QMSS Final Project - Alternative Data

The Physician Compare website was created by the Centers for Medicare & Medicaid Services (CMS) in December 2010 as required by the Affordable Care Act (ACA) of 2010 to help patients assess and find doctors and hospitals. This dataset contains the information supplied to patients via that website, including patient satisfaction surveys and performance scores across over 100 metrics.

Looking at individual physician scores:

- MIPS
- Performance by measure category
- Organization MIPS

Possible problems to solve:

- Prediction of whether to see a physician based on threshold
- Predict physician score based on all factors (predict MIPS?)
  - classification of whether you should go to a doctor based on MIPS score
  - train on several classification methods to analyze best approach for this data

```
# empty cells as NA
indiv_measure <- read.csv("alt_data/Individual_Measure_2017.csv",
                       header = T, na.strings=c("", "NA"))
indiv_mips <- read.csv("alt_data/Individual_MIPS_2017.csv",
                        header = T, na.strings=c("", "NA"))
group_mips <- read.csv("alt_data/Group_MIPS _2017.csv",
                        header = T, na.strings=c("", "NA"))
national <- read.csv("alt_data/National_2017.csv",
                     header = T, na.strings=c("", "NA"))
```

Individual Measures (performance by individual doctor for a task category)

Actually might want to ignore collection type, may have too many for each task category

```
# AFM affiliations too sparse
# remove measure.code -- maintain individuals as observations
# don't have inverse measure calculation method
# Reported on PC Live Site: only "N" measure in downloadable

idv_measure <- indiv_measure[c("NPI", "PAC.ID", "Last.Name",
                               "First.Name", "Measure.Title",
                               "Inverse.Measure",
                               "Measure.Performance.Rate",
                               "Denominator.Count")] %>%
  filter(., Inverse.Measure == "N") %>%
  separate(., Measure.Title, sep = ":", into = "Measure.Name", remove = T) %>%
  na.omit()
# score per title per doctor
# dups for the same measure title may have DIFFERENT scores
# take average by measure title
```

```r
# Some measure titles have multiple collection types
# about 8,000 of the 360,000 distict doctor-measure.title combinations

unique_source <- group_by(idv_measure, PAC.ID, Measure.Name) %>%
  summarise(., Performance.Rate = as.integer(mean(Measure.Performance.Rate)),
            Patient.Count = as.integer(mean(Denominator.Count)))

# merge back to claims (PAC.ID more reliable than NPI)
# remove unneccesary columns
remove <- c("Inverse.Measure",
            "Measure.Performance.Rate",
            "Denominator.Count")
idv_score <- inner_join(unique_source,
                        idv_measure[, !(names(idv_measure) %in% remove)],
                        by = c("PAC.ID","Measure.Name"))
idv_score <- distinct(idv_score)
```

```r
# let's use categories where the counts are greater than 1000
categories <- group_by(idv_score, Measure.Name) %>%
  summarise(., count = n()) %>%
  filter(., count >= 1000)
categories <- categories$Measure.Name
idv_score <- subset(idv_score, Measure.Name %in% categories)

col_title <- as.data.frame(cbind(
  "Name" = categories,
  "Count" = paste0("count_", categories)),
  stringsAsFactors = F)

num_dr <- length(unique(idv_score$PAC.ID))
dr_id <- unique(idv_score$PAC.ID)

df <- data.frame(matrix(NA, nrow = num_dr, ncol = ((length(categories) * 2) + 1)))
colnames(df) <- c("PAC_id", categories, col_title$Count)
df$PAC_id <- dr_id

write.csv(idv_score, file = "idv_score_preprocess.csv", sep = ",", na = "NA", col.names = T, row.names =
```

```
## Warning in write.csv(idv_score, file = "idv_score_preprocess.csv", sep =
## ",", : attempt to set 'col.names' ignored

## Warning in write.csv(idv_score, file = "idv_score_preprocess.csv", sep =
## ",", : attempt to set 'sep' ignored
```

```r
write.csv(df, file = "empty_PAC_preprocess.csv", sep = ",", na = "NA", col.names = T, row.names = F)
```

```
## Warning in write.csv(df, file = "empty_PAC_preprocess.csv", sep = ",", na =
## "NA", : attempt to set 'col.names' ignored

## Warning in write.csv(df, file = "empty_PAC_preprocess.csv", sep = ",", na =
## "NA", : attempt to set 'sep' ignored
```

```r
dr_perf <- read.csv2("dr_by_measure.csv", header = T, na.strings = c("", "NA"), sep = ",")
```

Individual MIPS (Merit-Based Incentive Payment System) - to adjust medicare part B payments or not

- ACI (Advancing Care Information)
  - ACI >= 0 : clinician reported the ACI category
  - ACI >= 50: clinicial acheived base score for ACI

- IA (Improvement Activities0)
- MIPS
  - MIPS < 30: Negative Payment Adjustment
  - MIPS >= 75: POsitive Payment Adjustment
  - 15% cost score, 45% quality score, 15% improvement activies score, 35% promoting interoperability (PI)

```r
indiv_mips <- distinct(indiv_mips)
colnames(indiv_mips)[2] <- "PAC_id"

# map MIPS to task scores
dr_score <- inner_join(indiv_mips, dr_perf, by = "PAC_id")
```

```r
# too much variance in state/zip data for unique organization IDs
# all but ~1000 group_mips PAC ID's are represented in here
national <- national[colnames(national)[c(1:5, 8:17, 19, 25)]]



dup_ID <- distinct(national) %>%
  group_by(., PAC.ID) %>%
  summarise(., count = n()) %>%
  filter(., count == 1) %>%
  .$PAC.ID

# extra observations for a doctor might bias data
# remove doctors of multiple organizations
dr_org <- subset(national, PAC.ID %in% dup_ID) %>%
  distinct(.)

# how many secondary specialties does a doctor have
dr_org$Num.Secondaries <- 4 - rowSums(is.na(dr_org[11:14]))
dr_org <- dr_org[c(1:10, 18, 16, 17)]
```

```r
# Need to merge by PAC ID, some organizations use the same legal name
org_mips <- inner_join(dr_org, group_mips, by = c("Group.Practice.PAC.ID" = "Organization.PAC.ID"))
```

```r
colnames(dr_score)[5:9] <- c("Score_Source_Dr", "Quality_Score_Dr",
                             "ACI_Score_Dr", "IA_Score_Dr", "MIPS_Score_Dr")
colnames(org_mips)[15:19] <- c("Score_Source_Org", "Quality_Score_Org",
                             "ACI_Score_Org", "IA_Score_Org", "MIPS_Score_Org")
colnames(org_mips)[2] <- "PAC_id"
```

In Common:
NPI
PAC_id
Last.Name
First.Name

Issues:
Sources.of.scores Quality.Category.Score
ACI.Category.Score IA.Category.Score Final.MIPS.Score

```r
dr_data <- inner_join(org_mips, dr_score, by = c("PAC_id"))[c(-1, -20, -21, -22)]
colnames(dr_data)[3:4] <- c("last_name", "first_name")

# actually going to remove the counts, as they don't add much and are too sparse
dr_data <- dr_data[seq(-57, -89, -1)]

# typage for math operations
for (i in seq(24, 56, 1)) {
  dr_data[, i] <- unfactor(dr_data[, i])
}

colnames(idv_score)[1] <- "PAC_id"

# average performance across all tasks
avg_score <- idv_score %>%
  group_by(., PAC_id) %>%
  summarise(., avg.perf = mean(Performance.Rate))

dr_data <- inner_join(dr_data, avg_score, by = "PAC_id")
```

```r
write.csv(dr_data, file = "full_doctor_scoring.csv",
          sep = ",", na = "NA", row.names = F, col.names = T)
```

```
## Warning in write.csv(dr_data, file = "full_doctor_scoring.csv", sep =
## ",", : attempt to set 'col.names' ignored

## Warning in write.csv(dr_data, file = "full_doctor_scoring.csv", sep =
## ",", : attempt to set 'sep' ignored
```