College football teams 2021 season analysis

1. Introduction

The dataset chosen for this project is the team statistics of the 2021 NCAA Division I FBS football season. This dataset contains various variables about the team's summarizing statistics for all the games they have played, containing aspects of offense, defense, special teams and so on. Each observation represents a team.

The datasets were acquired from Kaggle website. Link: https://www.kaggle.com/datasets/jeffgallini/college-football-team-stats-2019 (https://www.kaggle.com/datasets/jeffgallini/college-football-team-stats-2019). The topic and dataset are selected due to author's interest in college football. Data in year 2021 were specifically extracted due to it being the most recent college football season (and Texas did not perform particularly well…).

We are going to apply several classification and prediction methods on the dataset. Variables among the same "category", such as offense and defense, are expected to have more correlations than the others. Other potential relationships might be found between defense and turnovers (better defense might create more turnovers).

```
# Load the data
library(readr)

team <- read_csv("C:\\Users\\Yukun\\Desktop\\cfb21.csv")
```

```
## New names:
## * `` -> ...1
```

```
## Rows: 130 Columns: 152
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr   (4): Team, Win-Loss, Time of Possession, Average Time of Possession pe...
## dbl (137): ...1, Games, Off Rank, Off Yards/Play, Off TDs, Off Yards per Gam...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
head(team)
```

```
## # A tibble: 6 x 152
##    ...1 Team                 Games `Win-Loss` `Off Rank` `Off Plays` `Off Yards`
##   <dbl> <chr>                <dbl> <chr>           <dbl>       <dbl>       <dbl>
## 1     1 Air Force (Mountain~    13 10-3               51         947        5502
## 2     2 Akron (MAC)             12 2-10              106         783        4089
## 3     3 Alabama (SEC)           15 13-2                7        1119        7323
## 4     4 App State (Sun Belt)    14 10-4               30         980        6178
## 5     5 Arizona (Pac-12)        12 1-11              101         870        4271
## 6     6 Arizona St. (Pac-12)    13 8-5                75         807        5018
## # ... with 145 more variables: `Off Yards/Play` <dbl>, `Off TDs` <dbl>,
## #   `Off Yards per Game` <dbl>, `Def Rank` <dbl>, `Def Plays` <dbl>,
## #   `Yards Allowed` <dbl>, `Yards/Play Allowed` <dbl>, `Off TDs Allowed` <dbl>,
## #   `Total TDs Allowed` <dbl>, `Yards Per Game Allowed` <dbl>,
## #   `First Down Rank` <dbl>, `First Down Runs` <dbl>,
## #   `First Down Passes` <dbl>, `First Down Penalties` <dbl>,
## #   `First Downs` <dbl>, `First Down Def Rank` <dbl>, ...
```

```
# check if there are any missing values in the datasets
sum(is.na(team))
```

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

The dataset is already in a tidy format and does not contain any missing values, so no tidying is needed.

2. Exploratory Data Analysis

The main variables selected for this dataset (project) include two variables about offense ("offensive yards per game" & "points per game"), two about defense ("yards allowed per game" & "points allowed per game"), and two about other aspects ("penalty yards per game" & "turnover margin per game"). All the variables are numeric.

More variables can definitely be taken into considerations, but for the sake of visualization (graphs can be too compact if too many variables), we are only taking the six above.

```
# call the packages
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
## v ggplot2 3.3.5     v dplyr   1.0.7
## v tibble  3.1.6     v stringr 1.4.0
## v tidyr   1.2.0     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.1.3
```

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

```
library(cluster)

# select main variables
team_main <- team %>%
  select(`Off Yards per Game`, `Yards Per Game Allowed`, `Points Per Game`, `Avg Points per Game Allowed`, `Penalty Yards Pe
r Game`, `Avg Turnover Margin per Game`)

head(team_main)
```

```
## # A tibble: 6 x 6
##   `Off Yards per Game` `Yards Per Game Allow~` `Points Per Ga~` `Avg Points pe~`
##                  <dbl>                   <dbl>            <dbl>            <dbl>
## 1                 423.                    296.             31               19.8
## 2                 341.                    469              19.8             39.5
## 3                 488.                    304.             39.9             20.1
## 4                 441.                    348.             34.5             22.1
## 5                 356.                    371              17.2             31.4
## 6                 386                     326              28.4             20.8
## # ... with 2 more variables: `Penalty Yards Per Game` <dbl>,
## #   `Avg Turnover Margin per Game` <dbl>
```

```
# build a correlation matrix with correlation coefficients
cor(team_main, use = "pairwise.complete.obs")
```

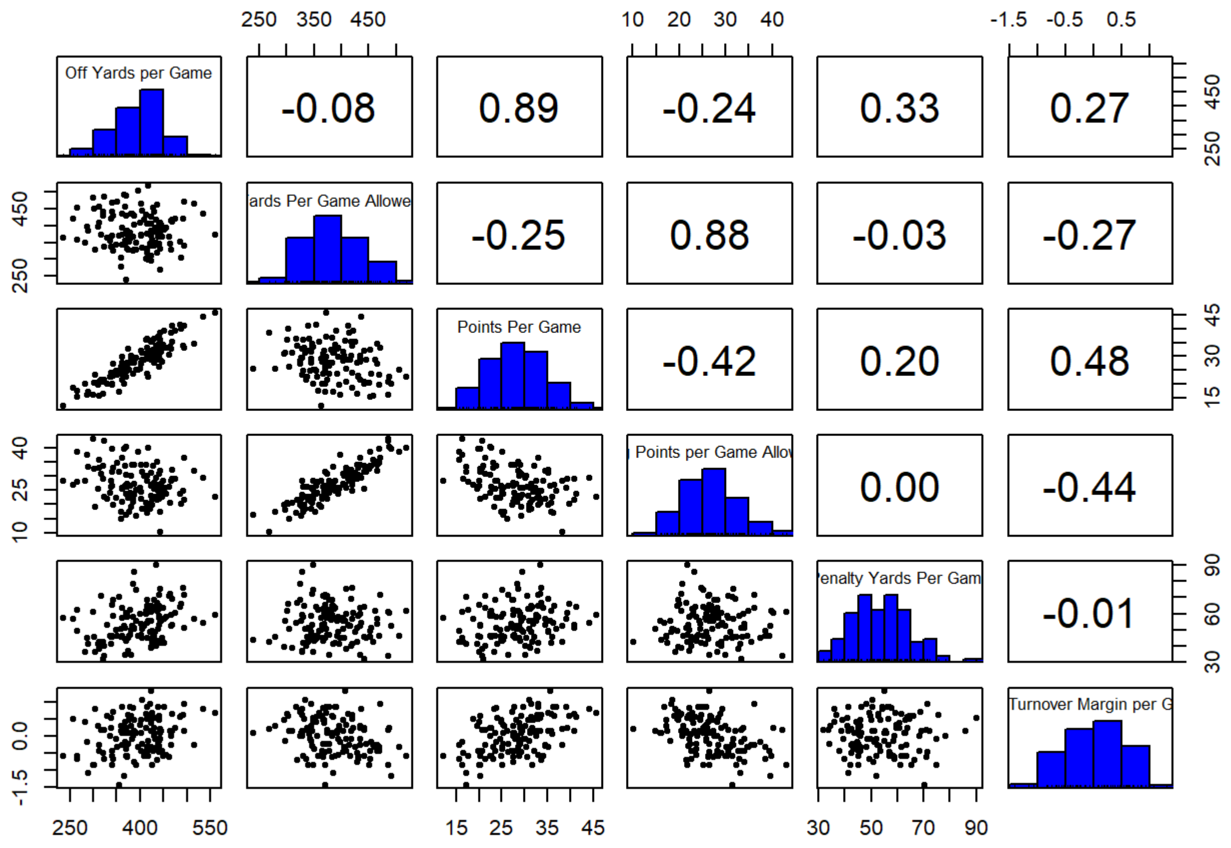
```
##                               Off Yards per Game Yards Per Game Allowed
## Off Yards per Game                     1.0000000           -0.07918910
## Yards Per Game Allowed                -0.0791891            1.00000000
## Points Per Game                        0.8875336           -0.25483215
## Avg Points per Game Allowed           -0.2426045            0.88169023
## Penalty Yards Per Game                 0.3302085           -0.03393834
## Avg Turnover Margin per Game           0.2744280           -0.27202720
##                               Points Per Game Avg Points per Game Allowed
## Off Yards per Game                  0.8875336                -0.242604538
## Yards Per Game Allowed             -0.2548321                 0.881690235
## Points Per Game                     1.0000000                -0.417230629
## Avg Points per Game Allowed        -0.4172306                 1.000000000
## Penalty Yards Per Game              0.2030106                -0.003560831
## Avg Turnover Margin per Game        0.4771059                -0.436219278
##                               Penalty Yards Per Game
## Off Yards per Game                       0.330208526
## Yards Per Game Allowed                  -0.033938343
## Points Per Game                          0.203010557
## Avg Points per Game Allowed             -0.003560831
## Penalty Yards Per Game                   1.000000000
## Avg Turnover Margin per Game            -0.007112996
##                               Avg Turnover Margin per Game
## Off Yards per Game                             0.274427974
## Yards Per Game Allowed                        -0.272027200
## Points Per Game                                0.477105850
## Avg Points per Game Allowed                   -0.436219278
## Penalty Yards Per Game                        -0.007112996
## Avg Turnover Margin per Game                   1.000000000
```

```
# build a correlation matrix with univariate and bivariate graphs
library(psych)
```

```
## Warning: package 'psych' was built under R version 4.1.3
```

```
##
## Attaching package: 'psych'
## The following objects are masked from 'package:ggplot2':
##
##     %+%, alpha
```

```
pairs.panels(team_main,
             method = "pearson", # correlation coefficient method
             hist.col = "blue", # color of histogram
             smooth = FALSE, density = FALSE, ellipses = FALSE)
```



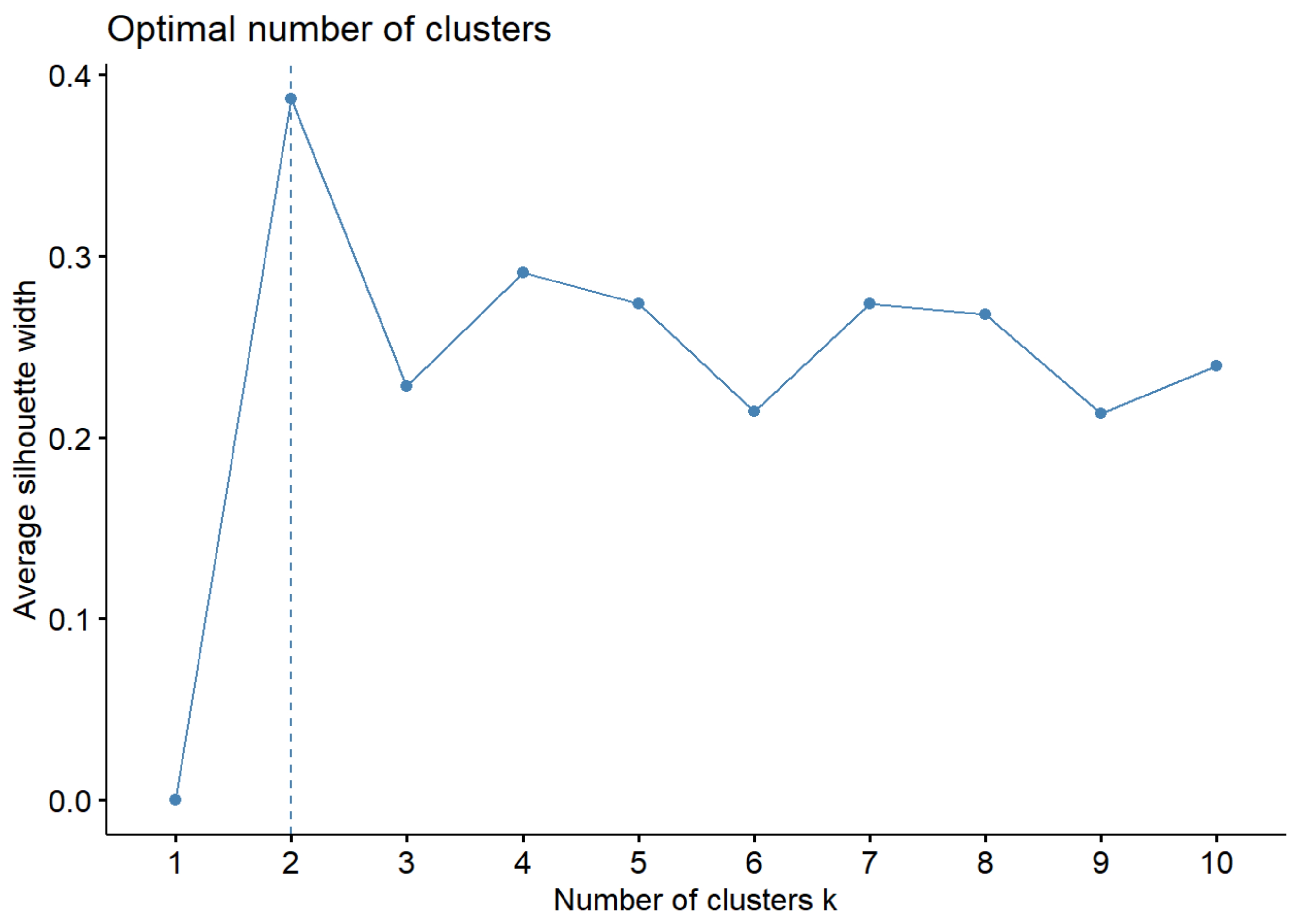
The most correlated pairs of variables are: 1) “offensive yards per game” & “points gained per game”, and 2) “yards allowed per game” & “points allowed per game”. Both of the pairs are positively correlated and significantly more correlated than other pairs. This makes sense since one pair contains variables directly related to offense, and the other is on defense.

3. Clustering

In this part, we will apply PAM clustering on four variables: “offensive yards per game”, “points gained per game”, “yards allowed per game” and “points allowed per game”. These are all numeric variables.

The Gower dissimilarities part is omitted here since the categorical variable contains too many layers and exceeds the threshold of visualization graph.

```r
# prepare data (select variables and scale them)
team_main2 <- team_main %>%
  select(`Off Yards per Game`, `Yards Per Game Allowed`, `Points Per Game`, `Avg Points per Game Allowed`) %>%
  scale

# determine the number of custers based on silhouette width
fviz_nbclust(team_main2, pam, method = "silhouette")
```



The average silhouette width indicates that 2 clusters are sufficient.

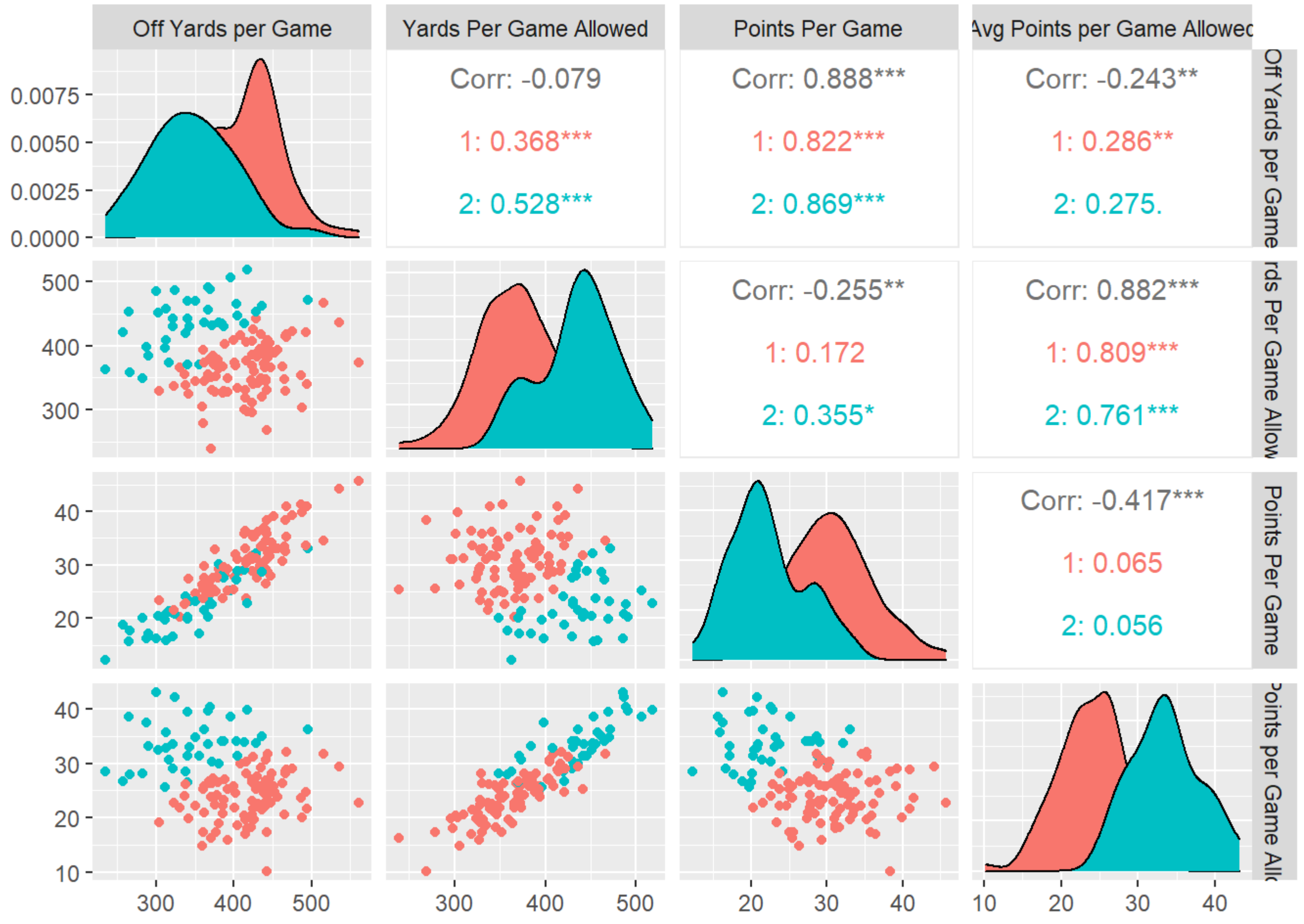
```
# perform PAM clustering algorithm, with 2 clusters
pam_results <- team_main2 %>%
  pam(k = 2)

# save cluster assignment as a column in the dataset
team_pam <- team_main %>%
  mutate(cluster = as.factor(pam_results$clustering))
```

```
# visualize all pairwise combinations of variables using ggpairs
library(GGally)
```

```
## Warning: package 'GGally' was built under R version 4.1.3
```

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

```
ggpairs(team_pam, columns = 1:4, aes(color = cluster))
```



- Visualizing clusters in a 2-dimension plot is included in the next “dimensionality reduction” part.

```
# find the how many observations each cluster contains
team_pam %>%
  group_by(cluster) %>%
  count()
```

```
## # A tibble: 2 x 2
## # Groups:   cluster [2]
##   cluster     n
##   <fct>   <int>
## 1 1          91
## 2 2          39
```

```
# find the means of each variable for each cluster
team_pam %>%
  group_by(cluster) %>%
  summarize_if(is.numeric, mean, na.rm = T)
```

```
## # A tibble: 2 x 7
##   cluster `Off Yards per Ga~` `Yards Per Gam~` `Points Per Ga~` `Avg Points pe~`
##   <fct>                 <dbl>            <dbl>            <dbl>            <dbl>
## 1 1                      418.             364.             30.9             23.8
## 2 2                      346.             436.             22.1             33.6
## # ... with 2 more variables: `Penalty Yards Per Game` <dbl>,
## #   `Avg Turnover Margin per Game` <dbl>
```

```
# Which teams are more representative of their cluster? Let's look at the final medoids
team[pam_results$id.med,]
```

```
## # A tibble: 2 x 152
##    ...1 Team                 Games `Win-Loss` `Off Rank` `Off Plays` `Off Yards`
##   <dbl> <chr>                <dbl> <chr>           <dbl>       <dbl>       <dbl>
## 1    58 Miami (OH) (MAC)        13 7-6                52         879        5500
## 2   104 Texas St. (Sun Belt)    12 4-8               104         829        4112
## # ... with 145 more variables: `Off Yards/Play` <dbl>, `Off TDs` <dbl>,
## #   `Off Yards per Game` <dbl>, `Def Rank` <dbl>, `Def Plays` <dbl>,
## #   `Yards Allowed` <dbl>, `Yards/Play Allowed` <dbl>, `Off TDs Allowed` <dbl>,
## #   `Total TDs Allowed` <dbl>, `Yards Per Game Allowed` <dbl>,
## #   `First Down Rank` <dbl>, `First Down Runs` <dbl>,
## #   `First Down Passes` <dbl>, `First Down Penalties` <dbl>,
## #   `First Downs` <dbl>, `First Down Def Rank` <dbl>, ...
```

```
# find the average silhouette width
pam_results$silinfo$avg.width
```

```
## [1] 0.3865166
```

Based on the counts and medoids of two clusters, especially their “win-loss” records, one cluster might represent teams who are “bowl eligible” and other one represents teams who are not. *“Bowl eligible” teams must have at least 6 total wins, of all the games they played in that season.

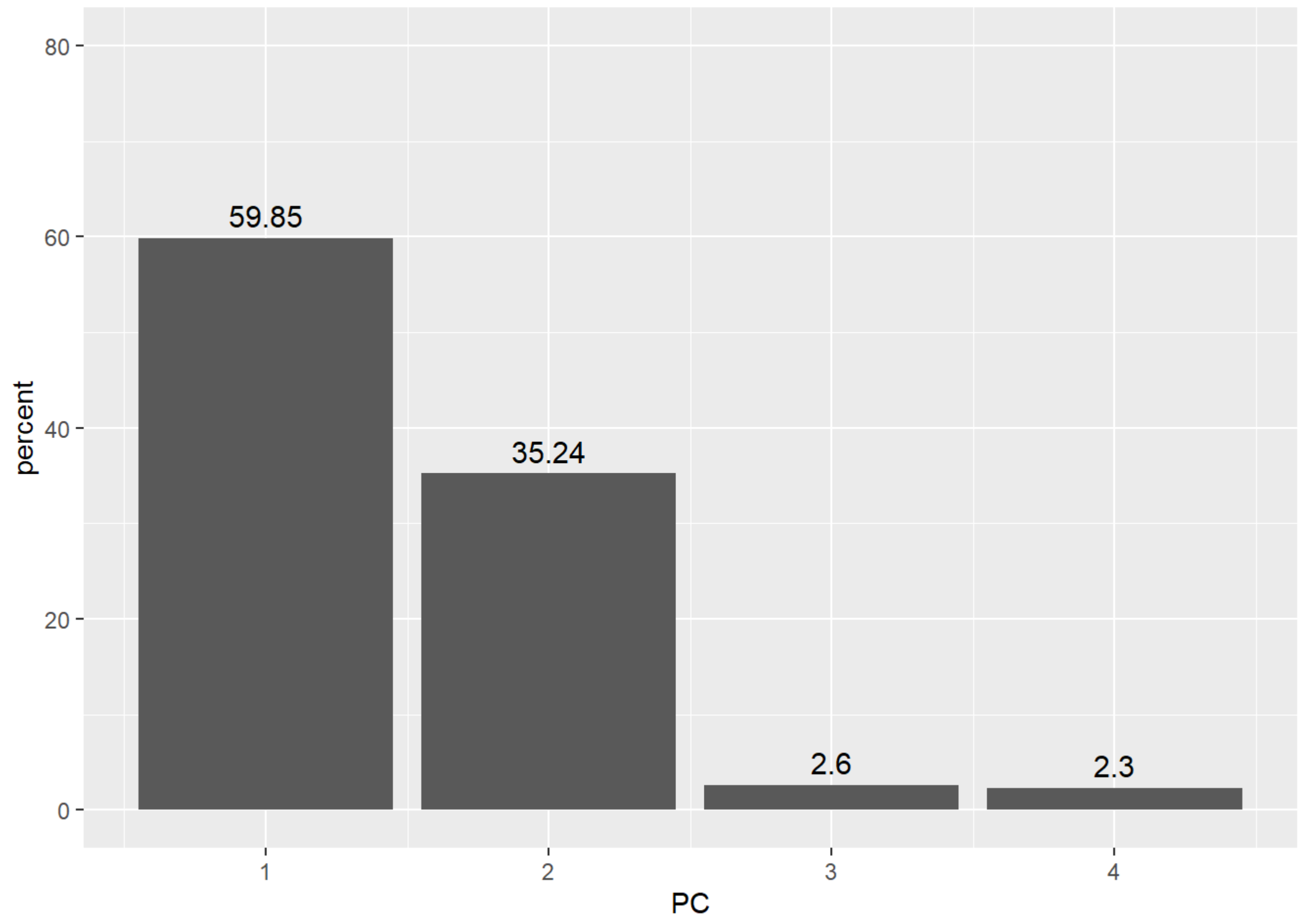
The teams who are “bowl eligible” tend to averagely have significant higher offense data and lower defense data, which means their offense and defense are both better than the other cluster. The two clusters also averagely have opposite signs on “turnover margin”, which means “bowl eligible” teams tend to force turnovers on their opponents, and “not bowl eligible teams” tend to turn the ball over to their opponents.

However, the average silhouette width is about 0.39, which is pretty low and indicates that the “structure is weak and could be artificial”.

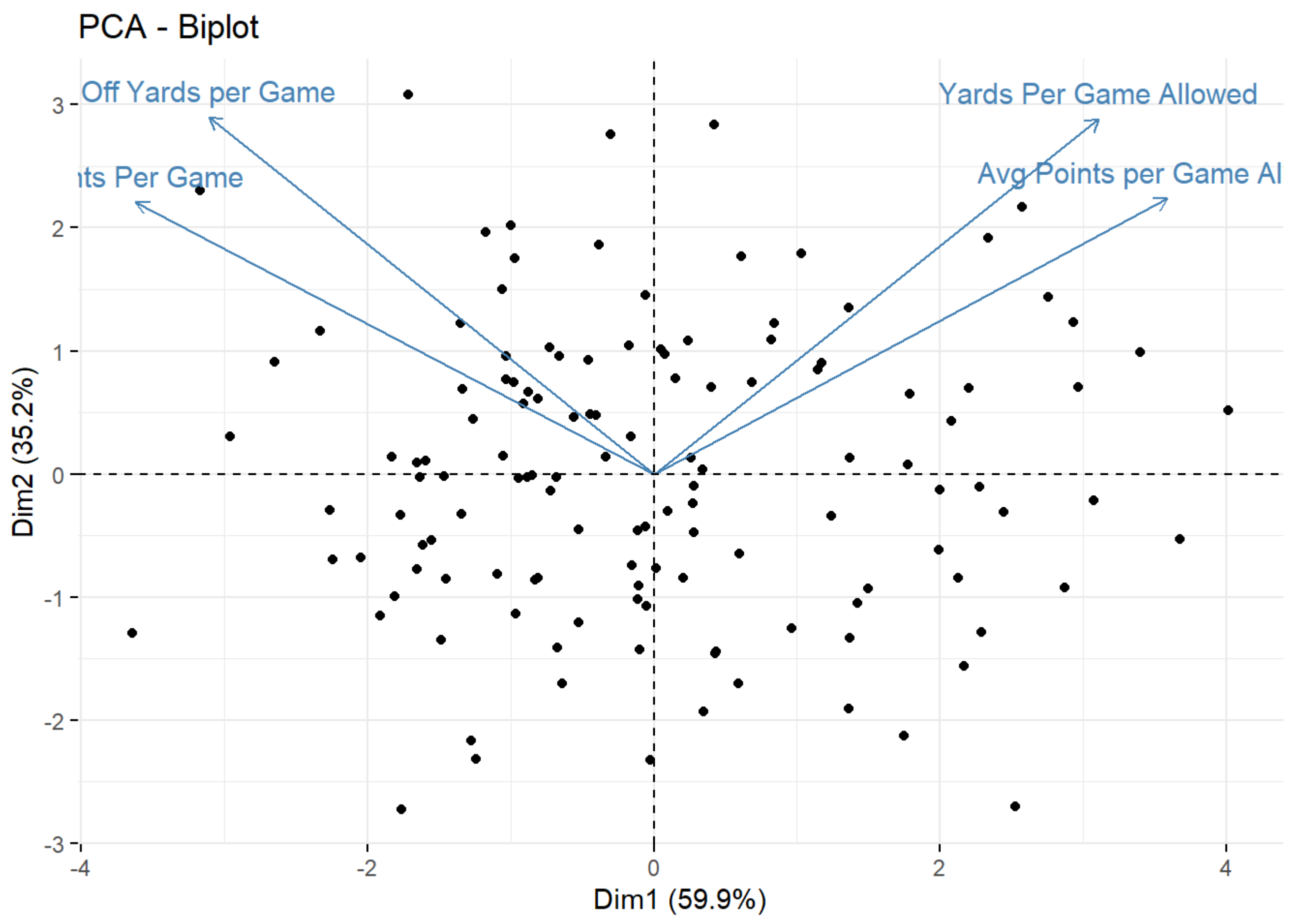
4. Dimentionality Reduction

In this part, we will apply PCA on the same four variables as above.

```r
# perform PCA with the function prcomp(), on the same variables as part 3
pca <- team_main2 %>%
  prcomp()

# save the percentage of variance explained by each component with sdev
percent <- 100 * (pca$sdev^2 / sum(pca$sdev^2))

var_explained <- data.frame(percent, PC = 1:length(percent))

# visualize the percentage of variance explained by each component
ggplot(var_explained, aes(x = PC, y = percent)) +
  geom_col() +
  geom_text(aes(label = round(percent, 2)), size = 4, vjust = -0.5) +
  ylim(0, 80)
```



```
# visualize the observations using the first 2 PCs
# use fviz_pca_biplot() function to also include variables
fviz_pca_biplot(pca, geom.ind = "point")
```



```
# take a look at the rotation matrix to see what variable contributed the most to each component
pca$rotation
```

```
##                                 PC1       PC2        PC3        PC4
## Off Yards per Game        -0.4614596 0.5621428 -0.3308363  0.6013300
## Yards Per Game Allowed     0.4618521 0.5589462  0.6604166  0.1952477
## Points Per Game           -0.5378354 0.4280502  0.2186117 -0.6926147
## Avg Points per Game Allowed  0.5333674 0.4339904 -0.6376568 -0.3472254
```

Based on the rotation matrix, we can see that all of the variables contribute positively to the second component. This means that teams with high scores on PC2 tend to have high values for all the 4 variables.

We also see that the variables of “offensive yards per game” and “points gained per game” contributed negatively to the first component while “yards allowed per game” and “points allowed per game” contributed positively. This means that teams with high scores on PC1 tend to have higher values for the 2 defense variables and lower values for the 2 offense variables.

From the variance graph, we can see that PC1 accounted for 59.85% of the total variations, PC2 35.24%, PC3 2.6% and PC4 2.3%. PC1 and PC2 combining composes most of the total variations of 95.09%.

5. Classification and Cross-validation

Since there is no binary variable in our original dataset, we will create one on our own in this part (as the response). Based on our discussion earlier, a response related to “bowl eligibility” seems reasonable. Then we will use logistic regression classifier to predict and perform k-fold cross-validation. The variables that will be used to predict are the same 4 as above.

```
# call the packages tidyverse, plotROC, and caret
library(tidyverse)
library(plotROC)
```

```
## Warning: package 'plotROC' was built under R version 4.1.3
```

```
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.3
```

```
## Loading required package: lattice
```

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

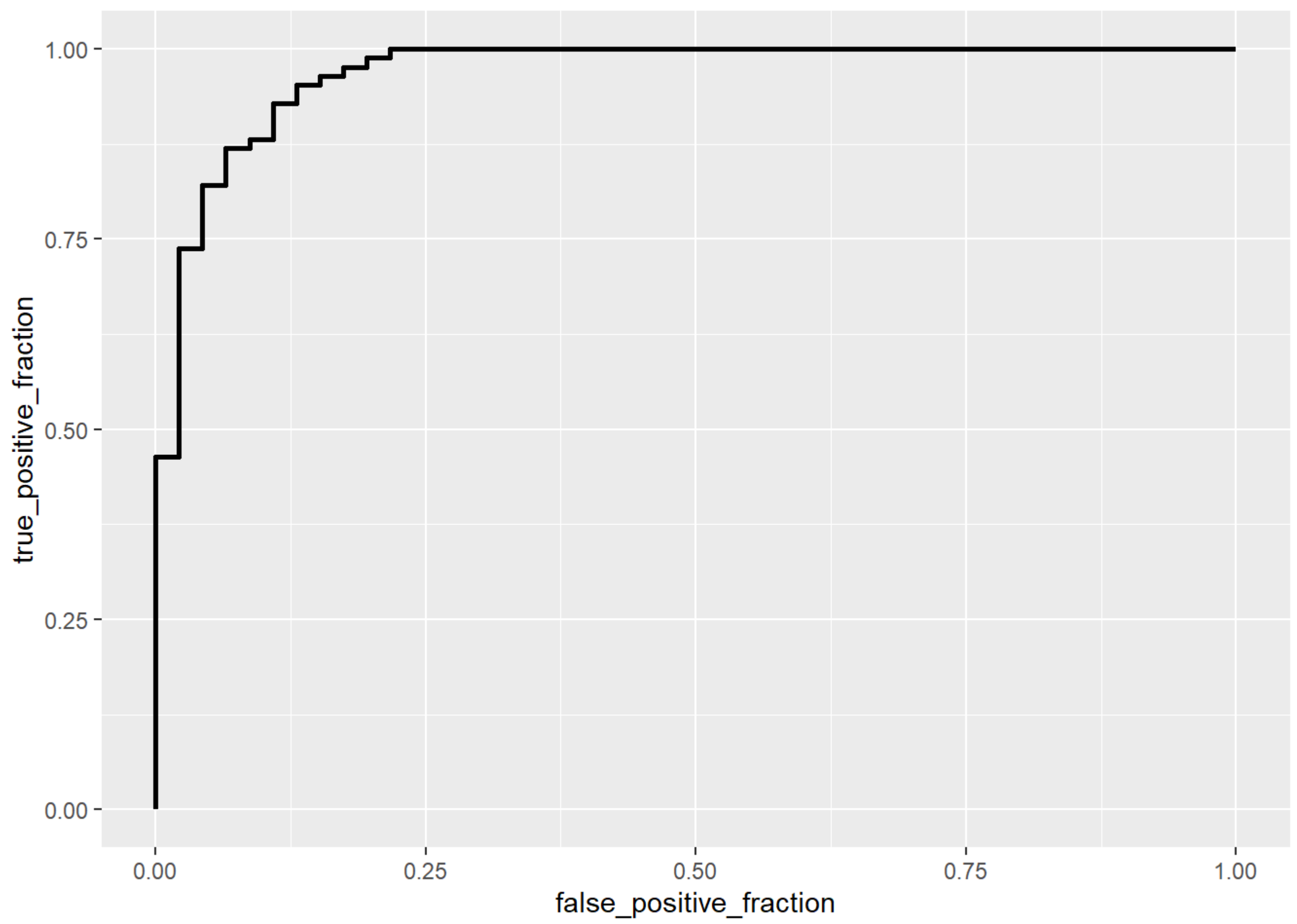
```
# create a binary variable on "bowl eligibility"
team_new <- team %>%
  mutate(`Number of wins` = as.numeric(sub("-.*", "", team$`Win-Loss`))) %>%
  mutate(`Bowl Eligibility` = case_when(`Number of wins` >= 6 ~ 1, `Number of wins` < 6 ~ 0))

# fit the logistic model with the entire dataset
fit <- glm(`Bowl Eligibility` ~ `Off Yards per Game`+`Yards Per Game Allowed`+`Points Per Game`+`Avg Points per Game Allowed
`,
           data = team_new,
           family = "binomial")

# calculate a predicted probability for bowl eligibility
log_team <- team_new %>%
  mutate(score = predict(fit, type = "response"),
         predicted = ifelse(score < 0.5, 0, 1)) %>%
  select(`Off Yards per Game`, `Yards Per Game Allowed`, `Points Per Game`, `Avg Points per Game Allowed`, `Bowl Eligibility
`, score, predicted)

head(log_team)
```

```
## # A tibble: 6 x 7
##   `Off Yards per Game` `Yards Per Game Allow~` `Points Per Ga~` `Avg Points pe~`
##                  <dbl>                   <dbl>            <dbl>            <dbl>
## 1                 423.                    296.             31               19.8
## 2                 341.                    469              19.8             39.5
## 3                 488.                    304.             39.9             20.1
## 4                 441.                    348.             34.5             22.1
## 5                 356.                    371              17.2             31.4
## 6                 386                     326              28.4             20.8
## # ... with 3 more variables: `Bowl Eligibility` <dbl>, score <dbl>,
## #   predicted <dbl>
```

```
# create a confusion matrix to compare true to predicted condition
table(log_team$`Bowl Eligibility`, log_team$predicted) %>% addmargins
```

```
##
##          0   1 Sum
##   0    40   6  46
##   1     4  80  84
##   Sum  44  86 130
```

```
# ROC curve
ROC <- ggplot(log_team) +
  geom_roc(aes(d = `Bowl Eligibility`, m = score), n.cuts = 0)
ROC
```



```
# calculate the area under the curve
calc_auc(ROC)
```

```
##   PANEL group       AUC
## 1     1    -1 0.9692029
```

AUC value indicates that we are doing great! Now let's do the k-fold cross-validation with the same classifier:

```
# choose number of folds
k = 10

# randomly order rows in the dataset
data <- team_new[sample(nrow(team_new)), ]

# create k folds from the dataset
folds <- cut(seq(1:nrow(data)), breaks = k, labels = FALSE)
```

Fit the model and repeat the process for each k-fold:

```
# use a for loop to get diagnostics for each test set
diags_k <- NULL

for(i in 1:k){
  # create training and test sets
  train <- data[folds != i, ] # all observations except in fold i
  test <- data[folds == i, ]  # observations in fold i

  # train model on training set (all but fold i)
  fit <- glm(`Bowl Eligibility` ~ `Off Yards per Game`+`Yards Per Game Allowed`+`Points Per Game`+`Avg Points per Game Allow
ed`,
             data = train,
             family = "binomial")

  # test model on test set (fold i)
  df <- data.frame(
    probability = predict(fit, newdata = test, type = "response"),
    outcome = test$`Bowl Eligibility`)

  # consider the ROC curve for the test dataset
  ROC <- ggplot(df) +
    geom_roc(aes(d = outcome, m = probability, n.cuts = 0))

  # get diagnostics for fold i (AUC)
  diags_k[i] <- calc_auc(ROC)$AUC
}
```

```
## Warning: Ignoring unknown aesthetics: n.cuts
## Ignoring unknown aesthetics: n.cuts
## Ignoring unknown aesthetics: n.cuts
## Ignoring unknown aesthetics: n.cuts
## Ignoring unknown aesthetics: n.cuts
## Ignoring unknown aesthetics: n.cuts
## Ignoring unknown aesthetics: n.cuts
## Ignoring unknown aesthetics: n.cuts
## Ignoring unknown aesthetics: n.cuts
## Ignoring unknown aesthetics: n.cuts
```

Finally, find the average performance on new data:

```
# average performance
mean(diags_k)
```

```
## [1] 0.9525
```

With scores over 90, the classifier is predicting new observations on a scale of “Great”. However, there are slight signs of overfitting since the average performance is slightly lower than the value of AUC.