PersonalDataWrangling

Adam Rogers

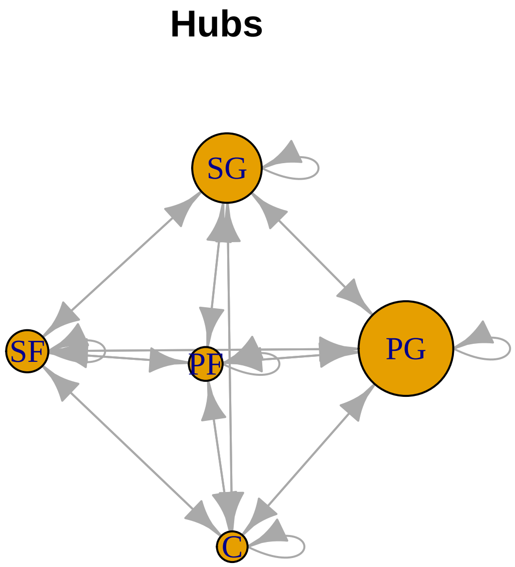
3/29/2022

```r
# Adam

# Load NBA play-by-play data with every court action
assist_network <- hoopR::load_nba_pbp(2022) %>%
  # Exclude All-Star Game
  filter(game_id != 401410564) %>%
  select(text, game_id) %>%
  # Grab only play descriptions that are assists
  filter(str_detect(text, "assist"))

# Create vector of players that scored
shooter <- str_extract(assist_network$text, "(.*)(?=\\smakes)")
# Create vector of players that passed
passer <- str_extract(assist_network$text, "(?<=\\().+?(?=\\sassists)")
# Combine into data frame with game_id named accordingly
assists_data <-
  cbind(data.frame(shooter), data.frame(passer), assist_network$game_id) %>%
  rename(game_id = "assist_network$game_id")

# Create vector of every game_id
game_id_vector <- as.vector(assists_data$game_id)
# Remove duplicates
game_id_vector_dup <- game_id_vector[!duplicated(game_id_vector)]

# Create initial player position frame from arbitrary game of another season
# This will be filtered out later
game_comb <- hoopR::espn_nba_player_box(400974437) %>%
  select(athlete_display_name, athlete_position_abbreviation) %>%
  # Create variable for shooter and passer
  mutate("shooter" = athlete_display_name) %>%
  mutate("passer" = athlete_display_name) %>%
  # Track game ID
  mutate(game_id = as.integer(400974437))

# Iterate over vector of every unique game_id
for (i in game_id_vector_dup) {
  # Create similar table for every game_id
  game_i <- hoopR::espn_nba_player_box(i) %>%
    select(athlete_display_name, athlete_position_abbreviation) %>%
    # Create variable for shooter and passer
    mutate("shooter" = athlete_display_name) %>%
    mutate("passer" = athlete_display_name) %>%
    # Track game ID
    mutate(game_id = as.integer(i))
  # Combine into single data frame
  game_comb <- rbind(game_comb, game_i)
}

# Filter arbitrary game
game_comb <- game_comb %>%
  filter(game_id != 400974437)

# Join to match shooter position
assist_pos <- assists_data %>%
  left_join(game_comb, by = c("shooter", "game_id")) %>%
  select(shooter, athlete_position_abbreviation, passer.x, game_id)

# Join to match passer position and tidy names
assist_pos <- assist_pos %>%
  left_join(game_comb, by = c("passer.x" = "passer", "game_id")) %>%
  rename(shooter = "shooter.x",
         shooter_pos = "athlete_position_abbreviation.x",
         passer = "passer.x",
         passer_pos = "athlete_position_abbreviation.y",
         game_id = "game_id") %>%
  select(passer_pos, shooter_pos)

# Create csv with shooter and passer names, positions, and gameIDs
write_csv(assist_pos, "assist_pos.csv")

assist_pos2 <- assist_pos %>%
  filter(shooter_pos != 'G', passer_pos != 'G',
         shooter_pos != 'F', passer_pos != 'F') %>%
  drop_na() %>%
  group_by(passer_pos, shooter_pos) %>%
  summarise(count = n())
```

```
## ‘summarise()‘ has grouped output by ’passer_pos’. You can override using the
## ‘.groups‘ argument.
```

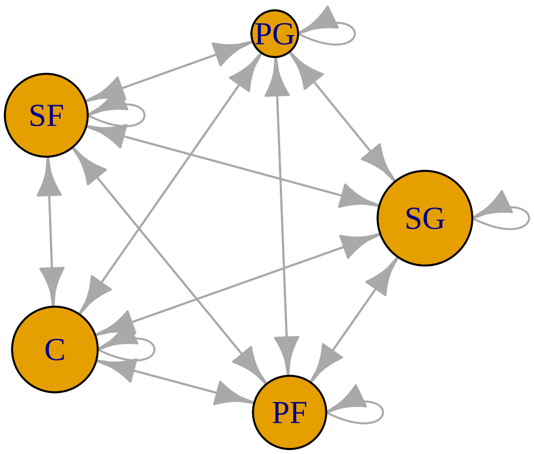
```
assist_el <- assist_pos2 %>%
 select(passer_pos, shooter_pos) %>%
 as.matrix()

assist_el_igraph <- graph_from_edgelist(el = assist_el, directed = TRUE)

assist_el_igraph <- set_edge_attr(graph = assist_el_igraph,
                   name = "count",
                   value = assist_pos2$count)

# Calculate hub score for each position
hs <- hub_score(assist_el_igraph, weights = E(assist_el_igraph)$count)$vector
# Calculate authority score for each position
as <- authority_score(assist_el_igraph, weights = E(assist_el_igraph)$count)$vector
# Plot hub score for each position
```

```
plot(assist_el_igraph, vertex.size=hs*50, main="Hubs")
```



```
# Plot authority score for each position
plot(assist_el_igraph, vertex.size=as*50, main="Authorities")
```

Authorities



```
assist_ggnetwork <- ggnetwork(assist_el_igraph)

ggplot(data = assist_ggnetwork, aes(x = x, y = y,
          xend = xend, yend = yend)) +

  geom_edges(arrow = arrow(type = "closed", angle = 10),
       color = "gray50",
       aes(size = count)) +
 geom_nodes(size = hs*30) +
  geom_nodelabel(aes(label = name), size = 3) +
 labs(title = "Migration among selected countries",
    subtitle = "Among females in 2000",
    size = "Number who migrated") +
 theme_blank()
```