Take-home\_Ex02

Lucas Luo

2025-05-21

## 📁 1. Load Knowledge Graph Data

The dataset provided is a directed multigraph in JSON format. It contains information about people in the music industry, songs, albums, record labels, and the relationships between them such as performances, collaborations, and influences.

# Read graph data  
graph\_data <- fromJSON("data/MC1\_graph.json")  
  
# Extract clean nodes and edges  
nodes\_tbl <- graph\_data$nodes %>% distinct(id, .keep\_all = TRUE)  
edges\_tbl <- graph\_data$links  
  
# Recreate id-to-row-index mapping  
id\_map <- tibble(id = nodes\_tbl$id, index = seq\_len(nrow(nodes\_tbl)))  
  
# Rebuild edges\_tbl to include row indices as from / to  
edges\_tbl <- graph\_data$links %>%  
 left\_join(id\_map, by = c("source" = "id")) %>%  
 rename(from = index) %>%  
 left\_join(id\_map, by = c("target" = "id")) %>%  
 rename(to = index) %>%  
 filter(!is.na(from), !is.na(to))  
  
# Build base tidygraph  
g <- tbl\_graph(nodes = nodes\_tbl, edges = edges\_tbl, directed = graph\_data$directed)

## 🔍 1.1 Exploratory Data Analysis (EDA): Graph Overview

# Total number of nodes and edges  
total\_nodes <- nrow(nodes\_tbl)  
total\_edges <- nrow(edges\_tbl)  
  
# Unique node types  
node\_types <- nodes\_tbl %>% count(`Node Type`, sort = TRUE)  
  
# Unique edge types  
edge\_types <- edges\_tbl %>% count(`Edge Type`, sort = TRUE)  
  
list(  
 Total\_Nodes = total\_nodes,  
 Total\_Edges = total\_edges,  
 Node\_Types = node\_types,  
 Edge\_Types = edge\_types  
)

$Total\_Nodes  
[1] 17412  
  
$Total\_Edges  
[1] 37857  
  
$Node\_Types  
 Node Type n  
1 Person 11361  
2 Song 3615  
3 RecordLabel 1217  
4 Album 996  
5 MusicalGroup 223  
  
$Edge\_Types  
 Edge Type n  
1 PerformerOf 13587  
2 RecordedBy 3798  
3 ComposerOf 3290  
4 ProducerOf 3209  
5 DistributedBy 3013  
6 LyricistOf 2985  
7 InStyleOf 2289  
8 InterpolatesFrom 1574  
9 LyricalReferenceTo 1496  
10 CoverOf 1429  
11 DirectlySamples 619  
12 MemberOf 568

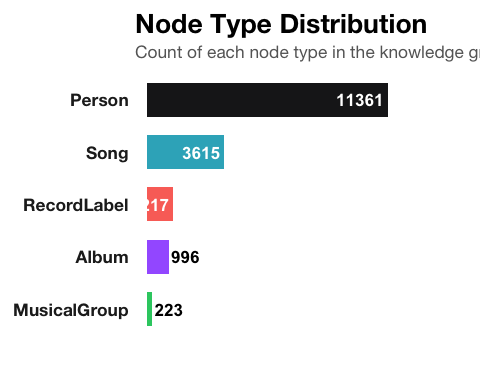
## 🧩 Node Types Description

Based on the data dictionary and actual values in the JSON file, we can classify the following node types:

| Node Type | Description |
| --- | --- |
| Person | Any individual contributor, including singers, lyricists, composers, etc. |
| MusicalGroup | Groups formed by multiple individuals, e.g., bands or choirs |
| Song | Individual song tracks with attributes like release date, genre, etc. |
| Album | Grouped collections of songs, with their own release and popularity markers |
| RecordLabel | Organizations involved in production, recording, or distribution |

We can also check their frequency in the dataset:

# Custom palette inspired by genre styles  
genre\_palette <- c(  
 "#A566FF", # Person (lavender)  
 "#2ECC71", # Song (green)  
 "#1C1C1E", # RecordLabel (black)  
 "#FA7268", # Album (salmon)  
 "#34B1C4" # MusicalGroup (aqua)  
)  
  
node\_types <- node\_types %>%   
 mutate(text\_color = ifelse(n > 1000, "white", "black"),  
 text\_hjust = ifelse(n > 1000, 1.1, -0.1))  
  
ggplot(node\_types, aes(x = reorder(`Node Type`, n), y = n, fill = `Node Type`)) +  
 geom\_col(width = 0.65, show.legend = FALSE) +  
 geom\_text(aes(label = n, hjust = text\_hjust, color = text\_color),   
 size = 4.5, fontface = "bold") +  
 coord\_flip(clip = "off") +  
 scale\_fill\_manual(values = genre\_palette) +  
 scale\_color\_identity() +  
 labs(  
 title = "<b style='font-size:20pt;'>Node Type Distribution</b>",  
 subtitle = "<span style='color:#666666;'>Count of each node type in the knowledge graph</span>",  
 x = NULL,  
 y = NULL  
 ) +  
 theme\_minimal(base\_family = "Helvetica Neue") +  
 theme(  
 panel.grid = element\_blank(),  
 axis.text.y = element\_text(face = "bold", size = 13, color = "#222222"),  
 axis.text.x = element\_blank(),  
 axis.ticks = element\_blank(),  
 plot.title = element\_markdown(size = 20, face = "bold", hjust = 0),  
 plot.subtitle = element\_markdown(size = 13, hjust = 0),  
 plot.margin = margin(10, 60, 30, 10)  
 )



## 🔗 Edge Types Reference Table

Below is the detailed explanation of all possible **Edge Types** in the knowledge graph, which describe different types of relationships between entities (nodes).

### 🎵 Creative and Collaboration Relationships

| Edge Type | Meaning |
| --- | --- |
| PerformerOf | A (person/group) performed B (song/album) |
| ComposerOf | A is the composer of B |
| LyricistOf | A is the lyricist of B |
| ProducerOf | A is the producer of B |

### 🏢 Company Relationships

| Edge Type | Meaning |
| --- | --- |
| RecordedBy | A (work) was recorded by B (record label) |
| DistributedBy | A (work) was distributed by B (record label) |

### 🎨 Style and Influence Relationships

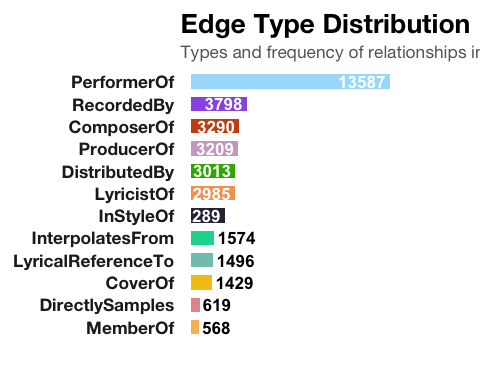
| Edge Type | Meaning |
| --- | --- |
| InStyleOf | A mimics the style of B |
| CoverOf | A is a cover version of B |
| InterpolatesFrom | A interpolates melody from B |
| LyricalReferenceTo | A makes a lyrical reference to B |
| DirectlySamples | A directly samples audio from B |

### 👥 Membership Relationships

| Edge Type | Meaning |
| --- | --- |
| MemberOf | A is a member of B (a music group or collective) |

Their distribution is plotted below:

edge\_palette <- c("#CA5310", "#F5C518", "#E5989B", "#38B000", "#2D3142", "#06D6A0", "#83C5BE",  
 "#F4A261", "#F6BD60", "#A9DEF9", "#CBAACB", "#9B5DE5")  
  
edge\_types <- edge\_types %>%   
 mutate(text\_color = ifelse(n > 2000, "white", "black"),  
 text\_hjust = ifelse(n > 2000, 1.1, -0.1))  
  
ggplot(edge\_types, aes(x = reorder(`Edge Type`, n), y = n, fill = `Edge Type`)) +  
 geom\_col(width = 0.65, show.legend = FALSE) +  
 geom\_text(aes(label = n, hjust = text\_hjust, color = text\_color),   
 size = 4.5, fontface = "bold") +  
 coord\_flip(clip = "off") +  
 scale\_fill\_manual(values = edge\_palette) +  
 scale\_color\_identity() +  
 labs(  
 title = "<b style='font-size:20pt;'>Edge Type Distribution</b>",  
 subtitle = "<span style='color:#666666;'>Types and frequency of relationships in the knowledge graph</span>",  
 x = NULL,  
 y = NULL  
 ) +  
 theme\_minimal(base\_family = "Helvetica Neue") +  
 theme(  
 panel.grid = element\_blank(),  
 axis.text.y = element\_text(face = "bold", size = 13, color = "#222222"),  
 axis.text.x = element\_blank(),  
 axis.ticks = element\_blank(),  
 plot.title = element\_markdown(size = 20, face = "bold", hjust = 0),  
 plot.subtitle = element\_markdown(size = 13, hjust = 0),  
 plot.margin = margin(10, 60, 30, 10)  
 )



These insights provide a foundation to interpret future patterns in artist collaboration, influence, and success prediction.

## 🌍 2. Sailor Shift’s Ego Network Overview (Full Connected Subgraph)

### 📌 Objective

This section extracts and visualizes the **full connected subgraph** centered on *Sailor Shift*, showcasing all directly and indirectly connected entities in the knowledge graph.

### 🌭️ Step 1: Extract All Connected Nodes

# Get Sailor Shift's node id  
sailor\_id <- nodes\_tbl %>%   
 filter(name == "Sailor Shift") %>%   
 pull(id)  
  
# Convert tidygraph to igraph for component extraction  
ig <- as.igraph(g)  
  
# Identify the connected component that includes Sailor Shift  
component\_id <- igraph::components(ig)$membership[which(V(ig)$id == sailor\_id)]  
connected\_ids <- V(ig)[igraph::components(ig)$membership == component\_id]$name

### 🌍 Step 2: Visualize Static Subgraph with Interactive 3-Hop Network

# Convert to igraph and compute 3-hop neighborhood  
ig <- as.igraph(g)  
sailor\_vid <- which(V(ig)$id == sailor\_id)  
neighbor\_ids <- ego(ig, order = 2, nodes = sailor\_vid, mode = "all")[[1]]  
  
sub3\_ids <- V(ig)[neighbor\_ids]$id  
sub3\_nodes\_tbl <- nodes\_tbl %>% filter(id %in% sub3\_ids)  
sub3\_edges\_tbl <- edges\_tbl %>% filter(source %in% sub3\_ids & target %in% sub3\_ids)  
  
# Prepare visNetwork data  
# Modify vis\_nodes to make "Sailor Shift" larger and bold  
vis\_nodes <- sub3\_nodes\_tbl %>%  
 mutate(size = ifelse(name == "Sailor Shift", 50, 20)) %>% # 🔍 Control node size  
 transmute(  
 id = id,  
 label = name,  
 group = `Node Type`,  
 title = paste0("<b>", name, "</b><br>Type: ", `Node Type`),  
 value = size # visNetwork uses 'value' to control node size  
 )  
  
# vis\_edges remains unchanged  
vis\_edges <- sub3\_edges\_tbl %>%  
 transmute(from = source, to = target, label = `Edge Type`, arrows = "to")  
  
# Render Interactive Graph with highlighted Sailor Shift  
visNetwork(vis\_nodes, vis\_edges, width = "100%", height = "700px") %>%  
 visEdges(smooth = TRUE) %>%  
 visOptions(highlightNearest = list(enabled = TRUE, degree = 2), nodesIdSelection = TRUE) %>%  
 visLayout(randomSeed = 2025) %>%  
 visLegend(useGroups = TRUE, main = "Node Type") %>%  
 visPhysics(stabilization = TRUE)

This full subgraph gives a comprehensive view of Sailor Shift’s artistic ecosystem, while the interactive 3-hop version allows users to explore relationships with greater clarity and responsiveness.

🔍 Insight: Sailor Shift’s Influence Ecosystem The Ego Network of Sailor Shift reveals a highly connected artistic ecosystem spanning songs, albums, musical groups, and record labels. Her position at the center of a dense subgraph highlights her role as a key figure in the Oceanus Folk community. The variety of edges (e.g., InterpolatesFrom, PerformerOf, MemberOf) suggests not only her creative versatility but also her multi-layered influence network—including direct artistic collaborations and indirect stylistic references.

Notably:

She interacts with all five node types (Person, Song, Album, MusicalGroup, RecordLabel),

Her connections include both creative influence (via stylistic or lyrical references) and professional affiliations (production, distribution, membership).

This network serves as a powerful visual entry point for investigating how and by whom Sailor Shift has been influenced, who she in turn has influenced, and what her broader role is within the Oceanus Folk genre.

🚀 Moving Forward: Task-based Subgraph Exploration With this foundation, we will now delve deeper into three core analytical tasks, each focused on a unique aspect of Sailor Shift’s artistic identity and influence. Each task will be accompanied by an interactive or static visualization powered by the knowledge graph.

## ✅ Task 1 – Who Has Influenced Sailor Shift?

### 🎯 Objective:

Use the knowledge graph to identify **who has influenced Sailor Shift over time**, by analyzing stylistic, lyrical, or sampling relationships represented in the graph. These insights will reveal patterns in her creative development.

### 🔍 Why This Approach?

Sailor Shift’s musical evolution is shaped by direct and indirect artistic influences. In a knowledge graph, influence is captured by specific edge types:

### 🎨 Style and Influence Relationships

| Edge Type | Meaning |
| --- | --- |
| InStyleOf | A mimics the style of B |
| CoverOf | A is a cover version of B |
| InterpolatesFrom | A interpolates melody from B |
| LyricalReferenceTo | A makes a lyrical reference to B |
| DirectlySamples | A directly samples audio from B |

These edge types describe different ways an artist or work may influence Sailor Shift. By filtering the graph for these edge types where **Sailor Shift is the target**, we construct a reliable picture of who influenced her and how.

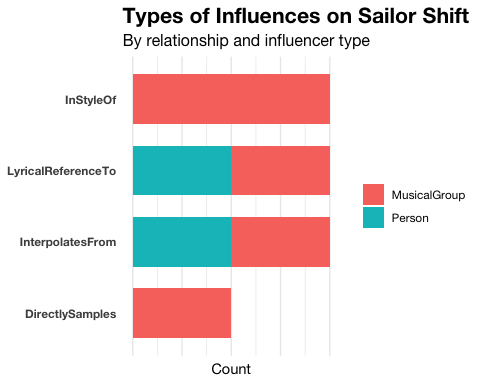
By filtering the graph for these edge types targeting Sailor Shift, we can construct a reliable view of her artistic lineage.

### 📌 Step 1: Extract Influencer Edges

influence\_types <- c("InStyleOf", "CoverOf", "InterpolatesFrom", "LyricalReferenceTo", "DirectlySamples")  
  
# Get edges that influence her  
influencer\_edges <- graph\_data$links %>%  
 filter(`Edge Type` %in% influence\_types, target == sailor\_id) %>%  
 left\_join(nodes\_tbl, by = c("source" = "id")) %>%  
 rename(influencer\_name = name, influencer\_type = `Node Type`) %>%  
 select(source, target, `Edge Type`, influencer\_name, influencer\_type)

### 📌 Step 2: Visualize Influencer Type Distribution

influencer\_edges %>%  
 count(`Edge Type`, influencer\_type) %>%  
 ggplot(aes(x = reorder(`Edge Type`, n), y = n, fill = influencer\_type)) +  
 geom\_col(width = 0.7) +  
 coord\_flip() +  
 labs(  
 title = "Types of Influences on Sailor Shift",  
 subtitle = "By relationship and influencer type",  
 x = NULL,  
 y = "Count"  
 ) +  
 theme\_minimal(base\_family = "Helvetica Neue") +  
 theme(  
 plot.title = element\_text(face = "bold", size = 16),  
 plot.subtitle = element\_text(size = 12),  
 axis.text.y = element\_text(face = "bold"),  
 panel.grid.major.y = element\_blank(),  
 axis.text.x = element\_blank(),  
 legend.title = element\_blank()  
 )



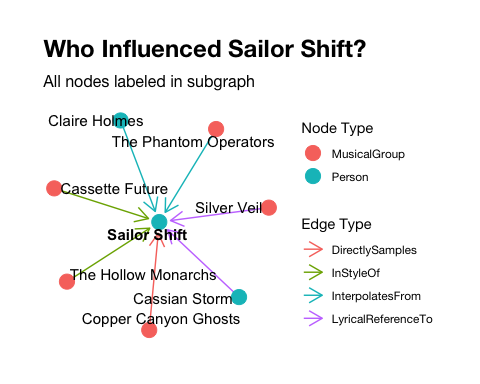
🔍 Insight: Who Influenced Sailor Shift?

The bar chart reveals the distribution of influence types on Sailor Shift, broken down by the influencer’s node type (MusicalGroup or Person). Key takeaways:

* **MusicalGroups are dominant** in all influence categories:
  + They account for **all** InStyleOf influences, suggesting that Sailor Shift’s stylistic inspirations largely stem from band-level aesthetics.
  + For DirectlySamples, MusicalGroup nodes are again the exclusive contributors, highlighting a tendency to sample from established group-based soundscapes.
* **Persons contribute notably to lyrical and melodic references**:
  + Half of the LyricalReferenceTo and InterpolatesFrom edges are from Person nodes. This suggests a more personal or individual-driven source of lyrical and melodic inspiration, possibly tied to solo artist storytelling.

### 📌 Step 3: Visualize Knowledge Graph of Influencers

# Extract relevant nodes  
influencer\_ids <- influencer\_edges$source  
  
influencer\_nodes <- nodes\_tbl %>%  
 filter(id %in% influencer\_ids | id == sailor\_id) %>%  
 distinct(id, .keep\_all = TRUE)  
  
influencer\_links <- graph\_data$links %>%  
 filter(source %in% influencer\_ids, target == sailor\_id)  
  
# Map to row indices for subgraph  
id\_map <- tibble(id = influencer\_nodes$id, index = seq\_len(nrow(influencer\_nodes)))  
  
influencer\_links\_mapped <- influencer\_links %>%  
 left\_join(id\_map, by = c("source" = "id")) %>%  
 rename(from = index) %>%  
 left\_join(id\_map, by = c("target" = "id")) %>%  
 rename(to = index) %>%  
 filter(!is.na(from), !is.na(to))  
  
# Create subgraph  
influencer\_graph <- tbl\_graph(nodes = influencer\_nodes, edges = influencer\_links\_mapped, directed = TRUE)  
  
# Draw the graph  
set.seed(2025)  
ggraph(influencer\_graph, layout = "kk") +  
 geom\_edge\_link(  
 aes(color = `Edge Type`),  
 arrow = arrow(length = unit(4, 'mm')),  
 end\_cap = circle(3, 'mm')  
 ) +  
 geom\_node\_point(aes(color = `Node Type`), size = 5) +  
 geom\_node\_text(  
 aes(label = name), # ✅ Show all node names  
 repel = TRUE,  
 size = 4,  
 fontface = ifelse(influencer\_nodes$name == "Sailor Shift", "bold", "plain")  
 ) +  
 labs(  
 title = "Who Influenced Sailor Shift?",  
 subtitle = "All nodes labeled in subgraph"  
 ) +  
 theme\_graph(base\_family = "Helvetica Neue")



### 🔍 Step 3 Insight: Direct Artistic Influences on Sailor Shift

The static knowledge graph in Step 3 provides a clear snapshot of **who directly influenced Sailor Shift**, based on well-defined artistic relationships such as DirectlySamples, InterpolatesFrom, InStyleOf, and LyricalReferenceTo.

#### ✨ Key Observations:

* **Diverse Influence Channels**: Sailor Shift’s inspirations are not confined to a single creative channel. She samples music (DirectlySamples), borrows stylistic cues (InStyleOf), lifts lyrical fragments (LyricalReferenceTo), and interpolates melodies (InterpolatesFrom).
* **Balanced Influencer Types**: Influences come from both **individual artists** and **musical groups**, indicating that her artistry draws from both personal voices and collective aesthetics.
* **Dense Inbound Edges**: The graph shows multiple inbound links converging on Sailor Shift, underscoring her role as a stylistic inheritor of a tightly connected creative network.

#### 🧠 Interpretation:

This direct influence map builds a foundation for understanding her creative identity. However, it only reflects surface-level lineage—i.e., **those explicitly linked to her** in the graph.

#### ⏭️ Why We Proceed to Step 4

While Step 3 gives us a high-fidelity view of Sailor Shift’s direct inspirations, it leaves several questions unanswered:

* **Who are the individuals behind the musical groups?**
* **Are there unseen intermediaries—like songs or albums—that serve as vehicles of influence?**
* **What multi-hop chains of inspiration exist beyond what is immediately visible?**

To answer these, we transition to **Step 4**, where we:

* Traverse **multi-layer influence chains** (not limited to 1-hop)
* Expand **MusicalGroup** nodes into their individual **Person** members (via MemberOf)
* Visualize **deep indirect relationships** that build Sailor Shift’s identity over time

### 📌 Step 4: Show All Influence Types in Indirect Graph

# Influence edge types  
influence\_types <- c("InStyleOf", "CoverOf", "InterpolatesFrom", "LyricalReferenceTo", "DirectlySamples")  
  
# Expand full influence paths recursively (e.g. up to 5 hops)  
max\_depth <- 5  
influence\_paths <- tibble()  
current\_targets <- sailor\_id  
  
for (i in 1:max\_depth) {  
 next\_edges <- edges\_tbl %>% filter(`Edge Type` %in% influence\_types, target %in% current\_targets)  
 if (nrow(next\_edges) == 0) break  
 next\_edges <- next\_edges %>% mutate(depth = paste0(i, "-hop"))  
 influence\_paths <- bind\_rows(influence\_paths, next\_edges)  
 current\_targets <- unique(next\_edges$source)  
}  
  
# Identify Musical Groups and expand their members  
musical\_group\_ids <- nodes\_tbl %>% filter(`Node Type` == "MusicalGroup") %>% pull(id)  
member\_edges <- edges\_tbl %>% filter(`Edge Type` == "MemberOf", target %in% musical\_group\_ids)  
  
# If any musical group is in influence path, bring in members  
group\_ids\_in\_path <- influence\_paths %>% filter(source %in% musical\_group\_ids) %>% pull(source)  
extra\_members <- member\_edges %>% filter(target %in% group\_ids\_in\_path)  
  
# Combine all edges  
full\_edges <- bind\_rows(  
 influence\_paths %>% mutate(hop = depth),  
 extra\_members %>% mutate(hop = "GroupMember")  
)  
  
# Related node ids  
related\_ids <- unique(c(full\_edges$source, full\_edges$target, sailor\_id))  
vis\_nodes <- nodes\_tbl %>% filter(id %in% related\_ids) %>%  
 mutate(  
 label = name,  
 group = `Node Type`,  
 value = ifelse(name == "Sailor Shift", 50, 20),  
 title = paste0("<b>", name, "</b><br>Type: ", `Node Type`)  
 ) %>%  
 select(id, label, group, value, title)  
  
vis\_edges <- full\_edges %>%  
 mutate(  
 arrows = "to",  
 label = `Edge Type`,  
 color = case\_when(  
 `Edge Type` == "DirectlySamples" ~ "#FF6B6B",  
 `Edge Type` == "InterpolatesFrom" ~ "#4D96FF",  
 `Edge Type` == "CoverOf" ~ "#FFD93D",  
 `Edge Type` == "InStyleOf" ~ "#6BCB77",  
 `Edge Type` == "LyricalReferenceTo" ~ "#C780FA",  
 `Edge Type` == "MemberOf" ~ "#BDBDBD",  
 TRUE ~ "#999999"  
 )  
 ) %>%  
 select(from = source, to = target, arrows, label, color)  
  
visNetwork(vis\_nodes, vis\_edges, width = "100%", height = "800px") %>%  
 visEdges(smooth = TRUE) %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) %>%  
 visLayout(randomSeed = 2025) %>%  
 visLegend(useGroups = TRUE, main = "Node Type") %>%  
 visPhysics(stabilization = TRUE)

#### 🔍 Step 4 Insight: Full Influence Graph with Group Member Expansion

The interactive graph in Step 4 reveals the **deep influence structure** surrounding Sailor Shift, integrating both direct and indirect artistic relationships and **group-to-individual decompositions**.

#### 🎯 Key Observations:

* **Musical Groups as Gateways**: Many key influences (e.g., *The Phantom Operators*, *Cassette Future*) are MusicalGroups. However, their true impact on Sailor Shift is clarified only after unfolding their members. This highlights the importance of surfacing latent human contributors.
* **Individuals Behind the Sound**:
  + *Juno Ray*, *Savannah Teal*, *Zara Quinn*, and others appear as hidden but crucial figures influencing Sailor Shift indirectly through their band affiliations.
  + These artists are connected through MemberOf → MusicalGroup → InfluenceType paths, showing that influence isn’t always linear or direct.
* **Multi-Relation Influence**: Relationships such as InterpolatesFrom, InStyleOf, and LyricalReferenceTo show that Sailor Shift does not just imitate style—she blends lyrical and musical fragments from diverse sources.
* **Dense Centrality**: Sailor Shift sits at the hub of a **diverse, multi-layered network**, receiving creative input from solo artists, collective groups, and intermediary compositions (albums/songs).

#### 🧠 Interpretation:

This graph confirms that Sailor Shift is not only influenced by prominent entities but is also indirectly shaped by **dozens of individuals**, whose contributions are embedded within groups. Her artistic identity emerges from a blend of:

* Stylized group traditions
* Personal lyrical echoes
* Collaborative musical lineages

It also suggests that to understand her evolution, one must look beyond direct citations and recognize the **ecosystem of creators** behind the visible front.

### 📌 Step 5: Expand Influence Path – Including Songs/Albums (≤ 5 Hops)

# Influence edge types  
influence\_types <- c("InStyleOf", "CoverOf", "InterpolatesFrom", "LyricalReferenceTo", "DirectlySamples")  
  
# Expand backward paths from Sailor Shift  
max\_depth <- 5  
influence\_paths <- tibble()  
current\_targets <- sailor\_id  
  
for (i in 1:max\_depth) {  
 next\_edges <- edges\_tbl %>% filter(`Edge Type` %in% influence\_types, target %in% current\_targets)  
 if (nrow(next\_edges) == 0) break  
 next\_edges <- next\_edges %>% mutate(depth = paste0(i, "-hop"))  
 influence\_paths <- bind\_rows(influence\_paths, next\_edges)  
 current\_targets <- unique(next\_edges$source)  
}  
  
# Identify Musical Groups and expand their members  
musical\_group\_ids <- nodes\_tbl %>% filter(`Node Type` == "MusicalGroup") %>% pull(id)  
member\_edges <- edges\_tbl %>% filter(`Edge Type` == "MemberOf", target %in% musical\_group\_ids)  
  
# If any musical group is in influence path, bring in members  
group\_ids\_in\_path <- influence\_paths %>% filter(source %in% musical\_group\_ids) %>% pull(source)  
extra\_members <- member\_edges %>% filter(target %in% group\_ids\_in\_path)  
  
# Identify all nodes in reachable paths  
direct\_path\_ids <- unique(c(influence\_paths$source, influence\_paths$target))  
  
# Identify intermediaries (song/album) linking between artists and Sailor Shift  
intermediate\_edges <- edges\_tbl %>%  
 filter(source %in% direct\_path\_ids | target %in% direct\_path\_ids) %>%  
 filter(`Edge Type` %in% c("PerformerOf", "AppearsOn")) %>%  
 mutate(hop = "Song/Album Link")  
  
# Combine all edges  
full\_edges <- bind\_rows(  
 influence\_paths %>% mutate(hop = depth),  
 intermediate\_edges,  
 extra\_members %>% mutate(hop = "GroupMember")  
)  
  
# Related node ids  
related\_ids <- unique(c(full\_edges$source, full\_edges$target, sailor\_id))  
vis\_nodes <- nodes\_tbl %>% filter(id %in% related\_ids) %>%  
 mutate(  
 label = name,  
 group = `Node Type`,  
 value = ifelse(name == "Sailor Shift", 50, 20),  
 font = list(size = ifelse(name == "Sailor Shift", 36, 22)),  
 title = paste0("<b>", name, "</b><br>Type: ", `Node Type`)  
 ) %>%  
 select(id, label, group, value, font, title)  
  
vis\_edges <- full\_edges %>%  
 mutate(  
 arrows = "to",  
 label = `Edge Type`,  
 color = case\_when(  
 `Edge Type` == "DirectlySamples" ~ "#FF6B6B",  
 `Edge Type` == "InterpolatesFrom" ~ "#4D96FF",  
 `Edge Type` == "CoverOf" ~ "#FFD93D",  
 `Edge Type` == "InStyleOf" ~ "#6BCB77",  
 `Edge Type` == "LyricalReferenceTo" ~ "#C780FA",  
 `Edge Type` == "MemberOf" ~ "#BDBDBD",  
 `Edge Type` == "PerformerOf" ~ "#00BFC4",  
 `Edge Type` == "AppearsOn" ~ "#F8766D",  
 TRUE ~ "#999999"  
 )  
 ) %>%  
 select(from = source, to = target, arrows, label, color)  
  
visNetwork(vis\_nodes, vis\_edges, width = "100%", height = "800px") %>%  
 visEdges(smooth = TRUE) %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) %>%  
 visLayout(randomSeed = 2025) %>%  
 visLegend(useGroups = TRUE, main = "Node Type") %>%  
 visPhysics(stabilization = TRUE)

#### 🔍 Insight (Step 5 – Song/Album Influence)

The network reveals a rich multi-hop influence chain leading into Sailor Shift’s artistry. Instead of just direct artistic references, this graph uncovers how individuals or groups shaped her through intermediary songs and albums.

#### 🎯 Key Observations:

Several musical groups contributed via performed songs, which were later interpolated or sampled by Sailor Shift.

Albums served as indirect influence bridges, housing songs that were referenced, covered, or remixed.

Collaboration depth is highlighted—some artists influence Sailor Shift through layered paths (e.g., PerformerOf → Song → InterpolatesFrom → Sailor Shift).

This illustrates how artistic influence flows across works, not just individuals—showing Sailor Shift’s stylistic lineage as both deep and distributed.