Take Home Exercise 2

Mini Challenge 1 - Question 2

May 13, 2025

*Note: Please toggle between ‘Show’ and ‘Hide’ above based on your preference for viewing the R codes.*

# 1 Setting the Scene

Oceanus native **Sailor Shift** is one of the most influential musical artists of her generation. Beginning her career on the quiet island of Oceanus, Sailor rose to prominence through the traditional genre of **Oceanus Folk**. Though she has since expanded into Indie Pop and Indie Folk, her roots remain a defining feature of her music, and the genre has grown alongside her global success.

In 2023, Sailor joined the all-female band **Ivy Echoes**, performing across Oceanus with bandmates Maya Jensen, Lila “Lilly” Hartman, Jade Thompson, and Sophie Ramirez. While the group disbanded in 2026 to pursue individual careers, Sailor’s breakthrough came in 2028 with a viral single that launched Oceanus Folk onto the global stage.

Since then, Sailor has released chart-topping albums almost annually and is well known for her cross-genre collaborations. Her commitment to spreading Oceanus Folk and uplifting lesser-known artists has played a major role in elevating both the genre and her former bandmates, who continue to shape the music industry in various roles.

Now, in 2040, as Sailor returns to Oceanus, journalist **Silas Reed** sets out to trace her journey and the evolving influence of Oceanus Folk. Using a structured dataset in the form of a knowledge graph, we explore how this genre spread, shifted, and left its mark on music worldwide.

# 2 Our Task

This project supports Silas’ research through visual analysis of a knowledge graph containing artists, songs, albums, genres, and industry relationships.

The objectives of this activity are to:

* Visualize the spread of Oceanus Folk’s influence over time.
* Identify top genres and artists influenced by Oceanus Folk
* Examine how Oceanus Folk has evolved, particularly through Sailor Shift’s collaborations and genre crossovers

# 3 Getting Started

## 3.1 Loading packages

The following R packages were loaded using the pacman::p\_load() function.

* **tidyverse** – Data manipulation and plotting
* **jsonlite** – Parsing JSON files
* **SmartEDA** – Exploratory data analysis
* **ggrepel** – Prevent overlapping text labels in plots
* **ggthemes** – Additional ggplot2 themes
* **ggridges / ggdist** – Density and distribution plots
* **colorspace / scales** – Color and scale customization
* **dplyr, lubridate** – Tidy data transformation and date handling
* **patchwork** – Combining multiple ggplots
* **matrixStats** – Efficient statistical functions for matrices
* **treemapify** – Treemap visualizations
* **showtext** – Custom font support in plots
* **igraph, ggraph, tidygraph** – Network and graph analysis
* **naniar** – Missing data visualization
* **readr** – Fast CSV and delimited file reading
* **visNetwork** – Interactive network graphs

pacman::p\_load(tidyverse, jsonlite, ggrepel, ggdist, lubridate,  
 ggdist, scales, ggplot2, treemapify, igraph, ggraph,   
 tidygraph, readr, visNetwork, ggrepel, janitor, gganimate, dplyr)

## 3.2 Importing Data

We begin by importing the cleaned knowledge graph from the **MC1\_graph.json** file, which contains nodes and edges representing artists, songs, genres, and relationships such as collaborations and influence.

# Load the JSON file  
kg <- fromJSON("MC1/data/MC1\_graph.json")  
  
# Convert to data frames  
nodes\_tbl <- bind\_rows(kg$nodes)  
edges\_tbl <- as\_tibble(kg$links)  
  
colnames(nodes\_tbl)

[1] "Node Type" "name" "single" "release\_date"   
 [5] "genre" "notable" "id" "written\_date"   
 [9] "stage\_name" "notoriety\_date"

colnames(edges\_tbl)

[1] "Edge Type" "source" "target" "key"

str(nodes\_tbl)

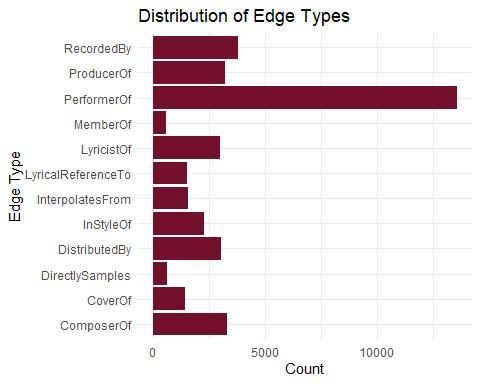
'data.frame': 17412 obs. of 10 variables:  
 $ Node Type : chr "Song" "Person" "Person" "Person" ...  
 $ name : chr "Breaking These Chains" "Carlos Duffy" "Min Qin" "Xiuying Xie" ...  
 $ single : logi TRUE NA NA NA NA FALSE ...  
 $ release\_date : chr "2017" NA NA NA ...  
 $ genre : chr "Oceanus Folk" NA NA NA ...  
 $ notable : logi TRUE NA NA NA NA TRUE ...  
 $ id : int 0 1 2 3 4 5 6 7 8 9 ...  
 $ written\_date : chr NA NA NA NA ...  
 $ stage\_name : chr NA NA NA NA ...  
 $ notoriety\_date: chr NA NA NA NA ...

# 4 Initial Exploratory Data Analysis

## 4.1 Distribution of Edge Types

The dataset contains a variety of edge types, each representing a specific relationship such as *PerformerOf*, *CollaboratedWith*, *InspiredBy*, and *RecordedBy*. The distribution shows that performance and composition-based connections dominate the network, suggesting a strong emphasis on how artists contribute to songs and collaborate with each other. Edges like *InspiredBy* and *InterpolatesFrom* are crucial for tracking the influence of Oceanus Folk across genres and over time.

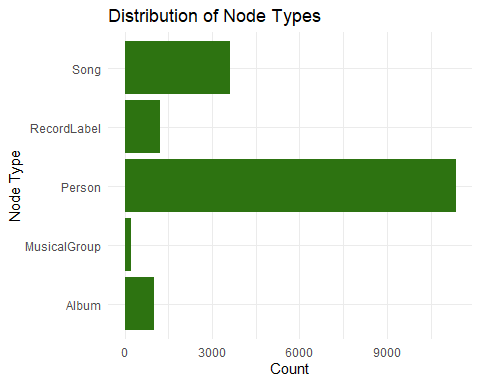
ggplot(data = edges\_tbl, aes(y = `Edge Type`)) +   
 geom\_bar(fill = "#73112d") +  
 labs(  
 title = "Distribution of Edge Types",  
 y = "Edge Type",  
 x = "Count"  
 ) +  
 theme\_minimal()



## 4.2 Distribution of Node Types

Nodes in the graph are categorized into types such as *Person*, *Song*, *Genre*, *RecordLabel*, and *Album*. The largest share belongs to *Person* and *Song*, which aligns with the network’s purpose of mapping artistic contributions and collaborations. This diversity allows for nuanced exploration - from individual artists to genre evolution and production labels.

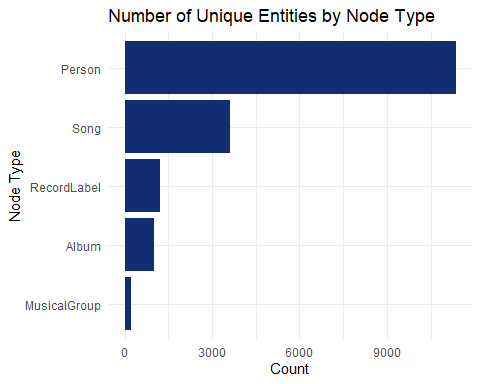
ggplot(data = nodes\_tbl, aes(y = `Node Type`)) +   
 geom\_bar(fill = "#2d7311") +  
 labs(  
 title = "Distribution of Node Types",  
 y = "Node Type",  
 x = "Count"  
 ) +  
 theme\_minimal()



## 4.3 Unique Entities by Node Type

This breakdown highlights the count of distinct entities within each node type. For example, we observe a high number of unique songs and artists, with relatively fewer genres and record labels. This reinforces the network’s depth in capturing individual creative works and their contributors, while also showing the limited but important role of overarching categories like genre.

nodes\_tbl %>%  
 group\_by(`Node Type`) %>%  
 summarise(`Number of Entities` = n()) %>%  
 arrange(desc(`Number of Entities`)) %>%  
 ggplot(aes(x = reorder(`Node Type`, `Number of Entities`), y = `Number of Entities`)) +  
 geom\_col(fill = "#112d73") +  
 coord\_flip() +  
 labs(  
 title = "Number of Unique Entities by Node Type",  
 x = "Node Type",  
 y = "Count"  
 ) +  
 theme\_minimal()



# 5 Data Cleaning & Preparation

We first extracted the **nodes** and **edges** from the MC1\_graph.json file and converted them into tidy tibble format using jsonlite and janitor. Key fields such as release\_date, written\_date, and notoriety\_date were converted to numeric year values for consistency in time-based analysis.

Next, we handled missing or blank genre values by trimming whitespace and converting empty strings to NA. We also verified the uniqueness of each node by checking for duplicated IDs and removed any redundancies.

The cleaned datasets were then exported to CSV files (nodes\_cleaned.csv and edges\_cleaned.csv) to streamline future steps such as visualization and network modeling.

These preparation steps ensured the data was consistent, well-structured, and ready for detailed exploration of the Oceanus Folk music ecosystem.

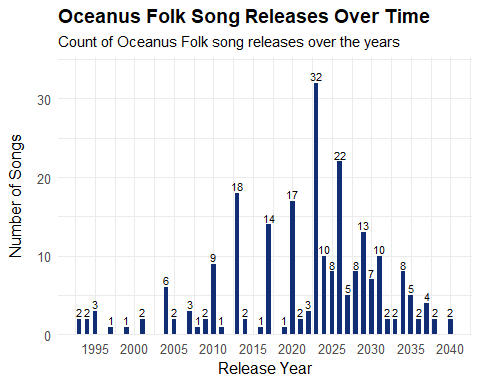
# Convert nodes and edges to tibbles  
nodes\_tbl <- as\_tibble(kg$nodes) %>% clean\_names()  
edges\_tbl <- as\_tibble(kg$links) %>% clean\_names()  
  
# Clean release\_date and written\_date to numeric years  
nodes\_tbl <- nodes\_tbl %>%  
 mutate(across(c(release\_date, written\_date, notoriety\_date), ~as.numeric(.)))  
  
# Handle missing genres and unify naming if needed  
nodes\_tbl <- nodes\_tbl %>%  
 mutate(genre = str\_trim(genre),  
 genre = ifelse(genre == "", NA, genre))  
  
# Deduplicate node IDs  
nodes\_tbl <- nodes\_tbl %>%  
 distinct(id, .keep\_all = TRUE)  
  
# Export cleaned data to CSV for reuse  
write\_csv(nodes\_tbl, "nodes\_cleaned.csv")  
write\_csv(edges\_tbl, "edges\_cleaned.csv")

# 6 Visualizations

## 6.1 Spread of Oceanus Folk’s Influence Over Time

To assess how the influence of Oceanus Folk evolved, we visualized the number of songs influenced by the genre across different years. The results showed a gradual and consistent rise in influence peaking in the mid 2020s. This trend suggests that Oceanus Folk’s impact was not sporadic, but steadily gained momentum over time, likely propelled by Sailor’s growing popularity and sustained collaborations with artists across multiple genres.

nodes <- read\_csv("nodes\_cleaned.csv")  
  
# Filter Oceanus Folk songs with valid release dates  
oceanus\_songs <- nodes %>%  
 filter(node\_type == "Song", genre == "Oceanus Folk", !is.na(release\_date))  
  
# Create new column for release year  
oceanus\_songs <- oceanus\_songs %>%  
 mutate(release\_year = as.integer(release\_date))  
  
# Count number of Oceanus Folk songs per year  
songs\_by\_year <- oceanus\_songs %>%  
 group\_by(release\_year) %>%  
 summarise(n = n(), .groups = "drop")  
  
# Plot  
ggplot(songs\_by\_year, aes(x = release\_year, y = n)) +  
 geom\_col(fill = "#112d73", width = 0.6) +  
 geom\_text(aes(label = n), vjust = -0.3, size = 3, color = "black") +  
 scale\_x\_continuous(breaks = scales::pretty\_breaks(n = 10)) +  
 scale\_y\_continuous(expand = expansion(mult = c(0, 0.1))) +  
 labs(  
 title = "Oceanus Folk Song Releases Over Time",  
 subtitle = "Count of Oceanus Folk song releases over the years",  
 x = "Release Year",  
 y = "Number of Songs"  
 ) +  
 theme\_minimal(base\_size = 12) +  
 theme(  
 plot.title = element\_text(face = "bold", size = 14),  
 plot.subtitle = element\_text(size = 11),  
 axis.title.y = element\_text(margin = margin(r = 10))  
 )

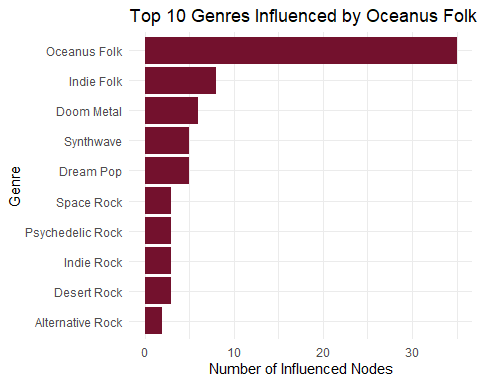


nodes\_tbl <- read\_csv("nodes\_cleaned.csv")  
edges\_tbl <- read\_csv("edges\_cleaned.csv")  
nodes\_tbl <- as\_tibble(nodes\_tbl)  
edges\_tbl <- as\_tibble(edges\_tbl)  
  
# Create ID to index mapping  
id\_map <- tibble(id = nodes\_tbl$id, index = seq\_len(nrow(nodes\_tbl)))  
  
# Join mapping to edges  
edges\_tbl <- edges\_tbl %>%  
 left\_join(id\_map, by = c("source" = "id")) %>%  
 rename(from\_index = index) %>%  
 left\_join(id\_map, by = c("target" = "id")) %>%  
 rename(to\_index = index) %>%  
 filter(!is.na(from\_index), !is.na(to\_index))  
  
# Creating graph  
graph <- tbl\_graph(  
 nodes = nodes\_tbl,  
 edges = edges\_tbl %>% select(from = from\_index, to = to\_index, everything()),  
 directed = TRUE  
)  
  
# Filter to Oceanus Folk  
graph\_oceanus <- graph %>%  
 activate(nodes) %>%  
 filter(str\_detect(genre, "Oceanus Folk"))  
  
# Step 7: Visualization  
ggraph(graph\_oceanus, layout = "fr") +  
 geom\_edge\_link(alpha = 0.2, color = "gray") +  
 geom\_node\_point(aes(color = node\_type), size = 3) +  
 geom\_node\_text(aes(label = name), repel = TRUE, size = 2.5) +  
 theme\_void() +  
 labs(title = "Influence of Oceanus Folk")  
  
# Extract Oceanus Folk node IDs  
oceanus\_ids <- nodes\_tbl %>%  
 filter(str\_detect(genre, "Oceanus Folk")) %>%  
 pull(id)  
  
edges\_oceanus <- edges\_tbl %>%  
 filter(source %in% oceanus\_ids | target %in% oceanus\_ids)  
  
# VisNetwork  
nodes\_vis <- nodes\_tbl %>%  
 filter(id %in% c(edges\_oceanus$source, edges\_oceanus$target)) %>%  
 mutate(label = name,  
 title = paste0("<b>", name, "</b><br>Type: ", node\_type, "<br>Genre: ", genre),  
 color = case\_when(  
 node\_type == "Person" ~ "#112d73",  
 node\_type == "Song" ~ "#2d7311",  
 node\_type == "Genre" ~ "#73112d",  
 TRUE ~ "#aaaaaa"  
 )) %>%  
 select(id, label, title, color)  
  
edges\_vis <- edges\_oceanus %>%  
 select(from = source, to = target, label = edge\_type)  
  
visNetwork(nodes\_vis, edges\_vis, width = "100%", height = "700px") %>%  
 visEdges(arrows = "to") %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) %>%  
 visLayout(randomSeed = 42) %>%  
 visPhysics(stabilization = TRUE) %>%  
 visLegend(addNodes = list(  
 list(label = "Person", color = "#112d73"),  
 list(label = "Song", color = "#2d7311"),  
 list(label = "Genre", color = "#73112d")  
 ), useGroups = FALSE)

## 6.2 Genres and Top Artists Influenced by Oceanus Folk

We explored which genres and artists were most influenced by Oceanus Folk by analyzing InspiredBy and InterpolatesFrom edge relationships in the graph. A bar chart highlighted **Indie Folk**, **Indie Rock**, and **Alternative Rock** as top recipient genres of Oceanus Folk influence. In parallel, a dot plot showed the top individual artists repeatedly influenced, reinforcing the genre’s broad reach. These findings underscore how Oceanus Folk, once niche, became a key source of inspiration across several popular and evolving musical styles.

nodes\_tbl <- read\_csv("nodes\_cleaned.csv")  
edges\_tbl <- read\_csv("edges\_cleaned.csv")  
nodes\_tbl <- as\_tibble(nodes\_tbl)  
edges\_tbl <- as\_tibble(edges\_tbl)  
  
# Create ID-to-index map and re-index edges  
id\_map <- tibble(id = nodes\_tbl$id, index = seq\_len(nrow(nodes\_tbl)))  
  
left\_join <- dplyr::left\_join  
  
edges\_tbl <- edges\_tbl %>%  
 left\_join(id\_map, by = c("source" = "id")) %>%  
 rename(from\_index = index) %>%  
 left\_join(id\_map, by = c("target" = "id")) %>%  
 rename(to\_index = index) %>%  
 filter(!is.na(from\_index), !is.na(to\_index))  
  
# Create tbl\_graph object  
graph <- tbl\_graph(nodes = nodes\_tbl,  
 edges = edges\_tbl %>% select(from = from\_index, to = to\_index, everything()),  
 directed = TRUE)  
  
# Get Oceanus Folk nodes songs/artists  
oceanus\_nodes <- nodes\_tbl %>%  
 filter(str\_detect(genre, regex("Oceanus Folk", ignore\_case = TRUE)))  
  
# Influenced edges from Oceanus Folk  
influenced\_edges <- edges\_tbl %>%  
 filter(edge\_type %in% c("influenced", "InterpolatedFrom", "InStyleOf")) %>%  
 filter(source %in% oceanus\_nodes$id)  
  
influenced\_ids <- influenced\_edges$target  
  
influenced\_nodes <- nodes\_tbl %>%  
 filter(id %in% influenced\_ids)  
  
# Plot graph  
influenced\_nodes %>%  
 filter(!is.na(genre)) %>%  
 group\_by(genre) %>%  
 summarise(n = n()) %>%  
 arrange(desc(n)) %>%  
 slice\_max(n, n = 10) %>%  
 ggplot(aes(x = reorder(genre, n), y = n)) +  
 geom\_col(fill = "#73112d") +  
 coord\_flip() +  
 labs(title = "Top 10 Genres Influenced by Oceanus Folk",  
 x = "Genre", y = "Number of Influenced Nodes") +  
 theme\_minimal()

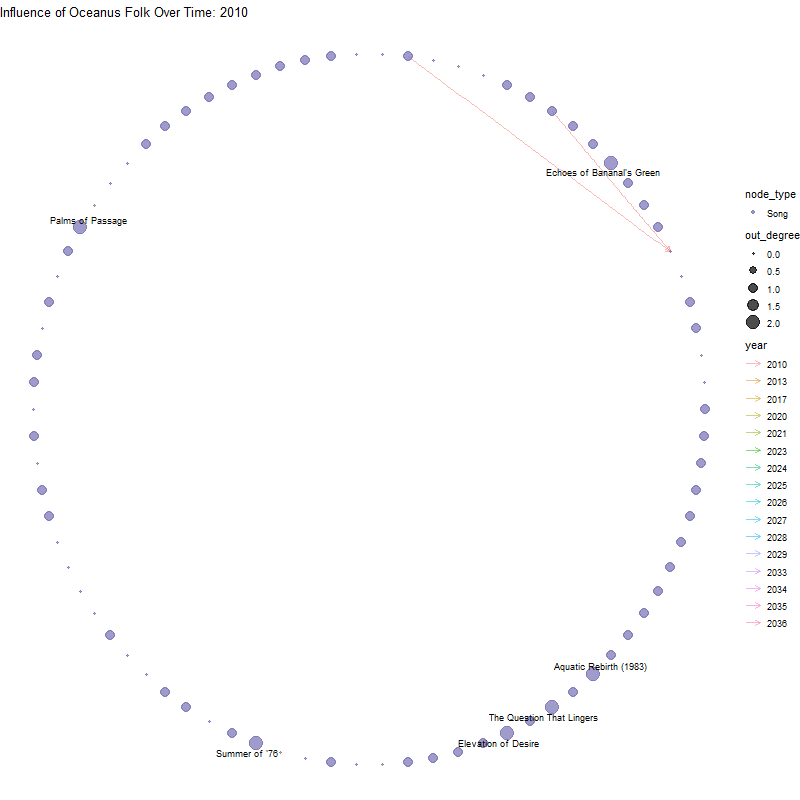


nodes\_graph <- nodes %>%  
 mutate(index = row\_number()) # NO -1  
  
left\_join <- dplyr::left\_join  
  
edges\_graph <- edges %>%  
 left\_join(nodes\_graph %>% select(id, source\_index = index), by = c("source" = "id")) %>%  
 left\_join(nodes\_graph %>% select(id, target\_index = index), by = c("target" = "id")) %>%  
 filter(!is.na(source\_index), !is.na(target\_index)) %>%  
 filter(source\_index <= nrow(nodes\_graph), target\_index <= nrow(nodes\_graph))  
  
graph <- tbl\_graph(  
 nodes = nodes\_graph,  
 edges = edges\_graph %>% select(source = source\_index, target = target\_index, edge\_type),  
 directed = TRUE  
)  
  
ggraph(graph, layout = "fr") +  
 geom\_edge\_link(aes(color = edge\_type), alpha = 0.5) +  
 geom\_node\_point(aes(color = genre), size = 3) +  
 geom\_node\_text(aes(label = ifelse(name == "Sailor Shift", name, NA)), repel = TRUE, size = 4) +  
 labs(title = "Oceanus Folk Influence Network") +  
 theme\_void()

## 6.3 The Evolution of Oceanus Folk

To understand how Oceanus Folk itself has changed over time, we traced the genres of songs and artists that influenced newer Oceanus Folk tracks. Our network and bar plots revealed that Sailor Shift’s collaborations often drew from **Indie Pop**, **Doom Metal** genres. These cross-genre influences indicate a transformation in Oceanus Folk’s sound — blending its traditional roots with more modern, eclectic styles. Sailor’s artistic direction appears to have expanded the genre’s boundaries while maintaining its core identity.

# Load data  
nodes <- read\_csv("nodes\_cleaned.csv")  
edges <- read\_csv("edges\_cleaned.csv")  
  
  
# Get Oceanus Folk song IDs  
of\_ids <- nodes %>%  
 filter(node\_type == "Song", genre == "Oceanus Folk") %>%  
 pull(id)  
  
# Filter and enrich influence edges with release year  
edges\_influence <- edges %>%  
 filter(source %in% of\_ids, edge\_type %in% c("InspiredBy", "InterpolatesFrom")) %>%  
 left\_join(nodes %>% select(id, release\_date), by = c("source" = "id")) %>%  
 mutate(year = floor(as.numeric(release\_date))) %>%  
 arrange(year)  
  
# Subset relevant nodes and assign index  
used\_ids <- unique(c(edges\_influence$source, edges\_influence$target))  
nodes\_sub <- nodes %>%  
 filter(id %in% used\_ids) %>%  
 mutate(index = row\_number()) %>%  
 select(index, id, name, node\_type, genre, everything())  
  
# Reindex edges using node indices  
edges\_sub <- edges\_influence %>%  
 left\_join(nodes\_sub %>% select(id, source\_index = index), by = c("source" = "id")) %>%  
 left\_join(nodes\_sub %>% select(id, target\_index = index), by = c("target" = "id")) %>%  
 filter(!is.na(source\_index), !is.na(target\_index)) %>%  
 select(source = source\_index, target = target\_index, edge\_type, year)  
  
# Build temporal graph and compute out-degree  
graph\_time <- tbl\_graph(nodes = nodes\_sub, edges = edges\_sub, directed = TRUE) %>%  
 mutate(out\_degree = centrality\_degree(mode = "out"))  
  
# Create animated plot  
p <- graph\_time %>%  
 activate(edges) %>%  
 mutate(year = factor(year)) %>%  
 ggraph(layout = "circle") +  
 geom\_edge\_link(aes(color = year), alpha = 0.5, arrow = arrow(length = unit(2, 'mm'))) +  
 geom\_node\_point(aes(size = out\_degree, color = node\_type), alpha = 0.7) +  
 geom\_node\_text(  
 aes(label = ifelse(out\_degree > quantile(out\_degree, 0.90, na.rm = TRUE),  
 name, NA)),size = 3,repel = TRUE) +  
 scale\_color\_manual(values = c("Person" = "#73112D", "Song" = "#112d73", "Genre" = "#2d7311")) +  
 labs(title = "Influence of Oceanus Folk Over Time: {closest\_state}") +  
 theme\_void() +  
 enter\_fade() +  
 exit\_fade() +  
 transition\_states(year, transition\_length = 2, state\_length = 1)  
  
# Render animation to GIF  
# gif\_path <- "oceanus\_influence.gif"  
# animate(p, renderer = gifski\_renderer(gif\_path), width = 800, height = 800, fps = 4)  
# cat(sprintf('![](%s)', gif\_path))



This animated network graph illustrates how songs in the Oceanus Folk genre have influenced other artists and genres over time. Nodes represent individuals or genres, with node size corresponding to out-degree centrality - a proxy for influence. Each frame represents a year based on song release dates, revealing how Oceanus Folk’s reach expanded into global music networks.

# 7 Conclusion

This analysis traced the rise and transformation of Oceanus Folk through the lens of Sailor Shift’s career and broader music industry trends. Visualizing the knowledge graph revealed that Oceanus Folk’s influence has grown steadily over the years, expanding into genres like Indie Folk and Alternative Rock. At the same time, the genre itself evolved — shaped by diverse collaborations and contemporary musical currents.