Process book
Group P- Healthcare coverage in the US

Team Members:

- Grace Yun Xin Kong (gyk2108@columbia.edu)
- Iris Shih-Yin Chen (sc4071@tc.columbia.edu)
- Fang Liu (fl2407@columbia.edu)

Overview and Motivation:

One thing that was discussed a lot during the last president election is Obamacare. The Patient Protection and Affordable Care Act (ACA)， also commonly known as Obamacare, was a major and wide-sweeping healthcare reform initiative enacted by President Obama. The ACA was signed into law by Obama on March 23, 2010, and changes were gradually implemented over the next few years, though the majority of the provisions went into effect only on January 1, 2014. Trump criticized Obamacare a lot. So will the ACA survive into 2017? Does it depend on the results of the 2016 election? Whether ACA is as terrible as trump said?

To answer those questions, we are interested in finding out how health insurance coverage in the USA differs from state to state, its relationship to indicators of health performance, and whether it can be predicted by political factors. More specifically, we want to see the impact of Affordable care act on the insurance coverage rate.

Quantitatively, we want see the change over time and the difference between states. We want to know in which states did insurance coverage expand under Obamacare? Which states observed the greatest decline in their uninsured rate? How does insurance coverage differ from state to state? We will also control some variables like income, mortality rate and rates of major diseases.

And then, we want to see whether higher insurance rate really benefit people. And how does health insurance coverage relate to participation uptake in preventative measures and treatment options (such as vaccinations, mammograms)?

And finally, we want to know what people think about Obamacare. To realize this, we will do some text analysis and see how do the sentiments towards Obamacare differ from state to state, and how does it relate to voting patterns? If feasible, we plan to analyze sentiments from twitter data, whether Obamacare is mentioned alongside positive or negative words.

We will make use of ggplot2, spatial data techniques, and text mining techniques in our visualization project.

Data

Our project need:

1. General state-level data (population, age distribution and income distribution).

- United States Census Bureau: https://www.census.gov/data.html

The United States Census bureau provides dataset that include information like population, average income and age distribution.

2. Insurance coverage and health indicators datasets

- State data on the Affordable Care Act (ACA) (from US National Library of Medicine): https://aspe.hhs.gov/compilation-state-data-affordable-care-act

  This state-level dataset includes coverage rates of the ACA, growth and expansion status of the ACA, employer vs. individual market coverage, and Medicaid / Medicare numbers. (filename: states_aca_general)

- State scorecard on various health indicators (from the Commonwealth Fund): http://datacenter.commonwealthfund.org/#ind=1/sc=1

  This state-level dataset has collated information about the health performance of the states along various dimensions. It includes insurance coverage rates, participation rates in prevention and treatment activities, and some measures of mortality. It also provides some differentiating information by race and income, allowing us to explore issues of equity.

- Health Insurance Marketplace Data (from US Department of Health and Human Services): https://www.kaggle.com/hhs/health-insurance-marketplace

  This dataset contains information on health and dental plans offered through the US Health Insurance Marketplace.

3. Twitter API data & New York Times API data

   For the text analysis, we use twitter API data and New York Times API data.

   For the twitter API, we will try to extract text data on individuals' thoughts and opinions on Obamacare, capturing positive and negative sentiments.

   And we downloaded the New York Times articles from LexisNexis and then use R to transform them into corpus that can be analysis with R.

Exploratory Data Analysis

We first cleaned our data and unified the name of some variables as the data sets have different sources. The more challenging part is the API data. At first we planned to download all tweets that mentioned ACA from 2008 to 2016. However, we realized that will be too much data and will be hard to clean up and analysis. So we decided to download the New York Times articles and do the long term text analysis based on the New York Times articles. For twitter API, we only focus on the most recent 1000 tweets and try to have a sense of people’s opinion about Affordable care act after the election.

- Download API Data

Get some tweets from Twitter to analyze and visualize Set up Twitter API: Selecting data including Obamacare, ACA, Affordable Care Act, and #ACA (n=1000)

```
library(httr)
#library(oauth)
library(ROAuth)
library(twitteR)
library(RCurl)
library(RJSONIO)
library(stringr)
#secretkey
#myapp <- oauth_app("twitter",
	# key = "liLn6XJFenGjtvWFwi5LnDS1M",
	# secret = "dsCBm9Kyaeu9GMKlM9xwKl7eKmDn6qsjP31LQtwMGkF6OQdLh6")
#Get OAuth credentials
#twitter_token <- oauth1.0_token(oauth_endpoints("twitter"), myapp)

# Declare Twitter API Credentials
api_key <- "liLn6XJFenGjtvWFwi5LnDS1M" # From dev.twitter.com
api_secret <- "dsCBm9Kyaeu9GMKlM9xwKl7eKmDn6qsjP31LQtwMGkF6OQdLh6" # From
dev.twitter.com
token <- "772176811455381505-PYuNAEqhHFc02r83WS9Y5dnsZciIY5v" # From dev.twitter.com
token_secret <- "mgRPwKeHZEw9Y486h2GMtCBxDztPfQLX1Msd5vog1hiwv" # From dev.twitter.com

# Create Twitter Connection
library("base64enc")
setup_twitter_oauth(api_key, api_secret, token, token_secret)

# Run Twitter Search. Format is searchTwitter("Search Terms", n=100, lang="en", geocode="lat,lng",
also accepts since and until).

tweets <- searchTwitter("Obamacare OR ACA OR 'Affordable Care Act' OR #ACA", n=1000, lang="en",
since="2014-08-20")

# Transform tweets list into a data frame
tweets.df <- twListToDF(tweets)
head(tweets.df,3)
```

Who was tweeting most in the sample we collected

```
counts=table(tweets.df$screenName)
barplot(counts)
# Let's do something hacky:
# Limit the data set to show only folk who tweeted twice or more in the sample
cc=subset(counts,counts>1)
barplot(cc,las=2,cex.names =0.3)
```

- Download and convert the New York time articles

The New York Times articles are downloaded from LexisNexis as two html files. And we used the code provide by professor to convert the data. The code didn’t work well at first and then we figured out it is because we there is an option to download the key word in bold and we selected that, and as we re-downloaded the html files the code works.

```
library(tm)
```

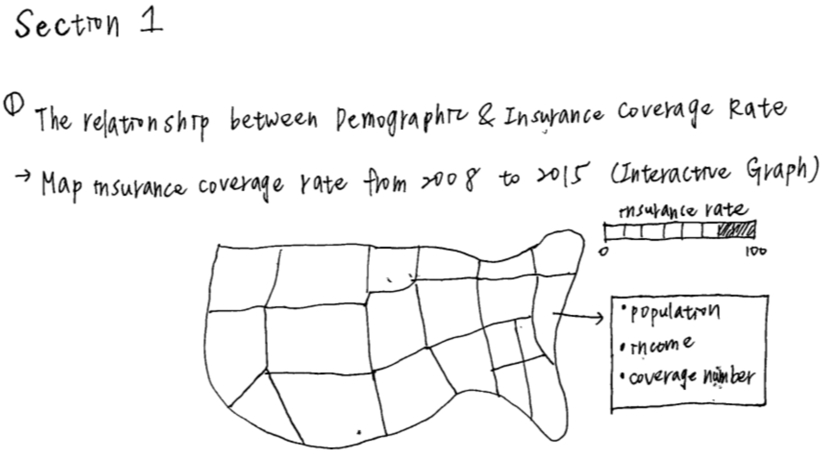
```
library(tm.plugin.lexisnexis)
library(readxl)
library(gtools)  # for smartbind
library(dplyr)   # for data_frame
library(lubridate)   # for date formatting
library(stringr)
library(tools)  # Title case
library(quanteda)
library(ggplot2)

# Combine CSV and HTML Files
data <- read_excel("LexisNexis/NYTimes_Metadata.xlsx")
colnames(data) <- tolower(colnames(data))

# Correct data
data$date <- substr(parse_date_time(data$date, c("mdy")),1,10)
data$author <- toTitleCase(tolower(data$byline))
data$byline <- NULL

## Get Text files
source1 <- LexisNexisSource("LexisNexis/The_New_York1.html")
source2 <- LexisNexisSource("LexisNexis/The_New_York2.html")

corpus1 <- Corpus(source1, readerControl = list(language = NA))
corpus2 <- Corpus(source2, readerControl = list(language = NA))

corpus <- c(corpus1, corpus2)

# Convert to quanteda corpus
corpus <- quanteda::corpus(corpus)

## Add Metadata
# Check: match(data$headline, corpus$documents$heading)
corpus$documents$datetimestamp <- substring(corpus$documents$datetimestamp, 1,4)
corpus$documents$date <- corpus$documents$datetimestamp
corpus$documents$description <- corpus$documents$id

# options(width = 200)
# kwic(corpus, "Trump")
#
save(corpus, file="nytimes.rda")
```

Design Evolution:

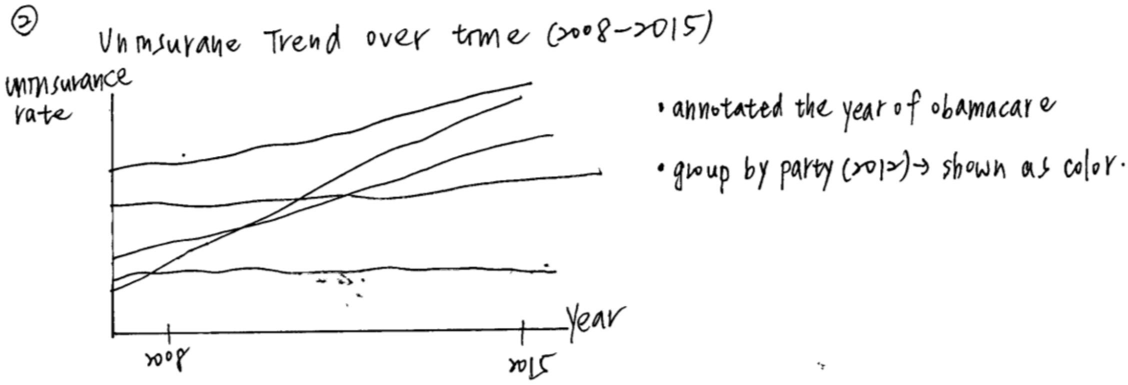
We started from drawing the graphs on paper. As we already have a general idea of our datasets, we were able to create some graph that can visualize information from different datasets. We divided our project into three sections.

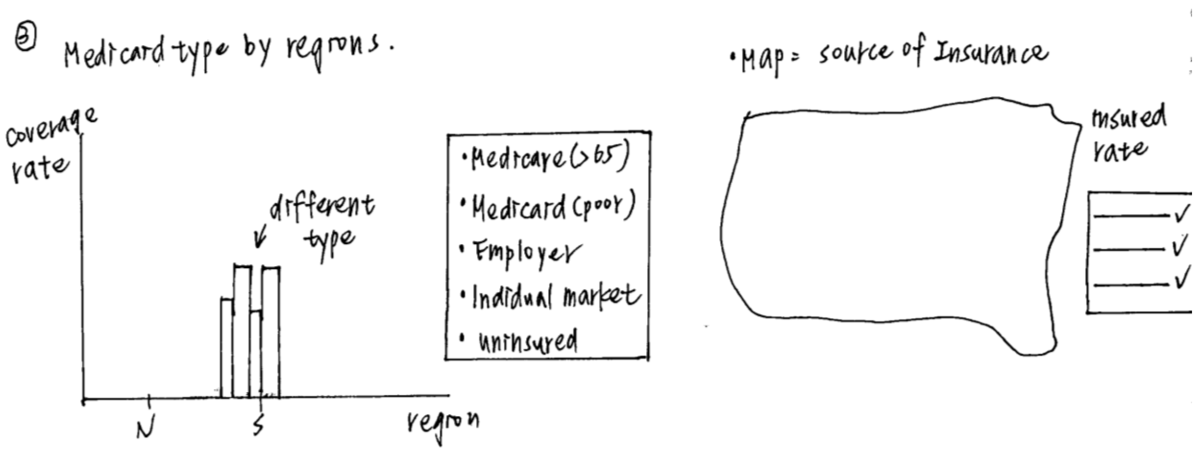
The first section focus on the general insurance coverage in the US.

Graph 1: The first graph will show the general domestic information and also an insurance coverage rate.



Graph 2 and 3 focus on insurance rate. Graph 2 will show the trend of uninsured rate while graph 3 is a bar graph that will show the source of insurance by region



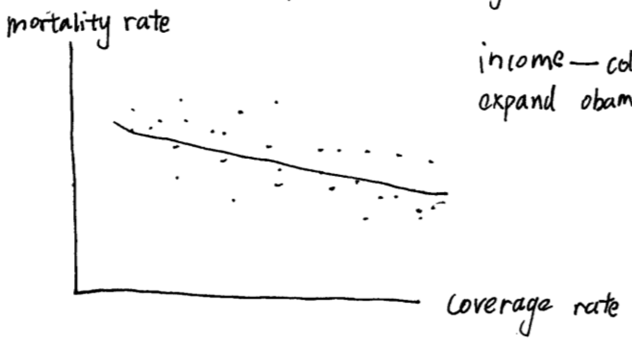


Section 2 tries to see the insurance coverage in depth. We want to control some variables and see their relationship with insurance coverage rate.

The graph 4 is supposed to show the relationship between mortality rate and insurance coverage rate. We want to control some omitted variable like income, and expansion of Obamacare,

Section 2

graph 4: mortality rate and insurance coverage rate (2015)



income — color

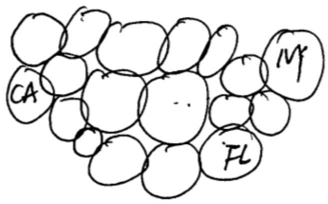
expand obamacare or not — shape.

Q: whether we want separate graphs for different age groups

Graph 5 will try to evaluate the relationship between the access of healthcare and insurance coverage. We divided health services into several categories based on our dataset like vaccine, dental and so on.

graph 5. Access of healthcare.

Bubble map.



- insurance coverage. — color
- drop down (access to different services)
  - cancer
  - vaccine
  - dental

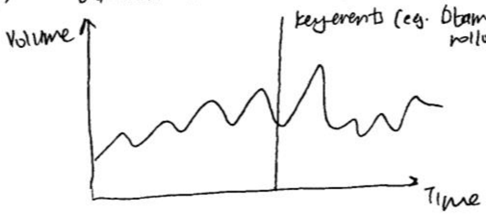
one for 2008, one for 2015.

The section 3 is the text analysis. We planned to first have a look of the change of numbers of tweets mentioned ACA and then sentiment analysis of tweets mentioned ACA over years.

Section #3 : Sentiment Analysis of Tweets on healthcare / Obamacare

⑥ Frequency of Tweets Mentioning Various Topics (over time)

(A) Overall volume of Tweets (on healthcare)



(B) Mentions of specific topics - multi-graph plot

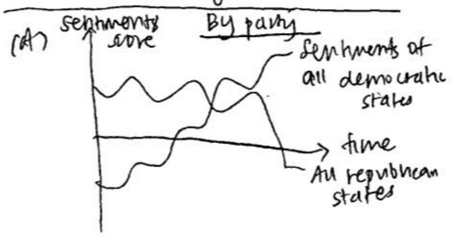
... etc...

Data format

Tweets	Date (with Year)	Location (⇑ state)	Democratic/Republican state	Sentiment score

⑦ Sentiment Analysis of Tweets

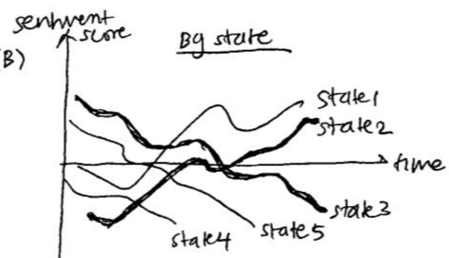
(A) By party



(2 lines)

- sentiment score calculated using dictionary of positive / negative words

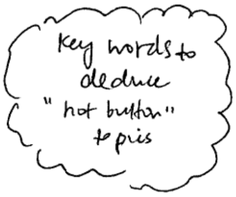
(B) By state



(50 lines - one for each state

map color : whether state has expanded.

⑧ Wordcloud of Tweets mentioning Obamacare



→ could divide into D/R states as well

→ using TF-IDF score ?

Working process

Part 1

```
m(list=ls())

library(leaflet)
library(maps)
library(rgdal)
library(DT)
library(ggplot2)
library(ggthemes)
library(plotly)
library(magrittr)
library(readxl)
library(plyr)
library(dplyr)
library(tidyr)
library(readr)
library(stringi)
library(RColorBrewer)
library(countrycode)
require(gridExtra)
## IMPORT DATA

# General ACA information
aca_general_raw <- read_excel("Data/states_aca_general.xlsx", sheet=2, col_names=FALSE)

# States health outcomes
health_ind_raw <- read_excel("Data/states_health_ind.xlsx", sheet=4, col_names=FALSE)

# Insurance coverage data
  # Long format
insurance_long <- read_excel("Data/insurance-clean.xlsx", sheet=1, col_names=TRUE)
  # Wide format
insurance <- read_excel("Data/insurance-clean.xlsx", sheet=2, col_names=TRUE)

# Demographic information
demographics1 <- read.csv("Data/Population_Age_Income.csv", header = TRUE)
  # Provided with Assignment 4 (want the yearly population data)
demographics2 <- read_excel("Data/PopulationEstimates.xls")

# Election Results
election <- read_excel("Data/US_Presidential_Results_by_State_1828-2016.xlsx", sheet=2,
col_names=FALSE)
## CLEAN DATA

# List of states abbreviations
state_abbreviations <- c("AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE", "FL", "GA", "HI", "ID", "IL",
"IN", "IA", "KS", "KY", "LA", "ME", "MD", "MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH",
"NJ", "NM", "NY", "NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC", "SD", "TN", "TX", "UT", "VT", "VA",
"WA", "WV", "WI", "WY")
```

```
## CLEAN INSURANCE RATES DATA (WIDE)

# Drop unnecessary columns
insurance <- insurance[1:50, -1]

# Rename variables
colnames(insurance) <- c("state_abb", "state", "uninsured_num_2008", "uninsured_pct_2008",
"uninsured_num_2009", "uninsured_pct_2009", "uninsured_num_2010", "uninsured_pct_2010",
"uninsured_num_2011", "uninsured_pct_2011", "uninsured_num_2012", "uninsured_pct_2012",
"uninsured_num_2013", "uninsured_pct_2013", "uninsured_num_2014", "uninsured_pct_2014",
"uninsured_num_2015", "uninsured_pct_2015")

# Create variables for insurance rates
insurance$insured_pct_2008 <- 100 - insurance$uninsured_pct_2008
insurance$insured_pct_2009 <- 100 - insurance$uninsured_pct_2009
insurance$insured_pct_2010 <- 100 - insurance$uninsured_pct_2010
insurance$insured_pct_2011 <- 100 - insurance$uninsured_pct_2011
insurance$insured_pct_2012 <- 100 - insurance$uninsured_pct_2012
insurance$insured_pct_2013 <- 100 - insurance$uninsured_pct_2013
insurance$insured_pct_2014 <- 100 - insurance$uninsured_pct_2014
insurance$insured_pct_2015 <- 100 - insurance$uninsured_pct_2015

# Reorder variables
insurance <- insurance[ , c("state_abb", "state", "uninsured_num_2008", "uninsured_pct_2008",
"insured_pct_2008", "uninsured_num_2009", "uninsured_pct_2009", "insured_pct_2009",
"uninsured_num_2010", "uninsured_pct_2010", "insured_pct_2010", "uninsured_num_2011",
"uninsured_pct_2011", "insured_pct_2011", "uninsured_num_2012", "uninsured_pct_2012",
"insured_pct_2012", "uninsured_num_2013", "uninsured_pct_2013", "insured_pct_2013",
"uninsured_num_2014", "uninsured_pct_2014", "insured_pct_2014", "uninsured_num_2015",
"uninsured_pct_2015", "insured_pct_2015")]


## CLEAN INSURANCE RATES DATA (LONG)

# Rename variables
colnames(insurance_long) <- c("year", "state", "population", "uninsured_num", "uninsured_pct",
"insured_num", "insured_pct")

# Add state abbreviations
  # First create a look-up table (based on wide data)
state_abb_lookup <- insurance[ , c("state", "state_abb")]
  # Then do the matching
insurance_long$state_abb <- state_abb_lookup$state_abb[match(insurance_long$state,
state_abb_lookup$state)]
  # Reorder variable
insurance_long <- insurance_long[ , c(1:2, ncol(insurance_long), 3:(ncol(insurance_long)-1))]

# Order observations (by state, then by year)
insurance_long <- insurance_long[order(insurance_long$state, insurance_long$year), ]
```

```
## CLEAN ACA GENERAL DATA

# Keep only relevant rows and columns
aca_general <- aca_general_raw[6:57, 1:73]

# Label variables for ACA general data
names(aca_general) <- c("state", "unins_all_pct_10", "unins_all_pct_15", "unins_all_decr_pct",
"unins_all_decr", "cov_emp", "cov_parents_plan", "lifetime_lim_preACA_tot",
"lifetime_lim_preACA_child", "lifetime_lim_preACA_adultM", "lifetime_lim_preACA_adultF",
"cov_private_tot", "cov_private_child", "cov_private_adultM", "cov_private_adultF",
"premium_emp_avg_growth_pct_00to10", "premium_emp_avg_growth_pct_10to15",
"premium_emp_savings_15", "premium_emp_savings_16", "MLR_rebate_beneficiaries_12",
"MLR_rebate_amt_12", "MLR_rebate_beneficiaries_13", "MLR_rebate_amt_13",
"MLR_rebate_beneficiaries_14", "MLR_rebate_amt_14", "MLR_rebate_beneficiaries_15",
"MLR_rebate_amt_15", "MLR_rebate_amt_12to15", "medicaid_enroll_13", "medicaid_enroll_16",
"medicaid_enroll_child_13", "medicaid_enroll_incr_13to16", "medicaid_full_duals",
"medicaid_partial_duals", "medicaid_full_or_partial_duals", "state_has_expanded",
"insurance_incr_medicaid", "cholest_scr_incr_medicaid", "mammogram_incr_medicaid",
"papsmear_incr_medicaid", "clinic_care_incr_medicaid", "all_care_incr_medicaid",
"phycisian_visit_ann_incr_medicaid", "depression_decr_medicaid", "good_health_incr_medicaid",
"deaths_ann_decr_medicaid", "catastrophic_oop_ann_decr_medicaid",
"indebted_ppl_decr_medicaid", "fed_spending_net_incr_inMil",
"uncompensated_care_decr_inMil", "mental_substance_elig_medicaid_share",
"mental_substance_elig_medicaid", "preexisting_condition_09", "cov_mkt_plan_16",
"cov_mkt_16", "receive_tax_credit", "avg_tax_credit", "receive_cost_sharing",
"avg_num_mkt_plans_avail_17", "cov_mkt_under75D_pct", "cov_mkt_under100D_pct",
"cov_offmkt_elig_tax_credit", "rate_review_funds_to_state", "HIECP_grant_award_to_state",
"medicare_enroll_16", "medicare_benef_donuthole", "medicare_benef_donuthole_savings_tot",
"medicare_benef_donuthole_savings_avg", "medicare_partB_free_prev_services",
"medicare_partB_free_prev_services_share", "medicare_incr_readmit_rate",
"medicare_avoided_readmit", "accountable_care_org_num")

# Trim white space on state variable
aca_general$state <- stri_trim_both(aca_general$state)

# Change format of variables
  # Apart from T/F variable for "state has expanded"
aca_general$state_has_expanded[aca_general$state_has_expanded=="yes"] <- TRUE
aca_general$state_has_expanded[aca_general$state_has_expanded=="no"] <- FALSE
  # Convert others to numeric form
for (j in c(2:35, 37:73)) {
  aca_general[[j]] <- as.numeric(aca_general[[j]])
}

# Save USA data separately
aca_general_USA <- aca_general[aca_general$state=="United States", ]

# Drop USA and DC data
aca_general <- aca_general[!(aca_general$state %in% c("United States", "District of Columbia")), ]
```

```
# Add state abbreviations
aca_general$state_abb <- state_abbreviations
aca_general <- aca_general[ , c(1, ncol(aca_general), 2:(ncol(aca_general)-1))]

# Drop raw data
rm(aca_general_raw)


## CLEAN HEALTH INDICATOR DATA

# Keep only relevant rows and columns
health_ind <- health_ind_raw[4:6703, c(1, 3:6)]

# Label variables
names(health_ind) <- c("state", "measure", "year", "rate", "rank")

# Simplify date (years) - only keep last year if range was given
health_ind$year[health_ind$year=="07/2009 - 06/2012"] <- "2012"
health_ind$year[health_ind$year=="07/2010 - 06/2013"] <- "2013"
health_ind$year[health_ind$year=="07/2012 - 06/2015"] <- "2015"
health_ind$year[health_ind$year=="10/2013-9/2014"] <- "2014"
health_ind$year[health_ind$year=="2008-09"] <- "2009"
health_ind$year[health_ind$year=="2010-11"] <- "2011"
health_ind$year[health_ind$year=="2011-12"] <- "2012"
health_ind$year[health_ind$year=="2011/12"] <- "2012"
health_ind$year[health_ind$year=="2012-13"] <- "2013"
health_ind$year[health_ind$year=="2013-14"] <- "2014"
health_ind$year[health_ind$year=="2013(Q2-Q4)"] <- "2013"
health_ind$year[health_ind$year=="2014-15"] <- "2015"
health_ind$year[health_ind$year=="2015(Q2-Q4)"] <- "2015"

# Export to excel to manually label variable names
unique_measures <- unique(as.factor(health_ind$measure))
  # Copied in from excel sheet
unique_measures_relabelled <- c("a.summary_access", "a.unins_adult", "a.unins_child",
"a.no_care_bc_cost_adult", "a.high_OOP_relative_under65", "a.at_risk_no_routine_doc_adult",
"a.no_dental_adult", "q.summary_prev_treat", "q.with_usual_care_adult",
"q.with_cancer_screening_adult", "q.with_vaccines_adult", "q.with_medical_home_child",
"q.with_prev_medical_dental_child", "q.with_mental_healthcare_child", "q.with_vaccines_infant",
"q.drug_should_avoid_medicare", "q.drug_should_avoid_3conditions_medicare",
"q.good_health_provider_medicare", "q.mortality_4conditions_medicare",
"q.CLABSI_infection_ratio", "q.info_recovery_hospitalized", "q.good_hospital_staff_hospitalized",
"q.improve_mobility_homehealth", "q.improved_wounds_homehealth", "q.sores_NHres",
"q.antipsychotic_med_NHres", "u.summary_avoidable_hosp_cost", "u.hosp_asthma_child",
"u.hosp_ambulatory_65to74yrs", "u.hosp_ambulatory_above75yrs",
"u.30day_hosp_readmit_medicare", "u.30day_hosp_readmit_NHres", "u.hosp_6mos_NHres",
"u.hosp_medicare_homehealth", "u.avoidable_ER_medicare", "u.tot_reimb_employer_ins",
"u.tot_reimb_medicare", "h.summary_healthy_lives", "h.deaths_amenable",
"h.yrs_lost_potential_life_before75", "h.deaths_breast_cancer_F", "h.deaths_colorectal_cancer",
"h.deaths_suicide", "h.deaths_infant_mortality", "h.poor_health_adult", "h.smoke_adult",
```

```
"h.obese_adult", "h.obese_child", "h.poor_dental_adult", "u.premium_emp_private",
"u.premium_emp_private_unadj", "u.reimb_medicare_unadj", "u.deaths_amenable_black",
"u.deaths_amenable_white", NA, "q.with_prev_screening_above50yrs", NA)

# Replace variable names with abbreviated version
for (i in 1:57) {
  health_ind$measure <- replace(health_ind$measure, health_ind$measure==unique_measures[i],
unique_measures_relabelled[i])
}

# Remove the NA's
health_ind <- health_ind[!is.na(health_ind$measure), ]

# Remove duplicates
  # By state, measure.year
health_ind <- health_ind[!duplicated(health_ind[ , 1:3]), ]

# Interact measure and year
health_ind$measure_year <- interaction(health_ind$measure, health_ind$year)

# Convert to character variable
health_ind$measure_year <- as.character(health_ind$measure_year)

# Pre-reshape: Store existing data as "long"
health_ind_long <- health_ind

# Reshape wide
health_ind <- as.data.frame(cbind(health_ind$state, health_ind$measure_year, health_ind$rate))
  # Drop rank data for now
names(health_ind) <- c("state", "measure_year", "rate")
health_ind <- reshape(health_ind,
                      idvar = "state",
                      timevar = "measure_year",
                      direction = "wide")

# Drop "rate" from the name
names(health_ind)[2:ncol(health_ind)] <- substr(names(health_ind), 6,
nchar(names(health_ind)))[2:ncol(health_ind)]

# Drop variables who have all entries as NA
  # Due to 'unbalanced panel' in reshaping
  # First create a vector to record if variable is all NA
variable_all_NA <- vector(mode="logical", length=ncol(health_ind))
for (j in 1:ncol(health_ind)) {
  if(sum(is.na(health_ind[[j]]))==52) {
    variable_all_NA[j] <- TRUE
  }
}
  # Only keep the variables that are NOT all NA
health_ind <- health_ind[ , !variable_all_NA]
```

```
# Change format of variables
  # To character variable
health_ind$state <- as.character(health_ind$state)
  # To numeric variable
for (j in c(2:164)) {
  health_ind[[j]] <- as.character(health_ind[[j]])
  health_ind[[j]] <- as.numeric(health_ind[[j]])
}

# Save USA data separately
health_ind_USA <- health_ind[health_ind$state=="United States", ]

# Drop USA and DC data
health_ind <- health_ind[!(health_ind$state %in% c("United States", "District of Columbia")), ]

# Label state abbreviations
health_ind$state_abb <- state_abbreviations
health_ind <- health_ind[ , c(1, ncol(health_ind), 2:(ncol(health_ind)-1))]

# Drop raw data
rm(health_ind_raw)


## CLEAN DEMOGRAPHIC (1) DATA

# Drop unnecessary row and column
demographics1 <- demographics1[-1,-1]

# Rename variables
colnames(demographics1) <- c("state","income","population",
"ppl_age0to4","ppl_age5to9","ppl_age10to14","ppl_age15to19","ppl_age20to24",
"ppl_age25to29","ppl_age30to34","ppl_age35to39","ppl_age40to44",
"ppl_age45to49","ppl_age50to54","ppl_age55to59","ppl_age60to64",
"ppl_age65to69","ppl_age70to74","ppl_age75to79","ppl_age80to84","ppl_age85plus")

# Drop observations DC and PR
demographics1 <- demographics1[!demographics1$state=="District of Columbia"
& !demographics1$state=="Puerto Rico", ]

# Label state abbreviations
demographics1$state_abb <- state_abbreviations
demographics1 <- demographics1[ , c(1, ncol(demographics1), 2:(ncol(demographics1)-1))]

# Change format of variables
  # To character variable
demographics1$state <- as.character(demographics1$state)
  # To numeric variable
for (j in c(3:ncol(demographics1))) {
  demographics1[[j]] <- as.character(demographics1[[j]])
  demographics1[[j]] <- as.numeric(demographics1[[j]])
}
```

```
# BEA regions
demographics1$BEA_region <- ""
demographics1$BEA_region[demographics1$state_abb %in% c("ME", "NH", "VT", "MA", "CT", "RI")]
<- "New England"
demographics1$BEA_region[demographics1$state_abb %in% c("NY", "NJ", "PA", "MD", "DE")] <-
"Mideast"
demographics1$BEA_region[demographics1$state_abb %in% c("WI", "IL", "MI", "IN", "OH")] <-
"Great Lakes"
demographics1$BEA_region[demographics1$state_abb %in% c("WV", "KY", "VA", "TN", "NC", "SC",
"AR", "LA", "MS", "AL", "GA", "FL")] <- "Southeast"
demographics1$BEA_region[demographics1$state_abb %in% c("ND", "SD", "NE", "KS", "MN", "IA",
"MO")] <- "Plains"
demographics1$BEA_region[demographics1$state_abb %in% c("MT", "ID", "WY", "UT", "CO")] <-
"Rocky Mountains"
demographics1$BEA_region[demographics1$state_abb %in% c("AZ", "NM", "TX", "OK")] <-
"Southwest"
demographics1$BEA_region[demographics1$state_abb %in% c("WA", "OR", "CA", "NV", "AK", "HI")]
<- "Far West"

# Tag income level (category)
demographics1$income_level <- ""
demographics1$income_level[demographics1$income >= 60000] <- "High"
demographics1$income_level[demographics1$income >= 50000 & demographics1$income < 60000]
<- "Upper Middle"
demographics1$income_level[demographics1$income >= 45000 & demographics1$income < 50000]
<- "Lower Middle"
demographics1$income_level[demographics1$income < 45000] <- "Low"
  # Convert to factor variable
demographics1$income_level <- factor(demographics1$income_level, levels = c("Low", "Lower
Middle", "Upper Middle", "High"))


## CLEAN DEMOGRAPHIC (2) DATA

colnames(demographics2) <- demographics2[2, ]

# Drop first few rows (non-observations)
demographics2 <- demographics2[-(1:2), ]

# Drop county-level data,  keep state-level data only
demographics2 <- demographics2[demographics2$Area_Name == "United States" |
demographics2$State != lag(demographics2$State), ]

# Keep only columns of population from 2010 to 2015
demographics2 <- demographics2[ , c("State", "Area_Name", "POP_ESTIMATE_2010",
"POP_ESTIMATE_2011", "POP_ESTIMATE_2012", "POP_ESTIMATE_2013", "POP_ESTIMATE_2014",
"POP_ESTIMATE_2015")]

# Change format of variables
  # To character variable
```

```
for (j in c(3:8)) {
  demographics2[[j]] <- as.numeric(demographics2[[j]])
}

# Rename variables
colnames(demographics2) <- c("state_abb", "state", "population_2010", "population_2011",
"population_2012", "population_2013", "population_2014", "population_2015")


## CLEAN ELECTION DATA

# Keep only 2012 and 2008 data, and data for the 50 states
election <- election[ 4:53 , c("X0", "X5", "X6", "X9", "X10")]

# Rename variables
colnames(election) <- c("state", "Dem_pct_2012", "Rep_pct_2012", "Dem_pct_2008",
"Rep_pct_2008")

# Add state abbreviations
  # Match with look-up table
election$state_abb <- state_abb_lookup$state_abb[match(election$state, state_abb_lookup$state)]

# Change format of variables
  # To numeric variable
for (j in c(4:6)) {
  election[[j]] <- as.numeric(election[[j]])
}

# Calculate winning party for each state
election$party_2012 <- ifelse(election$Dem_pct_2012 > election$Rep_pct_2012, "Democratic",
"Republican")
election$party_2008 <- ifelse(election$Dem_pct_2008 > election$Rep_pct_2008, "Democratic",
"Republican")

# Reorder variables
election <- election[ , c("state", "state_abb", "Dem_pct_2012", "Rep_pct_2012", "party_2012",
"Dem_pct_2008", "Rep_pct_2008", "party_2008")]
```

Health Insurance Coverage

Figure 1

Click the link to access our Shiny App on Uninsured Rates in the United States from 2008-2015.

```
## SOME DATA PREPARATION

# Look-up information into insurance and insurance_long, on:
  # Party (2012)
insurance_long$party_2012 <- election$party_2012[match(insurance_long$state, election$state)]
insurance$party_2012 <- election$party_2012[match(insurance$state, election$state)]
  # BEA region
```

```
insurance_long$BEA_region <- demographics1$BEA_region[match(insurance_long$state,
demographics1$state)]
insurance$BEA_region <- demographics1$BEA_region[match(insurance$state,
demographics1$state)]
  # Income level
insurance_long$income_level <- demographics1$income_level[match(insurance_long$state,
demographics1$state)]
insurance$income_level <- demographics1$income_level[match(insurance$state,
demographics1$state)]

# Also find nation-wide trend for insurance
insurance_long_US <- data.frame("year" = c(2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015),
"uninsured_pct" = c(16.8, 17.5, 18.2, 17.2, 16.9, 16.6, 13.3, 10.5))
```

Figure 2A

The following figure displays the trend in the uninsured rate between 2008 and 2015. Results are shown for the USA average (in black), as well as for every individual state (in color). The states are group by Bureau of Economic Analysis (BEA) regions - this allows us to analyze regional trends, and also to identify a state by filtering down to its respective region on this interactive plot.

We see a clear downward trend in the uninsured rate once most of the Affordable Care Act (ACA) provisions were implemented in 2014. The overall US uninsured rate decreased from 18.2% in 2010 to 10.5% in 2015.

However, note that the disparities in uninsured rates largely continue to persist among the states, with regional patterns. States of the Southeast and Southwest consistently had the highest uninsured rates, while states of New England consistenely had the lowest insured or highest health insurance coverage rates.

```
## PLOT 2A

plot2a <- (ggplot(insurance_long, aes(year, uninsured_pct))
        + geom_line(aes(color = BEA_region, group = state, label = uninsured_num))
        + scale_x_continuous(breaks = c(2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015))
        + scale_y_continuous(breaks = c(5, 10, 15, 20, 25, 30))
        + labs(x = "Year", y = "% Uninsured", color = "Region")
        + geom_line(data = insurance_long_US, lwd = 0.8)
        + geom_vline(xintercept = 2010, linetype = 2, color = "gray30")
        + geom_vline(xintercept = 2014, linetype = 2, color = "gray30")
        + annotate("text",x = 2009.4, y = 25, label = "Obama \nsigns ACA", color = "black",fontface = 2,
size = 3)
        + annotate("text",x = 2013.4, y = 24.5, label = "Most ACA \nprovisions \ntake effect", color =
"black",fontface = 2, size = 3)
        + annotate("text",x = 2014.2, y = 11, label = "USA \nAverage", color = "black",fontface = 2, size
= 3)
        + ggtitle("Change in Percentage Uninsured by State, \nfrom 2008 to 2015")
        + theme_minimal()
        + theme(plot.title = element_text(face="bold", hjust = 0, size = 12)))
ggplotly(plot2a, dynamicTicks = FALSE, tooltip = c("state", "year", "uninsured_pct",
"uninsured_num"))
```

Figure 2B

The following plot provides a visual depiction of the narrowing in dispersal of the states' uninsured rates over the period of 2008 to 2015. It can be seen that by 2015, there is a concentration of states' uninsured rates especially within the range of 7.5% to 12.5%.

```
## PLOT 2B
plot2b <- (ggplot(insurance_long, aes(year, uninsured_pct))
        + geom_violin(scale = "area", aes(group = year), fill = "dark blue", color = NA)
        + scale_x_continuous(breaks = c(2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015))
        + scale_y_continuous(breaks = c(5, 10, 15, 20, 25, 30))
        + labs(x = "Year", y = "% Uninsured")
        + ggtitle("Narrowing of Uninsured Rates Across States \nfrom 2008 to 2015")
        + theme_minimal()
        + theme(plot.title = element_text(face="bold", hjust = 0.5, size = 12)))
plot2b
```

Figure 3

In the following plot, we explore the difference between Democratic and Republican states, based on the 2012 US Presidential Election results. Generally, the democratic states consistently have had largely have a lower uninsured rate than the Republican states, though both groups experienced declines in the uninsured rates in 2014. The rate of decline of uninsured rates in 2014 for some Republican states appeared to exceed that of the Democratic states, mainly because of their higher starting point.

```
## PLOT 3
plot3 <- (ggplot(insurance_long, aes(year, uninsured_pct))
        + geom_line(aes(color = party_2012, group = state, label = uninsured_num))
        + scale_x_continuous(breaks = c(2008, 2009, 2010, 2011, 2012, 2013, 2014, 2015))
        + scale_y_continuous(breaks = c(5, 10, 15, 20, 25, 30))
        + scale_color_manual(values = c("steel blue", "firebrick1"))
        + labs(x = "Year", y = "% Uninsured", color = "Party \n(2012 Winner)")
        + geom_line(data = insurance_long_US, lwd = 1)
        + geom_vline(xintercept = 2010, linetype = 2, color = "gray30")
        + geom_vline(xintercept = 2014, linetype = 2, color = "gray30")
        + annotate("text",x = 2009.4, y = 25, label = "Obama \nsigns ACA", color = "black",fontface = 2,
size = 3)
        + annotate("text",x = 2013.4, y = 24.5, label = "Most ACA \nprovisions \ntake effect", color =
"black",fontface = 2, size = 3)
        + annotate("text",x = 2014.2, y = 11, label = "USA \nAverage", color = "black",fontface = 2, size
= 3)
        + ggtitle("Change in Percentage Uninsured by Party, \nfrom 2008 to 2015")
        + theme_minimal()
        + theme(plot.title = element_text(face="bold", hjust = 0, size = 12)))
ggplotly(plot3, dynamicTicks = FALSE,  tooltip = c("state", "year", "uninsured_pct",
"uninsured_num"))
```

Figure 4

We also know that the Medicaid expansion was found to be a big driver of the fall in uninsured rates (accounting for more than half of the decrease). Hypothesizing that this was highly correlated to party affiliation, we made the following bar graph. It confirmed our suspicions that Democratic states were more likely to opt for Medicaid expansion, whereas Republican states were more likely to opt out. This could be a possible explanation for the continued lower insured rates in Democratic states after 2014.

```
# Merge in Medicaid expansion data
insurance$expanded_medicaid <- aca_general$state_has_expanded[match(insurance$state,
aca_general$state)]

plot4 <- (ggplot(insurance, aes(label = state_abb))
  + geom_bar(aes(x = party_2012, fill = expanded_medicaid), position = "dodge")
  + scale_fill_manual(values = brewer.pal(n = 2, name = "Dark2")[c(2,1)])
  + labs(x = "Party (2012 Winner)", y = "Number of States", fill = "Expanded \nMedicaid")
  + ggtitle("Medicaid Expansion and Party of State")
  + theme_minimal()
  + theme(plot.title = element_text(face="bold", hjust = 0.5, size = 12)))
plot4
```

Figure 5

We want to provide a clearer comparison across of the gains in insurance coverage due to the ACA. Thus, we plotted the decrease in uninsured rate (or equivalently, the increase in insured rate) for the states from 2010 to 2015. We find that it is the West Coast states of Oregon, Nevada and California that led in gains in insurance coverage. Republican states are not all ranked at the bottom, and many of them are in the middle of the pack (again due to their higher starting point).

```
insurance$unins_pct_decr_10_15 <- insurance$uninsured_pct_2010 -
insurance$uninsured_pct_2015
insurance$unins_num_decr_10_15 <- insurance$uninsured_num_2010 -
insurance$uninsured_num_2015

## PLOT 5
plot5 <- (ggplot(insurance, aes(x = reorder(state, unins_pct_decr_10_15), y = unins_pct_decr_10_15,
fill = party_2012, label = unins_num_decr_10_15))
  + geom_col()
  + coord_flip()
  + labs(x = "", y = "Decrease in % Uninsured", fill = "Party \n(2012 Winner)")
  + scale_fill_manual(values = c("steel blue", "firebrick1"))
  + ggtitle("Decrease in in Percentage Uninsured from 2010 to 2015")
  + theme_minimal()
  + theme(plot.title = element_text(face="bold", hjust = 0, size = 12), axis.text.y = element_text(size =
6.5))
)
ggplotly(plot5, dynamicTicks = FALSE,  tooltip = c("unins_pct_decr_10_15",
"unins_num_decr_10_15"))
```

Figure 6A

This plot shows insured rates in 2015 at a glance, looking at how they differ by party (2012 winner), BEA region and income level of the states.

In terms of correlations, it appears that being a Democratic state is a strong predictor of a higher insurance rate. States in New England, the Mideast and the Great Lakes perform well in insurance rates, while states in the Southwest, Rocky Mountains and Southeast perform the poorest. As expected, high-income states have the highest insurance rates as more people would be able to afford insurance. Lower and lower-middle income states have lower coverage rates, but it is infact the lower-middle income states that have the lowest insurance rates.

```
## PLOT 6A
```

```
# Subset relevant variables
insurance_2010_2015 <- insurance[ , c("state", "insured_pct_2010", "unins_pct_decr_10_15",
"insured_pct_2015", "party_2012", "BEA_region", "income_level")]
names(insurance_2010_2015)[names(insurance_2010_2015)=="unins_pct_decr_10_15"] <-
"ins_pct_incr_10_15"

# Boxplots by party
plot6a1 <- (ggplot(insurance_2010_2015, aes(x = reorder(party_2012, insured_pct_2015), y =
insured_pct_2015))
        + geom_boxplot(aes(fill = party_2012, label = insured_pct_2015), color = "gray30")
        + coord_flip()
        + scale_fill_manual(values = c("steel blue", "firebrick"))
        + ggtitle("Democratic vs. Republican States")
        + theme_minimal()
        + theme(plot.title = element_text(face="bold", hjust = 0.37, size = 12), legend.position =
"none", axis.title.x = element_blank(), axis.title.y = element_blank()))

# Boxplots by BEA region
plot6a2 <- (ggplot(insurance_2010_2015, aes(x = reorder(BEA_region, insured_pct_2015), y =
insured_pct_2015))
        + geom_boxplot(aes(fill = BEA_region, label = insured_pct_2015), color = "gray30")
        + coord_flip()
        + scale_fill_manual(values = brewer.pal(n = 8, name = "Dark2"))
        + ggtitle("By Bureau of Economic Analysis (BEA) Region")
        + theme_minimal()
        + theme(plot.title = element_text(face="bold", hjust = 0.24, size = 12), legend.position =
"none", axis.title.x = element_blank(), axis.title.y = element_blank()))

# Boxplots by income level
plot6a3 <- (ggplot(insurance_2010_2015, aes(x = income_level, y = insured_pct_2015))
        + geom_boxplot(aes(fill = income_level, label = insured_pct_2015), color = "gray30")
        + coord_flip()
        + scale_fill_manual(values = brewer.pal(n = 4, name = "Dark2"))
        + labs(x = "", y = "% Insured (2015)")
        + ggtitle("By State's Income Level")
        + theme_minimal()
        + theme(plot.title = element_text(face="bold", hjust = 0.39, size = 12), legend.position =
"none", axis.title.y = element_blank()))

# Combine plots vertically
grid.arrange(plot6a1, plot6a2, plot6a3, nrow = 3, top = "2015 Insurance Rates at a Glance")
```

Figure 6B

We now make these same categorizations, but looking at the gains in insurance coverage between 2010 and 2015. We notice that Republican states at actually have higher gains in insurance rate than Democratic states (although Democratic states have a higher 3rd quartile). The Far West, Southwest and Southeast make the greatest gains in insurance coverage, but still had the lowest insured rates in 2015. The lower the income of the state, the greater the gains in insurance coverage although they still end up with the lowest insurance rates in 2015.

Figures 6A and 6B together suggest that the trends by party, region and income have been preserved from 2010 to 2015, but the gap between the high-performing groups and the low-performing groups has narrowed.

```
## PLOT 6B

# Boxplots by party
plot6b1 <- (ggplot(insurance_2010_2015, aes(x = reorder(party_2012, ins_pct_incr_10_15), y =
ins_pct_incr_10_15))
        + geom_boxplot(aes(fill = party_2012, label = ins_pct_incr_10_15), color = "gray30")
        + coord_flip()
        + scale_fill_manual(values = c("steel blue", "firebrick"))
        + ggtitle("Democratic vs. Republican States")
        + theme_minimal()
        + theme(plot.title = element_text(face="bold", hjust = 0.37, size = 12), legend.position =
"none", axis.title.x = element_blank(), axis.title.y = element_blank()))

# Boxplots by BEA region
plot6b2 <- (ggplot(insurance_2010_2015, aes(x = reorder(BEA_region, ins_pct_incr_10_15), y =
ins_pct_incr_10_15))
        + geom_boxplot(aes(fill = BEA_region, label = ins_pct_incr_10_15), color = "gray30")
        + coord_flip()
        + scale_fill_manual(values = brewer.pal(n = 8, name = "Dark2"))
        + ggtitle("By Bureau of Economic Analysis (BEA) Region")
        + theme_minimal()
        + theme(plot.title = element_text(face="bold", hjust = 0.24, size = 12), legend.position =
"none", axis.title.x = element_blank(), axis.title.y = element_blank()))

# Boxplots by income level
plot6b3 <- (ggplot(insurance_2010_2015, aes(x = income_level, y = ins_pct_incr_10_15))
        + geom_boxplot(aes(fill = income_level, label = ins_pct_incr_10_15), color = "gray30")
        + coord_flip()
        + scale_fill_manual(values = brewer.pal(n = 4, name = "Dark2"))
        + labs(x = "", y = "Increase in % Insured")
        + ggtitle("By State's Income Level")
        + theme_minimal()
        + theme(plot.title = element_text(face="bold", hjust = 0.39, size = 12), legend.position =
"none", axis.title.y = element_blank()))

# Combine plots vertically
grid.arrange(plot6b1, plot6b2, plot6b3, nrow = 3, top = "Increase in Insurance Rates (2010-2015) at a
Glance")
## SOME DATA RESHAPING

# Reshape aca_general (long), keeping the interesting quantities
  # Import uninsured number (2015)
aca_general$unins_all_15 <- insurance$uninsured_num_2015[match(aca_general$state,
insurance$state)]
  # Keep only select variables
aca_general_long <- aca_general[ , c("state", "state_abb", "cov_emp", "cov_mkt_plan_16",
"medicaid_enroll_16", "medicare_enroll_16", "unins_all_15")]
  # The actual reshaping
```

```
aca_general_long <- reshape(aca_general_long,
                varying = c("cov_emp", "cov_mkt_plan_16", "medicaid_enroll_16",
"medicare_enroll_16", "unins_all_15"),
                v.names = "Number",
                timevar = "Source",
                times = c("Employer", "Marketplace", "Medicaid / CHIP", "Medicare", "Uninsured"),
                new.row.names = 1:1000,
                direction = "long")

# Look-up information into aca_general_long, on:
  # Party (2012)
aca_general_long$party_2012 <- election$party_2012[match(aca_general_long$state,
election$state)]
  # BEA region
aca_general_long$BEA_region <- demographics1$BEA_region[match(aca_general_long$state,
demographics1$state)]
  # Income level
aca_general_long$income_level <- demographics1$income_level[match(aca_general_long$state,
demographics1$state)]
  # Population
aca_general_long$population_2015 <-
demographics2$population_2015[match(aca_general_long$state, demographics2$state)]

# Calculate percentage from number and population
aca_general_long$percentage <- aca_general_long$Number / aca_general_long$population_2015 *
100
```

Figure 7

Next, we investigate the main health insurance and coverage sources, namely employer coverage, the individual marketplace, Medicaid / Childrens' Health Insurance Program (CHIP), and Medicare for the aged. We broke down the results by party and income level, taking a simple (unweighted) average across the states in the categories. We observe that employer coverage rates increase as income level increases, especially for Democratic states. Democratic and low income states have more people on Medicaid / CHIP than Republican and low income states.

```
## PLOT 7

aca_general_long_by_party_income <- ddply(aca_general_long, c("party_2012", "income_level",
"Source"), summarise, avg_pct = mean(percentage))
aca_general_long_by_party_income$avg_pct <-
round(aca_general_long_by_party_income$avg_pct, digits = 1)

plot7 <- (ggplot(aca_general_long_by_party_income, aes(income_level, avg_pct, fill = Source))
        + geom_col(position = "dodge") + facet_grid(party_2012 ~ .)
        + labs(x = "Income Level", y = "Average % in State", color = "Source of \nInsurance")
        + scale_fill_manual(values = brewer.pal(n = 6, name = "Dark2")[c(1:3,6,4)])
        + ggtitle("Sources of Health Insurance, \nby Party and Income Level")
        + theme_minimal()
        + theme(plot.title = element_text(face="bold", hjust = 0, size = 12)))
ggplotly(plot7, tooltip = c("income_level", "Source", "avg_pct"))
rm(list=ls())
```

```
library(leaflet)
library(maps)
library(rgdal)
library(DT)
library(ggplot2)
library(ggthemes)
library(plotly)
library(magrittr)
library(readxl)
library(plyr)
library(dplyr)
library(tidyr)
library(readr)
library(stringi)
library(RColorBrewer)
library(countrycode)
require(gridExtra)
## IMPORT DATA

# General ACA information
aca_general_raw <- read_excel("Data/states_aca_general.xlsx", sheet=2,
col_names=FALSE)

# States health outcomes
health_ind_raw <- read_excel("Data/states_health_ind.xlsx", sheet=4,
col_names=FALSE)

# Insurance coverage data
  # Long format
insurance_long <- read_excel("Data/insurance-clean.xlsx", sheet=1,
col_names=TRUE)
  # Wide format
insurance <- read_excel("Data/insurance-clean.xlsx", sheet=2,
col_names=TRUE)

# Demographic information
demographics1 <- read.csv("Data/Population_Age_Income.csv", header = TRUE)
  # Provided with Assignment 4 (want the yearly population data)
demographics2 <- read_excel("Data/PopulationEstimates.xls")

# Election Results
election <- read_excel("Data/US_Presidential_Results_by_State_1828-
2016.xlsx", sheet=2, col_names=FALSE)
## CLEAN DATA

# List of states abbreviations
state_abbreviations <- c("AL", "AK", "AZ", "AR", "CA", "CO", "CT", "DE",
"FL", "GA", "HI", "ID", "IL", "IN", "IA", "KS", "KY", "LA", "ME", "MD",
"MA", "MI", "MN", "MS", "MO", "MT", "NE", "NV", "NH", "NJ", "NM", "NY",
"NC", "ND", "OH", "OK", "OR", "PA", "RI", "SC", "SD", "TN", "TX", "UT",
"VT", "VA", "WA", "WV", "WI", "WY")


## CLEAN INSURANCE RATES DATA (WIDE)
```

```
# Drop unnecessary columns
insurance <- insurance[1:50, -1]

# Rename variables
colnames(insurance) <- c("state_abb", "state", "uninsured_num_2008",
"uninsured_pct_2008", "uninsured_num_2009", "uninsured_pct_2009",
"uninsured_num_2010", "uninsured_pct_2010", "uninsured_num_2011",
"uninsured_pct_2011", "uninsured_num_2012", "uninsured_pct_2012",
"uninsured_num_2013", "uninsured_pct_2013", "uninsured_num_2014",
"uninsured_pct_2014", "uninsured_num_2015", "uninsured_pct_2015")

# Create variables for insurance rates
insurance$insured_pct_2008 <- 100 - insurance$uninsured_pct_2008
insurance$insured_pct_2009 <- 100 - insurance$uninsured_pct_2009
insurance$insured_pct_2010 <- 100 - insurance$uninsured_pct_2010
insurance$insured_pct_2011 <- 100 - insurance$uninsured_pct_2011
insurance$insured_pct_2012 <- 100 - insurance$uninsured_pct_2012
insurance$insured_pct_2013 <- 100 - insurance$uninsured_pct_2013
insurance$insured_pct_2014 <- 100 - insurance$uninsured_pct_2014
insurance$insured_pct_2015 <- 100 - insurance$uninsured_pct_2015

# Reorder variables
insurance <- insurance[ , c("state_abb", "state", "uninsured_num_2008",
"uninsured_pct_2008", "insured_pct_2008", "uninsured_num_2009",
"uninsured_pct_2009", "insured_pct_2009", "uninsured_num_2010",
"uninsured_pct_2010", "insured_pct_2010", "uninsured_num_2011",
"uninsured_pct_2011", "insured_pct_2011", "uninsured_num_2012",
"uninsured_pct_2012", "insured_pct_2012", "uninsured_num_2013",
"uninsured_pct_2013", "insured_pct_2013", "uninsured_num_2014",
"uninsured_pct_2014", "insured_pct_2014", "uninsured_num_2015",
"uninsured_pct_2015", "insured_pct_2015")]


## CLEAN INSURANCE RATES DATA (LONG)

# Rename variables
colnames(insurance_long) <- c("year", "state", "population",
"uninsured_num", "uninsured_pct", "insured_num", "insured_pct")

# Add state abbreviations
  # First create a look-up table (based on wide data)
state_abb_lookup <- insurance[ , c("state", "state_abb")]
  # Then do the matching
insurance_long$state_abb <-
state_abb_lookup$state_abb[match(insurance_long$state,
state_abb_lookup$state)]
  # Reorder variable
insurance_long <- insurance_long[ , c(1:2, ncol(insurance_long),
3:(ncol(insurance_long)-1))]

# Order observations (by state, then by year)
insurance_long <- insurance_long[order(insurance_long$state,
insurance_long$year), ]
```

```
## CLEAN ACA GENERAL DATA

# Keep only relevant rows and columns
aca_general <- aca_general_raw[6:57, 1:73]

# Label variables for ACA general data
names(aca_general) <- c("state", "unins_all_pct_10", "unins_all_pct_15",
"unins_all_decr_pct", "unins_all_decr", "cov_emp", "cov_parents_plan",
"lifetime_lim_preACA_tot", "lifetime_lim_preACA_child",
"lifetime_lim_preACA_adultM", "lifetime_lim_preACA_adultF",
"cov_private_tot", "cov_private_child", "cov_private_adultM",
"cov_private_adultF", "premium_emp_avg_growth_pct_00to10",
"premium_emp_avg_growth_pct_10to15", "premium_emp_savings_15",
"premium_emp_savings_16", "MLR_rebate_beneficiaries_12",
"MLR_rebate_amt_12", "MLR_rebate_beneficiaries_13", "MLR_rebate_amt_13",
"MLR_rebate_beneficiaries_14", "MLR_rebate_amt_14",
"MLR_rebate_beneficiaries_15", "MLR_rebate_amt_15",
"MLR_rebate_amt_12to15", "medicaid_enroll_13", "medicaid_enroll_16",
"medicaid_enroll_child_13", "medicaid_enroll_incr_13to16",
"medicaid_full_duals", "medicaid_partial_duals",
"medicaid_full_or_partial_duals", "state_has_expanded",
"insurance_incr_medicaid", "cholest_scr_incr_medicaid",
"mammogram_incr_medicaid", "papsmear_incr_medicaid",
"clinic_care_incr_medicaid", "all_care_incr_medicaid",
"phycisian_visit_ann_incr_medicaid", "depression_decr_medicaid",
"good_health_incr_medicaid", "deaths_ann_decr_medicaid",
"catastrophic_oop_ann_decr_medicaid", "indebted_ppl_decr_medicaid",
"fed_spending_net_incr_inMil", "uncompensated_care_decr_inMil",
"mental_substance_elig_medicaid_share", "mental_substance_elig_medicaid",
"preexisting_condition_09", "cov_mkt_plan_16", "cov_mkt_16",
"receive_tax_credit", "avg_tax_credit", "receive_cost_sharing",
"avg_num_mkt_plans_avail_17", "cov_mkt_under75D_pct",
"cov_mkt_under100D_pct", "cov_offmkt_elig_tax_credit",
"rate_review_funds_to_state", "HIECP_grant_award_to_state",
"medicare_enroll_16", "medicare_benef_donuthole",
"medicare_benef_donuthole_savings_tot",
"medicare_benef_donuthole_savings_avg",
"medicare_partB_free_prev_services",
"medicare_partB_free_prev_services_share", "medicare_incr_readmit_rate",
"medicare_avoided_readmit", "accountable_care_org_num")

# Trim white space on state variable
aca_general$state <- stri_trim_both(aca_general$state)

# Change format of variables
  # Apart from T/F variable for "state has expanded"
aca_general$state_has_expanded[aca_general$state_has_expanded=="yes"] <-
TRUE
aca_general$state_has_expanded[aca_general$state_has_expanded=="no"] <-
FALSE
  # Convert others to numeric form
for (j in c(2:35, 37:73)) {
```

```
  aca_general[[j]] <- as.numeric(aca_general[[j]])
}

# Save USA data separately
aca_general_USA <- aca_general[aca_general$state=="United States", ]

# Drop USA and DC data
aca_general <- aca_general[!(aca_general$state %in% c("United States",
"District of Columbia")), ]

# Add state abbreviations
aca_general$state_abb <- state_abbreviations
aca_general <- aca_general[ , c(1, ncol(aca_general),
2:(ncol(aca_general)-1))]

# Drop raw data
rm(aca_general_raw)


## CLEAN HEALTH INDICATOR DATA

# Keep only relevant rows and columns
health_ind <- health_ind_raw[4:6703, c(1, 3:6)]

# Label variables
names(health_ind) <- c("state", "measure", "year", "rate", "rank")

# Simplify date (years) - only keep last year if range was given
health_ind$year[health_ind$year=="07/2009 - 06/2012"] <- "2012"
health_ind$year[health_ind$year=="07/2010 - 06/2013"] <- "2013"
health_ind$year[health_ind$year=="07/2012 - 06/2015"] <- "2015"
health_ind$year[health_ind$year=="10/2013-9/2014"] <- "2014"
health_ind$year[health_ind$year=="2008-09"] <- "2009"
health_ind$year[health_ind$year=="2010-11"] <- "2011"
health_ind$year[health_ind$year=="2011-12"] <- "2012"
health_ind$year[health_ind$year=="2011/12"] <- "2012"
health_ind$year[health_ind$year=="2012-13"] <- "2013"
health_ind$year[health_ind$year=="2013-14"] <- "2014"
health_ind$year[health_ind$year=="2013(Q2-Q4)"] <- "2013"
health_ind$year[health_ind$year=="2014-15"] <- "2015"
health_ind$year[health_ind$year=="2015(Q2-Q4)"] <- "2015"

# Export to excel to manually label variable names
unique_measures <- unique(as.factor(health_ind$measure))
  # Copied in from excel sheet
unique_measures_relabelled <- c("a.summary_access", "a.unins_adult",
"a.unins_child", "a.no_care_bc_cost_adult", "a.high_OOP_relative_under65",
"a.at_risk_no_routine_doc_adult", "a.no_dental_adult",
"q.summary_prev_treat", "q.with_usual_care_adult",
"q.with_cancer_screening_adult", "q.with_vaccines_adult",
"q.with_medical_home_child", "q.with_prev_medical_dental_child",
"q.with_mental_healthcare_child", "q.with_vaccines_infant",
"q.drug_should_avoid_medicare",
"q.drug_should_avoid_3conditions_medicare",
```

```
"q.good_health_provider_medicare", "q.mortality_4conditions_medicare",
"q.CLABSI_infection_ratio", "q.info_recovery_hospitalized",
"q.good_hospital_staff_hospitalized", "q.improve_mobility_homehealth",
"q.improved_wounds_homehealth", "q.sores_NHres",
"q.antipsychotic_med_NHres", "u.summary_avoidable_hosp_cost",
"u.hosp_asthma_child", "u.hosp_ambulatory_65to74yrs",
"u.hosp_ambulatory_above75yrs", "u.30day_hosp_readmit_medicare",
"u.30day_hosp_readmit_NHres", "u.hosp_6mos_NHres",
"u.hosp_medicare_homehealth", "u.avoidable_ER_medicare",
"u.tot_reimb_employer_ins", "u.tot_reimb_medicare",
"h.summary_healthy_lives", "h.deaths_amenable",
"h.yrs_lost_potential_life_before75", "h.deaths_breast_cancer_F",
"h.deaths_colorectal_cancer", "h.deaths_suicide",
"h.deaths_infant_mortality", "h.poor_health_adult", "h.smoke_adult",
"h.obese_adult", "h.obese_child", "h.poor_dental_adult",
"u.premium_emp_private", "u.premium_emp_private_unadj",
"u.reimb_medicare_unadj", "u.deaths_amenable_black",
"u.deaths_amenable_white", NA, "q.with_prev_screening_above50yrs", NA)

# Replace variable names with abbreviated version
for (i in 1:57) {
  health_ind$measure <- replace(health_ind$measure,
health_ind$measure==unique_measures[i], unique_measures_relabelled[i])
}

# Remove the NA's
health_ind <- health_ind[!is.na(health_ind$measure), ]

# Remove duplicates
  # By state, measure.year
health_ind <- health_ind[!duplicated(health_ind[ , 1:3]), ]

# Interact measure and year
health_ind$measure_year <- interaction(health_ind$measure,
health_ind$year)

# Convert to character variable
health_ind$measure_year <- as.character(health_ind$measure_year)

# Pre-reshape: Store existing data as "long"
health_ind_long <- health_ind

# Reshape wide
health_ind <- as.data.frame(cbind(health_ind$state,
health_ind$measure_year, health_ind$rate))
  # Drop rank data for now
names(health_ind) <- c("state", "measure_year", "rate")
health_ind <- reshape(health_ind,
                        idvar = "state",
                        timevar = "measure_year",
                        direction = "wide")

# Drop "rate" from the name
```

```
names(health_ind)[2:ncol(health_ind)] <- substr(names(health_ind), 6,
nchar(names(health_ind)))[2:ncol(health_ind)]

# Drop variables who have all entries as NA
  # Due to 'unbalanced panel' in reshaping
  # First create a vector to record if variable is all NA
variable_all_NA <- vector(mode="logical", length=ncol(health_ind))
for (j in 1:ncol(health_ind)) {
  if(sum(is.na(health_ind[[j]]))==52) {
    variable_all_NA[j] <- TRUE
  }
}
  # Only keep the variables that are NOT all NA
health_ind <- health_ind[ , !variable_all_NA]

# Change format of variables
  # To character variable
health_ind$state <- as.character(health_ind$state)
  # To numeric variable
for (j in c(2:164)) {
  health_ind[[j]] <- as.character(health_ind[[j]])
  health_ind[[j]] <- as.numeric(health_ind[[j]])
}

# Save USA data separately
health_ind_USA <- health_ind[health_ind$state=="United States", ]

# Drop USA and DC data
health_ind <- health_ind[!(health_ind$state %in% c("United States",
"District of Columbia")), ]

# Label state abbreviations
health_ind$state_abb <- state_abbreviations
health_ind <- health_ind[ , c(1, ncol(health_ind), 2:(ncol(health_ind)-
1))]

# Drop raw data
rm(health_ind_raw)


## CLEAN DEMOGRAPHIC (1) DATA

# Drop unnecessary row and column
demographics1 <- demographics1[-1,-1]

# Rename variables
colnames(demographics1) <- c("state","income","population",
"ppl_age0to4","ppl_age5to9","ppl_age10to14","ppl_age15to19","ppl_age20to24
", "ppl_age25to29","ppl_age30to34","ppl_age35to39","ppl_age40to44",
"ppl_age45to49","ppl_age50to54","ppl_age55to59","ppl_age60to64",
"ppl_age65to69","ppl_age70to74","ppl_age75to79","ppl_age80to84","ppl_age85
plus")

# Drop observations DC and PR
```

```
demographics1 <- demographics1[!demographics1$state=="District of
Columbia" & !demographics1$state=="Puerto Rico", ]

# Label state abbreviations
demographics1$state_abb <- state_abbreviations
demographics1 <- demographics1[ , c(1, ncol(demographics1),
2:(ncol(demographics1)-1))]

# Change format of variables
  # To character variable
demographics1$state <- as.character(demographics1$state)
  # To numeric variable
for (j in c(3:ncol(demographics1))) {
  demographics1[[j]] <- as.character(demographics1[[j]])
  demographics1[[j]] <- as.numeric(demographics1[[j]])
}

# BEA regions
demographics1$BEA_region <- ""
demographics1$BEA_region[demographics1$state_abb %in% c("ME", "NH", "VT",
"MA", "CT", "RI")] <- "New England"
demographics1$BEA_region[demographics1$state_abb %in% c("NY", "NJ", "PA",
"MD", "DE")] <- "Mideast"
demographics1$BEA_region[demographics1$state_abb %in% c("WI", "IL", "MI",
"IN", "OH")] <- "Great Lakes"
demographics1$BEA_region[demographics1$state_abb %in% c("WV", "KY", "VA",
"TN", "NC", "SC", "AR", "LA", "MS", "AL", "GA", "FL")] <- "Southeast"
demographics1$BEA_region[demographics1$state_abb %in% c("ND", "SD", "NE",
"KS", "MN", "IA", "MO")] <- "Plains"
demographics1$BEA_region[demographics1$state_abb %in% c("MT", "ID", "WY",
"UT", "CO")] <- "Rocky Mountains"
demographics1$BEA_region[demographics1$state_abb %in% c("AZ", "NM", "TX",
"OK")] <- "Southwest"
demographics1$BEA_region[demographics1$state_abb %in% c("WA", "OR", "CA",
"NV", "AK", "HI")] <- "Far West"

# Tag income level (category)
demographics1$income_level <- ""
demographics1$income_level[demographics1$income >= 60000] <- "High"
demographics1$income_level[demographics1$income >= 50000 &
demographics1$income < 60000] <- "Upper Middle"
demographics1$income_level[demographics1$income >= 45000 &
demographics1$income < 50000] <- "Lower Middle"
demographics1$income_level[demographics1$income < 45000] <- "Low"
  # Convert to factor variable
demographics1$income_level <- factor(demographics1$income_level, levels =
c("Low", "Lower Middle", "Upper Middle", "High"))


## CLEAN DEMOGRAPHIC (2) DATA

colnames(demographics2) <- demographics2[2, ]

# Drop first few rows (non-observations)
```

```
demographics2 <- demographics2[-(1:2), ]

# Drop county-level data,  keep state-level data only
demographics2 <- demographics2[demographics2$Area_Name == "United States"
| demographics2$State != lag(demographics2$State), ]

# Keep only columns of population from 2010 to 2015
demographics2 <- demographics2[ , c("State", "Area_Name",
"POP_ESTIMATE_2010", "POP_ESTIMATE_2011", "POP_ESTIMATE_2012",
"POP_ESTIMATE_2013", "POP_ESTIMATE_2014", "POP_ESTIMATE_2015")]

# Change format of variables
  # To character variable
for (j in c(3:8)) {
  demographics2[[j]] <- as.numeric(demographics2[[j]])
}

# Rename variables
colnames(demographics2) <- c("state_abb", "state", "population_2010",
"population_2011", "population_2012", "population_2013",
"population_2014", "population_2015")


## CLEAN ELECTION DATA

# Keep only 2012 and 2008 data, and data for the 50 states
election <- election[ 4:53 , c("X0", "X5", "X6", "X9", "X10")]

# Rename variables
colnames(election) <- c("state", "Dem_pct_2012", "Rep_pct_2012",
"Dem_pct_2008", "Rep_pct_2008")

# Add state abbreviations
  # Match with look-up table
election$state_abb <- state_abb_lookup$state_abb[match(election$state,
state_abb_lookup$state)]

# Change format of variables
  # To numeric variable
for (j in c(4:6)) {
  election[[j]] <- as.numeric(election[[j]])
}

# Calculate winning party for each state
election$party_2012 <- ifelse(election$Dem_pct_2012 >
election$Rep_pct_2012, "Democratic", "Republican")
election$party_2008 <- ifelse(election$Dem_pct_2008 >
election$Rep_pct_2008, "Democratic", "Republican")

# Reorder variables
election <- election[ , c("state", "state_abb", "Dem_pct_2012",
"Rep_pct_2012", "party_2012", "Dem_pct_2008", "Rep_pct_2008",
"party_2008")]
## SOME DATA PREPARATION
```

```
# Look-up information into insurance and insurance_long, on:
  # Party (2012)
insurance_long$party_2012 <-
election$party_2012[match(insurance_long$state, election$state)]
insurance$party_2012 <- election$party_2012[match(insurance$state,
election$state)]
  # BEA region
insurance_long$BEA_region <-
demographics1$BEA_region[match(insurance_long$state, demographics1$state)]
insurance$BEA_region <- demographics1$BEA_region[match(insurance$state,
demographics1$state)]
  # Income level
insurance_long$income_level <-
demographics1$income_level[match(insurance_long$state,
demographics1$state)]
insurance$income_level <-
demographics1$income_level[match(insurance$state, demographics1$state)]

# Also find nation-wide trend for insurance
insurance_long_US <- data.frame("year" = c(2008, 2009, 2010, 2011, 2012,
2013, 2014, 2015), "uninsured_pct" = c(16.8, 17.5, 18.2, 17.2, 16.9, 16.6,
13.3, 10.5))
## PREPARATION FOR FURTHER PLOTS (INCLUDING DATA TABLE)

# Want to combine information on insurance coverage with key information
on health outcomes/access and affordability

key_indicators <- state_abb_lookup

## Merge in information to the data table on the following

# Demographic Information
  # Income level
key_indicators$income_level <-
demographics1$income_level[match(key_indicators$state_abb,
demographics1$state_abb)]

# Insurance Coverage
  # Percentage insured (2015)
key_indicators$insured_pct_2015 <-
insurance$insured_pct_2015[match(key_indicators$state_abb,
insurance$state_abb)]
  # Percentage insured (2014)
key_indicators$insured_pct_2014 <-
insurance$insured_pct_2014[match(key_indicators$state_abb,
insurance$state_abb)]
  # Indicator: Whether state has expanded medicare
key_indicators$state_has_expanded <-
aca_general$state_has_expanded[match(key_indicators$state_abb,
aca_general$state_abb)]

# Health outcomes and access
  # Mortality amenable to health care, deaths per 10000 population
```

```
key_indicators$deaths_amenable_2014 <- health_ind$h.deaths_amenable.2014
[match(key_indicators$state_abb, health_ind$state_abb)]
  # Years of Potential Life Lost before 75
key_indicators$years_life_lost <-
health_ind$h.yrs_lost_potential_life_before75.2014
[match(key_indicators$state_abb, health_ind$state_abb)]
  # Adults with a usual source of care (%)
key_indicators$usual_care_2015 <- health_ind$q.with_usual_care_adult.2015
[match(key_indicators$state_abb, health_ind$state_abb)]
  # Adults with age/gender-appropriate cancer screenings (%)
key_indicators$with_cancer_screening <-
health_ind$q.with_cancer_screening_adult.2014
[match(key_indicators$state_abb, health_ind$state_abb)]
  # Adults with age-appropriate vaccines (%)
key_indicators$with_vaccines <- health_ind$q.with_vaccines_adult.2015
[match(key_indicators$state_abb, health_ind$state_abb)]

# Affordability and cost-efficiency
  # Individuals under age 65 with high OOP medical costs relative to
annual household income
key_indicators$high_OOP_relative <-
health_ind$a.high_OOP_relative_under65.2015
[match(key_indicators$state_abb, health_ind$state_abb)]
  # Average annual growth in family premiums for employer coverage
(between 2010 and 2015)
key_indicators$premium_ann_growth_10_15 <-
aca_general$premium_emp_avg_growth_pct_10to15[match(key_indicators$state_a
bb, aca_general$state_abb)]
  # Marketplace consumers who could select a plan for less than $100
key_indicators$IM_plan_under_100 <-
aca_general$cov_mkt_under100D_pct[match(key_indicators$state_abb,
aca_general$state_abb)]
  # Net increase in federal spending (millions)
key_indicators$incr_fed_spending_mil <-
aca_general$fed_spending_net_incr_inMil[match(key_indicators$state_abb,
aca_general$state_abb)]
```

In this section, we are interested in investigating the how states' insurance coverage relates to two broad classes of outcomes - **health outcomes** and **cost containment outcomes**. We anticipate that an increase in health insurance coverage in a state would be correlated with increased health (though there is unlikely to be a significant causal relationship due to the short time span since the ACA went into effect). Moreover, an increase in health insurance coverage would likely shift some of the financial burden from individual out of pocket (OOP) costs to the premiums paid (pooled risk) and to the state.

We are interested in finding out which states perform well in which outcomes, as well as eventually develop an indicator of how states fare *overall*.

Health Outcomes and Access

Figure 8

We explore how health insurance rates correlates with health outcomes. In particular, we can look at the measure of mortality amenable to healthcare intervention - the number of deaths that could have been prevented by healthcare in the year 2014. We plot avoidable mortality (2014) against insurance rates (2014) and observe a negative relationship as expected. States that have higher rates of insurance also have better health in this respect. However, we also control for income levels as low income levels would be a major predictor of poor health, as confirmed in our plot. Yet, within each income level, the negative relationship remains, and it is especially pronounced among the upper-middle income countries.

```
## PLOT 8
plot8 <- (ggplot(key_indicators, aes(insured_pct_2014,
deaths_amenable_2014))
          + geom_smooth(aes(color = income_level, group = income_level),
method = "lm", lwd = 1, se = FALSE)
          + geom_smooth(color = "black", method = "lm", linetype = 2, lwd
= 1, se = FALSE)
          + geom_point(aes(color = income_level, shape =
state_has_expanded), size = 2)
          + geom_text(aes(label = state_abb), nudge_x = 0.5, nudge_y =
1.5, check_overlap = TRUE, size = 2.5, color = "gray30")
          + labs(x = "% Insured", y = "Deaths Preventable by Healthcare
Intervention \n(per 100,000 People)", color = "Income \nLevel", shape =
"State has \nExpanded \nMedicaid")
          + scale_color_manual(values = brewer.pal(n = 4, name = "Dark2"))
          + ggtitle("Relationship Between Avoidable Mortality and \nHealth
Insurance Coverage Among States")
          + theme_minimal()
          + theme(plot.title = element_text(face="bold", hjust = 0.5, size
= 12)))
plot8
## DATA PROCESSING FOR MAPS

# Obtain shape files of US states
map_states = map("state", fill = TRUE, plot = FALSE)

# Standardize name format with rest of assignment
  # Function to conver to proper case
properCase <- function(x) {
  s <- strsplit(x, " ")[[1]]
  paste(toupper(substring(s, 1,1)), substring(s, 2),
      sep="", collapse=" ")
}
  # Convert to proper case
map_states$state <- sapply(map_states$names, properCase)
  # Rename the main component of states with multiple parts in the map
map_states$state[map_states$state == "Massachusetts:main"] <-
"Massachusetts"
map_states$state[map_states$state == "Michigan:south"] <- "Michigan"
map_states$state[map_states$state == "New York:main"] <- "New York"
map_states$state[map_states$state == "North Carolina:main"] <- "North
Carolina"
map_states$state[map_states$state == "Virginia:main"] <- "Virginia"
map_states$state[map_states$state == "Washington:main"] <- "Washington"
  # Match with state abbreviations
```

```
map_states$state_abb <- state_abb_lookup$state_abb[match(map_states$state,
state_abb_lookup$state)]
  # Then strip off all the extra location information for states with
multiple parts (leaving just the state name)
for (i in 1:length(map_states$state)) {
  map_states$state[i] <- unlist(strsplit(map_states$state[i], ":"))[1]
}

# Get coordinates of state centers
states_centers <- state.center
state.center <- cbind(states_centers, state_abb_lookup)

# Health outcomes: Import key indicators data, matched by state
    # Remember to later match information based on state, not state
abbreviation (due to how we labelled above)
map_states$deaths_amenable_2014 <-
key_indicators$deaths_amenable_2014[match(map_states$state,
key_indicators$state)]
map_states$usual_care_2015 <-
key_indicators$usual_care_2015[match(map_states$state,
key_indicators$state)]
map_states$with_cancer_screening <-
key_indicators$with_cancer_screening[match(map_states$state,
key_indicators$state)]
map_states$with_vaccines <-
key_indicators$with_vaccines[match(map_states$state,
key_indicators$state)]

# Cost outcomes: Import key indicators data, matched by state
    # Remember to later match information based on state, not state
abbreviation (due to how we labelled above)
map_states$high_OOP_relative <-
key_indicators$high_OOP_relative[match(map_states$state,
key_indicators$state)]
map_states$premium_ann_growth_10_15 <-
key_indicators$premium_ann_growth_10_15[match(map_states$state,
key_indicators$state)]
map_states$IM_plan_under_100 <-
key_indicators$IM_plan_under_100[match(map_states$state,
key_indicators$state)]
map_states$incr_fed_spending_mil <-
key_indicators$incr_fed_spending_mil[match(map_states$state,
key_indicators$state)]
```

Figure 9

The following maps how the states performed in the following health outcomes:

1. Preventable mortality - deaths amenable to healthcare intervention, per 100,000 population (2014)

2. Adults with access to a usual source of care (2015)

3. Rate of age- and gender-appropriate cancer screenings in adults

4. Rate of age-appropriate vaccinations in adults (2014)

A green color always incidates a comparatively more favorable outcome, while a red color indicates a less favorable outcome (2015)

We note that the Northeast (New England) states perform consistently well across all health outcomes, while the Southeast states, as well as Texes, peform consistently poorly on all except for vaccination rates. The Western and Rocky Mountains states have a low avoidable mortality rate, despite having low rates of access to usual care and of cancer screening and vaccinations. In the measure of adult vaccination rates, the trend of the South faring poorer on health is reversed, as they do comparatively well in vaccinations.

```
## PLOT 9: LEAFLET MAP - HEALTH OUTCOMES
(leaflet(map_states) %>%
    setView(lat=39.8282, lng=-96 , zoom=4) %>%
    addPolygons(color = "#333333", weight = 1, smoothFactor = 0.5,
fillOpacity = 0) %>%
    addPolygons(group = "Preventable Deaths (2014)", fillColor =
~colorQuantile("RdYlGn", -deaths_amenable_2014)(-deaths_amenable_2014),
smoothFactor = 0.5, stroke = FALSE, fillOpacity = 0.6, popup =
paste("<b>State: </b>", map_states$state, "<br/>", "<b>Preventable
Mortality</b>", "<br/>", "<b>per 100,000: </b>",
map_states$deaths_amenable_2014)) %>%
    addPolygons(group = "Access to Usual Care (2015)", fillColor =
~colorQuantile("RdYlGn", usual_care_2015)(usual_care_2015), smoothFactor =
0.5, stroke = FALSE, fillOpacity = 0.6, popup = paste("<b>State: </b>",
map_states$state, "<br/>", "<b>Access to Usual</b>", "<br/>", "<b>Source
of Care: </b>", map_states$usual_care_2015, "%")) %>%
    addPolygons(group = "Cancer Screenings Rate (2014)", fillColor =
~colorQuantile("RdYlGn", with_cancer_screening)(with_cancer_screening),
smoothFactor = 0.5, stroke = FALSE, fillOpacity = 0.6, popup =
paste("<b>State: </b>", map_states$state, "<br/>", "<b>Cancer
Screenings</b>", "<br/>", "<b>Rate: </b>",
map_states$with_cancer_screening, "%")) %>%
    addPolygons(group = "Adult Vaccination Rate (2015)", fillColor =
~colorQuantile("RdYlGn", with_vaccines)(with_vaccines), smoothFactor =
0.5, stroke = FALSE, fillOpacity = 0.6, popup = paste("<b>State: </b>",
map_states$state, "<br/>", "<b>Vaccination</b>", "<br/>", "<b>Rate: </b>",
map_states$with_vaccines, "%")) %>%
    addLabelOnlyMarkers(data = filter(state.center, state_abb!="AK" &
state_abb!="HI"), lng = ~x, lat = ~y, label = ~state_abb, labelOptions =
labelOptions(noHide = T, direction = 'top', textOnly = T)) %>%
    addLayersControl(
      baseGroups = c("Preventable Deaths (2014)", "Access to Usual Care
(2015)", "Cancer Screenings Rate (2014)", "Adult Vaccination Rate
(2015)"),
      options = layersControlOptions(collapsed = FALSE)))
```

Affordability and Cost Outcomes

Figure 10

Figure 10 demonstrates a clear negative relationship between the insured rate and the percentage of people with high out-of-pocket (OOP) medical costs relative to their annaul household income. This suggests, as we expect, that as more people get insured, they are less likely to be paying excessive out-of-pocket costs for medical expenses. This relationship holds even after controlling for income level.

There was no conclusive relationship when the other two cost outcomes - annual growth in family premiums and increase in federal spending - were plotted against the insured rate.

```
## PLOT 10
plot10 <- (ggplot(key_indicators, aes(insured_pct_2015, high_OOP_relative, label = state_abb))
            + geom_smooth(aes(color = income_level), method = "lm", lwd = 1, se = FALSE)
            + geom_smooth(color = "black", method = "lm", linetype = 2, lwd = 1, se = FALSE)
            + geom_point(aes(color = income_level, shape = state_has_expanded), size = 2)
           + geom_text(aes(label = state_abb), nudge_y = -0.2, check_overlap = TRUE, size = 2.5, color = "gray30")

           + scale_color_manual(values = brewer.pal(n = 4, name = "Dark2"))
           + labs(x = "% Insured", y = "People with High OOP Costs Relative to Household Income", color = "Income \nLevel", shape = "State has \nExpanded \nMedicaid")
           + ggtitle("Relationship Between OOP Costs and Health \nInsurance Coverage Among States")
           + theme_minimal()
           + theme(plot.title = element_text(face="bold", hjust = 0.5, size = 12)))
plot10
```

Figure 11

Similarly, we map how the states performed in the following financial and cost outcomes:

1. Percentage of people with high OOP costs relative to annual household income (2015)

2. Average annual premium growth rate from 2010 to 2015

3. Affordability of the marketplace plans, measured by the marketplace consumers who can obtain a plan less than $100 (2017). Note that states that do not operate a insurance marketplace are shaded in gray.

4. Net increase in federal spending (in millions of dollars) over the period (up till 2016)

As before, a green color incidates a comparatively more favorable outcome, while a red color indicates a less favorable outcome.

We notice that states that do better in containing OOP costs tend to do worse in containing premium and marketplace plan rates as well as federal spending. These include some states like New York, New Jersey and Ohio in the Northeast and Midwest (Great Lakes). A notable exception is Massachussets, which performs well across the financial measures, possibly because of the headstart it got with "Romney-care", which actually served as a model for Obamacare.

```
## PLOT 11: LEAFLET MAP - AFFORDABILITY / COST OUTCOMES
marketplace_pct <- map_states$IM_plan_under_100
marketplace_pct <- marketplace_pct * 100
marketplace_pct[is.na(marketplace_pct)] <- "No State Marketplace"
marketplace_pct_symbol <- ifelse(marketplace_pct=="No State Marketplace",
"", "%")

(leaflet(map_states) %>%
    setView(lat=39.8282, lng=-96 , zoom=4) %>%
    addPolygons(color = "#333333", weight = 1, smoothFactor = 0.5,
fillOpacity = 0) %>%
    addPolygons(group = "Contained OOP Costs (2015)", fillColor =
~colorQuantile("RdYlGn", -high_OOP_relative)(-high_OOP_relative),
smoothFactor = 0.5, stroke = FALSE, fillOpacity = 0.6, popup =
paste("<b>State: </b>", map_states$state, "<br/>", "<b>People with High
OOP</b>", "<br/>", "<b>Relative to Income: </b>",
map_states$high_OOP_relative, "%")) %>%
    addPolygons(group = "Premium Growth Rate (2010-2015)", fillColor =
~colorQuantile("RdYlGn", -premium_ann_growth_10_15)(-
premium_ann_growth_10_15), smoothFactor = 0.5, stroke = FALSE, fillOpacity
= 0.6, popup = paste("<b>State: </b>", map_states$state, "<br/>",
"<b>Annual Premium Growth</b>", "<br/>", "<b>Rate (2010-2015): </b>",
map_states$premium_ann_growth_10_15, "%")) %>%
    addPolygons(group = "Affordable Marketplace Plan (2017)", fillColor =
~colorQuantile("RdYlGn", IM_plan_under_100)(IM_plan_under_100),
smoothFactor = 0.5, stroke = FALSE, fillOpacity = 0.6, popup =
paste("<b>State: </b>", map_states$state, "<br/>", "<b>Marketplace
Consumers</b>", "<br/>", "<b>Who Can Select Plan</b>", "<br/>","<b>Under
$100: </b>", marketplace_pct, marketplace_pct_symbol)) %>%
    addPolygons(group = "Increase in Federal Spending (2016)", fillColor =
~colorQuantile("RdYlGn", -incr_fed_spending_mil)(-incr_fed_spending_mil),
smoothFactor = 0.5, stroke = FALSE, fillOpacity = 0.6, popup =
paste("<b>State: </b>", map_states$state, "<br/>", "<b>Net Increase in
Federal</b>", "<br/>", "<b>Spending</b>: $",
map_states$incr_fed_spending_mil, "million")) %>%
    addLabelOnlyMarkers(data = filter(state.center, state_abb!="AK" &
state_abb!="HI"), lng = ~x, lat = ~y, label = ~state_abb, labelOptions =
labelOptions(noHide = T, direction = 'top', textOnly = T)) %>%
    addLayersControl(
      baseGroups = c("Contained OOP Costs (2015)", "Premium Growth Rate
(2010-2015)", "Affordable Marketplace Plan (2017)", "Increase in Federal
Spending (2016)"),
      options = layersControlOptions(collapsed = FALSE)))
```

Summary of States' Performance

```
# Create data table reporting values
data_table_values <- key_indicators[ , c("state", "insured_pct_2015",
"deaths_amenable_2014", "usual_care_2015", "high_OOP_relative",
"premium_ann_growth_10_15", "incr_fed_spending_mil")]

# Create data table reporting rank
```

```
data_table_rank <- data_table_values[ , c("state", "insured_pct_2015",
"deaths_amenable_2014", "usual_care_2015", "high_OOP_relative",
"premium_ann_growth_10_15")]

# Convert each function to rank
  # Rank 1 is always more favorable (regardless of definition of variable)
data_table_rank$r_insured_pct_2015 <- rank(-
data_table_values$insured_pct_2015, na.last = "keep", ties.method = "min")
data_table_rank$r_deaths_amenable_2014 <-
rank(data_table_values$deaths_amenable_2014, na.last = "keep", ties.method
= "min")
data_table_rank$r_usual_care_2015 <- rank(-
data_table_values$usual_care_2015, na.last = "keep", ties.method = "min")
data_table_rank$r_high_OOP_relative <-
rank(data_table_values$high_OOP_relative, na.last = "keep", ties.method =
"min")
data_table_rank$r_premium_ann_growth_10_15 <-
rank(data_table_values$premium_ann_growth_10_15, na.last = "keep",
ties.method = "min")
data_table_rank$r_incr_fed_spending_mil <-
rank(data_table_values$incr_fed_spending_mil, na.last = "keep",
ties.method = "min")
  # Remove value variables
data_table_rank$insured_pct_2015 <- NULL
data_table_rank$deaths_amenable_2014 <- NULL
data_table_rank$usual_care_2015 <- NULL
data_table_rank$high_OOP_relative <- NULL
data_table_rank$premium_ann_growth_10_15 <- NULL
data_table_rank$incr_fed_spending_mil <- NULL
```

Finally, we want to compare how the states do across both health and cost outcomes, and eventually get to some form of ranking of the states.

The following data table presents the rankings of the states on the following insurance coverage, health and affordability outcomes, providing a summary of the states' performance:

1. Percentage of population insured
2. Deaths preventable by healthcare intervention (per 100,000 people)
3. Proportion of adults with access to a usual source of care
4. Proportion of people with high out-of-pocket costs (relative to household income)
5. Annual rate of health insurance premium growth from 2010 to 2015
6. Net increase in federal spending on healthcare up till 2016

A rank of 1 always indicates the more favorable outcome (i.e. higher insurance and treatment access rate, lower preventable mortality, OOP costs, premium growth, and federal spending increase).

Figure 12

```
## PLOT 12: DATA TABLE RANKING STATES ON VARIOUS MEASURES
```

```
datatable(data_table_rank, rownames = FALSE, colnames = c("State",
"Percentage Insured", "Fewer Preventable Deaths", "Usual Care Access",
"OOP Contained", "Lower Annual Premium Growth", "Lower Increase in Federal
Spending"), caption = "Rank of the 50 States in Insurance Coverage, Health
and Cost Outcomes")
# Compute overall ranking of states by each of three measures
data_table_rank_cat <- data_table_rank
  # Insurance coverage
data_table_rank_cat$insurance_rank <- data_table_rank$r_insured_pct_2015
  # Health
data_table_rank_cat$health_rank_sum <-
data_table_rank$r_deaths_amenable_2014 + data_table_rank$r_usual_care_2015
data_table_rank_cat$health_rank <-
rank(data_table_rank_cat$health_rank_sum, na.last = "keep", ties.method =
"min")
  # Finance
data_table_rank_cat$cost_rank_sum <- data_table_rank$r_high_OOP_relative +
data_table_rank$r_premium_ann_growth_10_15 +
data_table_rank$r_incr_fed_spending_mil
data_table_rank_cat$cost_rank <- rank(data_table_rank_cat$cost_rank_sum,
na.last = "keep", ties.method = "min")

data_table_rank_cat <- data_table_rank_cat[ , c("state", "insurance_rank",
"health_rank", "cost_rank")]
data_table_rank_cat$sum_of_ranks <- data_table_rank_cat$insurance_rank +
data_table_rank_cat$health_rank + data_table_rank_cat$cost_rank
data_table_rank_cat$overall_rank <- rank(data_table_rank_cat$sum_of_ranks,
na.last = "keep", ties.method = "min")

data_table_rank_cat_long <- data_table_rank_cat
data_table_rank_cat_long <- reshape(data_table_rank_cat,
                      varying = c("insurance_rank", "health_rank",
"cost_rank"),
                      v.names = "Rank",
                      timevar = "Category",
                      times = c("Insurance", "Health", "Cost"),
                      new.row.names = 1:1000,
                      direction = "long")
```

Figure 13

We present a way of aggregating the rankings of the states in the above six health measures into an overall rank. We rank each state on each of three dimensions, insurance coverage, health outcomes, and cost control outcomes. For the health and cost control outcomes, since they have multiple subcomponents (2 and 3 respectively, as in the data table), we take an average over these subcomponent ranks. We then add up the ranks in the three dimensions to find an overall rank to order the states.

States at the top of this chart (with lower rank numbers) perform the best on the whole in their healthcare system. Mouse over the interactive plot to view the overall state rank. Note that you can *toggle the three criteria on and off* to include or exclude each criteria from the graph.

We see that many of the states with high-performing healthcare systems are New England states, including Vermont, Rhode Island, Massachussets and Connecticut, while Minnesota and Iowa also perform well. At the other end, the poorest performing states across all measures are Georgia, Oklahoma, Louisiana and Texas, mostly states in the Southeast and Southwest.

```
plot13 <- (ggplot(data_table_rank_cat_long, aes(x = reorder(state, -
overall_rank), y = Rank, fill = Category, label = overall_rank))
           + geom_col(position = "stack")
           + coord_flip()
           + labs(x = "", y = "", fill = "Rank \nCriteria")
           + scale_fill_manual(values = brewer.pal(n = 3, name =
"Dark2")[c(2,1,3)])
           + ggtitle("Overall Ranking of States Healthcare Systems")
  + theme_minimal()
  + theme(plot.title = element_text(face="bold", hjust = 0, size = 12),
          axis.text.y = element_text(size = 6.5),
          axis.ticks.x = element_blank(),
          axis.text.x = element_blank()))
ggplotly(plot13, tooltip = "overall_rank")
```

Perceptions of ACA (Text Analysis)

We are also interested in how people think about Affordable care act(ACA). We want to have a look of the tweets and newspaper related to ACA. To realize this, we use API to download articles realted to ACA on New york times from 2011-2017 and transform them into corpus to do the text analysis.

Research Questions:

1. The overll trend of public attention, which can be shown as the number of articles related to ACA in different years.
2. What people discuss about when they discuss ACA, which can be shown as the word frequencies.

We could see that people discussed ACA a lot when obama first signed it, and the election in 2017 made it a hot topic again. And not suprisingly, peopel always talk about Trump when they talk about ACA

```
library(tm)
library(tm.plugin.lexisnexis)
library(readxl)
library(gtools)  # for smartbind
library(dplyr)   # for data_frame
library(lubridate)   # for date formatting
library(stringr)
library(tools)  # Title case
library(quanteda)
library(ggplot2)
library(quanteda)
library(stringr)
library(tm)
library(qdap)
library(SnowballC)
library(dplyr)
library(tidytext)
```

```
library(wordcloud)
library(ggthemes)

# Combine CSV and HTML Files
data <- read_excel("LexisNexis/NYTimes_Metadata.xlsx")
colnames(data) <- tolower(colnames(data))

# Correct data
data$date <- substr(parse_date_time(data$date, c("mdy")),1,10)
data$author <- toTitleCase(tolower(data$byline))
data$byline <- NULL

## Get Text files
source1 <- LexisNexisSource("LexisNexis/The_New_York1.html")
source2 <- LexisNexisSource("LexisNexis/The_New_York2.html")

corpus1 <- Corpus(source1, readerControl = list(language = NA))
corpus2 <- Corpus(source2, readerControl = list(language = NA))

corpus <- c(corpus1, corpus2)

# Convert to quanteda corpus
corpus <- quanteda::corpus(corpus)

## Add Metadata
# Check: match(data$headline, corpus$documents$heading)
corpus$documents$datetimestamp <- substring(corpus$documents$datetimestamp, 1,4)
corpus$documents$date <- corpus$documents$datetimestamp
corpus$documents$description <- corpus$documents$id

# options(width = 200)
# kwic(corpus, "Trump")
#
save(corpus, file="nytimes.rda")
```

Figure 14

We could see that people discussed ACA a lot when Obama first signed it, and the election in 2017 made it a hot topic again. And not suprisingly, people always talk about Trump when they talk about ACA.

```
dat <- data.frame(date =corpus$documents$datetimestamp, levels=c(1:1000))
ggplot(data=dat, aes(x=date)) +geom_bar(stat="count") +
  ggtitle("Number of Articles Mentioning ACA by Year")+theme_tufte()+ylab("Number of articles") +
  xlab("year") + theme(plot.title = element_text(face="bold", hjust = 0.5, size = 12)) +
  theme_minimal()
load("nytimes.rda")
dfmtotal <- dfm(corpus, remove = stopwords("english"), stem = TRUE, removePunct = TRUE, removeNumbers = TRUE, tolower = TRUE, verbose = TRUE)
dfmtotal[, 1:5]
head(stopwords("english"), 20)
freq<-topfeatures(dfmtotal, 20)
wf <- data.frame(word=names(freq), freq=freq)
```

Figure 15

The graph shows that the frequently use words are health, insur and care. And other words like law, people and republican are also frequently used. It is not surprising that the articles most features on the health care issues, and yet the topics of people, policy and politics are also highly concerned in New York Times.

```
library(plotly)
pp <- plot_ly(subset(wf, freq>50), x = ~freq, y = ~reorder(word, freq), type =
"bar", orientation = 'h') %>%
  layout(title = "Word Frequency from Articles on ACA, 2011-2017",
         xaxis = list(title = "Top 30 Used Stem"),
         yaxis = list(title = "Frequency of Stem"),
         margin = list(l = 120, r = 10, t = 80, b = 80))
pp
load('nytimes.rda')
NYT_source <- VectorSource(corpus$documents[,1])
NYT <- VCorpus(NYT_source)
documents <- corpus$documents
removeNumPunct <- function(x){gsub("[^[:alpha:][:space:]]*", "", x)}
clean_corpus <- function(corpus){
corpus <- tm_map(corpus, removePunctuation)
corpus <- tm_map(corpus, content_transformer(tolower))
corpus <- tm_map(corpus, content_transformer(replace_symbol))
corpus <- tm_map(corpus, removeWords, c(stopwords("english"), "will", "can"))
# We could add more stop words as above
corpus <- tm_map(corpus, stripWhitespace)
corpus <- tm_map(corpus, removeNumbers)
corpus <- tm_map(corpus, content_transformer(removeNumPunct))
return(corpus)
}
NYT_clean <- clean_corpus(NYT)
NYT_stem <- tm_map(NYT_clean, stemDocument)
meta(NYT_stem, type = "local", tag = "author") <- documents$author
NYT_dtm <- DocumentTermMatrix(NYT_stem)
NYT_tdm <- TermDocumentMatrix(NYT_stem)

NYT_tdm2 <- tidy(NYT_tdm)
NYT_dtm2 <- tidy(NYT_dtm)
NYT_tidy <- tidy(NYT_stem)
NYT_tdm2 <- merge(NYT_tdm2, NYT_tidy, by.x = "document", by.y = "id", all.x =
TRUE)
NYT_dtm2 <- merge(NYT_dtm2, NYT_tidy, by.x = "document", by.y = "id", all.x =
TRUE)
```

Figure 16

Wordcloud by Year

In the beginning, the articles mainly focused on the law issues as we can see from the frequently used words such as federal, act, mandate, judge, reform, etc. As time passed, the topics mainly focused on financial issues, the words that articles frequently mentioned are workers, business, tax, financial, subsidies, medicaid, etc. In the latest two years, due to the presidential election, the topics are changed to political concerns. For example, the words like people, trump, republicans, house, and repeal are frequently used in the articles.

```
atxt <- documents$texts[documents$datetimestamp == "2011"]
btxt <- documents$texts[documents$datetimestamp == "2012"]
ctxt <- documents$texts[documents$datetimestamp == "2013"]
dtxt <- documents$texts[documents$datetimestamp == "2014"]
etxt <- documents$texts[documents$datetimestamp == "2015"]
ftxt <- documents$texts[documents$datetimestamp == "2016"]
```

```
gtxt <- documents$texts[documents$datetimestamp == "2017"]

clean.text <- function(x)
{
# tolower
x = tolower(x)
# remove rt
x = gsub("rt", "", x)
# remove at
x = gsub("@\\w+", "", x)
# remove punctuation
x = gsub("[[:punct:]]", "", x)
# remove numbers
x = gsub("[[:digit:]]", "", x)
# remove links http
x = gsub("http\\w+", "", x)
# remove tabs
x = gsub("[ |\t]{2,}", "", x)
# remove blank spaces at the beginning
x = gsub("^ ", "", x)
# remove blank spaces at the end
x = gsub(" $", "", x)
return(x)
}

aclean <- clean.text(atxt)
bclean <- clean.text(btxt)
cclean <- clean.text(ctxt)
dclean <- clean.text(dtxt)
eclean <- clean.text(etxt)
fclean <- clean.text(ftxt)
gclean <- clean.text(gtxt)

a <- paste(aclean, collapse=" ")
b <- paste(bclean, collapse=" ")
c <- paste(cclean, collapse=" ")
d <- paste(dclean, collapse=" ")
e <- paste(eclean, collapse=" ")
f <- paste(fclean, collapse=" ")
g <- paste(gclean, collapse=" ")

# put everything in a single vector
all <- c(a,b,c,d,e,f,g)
all <- removeWords(all, c("will", "can", "the",
"that","are","mrs","not","said","cou"))
# create corpus
corpus2 <- Corpus(VectorSource(all))
# create term-document matrix
cloudtdm <- TermDocumentMatrix(corpus2)
# convert as matrix
cloudtdm <- as.matrix(cloudtdm)
# add column names
colnames(cloudtdm) <- c("2011","2012","2013","2014","2015","2016","2017")
comparison.cloud(cloudtdm, random.order=FALSE,colors = brewer.pal(8,
"Dark2"),title.size=1.5, max.words = 200)
```

#Twitter API Sentiment/Text Analysis

Get some tweets from Twitter to analyze and visualize Set up Twitter API: Selecting data including Obamacare, ACA, Affordable Care Act, and #ACA (n=1000)

```
library(httr)
#library(oauth)
library(ROAuth)
library(twitteR)
library(RCurl)
library(RJSONIO)
library(stringr)
#secretkey
#myapp <- oauth_app("twitter",
                  # key = "liLn6XJFenGjtvWFwi5LnDS1M",
                  # secret = "dsCBm9Kyaeu9GMKlM9xwKl7eKmDn6qsjP31LQtwMGkF6OQdLh6")
#Get OAuth credentials
#twitter_token <- oauth1.0_token(oauth_endpoints("twitter"), myapp)

# Declare Twitter API Credentials
api_key <- "liLn6XJFenGjtvWFwi5LnDS1M" # From dev.twitter.com
api_secret <- "dsCBm9Kyaeu9GMKlM9xwKl7eKmDn6qsjP31LQtwMGkF6OQdLh6" # From
dev.twitter.com
token <- "772176811455381505-PYuNAEqhHFc02r83WS9Y5dnsZciIY5v" # From
dev.twitter.com
token_secret <- "mgRPwKeHZEw9Y486h2GMtCBxDztPfQLX1Msd5vog1hiwv" # From
dev.twitter.com

# Create Twitter Connection
library("base64enc")
setup_twitter_oauth(api_key, api_secret, token, token_secret)

# Run Twitter Search. Format is searchTwitter("Search Terms", n=100, lang="en",
geocode="lat,lng", also accepts since and until).

tweets <- searchTwitter("Obamacare OR ACA OR 'Affordable Care Act' OR #ACA",
n=1000, lang="en", since="2014-08-20")

# Transform tweets list into a data frame
tweets.df <- twListToDF(tweets)
head(tweets.df,3)
counts=table(tweets.df$screenName)
barplot(counts)
# Let's do something hacky:
# Limit the data set to show only folk who tweeted twice or more in the sample
cc=subset(counts,counts>1)
barplot(cc,las=2,cex.names =0.3)
tweets.df$text=sapply(tweets.df$text,function(row) iconv(row,to='UTF-8'))

#A helper function to remove @ symbols from user names...
trim <- function (x) sub('@','',x)

#A couple of tweet parsing functions that add columns to the dataframe
library(stringr)
#Pull out who a message is to
tweets.df$to=sapply(tweets.df$text,function(tweet)
str_extract(tweet,"^(@[[:alnum:]_]*)"))
tweets.df$to=sapply(tweets.df$to,function(name) trim(name))
```

```
#And here's a way of grabbing who's been RT'd
tweets.df$rt=sapply(tweets.df$text,function(tweet) trim(str_match(tweet,"^RT
(@[[:alnum:]_]*)")[2]))

#now we can plot a chart showing how often a particular person was RT'd in our
sample.
library(ggplot2)
ggplot()+geom_bar(aes(x=na.omit(tweets.df$rt)))
library(tidyr)
library(dplyr)
library(purrr)
tweets <- tweets.df %>%
  select(id, statusSource, text, created) %>%
  extract(statusSource, "source", "Twitter for (.*?)<") %>%
  filter(source %in% c("iPhone", "Android"))
table(tweets$source)

library(lubridate)
library(scales)
tweets %>%
  count(source, minute = minute(with_tz(created, "EST"))) %>%
  mutate(percent = n / sum(n)) %>%
  ggplot(aes(minute, percent, color = source)) +
  geom_line() +
  scale_y_continuous(labels = percent_format()) +
  labs(x = "Minute of day (EST)",
       y = "% of tweets",
       color = "")
tweet_picture_counts <- tweets %>%
  filter(!str_detect(text, '^"')) %>%
  count(source,
        picture = ifelse(str_detect(text, "t.co"),
                         "Picture/link", "No picture/link"))

ggplot(tweet_picture_counts, aes(source, n, fill = picture)) +
  geom_bar(stat = "identity", position = "dodge") +
  labs(x = "", y = "Number of tweets", fill = "")
```

figure 17_ Comparison of words Now that we're sure there's a difference, what can we say about the difference in the content? We'll use the tidytext package. We start by dividing into individual words using the unnest_tokens function, and removing some common stopwords. As we can see the most frequently used word when discussing ACA on Twitter is rt (Republican Party), care, act, afforadable, health, obama, and so on. From the table, we found that most people discussed about political issues (i.e., the topics of republican and democratic parties), affordable care act itself, and finanical concerns (i.e., bill, pay, etc).

```
library(tidytext)

reg <- "([^A-Za-z\\d#@']|'(?![A-Za-z\\d#@]))"
tweet_words <- tweets %>%
  filter(!str_detect(text, '^"')) %>%
  mutate(text = str_replace_all(text, "https://t.co/[A-Za-z\\d]+|&amp;", "")) %>%
  unnest_tokens(word, text, token = "regex", pattern = reg) %>%
  filter(!word %in% stop_words$word,
         str_detect(word, "[a-z]"))

library(qdap)
# Find the 20 most frequent terms: term_count
term_count1 <- freq_terms(tweet_words$word,20)
# Plot term_count
```

```
plot(term_count1, main = "Frequently used words on Twitter regarding ACA")
android_iphone_ratios <- tweet_words %>%
  count(word, source) %>%
  filter(sum(n) >= 5) %>%
  spread(source, n, fill = 0) %>%
  ungroup() %>%
  mutate_each(funs((. + 1) / sum(. + 1)), -word) %>%
  mutate(logratio = log2(Android / iPhone)) %>%
  arrange(desc(logratio))
library(ggthemes)
ggplot(data=android_iphone_ratios, aes(x = word, y = logratio)) +
  geom_bar(stat = "identity") +
  #geom_text(aes(label=pres, , y=0.005), color="white") +
  xlab(NULL) +  coord_flip() + theme_tufte()

ggplot(data=android_iphone_ratios, aes(x = word, y = Android)) +
  geom_bar(stat = "identity") +
  #geom_text(label= Android, color="red") +
  xlab(NULL) +  coord_flip() + theme_tufte()

ggplot(data=android_iphone_ratios, aes(x = word, y=iPhone)) +
  geom_bar(stat = "identity") +
  #geom_text(label= Android, color="red") +
  xlab(NULL) +  coord_flip() + theme_tufte()
```

Sentiment analysis: Since we've observed a difference in sentiment between the Android and iPhone tweets, let's try quantifying it. We'll work with the NRC Word-Emotion Association lexicon, available from the tidytext package, which associates words with 10 sentiments: positive, negative, anger, anticipation, disgust, fear, joy, sadness, surprise, and trust.

```
nrc <- sentiments %>%
  filter(lexicon == "nrc") %>%
  dplyr::select(word, sentiment)
```

To measure the sentiment of the Android and iPhone tweets, we can count the number of words in each category. (For example, we see that 41 of the 2331 words in the Android tweets were associated with "anger"). We then want to measure how much more likely the Android account is to use an emotionally-charged term relative to the iPhone account. Since this is count data, we can use a Poisson test to measure the difference:

```
sources <- tweet_words %>%
  group_by(source) %>%
  mutate(total_words = n()) %>%
  ungroup() %>%
  distinct(id, source, total_words)

by_source_sentiment <- tweet_words %>%
  inner_join(nrc, by = "word") %>%
  count(sentiment, id) %>%
  ungroup() %>%
  complete(sentiment, id, fill = list(n = 0)) %>%
  inner_join(sources) %>%
  group_by(source, sentiment, total_words) %>%
  summarize(words = sum(n)) %>%
  ungroup()

head(by_source_sentiment)
```

figure 18_ From the below table, we could see that most people feel quite positive about ACA, the words they used are expressed their trust, anticipation, joy and positive feeling.

```
library(broom)
```

```
sentiment_differences <- by_source_sentiment %>%
  group_by(sentiment) %>%
  do(tidy(poisson.test(.$words, .$total_words)))

#sentiment_differences

#library(reshape2)
#df.long <- melt(by_source_sentiment)
#df.long <- df.long[21:40,]
ggplot(by_source_sentiment, aes(x = sentiment, y = words)) +
  geom_bar(aes(fill=source), stat = "identity",position="dodge") +
  scale_fill_brewer(palette="Spectral") +
  coord_flip() +  theme(axis.text.x=element_text(angle=45, hjust=1), plot.title =
element_text(face="bold", hjust = 0.5, size = 12)) + ggtitle("Sentiment Analysis
from Tweets on ACA")+theme_tufte()+ylab("Counts") + xlab("Word-Emotion Association
lexicon")
```

Data Preparation using Twitter: The Twitter search API does not return an exhaustive list of tweets that match your search criteria, as Twitter only makes available a sample of recent tweets. For a more comprehensive search, we will need to use the Twitter streaming API, creating a database of results and regularly updating them, or use an online service that can do this. Now that we have tweet texts, we need to clean them up before doing any analysis. This involves removing content, such as punctuation, that has no emotional content, and removing any content that causes errors.

```
#text cleaning

library(tm)
# build a corpus, and specify the source to be character vectors
myCorpus <- Corpus(VectorSource(tweets.df$text))
# convert to lower case
# tm v0.6
myCorpus <- tm_map(myCorpus, content_transformer(tolower))
# tm v0.5-10
# myCorpus <- tm_map(myCorpus, tolower)
# remove URLs
removeURL <- function(x) gsub("http[^[:space:]]*", "", x)
# tm v0.6
myCorpus <- tm_map(myCorpus, content_transformer(removeURL))
# tm v0.5-10
# myCorpus <- tm_map(myCorpus, removeURL)
# remove anything other than English letters or space
removeNumPunct <- function(x) gsub("[^[:alpha:][:space:]]*", "", x)
myCorpus <- tm_map(myCorpus, content_transformer(removeNumPunct))
# remove punctuation
# myCorpus <- tm_map(myCorpus, removePunctuation)
# remove numbers
# myCorpus <- tm_map(myCorpus, removeNumbers)
# add two extra stop words: "available" and "via"
myStopwords <- c(stopwords('english'), "available", "via")
# remove "r" and "big" from stopwords
myStopwords <- setdiff(myStopwords, c("r", "big"))
# remove stopwords from corpus
myCorpus <- tm_map(myCorpus, removeWords, myStopwords)
# remove extra whitespace
myCorpus <- tm_map(myCorpus, stripWhitespace)
# keep a copy of corpus to use later as a dictionary for stem completion
myCorpusCopy <- myCorpus
# stem words
```

```
myCorpus <- tm_map(myCorpus, stemDocument)
# inspect the first 5 documents (tweets)
# inspect(myCorpus[1:5])
# The code below is used for to make text fit for paper width
for (i in c(1:2, 320)) {
cat(paste0("[", i, "] "))
writeLines(strwrap(as.character(myCorpus[[i]]), 60))
}
## [1] exampl call java code r
## [2] simul mapreduc r big data analysi use flight data rblogger
## [320] r refer card data mine now cran list mani use r function
## packag data mine applic
# tm v0.5-10
# myCorpus <- tm_map(myCorpus, stemCompletion)
# tm v0.6
stemCompletion2 <- function(x, dictionary) {
x <- unlist(strsplit(as.character(x), " "))
# Unexpectedly, stemCompletion completes an empty string to
# a word in dictionary. Remove empty string to avoid above issue.
x <- x[x != ""]
x <- stemCompletion(x, dictionary=dictionary)
x <- paste(x, sep="", collapse=" ")
PlainTextDocument(stripWhitespace(x))
}
#myCorpus <- lapply(myCorpus, stemCompletion2, dictionary=myCorpusCopy)
myCorpus <- Corpus(VectorSource(myCorpus))

# count frequency of "mining"
miningCases <- lapply(myCorpusCopy,
function(x) { grep(as.character(x), pattern = "\\<mining")} )
sum(unlist(miningCases))
## [1] 82
# count frequency of "miner"
minerCases <- lapply(myCorpusCopy,
function(x) {grep(as.character(x), pattern = "\\<miner")} )
sum(unlist(minerCases))
## [1] 5
# replace "miner" with "mining"
myCorpus <- tm_map(myCorpus, content_transformer(gsub),
pattern = "miner", replacement = "mining")

tdm <- TermDocumentMatrix(myCorpus,
control = list(wordLengths = c(1, Inf)))
tdm
## <<TermDocumentMatrix (terms: 822, documents: 320)>>
## Non-/sparse entries: 2460/260580
## Sparsity : 99%
## Maximal term length: 27
## Weighting : term frequency (tf)
#__figure 17___
#Frequent Words and Associations

term.freq <- rowSums(as.matrix(tdm))
term.freq <- subset(term.freq, term.freq >= 15)
df <- data.frame(term = names(term.freq), freq = term.freq)
library(ggplot2)
ggplot(df, aes(x = term, y = freq)) + geom_bar(stat = "identity") +
xlab("Terms") + ylab("Count") + coord_flip()
```

figure 19_ Unsurprisingly, Word Cloud shows that most frequently mentioned terms are health, care, affordable, aca, and act. Except the discussion of insurance policy itself, people do talk frequently about political-related topics, such as Republican Party (rt), Grand Old Party (gop), and Obama. Also, some terms related to push forward or hold back the policy, such as courage, urge and repeal. Financial words are used, like pay, bill and save. Finally, specific names of people and place were mentioned in the tweets (i.e., Larry Levitt and Mecklenburg).

```
#library(graph)
#source("http://bioconductor.org/biocLite.R")
#biocLite("Rgraphviz")
#library("Rgraphviz")
#plot(tdm, term = term.freq, corThreshold = 0.2, weighting = T)

m <- as.matrix(tdm)
# calculate the frequency of words and sort it by frequency
word.freq <- sort(rowSums(m), decreasing = T)
# colors
library(RColorBrewer)
pal <- brewer.pal(9, "BuGn")
pal <- pal[-(1:4)]

# plot word cloud
library(wordcloud)
wordcloud(words = names(word.freq), freq = word.freq, min.freq = 3,
random.order = F, colors = pal)
```

figure 20_

Topic Modelling

```
dtm <- as.DocumentTermMatrix(tdm)
library(topicmodels)
lda <- LDA(dtm, k = 8) # find 8 topics
(term <- terms(lda, 6)) # first 6 terms of every topic

chapter_topics <- tidy(lda, matrix = "beta")
chapter_topics

top_terms <- chapter_topics %>%
  group_by(topic) %>%
  top_n(5, beta) %>%
  ungroup() %>%
  arrange(topic, -beta)
top_terms
```

In text mining, we often have collections of documents, such as social media posts or news articles, that we'd like to divide into natural groups so that we can understand them separately. Topic modeling is a method for unsupervised classification of such documents, similar to clustering on numeric data, which finds natural groups of items even when we're not sure what we're looking for.

Latent Dirichlet allocation (LDA) is a particularly popular method for fitting a topic model. It treats each document as a mixture of topics, and each topic as a mixture of words. This allows documents to "overlap" each other in terms of content, rather than being separated into discrete groups, in a way that mirrors typical use of natural language.

Latent Dirichlet allocation is one of the most common algorithms for topic modeling. Without diving into the math behind the model, we can understand it as being guided by two principles: Every document is a mixture of topics and Every topic is a mixture of words.

This visualization lets us understand the eight topics that were extracted from the tweets. The most common words in topic 1 include "rt", "care", and "health", which suggests it may represent health insurance and republican issues. Those most common in topic 2 include "obama", "act", and "bill", suggeting that this topic represents issues related to obamacare. One important observation about the words in each topic is that some words, such as "act" are common within both topics. This is an advantage of topic modeling as opposed to "hard clustering" methods: topics used in natural language could have some overlap in terms of words.

```
library(ggplot2)

top_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()
```

Next Steps:

We want to do more for the text analysis part, maybe include some sentiment analysis. And for the New York Times analysis, when we convert the articles into metadata, we abandoned many variables, which also limit the possibility to explore some possible relationship, so we might try to do some text analysis while control other variables. As we were not able to download all the tweets we didn't finish the text analysis based on tweets. Maybe we would like to try that in the future.