

Untitled5

```
library(dplyr)
```

Attaching package: 'dplyr'

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

filter, lag

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

intersect, setdiff, setequal, union

```
library(tidyr)
library(stringr)
library(igraph)
```

Attaching package: 'igraph'

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

crossing

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

as_data_frame, groups, union

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

```
decompose, spectrum
```

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

```
union
```

```
library(showtext)
```

```
Loading required package: sysfonts
```

```
Loading required package: showtextdb
```

```
showtext_auto()
```

```
## 1. Load data -----
df <- read.csv("data.csv", stringsAsFactors = FALSE)

## 2. Build edge list with NO double edges -----
edges <- df %>%
  select(name, collaborators) %>%
  filter(!is.na(collaborators), collaborators != "") %>%
  separate_rows(collaborators, sep = ",") %>%
  mutate(collaborators = str_trim(collaborators)) %>%
  filter(collaborators != "", collaborators != name) %>%
  mutate(
    artist1 = pmin(name, collaborators),
    artist2 = pmax(name, collaborators)
  ) %>%
  distinct(artist1, artist2) %>%
  rename(from = artist1, to = artist2)

## 3. All artists (vertices) -----
all_artists <- sort(unique(c(df$name, edges$from, edges$to)))
vertices <- data.frame(name = all_artists, stringsAsFactors = FALSE)

## 3a. User info for SHAPES -----
user_attr <- df %>%
  select(name, user) %>%
  filter(!is.na(user), user != "") %>%
```

```

    mutate(user = gsub("\\s+", "", user)) %>%
  separate_rows(user, sep = ",") %>%
  distinct(name, user) %>%
  group_by(name) %>%
  summarise(user = paste(sort(unique(user)), collapse = ","), .groups = "drop")

## 3b. Genre → MAIN GENRE mapping -----
genre_map <- df %>%
  select(name, genre) %>%
  filter(!is.na(genre), genre != "") %>%
  separate_rows(genre, sep = ",") %>%
  mutate(
    genre = str_trim(genre),
    genre_lower = str_to_lower(genre),
    main_genre = case_when(
      str_detect(genre_lower, "hip hop|hiphop|rap|trap|drill|grime|phonk|gangsta|turntablism")
        ~ "Hip-Hop / Rap",
      str_detect(genre_lower, "r&b|soul|funk|motown")
        ~ "R&B / Soul / Funk",
      str_detect(genre_lower, "pop")
        ~ "Pop",
      str_detect(genre_lower,
                 "rock|metal|punk|emo|grunge|shoegaze|hardcore|mathcore|screamo|post-hardcore")
        ~ "Rock / Metal / Punk",
      str_detect(genre_lower,
                 "edm|electronic|electro|house|techno|trance|dubstep|drum and bass|dnb|vaporwave")
        ~ "Electronic / Dance",
      str_detect(genre_lower, "jazz|blues|swing")
        ~ "Jazz / Blues / Swing",
      str_detect(genre_lower, "country|folk|singer songwriter|americana|bluegrass|roots")
        ~ "Country / Folk / Songwriter",
      str_detect(genre_lower, "afro|latin|reggae|dancehall|soca|reggaeton|bossa|cumbia|samba")
        ~ "Latin / World / Afro / Reggae",
      str_detect(genre_lower, "classical|orchestral|opera|soundtrack|score|chamber|symphony")
        ~ "Classical / Score / Soundtrack",
      TRUE
        ~ "Other / Indie / Experimental"
    )
  ) %>%
  distinct(name, main_genre)

artist_genre_main <- genre_map %>%

```

```

group_by(name) %>%
  summarise(main_genre = first(main_genre), .groups = "drop")

## 3c. Attach user + main_genre + SHAPES to vertices -----
vertices <- vertices %>%
  left_join(user_attr, by = "name") %>%
  left_join(artist_genre_main, by = "name") %>%
  mutate(
    user_clean = ifelse(user == "" | is.na(user), NA, user),
    shape = dplyr::case_when(
      user_clean == "I" ~ "square",
      user_clean == "C" ~ "circle",
      user_clean == "M" ~ "rectangle",
      !is.na(user_clean) ~ "pie",    # any combination
      TRUE ~ "circle"
    )
  )

## 4. Create undirected igraph -----
g <- graph_from_data_frame(
  d = edges,
  vertices = vertices,
  directed = FALSE
)

V(g)$user      <- vertices$user[match(V(g)$name, vertices$name)]
V(g)$shape     <- vertices$shape[match(V(g)$name, vertices$name)]
V(g)$main_genre <- vertices$main_genre[match(V(g)$name, vertices$name)]

## 4b. Color palette for MAIN GENRES -----
main_genre_levels <- c(
  "Hip-Hop / Rap",
  "R&B / Soul / Funk",
  "Pop",
  "Rock / Metal / Punk",
  "Electronic / Dance",
  "Jazz / Blues / Swing",
  "Country / Folk / Songwriter",
  "Latin / World / Afro / Reggae",
  "Classical / Score / Soundtrack",
  "Other / Indie / Alternative / Experimental"
)

```

```

genre_colors <- c(
  "Hip-Hop / Rap" = "goldenrod1",
  "R&B / Soul / Funk" = "mediumorchid",
  "Pop" = "deepskyblue3",
  "Rock / Metal / Punk" = "firebrick2",
  "Electronic / Dance" = "springgreen3",
  "Jazz / Blues / Swing" = "darkorange2",
  "Country / Folk / Songwriter" = "sienna3",
  "Latin / World / Afro / Reggae" = "darkolivegreen3",
  "Classical / Score / Soundtrack" = "steelblue4",
  "Other / Indie / Alternative / Experimental" = "grey50"
)

## 4c. Pie support for combo user types -----
V(g)$pie <- replicate(vcount(g), c(1), simplify = FALSE)

V(g)$pie.color <- lapply(seq_len(vcount(g)), function(i) {
  mg <- V(g)$main_genre[i]
  col <- if (!is.na(mg) && mg %in% names(genre_colors)) genre_colors[[mg]] else "cyan"
  c(col)
})

## 5. Isolates in FULL graph -----
deg <- degree(g)
iso_verts <- V(g)[deg == 0]
V(g)$deg <- deg # keep degree as vertex attribute

## 6. Build PDF: legend + genre neighbourhoods + isolates -----
pdf("genre_communities_with_isolates_fixed.pdf", width = 10, height = 10)

## PAGE 1: LEGEND -----
par(mar = c(1, 1, 3, 1))
plot.new()

legend(
  "topleft",
  legend = c(
    "C only (circle)",
    "I only (square)",
    "M only (rectangle)",
    "Combination (I/C/M) (pie)"
  ),
  pch = c(21, 22, 22, 21),

```

```

  pt.bg  = "grey80",
  pt.cex = 1.4,
  title  = "Node shape = user type",
  bty    = "n"
)

legend(
  "bottomleft",
  legend = main_genre_levels,
  pch    = 15,
  col    = genre_colors[main_genre_levels],
  pt.cex = 1.4,
  title  = "Node color = main genre",
  bty    = "n"
)

title("Legend", line = 1)

## PAGES: ONE PAGE PER MAIN GENRE (only connected artists) -----
set.seed(123)

for (mg in main_genre_levels) {

  # primary = artists in this main genre that actually have at least 1 edge
  primary <- V(g)[!is.na(main_genre) & main_genre == mg & deg > 0]

  if (length(primary) == 0) next # no connected artists for this genre

  # 1-step ego neighbourhood: those artists + all their neighbours (any genre)
  neigh_list <- ego(g, order = 1, nodes = primary, mode = "all")
  vids <- sort(unique(unlist(neigh_list)))
  subg <- induced_subgraph(g, vids = vids)

  if (ecount(subg) == 0) next # safety

  coords <- layout_with_fr(subg)

  main_g_sub <- V(subg)$main_genre
  vcols <- genre_colors[main_g_sub]
  vcols[is.na(vcols)] <- "grey80"

  par(mar = c(0, 0, 3, 0))
}

```

```

plot(
  subg,
  layout      = coords,
  vertex.size = 5,
  vertex.label = V(subg)$name,
  vertex.label.cex = 0.7,
  vertex.label.color = "black",
  vertex.shape = V(subg)$shape,
  vertex.color = vcols,
  vertex.pie = V(subg)$pie,
  vertex.pie.color = V(subg)$pie.color,
  edge.width = 0.7,
  edge.color = "grey80",
  main = paste0(
    "Artist Collaboration Network - ", mg,
    " (n = ", vcount(subg), ", edges = ", ecount(subg), ")"
  )
)
}

## FINAL PAGE: ALL ISOLATED ARTISTS -----
if (length(iso_verts) > 0) {
  sub_iso   <- induced_subgraph(g, vids = iso_verts)
  coords_iso <- layout_in_circle(sub_iso)

  main_g_iso <- V(sub_iso)$main_genre
  vcols_iso  <- genre_colors[main_g_iso]
  vcols_iso[is.na(vcols_iso)] <- "grey80"

  par(mar = c(0, 0, 3, 0))
  plot(
    sub_iso,
    layout      = coords_iso,
    vertex.size = 5,
    vertex.label = V(sub_iso)$name,
    vertex.label.cex = 0.7,
    vertex.label.color = "black",
    vertex.shape = V(sub_iso)$shape,
    vertex.color = vcols_iso,
    vertex.pie = V(sub_iso)$pie,
    vertex.pie.color = V(sub_iso)$pie.color,
    edge.width = 0.7,

```

```

    edge.color      = "grey80",
    main = paste0(
      "Isolated Artists (no collaborations in full network) - n = ", vcount(sub_iso)
    )
  )
}

dev.off()

```

pdf
2

```

library(dplyr)
library(igraph)

# degrees & components from your final graph g
deg <- degree(g)
comp <- components(g)

network_summary <- tibble(
  n_artists      = vcount(g),
  n_edges        = ecount(g),
  density         = edge_density(g),
  n_isolates     = sum(deg == 0),
  pct_isolates   = sum(deg == 0) / vcount(g),
  n_components   = comp$no,
  largest_component = max(comp$csize),
  median_degree  = median(deg),
  mean_degree    = mean(deg),
  max_degree     = max(deg)
)

network_summary

```

```

# A tibble: 1 x 10
  n_artists n_edges density n_isolates pct_isolates n_components
  <dbl>     <dbl>    <dbl>      <int>       <dbl>        <dbl>
1       282      786  0.0198      141        0.5        148
# i 4 more variables: largest_component <dbl>, median_degree <dbl>,
#   mean_degree <dbl>, max_degree <dbl>

```

```

vertex_df <- tibble(
  name      = V(g)$name,
  main_genre = V(g)$main_genre,
  user      = V(g)$user,
  shape     = V(g)$shape,
  deg       = deg
) %>%
  mutate(
    user_category = case_when(
      is.na(user) | user == "" ~ "None",
      user == "I"           ~ "I only",
      user == "C"           ~ "C only",
      user == "M"           ~ "M only",
      TRUE                 ~ "Combination (I/C/M)"
    )
  )
)

head(vertex_df)

```

	# A tibble: 6 x 6	name	main_genre	user	shape	deg	user_category
		<chr>	<chr>	<chr>	<chr>	<dbl>	<chr>
1	Other / Indie / Experimental	C	circ~	O	C only		
2	21 Savage	Hip-Hop / Rap	M	rect~	39	M only	
3	2hollis	Electronic / Dance	I	squa~	0	I only	
4	42 Dugg	Hip-Hop / Rap	M	rect~	16	M only	
5	6ix9ine	Hip-Hop / Rap	M	rect~	8	M only	
6	A Beacon School	Pop	C	circ~	0	C only	

```

genre_summary <- vertex_df %>%
  group_by(main_genre) %>%
  summarise(
    n_artists    = n(),
    n_isolates   = sum(deg == 0),
    pct_isolates = n_isolates / n_artists,
    mean_degree  = mean(deg),
    median_degree = median(deg),
    max_degree   = max(deg),
    .groups = "drop"
  ) %>%
  arrange(desc(n_artists))

```

```
genre_summary
```

```
# A tibble: 10 x 7
  main_genre      n_artists n_isolates pct_isolates mean_degree median_degree
  <chr>            <int>       <int>        <dbl>        <dbl>        <dbl>
1 Hip-Hop / Rap     85          5      0.0588      15.2        13
2 Other / Indie / ~    58         39      0.672       1.07        0
3 Pop                  50         39      0.78        1.2        0
4 Rock / Metal / P~    37         33      0.892       0.108       0
5 Electronic / Dan~   23         18      0.783        1        0
6 Classical / Scor~   12          0       0        5.67        6
7 R&B / Soul / Funk    8          0       0        5.38       4.5
8 Country / Folk / ~   5          5       1        0        0
9 Jazz / Blues / S~    2          2       1        0        0
10 <NA>                 2          0       0        10        10
# i 1 more variable: max_degree <dbl>
```

```
edge_df <- as_data_frame(g, what = "edges") %>%
  rename(artist1 = from, artist2 = to) %>%
  mutate(
    genre1 = V(g)$main_genre[match(artist1, V(g)$name)],
    genre2 = V(g)$main_genre[match(artist2, V(g)$name)]
  )

# same vs cross-genre edges
same_cross_summary <- edge_df %>%
  mutate(same_genre = !is.na(genre1) & genre1 == genre2) %>%
  summarise(
    total_edges      = n(),
    same_genre_edges = sum(same_genre),
    cross_genre_edges = total_edges - same_genre_edges,
    pct_same        = same_genre_edges / total_edges,
    pct_cross        = cross_genre_edges / total_edges
  )

same_cross_summary
```

```
total_edges same_genre_edges cross_genre_edges pct_same pct_cross
1           786                NA             NA        NA        NA
```

```
# matrix of edges between genre pairs (for a heatmap/table)
genre_pair_counts <- edge_df %>%
  mutate(
    g_low = pmin(genre1, genre2),
    g_high = pmax(genre1, genre2)
  ) %>%
  count(g_low, g_high, name = "n_edges") %>%
  arrange(desc(n_edges))

genre_pair_counts
```

	g_low	g_high	n_edges
1	Hip-Hop / Rap	Hip-Hop / Rap	588
2	Hip-Hop / Rap	Pop	39
3	Hip-Hop / Rap	R&B / Soul / Funk	30
4	Classical / Score / Soundtrack	Classical / Score / Soundtrack	29
5	Hip-Hop / Rap	Other / Indie / Experimental	22
6	<NA>	<NA>	20
7	Electronic / Dance	Hip-Hop / Rap	11
8	Classical / Score / Soundtrack	Other / Indie / Experimental	10
9	Electronic / Dance	Other / Indie / Experimental	8
10	Other / Indie / Experimental	Pop	7
11	Other / Indie / Experimental	Other / Indie / Experimental	5
12	R&B / Soul / Funk	R&B / Soul / Funk	5
13	Electronic / Dance	Pop	4
14	Pop	Pop	4
15	Other / Indie / Experimental	Rock / Metal / Punk	2
16	Other / Indie / Experimental	R&B / Soul / Funk	1
17	Rock / Metal / Punk	Rock / Metal / Punk	1

```
# Degree centrality
top_degree <- vertex_df %>%
  arrange(desc(deg)) %>%
  slice_head(n = 15)

top_degree
```

```
# A tibble: 15 x 6
  name      main_genre   user   shape      deg user_category
  <chr>     <chr>       <chr> <chr>     <dbl> <chr>
  1 Future   Hip-Hop / Rap M   rectangle   43 M only
```

2	Young Thug	Hip-Hop / Rap M	rectangle	41 M only
3	21 Savage	Hip-Hop / Rap M	rectangle	39 M only
4	Lil Baby	Hip-Hop / Rap M	rectangle	39 M only
5	Gunna	Hip-Hop / Rap M	rectangle	38 M only
6	Lil Durk	Hip-Hop / Rap M	rectangle	38 M only
7	Gucci Mane	Hip-Hop / Rap M	rectangle	37 M only
8	Offset	Hip-Hop / Rap M	rectangle	37 M only
9	A\$AP Rocky	Hip-Hop / Rap M	rectangle	33 M only
10	Travis Scott	Hip-Hop / Rap C,M	pie	33 Combination (I/C/M)
11	Lil Uzi Vert	Hip-Hop / Rap M	rectangle	32 M only
12	Drake	Hip-Hop / Rap M	rectangle	30 M only
13	Quavo	Hip-Hop / Rap M	rectangle	30 M only
14	Kodak Black	Hip-Hop / Rap M	rectangle	29 M only
15	Lil Yachty	Hip-Hop / Rap I,M	pie	29 Combination (I/C/M)

```
# Betweenness centrality
btw <- betweenness(g, directed = FALSE)
top_betweenness <- tibble(
  name      = V(g)$name,
  main_genre = V(g)$main_genre,
  betweenness = btw
) %>%
  arrange(desc(betweenness)) %>%
  slice_head(n = 15)

top_betweenness
```

	# A tibble: 15 x 3		
	name	main_genre	betweenness
	<chr>	<chr>	<dbl>
1	A\$AP Rocky	Hip-Hop / Rap	1094.
2	Gucci Mane	Hip-Hop / Rap	834.
3	Yung Lean	Hip-Hop / Rap	755
4	Travis Scott	Hip-Hop / Rap	545.
5	Dean Blunt	Pop	445
6	21 Savage	Hip-Hop / Rap	408.
7	Lil Yachty	Hip-Hop / Rap	346.
8	Future	Hip-Hop / Rap	297.
9	Lil Baby	Hip-Hop / Rap	254.
10	Drake	Hip-Hop / Rap	252.
11	Kanye West	Hip-Hop / Rap	246.
12	Offset	Hip-Hop / Rap	238.

13	Tiesto	Other / Indie / Experimental	227.
14	DJ Candlestick	Hip-Hop / Rap	212.
15	Brent Faiyaz	R&B / Soul / Funk	207.

```
# build artist-neighbor-genre table (both directions of each edge)
neighbors_by_genre <- bind_rows(
  edge_df %>% transmute(artist = artist1, neighbor_genre = genre2),
  edge_df %>% transmute(artist = artist2, neighbor_genre = genre1)
)

bridge_artists <- neighbors_by_genre %>%
  filter(!is.na(neighbor_genre)) %>%
  group_by(artist) %>%
  summarise(
    n_neighbors      = n(),
    n_neighbor_genres = n_distinct(neighbor_genre),
    .groups = "drop"
  ) %>%
  left_join(vertex_df, by = c("artist" = "name")) %>%
  arrange(desc(n_neighbor_genres), desc(n_neighbors)) %>%
  slice_head(n = 15)

bridge_artists
```

	artist	n_neighbors	n_neighbor_genres	main_genre	user	shape	deg
	<chr>	<int>	<int>	<chr>	<chr>	<chr>	<dbl>
1	Future	43	5	Hip-Hop / Rap	M	rect~	43
2	21 Savage	37	5	Hip-Hop / Rap	M	rect~	39
3	A\$AP Rocky	31	5	Hip-Hop / Rap	M	rect~	33
4	Gucci Mane	37	4	Hip-Hop / Rap	M	rect~	37
5	Travis Scott	33	4	Hip-Hop / Rap	C,M	pie	33
6	Kanye West	20	4	Hip-Hop / Rap	C,I,M	pie	20
7	The Creator	10	4	<NA>	<NA>	circ~	10
8	Tyler	10	4	<NA>	<NA>	circ~	10
9	Tiesto	9	4	Other / Indie	~ M	rect~	9
10	Bladee	6	4	Hip-Hop / Rap	I	squa~	6
11	Joel Corry	5	4	Pop	M	rect~	5
12	Ariana Grande	4	4	Pop	M	rect~	4
13	Young Thug	41	3	Hip-Hop / Rap	M	rect~	41
14	Lil Durk	38	3	Hip-Hop / Rap	M	rect~	38
15	Drake	30	3	Hip-Hop / Rap	M	rect~	30

```

# i 1 more variable: user_category <chr>

user_summary <- vertex_df %>%
  count(user_category, name = "n_artists") %>%
  mutate(pct = n_artists / sum(n_artists))

user_summary

# A tibble: 5 x 3
  user_category      n_artists      pct
  <chr>                <int>    <dbl>
1 C only                  87  0.309
2 Combination (I/C/M)     17  0.0603
3 I only                  84  0.298
4 M only                  92  0.326
5 None                     2  0.00709

user_by_genre <- vertex_df %>%
  group_by(main_genre, user_category) %>%
  summarise(n = n(), .groups = "drop") %>%
  group_by(main_genre) %>%
  mutate(pct = n / sum(n)) %>%
  arrange(main_genre, desc(pct))

user_by_genre

# A tibble: 27 x 4
# Groups:   main_genre [10]
  main_genre           user_category      n      pct
  <chr>                <chr>        <int>    <dbl>
1 Classical / Score / Soundtrack I only          12  1
2 Country / Folk / Songwriter   I only          3  0.6
3 Country / Folk / Songwriter   C only          1  0.2
4 Country / Folk / Songwriter   M only          1  0.2
5 Electronic / Dance           C only         10  0.435
6 Electronic / Dance           I only          9  0.391
7 Electronic / Dance           M only          3  0.130
8 Electronic / Dance           Combination (I/C/M) 1  0.0435
9 Hip-Hop / Rap               M only         69  0.812
10 Hip-Hop / Rap              I only          7  0.0824
# i 17 more rows

```

```
isolates_by_genre <- vertex_df %>%
  filter(deg == 0) %>%
  count(main_genre, name = "n_isolates") %>%
  arrange(desc(n_isolates))
```

```
isolates_by_genre
```

```
# A tibble: 7 x 2
  main_genre          n_isolates
  <chr>                  <int>
1 Other / Indie / Experimental     39
2 Pop                          39
3 Rock / Metal / Punk            33
4 Electronic / Dance             18
5 Country / Folk / Songwriter      5
6 Hip-Hop / Rap                  5
7 Jazz / Blues / Swing              2
```