

R Studio Input

```
# =====  
# FINAL TEMPORAL BURST ANALYSIS + PLOTTING  
# Document-Level Frequency + Structural Break Detection  
# =====  
  
library(dplyr)  
library(stringr)  
library(tidytext)  
library(tidyr)  
library(changepoint)  
  
# -----  
# STEP 1: LOAD DATA  
# -----  
  
data <- read.csv("your_dimensions_file.csv",  
                skip = 1,  
                stringsAsFactors = FALSE,  
                encoding = "UTF-8")  
  
names(data)[3] <- "Title"  
names(data)[4] <- "Abstract"  
names(data)[6] <- "Year"  
  
data <- data[, c("Title", "Abstract", "Year")]  
data <- data[!is.na(data$Year), ]  
data$Year <- as.numeric(data$Year)  
data$text <- tolower(paste(data$Title, data$Abstract))  
  
# -----  
# STEP 2: BIGRAM EXTRACTION  
# -----  
  
tidy_terms <- data %>%  
  mutate(doc_id = row_number()) %>%  
  unnest_tokens(bigram, text, token = "ngrams", n = 2)  
tidy_terms <- tidy_terms %>%  
  separate(bigram, into = c("word1", "word2"), sep = " ") %>%  
  filter(!word1 %in% stop_words$word,  
         !word2 %in% stop_words$word) %>%  
  unite(bigram, word1, word2, sep = " ")  
  
# -----  
# STEP 3: REMOVE NUMERIC / CITATION ARTIFACTS  
# -----
```

```

tidy_terms <- tidy_terms %>%
  filter(!str_detect(bigram, "^[0-9]")) %>%
  filter(!str_detect(bigram, "[0-9]{4}")) %>%
  filter(!str_detect(bigram, "et al")) %>%
  filter(!str_detect(bigram, "al [0-9]"))
# -----

# STEP 4: DOCUMENT-LEVEL YEARLY FREQUENCY
# -----

bigram_yearly <- tidy_terms %>%
  distinct(doc_id, Year, bigram) %>%
  group_by(bigram, Year) %>%
  summarise(Frequency = n(), .groups = "drop")
all_years <- seq(min(data$Year), max(data$Year))
# Keep terms with minimum frequency
bigram_totals <- bigram_yearly %>%
  group_by(bigram) %>%
  summarise(Total = sum(Frequency)) %>%
  filter(Total >= 3)
bigram_yearly <- bigram_yearly %>%
  filter(bigram %in% bigram_totals$bigram)
# -----

# STEP 5: BURST CALCULATION
# -----

burst_results <- data.frame()
for(term in unique(bigram_yearly$bigram)){
  term_data <- bigram_yearly %>%
    filter(bigram == term)
  term_data <- data.frame(Year = all_years) %>%
    left_join(term_data, by="Year")

  term_data$Frequency[is.na(term_data$Frequency)] <- 0
  if(sum(term_data$Frequency) < 3) next
  cpt_model <- tryCatch(
    cpt.mean(term_data$Frequency, method="PELT"),
    error = function(e) NULL
  )
  if(is.null(cpt_model)) next

```

```

break_index <- cpts(cpt_model)
iflength(break_index) == 0) next

break_year <- term_data$Year[break_index[1]]

baseline <- term_data %>% filter(Year < break_year)
surge <- term_data %>% filter(Year >= break_year)

if(nrow(baseline) == 0 | nrow(surge) == 0) next

baseline_avg <- mean(baseline$Frequency)
peak_freq <- max(surge$Frequency)
peak_year <- surge$Year[which.max(surge$Frequency)]

burst_strength <- (peak_freq - baseline_avg) / (baseline_avg + 1)

burst_results <- rbind(burst_results,
  data.frame(
    Term = term,
    Baseline_Average = round(baseline_avg,2),
    Peak_Year = peak_year,
    Peak_Frequency = peak_freq,
    Structural_Shift_Year = break_year,
    Burst_Strength_Index = round(burst_strength,2)
  ))
}

burst_results <- burst_results %>%
  arrange(desc(Burst_Strength_Index))

cat("\n===== FULL BURST TABLE =====\n")
print(head(burst_results, 20))

# -----
# OPTIONAL: MERGE COST OVERRUN VARIANTS
# -----

if(any(burst_results$Term %in% c("cost overrun","cost overruns"))){
  merged <- burst_results %>%
    filter(Term %in% c("cost overrun","cost overruns")) %>%
    slice_max(Burst_Strength_Index, n = 1)
  merged$Term <- "cost overrun(s)"
  burst_results <- burst_results %>%
    filter(!Term %in% c("cost overrun","cost overruns"))
  burst_results <- rbind(burst_results, merged) %>%
    arrange(desc(Burst_Strength_Index))
}

# -----

```

```

> # STEP 6: PLOT FUNCTION

> # -----

>

> plot_theme <- function(selected_term){
+
+   term_data <- tidy_terms %>%
+     distinct(doc_id, Year, bigram) %>%
+     filter(bigram == selected_term |
+       (selected_term == "cost overrun(s)" &
+         bigram %in% c("cost overrun", "cost overruns"))) %>%
+     group_by(Year) %>%
+     summarise(Frequency = n())
+
+   term_data <- data.frame(Year = all_years) %>%
+     left_join(term_data, by="Year")
+
+   term_data$Frequency[is.na(term_data$Frequency)] <- 0
+
+   cpt_model <- tryCatch(
+     cpt.mean(term_data$Frequency, method="PELT"),
+     error = function(e) NULL
+   )
+
+   break_year <- NA
+
+   if(!is.null(cpt_model)){
+     break_index <- cpts(cpt_model)
+
+     if(length(break_index) > 0){
+       break_year <- term_data$Year[break_index[1]]
+     }
+   }
+
+   plot(term_data$Year,
+     term_data$Frequency,
+     type = "l",
+     lwd = 3,
+     col = "black",
+     xlab = "Year",

```

```
+   ylab = "Number of Documents",
+   main = paste("Temporal Trend:", selected_term),
+   xlim = c(2000, 2030),
+   xaxt = "n")
+ axis(1, at = seq(2000, 2030, 5))
+ if(!is.na(break_year)){
+   abline(v = break_year,
+         col = "red",
+         lwd = 2,
+         lty = 2)
+ }
+ }
> # -----
> # STEP 7: GENERATE PLOTS
> # -----
```