DAR F23 Project Status Notebook Template

Hockey Analytics

Caleb Smith

2023-10-16

Contents

Weekly Work Summary

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

- RCS ID: **Always** include this!

smithc22

- Project Name: **Always** include this!

Hockey Analytics

- Summary of work since last week
  - Describe the important aspects of what you worked on and accomplished

  Re-ran clustering analysis with the new features Attempted to handle angle better and analysed the results of doing so

- NEW: Summary of github issues added and worked
  - Issues that you've submitted
  - Issues that you've self-assigned and addressed

  *N/A

- Summary of github commits
  - include branch name(s) dar-smithc22
  - include browsable links to all external files on github
  - Include links to shared Shiny apps

- List of presentations, papers, or other outputs
  - Include browsable links

  https://docs.google.com/presentation/d/1tc2zpWd-nTNv9Cp7yucLfOVzOJNdGTWe10eFp3ulN0U/edit#slide=id.g28b7e1d64e4_1_3 https://docs.google.com/presentation/d/1tc2zpWd-nTNv9Cp7yucLfOVzOJNdGTWe10eFp3ulN0U/edit#slide=id.g28b7e1d64e4_1_3

- 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

  As far as I remember all work here is my own

- **Required:** Provide illustrating figures and/or tables

Personal Contribution

- Clearly defined, unique contribution(s) done by you: code, ideas, writing…
- Include github issues you've addressed

  The code here is the main contribution I've made over the past few weeks

Analysis: Question 1: Are there failure types we haven't identified yet?

Question being asked

Provide in natural language a statement of what question you're trying to answer

Are there more modes of failures that I missed in the last analysis?

Data Preparation

Provide in natural language a description of the data you are using for this analysis I'm using the older data frame here without more detail on defenders, since it wasn't created yet at the time of this analysis

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

1. Drop the goals column, as we are trying to classify types of shots in a manner that would let us predict the outcome.
2. One hot encode
3. Scale the data for even treatment of features
4. Pull the goals out, since we are focusing on modes of failures. Doing this after scaling makes assigning goals to clusters after the fact easier, and since there are so few goals it doesn't mess with the scaling that much

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
#This is taken from my previous notebook
shots <- readRDS("shots_stats_goal.df.Rds")#Need to make sure this is grabb
insertCol <- function(data,loc,col){
  newData <- cbind(data[,1:loc],col)
  newData <- cbind(newData,data[,(loc+1):ncol(data)])
  return(newData)
}
library(data.table)
library(mltools)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
## The following objects are masked from 'package:data.table':
##
##     between, first, last

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

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

```
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

##
## Attaching package: 'car'

## The following object is masked from 'package:dplyr':
##
##     recode
```

```
library(grid)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'

## The following object is masked from 'package:dplyr':
##
##     combine
```

```r
library(kableExtra)
```

```
##
## Attaching package: 'kableExtra'

## The following object is masked from 'package:dplyr':
##
##     group_rows
```

```r
library(heatmaply)
```

```
## Loading required package: viridis

## Loading required package: viridisLite

##
## ======================
## Welcome to heatmaply version 1.4.2
##
## Type citation('heatmaply') for how to cite the package.
## Type ?heatmaply for the main documentation.
##
## The github page is: https://github.com/talgalili/heatmaply/
## Please submit your suggestions and bug-reports at: https://github.com/talgalili/heatmaply/issues
## You may ask questions at stackoverflow, use the r and heatmaply tags:
##   https://stackoverflow.com/questions/tagged/heatmaply
## ======================
```

```r
library(gplots)
```

```
##
## Attaching package: 'gplots'

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

```r
library(tibble)
shotsNum <- shots
goals <- shots[shots$outcomes.goal == 1,]
#One-hot encoding and making sure goals are dropped
shotsNum$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)
#Doing something similar to the goals to allow me to figure out which cluster they're in
goalsScaledCol <- goals
#goalsScaled$goalieScreened = as.factor(goalsScaled$goalieScreened)
#goalsScaled$oppDefenders = as.factor(goalsScaled$oppDefenders)
#goalsScaled$sameDefenders = as.factor(goalsScaled$sameDefenders)
#goalsScaled <- goalsScaled[,1:11]
#goalsScaled <- one_hot(dt = as.data.table(goalsScaled)) #Need to add 0 vector of opp_defender4, same d
#goalsScaled <- as.tibble(goalsScaled)
#goalsScaled %>% add_column(sameDefenders_1 = numeric(9),.after = "sameDefenders_0")
```

```
#this is garbage, I'm just going to write an insert function
#goalsScaledCol <- insertCol(goalsScaled,8,numeric(9))
#goalsScaledCol <- insertCol(goalsScaledCol,9,numeric(9))
#goalsScaledCol <- insertCol(goalsScaledCol,13,numeric(9))
#Turns out I don't need to insert those columns. Oh well, now i have it for later

#Getting right format attempt 2
goalsScaledCol <- apply(goals,2,as.numeric)
```

```
## Warning in apply(goals, 2, as.numeric): NAs introduced by coercion
```

```
#goalScreened <- ifelse(goals$goalieScreened,2,1) #time to make another transform function

#goalsScaled <- scale(goalsScaledCol, attr(temp, "scaled:center"), attr(temp, "scaled:scale"))

#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$outcomes.goal == 0)
shots2NoGoals <- shots2 %>% filter(shots2$outcomes.goal == 1)
temp <- scale(apply(shots2NoGoals,2,as.numeric))
goalsScaled <- scale(goalsScaledCol, attr(temp, "scaled:center"), attr(temp, "scaled:scale"))
#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


assignCluster <- function(centers,data){
  distVec <- numeric(nrow(centers))
  for(i in 1:(nrow(centers))){
    distVec[i] <- dist(rbind(data,centers[i,2:(ncol(centers))]))
  }
  return(which.min(distVec))
}

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))
  }
}
#Not coloring by cluster
plotAllNoCluster <- 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))+
  geom_bar()
    }else{
  p <-  ggplot(data = data,aes_string(x = axises))+
  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)
}
```

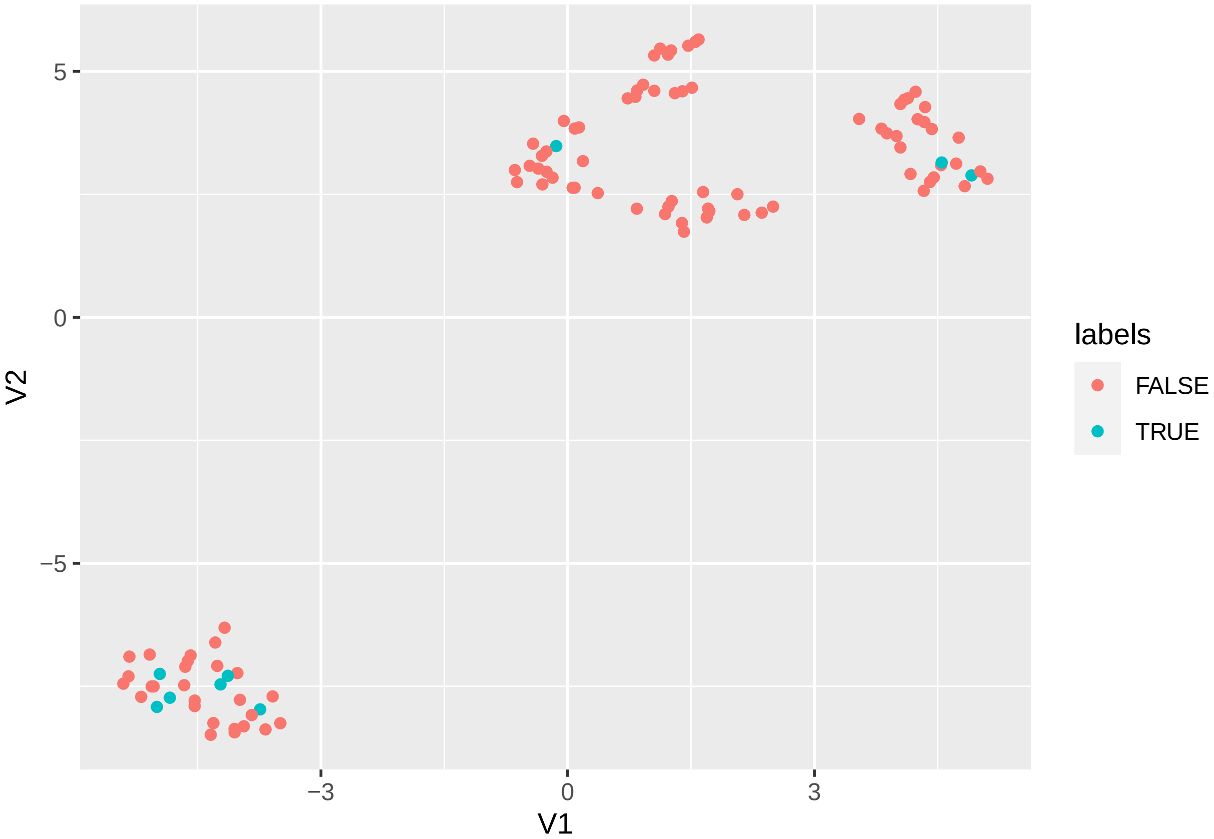
Analysis: Methods and results

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

I'm trying to see if I can use UMAP and K-means to go from three clusters to five, as was suggested at a meeting. All scatterplots will be of UMAP projections. To assess clusters, I will first form and graph a UMAP projection. Then, combined with the elbow test, I will assess and plot the clusters to determine the proper number. To analyse my clusters after creation, I will create a heat map and possibly a summary table. I will name the clusters to keep the differences between them easy to remember.

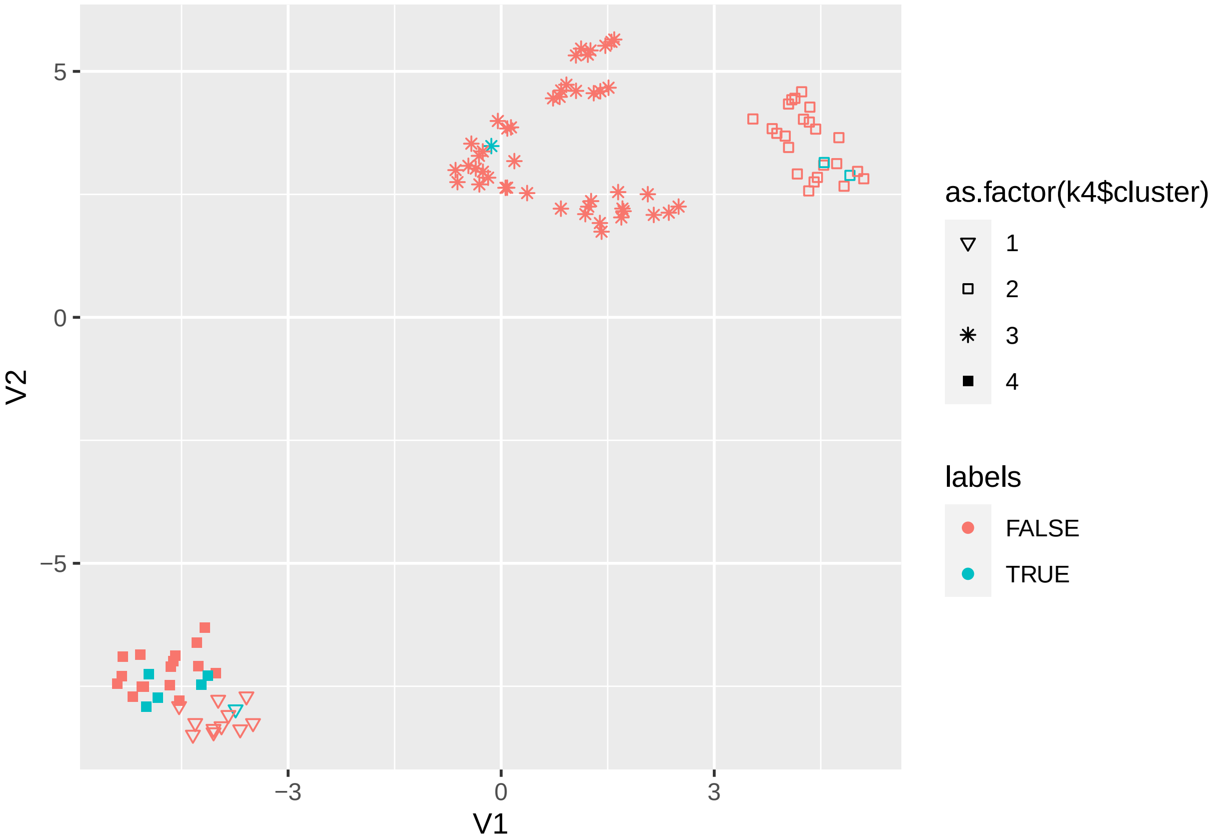
Provide clearly commented analysis code; include code for tables and figures!

```
# 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.

#All shots were under 300 distance
#plotAllNoCluster(goals,TRUE,"Goals_Only.pdf")
#Don't worry about this, not important

#Trying to do 5 clusters instead of 3 on the UMAP
umapNoDistance <- readRDS("UMAP_Distance_Dropped")

ggplot()+
 geom_point(data = as.data.frame(umapNoDistance$layout),aes(x = V1,y = V2,color = labels))
```



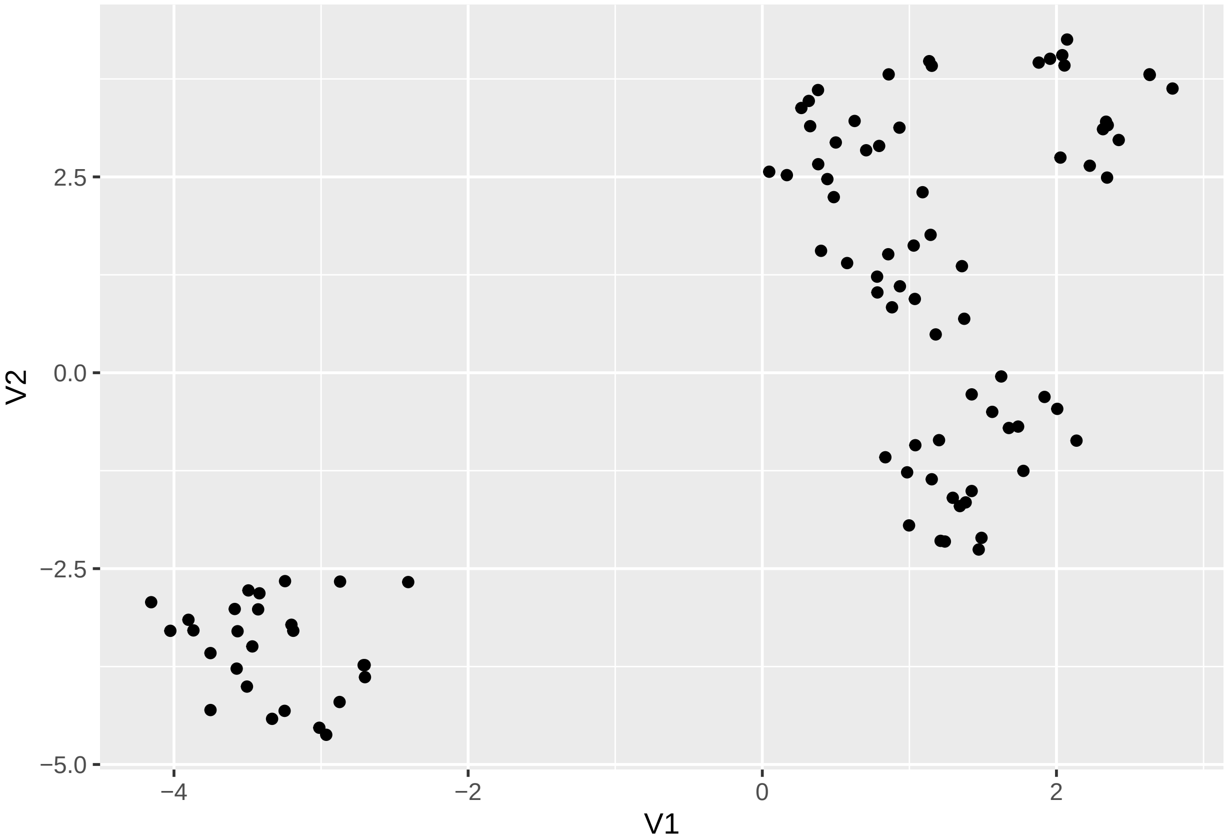
The goal is to see if I can break this up into more than the three obvious clusters, since it looks like the 'V' formation could potentially be split

```
k4 <- kmeans(umapNoDistance$layout,4)

ggplot()+
  geom_point(data = as.data.frame(umapNoDistance$layout),aes(x = V1,y = V2,color = labels,shape = as.fa
  scale_shape_manual(values = c(25,22,8,15))
```



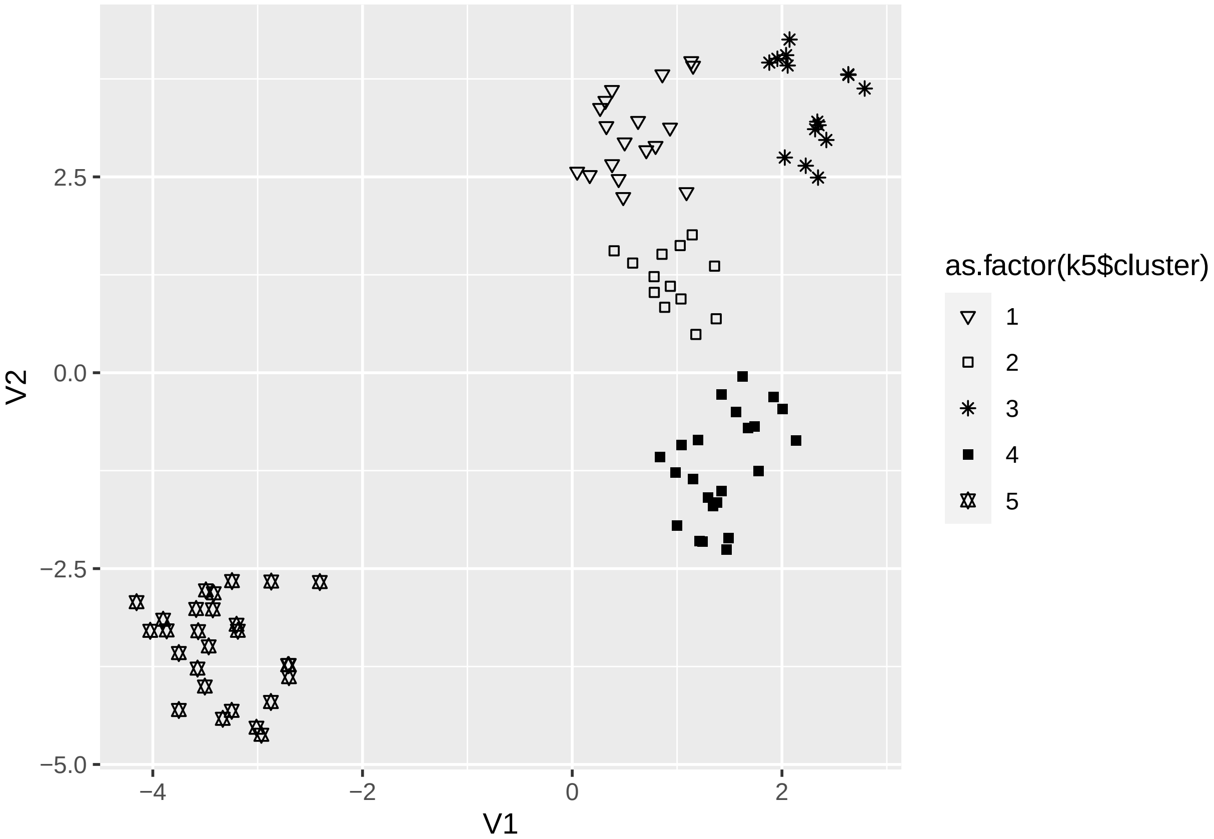
Unfortunately, K-means ends up just splitting the goals corner up in a way that doesn't really make sense. I'll try it on the umap I made of the fails and see if I can put that into more clusters

```
#Doesn't seem to want to work. I'll try the one I ran on the fails instead
umapFails <- umap(fails,n_components = 2, config = custom.config)
ggplot()+
 geom_point(data = as.data.frame(umapFails$layout),aes(x = V1,y = V2))
```



This is fairly similar to the previous projection, with three main clusters and one of the clusters that could be divided into 3.

```
k5 <- kmeans(umapFails$layout,5)
#This looks pretty
ggplot()+
  geom_point(data = as.data.frame(umapFails$layout),aes(x = V1,y = V2,shape = as.factor(k5$cluster))) +
  scale_shape_manual(values = c(25,22,8,15,11))
```



Fortunately, K-means is able to break up the clusters into nice breaks here, which is exactly what I was hoping for.

```
#Analyzing this
ggData <- cbind.data.frame(shots2NoGoals,k5$cluster)
ggData[,13] <- as.factor(k5$cluster)
colnames(ggData) <- append(colnames(shots2),"cluster")

plotAll(ggData,TRUE,"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.
```

```
## pdf
##   2
```

```
#Heatmap as well since 5 clusters on density graphs starts looking disgusting
#Data preperation
shotsNumRegular <- as.data.frame(scale(apply(shots2NoGoals,2,as.numeric)))[,1:11]
preimageCenters <- shotsNumRegular%>% group_by(as.factor(k5$cluster) ) %>% summarise_all(.funs =  mean)
  #apply(fails,2,median) #do this for each factor
heatData <- as.matrix(preimageCenters[,2:11])

goalClust <- numeric(5)
for(i in 1:(nrow(goalsScaledCol))){
  goalClust[assignCluster(preimageCenters,goalsScaled[i,])] <- goalClust[assignCluster(preimageCenters,g
}
goalClust#Almost all 5, a 4 and a 1 in there
```

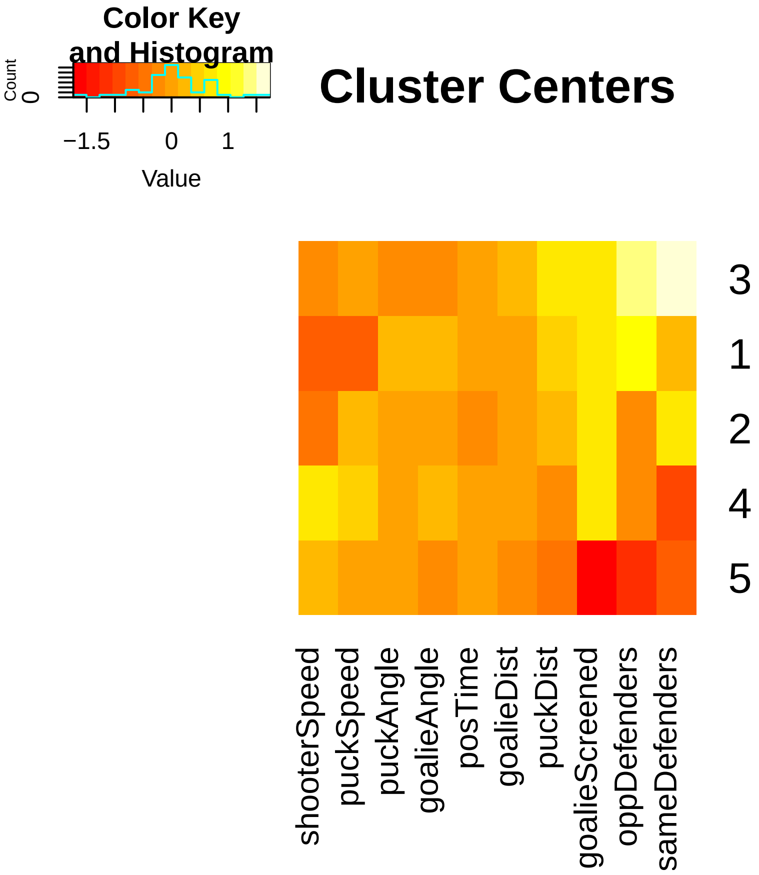
```
## [1] 1 0 0 1 7
```

```
clusterPercents <- goalClust/k5$size
clusterPercents
```

```
## [1] 0.05555556 0.00000000 0.00000000 0.04347826 0.25925926
```

The majority of the goals were in cluster 5, with 1 in 4 and 1 in 1. This will be kept in mind in the analysis at the end of the question. The heatmap below is of the cluster centers, and will be used along with the summary table (which includes the means of each feature for each cluster) to name the clusters

```
#goalsScaled
#preimageCenters

heatmap.2(heatData,
          main = "Cluster Centers",
          scale = "none",
          trace = "none",
          dendrogram = "none",
          Colv = TRUE,
          Rowv = TRUE,
          margins = c(8,16))
```



```
#finding which cluster the goals are in
#goalProj <- predict(umapFails,goals)
#Let's make a summary table as well
summaryTable <- shots2NoGoals %>% group_by(as.factor(k5$cluster)) %>% summarise_all(.funs = mean)
kbl(summaryTable[,1:8],booktabs = T) %>% kable_styling(latex_options = c("striped","scaled_down"),full_
```

as.factor(k5$cluster)	puckDist	puckAngle	puckSpeed	shooterSpeed	goalieDist	goalieAngle	posTime
1	497.5395	95.76245	31.33067	11.28901	73.55111	88.37170	32.94444
2	479.7380	83.37737	44.56910	11.62327	68.04036	77.84730	24.84615
3	529.8660	74.37155	41.96123	13.57873	79.63267	65.44688	32.93333
4	389.3093	82.23158	47.74488	20.97168	73.41236	82.64432	33.91304
5	330.4890	84.04700	41.36843	18.13588	62.18259	68.31499	30.40741

```
kbl(summaryTable[,9:ncol(summaryTable)],booktabs = T) %>% kable_styling(latex_options = c("striped","sca
```

sameDefenders	oppDefenders	goalieScreened	rightHanded	outcomes.goal
0.7222222	2.888889	2	0.6111111	1
1.0000000	2.000000	2	0.6923077	1
1.8000000	3.266667	2	0.2000000	1
0.0000000	2.000000	2	0.5652174	1
0.0740741	1.111111	1	0.3703704	1

Discussion of results

Provide in natural language a clear discussion of your observations.

Cluster 1 'Lousy Shot': Slow, both shooter and puck speed are noticeably low. Far away, most opposing defenders. Looks like this is the defensive pressure cluster where they were getting mobbed, will check when Amy has the new features.

Cluster 2 'Too far': Shooter advantage, they have more people involved with their team than the opposing team has. The shooter is moving pretty slowly, but the puck still has a decent speed. Unfortunately, they appear to be too far out to get it

Cluster 3 'Lousy shot, left side': Full house, lots of players on both teams, pretty far from the goal. The hail mary one from the three cluster group. The angle difference between puck and goalie seems solid

Cluster 4 'Goalie Save': Looks like it has a good chance, close to the goal and very fast. Appears to have less team assistance than 5, with more opposing team members. Also has a much more centered goalie angle, cluster 5 is biased left of the goalie!

Cluster 5: Perfect - 25% of these were scores, the only significant amount. From the summary table we can see they are significantly more likely to be left hand. Also, the goalie isn't screened and there are less defenders involved. Most notably, there is a 16 degree difference between the puck angle and the goalie angle. While the puck shot seemed to be reasonably straight on, the goalie was WAY out of position to the left

Analysis: Question 2 : Can we make these clusters more descriptive by adding more features?

Question being asked

Provide in natural language a statement of what question you're trying to answer

As you probably noticed, the cluster names are rather vague in the previous analysis. However, thanks to some work from Amy and Dr. Morgan, we are able to get more data about the defenders on the ice, and what caused the shot to miss. I'm seeing how releveant this is to modes of failure.

Data Preparation

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

I'm using all the shots from our newest RDS

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

1. Drop the goals column, as we are trying to classify types of shots in a manner that would let us predict the outcome.
2. One hot encode
3. Scale the data for even treatment of features

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

Reading in our new data frame and running the same preparation it was given in question 1, and also making the differential between the puck and goalie a factor. After a discussion with Dr. Morgan on the WebEx, I dropped closestDef as irrelevant.

```
# Include all data processing code (if necessary), clearly commented
#screw it im gonna see if i can get a decent umap with the goals to determine their cluster
new_df <- readRDS("shots_stats_goal.df.newest.Rds")
shotsNew <- new_df[,!names(new_df) %in% c("closestDef","outcomes","shotOutcome")]
shotsNew$angleDiff <- abs(new_df$goalieAngle - new_df$puckAngle)
shotsNumNew <- shotsNew
sumTab <- shotsNew
#Now rerunning all the scaling to make it work
#shotsNew$goalieScreened = as.factor(shotsNew$goalieScreened)
shotsNew$NumOffense = as.factor(shotsNew$NumOffense)
shotsNew$NumDefense = as.factor(shotsNew$NumDefense)
#shotsNew$rightHanded = as.factor(shotsNew$rightHanded)
shotsQ3 <- shotsNew
shotsNew <- one_hot(dt = as.data.table(shotsNew))
shotsNew <- scale(shotsNew)

custom.config2 <- umap.defaults
custom.config2$random_state <- 10112023
```

Analysis: Methods and Results

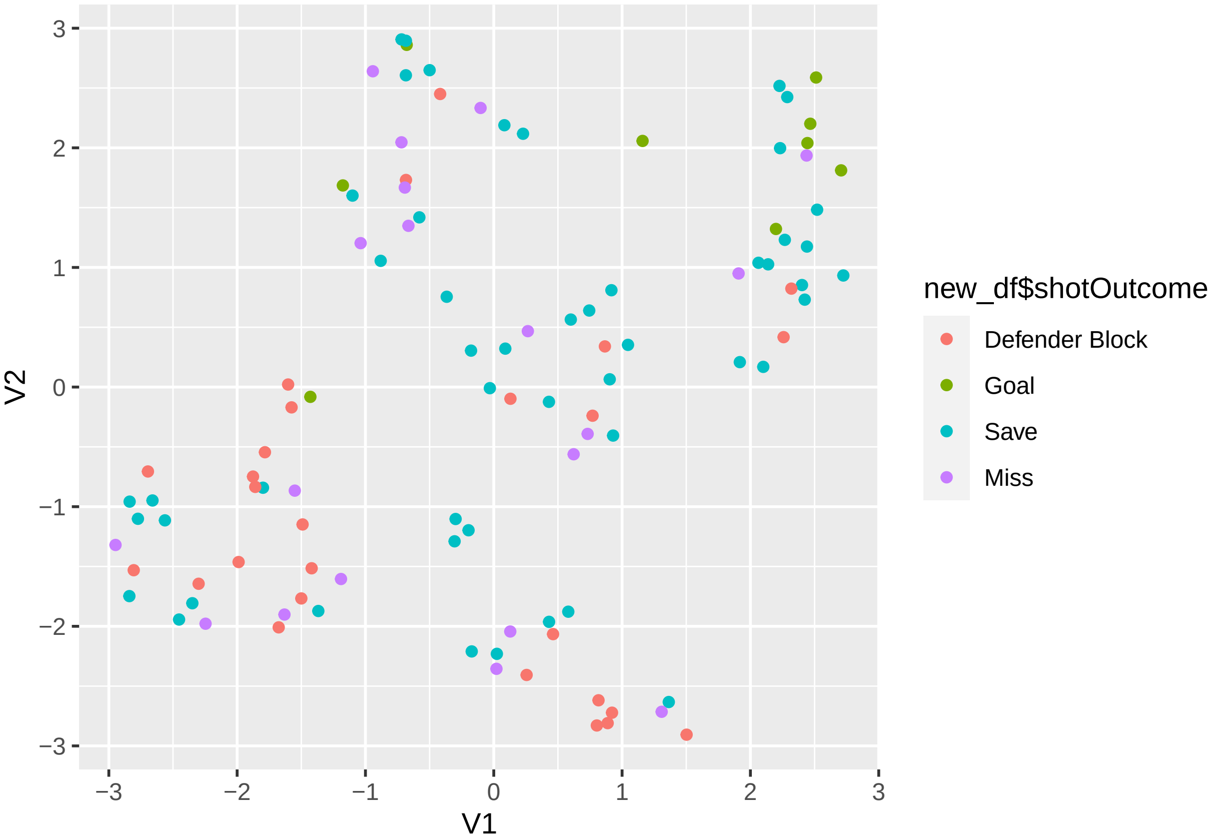
Describe in natural language a statement of the analysis you're trying to do

Rerunning analysis I did for question 1 with the new data frame

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

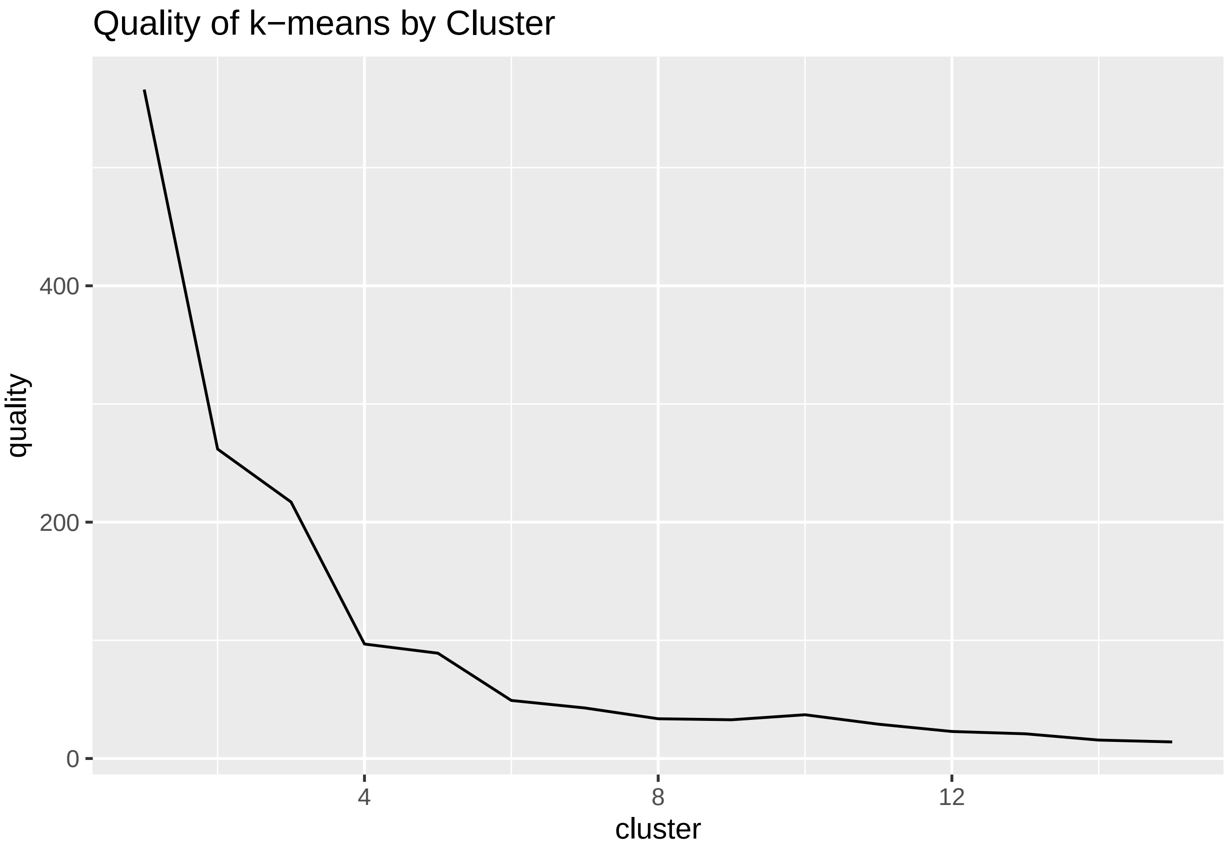
```
# Include all analysis code, clearly commented
# If not possible, screen shots are acceptable.
# If your contributions included things that are not done in an R-notebook,
#   (e.g. researching, writing, and coding in Python), you still need to do
#   this status notebook in R.  Describe what you 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.
newMap <- umap(shotsNew,custom.config2)
```

```r
ggplot()+
 geom_point(data = as.data.frame(newMap$layout),aes(x = V1,y = V2,color = new_df$shotOutcome))
```



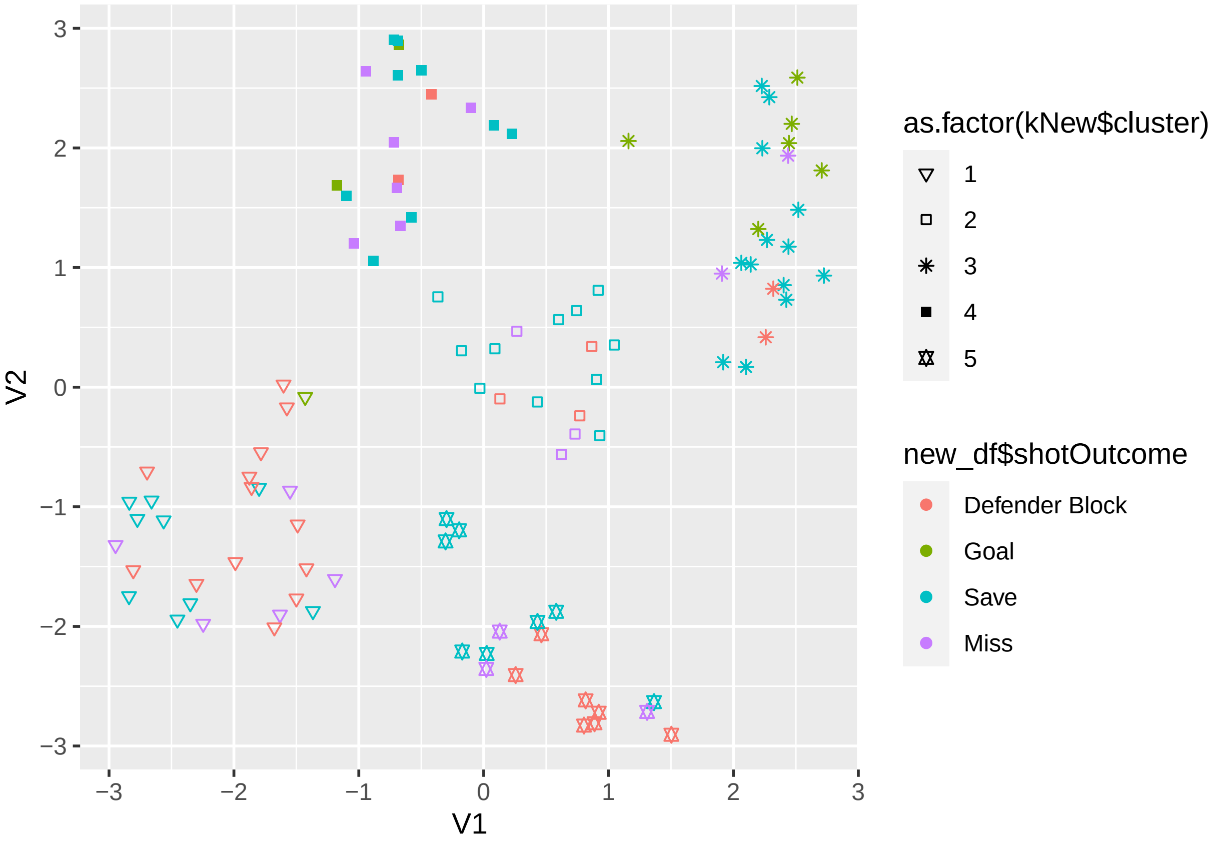
Above plot is a plot of the UMAP run with the new data frame. I ran it with a different seed, but is seems fairly consistent between seeds. Not nearly as nice as the previous UMAP with clusters, although there's still a bunch of goals hanging out in the corner which is good to see

```r
wssplot(newMap$layout,15,100)
```



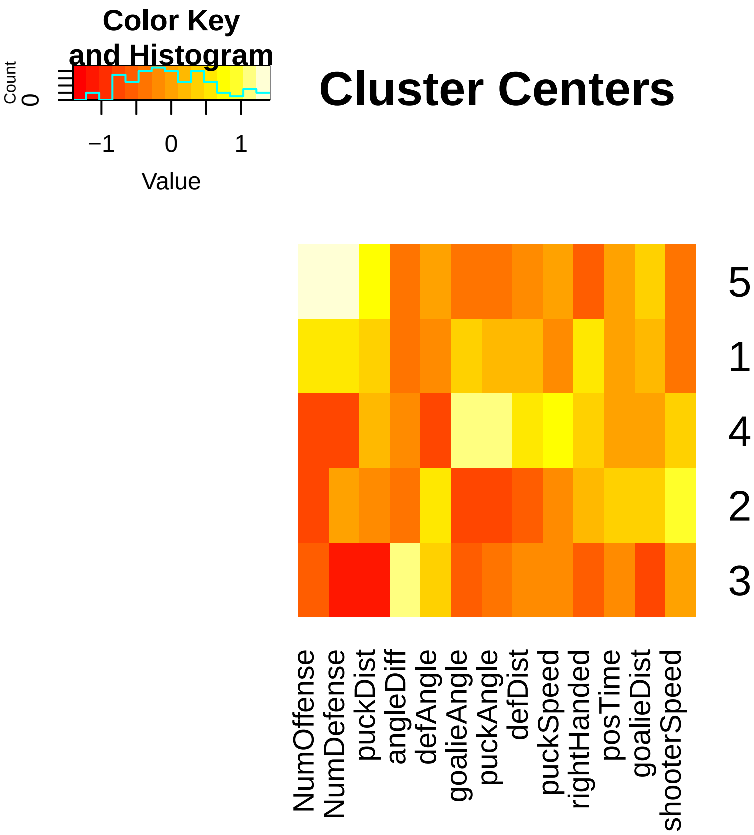
Plot of the cluster differences for the elbow test. Four and six are the main candidates. Four looks ok I guess, six looks great except it has one cluster with only 8 shots in it, so I also tried 5 to see if that would get rid of it, and it did. As a result, it is the one I'm going with

```
kNew <- kmeans(newMap$layout,5)
ggplot()+
  geom_point(data = as.data.frame(newMap$layout),aes(x = V1,y = V2,shape = as.factor(kNew$cluster),colo
  scale_shape_manual(values = c(25,22,8,15,11,12))
```



Five clusters makes the most reasonable looking graph. Unfortunately, it still isn't quite as nice as earlier. Now I'll run a similar analysis on the clusters as I did previously

```
shotsNumNew <- as.data.frame(scale(apply(shotsNumNew,2,as.numeric)))
preimageCentersNew <- shotsNumNew%>% group_by(as.factor(kNew$cluster) ) %>% summarise_all(.funs =  mean
  #apply(fails,2,median) #do this for each factor
heatDataNew <- as.matrix(preimageCentersNew[,2:(ncol(preimageCentersNew))])
#head(heatDataNew)
heatmap.2(heatDataNew,
          main = "Cluster Centers",
          scale = "none",
          trace = "none",
          dendrogram = "none",
          Colv = TRUE,
          Rowv = TRUE,
          margins = c(8,16))
```



Heatmap will be discussed under “Discussion of Results,” along with the summary table, which one again is the means for each cluster

```
sumTab$wasSuccess <- ifelse(new_df$outcomes == 2,1,0)
sumTab$wasDB <- ifelse(new_df$shotOutcome == "Defender Block",1,0)
sumTab$wasMiss <- ifelse(new_df$shotOutcome == "Miss",1,0)
summaryTable <- sumTab %>% group_by(as.factor(kNew$cluster)) %>% summarise_all(.funs = mean)
kbl(summaryTable[,1:8],booktabs = T) %>% kable_styling(latex_options = c("striped","scaled_down"),full_
```

as.factor(kNew$cluster)	puckDist	puckAngle	puckSpeed	shooterSpeed	goalieDist	goalieAngle	posTime
1	480.2189	92.74279	37.93688	12.24698	73.54123	89.42210	30.03571
2	352.1640	58.75496	37.85715	22.76281	77.99395	49.67319	36.35294
3	204.9807	73.83406	39.00433	15.23265	48.45446	56.86832	23.04348
4	444.3973	125.80491	54.42665	18.94310	70.50842	124.21303	27.10526
5	544.7923	72.74759	40.63425	12.48367	78.61590	61.27511	30.83333

```
kbl(summaryTable[,9:ncol(summaryTable)],booktabs = T) %>% kable_styling(latex_options = c("striped","sca
```

NumOffense	NumDefense	rightHanded	defDist	defAngle	angleDiff	wasSuccess	wasDB	wasMiss
0.8928571	2.535714	0.7500000	206.8147	86.95016	12.77057	0.0357143	0.4642857	0.1785714
0.0000000	2.000000	0.5294118	103.6250	133.28207	12.28638	0.0000000	0.1764706	0.1764706
0.1304348	1.086957	0.1739130	165.5673	121.62466	51.78782	0.2608696	0.0869565	0.0869565
0.0000000	1.473684	0.6315789	277.6410	54.77008	15.03603	0.1052632	0.1052632	0.3157895
1.5000000	3.222222	0.1666667	169.2922	101.53911	12.63375	0.0000000	0.3888889	0.1666667

Discussion of results

Provide in natural language a clear discussion of your observations.

Cluster 1: Pass to the enemy team. About half of these shots were defender blocks, I'm going to add the angle difference between the closest defender and the shot angle for the next analysis, as the mean defender angle and the mean puck angle were way closer here than in any of the other clusters.

Cluster 2: Defensive pressure/panic shots. While they were at a pretty decent distance, they held onto the puck for the longest of any of the clusters. Additionally, there were 2 defenders and no offense, and the defenders tended to be pretty close. Over 2/3s of these were caught by the goalie, more than normal, so I think these were the panic shots

Cluster 3: Perfect shot. Everything was aligned. On average, the goalie was off by 50 degrees! The shooter was very close, there were very few defensive players involved, the shooter took the shot quickly, and there also happened to be left handed

Cluster 4: Wide open. The nearest defender was over 277 units away, further than any of the other clusters. They also had a slightly better shot at the goal, as the difference in angle between the goalie and the shooter was 15 degrees, slightly higher than normal. As a result, they were able to achieve a 10% success rate. Most of the time the goalie blocked it, although they had a lot of misses as well. On the bright side, they also hit the puck really hard and got it fast. Not sure if this or being over 30 degrees to the right of the goalie was causing these misses

Cluster 5: "Traffic Jam" - Thanks for the name Ashley. We saw this several weeks earlier, but now that we have DB we can confirm it. They are shooting from very far out with a lot of other players involved (on average, 1.5 offense and over 3 defense). As a result, over a third of the time a defender blocks it.

Analysis: Question 3: Defender angles

Question being asked

Provide in natural language a statement of what question you're trying to answer

Is one of the clusters being caused by players shooting directly into the defenders?

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

I'm using the data and proccessing from question 2, but I'm adding in the difference in angle between the shot and the nearest defender

```r
# Include all data processing code (if necessary), clearly commented
shotsQ3$defAngleDiff <- abs(new_df$defAngle - new_df$puckAngle)
tableQ3 <- shotsQ3
sumTabQ3 <- shotsQ3
shotsQ3 <- one_hot(dt = as.data.table(shotsQ3))
shotsQ3 <- scale(shotsQ3)

sumTabQ3$wasSuccess <- ifelse(new_df$outcomes == 2,1,0)
sumTabQ3$wasDB <- ifelse(new_df$shotOutcome == "Defender Block",1,0)
sumTabQ3$wasMiss <- ifelse(new_df$shotOutcome == "Miss",1,0)
sumTabQ3$NumDefense <- as.numeric(sumTabQ3$NumDefense)
sumTabQ3$NumOffense <- as.numeric(sumTabQ3$NumOffense)
```

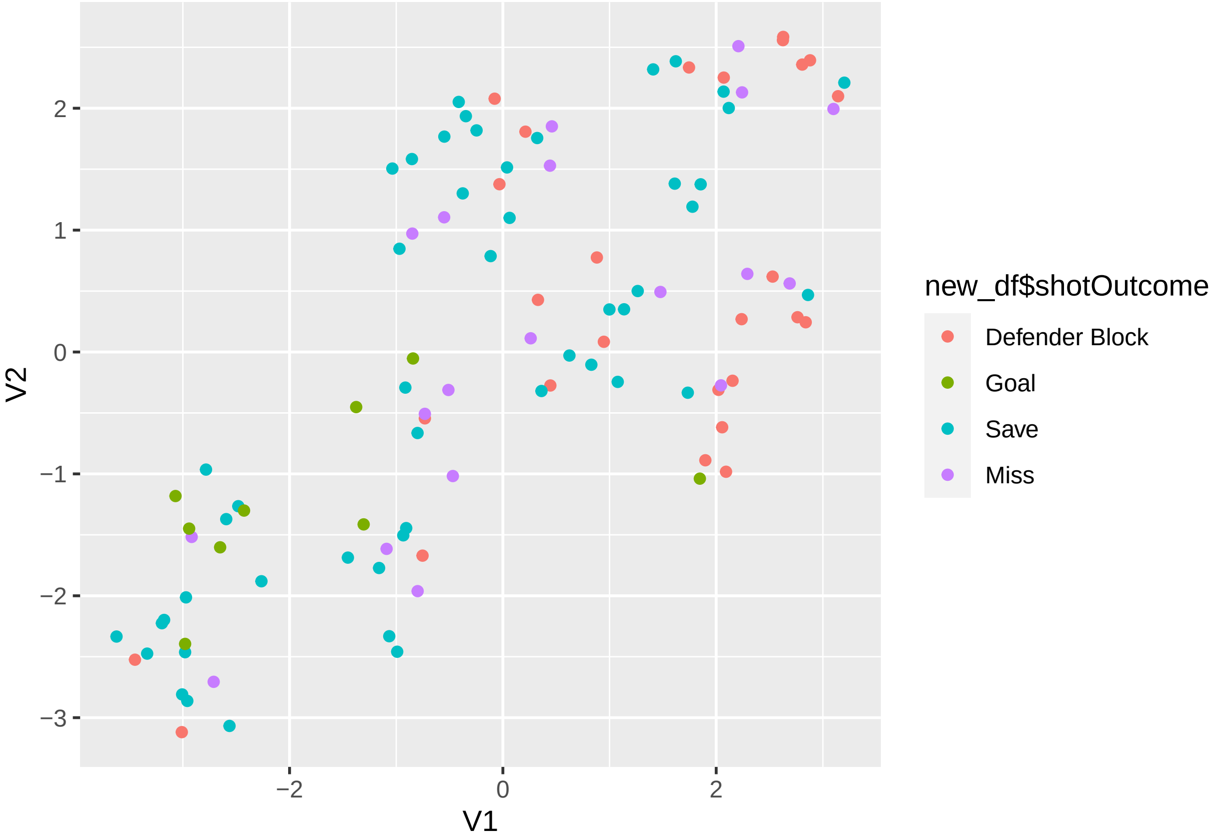
Analysis methods used

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

The same analysis from question 2 with the added feature

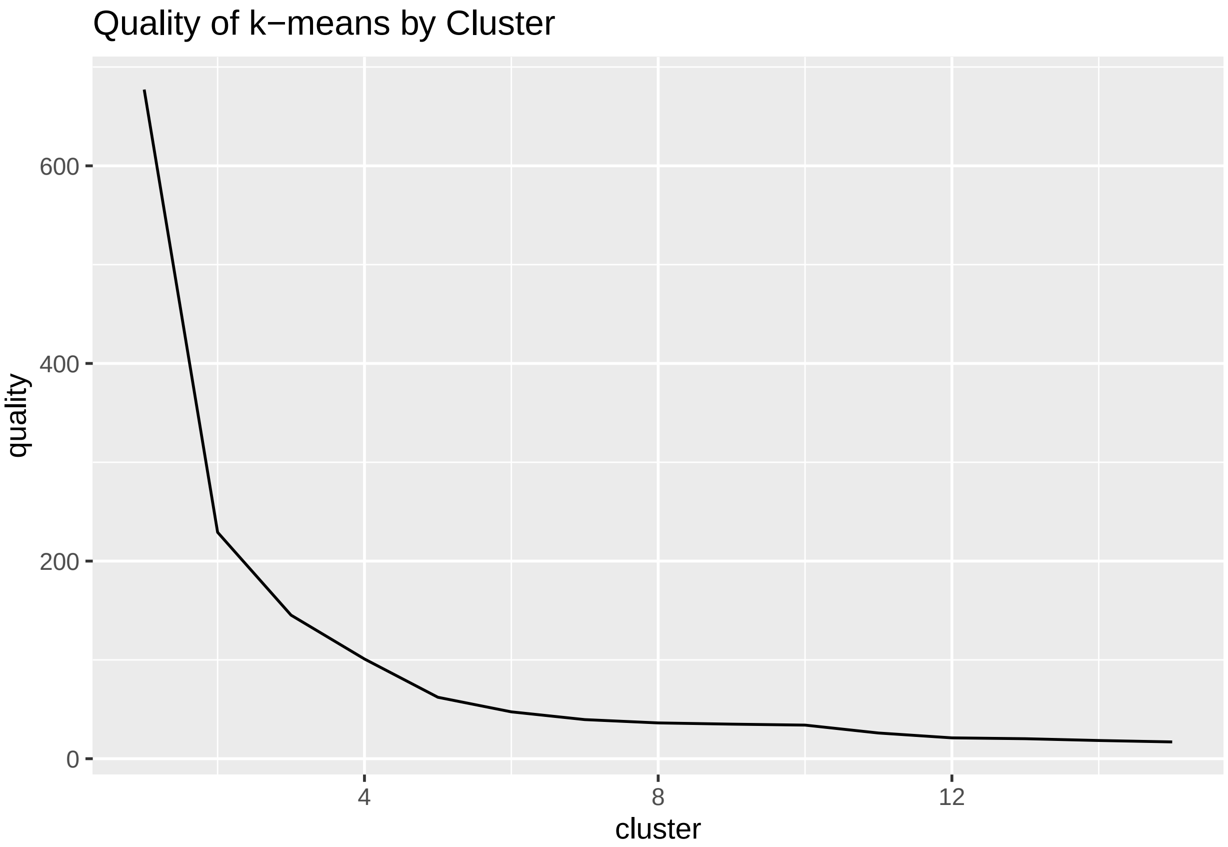
Provide clearly commented analysis code; include code for tables and figures!

```
# 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.
map3 <- umap(shotsQ3,custom.config)

ggplot()+
 geom_point(data = as.data.frame(map3$layout),aes(x = V1,y = V2,color = new_df$shotOutcome))
```

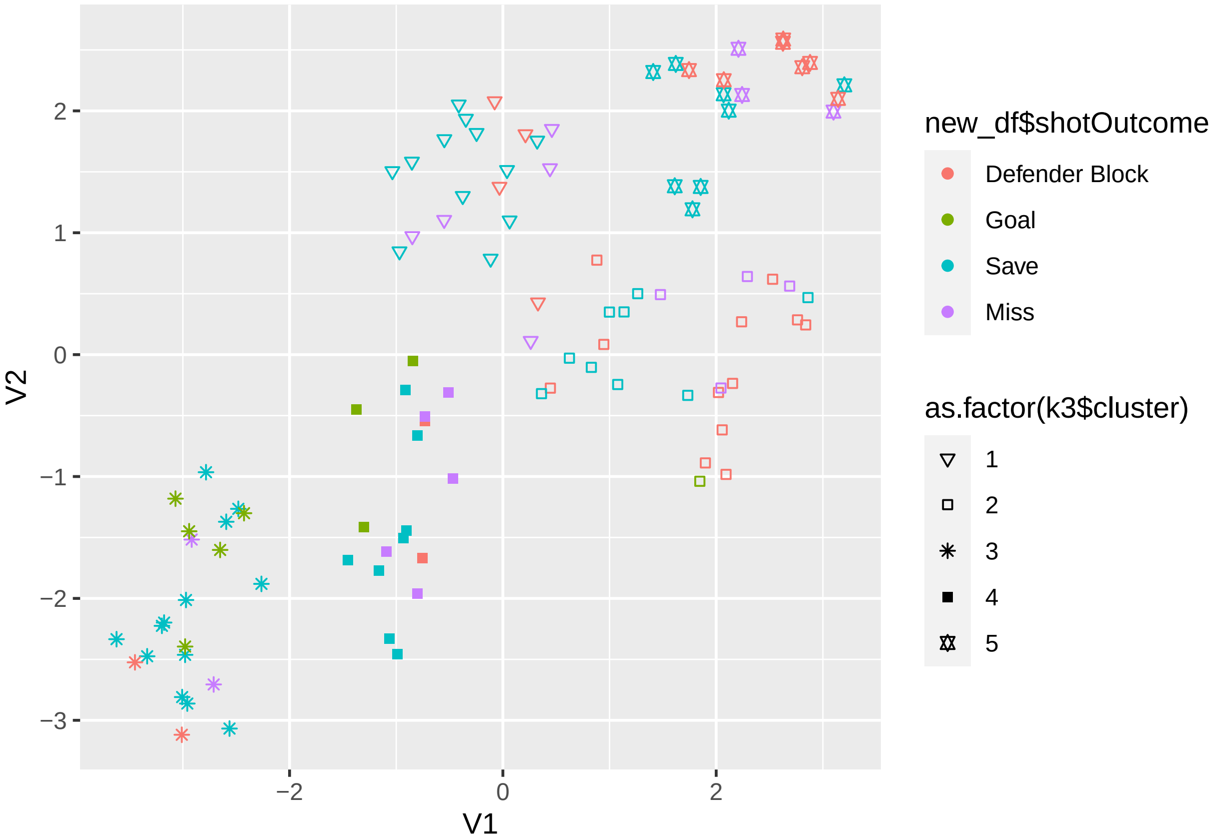


This didn't really cluster nicely, but I'll see if K-means agrees with me on that front

```
wssplot(map3$layout)
```

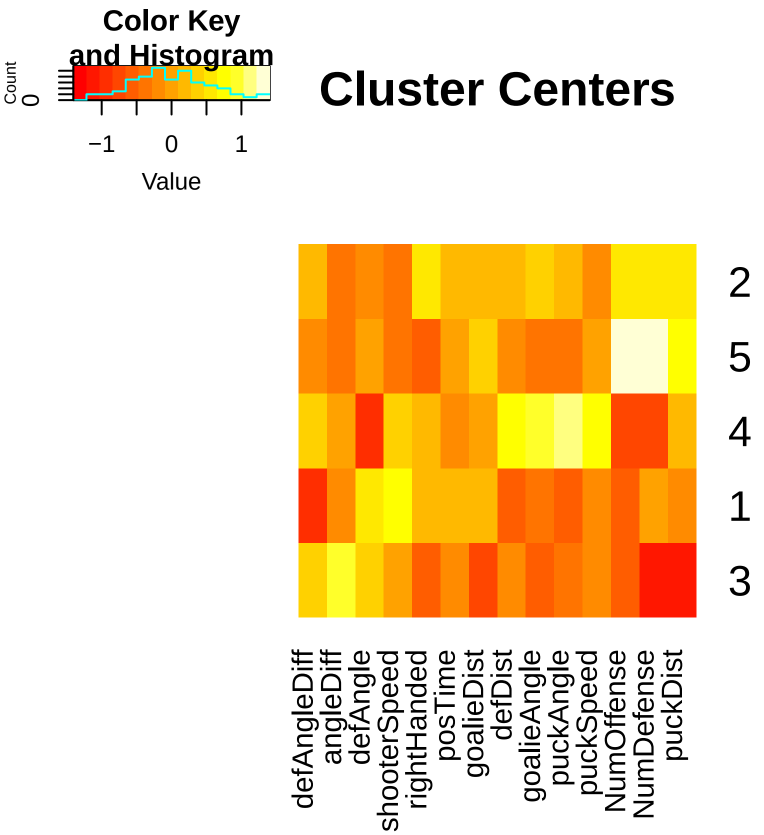


```
#5 clusters?
k3 <- kmeans(map3$layout,5)
ggplot()+
  geom_point(data = as.data.frame(map3$layout),aes(x = V1,y = V2,shape = as.factor(k3$cluster),color = r
  scale_shape_manual(values = c(25,22,8,15,11,12))
```



That actually doesn't look that bad, although it does split the goals into two seperate clusters. Heatmap and table of means time!

```
shotsNumNew <- as.data.frame(scale(apply(tableQ3,2,as.numeric)))
preimageCentersNew <- shotsNumNew%>% group_by(as.factor(k3$cluster) ) %>% summarise_all(.funs =  mean)
  #apply(fails,2,median) #do this for each factor
heatDataNew <- as.matrix(preimageCentersNew[,2:(ncol(preimageCentersNew))])
#head(heatDataNew)
heatmap.2(heatDataNew,
          main = "Cluster Centers",
          scale = "none",
          trace = "none",
          dendrogram = "none",
          Colv = TRUE,
          Rowv = TRUE,
          margins = c(8,16))
```



```r
sumTab$wasSuccess <- ifelse(new_df$outcomes == 2,1,0)
sumTab$wasDB <- ifelse(new_df$shotOutcome == "Defender Block",1,0)
sumTab$wasMiss <- ifelse(new_df$shotOutcome == "Miss",1,0)
summaryTable <- sumTabQ3 %>% group_by(as.factor(kNew$cluster)) %>% summarise_all(.funs = mean)
kbl(summaryTable[,1:8],booktabs = T) %>% kable_styling(latex_options = c("striped","scaled_down"),full_
```

as.factor(kNew$cluster)	puckDist	puckAngle	puckSpeed	shooterSpeed	goalieDist	goalieAngle	posTime
1	480.2189	92.74279	37.93688	12.24698	73.54123	89.42210	30.03571
2	352.1640	58.75496	37.85715	22.76281	77.99395	49.67319	36.35294
3	204.9807	73.83406	39.00433	15.23265	48.45446	56.86832	23.04348
4	444.3973	125.80491	54.42665	18.94310	70.50842	124.21303	27.10526
5	544.7923	72.74759	40.63425	12.48367	78.61590	61.27511	30.83333

```r
kbl(summaryTable[,9:17],booktabs = T) %>% kable_styling(latex_options = c("striped","scaled_down"),full_
```

NumOffense	NumDefense	rightHanded	defDist	defAngle	angleDiff	defAngleDiff	wasSuccess	wasDB
1.892857	2.535714	0.7500000	206.8147	86.95016	12.77057	84.15031	0.0357143	0.4642857
1.000000	2.000000	0.5294118	103.6250	133.28207	12.28638	74.52711	0.0000000	0.1764706
1.130435	1.086957	0.1739130	165.5673	121.62466	51.78782	86.49131	0.2608696	0.0869565
1.000000	1.473684	0.6315789	277.6410	54.77008	15.03603	80.49837	0.1052632	0.1052632
2.500000	3.222222	0.1666667	169.2922	101.53911	12.63375	79.58732	0.0000000	0.3888889

```r
kbl(summaryTable[,18:ncol(summaryTable)],booktabs = T) %>% kable_styling(latex_options = c("striped","s
```

wasMiss
0.1785714
0.1764706
0.0869565
0.3157895
0.1666667

Discussion of results

Provide in natural language a clear discussion of your observations.

It produced basically the same table as previously, there wasn't anything added by putting the defender angle difference in. I think this is because the closest defender is not necessarily the person best in position to block the shot. This analysis doesn't reveal any new information.

Summary and next steps

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

Overall, there appears to be 5 separate clusters as outlined in Question 2, with one major goal cluster and another that has a few clusters. For next steps, I could work on making this analysis more presentable to a wider audience by improving graphs and also potentially working with some new data, as apparently there is an updated data frame that will be pushed tonight.