Take-home\_Ex02

Shermainn

2025-05-10

# Import Packages

In the code chunk below, p\_load() of **pacman** package is used to load the R packages into R environment.

pacman::p\_load(tidyverse, jsonlite,  
 tidygraph, ggraph,  
 SmartEDA, dplyr, visNetwork, igraph)

# Import Data

In the code chunk below, fromJSON() of **jsonlite** package is used to import *mc3.json* file into R and save the output object

MC3 <- fromJSON("data/MC3\_graph.json")  
MC3\_schema <- fromJSON("data/MC3\_schema.json")

## Inspect Knowledge Graph Structure

In the code chunk below glimpse() is used to reveal the structure of *mc3* knowledge graph.

glimpse(MC3)

## Extract Edges and Nodes

Next, as\_tibble() of **tibble** package package is used to extract the nodes and links tibble data frames from *mc3* tibble dataframe into two separate tibble dataframes called *mc3\_nodes* and *mc3\_edges* respectively.

mc3\_nodes <- as\_tibble(MC3$nodes)  
mc3\_edges <- as\_tibble(MC3$edges)

## Initial EDA

It is time for us to apply appropriate EDA methods to examine the data.

In the code chunk below, ExpCatViz() of SmartEDA package is used to reveal the frequency distribution of all categorical fields in *mc3\_nodes* tibble dataframe.

On the other hands, code chunk below uses ExpCATViz() of SmartEDA package to reveal the frequency distribution of all categorical fields in *mc3\_edges* tibble dataframe.

ExpCatViz(data=mc3\_nodes,  
 col="lightblue")

# Data Cleaning and Wrangling

### Filter Values

Code chunk below performs the following data cleaning tasks:

* convert values in id field into character data type,
* exclude records with id value are na,
* exclude records with similar id values,
* exclude thing\_collected field, and
* save the cleaned tibble dataframe into a new tibble datatable called mc3\_nodes\_cleaned.

mc3\_nodes\_cleaned <- mc3\_nodes %>%  
 mutate(id = as.character(id)) %>%  
 filter(!is.na(id)) %>%  
 distinct(id, .keep\_all = TRUE) %>%  
 select(-thing\_collected)

### Rename and remove unnecessary values

Next, the code chunk below will be used to:

* rename source and target fields to from\_id and to\_id respectively,
* convert values in from\_id and to\_id fields to character data type,
* exclude values in from\_id and to\_id which not found in the id field of mc3\_nodes\_cleaned,
* exclude records whereby from\_id and/or to\_id values are missing, and
* save the cleaned tibble dataframe and called it mc3\_edges\_cleaned.

mc3\_edges\_cleaned <- mc3\_edges %>%  
 rename(from\_id = source,   
 to\_id = target) %>%  
 mutate(across(c(from\_id, to\_id),   
 as.character)) %>%  
 filter(from\_id %in% mc3\_nodes\_cleaned$id,   
 to\_id %in% mc3\_nodes\_cleaned$id) %>%  
 filter(!is.na(from\_id), !is.na(to\_id))

### Create mapping of nodes

Next, code chunk below will be used to create mapping of character id in mc3\_nodes\_cleaned to row index.

node\_index\_lookup <- mc3\_nodes\_cleaned %>%  
 mutate(.row\_id = row\_number()) %>%  
 select(id, .row\_id)

### Join nodes to edges

Next, the code chunk below will be used to join and convert from\_id and to\_id to integer indices. At the same time we also drop rows with unmatched nodes.

mc3\_edges\_indexed <- mc3\_edges\_cleaned %>%  
 left\_join(node\_index\_lookup,   
 by = c("from\_id" = "id")) %>%  
 rename(from = .row\_id) %>%  
 left\_join(node\_index\_lookup,   
 by = c("to\_id" = "id")) %>%  
 rename(to = .row\_id) %>%  
 select(from, to, is\_inferred, type) %>%  
 filter(!is.na(from) & !is.na(to))

### Subset Nodes by Edges

Next the code chunk below is used to subset nodes to only those referenced by edges.

used\_node\_indices <- sort(  
 unique(c(mc3\_edges\_indexed$from,   
 mc3\_edges\_indexed$to)))  
  
mc3\_nodes\_final <- mc3\_nodes\_cleaned %>%  
 slice(used\_node\_indices) %>%  
 mutate(new\_index = row\_number())

### Rebuild new index

We will then use the code chunk below to rebuild lookup from old index to new index.

old\_to\_new\_index <- tibble(  
 old\_index = used\_node\_indices,  
 new\_index = seq\_along(  
 used\_node\_indices))

### Update edges to match new node table

Lastly, the code chunk below will be used to update edge indices to match new node table.

mc3\_edges\_final <- mc3\_edges\_indexed %>%  
 left\_join(old\_to\_new\_index,   
 by = c("from" = "old\_index")) %>%  
 rename(from\_new = new\_index) %>%  
 left\_join(old\_to\_new\_index,   
 by = c("to" = "old\_index")) %>%  
 rename(to\_new = new\_index) %>%  
 select(from = from\_new, to = to\_new,   
 is\_inferred, type)

## Build tidygraph Object

mc3\_graph <- tbl\_graph(  
 nodes = mc3\_nodes\_final,  
 edges = mc3\_edges\_final,  
 directed = TRUE  
)  
  
str(mc3\_graph)

## Visualize knowledge graph

Several of the **ggraph** layouts involve randomisation. In order to ensure reproducibility, it is necessary to set the seed value before plotting by using the code chunk below.

set.seed(1234)

In the code chunk below, **ggraph** functions are used to create the whole graph.

ggraph(mc3\_graph,   
 layout = "fr") +  
 geom\_edge\_link(alpha = 0.3,   
 colour = "gray") +  
 geom\_node\_point(aes(color = `type`),   
 size = 4) +  
 geom\_node\_text(aes(label = type),   
 repel = TRUE,   
 size = 2.5) +  
 theme\_void()

# Mini Case 3

## Background

Over the past decade, the community of Oceanus has faced numerous transformations and challenges evolving from its fishing-centric origins. Following major crackdowns on illegal fishing activities, suspects have shifted investments into more regulated sectors such as the ocean tourism industry, resulting in growing tensions. This increased tourism has recently attracted the likes of international pop star Sailor Shift, who announced plans to film a music video on the island.

Clepper Jessen, a former analyst at FishEye and now a seasoned journalist for the Hacklee Herald, has been keenly observing these rising tensions. Recently, he turned his attention towards the temporary closure of Nemo Reef. By listening to radio communications and utilizing his investigative tools, Clepper uncovered a complex web of expedited approvals and secretive logistics. These efforts revealed a story involving high-level Oceanus officials, Sailor Shift’s team, local influential families, and local conservationist group The Green Guardians, pointing towards a story of corruption and manipulation.

Your task is to develop new and novel visualizations and visual analytics approaches to help Clepper get to the bottom of this story.

## Tasks and Questions:

Clepper diligently recorded all intercepted radio communications over the last two weeks. With the help of his intern, they have analyzed their content to identify important events and relationships between key players. The result is a knowledge graph describing the last two weeks on Oceanus. Clepper and his intern have spent a large amount of time generating this knowledge graph, and they would now like some assistance using it to answer the following questions.

1. Clepper found that messages frequently came in at around the same time each day.
   * Develop a graph-based visual analytics approach to identify any daily temporal patterns in communications.
   * How do these patterns shift over the two weeks of observations?
   * Focus on a specific entity and use this information to determine who has influence over them.

## Extracting Graph Elements

To identify any patterns in communications between the different people or companies, only “sub\_type”, “name”, “content” and “timestamp” columns were used to do these analysis.

### 1. Make an entity list

entity\_list <- subset(mc3\_nodes\_final, type == "Entity", select = c(sub\_type, name))  
  
entity\_list

### 2. Make a Content for communications List

comms\_list <- subset(mc3\_nodes\_final, type == "Event" & label == "Communication", select = c(timestamp, content))  
  
comms\_list

### 3. Make a new list - to identify the entities involved in the messages in the communications

clean\_text <- function(x) {  
 x <- tolower(x)  
 x <- gsub("[^a-z0-9 ]", "", x) # remove punctuation  
 trimws(x)  
}  
  
entity\_names <- unique(mc3\_nodes\_final$name)  
  
# Pre-clean both  
entity\_names\_clean <- sapply(entity\_names, clean\_text)  
content\_clean <- sapply(comms\_list$content, clean\_text)  
  
# Match using the cleaned versions  
comms\_entity\_list <- mapply(function(ts, content) {  
 matches <- entity\_names[ sapply(entity\_names\_clean, function(entity) grepl(entity, content, fixed = TRUE)) ]  
 list(timestamp = ts, entities = matches)  
}, ts = comms\_list$timestamp, content = content\_clean, SIMPLIFY = FALSE)

### 4. Create new Edges data frame (Entity -> Entity)

For ggraph to plot the edges correctly, we need to assigns weights to the edges. For simplicity, let us assign the number of edges with the same source, target and type.

It will then be changed from source and target to from and to, respectively for compatibility with network functions we will use.

edges\_list <- lapply(comms\_entity\_list, function(x) {  
 ents <- x$entities  
 if(length(ents) < 2) return(NULL) # Need at least 2 to make an edge  
   
 # Create all pairwise combinations (undirected edges)  
 pairs <- t(combn(ents, 2))  
   
 data.frame(  
 from = pairs[,1],  
 to = pairs[,2],  
 timestamp = x$timestamp,  
 stringsAsFactors = FALSE  
 )  
})  
  
# Combine all into one data frame  
edges\_edges\_df <- do.call(rbind, edges\_list)  
  
head(edges\_edges\_df)  
  
weighted\_edges <- edges\_edges\_df %>%  
 count(from, to, name = "weight")  
head(weighted\_edges)

## Visualize Interactive Network Graph of Communications with visNetwork

### Communication Network (As a whole)

This is the baseline of the communication network based on the sub-types of organization, person, group, vessel, location and monitoring.

nodes <- mc3\_nodes\_final %>%  
 distinct(name, .keep\_all = TRUE) %>%  
 mutate(id = name, label = name) %>%  
 select(id, sub\_type)  
  
nodes$group <- nodes$sub\_type  
  
edges <- edges\_edges\_df %>%  
 mutate(arrows = "to",  
 title = paste0("From: ", from, "<br>To: ", to))  
  
visNetwork(nodes, edges) %>%  
 visEdges(smooth = FALSE, arrows = "to") %>%  
 visNodes(font = list(size = 20)) %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) %>%  
 visIgraphLayout(layout = "layout\_with\_fr") %>%  
 visLegend()

### Narrowed down specific sub-types

Since this is too big of a network, it will be narrowed down to organization, person and vessel based on the sub-types as the other 3 sub-types seems to show little correlation as shown in the network below.

important\_types are the 3 sub-types that we are looking more in-depth to see if there is any relationship between the 3 types. Making use of the group attribute that is created to clearly identify the different types of nodes inimportant\_types.

For ease of selecting a particular entity, a dropdown list with all the entities (nodes) that are present in the graph is shown.

Mouse pointer hover action is also included on the graph so that the user can hover the mouse pointer over the graph to look at the possible different ‘groups’ of connectivity as well as the number of messages sent to each ‘group’.

* The thicker the lines, the more messages are communicated between the sub-types.
* Outliers are isolated - no direct connections exclusively between them.

important\_types <- c("Organization", "Person", "Location")  
filtered\_nodes <- mc3\_nodes\_final %>%  
 filter(sub\_type %in% important\_types)  
  
filtered\_edges <- edges\_edges\_df %>%  
 filter(from %in% filtered\_nodes$name, to %in% filtered\_nodes$name)  
  
nodes <- filtered\_nodes %>%  
 rename(id = name, group = sub\_type, label\_new = name)  
  
edges <- filtered\_edges %>%  
 group\_by(from, to) %>%  
 summarise(value = n(), .groups = "drop") %>%  
 mutate(arrows = "to", title = paste("Messages:", value))  
  
visNetwork(nodes, edges) %>%  
 visGroups(groupname = "Organization", color = "tomato") %>%  
 visGroups(groupname = "Person", color = "skyblue") %>%  
 visGroups(groupname = "Location", color = "lightgreen") %>%  
 visIgraphLayout(layout = "layout\_with\_fr") %>%  
 visPhysics(enabled = FALSE) %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) %>%  
 visLegend()

Organization to Person Communication

important\_types <- c("Organization", "Person")  
filtered\_nodes <- mc3\_nodes\_final %>%  
 filter(sub\_type %in% important\_types)  
  
filtered\_edges <- edges\_edges\_df %>%  
 filter(from %in% filtered\_nodes$name, to %in% filtered\_nodes$name)  
  
nodes <- filtered\_nodes %>%  
 rename(id = name, group = sub\_type, label\_new = name)  
  
edges <- filtered\_edges %>%  
 group\_by(from, to) %>%  
 summarise(value = n(), .groups = "drop") %>%  
 mutate(arrows = "to", title = paste("Messages:", value))  
  
visNetwork(nodes, edges) %>%  
 visGroups(groupname = "Organization", color = "tomato") %>%  
 visGroups(groupname = "Person", color = "skyblue") %>%  
 visIgraphLayout(layout = "layout\_with\_fr") %>%  
 visPhysics(enabled = FALSE) %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) %>%  
 visLegend()

### Persons’ Communication

important\_types <- c("Person")  
filtered\_nodes <- mc3\_nodes\_final %>%  
 filter(sub\_type %in% important\_types)  
  
filtered\_edges <- edges\_edges\_df %>%  
 filter(from %in% filtered\_nodes$name, to %in% filtered\_nodes$name)  
  
nodes <- filtered\_nodes %>%  
 rename(id = name, group = sub\_type, label\_new = name)  
  
edges <- filtered\_edges %>%  
 group\_by(from, to) %>%  
 summarise(value = n(), .groups = "drop") %>%  
 mutate(arrows = "to", title = paste("Messages:", value))  
  
visNetwork(nodes, edges) %>%  
 visGroups(groupname = "Organization", color = "tomato") %>%  
 visGroups(groupname = "Person", color = "skyblue") %>%  
 visIgraphLayout(layout = "layout\_with\_fr") %>%  
 visPhysics(enabled = FALSE) %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) %>%  
 visLegend()

## 💡 Observations from Communication Network

* Nemo Reef has the highest communications among the locations shown.
* Miranda Jordan and Liam Thomson do not have any direct connections between any people as they are isolated.
* Boss seems to be entity that has the most influence among the organizations as Boss has the most edges in terms of communication.
* V. Miesel Shipping seems to be the entity that has the most influence among the persons with the most edges as well in terms of communication.

### Time-based Pattern Shifts

To detect behavioural changes between individuals and organizations based on the clusters forming or active nodes being isolated.

Firstly, Filter the dates to week 1 (first 7 days) and week 2 (next 7 days) from 01-10-2040 to 07-10-2040 and from 08-10-2040 to 15-10-2040.

# Ensure date\_only is created from timestamp  
mc3\_nodes\_final$date\_only <- as.Date(mc3\_nodes\_final$timestamp)  
  
end\_date <- max(mc3\_nodes\_final$date\_only, na.rm = TRUE)  
  
wk2\_start <- end\_date - 7  
wk1\_start <- end\_date - 14  
  
# Filter Week 1 (first 7 days)  
week1 <- mc3\_nodes\_final %>%  
 filter(date\_only >= wk1\_start & date\_only < wk2\_start) %>%  
 filter(type == "Event", label == "Communication")  
   
# Filter Week 2 (next 7 days)  
week2 <- mc3\_nodes\_final %>%  
 filter(date\_only >= wk2\_start | date\_only <= end\_date) %>%  
 filter(type == "Event", label == "Communication")

### Comparison of Week 1 and Week 2 Communications

week1\_ids <- week1$timestamp  
week2\_ids <- week2$timestamp  
  
edges\_week1 <- edges\_edges\_df %>% filter(timestamp %in% week1$timestamp)  
edges\_week2 <- edges\_edges\_df %>% filter(timestamp %in% week2$timestamp)  
  
  
edges\_wk1 <- edges\_week1 %>% select(from, to)  
edges\_wk2 <- edges\_week2 %>% select(from, to)  
  
nodes\_wk1 <- data.frame(  
 id = unique(c(edges\_week1$from, edges\_week1$to)),  
 label = unique(c(edges\_week1$from, edges\_week1$to))  
)  
  
edges\_wk1 <- edges\_week1 %>%  
 select(from, to)  
  
nodes\_wk2 <- data.frame(  
 id = unique(c(edges\_week2$from, edges\_week2$to)),  
 label = unique(c(edges\_week2$from, edges\_week2$to))  
)  
  
edges\_wk2 <- edges\_week2 %>%  
 select(from, to)  
  
visNetwork(nodes\_wk1, edges\_wk1, width = "100%", height = "400px") %>%  
 visIgraphLayout(layout = "layout\_with\_fr") %>%  
 visPhysics(enabled = FALSE) %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) %>%  
 visLegend()  
  
visNetwork(nodes\_wk2, edges\_wk2, width = "100%", height = "400px") %>%  
 visIgraphLayout(layout = "layout\_with\_fr") %>%  
 visPhysics(enabled = FALSE) %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) %>%  
 visLegend()

## 💡 Observations from Communication Time-based Pattern shifts

* As an overall, most communication are reduced when transitioning to week 2.

## Heatmap by Grouped Type (Organization, People, Vessel)

The heatmap is created to analyzed on the communication (i.e. number of messages) between the individuals as well as other subtypes as shown below. I will be narrowing it down to people’s communication to see if there is any suspicious activity going on between them.

### Heatmap (By type)

by\_type <- edges\_with\_meta %>%  
 group\_by(date, hour, sub\_type) %>%  
 summarise(message\_count = n(), .groups = "drop")  
  
ggplot(by\_type, aes(x = hour, y = date, fill = message\_count)) +  
 geom\_tile() +  
 facet\_wrap(~ sub\_type) +  
 scale\_fill\_gradient(low = "white", high = "purple") +  
 theme\_minimal() +  
 labs(  
 title = "Communication Patterns by Type",  
 x = "Hour of Day", y = "Date", fill = "Count"  
 )

### Heatmap (Individuals)

by\_person <- edges\_with\_meta %>%  
 filter(sub\_type == "Person") %>%  
 group\_by(date, hour) %>%  
 summarise(message\_count = n(), .groups = "drop")  
  
ggplot(by\_person, aes(x = hour, y = date, fill = message\_count)) +  
 geom\_tile(color = "white") +  
 scale\_fill\_gradient(low = "white", high = "purple") +  
 theme\_minimal() +  
 labs(  
 title = "Communication Patterns (Senders Who Are Persons)",  
 x = "Hour of Day", y = "Date", fill = "Message Count"  
 )

by\_person\_detail <- edges\_with\_meta %>%  
 filter(sub\_type == "Person") %>%  
 group\_by(date, hour, name) %>%  
 summarise(message\_count = n(), .groups = "drop")  
  
ggplot(by\_person\_detail, aes(x = hour, y = date, fill = message\_count)) +  
 geom\_tile() +  
 facet\_wrap(~ name) +  
 scale\_fill\_gradient(low = "white", high = "purple") +  
 theme\_minimal() +  
 labs(  
 title = "Message Timing for Each Person",  
 x = "Hour of Day", y = "Date", fill = "Count"  
 )

### Top 5 people that communicates the most

# Count total messages sent by each person  
top\_5\_people <- edges\_with\_meta %>%  
 filter(sub\_type == "Person") %>%  
 count(name, sort = TRUE) %>%  
 slice\_head(n = 5) %>%  
 pull(name) # extract just the names  
  
top\_people\_data <- edges\_with\_meta %>%  
 filter(sub\_type == "Person", name %in% top\_5\_people) %>%  
 group\_by(date, hour, name) %>%  
 summarise(message\_count = n(), .groups = "drop")  
  
ggplot(top\_people\_data, aes(x = hour, y = date, fill = message\_count)) +  
 geom\_tile(color = "white") +  
 facet\_wrap(~ name) +  
 scale\_fill\_gradient(low = "white", high = "darkorange") +  
 theme\_minimal() +  
 labs(  
 title = "Top 5 Most Active People: Communication Heatmap",  
 x = "Hour of Day", y = "Date", fill = "Messages Sent"  
 )

## 💡 Observations from Communication Heatmap

* It is shown that the top 5 individuals communicating are Sam, Davis, The Intern, Mrs Money and Boss has the most communication among all the individuals.
* The communication is mostly done between the 8th and 10th hour of the day.
* There isn’t really a distinguished date where they communicated the most.

top\_5\_people <- edges\_with\_meta %>%  
 filter(sub\_type == "Person") %>%  
 count(name, sort = TRUE) %>%  
 slice\_head(n = 5) %>%  
 pull(name)  
  
receiver\_meta <- mc3\_nodes\_final %>%  
 select(id, to\_name = name, to\_type = sub\_type)  
  
edges\_top5 <- edges\_with\_meta %>%  
 left\_join(receiver\_meta, by = c("to" = "id")) %>%  
 filter(name %in% top\_5\_people, to\_name %in% top\_5\_people)  
  
edge\_list <- edges\_top5 %>%  
 group\_by(name, to\_name) %>%  
 summarise(value = n(), .groups = "drop") %>%  
 rename(from = name, to = to\_name)  
  
nodes\_list <- unique(c(edge\_list$from, edge\_list$to))  
  
# Create nodes dataframe  
nodes <- data.frame(  
 id = nodes\_list,  
 label = nodes\_list,  
 title = nodes\_list,  
 group = "Person",   
 stringsAsFactors = FALSE  
)  
  
edges <- edge\_list %>%  
 mutate(  
 width = value, # edge thickness  
 label = as.character(value)   
 )  
  
visNetwork(nodes, edge\_list, width = "100%") %>%  
 visEdges(arrows = "to") %>%  
 visOptions(highlightNearest = TRUE, nodesIdSelection = TRUE) %>%  
 visLayout(randomSeed = 123) %>%  
 visPhysics(stabilization = TRUE) %>%  
 visLegend()

## 💡 Observations from Communication Heatmap

* Boss is the suspected mastermind and most suspicious as everyone is sending him all the updates and messages to him and shows that he has the most communication among the 5 people.