Playground

Frances Hung

10/21/2017

Motivation

According to the CDC, suicide was the 10th leading cause of death in the US in 2015, and the 2nd leading cause of death among adolescents and young adults. Psychological disorders, particularly depression, are a significant risk factor for suicide especially when they go untreated. There is no reliable way to predict who is at risk for committing suicide, because most screening approaches depend on self-report information and people contemplating on suicide would often deny it when asked.

In the first part of our project, we hence aim to build a logistic regression model to identify important variables in predicting suicide rates. Due to the limits of our data, we consider the period from 2004-2015, within the scope of cities in California. One interesting explanatory variable we use is Google search term data (under the product of "Google Trends"). Our hypothesis is that individuals are more likely to tell the truth to Google, than on a questionnaire. In the second part of our project, we build a series of maps using the ArcGIS software. Using suicide rate and mental health treatment facilities data as well as Google search term data, our project aims to map the demand for and supply of mental health treatment in California cities.

Ultimately, we hope to shed some light on important explanatory variables correlated with suicide rates (with the regression model), and to help identify cities where there is a large treatment service gap (with the maps) so that we can address this problem in a more data-driven way.

- use result/visualization as hook

Variable choice (to be moved to preceding corresponding R chunks)

Ideally, the response variable that we are interested in is the gap between the demand and supply of mental health treatment. Which areas are over/under-served, and why? This would be very useful information to policy makers, mental health service providers, related non-profits and such. However, such a variable does not exist (or we could not find it), and we would have had to create an algorithm to derive this data from other existing variables. We could not decide on an accurate way to code "demand" (and what weights to give each component). Furthermore, even though "supply" is more straightforward, there also exists discrepancies between the size of the facilities, or the affordability of the services that would need to be captured by our variable. In the end, we decided that we would use suicide rate as a response variable, although we agreed that it would be an interesting extension to look at service gap. We also hope that our GIS maps would help our audience to begin to think about and identify areas which are under-served.

The original datasets we start with include: - List of verified mental health treatment clinics and facilities (downloaded from ReferenceUSA). We only included places with a certified psychiatrist or psychologist, and which focuses on general mental health (excluding substance abuse facilities) - Google search frequency by city on "depression" as a mood (to exclude unrelated searches on economic depression etc) from 2004-2015 (downloaded from Google Trends). The "hits" values are calculated on a scale from 0 to 100, where 100 is the location with the most popularity as a fraction of total searches, where 50 indicates a location which is half as popular and so on. - Number of suicides by zipcode from 2004-2015. We downloaded leading causes of death data from California Health and Human Services Agency and filtered for cause of death is suicide. - Demographic data downloaded from SimplyAnalytics, including racial and gender makeup, age, marriage, education level, employment, income, healthcare, etc. - Cities long lat data (if possible, could we find a dataset which has a more exhaustive list, or I could ask Warren..)

Playground

We originally intended to look at suicide rate and Google Trends data from one year, eg. 2015, but the logit model returned no significant variables as both suicide rate and depression search fluctuate a lot each year, influenced by factors like celebrity suicides which are not directly relevant to population mental health. Hence, we decided to aggregate suicide rate and Google Trends data over 12 years (constrained by data availability), from 2004 to 2015. Since demographic information is fairly stable over time, we used demographic information from the most recent year to train our model.

```
require(gtrendsR)
```

```
## Loading required package: gtrendsR
```

```
require(ggplot2)
```

```
## Loading required package: ggplot2
```

```
require(dplyr)
```

```
## Loading required package: dplyr
```

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

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

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

```
require(zipcode)
```

```
## Loading required package: zipcode
```

```
data("zipcode")
require(ggmap)
```

```
## Loading required package: ggmap
```

Making Dataframes

Longitude and latitude of cities (for mapping them later)

```
cities_longlat<-read.csv("cal_cities.csv",header=TRUE) %>% select(c(location,Latitude,Longitude))
```

Full gtrends data for all cities for depression search

```
gtrends_full <- read.csv("gtrends_20042015_full.csv") %>%
  `colnames<-`(c("location", "hits"))
```

Prepping facilities data to find number of facilities per city

```
facilities<-read.csv("facilities_final.csv",header = TRUE)
colnames(facilities)[7]<-"zip"
facilities$zip<-as.character(facilities$zip)
city_facs<-facilities %>% group_by(City) %>% summarise(facility_cnt=n())
colnames(city_facs)[1]<-"location"
```

Prepping suicide data to find number of suicides per city, 2004-2015

```
suis<-read.csv(file="death.csv",header=TRUE) %>% filter(Causes.of.Death=="SUI") %>% filter(Year >= 2004)
colnames(suis)[2]<-"zip"
suis$zip<-as.character(suis$zip)
suis2 <-inner_join(zipcode,suis,by="zip")
# aggregate suicide data across all the years for each city
city_suis<-suis2 %>% group_by(city) %>% summarise(suicides=sum(Count))
colnames(city_suis)[1]<-"location"

# wrangled the data for the purpose of GIS (to use later)
zip_suis <- suis %>% group_by(zip) %>% summarise(suicides=sum(Count))
```

Adding in city demographic data for 2017

```
citydem<-read.csv("citydems.csv",header=TRUE)
citydem2<-read.csv("citydems2.csv",header=TRUE)
citydem2$Name<-gsub(",.*","",citydem2$Name)
citydem$Name<-gsub(",.*","",citydem$Name)
citydem$FIPS<-NULL
citydem2$FIPS<-NULL
colnames(citydem)<-c("location", "male", "female","healthcare","bluecollar","whitecollar","nonfamily","n
citydem<-inner_join(citydem,city_facs,by="location")
# Remove the Burbank and Mountain View entries that refer to census-designated areas (duplicate names w
citydem <- citydem [-c(36,121), ]
citydem2 <- citydem2 [-c(636,803), ]
citydem$facility_cnt<-citydem$facility_cnt*100000/citydem$pop
colnames(citydem2)<-c("location","healthcarepp","activities","socialRec","entertainment","pov","presdrug

# data weangling for GIS mapping
zipcode_dem <- read.csv("explansToViz-zipcode.csv")
zipcode_dem$Name<-gsub(",.*","",zipcode_dem$Name)
zipcode_dem$FIPS<-NULL
colnames(zipcode_dem)[1]<-"zip"
zipcode_dem$zip<-as.character(zipcode_dem$zip)
zipcode_dem2 <- zipcode_dem %>% left_join(zip_suis,by="zip") %>% left_join(zipcode,by="zip") %>% filter
write.csv(zipcode_dem2,"gis_zip_dem.csv")
# add city area (in square miles) info from GIS
landArea <- foreign::read.dbf("LandCity.dbf")
landArea <- landArea[,c(1,3)]
colnames(landArea) <- c("location", "landArea")
```

Append all dataframes

```
logtable <- inner_join(citydem,gtrends_full,by="location") %>% inner_join(city_suis,by="location") %>% i
# now the table has 174 cities, whereas the full list of gtrends had 200. Not a big loss

logtable_crop <- logtable [,-c(1)]
# normalize the explanatory variables data frame
logtable_crop_normalized <- scale(logtable_crop) %>% data.frame()

# for purpose of identifying which cities to map
zip.no <- zipcode_dem2 %>% group_by(city) %>% summarise(zip.n=n())
colnames(zip.no)[1]<-"location"
gistable <- logtable %>% select(c(1,19,24)) %>% left_join(zip.no,by="location")
# We wanted to choose one city with high suicide rate, and one with a low suicide rate. The two cities
```

Build a logit model to predict suicide rate at the city level

```
set.seed(47)
model_full<-lm((suicides)~.,data=logtable_crop_normalized)
summary(model_full)
```

```
##
## Call:
## lm(formula = (suicides) ~ ., data = logtable_crop_normalized)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.22557 -0.06899 -0.00180  0.05733  0.72677
##
## Coefficients: (1 not defined because of singularities)
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -1.605e-15  9.710e-03   0.000 1.000000
## male           1.115e-03  1.627e-02   0.068 0.945491
## female               NA         NA      NA       NA
## healthcare    -1.618e-01  7.436e-02  -2.176 0.031173 *
## bluecollar     1.185e-02  4.023e-02   0.294 0.768804
## whitecollar   -3.798e-02  5.205e-02  -0.730 0.466835
## nonfamily     -1.623e-01  5.393e-02  -3.009 0.003101 **
## medAge         9.488e-02  4.099e-02   2.315 0.022043 *
## AmInd          8.896e-02  1.780e-02   4.996 1.68e-06 ***
## whiteNonHisp  -2.272e-01  2.807e-01  -0.809 0.419616
## hisp          -2.652e-01  2.194e-01  -1.208 0.228893
## white          7.713e-03  9.891e-02   0.078 0.937955
## black         -1.785e-01  8.113e-02  -2.200 0.029437 *
## asian         -7.441e-02  1.581e-01  -0.471 0.638585
## medIncome      4.801e-02  6.853e-02   0.701 0.484734
## lessHS         7.282e-03  3.656e-02   0.199 0.842418
## HS            -7.293e-02  4.971e-02  -1.467 0.144601
## Bachelors     -2.760e-02  4.618e-02  -0.598 0.551018
## pop           -8.469e-03  1.128e-02  -0.751 0.453785
## unmarriedMpop  3.667e-02  3.052e-02   1.202 0.231467
## unemployed    -4.124e-02  1.361e-02  -3.030 0.002905 **
## facility_cnt   1.138e-01  3.364e-02   3.381 0.000930 ***
## hits           2.228e-03  1.388e-02   0.160 0.872731
## healthcarepp  -3.657e+01  1.706e+01  -2.144 0.033755 *
## activities     4.908e+00  4.503e+00   1.090 0.277572
## socialRec     -1.018e+01  5.245e+00  -1.940 0.054330 .
## entertainment  2.867e+01  1.363e+01   2.104 0.037173 *
## pov            5.372e-02  3.502e-02   1.534 0.127221
## presdrugs      1.412e+01  6.411e+00   2.202 0.029258 *
## healthcarebiz -1.012e-01  8.135e-02  -1.244 0.215573
## landArea      -5.971e-03  1.664e-02  -0.359 0.720289
## pop_dens       6.066e-02  1.714e-02   3.539 0.000543 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1281 on 143 degrees of freedom
## Multiple R-squared:  0.9864, Adjusted R-squared:  0.9836
## F-statistic: 346.8 on 30 and 143 DF,  p-value: < 2.2e-16
```

Significant variables include: Facility density (#mental health facilities/100,000 people) (+), % population native (+), median age (+), healthcare spending per person (big -), social/recreation/gym club spending (big -), prescription drug spending (big +), population density (+), unemployment rate (-), and movies/parks/museum spending (big +).

```
#ggplot(logtable_crop,aes(x=whitecollar,y=log(suicides)))+geom_point()
```

```
# clustering_table<-logtable_full_crop_normalized %>% select(presdrugs,socialRec,healthcarepp,facility_
# rownames(clustering_table)<-logtable_full[,1]
# dist_whole<-dist(clustering_table)
# cluster_whole<-hclust(dist_whole,method="centroid")
# plot(cluster_whole, labels=logtable_full[,1])
# groups=cutree(cluster_whole,k=20)
# groups
# x<-cbind(clustering_table, groups)
#
# suis_cluster<-function(clus) {
#   sd<-logtable_full %>% filter(location %in% (rownames(subset(x,groups==clus)))) %>% .[["suicides"]] 
#   mean<-logtable_full %>% filter(location %in% (rownames(subset(x,groups==clus)))) %>% .[["suicides"]
#   #View(logtable_full%>% filter(location %in% (rownames(subset(x,groups==clus)))))
#   return(c(mean,sd))
# }
#  suis_cluster(5)
# #(lapply(1:12,suis_cluster))
```

```
# set.seed(10)
# kcluster<-kmeans(clustering_table,20, nstart=20)$cluster
# kcluster
# y<-cbind(clustering_table,kcluster)
# logtable_full %>% filter(location %in% (rownames(subset(y,groups==17))))
```

}}