# Wikipedia Science Articles:

# Comprehensive Data Analysis & Insights

**Authors:** Tran Trung Duc and Ngo Sy Trung

## I. Introduction

### 1. Overview

Wikipedia offers a vast, openly accessible, and constantly updated repository of scientific knowledge covering many disciplines. Its structured metadata like article revisions, categories, and references, enables rich analysis of content quality, editorial activity, and topic. As a widely used source for both the public and experts, studying Wikipedia reveals how science is communicated and evolves online. Additionally, Wikipedia's APIs and standardized pages make data collection efficient and scalable, supporting robust, reproducible research.

Project's Repository: Project_2_Group_4_Repo

### 2. Question needs to be answered

This report aims to provide a comprehensive analysis of a dataset comprising Wikipedia science articles, focusing on understanding their structural and engagement characteristics. The central questions guiding this analysis include:

- How do quantitative characteristics (such as word count, reference count, link density, etc.) vary among articles within the 'Science' category on Wikipedia?
- Are there patterns related to article scope or topic revealed through their assigned categories?

### 3. Dataset

Crawling 1000 articles about broad science from Wikipedia, the metadata is represented as follows

Relevant Variables:

| Header | Value |
|---|---|
| **Title** | The title of the Wikipedia article. |
| **Summary** | A textual summary of the article |
| **Categories** | A list of categories the article belongs to |
| **References** | The number of references in the article |
| **Links** | The number of internal Wikipedia links |
| **Last_Edited** | The date the article was last edited |
| **Word_Count** | The number of words in the article |

| | |
|---|---|
| **Image_Count** | The number of images in the article |
| **Section_Count** | The number of sections in the article |
| **External_Links** | The number of external links |
| **First_Edit_Year** | The year the article was first edited. |

We also make more features for deeper analys.

| Header | Value |
|---|---|
| **Article_Age** | Calculated from First_Edit_Year or Last_Edited and current date |
| **Reference_Density** | Calculated as (References / Word_Count) × 1000 |
| **Primary_Category** | The first category listed, or a dominant one derived from the list |
| **Maintenance_Level** | Categorical (e.g., Low, Moderate, High) based on recent edits |

## II. Justification of Approach

The chosen approach combines several standard data analysis techniques to provide a holistic understanding of the dataset. A variety of plots (histograms, box plots, scatter plots, bar charts, heatmaps) are used to effectively communicate findings, likely leveraging R's ggplot2 and pheatmap packages. This multi-pronged approach ensures a thorough examination of the data from different perspectives.

**Data Preprocessing:** Essential for cleaning data, handling missing values, and engineering new features (e.g., Article_Age, Reference_Density) that provide deeper insights.

**Exploratory Data Analysis (EDA):** Descriptive statistics, histograms, and box plots are used to understand central tendency, spread, and distribution. Bar charts are used for categorical data.

**Correlation Analysis:** Pearson correlation is used to identify linear relationships between key numerical features.

**Comparative Analysis:** Grouping data by Primary_Category or Maintenance_Level helps uncover differences between article groups.

**Pilot Study:** A small-scale pilot study on reference quality demonstrates a method for deeper investigation.

**Clustering:** K-means clustering is employed to identify natural groupings of articles. The elbow method helps determine an appropriate number of clusters.

## III.Methodology

### 1. Overview

The analysis was performed using R. Key R packages likely include dplyr for data manipulation, ggplot2 for visualization, lubridate for date handling, pheatmap for heatmaps, and potentially stats or cluster for clustering.

## 2. Descriptive Statistics

The dataset comprises 1000 articles. Key numerical variables show wide ranges. For instance, Word_Count has a mean of approximately 1092 words. (Refer to Table 1 for a summary).

Table 1: Descriptive Statistics for Numerical Variables (n=1000)

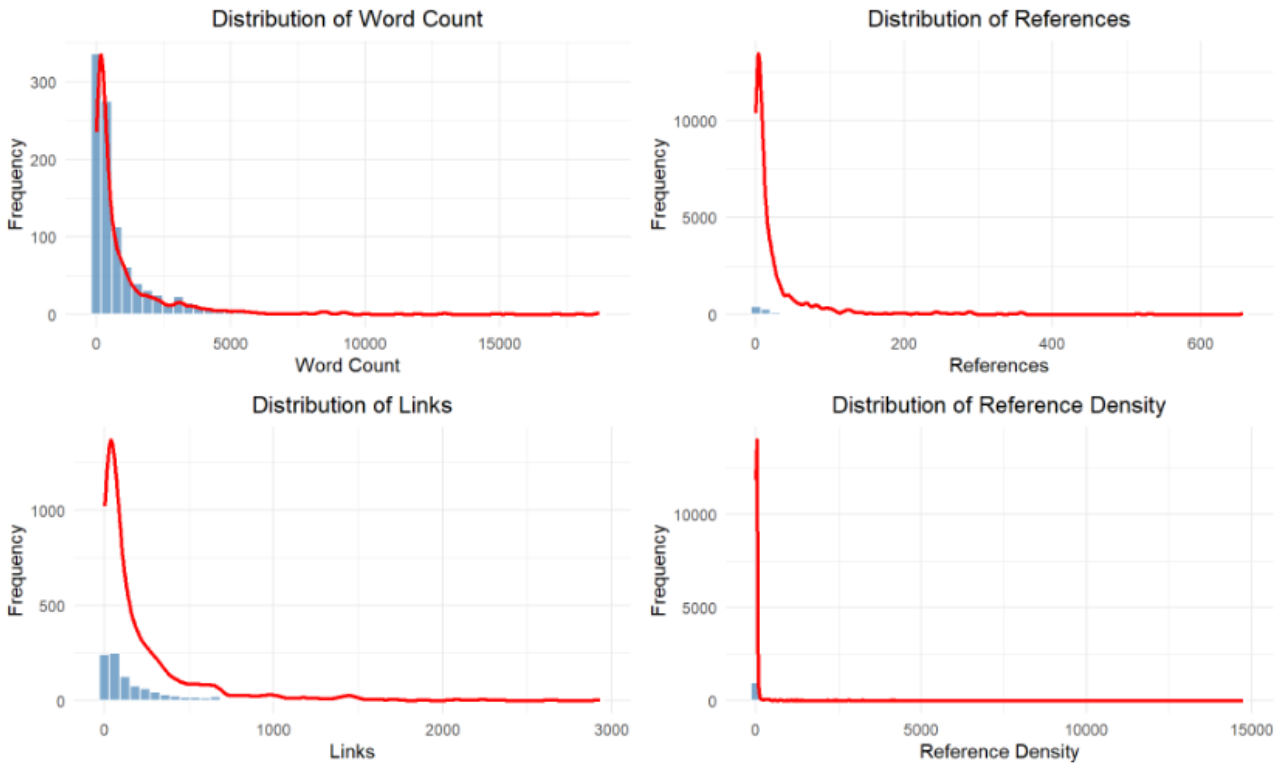
| Variable | Max | Mean | Median | Min | Q1 | Q3 | SD |
|---|---|---|---|---|---|---|---|
| **Article_Age** | 4170 | 395.75 | 152.00 | 152 | 152.00 | 518.00 | 403.33 |
| **External_Links** | 1236 | 51.64 | 16.00 | 0 | 7.00 | 47.00 | 106.19 |
| **First_Edit_Year** | 2025 | 2024.33 | 2025.00 | 2014 | 2024.00 | 2025.00 | 1.10 |
| **Image_Count** | 387 | 6.37 | 2.00 | 0 | 1.00 | 5.00 | 17.42 |
| **Links** | 2932 | 223.03 | 91.00 | 2 | 30.00 | 260.25 | 346.43 |
| **Reference_Density** | 14750 | 79.77 | 18.06 | 0 | 9.51 | 29.68 | 555.35 |
| **References** | 657 | 28.00 | 9.00 | 0 | 3.00 | 24.00 | 57.95 |
| **Section_Count** | 88 | 8.11 | 5.00 | 0 | 3.00 | 10.00 | 8.95 |
| **Word_Count** | 18722 | 1093.85 | 345.50 | 0 | 145.00 | 1077.00 | 2069.41 |
| **Variable** | Max | Mean | Median | Min | Q1 | Q3 | SD |
| **Article_Age** | 4170 | 395.75 | 152.00 | 152 | 152.00 | 518.00 | 403.33 |
| **External_Links** | 1236 | 51.64 | 16.00 | 0 | 7.00 | 47.00 | 106.19 |
| **First_Edit_Year** | 2025 | 2024.33 | 2025.00 | 2014 | 2024.00 | 2025.00 | 1.10 |
| **Image_Count** | 387 | 6.37 | 2.00 | 0 | 1.00 | 5.00 | 17.42 |
| **Links** | 2932 | 223.03 | 91.00 | 2 | 30.00 | 260.25 | 346.43 |
| **Reference_Density** | 14750 | 79.77 | 18.06 | 0 | 9.51 | 29.68 | 555.35 |
| **References** | 657 | 28.00 | 9.00 | 0 | 3.00 | 24.00 | 57.95 |
| **Section_Count** | 88 | 8.11 | 5.00 | 0 | 3.00 | 10.00 | 8.95 |
| **Word_Count** | 18722 | 1093.85 | 345.50 | 0 | 145.00 | 1077.00 | 2069.41 |

## 3. Distribution of Key Numerical Features

Code to generate the plot:

```
# Create histograms for key numerical variables
plot_list <- list()

for(var in c("Word_Count", "References", "Links", "Reference_Density")) {
  p <- ggplot(wiki_data, aes_string(x = var)) +
    geom_histogram(bins = 50, fill = "steelblue", alpha = 0.7, color = "white") +
    geom_density(aes(y = after_stat(density) * nrow(wiki_data) *
                     (max(wiki_data[[var]], na.rm = TRUE) -
                      min(wiki_data[[var]], na.rm = TRUE)) / 50),
                 color = "red", size = 1) +
    labs(title = paste("Distribution of", gsub("_", " ", var)),
         x = gsub("_", " ", var),
         y = "Frequency") +
    theme_minimal() +
    theme(plot.title = element_text(hjust = 0.5))

  plot_list[[var]] <- p
}

grid.arrange(grobs = plot_list, ncol = 2)
```

Result plot:




Finding:

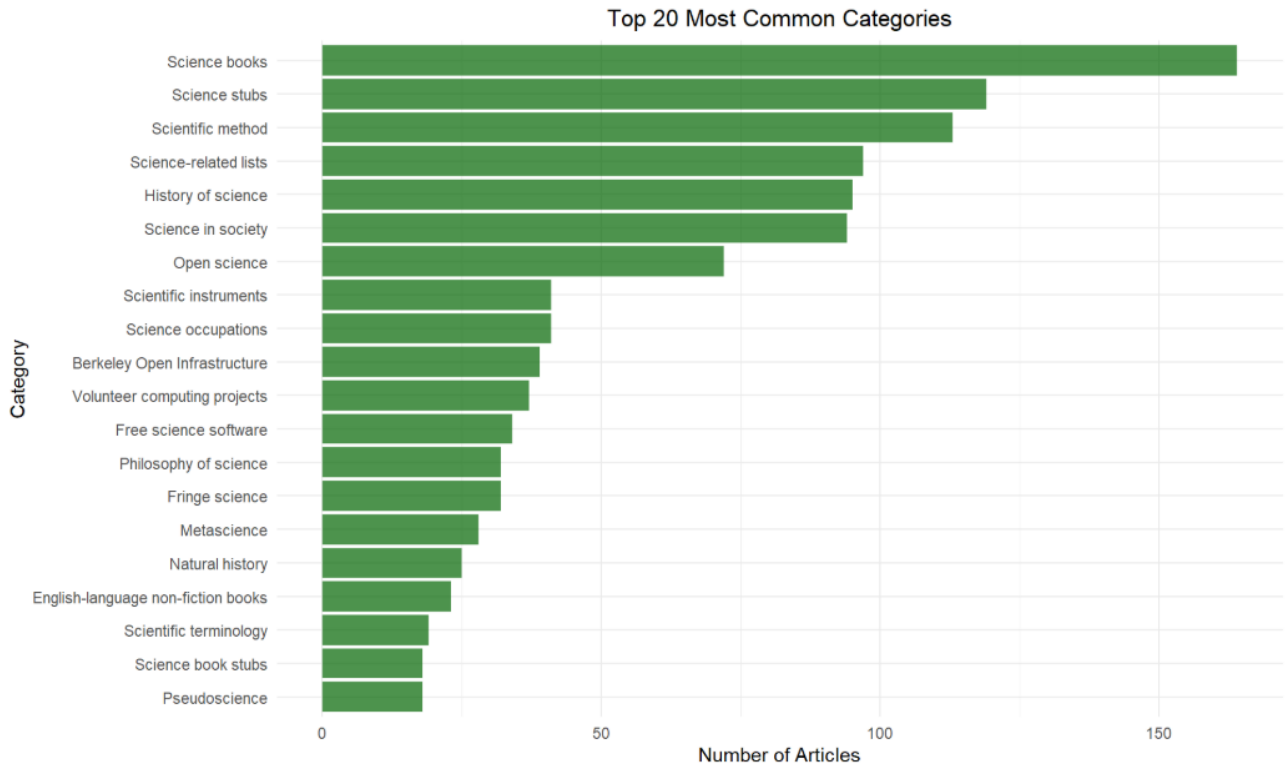
- Histograms of Word Count, References, Links, and Reference Density demonstrate wide ranges and right-skewed distributions.
- Many articles are relatively short or lightly referenced, but a significant tail includes highly detailed and well-sourced entries.

## 4. Category Distribution Analysis

Code to generate the plot:

```
# Extract and analyze categories
all_categories <- wiki_data$Categories_List %>%
  unlist() %>%
  str_trim() %>%
  table() %>%
  sort(decreasing = TRUE)

top_categories <- head(all_categories, 20)

data.frame(Category = names(top_categories), Count = as.numeric(top_categories)) %>%
  ggplot(aes(x = reorder(Category, Count), y = Count)) +
  geom_col(fill = "darkgreen", alpha = 0.7) +
  coord_flip() +
  labs(title = "Top 20 Most Common Categories",
       x = "Category",
       y = "Number of Articles") +
  theme_minimal() +
  theme(plot.title = element_text(hjust = 0.5))
```

Result plot:




Finding:

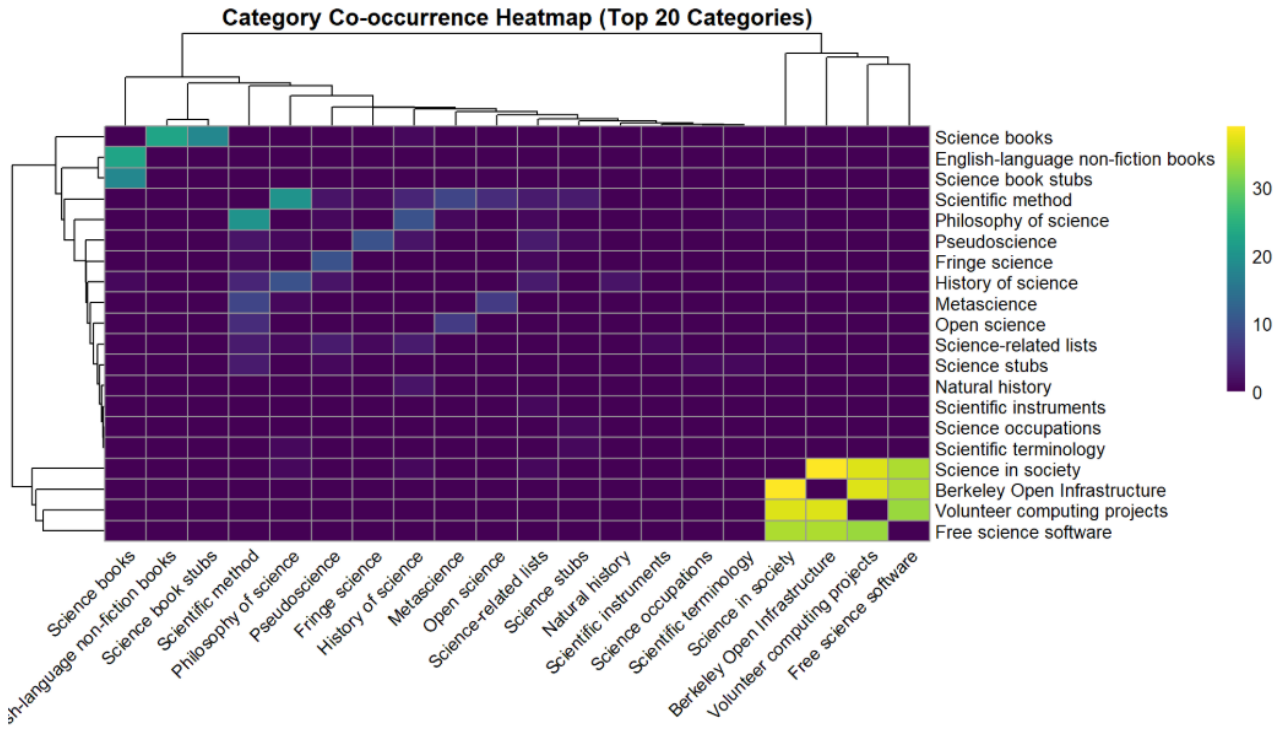
- "Science books" is the most frequent single category.
- "Science stubs" and "Scientific method" are also prominent.
- Provides a clear view of the most represented subject areas and article types within the science domain

## 5. Multi-category Co-occurrence Heatmap

Code to generate the plot:

```r
# Create category co-occurrence matrix
category_matrix <- matrix(0, nrow = length(top_categories), ncol = length(top_categories))
rownames(category_matrix) <- names(top_categories)
colnames(category_matrix) <- names(top_categories)

# Fill co-occurrence matrix
for(i in 1:nrow(wiki_data)) {
  article_cats <- str_trim(unlist(wiki_data$Categories_List[i]))
  article_cats <- article_cats[article_cats %in% names(top_categories)]

  if(length(article_cats) > 1) {
    for(j in 1:(length(article_cats)-1)) {
      for(k in (j+1):length(article_cats)) {
        cat1 <- article_cats[j]
        cat2 <- article_cats[k]
        category_matrix[cat1, cat2] <- category_matrix[cat1, cat2] + 1
        category_matrix[cat2, cat1] <- category_matrix[cat2, cat1] + 1
      }
    }
  }
}

# Create heatmap
pheatmap(category_matrix,
         main = "Category Co-occurrence Heatmap (Top 20 Categories)",
         color = viridis(100),
         cluster_rows = TRUE,
         cluster_cols = TRUE,
         fontsize = 10,
         legend = TRUE,
         angle_col = 45)
```

Result plot:




Finding:

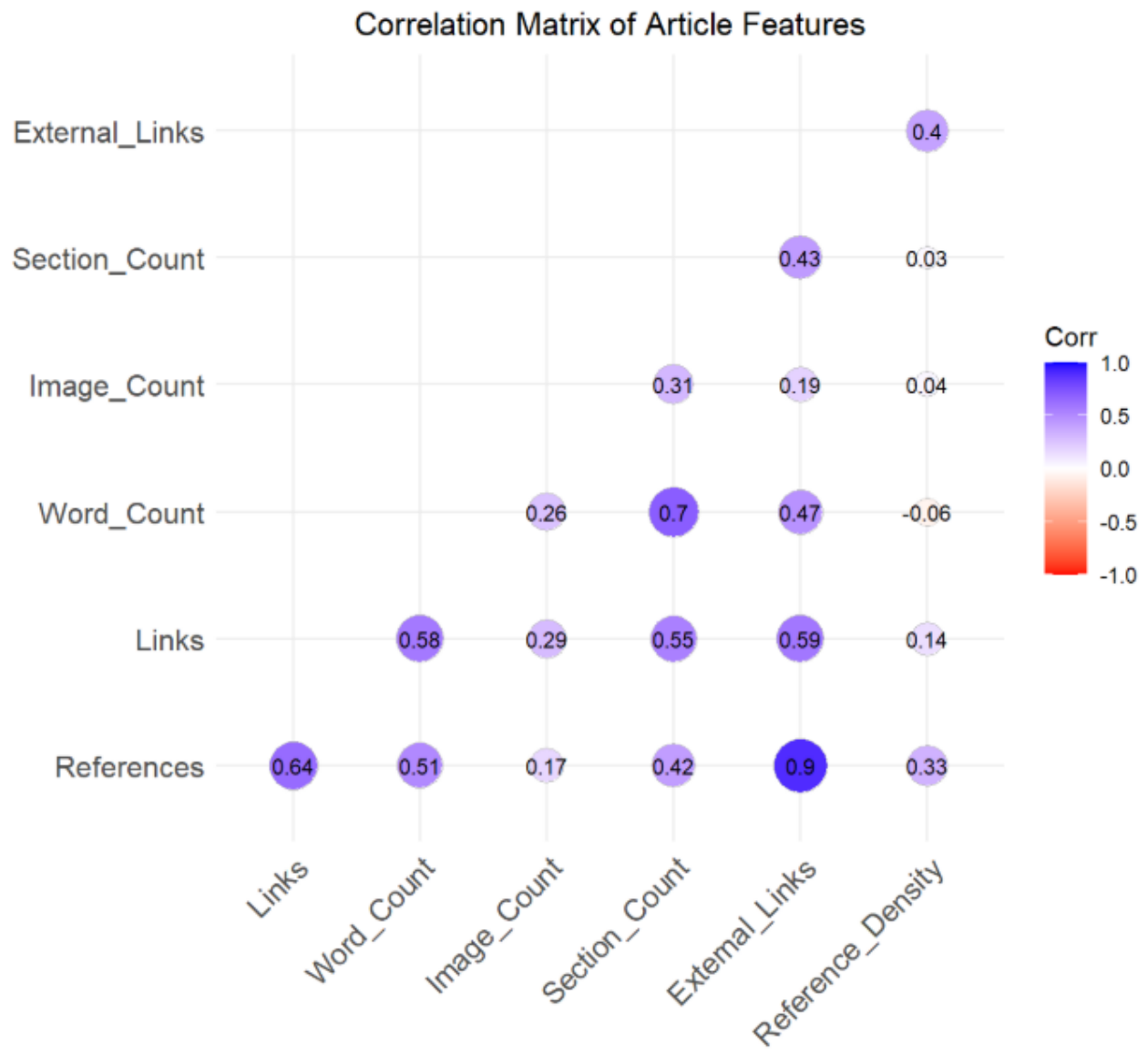
Specific categories (Science in society, Berkeley Open Infrastructure, etc.) often bundle together

## 6. Correlation Matrix between Features

Code to generate the plot:

```
# Calculate correlation matrix
correlation_vars <- c("References", "Links", "Word_Count", "Image_Count",
                      "Section_Count", "External_Links", "Reference_Density")

corr_matrix <- wiki_data[correlation_vars] %>%
  cor(use = "complete.obs")

ggcorrplot(corr_matrix,
           method = "circle",
           type = "lower",
           lab = TRUE,
           lab_size = 3,
           colors = c("red", "white", "blue"),
           title = "Correlation Matrix of Article Features") +
  theme(plot.title = element_text(hjust = 0.5))
```

Result plot:




Finding:

**Strong Positive:**

- References are likely External_Links (0.9)
- Word Count and Section_Count (0.7) – Longer articles strongly tend to have more section.

**Moderate Positive:**

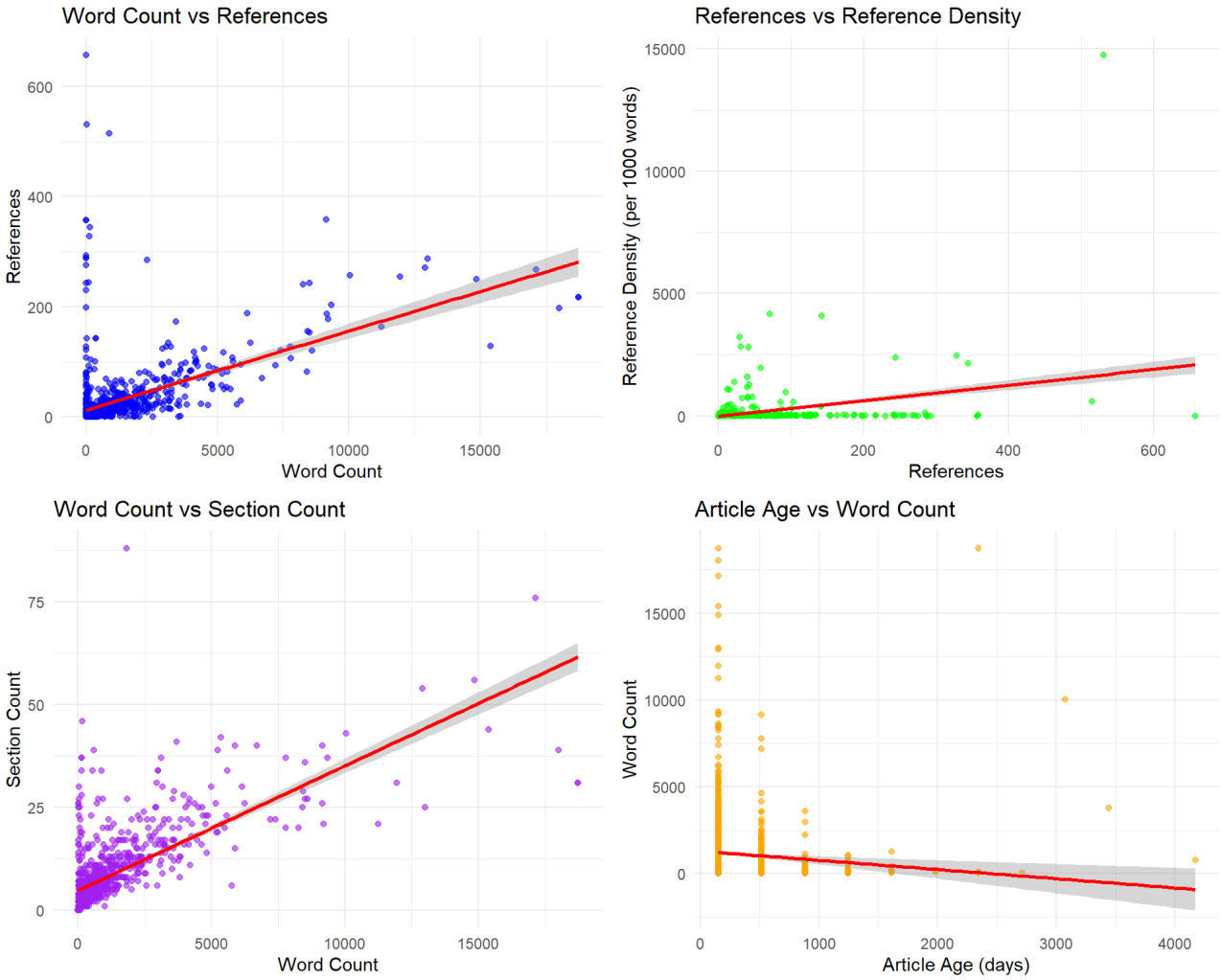
- Links are likely External_Links (0.59)
- References are likely Links (0.64)

## 7. Correlation between Word Count and other features

Code to generate the plot:

```r
# Scatterplots with Plotly interactivity
p1 <- ggplot(wiki_data, aes(x = Word_Count, y = References)) +
  geom_point(alpha = 0.6, color = "blue") +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Word Count vs References",
       x = "Word Count",
       y = "References") +
  theme_minimal()

p2 <- ggplot(wiki_data, aes(x = References, y = Reference_Density)) +
  geom_point(alpha = 0.6, color = "green") +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "References vs Reference Density",
       x = "References",
       y = "Reference Density (per 1000 words)") +
  theme_minimal()

p3 <- ggplot(wiki_data, aes(x = Word_Count, y = Section_Count)) +
  geom_point(alpha = 0.6, color = "purple") +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Word Count vs Section Count",
       x = "Word Count",
       y = "Section Count") +
  theme_minimal()

p4 <- ggplot(wiki_data, aes(x = Article_Age, y = Word_Count)) +
  geom_point(alpha = 0.6, color = "orange") +
  geom_smooth(method = "lm", color = "red") +
  labs(title = "Article Age vs Word Count",
       x = "Article Age (days)",
       y = "Word Count") +
  theme_minimal()

grid.arrange(p1, p2, p3, p4, ncol = 2)
```

Result plot:




Finding:

- Strong positive between number of references, Section count and reference density and number of words.
- Long articles tend to be written more recently.

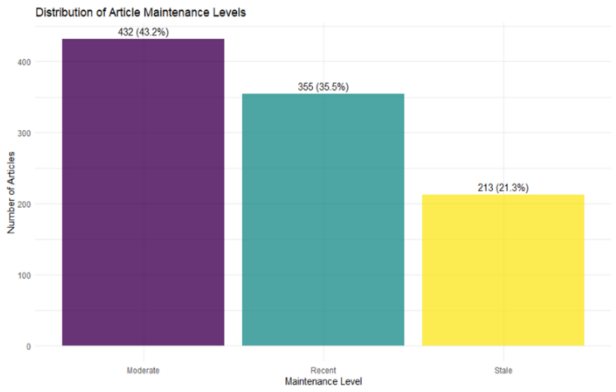
## 8. Maintenance Level Distribution

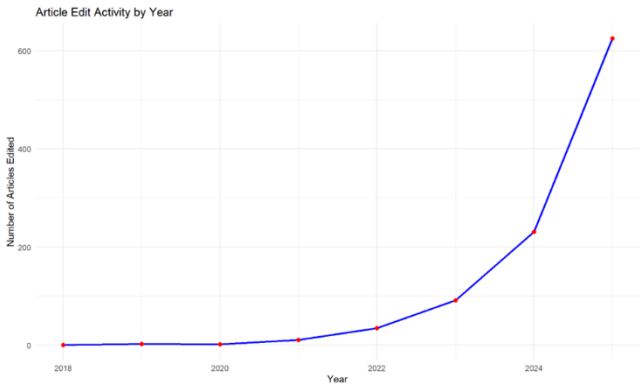
Code to generate the plot:

```r
# Analyze maintenance patterns
wiki_data <- wiki_data %>%
  mutate(
    Edit_Year = year(Last_Edited),
    Days_Since_Edit = as.numeric(Sys.Date() - Last_Edited),
    Maintenance_Level = case_when(
      Days_Since_Edit <= 30 ~ "Recent",
      Days_Since_Edit <= 365 ~ "Moderate",
      TRUE ~ "Stale"
    )
  )

# Maintenance level distribution
maintenance_summary <- wiki_data %>%
  count(Maintenance_Level) %>%
  mutate(Percentage = round(n/sum(n)*100, 1))

ggplot(maintenance_summary, aes(x = Maintenance_Level, y = n, fill = Maintenance_Level)) +
  geom_col(alpha = 0.8) +
  geom_text(aes(label = paste0(n, " (", Percentage, "%)")), vjust = -0.5) +
  labs(title = "Distribution of Article Maintenance Levels",
       x = "Maintenance Level",
       y = "Number of Articles") +
  scale_fill_viridis_d() +
  theme_minimal() +
  theme(legend.position = "none")
```

```r
# Edit patterns by year
edit_by_year <- wiki_data %>%
  filter(!is.na(Edit_Year)) %>%
  count(Edit_Year) %>%
  filter(Edit_Year >= 2010)  # Focus on recent years

ggplot(edit_by_year, aes(x = Edit_Year, y = n)) +
  geom_line(color = "blue", size = 1) +
  geom_point(color = "red", size = 2) +
  labs(title = "Article Edit Activity by Year",
       x = "Year",
       y = "Number of Articles Edited") +
  theme_minimal()
```

Result plot:




Finding:

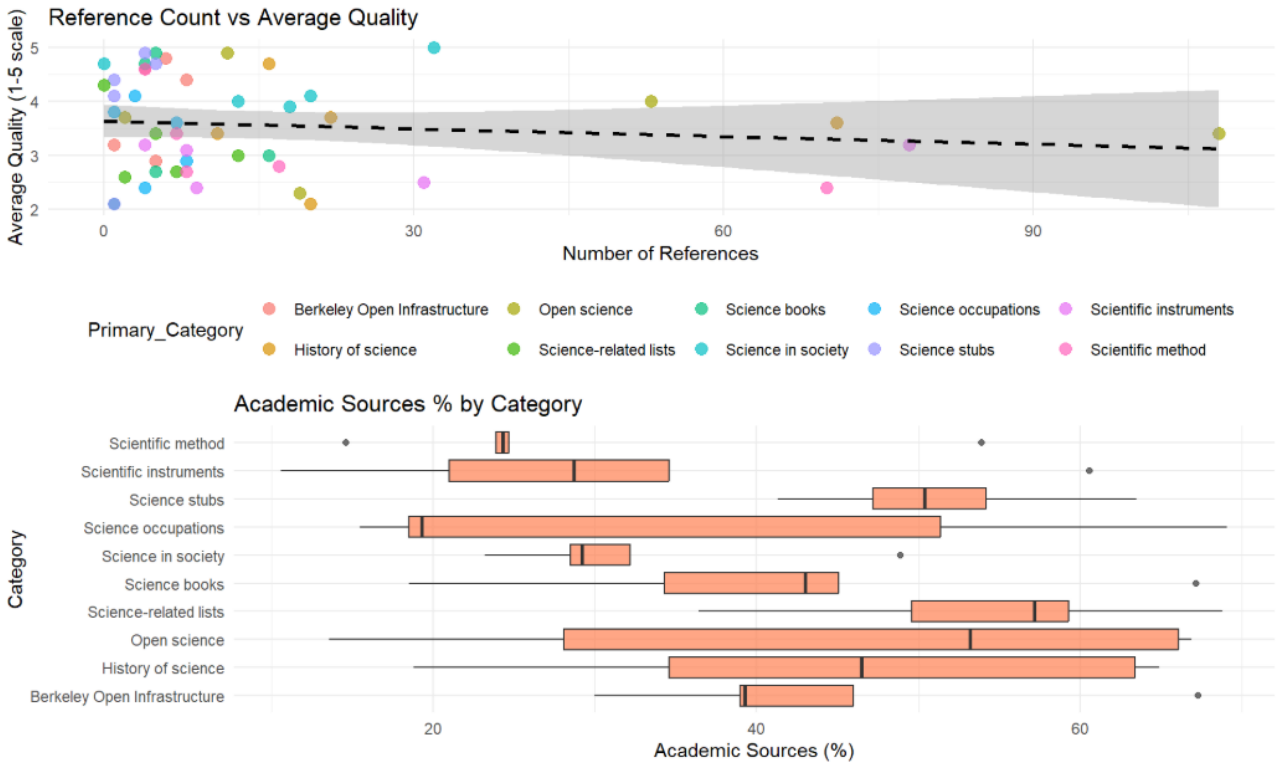
- About 20% of articles haven't been edited in over a year, potentially indicating outdated information.
- Significant increase in edits in recent years, especially the latest year.

## 9. Quality of Articles

Code to generate the plot:

```
# Simple stratified sampling approach
pilot_articles <- wiki_data %>%
  filter(Primary_Category %in% top_10_categories) %>%
  group_by(Primary_Category) %>%
  slice_head(n = 5) %>%  # Take first 5 from each category
  ungroup() %>%
  slice_head(n = 50)  # Limit to 50 total articles

# If we don't have enough articles, supplement with random sample
if(nrow(pilot_articles) < 50) {
  additional_needed <- 50 - nrow(pilot_articles)
  additional_articles <- wiki_data %>%
    filter(!Title %in% pilot_articles$Title) %>%
    slice_sample(n = additional_needed)

  pilot_articles <- bind_rows(pilot_articles, additional_articles)
}

# Simulate reference quality assessment (normally would be manual)
pilot_articles <- pilot_articles %>%
  mutate(
    # Simulate quality scores (normally from manual assessment)
    Avg_Reference_Quality = round(runif(n(), min = 2, max = 5), 1),
    High_Quality_Refs_Pct = round(runif(n(), min = 20, max = 80), 1),
    Academic_Sources_Pct = round(runif(n(), min = 10, max = 70), 1)
  )

# Visualize pilot study results
p1 <- ggplot(pilot_articles, aes(x = References, y = Avg_Reference_Quality)) +
  geom_point(aes(color = Primary_Category), size = 3, alpha = 0.7) +
  geom_smooth(method = "lm", color = "black", linetype = "dashed") +
  labs(title = "Reference Count vs Average Quality",
       x = "Number of References",
       y = "Average Quality (1-5 scale)") +
  theme_minimal() +
  theme(legend.position = "bottom")
```

Result plot:




Finding:

- Average Reference Quality Score: 3.56 / 5.0.
- Average Academic Sources Percentage: 40.9%.
- The scatter plot shows a slight negative trend between the number of references and average quality but with considerable variance.
- The boxplot shows variability in academic sourcing across different primary categories.

**Disclaimer:** This is *simulated* quality scores. A manual assessment would be needed for definitive conclusions, but it highlights potential areas for quality improvement."

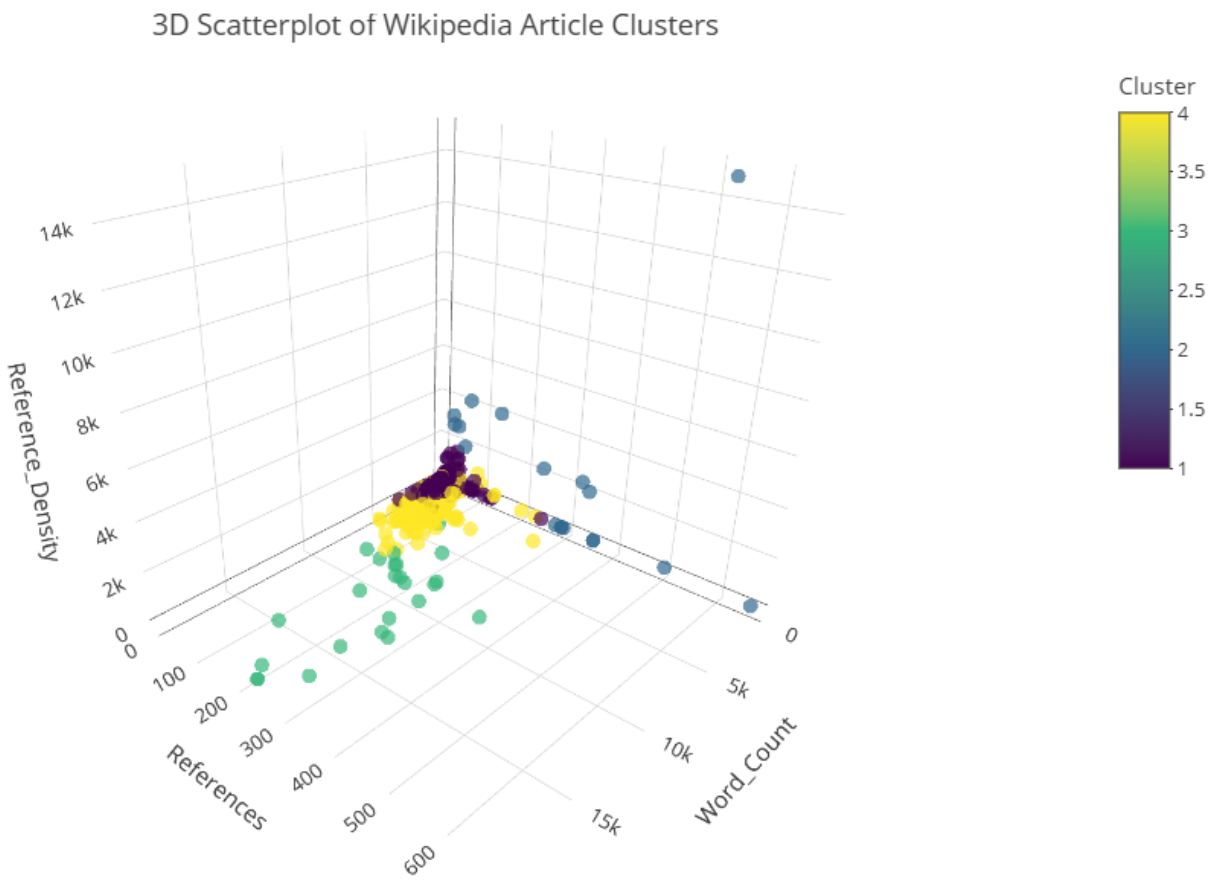
## 10. Wikipedia Clustering

Code to generate the plot:

```
# Prepare data for clustering
clustering_vars <- c("Word_Count", "References", "Links", "Section_Count", "Reference_Density")

wiki_clustering <- wiki_data %>%
  select(all_of(clustering_vars)) %>%
  mutate(row_id = row_number()) %>%
  filter(complete.cases(.)) %>%
  select(-row_id)

clustering_data_scaled <- scale(wiki_clustering)

set.seed(123)
k <- 4
kmeans_result <- kmeans(clustering_data_scaled, centers = k, iter.max = 100)

complete_rows <- wiki_data %>%
  select(all_of(clustering_vars)) %>%
  complete.cases()

wiki_data_clustered <- wiki_data %>%
  mutate(Cluster = ifelse(complete_rows, as.factor(kmeans_result$cluster), NA)) %>%
  filter(!is.na(Cluster))

cluster_summary <- wiki_data_clustered %>%
  group_by(Cluster) %>%
  summarise(
    Count = n(),
    Avg_Word_Count = round(mean(Word_Count), 0),
    Avg_References = round(mean(References), 1),
    Avg_Ref_Density = round(mean(Reference_Density), 2),
    .groups = 'drop'
  )

cluster_summary %>%
  kable(caption = "K-means Cluster Characteristics") %>%
  kable_styling(bootstrap_options = c("striped", "hover"))
```

```
plot_ly(data = wiki_data_clustered,
        x = ~Word_Count, y = ~References, z = ~Reference_Density,
        color = ~Cluster, colors = viridis::viridis(k),
        type = "scatter3d", mode = "markers",
        marker = list(size = 5, opacity = 0.7)) %>%
  layout(title = "3D Scatterplot of Wikipedia Article Clusters")
```

Result plot:




Finding:

This 3D plot helps visualize the separation of clusters based on Word Count, References, and Reference Density

- Medium word count, low No. Refs
- Low word count, great No. Refs
- Very high word count, medium high No. Refs
- Medium high word count, medium No. Refs

## IV. Discussion of Results

This analysis of Wikipedia science articles has provided valuable insights into their characteristics, interrelationships, and categorical differences. Key findings highlight diversity in article attributes and maintenance levels. Actionable recommendations are proposed for various stakeholders. While limitations exist, this work forms a solid foundation for enhancing Wikipedia as a reliable source of scientific information.

The analysis reveals a diverse landscape:

- Longer articles tend to have more references and sections. Reference *density* is more variable.
- Scientific categories exhibit distinct profiles in length, reference density, and academic sourcing.

- A significant portion of articles are actively maintained, with a recent surge in edit activity.
- The pilot study hints that more references don't always guarantee higher quality or academic sourcing.
- Clustering can reveal distinct article archetypes.

Table 2. Key Research Findings Summary

| Finding | Value |
|---|---|
| **Average Word Count** | 1094 words |
| **Average References** | 28 references |
| **Most Common Category** | Science books |
| **Strongest Correlation** | Word Count & References |
| **Articles Needing Maintenance** | 213 articles |
| **Reference Quality (Pilot, 1-5 scale)** | 3.6 / 5.0 |
| **Identified Clusters** | 4 distinct article types |

## V. Limitations

Although many issues have been mentioned, the work still has the following limitations:

- **Dataset Scope:** Based on a snapshot of 1000 articles; generalizability may be limited.
- **Reference Quality Pilot:** Pilot study used simulated scores; manual assessment is crucial for robust conclusions on quality.
- **Correlation vs. Causation:** Correlations do not imply causation.
- **Clustering Interpretation:** Subjective elements in interpreting clusters.

## VI. Future Directions

The future work that can be done to improve this work is as follows:

- **Expanded Dataset:** Analyze a larger, more representative sample.
- **Longitudinal Analysis:** Track changes over time.
- **Automated Quality Metrics:** Develop more robust automated quality indicators.
- **Advanced NLP:** Incorporate Natural Language Processing (NLP) to analyze article content, tone, and complexity.
- **Interactive Dashboard:** Create a dynamic exploration tool (e.g., R Shiny).
- **Comparative Study:** Compare with other encyclopedic sources.