# Connnected Component Analysis

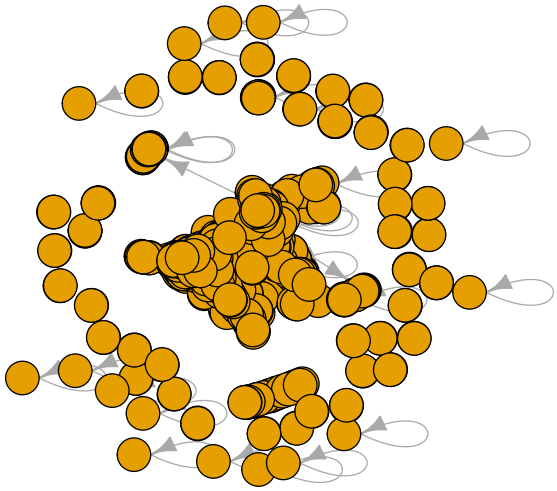
Johanes Panjaitan(39809579), Yuxuan Sun (27929934)

2025-12-04

## Contents

# 1 Data Loading and Preprocessing

```
# Load data
nodes_connected <- read.csv("data/nodes_connected.csv")
edges_connected <- read.csv("data/edges_connected.csv")
nodes_all_raw <- read.csv("data/nodes.csv")

# Filter out invalid nodes from nodes_all
nodes_all <- nodes_all_raw %>%
  filter(
    !is.na(title) & trimws(title) != "",  # Valid title
    !is.na(local_id),                     # Valid ID
    !is.na(year) & year <= 2025,  # Valid year range
    !is.na(citations) & citations >= 0,   # Valid citation count
    !is.na(references) & references >= 0, # Valid reference count
    !is.na(subtopic) & trimws(subtopic) != "",  # Valid subtopic
    !is.na(institution) & trimws(institution) != "" , # Valid institution
    !is.na(country) & country!=""
  )

# Create graph from connected component
connected_graph <- graph_from_data_frame(edges_connected, vertices = nodes_connected, directed = TRUE)
connected_graph <- simplify(connected_graph)
plot(connected_graph, vertex.label=NA)
```



```r
# Use nodes_connected as papers_df for consistency with analysis
papers_df <- nodes_connected

# Display basic information
cat("Connected component: ", vcount(connected_graph), "nodes,",
    ecount(connected_graph), "edges\n")
```

```
## Connected component:  1582 nodes, 3757 edges
```

```r
cat("Total papers in dataset: ", nrow(nodes_all), "\n")
```

```
## Total papers in dataset:  2588
```

# 2 Network Overview & Basic Statistics

## 2.1 Network Size and Structure

```r
# Calculate basic network statistics
stats_df <- data.frame(
  Metric = c(
    "Total Nodes",
    "Total Edges",
    "Average Degree",
    "Average In-Degree",
    "Average Out-Degree",
```

```
    "Network Density",
    "Network Diameter",
    "Average Path Length",
    "Number of Weakly Connected Components",
    "Number of Strongly Connected Components"
  ),
  Value = c(
    vcount(connected_graph),
    ecount(connected_graph),
    mean(degree(connected_graph)),
    mean(degree(connected_graph, mode = "in")),
    mean(degree(connected_graph, mode = "out")),
    edge_density(connected_graph),
    diameter(connected_graph, directed = TRUE),
    mean_distance(connected_graph, directed = TRUE),
    count_components(connected_graph, mode = "weak"),
    count_components(connected_graph, mode = "strong")
  )
)

kable(stats_df, digits = 3, caption = "Network Basic Statistics")
```

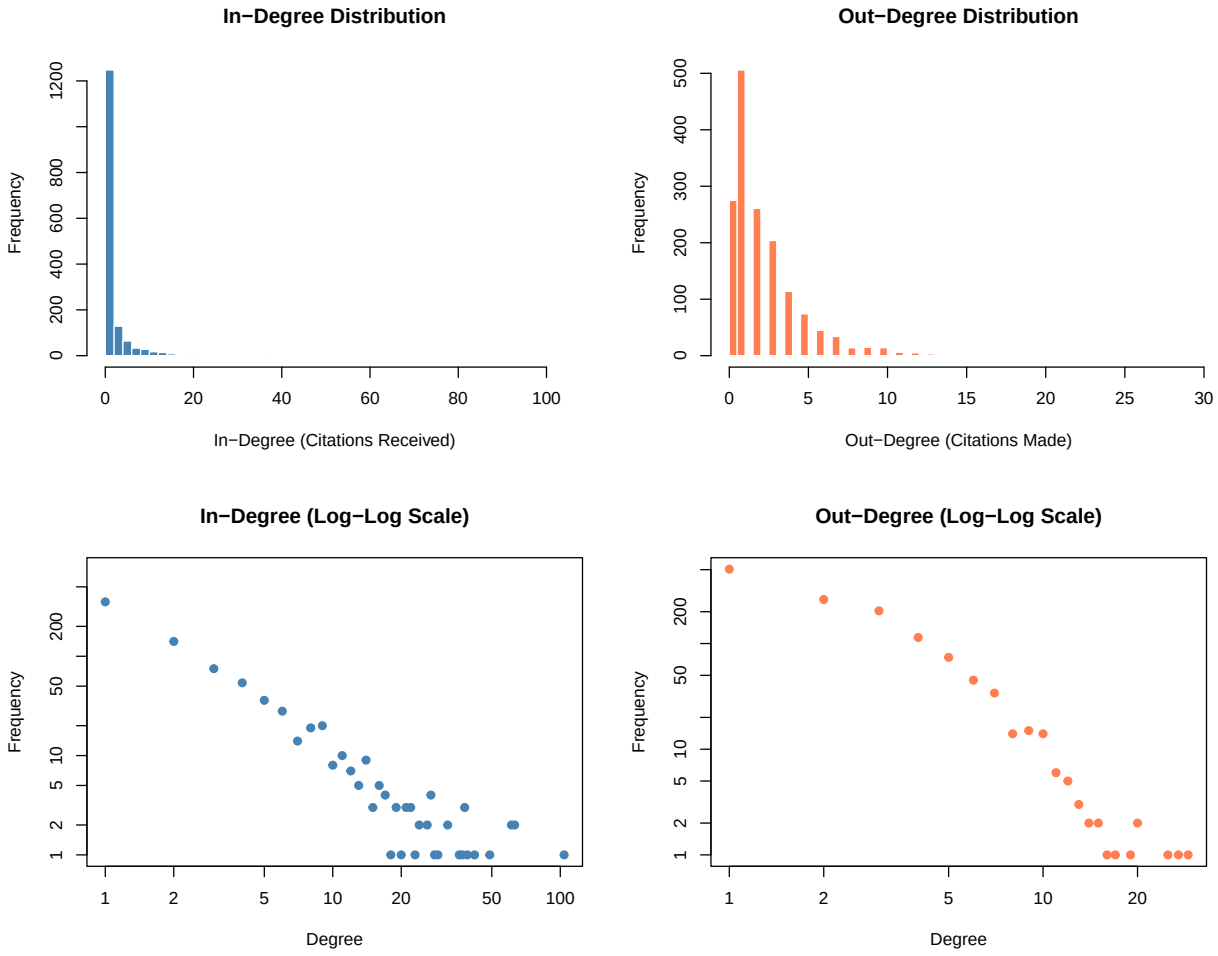
Table 1: Network Basic Statistics

| Metric | Value |
|---|---|
| Total Nodes | 1582.000 |
| Total Edges | 3757.000 |
| Average Degree | 4.750 |
| Average In-Degree | 2.375 |
| Average Out-Degree | 2.375 |
| Network Density | 0.002 |
| Network Diameter | 10.000 |
| Average Path Length | 3.148 |
| Number of Weakly Connected Components | 58.000 |
| Number of Strongly Connected Components | 1569.000 |

## 2.2 Degree Distribution

```
# Calculate degrees
in_deg <- degree(connected_graph, mode = "in")
out_deg <- degree(connected_graph, mode = "out")
total_deg <- degree(connected_graph, mode = "all")

# Create degree distribution plots
par(mfrow = c(2, 2))

# In-degree distribution
hist(in_deg, breaks = 50, main = "In-Degree Distribution",
     xlab = "In-Degree (Citations Received)", col = "steelblue", border = "white")
```

```r
# Out-degree distribution
hist(out_deg, breaks = 50, main = "Out-Degree Distribution",
     xlab = "Out-Degree (Citations Made)", col = "coral", border = "white")

# Log-log plot for in-degree
in_deg_table <- table(in_deg)
plot(as.numeric(names(in_deg_table)), as.numeric(in_deg_table),
     log = "xy", main = "In-Degree (Log-Log Scale)",
     xlab = "Degree", ylab = "Frequency", pch = 19, col = "steelblue")

# Log-log plot for out-degree
out_deg_table <- table(out_deg)
plot(as.numeric(names(out_deg_table)), as.numeric(out_deg_table),
     log = "xy", main = "Out-Degree (Log-Log Scale)",
     xlab = "Degree", ylab = "Frequency", pch = 19, col = "coral")
```




```r
par(mfrow = c(1, 1))
```

# 3 Research Question 1: Most Impactful Papers

## 3.1 Multiple Centrality Metrics

```
# Calculate multiple centrality metrics
V(connected_graph)$pagerank <- page_rank(connected_graph)$vector
V(connected_graph)$in_degree <- degree(connected_graph, mode = "in")
V(connected_graph)$betweenness <- betweenness(connected_graph, directed = TRUE)
V(connected_graph)$eigenvector <- eigen_centrality(connected_graph, directed = TRUE)$vector

# Create centrality data frame
centrality_df <- data.frame(
  local_id = V(connected_graph)$name,
  pagerank = V(connected_graph)$pagerank,
  in_degree = V(connected_graph)$in_degree,
  betweenness = V(connected_graph)$betweenness,
  eigenvector = V(connected_graph)$eigenvector
)

# Merge with paper metadata
centrality_df <- centrality_df %>%
  left_join(papers_df, by = "local_id")
```

## 3.2 Top 10 Most Impactful Papers (by PageRank)

```
# Top papers by PageRank
top_papers <- centrality_df %>%
  arrange(desc(pagerank)) %>%
  select(title, first_author, year, pagerank) %>%
  head(10)

kable(top_papers, digits = 4, caption = "Top 10 Papers by PageRank")
```

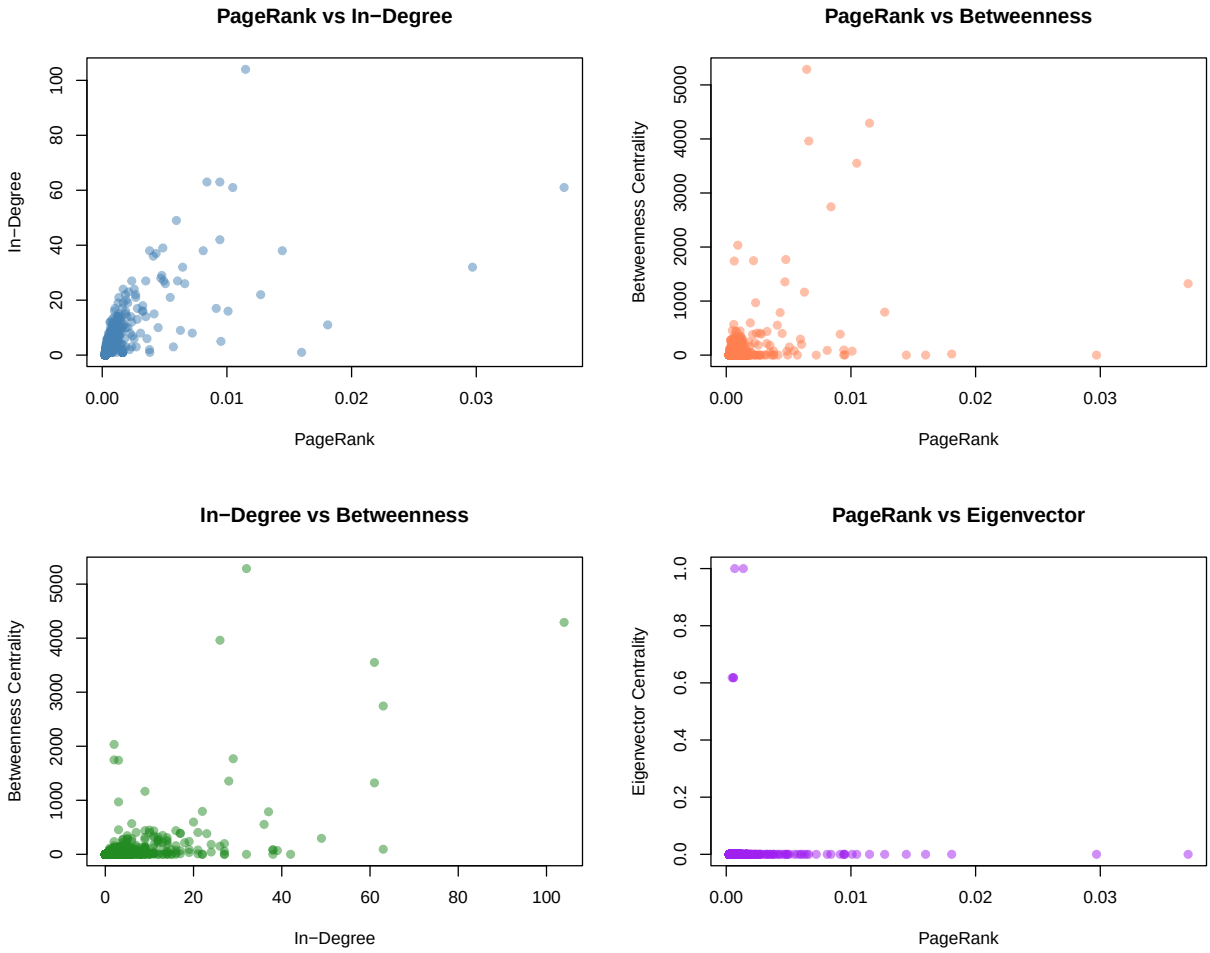
Table 2: Top 10 Papers by PageRank

| title | first_author | year | pagerank |
|---|---|---|---|
| Artificial intelligence in healthcare | Kun-Hsing Yu | 2018 | 0.0370 |
| Developing specific reporting guidelines for diagnostic accuracy studies assessing AI interventions: The STARD-AI Steering Group | Viknesh Sounderajah | 2020 | 0.0297 |
| Framing the challenges of artificial intelligence in medicine | Kun-Hsing Yu | 2018 | 0.0181 |
| Artificial intelligence (AI) systems for interpreting complex medical datasets | Rb Altman | 2017 | 0.0160 |
| Potential Liability for Physicians Using Artificial Intelligence | W. Nicholson Price | 2019 | 0.0144 |
| The “inconvenient truth” about AI in healthcare | Trishan Panch | 2019 | 0.0127 |
| Large language models in medicine | Arun James Thirunavukarasu | 2023 | 0.0115 |
| AI in health and medicine | Pranav Rajpurkar | 2022 | 0.0105 |

| title | first_author | year | pagerank |
|---|---|---|---|
| AI-Assisted Decision-making in Healthcare | Tamra Lysaght | 2019 | 0.0101 |
| An Ethics Framework for Big Data in Health and Research | Vicki Xafis | 2019 | 0.0095 |

### 3.3 Comparison of Ranking Metrics

```
# Create ranking comparison
top_by_pagerank <- centrality_df %>% arrange(desc(pagerank)) %>% head(20)
top_by_indegree <- centrality_df %>% arrange(desc(in_degree)) %>% head(20)
top_by_betweenness <- centrality_df %>% arrange(desc(betweenness)) %>% head(20)

# Scatter plots comparing metrics
par(mfrow = c(2, 2))

plot(centrality_df$pagerank, centrality_df$in_degree,
     xlab = "PageRank", ylab = "In-Degree",
     main = "PageRank vs In-Degree", pch = 19, col = alpha("steelblue", 0.5))

plot(centrality_df$pagerank, centrality_df$betweenness,
     xlab = "PageRank", ylab = "Betweenness Centrality",
     main = "PageRank vs Betweenness", pch = 19, col = alpha("coral", 0.5))

plot(centrality_df$in_degree, centrality_df$betweenness,
     xlab = "In-Degree", ylab = "Betweenness Centrality",
     main = "In-Degree vs Betweenness", pch = 19, col = alpha("forestgreen", 0.5))

plot(centrality_df$pagerank, centrality_df$eigenvector,
     xlab = "PageRank", ylab = "Eigenvector Centrality",
     main = "PageRank vs Eigenvector", pch = 19, col = alpha("purple", 0.5))
```




```
par(mfrow = c(1, 1))
```

## 3.4 Papers Ranking High on Multiple Metrics

```
# Normalize metrics to [0,1] for comparison
centrality_df <- centrality_df %>%
  mutate(
    pagerank_norm = (pagerank - min(pagerank)) / (max(pagerank) - min(pagerank)),
    indegree_norm = (in_degree - min(in_degree)) / (max(in_degree) - min(in_degree)),
    betweenness_norm = (betweenness - min(betweenness)) / (max(betweenness) - min(betweenness)),
    combined_score = pagerank_norm + indegree_norm + betweenness_norm
  )

# Top papers by combined metrics
multi_metric_top <- centrality_df %>%
  arrange(desc(combined_score)) %>%
  select(title, year, pagerank_norm, indegree_norm, betweenness_norm, combined_score) %>%
  head(10)
library(kableExtra)
```

```r
kable(multi_metric_top, digits = 3,
      caption = "Top 10 Papers by Combined Metrics",
      booktabs = TRUE) %>%
  kableExtra::kable_styling() %>%
  column_spec(1, width = "15em") %>%   # title column wide
  column_spec(2, width = "2em") %>% # year
  column_spec(3:5, width="6em") %>%
  column_spec(6, width = "8.5em")       # numeric columns
```

Table 3: Top 10 Papers by Combined Metrics

| title | year | pagerank_norm | indegree_norm | betweenness_norm | combined_score |
|---|---|---|---|---|---|
| Large language models in medicine | 2023 | 0.306 | 1.000 | 0.812 | 2.117 |
| Artificial intelligence in healthcare | 2018 | 1.000 | 0.587 | 0.250 | 1.837 |
| AI in health and medicine | 2022 | 0.278 | 0.587 | 0.672 | 1.536 |
| The shaky foundations of large language models and foundation models for electronic health records | 2023 | 0.169 | 0.308 | 1.000 | 1.476 |
| Foundation models for generalist medical artificial intelligence | 2023 | 0.222 | 0.606 | 0.519 | 1.346 |
| Multimodal biomedical AI | 2022 | 0.173 | 0.250 | 0.749 | 1.172 |
| Developing specific reporting guidelines for diagnostic accuracy studies assessing AI interventions: The STARD-AI Steering Group | 2020 | 0.800 | 0.308 | 0.000 | 1.108 |
| Reporting guidelines for clinical trial reports for interventions involving artificial intelligence: the CONSORT-AI extension | 2020 | 0.250 | 0.606 | 0.018 | 0.873 |
| Potential Liability for Physicians Using Artificial Intelligence | 2019 | 0.386 | 0.365 | 0.000 | 0.751 |
| Creation and Adoption of Large Language Models in Medicine | 2023 | 0.123 | 0.279 | 0.335 | 0.737 |

# 4 Research Question 2: Oldest Papers with Lasting Relevance

## 4.1 Papers with High Closeness Centrality (Sorted by Age)

```r
# Calculate closeness centrality for ALL papers
all_paper_ids <- V(connected_graph)$name
closeness_scores <- closeness(connected_graph, vids = all_paper_ids, mode = "all")

# Add closeness to centrality_df
centrality_df$closeness <- closeness_scores[match(centrality_df$local_id, names(closeness_scores))]

# Filter for papers with valid closeness (not NA or 0) and sort by year (oldest first), then by closene
oldest_papers <- centrality_df %>%
```

```r
  filter(!is.na(closeness), closeness > 0, !is.na(year)) %>%
  arrange(year, desc(closeness)) %>%
  head(10)

# Create display table
oldest_papers_display <- oldest_papers %>%
  select(title, year, closeness, in_degree)

kable(oldest_papers_display, digits = 3,
      caption = "Papers sorted based on year and closeness centrality")
```

Table 4: Papers sorted based on year and closeness centrality

| title | year | closeness | in_degree |
|---|---|---|---|
| Exploring big educational learner corpora for SLA research | 2015 | 1.000 | 1 |
| Incremental Dependency Parsing and Disfluency Detection in Spoken Learner English | 2015 | 0.333 | 1 |
| How to Train good Word Embeddings for Biomedical NLP | 2016 | 0.333 | 1 |
| AI as evaluator: Search driven playtesting of modern board games | 2017 | 1.000 | 0 |
| Findings of the VarDial Evaluation Campaign 2017 | 2017 | 0.200 | 3 |
| A Report on the 2017 Native Language Identification Shared Task | 2017 | 0.167 | 1 |
| Investigating the cross-lingual translatability of VerbNet-style classification | 2017 | 0.004 | 1 |
| What This Computer Needs Is a Physician | 2017 | 0.000 | 13 |
| Artificial intelligence (AI) systems for interpreting complex medical datasets | 2017 | 0.000 | 1 |
| Segmenting and POS tagging Classical Tibetan using a memory-based tagger | 2018 | 1.000 | 1 |

## 4.2 Citation Longevity Analysis

```r
# Analyze citations to oldest papers from recent papers (2020-2025)
recent_papers <- papers_df %>% filter(year >= 2020 & year <= 2025)

# Get edges from recent papers to oldest papers
citation_longevity <- data.frame()

for (old_id in oldest_papers$local_id) {
  # Get papers that cite this old paper
  citing_papers <- neighbors(connected_graph, old_id, mode = "in")
  citing_ids <- V(connected_graph)[citing_papers]$name

  # Check which citing papers are recent
  recent_citations <- sum(citing_ids %in% recent_papers$local_id)
  total_citations <- length(citing_ids)

  citation_longevity <- rbind(citation_longevity, data.frame(
    local_id = old_id,
    total_citations = total_citations,
    recent_citations = recent_citations,
    recent_ratio = ifelse(total_citations > 0, recent_citations / total_citations, 0)
  ))
```

```
}

# Merge with paper info
citation_longevity <- citation_longevity %>%
  left_join(papers_df, by = "local_id") %>%
  arrange(desc(recent_citations)) %>%
  select(title, year, total_citations, recent_citations)

kable(head(citation_longevity, 10), digits = 3,
      caption = "Papers that was cited by most recent papers")
```

Table 5: Papers that was cited by most recent papers

| title | year | total_citations | recent_citations |
|---|---|---|---|
| What This Computer Needs Is a Physician | 2017 | 13 | 7 |
| Exploring big educational learner corpora for SLA research | 2015 | 1 | 1 |
| Incremental Dependency Parsing and Disfluency Detection in Spoken Learner English | 2015 | 1 | 1 |
| How to Train good Word Embeddings for Biomedical NLP | 2016 | 1 | 1 |
| Findings of the VarDial Evaluation Campaign 2017 | 2017 | 3 | 1 |
| Investigating the cross-lingual translatability of VerbNet-style classification | 2017 | 1 | 1 |
| AI as evaluator: Search driven playtesting of modern board games | 2017 | 0 | 0 |
| A Report on the 2017 Native Language Identification Shared Task | 2017 | 1 | 0 |
| Artificial intelligence (AI) systems for interpreting complex medical datasets | 2017 | 1 | 0 |
| Segmenting and POS tagging Classical Tibetan using a memory-based tagger | 2018 | 1 | 0 |

# 5 Research Question 3: Subtopic Concentration

## 5.1 Community Detection

```
# Apply Louvain community detection
set.seed(42)
communities <- cluster_louvain(as.undirected(connected_graph))

# Add community membership to vertices
V(connected_graph)$community <- membership(communities)

# Calculate modularity
modularity_score <- modularity(communities)
cat("Modularity score:", modularity_score, "\n")
```

```
## Modularity score: 0.5981146
```

```
cat("Number of communities detected:", length(communities), "\n")
```

```
## Number of communities detected: 77
```

## 5.2 Community Statistics

```
# Calculate statistics for each community
community_stats <- data.frame()

for (comm_id in unique(V(connected_graph)$community)) {
  # Get subgraph for this community
  comm_nodes <- V(connected_graph)[V(connected_graph)$community == comm_id]
  subgraph <- induced_subgraph(connected_graph, comm_nodes)

  # Calculate statistics
  comm_size <- vcount(subgraph)
  comm_edges <- ecount(subgraph)
  comm_density <- edge_density(subgraph)

  community_stats <- rbind(community_stats, data.frame(
    community = comm_id,
    size = comm_size,
    edges = comm_edges,
    density = comm_density,
    avg_degree = mean(degree(subgraph))
  ))
}

community_stats <- community_stats %>%
  arrange(desc(size))

kable(head(community_stats, 10), digits = 3,
      caption = "Top 10 Communities by Size")
```

Table 6: Top 10 Communities by Size

| community | size | edges | density | avg_degree |
|---|---|---|---|---|
| 1 | 251 | 546 | 0.009 | 4.351 |
| 4 | 160 | 249 | 0.010 | 3.113 |
| 5 | 155 | 442 | 0.019 | 5.703 |
| 10 | 135 | 283 | 0.016 | 4.193 |
| 11 | 133 | 217 | 0.012 | 3.263 |
| 8 | 124 | 178 | 0.012 | 2.871 |
| 2 | 85 | 107 | 0.015 | 2.518 |
| 19 | 59 | 76 | 0.022 | 2.576 |
| 24 | 53 | 56 | 0.020 | 2.113 |
| 14 | 51 | 59 | 0.023 | 2.314 |

## 5.3 Predominant Topics by Community

```
# Analyze subtopics within each community
community_topics <- data.frame()
```

```r
for (comm_id in head(unique(V(connected_graph)$community), 10)) {
  # Get papers in this community
  comm_paper_ids <- V(connected_graph)[V(connected_graph)$community == comm_id]$name

  # Get subtopics for these papers
  comm_papers <- papers_df %>%
    filter(local_id %in% comm_paper_ids)

  # Count subtopic frequencies (subtopic is singular, not a list)
  subtopic_freq <- comm_papers %>%
    count(subtopic, sort = TRUE) %>%
    head(5)

  subtopic_freq$community <- comm_id
  community_topics <- rbind(community_topics, subtopic_freq)
}

if (nrow(community_topics) > 0) {
  kable(head(community_topics, 20), caption = "Top Subtopics by Community")
}
```

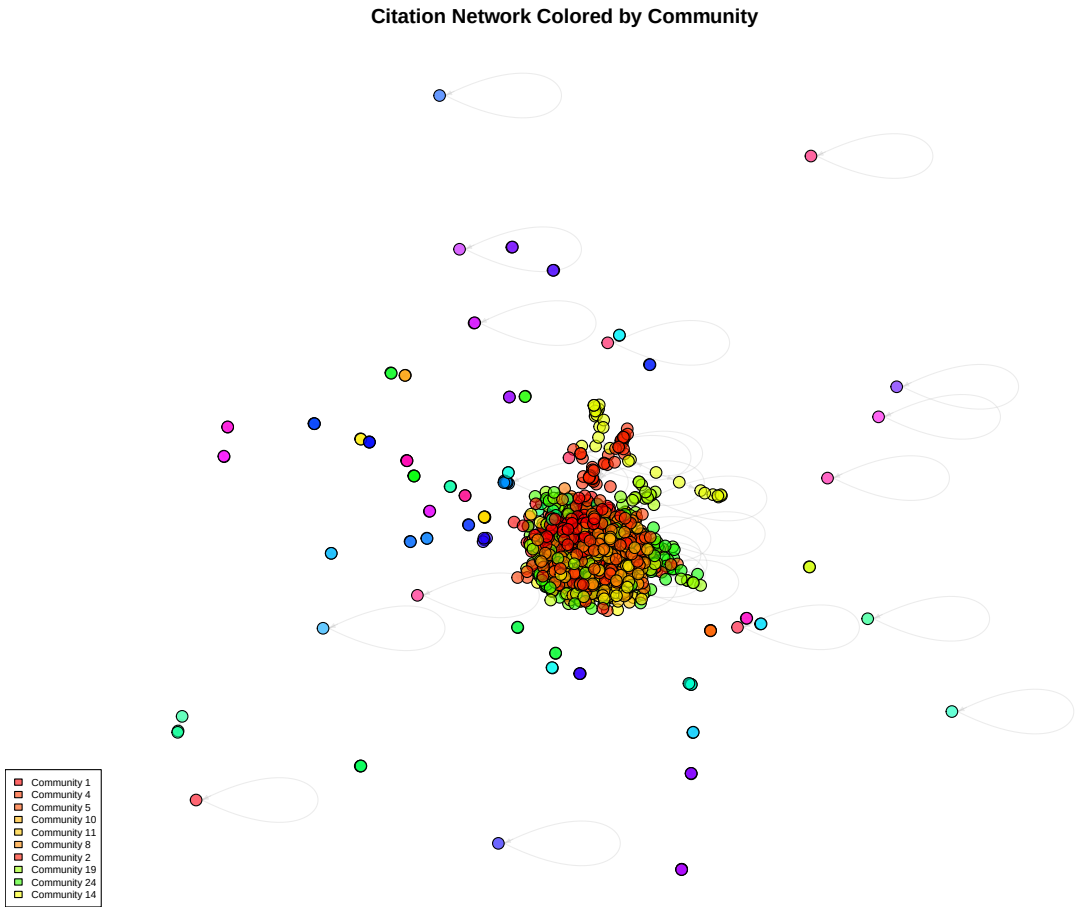
Table 7: Top Subtopics by Community

| subtopic | n | community |
|---|---|---|
| Artificial Intelligence in Healthcare and Education | 209 | 1 |
| Machine Learning in Healthcare | 34 | 1 |
| Artificial Intelligence in Healthcare | 2 | 1 |
| Machine Learning in Bioinformatics | 2 | 1 |
| Artificial Intelligence in Games | 1 | 1 |
| Artificial Intelligence in Healthcare and Education | 70 | 2 |
| Machine Learning in Healthcare | 9 | 2 |
| Artificial Intelligence in Healthcare | 3 | 2 |
| Artificial Intelligence in Law | 1 | 2 |
| Machine Learning and Data Classification | 1 | 2 |
| Machine Learning in Materials Science | 46 | 3 |
| Artificial Intelligence in Healthcare and Education | 145 | 4 |
| Machine Learning in Healthcare | 9 | 4 |
| Artificial Intelligence in Healthcare | 5 | 4 |
| Natural Language Processing Techniques | 1 | 4 |
| Artificial Intelligence in Healthcare and Education | 138 | 5 |
| Machine Learning in Healthcare | 16 | 5 |
| Machine Learning and Data Classification | 1 | 5 |
| Machine Learning in Healthcare | 2 | 6 |
| Natural Language Processing Techniques | 1 | 6 |

## 5.4 Community Visualization

```r
# Create layout for visualization
set.seed(42)
layout_fr <- layout_with_fr(connected_graph)
```

```r
# Color palette for communities
num_communities <- length(unique(V(connected_graph)$community))
colors <- rainbow(num_communities, alpha = 0.6)

# Plot network colored by community
plot(connected_graph,
     vertex.color = colors[V(connected_graph)$community],
     vertex.size = 3,
     vertex.label = NA,
     edge.arrow.size = 0.3,
     edge.color = alpha("gray", 0.3),
     layout = layout_fr,
     main = "Citation Network Colored by Community")

# Add legend for top communities
top_communities <- head(unique(community_stats$community), 10)
legend("bottomleft",
       legend = paste("Community", top_communities),
       fill = colors[top_communities],
       cex = 0.6)
```




## 5.5 Inter-Community Connections

```
# Calculate edges between communities
edge_list <- as_edgelist(connected_graph, names = TRUE)
edge_communities <- data.frame(
  from_comm = V(connected_graph)$community[match(edge_list[,1], V(connected_graph)$name)],
  to_comm = V(connected_graph)$community[match(edge_list[,2], V(connected_graph)$name)]
)

# Count inter vs intra-community edges
edge_communities$edge_type <- ifelse(
  edge_communities$from_comm == edge_communities$to_comm,
  "Intra-community",
  "Inter-community"
)

edge_type_summary <- table(edge_communities$edge_type)
kable(as.data.frame(edge_type_summary),
      caption = "Intra-community vs Inter-community Edges")
```

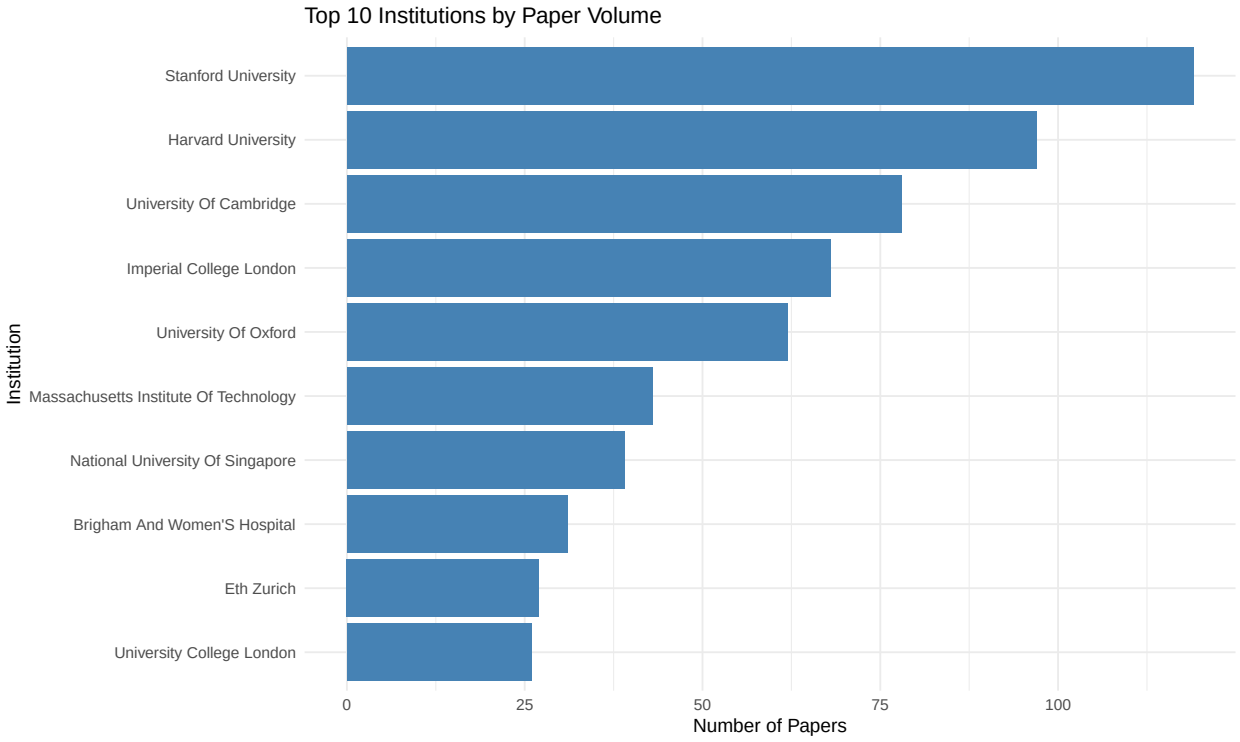
Table 8: Intra-community vs Inter-community Edges

| Var1 | Freq |
|---|---|
| Inter-community | 1102 |
| Intra-community | 2655 |

# 6 Research Question 4: Institution/Country Output

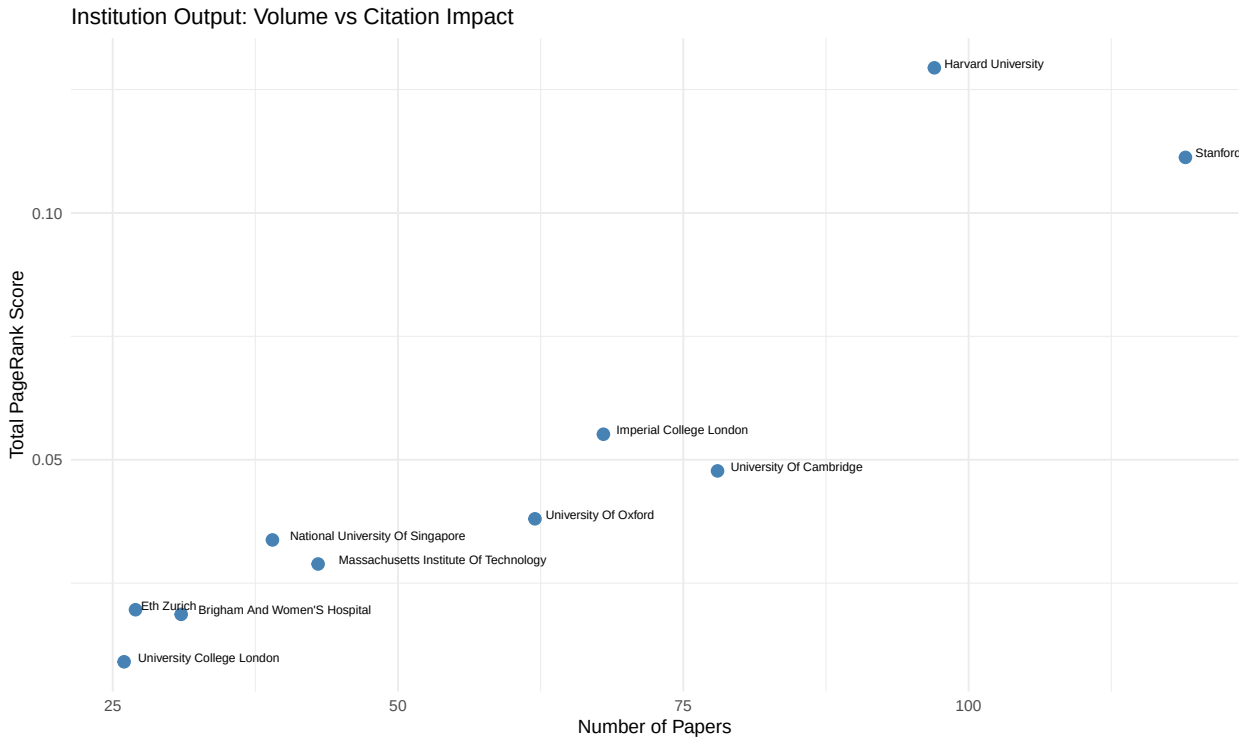
## 6.1 Paper Count by Institution

```
# Count papers by institution
if ("institution" %in% names(papers_df)) {
  institution_counts <- papers_df %>%
    count(institution, sort = TRUE) %>%
    head(10)

  # Bar plot
  ggplot(institution_counts, aes(x = reorder(institution, n), y = n)) +
    geom_bar(stat = "identity", fill = "steelblue") +
    coord_flip() +
    labs(title = "Top 10 Institutions by Paper Volume",
         x = "Institution", y = "Number of Papers") +
    theme_minimal()
}
```



### 6.2 Citation-Weighted Impact by Institution

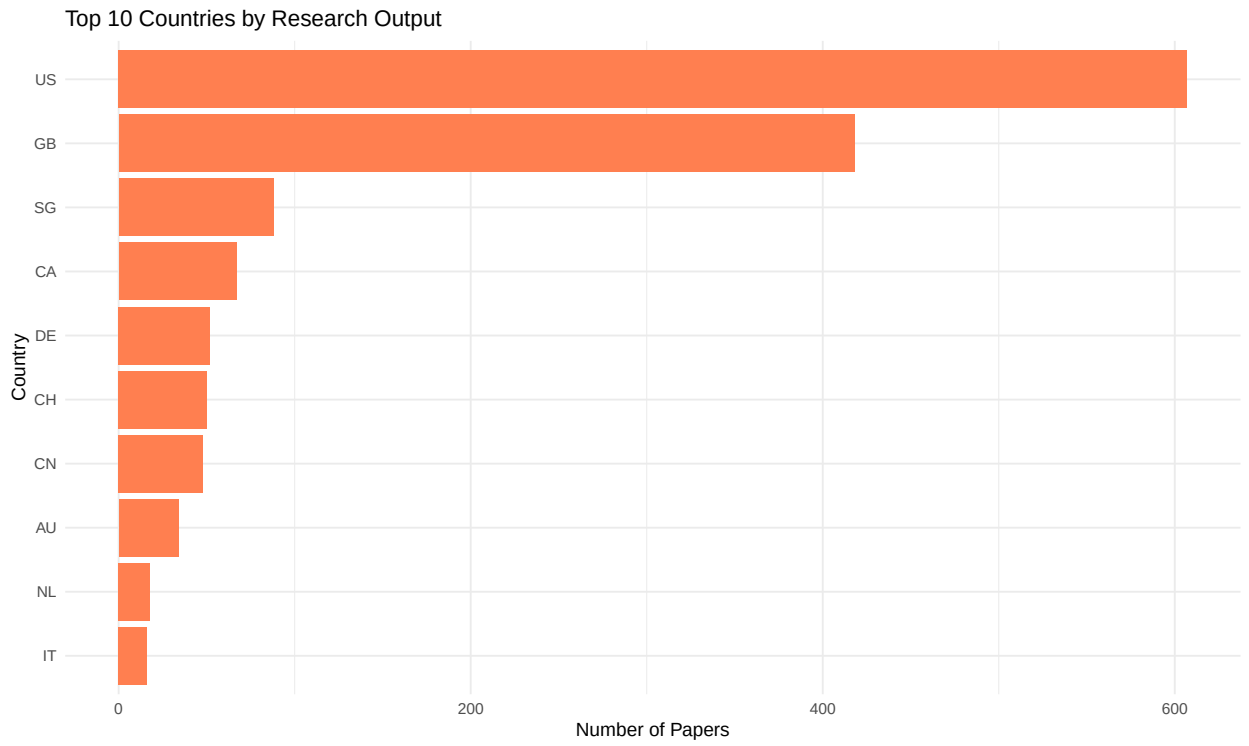
```
if ("institution" %in% names(papers_df)) {
  # Calculate total PageRank by institution
  institution_impact <- centrality_df %>%
    group_by(institution) %>%
    summarise(
      paper_count = n(),
      total_pagerank = sum(pagerank, na.rm = TRUE),
      avg_pagerank = mean(pagerank, na.rm = TRUE),
      total_citations = sum(in_degree, na.rm = TRUE),
      avg_citations = mean(in_degree, na.rm = TRUE)
    ) %>%
    arrange(desc(total_pagerank)) %>%
    head(20)

  kable(institution_impact, digits = 4,
        caption = "Top Institutions by Citation Impact")

  # Comparison plot: Volume vs Impact
  comparison_df <- institution_counts %>%
    left_join(institution_impact, by = "institution") %>%
    filter(!is.na(total_pagerank))

  ggplot(comparison_df, aes(x = n, y = total_pagerank, label = institution)) +
    geom_point(size = 3, color = "steelblue") +
    geom_text(hjust = -0.1, vjust = 0, size = 2.5) +
    labs(title = "Institution Output: Volume vs Citation Impact",
         x = "Number of Papers", y = "Total PageRank Score") +
    theme_minimal()
}
```




## 6.3 Country-Level Analysis

```r
if ("country" %in% names(papers_df)) {
  # Count papers by country
  country_counts <- papers_df %>%
    filter(!is.na(country)&country!="") %>%
    count(country, sort = TRUE) %>%
    head(10)

  # Country impact
  country_impact <- centrality_df %>%
    filter(!is.na(country)&country!="") %>%
    group_by(country) %>%
    summarise(
      paper_count = n(),
      total_pagerank = sum(pagerank, na.rm = TRUE),
      avg_pagerank = mean(pagerank, na.rm = TRUE)
    ) %>%
    arrange(desc(total_pagerank)) %>%
    head(10)

  kable(country_impact, digits = 4, caption = "Top Countries by Research Impact")

  # Visualization
  ggplot(country_counts, aes(x = reorder(country, n), y = n)) +
    geom_bar(stat = "identity", fill = "coral") +
    coord_flip() +
```

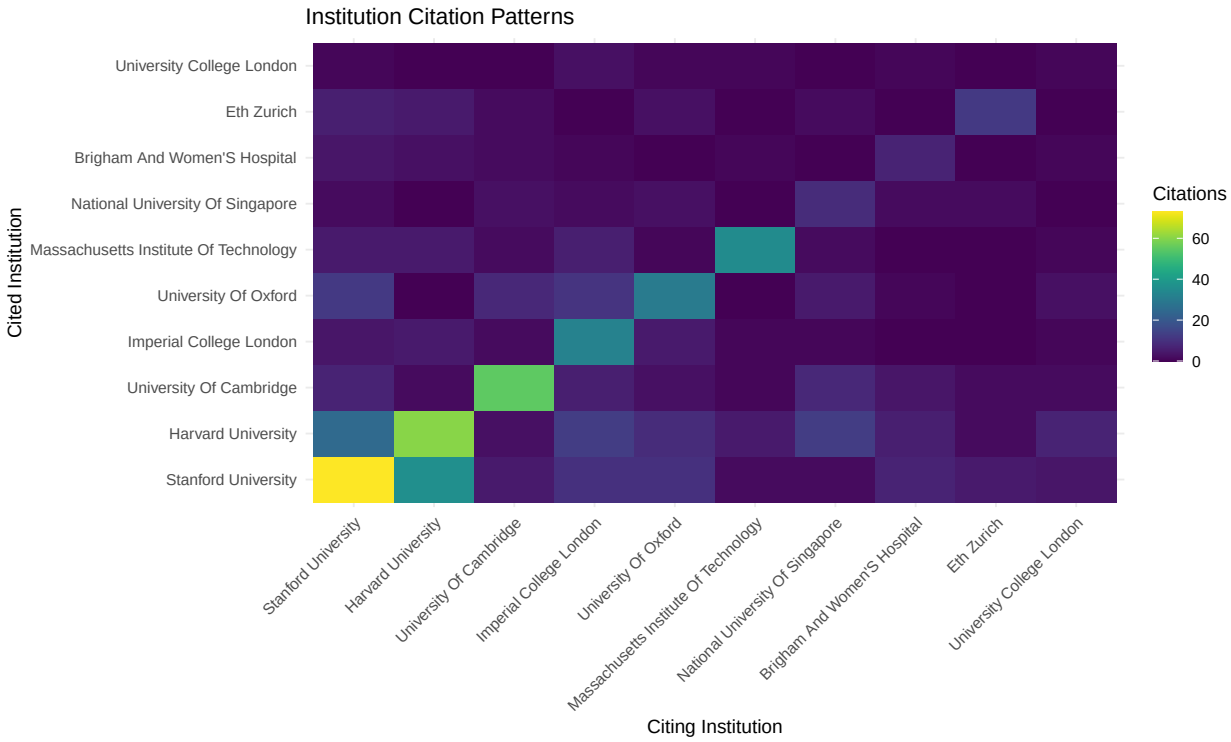
```
    labs(title = "Top 10 Countries by Research Output",
         x = "Country", y = "Number of Papers") +
    theme_minimal()
}
```




## 6.4 Institution Collaboration Patterns

```
if ("institution" %in% names(papers_df)) {
  # Find co-authorship between institutions (papers citing each other)
  top_institutions <- head(institution_counts$institution, 10)

  collaboration_matrix <- matrix(0, nrow = length(top_institutions),
                                 ncol = length(top_institutions))
  rownames(collaboration_matrix) <- top_institutions
  colnames(collaboration_matrix) <- top_institutions

  # Count citations between institutions
  for (i in 1:length(top_institutions)) {
    for (j in 1:length(top_institutions)) {
      inst_i_papers <- papers_df %>% filter(institution == top_institutions[i]) %>% pull(local_id)
      inst_j_papers <- papers_df %>% filter(institution == top_institutions[j]) %>% pull(local_id)

      # Count edges from i to j
      edges_ij <- sum(edge_list[,1] %in% inst_i_papers & edge_list[,2] %in% inst_j_papers)
      collaboration_matrix[i, j] <- edges_ij
    }
  }
```

```
  # Heatmap
  library(reshape2)
  collab_melt <- melt(collaboration_matrix)

  ggplot(collab_melt, aes(x = Var1, y = Var2, fill = value)) +
    geom_tile() +
    scale_fill_viridis() +
    theme_minimal() +
    theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
    labs(title = "Institution Citation Patterns",
         x = "Citing Institution", y = "Cited Institution",
         fill = "Citations")
}
```




# 7 Research Question 5: Research Directions

## 7.1 Temporal Analysis of Citation Patterns

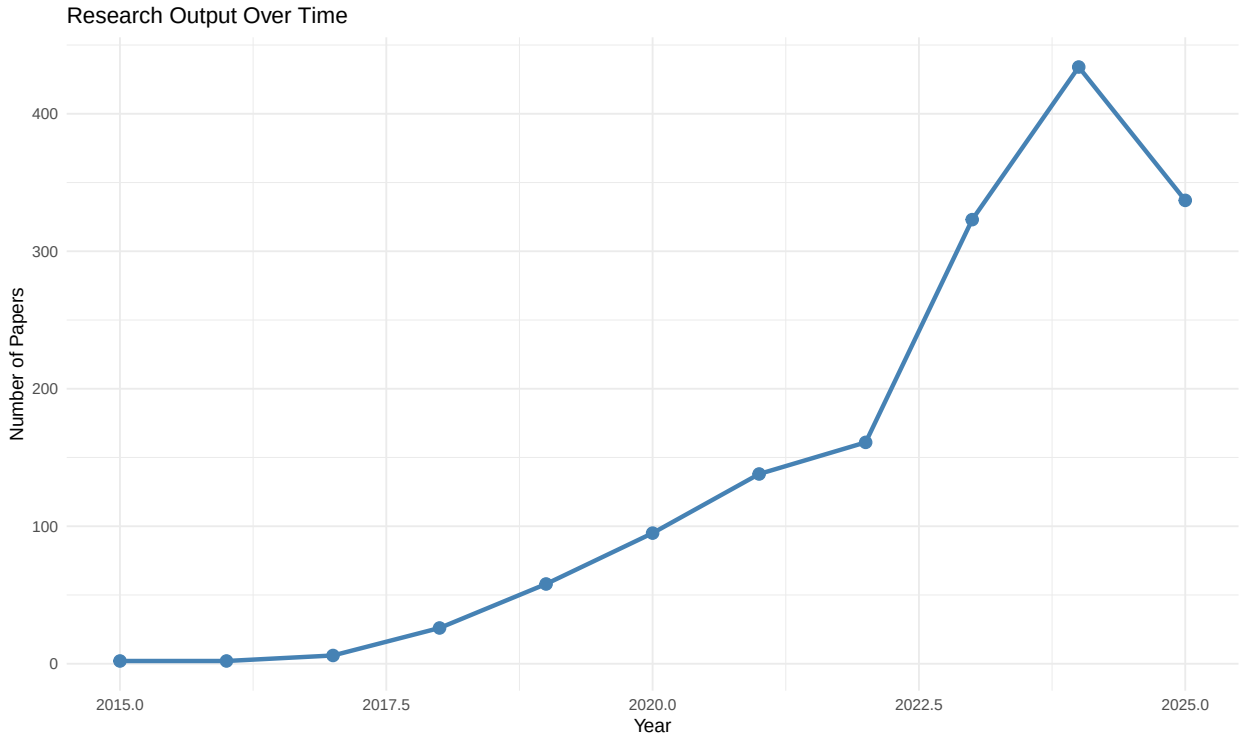
```
# Divide into time periods
papers_df <- papers_df %>%
  mutate(era = case_when(
    year >= 2015 & year <= 2018 ~ "2015-2018",
    year >= 2019 & year <= 2021 ~ "2019-2021",
    year >= 2022 & year <= 2025 ~ "2022-2025",
    TRUE ~ "Other"
```

```
  ))

# Count papers by era
era_counts <- papers_df %>%
  count(era) %>%
  filter(era != "Other")

kable(era_counts, caption = "Papers by Time Period")
```

Table 9: Papers by Time Period

| era | n |
|---|---|
| 2015-2018 | 36 |
| 2019-2021 | 291 |
| 2022-2025 | 1255 |

```
# Plot papers over time
papers_by_year <- papers_df %>%
  count(year) %>%
  filter(year >= 2015 & year <= 2025)

ggplot(papers_by_year, aes(x = year, y = n)) +
  geom_line(size = 1.2, color = "steelblue") +
  geom_point(size = 3, color = "steelblue") +
  labs(title = "Research Output Over Time",
       x = "Year", y = "Number of Papers") +
  theme_minimal()
```



## 7.2 Emerging Bridge Papers (2022-2024)

```
# Recent papers with high betweenness
recent_bridge <- centrality_df %>%
  filter(year >= 2020 & year <= 2025) %>%
  arrange(desc(betweenness)) %>%
  select(title, first_author, year, betweenness, in_degree) %>%
  head(10)

kable(recent_bridge, digits = 3,
      caption = "Recent Papers with High Betweenness (Bridge Papers)")
```

Table 10: Recent Papers with High Betweenness (Bridge Papers)

| title | first_author | year | betweenness | in_degree |
|---|---|---|---|---|
| The shaky foundations of large language models and foundation models for electronic health records | Michael Wornow | 2023 | 5287.655 | 32 |
| Large language models in medicine | Arun James Thirunavukarasu | 2023 | 4291.818 | 104 |
| Multimodal biomedical AI | Julián N. Acosta | 2022 | 3960.812 | 26 |
| AI in health and medicine | Pranav Rajpurkar | 2022 | 3551.094 | 61 |
| Foundation models for generalist medical artificial intelligence | Michael Moor | 2023 | 2744.235 | 63 |
| QUEST-AI: A System for Question Generation, Verification, and Refinement using AI for USMLE-Style Exams | Suhana Bedi | 2023 | 2034.000 | 2 |
| Creation and Adoption of Large Language Models in Medicine | Nigam H. Shah | 2023 | 1768.982 | 29 |
| Ensuring that biomedical AI benefits diverse populations | James Zou | 2021 | 1747.330 | 2 |
| A Systematic Review of Testing and Evaluation of Healthcare Applications of Large Language Models (LLMs) | Suhana Bedi | 2024 | 1740.675 | 3 |
| AI recognition of patient race in medical imaging: a modelling study | Judy Wawira Gichoya | 2022 | 1354.800 | 28 |

## 7.3 Trend-Setting Papers

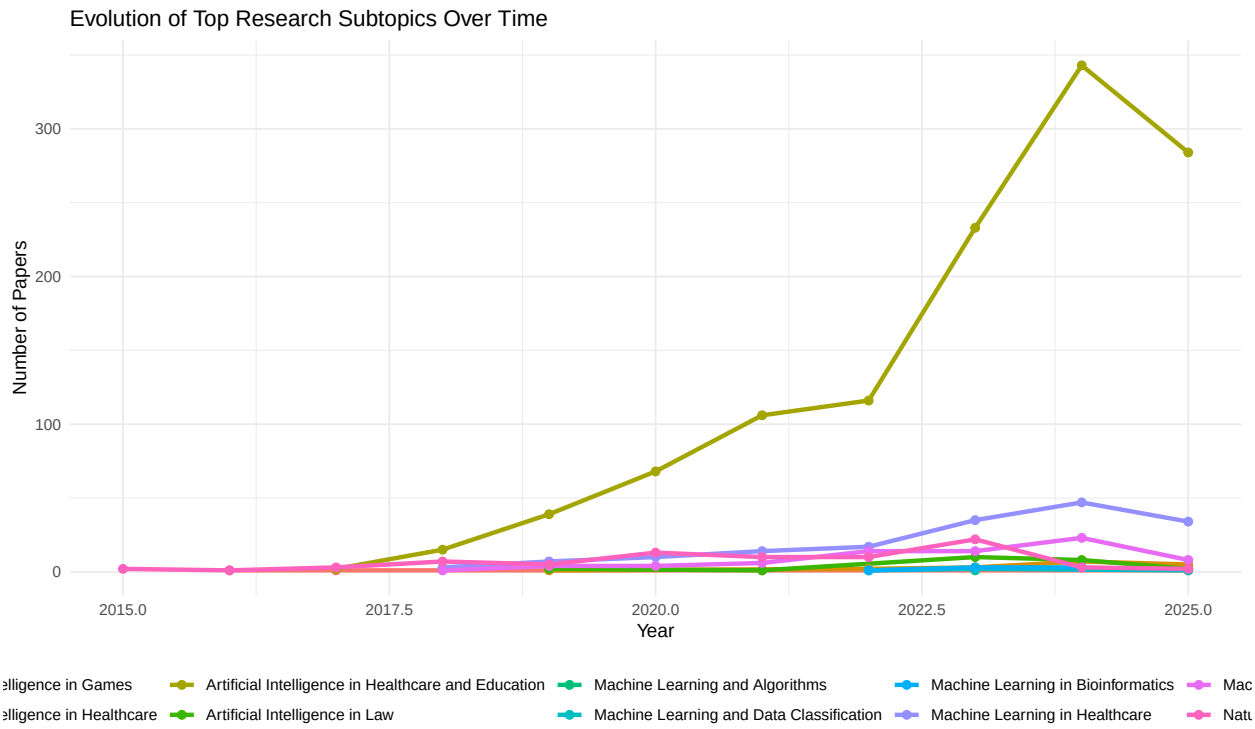
```
# Recent papers with high PageRank (rapid impact)
recent_impact <- centrality_df %>%
  filter(year >= 2022 & year <= 2025) %>%
  arrange(desc(betweenness)) %>%
  arrange(desc(pagerank)) %>%
  select(title, first_author, year, pagerank, in_degree) %>%
  head(10)

kable(recent_impact, digits = 4, caption = "Recent High-Impact Papers (2022-2024)")
```

Table 11: Recent High-Impact Papers (2022-2024)

| title | first_author | year | pagerank | in_degree |
|---|---|---|---|---|
| Large language models in medicine | Arun James Thirunavukarasu | 2023 | 0.0115 | 104 |
| AI in health and medicine | Pranav Rajpurkar | 2022 | 0.0105 | 61 |
| Foundation models for generalist medical artificial intelligence | Michael Moor | 2023 | 0.0084 | 63 |
| Multimodal biomedical AI | Julián N. Acosta | 2022 | 0.0066 | 26 |
| The shaky foundations of large language models and foundation models for electronic health records | Michael Wornow | 2023 | 0.0065 | 32 |
| The Diagnostic and Triage Accuracy of the GPT-3 Artificial Intelligence Model | David M Levine | 2023 | 0.0063 | 9 |
| Creation and Adoption of Large Language Models in Medicine | Nigam H. Shah | 2023 | 0.0048 | 29 |
| AI recognition of patient race in medical imaging: a modelling study | Judy Wawira Gichoya | 2022 | 0.0047 | 28 |
| Reporting guideline for the early-stage clinical evaluation of decision support systems driven by artificial intelligence: DECIDE-AI | Baptiste Vasey | 2022 | 0.0043 | 37 |
| GenSLMs: Genome-scale language models reveal SARS-CoV-2 evolutionary dynamics | Maxim Zvyagin | 2022 | 0.0038 | 1 |

## 7.4 Evolution of Topics Over Time

```
# Get top topics overall
all_topics <- papers_df %>%
  count(subtopic, sort = TRUE) %>%
  head(10)

top_topics <- all_topics$subtopic

# Count by year for each top topic
topic_timeline <- papers_df %>%
  filter(subtopic %in% top_topics) %>%
  count(year, subtopic) %>%
  filter(year >= 2015 & year <= 2025)

# Plot evolution
ggplot(topic_timeline, aes(x = year, y = n, color = subtopic, group = subtopic)) +
  geom_line(size = 1.2) +
  geom_point(size = 2) +
  labs(title = "Evolution of Top Research Subtopics Over Time",
       x = "Year", y = "Number of Papers",
       color = "Subtopic") +
  theme_minimal() +
  theme(legend.position = "bottom")
```




## 7.5 Emerging Communities (Recent Papers)

```
# Identify communities dominated by recent papers
recent_paper_ids <- papers_df %>%
  filter(year >= 2020 & year <= 2025) %>%
  pull(local_id)

community_recency <- data.frame()

for (comm_id in unique(V(connected_graph)$community)) {
  comm_paper_ids <- V(connected_graph)[V(connected_graph)$community == comm_id]$name

  recent_count <- sum(comm_paper_ids %in% recent_paper_ids)
  total_count <- length(comm_paper_ids)
  recent_ratio <- recent_count / total_count

  community_recency <- rbind(community_recency, data.frame(
    community = comm_id,
    total_papers = total_count,
    recent_papers = recent_count,
    recent_ratio = recent_ratio
  ))
}

# Communities with high proportion of recent papers (emerging topics)
emerging_communities <- community_recency %>%
  filter(total_papers >= 10) %>%  # Only consider sizeable communities
  arrange(desc(recent_ratio)) %>%
```

```
  head(10)

kable(emerging_communities, digits = 3,
      caption = "Emerging Communities (High Proportion of Recent Papers)")
```

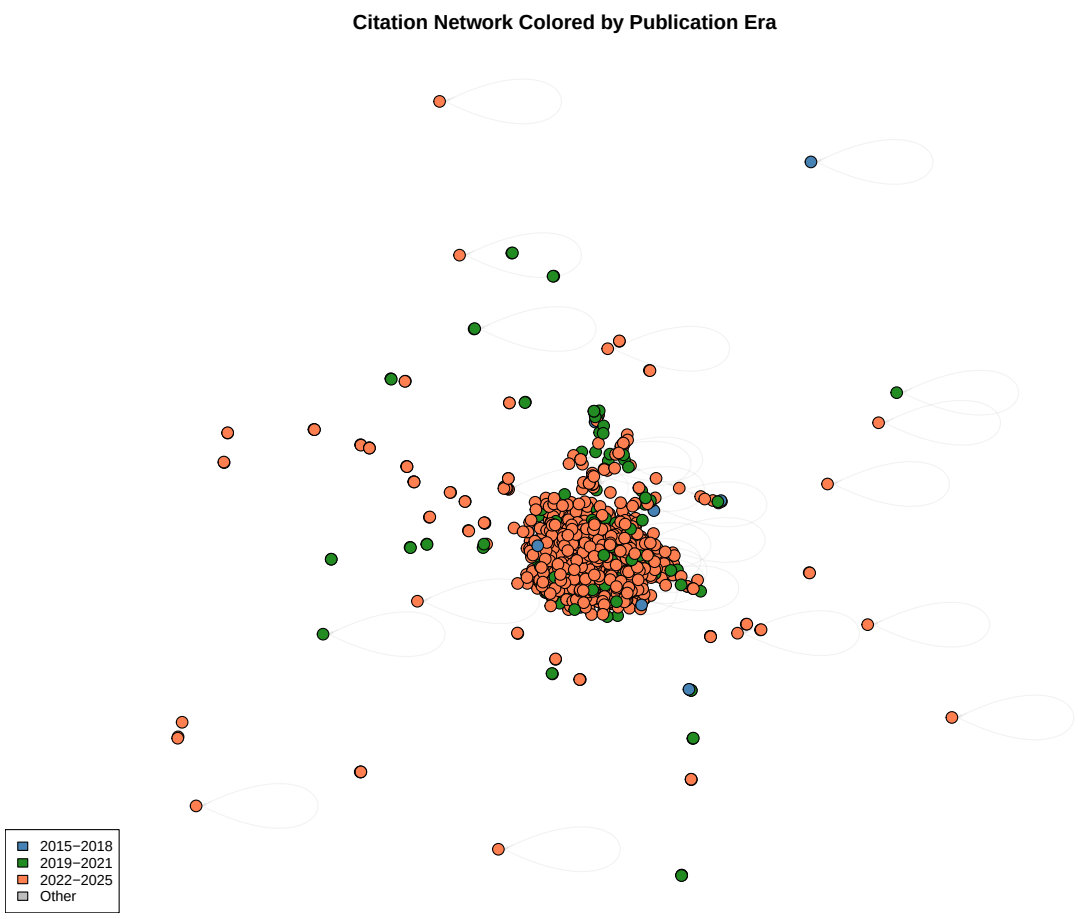
Table 12: Emerging Communities (High Proportion of Recent Papers)

| community | total_papers | recent_papers | recent_ratio |
|---|---|---|---|
| 1 | 251 | 251 | 1.000 |
| 26 | 19 | 19 | 1.000 |
| 11 | 133 | 131 | 0.985 |
| 19 | 59 | 58 | 0.983 |
| 5 | 155 | 152 | 0.981 |
| 2 | 85 | 83 | 0.976 |
| 14 | 51 | 49 | 0.961 |
| 17 | 43 | 41 | 0.953 |
| 7 | 42 | 40 | 0.952 |
| 10 | 135 | 128 | 0.948 |

## 7.6 Network Visualization by Publication Year

```
# Add year to vertices
vertex_years <- papers_df %>%
  select(local_id, year) %>%
  filter(local_id %in% V(connected_graph)$name)

V(connected_graph)$year <- vertex_years$year[match(V(connected_graph)$name, vertex_years$local_id)]

# Color by era
V(connected_graph)$era <- case_when(
  V(connected_graph)$year >= 2015 & V(connected_graph)$year <= 2018 ~ 1,
  V(connected_graph)$year >= 2019 & V(connected_graph)$year <= 2021 ~ 2,
  V(connected_graph)$year >= 2022 & V(connected_graph)$year <= 2025 ~ 3,
  TRUE ~ 4
)

era_colors <- c("steelblue", "forestgreen", "coral", "gray")

# Plot network colored by time period
plot(connected_graph,
     vertex.color = era_colors[V(connected_graph)$era],
     vertex.size = 3,
     vertex.label = NA,
     edge.arrow.size = 0.2,
     edge.color = alpha("gray", 0.2),
     layout = layout_fr,
     main = "Citation Network Colored by Publication Era")

legend("bottomleft",
```

```
       legend = c("2015-2018", "2019-2021", "2022-2025", "Other"),
       fill = era_colors,
       cex = 0.8)
```

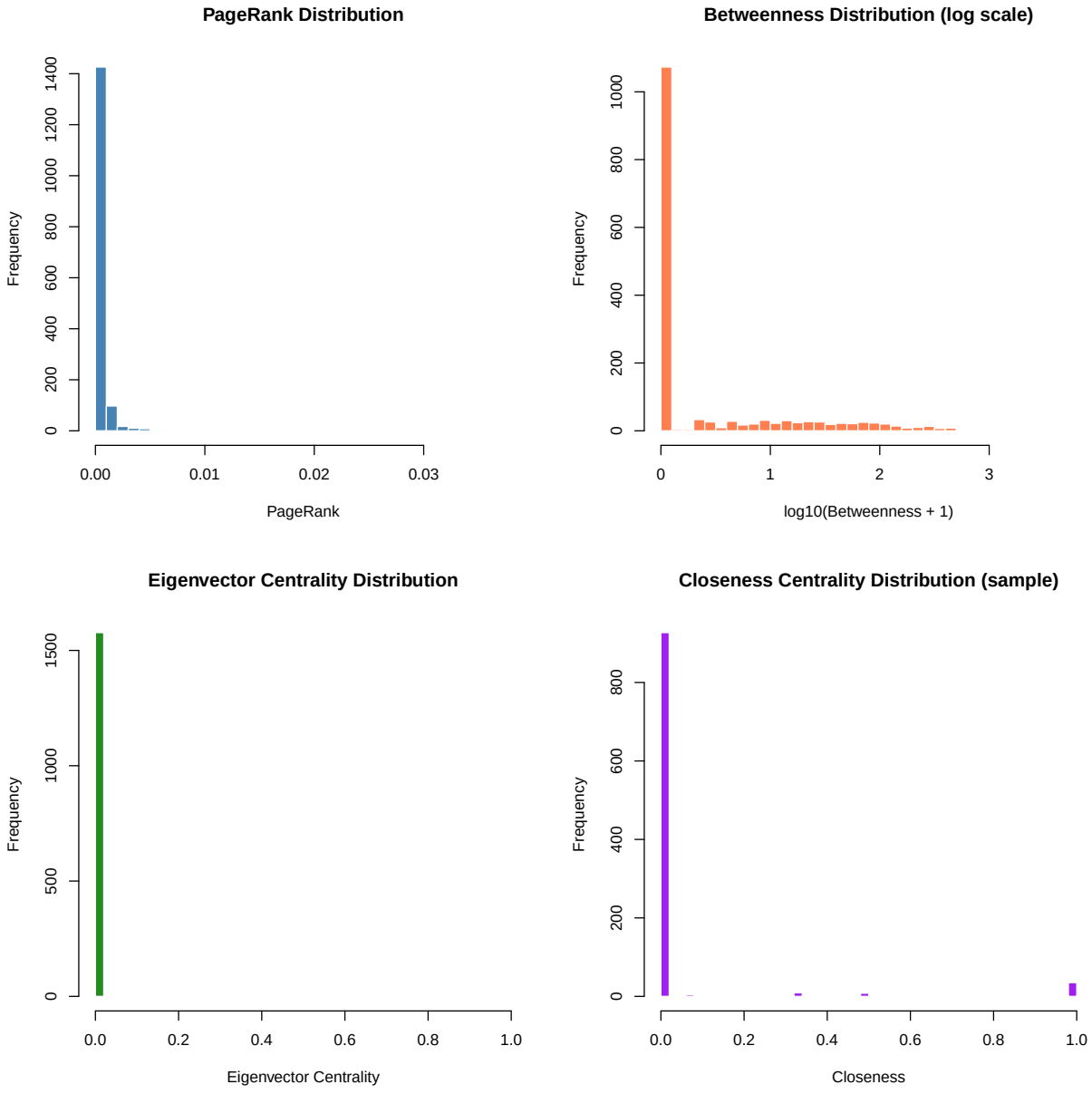



# 8 Advanced Network Analysis

## 8.1 Centrality Distributions

```
par(mfrow = c(2, 2))

# PageRank distribution
hist(centrality_df$pagerank, breaks = 50,
     main = "PageRank Distribution", xlab = "PageRank",
     col = "steelblue", border = "white")

# Betweenness distribution
hist(log10(centrality_df$betweenness + 1), breaks = 50,
```

```
     main = "Betweenness Distribution (log scale)", xlab = "log10(Betweenness + 1)",
     col = "coral", border = "white")

# Eigenvector centrality
hist(centrality_df$eigenvector, breaks = 50,
     main = "Eigenvector Centrality Distribution", xlab = "Eigenvector Centrality",
     col = "forestgreen", border = "white")

# Closeness centrality (sample for speed)
sample_nodes <- sample(V(connected_graph), min(1000, vcount(connected_graph)))
closeness_sample <- closeness(connected_graph, vids = sample_nodes, mode = "all")
hist(closeness_sample, breaks = 50,
     main = "Closeness Centrality Distribution (sample)", xlab = "Closeness",
     col = "purple", border = "white")
```




```r
par(mfrow = c(1, 1))
```

## 8.2 K-Core Decomposition

```r
# K-core decomposition
kcore_values <- coreness(connected_graph, mode = "all")
V(connected_graph)$kcore <- kcore_values

kcore_summary <- data.frame(
  kcore = sort(unique(kcore_values), decreasing = TRUE)
) %>%
```

```
  rowwise() %>%
  mutate(num_nodes = sum(kcore_values >= kcore))

kable(head(kcore_summary, 15), caption = "K-Core Decomposition Summary")
```

Table 13: K-Core Decomposition Summary

| kcore | num_nodes |
|---|---|
| 6 | 35 |
| 5 | 230 |
| 4 | 459 |
| 3 | 724 |
| 2 | 1092 |
| 1 | 1582 |

```
# Plot k-core distribution
ggplot(data.frame(kcore = kcore_values), aes(x = kcore)) +
  geom_histogram(binwidth = 1, fill = "steelblue", color = "white") +
  labs(title = "K-Core Distribution",
       x = "K-Core Value", y = "Number of Nodes") +
  theme_minimal()
```