final_wrangling - May Mishima

```r
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.2     v tibble    3.2.1
v lubridate 1.9.4     v tidyr     1.3.1
v purrr     1.0.4
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
```

```r
library(purrr)
library(dplyr)
library(stringr)
library(readr)
library(readr)
library(mosaic)
```

```
Registered S3 method overwritten by 'mosaic':
  method                           from
  fortify.SpatialPolygonsDataFrame ggplot2

The 'mosaic' package masks several functions from core packages in order to add
additional features.  The original behavior of these functions should not be affected by thi

Attaching package: 'mosaic'

The following object is masked from 'package:Matrix':
    mean

The following objects are masked from 'package:dplyr':

    count, do, tally

The following object is masked from 'package:purrr':

    cross

The following object is masked from 'package:ggplot2':

    stat

The following objects are masked from 'package:stats':

    binom.test, cor, cor.test, cov, fivenum, IQR, median, prop.test,
    quantile, sd, t.test, var

The following objects are masked from 'package:base':

    max, mean, min, prod, range, sample, sum
```

```r
library(GGally)
library(gridExtra)
```

```
Attaching package: 'gridExtra'

The following object is masked from 'package:dplyr':

    combine
```

```r
library(igraph)
```

```
Attaching package: 'igraph'

The following object is masked from 'package:mosaic':
    compare

The following objects are masked from 'package:lubridate':

    %--%, union

The following objects are masked from 'package:dplyr':

    as_data_frame, groups, union

The following objects are masked from 'package:purrr':

    compose, simplify

The following object is masked from 'package:tidyr':

    crossing

The following object is masked from 'package:tibble':

    as_data_frame

The following objects are masked from 'package:stats':

    decompose, spectrum

The following object is masked from 'package:base':

    union
```

```r
library(ggnetwork)
library(kableExtra)
```

```

Attaching package: 'kableExtra'

The following object is masked from 'package:dplyr':

    group_rows
```

```r
library(knitr)
```

Citations for the Data Used

- Baseball-Reference (Last updated 2025, November 2) 2025 Major League Baseball Standard Pitching: Player Standard Pitching. Sports Reference LLC. Retrieved on October 26, https://www.baseball-reference.com/leagues/majors/2025-standard-pitching.shtml.
- Baseball-Reference (Last updated 2025, November 2) 2024 Major League Baseball Standard Pitching: Player Standard Pitching. Sports Reference LLC. Retrieved on October 26, https://www.baseball-reference.com/leagues/majors/2024-standard-pitching.shtml.
- Baseball-Reference (Last updated 2025, November 2) 2023 Major League Baseball Standard Pitching: Player Standard Pitching. Sports Reference LLC. Retrieved on October 26, https://www.baseball-reference.com/leagues/majors/2023-standard-pitching.shtml.
- Baseball-Reference (Last updated 2025, November 2) 2022 Major League Baseball Standard Pitching: Player Standard Pitching. Sports Reference LLC. Retrieved on October 26, https://www.baseball-reference.com/leagues/majors/2022-standard-pitching.shtml.
- Baseball-Reference (Last updated 2025, November 2) 2021 Major League Baseball Standard Pitching: Player Standard Pitching. Sports Reference LLC. Retrieved on October 26, https://www.baseball-reference.com/leagues/majors/2021-standard-pitching.shtml.
- Baseball-Reference (Last updated 2025, November 18) 2020 Major League Baseball Standard Pitching: Player Standard Pitching. Sports Reference LLC. Retrieved on November 18, https://www.baseball-reference.com/leagues/majors/2020-standard-pitching.shtml.
- Baseball-Reference (Last updated 2025, November 18) 2019 Major League Baseball Standard Pitching: Player Standard Pitching. Sports Reference LLC. Retrieved on November 18, https://www.baseball-reference.com/leagues/majors/2019-standard-pitching.shtml.
- Baseball-Reference (Last updated 2025, November 18) 2018 Major League Baseball Standard Pitching: Player Standard Pitching. Sports Reference LLC. Retrieved on November 18, https://www.baseball-reference.com/leagues/majors/2018-standard-pitching.shtml.
- Baseball-Reference (Last updated 2025, November 18) 2017 Major League Baseball Standard Pitching: Player Standard Pitching. Sports Reference LLC. Retrieved on November 18, https://www.baseball-reference.com/leagues/majors/2017-standard-pitching.shtml.
- Baseball-Reference (Last updated 2025, November 18) 2016 Major League Baseball Standard Pitching: Player Standard Pitching. Sports Reference LLC. Retrieved on November 18, https://www.baseball-reference.com/leagues/majors/2021-standard-pitching.shtml.

- Baseball-Reference (Last updated 2025, November 18) 2015 Major League Baseball Standard Pitching: Player Standard Pitching. Sports Reference LLC. Retrieved on November 18, https://www.baseball-reference.com/leagues/majors/2015-standard-pitching.shtml.

- Baseball-Reference (Last updated 2025, November 2) 2025 Major League Baseball Standard Batting: Player Standard Batting. Sports Reference LLC. Retrieved on October 24, https://www.baseball-reference.com/leagues/majors/2025-standard-batting.shtml.

- Baseball-Reference (Last updated 2025, November 2) 2024 Major League Baseball Standard Batting: Player Standard Batting. Sports Reference LLC. Retrieved on October 24, https://www.baseball-reference.com/leagues/majors/2024-standard-batting.shtml.

- Baseball-Reference (Last updated 2025, November 2) 2023 Major League Baseball Standard Batting: Player Standard Batting. Sports Reference LLC. Retrieved on October 24, https://www.baseball-reference.com/leagues/majors/2023-standard-batting.shtml.

- Baseball-Reference (Last updated 2025, November 2) 2022 Major League Baseball Standard Batting: Player Standard Batting. Sports Reference LLC. Retrieved on October 24, https://www.baseball-reference.com/leagues/majors/2022-standard-batting.shtml.

- Baseball-Reference (Last updated 2025, November 2) 2021 Major League Baseball Standard Batting: Player Standard Batting. Sports Reference LLC. Retrieved on October 24, https://www.baseball-reference.com/leagues/majors/2021-standard-batting.shtml.

- Baseball-Reference (Last updated 2025, November 18) 2020 Major League Baseball Standard Batting: Player Standard Batting. Sports Reference LLC. Retrieved on November 18, https://www.baseball-reference.com/leagues/majors/2020-standard-batting.shtml.

- Baseball-Reference (Last updated 2025, November 18) 2019 Major League Baseball Standard Batting: Player Standard Batting. Sports Reference LLC. Retrieved on November 18, https://www.baseball-reference.com/leagues/majors/2019-standard-batting.shtml.

- Baseball-Reference (Last updated 2025, November 18) 2018 Major League Baseball Standard Batting: Player Standard Batting. Sports Reference LLC. Retrieved on November 18, https://www.baseball-reference.com/leagues/majors/2018-standard-batting.shtml.

- Baseball-Reference (Last updated 2025, November 18) 2017 Major League Baseball Standard Batting: Player Standard Batting. Sports Reference LLC. Retrieved on November 18, https://www.baseball-reference.com/leagues/majors/2017-standard-batting.shtml.

- Baseball-Reference (Last updated 2025, November 18) 2016 Major League Baseball Standard Batting: Player Standard Batting. Sports Reference LLC. Retrieved on November 18, https://www.baseball-reference.com/leagues/majors/2016-standard-batting.shtml.

- Baseball-Reference (Last updated 2025, November 18) 2015 Major League Baseball Standard Batting: Player Standard Batting. Sports Reference LLC. Retrieved on November 18, https://www.baseball-reference.com/leagues/majors/2015-standard-batting.shtml .

- Baseball-Reference (Last updated 2025, December 10) 2025 Major League Detailed Standings. Sports Reference LLC. Retrieved on December 10, https://www.baseball-reference.com/leagues/MLB-standings.shtml
- Baseball-Reference (Last updated 2025, December 10) 2024 Major League Detailed Standings. Sports Reference LLC. Retrieved on December 10, https://www.baseball-reference.com/leagues/majors/2024-standings.shtml
- Baseball-Reference (Last updated 2025, December 10) 2023 Major League Detailed Standings. Sports Reference LLC. Retrieved on December 10, https://www.baseball-reference.com/leagues/majors/2023-standings.shtml
- Baseball-Reference (Last updated 2025, December 10) 2022 Major League Detailed Standings. Sports Reference LLC. Retrieved on December 10, https://www.baseball-reference.com/leagues/majors/2022-standings.shtml
- Baseball-Reference (Last updated 2025, December 10) 2021 Major League Detailed Standings. Sports Reference LLC. Retrieved on December 10, https://www.baseball-reference.com/leagues/majors/2021-standings.shtml
- Baseball-Reference (Last updated 2025, December 10) 2020 Major League Detailed Standings. Sports Reference LLC. Retrieved on December 10, https://www.baseball-reference.com/leagues/majors/2020-standings.shtml
- Baseball-Reference (Last updated 2025, December 10) 2019 Major League Detailed Standings. Sports Reference LLC. Retrieved on December 10, https://www.baseball-reference.com/leagues/majors/2019-standings.shtml
- Baseball-Reference (Last updated 2025, December 10) 2018 Major League Detailed Standings. Sports Reference LLC. Retrieved on December 10, https://www.baseball-reference.com/leagues/majors/2018-standings.shtml
- Baseball-Reference (Last updated 2025, December 10) 2017 Major League Detailed Standings. Sports Reference LLC. Retrieved on December 10, https://www.baseball-reference.com/leagues/majors/2017-standings.shtml
- Baseball-Reference (Last updated 2025, December 10) 2016 Major League Detailed Standings. Sports Reference LLC. Retrieved on December 10, https://www.baseball-reference.com/leagues/majors/2016-standings.shtml
- Baseball-Reference (Last updated 2025, December 10) 2015 Major League Detailed Standings. Sports Reference LLC. Retrieved on December 10, https://www.baseball-reference.com/leagues/majors/2015-standings.shtml

Pitchers Data Wrangling

```
#importing pitching data 10/26
pitchers25 <- read.csv("pwar2025 - Sheet1.csv")
pitchers24 <- read.csv("pwar2024 - Sheet1.csv")
pitchers23 <- read.csv("pwar2023 - Sheet1.csv")
pitchers22 <- read.csv("pwar2022 - Sheet1.csv")
pitchers21 <- read.csv("pwar2021 - Sheet1.csv")
pitchers20 <- read.csv("pwar2020 - Sheet1.csv")
#accessed 11/18
pitchers19 <- read.csv("pitching2019 - Sheet1.csv")
pitchers18 <- read.csv("pitching2018 - Sheet1.csv")
pitchers17 <- read.csv("pitching2017 - Sheet1.csv")
pitchers16 <- read.csv("pitching2016 - Sheet1.csv")
pitchers15 <- read.csv("pitching2015 - Sheet1.csv")

#remove multi-teams(2TM, 3TM, etc) by filtering for multi-teams
pitchers25 <- pitchers25 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
pitchers24 <- pitchers24 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
pitchers23 <- pitchers23 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
pitchers22 <- pitchers22 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
pitchers21 <- pitchers21 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
pitchers20 <- pitchers20 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
pitchers19 <- pitchers19 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
pitchers18 <- pitchers18 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
pitchers17 <- pitchers17 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
pitchers16 <- pitchers16 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
pitchers15 <- pitchers15 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))

#select relevant columns
pitchers25 <- pitchers25 |> select(Player, Team)
```

```
pitchers24 <- pitchers24 |> select(Player, Team)
pitchers23 <- pitchers23 |> select(Player, Team)
pitchers22 <- pitchers22 |> select(Player, Team)
pitchers21 <- pitchers21 |> select(Player, Team)
pitchers20 <- pitchers20 |> select(Player, Team)
pitchers19 <- pitchers19 |> select(Player, Team)
pitchers18 <- pitchers18 |> select(Player, Team)
pitchers17 <- pitchers17 |> select(Player, Team)
pitchers16 <- pitchers16 |> select(Player, Team)
pitchers15 <- pitchers15 |> select(Player, Team)

#binding rows, creating Season variable, and cleaning names
pitchers_all <- bind_rows(
  mutate(pitchers15, Season = 2015),
  mutate(pitchers16, Season = 2016),
  mutate(pitchers17, Season = 2017),
  mutate(pitchers18, Season = 2018),
  mutate(pitchers19, Season = 2019),
  mutate(pitchers20, Season = 2020),
  mutate(pitchers21, Season = 2021),
  mutate(pitchers22, Season = 2022),
  mutate(pitchers23, Season = 2023),
  mutate(pitchers24, Season = 2024),
  mutate(pitchers25, Season = 2025)
) |>
  mutate(
    Player = str_squish(str_replace_all(Player, "[*#]", ""))
  )
```

#Position Player Wrangling and Overall Combination

```
#importing Overall WAR data 10/24
overalldata25 <- read.csv("WAR2025 - Sheet1.csv")
overalldata24 <- read.csv("WAR2024 - Sheet1.csv")
overalldata23 <- read.csv("WAR2023 - Sheet1.csv")
overalldata22 <- read.csv("WAR2022 - Sheet1.csv")
overalldata21 <- read.csv("WAR2021 - Sheet1.csv")
overalldata20 <- read.csv("WAR2020 - Sheet1.csv")
overalldata19 <- read.csv("players2019 - Sheet1.csv")
overalldata18 <- read.csv("players2018 - Sheet1.csv")
overalldata17 <- read.csv("players2017 - Sheet1.csv")
overalldata16 <- read.csv("players2016 - Sheet1.csv")
```

```r
overalldata15 <- read.csv("players2015 - Sheet1.csv")

#removing any duplicates
overalldata25 <- overalldata25 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
overalldata24 <- overalldata24 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
overalldata23 <- overalldata23 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
overalldata22 <- overalldata22 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
overalldata21 <- overalldata21 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
overalldata20 <- overalldata20 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
overalldata19 <- overalldata19 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
overalldata18 <- overalldata18 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
overalldata17 <- overalldata17 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
overalldata16 <- overalldata16 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))
overalldata15 <- overalldata15 |>
  filter(!str_detect(Team, "^(2TM|3TM|4TM|5TM)$"))

#selecting relevant columns
overalldata25 <- overalldata25 |> select(Player, Team)
overalldata24 <- overalldata24 |> select(Player, Team)
overalldata23 <- overalldata23 |> select(Player, Team)
overalldata22 <- overalldata22 |> select(Player, Team)
overalldata21 <- overalldata21 |> select(Player, Team)
overalldata20 <- overalldata20 |> select(Player, Team)
overalldata19 <- overalldata19 |> select(Player, Team)
overalldata18 <- overalldata18 |> select(Player, Team)
overalldata17 <- overalldata17 |> select(Player, Team)
overalldata16 <- overalldata16 |> select(Player, Team)
overalldata15 <- overalldata15 |> select(Player, Team)

#bind rows and remove last symbols
all_overall <- bind_rows(
  mutate(overalldata15, Season = 2015),
  mutate(overalldata16, Season = 2016),
  mutate(overalldata17, Season = 2017),
  mutate(overalldata18, Season = 2018),
  mutate(overalldata19, Season = 2019),
  mutate(overalldata20, Season = 2020),
  mutate(overalldata21, Season = 2021),
  mutate(overalldata22, Season = 2022),
  mutate(overalldata23, Season = 2023),
  mutate(overalldata24, Season = 2024),
  mutate(overalldata25, Season = 2025)
) |>
  mutate(
    Player = str_squish(str_replace_all(Player, "[*#]", ""))
  )

#join datasets into one
all <- full_join(pitchers_all, all_overall, by = c("Player", "Season", "Team"))

#adjusting for teams that have changed names since 2000
all <- all |>
  mutate(
    Team = case_when(
      Team == "ANA" ~ "LAA",
      Team == "FLA" ~ "MIA",
      Team == "MON" ~ "WSN",
      Team == "TBD" ~ "TBR",
      Team == "TBA" ~ "TBR",
      Team == "OAK" ~ "ATH",
      TRUE ~ Team
    )
  )
```

#Win Percentage Wrangling

```
#retrieved 12/7
win25 <- read.csv("win2025 - Sheet1.csv")
win24 <- read.csv("win2024 - Sheet1.csv")
win23 <- read.csv("win2023 - Sheet1.csv")
win22 <- read.csv("win2022 - Sheet1.csv")
win21 <- read.csv("win2021 - Sheet1.csv")
win20 <- read.csv("win2020 - Sheet1.csv")
win19 <- read.csv("win2019 - Sheet1.csv")
```

```r
win18 <- read.csv("win2018 - Sheet1.csv")
win17 <- read.csv("win2017 - Sheet1.csv")
win16 <- read.csv("win2016 - Sheet1.csv")
win15 <- read.csv("win2015 - Sheet1.csv")

#select relevant columns
win25 <- win25 |> select(Tm, W.L.)
win24 <- win24 |> select(Tm, W.L.)
win23 <- win23 |> select(Tm, W.L.)
win22 <- win22 |> select(Tm, W.L.)
win21 <- win21 |> select(Tm, W.L.)
win20 <- win20 |> select(Tm, W.L.)
win19 <- win19 |> select(Tm, W.L.)
win18 <- win18 |> select(Tm, W.L.)
win17 <- win17 |> select(Tm, W.L.)
win16 <- win16 |> select(Tm, W.L.)
win15 <- win15 |> select(Tm, W.L.)

#combining rows
all_win <- bind_rows(
  mutate(win15, Season = 2015),
  mutate(win16, Season = 2016),
  mutate(win17, Season = 2017),
  mutate(win18, Season = 2018),
  mutate(win19, Season = 2019),
  mutate(win20, Season = 2020),
  mutate(win21, Season = 2021),
  mutate(win22, Season = 2022),
  mutate(win23, Season = 2023),
  mutate(win24, Season = 2024),
  mutate(win25, Season = 2025)
)

#adjusting for teams that have changed names since 2015
all_win_clean <- all_win |>
  mutate(
    Tm = case_when(
      Tm == "Los Angeles Angels of Anaheim" ~ "Los Angeles Angels",
      Tm == "Cleveland Indians" ~ "Cleveland Guardians",
      Tm == "Oakland Athletics" ~ "Athletics",
      TRUE ~ Tm
    )
  )

#obtain average win% per team across seasons
avg_winpct <- all_win_clean |>
  group_by(Tm) |>
  summarize(AvgWinPct = mean(W.L., na.rm = TRUE))

#adjusting for teams that have changed names since 2015
avg_winpct |>
  mutate(
    Tm = case_when(
      Tm == "Los Angeles Angels of Anaheim" ~ "Los Angeles Angels",
      Tm == "Cleveland Indians" ~ "Cleveland Guardians",
      Tm == "Oakland Athletics" ~ "Athletics",
      TRUE ~ Tm
    ))
```

```
# A tibble: 30 x 2
   Tm                    AvgWinPct
   <chr>                     <dbl>
 1 Arizona Diamondbacks      0.480
 2 Athletics                 0.474
 3 Atlanta Braves            0.532
 4 Baltimore Orioles         0.458
 5 Boston Red Sox            0.527
 6 Chicago Cubs              0.542
 7 Chicago White Sox         0.441
 8 Cincinnati Reds           0.456
 9 Cleveland Guardians       0.553
10 Colorado Rockies          0.430
# i 20 more rows
```

Transitivity and Networks for each MLB team

```
#NYY network
#filter just NYY
nyy_data <- all |>
  filter(Team == "NYY", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)
```

```r
#calculate overlap for every pair (edges and weights)
pairs_nyy <- nyy_data |>
  inner_join(nyy_data, by = "Season") |>
  #removes duplicates and opposite combos
  filter(Player.x < Player.y) |>
  #how many seasons overlap
  count(Player.x, Player.y, name = "overlap")
```

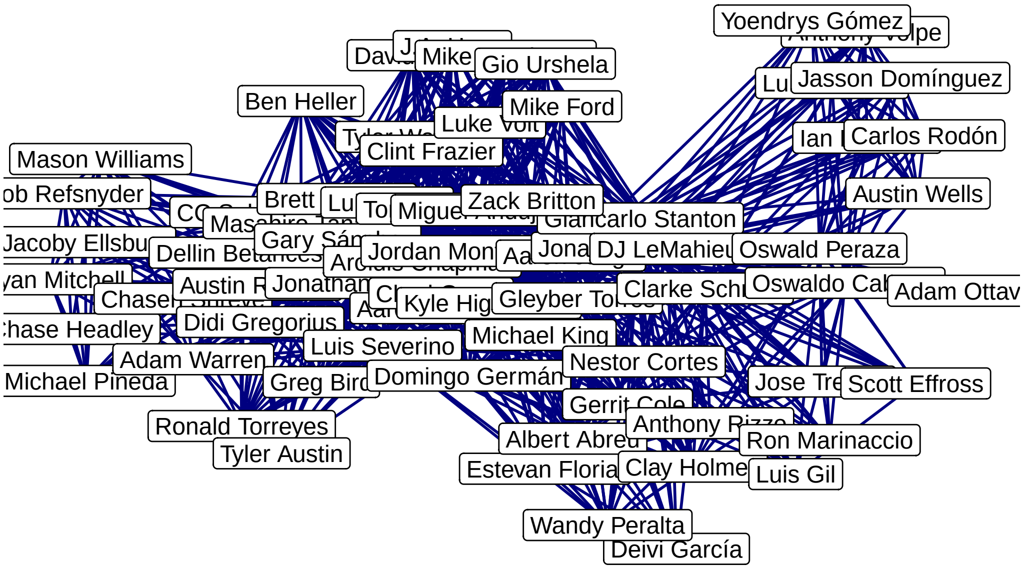
```
Warning in inner_join(nyy_data, nyy_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```r
#edge is set to pairs that have at least 3 seasons together
edges_nyy <- pairs_nyy |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_nyy <- graph_from_data_frame(edges_nyy, directed = FALSE)

#compute transitivity
trans_nyy <- transitivity(g_nyy, type = "global")
trans_nyy
```

```
[1] 0.6920599
```

```r
nyy_network <- ggnetwork(g_nyy)

ggplot(nyy_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "navy") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("New York Yankees Social Network 2015-2025")
```

New York Yankees Social Network 2015−2025



```
#ATL network
#filter just ATL
atl_data <- all |>
  filter(Team == "ATL", Season >= 2015, Season <= 2025 )|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_atl <- atl_data |>
  inner_join(atl_data, by = "Season") |>
  #removes duplicates and opposite combos
  filter(Player.x < Player.y) |>
  #how many seasons did they overlap for?
  count(Player.x, Player.y, name = "overlap")
```

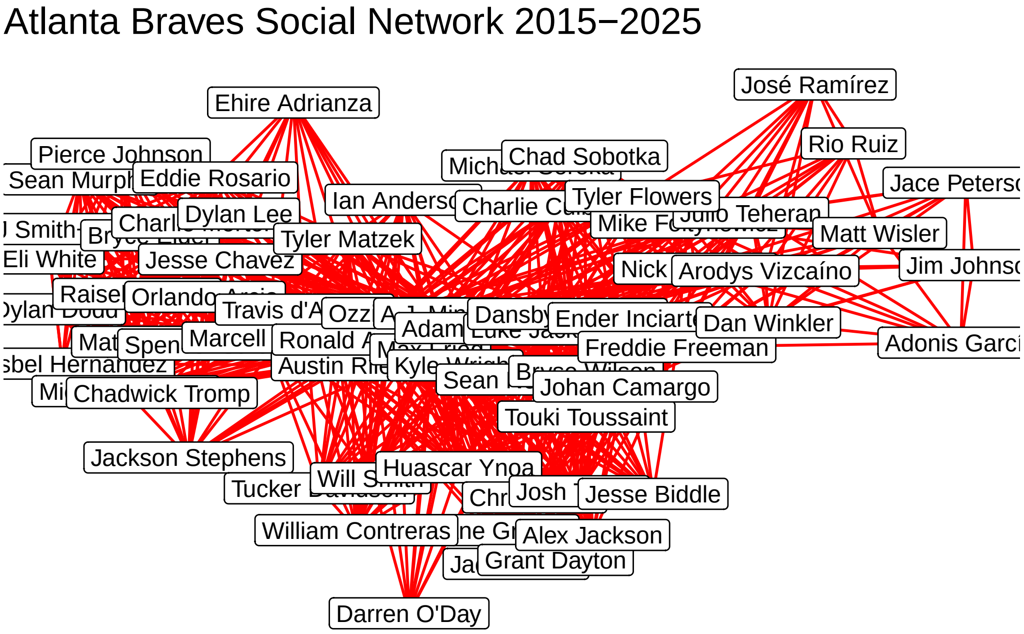
```
Warning in inner_join(atl_data, atl_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 5 seasons together
edges_atl <- pairs_atl |>
  filter(overlap >= 3)
```

```
#put into graph frame, false to allow for transitivity
g_atl <- graph_from_data_frame(edges_atl, directed = FALSE)

#compute transitivity
trans_atl <- transitivity(g_atl, type = "global")
trans_atl
```

```
[1] 0.6878162
```

```
atl_network <- ggnetwork(g_atl)

ggplot(atl_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "red") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Atlanta Braves Social Network 2015-2025")
```



```
#Boston Red Sox
#filter just BOS
bos_data <- all |>
```

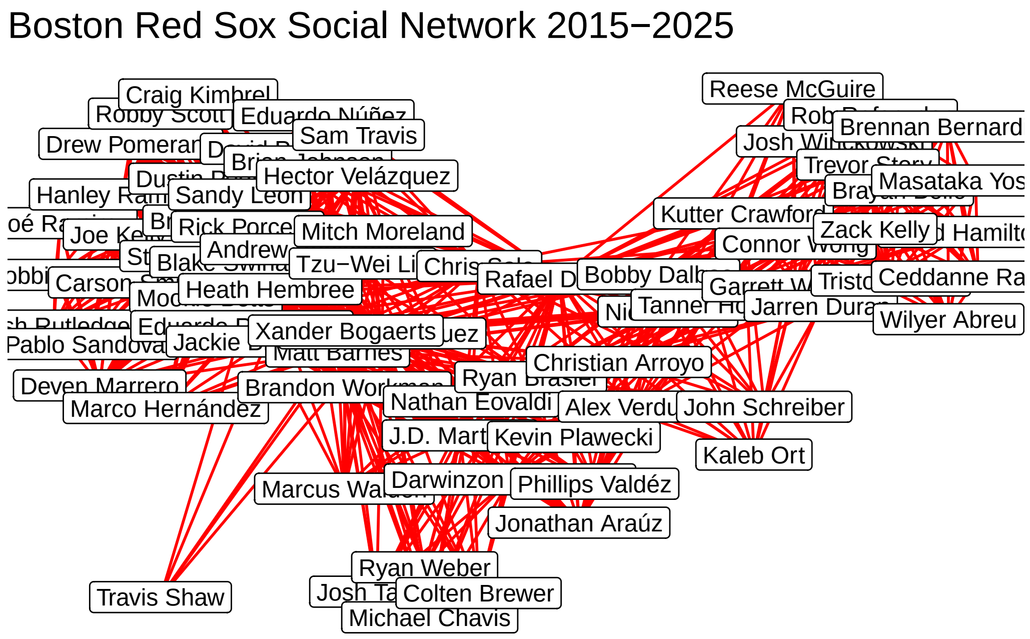
```
  filter(Team == "BOS", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_bos <- bos_data |>
  inner_join(bos_data, by = "Season") |>
  #removes duplicates and opposite combos
  filter(Player.x < Player.y) |>
  #how many seasons together?
  count(Player.x, Player.y, name = "overlap")
```

```
Warning in inner_join(bos_data, bos_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_bos <- pairs_bos |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_bos <- graph_from_data_frame(edges_bos, directed = FALSE)

#compute transitivity
trans_bos <- transitivity(g_bos, type = "global")
trans_bos
```

```
[1] 0.6989435
```

```
bos_network <- ggnetwork(g_bos)

ggplot(bos_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "red") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Boston Red Sox Social Network 2015-2025")
```


Boston Red Sox Social Network 2015−2025

```
#LAD network
#filter just LAD
lad_data <- all |>
  filter(Team == "LAD", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_lad <- lad_data |>
  inner_join(lad_data, by = "Season") |>
  #removes duplicates and opposite combos
  filter(Player.x < Player.y) |>
  #how many seasons together?
  count(Player.x, Player.y, name = "overlap")
```

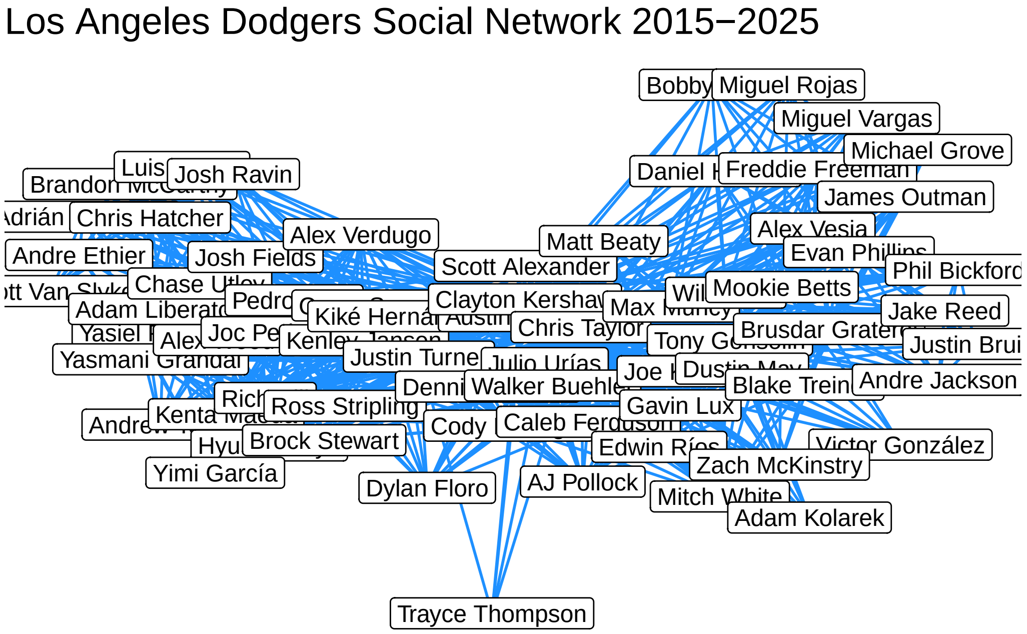
```
Warning in inner_join(lad_data, lad_data, by = "Season"): Detected an unexpected many-to-man
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_lad <- pairs_lad |>
  filter(overlap >= 3)
```

```
#put into graph frame, false to allow for transitivity
g_lad <- graph_from_data_frame(edges_lad, directed = FALSE)

#compute transitivity
trans_lad <- transitivity(g_lad, type = "global")
trans_lad
```

```
[1] 0.6566315
```

```
lad_network <- ggnetwork(g_lad)

ggplot(lad_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "dodgerblue") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Los Angeles Dodgers Social Network 2015-2025")
```



```
#TOR network
#filter just TOR and the seasons
tor_data <- all |>
```

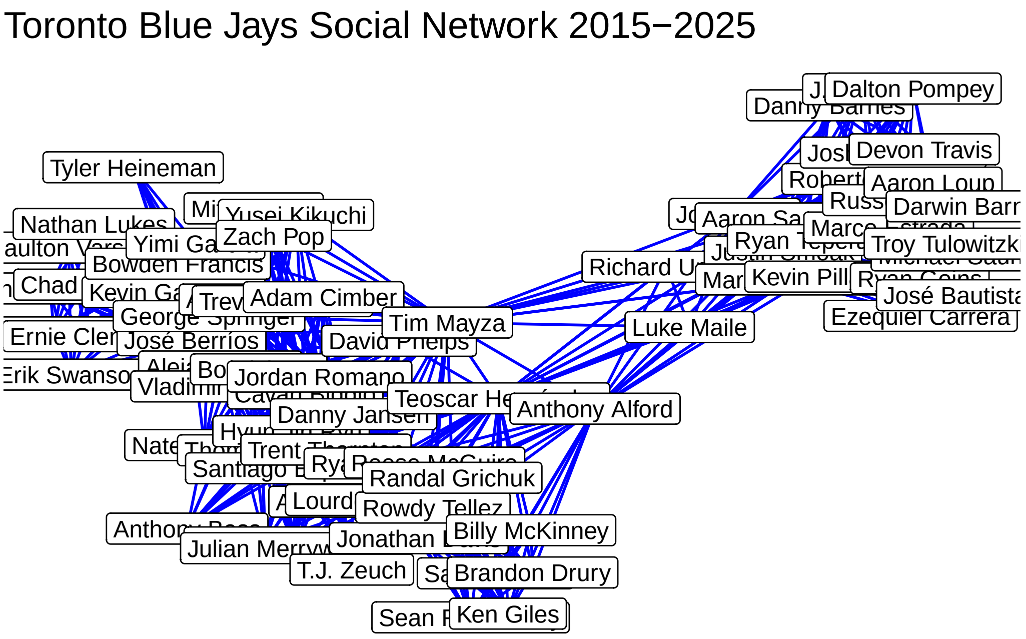
```
  filter(Team == "TOR", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_tor <- tor_data |>
  inner_join(tor_data, by = "Season") |>
  #removes duplicates and opposite combos
  filter(Player.x < Player.y) |>
  #how many combined seasons?
  count(Player.x, Player.y, name = "overlap")
```

```
Warning in inner_join(tor_data, tor_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least2 seasons together
edges_tor <- pairs_tor |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_tor <- graph_from_data_frame(edges_tor, directed = FALSE)

#compute transitivity
trans_tor <- transitivity(g_tor, type = "global")
trans_tor
```

```
[1] 0.694812
```

```
tor_network <- ggnetwork(g_tor)

ggplot(tor_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "blue") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Toronto Blue Jays Social Network 2015-2025")
```



```
#Tampa Bay Rays
#filter just TBR
tbr_data <- all |>
  filter(Team == "TBR", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_tbr <- tbr_data |>
  inner_join(tbr_data, by = "Season") |>
  #removes duplicates and opposite combos
  filter(Player.x < Player.y) |>
  #how many combined seasons?
  count(Player.x, Player.y, name = "overlap")
```

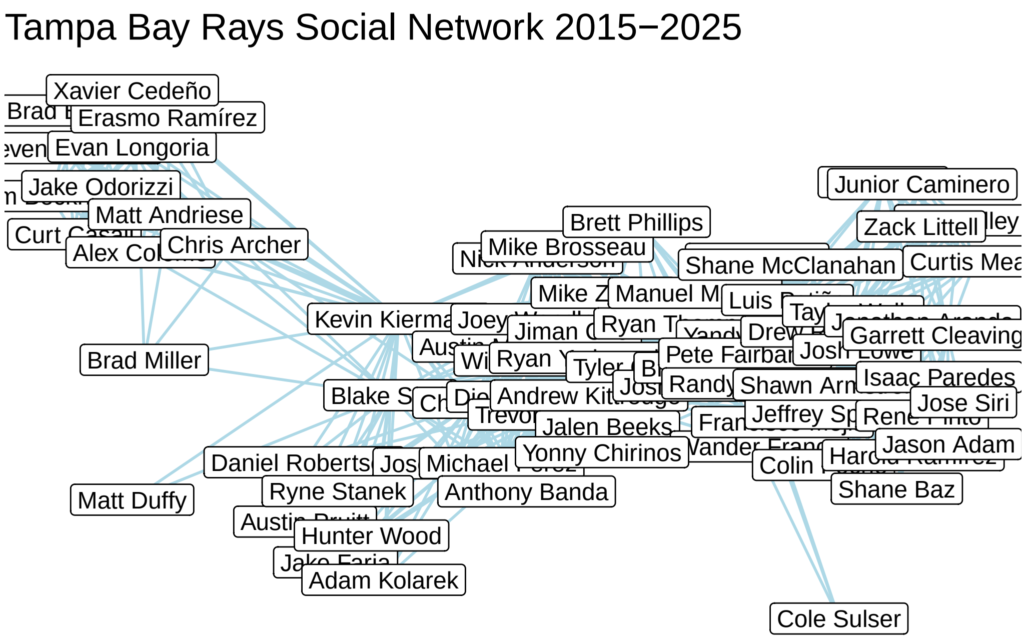
```
Warning in inner_join(tbr_data, tbr_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_tbr <- pairs_tbr |>
  filter(overlap >= 3)
```

```
#put into graph frame, false to allow for transitivity
g_tbr <- graph_from_data_frame(edges_tbr, directed = FALSE)

#compute transitivity
trans_tbr <- transitivity(g_tbr, type = "global")
trans_tbr
```

```
[1] 0.6596733
```

```
tbr_network <- ggnetwork(g_tbr)

#network building
ggplot(tbr_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "lightblue") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Tampa Bay Rays Social Network 2015-2025")
```



```
#BAL network
#filter just BAL
```

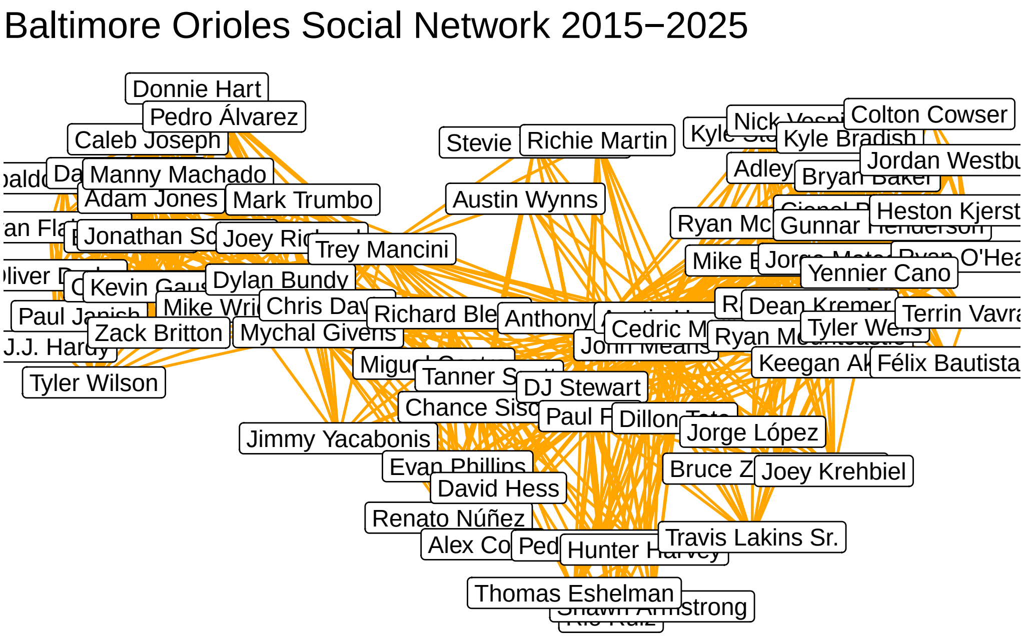
```r
bal_data <- all |>
  filter(Team == "BAL", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_bal <- bal_data |>
  inner_join(bal_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

```
Warning in inner_join(bal_data, bal_data, by = "Season"): Detected an unexpected many-to-man
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```r
#edge is set to pairs that have at least 3 seasons together
edges_bal <- pairs_bal |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_bal <- graph_from_data_frame(edges_bal, directed = FALSE)

#compute transitivity
trans_bal <- transitivity(g_bal, type = "global")
trans_bal
```

```
[1] 0.6928982
```

```r
bal_network <- ggnetwork(g_bal)

ggplot(bal_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "orange") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Baltimore Orioles Social Network 2015-2025")
```



Baltimore Orioles Social Network 2015−2025

```
#CLE network
#filter just CLE
cle_data <- all |>
  filter(Team == "CLE", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_cle <- cle_data |>
  inner_join(cle_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

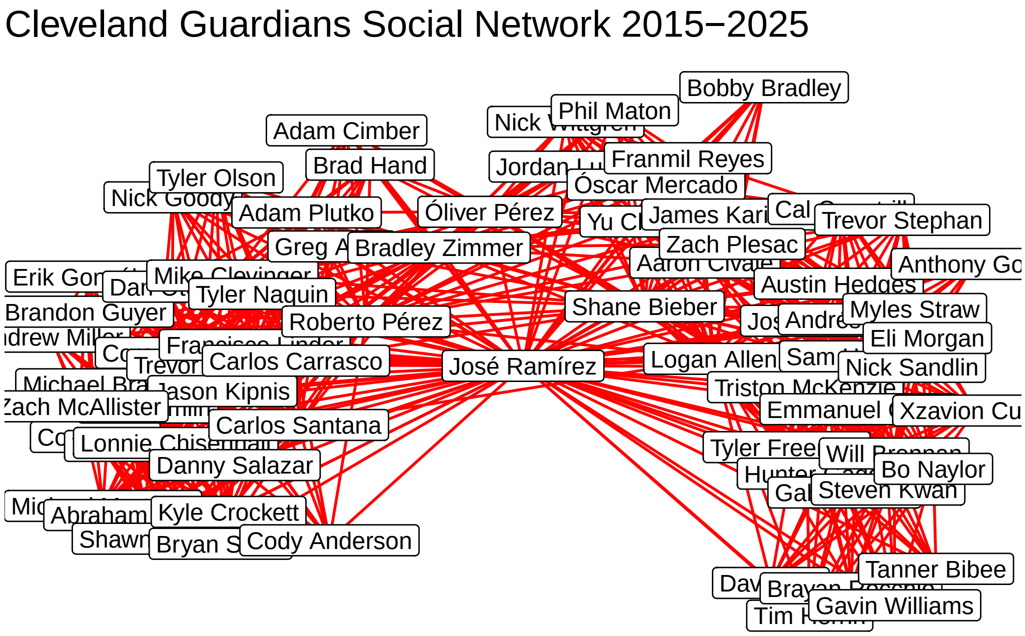
```
Warning in inner_join(cle_data, cle_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_cle <- pairs_cle |>
  filter(overlap >= 3)
```

```
#put into graph frame, false to allow for transitivity
g_cle <- graph_from_data_frame(edges_cle, directed = FALSE)

#compute transitivity
trans_cle <- transitivity(g_cle, type = "global")
trans_cle
```

```
[1] 0.6915636
```

```
cle_network <- ggnetwork(g_cle)

ggplot(cle_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "red") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Cleveland Guardians Social Network 2015-2025")
```



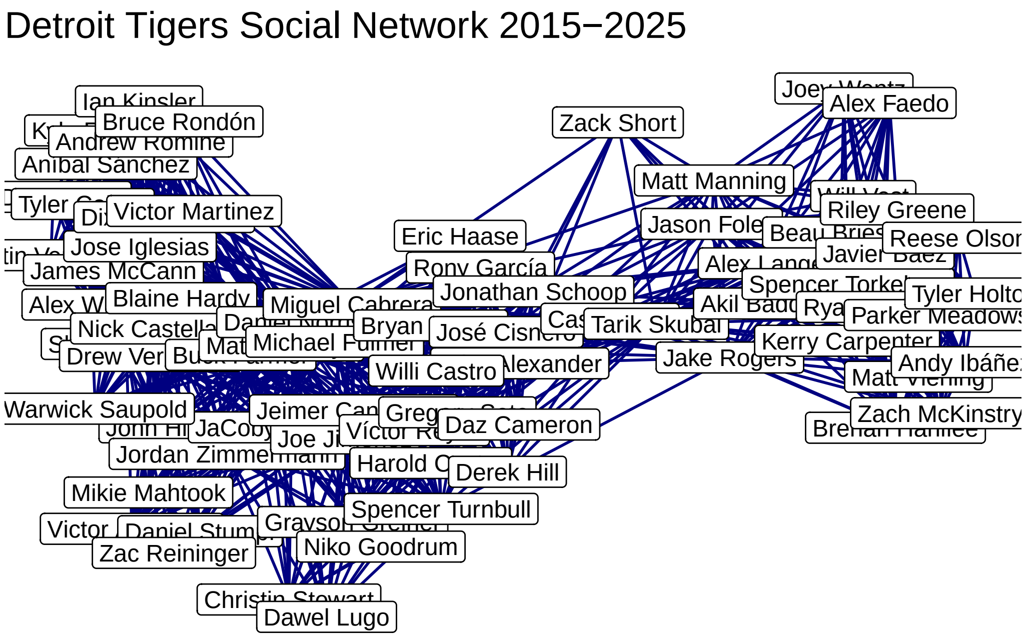
```
#DET network
#filter just DET
det_data <- all |>
  filter(Team == "DET", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_det <- det_data |>
  inner_join(det_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

```
Warning in inner_join(det_data, det_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_det <- pairs_det |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_det <- graph_from_data_frame(edges_det, directed = FALSE)

#compute transitivity
trans_det <- transitivity(g_det, type = "global")
trans_det
```

```
[1] 0.717107
```

```
det_network <- ggnetwork(g_det)

ggplot(det_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "navy") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Detroit Tigers Social Network 2015-2025")
```



```
#KCR network
#filter just KCR
kcr_data <- all |>
  filter(Team == "KCR", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_kcr <- kcr_data |>
  inner_join(kcr_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

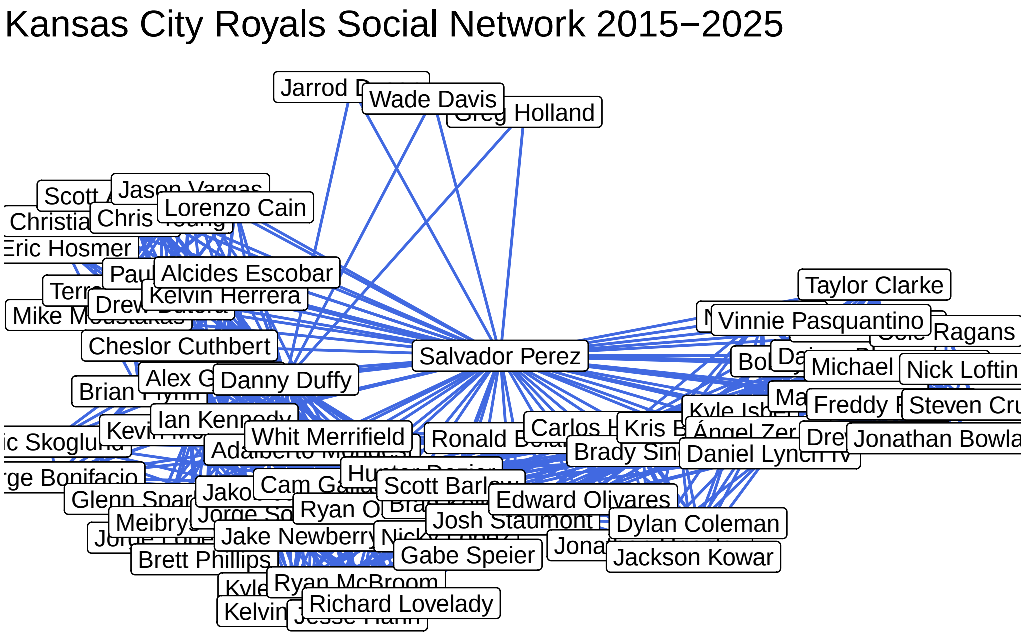
```
Warning in inner_join(kcr_data, kcr_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_kcr <- pairs_kcr |>
  filter(overlap >= 3)
```

```r
#put into graph frame, false to allow for transitivity
g_kcr <- graph_from_data_frame(edges_kcr, directed = FALSE)

#compute transitivity
trans_kcr <- transitivity(g_kcr, type = "global")
trans_kcr
```

```
[1] 0.6826061
```

```r
kcr_network <- ggnetwork(g_kcr)

ggplot(kcr_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "royalblue") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Kansas City Royals Social Network 2015-2025")
```



```r
#MIN network
#filter just MIN
min_data <- all |>
  filter(Team == "MIN", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_min <- min_data |>
  inner_join(min_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

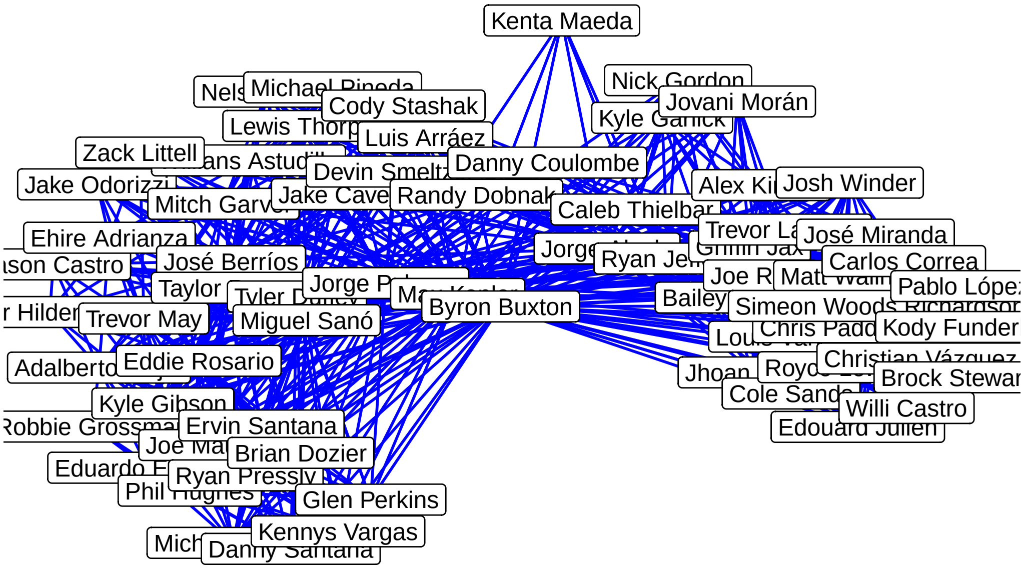
```
Warning in inner_join(min_data, min_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```r
#edge is set to pairs that have at least 3 seasons together
edges_min <- pairs_min |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_min <- graph_from_data_frame(edges_min, directed = FALSE)

#compute transitivity
trans_min <- transitivity(g_min, type = "global")
trans_min
```

```
[1] 0.690672
```

```r
min_network <- ggnetwork(g_min)

ggplot(min_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "blue") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Minnesota Twins Social Network 2015-2025")
```

Minnesota Twins Social Network 2015−2025



```
#CHW network
#filter just CHW
chw_data <- all |>
  filter(Team == "CHW", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_chw <- chw_data |>
  inner_join(chw_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

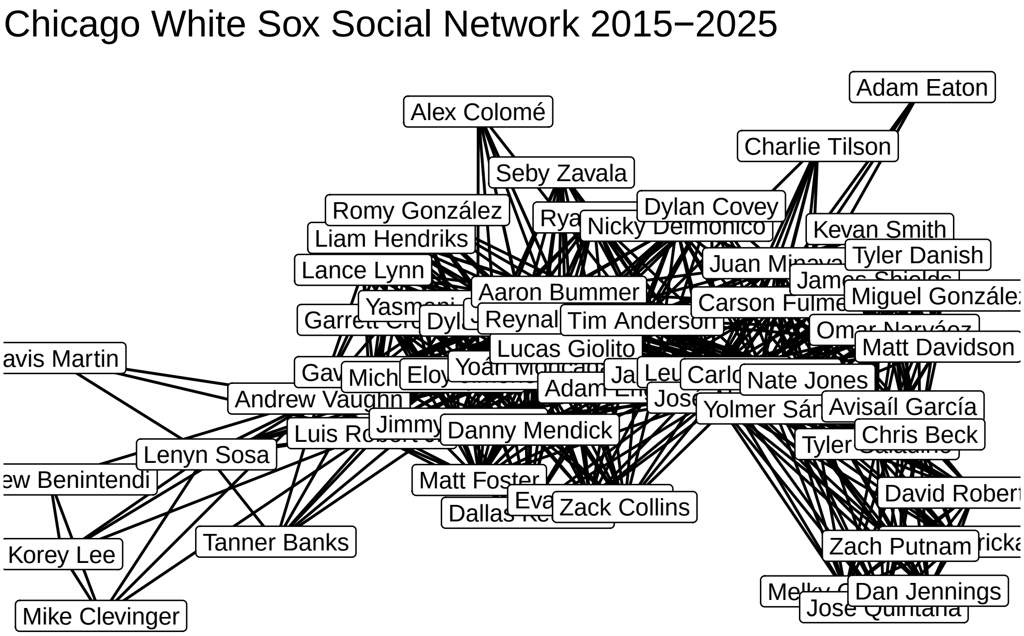
```
Warning in inner_join(chw_data, chw_data, by = "Season"): Detected an unexpected many-to-man
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_chw <- pairs_chw |>
  filter(overlap >= 3)
```

```
#put into graph frame, false to allow for transitivity
g_chw <- graph_from_data_frame(edges_chw, directed = FALSE)

#compute transitivity
trans_chw <- transitivity(g_chw, type = "global")
trans_chw
```

```
[1] 0.6666881
```

```
chw_network <- ggnetwork(g_chw)

ggplot(chw_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "black") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Chicago White Sox Social Network 2015-2025")
```



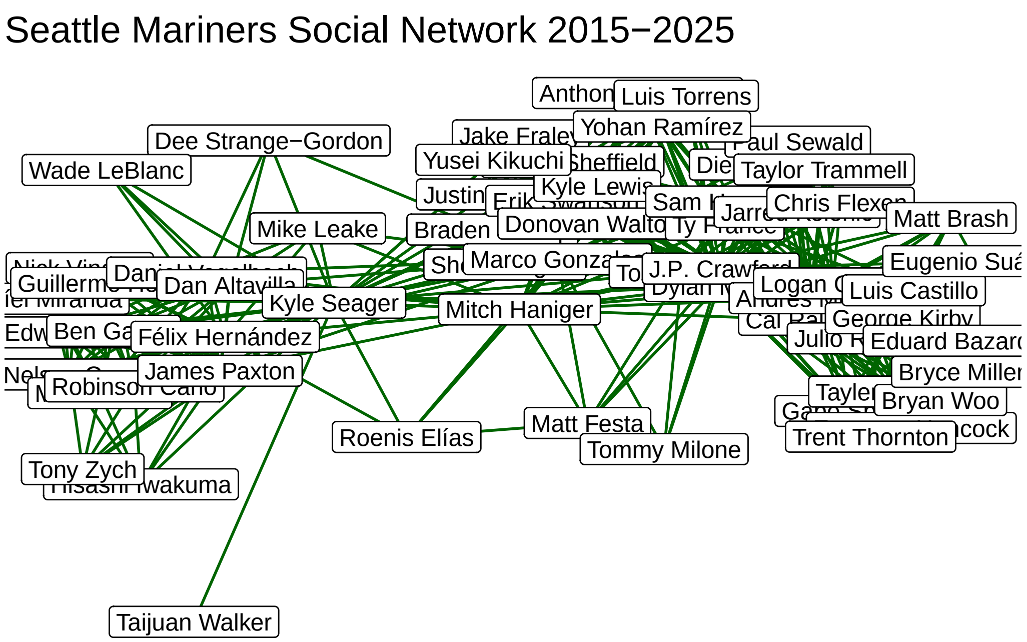
```
#SEA network
#filter just SEA
sea_data <- all |>
  filter(Team == "SEA", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_sea <- sea_data |>
  inner_join(sea_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

```
Warning in inner_join(sea_data, sea_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_sea <- pairs_sea |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_sea <- graph_from_data_frame(edges_sea, directed = FALSE)

#compute transitivity
trans_sea <- transitivity(g_sea, type = "global")
trans_sea
```

```
[1] 0.645417
```

```
sea_network <- ggnetwork(g_sea)

ggplot(sea_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "darkgreen") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Seattle Mariners Social Network 2015-2025")
```


Seattle Mariners Social Network 2015−2025

```
#HOU network
#filter just HOU
hou_data <- all |>
  filter(Team == "HOU", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_hou <- hou_data |>
  inner_join(hou_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

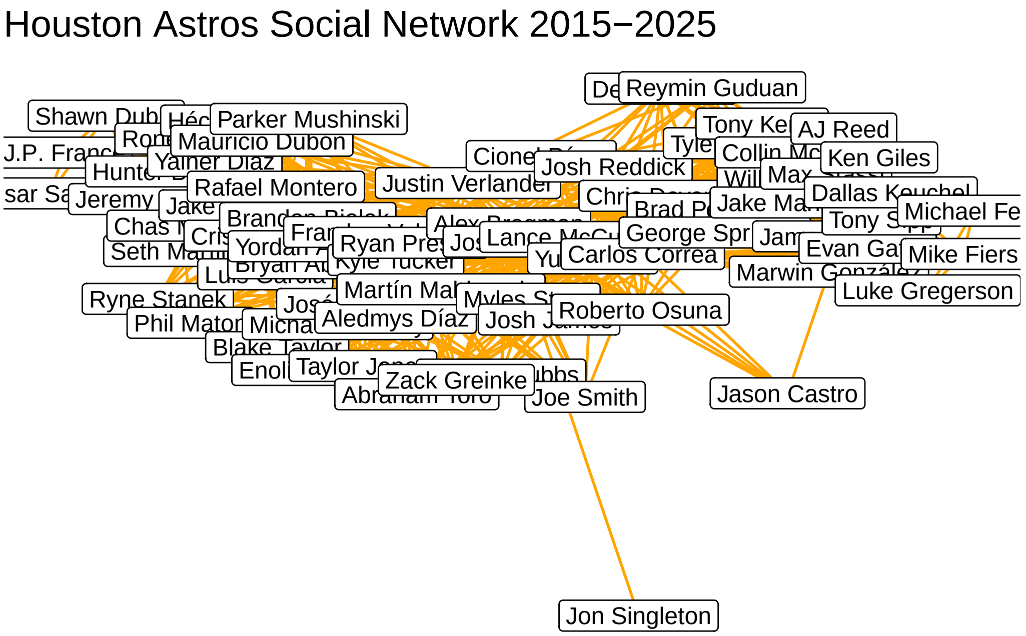
```
Warning in inner_join(hou_data, hou_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_hou <- pairs_hou |>
  filter(overlap >= 3)
```

```
#put into graph frame, false to allow for transitivity
g_hou <- graph_from_data_frame(edges_hou, directed = FALSE)

#compute transitivity
trans_hou <- transitivity(g_hou, type = "global")
trans_hou
```

```
[1] 0.6810345
```

```
hou_network <- ggnetwork(g_hou)

ggplot(hou_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "orange") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Houston Astros Social Network 2015-2025")
```



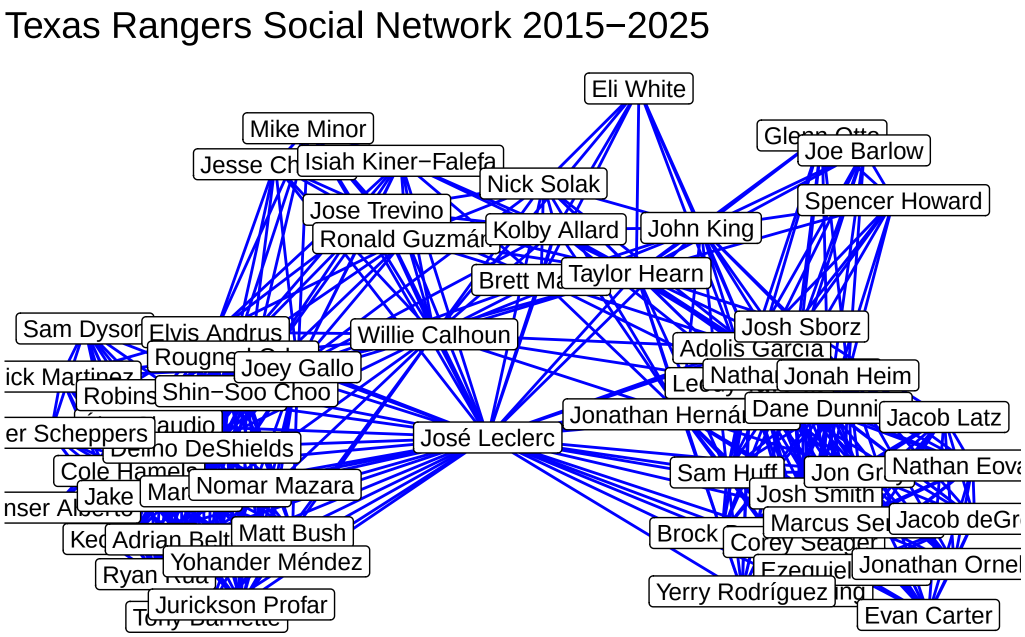
```
#TEX network
#filter just TEX
tex_data <- all |>
  filter(Team == "TEX", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_tex <- tex_data |>
  inner_join(tex_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

```
Warning in inner_join(tex_data, tex_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_tex <- pairs_tex |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_tex <- graph_from_data_frame(edges_tex, directed = FALSE)

#compute transitivity
trans_tex <- transitivity(g_tex, type = "global")
trans_tex
```

```
[1] 0.7467249
```

```
tex_network <- ggnetwork(g_tex)

ggplot(tex_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "blue") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Texas Rangers Social Network 2015-2025")
```



Texas Rangers Social Network 2015−2025

```
#ATH network
#filter just ATH
ath_data <- all |>
  filter(Team == "ATH", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_ath <- ath_data |>
  inner_join(ath_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

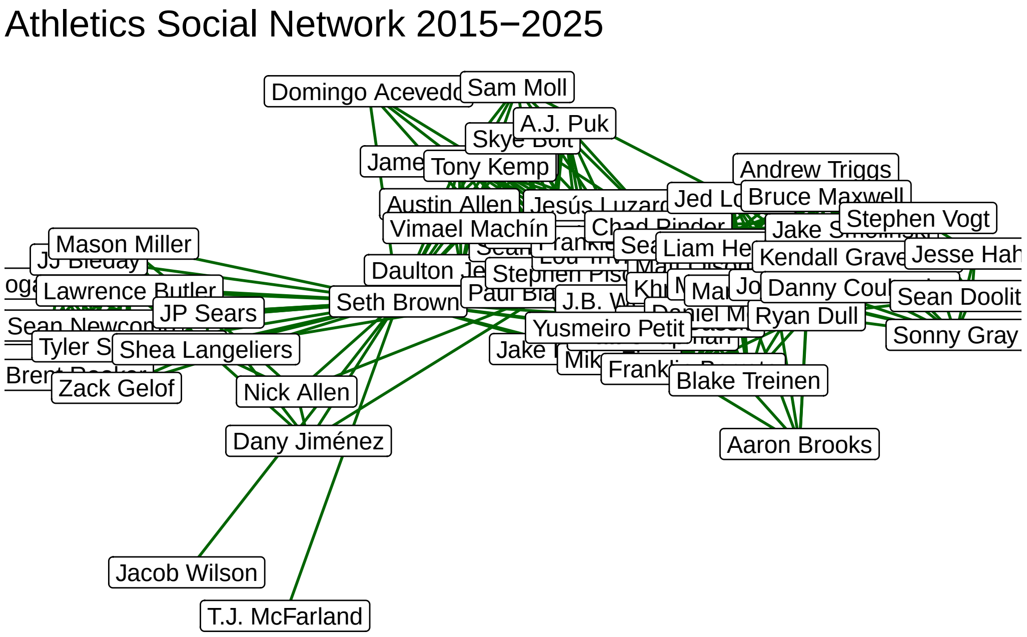
```
Warning in inner_join(ath_data, ath_data, by = "Season"): Detected an unexpected many-to-man
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_ath <- pairs_ath |>
  filter(overlap >= 3)
```

```
#put into graph frame, false to allow for transitivity
g_ath <- graph_from_data_frame(edges_ath, directed = FALSE)

#compute transitivity
trans_ath <- transitivity(g_ath, type = "global")
trans_ath
```

```
[1] 0.7030928
```

```
ath_network <- ggnetwork(g_ath)

ggplot(ath_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "darkgreen") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Athletics Social Network 2015-2025")
```



```
#LAA network
#filter just LAA
laa_data <- all |>
  filter(Team == "LAA", Season >= 2015, Season <= 2025)|>
```

```r
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_laa <- laa_data |>
  inner_join(laa_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

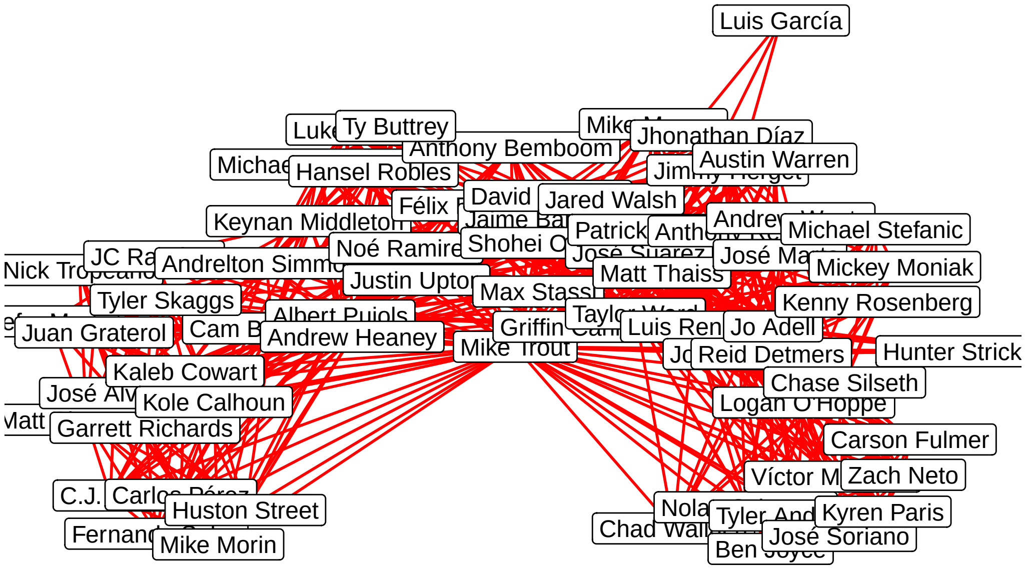
```
Warning in inner_join(laa_data, laa_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```r
#edge is set to pairs that have at least 3 seasons together
edges_laa <- pairs_laa |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_laa <- graph_from_data_frame(edges_laa, directed = FALSE)

#compute transitivity
trans_laa <- transitivity(g_laa, type = "global")
trans_laa
```

```
[1] 0.658548
```

```r
laa_network <- ggnetwork(g_laa)

ggplot(laa_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "red") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Los Angeles Angels Social Network 2015-2025")
```

Los Angeles Angels Social Network 2015−2025



```
#PHI network
#filter just PHI
phi_data <- all |>
  filter(Team == "PHI", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_phi <- phi_data |>
  inner_join(phi_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

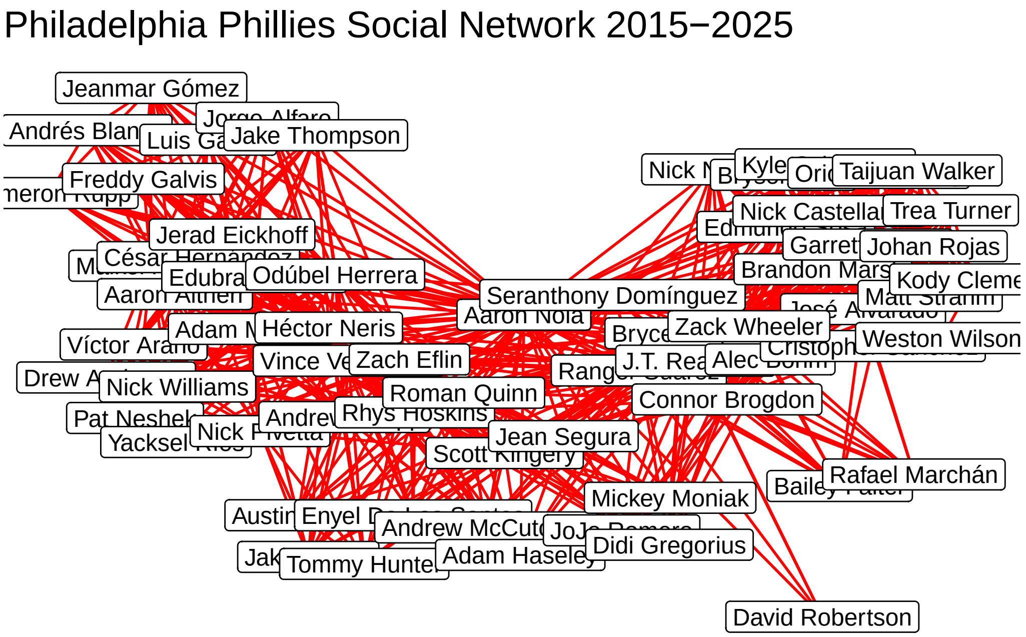
```
Warning in inner_join(phi_data, phi_data, by = "Season"): Detected an unexpected many-to-man
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_phi <- pairs_phi |>
  filter(overlap >= 3)
```

```
#put into graph frame, false to allow for transitivity
g_phi <- graph_from_data_frame(edges_phi, directed = FALSE)

#compute transitivity
trans_phi <- transitivity(g_phi, type = "global")
trans_phi
```

```
[1] 0.6752159
```

```
phi_network <- ggnetwork(g_phi)

ggplot(phi_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "red") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Philadelphia Phillies Social Network 2015-2025")
```



```
#NYM network
#filter just NYM
nym_data <- all |>
```

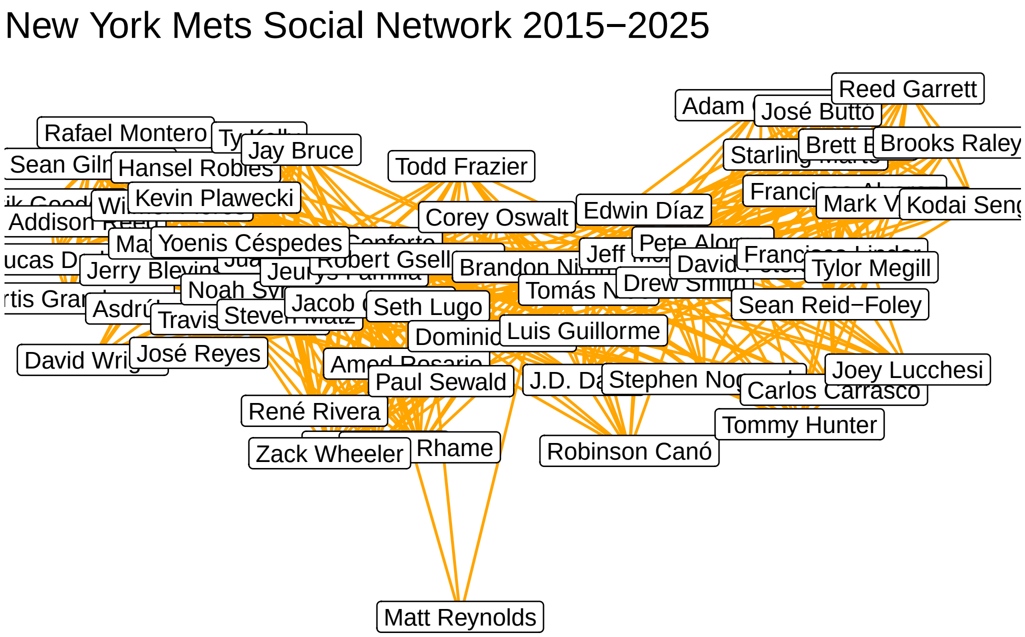
```
  filter(Team == "NYM", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_nym <- nym_data |>
  inner_join(nym_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

```
Warning in inner_join(nym_data, nym_data, by = "Season"): Detected an unexpected many-to-man
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_nym <- pairs_nym |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_nym <- graph_from_data_frame(edges_nym, directed = FALSE)

#compute transitivity
trans_nym <- transitivity(g_nym, type = "global")
trans_nym
```

```
[1] 0.6805505
```

```
nym_network <- ggnetwork(g_nym)

ggplot(nym_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "orange") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("New York Mets Social Network 2015-2025")
```



```
#MIA network
#filter just MIA
mia_data <- all |>
  filter(Team == "MIA", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_mia <- mia_data |>
  inner_join(mia_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

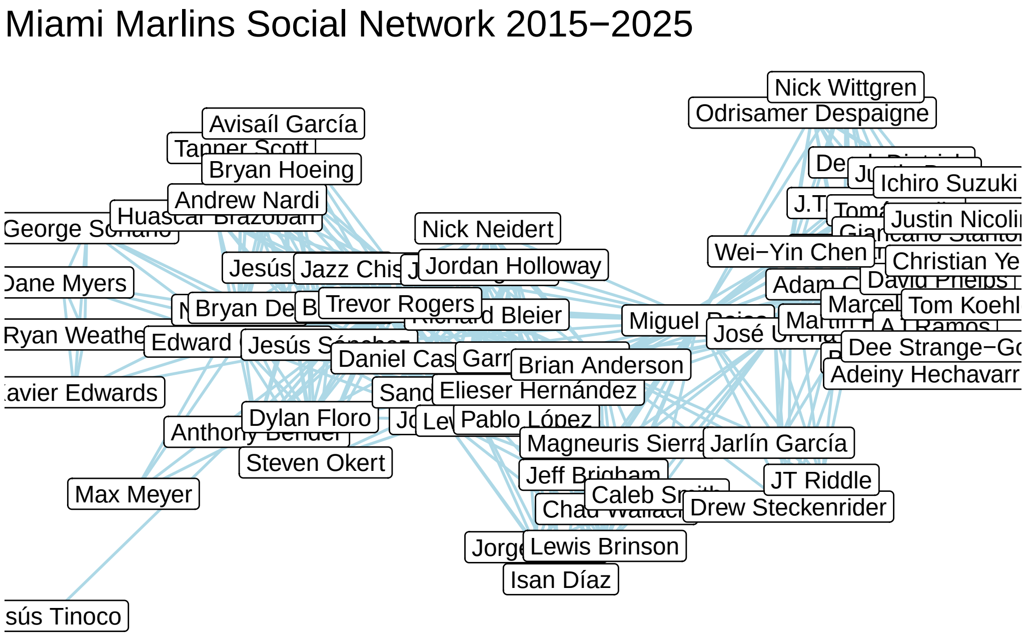
```
Warning in inner_join(mia_data, mia_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_mia <- pairs_mia |>
  filter(overlap >= 3)
```

```
#put into graph frame, false to allow for transitivity
g_mia <- graph_from_data_frame(edges_mia, directed = FALSE)

#compute transitivity
trans_mia <- transitivity(g_mia, type = "global")
trans_mia
```

```
[1] 0.7153488
```

```
mia_network <- ggnetwork(g_mia)

ggplot(mia_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "lightblue") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Miami Marlins Social Network 2015−2025")
```



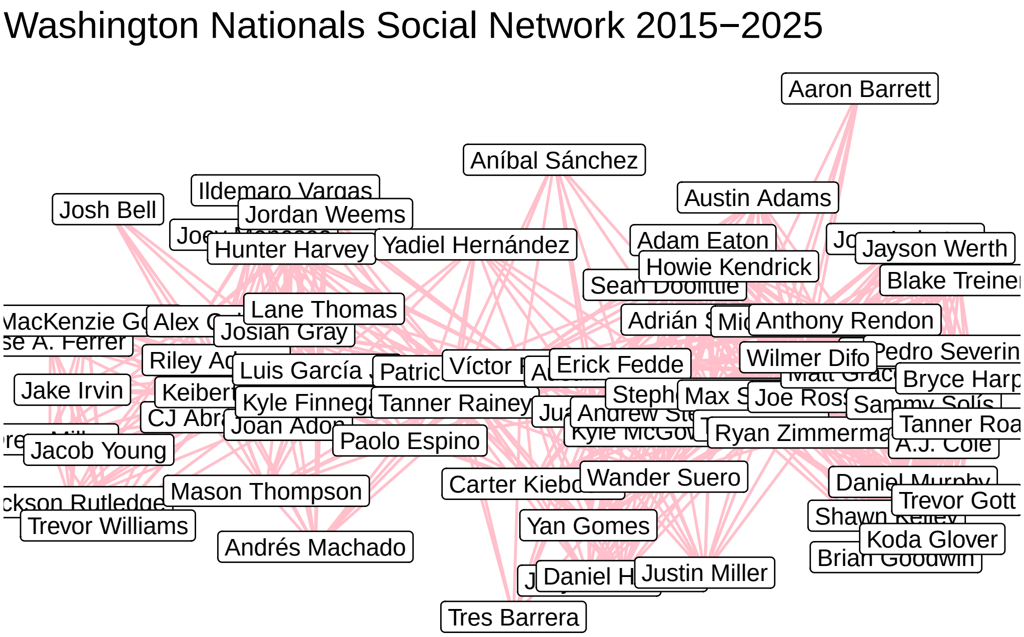
```
#WSN network
#filter just WSN
wsn_data <- all |>
  filter(Team == "WSN", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_wsn <- wsn_data |>
  inner_join(wsn_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

```
Warning in inner_join(wsn_data, wsn_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_wsn <- pairs_wsn |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_wsn <- graph_from_data_frame(edges_wsn, directed = FALSE)

#compute transitivity
trans_wsn <- transitivity(g_wsn, type = "global")
trans_wsn
```

```
[1] 0.6900754
```

```
wsn_network <- ggnetwork(g_wsn)

ggplot(wsn_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "pink") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Washington Nationals Social Network 2015-2025")
```



```r
#MIL network
#filter just MIL
mil_data <- all |>
  filter(Team == "MIL", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_mil <- mil_data |>
  inner_join(mil_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

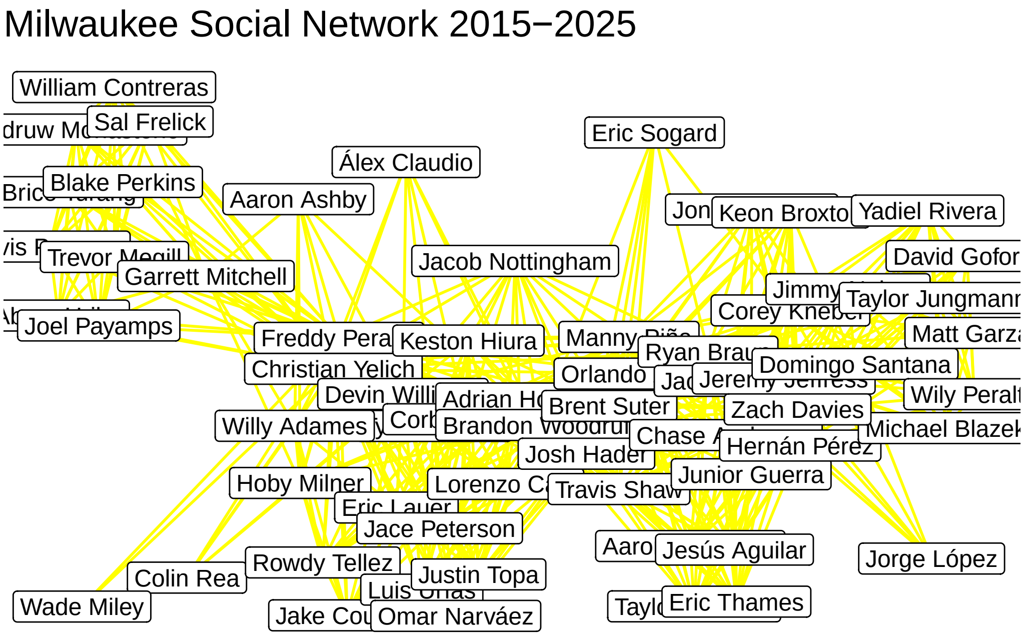
```
Warning in inner_join(mil_data, mil_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```r
#edge is set to pairs that have at least 3 seasons together
edges_mil <- pairs_mil |>
  filter(overlap >= 3)
```

```r
#put into graph frame, false to allow for transitivity
g_mil <- graph_from_data_frame(edges_mil, directed = FALSE)

#compute transitivity
trans_mil <- transitivity(g_mil, type = "global")
trans_mil
```

```
[1] 0.6627702
```

```r
mil_network <- ggnetwork(g_mil)

ggplot(mil_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "yellow") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Milwaukee Social Network 2015-2025")
```



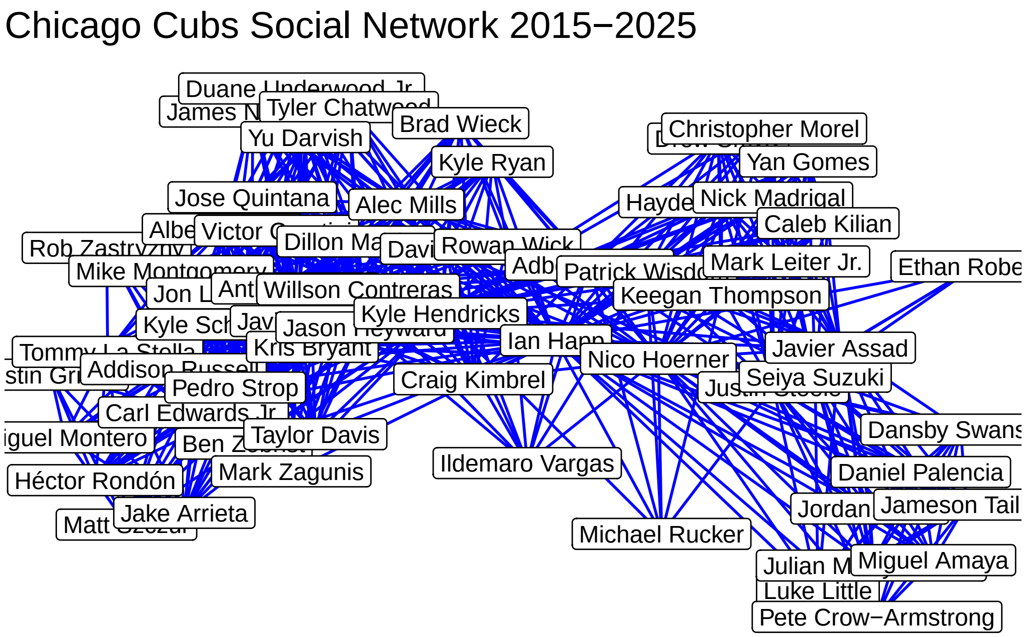
```r
#CHC network
#filter just CHC
chc_data <- all |>
  filter(Team == "CHC", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_chc <- chc_data |>
  inner_join(chc_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

```
Warning in inner_join(chc_data, chc_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```r
#edge is set to pairs that have at least 3 seasons together
edges_chc <- pairs_chc |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_chc <- graph_from_data_frame(edges_chc, directed = FALSE)

#compute transitivity
trans_chc <- transitivity(g_chc, type = "global")
trans_chc
```

```
[1] 0.684502
```

```r
chc_network <- ggnetwork(g_chc)

ggplot(chc_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "blue") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Chicago Cubs Social Network 2015-2025")
```


Chicago Cubs Social Network 2015−2025

```
#CIN network
#filter just CIN
cin_data <- all |>
  filter(Team == "CIN", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_cin <- cin_data |>
  inner_join(cin_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

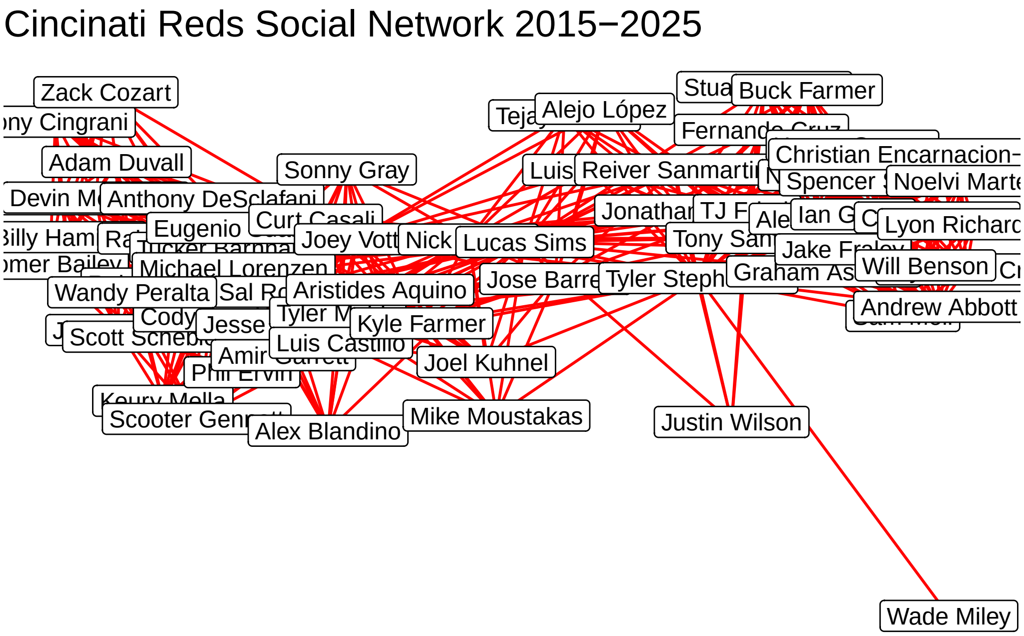
```
Warning in inner_join(cin_data, cin_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_cin <- pairs_cin |>
  filter(overlap >= 3)
```

```
#put into graph frame, false to allow for transitivity
g_cin <- graph_from_data_frame(edges_cin, directed = FALSE)

#compute transitivity
trans_cin <- transitivity(g_cin, type = "global")
trans_cin
```

```
[1] 0.7213327
```

```
cin_network <- ggnetwork(g_cin)

ggplot(cin_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "red") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Cincinati Reds Social Network 2015-2025")
```



```
#STL network
#filter just STL
stl_data <- all |>
  filter(Team == "STL", Season >= 2015, Season <= 2025)|>
```

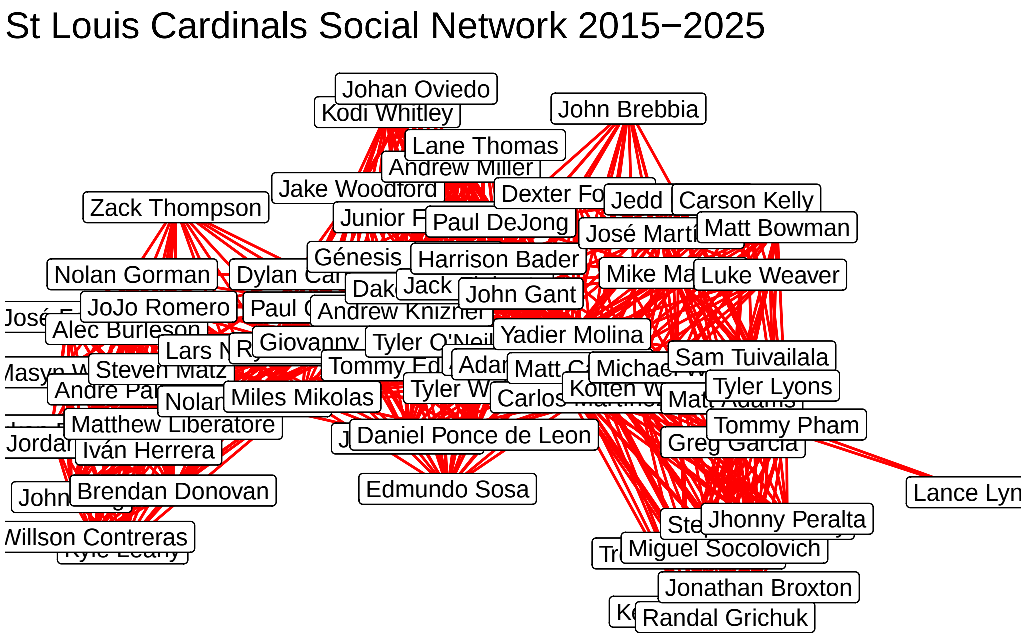
```r
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_stl <- stl_data |>
  inner_join(stl_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

```
Warning in inner_join(stl_data, stl_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```r
#edge is set to pairs that have at least 3 seasons together
edges_stl <- pairs_stl |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_stl <- graph_from_data_frame(edges_stl, directed = FALSE)

#compute transitivity
trans_stl <- transitivity(g_stl, type = "global")
trans_stl
```

```
[1] 0.7250806
```

```r
stl_network <- ggnetwork(g_stl)

ggplot(stl_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "red") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("St Louis Cardinals Social Network 2015-2025")
```



St Louis Cardinals Social Network 2015−2025

```
#PIT network
#filter just PIT
pit_data <- all |>
  filter(Team == "PIT", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_pit <- pit_data |>
  inner_join(pit_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

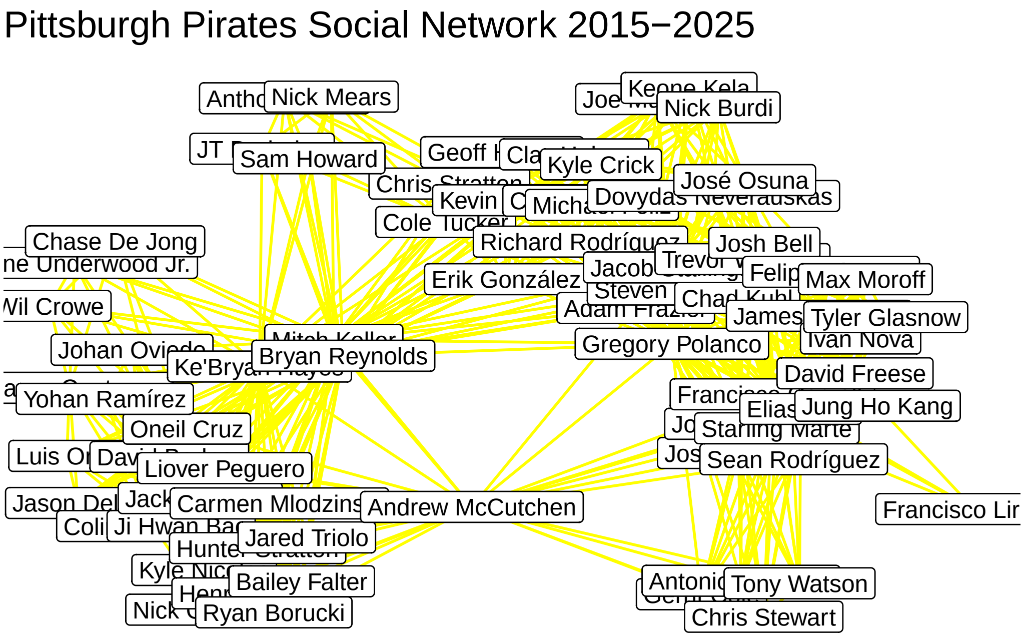
```
Warning in inner_join(pit_data, pit_data, by = "Season"): Detected an unexpected many-to-man
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_pit <- pairs_pit |>
  filter(overlap >= 3)
```

```r
#put into graph frame, false to allow for transitivity
g_pit <- graph_from_data_frame(edges_pit, directed = FALSE)

#compute transitivity
trans_pit <- transitivity(g_pit, type = "global")
trans_pit
```

```
[1] 0.6945594
```

```r
pit_network <- ggnetwork(g_pit)

ggplot(pit_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "yellow") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Pittsburgh Pirates Social Network 2015-2025")
```



Pittsburgh Pirates Social Network 2015−2025

```r
#SDP network
#filter just SDP
sdp_data <- all |>
```

```r
  filter(Team == "SDP", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_sdp <- sdp_data |>
  inner_join(sdp_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

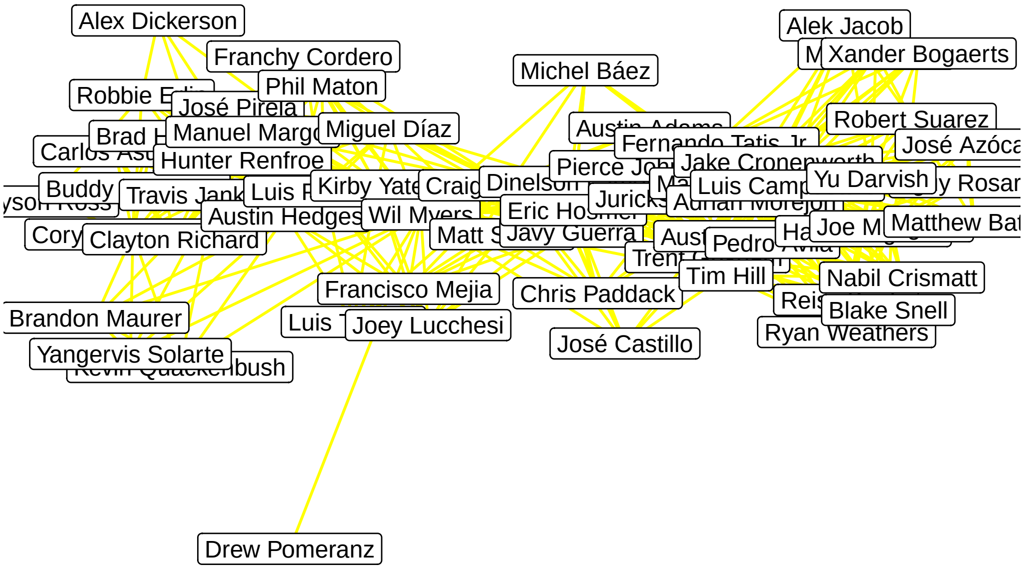
```
Warning in inner_join(sdp_data, sdp_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```r
#edge is set to pairs that have at least 3 seasons together
edges_sdp <- pairs_sdp |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_sdp <- graph_from_data_frame(edges_sdp, directed = FALSE)

#compute transitivity
trans_sdp <- transitivity(g_sdp, type = "global")
trans_sdp
```

```
[1] 0.630527
```

```r
sdp_network <- ggnetwork(g_sdp)

ggplot(sdp_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "yellow") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("San Diego Padres Social Network 2015-2025")
```

San Diego Padres Social Network 2015−2025



```
#SFG network
#filter just SFG
sfg_data <- all |>
  filter(Team == "SFG", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_sfg <- sfg_data |>
  inner_join(sfg_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

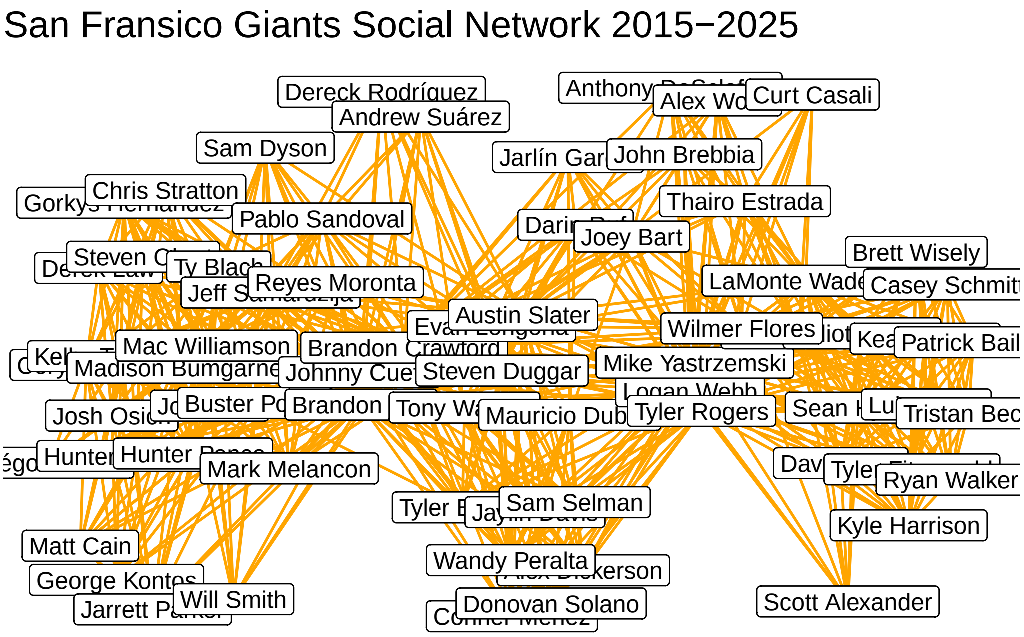
```
Warning in inner_join(sfg_data, sfg_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_sfg <- pairs_sfg |>
  filter(overlap >= 3)
```

```
#put into graph frame, false to allow for transitivity
g_sfg <- graph_from_data_frame(edges_sfg, directed = FALSE)

#compute transitivity
trans_sfg <- transitivity(g_sfg, type = "global")
trans_sfg
```

```
[1] 0.6534442
```

```
sfg_network <- ggnetwork(g_sfg)

ggplot(sfg_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "orange") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("San Fransico Giants Social Network 2015-2025")
```



```
#ARI network
#filter just ARI
ari_data <- all |>
```

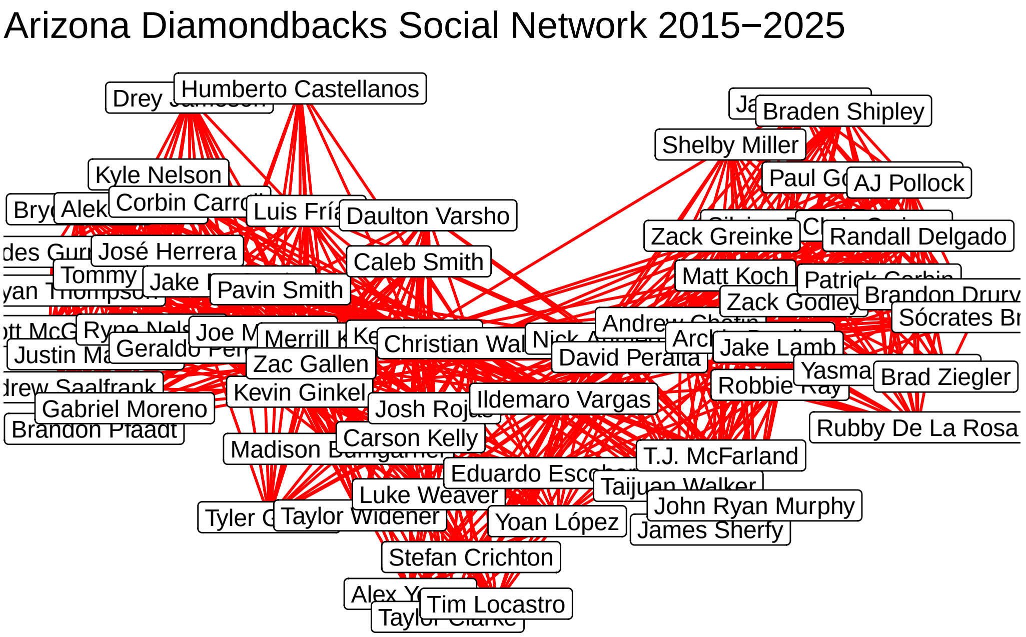
```
  filter(Team == "ARI", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_ari <- ari_data |>
  inner_join(ari_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

```
Warning in inner_join(ari_data, ari_data, by = "Season"): Detected an unexpected many-to-man
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```
#edge is set to pairs that have at least 3 seasons together
edges_ari <- pairs_ari |>
  filter(overlap >= 3)

#put into graph frame, false to allow for transitivity
g_ari <- graph_from_data_frame(edges_ari, directed = FALSE)

#compute transitivity
trans_ari <- transitivity(g_ari, type = "global")
trans_ari
```

```
[1] 0.6916697
```

```
ari_network <- ggnetwork(g_ari)

ggplot(ari_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "red") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Arizona Diamondbacks Social Network 2015-2025")
```



```r
#COL network
#filter just COL
col_data <- all |>
  filter(Team == "COL", Season >= 2015, Season <= 2025)|>
  distinct(Player, Season)

#calculate overlap for every pair (edges and weights)
pairs_col <- col_data |>
  inner_join(col_data, by = "Season") |>
  filter(Player.x < Player.y) |>
  #removes duplicates and opposite combos
  count(Player.x, Player.y, name = "overlap")
```

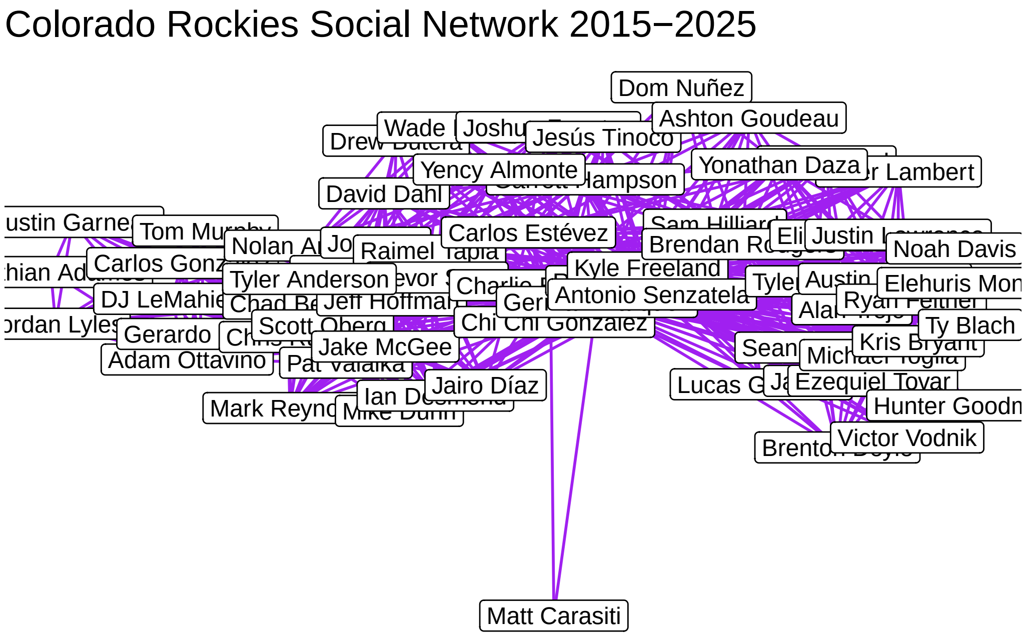
```
Warning in inner_join(col_data, col_data, by = "Season"): Detected an unexpected many-to-many
i Row 1 of `x` matches multiple rows in `y`.
i Row 1 of `y` matches multiple rows in `x`.
i If a many-to-many relationship is expected, set `relationship =
  "many-to-many"` to silence this warning.
```

```r
#edge is set to pairs that have at least 3 seasons together
edges_col <- pairs_col |>
  filter(overlap >= 3)
```

```r
#put into graph frame, false to allow for transitivity
g_col <- graph_from_data_frame(edges_col, directed = FALSE)

#compute transitivity
trans_col <- transitivity(g_col, type = "global")
trans_col
```

```
[1] 0.6650041
```

```r
col_network <- ggnetwork(g_col)

ggplot(col_network,
       aes(x = x, y = y, xend = xend, yend = yend)) +
  geom_edges(color = "purple") +
  geom_nodes() +
  geom_nodelabel(aes(label = name), size = 3) +
  theme_blank() +
  ggtitle("Colorado Rockies Social Network 2015-2025")
```



```r
#connecting all team codes to transitivity scores
transitivity_scores <- tibble(
  Team = c("NYY", "TOR", "BAL", "BOS", "ATL", "TBR", "LAD", "CLE", "DET", "KCR",
           "MIN", "CHW", "SEA", "HOU", "TEX", "ATH", "LAA", "PHI", "NYM", "MIA",
           "WSN", "MIL", "CHC", "CIN", "STL", "PIT", "SDP", "SFG", "ARI", "COL"),
  Transitivity = c(trans_nyy, trans_tor, trans_bal, trans_bos, trans_atl,
                   trans_tbr, trans_lad, trans_cle, trans_det, trans_kcr,
                   trans_min, trans_chw, trans_sea, trans_hou, trans_tex,
                   trans_ath, trans_laa, trans_phi, trans_nym, trans_mia,
                   trans_wsn, trans_mil, trans_chc, trans_cin, trans_stl,
                   trans_pit, trans_sdp, trans_sfg, trans_ari, trans_col))
```

#Transitivity Score Table

```
#transitivity score table
transitivity_scores |>
  arrange(desc(Transitivity)) |>
  kable(
    caption = "Team Transitivity Scores (2015-2025)",
    booktabs = TRUE
  )
```

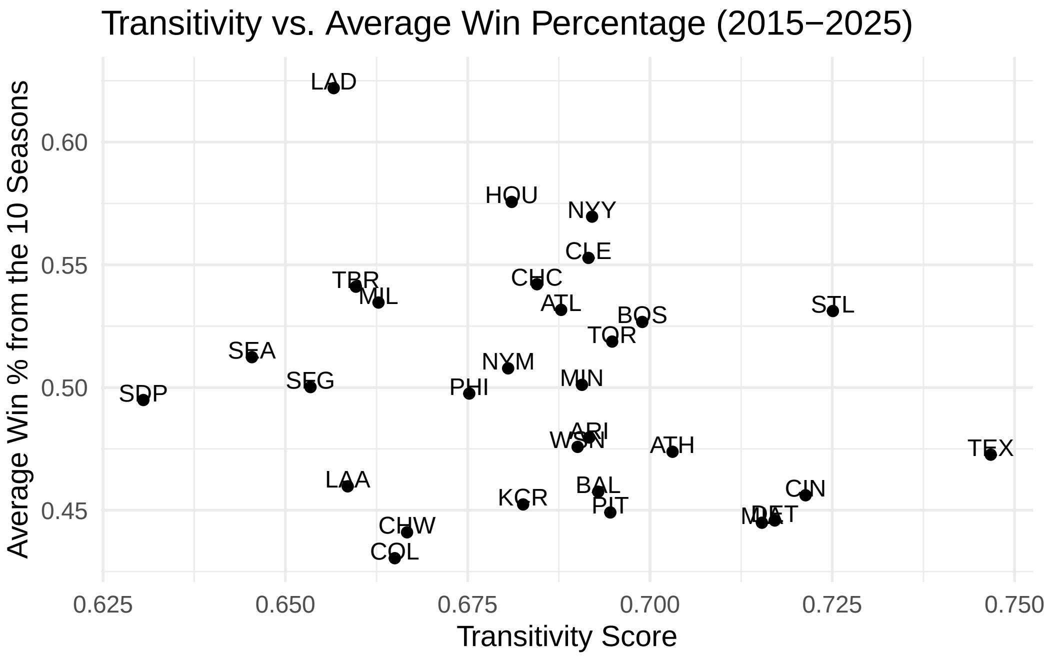
Table 1: Team Transitivity Scores (2015–2025)

Team	Transitivity
TEX	0.7467249
STL	0.7250806
CIN	0.7213327
DET	0.7171070
MIA	0.7153488
ATH	0.7030928
BOS	0.6989435
TOR	0.6948120
PIT	0.6945594
BAL	0.6928982
NYY	0.6920599
ARI	0.6916697
CLE	0.6915636
MIN	0.6906720
WSN	0.6900754
ATL	0.6878162
CHC	0.6845020
KCR	0.6826061
HOU	0.6810345
NYM	0.6805505

Team	Transitivity
PHI	0.6752159
CHW	0.6666881
COL	0.6650041
MIL	0.6627702
TBR	0.6596733
LAA	0.6585480
LAD	0.6566315
SFG	0.6534442
SEA	0.6454170
SDP	0.6305270

##SLR transivitiy vs winning%

```r
# Fix team codes in avg_winpct
avg_winpct <- avg_winpct |>
  mutate(
    Team = case_when(
      str_detect(Tm, "Los Angeles Angels") ~ "LAA",
      str_detect(Tm, "Cleveland Guardians") ~ "CLE",
      str_detect(Tm, "Athletics") ~ "ATH",
      str_detect(Tm, "Tampa Bay Rays") ~ "TBR",
      str_detect(Tm, "Arizona Diamondbacks") ~ "ARI",
      str_detect(Tm, "Atlanta Braves") ~ "ATL",
      str_detect(Tm, "Baltimore Orioles") ~ "BAL",
      str_detect(Tm, "Boston Red Sox") ~ "BOS",
      str_detect(Tm, "Chicago Cubs") ~ "CHC",
      str_detect(Tm, "Chicago White Sox") ~ "CHW",
      str_detect(Tm, "Cincinnati Reds") ~ "CIN",
      str_detect(Tm, "Colorado Rockies") ~ "COL",
      str_detect(Tm, "Detroit Tigers") ~ "DET",
      str_detect(Tm, "Houston Astros") ~ "HOU",
      str_detect(Tm, "Kansas City Royals") ~ "KCR",
      str_detect(Tm, "Los Angeles Dodgers") ~ "LAD",
      str_detect(Tm, "Miami Marlins") ~ "MIA",
      str_detect(Tm, "Milwaukee Brewers") ~ "MIL",
      str_detect(Tm, "Minnesota Twins") ~ "MIN",
      str_detect(Tm, "New York Mets") ~ "NYM",
      str_detect(Tm, "New York Yankees") ~ "NYY",
      str_detect(Tm, "Philadelphia Phillies") ~ "PHI",
      str_detect(Tm, "Pittsburgh Pirates") ~ "PIT",
      str_detect(Tm, "San Diego Padres") ~ "SDP",
      str_detect(Tm, "San Francisco Giants") ~ "SFG",
      str_detect(Tm, "Seattle Mariners") ~ "SEA",
      str_detect(Tm, "St. Louis Cardinals") ~ "STL",
      str_detect(Tm, "Texas Rangers") ~ "TEX",
      str_detect(Tm, "Toronto Blue Jays") ~ "TOR",
      str_detect(Tm, "Washington Nationals") ~ "WSN",
      TRUE ~ Tm
    )
  )
#merge datasets by team
slr_data <- transitivity_scores |>
  inner_join(avg_winpct, by = "Team")

#scatterplot to show relationship
ggplot(slr_data, aes(x = Transitivity, y = AvgWinPct, label = Team)) +
  geom_point() +
  geom_text(nudge_y = 0.003, size = 3) +
  theme_minimal() +
  labs(
    title = "Transitivity vs. Average Win Percentage (2015-2025)",
    x = "Transitivity Score",
    y = "Average Win % from the 10 Seasons"
  )
```



```
#testing if we can do linear regression, we should not
slr <- lm(AvgWinPct ~ Transitivity, data = slr_data)
summary(slr)
```

```
Call:
lm(formula = AvgWinPct ~ Transitivity, data = slr_data)

Residuals:
      Min        1Q    Median        3Q       Max 
-0.077810 -0.036710 -0.005235  0.031959  0.110297 

Coefficients:
             Estimate Std. Error t value Pr(>|t|)   
(Intercept)    0.7813     0.2320   3.368  0.00222 **
Transitivity  -0.4106     0.3384  -1.214  0.23507   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.04628 on 28 degrees of freedom
Multiple R-squared:  0.04997,	Adjusted R-squared:  0.01604 
F-statistic: 1.473 on 1 and 28 DF,  p-value: 0.2351
```

```
cor(AvgWinPct ~ Transitivity, data = slr_data)
```

```
[1] -0.2235319
```

```
graph_list <- list(
  NYY = g_nyy,
  BOS = g_bos,
  BAL = g_bal,
  TOR = g_tor,
  TBR = g_tbr,
  CLE = g_cle,
  DET = g_det,
  KCR = g_kcr,
  MIN = g_min,
  CHW = g_chw,
  SEA = g_sea,
  HOU = g_hou,
  TEX = g_tex,
  ATH = g_ath,
  LAA = g_laa,
  PHI = g_phi,
  NYM = g_nym,
  MIA = g_mia,
  WSN = g_wsn,
  MIL = g_mil,
  CHC = g_chc,
  CIN = g_cin,
  STL = g_stl,
  PIT = g_pit,
  SDP = g_sdp,
  SFG = g_sfg,
  ARI = g_ari,
  COL = g_col,
  LAD = g_lad,
  ATL = g_atl
)
```

```
saveRDS(graph_list, "networks.rds")
saveRDS(transitivity_scores, "transitivity.rds")
saveRDS(slr_data, "slr_data.rds")
```