AssistsAnalysis

In soccer, the most valuable players are those who score the most goals. However, goalscorers on their own can't do much, they need someone look at those providers, the playmakers. It could be argued that assisting is harder than scoring, since it involves picking out a player in space, d of him (if he is making a run behind defense), and doing all of this while under pressure from the opponent's defenders.

I'll be using data from the English Women Super League (FAWSL), provided by statsbomb (https://statsbomb.com/), which provides event track end locations & pass height. Statsbomb has opensourced the data from both the 2018/19 & 2019/20 seasons.

During this period, there have been 383 assisted goals, and the top five assist providers are:

Vivianne Miedema - Arsenal: 18

Bethany Mead - Arsenal: 14

Caroline Weir - Manchester City: 10

Danielle van de Donk - Arsenal: 10

Keira Walsh - Manchester City: 10

Incidently, these are the only players with double digits assists. In addition to assisted goals, I also include all assisted shots. I do this for two assisted shots compared to only 383 assisted goals. 2- I want to analyze playmaking skill not goal scoring skill; a good playmaker can create a g while the other might shoot the ball wide - the playmaker ability shouldn't be judged by this miss.

Here is a snapshot of the original data:

id	player.name	xA	play_pattern.name	pass.length	pass.angle	pass.height.name	pass.body_part.name	sta
bdaec1e8-b743-4128-b5ec-a82b1d95cf28	Rachel Rowe	0.02395784	From Throw In	9.725224	1.498756300	Ground Pass	Right Foot	91
fd1f9f4a-3ed5-4797-8160-70a588729ad5	Remi Allen	0.29257497	Regular Play	15.156847	1.213570000	High Pass	Head	10
910f8e78-7b3d-4f00-9874-216fd5176be4	Amalie Vevle Eikeland	0.05880328	From Free Kick	3.883298	-0.602287350	High Pass	Head	10
afb3671e-2658-4139-a047-b7dec26cd247	Jade Moore	0.22572555	From Throw In	24.515300	-1.776191700	Low Pass	Right Foot	11
a151418f-f5f1-4598-8913-51717d59c835	Jade Moore	0.41332054	Regular Play	16.542370	-1.339013300	Low Pass	Right Foot	11
99970782-23ea-4bd5-8d4d-ab03a58f9661	Fara Williams	0.02900176	From Free Kick	23.678050	0.221417460	Low Pass	Right Foot	75
1a3b13d8-aad4-4358-a160-cb6a18c4c7ae	Remi Allen	0.08613400	From Keeper	41.046803	-0.317128100	High Pass	Right Foot	48
b63f95bc-1206-4be2-b250-e0d523558779	Jade Moore	0.31610504	From Counter	55.986607	-0.344345570	High Pass	Right Foot	26
e398bf0f-f059-4f09-abdc-d3aebb671d32	Rachel Rowe	0.26187900	Regular Play	16.031220	-0.581082300	Ground Pass	Right Foot	91
2975780a-1599-483a-828b-0a54cda44b4f	Keira Walsh	0.11667784	From Corner	35.833645	-1.978253100	Ground Pass	Left Foot	12
80685559-b412-4a20-aa11-64cdefd830ea	Jill Scott	0.06016142	From Corner	4.427189	2.819842000	Ground Pass	Right Foot	10
7c8d2f36-e660-4777-9edc-7066e1e4b57e	Katie McCabe	0.05458994	Regular Play	23.631546	1.220940700	Low Pass	Left Foot	92

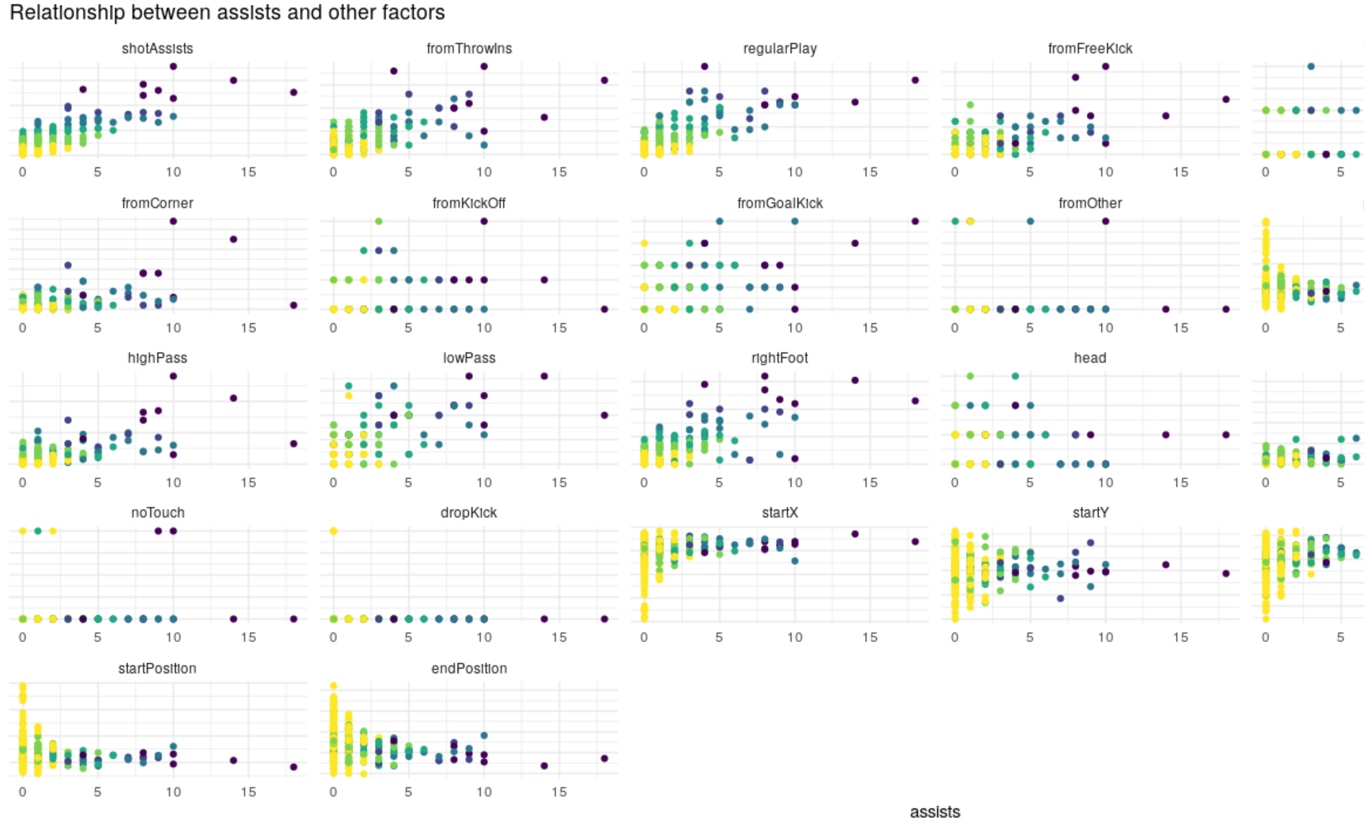
As mentioned before, the data here describes events, I still need to aggregate it in order to get the totals. The final dataset looks like this:

	player.name	shotAssists	assists	fromThrowIns	regularPlay	fromFreeKick	fromKeeper	fromCounter	fromCorner	fromKickOff	fromGoalKick	fro
1	Vivianne Miedema	63	18	16	27	10	2	2	2	0	4	0
2	Bethany Mead	75	14	8	19	7	1	1	35	1	3	0
3	Caroline Weir	89	10	5	18	16	0	4	44	1	0	1
4	Danielle van de Donk	57	10	19	21	2	1	4	6	3	1	0
5	Keira Walsh	39	10	2	18	3	0	7	5	0	4	0
6	Fara Williams	65	9	11	19	7	1	6	18	1	2	0
7	Janine Beckie	42	9	13	19	4	1	2	2	0	1	0
8	Katie McCabe	33	9	4	18	5	1	0	4	0	1	0
9	Erin Cuthbert	60	8	10	18	8	0	4	18	0	2	0
10	Jill Scott	43	8	6	23	3	0	7	2	1	1	0
11	Lucy Staniforth	71	8	10	18	14	1	7	18	1	2	0
12	Ramona Bachmann	37	8	12	10	3	0	4	7	0	1	0
13	Guro Reiten	38	7	10	16	3	0	0	8	1	0	0
14	Jonna Andersson	41	7	10	13	7	2	1	6	1	1	0
15	Kim Little	38	7	7	9	6	2	2	11	0	1	0
16	Inessa Kaagman	33	6	4	9	6	1	2	9	0	2	0
17	Julia Simic	25	6	7	6	4	0	3	2	1	2	0
18	Charlie Wellings	34	5	9	15	2	1	3	1	1	1	1

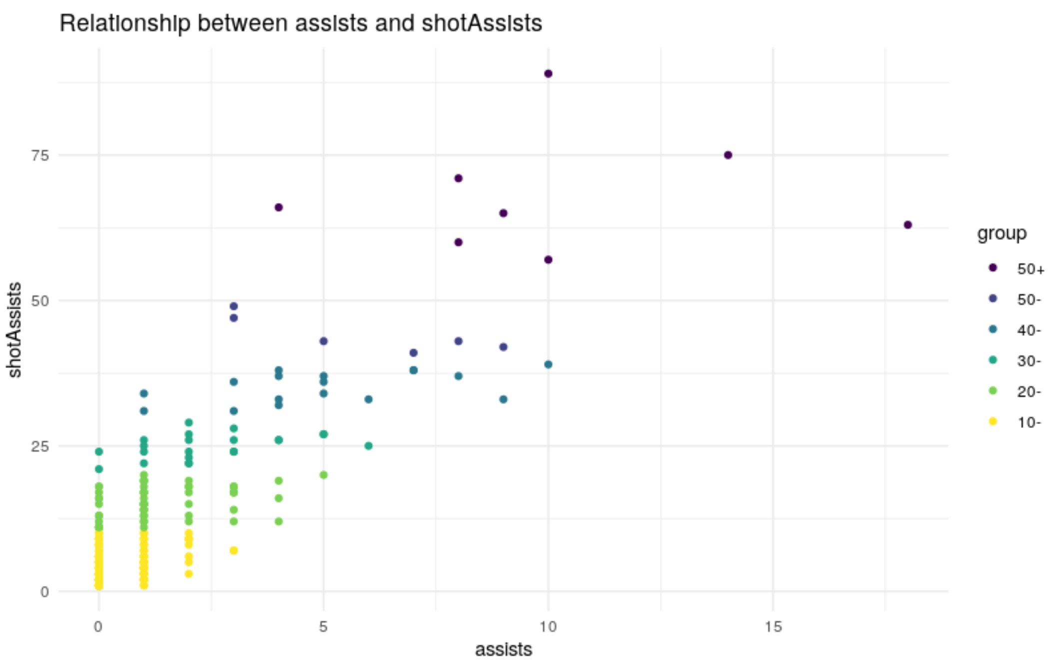
The aggregated data doesn't include any categorical variables, since these have been turned into counts (for example, Vivianne Miedema has p players into separate bins, determined by the number of assists they provide. In total, we have 6 groups, from those who have provided less thar assists (50+):

	Var1	Freq
1	50+	8
2	50-	6
3	40-	17
4	30-	22
5	20-	54
6	10-	148

Now, let's have a look at the relationship between assists and the different features:



Each figure describes the relationship between assists and a specific feature. One thing of note here is that there is no distinct relationship b
mentioned here that the top left image shows the relationship between goal assists and shot assists, so it doesn't actually show a relationship be
between two aggregate values. What I'm trying to say is that shot assists are a good proxy of the goal assists:

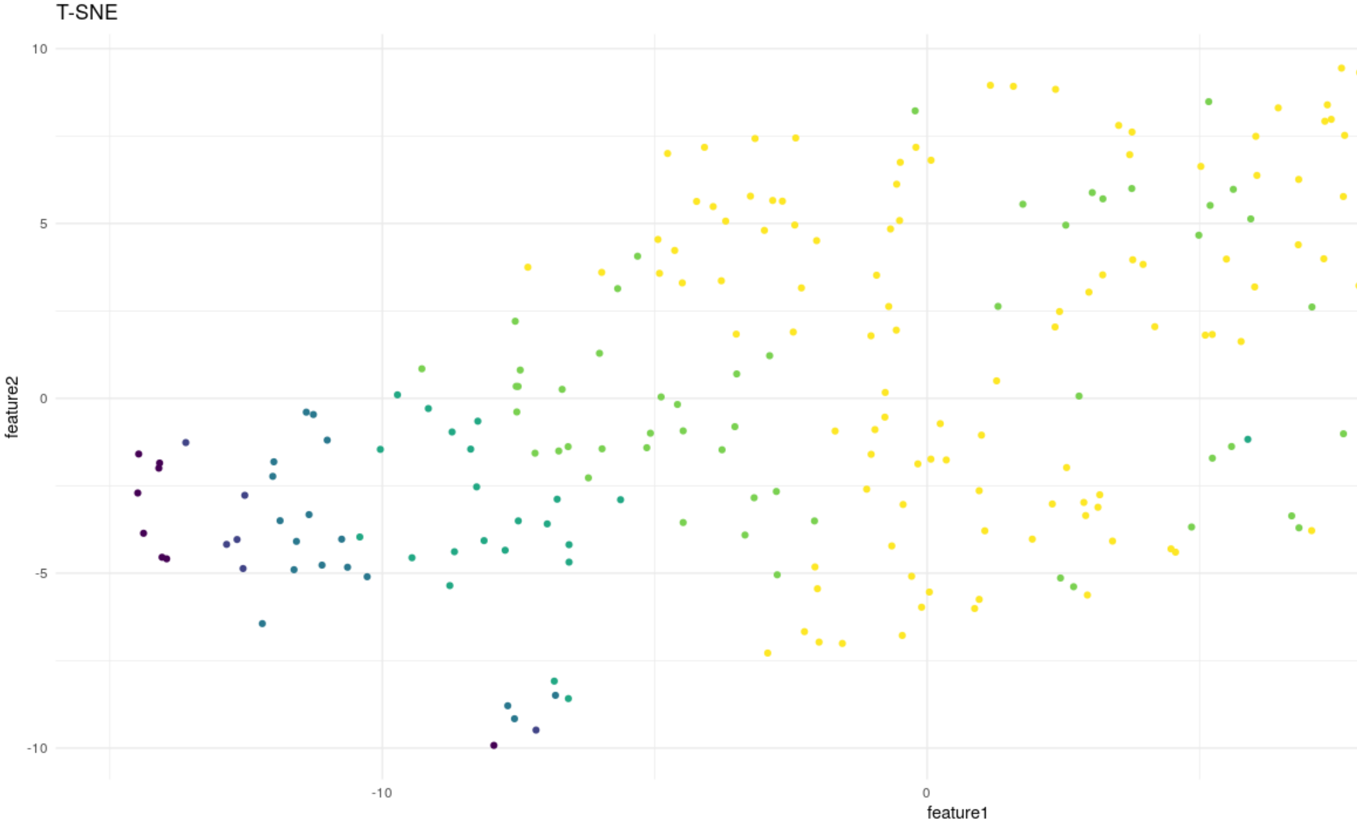


A good dimensionality reduction technique should be able to separate the distinct groups when plotted. This will be the main measure of how effe

Dimensionality Reduction

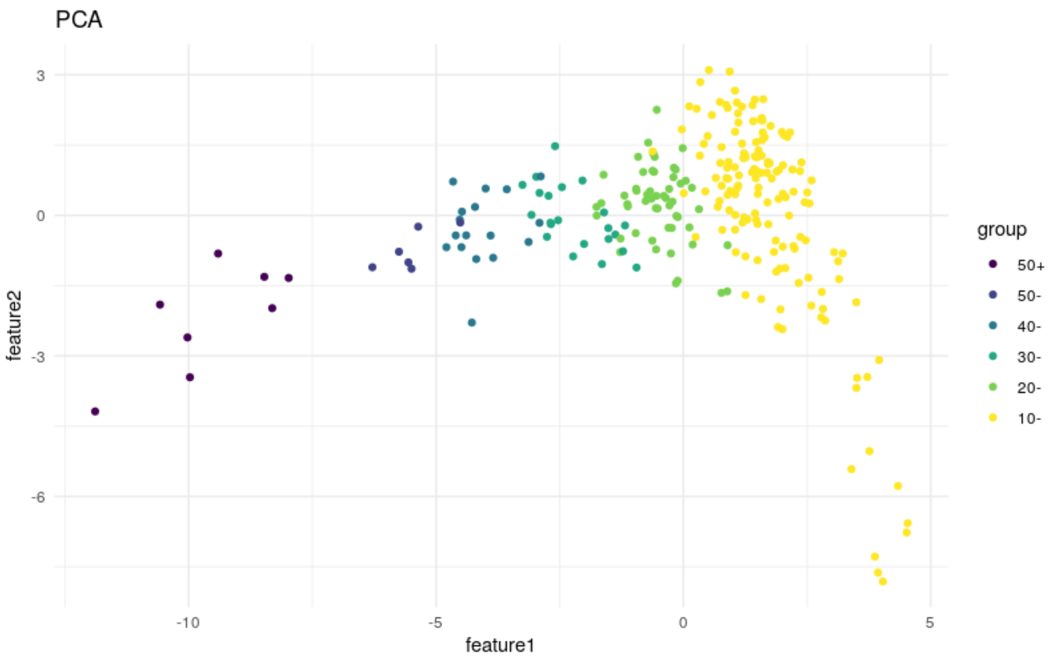
1- T-SNE

I start with the T-SNE technique. Here is the result:



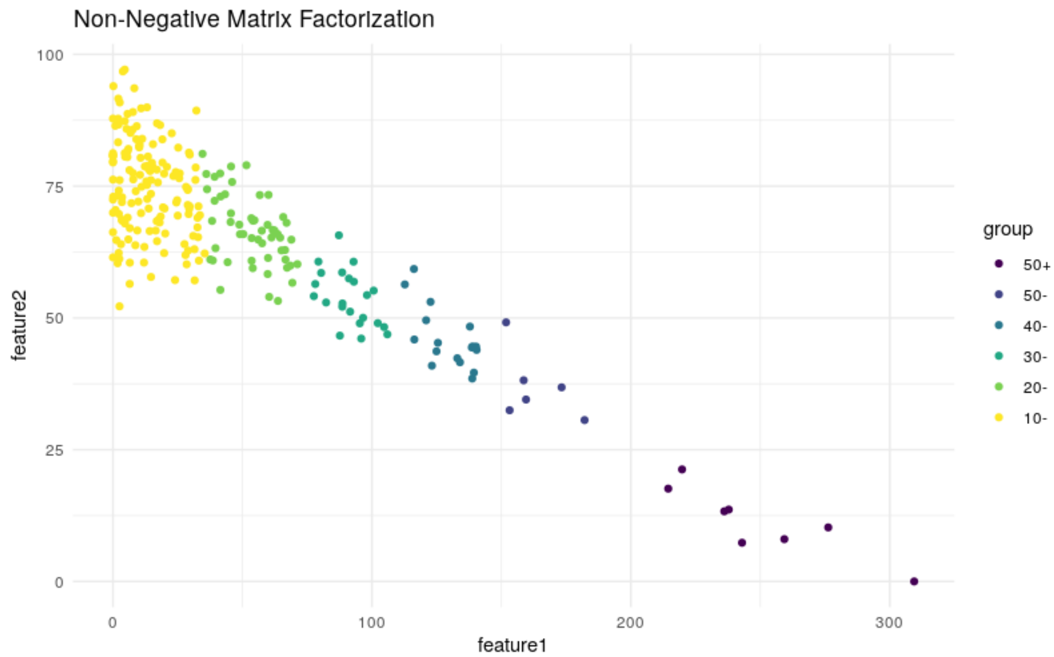
2- PCA

Next, I look at PCA. Here is the result:



3- Non-negative Matrix Factorization

Finally, non-negative matrix factorization. Here is the result:



From these figures, we can say confidently that non-negative matrix factorization is the best dimensionality reduction technique for our

Appendix I - Code

```
# libraries
library(StatsBombR)
library(dplyr)
library(ggplot2)

########################################################################################## RETRIEVING DATA
# 1- get free competitions
# 2- get free matches
# 3- get FA Women's Super League matches

getAssists <- function(fromSource=FALSE){
  if (fromSource){
    Comp <- FreeCompetitions()
    Matches <- FreeMatches(Comp)
    FAWSL <- filter(Matches, competition.competition_name == 'FA Women\'s Super League')

    # find common columns
    col_counter <- data.frame(x=character(0), y=numeric(0), stringsAsFactors=FALSE)
    colnames(col_counter) <- c('colname', 'colcount')
    `%notin%` <- Negate(`%in%`)

    for (i in FAWSL$match_id){
      print(i)
      colz <- c(colnames(get.matchFree(filter(FAWSL, match_id == i))))
      for (j in 1:length(colz)){
        if(colz[j] %notin% col_counter$colname){
          col_counter[nrow(col_counter) + 1,] = list(colz[j], 1)
        } else {
          col_counter$colcount[col_counter$colname == colz[j]] <- col_counter$colcount[col_counter$colname == colz[j]]+1
        }
      }
    }

    # get match events
    colkeys <- col_counter$colname[col_counter$colcount == 194]

    FAWSLEvents <- data.frame()
    for (i in FAWSL$match_id){
      print(i)
      event <- select(get.matchFree(filter(FAWSL, match_id == i)), all_of(colkeys))
      event$match_name <- paste(filter(FAWSL, match_id == i)$home_team.home_team_name,
                                'v',
                                filter(FAWSL, match_id == i)$away_team.away_team_name)
      event$match_date <- filter(FAWSL, match_id == i)$match_date
```

```
  FAWSLEvents <- rbind(FAWSLEvents, event)
}

# get assisted shots
FAWSLEvents$xA <- NA
FAWSLXG <- FAWSLEvents[!is.na(FAWSLEvents$shot.key_pass_id),c('shot.key_pass_id','shot.statsbomb_xg')]
FAWSLEvents[FAWSLEvents$id %in% FAWSLXG$shot.key_pass_id,'xA'] <- cbind(FAWSLEvents[FAWSLEvents$id %in% FAWSLXG$shot.key
xADataset_M <- FAWSLEvents[!is.na(FAWSLEvents$xA),]
xADataset_M <- select(xADataset_M,
                      id,
                      player.name,
                      xA,
                      location,
                      play_pattern.name,
                      starts_with('pass'),
                      -pass.assisted_shot_id,
                      -pass.shot_assist,
                      -pass.recipient.id,
                      -pass.recipient.name,
                      -pass.height.id,
                      -pass.type.id,
                      -pass.body_part.id,
                      -pass.outcome.id,
                      -pass.cross,
                      -pass.switch,
                      -pass.type.name,
                      -pass.outcome.name
)

xADataset_M$start.X <- NA
xADataset_M$start.Y <- NA
xADataset_M$end.X <- NA
xADataset_M$end.Y <- NA
for (i in c(1:nrow(xADataset_M))){
  xADataset_M[i, 'start.X'] <- unlist(xADataset_M[i,'location'])[1]
  xADataset_M[i, 'start.Y'] <- unlist(xADataset_M[i,'location'])[2]
  xADataset_M[i, 'end.X'] <- unlist(xADataset_M[i,'pass.end_location'])[1]
  xADataset_M[i, 'end.Y'] <- unlist(xADataset_M[i,'pass.end_location'])[2]
}
xADataset_M <- select(xADataset_M, -location, -pass.end_location)

# missing values
apply(is.na(xADataset_M), 2, sum)
xADataset_M[is.na(xADataset_M$pass.body_part.name),'pass.body_part.name'] <- 'Other'

# handling categorical columns
xADataset_M$play_pattern.name <- as.factor(xADataset_M$play_pattern.name)
xADataset_M$pass.height.name <- as.factor(xADataset_M$pass.height.name)
xADataset_M$pass.body_part.name <- as.factor(xADataset_M$pass.body_part.name)
assistedShots <- FAWSLEvents[!is.na(FAWSLEvents$shot.outcome.name) & FAWSLEvents$shot.outcome.name=='Goal' & !is.na(FAWS
assists <- FAWSLEvents[FAWSLEvents$id %in% assistedShots$shot.key_pass_id,]
xADataset_M$pass.outcome <- ifelse(xADataset_M$id %in% assists$id, 'Goal', 'No goal')

totalXA <- xADataset_M %>%
  group_by(player.name) %>%
  summarise(
    shotAssists=n(),
    assists=sum(pass.outcome=='Goal'),
    fromThrowIns=n_distinct(id[play_pattern.name=='From Throw In']),
    regularPlay=n_distinct(id[play_pattern.name=='Regular Play']),
    fromFreeKick=n_distinct(id[play_pattern.name=='From Free Kick']),
    fromKeeper=n_distinct(id[play_pattern.name=='From Keeper']),
    fromCounter=n_distinct(id[play_pattern.name=='From Counter']),
    fromCorner=n_distinct(id[play_pattern.name=='From Corner']),
    fromKickOff=n_distinct(id[play_pattern.name=='From Kick Off']),
    fromGoalKick=n_distinct(id[play_pattern.name=='From Goal Kick']),
    fromOther=n_distinct(id[play_pattern.name=='Other']),
    passLength=mean(pass.length),
    groundPass=n_distinct(id[pass.height.name=='Ground Pass']),
    highPass=n_distinct(id[pass.height.name=='High Pass']),
    lowPass=n_distinct(id[pass.height.name=='Low Pass']),
    rightFoot=n_distinct(id[pass.body_part.name=='Right Foot']),
    head=n_distinct(id[pass.body_part.name=='Head']),
    leftFoot=n_distinct(id[pass.body_part.name=='Left Foot']),
    other=n_distinct(id[pass.body_part.name=='Other']),
    noTouch=n_distinct(id[pass.body_part.name=='No Touch']),
    dropKick=n_distinct(id[pass.body_part.name=='Drop Kick']),
```

```
        startX=mean(start.X),
        startY=mean(start.Y),
        endX=mean(end.X),
        endY=mean(end.Y)
      ) %>%
      arrange(desc(assists))
    totalXA$group <- ifelse(totalXA$shotAssists<=10,'10-',
                            ifelse(totalXA$shotAssists>10 & totalXA$shotAssists<=20,'20-',
                                   ifelse(totalXA$shotAssists>20 & totalXA$shotAssists<=30, '30-',
                                          ifelse(totalXA$shotAssists>30 & totalXA$shotAssists<=40, '40-',
                                                 ifelse(totalXA$shotAssists>40 & totalXA$shotAssists<=50, '50-','50+')))))
    totalXA$group <- factor(x=totalXA$group, levels=c('50+','50-','40-','30-','20-','10-'))
    write.csv(totalXA, 'totalXA.csv', row.names = FALSE)
  } else {
    totalXA <- read.csv('totalXA.csv')
  }
  return (totalXA)
}

##############################################################################################################

totalXA <- getAssists(fromSource=FALSE)

##############################################################################################################

totalXA %>%
  select(-player.name) %>%
  reshape2::melt(id.vars=c('assists','group')) %>%
  ggplot() +
  aes(x=assists, y=value, color=group) +
  geom_point() +
  scale_colour_viridis_d() +
  facet_wrap(~variable, scales = "free") +
  theme_minimal() +
  theme(axis.title.y=element_blank(),
        axis.text.y=element_blank(),
        axis.ticks.y=element_blank()) +
  labs(title = "Relationship between assists and other factors",
       x ='assists')

ggplot(totalXA) +
  aes(x=assists, y=shotAssists, color=group) +
  geom_point() +
  scale_colour_viridis_d() +
  theme_minimal() +
  labs(title = "Relationship between assists and shotAssists")

######################################################################################## DIMENSION REDUCTION

# TNSE
set.seed(823)
RtsneAssists <- Rtsne::Rtsne(
  X=select(totalXA,-player.name,-shotAssists,-assists,-group)
)

RTSNEFeatures <- data.frame(RtsneAssists$Y,totalXA$group)
colnames(RTSNEFeatures) <- c('feature1','feature2','group')
ggplot(RTSNEFeatures) +
  aes(x=feature1, y=feature2, color=group) +
  geom_point() +
  scale_color_viridis_d() +
  theme_minimal() +
  labs(title = "T-SNE")


# PRCOMP
prcompAssists <- prcomp(
  x = select(totalXA,-player.name,-shotAssists,-assists,-group),
  center = TRUE,
  scale. = TRUE,
  rank = 2
)

PRCCompFeatures <- data.frame(prcompAssists$x[,1:2],totalXA$group)
colnames(PRCCompFeatures) <- c('feature1','feature2','group')
ggplot(PRCCompFeatures) +
  aes(x=feature1, y=feature2, color=group) +
```

```
  geom_point() +
  scale_color_viridis_d() +
  theme_minimal() +
  labs(title = "PCA")


# NONNEGATIVE
nmfAssists <- NMF::nmf(
  x = select(totalXA,-player.name,-shotAssists,-assists,-group),
  rank = 2
)
basis_acq <- NMF::basis(nmfAssists)
coef_acq <- NMF::coef(nmfAssists)
t(round(head(coef_acq),3)) %>% View()

nonNegFeatures <- data.frame(basis_acq, totalXA$group)
colnames(nonNegFeatures) <- c('feature1','feature2','group')
ggplot(nonNegFeatures) +
  aes(x=feature1, y=feature2, color=group) +
  geom_point() +
  scale_color_viridis_d() +
  theme_minimal() +
  labs(title = "Non-Negative Matrix Factorization")
```