Text Analysis with JSTOR Archives
Supplementary File 2018

The following R code walks through the preparation of the data and plotting of the figure in the main text of the article. JSTOR provides instructions on creating and requesting a dataset (https://www.jstor.org/dfr/about/creating-datasets), and the following code will require access to the two delivered folders: "metadata" and "ngram1".

The "Prep #1" script uses a function to extract JSTOR xml metadata into an R data frame. The analysis in the article uses two separate datasets and only the ASR-specific extraction code is reproduced below. The AJS-dataset was extracted in near-identical fashion.

The "Prep #2" script uses a function to convert JSTOR ngram data into a "pseudo raw text" format by multiplying each word by its associated frequency. What results is essentially the full text of the article, sorted by word. In other words, if "sociology" was the most frequent word at 150 uses, the first 150 words of the pseudo raw text will be "sociology".

The "Prep #3" script joins the resultant data frames together using the file name as an identifier.

With this data frame complete, the concluding script details plotting decisions, including the frequency analysis of ASR articles and the authorship and regression analysis of AJS articles.

```
################################
# JSTOR Data Prep #1
# Parsing out xml files
################################

# Load packages
require(xml2)
require(tidyverse)
require(plyr)
require(dplyr)

# Identify path to metadata folder and list files
path <- "/Users/jberna5/Desktop/ASR JSTOR/metadata"
files <- list.files(path)

# Initialize empty set
final_data <- NULL

# Using the xml2 package: for each file, extract metadata and append row to final_data
for (x in files){
  path <- read_xml(paste0(path1, "/", x))

  # File name
  file <- x

  # Article type
  type <- xml_find_all(path, "/article/@article-type") %>%
    xml_text()
```

```
# Title
title <- xml_find_all(path, xpath = "/article/front/article-meta/title-group/article-title") %>%
  xml_text()

# Author names
authors <- xml_find_all(path, xpath = "/article/front/article-meta/contrib-group/contrib") %>%
  xml_text()
auth1 <- authors[1]
auth2 <- authors[2]
auth3 <- authors[3]
auth4 <- authors[4]
auth5 <- authors[5]
auth6 <- authors[6]
auth7 <- authors[7]
auth8 <- authors[8]
auth9 <- authors[9]
auth10 <- authors[10]

# Affiliations
affil <- xml_find_all(path, xpath = "/article/front/article-meta/contrib-group/aff") %>%
  xml_text()
affil1 <- affil[1]
affil2 <- affil[2]
affil3 <- affil[3]
affil4 <- affil[4]
affil5 <- affil[5]
affil6 <- affil[6]
affil7 <- affil[7]
affil8 <- affil[8]
affil9 <- affil[9]
affil10 <- affil[10]

# Abstract
abstract <- xml_find_all(path, xpath = "/article/front/article-meta/abstract") %>%
  xml_text()

# Month
month <- xml_find_all(path, xpath = "/article/front/article-meta/pub-date/month") %>%
  xml_text()

# Year
year <- xml_find_all(path, xpath = "/article/front/article-meta/pub-date/year") %>%
  xml_text()

# Volume
vol <- xml_find_all(path, xpath = "/article/front/article-meta/volume") %>%
  xml_text()

# Issue
iss <- xml_find_all(path, xpath = "/article/front/article-meta/issue") %>%
  xml_text()

# First page
fpage <- xml_find_all(path, xpath = "/article/front/article-meta/fpage") %>%
  xml_text()

# Last page
lpage <- xml_find_all(path, xpath = "/article/front/article-meta/lpage") %>%
  xml_text()

# Footnote
notes <- xml_find_all(path, xpath = "/article/front/notes") %>%
```

```
    xml_text()

  # Bind all together
  article_meta <- cbind(file, type, title,
                        auth1, auth2, auth3, auth4, auth5, auth6, auth7, auth8, auth9, auth10,
                        affil1, affil2, affil3, affil4, affil5, affil6, affil7, affil8, affil9, affil10,
                        abstract, month, year, vol, iss, fpage, lpage, notes)

  final_data <- rbind.fill(final_data, data.frame(article_meta, stringsAsFactors = FALSE))

  # Print progress
  if (nrow(final_data) %% 250 == 0){
    print(paste0("Extracting document # ", nrow(final_data)))
    print(Sys.time())
  }
}

# Check output
names(final_data)
str(final_data)

# Shorter name
fd <- final_data

# Adjust data types
fd$type <- as.factor(fd$type)
fd$month <- as.numeric(fd$month)
fd$year <- as.numeric(fd$year)
fd$vol <- as.numeric(fd$vol)
fd$iss <- as.numeric(fd$iss)

# Remove variables if ALL rows are "NA"
not_all_na <- function(x) any(!is.na(x))
fd <- fd %>% select_if(not_all_na)

# Create date variable
fd$date <- paste("01", fd$month, fd$year, sep = "-")
fd$date <- as.Date(fd$date, "%d-%m-%Y")
class(fd$date)
head(fd$date)

##################
# Page variables
##################
# Examine for any unusual values ("cover", "ix", etc)
View(count(fd$fpage))
View(count(fd$lpage))

# Convert to numeric (roman numerals converted to NA by default)
fd$fpage <- as.numeric(fd$fpage)
fd$lpage <- as.numeric(fd$lpage)
fd$fpage[fd$fpage == ""] <- NA
fd$lpage[fd$lpage == ""] <- NA

# Create length variable
fd$length <- fd$lpage - fd$fpage
# Are you getting negative length variables? See "negative page length.R"

# Examine data
View(arrange(fd, desc(length)))

ggplot(fd, aes(length)) +
```

```
  geom_histogram(bins = 50)

####################
# Explore data
####################
# Examine type categories
dplyr::count(fd, type)
# Careful of "research-articles" with titles like "Book Review" or "Review"
filter(fd, type == "research-articles" & title == "Book Reviews")

# Looks like JSTOR stopped hosting ASR book reviews around 1970...
ggplot(fd, aes(year)) +
  geom_bar(aes(fill = type))

# Save as .RData file for future work
save(fd, file = "/Users/jberna5/Desktop/ASRjstor.RData")

#############################################################
# JSTOR Data Prep #2
# Read ngrams and convert to "pseudo raw text"
#############################################################
require(tidyverse)
require(quanteda)
require(dplyr)
#########################################################

# jstor_expand() takes a file and returns the name and the pseudo raw text
# Use map to run this over all files in n_files
# Once you have a function, you can use safely() to get around any errors


# Set up files paths
path <- "/Users/jberna5/Desktop/ASR JSTOR/ngram1/"
n_files <- list.files(path)

########################
# Writing function
########################
# Input the file name
jstor_expand <- function(fname){
  text1 <- read.table(paste0(path, fname))
  text1 <- text1 %>%
    rename(word = V1, n = V2)
  text1$word <- as.character(text1$word)
  text1$n <- as.numeric(text1$n)
  text1$file <- fname

  # Check to see if a digit is present and count characters.
  # Then drop if a number or less than 3 characters.
  # Drop these two variables
  text1 <- text1 %>%
    mutate(digit = str_detect(word, "[:digit:]"),
           len = str_length(word)) %>%
    filter(digit == F & len > 2) %>%
    select(-digit, -len)

  # Repeat word by its frequency, then collapse and save as a pseudo raw text
  text1$ex <- NA
  for (y in 1:nrow(text1)){
    # Repeat x word, n times...
    a <- rep(text1$word[y], text1$n[y])
    # Collapse this into a single string and save as 'ex' variable.
```

```
    text1$ex[y] <- str_c(a, collapse = " ")
  }

  text_collapsed <- str_c(text1$ex, collapse = " ")
  return <- cbind(fname, text_collapsed)

  # Print out for progress
  index <- str_which(n_files, fname)
  if (index %% 5 == 0){
    print(paste0("Expanding file # ", index, "/", length(n_files)))
    print(Sys.time())
  }
  return
}

####################################

# safely() wraps around function for error-proofing
safely_jstor <- safely(jstor_expand)

# Map the safely_jstor function onto each item in n_files. Save as list "asr_expand"
asr_expand <- map(n_files, safely_jstor)

# Unlist result into a dataframe
asr_df <- data.frame(matrix(unlist(asr_expand), nrow=length(asr_expand), byrow=T), stringsAsFactors=FALSE)

# Manipulate data types
asr_df <- asr_df %>%
  mutate(file = unlist(X1), text = as.character(X2)) %>%
  select(-X1, -X2)
str(asr_df)

# Save when done
save(asr_df, file = "~/Desktop/ASRtext.RData")

####################################
# JSTOR Data Prep #3
# Merge metadata with pseudo raw text
####################################
require(tidyverse)
####################################
# Load ASRtext.RData (full text)
load(file.choose())
full_text <- asr_df
rm(asr_df)

# Load ASRjstor.RData (metadata)
load(file.choose())
meta <- fd
rm(fd)

# Make joining column
meta$id <- str_replace_all(meta$file, ".xml", "")
meta$id[1:10]
full_text$id <- str_replace_all(full_text$file, "-ngram1.txt", "")
full_text$id[1:10]

# # Option 1: Join all rows no matter what
# merged <- full_join(meta, full_text, by = "id")

# Option 2: Join only rows with matching ids
merged <- inner_join(meta, full_text, by = "id")
```

```
# Examine
names(merged)
View(merged[1:10,])


save(merged, file = "~/Desktop/ASRjoined.RData")

#######################################
# JSTOR Data Plotting #1
#
#######################################
require(broom)
require(ggplot2)
require(RColorBrewer)
require(scales)
require(gridExtra)

#############################################################################
# American Sociological Review - keyword search
#############################################################################
# Load "ASRjoined.RData" - 14231 obs of 25 vars.
load(file.choose())

# Save as asr_arts
asr_arts <- filter(merged, type == "research-article" & title != "Book Review" & title != "Book Reviews" & title !=
"Book Reviewers" & title != "Commentary and Debate" & title != "Book Notes")
asr_arts$year <- as.numeric(as.character(asr_arts$year))


# "Holy Trinity" search
asr_arts$race <- str_count(asr_arts$text, "race")
asr_arts$class <- str_count(asr_arts$text, "class")
asr_arts$gender <- str_count(asr_arts$text, "gender")

# Create yearly totals
ht_plot <- asr_arts %>%
  group_by(year) %>%
  summarize(race = sum(race),
            class = sum(class),
            gender = sum(gender)) %>%
  ungroup() %>%
  gather("ht", "n", race:gender)

ht_plot_p <- filter(ht_plot, n > 0)

# Save variable as factor
ht_plot_p$ht <- factor(ht_plot_p$ht, levels = c("class", "race", "gender"))

# Choosing colors
colors <- hue_pal()(3)
colors2 <- c(colors[2], colors[3], colors[1])

# Final Tweaking
asr_ht <- ggplot(ht_plot_p, aes(year, log2(n))) +
  geom_point(aes(color = ht, alpha = n), size = 3) +
  scale_alpha_continuous(range = c(0.3, 0.8), guide = F) +
  geom_smooth(aes(color = ht), se = F, size = 2) +
  scale_color_manual(values = colors2, name = "") +
  scale_y_continuous(breaks = seq(0, 13, by = 1), labels = c(1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096,
8192))  +
  scale_x_continuous(breaks = seq(1900, 2000, by = 20)) +
```

```
  labs(title = "American Sociological Review", subtitle = "Word frequencies per year (log2 scale)", y = "Yearly
Mentions", x = NULL, caption = "") +
  theme(text=element_text(size = 14, family="Times New Roman"),
        plot.subtitle = element_text(size = 12),
        #legend.position = "bottom",
        legend.title=element_blank(),
        axis.text = element_text(size = 14),
        panel.background = element_rect(fill="white"),
        panel.grid.minor = element_line(color="grey90"),
        panel.grid.major = element_line(color="grey90"),
        plot.margin = unit(c(0.5,0.5,0.5,0.5), "cm"),
        legend.position = c(.9,.5))
asr_ht


##############################################################################
# American Journal of Sociology - Page length as function of date * author
##############################################################################
# Load "AJSjstor4.RData" - 23885 obs of 36 vars.
load(file.choose())

# Filter by type
ajs_arts <- filter(merged, type == "research-article")

# Eliminate problem titles
'%ni%' <- Negate('%in%')

problem_titles <- c("Book Review", "Book Reviews", "Book Reviewers", "Commentary and Debate", "Book Notes")

ajs_arts <- ajs_arts %>%
  filter(title %ni% problem_titles) %>%
  filter(str_detect(title, "Acknowledgments") == FALSE) %>%
  filter(str_detect(title, "Contents of Volume") == FALSE) %>%
  filter(str_detect(title, "Call for Papers") == FALSE) %>%
  filter(str_detect(title, "Errata") == FALSE)

ajs_arts$year <- as.numeric(as.character(ajs_arts$year))

# Create authorship variable from number of author columns != NA
ajs_arts$nauth <- 1
ajs_arts$nauth[!is.na(ajs_arts$auth2)] <- 2
ajs_arts$nauth[!is.na(ajs_arts$auth3)] <- 3
ajs_arts$nauth[!is.na(ajs_arts$auth4)] <- 4
ajs_arts$nauth <- as.integer(ajs_arts$nauth)
ajs_arts$title <- as.character(ajs_arts$title)

# Run linear model with interaction term
ajs_mod <- lm(length ~ year + factor(nauth) + year:factor(nauth), data = ajs_arts)
# Save R-squared
rsq <- summary(ajs_mod)$r.squared
# Add to existing data
ajs_res <- augment(ajs_mod)

# Choose colors
colz <- brewer.pal(8, "Blues")
colz <- colz[5:8]

# Plot results
inter <- ggplot(ajs_res, aes(year, length)) +
  geom_point(position = "jitter", aes(alpha = .resid, fill = .resid, size = .resid), pch = 21) +
  scale_size(guide = F) +
  scale_alpha(range = c(0.3, 0.8), guide = F) +
```

```
  scale_fill_continuous(low = "black", high = "red", guide = F) +
  geom_line(size = 2, aes(y = .fitted, color = factor.nauth.)) +
  scale_color_manual(values = colz, name = "Authors") +
  scale_y_continuous(limits = c(0, 150), breaks = seq(0, 150, by = 25)) +
  scale_x_continuous(limits = c(1897, 2015)) +
  labs(title = "American Journal of Sociology", subtitle = "Article length as function of ¥"date x author¥"", caption
= paste("R-Squared =", round(rsq, 2)), x = NULL, y = "Page Length") +
  theme(text=element_text(size = 14, family="Times New Roman"),
        plot.subtitle = element_text(size = 12),
        axis.text = element_text(size = 14),
        panel.background = element_rect(fill="white"),
        panel.grid.minor = element_line(color="grey90"),
        panel.grid.major = element_line(color="grey90"),
        plot.margin = unit(c(0.5,0.5,0.5,0.5), "cm")) +
  geom_text(aes(label = ifelse(length>125, "Albion Small (1916, 21:6)", "")),
            nudge_x = 43)

inter


# Combine plots into one figure
g <- grid.arrange(asr_ht, inter, ncol = 2)

ggsave(g, file = "~/Desktop/socius.tiff", width = 10, height = 5, units = "in", dpi = 1200)

ggsave(g, file = "~/Desktop/socius.jpg", width = 10, height = 5, units = "in", dpi = 600)
```