Following the Leader: Understanding Relational Voting in the United Nations General Assembly

MACS 33002

Ipek Cinar, Pete Cuppernull, Seth Harrison, Casey Mallon

Group Duties:

Ipek Cinar: Producing findings, creating graphics and developing animations for illustration, merging and cleaning the code and providing code feedback, contributing to paper

Pete Cuppernull: Cleaning data, developing unsupervised model, performing data analysis, performing clustering analysis, contributing to paper

Seth Harrison: Editing the presentation, contributing to and editing the paper

Casey Mallon: Contextualizing research, examining existing literature, writing the paper and powerpoint

Which countries lead the charge on human rights issues? The United Nations (UN) is the central node around which modern international cooperation takes place. In seeking to formalize international norms—which are "social facts" of a community that reflect shared intentions and understandings—into international law, states come together through the UNto vote on pivotal global issues such as climate change, norms of warfare, and increasingly to establish and uphold norms of human rights.[1] Human rights include matters regarding the political, social, and economic equality and maintenance of rights for all persons, regardless of group affiliation.[2] Human rights issues are increasingly important to international debate given the increased potential for populations to move across borders, bringing into question how the human rights of persons are to be guaranteed outside of their home countries.[3]

In this paper, we develop an unsupervised topic model to classify UN General Assembly (UNGA) human rights resolutions and then examine the presence (or lack thereof) of "leaders" on certain human rights issues. We define leaders as states which exist within the largest voting blocs at the UNGA—effectively, a vote "leader" is a state which disproportionately exhibits popular voting behavior, suggesting that their vote preferences are driving the voting behavior of other states. We establish empirical evidence to suggest the existence of human rights policy leaders, an existence which has largely been taken as an assumption in human rights literature.

[1] Payne, Rodger A. "Persuasion, Frames and Norm Construction." European Journal of International Relations 7, no. 1 (2001): 37–61.

[2]See Park, Han S. 1987. "Correlates of human rights: Global tendencies." *Human Rights Quarterly* 9: 405-413

[3] See Giordani, Paolo E. and Michele Ruta. 2012. "Coordination in immigration policy." *Journal of International Economics* 89: 55-67., Binder, Martin. 2015. "Paths to intervention: What explains the UN's selective response to humanitarian crises?" *Journal of Peace Research* 52(6): 712-726., Dowty, Alan and Gil Loescher. 1996. "Refugee Flows as Grounds for International Action." *International Security* 21(1): 43-71., Noll, Gregor. 2010. "Why Human Rights Fail to Protect Undocumented Migrants." *Journal of Migration Law* 12: 241-272., Bosniak, Linda S. 1991. "Human rights, state sovereignty and the protection of undocumented migrants under the International Migrant Workers Convention." *International Migration Review* 25: 737-770.

The contribution of our study is thus twofold: we close empirical gaps left by scholars who have simply assumed the existence of policy leaders, and we leverage an empirical method (unsupervised topic models) which has been largely overlooked in this body of literature.

The Importance of International Organizations

International organizations (IOs) are platforms through which states discuss, debate, and resolve international disputes, formed with the primary goal of preventing interstate war.. The most prominent of these IOs is the UN. Although the effectiveness of the UN to prevent war is debated among scholars, one cannot deny that there are tangible results stemming from international participation in the UN, including the development of more robust international law.[4] This paper, therefore, continues in the tradition of institutionalism, foundationally asserting that efforts of international cooperation take place within institutional contexts—the UN is the most appropriate venue in which to study international cooperation because it is the most comprehensive and inclusive IO.[5]

Many scholars studying the UN acknowledge the analytic power of utilizing UNGA voting records to measure state voting preferences and positions.[6] However, existing scholars

[4] See, for example, Mearsheimer, John J. 1994. "The false promise of international institutions." *International security* 19(3): 5-49 for an example of the realist approach to IOs and Keohane, Robert O. 1988. "International institutions: Two approaches." *International studies quarterly* 32(4): 379-396 for an example of the liberal institutionalist approach to IOs. See Cortell, Andrew P., and James W. Davis Jr. 1996. "How do international institutions matter? The domestic impact of international rules and norms." *International Studies Quarterly* 40(4): 451-478; Hawkins, Darren. 2004. "Explaining costly international institutions: Persuasion and enforceable human rights norms." *International Studies Quarterly* 48(4): 779-804; Finnemore, Martha, and Kathryn Sikkink. 1998. "International norm dynamics and political change." *International organization* 52(4): 887-917; Finnemore, Martha. 1993. "International organizations as teachers of norms: the United Nations Educational, Scientific, and Cultural Organization and science policy." *International organization* 47(4): 565-597.

[5] See Keohane, Robert O. 1988. "International institutions: Two approaches." *International studies quarterly* 32(4): 379-396.

[6] See Brazys, Samuel and Diana Panke. 2017. " Why do states change positions in the United Nations General Assembly?" International Political Science Review 38(1): 70-84. Dreher, Axel and James Raymond Vreeland. 2011. "Buying Votes and International Organizations." Center for European, Governance and Economic Development Research Discussion Papers 123. Bueno de Mesquita B (1983) *The War Trap*. New Haven, CT: Yale University Press. Kim

assume the existence of policy leaders and incorporate measures of such leaders into their models. For example, Bailey et. al (2017) develops an ideal point model that finds support for Cold War period voting blocs based on latent preferences, but still takes for granted the existence of policy leaders, the US and USSR, based on the existing Cold War balance of power.These scholars assume that because a state is a great power (via its military or economic capabilities), it has a commensurate amount of leadership within IOs. There is no empirical basis, however, to suggest that institutional dynamics mirror the overarching balance of power in the international system, though the case has been alluded to by liberal institutionalists.. Our study, therefore, seeks to demonstrate the existence of policy leaders on specific issues rather than taking for granted the institutional mirroring of balance of power politics.

Human Rights Issues

Human rights are generally studied in the context of the UN because the UN is the primary "authority" for these issues.[7] Human rights issues have become more complex and nuanced asa result of civil wars, overall violence, and natural disasters, and human rights crises pose some of the greatest international challenges facing states today. Accordingly, a new body of research has developed to answer the normative and empirical questions posed by these developments. [8] We choose to study policy leaders at the UN within the context of human rights

---

SY and Russett B (1996) The new politics of voting align- ments in the United Nations General Assembly. *International Organization* 50(4): 629–652.

[7] Assembly, UN General. 1948. "Universal declaration of human rights." *UN General Assembly* 302, no. 2.

[8] See Hawkins, Darren. 2004. "Explaining Costly International Institutions: Persuasion and Enforceable Human Rights Norms." *International Studies Quarterly*. Volume 48(4): 779–804. Hathaway, James C. 1991. "Reconceiving refugee law as human rights protection." *Journal of Refugee Studies* 4(2): 113-131. Salehyan, Idean. 2007. "Refugees and the study of civil war." *Civil Wars* 9(2): 127-141. Binder, Martin. 2015. "Paths to intervention: What explains the UN's selective response to humanitarian crises?." *Journal of Peace Research* 52(6): 712-726. Domagała, Arkadiusz. 2018. "For and Against: Analysing the Determinants of Humanitarian Intervention. Libya (2011) and Syria (2011–2013) Compared." *Polish Political Science Review* 6(1): 34-49. Geiger, Martin, and Antoine Pécoud. 2017. "International organisations and the securitisation of migration." In *Handbook on Migration and*

and contribute to this growing body of research given its increased importance in international politics. Our methodology, however, could likely be applied to other UN issue areas and other international bodies in which states regularly vote.

Data and Methodology

We develop an unsupervised topic model to examine UNGA voting records and identify clusters of issue areas. Leveraging these classifications, we shed light upon which states are vote leaders and how states vote differently between issue areas. We use the text of the vote proposals to categorize the issue areas that are voted upon; the dataset we use includes every roll-call vote held in the UNGA from 1946-2018.[9] Coding for the dataset is based on existing compilations of UNGA roll call votes.[10] Each observation in the dataset is a country-proposal and is assigned one of five vote choice values: Yes, No, Abstain, Absent, or Not a member. In addition to coding vote choice, the data is coded topically, allowing for easy identification of human rights-related votes.

We measure which states are the leaders on given topics over time by ranking the percentage of other states that vote in accordance with that state across UN proposals. With this

---

Security. Edward Elgar Publishing. Thouez, Colleen. 2019. "Strengthening migration governance: the UN as 'wingman'." *Journal of Ethnic and Migration Studies* 45(8): 1242-1257.

[9] Erik Voeten "Data and Analyses of Voting in the UN General Assembly" Routledge Handbook of International Organization, edited by Bob Reinalda (published May 27, 2013). Available at SSRN: http://ssrn.com/abstract=2111149

[10] Sources identified in the codebook include: Inter-university Consortium for Political and Social Research (ICPSR*). United Nations Roll Call Data, 1946-1985* [Computer file]. Ann Arbor, MI: Inter-university Consortium for Political and Social Research [producer and distributor], 1982.; Gartzke, Erik and Dong-Joon Jo. *UN General Assembly Voting* V3.0 January 2002. http://dss.ucsd.edu/~egartzke/;Jo, Dong-Joon. *Dundas Dataset.* Used in: Jo, Dong-Joon. 2000. "Power Resources and Influence at the UN General Assembly." *Presented at the Annual Meeting of the Northeastern Political Science Association*.; Schopen, Lynn; Newcombe, Hanna; Young, Chris; Wert, James, *Nations on Record: United Nations General Assembly Roll-Call Votes (1946-1973)*. Oakville-Dundas, ON: Canadian Peace Research Institute, 1975. (and subsequent supplements).;Kim, Soo Yeon; Russett, Bruce, "The new politics of voting alignments in the United Nations General Assembly." *International Organization*. Aut 1996, 50, (4), 629 - 652.;Various UN Resources including the *Official Records to the Proceedings of the United Nations General Assembly, UNBISNET*: http://unbisnet.un.org/, and UN documentation on-line: http://www.un.org/documents/resga.htm.

identification strategy, we assume that on average, a state that votes within a large voting bloc has influenced the voting behavior of other states. Thus, we consider states which have the highest percentage of other states vote in accordance with them as a vote leader.

Our data set contains 978 unique human rights proposals. To process the text of the proposals, we remove numbers and stop words from the proposal texts and stem the remaining text. We then create a document-term matrix with all features and term-frequency weighting. Before proceeding to classification, we assess the clusterability of the documents by evaluating dissimilarity plots, shown below in Figure 1.

Figure 1: Dissimilarity Plots of Document-Term Matrix

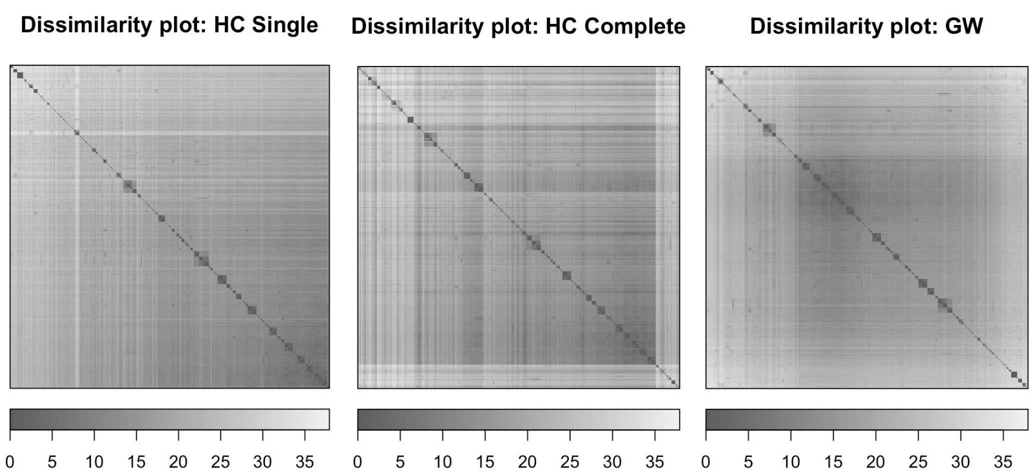


Figure 1 shows relatively weak stratification in the document-term matrix with small clusters of highly correlated documents. This informs us that there may be valuable clusters within the data, but we will likely need a high number of clusters in the classifier to capture these specific smaller groups. After doing so, as indicated by some of the larger stratification in Figure 1, we may be able to manually aggregate some of these smaller categories based on context.

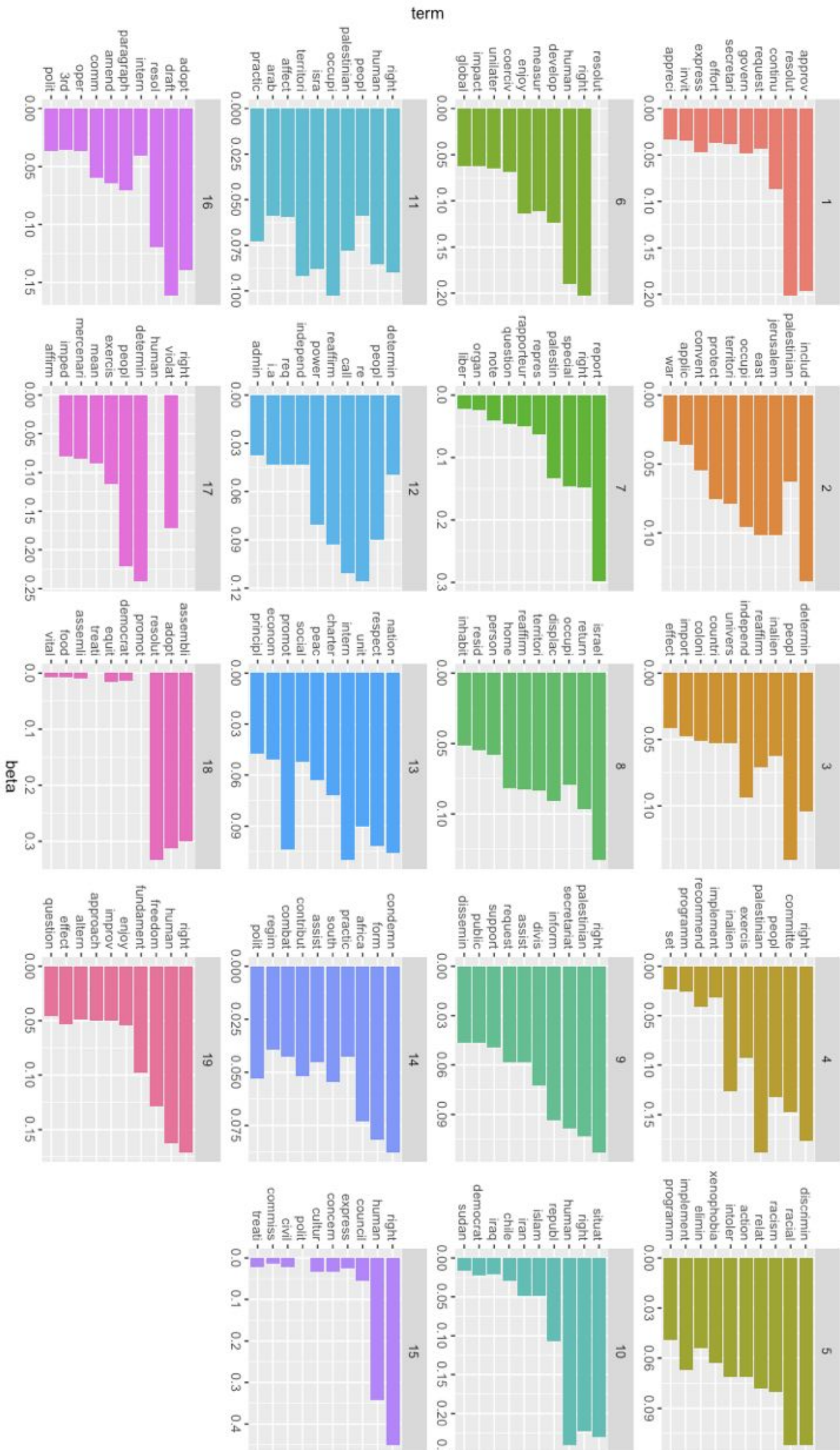
For classification, we use a latent dirichlet allocation (LDA) model which leverages the variational expectation maximization algorithm.[11] We tuned both the number of clusters in the model and the sparsity of the document-term matrix to select the model which made the most confident classifications—we measure classification confidence by the difference between the top two classification weights per proposal in the document-term gamma matrix.[12] The best performing model has 19 clusters with a sparsity of .99.

The most common words in each cluster are presented in Figure 2:

[11] We also pursued k-means clustering of the documents, which can be viewed in the Appendix. This resulted in clusters which appeared less intuitive than the LDA classification, and we thus chose to use the LDA classification for the rest of the project. However, a silhouette width validation of k-means models with the number of clusters ranging from 2-20 showed that models with higher numbers of clusters performed approximately the same as those with fewer clusters. This gave us further confidence that our method of using the unsupervised model to identify a higher number of clusters and manually aggregating these clusters based on context after the fact was a valid approach.

[12] The code for this step is captured in the "gamma_difference2" function in the Appendix.

Figure 2: Most Common Words in Cluster Assignments

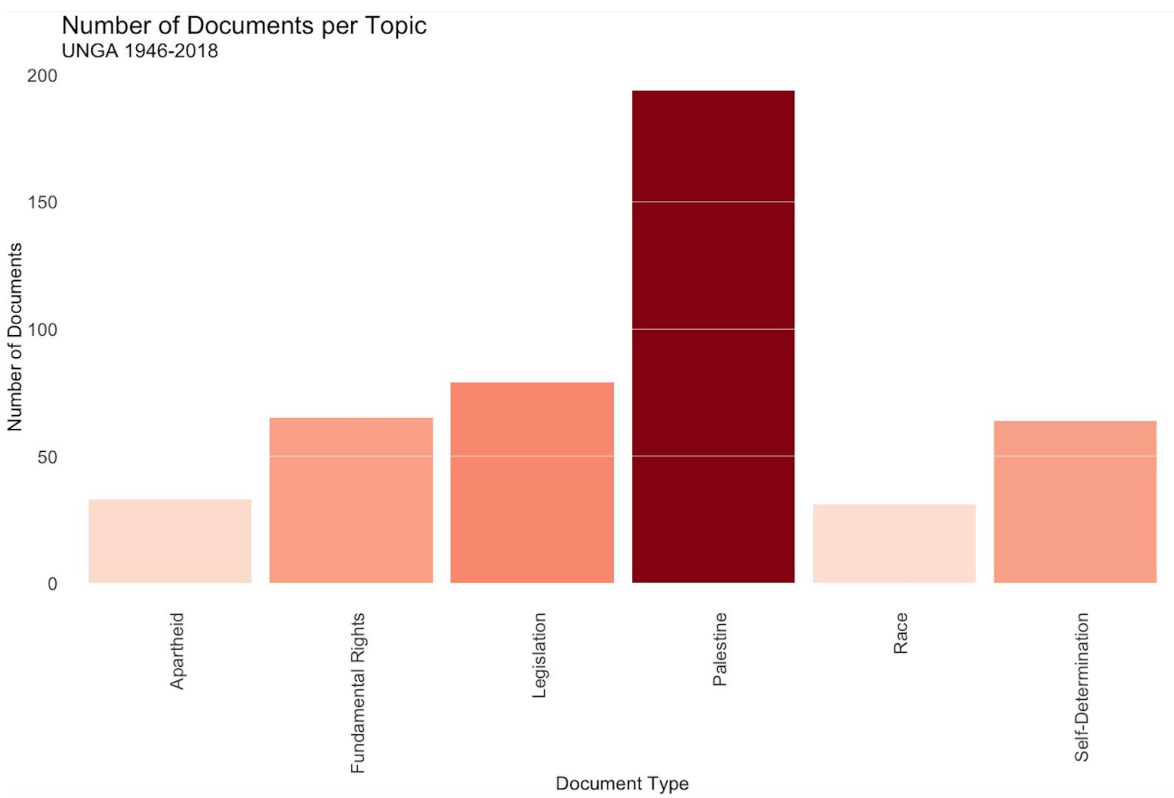


As expected, we see some clusters identifying specific groups of highly correlated documents—for example, cluster 14 seems to collect proposals condemning the South African apartheid regime, and cluster 5 captures proposals linked to issues of racial discrimination. Other clusters, however, do not appear to be entirely unique and can be manually aggregated for further analysis—for example, clusters 2, 7, 8, 9, and 11 all appear to pertain to the Israeli-Palestinian conflict, and we choose to aggregate these for further examination.

Findings

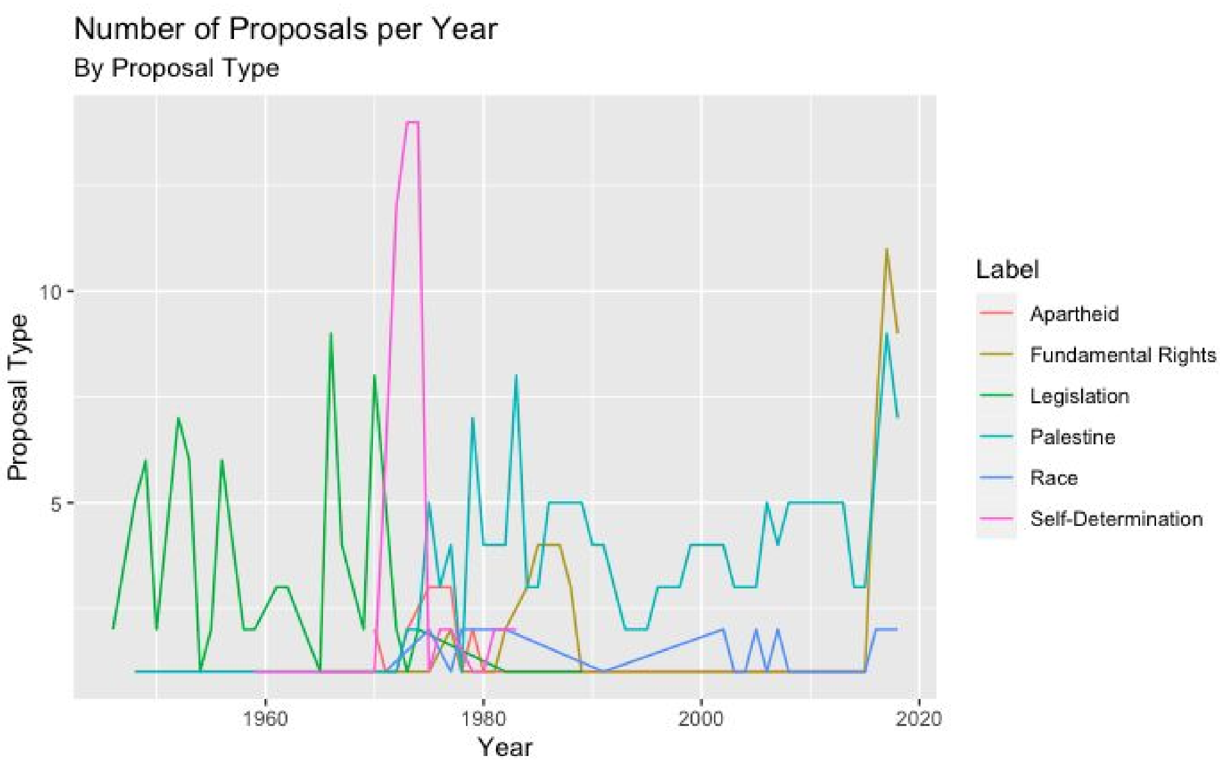
After assigning and aggregating the clusters with their appropriate topic labelings, we examine the distribution of the documents across the different topics. In Figure 3 below, we plot the number of documents that have been classified under each topic using our model.

Figure 3: Number of Documents Per Topic



From the figure, while we see that the number of documents identified as pertaining to the *Israeli-Palestinian conflict* remains a large percentage, our model identifies proposals related to *Legislation* as the second most popular issue area. However, in order to not draw any conclusions based on just the overall number of documents and rather be able to track the changes in vote patterns across the years, we break down the distribution of documents at the year level. In Figure 4 below, we visualize how the number of documents classified for each issue area.

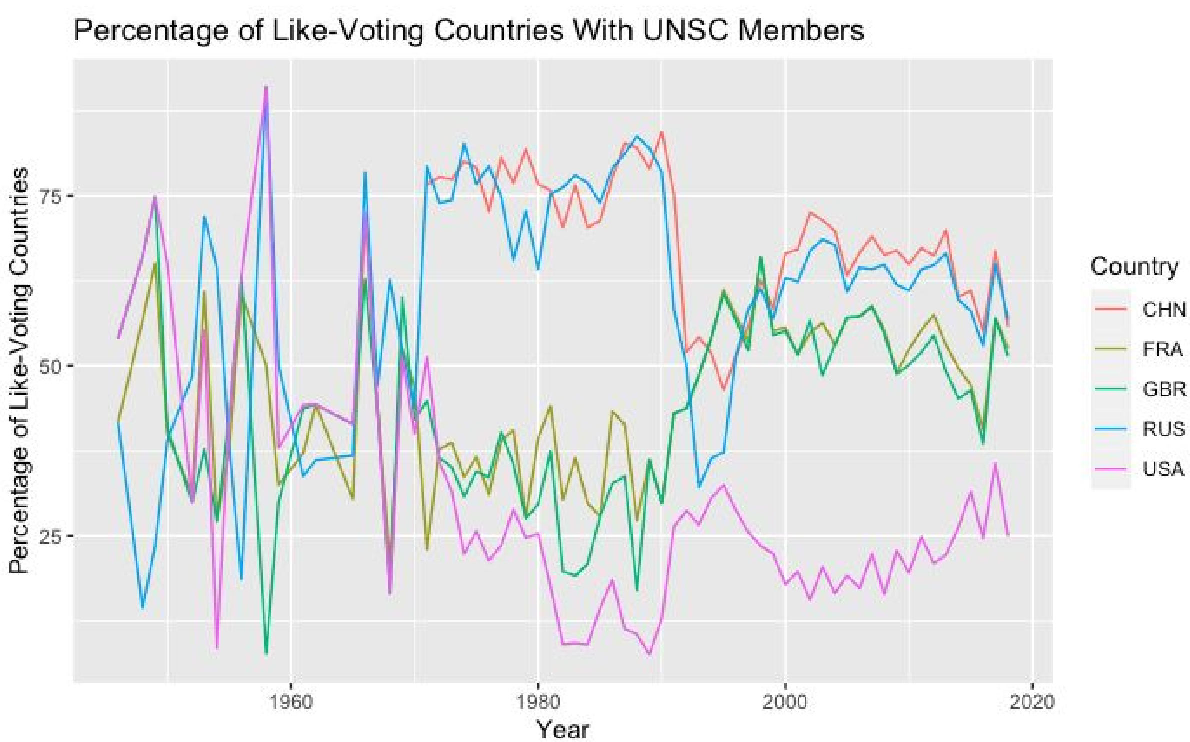
Figure 4: Number of Documents per Proposal Types



Upon further inspection, we see that the number of documents classified within each area is not distributed uniformly over time. Notably, the overwhelming majority of documents allocated to the *Legislation* topic is concentrated in the period at the beginning of the dataset, from 1946 until about 1975. The documents that are grouped under the *Israeli-Palestinian conflict* make up a large portion of the documents of the period starting after 1972. It is important to note the relative time frames as we transition into analyzing leaders on certain human rights issues as well as specific issue areas identified by our model.

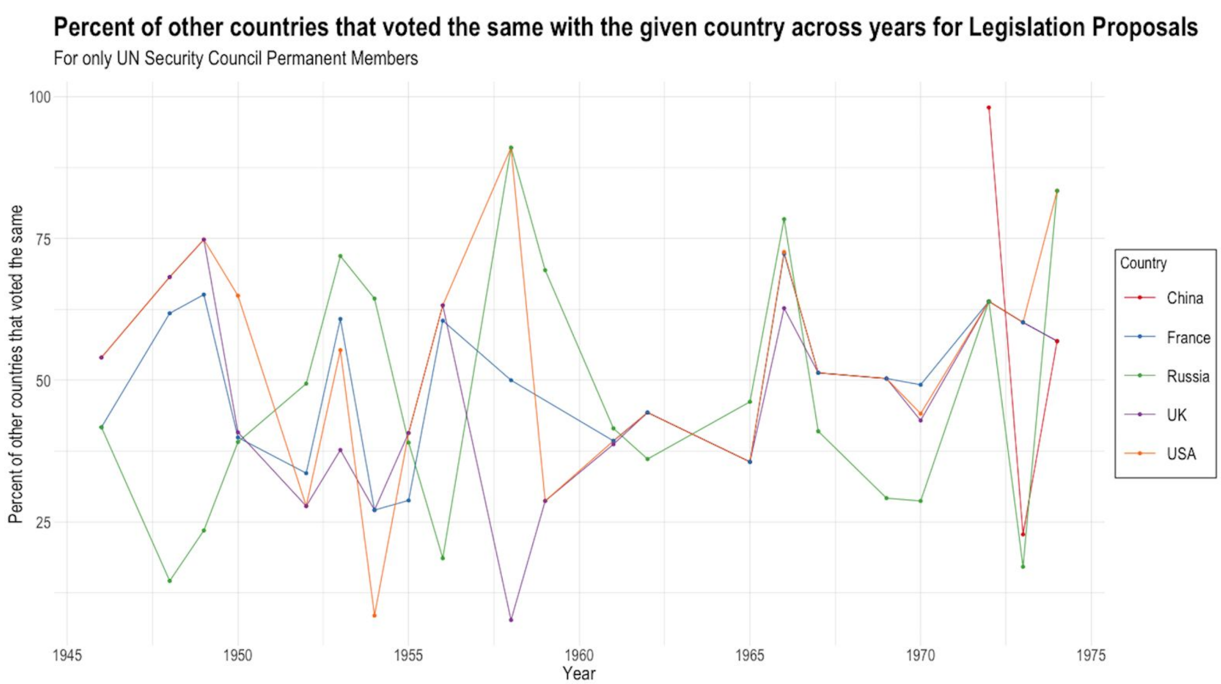
For the sake of parsimony, in terms of vote leaders, we exclusively focus on the permanent members of the UN Security Council (UNSC) consisting of China, France, Russia, United Kingdom and United States of America. In Figure 5 below, we start by first demonstrating how the relative order of vote leaders change in between these countries across the entire time span of the dataset by aggregating all the proposals.

Figure 5: Percentage of Like-Voting Countries With UNSC Members



In order to examine how the voting patterns and vote leaders' positioning change among these states for the specific types of issues we have identified with the model, we focus again on the top two issue areas: Israeli-Palestine Conflict and Legislation. In Figure 6 below, we illustrate how the vote leader changes among the permanent members of the UNSC over the course of 1946-1975 with regards to proposals on Legislation.

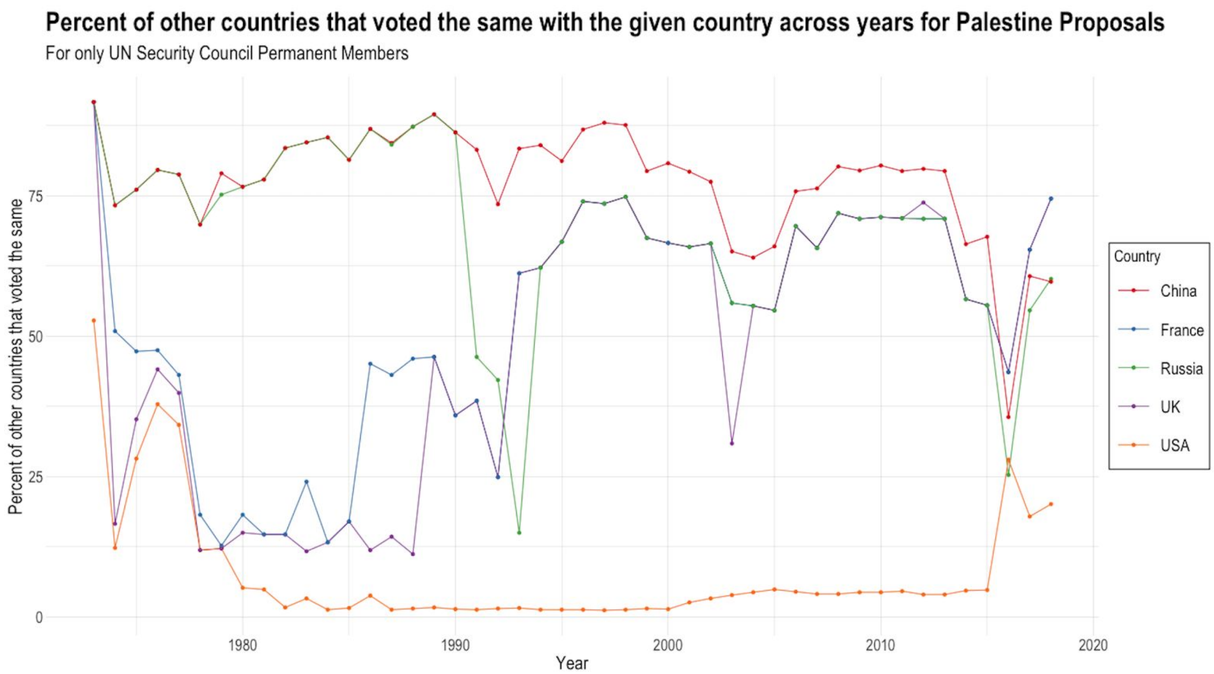
Figure 6: Like-Voting On Legislation Proposals



In this figure, we see that up until 1960, there is no clear vote leader. Over this period, the concept of human rights was not well-established; the legislation proposals sought to establish the existence of human rights as a legal principle. During the 1960s, we begin to see some correlation in vote leadership, suggesting international cooperation in this topic area—though the conventional view of the period was one of heightened international discord.

We apply the same procedure to proposals belonging to the Israel-Palestine Conflict category, and focus on the period from 1972 onwards to elucidate how the trends in voting as well as vote leaders exhibit deviation within votes on issues related to Israeli-Palestinian conflict in Figure 7 below.

Figure 7: Like-Voting On Palestine Proposals



Like-voting patterns related to Israeli-Palestinian conflict most clearly show a story of United States' divergence. Beginning in 1980, the United States dropped well below the other like-voting percentages of other UNSC members. The percentage of other countries that voted the same as the United States remained close to zero from 1980 until 2000. A recent increase in like voting with United States on the Israeli-Palestinian conflict has changed the ordinal distrubution of UNSC members, as Russia has recently garnered the least like-votes on the issue. Another interesting finding concerns Russia and China. Despite widespread allegations of human rights violations against both countries, we see that the two states have emerged as vote leaders on this issue, meaning most states agree with their positions.

Conclusion

In this paper, we provided an empirical basis for the existence of vote leaders on human rights issues in IOs. Using a corpus of draft resolutions from the UNGA, we conducted unsupervised topic modeling on 978 human rights proposals to identify 19 proposal clusters. To enhance interpretability, we aggregated these 19 clusters into six issue areas. As a plausibility probe, we closely investigated two of these issue areas: legislation proposals and the Israeli-Palestinian conflict, finding consensus between UNSC members on legislation proposals and divergent voting behavior by the United States on human rights proposals surrounding the Israeli-Palestinian conflict.

We identify three ways to expand the approach. First, unsupervised topic models can allow us to better understand UNGA resolutions outside the human rights scope. Using a similar technique it is possible to filter the resolution dataset for resolutions pertaining to economic development, colonialism, arms control and disarmament, or nuclear weapons. Second, similar analyses could stand to evaluate international organizations other than the UNGA. Potential candidates include analyzing different corpuses from the World Trade Organization, or International Monetary Fund. Finally, beyond the scope of international organizations entirely, the techniques used in this paper can be used for different legislative chambers. Here, the United States Congress represents a particularly interesting area of potential extension. Ultimately, as world events continue to demonstrate the need for political organizations, it is increasingly important that we understand how these organizations are structured.

Machine Learning Final Project Appendix

Pete Cuppernull, Ipek Cinar

03/11/2020

```
knitr::opts_chunk$set(echo = TRUE)
library(tidyverse)
library(topicmodels)
library(tidytext)
library(tm)
library(gganimate)
library(gifski)
library(seriation)
library(clValid)
library(transformr)
library(viridis)
library(ggplot2)
library(gganimate)
library(babynames)
library(hrbrthemes)
library(viridis)
set.seed(1414)
```

Data Preprocessing

We start by importing the data:

```
getwd()
data <- read_csv("~/Desktop/Intro to Machine Learning/Project/UNdata.csv")
data_clean <- data %>%
  filter(hr == 1,
         member == 1,
         vote != 9) %>%
  select(-amend, -me, -nu, -di, -hr, -co, -ec, -(Meeting:Related.Documents))
```

We proceed with creating a dataframe with one row per proposal:

```
unique_proposals <- data_clean %>%
  distinct(resid, .keep_all = TRUE)

unique_proposals %>%
  group_by(year) %>%
  summarize(n = n())
```

Research Design:

We then go ahead and create topic model for UN Proposals:

```
##Create DTM
proposal_tokens <- unique_proposals %>%
   unnest_tokens(output = word, input = descr) %>%
```

```r
  # remove numbers
  filter(!str_detect(word, "^[0-9]*$")) %>%
  # remove stop words
  anti_join(stop_words) %>%
  # stem the words
  mutate(word = SnowballC::wordStem(word))

proposal_dtm <- proposal_tokens %>%
# get count of each token in each document
count(resid, word) %>%
# create a document-term matrix with all features and tf weighting
cast_dtm(document = resid, term = word, value = n)

#test out sparsity level (remove words that only appear in a small # of docs)
proposal_dtm <- removeSparseTerms(proposal_dtm, sparse = .99)

#Find the sum of words in each Document
rowTotals <- apply(proposal_dtm , 1, sum)
#get rid of proposals qwith no words (maybe errors in the db or something)
proposal_dtm.new   <- proposal_dtm[rowTotals> 0, ]

##DTM is done, lets check out some topic models now
proposal_lda <- LDA(proposal_dtm.new, k = 10, control = list(seed = 123))

proposal_topics <- tidy(proposal_lda, matrix = "beta")
proposal_topics

proposal_terms <- proposal_topics %>%
  group_by(topic) %>%
  top_n(20, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)

proposal_terms

proposal_chart_v1 <- proposal_terms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
```

We also try to tune the model a little bit so see how good we can make the document classifier. The main ways we can tune this is by tweaking the sparsity (i.e. how many documents does a given word have to be in in order to have it count towards informing the classifier) and the number of classes the classifier is dividing documents up into (k).

Tuning the Model:

```r
#doing a gamma matrix will tell us the score that each document gets
# for each class - i.e. it will give a higher score to classes
# it is more likely to fall into, with the highest score
# becuase the class it gets sent to
td_lda_docs <- tidy(proposal_lda, matrix = "gamma")
```

```r
  td_lda_docs

  doc_classes <- td_lda_docs %>%
    group_by(document) %>%
    top_n(1) %>%
    ungroup()

# Let's write a couple functions to try out tons of combinations
# of sparsity and k to determine the ideal tuning parameters -
# we'll measure this by the difference between the top two gammas
# for each doc (i.e. how confident is the classifier that a given
# doc should be in the top rated class vs the next best rated class)
#write code to do this gamma difference scoring
    doc_classes <- td_lda_docs %>%
    group_by(document) %>%
    top_n(2) %>%
    ungroup()

gamma_difference <- doc_classes %>%
      group_by(document) %>%
      mutate(id = row_number()) %>%
      select(-topic) %>%
      #get gammas of top two classes
      spread(key = id, value = gamma) %>%
      #find difference between top two classes
      mutate(gamma_dif = abs(`1`-`2`)) %>%
      ungroup() %>%
      #take the mean of the gamma differences to provide
      #a single score for the whole model
      summarize(gamma_dif = mean(gamma_dif))

#now write function to test out different levels of sparsity
#and k -- this will take one argument of sparcity and spit out
#the gamma differences for k = 2:20 for that level of sparsity


gamma_difference2 <- function(sparsity) {
  #same code as above
  proposal_dtm <- removeSparseTerms(proposal_dtm, sparse = sparsity)

#Find the sum of words in each Document
rowTotals <- apply(proposal_dtm , 1, sum)
proposal_dtm.new   <- proposal_dtm[rowTotals> 0, ]

#create a function which finds the gamma difference like
#we did above -- this takes k as an argument, and we'll
#map over a range of k's below
gamma_mapper <- function(k){
  proposal_lda <- LDA(proposal_dtm.new, k = k, control = list(seed = 123))

  td_lda_docs <- tidy(proposal_lda, matrix = "gamma")

  doc_classes <- td_lda_docs %>%
```

```r
    group_by(document) %>%
    top_n(2) %>%
    ungroup()

  gamma_difference <- doc_classes %>%
      arrange(document) %>% #order by docs
      group_by(document) %>%
      mutate(id = row_number()) %>%
      select(-topic) %>%
      spread(key = id, value = gamma) %>%
      mutate(gamma_dif = abs(`1`-`2`)) %>%
      ungroup() %>%
      summarize(gamma_dif = mean(gamma_dif))

  gamma_difference
}

#now take that function we just wrote and map over k 2:20
map_dfr(2:20, gamma_mapper, .id="id") %>%
  #create id column so we know which gamma difference
  # corresponds to which value of k
  mutate(id = as.numeric(id) + 1) %>%
  rename(number_of_classes = id)
}

#now run this over sparsity values of .9-.99 --
# this will take a few minutes to run because
# its doing 190 different classifiers

model_iterations <- map_dfr(seq(.90, .99, .01), gamma_difference2,
                            .id = "sparsity") %>%
  mutate(sparsity = as.numeric(sparsity)) %>%
  mutate(sparsity = if_else(sparsity == 1, .90,
                    if_else(sparsity == 2, .91,
                    if_else(sparsity == 3, .92,
                    if_else(sparsity == 4, .93,
                    if_else(sparsity == 5, .94,
                    if_else(sparsity == 6, .95,
                    if_else(sparsity == 7, .96,
                    if_else(sparsity == 8, .97,
                    if_else(sparsity == 9, .98,
                    if_else(sparsity == 10, .99, 99))))))))))))

model_iterations <- model_iterations %>%
  arrange(desc(gamma_dif))

##Here, we can see that sparcity of .99  generally works
# the best -- I'm going to pick the top few models and
# check out the results in a table.

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

#write a quick function to plugin in sparsity and k to view the model,
```

```r
view_model <- function(sparsity, k){
  proposal_dtm <- removeSparseTerms(proposal_dtm, sparse = sparsity)

#Find the sum of words in each Document
rowTotals <- apply(proposal_dtm , 1, sum)
proposal_dtm.new   <- proposal_dtm[rowTotals> 0, ]

##DTM is done, lets check out some topic models now
proposal_lda <- LDA(proposal_dtm.new, k = k, control = list(seed = 123))

proposal_topics <- tidy(proposal_lda, matrix = "beta")

proposal_terms <- proposal_topics %>%
  group_by(topic) %>%
  top_n(20, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
proposal_terms %>%
  mutate(term = reorder(term, beta)) %>%
  ggplot(aes(term, beta, fill = factor(topic))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ topic, scales = "free") +
  coord_flip()
}

##Top 6 Models
model_iterations %>%
  head()

#mod 1
view_model(.99, 5)
##Of these five topics, #1 seems to be general HR law adoptions,
# 2-4 seems to revolve around Palestine, and 5 seems
# to be anti-racism stuff

#mod 2
view_model(.99, 7)
##Seems like there are 4 here: #1 and 4 are general HR adoptions,
# 2-3 are palestine, 5 is racism, 6 has something about
# globalization (72b?), 7 not sure

#mod 3
view_model(.99, 6)
##1 is general adoptions, 2-3 palestine, 4 is maybe anti-colonial
# stuff?, 5 is racism, 6 unsure

#mod 4
view_model(.99, 10)
##here we have palestine stuff, maybe one talking about rights of
# displaced people (3), racism

#mod 5
view_model(.99, 9)
```

```
##pretty much ony palestine and racism

#mod 6
view_model(.97, 3)
##nothing

##I think the gamma difference process might select against models
# with higher k - I'm going to filter this again with just k between 15 and 20.
model_iterations %>%
  filter(number_of_classes > 15) %>%
  head(3)

#mod7
view_model(.99, 19)
##This is pretty cool, we get the normal categories plus ones
# for natural resources and condemnations of South African apartheid

#mod8
view_model(.99, 16)
##not quite as cool, pretty noisey

#mod7
view_model(.99, 20)
##you get the south african stuff again but this feels a little noisier

# Mod7 looks cool!
```

We then merge Topic Classifications back to main dataframe (selecting the model with 0.99 sparsity and 19 topics):

```
##create simple df of document ID and final class --
# using sparsity .99 and 19 terms
proposal_dtm <- removeSparseTerms(proposal_dtm, sparse = .99)

rowTotals <- apply(proposal_dtm , 1, sum)
proposal_dtm.new   <- proposal_dtm[rowTotals> 0, ]

proposal_lda <- LDA(proposal_dtm.new, k = 19, control = list(seed = 123))

td_lda_docs <- tidy(proposal_lda, matrix = "gamma")

doc_classes_final <- td_lda_docs %>%
    group_by(document) %>%
    top_n(1) %>%
    ungroup() %>%
    mutate(document = as.numeric(document))

#Manually Assign Groupings
doc_classes_final$group[doc_classes_final$topic == 1] <- "general"
doc_classes_final$group[doc_classes_final$topic == 2] <- "palestine"
doc_classes_final$group[doc_classes_final$topic == 3] <- "general"
doc_classes_final$group[doc_classes_final$topic == 4] <- "palestine"
doc_classes_final$group[doc_classes_final$topic == 5] <- "race"
doc_classes_final$group[doc_classes_final$topic == 6] <- "general"
```

```r
doc_classes_final$group[doc_classes_final$topic == 7] <- "palestine"
doc_classes_final$group[doc_classes_final$topic == 8] <- "palestine"
doc_classes_final$group[doc_classes_final$topic == 9] <-  "unsure"
doc_classes_final$group[doc_classes_final$topic == 10] <- "general"
doc_classes_final$group[doc_classes_final$topic == 11] <- "palestine"
doc_classes_final$group[doc_classes_final$topic == 12] <- "self-determination"
doc_classes_final$group[doc_classes_final$topic == 13] <-  "general"
doc_classes_final$group[doc_classes_final$topic == 14] <- "apartheid"
doc_classes_final$group[doc_classes_final$topic == 15] <- "unsure"
doc_classes_final$group[doc_classes_final$topic == 16] <- "legislation"
doc_classes_final$group[doc_classes_final$topic == 17] <-  "unsure"
doc_classes_final$group[doc_classes_final$topic == 18] <- "unsure"
doc_classes_final$group[doc_classes_final$topic == 19] <-  "fundamental rights"

#join back to main df
data_clean_groups <- data_clean %>%
  left_join(doc_classes_final, by = c("resid" = "document"))

#make sure we didnt miss any
data_clean_groups %>%
  filter(is.na(group))
```

Dissimilarity Plots:

```r
#Prep and Create Distance Matrix
new_proposal_dtm <- removeSparseTerms(proposal_dtm, sparse = .99)

prop_matrix <- as.matrix(new_proposal_dtm)

pop_scaled <- as.data.frame(prop_matrix) %>%
  rownames_to_column("doc_id") %>%
  select(-doc_id) %>%
  scale()

pop_scaled2 <- pop_scaled[sample(seq_len(nrow(pop_scaled))),]


predictor_distance <- dist(as.matrix(pop_scaled2 ), method = "euclidean")

#Three diss plot type that shoed some stratification
dissplot(predictor_distance,
         method = "HC_single",
         options = list(main = "Dissimilarity plot: HC Single"))

dissplot(predictor_distance,
         method = "HC_complete",
         options = list(main = "Dissimilarity plot: HC Complete"))

dissplot(predictor_distance,
         method = "GW",
         options = list(main = "Dissimilarity plot: GW"))
```

K-means for clustering:

```r
#prep matrix
matrix <- as.matrix(proposal_dtm.new)
#scale columns
matrix <- scale(matrix)
#run
kmeans <- kmeans(matrix,
                 centers = 10,
                 nstart = 15)
#clean up results a bit
t <- as.table(kmeans$cluster)
t <- data.frame(t)
colnames(t)[colnames(t)=="Freq"] <- "Assignment"
colnames(t)[colnames(t)=="Var1"] <- "Proposal_id"


t #this gives us the class assignments

#check out number of proposals per class - it seems like
# nearly half ar ein one group -- possibly problematic,
t %>%
  count(Assignment) #vast majority are in one class, prob not good

#now umping through some hoops to manually create a beta matrix,
#like we did with the automatic LDA classifier earlier on
dtm_df <- as.data.frame(as.matrix(proposal_dtm.new))

dtm_df2 <- cbind(t, dtm_df) %>%
  select(-Proposal_id)


create_row <- function(x){
  dtm_df2 %>%
  filter(Assignment == x) %>%
  #select(-Proposal_id) %>%
  colSums()
}

c1 <- create_row(1)
c2 <- create_row(2)
c3 <- create_row(3)
c4 <- create_row(4)
c5 <- create_row(5)
c6 <- create_row(6)
c7 <- create_row(7)
c8 <- create_row(8)
c9 <- create_row(9)
c10 <- create_row(10)


kmeans_beta <- as.data.frame(rbind(c1, c2, c3, c4, c5, c6,
                                   c7, c8, c9, c10))
```

```r
kmeans_beta <- cbind(1:10, kmeans_beta) %>%
  select(-Assignment) %>%
  rename(assignment = `1:10`)

#created beta matrix, now jump down a bit to do the same cgart
# of the most frequent words per class, like we did above


#this give us the words that tend to appear in only one category
# - we shoulmdnt use this because it is biased towards showing
# infrequent qwords, ust leaving the code in here for now though

#kmeans_topic_words_unique <- kmeans_beta %>%
#  gather(key = term,
#         value = n,
#         -assignment) %>%
#  group_by(term) %>%
#  mutate(percent = n/sum(n)) %>%
#  ungroup() %>%
#  group_by(assignment) %>%
#  top_n(10, percent) %>%
#  ungroup() %>%
#  arrange(assignment, -percent)

#we should use this, which shows the most popular words per class
kmeans_topic_words <- kmeans_beta %>%
  gather(key = term,
         value = n,
         -assignment) %>%
  group_by(term) %>%
  ungroup() %>%
  group_by(assignment) %>%
  top_n(6, n) %>%
  ungroup() %>%
  arrange(assignment, -n)

kmeans_topic_words %>%
  mutate(term = reorder(term, n)) %>%
  ggplot(aes(term, n, fill = factor(assignment))) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ assignment, scales = "free") +
  coord_flip()
```

So this is working (we can see South Africa stuff, Palestine, etc), but the clusters seems to be pretty noisey. Let's run some silhouette width validation to see what the optimum number of clusters are (this is basically copy and pasted from the last problem set).

```r
kmeans_valid <- clValid(as.matrix(matrix),
        nClust = 2:15,
        clMethods = "kmeans",
        validation = "internal")

kmeans_measures <- measures(kmeans_valid)
```

```r
km_df <- as.data.frame(kmeans_measures)
colnames(km_df) <- c("2", "3", "4", "5", "6", "7","8", "9",
                     "10", "11", "12", "13", "14", "15")

km_final_df <- km_df %>%
  rownames_to_column() %>%
  gather(key = "clusters",
         value = "score",
         2:15) %>%
  mutate(model = "kmeans")

km_final_df %>%
  filter(rowname == "Silhouette")
```

We can see that all of the Silhouette widths are pretty low – this is somewhat expected because the cluster densities will be lower given the high number of predictors. While this does little to help us choose an optimum number of clusters, it does highlight the importance of manually reviewing the clustering results and aggregating smaller subtopics after the fact.

Findings

1) Number of Documents per each Topic classification:

We start by plotting the number of documents that have been classified under each topic using the algorithm. The static plot looks as the following:

```r
doc_classes_summary <- doc_classes_final %>%
  group_by(group) %>% summarise(count = n())

# Filtering out unsure and general as document types:
# unique(doc_classes_final$group)
doc_classes_summary <- subset(doc_classes_summary,
                              group!="general" & group!="unsure")
doc_classes_summary$group <- str_to_title(doc_classes_summary$group)

# Static: Plotting number of documents per topic across
# entire time range of the dataset (1946-2018)
ggplot(doc_classes_summary, aes(x=reorder(group, -count),
                 y=count,fill=rownames(doc_classes_summary))) +
  geom_bar(stat = "identity") +
  guides(fill=FALSE)+
  coord_flip() +
  theme_bw() + labs(x="Topic Name", y="Number of Documents") +
  ylim(0,max(doc_classes_summary$count)) +
  ggtitle("Number of Documents per Topic", subtitle = "UNGA 1946-2018")
#ggsave("~/Desktop/Intro to Machine Learning/Project/Final Visuals/document_classes_barplot.png", width
```

We can also show it through a cool animation by integrating transitions between topics:

```r
# Animated: Plotting number of documents per topic across
# entire time range of the dataset (1946-2018)
p <- ggplot(doc_classes_summary, aes(group, count, fill = count)) +
  geom_col() +
  scale_fill_distiller(palette = "Reds", direction = 1) +
```

```
  labs(x = "Document Type", y = "Number of Documents",
       title = "Number of Documents per Topic",
       subtitle = "UNGA 1946-2018") +
  guides(fill = FALSE) +
  theme_minimal() +
  theme(axis.text.x=element_text(angle=90,hjust=1, size = 23),
        axis.text.y = element_text(size = 23),
        axis.title = element_text(size = 24),
        plot.title = element_text(size=32),
        plot.subtitle = element_text(size=25),
    panel.grid = element_blank(),
    panel.grid.major.y = element_line(color = "white"),
    panel.ontop = TRUE,
  )
p <- p + transition_states(group, wrap = FALSE) +
  shadow_mark() +
  enter_grow() +
  enter_fade()
#animate(p,100,fps = 10,duration = 10, width = 1500, height = 1000,
#        renderer = gifski_renderer("~/Desktop/Intro to Machine Learning/Project/Final Visuals/anim_doc
```

And this is another cool animation to plot how the number of documents per topic change with each year:

```
data_groups_overyears <- data_clean_groups %>%
  distinct(resid, .keep_all = TRUE) %>%
  group_by(year, group) %>%
  summarize(n = n()) %>%
  filter(group %in% c("legislation", "palestine",
                      "self-determination", "fundamental rights",
                      "apartheid", "race"))
data_groups_overyears$group <- str_to_title(data_groups_overyears$group)

data_groups_overyears <- data_groups_overyears %>%
  group_by(year) %>%
  arrange(desc(n)) %>%
  mutate(rank = row_number())

# Animated barplot
static_plot<-ggplot(data_groups_overyears,
                    aes(rank,group=group,
                        fill=as.factor(group),
                        color=as.factor(group))) +
 geom_tile(aes(y = n/2,
 height = n,
 width = 0.9), alpha = 0.8, color = NA) +
 geom_text(aes(y = 0, label = paste(group, " ")),
           vjust = 0.2, hjust = 1, size = 8) +
 geom_text(aes(y=n,label = paste(" ",n)), hjust=0, size = 8)+
 coord_flip(clip = "off", expand = TRUE) +
 scale_x_reverse() +
 guides(color = FALSE, fill = FALSE) +
 theme_minimal() +
 theme(
 plot.title=element_text(size=25, hjust=0.5, face="bold",
```

```
                         colour="grey", vjust=-1),
 plot.subtitle=element_text(size=18, hjust=0.5, face="italic",
                            color="grey"),
 plot.caption =element_text(size=8, hjust=0.5, face="italic",
                            color="grey"),
 axis.ticks.y = element_blank(),
 axis.text.y = element_blank(),
 plot.margin = margin(1,1,1,4, "cm")
 )

plt<-static_plot + transition_states(states = year,
                                     transition_length = 4,
                                     state_length = 1) +
 ease_aes("cubic-in-out") +
 #view_follow(fixed_x = TRUE) +
 labs(title = "Volume of Proposals per Type per Year: {closest_state}",
 subtitle = "UNGA 1946-2018",
 #caption = "Data Source: World Bank Data",
 x="",y="Number of Documents") +
  theme(axis.text = element_text(size = 16),
        axis.title = element_text(size = 18))
animate(plt,100,fps = 20,duration = 30, width = 1500, height = 800,
        renderer = gifski_renderer("~/Desktop/Intro to Machine Learning/Project/Final Visuals/anim_barpl
```

2) Vote leaders:

For vote leaders, we are interested in seeing states that are the "leaders" on given topics over time by ranking the percentage of other states that vote in accordance with that state across UN proposals.

We start by constructing Voting Network for proposal type by year:

```
#data_clean_groups <- read_csv("~/Desktop/df.csv")

#Calculate percent of other countries that voted with them
##ALL PROPOSAL TYPES
voting_percentages <- data_clean_groups %>%
  select(X1, ccode, vote, Country, year, resid, group) %>%
  filter(vote != 8) %>%
  mutate(vote = as.factor(vote)) %>%
  group_by(resid) %>%
  add_count(vote) %>%
  mutate(percent = n/sum(unique(n))) %>%
  ungroup() %>%
  group_by(ccode, Country, year) %>%
  summarize(popular_vote = mean(percent)) %>%
  mutate(perm_sc_member = if_else(ccode %in% c(2, 200,
                                              220, 365, 710), 1, 0)) %>%
  ungroup()

# Dataframe for UN Security Council permanent members only:
voting_percent_sc <- voting_percentages %>%
  filter(perm_sc_member == 1) %>%
  group_by(year) %>%
```

```r
  arrange(desc(popular_vote)) %>%
  mutate(rank = row_number()) %>%
  mutate(popular_vote = round(popular_vote*100, 1)) %>%
  mutate(Country_fullname = ifelse(Country == "USA", "USA",
                            ifelse(Country == "CHN", "China",
                            ifelse(Country == "GBR", "UK",
                            ifelse(Country == "RUS", "Russia",
                            ifelse(Country == "FRA", "France", NA))))))

## Proposal Type - Breakdown
voting_percentages_groups <- data_clean_groups %>%
  select(X1, ccode, vote, Country, year, resid, group) %>%
  filter(vote != 8) %>%
  mutate(vote = as.factor(vote)) %>%
  group_by(resid) %>%
  add_count(vote) %>%
  mutate(percent = n/sum(unique(n))) %>%
  ungroup() %>%
  group_by(ccode, Country, year, group) %>%
  summarize(popular_vote = mean(percent)) %>%
  mutate(perm_sc_member = if_else(ccode %in% c(2, 200,
                                               220, 365, 710), 1, 0)) %>%
  ungroup()

# Test out vote leader for Palestine proposals - all members
voting_percent_palestine <- voting_percentages_groups %>%
  # Filtering for years after 1972 since most documents
  # for Palestine are concentrated there:
  filter(year > 1972) %>%
  filter(group == "palestine") %>%
  group_by(year) %>%
  arrange(desc(popular_vote)) %>%
  mutate(rank = row_number()) %>%
  top_n(10, desc(rank)) %>%
  mutate(popular_vote = round(popular_vote*100, 1))

# Dataframe for UN Security Council permanent members only:
# Test out vote leaders for Palestine proposals - only SC members
voting_percent_palestine_sc <- voting_percentages_groups %>%
  filter(perm_sc_member == 1) %>%
  # Filtering for years after 1972 since most documents
  # for Palestine are concentrated there:
  filter(year > 1972) %>%
  filter(group == "palestine") %>%
  group_by(year) %>%
  arrange(desc(popular_vote)) %>%
  mutate(rank = row_number()) %>%
  mutate(popular_vote = round(popular_vote*100, 1)) %>%
  mutate(Country_fullname = ifelse(Country == "USA", "USA",
                            ifelse(Country == "CHN", "China",
                            ifelse(Country == "GBR", "UK",
                            ifelse(Country == "RUS", "Russia",
                            ifelse(Country == "FRA", "France", NA))))))
```

```r
# Test out vote leader for legislation proposals - all members
voting_percent_legislation <- voting_percentages_groups %>%
  # Filtering for years up until 1975 since most documents
  # for Legislation are concentrated there:
  filter(year < 1976) %>%
  filter(group == "legislation") %>%
  group_by(year) %>%
  arrange(desc(popular_vote)) %>%
  mutate(rank = row_number()) %>%
  top_n(10, desc(rank)) %>%
  mutate(popular_vote = round(popular_vote*100, 1))

# Dataframe for UN Security Council permanent members only:
# Test out vote leaders for Palestine proposals - only SC members
voting_percent_legislation_sc <- voting_percentages_groups %>%
  filter(perm_sc_member == 1) %>%
  # Filtering for years up until 1975 since most documents
  # for Legislation are concentrated there:
  filter(year < 1976) %>%
  filter(group == "legislation") %>%
  group_by(year) %>%
  arrange(desc(popular_vote)) %>%
  mutate(rank = row_number()) %>%
  mutate(popular_vote = round(popular_vote*100, 1)) %>%
  mutate(Country_fullname = ifelse(Country == "USA", "USA",
                            ifelse(Country == "CHN", "China",
                            ifelse(Country == "GBR", "UK",
                            ifelse(Country == "RUS", "Russia",
                            ifelse(Country == "FRA", "France", NA))))))

#voting_percent_palestine_sc %>%
#  arrange(year)

#voting_percent_sc %>%
#  arrange(year)
```

We first plot how the vote leader changes within the UN Security Council Permanent Members (only) over the years 1946-2018 by aggregating all the proposals:

```r
# The code velow benefited greatly from the following link:
# https://towardsdatascience.com/how-to-create-plots-with-beautiful-animation-using-gganimate-912f4279b
static_plot<-ggplot(voting_percent_sc,
                    aes(rank,group=Country_fullname,
                        fill=as.factor(Country_fullname),
                        color=as.factor(Country_fullname))) +
 geom_tile(aes(y = popular_vote/2,
 height = popular_vote,
 width = 0.9), alpha = 0.8, color = NA) +
 geom_text(aes(y = 0, label = paste(Country_fullname, " ")),
          vjust = 0.2, hjust = 1, size = 8) +
 geom_text(aes(y=popular_vote,label = paste(" ",popular_vote)),
          hjust=0, size = 8)+
 coord_flip(clip = "off", expand = TRUE) +
 #scale_y_continuous(labels = scales::comma) +
```

```r
  scale_x_reverse() +
  guides(color = FALSE, fill = FALSE) +
  theme_minimal() +
  theme(
  plot.title=element_text(size=25, hjust=0.5, face="bold",
                          colour="grey", vjust=-1),
  plot.subtitle=element_text(size=18, hjust=0.5, face="italic",
                             color="grey"),
  plot.caption =element_text(size=8, hjust=0.5, face="italic",
                             color="grey"),
  axis.ticks.y = element_blank(),
  axis.text.y = element_blank(),
  plot.margin = margin(1,1,1,4, "cm")
  )

plt<-static_plot + transition_states(states = year,
                                     transition_length = 4,
                                     state_length = 1) +
  ease_aes("cubic-in-out") +
  #view_follow(fixed_x = TRUE) +
  labs(title = "Percent of other countries that voted the same with given country: {closest_state}",
  subtitle = "For UN Security Council Permanent Members only",
  #caption = "Data Source: World Bank Data",
  x="",y="Percent of other countries that voted the same") +
   theme(axis.text = element_text(size = 18),
         axis.title = element_text(size = 20))
#animate(plt,100,fps = 20,duration = 30, width = 1200, height = 800,
        renderer = gifski_renderer("~/Desktop/Intro to Machine Learning/Project/Final Visuals/anim_barpl
```

We can also visualize the grpah above with a line plot:

```r
animate_permsc_lp_all <- voting_percent_sc %>%
  ggplot(aes(x=year, y=popular_vote, group = Country_fullname,
             color = Country_fullname)) +
  geom_line() +
    geom_point(aes(group=seq_along(year))) +
    scale_color_brewer(palette = "Set1") +
    ggtitle("Percent of other countries that voted the same with the given country across years for All
    theme_ipsum(plot_title_size = 30, subtitle_size = 22,
                axis_title_size = 21, axis_text_size = 18,
                axis_title_just = "mc") +
   theme(legend.text=element_text(size=19),
         legend.title = element_text(size = 18),
         legend.key.size = unit(3, "line"),
         legend.background = element_rect(size=0.5,
                                          linetype="solid", colour ="black")) +
    ylab("Percent of other countries that voted the same") + xlab("Year") +
  labs(color='Country') +
    transition_reveal(year)
#animate(animate_permsc_lp_all, width = 1400, height = 800,
        #renderer = gifski_renderer("~/Desktop/Intro to Machine Learning/Project/Final Visuals/anim_lin
```

We then plot how the vote leader changes within the UN Security Council Permanent Members (only) over the years 1972-2018 with regards to proposals relating to Palestine (using the classification we developed via topic modeling):

```
animate_permsc_lp_palestine <- voting_percent_palestine_sc %>%
  ggplot(aes(x=year, y=popular_vote, group = Country_fullname,
             color = Country_fullname)) +
  geom_line() +
    geom_point(aes(group=seq_along(year))) +
    scale_color_brewer(palette = "Set1") +
    ggtitle("Percent of other countries that voted the same with the given country across years for Pal
    theme_ipsum(plot_title_size = 30, subtitle_size = 22,
                axis_title_size = 21, axis_text_size = 18,
                axis_title_just = "mc") +
   theme(legend.text=element_text(size=19),
         legend.title = element_text(size = 18),
         legend.key.size = unit(3, "line"),
         legend.background = element_rect(size=0.5,
                                          linetype="solid", colour ="black")) +
    ylab("Percent of other countries that voted the same") + xlab("Year") +
  labs(color='Country') +
    transition_reveal(year)
#animate(animate_permsc_lp_palestine, width = 1400, height = 800,
        #renderer = gifski_renderer("~/Desktop/Intro to Machine Learning/Project/Final Visuals/anim_lin
```

We also plot how the vote leader changes within the UN Security Council Permanent Members (only) over the years 1946-1975 with regards to proposals relating to Legislation (using the classification we developed via topic modeling):

```
animate_permsc_lp_legislation <- voting_percent_legislation_sc %>%
  ggplot(aes(x=year, y=popular_vote,
             group = Country_fullname,
             color = Country_fullname)) +
  geom_line() +
    geom_point(aes(group=seq_along(year))) +
    scale_color_brewer(palette = "Set1") +
    ggtitle("Percent of other countries that voted the same with the given country across years for Leg
    theme_ipsum(plot_title_size = 30, subtitle_size = 22,
                axis_title_size = 21, axis_text_size = 18,
                axis_title_just = "mc") +
   theme(legend.text=element_text(size=19),
         legend.title = element_text(size = 18),
         legend.key.size = unit(3, "line"),
         legend.background = element_rect(size=0.5, linetype="solid",
                                          colour ="black")) +
    ylab("Percent of other countries that voted the same") +
  xlab("Year") +
  labs(color='Country') +
    transition_reveal(year)
#animate(animate_permsc_lp_legislation, width = 1400, height = 800,
        #renderer = gifski_renderer("~/Desktop/Intro to Machine Learning/Project/Final Visuals/anim_lin
```

Furthermore, in order to see how more developed versus less developed countries vote and see how all countries within the UN vote in general, we control for GDP and plot the percentage of yes votes given by counties first for all proposals:

```
# The code below benefited  from:
# https://towardsdatascience.com/animating-your-data-visualizations-like-a-boss-using-r-f94ae20843e3
```

```r
library(plotly)
library(gapminder)
library(countrycode)

# Adding the country codes to gapminder dataset in order to merge later with UN Data:
gapminder_new <- gapminder %>% mutate(countryabrv = countrycode(country,
                                                                "country.name",
                                                                "iso3c"))
# Reordering columns in gapminder:
gapminder_new <- gapminder_new %>%  dplyr::select("country",
                                                  "countryabrv",
                                                  "continent",
                                                  "year",
                                                  "pop",
                                                  "gdpPercap")

voting_yespercent_groups_country <- data_clean_groups %>%
  select(X1, ccode, vote, Country, year, resid, group) %>%
  filter(vote != 8) %>% group_by(year, Country) %>%
  summarise(totproposals = n(),
            totyes = sum(vote == 1),
            propyes = round(totyes/totproposals, digits = 2))
# Filtering columns in voting_percentages_groups:
voting_yespercent_groups_country <- voting_yespercent_groups_country %>%
  dplyr::select(Country, year, propyes)
# Renaming a column for merging purposes:
colnames(voting_yespercent_groups_country)[1] <- "countryabrv"

# Left joining voting_yespercent_groups_country onto gapminder,
# (since gapminder has less rows):
joined_df <- left_join(gapminder_new,
                       voting_yespercent_groups_country,
                       by = c("countryabrv", "year")) %>% na.omit()

# Plotting  percentage of yes votes versus GDP per capita for each year
fig <- joined_df %>%
  plot_ly(
    x = ~gdpPercap,
    y = ~propyes,
    size = ~pop,
    color = ~continent,
    frame = ~year,
    text = ~country,
    hoverinfo = "text",
    type = 'scatter',
    mode = 'markers'
  ) %>%
  layout(
    xaxis = list(
      type = "log", title = "GDP per Capita"
    ),
    yaxis = list(title = "Percantage of Yes votes (in a given year)")
  )
```

```
fig <- fig %>% layout(
    xaxis = list(
      type = "log"
    )
  )
fig <- fig %>% animation_opts(
    1000, easing = "elastic", redraw = FALSE
  )
fig <- fig %>% animation_button(
    x = 1, xanchor = "right", y = 0, yanchor = "bottom"
  )
fig <- fig %>% animation_slider(
    currentvalue = list(prefix = "YEAR ", font = list(color="red"))
  )
fig
#htmlwidgets::saveWidget(fig, "~/Desktop/Intro to Machine Learning/Project/Final Visuals/gdpversusyesvo
```

We then subset it to only Palestine proposals (and again only for years after 1972 since these are the years where the proposals for Palestine is concentrated):

```
library(plotly)
library(gapminder)
library(countrycode)

# Adding the country codes to gapminder dataset
# in order to merge later with UN Data:
gapminder_new <- gapminder %>% mutate(countryabrv = countrycode(country,
                                                                "country.name",
                                                                "iso3c"))
# Reordering columns in gapminder:
gapminder_new <- gapminder_new %>%  dplyr::select("country",
                                                  "countryabrv",
                                                  "continent",
                                                  "year",
                                                  "pop",
                                                  "gdpPercap")

voting_yespercent_groups_country_palestine <- data_clean_groups %>%
  filter(group == "palestine") %>%
  filter(year > 1972) %>%
  select(X1, ccode, vote, Country, year, resid, group) %>%
  filter(vote != 8) %>% group_by(year, Country) %>%
  summarise(totproposals = n(),
            totyes = sum(vote == 1),
            propyes = round(totyes/totproposals, digits = 2))
# Filtering columns in voting_percentages_groups:
voting_yespercent_groups_country_palestine <- voting_yespercent_groups_country_palestine %>%
  dplyr::select(Country, year, propyes)
# Renaming a column for merging purposes:
colnames(voting_yespercent_groups_country_palestine)[1] <- "countryabrv"

# Left joining voting_yespercent_groups_country_palestine
# onto gapminder, (since gapminder has less rows):
joined_df_palestine <- left_join(gapminder_new,
```

```r
                                  voting_yespercent_groups_country_palestine,
                                  by = c("countryabrv", "year")) %>% na.omit()

# Plotting  percentage of yes votes versus GDP per capita for each year
fig_palestine <- joined_df_palestine %>%
  plot_ly(
    x = ~gdpPercap,
    y = ~propyes,
    size = ~pop,
    color = ~continent,
    frame = ~year,
    text = ~country,
    hoverinfo = "text",
    type = 'scatter',
    mode = 'markers'
  ) %>%
  layout(
    xaxis = list(
      type = "log", title = "GDP per Capita"
    ),
    yaxis = list(title = "Percantage of Yes votes in Palestine Proposals (in a given year)")
  )
fig_palestine <- fig_palestine %>% layout(
    xaxis = list(
      type = "log"
    )
  )
fig_palestine <- fig_palestine %>% animation_opts(
    1000, easing = "elastic", redraw = FALSE
  )
fig_palestine <- fig_palestine %>% animation_button(
    x = 1, xanchor = "right", y = 0, yanchor = "bottom"
  )
fig_palestine <- fig_palestine %>% animation_slider(
    currentvalue = list(prefix = "YEAR ", font = list(color="red"))
  )
fig_palestine
#htmlwidgets::saveWidget(fig_palestine, "~/Desktop/Intro to Machine Learning/Project/Final Visuals/gdpv
```

We conclude by doing the same thing for Legislation proposals (again only for years between 1946-1975 since these are the years where the proposals for Legislation is concentrated):

```r
# Adding the country codes to gapminder dataset in order to merge later with UN Data:
gapminder_new <- gapminder %>% mutate(countryabrv = countrycode(country,
                                                              "country.name",
                                                              "iso3c"))
# Reordering columns in gapminder:
gapminder_new <- gapminder_new %>%  dplyr::select("country",
                                                  "countryabrv",
                                                  "continent",
                                                  "year",
                                                  "pop",
                                                  "gdpPercap")
```

```r
voting_yespercent_groups_country_legislation <- data_clean_groups %>%
  filter(group == "legislation") %>%
  filter(year < 1976) %>%
  select(X1, ccode, vote, Country, year, resid, group) %>%
  filter(vote != 8) %>% group_by(year, Country) %>%
  summarise(totproposals = n(),
            totyes = sum(vote == 1),
            propyes = round(totyes/totproposals, digits = 2))
# Filtering columns in voting_percentages_groups:
voting_yespercent_groups_country_legislation <- voting_yespercent_groups_country_legislation %>%
  dplyr::select(Country, year, propyes)
# Renaming a column for merging purposes:
colnames(voting_yespercent_groups_country_legislation)[1] <- "countryabrv"

# Left joining voting_yespercent_groups_country_legislation
# onto gapminder, (since gapminder has less rows):
joined_df_legislation <- left_join(gapminder_new,
                                   voting_yespercent_groups_country_legislation,
                                   by = c("countryabrv", "year")) %>% na.omit()

# Plotting  percentage of yes votes versus GDP per capita for each year
fig_legislation <- joined_df_legislation %>%
  plot_ly(
    x = ~gdpPercap,
    y = ~propyes,
    size = ~pop,
    color = ~continent,
    frame = ~year,
    text = ~country,
    hoverinfo = "text",
    type = 'scatter',
    mode = 'markers'
  ) %>%
  layout(
    xaxis = list(
      type = "log", title = "GDP per Capita"
    ),
    yaxis = list(title = "Percantage of Yes votes in Legislation Proposals (in a given year)")
  )
fig_legislation <- fig_legislation %>% layout(
    xaxis = list(
      type = "log"
    )
  )
fig_legislation <- fig_legislation %>% animation_opts(
    1000, easing = "elastic", redraw = FALSE
  )
fig_legislation <- fig_legislation %>% animation_button(
    x = 1, xanchor = "right", y = 0, yanchor = "bottom"
  )
fig_legislation <- fig_legislation %>% animation_slider(
    currentvalue = list(prefix = "YEAR ", font = list(color="red"))
  )
```

```r
fig_legislation
#htmlwidgets::saveWidget(fig_legislation, "~/Desktop/Intro to Machine Learning/Project/Final Visuals/gd
```