groupact1.2

Jessamine Paula Orada

2025-12-01

{r setup, include=FALSE} knitr::opts_chunk$set(echo = TRUE)

“‘{r} library(rvest)
library(dplyr)
library(stringr)
library(lubridate) library(ggplot2)
library(knitr)
library(kableExtra)

Creating Objects for Data Storage (Optional, included for completeness)

titles <- character(0) authors <- character(0) submission_dates <- character(0)

```{r}
# URL for astro-ph.SR recent papers
base_url <- "https://arxiv.org/list/astro-ph.SR/recent?skip="

all_papers <- list()

# Loop to get exactly 200 papers (4 batches of 50: skip=0, 50, 100, 150)
starts <- seq(from = 0, to = 150, by = 50)

for (i in starts) {

  # Construct URL
  url <- paste0(base_url, i, "&show=50")
  print(paste("Scraping:", url))

  tryCatch({
    page <- read_html(url)

    # Extract paper elements from the listing
    papers_dd <- page %>% html_nodes("dd") # Paper info (Title, Authors, Abstract, Meta)
    papers_dt <- page %>% html_nodes("dt") # arXiv IDs/links

    # Extract data from each paper
    for (j in seq_along(papers_dd)) {

      # Title
```

```
      title <- papers_dd[j] %>%
        html_node("div.list-title") %>%
        html_text(trim = TRUE) %>%
        str_remove("Title:\\s*")

      # Authors
      author <- papers_dd[j] %>%
        html_node("div.list-authors") %>%
        html_text(trim = TRUE) %>%
        str_remove("Authors:\\s*")

      # Abstract
      abstract <- papers_dd[j] %>%
        html_node("p.mathjax") %>%
        html_text(trim = TRUE)

      # Submission info (contains dates)
      meta <- papers_dd[j] %>%
        html_node("div.list-comments") %>%
        html_text(trim = TRUE)

      # Get arXiv ID from dt tag
      arxiv_id <- papers_dt[j] %>%
        html_node("a[title='Abstract']") %>%
        html_text(trim = TRUE)

      # Create temporary dataframe for this paper
      temp_df <- data.frame(
        arxiv_id = ifelse(is.na(arxiv_id), "", arxiv_id),
        title = ifelse(is.na(title), "", title),
        author = ifelse(is.na(author), "", author),
        abstract = ifelse(is.na(abstract), "", abstract),
        meta_raw = ifelse(is.na(meta), "", meta),
        stringsAsFactors = FALSE
      )

      all_papers[[length(all_papers) + 1]] <- temp_df
    }

  }, error = function(e) {
    print(paste("Error on page starting at", i, ":", e$message))
  })

  # Wait 3 seconds between requests to be respectful
  Sys.sleep(3)
}

# Combine all lists into one dataframe
df_papers <- bind_rows(all_papers)

# Check count
print(paste("Total papers extracted:", nrow(df_papers)))
```

“‘{r} # Cleaning the Data df_clean <- df_papers %>% filter(title != “” & author != “ “) %>% # Remove

empty rows mutate(# Create DOI from arXiv ID doi = paste0("10.48550/arXiv.", arxiv_id),

```
# Extract submission date from meta_raw (e.g., "Mon, 1 Jan 2024")
submission_date_text = str_extract(meta_raw,
                                   "[A-Z][a-z]{2},\\s*\\d{1,2}\\s+[A-Z][a-z]{2}\\s+\\d{4}"),

# Parse the date
submission_date = dmy(submission_date_text),

# Extract "originally announced" date as backup
announced_text = str_extract(meta_raw,
                             "originally announced\\s+[A-Z][a-z]+\\s+\\d{4}"),
announced_text = str_remove(announced_text, "originally announced\\s+"),
originally_announced = my(announced_text)
```

) %>%

Use originally_announced if submission_date is missing mutate(submission_date = if_else(is.na(submission_date), originally_announced, submission_date)) %>%

Remove rows where we couldn't extract any date filter(!is.na(submission_date))

Preview

head(df_clean %>% select(arxiv_id, title, submission_date, doi))

Arranging Papers by Date

df_sorted <- df_clean %>% arrange(submission_date) %>% mutate(Rank = row_number())

Show summary

print(paste("Total papers after cleaning:", nrow(df_sorted))) print(paste("Date range:", min(df_sorted$submission_date), "to", m

```
```{r}
# Monthly Time Series Plot
papers_per_month <- df_sorted %>%
  mutate(month_year = floor_date(submission_date, "month")) %>%
  group_by(month_year) %>%
  summarise(count = n(), .groups = "drop")

ggplot(papers_per_month, aes(x = month_year, y = count)) +
  geom_line(color = "darkblue", size = 1.2) +
  geom_point(color = "red", size = 3) +
  labs(title = "Time Series: Astrophysics Papers (arXiv astro-ph.SR)",
       subtitle = "Frequency of papers submitted per month",
       x = "Date",
       y = "Number of Papers") +
  theme_minimal(base_size = 12)

# Weekly Time Series Plot
papers_per_week <- df_sorted %>%
  mutate(week_year = floor_date(submission_date, "week")) %>%
```
```

```
  group_by(week_year) %>%
  summarise(count = n(), .groups = "drop")

ggplot(papers_per_week, aes(x = week_year, y = count)) +
  geom_line(color = "#10b981", size = 1.2) +
  geom_point(color = "#7c3aed", size = 2) +
  labs(title = "Weekly Paper Submissions",
       subtitle = "Papers submitted per week",
       x = "Week",
       y = "Number of Papers") +
  theme_minimal(base_size = 12)

# Daily Time Series Plot
papers_per_day <- df_sorted %>%
  group_by(submission_date) %>%
  summarise(count = n(), .groups = "drop")

ggplot(papers_per_day, aes(x = submission_date, y = count)) +
  geom_line(color = "#6366f1", size = 1) +
  geom_point(color = "#ec4899", size = 2) +
  labs(title = "Daily Paper Submissions",
       subtitle = "Number of papers submitted each day",
       x = "Date",
       y = "Number of Papers") +
  theme_minimal(base_size = 12)
```

“‘{r} # Display Sample Papers Table papers_display <- df_sorted %>% select(Rank, arxiv_id, title, author, submission_date, doi) %>% head(25)

kable(papers_display, caption = “First 25 Papers from astro-ph.SR”, col.names = c(“Rank”, “arXiv ID”, “Title”, “Authors”, “Submission Date”, “DOI”), booktabs = TRUE) %>% kable_styling(latex_options = c(“scale_down”, “striped”, “hold_position”), font_size = 8)

Data Export (Code is shown, but commented out/set to eval=FALSE in Rmd to prevent execution by default)

write.csv(df_sorted, “arxiv_astrophysics_papers.csv”, row.names = FALSE)

write.csv(papers_per_month, “monthly_submissions.csv”, row.names = FALSE)

“‘