Exploring Trends in CA Health Insurance Coverage Using Text Data From Bills

Bethan Cordone and Itzel Melgoza

Introduction

On a national level, California ranks high in regard to ethnic and socioeconomic diversity. Furthermore, itb s large population has made the collection of health-related data feasible. For example, the California Health Interview Survey (CHIS) conducted by the UCLA Center for Health Policy Research is the largest state health survey in the nation. CHIS collects data on a variety of health conditions and demographic information. The general topics covered through CHIS are health status, health conditions, mental health, health behaviors, womenb s health, dental health, neighborhood and housing, access to and use of health care, food environment, health insurance, public program eligibility, bullying, parental involvement, child care and school, employment, income, and respondent characteristics which consists of demographic information. We were interested in exploring changes in health insurance coverage over time due to efforts the state has made to provide widespread health insurance for all. California is also one of five states to enact the Individual Mandate which penalizes individuals for not having health insurance. As the CHIS data provides information on the type of insurance held by each respondent, we were able to use this data to find the percent change in number of people unisured and percent change in number of people covered through Medi-Cal from the years 2011-2020.

While there are associations between the demographic data included in CHIS and the type of health insurance individuals have, we were interested in observing how the trends in health insurance coverage are associated with legislation passed in California and the US. This is because we found unsurprising associations between demographic variables and health insurance type based on an initial exploratory data analysis. For example, most individuals making less than the federal poverty level were covered through Medi-Cal (Californiab s Medicaid Health Program). This wasnb t surprising considering that Medi-Cal insures people of low-income. Furthermore, we performed a preliminary analysis to find the demographic variables associated with type of health insurance coverage using a Random Forest Model. The results of the Random Forest revealed obvious, weak assocations between demographic variables and type of health insurance coverage. This further motivated us to explore how legislation related to health policy could help explain the trends in coverage we observed over time.

Using data from LegiScan, we found bills related to health insurance that were passed in the range of years we were interested in. We found token words from the bill descriptions and used these to create a word frequency matrix. We performed clustering analysis (k means and hierarchical) to explore different categorizations of bills. We also performed a sentiment analysis on the token words from each bill. We then used the word frequency matrix to see if there was a relationship between different words within the bills and the percent change in coverage that we observed for each year interval. We used PCA and linear regression to explore this relationship. Finally we used linear regression to see if the cluster category and sentiment was associated with the percent change within the year.

Given that our data mining project address whether changes in health insurance over time can be explained by policy changes occurring either in California and on a national level, the results of this data mining project would be useful for policymakers, public health officials and researchers, and even physicians. Understanding the role of health policy changes, if one is observed, on changes in health insurance coverage can inform these individuals as to what type of legislative reform is useful in ensuring more people are insured.

California Health Interview Survey (CHIS) Data Exploration

Source and Summary of the Data

The California Health Interview Survey (CHIS) is a web and telephone-based annual health survey conducted by the UCLA Center for Health Policy Research. Prior to the 2019-2020 cycle, interviews in all languages were administered using a computer-assisted telephone interviewing (CATI) system. The 2019-2020 cycle also allowed respondents to complete the survey through the web as well. The survey was conducted every other year since 2001 and continually beginning in 2011. In 2011, they also moved towards a biennial survey model. For the purposes of this project, we decided to focus on the years in which there is continuous data. Therefore, the data spans from 2011 to 2020. In regard to who is interviewed, CHIS selects households and from each household asks an adult, teen (12-17 years old), and child (<=11 years old) to participate. The CHIS sample excludes individuals who live in group quarter residencies (ex. correctional institutions, nursing homes, college dormitories, military bases). Given that some households may not have teens or children to participate due to the household composition, we decided to focus on the adult respondents for the purpose of this project.

Unfortunately, while CHIS collects data on a variety of health conditions and demographic information, there is no regional information such as the California county in which the respondent lives has been removed to preserve confidentiality. The data consists of both source variables and constructed variables. Source variables are based on a single question asked during the CHIS interview and are generally labeled with two letters followed by numbers. Constructed variables are usually based on multiple questions asked during the interview. Constructed variables are generally acronyms and/or abbreviations.

Sampling Design:

Prior to 2019, CHIS used random-digit-dial (RDD) sampling to reach participants. Separate RDD samples were drawn to address numbers associated with landline telephone services and cellular service. Interviews were conducted in five languages (English, Spanish, Korean, Vietnamese, and Chinese (Cantonese and Mandarin dialects)) with the sixth language of Tagalog being introduced in the 2013-2014 cycle. The 58 counties in California were grouped into 44 geographic sampling strata with 14 sub-strata being created for two of the largest metropolitan areas (Los Angeles County and San Diego County). To increase the sample size in specified geographic areas or for certain ethnic and race groups, supplemental sampling was performed.

Beginning in 2019, they moved towards conducting stratified address-based sampling (ABS) due to declines in telephone interview response rates. The CHIS 2019-2020 geographic stratification remained the same as previous years. For CHIS 2019-2020, a model built from CHIS 2017-2018 data was used to determine the likelihod that groups of interest were associated with the sampled addresses. These groups of interest included

households with any of the following attributes: Korean, Vietnamese, Other Asian, Hispanic or Spanish-Speaker, Low Educational Attainment or not a US Citizen, and Have children (under 19). There continued to be geographic area oversampling for San Diego County.

The number of adults respondents per year are as follows: 2011: 22580 2012: 20355 2013: 20724 2014: 19516 2015: 21034 2016: 21055 2017: 21153 2018: 21177 2019: 22160 2020: 21949

Given the relatively similar number of respondents per year, the data appears to be suitable for observing trends in the population across years.

Weights:

To account for response biases in sampling and underrepresentation of certain groups, each yearly dataset includes weights which are meant to be used to provide accurate population estimates. The weights are created using the California Department of Financeb s Population Estimates and Population Projections which account for demographic and geographic variables. While we acknowledge that not including weights can result in inaccurate conclusions regarding population estimates and statistical parameters, weights will not be used in this project.

Missing Values

```
CHIS_adult_data <- read.csv("~/Desktop/Stat 3106/CHIS_adult_data.csv")
sum(is.na(CHIS_adult_data))
```

```
## [1] 0
```

In the CHIS data, there are no NA values. There are hardly any missing variables as most missing values are replaced through imputation. The two consistent imputation methods across years were firstly a completely random selection from the observed distribution of respondents and secondly hot deck imputation. To denote issues with data quality, CHIS uses the following negative values:

-1: INAPPLICABLE -2: PROXY SKIPPED -5: CHILD/HOUSEHOLD INFORMATION NOT COLLECTED FOR TEEN AND CHILD INTERVIEWS -7: REFUSED -8: DONb T KNOW -9: NOT ASCERTAINED

Exploratory Data Analysis

```
dim(CHIS_adult_data)
```

```
## [1] 211703    241
```

The data for adult respondents interviewed by CHIS during the years 2011-2020 consists of 211703 rows (respondents) and 241 variables.

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
CHIS_adult_data <- CHIS_adult_data %>% select(-starts_with("RAKED"))
```

```
dim(CHIS_adult_data)
```

```
## [1] 211703    160
```

After removing the variables related to weighting the variables, there are a total of 160 variables left.

First, we need to change the non-continuous variables to factors. The continuous variables are: HHSIZE_P1, HGHTI_P, HGHTM_P, HEIGHM_P, WEIGHK_P, WGHTK_P, WGHTP_P, BMI_P, INS12M, AK10_P, AK10A_P, and year.

```
cont_cols <- c("HHSIZE_P1", "HGHTI_P", "HGHTM_P",
               "HEIGHM_P", "WEIGHK_P", "WGHTK_P",
               "WGHTP_P", "BMI_P", "INS12M", "AK10_P",
               "AK10A_P", "year")
non_cont_cols <- names(CHIS_adult_data)[names(CHIS_adult_data) %in% cont_cols == FALSE]
CHIS_adult_data[non_cont_cols] <- lapply(CHIS_adult_data[non_cont_cols], factor)
```

Trends in Health Insurance Coverage Over Time

The variable INS64_P is the current health coverage held by the respondent at the time of the survey. This variable is applicable to people under 65, with individuals over 65 being assigned a value of -1.

The levels of the different insurance types are defined as below: 1 UNINSURED 2 MEDI-CAL (MEDICAID) 3 MEDICARE 4 EMPLOYMENT-BASED 5 PRIVATELY PURCHASED 6 CHIP/OTHER PUBLIC PRGM

```
levels(CHIS_adult_data$INS64_P)
```

```
## [1] "-1" "1"  "2"  "3"  "4"  "5"  "6"
```

```
levels(CHIS_adult_data$INS64_P) <- c("Skipped-Age>=65", "Uninsured",
                                     "Medi-cal (Medicaid)", "Medicare",
                                     "Employment-Based", "Privately Purchased",
                                     "CHIP/Other Public Prgm")
levels(CHIS_adult_data$INS64_P)
```

```
## [1] "Skipped-Age>=65"        "Uninsured"              "Medi-cal (Medicaid)"
## [4] "Medicare"               "Employment-Based"       "Privately Purchased"
## [7] "CHIP/Other Public Prgm"
```

```
dat_1 <-CHIS_adult_data %>% filter(INS64_P=="Uninsured")
df_1 <- as.data.frame(table(dat_1$year))
df_1$instype <- "Uninsured"

dat_2 <-CHIS_adult_data %>% filter(INS64_P=="Medi-cal (Medicaid)")
df_2 <- as.data.frame(table(dat_2$year))
df_2$instype <- "Medi-cal (Medicaid)"

dat_3 <-CHIS_adult_data %>% filter(INS64_P=="Medicare")
df_3 <- as.data.frame(table(dat_3$year))
df_3$instype <- "Medicare"

dat_4 <-CHIS_adult_data %>% filter(INS64_P=="Employment-Based")
df_4 <- as.data.frame(table(dat_4$year))
df_4$instype <- "Employment-Based"

dat_5 <-CHIS_adult_data %>% filter(INS64_P=="Privately Purchased")
df_5 <- as.data.frame(table(dat_5$year))
df_5$instype <- "Privately Purchased"

dat_6 <-CHIS_adult_data %>% filter(INS64_P=="CHIP/Other Public Prgm")
df_6 <- as.data.frame(table(dat_6$year))
df_6$instype <- "CHIP/Other Public Prgm"

allins_df <- rbind(df_1,df_2,df_3,df_4,df_5, df_6)
dim(allins_df)
```

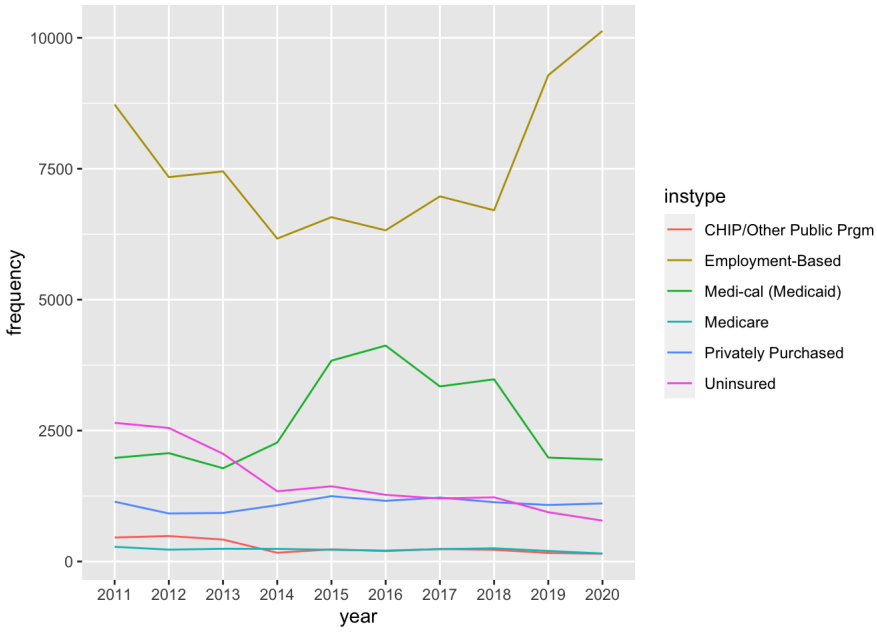
```
## [1] 60  3
```

```
names(allins_df) <- c("year", "frequency", "instype")
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
ggplot(allins_df) + geom_line(aes(year, frequency, group=instype, color=instype))
```



There are observable trends in changes in insurance type over time. For example, from 2013-2014, there was a decrease in people who are uninsured and individuals on privately purchased health insurance. Meanwhile, there was an increase in the number of individuals who were insured through Medi-cal.

Health Insurance Type & Ethnicity

To further understand who holds certain types of health insurance, we can look at the interactions between insurance type and ethnicity using the variable OMBSRR_P1.

OMBSRR_P1: OMB/CURRENT DOF RACE - ETHNICITY (PUF 1 YR RECODE)

The levels of OMBSRR_P1 are: 1 HISPANIC 2 WHITE, NON-HISPANIC (NH) 3 AFRICAN AMERICAN ONLY, NOT HISPANIC 4 AMERICAN INDIAN/ALASKAN NATIVE ONLY, NH 5 ASIAN ONLY, NH 6 OTHER/TWO OR MORE RACES

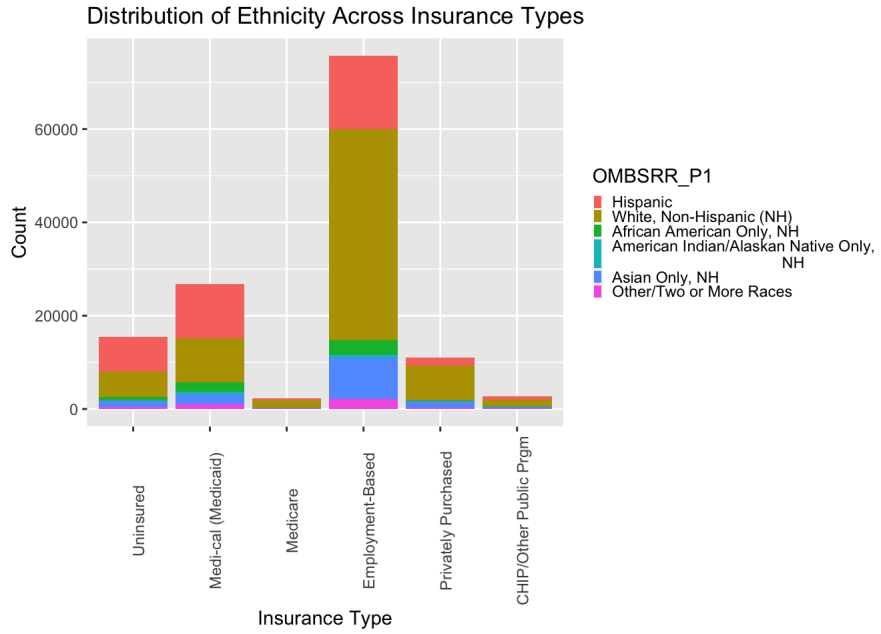
```
levels(CHIS_adult_data$OMBSRR_P1)
```

```
## [1] "1" "2" "3" "4" "5" "6"
```

```
levels(CHIS_adult_data$OMBSRR_P1) <- c("Hispanic", "White, Non-Hispanic (NH)",
                                       "African American Only, NH",
                                       "American Indian/Alaskan Native Only,
                                       NH", "Asian Only, NH", "Other/Two or More Races")
levels(CHIS_adult_data$OMBSRR_P1)
```

```
## [1] "Hispanic"
## [2] "White, Non-Hispanic (NH)"
## [3] "African American Only, NH"
## [4] "American Indian/Alaskan Native Only, \n                                    NH"
## [5] "Asian Only, NH"
## [6] "Other/Two or More Races"
```

```
CHIS_known_ins <- CHIS_adult_data %>% filter(INS64_P != "Skipped-Age>=65")

ggplot(CHIS_known_ins, aes(x = INS64_P, fill = OMBSRR_P1)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 90),
        legend.key.size = unit(0.2, "cm")) +
  labs(title = "Distribution of Ethnicity Across Insurance Types",
       x = "Insurance Type",
       y = "Count")
```



Across years, most people under 65 have employment-based coverage plans. There is no obvious distribution of ethnicity across insurance types.

Health Insurance & Poverty Level

The levels of poverty level (POVLL) are as follows: 1 0-99% FPL 2 100-199% FPL 3 200-299% FPL 4 300% FPL AND ABOVE

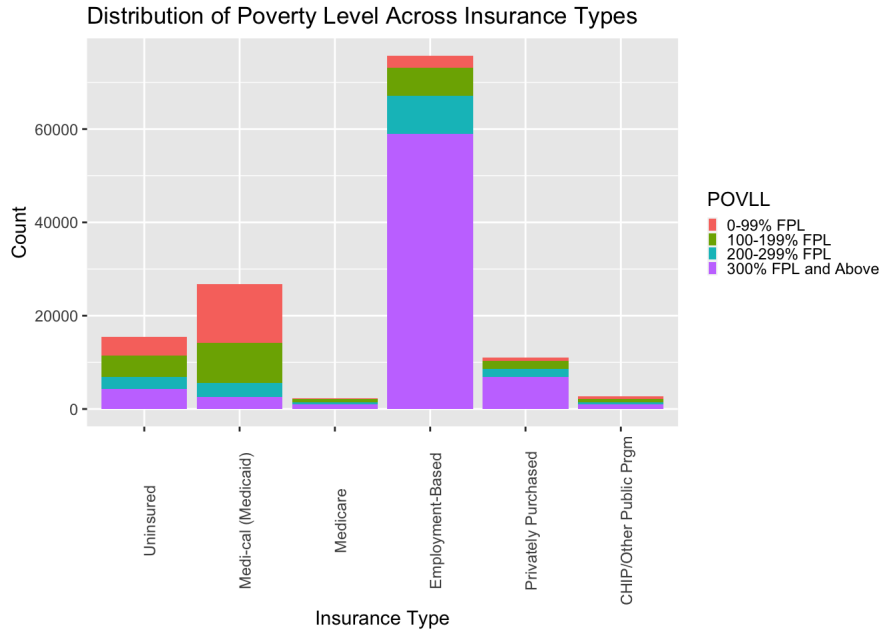
```
levels(CHIS_adult_data$POVLL)
```

```
## [1] "1" "2" "3" "4"
```

```
levels(CHIS_adult_data$POVLL) <- c("0-99% FPL", "100-199% FPL",
                                   "200-299% FPL", "300% FPL and Above")
levels(CHIS_adult_data$POVLL)
```

```
## [1] "0-99% FPL"          "100-199% FPL"       "200-299% FPL"
## [4] "300% FPL and Above"
```

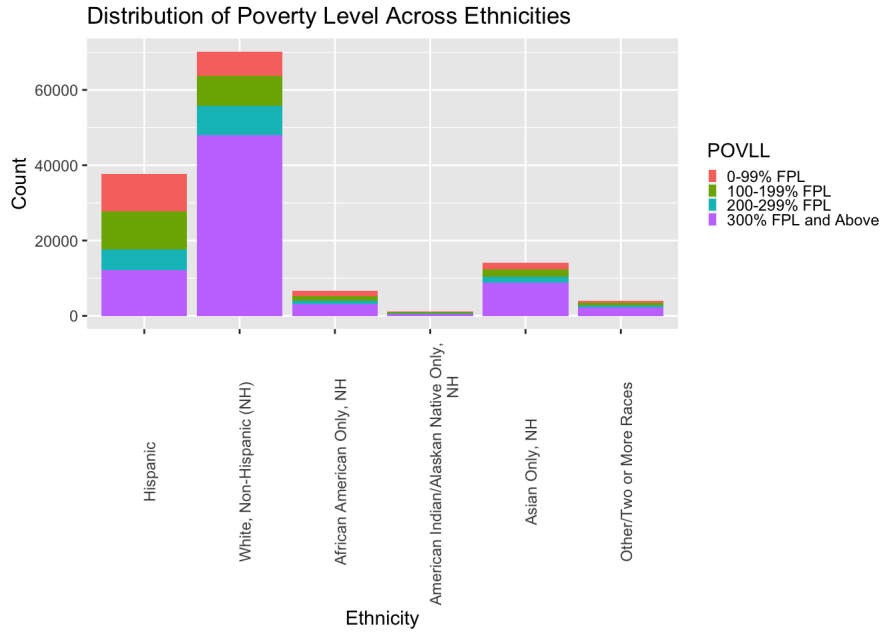
```
CHIS_known_ins <- CHIS_adult_data %>% filter(INS64_P != "Skipped-Age>=65")
ggplot(CHIS_known_ins, aes(x = INS64_P, fill = POVLL)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 90),
        legend.key.size = unit(0.2, "cm")) +
  labs(title = "Distribution of Poverty Level Across Insurance Types",
       x = "Insurance Type",
       y = "Count")
```



As expected, most people who live in a poverty level of 0-99% FPL are covered through Medi-Cal.

Poverty Level and Ethnicity

```
ggplot(CHIS_known_ins, aes(x = OMBSRR_P1, fill = POVLL)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 90),
        legend.key.size = unit(0.2, "cm")) +
  labs(title = "Distribution of Poverty Level Across Ethnicities",
       x = "Ethnicity",
       y = "Count")
```



The majority of respondents with 300% FPL and above were White, Non-Hispanic.

First-Generation Status and Insurance Type

It is also of interest to explore how being first-generation American can affect the type of coverage obtained. There are varying definitions for first-generation American with the two most common being a US-born child of foreign-born parents OR being foreign-born and gaining citizenship status (naturalized citizen). AH33NEW: 1 = BORN IN U.S., 2= BORN OUTSIDE U.S. AH34NEW: -1 = INAPPLICABLE, 1 = BORN IN U.S., 2= BORN OUTSIDE U.S. AH35NEW: -1 = INAPPLICABLE, 1 = BORN IN U.S., 2= BORN OUTSIDE U.S. CITIZEN2: 1= US-BORN CITIZEN, 2=NATURALIZED CITIZEN, 3 = NON-CITIZEN

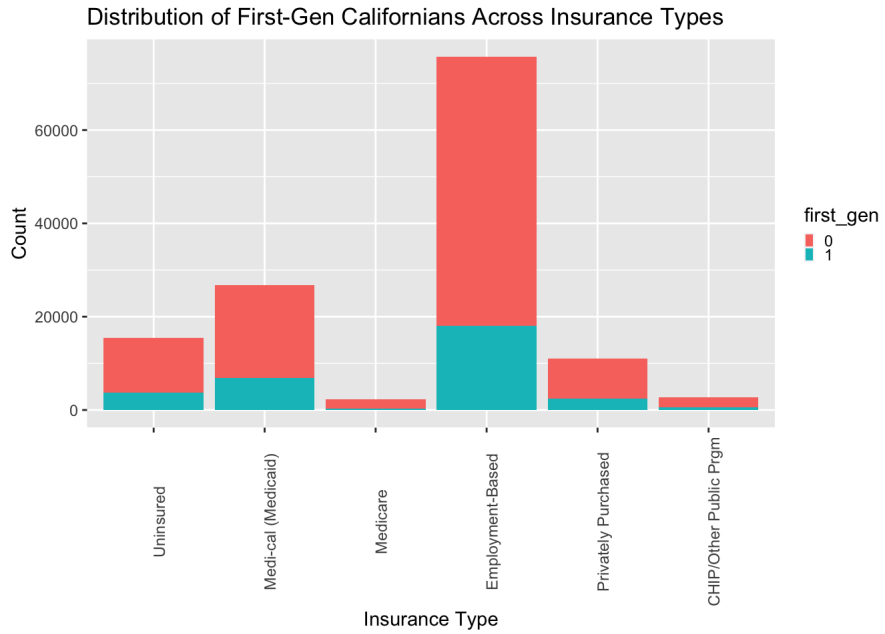
```
levels(CHIS_adult_data$AH33NEW) <- c("BORN IN U.S.", "BORN OUTSIDE U.S.")

levels(CHIS_adult_data$AH34NEW) <- c("INAPPLICABLE", "BORN IN U.S.",
                                     "BORN OUTSIDE U.S.")

levels(CHIS_adult_data$AH35NEW) <- c("INAPPLICABLE", "BORN IN U.S.",
                                     "BORN OUTSIDE U.S.")

levels(CHIS_adult_data$CITIZEN2) <- c("US-BORN CITIZEN",
                                      "NATURALIZED CITIZEN", "NON-CITIZEN")

CHIS_adult_data$first_gen <- ifelse(CHIS_adult_data$AH33NEW=="BORN IN U.S." &
                                     CHIS_adult_data$AH34NEW=="BORN OUTSIDE U.S." &
                                     CHIS_adult_data$AH35NEW == "BORN OUTSIDE U.S." |CHIS_adult_data$AH33NEW=="BORN OUTSIDE
U.S." & CHIS_adult_data$CITIZEN2 == "NATURALIZED CITIZEN" , 1, 0)
```

```
head(CHIS_adult_data %>% select(AH33NEW, AH34NEW, AH35NEW,CITIZEN2, first_gen), 6)
```

AH33NEW <fct>	AH34NEW <fct>	AH35NEW <fct>	CITIZEN2 <fct>	first_gen <dbl>
1 BORN IN U.S.	BORN OUTSIDE U.S.	BORN OUTSIDE U.S.	US-BORN CITIZEN	1
2 BORN IN U.S.	BORN IN U.S.	BORN IN U.S.	US-BORN CITIZEN	0
3 BORN OUTSIDE U.S.	INAPPLICABLE	INAPPLICABLE	NATURALIZED CITIZEN	1
4 BORN IN U.S.	BORN IN U.S.	BORN IN U.S.	US-BORN CITIZEN	0
5 BORN OUTSIDE U.S.	INAPPLICABLE	INAPPLICABLE	NON-CITIZEN	0
6 BORN IN U.S.	BORN IN U.S.	BORN OUTSIDE U.S.	US-BORN CITIZEN	0

6 rows

```
CHIS_adult_data$first_gen <- as.factor(CHIS_adult_data$first_gen)
CHIS_known_ins <- CHIS_adult_data %>% filter(INS64_P != "Skipped-Age>=65")
library(ggplot2)
ggplot(CHIS_known_ins, aes(x = INS64_P, fill = first_gen)) +
  geom_bar() +
  theme(axis.text.x = element_text(angle = 90),
        legend.key.size = unit(0.2, "cm")) +
  labs(title = "Distribution of First-Gen Californians Across Insurance Types",
       x = "Insurance Type",
       y = "Count")
```



Most first-generation Americans were covered through Employment-Based plans.

Preliminary Random Forest

Based off of the EDA, we are interested in exploring what factors predict what type of insurance an individual under 65 has with the levels ranging from: 1 UNINSURED 2 MEDI-CAL (MEDICAID) 3 MEDICARE 4 EMPLOYMENT-BASED 5 PRIVATELY PURCHASED 6 CHIP/OTHER PUBLIC PRGM

```
CHIS_adult_data$year <- as.character(CHIS_adult_data$year)
CHIS_under_65 <- CHIS_adult_data %>% filter(INS64_P != "Skipped-Age>=65")
CHIS_over_65 <- CHIS_adult_data %>% filter(INS64_P == "Skipped-Age>=65")
```

```
CHIS_under_65$INS64_P <- as.factor(CHIS_under_65$INS64_P)
train <- CHIS_under_65 %>% filter(year !="2020")
test <- CHIS_under_65 %>% filter(year == "2020")
#train$INS64_P <- droplevels(train$INS64_P)
```

```
my_rf2 <- randomForest(INS64_P ~ . -PUF1Y_ID-AI4-AI15-AI15A-AI22C-AI25-AH74-AH75-AI28-AH103-AH100-AJ102-AJ103-INSMC-INSMD-INSEM-INSPS-INSPR-INSOG-IHS-INS-INS65-INS12M-UNINSANY-INSANY-AI25NEW-INST_12-OFFTK-AH101_P-AH71_P1-AH72_P1-AH98_P1-AH99_P1-HMO-AI22A_P, train, importance=TRUE)
```

```
my_rf <- readRDS("~/Desktop/Stat 3106/my_rf2.rda")
```

```
library(randomForest)
```

```
## Warning: package 'randomForest' was built under R version 4.1.2
```

```
## randomForest 4.7-1
```

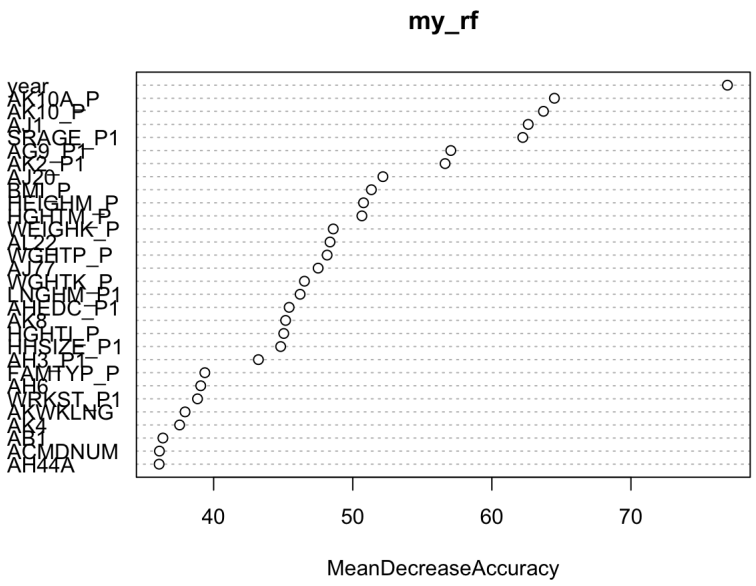
```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
varImpPlot(my_rf, type=1, offset=0.4)
```




```
#test$INS64_P <- droplevels(test$INS64_P)
#rf_preds <- predict(my_rf, newdata=test)
```

classification error

```
#rf_class_error <- rf_preds != test$INS64_P
#rf_class_error <- mean(rf_class_error)
#print(rf_class_error)
```

precision and recall

```
#rf_conf <- table(rf_preds, test$INS64_P)
```

```
#rf_precision <- rf_conf[2,2]/(rf_conf[2,2]+rf_conf[2,1])
#print(rf_precision)
#rf_recall <-rf_conf[2,2]/(rf_conf[2,2]+rf_conf[1,2])
#print(rf_recall)
```

Based off of the Random Forest, the top variables were year, AK10A_P, and AK10_P. AK10A_P and AK10_P are the respondentb s spouseb s earning last month and the respondentb s earning last month, respectively. The classification error was 0.2100799, the precision was 0.9313725, and the recall was 0.7804332. Firstly, it is an obvious insight that income is related to type of health insurance as some programs such as Medi-Cal have income eligibility requirements. Secondly, the placement of AK10A_P and AK10_P on the plot suggests that they may just be acting as noise. Lastly, the model will choose continuous variables as better predictors as they have more values and are better able to split things into categories. Given that the Random Forest model could not provide non-obvious insights into type of health insurance type held by a respondent, we were further motivated to narrow our focus to the percent change in coverage over time as well as to explore whether legislation is associated with these trends.

LegiScan Bill Data

Source and Summary of the Data

The data came from LegiScan which is a data base for legislation from all 50 states, Washington D.C., and the U.S. Congress. The csv downloads for each region contained data on the titles, descriptions, status, identification, and committee for every bill proposed. We obtained this data for the state of CA as well as the US congress. We downloaded the separate csv files for each 2 year session and combined them to form a dataset for the US and CA. We then combined these datasets to from one with all the bills in CA and the US ranging from 2009 to 2021 and labeled the bills belonging to CA and the US.

Feature generation from legislation data

The code below analyzes the text descriptions of Federal and CA bills. Only bills relating to health insurance are selected. We found token words from the descriptions for these bills and created a matrix with the count per bill and per token. We used this matrix to perform cluster analysis on the bills and a sentiment analysis.

We only used a basic keyword search to separate out the health insurance bills from the rest of the legislation but we didnb t have a easy way to differentiate these types of bills from each other. We used clustering in an attempt to create additional categories within the bills based on similar concepts. We were interested in using clustering to see if the resulting clusters would reveal bills that were related to specific type of health insurance (Ex. Medicare, Medi-Cal) or addressed specific sub-populations (Ex. veterans, people of low socioeconomic status). Observing these types of similarities between bills could help guide future analyses regarding the association between bills and percent change in coverage over time.

In regard to the sentiment analysis, we were interested in seeing whether we could create an index for the sentiment of a bill and whether the sentiment of a bill could affect the percent change in coverage for the year after it was passed. For example, we could expect bills with negative sentiments to be associated with bills calling for decreased funding for federally-funded health insurance programs. A possible limitation associated of the sentiment analysis is that we are using legal text which is likely written to not be emotionally-charged. Unsurprisingly, not many words from the sentiments dataset were used in the bills.

Packages

```
library(readr)
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
library(lubridate)
library(stringr)
library(dplyr)

# for the token words
library("udpipe")
```

```
## Warning: package 'udpipe' was built under R version 4.1.2
```

```
# for stopwords
library(stopwords)

# for comparing the clusters
library(fpc)

# library
library(tidyverse)
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

```
library(tidytext)
library(textdata)
```

```
## Warning: package 'textdata' was built under R version 4.1.2
```

```
library(glue)
```

```
## Warning: package 'glue' was built under R version 4.1.2
```

Subsetting

This section explores which subset of bills to analyze further. Since we wanted to explore how trends in health insurance coverage overtime correspond with health insurance legislation changes, we did a keyword search through the bill descriptions to obtain bills that were relevant. This included keywords health insurance, Medi-cal, Medicare, and Medicaid. We also subset the data so that it only included bills with status b passedb as bills with other statuses did not become law and therefore it can be reasonably assumed to not have effects.

```
bills <- read.csv("~/Downloads/bills_raw.csv")
# this is only bills that passed
bills_passed <-  bills[bills$status_desc == "Passed" ,]

# much less but still many bills
bills_passed %>% dim
```

```
## [1] 15300    16
```

```
bills_passed <- bills_passed %>% select(-X)

bills_passed %>% dim
```

```
## [1] 15300    15
```

```
# keywords
toMatch <- c("Health Insurance", "Medicare", "Medicaid", "Medi-cal")
matches <- grep(paste(toMatch,collapse="|"),
                bills_passed$description, ignore.case = T)

# look at the results of these matches on the data
HI_passed <- bills_passed[matches, ]

# we get 206 bills with includes the medicare, medicaid, and medi-cal
HI_passed %>% dim
```

```
## [1] 206  15
```

```
# how many CA bills?
# the majority of bills are from CA
# most are related to Medi-cal
HI_passed[HI_passed$level == "CA", ] %>% dim
```

```
## [1] 110  15
```

```
# how many US bills?
# 96 federal bills
HI_passed[HI_passed$level == "US", ] %>% dim
```

```
## [1] 96 15
```

Cleaning the text data

To generate our text features, we used a matrix of words counts of the tokens taken from the bill descriptions. Before creating the token words, we processed and cleaned the bill descriptions. We used all lower case letters and the number and year values were replaced with NUMBER and YEAR. We removed all non alphabetic characters and deleted extra spaces.

```
# use the bill descriptions to cluster the data
des <- HI_passed$description

# lets lowercase all of the words
des <- tolower(des)
```

```
# lets replace YEARS and NUMBERS
# replace % sign and all punctuation
des <- gsub("([^0-9])(19|20)[0-9]{2}([^0-9])", "YEAR", des)

des <- gsub("\\$?[0-9]+[,\\/\\.]?[0-9]*", "NUMBERS", des)
```

```
des <- gsub("[^[:alnum:]]", " ", des)

# removes extra spaces between words
des <- str_squish(des)
```

Token matrix

We calculated the token words for each bill description and created a matrix where the number of rows corresponded to the bills and the columns were token words. The content of the matrix contained counts of each token word per bill.

A histogram of the appearance percentage reveals that the majority of words appear in less than 10% of the bills. The number of words appearing in 10% to 20% is significantly lower and the frequency of words appearing in higher percentages further drops off.

```
# download token words algorithm
dl <- udpipe_download_model(language="english")
eng_model <- udpipe_load_model(file=dl$file_model)
```

```
# function to extra token word counts for each element (bill)
token <- function(e) {
  out_udpipe <- udpipe_annotate(eng_model, x= e,
                                tagger='default', parser='none')
  y <- as.data.frame(out_udpipe)
  return(table(y$lemma))
}
```

```
# create a list of all the token words for each bill
toke <- lapply(des , token)

# bind the rows such that non matching columns contain NA values
token_df <- do.call(bind_rows, toke)
```

```
# this is the code for reading in the matrix as a csv file so the last step does not have to be repeated
token_df <- read.csv("~/Downloads/token2.0.csv")
token_df <- token_df[ ,-1]
names(token_df)[1] <- "'s"

token_df[1:3,1:3]
```

	's <int>	a <int>	ability <int>
1	15	91	1
2	3	50	*NA*
3	6	28	*NA*

3 rows

```
# replace the NA's with 0
token_df <- as.data.frame(token_df)
token_df[is.na(token_df)] <- 0
```

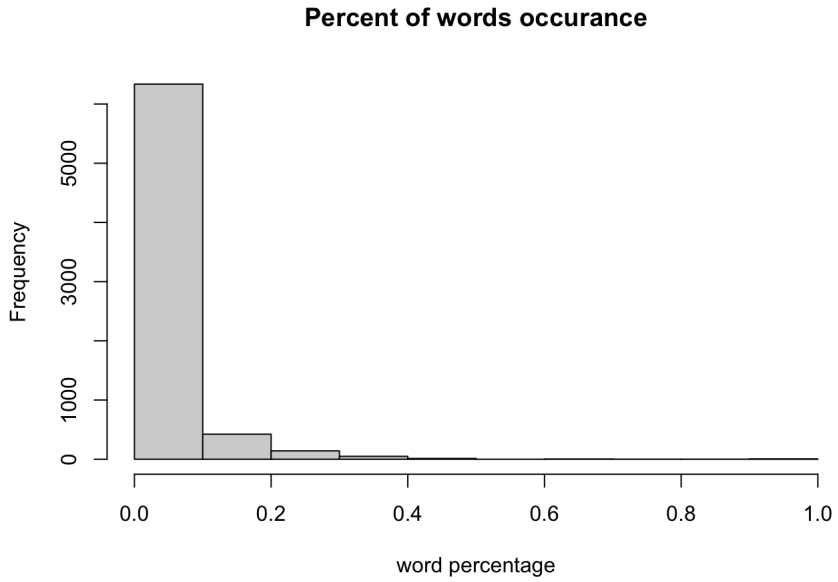
```
# look at the values
max(token_df)
```

```
## [1] 726
```

```
min(token_df)
```

```
## [1] 0
```

```
# calculate the appearance percentage
percent <- apply(token_df, 2, function(x) mean(x > 0))
# this might not be right because I think I messed something up
hist(percent, main = "Percent of words occurance", xlab = "word percentage")
```



Clustering

We performed k means and hierarchical clustering on the token matrix. We used the TF-IDF matrix method to scale the text data in preparation for the clustering algorithms. We then analyzed the cluster to understand the bill properties and create a categorization variable.

TF-IDF matrix

```
# calculate the TF matrix
TF <- t(apply(token_df, 1, function(x) x/max(x)))
TF %>% dim
```

```
## [1]  206 6974
```

```
TF[1:3, 1:3]
```

```
##             's         a     ability
## [1,] 0.07009346 0.4252336 0.004672897
## [2,] 0.02459016 0.4098361 0.000000000
## [3,] 0.12244898 0.5714286 0.000000000
```

```
# all below 1
sum(TF > 1)
```

```
## [1] 0
```

```
# calculate the IDF matrix
N <- nrow(token_df)

n_t <- apply(token_df, 2, function(x) sum(x != 0))

IDF <- log(N/n_t)
IDF <- matrix(rep(IDF, N), byrow = T, nrow=N)

IDF %>% dim
```

```
## [1]  206 6974
```

```
IDF[1:3, 1:3]
```

```
##          [,1]      [,2]     [,3]
## [1,] 1.037417 0.04976151 2.437504
## [2,] 1.037417 0.04976151 2.437504
## [3,] 1.037417 0.04976151 2.437504
```

```
range(IDF)
```

```
## [1] 0.000000 5.327876
```

```
# calculate TF_IDF
TF_IDF <- TF * IDF

# show TF_IDF
TF_IDF[1:3, 1:3]
```

```
##              's          a    ability
## [1,] 0.07271613 0.02116027 0.01139021
## [2,] 0.02551025 0.02039406 0.00000000
## [3,] 0.12703062 0.02843515 0.00000000
```

```
# calculate the max TF_IDF for each token
mtfidf <- apply(TF_IDF, 2, max)
mtfidf %>% head
```

```
##         's          a    ability      about     access    account
## 0.82128824 0.04976151 0.02195950 0.15355874 0.73886428 0.14720446
```

```
mtfidf %>% length
```

```
## [1] 6974
```

```
range(mtfidf)
```

```
## [1] 0.000000 3.995907
```

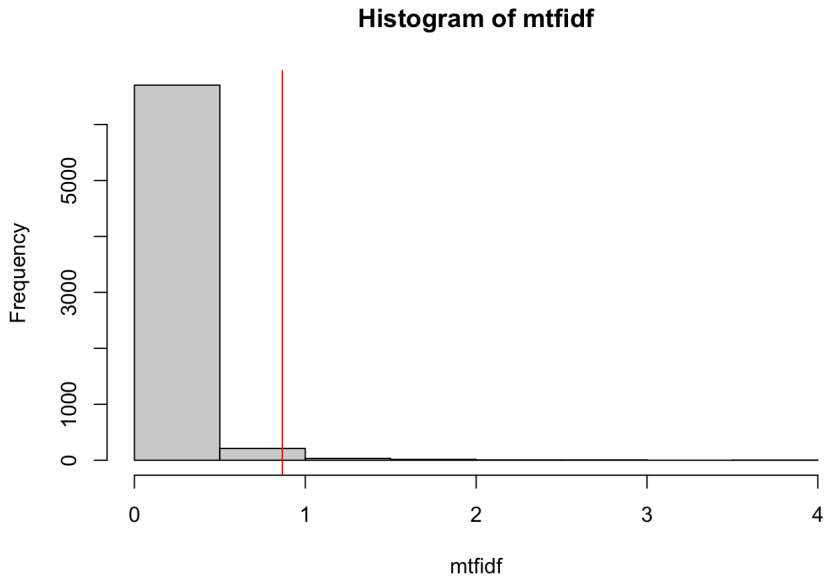
```
# 186 that are greater than 1
sum(mtfidf > 1)
```

```
## [1] 60
```

Stopwords / Determining Top Tokens

In order to limit the tokens used for clustering, we used the mtfidf values to determine b topb tokens. We initially used the stopwords dataset to determine which tokens had a higher mtfidf value than the maximum mtfididk value of the stopwords, but this method only yielded 89 words that were higher than the stopwords maximum line. We decided to instead include the top 5% of tokens, categorized by mtfidf value, which was 325 tokens. We wanted the number of tokens to be greater than the number of bills but not so large as to include tokens that were significant to many bills as this would interfere with clustering.

```
stop_words <- stopwords::stopwords("en")
matches <- names(mtfidf) %in% stop_words
sel_col <- names(mtfidf)[which(names(mtfidf) %in% stop_words)]

mtfidf_stop <- mtfidf[sel_col]
max_mtfidf_stop <- max(mtfidf_stop)
hist(mtfidf)
abline(v=max_mtfidf_stop, col = "red")
```




```
max_mtfidf_stop
```

```
## [1] 0.8654447
```

```
range(mtfidf)
```

```
## [1] 0.000000 3.995907
```

```
# there are 34 words greater than the line
sum(mtfidf > max_mtfidf_stop)
```

```
## [1] 89
```

```
# include the top 5% of words which is about 325 words which is more than the number of rows
top_words <- sort(mtfidf, decreasing = T)[1:325]
```

```
# include the top 5% of the words
top <- TF_IDF[ ,names(top_words)]

# has the correct dimensions
top %>% dim
```

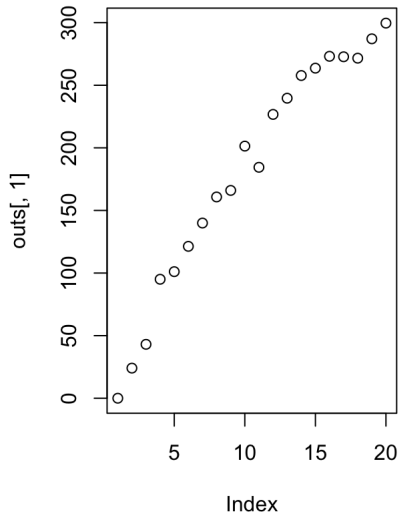
```
## [1] 206 325
```

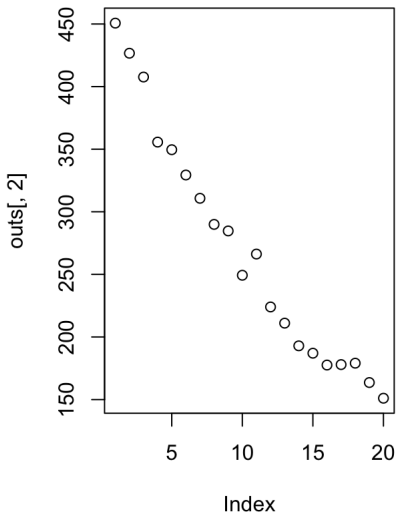
```
# add names to make things easier later on
names(top) <- names(token_df[, names(top_words)])
```

K means Clustering

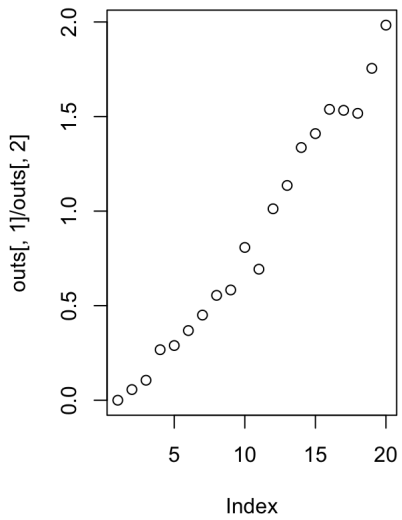
We applied the k means clustering algorithm to the TF_IDF matrix of the top tokens. Based on the plots of the ratio of betweenss and tot.withinss, we chose k = 3 clusters as this is were the first kink appears in the graph. A table of the k means cluster reveals that that two of the clusters have 4 and 5 bills while the last cluster has 197 bills.

```
set.seed(123)
k_max <- 20
outs <- matrix(NA, ncol=2, nrow=k_max)

for(k_guess in seq_len(k_max)){
  km_out <- kmeans(top, centers=k_guess)
  outs[k_guess, 1] <- km_out$betweenss
  outs[k_guess, 2] <- km_out$tot.withinss
}

par(mfrow=c(1, 2))
plot(outs[, 1])
plot(outs[, 2])
```






```
plot(outs[ ,1]/outs[ ,2])
```




```
# set k = 3 clusters
k <- 3
km <- kmeans(top, centers = k)

# look at distribution of clusters
table(km$cluster)
```

```
##
##   1   2   3
## 197   4   5
```

Analysis of K means Clusters

After performing K-means clustering, it was necessary to evaluate the clusters based on the top tokens associated with each. Finding these top tokens along with which bills were placed in each cluster could give us insight into the topics of the health insurance bills. It would help reveal whether they were related to specific type of health insurance coverage or addressed specific sub-populations.

```
# assign the clusters to the data_frame
top_df <- as.data.frame(top)
top_df$cluster <- km$cluster

top_df %>% dim
```

```
## [1] 206 326
```

```
# function to extract the top words
keywords <- function(clust, df) {
  cluster_sel <-  df %>% filter(cluster == clust) %>% select(-cluster)
  maxes <- apply(cluster_sel,2,function(x) max(x))
  k_words <- maxes[maxes > 0]
  if(length(k_words) < 10) {
    k_words <- c(k_words, rep(NA, (10-length(k_words))))
  }
  return(k_words[1:10])
}
```

```
# create a dataframe with the top 10 words from each bill
key <- c()
for(i in 1:k) {
  key <- cbind(key, names(keywords(clust=i, df=top_df)))
}

key <- as.data.frame(key)
names(key) <- paste("cluster", 1:k)
```

```
# look at the top words for each cluster
key
```

cluster 1 <chr>	**cluster 2** <chr>	**cluster 3** <chr>
alzheimer	strengthening	therapeutic
strengthening	payer	trial
payer	ofYEAR	accessory
pace	repay	ssi
wtc	enhancement	device
ofYEAR	relative	supervision
repay	secondary	speech
independence	taxpayer	throughyear
enhancement	ofyear	critical
relative	medicare	hospital

1-10 of 10 rows

```
table(km$cluster)
```

```
##
##   1   2   3
## 197   4   5
```

Next, we wanted to examine the titles of the bills assigned to each cluster to understand how the k-means algorithm categorized the bills. We were also interested in seeing how many bills in each cluster were passed per year. This could give us an idea of how the number of bills passed per year can explain changes in percent of people covered through a health insurance plan.

```
# sort this cluster based on the size of the k's
ranked_clust <- sort(table(km$cluster), decreasing = F) %>% names

ranked_clust
```

```
## [1] "2" "3" "1"
```

```
# add the cluseters to the main dataset
HI_passed$cluster <- km$cluster
```

```
# this is the smallest cluster containing 4 bills
cs <- HI_passed[HI_passed$cluster == ranked_clust[1], ]

# take a look at the 4 titles
cs$title
```

```
## [1] "Medicare Secondary Payer Enhancement Act of 2010."
## [2] "Strengthening Medicare and Repaying Taxpayers Act of"
## [3] "Medicare: dental care."
## [4] "Medicare: observation status."
```

```
# what state is this cluster from
table(cs$level)
```

```
##
## CA
##  4
```

```
# all passed in different years between 2010 and 2015
table(year(ymd(cs$status_date)))
```

```
##
## 2010 2011 2012 2015
##    1    1    1    1
```

```
# these are all CA bills related to Medicare
```

The smallest cluster (4 bills) contains exclusively CA bills related to Medicare. All these bills were passed between 2010 and 2015. The top tokens for this cluster were strengthening and payer. This makes sense since the bills are related to strengthening Medicare and determining payments.

```
# the middle cluster with 5 bills
cm <- HI_passed[HI_passed$cluster == ranked_clust[2], ]

cm$title[1:3]
```

```
## [1] "Improving Access to Clinical Trials Act of 2009"
## [2] "To provide for the extension of the enforcement instruction on supervision requirements for outpatient therapeutic s
ervices in critical access and small rural hospitals through 2014."
## [3] "Ensuring Access to Clinical Trials Act of 2015"
```

```
# all federal bills
table(cm$level)
```

```
##
## US
##  5
```

```
# 1 bill pased in 2010 and 2014 and 3 bills past in 2015
table(year(ymd(cm$status_date)))
```

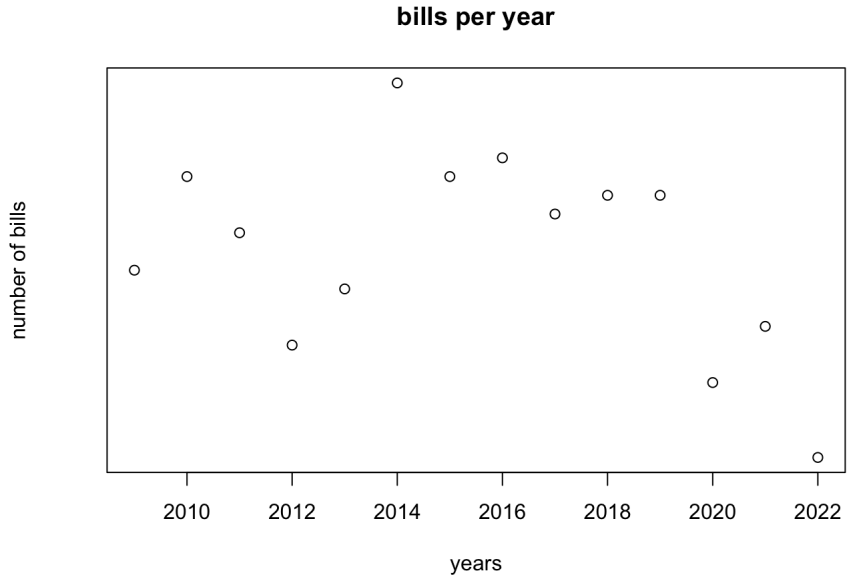
```
##
## 2010 2014 2015
##    1    1    3
```

The second biggest cluster (5 bills) contains exclusively US bills which are related to access to clinical trials. The top tokens for this cluster are therapeutic and trial. This makes sense in the context of the cluster.

```
cl <- HI_passed[HI_passed$cluster == ranked_clust[3], ]

# the rest of the CA and US bills
table(cl$level)
```

```
##
##  CA  US
## 106  91
```

```
# make a plot of the year distribution
years <- table(year(ymd(cl$status_date))) %>% names %>% as.numeric
counts <- table(year(ymd(cl$status_date))) %>% unname

# there does not appear to be any pattern in the year distribution of bills in this cluster
plot(years, counts, xlab = "years",
     ylab = "number of bills", main = "bills per year")
```



The largest cluster contained 197 bills of which 106 were b CAb bills and 91 were b USb bills. The distribution of the bills across years was pretty uniform. The content of this cluster was unclear as this is the cluster that the remaining bills were placed into.

The results of the first two clusters indicate that the algorithm successfully categorized the bills into meaningful groups. The K-means algorithm indicated that 3 clusters was the ideal k to use. The use of k=3 resulted in two clusters that show obvious thematic specificity in the type of bills chosen. The last cluster did not reveal a cluster of bills that were obviously related. With a larger k, itb s possible that the largest of the three clusters would be separated into smaller clusters. However, itb s unlikely that the bills assigned to these smaller clusters would be obviously related.

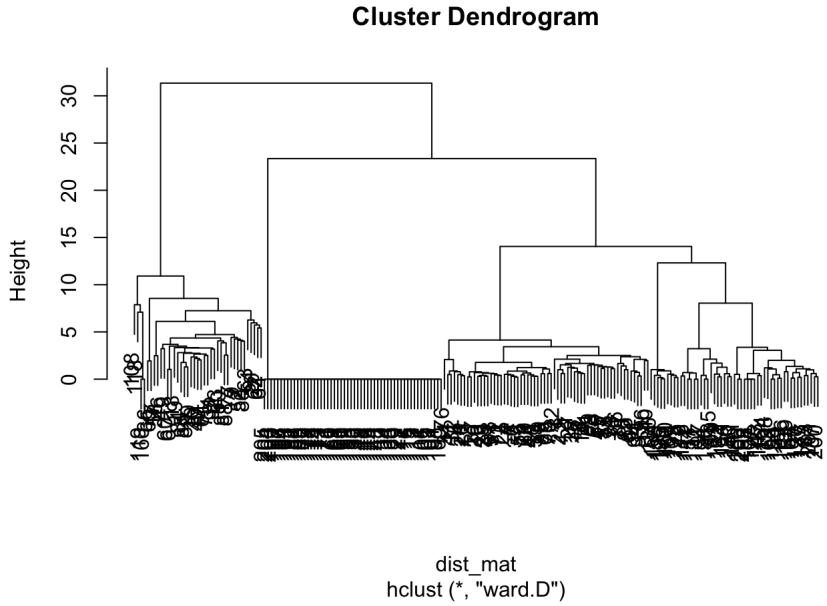
Hierachical clustering

We also tried hierarchical clustering on the data and used the b Ward.Db method because this yielded the most usable dendrograms. The other methods (b singleb and b completeb) yielded dendrograms that branched into many tiny clusters extremely early. The number of clusters was chosen to be 3. The distribution of bills in each cluster was distributed more evenly than in the k means, the smallest cluster had 39 bills, the second smallest had 54, and the third largest had 113.

```
set.seed(123)
top %>% dim
```

```
## [1] 206 325
```

```
dist_mat <- dist(top)
h_out <- hclust(dist_mat, method = "ward.D")
plot(h_out)
```




```
h_clust <- cutree(h_out, k = 3)
```

```
table(h_clust)
```

```
## h_clust
##   1   2   3
## 113  39  54
```

Analysis of Hierarchical Clustering

```
# assign the clusters to the data_frame
top_df$h_cluster <- h_clust
top_df %>% dim
```

```
## [1] 206 327
```

```
# function to extract the top words
keywords_h <- function(clust, df) {
  cluster_sel <-  df %>% filter(h_cluster == clust) %>%
    select(-c(cluster, h_cluster))
  maxes <- apply(cluster_sel,2,function(x) max(x))
  k_words <- maxes[maxes > 0]
  if(length(k_words) < 10) {
    k_words <- c(k_words, rep(NA, (10-length(k_words))))
  }
  return(k_words[1:10])
}
```

```
# create a dataframe with the top 10 words from each bill
key <- c()
for(i in 1:k) {
  key <- cbind(key, names(keywords_h(clust=i, df=top_df)))
}

key <- as.data.frame(key)
names(key) <- paste("cluster", 1:k)
```

```
key
```

cluster 1 <chr>	cluster 2 <chr>	cluster 3 <chr>
strengthening	alzheimer	
payer	strengthening	
pace	payer	

cluster 1 <chr>	cluster 2 <chr>	cluster 3 <chr>
ofYEAR	pace	
repay	wtc	
independence	ofYEAR	
enhancement	repay	
relative	independence	
secondary	enhancement	
quot	relative	

1-10 of 10 rows

```
h_rank <- sort(table(h_clust), decreasing = F) %>% names

# bills on the status of medicare in the "US" and "CA"
HI_passed$h_clust <- h_clust

HI_passed %>% dim
```

```
## [1] 206  17
```

```
cs_h <- HI_passed[HI_passed$h_clust == h_rank[1], ]

cs_h$title[1:3]
```

```
## [1] "James Zadroga 9/11 Health and Compensation Act of 2010"
## [2] "To amend title XVIII of the Social Security Act to delay the date on which the accreditation requirement under the M
edicare Program applies to suppliers of durable medical equipment that are pharmacies."
## [3] "Preservation of Access to Care for Medicare Beneficiaries and Pension Relief Act of 2010"
```

```
# most of these are "US" bills
table(cs_h$level)
```

```
##
## CA US
##  4 35
```

```
# most of the bills are in 2015
table(year(ymd(cs_h$status_date)))
```

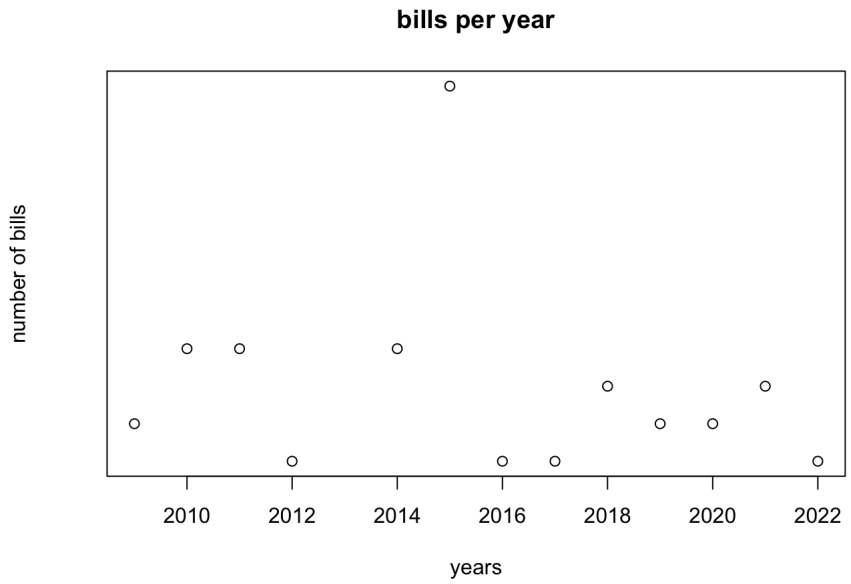
```
##
## 2009 2010 2011 2012 2014 2015 2016 2017 2018 2019 2020 2021 2022
##    2    4    4    1    4   11    1    1    3    2    2    3    1
```

```
# these are the Medicare bills
cs_h[cs_h$level == "CA", 1:3]
```

	bill_id <dbl>	session_id <int>	bill_number <chr>
92925	205887	30	AJR42
97866	330789	82	AJR12
97884	412335	82	AJR30
109557	777583	1120	SJR8

4 rows

```
# make a plot of the year distribution
years <- table(year(ymd(cs_h$status_date))) %>% names %>% as.numeric
counts <- table(year(ymd(cs_h$status_date))) %>% unname

# there does not appear to be any pattern in the year distribution of bills in this cluster
plot(years, counts, xlab = "years",
     ylab = "number of bills", main = "bills per year")
```



The smallest cluster contains 39 bills. All the bills are b USb except for 4 bills that are related to the b CAb Medicare bills found in the k means clustering. For the distribution across years it seems that 2015 had a large number of bills passed.

```
cm_h <- HI_passed[HI_passed$h_clust == h_rank[2], ]

cm_h$title[1:4]
```

```
## [1] "Medi-Cal: designated public hospitals: seismic safety"
## [2] "Medi-Cal: providers: remedies."
## [3] "Medi-Cal: proof of eligibility."
## [4] "Medi-Cal: eligibility."
```

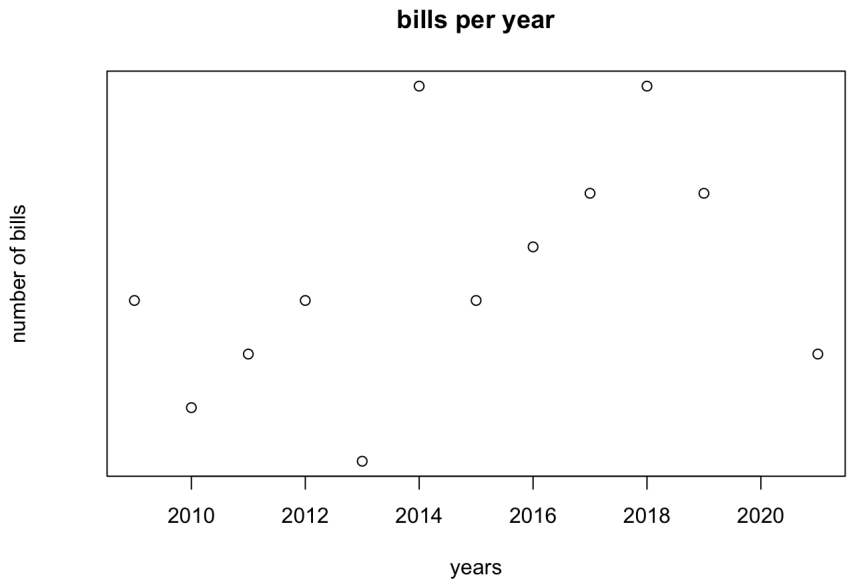
```
# all federal bills
table(cm_h$level)
```

```
##
## CA
## 54
```

```
# seems an adjustment every year
# pretty even distribution throughout the year
table(year(ymd(cm_h$status_date)))
```

```
##
## 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2021
##    4    2    3    4    1    8    4    5    6    8    6    3
```

```
# make a plot of the year distribution
years <- table(year(ymd(cm_h$status_date))) %>% names %>% as.numeric
counts <- table(year(ymd(cm_h$status_date))) %>% unname

# there does not appear to be any pattern in the year distribution of bills in this cluster
plot(years, counts, xlab = "years",
     ylab = "number of bills", main = "bills per year")
```



The middle cluster contained 54 bills which were all from b CAb and related to Medi-Cal. The year distribution plot is shown below. There may be non-trivial patterns. Additionally, the most bills are passed in 2014 and 2018.

```
cl_h <- HI_passed[HI_passed$h_clust== h_rank[3], ]

cl_h$title[1:4]
```

```
## [1] "Children's Health Insurance Program Reauthorization Act of 2009"
## [2] "FAA Air Transportation Modernization and Safety Improvement Act"
## [3] "Legislative Branch Appropriations Act, 2010"
## [4] "Veterans' Benefits Act of 2010"
```

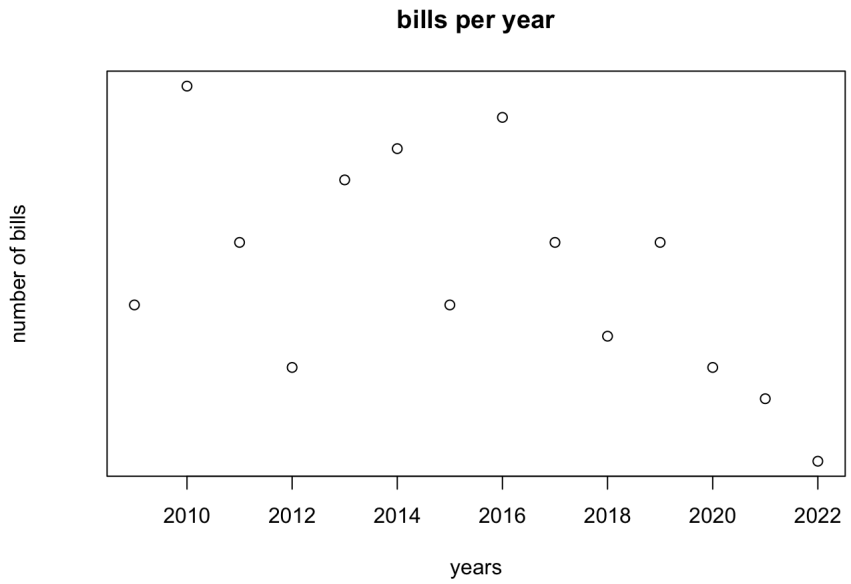
```
# about equal dist
table(cl_h$level)
```

```
##
## CA US
## 52 61
```

```
table(year(ymd(cl_h$status_date)))
```

```
##
## 2009 2010 2011 2012 2013 2014 2015 2016 2017 2018 2019 2020 2021 2022
##    7   14    9    5   11   12    7   13    9    6    9    5    4    2
```

```
# make a plot of the year distribution
years <- table(year(ymd(cl_h$status_date))) %>% names %>% as.numeric
counts <- table(year(ymd(cl_h$status_date))) %>% unname


# there does not appear to be any pattern in the year distribution of bills in this cluster
plot(years, counts, xlab = "years",
     ylab = "number of bills", main = "bills per year")
```



The year that had the maximum bills passed from this cluster was 2010. Additionally, the bills in the cluster were split between the b USb and b CAb so these were likely the bills that were leftover from the other clusters.

Results of Clustering & Implications for Percent Change in Medi-Cal Coverage

After trying both K-means and hierarchical clustering, the hierarchical clustering revealed a cluster of bills that can help explain percent change in Medi-Cal Coverage over time. The hierarchical cluster of 54 bills in which all are related to Medi-Cal can provide valuable insight into the relationship between legislation and health insurance coverage. The analysis of this cluster revealed that the greatest amount of these Medi-Cal related bills were passed in 2014 and 2018. Interestingly, EDA of the CHIS data revealed that from 2014 to 2015 there was a steep increase in the percent of respondents covered through Medi-Cal. Furthermore,from 2018 to 2019 there was a steep decrease in the percent of respondents covered through Medi-Cal. While this association between the number of bills passed and trends in percent of people covered do not imply a direct causal relationship, these results can motivate further exploration into the exact nature of the bills and the type of reform which they implemented. The results of the hierarchical clustering thus motivated us to specifically focus on the percent change in Medi-Cal coverage over time and on the percent of uninsured respondents over time. We were interested in the uninsured population because the expansion of programs like Medi-Cal should theoretically decrease the number of uninsured people.

Sentiment Feature

Given that the trends in health insurance coverage over time show rises and falls, itb s possible that not all bills which are passed are advocating for increased funding for federally funded health insurance programs. Therefore, we were interested in examining the sentiment of each bill based on its description.

We used the b afinnb sentiment dataset which had sentiment words with values ranging from -5 to 5. We matched the sentiment words with the tokens to find the overlap. We then created a matrix were the rows were words and the columns were the bills with an additional column for the sentiment index for each words. The rest of the matrix contained values for the word counts. We used this matrix to calculate the average sentiment for each bill by multiplying each column by the sentiment value and dividing by the sum of the words.

```
token_df[1:3, 1:3]
```

	's <dbl>	a <dbl>	ability <dbl>
1	15	91	1
2	3	50	0
3	6	28	0

3 rows

```
token_df[1,names(token_df) > 0] %>% names %>% head
```

```
## [1] "a"       "ability"  "about"    "access"   "account"  "accurate"
```

```
# downloaded the sentiments
sentiments <- get_sentiments("afinn")
token_words <- as.data.frame(names(token_df))
names(token_words) <- "word"

sent_overlap <- inner_join(get_sentiments("afinn"), token_words)
```

```
## Joining, by = "word"
```

```
sent_overlap %>% dim
```

```
## [1] 420   2
```

```
sent_overlap[1:3, ]
```

word <chr>	value <dbl>
abandon	-2
abandoned	-2
ability	2

3 rows

```
token_t <- as.data.frame(t(token_df))

# label all the bills with the bills numbers
names(token_t) <- HI_passed$bill_id

token_t$word <- rownames(token_t)

bills_overlap <- inner_join(token_t, sent_overlap)
```

```
## Joining, by = "word"
```

```
# we want to check it there are bills that have all 0 words for the sentiments words
test <- apply(bills_overlap[ ,1:206], 2, sum)

# there are 100 bills that do not have overlap with sentiment words
sum(test > 0)
```

```
## [1] 100
```

```
sent_index <- apply(bills_overlap[ ,1:206], 2, function(x) sum(x*bills_overlap$value)/sum(x))
# replace the NaN values

# there are no 0 sentiments for those
sum(na.omit(sent_index) == 0)
```

```
## [1] 0
```

```
# set the bills with no sentiment index to 0
sent_index[is.nan(sent_index)] <- 0
```

Master Data Frame (Clusters/Sentiments/Bill Data)

We added the two cluster types along with the sentiment values to the dataframe of bills so we could use it in later analysis.

```
# put this all together with everything else

# sent index for the bills
sent_index %>% length
```

```
## [1] 206
```

```
HI_passed$h_clust %>% length
```

```
## [1] 206
```

```
HI_passed$cluster %>% length
```

```
## [1] 206
```

```
# add this column to the data frame
HI_passed$sentiment <- sent_index

HI_passed %>% dim
```

```
## [1] 206  18
```

Change in % of Coverage Over Time

Change in % of Individuals Covered Through Medi-Cal Over Time

Finding % change per year

Given that the clustering results revealed a cluster of bills primarily related to Medi-Cal, we decided to explore how the bills passed within a given year are associated the change in percent of people covered through Medi-Cal the following year. Specifically, web re interested in observing whether the tokens of a bill are associated with the change in percent of people covered through Medi-Cal the following year.

For adults under 65, this meant that the variable INS64_P should be b Medi-cal (Medicaid)b and for adults over 65 this meant that the variable INS65 should be b Medicare+Medi-cal(Medicaid).b We found the percent of adults covered by Med-Cal per year and then found the changes from year to year.

```
levels(CHIS_adult_data$INS64_P) #for adults < 65
```

```
## [1] "Skipped-Age>=65"        "Uninsured"              "Medi-cal (Medicaid)"
## [4] "Medicare"               "Employment-Based"       "Privately Purchased"
## [7] "CHIP/Other Public Prgm"
```

```
levels(CHIS_adult_data$INS65) #for adults >=65
```

```
## [1] "-1" "1"  "2"  "3"  "4"  "5"
```

```
levels(CHIS_adult_data$INS65) <- c("Skipped-Age<65",
                                   "Medicare+Medi-cal(Medicaid)",
                                   "Medicare+Other", "Medicare Only",
                                   "Other Only", "Uninsured")
levels(CHIS_adult_data$INS65)
```

```
## [1] "Skipped-Age<65"              "Medicare+Medi-cal(Medicaid)"
## [3] "Medicare+Other"              "Medicare Only"
## [5] "Other Only"                  "Uninsured"
```

```
MD_peryr_count_df <- CHIS_adult_data %>%
  group_by(year) %>%
  summarize(num_row_per_year = n(),
            MD_64 = sum(INS64_P=="Medi-cal (Medicaid)"),
            MD_65 = sum(INS65=="Medicare+Medi-cal(Medicaid)"))
head(MD_peryr_count_df)
```

year <chr>	num_row_per_year <int>	MD_64 <int>	MD_65 <int>
2011	22580	1978	969
2012	20355	2069	1251
2013	20724	1781	1072
2014	19516	2271	1293
2015	21034	3835	1273
2016	21055	4123	1443

6 rows

```
MD_perc_df <- MD_peryr_count_df %>% mutate(MD_perc = (MD_64+MD_65)/num_row_per_year)
head(MD_perc_df)
```

year <chr>	num_row_per_year <int>	MD_64 <int>	MD_65 <int>	MD_perc <dbl>
2011	22580	1978	969	0.1305137
2012	20355	2069	1251	0.1631049
2013	20724	1781	1072	0.1376665
2014	19516	2271	1293	0.1826194
2015	21034	3835	1273	0.2428449
2016	21055	4123	1443	0.2643553

6 rows

```
#calculating change per year
MD_perc_final <- MD_perc_df %>% select(c(year,MD_perc))
MD_perc_changes <- diff(MD_perc_final$MD_perc)
yr_changes <- c("2011-2012", "2012-2013", "2013-2014",
                "2014-2015", "2015-2016", "2016-2017",
                "2017-2018", "2018-2019", "2019-2020")
year_bill_passed <- c("2011","2012", "2013","2014",
                      "2015","2016","2017","2018","2019")
MD_change_df <- data.frame(year_int = yr_changes, year_passed=year_bill_passed, MD_perc_change=MD_perc_changes)
```

After calculating the change in percent of people covered through Medi-Cal per year, we created a dataframe that also included the interval of years in which the change occured as well as the year which bills which would have an effect on percent of individuals covered passed. For example, we would expect the change observed from 2011 to 2012 to be associated with the bills passed in 2011.

Combing with bills dataset

Next, we needed to combine the dataframe with the percent changes with the bills that were passed and their associated tokens.

```
HI_passed <- read.csv("~/Downloads/HI_passed_clust_sent.csv")
token_df <- read.csv("~/Downloads/token2.0.csv")
token_df <- token_df[ ,-1]
names(token_df)[1] <- "'s"
length(unique(HI_passed$bill_id))
```

```
## [1] 206
```

Each row in token_df corresponds to a row in HI_passed. Each row is a unique bill. Next, we appended the year in which the bill was passed to the token data frame.

```
year_passed <-substr(HI_passed$status_date, 1,4)
token_df$year_passed <- year_passed
```

We were then able to merge the token dataframe with the dataframe containing the percent changes per year by merging based on the year_passed column.

```
MD_perc_token_df <- token_df %>% inner_join(MD_change_df, by="year_passed")
```

```
dim(MD_perc_token_df)
```

```
## [1]  153 6977
```

```
head(names(MD_perc_token_df))
```

```
## [1] "'s"      "a"       "ability" "about"   "access"  "account"
```

```
range(MD_perc_token_df$year_passed)
```

```
## [1] "2011" "2019"
```

```
MD_perc_token_df %>% select(year_passed, MD_perc_change) %>%head
```

	year_passed <chr>	MD_perc_change <dbl>
1	2011	0.03259116
2	2011	0.03259116
3	2011	0.03259116
4	2013	0.04495292
5	2011	0.03259116
6	2011	0.03259116

6 rows

As expected, bills that were passed during the same year should be associated with the same percent change of people covered through Medi-Cal. The rows that were removed from token_df correspond to the years before 2011 and after 2019. After merging, we are left with a dataframe of 53 bills and 6975 tokens. Given that one of the assumptions for proceeding with regression is that the number of rows (observations) is greater than the number of columns (variables/tokens), we attempted to address this issue of too many dimensions through PCA.

Principal Components Analysis (PCA) + Linear Regression

```
head(names(MD_perc_token_df))
```

```
## [1] "'s"      "a"       "ability" "about"   "access"  "account"
```

```
tail(names(MD_perc_token_df))
```

```
## [1] "urgency"        "therefor"       "renumber"       "year_passed"
## [5] "year_int"       "MD_perc_change"
```

```
tokens_only <- MD_perc_token_df %>% select(-c(year_passed, year_int, MD_perc_change))
```

```
dim(tokens_only)
```

```
## [1]  153 6974
```

```
unique(sapply(tokens_only, class))
```

```
## [1] "integer"
```

We first changed all NA values in the tokens_only datagrame to 0 and then took the log and added 1 to account for skewed counts and counts that are equal to zero.

```
tokens_only[is.na(tokens_only)] <- 0
```

```
tokens_only2 <- log(tokens_only + 1)
```

We then created a train and test set with the test set consisting of the bills passed in 2019. The test set was thus roughly 11% of the data.

```
dim(tokens_only2)
```

```
## [1]  153 6974
```

```
tokens_only2$year_passed <- MD_perc_token_df$year_passed
train_ind <- tokens_only2$year_passed != "2019"
test_ind <- !train_ind
```

```
tokens_mat <- as.matrix(tokens_only2 %>% select(-year_passed))
train <- tokens_mat[train_ind,]
test <- tokens_mat[test_ind,]
```

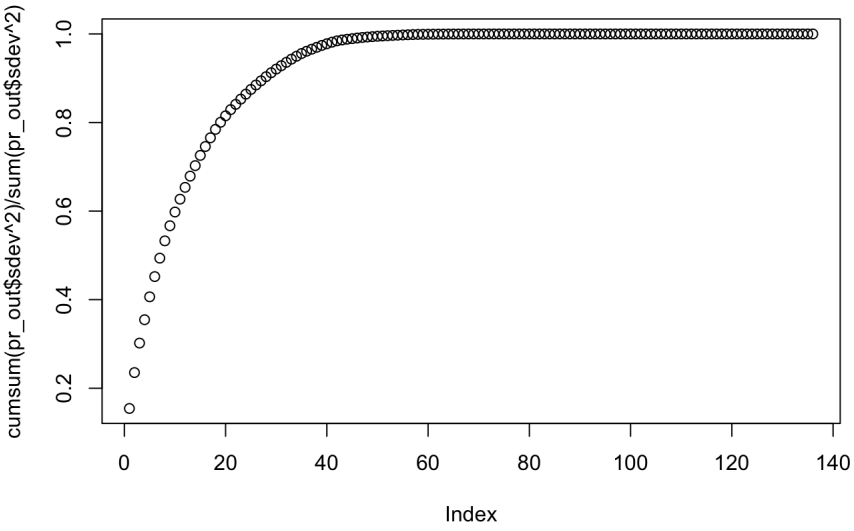
After splitting into the train and test set, there were many tokens that appeared in the test set that didnb t appear in the train set. This mean that these tokens had values of zeroes for all the bills in the test set and thus had variances of zero. As this would have prevented us from running PCA, we decided to remove these tokens from the train set and will subsequently remove them from the test set.

```
train_hasvar <- train[ , which(apply(train, 2, var) != 0)]
head(which(apply(train, 2, var)==0))
```

```
##         acs actuarially     affirms   americans  automantic      broker
##           9          13          22          41          68          88
```

```
pr_out <- prcomp(train_hasvar, scale=TRUE, center=TRUE)
```

```
plot(cumsum(pr_out$sdev^2) / sum(pr_out$sdev^2))
```




```
k <- which(cumsum(pr_out$sdev^2) / sum(pr_out$sdev^2) > 0.9)[1]
k
```

```
## [1] 28
```

Based off of the 0.9 cutoff rule, the smallest k is 28.

```
pr_train <- pr_out$x[,1:k]
perc_change_train <- MD_perc_token_df$MD_perc_change[train_ind]
design_matrix <- as.data.frame(cbind(perc_change_train, pr_train))
lm_pca <- lm(perc_change_train ~ ., data = design_matrix)
```

```
lm_summary <- summary(lm_pca)$coefficients
coeffs_ordered <- order(lm_summary[,"Pr(>|t|)"], decreasing=FALSE)
lm_pca_coefs <- lm_summary[coeffs_ordered,]
head(lm_pca_coefs)
```

```
##           Estimate   Std. Error   t value   Pr(>|t|)
## PC10  0.0008536764 0.0003566920  2.393315 0.01843847
## PC4   0.0006264677 0.0002732553  2.292610 0.02382650
## PC8   0.0004963332 0.0003176213  1.562657 0.12108609
## PC3  -0.0003732301 0.0002433030 -1.534013 0.12797794
## PC28  0.0009111978 0.0006488601  1.404305 0.16312399
## PC18  0.0006239054 0.0004552548  1.370453 0.17341374
```

After sorting the features according to p-values, we used the rotation matrix to find the top words associated with the top two loadings. These loadings were the only ones that were significant ($p<0.05$).

```
features <- rownames(lm_pca_coefs)

feat_1 <- pr_out$rotation[,features[1]]
feat_2 <- pr_out$rotation[,features[2]]

feat_1 <- abs(feat_1)
feat_2 <- abs(feat_2)

feat_1 <- feat_1[order(feat_1, decreasing=T)]
feat_2 <- feat_2[order(feat_2, decreasing=T)]

head(feat_1)
```

```
##   subsidize     married recognition     subpart     banking        bond
##  0.05599903  0.05555459  0.05204238  0.05204238  0.05204238  0.05204238
```

```
head(feat_2)
```

```
##         ph extramural     outlet     resale       ball    bearing
## 0.05057755 0.05057755 0.05057755 0.05057755 0.05057755 0.05057755
```

Next, we will used regression model to estimate the percent change in Medi-Cal coverage for the test set.

```
dim(train_hasvar)
```

```
## [1]  136 5576
```

```
dim(train)
```

```
## [1]  136 6974
```

```
dim(test)
```

```
## [1]   17 6974
```

We first subsetted the columns in the test set to match those in the train set.

```
matches_ind <- colnames(test) %in% colnames(train_hasvar)
length(matches_ind)
```

```
## [1] 6974
```

```
sum(matches_ind)
```

```
## [1] 5576
```

```
wanted_cols <- colnames(test)[which(colnames(test) %in% colnames(train_hasvar))]
test_hasvar <- test[,wanted_cols]
dim(test_hasvar)
```

```
## [1]   17 5576
```

```
pcs_test <- predict(pr_out, test_hasvar)
y_pred <- predict(lm_pca, as.data.frame(pcs_test[, 1:k]))
```

```
library(Metrics)
perc_change_test <- MD_perc_token_df$MD_perc_change[test_ind]
rmse(perc_change_test, y_pred)
```

```
## [1] 0.005649873
```

```
summary(lm_pca)$r.squared
```

```
## [1] 0.2242653
```

After PCA and regression, we obtained 2 significant loadings and the top words associated with them. Some of the top words for each of the principal components do seem associated with health insurance coverage if we look at words such as subsidize and married. Given that Medi-Cal provides health coverage to people of low-income and people generally belonging to underserved communities (https://www.coveredca.com/health/medi-cal/ (https://urldefense.proofpoint.com/v2/url?u=https-3A__www.coveredca.com_health_medi-2Dcal_&d=DwMGaQ&c=009klHSCxuh5AI1vNQzSO0KGjl4nbi2Q0M1QLJX9BeE&r=inE4QZjHDRWM9uG-Q3HITl8NuKP1vPdpVWQINHUhqAc&m=PIjFY7YdNYAecQ1RIoAf9a_pPfmsPvUUbmBEJMZNnjQ9rOHsaI6ZryVJF5CB4w3X&s=G1jFYxL_z2jxqIYE5M1RnelWkGd subsidize seems relevant. An individualb s marriage status could also be related to the type of health insurance they are covered by (https://www.coveredca.com/marketing-blog/what-newlyweds-should-know-about-health-insurance/ (https://urldefense.proofpoint.com/v2/url?u=https-3A__www.coveredca.com_marketing-2Dblog_what-2Dnewlyweds-2Dshould-2Dknow-2Dabout-2Dhealth-2Dinsurance_&d=DwMGaQ&c=009klHSCxuh5AI1vNQzSO0KGjl4nbi2Q0M1QLJX9BeE&r=inE4QZjHDRWM9uG-Q3HITl8NuKP1vPdpVWQINHUhqAc&m=PIjFY7YdNYAecQ1RIoAf9a_pPfmsPvUUbmBEJMZNnjQ9rOHsaI6ZryVJF5CB4w3X&s=ptZDWTDd5NlykdYkSvKwyEjrloV Some words donb t appear to be obviously related such as banking, bond, and extramural.

After predicting with the model, the resulting RMSE was 0.005649873. Given that the RMSE is low, it appears that the tokens associated with a bill are associated with the percent change in people covered through Medi-Cal the following year. However, the low R^2 suggests that the tokens arenb t explaining much of the variability in the percent change in coverage through Medi-Cal. This is not unexpected due to the fact that there are likely many external factors that help explain this variability. However, the results do indicate a form of association between the tokens found in bill descriptions and the percent change in Medi-Cal coverage.

Change in % of Uninsured Individuals Over Time

Finding % change per year

```
UN_peryr_count_df <- CHIS_adult_data %>%
  group_by(year) %>%
  summarize(num_row_per_year = n(),
            UN_64 = sum(INS64_P=="Uninsured"),
            UN_65 = sum(INS65=="Uninsured"))
head(UN_peryr_count_df)
```

year <chr>	num_row_per_year <int>	UN_64 <int>	UN_65 <int>
2011	22580	2648	37
2012	20355	2550	44
2013	20724	2057	42
2014	19516	1340	32
2015	21034	1436	37
2016	21055	1272	35

6 rows

```
UN_perc_df <- UN_peryr_count_df %>% mutate(UN_perc = (UN_64+UN_65)/num_row_per_year)
head(UN_perc_df)
```

year <chr>	num_row_per_year <int>	UN_64 <int>	UN_65 <int>	UN_perc <dbl>
2011	22580	2648	37	0.11891054
2012	20355	2550	44	0.12743798
2013	20724	2057	42	0.10128354
2014	19516	1340	32	0.07030129
2015	21034	1436	37	0.07002948
2016	21055	1272	35	0.06207552

6 rows

```
#calculating change per year
UN_perc_final <- UN_perc_df %>% select(c(year,UN_perc))
UN_perc_changes <- diff(UN_perc_final$UN_perc)
yr_changes <- c("2011-2012", "2012-2013", "2013-2014",
                "2014-2015", "2015-2016", "2016-2017",
                "2017-2018", "2018-2019", "2019-2020")
year_bill_passed <- c("2011","2012", "2013","2014","2015",
                      "2016","2017","2018","2019")
UN_change_df <- data.frame(year_int = yr_changes, year_passed=year_bill_passed, UN_perc_change=UN_perc_changes)
```

Combining with Bills Dataset

```
HI_passed <- read.csv("~/Downloads/HI_passed_clust_sent.csv")
token_df <- read.csv("~/Downloads/token2.0.csv")
token_df <- token_df[ ,-1]
names(token_df)[1] <- "'s"
length(unique(HI_passed$bill_id))
```

```
## [1] 206
```

```
year_passed <-substr(HI_passed$status_date, 1,4)
token_df$year_passed <- year_passed
```

```
UN_perc_token_df <- token_df %>% inner_join(UN_change_df, by="year_passed")
```

```
dim(UN_perc_token_df)
```

```
## [1]  153 6977
```

```
head(names(UN_perc_token_df))
```

```
## [1] "'s"      "a"       "ability" "about"   "access"  "account"
```

```
range(UN_perc_token_df$year_passed)
```

```
## [1] "2011" "2019"
```

```
UN_perc_token_df %>% select(year_passed, UN_perc_change) %>%head
```

	year_passed <chr>	UN_perc_change <dbl>
1	2011	0.008527436
2	2011	0.008527436
3	2011	0.008527436
4	2013	-0.030982245
5	2011	0.008527436
6	2011	0.008527436

6 rows

Principal Components Analysis (PCA) + Linear Regression

```
head(names(UN_perc_token_df))
```

```
## [1] "'s"      "a"       "ability" "about"   "access"  "account"
```

```
tail(names(UN_perc_token_df))
```

```
## [1] "urgency"        "therefor"       "renumber"       "year_passed"
## [5] "year_int"       "UN_perc_change"
```

```
tokens_only <- UN_perc_token_df %>% select(-c(year_passed, year_int, UN_perc_change))
```

```
dim(tokens_only)
```

```
## [1]  153 6974
```

```
unique(sapply(tokens_only, class))
```

```
## [1] "integer"
```

```
tokens_only[is.na(tokens_only)] <- 0
```

```
tokens_only2 <- log(tokens_only + 1)
```

```
dim(tokens_only2)
```

```
## [1]  153 6974
```

```
tokens_only2$year_passed <- UN_perc_token_df$year_passed
train_ind <- tokens_only2$year_passed != "2019"
test_ind <- !train_ind
```

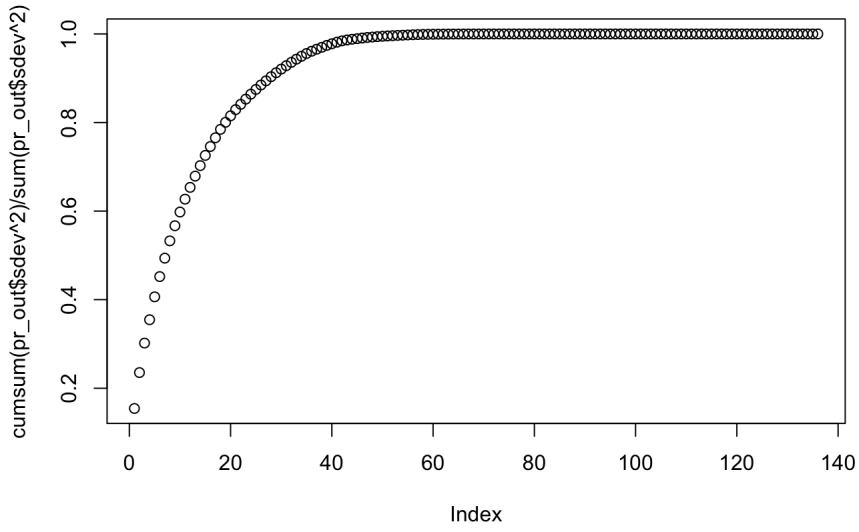
```
tokens_mat <- as.matrix(tokens_only2 %>% select(-year_passed))
train <- tokens_mat[train_ind,]
test <- tokens_mat[test_ind,]
```

```
train_hasvar <- train[ , which(apply(train, 2, var) != 0)]
head(which(apply(train, 2, var)==0))
```

```
##         acs actuarially     affirms   americans  automantic      broker
##           9          13          22          41          68          88
```

```
pr_out <- prcomp(train_hasvar, scale=TRUE, center=TRUE)
```

```
plot(cumsum(pr_out$sdev^2) / sum(pr_out$sdev^2))
```




```
k <- which(cumsum(pr_out$sdev^2) / sum(pr_out$sdev^2) > 0.9)[1]
k
```

```
## [1] 28
```

```
pr_train <- pr_out$x[,1:k]
perc_change_train <- UN_perc_token_df$UN_perc_change[train_ind]
design_matrix <- as.data.frame(cbind(perc_change_train, pr_train))
lm_pca <- lm(perc_change_train ~ ., data = design_matrix)
```

```
lm_summary <- summary(lm_pca)$coefficients
coeffs_ordered <- order(lm_summary[,"Pr(>|t|)"], decreasing=FALSE)
lm_pca_coefs <- lm_summary[coeffs_ordered,]
head(lm_pca_coefs)
```

```
##                 Estimate   Std. Error   t value     Pr(>|t|)
## (Intercept) -7.401426e-03 9.461774e-04 -7.822450 3.896739e-12
## PC25        -3.080243e-04 1.236377e-04 -2.491347 1.426161e-02
## PC1          7.422805e-05 3.237250e-05  2.292936 2.380703e-02
## PC26        -2.902045e-04 1.268599e-04 -2.287597 2.412763e-02
## PC18         1.882542e-04 9.224023e-05  2.040912 4.372107e-02
## PC7         -1.184120e-04 6.235453e-05 -1.899013 6.025615e-02
```

```
features <- rownames(lm_pca_coefs)

feat_1 <- pr_out$rotation[,features[2]]
feat_2 <- pr_out$rotation[,features[3]]
feat_3 <- pr_out$rotation[,features[4]]

feat_1 <- abs(feat_1)
feat_2 <- abs(feat_2)
feat_3 <- abs(feat_3)

feat_1 <- feat_1[order(feat_1, decreasing=T)]
feat_2 <- feat_2[order(feat_2, decreasing=T)]
feat_3 <- feat_3[order(feat_3, decreasing=T)]

head(feat_1)
```

```
##      toxic   sentence    abandon   apparent  ascertain carrollton
##  0.1146927  0.1146927  0.1146927  0.1146927  0.1146927  0.1146927
```

```
head(feat_2)
```

```
##    reserve   prohibit  committee   national     unless     system
## 0.03156659 0.03071766 0.03042044 0.03033276 0.03020163 0.03008433
```

```
head(feat_3)
```

```
##        book       mippa        fhfa origination        mass        bone
##  0.09991777  0.09991777  0.09991777  0.09991777  0.09991777  0.09991777
```

```
dim(train_hasvar)
```

```
## [1]  136 5576
```

```
dim(train)
```

```
## [1]  136 6974
```

```
dim(test)
```

```
## [1]   17 6974
```

```
matches_ind <- colnames(test) %in% colnames(train_hasvar)
length(matches_ind)
```

```
## [1] 6974
```

```
sum(matches_ind)
```

```
## [1] 5576
```

```
wanted_cols <- colnames(test)[which(colnames(test) %in% colnames(train_hasvar))]
test_hasvar <- test[,wanted_cols]
dim(test_hasvar)
```

```
## [1]   17 5576
```

```
pcs_test <- predict(pr_out, test_hasvar)
y_pred <- predict(lm_pca, as.data.frame(pcs_test[, 1:k]))
```

```
library(Metrics)
perc_change_test <- UN_perc_token_df$UN_perc_change[test_ind]
rmse(perc_change_test, y_pred)
```

```
## [1] 0.0004181913
```

```
summary(lm_pca)$r.squared
```

```
## [1] 0.291357
```

After PCA and regression, we obtained 4 significant loadings and the top words associated with them. Some of the top words for each of the principal components do seem associated with health insurance coverage in CA if we look at words such as mippa, committee, national, system. Mippa is the Medicare Improvements for Patients and Providers Act (https://www.ncoa.org/article/mippa (https://urldefense.proofpoint.com/v2/url?u=https-3A__www.ncoa.org_article_mippa&d=DwMGaQ&c=009klHSCxuh5AI1vNQzSO0KGjl4nbi2Q0M1QLJX9BeE&r=inE4QZjHDRWM9uG-Q3HlTl8NuKP1vPdpVWQINHUhqAc&m=PIjFY7YdNYAecQ1RIoAf9a_pPfmsPvUUbmBEJMZNnjQ9rOHsaI6ZryVJF5CB4w3X&s=5qnveN8zqhpDIk5G-Gp09Ky_uorzT-H4gNnRW1X4Vrs&e=)) which aims to provide grants to states and tribes. MIPPA funding is used to perform outreach and to provide financial assistance to eligible Medicare beneficiaries to help reduce their premiums and deductibles (https://www.retireguide.com/medicare/basics/history/mippa/ (https://urldefense.proofpoint.com/v2/url?u=https-3A__www.retireguide.com_medicare_basics_history_mippa_&d=DwMGaQ&c=009klHSCxuh5AI1vNQzSO0KGjl4nbi2Q0M1QLJX9BeE&r=inE4QZjHDRWM9uG-Q3HlTl8NuKP1vPdpVWQINHUhqAc&m=PIjFY7YdNYAecQ1RIoAf9a_pPfmsPvUUbmBEJMZNnjQ9rOHsaI6ZryVJF5CB4w3X&s=Cr5B_t_Na5zWhDG5CmOlp2gqZ that the amount of uninsured people is likely to be affected by national policy changes, the top words are not surprising. Some words donb t appear to be obviously related such as toxic, fhfa, and carrollton.

After predicting with the model, the resulting RMSE was 0.0004181913. Given that the RMSE is low, it appears that the tokens associated with a bill are associated with the percent change in uninsured people the following year. However, the low R^2 suggests that the tokens arenb t explaining much of the variability in the percent change in uninsured people.

Sentiment + Cluster Data Regression

Using the sentiment of each bill that was calculated earlier, along with the assigned cluster of each bill, we examined whether these features were associated with the percent change in people covered through Medi-Cal and people uninsured. ### % Change in Medi-Cal Coverage

```
#merge HI_passed with MD_change_df
year_passed <-substr(HI_passed$status_date, 1,4)
HI_passed$year_passed <- year_passed
MD_sent_df <- HI_passed %>% inner_join(MD_change_df, by="year_passed")
```

```
head(MD_sent_df %>% select(c(status_date, year_passed, MD_perc_change)))
```

	status_date <chr>	year_passed <chr>	MD_perc_change <dbl>
1	2011-01-02	2011	0.03259116
2	2011-01-04	2011	0.03259116
3	2011-04-14	2011	0.03259116
4	2013-01-02	2013	0.04495292
5	2011-11-21	2011	0.03259116
6	2011-04-15	2011	0.03259116

6 rows

```
train_ind <- MD_sent_df$year_passed != "2019"
test_ind <- !train_ind
```

```
MD_sent_df_final <- MD_sent_df %>% select(c(MD_perc_change,h_clust, cluster, sentiment))
MD_sent_df_final$h_clust <- as.factor(MD_sent_df_final$h_clust)
MD_sent_df_final$cluster <- as.factor(MD_sent_df_final$cluster)
head(MD_sent_df_final)
```

	MD_perc_change <dbl>	h_clust <fct>	cluster <fct>	sentiment <dbl>
1	0.03259116	2	1	0.9223301
2	0.03259116	2	1	0.9411765

	MD_perc_change <dbl>	h_clust <fct>	cluster <fct>	sentiment <dbl>
3	0.03259116	1	1	-0.3333333
4	0.04495292	1	1	0.3551913
5	0.03259116	1	1	0.6363636
6	0.03259116	1	1	0.4970930

6 rows

```
train <- MD_sent_df_final[train_ind,]
test <- MD_sent_df_final[test_ind,]
```

```
sent_lm <- lm(MD_perc_change~ ., data = train)
summary(sent_lm)
```

```
##
## Call:
## lm(formula = MD_perc_change ~ ., data = train)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.13366 -0.03125  0.01878  0.03762  0.07081
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.003200   0.007054   0.454    0.651
## h_clust2    -0.007933   0.014966  -0.530    0.597
## h_clust3    -0.013787   0.011283  -1.222    0.224
## cluster2     0.004657   0.034323   0.136    0.892
## cluster3     0.033214   0.030705   1.082    0.281
## sentiment    0.014445   0.011773   1.227    0.222
##
## Residual standard error: 0.055 on 130 degrees of freedom
## Multiple R-squared:  0.0388, Adjusted R-squared:  0.00183
## F-statistic:  1.05 on 5 and 130 DF,  p-value: 0.3916
```

Based off of the model including the clusters assigned through k-means and hierarchical clustering and the sentiment of each bill, these variables were not significant. Itb s not surprising that sentiment wasnb t significant given that legal text does not usually include emotionally-charged language.

% Change in Uninsured

```
#merge HI_passed with UN_change_df
year_passed <-substr(HI_passed$status_date, 1,4)
HI_passed$year_passed <- year_passed
UN_sent_df <- HI_passed %>% inner_join(UN_change_df, by="year_passed")
```

```
head(UN_sent_df %>% select(c(status_date, year_passed, UN_perc_change)))
```

	status_date <chr>	year_passed <chr>	UN_perc_change <dbl>
1	2011-01-02	2011	0.008527436
2	2011-01-04	2011	0.008527436
3	2011-04-14	2011	0.008527436
4	2013-01-02	2013	-0.030982245
5	2011-11-21	2011	0.008527436
6	2011-04-15	2011	0.008527436

6 rows

```
train_ind <- UN_sent_df$year_passed != "2019"
test_ind <- !train_ind
```

```
UN_sent_df_final <- UN_sent_df %>% select(c(UN_perc_change,h_clust, cluster, sentiment))
UN_sent_df_final$h_clust <- as.factor(UN_sent_df_final$h_clust)
UN_sent_df_final$cluster <- as.factor(UN_sent_df_final$cluster)
head(UN_sent_df_final)
```

	UN_perc_change	h_clust	cluster	sentiment
	<dbl>	<fct>	<fct>	<dbl>
1	0.008527436	2	1	0.9223301
2	0.008527436	2	1	0.9411765
3	0.008527436	1	1	-0.3333333
4	-0.030982245	1	1	0.3551913
5	0.008527436	1	1	0.6363636
6	0.008527436	1	1	0.4970930

6 rows

```
train <- UN_sent_df_final[train_ind,]
test <- UN_sent_df_final[test_ind,]
```

```
sent_lm <- lm(UN_perc_change~ ., data = train)
summary(sent_lm)
```

```
##
## Call:
## lm(formula = UN_perc_change ~ ., data = train)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.023626 -0.008950  0.003996  0.008037  0.017057
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0083093  0.0015168  -5.478 2.14e-07 ***
## h_clust2     0.0033351  0.0032182   1.036    0.302
## h_clust3     0.0009526  0.0024263   0.393    0.695
## cluster2    -0.0039935  0.0073806  -0.541    0.589
## cluster3    -0.0011832  0.0066025  -0.179    0.858
## sentiment    0.0006611  0.0025316   0.261    0.794
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.01183 on 130 degrees of freedom
## Multiple R-squared:  0.01099,    Adjusted R-squared:  -0.02705
## F-statistic: 0.2888 on 5 and 130 DF,  p-value: 0.9185
```

Similar to the model for the percent change in Medi-Cal, this model indicated that the clusters assigned through k-means and hierarchical clustering and the sentiment of each bill were not significant.

Conclusion

Our project aimed to answer the question of whether changes in health insurance over time can be explained by policy changes occurring either in California and on a national level. During our initial Random Forest model used to identify demographic features associated with health insurance type, we expect income to be highly predictive. The results of that analysis confirmed our expectations. However, we were surprised to see a lack of demographic info such as race or first-generation status have a large effect. As we moved towards explaining the trends in health insurance coverage through clustering and sentiment analysis of the bills based on their tokens, we didnb t expect the sentiment of the bill to show a large effect, but we did expect to see more meaningful clusters. We were surprised to see none related to specific sub-populations, but we werenb t surprised to see some that focused specifically on Medi-Cal. The insight provided from the clustering was mainly found by close examination of the hierarchcical clusters and evaluating the yearly trends in the number of Medi-Cal bills passed. We observed a similar trend between the number of bills passed and the degree of change in percent of people covered between 2014-2015 and 2018-2019. Based off of this insight, we decided to focus on the percent change in people covered through Medi-Cal as well as the percent change in uninsured people.

We proceeded with calculating the percent change in people covered through Medi-Cal as well as the percent change in uninsured people over the range of years included in the CHIS dataset. After merging this with the billsb tokens based on the year the bills were passed, we found an association between a billsb tokens and the percent change for both Medi-cal coverage and uninsured individuals.

In regards to the robustness of our results, we got significant p values for some of the top words obtained after PCA and regression which indicates that there was a significant correlation between bills containing these words and changes in percent of Medi-cal coverage and the uninsured. The health survey data was based on a sample of the California population. The data collected was extensive and efforts were made to try and make sure that it was representative of the entire California population. In regard to the CHIS data, there thus does not seem to be too much uncertainty in terms of the quality of the data and its ability to accurately present changes in insurance coverage over time. If the data included different respondents or if the algorithms were tested on a different year, the results of the algorithm would likely be similar.

The large uncertainty that is introduced comes from the creation of the billsb descriptions. The descriptions were not compiled by us and therefore we donb t have control over how they were created. If this analysis was run again on the entire text of the bill it is very possible that different words would show up as significant. The sentences that were chosen for the bill description could potentially have a strong effect on the results. If a dataset with different bill descriptions was used then the results of the analyses we ran could vary.

In regard to the potential implications of data snooping for our project, there was a lot of our own judgement put in in regard to including which bills to tokenize and use to predict the percent change in coverage per year. For example, we first filtered the bills to find those related to health insurance and then we decided to look at percent change in Medi-Cal coverage based on the understanding that one cluster of bills was specifically related to Medi-Cal. While these decisions were made with the intent of answering our question, they also introduced some bias in terms of which population of respondents we were interested in.

Ultimately, the finding that there is an association between a billsb tokens and the percent change for both Medi-Cal coverage and uninsured individuals, does not indicate a causal relationship between legislation and trends in health insurance. However, it can provide grounds for closer examination of the nature of the bills passed and the trends in health insurance coverage the following year. Given that much time is dedicated to drafting policies related to health care and that California continues to push efforts to increase the number of individuals eligible for Medi-Cal, it seems worthwhile to carefully examine the effects of these policies on decreasing the percent of uninsured people. Furthermore, an interesting future direction would be to explore how changes in health insurance coverage over time are associated with changes in health outcomes (ex. decreased rates of asthma, better) and health access (ex. more timely availability of doctor appointments) over time.

Critique

We critiqued Julia Angkeowb s and Karina Fengb s project.

What was the initial motivation for tackling the project?

The initial motivation for the project was based on a theory called event segmentation that hypothesizes that working memory is affected by how a given stimulus was perceived. It has been hypothesized that people parse events into different logical segments that help them encode events and recall information from memory. They are hoping to explore the questions of whether 1) Can an algorithm be created that will help psychologists identify the story that participants were assigned according to their recall? and 2) Does priming matter in memory recall? Do people recall stories in a certain schematic? The results of this project could be useful or psychologists, researchers, and even people in advertising to figure out how memory works.

What datasets were used?

The project uses a dataset of 373 participants along with 16 recalls and their story IDs and priming IDs. This dataset was then converted into two large datasets through 1) Universal sentence encoder (converts each recall into large numerical vectors for analysis) and 2) TFIDF (converts each recall into numerical values that highlight how important a given token is in a recall).

What aspect of the project is considered a data-mining and what is discovered?

They used data mining to convert the recall into tokens and to then use these tokens to explore whether there was an effect of priming group on recall. Using the tokens, theyb re hoping to separate into clusters based on story id. They are also trying to outline an algorithm that will help psychologists identify the story that participants were assigned according to their recall. The act of identifying which words/tokens are important in separating out stories would be difficult to do given the sheer number of stories and number of words per story so data mining was a useful tool to implement.

With the clustering results they hope to identify the top words per cluster and examine whether the clustering algorithm successfully separated the clusters based on priming group. If they are able to effectively separate on priming group, then the results will show that priming has an effect on recall.

Is there anything you would have done differently?

We would have considered other algorithms besides clustering with TF-IDF such as sentiment or PCA on the words frequency matrix. We would also have used a method that could test how well the recall token words predicted story ID.