Data Analysis

Katerina, Remy and Helena

11/15/2019

Add 1 paragraph describing your data set and 1 paragraph outlining your proposed in-class and out-of-class methods.

The Dialect Survey uses a series of questions, including rhyming word pairs and vocabulary words, to explore the distribution of dialects in American. We have 122 survey responses from 47,472 people from different city, state and zip code areas. The majority of participants were from the east coast, and approximately a third of the particpants were in the 20-29 age range.

In this project, we would use these data to fit a KNN classification model showing predicted class membership at each location in the US based on dialect. Our explanatory variable is zipcode and response is dialect. In addition, our second analysis will be based on a topic we haven't learned in class such as, neural networks to help us cluster, classify and recognize patterns of the data.

```r
##Libraries
library(ggplot2)
```

```
## Registered S3 methods overwritten by 'ggplot2':
##   method         from
##   [.quosures     rlang
##   c.quosures     rlang
##   print.quosures rlang
```

```
## Registered S3 method overwritten by 'dplyr':
##   method               from
##   as.data.frame.tbl_df tibble
```

```r
library(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
```

```r
library(readr)
library(caret)
```

```
## Loading required package: lattice
```

```r
dialect_survey<-read_csv("dialect_survey.csv")
```

```
## Parsed with column specification:
## cols(
##   .default = col_integer(),
##   ID = col_character(),
##   CITY = col_character(),
##   STATE = col_character(),
##   ZIP = col_character()
## )
```

```
## See spec(...) for full column specifications.
```

```
zip_codes<-read_csv("zipcodes.csv")
```

```
## Parsed with column specification:
## cols(
##   ZIP = col_character(),
##   LAT = col_double(),
##   LNG = col_double()
## )
```

```
##dataset with only ID, state, city, zip code, and answer to lawyer question
lawyer_response<-dialect_survey[,c(1:4, 18)]

for(i in seq_len(ncol(lawyer_response))){
  print(names(lawyer_response)[i])
  print(sum(is.na(lawyer_response[[i]])))
}
```

```
## [1] "ID"
## [1] 0
## [1] "CITY"
## [1] 537
## [1] "STATE"
## [1] 3
## [1] "ZIP"
## [1] 0
## [1] "Q014"
## [1] 0
```

```
##we examined whether there are missing values and found 3 pieces of missing data for state, and 537 fo

##cleaned up our data: got rid of city and ID
lawyer_response<-dialect_survey[,c(3, 4, 18)]

lawyer_response_cleaned<-mutate(lawyer_response, ZIP=substr(ZIP, 2, 6))
lawyer_response_cleaned
```

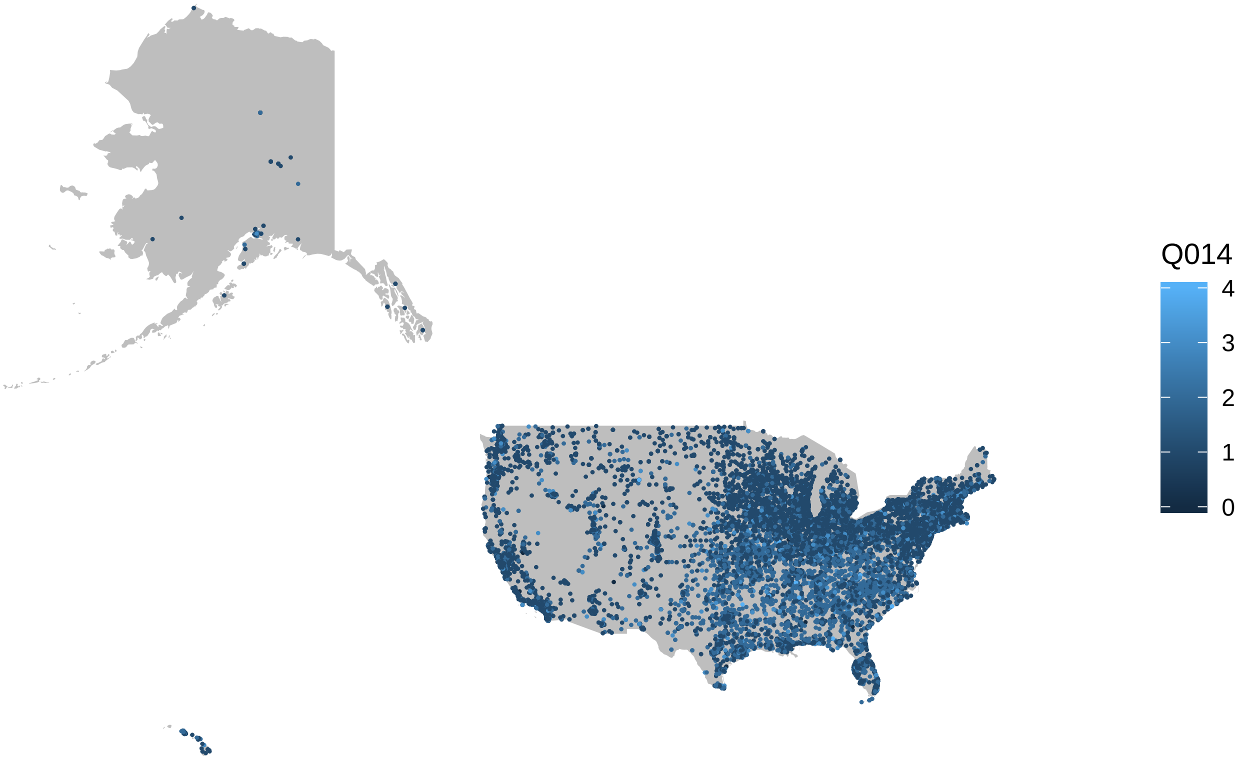
```
## # A tibble: 47,471 x 3
##    STATE ZIP    Q014
##    <chr> <chr> <int>
##  1 ID    83704     0
##  2 MA    01201     1
##  3 VT    05401     1
##  4 PA    18042     1
##  5 MA    01730     1
##  6 TX    77479     2
##  7 MA    02066     1
##  8 MD    21044     1
##  9 MN    56150     1
## 10 MA    01033     1
## # ... with 47,461 more rows
```

```
results<-left_join(x=lawyer_response_cleaned, y=zip_codes, by="ZIP", copy=FALSE)
results<-results[!is.na(results$LAT) & !is.na(results$LNG), ]




# Get the world polygon and extract USA
library(maps)
USA <- map_data("world") %>% filter(region=="USA")


# Left chart
g<-ggplot() +
  geom_polygon(data = USA, aes(x=long, y = lat, group = group), fill="grey") +
  geom_point(data=results, aes(x=LNG, y=LAT, color=Q014), size=0.1) +
  theme_void()+coord_map(xlim = c(-180, -50),ylim = c(18, 72))
g
```



```
results %>% group_by(Q014)%>% count()
```

```
## # A tibble: 5 x 2
## # Groups:   Q014 [5]
##    Q014     n
##   <int> <int>
## 1     0   426
## 2     1 33083
## 3     2 10414
## 4     3  2376
## 5     4   136
```

```r
#KNN Classification Model

library(ISLR)

results$Q014 = factor(results$Q014,
                      levels = c("0","1","2","3","4"))

set.seed(87053)
train_inds <- caret::createDataPartition(results$Q014, p = 0.8)
Data_train <- results %>% dplyr::slice(train_inds[[1]])
Data_test <- results %>% dplyr::slice(-train_inds[[1]])
val_folds <- caret::createFolds(Data_train$Q014, k = 10)

#Select K for K nearest neighbors classification

k_vals <- c(1:10, 25, 50, 75, 100, 150, 200, 250, 300)
results2 <- expand.grid(
    fold_ind = seq_len(10),
    k = k_vals,
    val_class_error = NA
  )
for(i in seq_len(10)) {
  train_data <- Data_train %>% dplyr::slice(-val_folds[[i]])
  val_data <- Data_train %>% dplyr::slice(val_folds[[i]])

  for(k in k_vals) {
    knn_fit <- train(
      form = Q014 ~LAT+LNG,
      data = train_data,
      method = "knn",
      preProcess = "scale",
      trControl = trainControl(method = "none"),
      tuneGrid = data.frame(k = k)
    )

    # get predicted values
    y_hats <- predict(knn_fit, newdata = val_data, type = "raw")

    # classification error rate
    save_ind <- which(results2$fold_ind == i & results2$k == k)
    results2$val_class_error[save_ind] <- mean(y_hats != val_data$Q014)
  }
}
results2 %>%
  group_by(k) %>%
  summarize(mean(val_class_error))
```

```
## # A tibble: 18 x 2
##         k `mean(val_class_error)`
##     <dbl>                   <dbl>
##  1     1                   0.321
##  2     2                   0.312
##  3     3                   0.299
##  4     4                   0.292
##  5     5                   0.286
##  6     6                   0.282
##  7     7                   0.280
##  8     8                   0.279
##  9     9                   0.277
## 10    10                   0.277
## 11    25                   0.274
## 12    50                   0.272
## 13    75                   0.272
## 14   100                   0.270
## 15   150                   0.271
## 16   200                   0.271
## 17   250                   0.271
## 18   300                   0.271
```

```
##Neural Network
library(neuralnet)
```

```
##
## Attaching package: 'neuralnet'

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

```
results_tiny_train<-results[1:10,]
results_tiny_test<-results[11:20,]


nn=neuralnet(Q014~LAT+LNG,data=results_tiny_train, hidden=3,act.fct = "logistic",
                linear.output = FALSE)
plot(nn)

predict_nn=compute(nn, results_tiny_test)

predict_nn$net.result
```

```
##              [,1]         [,2]         [,3]
##  [1,] 0.09486156 0.0001721693 0.9591641537
##  [2,] 0.01218778 0.9801897212 0.0001970873
##  [3,] 0.89083812 0.0782823162 0.0519943090
##  [4,] 0.01225681 0.9800825872 0.0001980470
##  [5,] 0.10268400 0.0001968101 0.9537389790
##  [6,] 0.01222183 0.9801368635 0.0001975609
##  [7,] 0.80436544 0.1528254828 0.0272894125
##  [8,] 0.01213964 0.9802644454 0.0001964177
##  [9,] 0.01245638 0.9797730624 0.0002008173
## [10,] 0.01214157 0.9802614486 0.0001964446
```

https://www.r-graph-gallery.com/330-bubble-map-with-ggplot2.html

https://www.datacamp.com/community/tutorials/neural-network-models-r