -0.688967356  0.904116728
## 178 -0.684258020  2.02377669 -2.03448665 -0.280055110  0.532498559
## 179 -1.285552291  0.89228904 -1.44296412 -0.787070731  0.142245995
## 180 -0.929142661  2.16979222 -1.07034712 -0.482380103  0.926965696
## 181 -1.832747859  0.79010143 -1.21291605 -0.390548526  0.669046447
## 182 -0.224010234  2.00103357 -1.60528947 -0.359185258  0.755876666
## 183 -1.984458934  0.89383394 -1.55023354 -0.482381598  0.764849367
## 184 -2.636561902 -0.16371338 -1.06023981 -0.845092054  0.480981581
## 185 -0.811129278  2.45099427 -1.70626521  0.070833967  0.722384990
## 186 -0.331842245  2.62665799 -0.97686673 -0.180662025  0.555266877
## 187 -0.562450396  1.24868512  0.82142240 -0.462809750 -0.949843070
```

Il barplot degli autovalori mostra che le prime tre componenti principali spiegano la maggior parte della varianza nei dati, con autovalori significativamente superiori a 1. Questo suggerisce che queste componenti sono importanti e dovrebbero essere considerate nell'analisi. Le componenti successive hanno autovalori inferiori a 1, indicando che spiegano una varianza marginale e possono essere trascurate. Pertanto, possi-

amo concentrare la nostra analisi sulle prime tre componenti principali per ottenere una rappresentazione significativa della varianza nel dataset

```
str(df_clean)
```

```
## 'data.frame':    187 obs. of  21 variables:
##  $ Player.Name                   : chr  "Giuseppe Meazza" "Silvio Piola" "Roberto Baggio" "Alessandr
##  $ Presenze.Totali               : int  492 612 604 777 276 624 643 476 534 566 ...
##  $ Gol                           : int  307 320 277 316 144 288 266 236 181 162 ...
##  $ Assist                        : int  3 7 152 184 10 46 10 34 9 13 ...
##  $ Sostituito.In                 : int  70 4 62 168 70 155 29 112 26 2 ...
##  $ Sostituito.Out                : int  92 1 144 257 92 189 58 102 49 13 ...
##  $ Cartellino.Giallo             : int  0 0 31 57 0 38 7 49 2 0 ...
##  $ Cartellino.Rosso              : int  6 6 2 0 2 1 1 6 5 0 ...
##  $ Gol.su.Rigore                 : int  16 26 93 70 9 15 41 25 5 17 ...
##  $ Gol.per.Minuti                : num  143 171 172 170 172 149 207 141 251 313 ...
##  $ Minuti.Giocati                : num  44 54.6 47.6 53.8 24.7 ...
##  $ Presenze.in.Nazionale         : int  53 34 56 91 47 57 61 49 64 70 ...
##  $ Gol.in.Nazionale              : int  33 30 27 27 25 25 25 23 23 22 ...
##  $ Assist.in.Nazionale           : int  0 0 14 11 0 4 2 4 0 3 ...
##  $ Sostituito.In..Nazionale.     : int  7 6 11 30 7 15 17 3 7 2 ...
##  $ Sostituito.Out..Nazionale.    : int  8 4 16 43 8 25 18 26 9 10 ...
##  $ Cartellino.Giallo.in.Nazionale: int  0 0 3 5 0 0 0 4 2 0 ...
##  $ Cartellino.Rosso.in.Nazionale : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Gol.su.Rigore.in.Nazionale    : int  3 1 7 6 1 2 2 0 1 1 ...
##  $ Gol.per.Minuti.in.Nazionale   : num  147 103 152 191 170 141 168 151 211 263 ...
##  $ Minuti.Giocati.in.Nazionale   : num  4.86 3.09 4.1 5.15 4.26 ...
```

Clustering

```
df_clean <- na.omit(df_clean)

# Escludere la colonna Player.Name per la normalizzazione
df_clean_numeric <- df_clean %>% select(-Player.Name)

# Verifica il tipo di ogni colonna
str(df_clean_numeric)
```

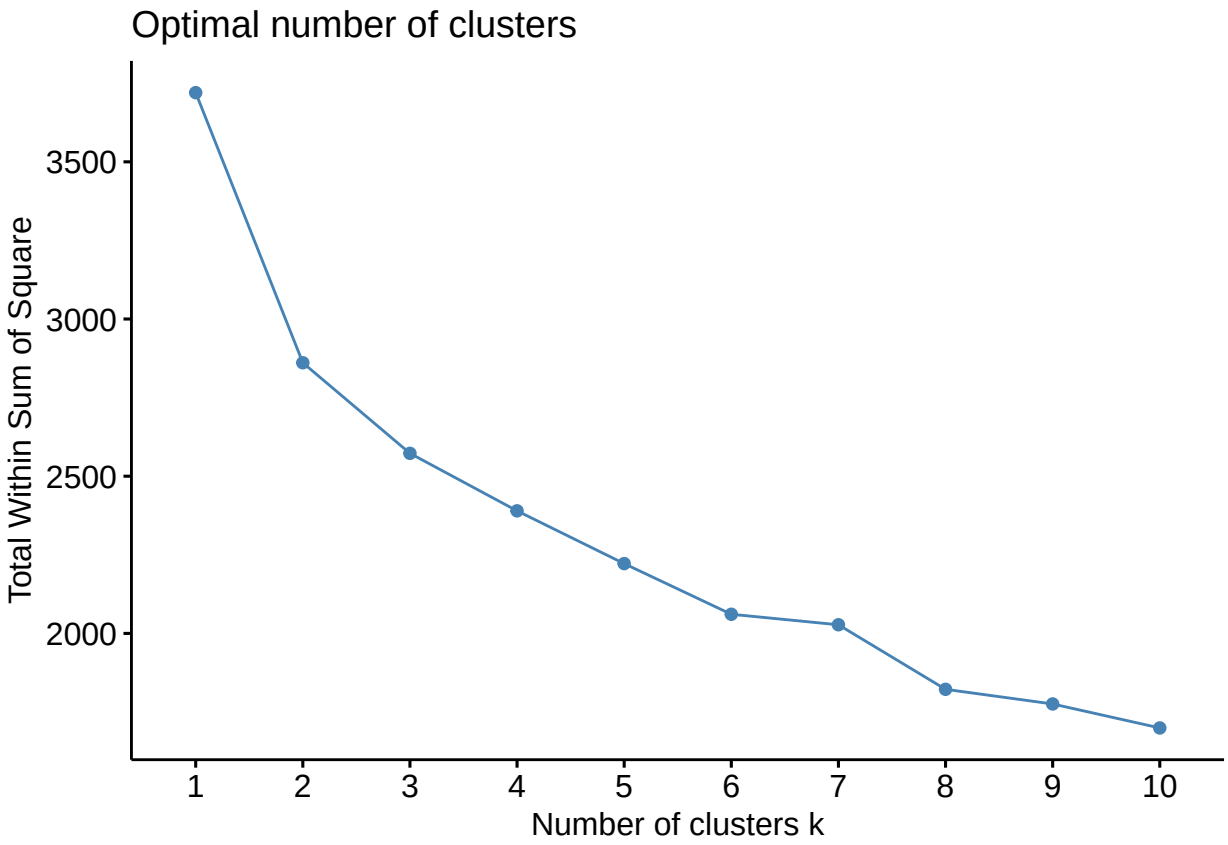
```
## 'data.frame':    187 obs. of  20 variables:
##  $ Presenze.Totali               : int  492 612 604 777 276 624 643 476 534 566 ...
##  $ Gol                           : int  307 320 277 316 144 288 266 236 181 162 ...
##  $ Assist                        : int  3 7 152 184 10 46 10 34 9 13 ...
##  $ Sostituito.In                 : int  70 4 62 168 70 155 29 112 26 2 ...
##  $ Sostituito.Out                : int  92 1 144 257 92 189 58 102 49 13 ...
##  $ Cartellino.Giallo             : int  0 0 31 57 0 38 7 49 2 0 ...
##  $ Cartellino.Rosso              : int  6 6 2 0 2 1 1 6 5 0 ...
##  $ Gol.su.Rigore                 : int  16 26 93 70 9 15 41 25 5 17 ...
##  $ Gol.per.Minuti                : num  143 171 172 170 172 149 207 141 251 313 ...
##  $ Minuti.Giocati                : num  44 54.6 47.6 53.8 24.7 ...
##  $ Presenze.in.Nazionale         : int  53 34 56 91 47 57 61 49 64 70 ...
```

```
##  $ Gol.in.Nazionale              : int  33 30 27 27 25 25 25 23 23 22 ...
##  $ Assist.in.Nazionale           : int  0 0 14 11 0 4 2 4 0 3 ...
##  $ Sostituito.In..Nazionale.     : int  7 6 11 30 7 15 17 3 7 2 ...
##  $ Sostituito.Out..Nazionale.    : int  8 4 16 43 8 25 18 26 9 10 ...
##  $ Cartellino.Giallo.in.Nazionale: int  0 0 3 5 0 0 0 4 2 0 ...
##  $ Cartellino.Rosso.in.Nazionale : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Gol.su.Rigore.in.Nazionale    : int  3 1 7 6 1 2 2 0 1 1 ...
##  $ Gol.per.Minuti.in.Nazionale   : num  147 103 152 191 170 141 168 151 211 263 ...
##  $ Minuti.Giocati.in.Nazionale   : num  4.86 3.09 4.1 5.15 4.26 ...
```

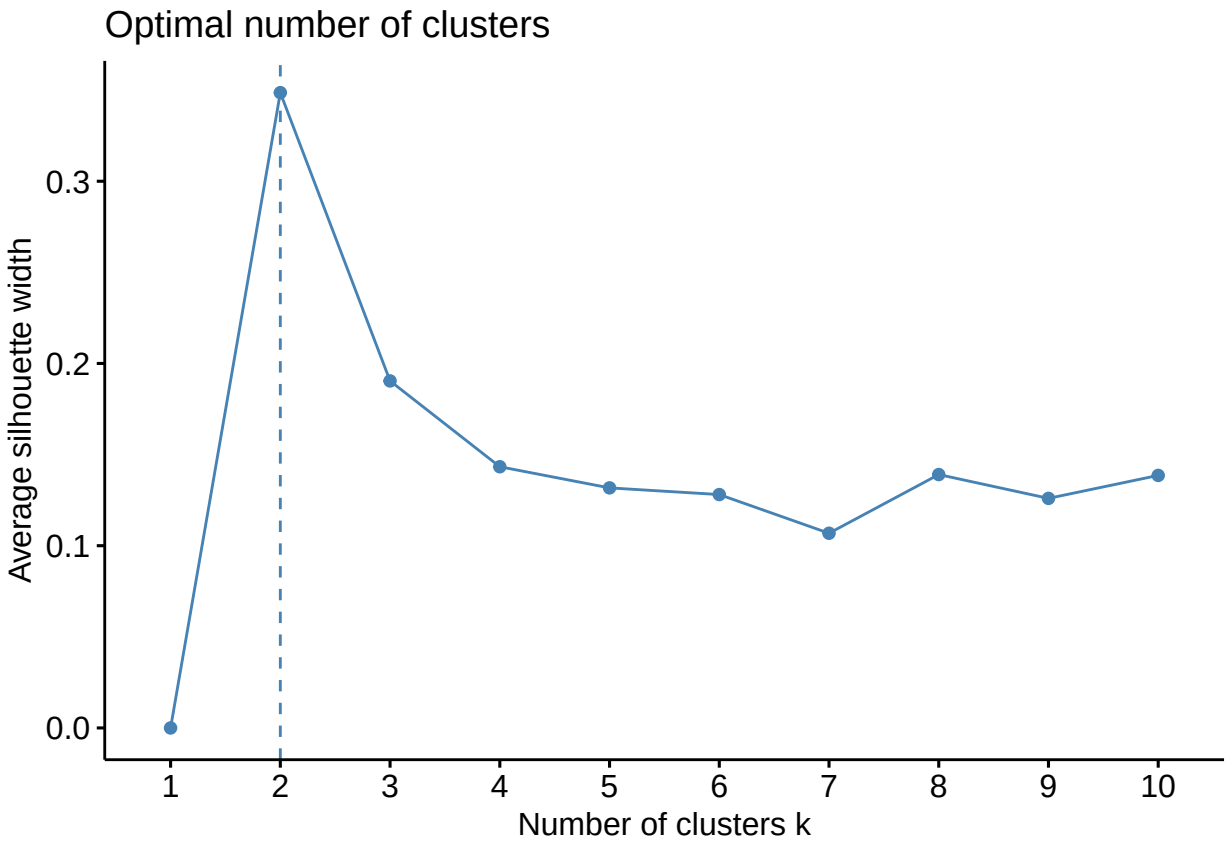
```
df_clean_numeric <- df_clean_numeric %>%
  mutate(across(everything(), as.numeric))

# Verifica nuovamente il tipo di ogni colonna
str(df_clean_numeric)
```

```
## 'data.frame':    187 obs. of  20 variables:
##  $ Presenze.Totali               : num  492 612 604 777 276 624 643 476 534 566 ...
##  $ Gol                           : num  307 320 277 316 144 288 266 236 181 162 ...
##  $ Assist                        : num  3 7 152 184 10 46 10 34 9 13 ...
##  $ Sostituito.In                 : num  70 4 62 168 70 155 29 112 26 2 ...
##  $ Sostituito.Out                : num  92 1 144 257 92 189 58 102 49 13 ...
##  $ Cartellino.Giallo             : num  0 0 31 57 0 38 7 49 2 0 ...
##  $ Cartellino.Rosso              : num  6 6 2 0 2 1 1 6 5 0 ...
##  $ Gol.su.Rigore                 : num  16 26 93 70 9 15 41 25 5 17 ...
##  $ Gol.per.Minuti                : num  143 171 172 170 172 149 207 141 251 313 ...
##  $ Minuti.Giocati                : num  44 54.6 47.6 53.8 24.7 ...
##  $ Presenze.in.Nazionale         : num  53 34 56 91 47 57 61 49 64 70 ...
##  $ Gol.in.Nazionale              : num  33 30 27 27 25 25 25 23 23 22 ...
##  $ Assist.in.Nazionale           : num  0 0 14 11 0 4 2 4 0 3 ...
##  $ Sostituito.In..Nazionale.     : num  7 6 11 30 7 15 17 3 7 2 ...
##  $ Sostituito.Out..Nazionale.    : num  8 4 16 43 8 25 18 26 9 10 ...
##  $ Cartellino.Giallo.in.Nazionale: num  0 0 3 5 0 0 0 4 2 0 ...
##  $ Cartellino.Rosso.in.Nazionale : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ Gol.su.Rigore.in.Nazionale    : num  3 1 7 6 1 2 2 0 1 1 ...
##  $ Gol.per.Minuti.in.Nazionale   : num  147 103 152 191 170 141 168 151 211 263 ...
##  $ Minuti.Giocati.in.Nazionale   : num  4.86 3.09 4.1 5.15 4.26 ...
```

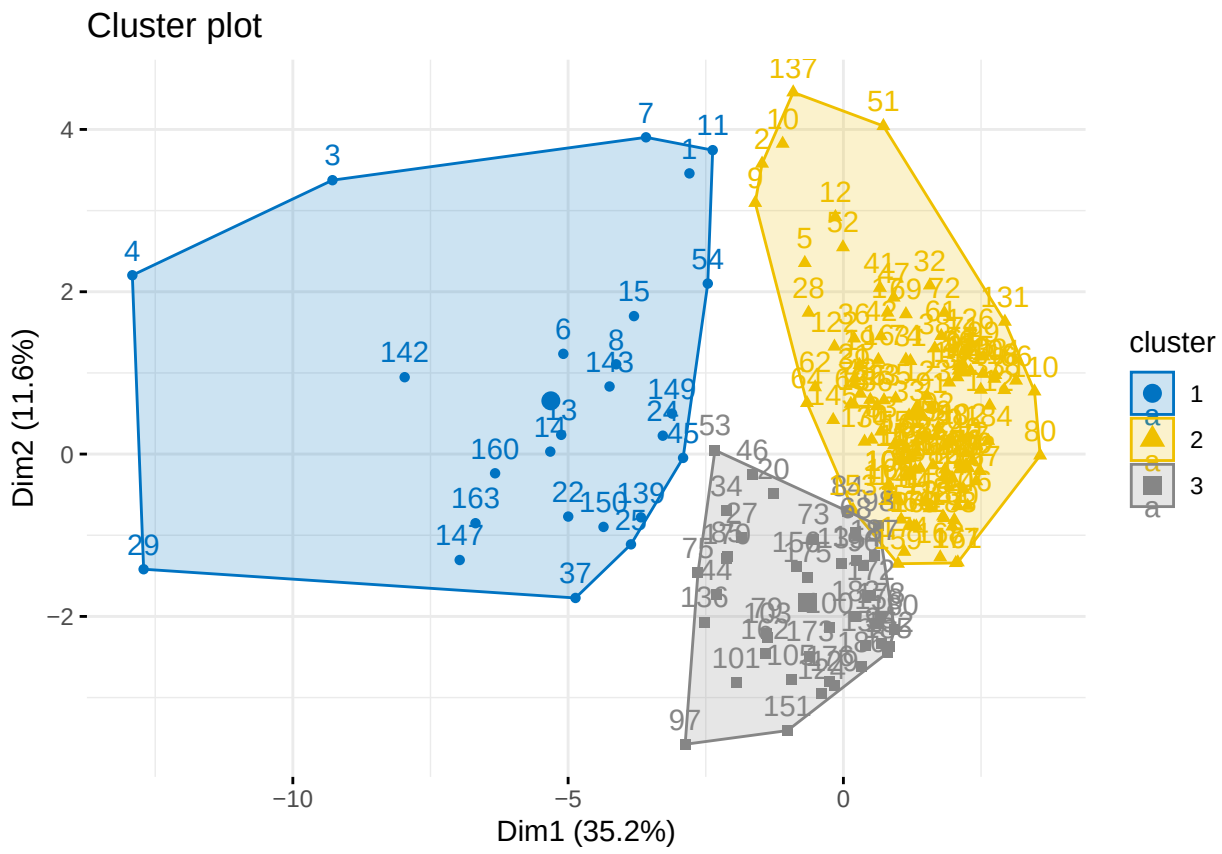
```
# Normalizzare i dati
df_clean_scaled <- scale(df_clean_numeric)

# Utilizzare il metodo Elbow per trovare il numero ottimale di cluster
fviz_nbclust(df_clean_scaled, kmeans, method = "wss")
```




```
# Utilizzare il metodo della silhouette per trovare il numero ottimale di cluster
fviz_nbclust(df_clean_scaled, kmeans, method = "silhouette")
```




```r
set.seed(123)
kmeans_result <- kmeans(df_clean_scaled, centers = 3, nstart = 25)

# Aggiungere i cluster al dataframe
df_clean$cluster <- as.factor(kmeans_result$cluster)

# Visualizzare i cluster ottenuti
fviz_cluster(kmeans_result, data = df_clean_scaled,
             ellipse.type = "convex",
             palette = "jco",
             ggtheme = theme_minimal())
```




```
# Aggiungere i cluster al dataframe originale
df$cluster <- as.factor(kmeans_result$cluster)

# Calcolare le medie delle variabili per ciascun cluster
cluster_means <- df %>%
  group_by(cluster) %>%
  summarise_all(list(mean = mean), na.rm = TRUE)
```

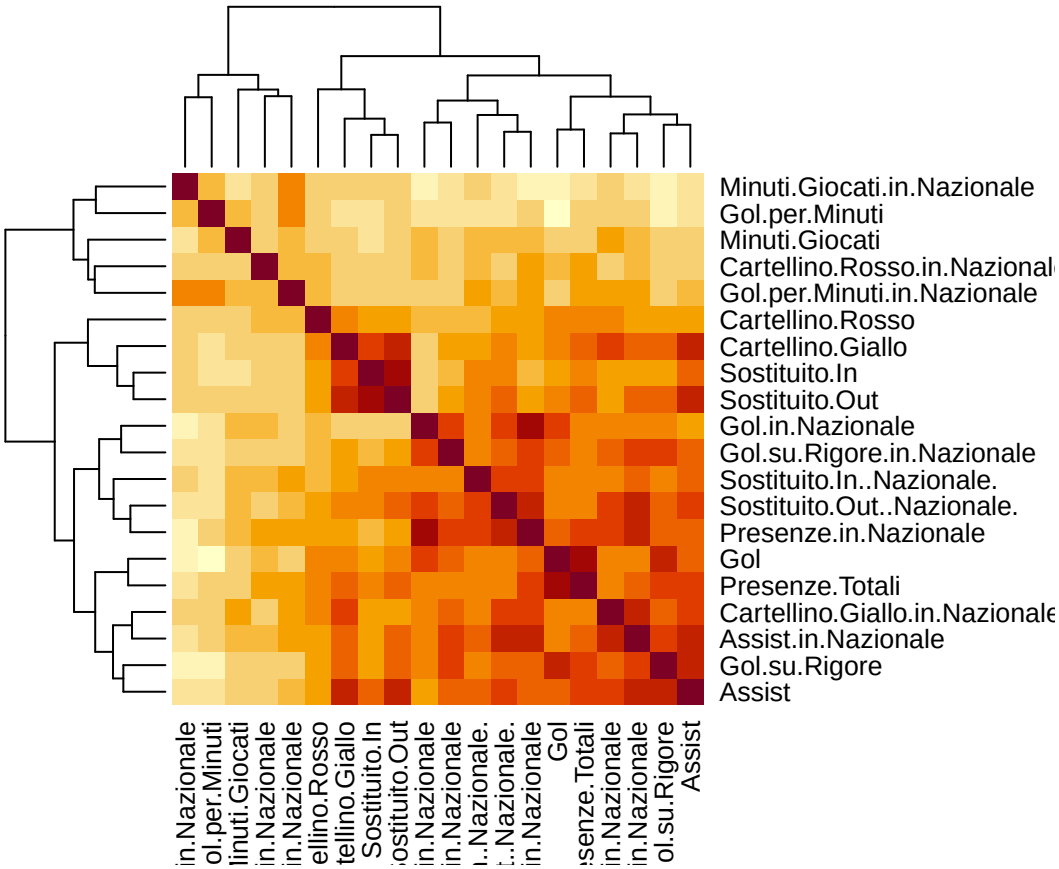
```
## Warning: There were 3 warnings in `summarise()`.
## The first warning was:
## i In argument: `Player.Name_mean = (function (x, ...) ...`.
## i In group 1: `cluster = 1`.
## Caused by warning in `mean.default()`:
## ! argument is not numeric or logical: returning NA
## i Run `dplyr::last_dplyr_warnings()` to see the 2 remaining warnings.
```

```
# Visualizzare le medie
print(cluster_means)
```

```
## # A tibble: 3 x 22
##   cluster Player.Name_mean Presenze.Totali_mean Gol_mean Assist_mean
##   <fct>              <dbl>                <dbl>    <dbl>       <dbl>
## 1 1                     NA                 597.     225.        76.1
## 2 2                     NA                 287.     100.         7.06
## 3 3                     NA                 462.     131.        42.2
```

```
## # i 17 more variables: Sostituito.In_mean <dbl>, Sostituito.Out_mean <dbl>,
## #   Cartellino.Giallo_mean <dbl>, Cartellino.Rosso_mean <dbl>,
## #   Gol.su.Rigore_mean <dbl>, Gol.per.Minuti_mean <dbl>,
## #   Minuti.Giocati_mean <dbl>, Presenze.in.Nazionale_mean <dbl>,
## #   Gol.in.Nazionale_mean <dbl>, Assist.in.Nazionale_mean <dbl>,
## #   Sostituito.In..Nazionale._mean <dbl>,
## #   Sostituito.Out..Nazionale._mean <dbl>, ...
```

```
correlation_matrix <- cor(df_clean_scaled)
heatmap(correlation_matrix, symm = TRUE)
```




```
train_set <- read.csv("C:/Users/filip/Desktop/attaccanti italiani/dataset scraped/train_set.csv")
test_set <- read.csv("C:/Users/filip/Desktop/attaccanti italiani/dataset scraped/test_set.csv")
```

```
train_set$Gol.per.Minuti <- as.numeric(gsub("'", "", train_set$Gol.per.Minuti))
train_set$Gol.per.Minuti.in.Nazionale <- as.numeric(gsub("'", "", train_set$Gol.per.Minuti.in.Nazionale))
train_set$Minuti.Giocati.in.Nazionale <- as.numeric(gsub("'", "", train_set$Minuti.Giocati.in.Nazionale))
train_set$Minuti.Giocati <- as.numeric(gsub("'", "", train_set$Minuti.Giocati))

test_set$Gol.per.Minuti <- as.numeric(gsub("'", "", test_set$Gol.per.Minuti))
test_set$Gol.per.Minuti.in.Nazionale <- as.numeric(gsub("'", "", test_set$Gol.per.Minuti.in.Nazionale))
test_set$Minuti.Giocati.in.Nazionale <- as.numeric(gsub("'", "", test_set$Minuti.Giocati.in.Nazionale))
test_set$Minuti.Giocati <- as.numeric(gsub("'", "", test_set$Minuti.Giocati))

str(train_set)
```

```
## 'data.frame':    140 obs. of  21 variables:
##  $ Player.Name                   : chr  "Giuseppe Meazza" "Silvio Piola" "Roberto Baggio" "Alessandr
##  $ Presenze.Totali               : int  492 612 604 777 276 624 643 476 534 566 ...
##  $ Gol                           : int  307 320 277 316 144 288 266 236 181 162 ...
##  $ Assist                        : int  3 7 152 184 10 46 10 34 9 13 ...
##  $ Sostituito.In                 : int  70 4 62 168 70 155 29 112 26 2 ...
##  $ Sostituito.Out                : int  92 1 144 257 92 189 58 102 49 13 ...
##  $ Cartellino.Giallo             : int  0 0 31 57 0 38 7 49 2 0 ...
##  $ Cartellino.Rosso              : int  6 6 2 0 2 1 1 6 5 0 ...
##  $ Gol.su.Rigore                 : int  16 26 93 70 9 15 41 25 5 17 ...
##  $ Gol.per.Minuti                : num  143 171 172 170 172 149 207 141 251 313 ...
##  $ Minuti.Giocati                : num  44 54.6 47.6 53.8 24.7 ...
##  $ Presenze.in.Nazionale         : int  53 34 56 91 47 57 61 49 64 70 ...
##  $ Gol.in.Nazionale              : int  33 30 27 27 25 25 25 23 23 22 ...
##  $ Assist.in.Nazionale           : int  0 0 14 11 0 4 2 4 0 3 ...
##  $ Sostituito.In..Nazionale.     : int  7 6 11 30 7 15 17 3 7 2 ...
##  $ Sostituito.Out..Nazionale.    : int  8 4 16 43 8 25 18 26 9 10 ...
##  $ Cartellino.Giallo.in.Nazionale: int  0 0 3 5 0 0 0 4 2 0 ...
##  $ Cartellino.Rosso.in.Nazionale : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Gol.su.Rigore.in.Nazionale    : int  3 1 7 6 1 2 2 0 1 1 ...
##  $ Gol.per.Minuti.in.Nazionale   : num  147 103 152 191 170 141 168 151 211 263 ...
##  $ Minuti.Giocati.in.Nazionale   : num  4.86 3.09 4.1 5.15 4.26 ...
```

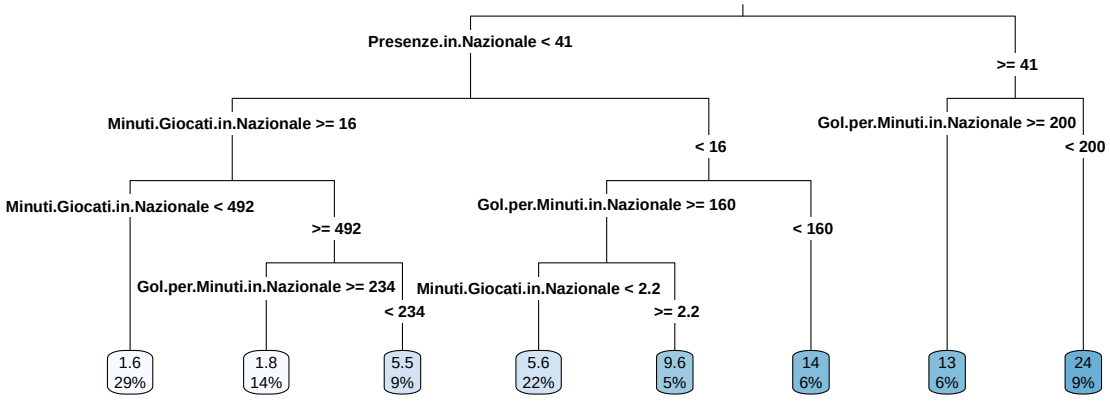
Spiegazione del Clustering

Nella fase di clustering, abbiamo raggruppato i giocatori in base alle loro caratteristiche prestazionali per identificare gruppi omogenei. Dopo aver pulito e normalizzato i dati, abbiamo determinato il numero ottimale di cluster utilizzando i metodi Elbow e della silhouette. Il metodo Elbow ha aiutato a identificare il punto in cui l'aggiunta di ulteriori cluster non riduceva significativamente la varianza interna ai cluster, mentre il metodo della silhouette ha valutato quanto bene ogni punto si adattava al proprio cluster rispetto agli altri. Abbiamo quindi eseguito il clustering K-means con il numero ottimale di cluster (3 in questo caso), partizionando i dati in gruppi con caratteristiche simili. I cluster risultanti sono stati visualizzati per mostrare la distribuzione dei giocatori, e sono state calcolate le medie delle variabili per ciascun cluster per comprenderne le caratteristiche distintive. Infine, abbiamo creato una heatmap della matrice di correlazione per visualizzare le relazioni tra le variabili normalizzate, identificando variabili fortemente correlate e comprendendo meglio la struttura dei dati. Questi risultati ci permettono di segmentare i giocatori in gruppi omogenei, facilitando l'analisi comparativa delle prestazioni.

CART

```r
train_player_names <- train_set$Player.Name
test_player_names <- test_set$Player.Name

# Assicurarsi che la variabile target sia numerica
train_set$Gol.in.Nazionale <- as.numeric(train_set$Gol.in.Nazionale)
test_set$Gol.in.Nazionale <- as.numeric(test_set$Gol.in.Nazionale)

# Rimuovere solo la colonna `Player.Name` dai dataset
train_set <- subset(train_set, select = -c(Player.Name))
test_set <- subset(test_set, select = -c(Player.Name))
```

```r
# Rimuovere righe con valori mancanti
train_set <- na.omit(train_set)
test_set <- na.omit(test_set)

# Creare il modello CART
cart_model <- rpart(Gol.in.Nazionale ~ ., data = train_set, method = "anova")

# Visualizzare l'albero di decisione
rpart.plot(cart_model, type = 3, digits = 2)
```




```r
# Fare previsioni sul test set
cart_predictions <- predict(cart_model, newdata = test_set)

# Arrotondare le predizioni ai valori interi più vicini
cart_predictions <- round(cart_predictions)

# Calcolare il Mean Absolute Error (MAE)
mae <- mean(abs(cart_predictions - test_set$Gol.in.Nazionale))
print(paste("Mean Absolute Error:", round(mae, 2)))
```

```
## [1] "Mean Absolute Error: 4.91"
```

```r
# Calcolare il Mean Squared Error (MSE)
mse <- mean((cart_predictions - test_set$Gol.in.Nazionale)^2)
print(paste("Mean Squared Error:", round(mse, 2)))
```

```
## [1] "Mean Squared Error: 53"
```

```
# Calcolare il R-squared
r_squared <- 1 - sum((cart_predictions - test_set$Gol.in.Nazionale)^2) / sum((mean(train_set$Gol.in.Naz
print(paste("R-squared:", round(r_squared, 2)))
```

```
## [1] "R-squared: -0.69"
```

```
# Aggiungere le predizioni e i nomi dei giocatori al test set
test_set$Predicted_Goals <- cart_predictions
test_set$Player.Name <- test_player_names

# Calcolare la differenza tra i gol effettivamente fatti e i gol predetti
test_set$Difference <- test_set$Gol.in.Nazionale - test_set$Predicted_Goals

# Suddividere i giocatori in overperforming e underperforming
test_set$Performance <- ifelse(test_set$Difference > 0, "Overperforming", "Underperforming")

# Visualizzare i risultati
head(test_set)
```

```
##   Presenze.Totali Gol Assist Sostituito.In Sostituito.Out Cartellino.Giallo
## 1             244  82     15            92             84                24
## 2             593 329     74           100            170                86
## 3             483 172     42           116            112                43
## 4             284  90     11           146             93                35
## 5             258  73     25            88             83                17
## 6             164  55     16            55             57                26
##   Cartellino.Rosso Gol.su.Rigore Gol.per.Minuti Minuti.Giocati
## 1                1             4            171         14.033
## 2                3            75            131         43.212
## 3                0            30            196         33.675
## 4                3             2            144         12.944
## 5                0             6            212         15.470
## 6                3             5            179          9.848
##   Presenze.in.Nazionale Gol.in.Nazionale Assist.in.Nazionale
## 1                    15                1                   0
## 2                    57               17                   8
## 3                    44               12                   7
## 4                    15                4                   2
## 5                    27                6                   4
## 6                     0                0                   0
##   Sostituito.In..Nazionale. Sostituito.Out..Nazionale.
## 1                        10                          5
## 2                         9                         30
## 3                        22                         15
## 4                        10                          3
## 5                        11                         13
## 6                         0                          0
##   Cartellino.Giallo.in.Nazionale Cartellino.Rosso.in.Nazionale
## 1                              0                             0
## 2                              6                             0
## 3                              4                             0
```

```
## 4                             1                             0
## 5                             0                             0
## 6                             0                             0
##   Gol.su.Rigore.in.Nazionale Gol.per.Minuti.in.Nazionale
## 1                          0                         570
## 2                          2                         232
## 3                          1                         177
## 4                          0                         155
## 5                          0                         224
## 6                          0                           0
##   Minuti.Giocati.in.Nazionale Predicted_Goals       Player.Name Difference
## 1                     570.000               2 Gianluca Scamacca         -1
## 2                       3.947              13     Ciro Immobile          4
## 3                       2.128              24    Andrea Belotti        -12
## 4                     620.000               6        Moise Kean         -2
## 5                       1.344               6 Giacomo Raspadori          0
## 6                       0.000              14     Lorenzo Lucca        -14
##       Performance
## 1 Underperforming
## 2  Overperforming
## 3 Underperforming
## 4 Underperforming
## 5 Underperforming
## 6 Underperforming
```

```
# Visualizzare solo i giocatori overperforming
overperforming_players <- test_set[test_set$Performance == "Overperforming", ]
print("Overperforming Players:")
```

```
## [1] "Overperforming Players:"
```

```
for (player in overperforming_players$Player.Name) {
  print(paste("Overperforming:", player))
}
```

```
## [1] "Overperforming: Ciro Immobile"
## [1] "Overperforming: Mario Balotelli"
## [1] "Overperforming: Domenico Berardi"
## [1] "Overperforming: Lorenzo Colombo"
## [1] "Overperforming: Stephan El Shaarawy"
```

Risultati CART

Per prevedere il numero di gol segnati in nazionale dai giocatori, abbiamo utilizzato un modello di albero decisionale (CART - Classification and Regression Tree). Dopo aver caricato e pulito i dataset di allenamento e di test, abbiamo costruito il modello CART utilizzando il dataset di allenamento. Il modello è stato addestrato a partire dalle variabili disponibili, escludendo il nome del giocatore, che è stato conservato separatamente per l'identificazione.

Una volta addestrato il modello, abbiamo fatto previsioni sui dati del test set e arrotondato le predizioni ai valori interi più vicini, poiché i gol sono una variabile discreta. Abbiamo calcolato le metriche di valutazione del modello, ottenendo un Mean Absolute Error (MAE) di 4.91, un Mean Squared Error (MSE) di 53 e un

R-squared di -0.69. Questi risultati indicano che il modello ha prestazioni subottimali e una capacità molto limitata di spiegare la varianza nei dati, suggerendo la necessità di miglioramenti o di considerare modelli alternativi.

Random Forest

```
# Identificare le variabili character
character_vars_train <- sapply(train_set, is.character)
character_vars_test <- sapply(test_set, is.character)

# Rimuovere le variabili character dai dataset
train_set <- train_set[, !character_vars_train]
test_set <- test_set[, !character_vars_test]

train_set$Gol.in.Nazionale <- as.numeric(train_set$Gol.in.Nazionale)
test_set$Gol.in.Nazionale <- as.numeric(test_set$Gol.in.Nazionale)

# Rimuovere righe con valori mancanti
train_set <- na.omit(train_set)
test_set <- na.omit(test_set)

# Creare il modello Random Forest per la regressione
rf_model <- randomForest(Gol.in.Nazionale ~ ., data = train_set, ntree = 100)

# Fare previsioni sul test set
rf_predictions <- predict(rf_model, newdata = test_set)

rf_predictions <- round(rf_predictions)

# Calcolare il Mean Absolute Error (MAE)
mae <- mean(abs(rf_predictions - test_set$Gol.in.Nazionale))
print(paste("Mean Absolute Error:", round(mae, 2)))
```

```
## [1] "Mean Absolute Error: 3.36"
```

```
# Calcolare il Mean Squared Error (MSE)
mse <- mean((rf_predictions - test_set$Gol.in.Nazionale)^2)
print(paste("Mean Squared Error:", round(mse, 2)))
```

```
## [1] "Mean Squared Error: 18.64"
```

```
# Calcolare il R-squared
r_squared <- 1 - sum((rf_predictions - test_set$Gol.in.Nazionale)^2) / sum((mean(train_set$Gol.in.Nazio
print(paste("R-squared:", round(r_squared, 2)))
```

```
## [1] "R-squared: 0.41"
```

```r
# Aggiungere le predizioni e i nomi dei giocatori al test set
test_set$Predicted_Goals <- rf_predictions
test_player_names <- test_set$Player.Name

# Calcolare la differenza tra i gol effettivamente fatti e i gol predetti
test_set$Difference <- test_set$Gol.in.Nazionale - test_set$Predicted_Goals

# Suddividere i giocatori in overperforming e underperforming
test_set$Performance <- ifelse(test_set$Difference > 0, "Overperforming", "Underperforming")
```

Risultati Random Foresest

I risultati dell'analisi ci hanno permesso di identificare chiaramente quali giocatori hanno superato le aspettative (overperforming) e quali hanno reso meno rispetto a quanto previsto (underperforming). Questa classificazione può fornire spunti utili per gli allenatori e gli analisti per comprendere meglio le prestazioni dei giocatori in nazionale.

In particolare i giocatori che hanno superato le aspettative sono; - Davide Frattesi - Matteo Pessina - Pietro Iemmello

Le differenze tra i giocatori classificati come overperforming e underperforming nei modelli CART e Random Forest possono essere attribuite alle caratteristiche distintive di questi algoritmi. Il modello CART, che utilizza un singolo albero decisionale, è più sensibile al rumore nei dati e può sovra-adattarsi alle specifiche del dataset di addestramento, portando a previsioni meno stabili. Al contrario, Random Forest, costruendo molti alberi decisionali e aggregando i loro risultati, tende a ridurre l'overfitting e a fornire previsioni più stabili e robuste. Questa aggregazione permette a Random Forest di essere meno influenzato dalle peculiarità del dataset di addestramento, risultando in una classificazione dei giocatori che potrebbe differire significativamente da quella ottenuta con un singolo albero CART. Pertanto, le differenze nei risultati tra i due modelli sono una manifestazione delle loro diverse capacità di generalizzare dai dati di addestramento ai dati di test