SOC Final Project Paper

Denise Bradford

11/22/2020

Introduction

What NCAA football conference has the best defensive players?

Tenured football analysts, head coaches, defensive coordinators and historical football drafts tell us that the best defensive teams are likely to come from one of the power 5 NCAA conferences (ACC, SEC, Big Ten, Big 12, or PAC-12) but which conference has most defensively dominate plays compared to the others. An interesting question to ask would be whether the team has the most sacks, interceptions,interceptions for a touchdown, fumbles, fumbles for a touchdown that modifies the relationship of the teams defense.

Data Description

Data was pulled from the cfbscrapR package: cfbscrapR is an R package for working with CFB data. It is an R API wrapper around https://collegefootballdata.com/. It provides users the capability to retrieve data from a plethora of endpoints and supplement that data with additional information (Expected Points Added/Win Probability added).

We will use the *cfb_pbp_data* function that will pull data for the last 5 years, by week and play-by-play in each game from that week within the year. We will focus on the power 5 conferences along with the teams that they played. List of variables with definitions: - offense_play: the team that has possession of the ball during the play (CHR) - defense_play: the team that is defending their goal (CHR) - offense_conference/defense_conference: the conference that the team is associated with in the NCAA (CHR) - home: the home team (CHR) - away: the away team (CHR) - play_type*: each observation will have a play type description for the result of the down (CHR) - yards_gained: the number of yards that are gained during the time that the ball is in play, this value can be negative if the team moves further a way from the defenders goal. (NUM)* using spread and count functions in dplyr, we were able to count the number of times a sack, fumble, interception, etc. occurred during a game.

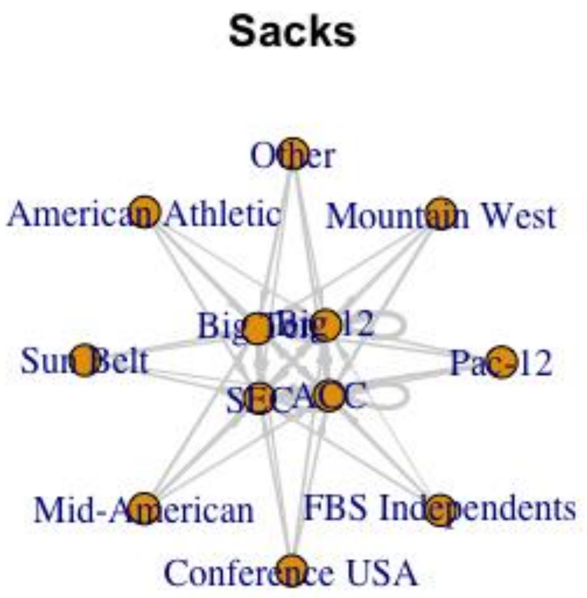
Analysis Plan

A multiblockmodel will be used to determine the best conference defensive teams. We will determine the conferences with the most sack ties, fumble ties and interception ties. If time permits, exploring the best team in each of the conferences will be determined using the blockmodel methodology. In future works, using the information from the regular session to make the conference championship, a two-mode affiliation network maybe useful to understand the "best" defense.

Preliminary Data Analysis

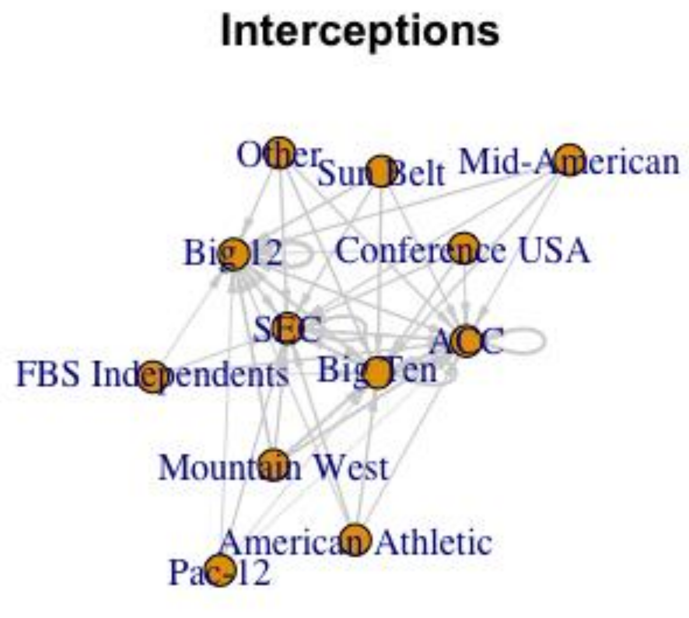
```
defense_all <- graph_from_data_frame(all_def_data2.mean, directed = TRUE)
summary(defense_all)

## IGRAPH 37e4520 DN-- 12 48 --
## + attr: name (v/c), fumble.mean (e/n), interception.mean (e/n),
## | sack.mean (e/n), weight.mean (e/n)
```

```
#Sacks
edgelist_sack=all_def_data2.mean[all_def_data2.mean$sack.mean>0, c("defense_conference",
"offense_conference", "sack.mean")]
def_sack=graph_from_data_frame(d=edgelist_sack, directed=T)
def_sack=igraph::simplify(def_sack, edge.attr.comb="mean")

sack_mat=as_adjacency_matrix(def_sack, attr="sack.mean", sparse=F)
sack_mat_in=t(sack_mat)

sack_mat_std=(sack_mat-mean(sack_mat))/sd(sack_mat)
sack_mat_in_std=t(sack_mat_std)

defense_sack <- delete.edges(defense_all, E(defense_all)[get.edge.attribute(defense_all,n
ame = "sack.mean")==0])

#par(mfrow=c(2,2))
sack_layout <- layout_nicely(defense_sack)
plot(defense_sack, layout=sack_layout, main = "Sacks", edge.arrow.size=.5, edge.arrow.wid
th=.5,
     edge.color="light gray", edge.width=E(defense_sack)$sack.mean/2)
```



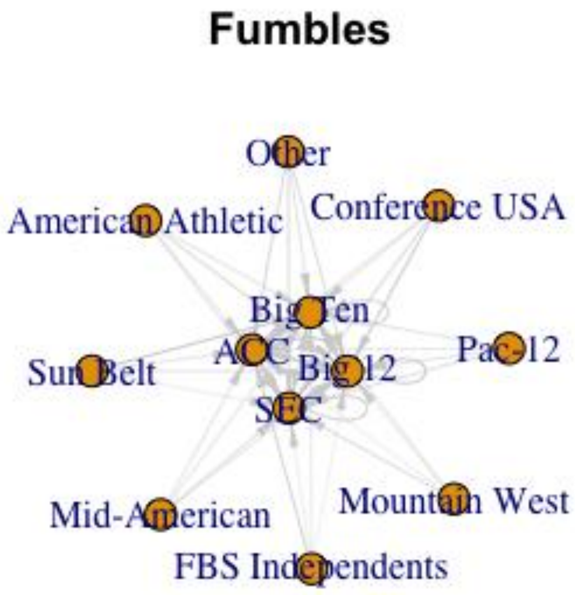
```
#interceptions
edgelist_interception=
  all_def_data2.mean[all_def_data2.mean$interception.mean>0, c("defense_conference", "off
ense_conference", "interception.mean")]
def_interception=graph_from_data_frame(d=edgelist_interception, directed=T)
def_interception=igraph::simplify(def_interception, edge.attr.comb="mean")
```

```
interception_mat=as_adjacency_matrix(def_interception, attr="interception.mean", sparse=F
)
interception_mat_in=t(interception_mat)

interception_mat_std=(interception_mat-mean(interception_mat))/sd(interception_mat)
interception_mat_in_std=t(interception_mat_std)

defense_interception <- delete.edges(defense_all, E(defense_all)[get.edge.attribute(defen
se_all,name = "interception.mean")==0])

interception_layout <- layout_with_dh(defense_interception)
plot(defense_interception, layout=interception_layout, main = "Interceptions",
     edge.arrow.size=.5, edge.arrow.width=.5,
     edge.color="light gray",
     edge.width=E(defense_interception)$interception.mean)
```

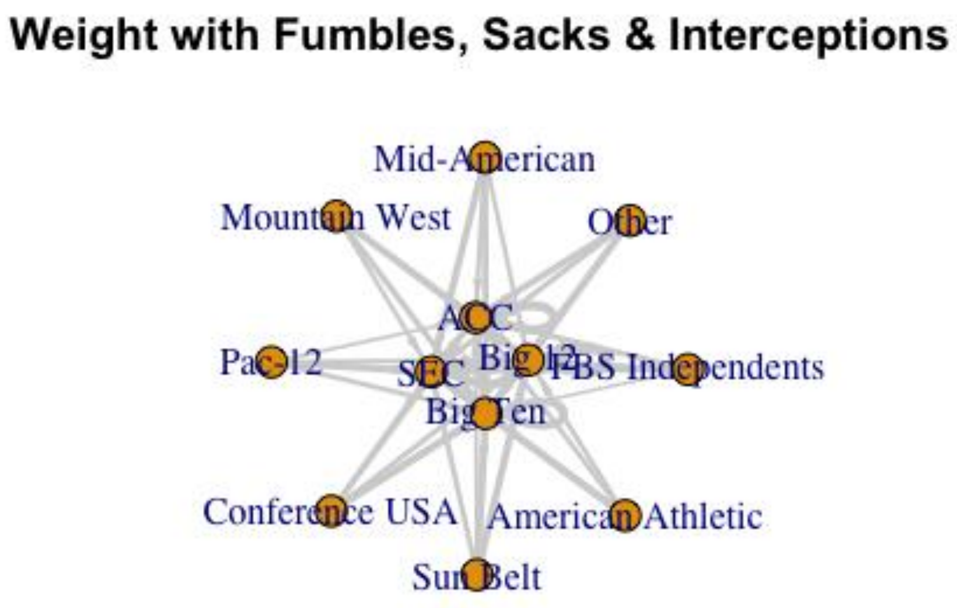


```
#fumbles
edgelist_fumble=
  all_def_data2.mean[all_def_data2.mean$fumble.mean>0, c("defense_conference", "offense_c
onference", "fumble.mean")]
def_fumble=graph_from_data_frame(d=edgelist_fumble, directed=F)
def_fumble=igraph::simplify(def_fumble, edge.attr.comb="mean")

fumble_mat=as_adjacency_matrix(def_fumble, attr="fumble.mean", sparse=F)
fumble_mat_in=t(fumble_mat)

fumble_mat_std=(fumble_mat-mean(fumble_mat))/sd(fumble_mat)
fumble_mat_in_std=t(fumble_mat_std)
```

```
defense_fumble <- delete.edges(defense_all, E(defense_all)[get.edge.attribute(defense_all
,name = "fumble.mean")==0])

fumble_layout <- igraph::layout_nicely(defense_fumble)
plot(defense_fumble, layout=fumble_layout, main = "Fumbles",
     edge.arrow.size=.5, edge.arrow.width=.5,
     edge.color="light gray",
     edge.width=E(defense_fumble)$fumble.mean)
```



```
#weight of all defense acts
edgelist_weight=
  all_def_data2.mean[all_def_data2.mean$weight.mean>0, c("defense_conference", "offense_c
onference", "weight.mean")]
def_weight=graph_from_data_frame(d=edgelist_weight, directed=F)
def_weight=igraph::simplify(def_weight, edge.attr.comb="mean")

weight_mat=as_adjacency_matrix(def_weight, attr="weight.mean", sparse=F)
weight_mat_in=t(weight_mat)

weight_mat_std=(weight_mat-mean(weight_mat))/sd(weight_mat)
weight_mat_in_std=t(weight_mat_std)

defense_weight <- delete.edges(defense_all, E(defense_all)[get.edge.attribute(defense_all
,name = "weight.mean")==0])

weight_layout <- igraph::layout_nicely(defense_weight)
plot(defense_weight, layout=weight_layout, main = "Weight with Fumbles, Sacks & Intercept
ions",
     edge.arrow.size=.5, edge.arrow.width=.5,
     edge.color="light gray",
     edge.width=E(defense_weight)$weight.mean/2)
```



Calculating Distance Between Nodes in the Network based on Pattern of Ties

Our distance matrix will suggests that Big Ten has similar ties as SEC and different ties from American Athletic. Similarly the Big 12 has similar ties to the SEC and different ties from the Sun Belt Conference. Lastly, the matrix will suggest that the ACC has similar ties to SEC and differs from the Mountain West Conference. Now looking at the distances visualizing the distances using MDS, using the euclidean distance calculation, where we will see but using the cmdscale function. We will also look at the distance data in the network layout. We can see that there are a few conferences that are visually different but for the most part the data are clustered around each other. This is due to the fact that we are only interested in the defense dominance in the Power 5 conferences (ACC, SEC, Big Ten, Big 12, and Pac 12) and who they have played in the last 5 years.

```
sacks_interception_fumble_std = cbind(sack_mat_std, sack_mat_in_std,
                                      interception_mat_std,interception_mat_in_std,
                                      fumble_mat_std)
dim(sacks_interception_fumble_std)

## [1] 12 60

euclid_dist = dist(x=sacks_interception_fumble_std, method = "euclidean")
euclid_dist
```

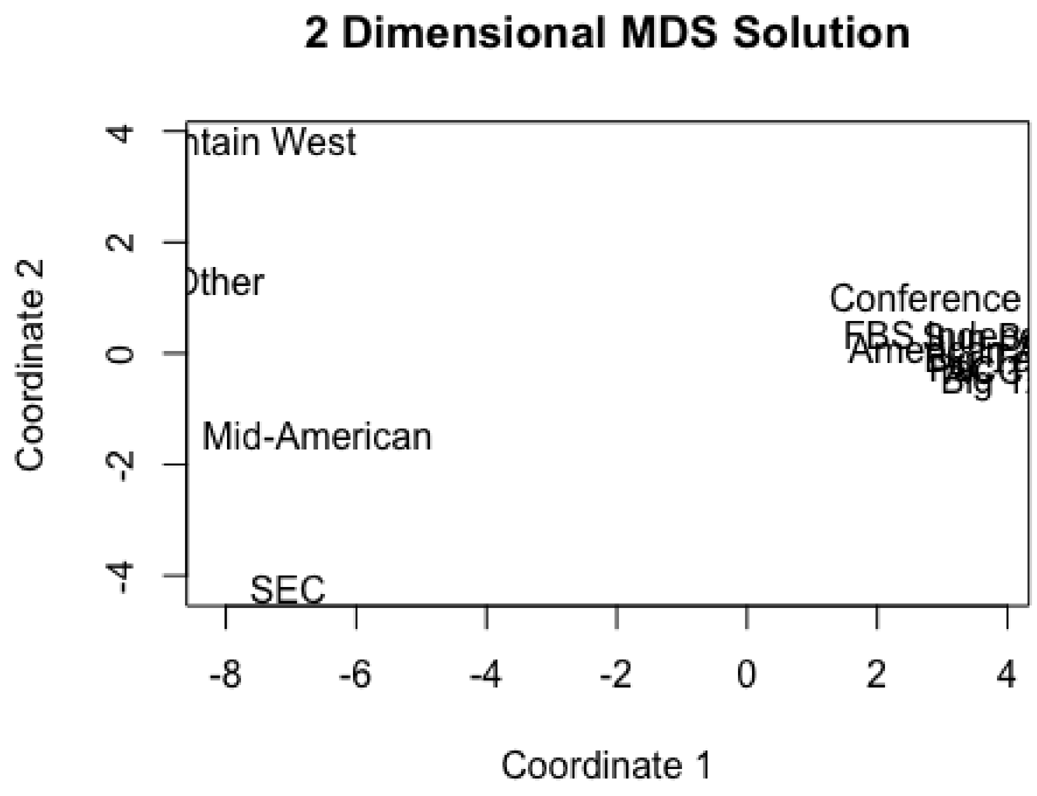
```
##                      Big Ten    Big 12       ACC       SEC Mountain West
## Big 12              8.078066                                        
## ACC                 8.335778  6.794084                              
## SEC                 7.597266  6.105879  8.230148                    
## Mountain West      12.906903 11.356887 12.267393 12.720641          
## Mid-American       12.755365 11.018531 11.814751 12.635372      2.633456
## Other              12.808790 11.259326 12.167404 12.679624      2.519152
## Sun Belt           13.068299 11.284331 11.962571 12.822675      3.183151
## Pac-12             12.698085 11.139876 11.755198 12.817129      3.661106
## FBS Independents   12.422834 10.952132 12.270142 12.320934      3.582518
## Conference USA     12.767019 11.461165 12.007564 12.804496      3.444326
## American Athletic  12.693002 11.064244 11.826542 12.644365      2.651279
##                    Mid-American     Other  Sun Belt    Pac-12 FBS Independents
## Big 12                                                                        
## ACC                                                                           
## SEC                                                                           
## Mountain West                                                                 
## Mid-American                                                                  
## Other                  1.690801                                               
## Sun Belt               2.523730  2.734885                                     
## Pac-12                 1.777191  2.595150  3.095178                           
## FBS Independents       2.469988  2.760513  3.459124  2.823148                 
## Conference USA         2.404345  1.914079  3.297295  2.345321         3.446859
## American Athletic      1.344054  1.486266  2.539468  1.924618         2.789252
##                    Conference USA
## Big 12                           
## ACC                              
## SEC                              
## Mountain West                    
## Mid-American                     
## Other                            
## Sun Belt                         
## Pac-12                           
## FBS Independents                 
## Conference USA                   
## American Athletic       1.488236
```

```
fit=cmdscale(euclid_dist, k=2)
fit
```

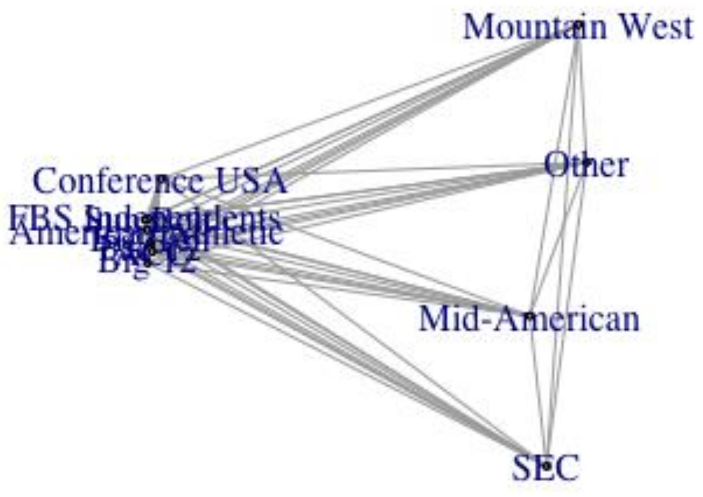
```
##                         [,1]       [,2]
## Big Ten            -7.906094  3.8453069
## Big 12             -6.587463 -1.4782345
## ACC                -7.038150 -4.2208802
## SEC                -8.124495  1.3235155
## Mountain West       3.763835  0.2997867
## Mid-American        3.696126 -0.2275109
## Other               3.849250  0.2813055
## Sun Belt            3.782340 -0.5132488
## Pac-12              3.650870 -0.3025905
## FBS Independents    3.411575  1.0263614
## Conference USA      3.797643  0.0939065
## American Athletic   3.704563 -0.1277176
```

```
x=fit[,1]
y=fit[,2]
```

```
plot(x, y, xlab = "Coordinate 1", ylab = "Coordinate 2", main = "2 Dimensional MDS Soluti
on", type = "n")
text(x,y, labels = unique(all_def_data2.mean$offense_conference), cex = 1)
```



```
g <- graph.full(nrow(as.matrix(euclid_dist)))
V(g)$label <- unique(all_def_data2.mean$offense_conference)
layout <- layout.mds(g, dist = as.matrix(euclid_dist))
plot(g, layout = layout, vertex.size = 3)
```



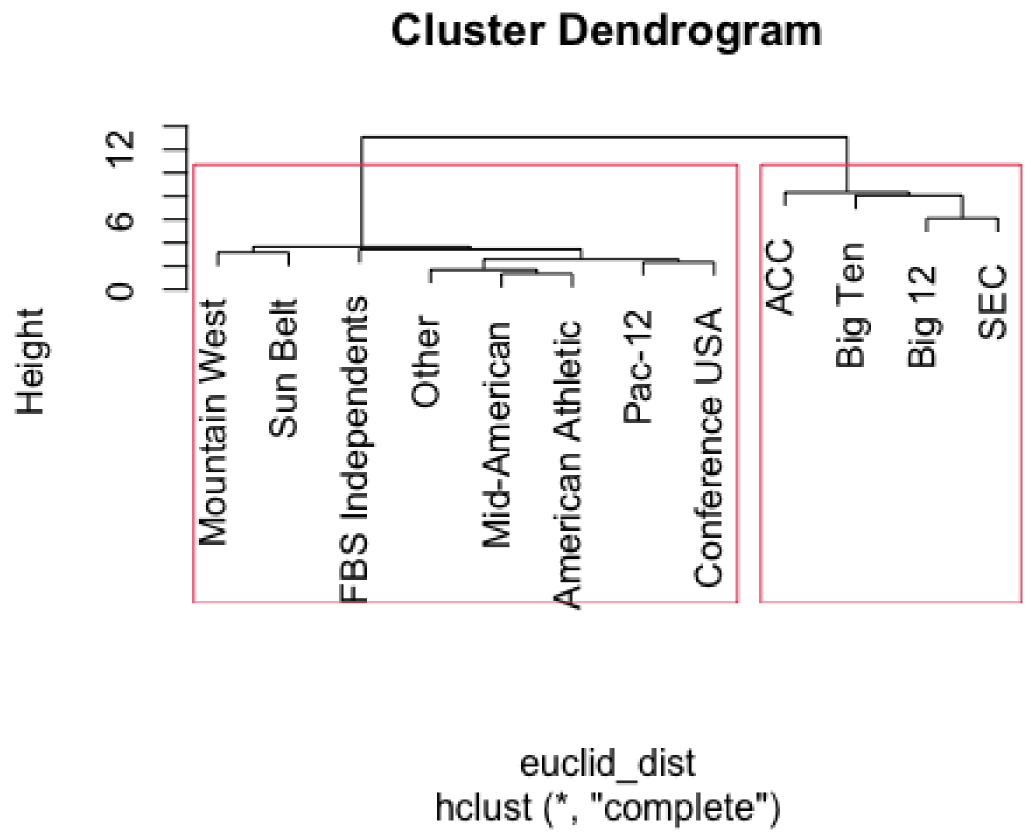
Hierarchical Clustering

Using the standard clustering visualization, a dendrogram, the inputs are the hclust output from above and h, the height at which you want to cut the tree. Here we will set h to 9, looking at clusters with distance less than 9.

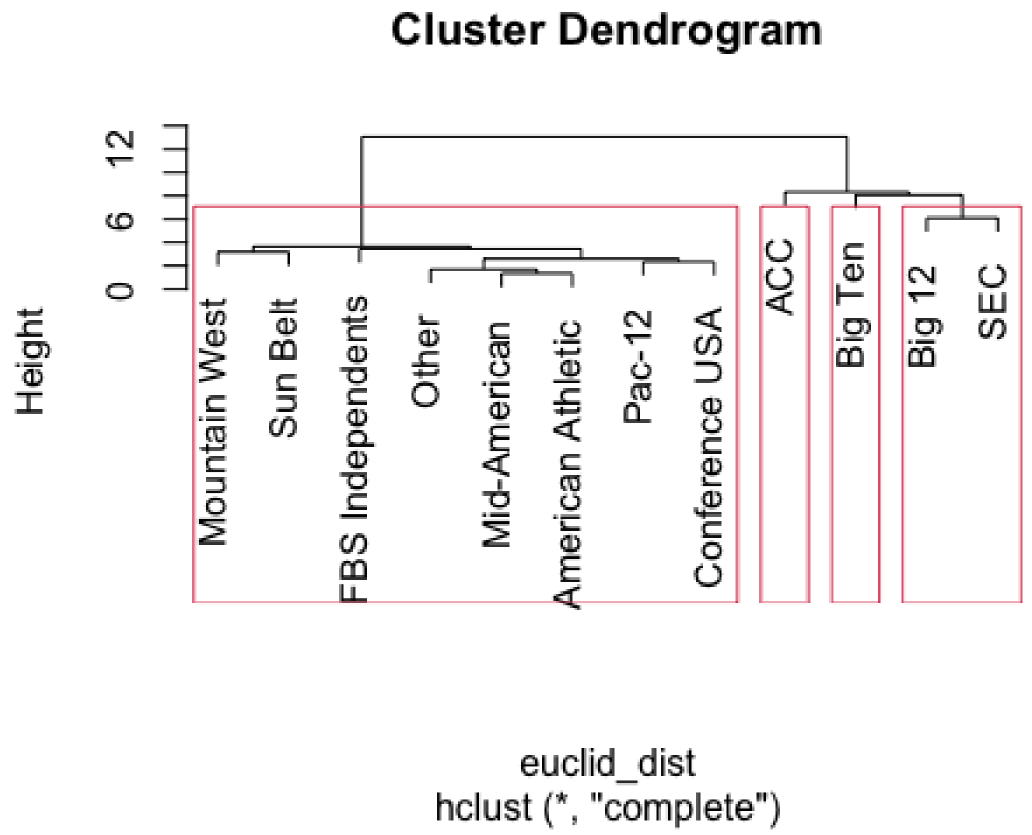
We are expecting the data to cluster around the Power 5 conferences but it looks like the Pac 12 isn't as dominate as we thought. The analysis of the clustering will suggest that the PAC 12 will fall within the other conferences in the dominance of defense performance in the last 5 years.

We will then extract the position memberships for height equal to 9. We will use the ward.d2 method in order to minimize the euclidean distance between our clusters. Based on this analysis we will look to fit 2 clusters that will separate the two groups. Using the NbClust function will help with the distance metric.

```
hc=hclust(euclid_dist)
plot(hc)
plot.clusters = rect.hclust(hc, h=9)
```



```r
plot(hc)
plot.clusters2 = rect.hclust(hc, h=8)
```



```
hc_ids=cutree(hc, h=8)
hc_ids

##          Big Ten            Big 12               ACC               SEC
##                1                 2                 3                 2
##    Mountain West      Mid-American             Other          Sun Belt
##                4                 4                 4                 4
##           Pac-12  FBS Independents    Conference USA American Athletic
##                4                 4                 4                 4

clusters=NbClust(sacks_interception_fumble_std, distance = "euclidean",
                 min.nc=2, max.nc=6,
                  method = "ward.D2", index = "ch")
clusters

## $All.index
##       2       3       4       5       6
## 29.7410 21.8048 24.2546 29.0743 27.8211
##
## $Best.nc
## Number_clusters     Value_Index
##           2.000          29.741
##
## $Best.partition
##          Big Ten            Big 12               ACC               SEC
##                1                 1                 1                 1
##    Mountain West      Mid-American             Other          Sun Belt
##                2                 2                 2                 2
##           Pac-12  FBS Independents    Conference USA American Athletic
##                2                 2                 2                 2
```

Role Analysis

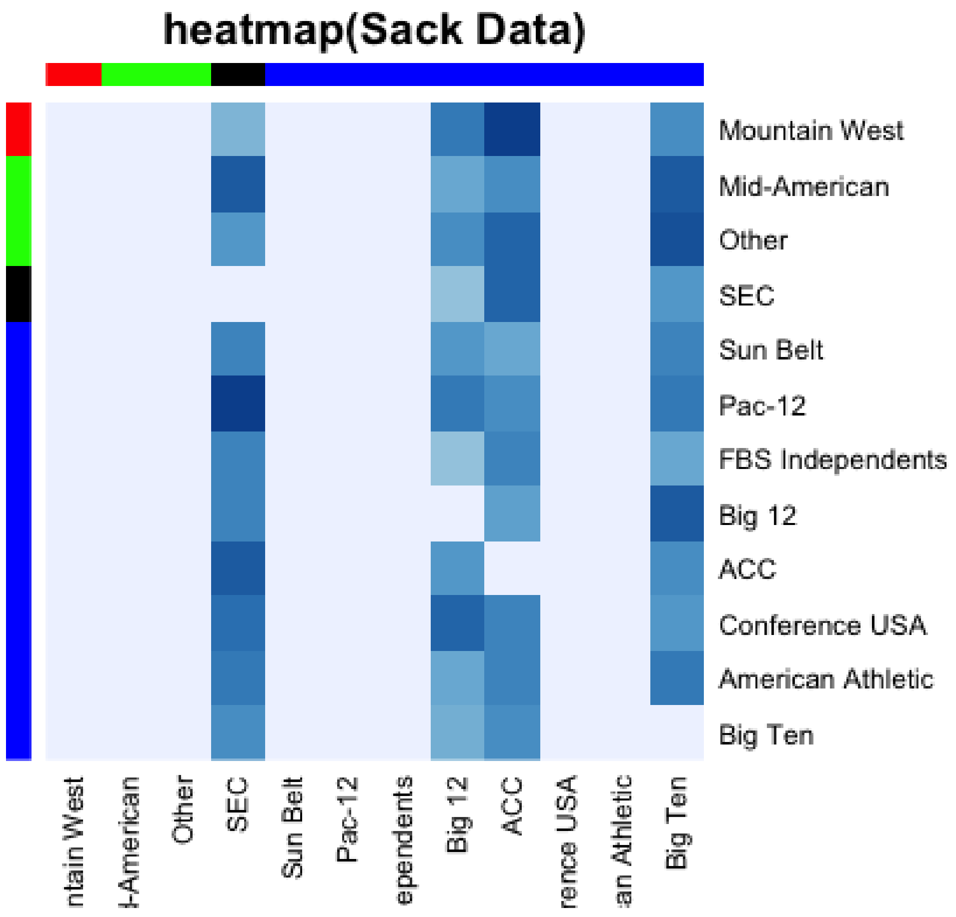
The colors in the heatmap for the for the sack interactions the darker will suggest a stronger interaction between the conferences. In our case, we will have two colors to denote the position of each node. Red corresponds to nodes in position 3; green corresponds to nodes in position 4. We can see that most sack interactions happen outside of positions. There us a fair amount of sack interaction between position 3 (black) and position 4 (blue). We expect some of this to occur from the Power 5 conferences but what we weren't expecting the sack interactions from from the PAC 12 not dominating over any other conference in sacks. Based on the type of offenses that the Pac 12 play, their defenses show know how to read and cover the pass routes. In the the data we can seem to see that in the sack averages.

In the heatmap for the fumble interactions will suggest that the fumbles occur in the same positions. As you can see the more dominating conference the Big 12 with a stronger number of interactions from Conference USA, which are in the same position in the data.

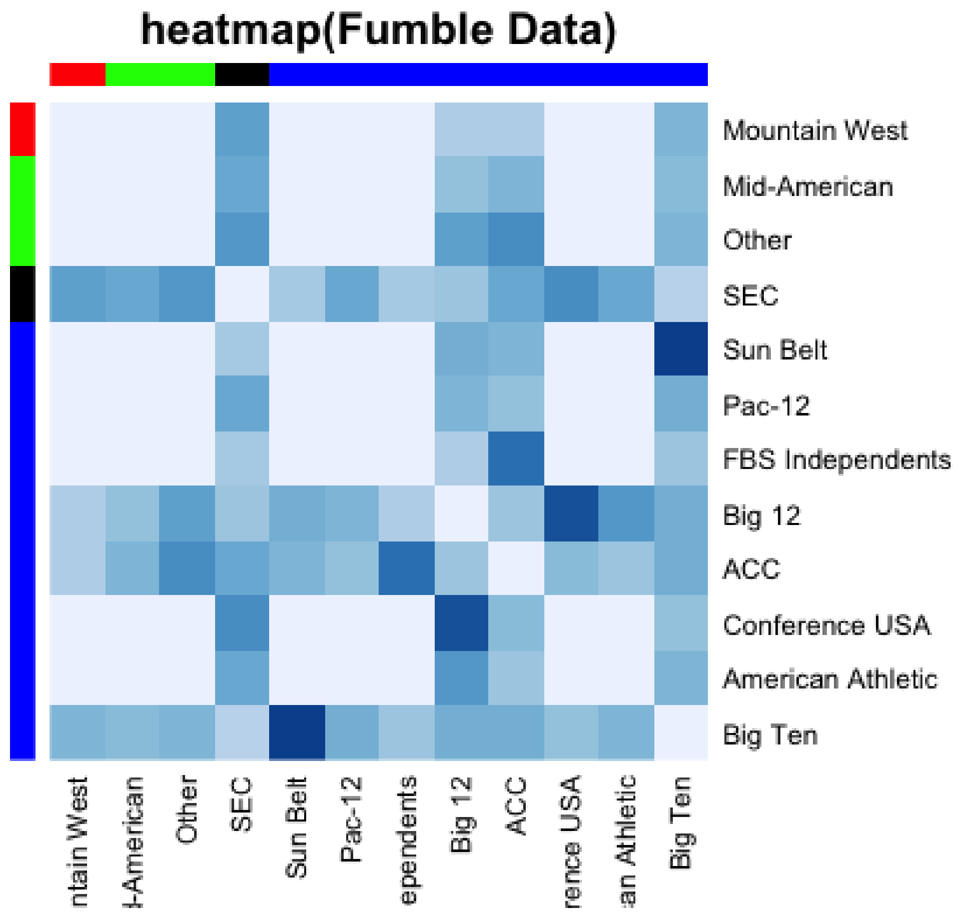
In the heatmap for the interception interactions will suggest that the interceptions occur in the dominating Power 5 conferences with ACC, Big Ten, Big 12 and SEC dominating every other conference in the analysis. This can suggest that the Pac 12 conference doesn't have the defensive teams that one would suggest. These interactions will suggest that interactions don't matter what positions are determined.

```
id_dat = data.frame(ids=unique(all_def_data2.mean$offense_conference), position=hc_ids)
id_dat = id_dat[order(id_dat$position),]
id_dat
```

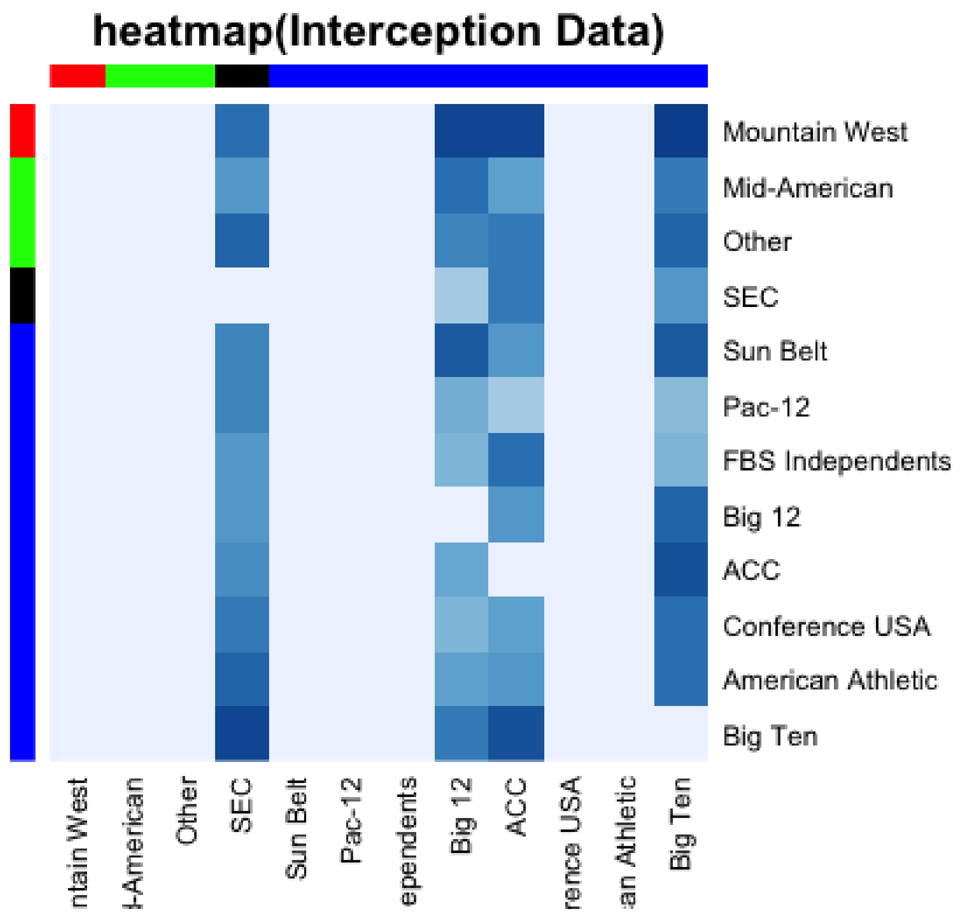
```
##                                  ids position
## Big Ten                Mountain West        1
## Big 12                  Mid-American        2
## SEC                            Other        2
## ACC                              SEC        3
## Mountain West               Sun Belt        4
## Mid-American                  Pac-12        4
## Other               FBS Independents        4
## Sun Belt                      Big 12        4
## Pac-12                           ACC        4
## FBS Independents      Conference USA        4
## Conference USA     American Athletic        4
## American Athletic             Big Ten        4

fumble_mat_rearrange = fumble_mat[id_dat$ids, id_dat$ids]
sack_mat_rearrange = sack_mat[id_dat$ids, id_dat$ids]
interception_mat_rearrange = interception_mat[id_dat$ids, id_dat$ids]

column_cols=c("red","green","black","blue","purple")

heatmap(sack_mat_rearrange, Rowv=NA, Colv=NA, revC=T,
        col=colorRampPalette(brewer.pal(6, "Blues"))(25),
        ColSideColors=column_cols[id_dat$position],
        RowSideColors=column_cols[id_dat$position], symm=T,
        main = "heatmap(Sack Data)")
```



```
heatmap(fumble_mat_rearrange, Rowv=NA, Colv=NA, revC=T,
        col=colorRampPalette(brewer.pal(6, "Blues"))(25),
        ColSideColors=column_cols[id_dat$position],
        RowSideColors=column_cols[id_dat$position], symm=T,
        main = "heatmap(Fumble Data)")
```



```
heatmap(interception_mat_rearrange, Rowv=NA, Colv=NA, revC=T,
        col=colorRampPalette(brewer.pal(6, "Blues"))(25),
        ColSideColors=column_cols[id_dat$position],
        RowSideColors=column_cols[id_dat$position], symm=T,
        main = "heatmap(Interception Data)")
```



Constructing the Blockmodel

The sack blockmodel shows us the density of within and between block ties. Here, we can see that block 2 sends most sack ties to Block 3, and a little less to block 2. The fumble blockmodel shows that block 1 send most fumble ties to Block 3 with Block 4 being a close second. Finally the The interception blockmodel shows that Block 1 will send the most interception ties to Block 3 with Block 2 following in a close second.

The blockmodel interpretations: - Looking at block 2, as the pattern of sack, fumble and interception ties are so distinct here.Looking at the blockmodel figure, has heavy sacks and interceptions with block 1 and block 3. They also do quite a bit more sack and interceptions than the other conferences in the last 5 years. We can see this by looking at the total level of sack and interception interactions for each block: - In short, this is block that incorporates the Big 12 and SEC conferences, who has more sacks and interceptions in the network but will fumble with other other conferences. In fact, the Big 12 and the SEC conferences that have similar football offensive play, but we arrived at the behavioral role without knowing that, and it was possible that another actor could have played that role.

Now looking at the more complicated block 3. Block 3 consists of nodes who are mostly sack conferences and interceptions with other conferences in block 3. They have much lower levels of sack interaction and interception interactions with block 4, these teams may not be able to play with the teams that are in the conferences of block 2. For example, those playing in block 2 will only play someone in block 3 as a rare out of conference game, which are games that are chosen based on money and distance from the university. For example, Alabama (SEC) will play a team like South Alabama (Conference USA) because of the distance for each of the teams.

Given this kind of analysis, an NFL Scout could use these revealed roles to predict other outcomes, such as the emergence of defense dominate teams in a moment of conference championships, teams that have a potential better secondary than most, finding the better defensive linemen in the time of the NFL draft/combine. We can also ask who ends up playing different roles in the Power 5 Conference teams.

In conclusion, we can see that the conferences in block 2, which include the Big 12 and SEC Conferences, one could say they have the "best defense teams." This is hard to say with certainty because we would use better metrics that will include yards for a loss and targets. This will give a better defintion of the overall defensive team.

```
#detach(package:igraph)
library(sna)

blockmod_sack=blockmodel(sack_mat, ec=hc_ids)
blockmod_sack

##
## Network Blockmodel:
##
## Block membership:
##
##  1  2  3  4  5  6  7  8  9 10 11 12
##  1  2  3  2  4  4  4  4  4  4  4  4
##
## Reduced form blockmodel:
##
##   1 2 3 4 5 6 7 8 9 10 11 12
##          Block 1  Block 2  Block 3  Block 4
## Block 1      NaN 2.842246 2.444444 2.720797
## Block 2 2.191667 1.997685 2.741015 2.568010
## Block 3 2.551724 2.636628      NaN 2.732305
## Block 4 0.000000 0.000000 0.000000 0.000000

blockmod_fumble=blockmodel(fumble_mat, ec=hc_ids)
blockmod_fumble

##
## Network Blockmodel:
##
## Block membership:
##
##  1  2  3  4  5  6  7  8  9 10 11 12
##  1  2  3  2  4  4  4  4  4  4  4  4
##
## Reduced form blockmodel:
##
##   1 2 3 4 5 6 7 8 9 10 11 12
##           Block 1   Block 2   Block 3   Block 4
## Block 1       NaN 0.3682932 0.5178799 0.5091104
## Block 2 0.3682932 0.3425926 0.4397903 0.5024869
## Block 3 0.5178799 0.4397903       NaN 0.4672434
## Block 4 0.5091104 0.5024869 0.4672434 0.0000000

blockmod_interception=blockmodel(interception_mat, ec=hc_ids)
blockmod_interception
```

```
##
## Network Blockmodel:
##
## Block membership:
##
##  1  2  3  4  5  6  7  8  9 10 11 12
##  1  2  3  2  4  4  4  4  4  4  4  4
##
## Reduced form blockmodel:
##
##   1 2 3 4 5 6 7 8 9 10 11 12
##          Block 1   Block 2   Block 3  Block 4
## Block 1      NaN 1.1363636 1.4074074 1.139792
## Block 2 1.333333 0.7199074 0.9207188 1.081177
## Block 3 1.379310 1.0813953       NaN 1.018404
## Block 4 0.000000 0.0000000 0.0000000 0.000000
```

```
blockedges_sack=block_model_edgelist_function(block_model=blockmod_sack,
                                                relation_label="sack",
                                                directed=T)
head(blockedges_sack)
```

```
##   sender receiver   weight  Tie WeightRecode
## 1      1        2 2.842246 sack            2
## 2      1        3 2.444444 sack            1
## 3      1        4 2.720797 sack            1
## 4      2        1 2.191667 sack            1
## 5      2        2 1.997685 sack            1
## 6      2        3 2.741015 sack            1
```

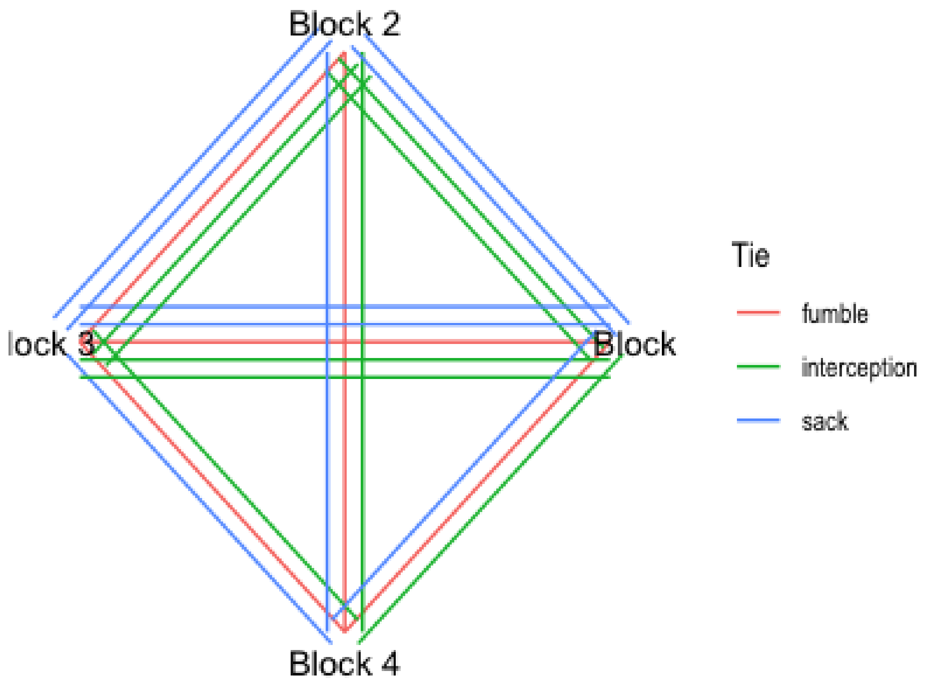
```
blockedges_fumble=block_model_edgelist_function(block_model=blockmod_fumble,
                                                relation_label="fumble",
                                                directed=F)
head(blockedges_fumble)
```

```
##   sender receiver    weight    Tie WeightRecode
## 1      1        2 0.3682932 fumble            0
## 2      1        3 0.5178799 fumble            2
## 3      1        4 0.5091104 fumble            1
## 4      2        2 0.3425926 fumble            0
## 5      2        3 0.4397903 fumble            1
## 6      2        4 0.5024869 fumble            1
```

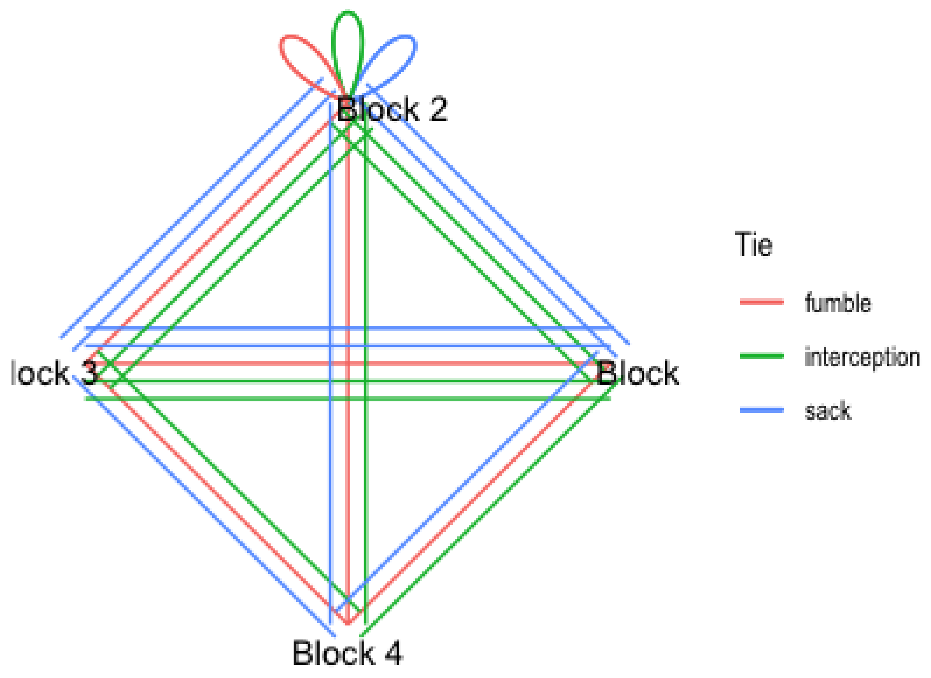
```
blockedges_interception=block_model_edgelist_function(block_model=blockmod_interception,
                                                relation_label="interception",
                                                directed=T)
head(blockedges_interception)
```

```
##   sender receiver    weight          Tie WeightRecode
## 1      1        2 1.1363636 interception            1
## 2      1        3 1.4074074 interception            2
## 3      1        4 1.1397916 interception            1
## 4      2        1 1.3333333 interception            2
## 5      2        2 0.7199074 interception            1
## 6      2        3 0.9207188 interception            1
```

```
block_dat=rbind(blockedges_sack, blockedges_fumble, blockedges_interception)

block_dat=block_dat[block_dat$WeightRecode %in% c("1", "2"),]

plot_net=tbl_graph(nodes=data.frame(ids=paste("Block", 1:4)), edges=block_dat, directed=F
)

ggraph(plot_net, layout='circle') +
  geom_edge_parallel(aes(colour=Tie))+
  geom_node_text(aes(label=ids),
                 nudge_x=c(.15, 0, -.15, 0),
                 nudge_y=c(0, .10, 0, -.10))+
  theme_graph()
```



```
ggraph(plot_net, layout='circle') +
  geom_edge_parallel(aes(colour=Tie))+
  geom_edge_loop(aes(colour="sack", span=45, strength=.5))+
  geom_edge_loop(aes(colour="interception", direction=90, span=45, strength=.5))+
  geom_edge_loop(aes(colour="fumble", direction=135, span=45, strength=.5))+
  geom_node_text(aes(label=ids),
                 nudge_x=c(.16, .17, -.16, 0),
                 nudge_y=c(-.03, -.02, -.03, -.10))+
  theme_graph()
```



```
rowSums(blockmod_sack$block.model, na.rm=T)

##  Block 1  Block 2  Block 3  Block 4
## 8.007487 9.498377 7.920657 0.000000

round(blockmod_sack$block.model[,2], 3)

## Block 1 Block 2 Block 3 Block 4
##   2.842   1.998   2.637   0.000

rowSums(blockmod_fumble$block.model, na.rm=T)

##  Block 1  Block 2  Block 3  Block 4
## 1.395284 1.653163 1.424914 1.478841

round(blockmod_fumble$block.model[,2], 3)

## Block 1 Block 2 Block 3 Block 4
##   0.368   0.343   0.440   0.502

rowSums(blockmod_interception$block.model, na.rm=T)

##  Block 1  Block 2  Block 3  Block 4
## 3.683563 4.055136 3.479110 0.000000

round(blockmod_interception$block.model[,2], 3)

## Block 1 Block 2 Block 3 Block 4
##   1.136   0.720   1.081   0.000
```