ClusteringDefenders

LZ

26 4 2023

Clustering Defenders

The goal of this project is to profile defenders of the English Championship through clustering. We will perform clustering on player stats of Season 22/23 scraped from FBref (https://fbref.com/en/comps/10/stats/Championship-Stats) with the `worldfootballR` package (https://github.com/JaseZiv/worldfootballR). The data has already been downloaded and saved, see the Markdown document named 'Download Player Stats' for more information (also on the metrics).

We load the tibble with the player stats:

```
library(tidyverse)
load(paste0(getwd(), "/data/Championship22_23_playerStats.RData"))
NL22_all
```

```
## # A tibble: 733 x 110
##       Rk Player Nation Pos   Squad Age    Born `MP_Playing Ti~` `Starts_Playin~`
##    <int> <chr>  <chr>  <chr> <chr> <chr> <int>            <int>            <int>
##  1     1 Max A~ eng E~ DF    Norw~ 23-1~  2000               43               42
##  2     2 Thelo~ no NOR FW,MF Wiga~ 20-3~  2002               39               11
##  3     3 Nelso~ eng E~ DF    Read~ 19-2~  2003                3                0
##  4     4 Kelvi~ <NA>   DF,MF Read~ <NA>     NA                7                0
##  5     5 Finla~ eng E~ FW,MF Pres~ 18-1~  2005                4                0
##  6     6 Elija~ eng E~ FW    Luto~ 25-1~  1998               41               38
##  7     7 Toby ~ eng E~ MF,FW Watf~ 18-0~  2005                4                0
##  8     8 Alber~ gh GHA FW,MF QPR   35-1~  1987               36               10
##  9     9 Micha~ eng E~ FW    Watf~ 17-2~  2005                1                0
## 10    10 Benik~ cd COD FW    Mill~ 30-0~  1993               19                8
## # ... with 723 more rows, and 101 more variables: `Min_Playing Time` <dbl>,
## #   `Mins_Per_90_Playing Time` <dbl>, Gls <int>, Ast <int>, `G+A` <int>,
## #   G_minus_PK <int>, PK <int>, PKatt <int>, xG_Expected <dbl>,
## #   npxG_Expected <dbl>, xAG_Expected <dbl>, `npxG+xAG_Expected` <dbl>,
## #   PrgC_Progression <int>, PrgP_Progression <int>, PrgR_Progression <int>,
## #   `Gls_Per 90 Minutes` <dbl>, `Ast_Per 90 Minutes` <dbl>,
## #   `G+A_Per 90 Minutes` <dbl>, `G_minus_PK_Per 90 Minutes` <dbl>, ...
```

We filter all players that have a defending position:

```
NL22_all$Pos %>% table()
```

```
## .
##    DF DF,FW DF,MF    FW FW,DF FW,MF    GK    MF MF,DF MF,FW
##   201    32    29   109    16    80    52   134    16    64
```

```
NL22_defenders <- NL22_all %>% filter(Pos %in% c("DF", "DF,FW", "DF,MF"))
dim(NL22_defenders)
```

```
## [1] 262 110
```

Some players have changed teams during the season, so we sum up the total play time for them.

```
NL22_defenders <- NL22_defenders %>%  group_by(Player) %>%
  mutate(MinsPer90_total = sum(Mins_Per_90, na.rm = TRUE)) %>%
  ungroup()
```

Now we can filter those defenders with at least 180 minutes played in total during the season 22/23.

```
NL22_defenders <- NL22_defenders %>% filter(MinsPer90_total >= 2)
```

For those who changed teams, we compute the weighted mean for each metric, where the weights are chosen as minutes played within teams, divided by minutes played in total. These are the players that appear multiple times in the data, with a selection of variables.

```
NL22_defenders %>%
  filter(Player %in% NL22_defenders[duplicated(NL22_defenders$Player),  ]$Player) %>%
  select(Player, Squad, Mins_Per_90, Tkl_Tackles, TklW_Tackles)
```

```
## # A tibble: 12 x 5
##    Player                Squad          Mins_Per_90 Tkl_Tackles TklW_Tackles
##    <chr>                 <chr>                <dbl>       <dbl>        <dbl>
##  1 Rarmani Edmonds-Green Wigan Athletic         2.7       2.59         1.48
##  2 Rarmani Edmonds-Green Huddersfield           8.5       1.88         1.18
##  3 Dominic Hyam          Blackburn             34.1       1.44         0.762
##  4 Dominic Hyam          Coventry City          2         2            1
##  5 Martin Kelly          West Brom              4.1       0.976        0.976
##  6 Martin Kelly          Wigan Athletic         0.8       0            0
##  7 Luke McNally          Burnley                0       NaN          NaN
##  8 Luke McNally          Coventry City         17         2.47         1.47
##  9 Curtis Nelson         Cardiff City           5.3       1.13         0.377
## 10 Curtis Nelson         Blackpool             15         1            0.533
## 11 Bailey Wright         Rotherham Utd          4.5       0.667        0.444
## 12 Bailey Wright         Sunderland             7.6       1.18         0.526
```

Now ´the new value for Rarmani Edmonds-Greens `Tkl_Tackles` statistics should be

$$(2.5926 \cdot 2.7 + 8.5 \cdot 1.8824)/(2.7 + 8.5).$$

The following function does that.

```
weighted_avg_duplicates <- function(df) {

  duplicated_ids <-  df[duplicated(df$Player),  ]$Player

  df_duplicates <- df %>% filter(Player %in% duplicated_ids)

  df_not_duplicated <- df %>% filter(!(Player %in% duplicated_ids))

  df_nest <- df_duplicates %>% group_by(Player, Born, Nation) %>% nest()

  weighted_avg <- function(weights, values) {
    sum(weights*values, na.rm = TRUE)/sum(weights, na.rm = TRUE)
  }
  paste_chr_info <- function(vec) {

    if(vec[1] == vec[2]) {
```

```
    return(vec[1])
  }
  else {
   paste(vec, collapse = ",")
  }
}

df_duplicates <- df_nest %>%
  mutate(data = purrr::map(data, ~ {

  mins.tmp <- .x$Mins_Per_90
  x_no_avg <- .x %>% select(Rk, Pos, Squad, Age)

  x_no_avg <- x_no_avg %>% summarise(across(where(is.character), paste_chr_info))
  summarise_vars <- colnames(.x)
  summarise_vars <- summarise_vars[!(summarise_vars %in% c("Rk", "Pos","Squad", "Age"))]

  xnew <- .x %>% summarise_at(summarise_vars,
                      ~ weighted_avg(weights = mins.tmp, values = .x))

  xnew <-  x_no_avg %>% bind_cols(xnew)
  xnew$Rk <- .x$Rk[1]
  xnew
  }
  )) %>%
  ungroup() %>%
  unnest(cols = data)

df_not_duplicated %>% bind_rows(df_duplicates)

}
```

```
NL22_defenders <- weighted_avg_duplicates(NL22_defenders)
```

Choosing relevant defending metrics

Not all available metrics are important for measuring the quality of a defender. We therefore define several metrics that we consider to be of central importance for defenders. First of all, we add some further metrics (Cards Per 90, Fouls per Tackle, Tackles Won).

```
NL22_defenders  <- NL22_defenders %>%
  mutate(CardsPer90 = CrdY + CrdR, FoulsPerTackle = Fls/Tkl_Tackles,
         TacklesWonPercent = TklW_Tackles/Tkl_Tackles)
```

We have metrics of central importance, then some additional ones, and some measuring player aggressiveness:

```
metrics_central <- c("Blocks_Blocks", "Sh_Blocks", "Clr", "Int",
                     "Won_Aerial Duels", "TacklesWonPercent",
                   "Tkl_percent_Challenges", "Cmp_percent_Total",
                   "Lost_Challenges", "Err", "Mins_Per_90")

metrics_add <- c("Carries_Carries", "PrgC_Carries", "Mis_Carries",
                 "Dis_Carries", "PrgDist_Carries",
                 "CrsPA", "Cmp_percent_Long", "Cmp_percent_Short",
                 "Cmp_percent_Medium", "PKcon", "OG")
```

```
metrics_agg <- c("Fls",  "Tkl_Tackles", "CardsPer90", "FoulsPerTackle")

metrics <- c(metrics_central, metrics_add, metrics_agg)
```

Cluster analysis

Now we are ready for the cluster analysis. Our new tibble of centre backs only consists of the chosen metrics and the Player's name and minutes played:

```
cbs_ana <- NL22_defenders[ , c("Player", "Mins_Per_90", metrics)]
cbs_ana
```

```
## # A tibble: 226 x 28
##    Player       Mins_Per_90 Blocks_Blocks Sh_Blocks   Clr   Int `Won_Aerial Du~`
##    <chr>              <dbl>         <dbl>     <dbl> <dbl> <dbl>            <dbl>
##  1 Max Aarons          40.8         0.833     0.392  1.79 1.15             0.441
##  2 Anel Ahmedh~        29.8         0.940     0.403  3.19 1.85             2.05
##  3 Semi Ajayi          13.2         0.833     0.379  2.95 1.36             3.11
##  4 Ajibola Ale~        15.6         1.47      0.513  3.78 1.86             2.18
##  5 Ryan Andrews         3.4         0.588     0      2.35 2.35             2.06
##  6 Robert Atki~        23.8         0.924     0.462  3.91 1.51             5.46
##  7 Daniel Ayala        22           1.09      0.727  4.73 1.64             3.09
##  8 George Bald~        29.4         1.19      0.272  1.60 0.782            1.12
##  9 Daniel Ball~        17.9         1.28      0.838  5.03 1.62             3.69
## 10 Leon Balogun        13.7         1.17      0.511  5.26 2.12             3.80
## # ... with 216 more rows, and 21 more variables: TacklesWonPercent <dbl>,
## #   Tkl_percent_Challenges <dbl>, Cmp_percent_Total <dbl>,
## #   Lost_Challenges <dbl>, Err <dbl>, Mins_Per_90 <dbl>, Carries_Carries <dbl>,
## #   PrgC_Carries <dbl>, Mis_Carries <dbl>, Dis_Carries <dbl>,
## #   PrgDist_Carries <dbl>, CrsPA <dbl>, Cmp_percent_Long <dbl>,
## #   Cmp_percent_Short <dbl>, Cmp_percent_Medium <dbl>, PKcon <dbl>, OG <dbl>,
## #   Fls <dbl>, Tkl_Tackles <dbl>, CardsPer90 <dbl>, FoulsPerTackle <dbl>
```

```
cbs_ana <- cbs_ana[complete.cases(cbs_ana), ]
```

Since these are still many metrics, we perform a cluster analysis in order to reduce dimension.

```
pca <- prcomp(cbs_ana[ , metrics], scale. = TRUE)
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4    PC5     PC6     PC7
## Standard deviation     2.5079 1.9958 1.56837 1.26772 1.1827 1.12630 1.07669
## Proportion of Variance 0.2419 0.1532 0.09461 0.06181 0.0538 0.04879 0.04459
## Cumulative Proportion  0.2419 0.3951 0.48972 0.55153 0.6053 0.65412 0.69871
##                            PC8     PC9    PC10   PC11    PC12   PC13    PC14
## Standard deviation     1.01354 0.95571 0.91143 0.8470 0.83281 0.7750 0.75440
## Proportion of Variance 0.03951 0.03513 0.03195 0.0276 0.02668 0.0231 0.02189
## Cumulative Proportion  0.73822 0.77335 0.80530 0.8329 0.85957 0.8827 0.90456
##                           PC15    PC16    PC17   PC18    PC19    PC20    PC21
## Standard deviation     0.69746 0.66531 0.54640 0.5073 0.46437 0.43889 0.41464
## Proportion of Variance 0.01871 0.01702 0.01148 0.0099 0.00829 0.00741 0.00661
## Cumulative Proportion  0.92327 0.94030 0.95178 0.9617 0.96997 0.97738 0.98399
##                           PC22    PC23    PC24    PC25    PC26
## Standard deviation     0.37063 0.33173 0.27078 0.25072 0.18049
```

```
## Proportion of Variance 0.00528 0.00423 0.00282 0.00242 0.00125
## Cumulative Proportion  0.98928 0.99351 0.99633 0.99875 1.00000
```

We see that the first 10 Principal Components explain approx 80% of variance.

```
rownames(pca$x) <- cbs_ana$Player
n_pcs <- 10
df_pca <- as_tibble(reshape2::melt(pca$x[, 1:n_pcs]))
df_pca <- df_pca %>% rename("Player" = Var1) %>% left_join(cbs_ana %>% select(Player))
```

```
## Joining, by = "Player"
```

```
df_pca
```

```
## # A tibble: 2,250 x 3
##    Player           Var2   value
##    <chr>            <fct>  <dbl>
##  1 Max Aarons       PC1    1.68
##  2 Anel Ahmedhodzic PC1    0.610
##  3 Semi Ajayi       PC1   -2.40
##  4 Ajibola Alese    PC1   -0.259
##  5 Ryan Andrews     PC1    1.31
##  6 Robert Atkinson  PC1   -1.14
##  7 Daniel Ayala     PC1   -3.89
##  8 George Baldock   PC1    2.11
##  9 Daniel Ballard   PC1   -2.45
## 10 Leon Balogun     PC1   -1.55
## # ... with 2,240 more rows
```

hierarchical clustering

We use Ward's method (other methods yielded way worse separations).

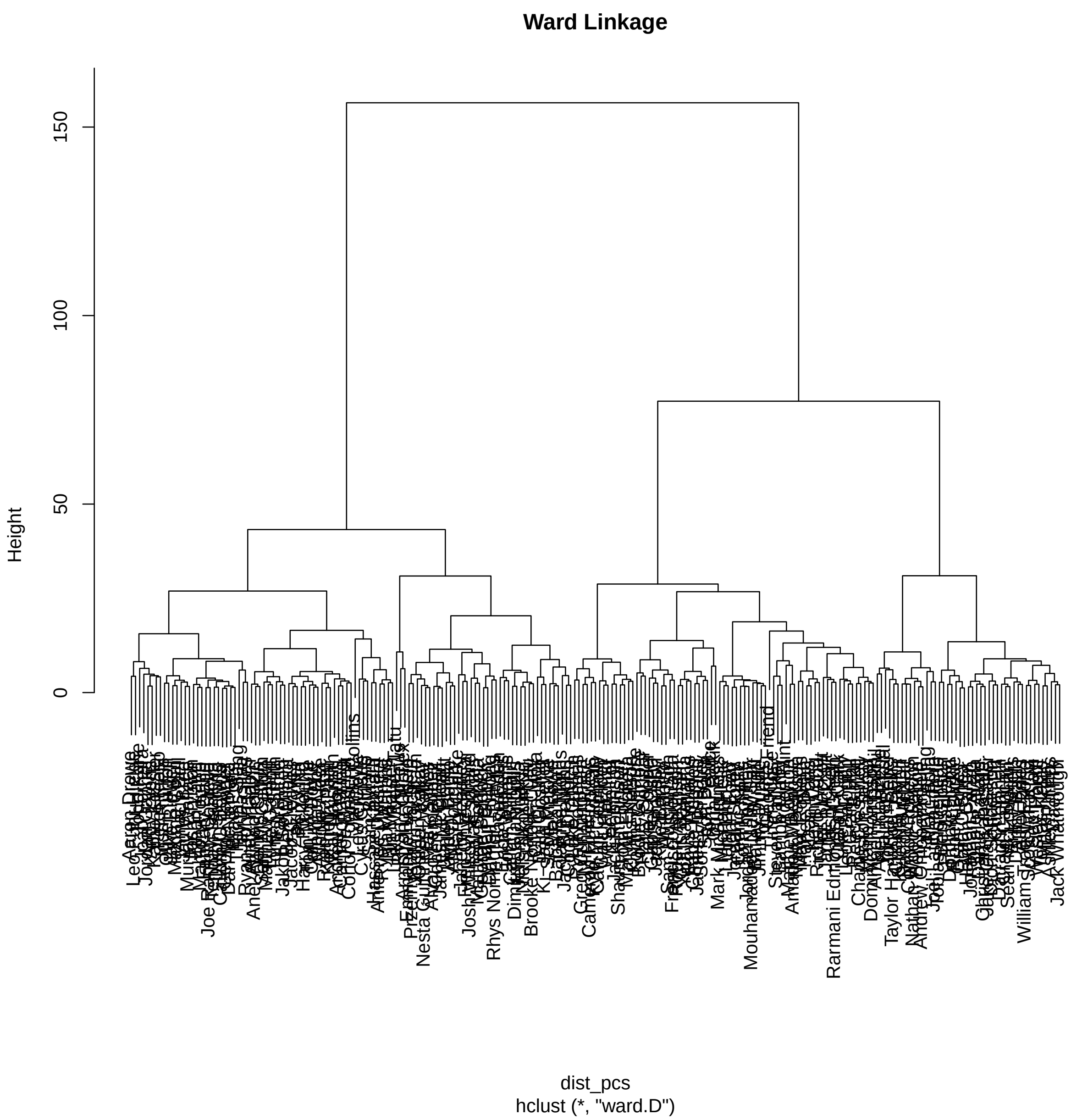
```
#dist_pcs <- dist(pca$x[ , 1:8], method = "max")

dist_pcs <- dist(pca$x[ , 1:n_pcs])

clust_pcs <- hclust(dist_pcs, method = "ward.D")
```

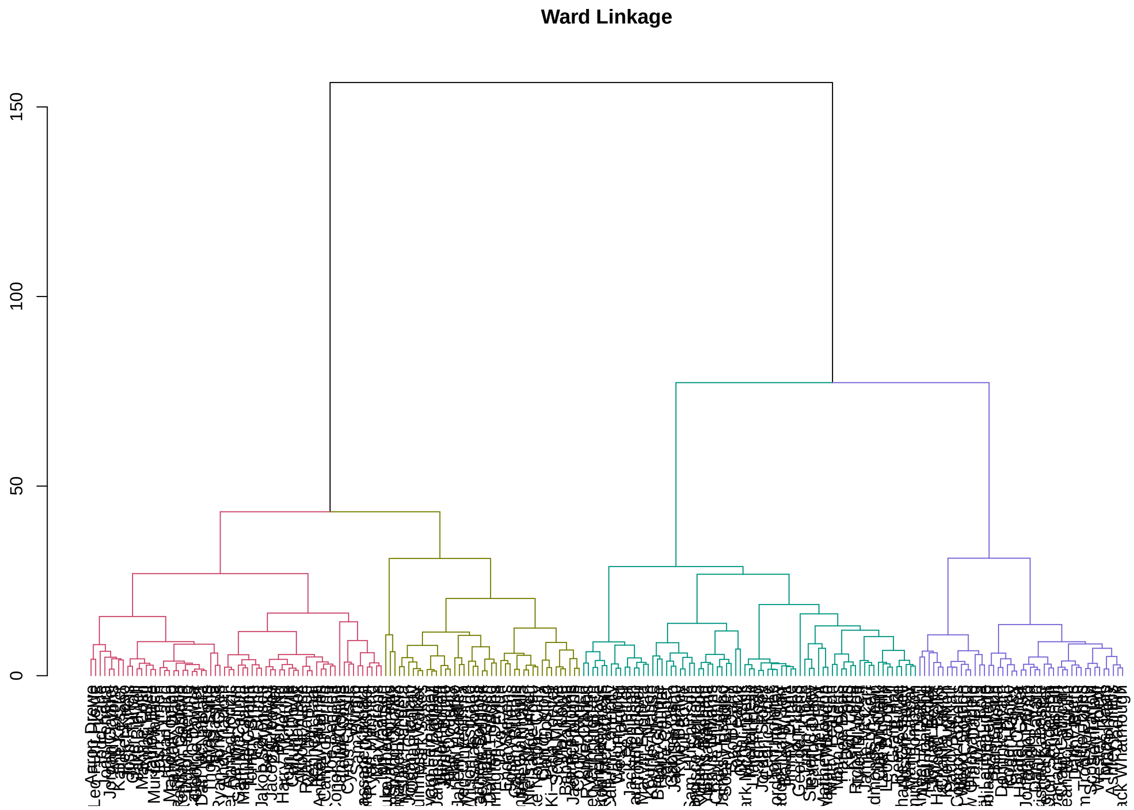
Now we plot the dendrogram for a first time.

```
plot(clust_pcs, main = "Ward Linkage")
```



It seems reasonable to cut the dendrogram such that we obtain 4 clusters.

```
cut_cl <- cutree(clust_pcs, k = 4)
clust_pcs <- as.dendrogram(clust_pcs)
dend_cl <- dendextend::color_branches(clust_pcs, k = 4)
plot(dend_cl, main = "Ward Linkage")
```



We count the number of players per cluster:

```
table(cut_cl)
```

```
## cut_cl
##  1  2  3  4
## 64 45 43 73
```

```
tib_clusters <- tibble(cluster = cut_cl) %>% mutate(Player = names(cut_cl))
```

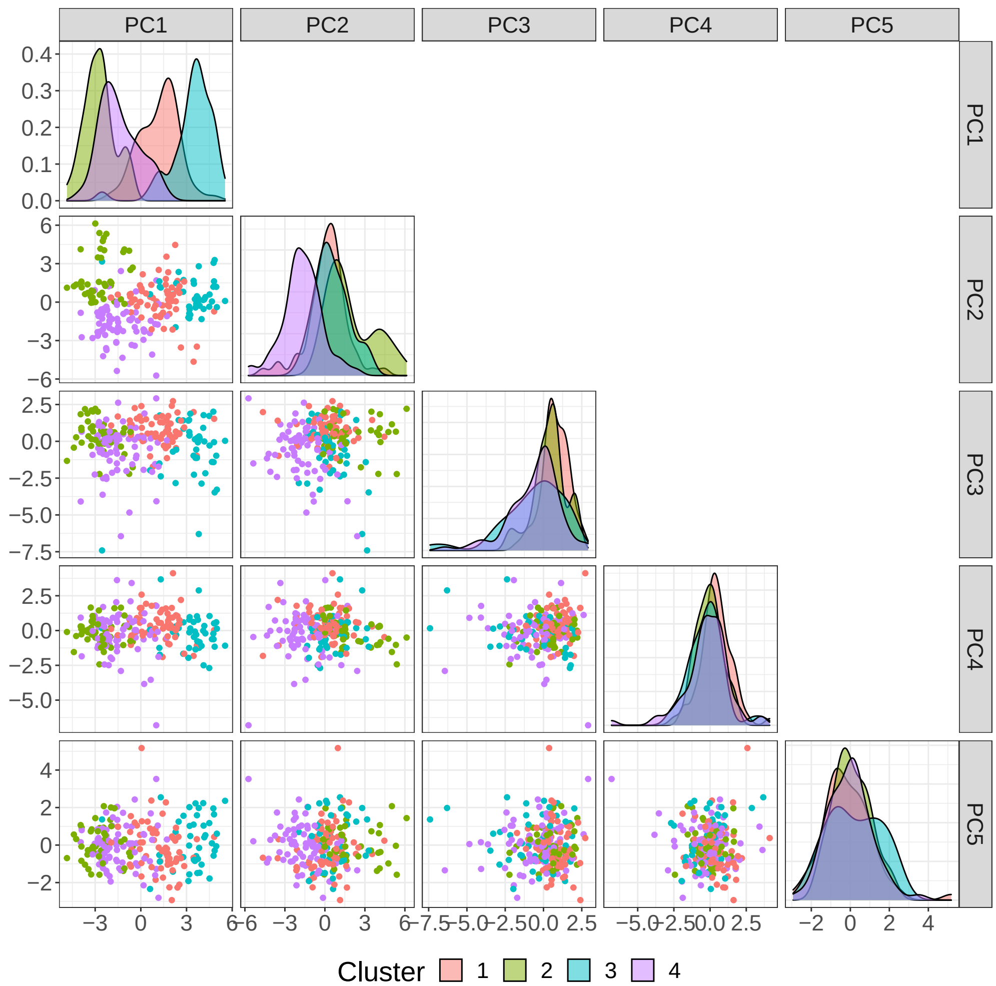
Join the principal component scores with the obtained cluster labels, and plot the densities of principal component scores within clusters, as well as pair-wise plots of the principal component scores.

```
df_pca <- df_pca %>% left_join(tib_clusters)
```

```
## Joining, by = "Player"
```

```
df_pca %>% pivot_wider(names_from = Var2, values_from = value) %>%
  GGally::ggpairs(columns = 3:7, aes(colour = as.factor(cluster)), upper = "blank",
                  diag = list(continuous = GGally::wrap("densityDiag", alpha = 0.5)),
                  legend = c(1,1))+
  labs(fill = "Cluster")+
  theme_bw() + theme(text = element_text(size = 20), legend.position = "bottom")
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```



The distributions of principal component scores are well separated between clusters (at least for the first two principal components). We can now try to interprete what the scores on the differend PCs actually mean. For this, we look at the matrix of loadings.

```r
loadings <- as_tibble(pca$rotation[ , 1:n_pcs]) %>%
  mutate_all( ~ { ifelse(abs(.x)> 0.1, round(.x,2),  "")}) %>%
  mutate(metric = rownames(pca$rotation))

knitr::kable(loadings[, c(n_pcs+1, 1:n_pcs)])
```

metric	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
Blocks_Blocks		-0.17		-0.34	0.2	-0.43	0.28	-0.26	-0.22	0.15
Sh_Blocks	-0.26	-0.19		-0.26		-0.18	0.12	-0.13	-0.21	
Clr	-0.29	-0.2	-0.12		0.11					
Int		-0.13			0.44	0.26	0.26	0.13	0.25	0.45
Won_Aerial Duels	-0.27	-0.21			0.1				-0.11	
TacklesWonPercent				0.12	0.14	-0.6	-0.43	-0.12		-0.15
Tkl_percent_Challenges			0.14	0.56	0.35		0.22		-0.17	-0.22
Cmp_percent_Total	-0.23	0.37						-0.18		
Lost_Challenges	0.25			-0.47			-0.14	-0.18	0.27	
Err				-0.37	0.22			0.53	-0.28	-0.26
Mins_Per_90			0.16		-0.3	-0.26	0.55			0.16

metric	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9	PC10
Carries_Carries	-0.12	0.42			0.12			0.14	-0.11	
PrgC_Carries	0.29	0.21					0.1	0.28	-0.11	0.15
Mis_Carries	0.32				0.12					-0.13
Dis_Carries	0.31	0.13								-0.28
PrgDist_Carries		0.41						0.28	-0.14	0.16
CrsPA	0.28		0.11			-0.21		0.26		0.15
Cmp_percent_Long	-0.25	0.2		-0.2	0.12					-0.17
Cmp_percent_Short		0.4				-0.11		-0.22		
Cmp_percent_Medium	-0.29	0.26						-0.13		
PKcon			-0.26	0.17	0.15	-0.38	-0.17	0.22	0.4	0.34
OG				-0.12	-0.18	-0.19	0.37	0.32	0.58	-0.45
Fls	0.11		-0.49		0.18		0.26	-0.13		-0.17
Tkl_Tackles	0.24		0.18		0.48			-0.16	0.16	-0.17
CardsPer90			-0.47		0.2			-0.12		
FoulsPerTackle			-0.56		-0.18				-0.19	

Interpretation

- first PC separates all clusters quite well:
- clearest separation between clusters 1,2 and 2,3; 1,3; 3,4;
- some overlap of clusters 1 and 4, but rather due to large variation; modus is well separated;
- cluster 2 and 4 overlap most
- second PC separates cluster 4 from the rest.
- PC3 doesn't separate too well between the groups
- first PC
  - high negative loads on Shots blocked, Clearances, Won Aerial Duels, completed pass percentage (also long + medium completed pass perc.)
  - moderately negative load on carries
  - high positive loads on lost challenges, Progressive carries, Mis_carriesm dispossessed carries, Crosses into penalty area, Number of tackles
  - moderately positive load on Fouls
  - $\Rightarrow$ high negative scores on PC1 can be interpreted as having good positioning and defendings skills (able to clear, block shots, win aerial duels) while not taking too many risks (few tackles, fouls; safe passes, few failed or dispossessed carries)
  - confident and safe playing style
  - contarily, high scores on PC1 are related to rather bad positioning and
  - having to tackle/foul more, less safe playing style
- second PC
- has moderately high negative loads on Blocks in total, shots blocked, clearances, intercepts and aerial duels,
- high positive loads on pass completion percentage (all distance categories), carries, progressive carries, progressive distance carried
- balances classic defending skills and ball playing qualities,
- high scores on PC2 related to great ball playing qualities,
- low scores to great defending skills and less good ball playing ability.

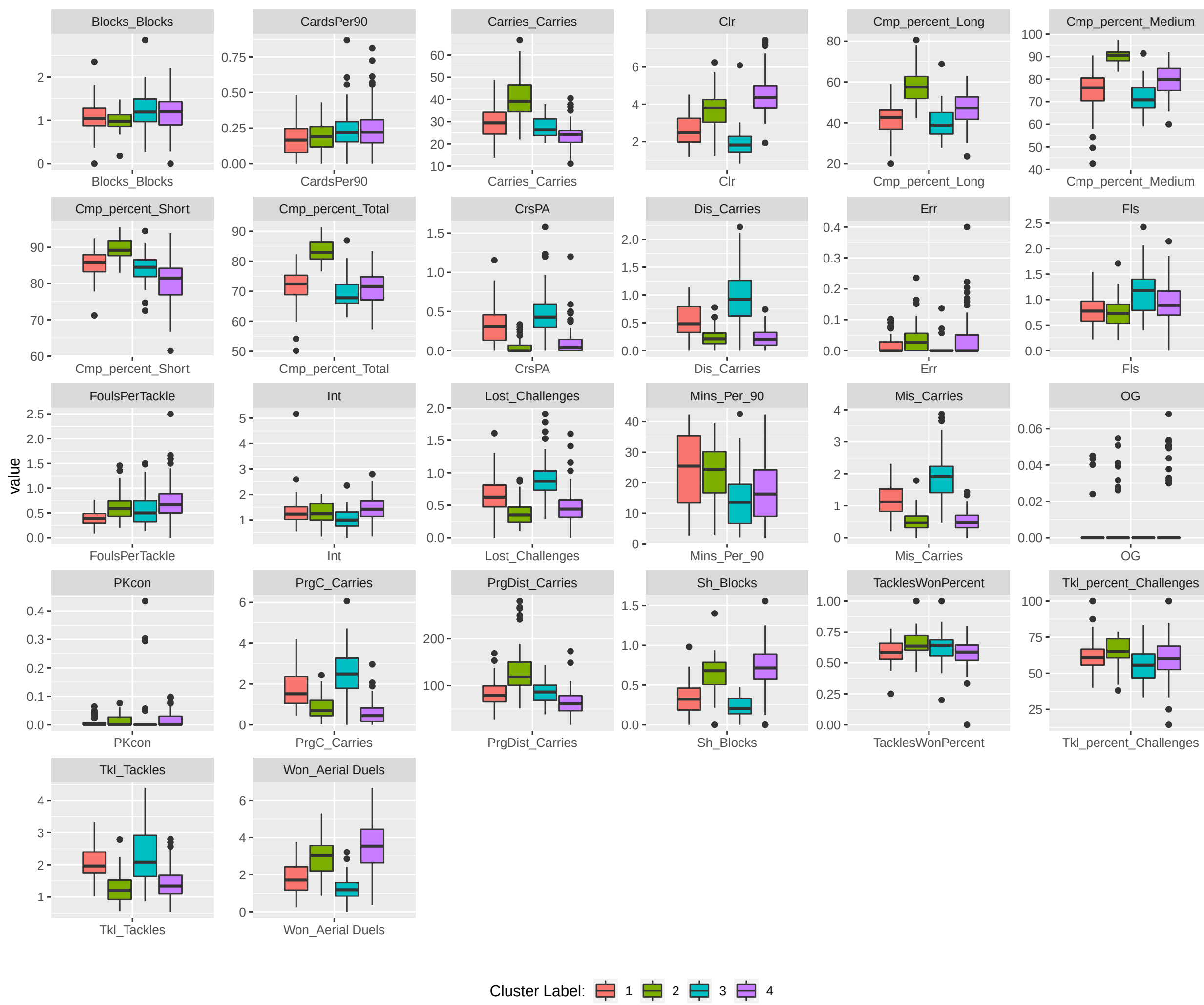
- Clusters 1 and 3 are quite balanced, cluster 4 with negative scores, cluster 2 with positive scores
- third PC
  - loadings suggest that PC3 measures 'intelligent'/good tackling skills or experience/routine
  - positive loadings on Tkl_percent_challenges, Mins_Per_90, Total Tackles
- negative loadings on conceded penalties, Fouls, CardsPer90, FoulsPerTackle
- having low scores on PC3 is rather unpleasant.

Groupwise boxplots of original stats

```
cbs_ana <- cbs_ana %>% select(-"Mins_Per_90") %>% left_join(tib_clusters)
```

```
## Joining, by = "Player"
```

```
cbs_ana %>% pivot_longer(cols = Blocks_Blocks:FoulsPerTackle) %>%
  ggplot(aes( x = name, y = value, fill = as.factor(cluster))+
  geom_boxplot()+
  facet_wrap(~name, scales = "free")+
  labs(fill = "Cluster Label:", x = "")+
  theme(legend.position = "bottom")
```



Finally, we get a list of players assigned to clusters.

```
cluster_list <- df_pca %>% select(Player, cluster) %>% unique() %>%
  group_by(cluster) %>%
  nest()%>%
  mutate(playerlist = purrr::map_chr(data, ~ paste(unlist(.x), collapse = ", "))) %>%
  select(-data) %>% ungroup()

knitr::kable(cluster_list)
```

cluster	playerlist
1	Max Aarons, Anel Ahmedhodzic, Ajibola Alese, Amari'i Bell, Joe Bennett, Jake Bidwell, James Bree, Sam Byram, Cyrus Christie, Dennis Cirkin, Matthew Clarke, Maxime Colin, Jamilu Collins, Lewie Coyle, Fankaty Dabo, Tendayi Darikwa, Jay Dasilva, Anfernee Dijksteel, Callum Doyle, Aaron Drewe, João Ferreira, Morgan Fox, Leo Fuhr Hjelde, Jordan Gabriel, Luke Garbutt, Jacob Greaves, Andrew Hughes, Trai Hume, Ryan John Giles, Hassane Kamara, Todd Kane, Kaine Kessler, Peter Kioso, Ethan Laird, Andy Lyons, Ian Maatsen, Ryan Manning, Sam McCallum, Tom McIntyre, Dan McNamara, Kal Naismith, Perry Ng, Ryan Nyambe, Callum O'Dowda, Kenneth Paal, Harry Pickering, Brad Potts, Cameron Pring, Joe Rankin-Costello, Mahlon Romeo, Josh Ruffels, Thomas Sang, Tommy Smith, Dujon Sterling, Jakob Sørensen, George Tanner, Conor Townsend, Oliver Turton, Josh Tymon, Vitinho, Zak Vyner, Murray Wallace, Ben Wilmot, Andy Yiadom
2	Semi Ajayi, Daniel Ayala, Daniel Ballard, Danny Batth, Jordan Beyer, Marc Bola, Ben Cabango, Hayden Carter, Craig Cathcart, Jake Clarke-Salter, Ameen Al Dakhil, Harry Darling, Robert Dickie, Hjalmar Ekdal, Tobias Figueiredo, Dael Fry, Ben Gibson, Grant Hanley, Taylor Harwood-Bellis, Wesley Hoedt, Phil Jagielka, Alfie Jones, Christian Kabasele, Tomáš Kalas, Joël Latibeaudière, Darragh Lenihan, Sean McLoughlin, Paddy McNair, Kyle Naughton, Dara O'Shea, Andrew Omobamidele, Jonathan Panzo, Ashley Phillips, Erik Pieters, Connor Roberts, Dion Sanderson, Charlie Taylor, Curtis Tilt, William Troost-Ekong, Axel Tuanzebe, Scott Wharton, Jack Whatmough, Nathan Wood-Gordon, Dominic Hyam, Martin Kelly
3	Ryan Andrews, George Baldock, Jayden Bogle, Cohen Bramall, Callum Brittain, Jack Burroughs, Harry Clarke, Alfie Doughty, Cody Drameh, Tayo Edun, Álvaro Fernández, Tariqe Fosu, Darnell Furlong, Dimitris Giannoulis, Lynden Gooch, Dan Gosling, Jordan Graham, Nesta Guinness-Walker, Jaheim Headley, Ki-Jana Hoever, Junior Hoilett, Isaiah Jones, Osman Kakay, Emmanuel Longelo, Max Lowe, Scott Malone, Luke Mbete-Tatu, James McClean, Clinton Mola, James Morris, Jeremy Ngakia, Niels Nkounkou, Rhys Norrington-Davies, Brooke Norton-Cuffy, Armstrong Okoflex, Fred Onyedinma, Przemyslaw Placheta, Baba Rahman, Matthew Sorinola, Enda Stevens, Mark Sykes, Josh Williams, Josh Wilson-Esbrand
4	Robert Atkinson, Leon Balogun, Kyle Bartley, Chris Basham, Tyler Blackett, Will Boyle, Sonny Bradley, Reece Burke, Steven Caulker, Ciaran Clark, Callum Connolly, Jake Cooper, Charlie Cresswell, Greg Cunningham, Scott Dann, Harlee Dean, Bambo Diaby, Jimmy Dunne, John Egan, Marvin Ekpiteta, Callum Elder, Aden Flint, George Friend, Mario Gaspar, Brodie Gilmore, Grant Hall, Wes Harding, Kortney Hause, Michal Helik, Tom Holmes, Charlie Hughes, Cameron Humphreys, James Husband, Sam Hutchinson, Shaun Hutchinson, Jason Kerr, Cédric Kipré, Timm Klose, Tom Lees, Liam Lindsay, Tom Lockyer, Kevin Long, Matthew Lowton, Amadou Mbengue, Jamie McCart, Kyle McFadzean, Mark McGuinness, Yuta Nakayama, Luke O'Nien, Gabriel Osho, Tom Pearce, Matty Pearson, Lee Peltier, Ryan Porteous, Dan Potts, Omar Rekik, Marc Roberts, Jack Robinson, Michael Rose, Mouhamadou-Naby Sarr, Francisco Sierralta, Jack Simpson, Harry Souttar, Jordan Storey, Dominic Thompson, Jordan Thorniley, Auston Trusty, Rhys Williams, Richard Wood, Rarmani Edmonds-Green, Luke McNally, Curtis Nelson, Bailey Wright