DAR F23 Project Status Notebook Template

Hockey Analytics

Caleb Smith

2023-09-29

Contents

Weekly Work Summary

NOTE: Follow an outline format; use bullets to express individual points.

- RCS ID: smithc22
- Project Name: Hockey Analytics
- Summary of work since last week
  - Describe the important aspects of what you worked on and accomplished I used UMAP and Kmeans to create clusters, than visualized and analyzed them.
- NEW: Summary of github issues added and worked
  - N/A
- Summary of github commits
  - include branch name(s)
  - include browsable links to all external files on github
  - Include links to shared Shiny apps -The commit for sending this assignment is going to be my first real commit.
- List of presentations, papers, or other outputs
  - N/A
- List of references (if necessary)
- Indicate any use of group shared code base
- Indicate which parts of your described work were done by you or as part of joint efforts *Jeff helped me with the code for the density plots
- **Required:** Provide illustrating figures and/or tables

Personal Contribution

- Clearly defined, unique contribution(s) done by you: code, ideas, writing… This code is the main contribution. I shared my data processing code with Lieben, so it should show up in his notebook in some form.
- Include github issues you've addressed N/A

Analysis: Question 1: Modes of Failure

Question being asked

Provide in natural language a statement of what question you're trying to answer I was trying to determine what the modes of failure were for goals. At the suggestion of Dr. Morgan, I dropped all the goals from my analysis, as there were so few of them we believed there wasn't much information to be gained from studying them

Data Preparation

Provide in natural language a description of the data you are using for this analysis I read all of the data in from the RDS object Mohammed created. *Include a step-by-step description of how you prepare your data for analysis* I then used one hot encoding for categorical variables to fix the issue with categorical variables I discovered in a PCA I did earlier. I also made sure that I dropped whether a shot was a goal or not, as this wasn't relevant to any of the clustering. Labels were created out of whether the shot was a goal or not. Finally, the data was scaled to make sure features were treated fairly in clustering. *If you're re-using dataframes prepared in another section, simply re-state what data you're using*

```
# Include all data processing code (if necessary), clearly commented
#Data processing, library calls, and other basic setup:
shots <- readRDS("shots_stats_goal.df.Rds")#Need to make sure this is grabb
library(data.table)
library(mltools)
library(plotly)
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(umap)
library(ggplot2)
library(scatterplot3d)
library(rgl)
```

```
## Warning in rgl.init(initValue, onlyNULL): RGL: unable to open X11 display
```

```
## Warning: 'rgl.init' failed, running with 'rgl.useNULL = TRUE'.
```

```
library(car)
```

```
## Loading required package: carData
```

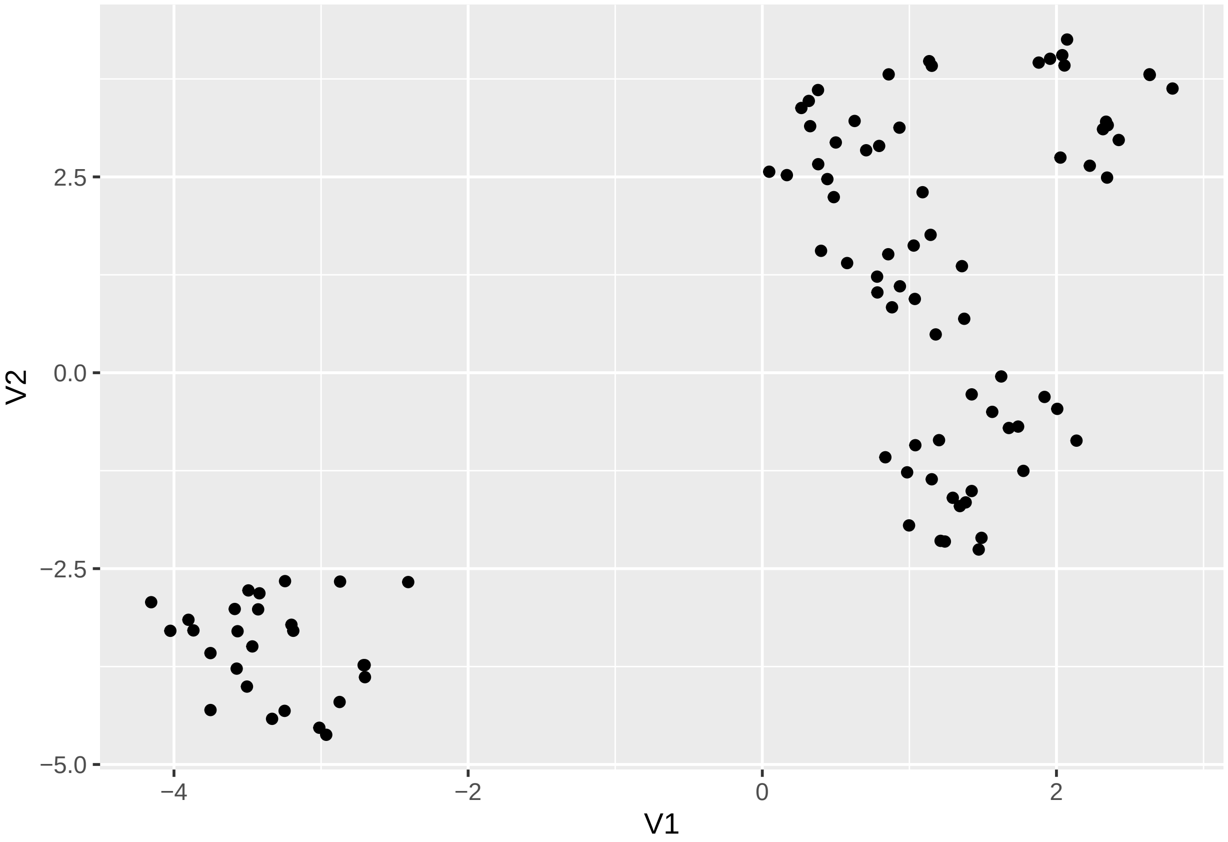
```
library(grid)
library(gridExtra)
library(kableExtra)
shotsNum <- shots
#One-hot encoding and making sure goals are dropped
shotsNum$goalieScreened = as.factor(shotsNum$goalieScreened)
shotsNum$oppDefenders = as.factor(shotsNum$oppDefenders)
shotsNum$sameDefenders = as.factor(shotsNum$sameDefenders)
shotsNum <- shotsNum[,1:11]
shotsNum <- one_hot(dt = as.data.table(shotsNum))
labels <- ifelse(shots$outcomes.goal == 1, TRUE, FALSE)#creating labels to use for graph coloring
shotsNum <- scale(shotsNum)
#This is needed for finding some of the medians
shots2 <- shots
shots2[,10] <- as.numeric(shots$goalieScreened)
shots2[,12] <- as.numeric(shots$outcomes.goal)
fails <- as.data.frame(shotsNum) %>% filter(shots$outcomes.goal == 0)

#Seed setting - UMAP still acts wonky regardless of this, so I'll be reading it in from RDS files for c
custom.config <- umap.defaults
custom.config$random_state <- 2392023
set.seed(100)

select <- dplyr::select

#General Functions
plotAll <- function(data,toFile = FALSE,fileName = "graph.pdf"){
  plotsList <- list()
  for(i in 1:(ncol(data)-1) ){
    p <- NULL
    axis <- colnames(data)
    axises <- axis[i]
    if(i >= 8){#Hard coded to do barplots on the categorical variables. Will change this later to check
      p <-  ggplot(data = data,aes_string(x = axises,fill = "cluster"))+
  geom_bar()
    }else{
  p <-  ggplot(data = data,aes_string(x = axises,color = "cluster"))+
  geom_density()
    }
  plotsList[[i]] <- p
  }
  if(toFile){
  pdf(fileName, width = 8, height = 12)
  do.call("grid.arrange", c(plotsList, ncol = 3))
  dev.off()
  }else{
    do.call("grid.arrange", c(plotsList, ncol = 3))
  }
}
#Generates a plot to use the elbow test for K-means. I stole this from IDM
```

```
wssplot <- function(data, nc=15, seed=100){
  wss <- data.frame(cluster=1:nc, quality=c(0))
  for (i in 1:nc){
    set.seed(seed)
    wss[i,2] <- kmeans(data, centers=i)$tot.withinss}
  ggplot(data=wss,aes(x=cluster,y=quality)) +
    geom_line() +
    ggtitle("Quality of k-means by Cluster")
}
#Displays medians of the data for each cluster passed
displayMeds <- function(data, clusters,nClust){
  temp <- data %>% filter(clusters == 1)
  results <- apply(temp,2,median)
  for(i in 2:nClust){
    temp <- data %>% filter(clusters == i)
    results <- rbind.data.frame(results,apply(temp,2,median))
  }
  return(results)
}
```

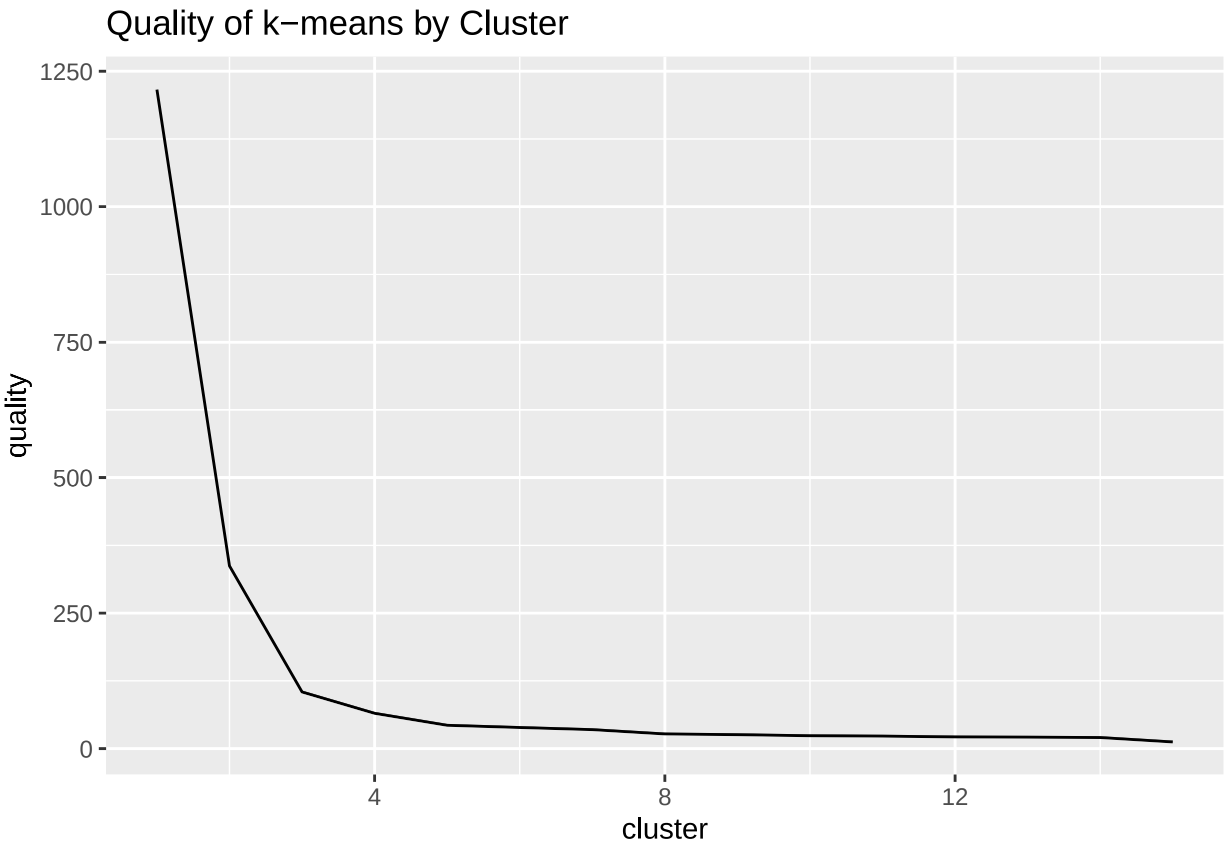
Analysis: Methods and results

Describe in natural language a statement of the analysis you're trying to do UMAP and Kmeans were combined to cluster the data, at which point plots were made to analyse the differences among clusters in the different features. I read this article on what UMAP was actually doing to get the idea to use it for clustering: https://pair-code.github.io/understanding-umap/. It is primarily useful in allowing the clusters that result to be graphed in 2D space so that we can see the relations between points. *Provide clearly commented analysis code; include code for tables and figures!*

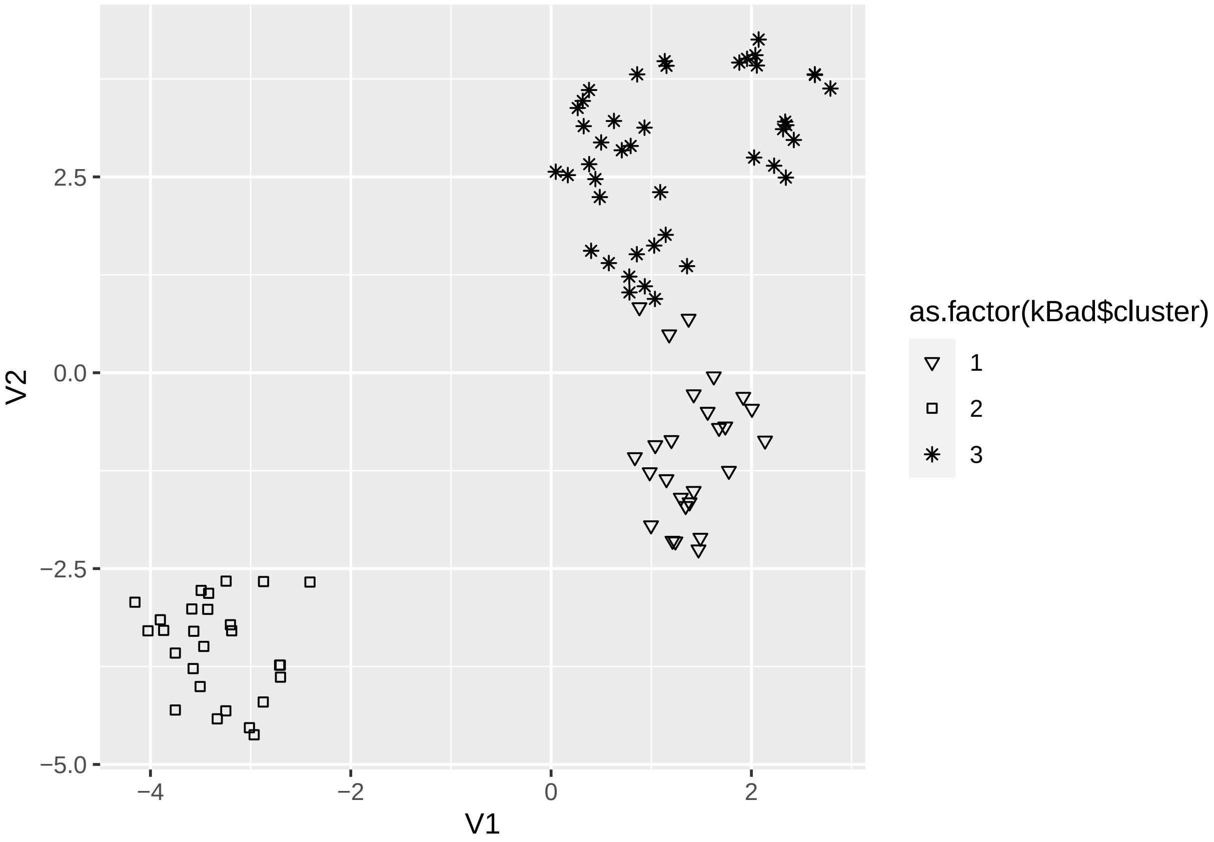
```
# Include all analysis code, clearly commented
# If not possible, screen shots are acceptable.
# If your contributions included things that are not done in an R-notebook,
#   (e.g. researching, writing, and coding in Python), you still need to do
#   this status notebook in R.  Describe what you did here and put any products
#   that you created in github. If you are writing online documents (e.g. overleaf
#   or google docs), you can include links to the documents in this notebook
#   instead of actual text.

#This umap is deciding to behave today so I don't have to read it from RDS
umapFails <- umap(fails,n_components = 2, config = custom.config)
#graph of the umap projection
ggplot()+
 geom_point(data = as.data.frame(umapFails$layout),aes(x = V1,y = V2))
```



```
#Kmeans, elbow plot indicated 3
wssplot(umapFails$layout,15,100)
```

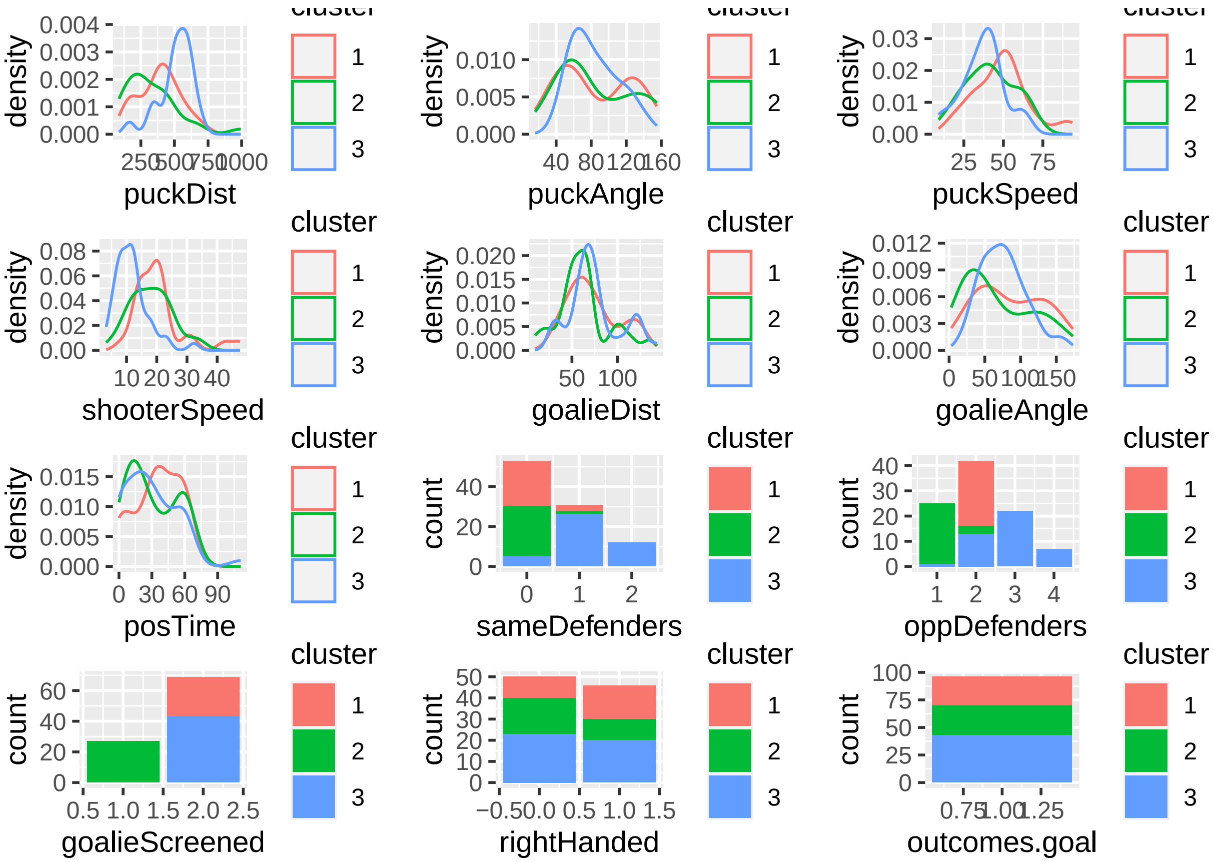


```
kBad <- kmeans(umapFails$layout,3)

#Graph of the clusters
ggplot()+
  geom_point(data = as.data.frame(umapFails$layout),aes(x = V1,y = V2,shape = as.factor(kBad$cluster)))
  scale_shape_manual(values = c(25,22,8))
```



```
#creating summary plots
shots2NoGoals <- shots2 %>% filter(shots2$outcomes.goal == 1)
ggData2 <- cbind.data.frame(shots2NoGoals,kBad$cluster)
ggData2[,13] <- as.factor(kBad$cluster)
colnames(ggData2) <- append(colnames(shots2),"cluster")

plotAll(ggData2,FALSE,"NoGoalGraphs.pdf")
```

```
## 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.
```



Discussion of results

Provide in natural language a clear discussion of your observations.

Note that these questions aren't in chronological order - I performed this analysis after the no distance analysis. The results are very similar, indicating that the goals had a very low influence on the actual clustering due to how few of them there were. Cluster 1 had a faster puck than the other two and also had the same apperance of defensive pressure, as well as the most right handed players. It also had a longer possession time. Cluster 3 was slow and far out - Probably in the "complete miss category". Cluster 1, the close one, didn't have goalie screened. It seems the close shots (the ones that are most likely to go in) aren't screened? (makes sense.) May have to look at whether screening matters when only considering further away shots

Analysis: Question 2 - Minor modes of failure

Question being asked

Provide in natural language a statement of what question you're trying to answer The initial feature selection work indicates that distance from the goal is by far the most important factor in a successful shot. I wanted to know what other, more subtle things had an impact on goals being scored.

Data Preparation

Provide in natural language a description of the data you are using for this analysis I'm using all rows from the scaled and encoded dataframe, but I dropped distance to use in this analysis so that it didn't swamp the other variables when I went to cluster

Include a step-by-step description of how you prepare your data for analysis

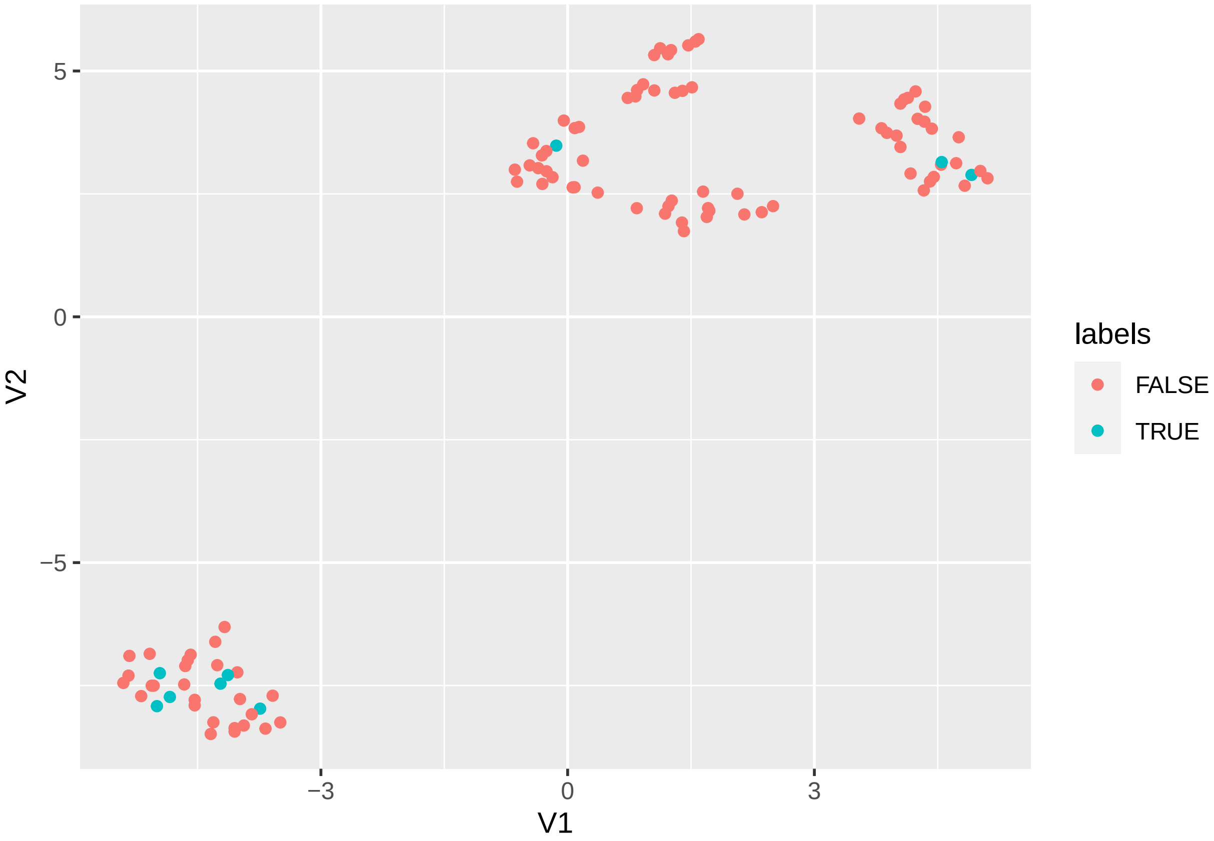
If you're re-using dataframes prepared in another section, simply re-state what data you're using

```
# Include all data processing code (if necessary), clearly commented
noDist <- shotsNum[,2:17] #Drops puckDistance from the features list
```

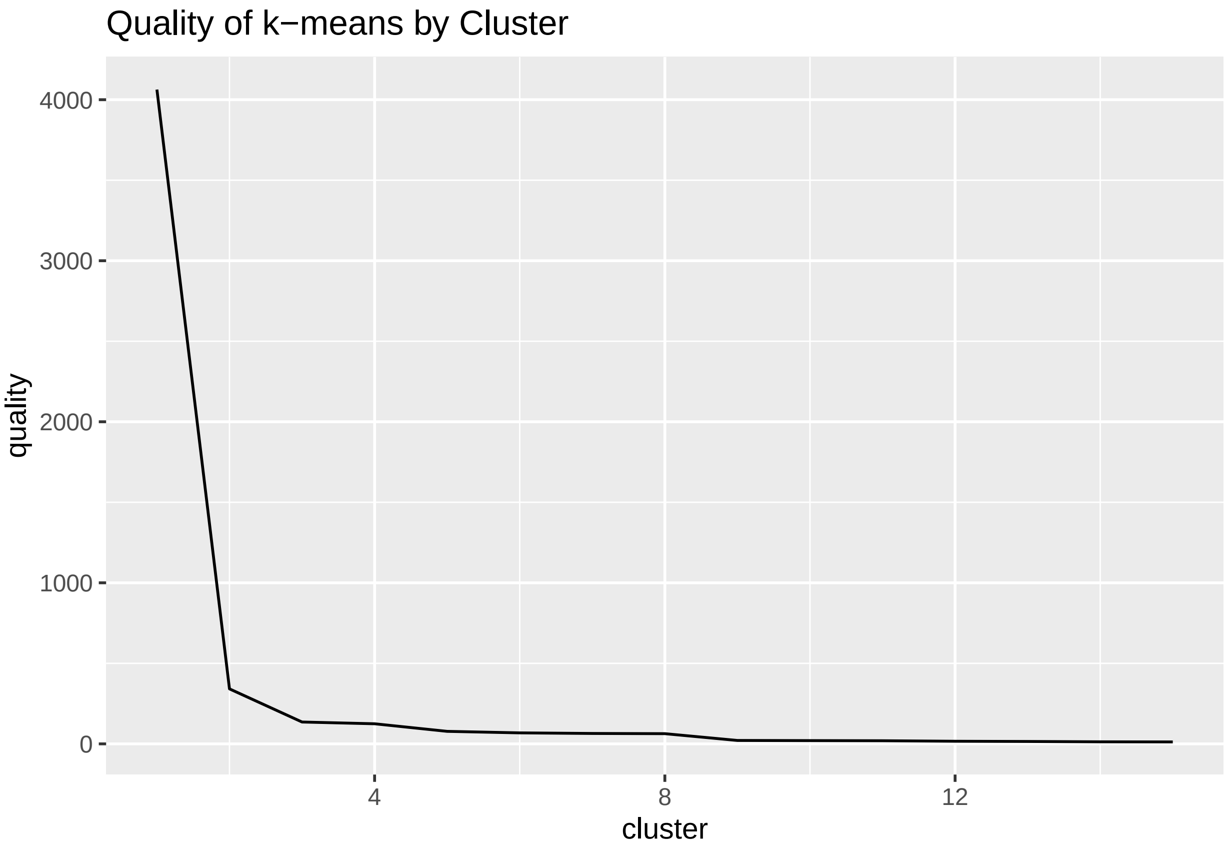
Analysis: Methods and Results

Describe in natural language a statement of the analysis you're trying to do I'm trying to use UMAP and Kmeans to cluster the data and look for different types of failures by analysing the graphs of each cluster. Note that this reads in UMAP_Distance_Dropped and K-means-Distance-Dropped RDS files because I couldn't get it to be reproducible otherwise for some reason. These will be included in the student data folder. *Provide clearly commented analysis code; include code for tables and figures!*

```
# Include all analysis code, clearly commented
# If not possible, screen shots are acceptable.
# If your contributions included things that are not done in an R-notebook,
#   (e.g. researching, writing, and coding in Python), you still need to do
#   this status notebook in R.  Describe what you did here and put any products
#   that you created in github. If you are writing online documents (e.g. overleaf
#   or google docs), you can include links to the documents in this notebook
#   instead of actual text.

#umapNoDistance <- umap(noDist,n_components = 2, config = custom.config)
umapNoDistance <- readRDS("UMAP_Distance_Dropped")
ggplot()+
 geom_point(data = as.data.frame(umapNoDistance$layout),aes(x = V1,y = V2,color = labels))
```



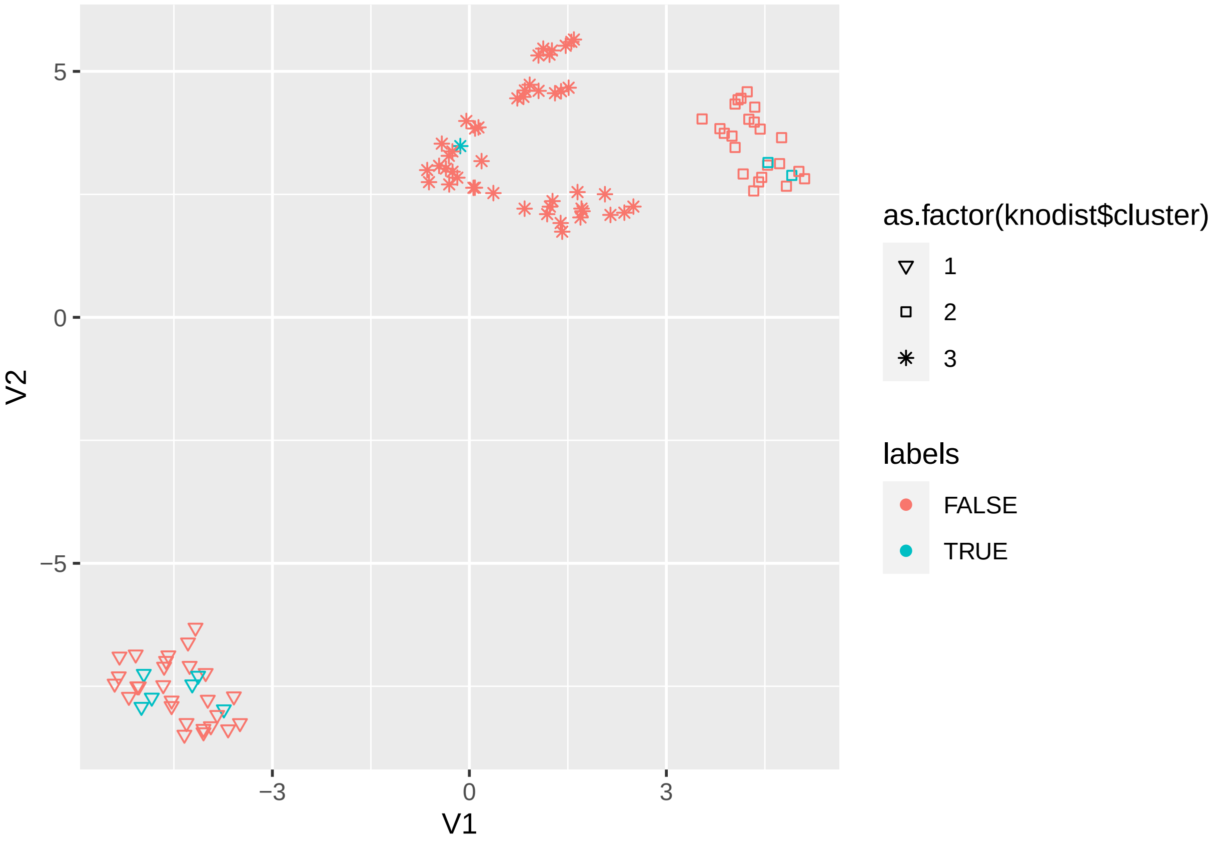
```
wssplot(umapNoDistance$layout)#elbow test indicates 3 clusters here
```



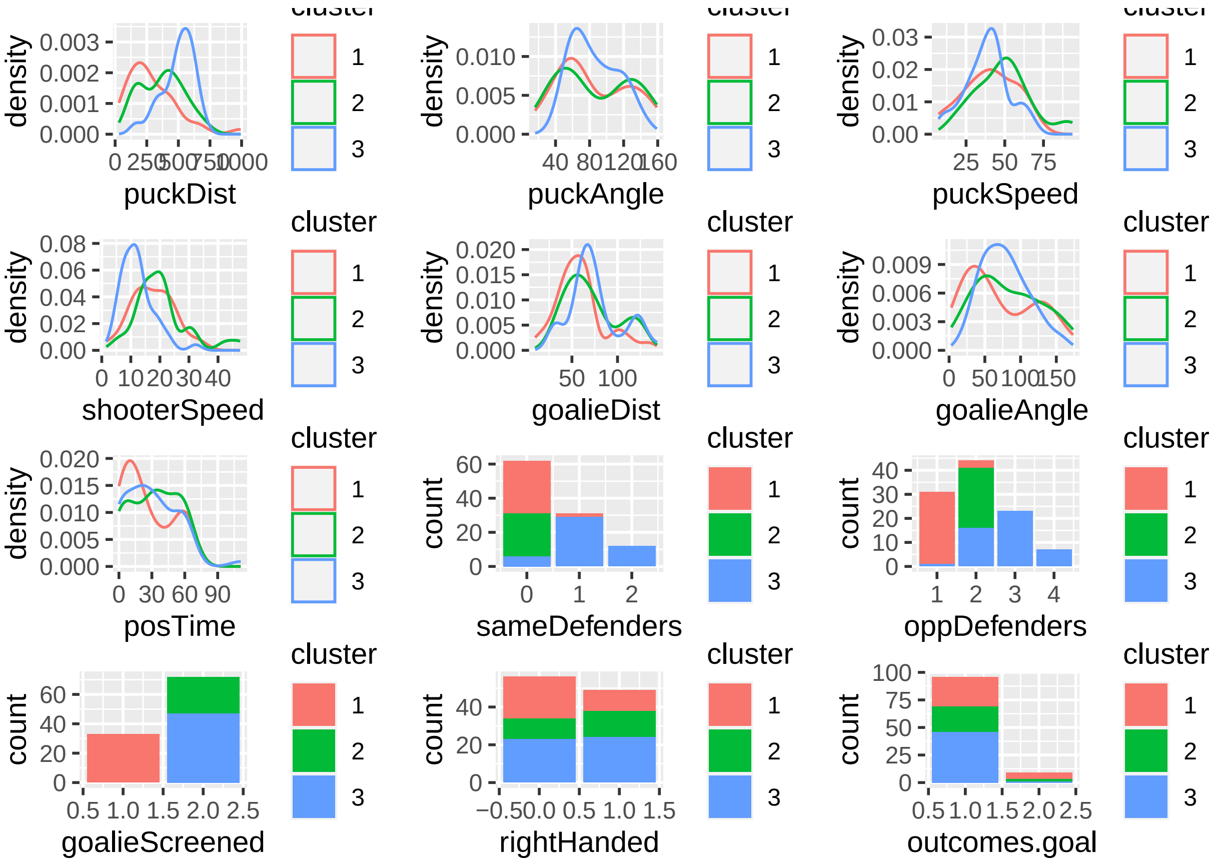
```r
#knodist <- kmeans(umapNoDistance$layout,3) Also acting up
knodist <- readRDS("K-means-Distance-Dropped")

#Preparing a summary table
medNoDist <- displayMeds(as.data.frame(shots2),knodist$cluster,3)
colnames(medNoDist) <- colnames(shots)
kable(medNoDist)
```

puckDist	puckAngle	puckSpeed	shooterSpeed	goalieDist	goalieAngle	posTime	sameDefenders	oppDefende
236.7666	67.29157	41.50166	17.42062	53.78814	54.29706	18	0	
391.9232	73.15754	50.16104	18.89086	63.93345	71.95786	34	0	
514.3987	80.01730	38.93087	11.45568	68.48097	74.29939	26	1	

```r
#Plot of the clusters
ggplot()+
  geom_point(data = as.data.frame(umapNoDistance$layout),aes(x = V1,y = V2,color = labels,shape = as.fa
  scale_shape_manual(values = c(25,22,8))
```



```
#Creating plots of all the features by cluster
ggData <- cbind.data.frame(shots2,knodist$cluster)
ggData[,13] <- as.factor(knodist$cluster)

colnames(ggData) <- append(colnames(shots2),"cluster")

plotAll(ggData)
```



Discussion of results

Provide in natural language a clear discussion of your observations. Suprisingly, data still seemed to stratify by distance to a degree, indicating strong correlation between distance and other variables. The cluster with most of the goals didn't have a screened goalie, indicating that maybe the goalie being screened isn't as important as initially believed. Cluster 2, while still reasonably close to the goal, had far fewer successful goals, so the differences between this and cluster 1 are likely to be the most instructive for other causes of goal failure. One major difference is cluster 2 had 2 opposing defenders but no friendly defenders for the shooter, in contrast to minimal involvment from either team in the cluster 1 shots. This indicates that some of the cluster 2 shots potentially missed due to defensive pressure on the shooter. Cluster 2 also had fewer left handed players than cluster 1.

Analysis: Question 3 N/A

Question being asked

Provide in natural language a statement of what question you're trying to answer What are the differences between different types of failures?

Data Preparation

Provide in natural language a description of the data you are using for this analysis

Include a step-by-step description of how you prepare your data for analysis

If you're re-using dataframes prepared in another section, simply re-state what data you're using

```
# Include all data processing code (if necessary), clearly commented
```

Analysis methods used

Describe in natural language a statement of the analysis you're trying to do

Provide clearly commented analysis code; include code for tables and figures!

```
# Include all analysis code, clearly commented
# If not possible, screen shots are acceptable.
# If your contributions included things that are not done in an R-notebook,
#   (e.g. researching, writing, and coding in Python), you still need to do
#   this status notebook in R.  Describe what you did here and put any products
#   that you created in github. If you are writing online documents (e.g. overleaf
#   or google docs), you can include links to the documents in this notebook
#   instead of actual text.
```

Discussion of results

Provide in natural language a clear discussion of your observations.

Summary and next steps

Provide in natural language a clear summary and your proposed next steps.

There appears to broadly be 3 types of shots, stratified by distance but also with a few other key differences, including goalie screening, speed, and involvement other players. For next steps, I am interested in seeing some new features related to pressure from the defense on the shooter - perhaps distance of the closest opposing player to the shooter? Since one of the clusters has more opposing players involved than players from the shooter's team. I'm also interested to investigate goalie screening more, since it seems that it may not be increasing the chance of a goal being scored.

##Garbage Code Here is some of the other stuff I tried, but it mostly ran into problems because I hadn't heard about 1 hot encoding yet. This was especially problematic for PCA. I also wasn't scaling it at this point, but it looks like UMAP automatically scales the data when it runs its analysis.

```
#a PCA out of curiosity

my.pca <- prcomp(shotsNum,retx=TRUE, center=TRUE, scale=TRUE)

primaryComponents.df <- as.data.frame(my.pca$x)
ggplot()+
  geom_point(data = primaryComponents.df,mapping = aes(x=PC1,y=PC2,color = labels))
#PC1 is still a good fit
#PC5 has decent bunching, as does 6
#PC 7 looks good
#Anythign after 8 is irrelevant
#Doing a 3D plot to see if I can get it to cluster nicely
#library("plot3D")
#library("rgl")
#3 and 6 kind of work as a third vector, 9 works really nicely as a second vector
#Wait, I should probably take outcome goal out of my pca lol
#scatter3D(x = my.pca$x[,1],y = my.pca$x[,5], z = my.pca$x[,7],colvar = labels, theta = 0,phi = 45)

#fig <- plot_ly(as.data.frame(my.pca$x), x = ~PC1, y = ~PC6, z = ~PC7, color = ~labels, colors = c('#BF
#fig <- fig %>% add_markers()
#fig <- fig %>% layout(scene = list(xaxis = list(title = 'PC1'),
#                      yaxis = list(title = 'PC6',
#                      zaxis = list(title = 'PC7')))
```

```
#fig


#pca on transpose, not exactly sure how to analyze this
#hockeyTranspose <- t(hockeyTrainNumPCA)
#pcaT <- prcomp(hockeyTranspose,retx=TRUE, center=TRUE, scale=TRUE)
#ggplot(data = as.data.frame(pcaT$x),aes(x=PC1,y=PC2,label = rownames(pcaT$x)))+
#  geom_text()
#UMap
important <- cbind.data.frame(my.pca$rotation[,1],my.pca$rotation[,7])
colnames(important) <- c("PC1","PC7")

reduced <-cbind.data.frame(my.pca$x[,1],my.pca$x[,7])
colnames(reduced) <- c("PC1","PC7")
#Clustering to see if it works

wssplot(important,10,20)
#3 or 5
k <- kmeans(reduced,centers = 5)

ggplot()+
  geom_point(data = primaryComponents.df,mapping = aes(x=PC1,y=PC2,color = as.factor(k$cluster),shape =
  scale_shape_manual(values = c(2,19))+
  scale_fill_discrete(labels = c("Goal","Cluster"))


hockeyMap <- umap(shotsNum,n_components = 3)
ggplot()+
  geom_point(data = as.data.frame(hockeyMap$layout),aes(x = V1,y = V2,color = labels))
#2 components gives 3 clusters
#scatter3D(x = hockeyMap$layout[,1],y=hockeyMap$layout[,2],z=hockeyMap$layout[,3],colvar = labels,
#          theta = 0, phi = 30) #Use this instead of the 2 components for the other one


#PCA without categorical
hockeyNoCat <- shotsNum[,1:7]
noCatPCA <- prcomp(hockeyNoCat,retx=TRUE, center=TRUE, scale=TRUE)
ggplot()+
  geom_point(data = as.data.frame(noCatPCA$x),mapping = aes(x=PC1,y=PC6,color = labels))#PC6 looks inte

#umap without cat
umapNoCat <- umap(hockeyNoCat,n_components = 2)
ggplot()+
  geom_point(data = as.data.frame(umapNoCat$layout),aes(x = V1,y = V2,color = labels))
#Looks very similar to the 3 component one, i must choose. 3D ultimately looks better

uk <- kmeans(umapNoCat$layout,centers = 3)
k1 <- shotsNum %>% filter(uk$cluster == 1) #shot from too far
k2 <- shotsNum %>% filter(uk$cluster == 2)#Decent Chance
k3 <- shotsNum %>% filter(uk$cluster == 3)#Failed for other reasons
#We got 3 clusters, looks like the dogleg is from categorical variables

#I have no clue which library scatter3D is from
```

```
umapNoCat3 <- umap(hockeyNoCat,n_components = 3)
#scatter3D(x = umapNoCat3$layout[,1],y=umapNoCat3$layout[,2],z=umapNoCat3$layout[,3],colvar = labels,
#          theta = 45, phi = 30) #Removing categorical variables doesn't seem to fix the problem of get
#At least the 3 components seems to have the clusters
#z and x could be two seperate modes of failure here
#saveRDS(umapNoCat3$layout,file = "UMap projection: 3, maybe 4 clusters") This one was only on the trai

#out of curiosity, what if I just straight up remove distance?
noDist <- shotsNum[2:11]
umapNoDist <- umap(noDist,n_components = 2)
ggplot()+
  geom_point(data = as.data.frame(umapNoDist$layout),aes(x = V1,y = V2,color = labels,shape = as.factor
  scale_shape_manual(values = c(25,22,8))
#Just going to do it without the clusters since mine aren't super clean, leave that to the actual clust
ggplot()+
 geom_point(data = as.data.frame(umapNoDist$layout),aes(x = V1,y = V2,color = labels))
#Two obvious clusters with minimals goals in them, even without distance. This is spicy
wssplot(umapNoDist$layout,10)
distK <- kmeans(umapNoDist$layout,3)
#checking cluster 2
badPlays <- cbind(shotsNum,labels) %>% filter(distK$cluster == 2)
badPlays2 <- cbind(shotsNum,labels) %>% filter(distK$cluster == 1)

goodDist <- cbind(shots,umapNoDist$layout) %>% filter(outcomes.goal == 1)
```