Hospital Data Report

Zach Christensen

September 4, 2015

Healthcare-Associated Infections, State

The Healthcare-Associated Infections (HAI) measures - state data. These measures are developed by Centers for Disease Control and Prevention (CDC) and collected through the National Healthcare Safety Network (NHSN). They provide information on infections that occur while the patient is in the hospital. These infections can be related to devices, such as central lines and urinary catheters, or spread from patient to patient after contact with an infected person or surface. Many healthcare associated infections can be prevented when the hospitals use CDC-recommended infection control steps.

The metric reported is standardized infection ratios (SIRs). SIRs compare the actual number of Healthcare-Assiocated Infections at each hospital to the predicted number of infections. These ratios are adjusted for various risk factors and population traits. More information about SIRs and how they are calculated can be found here: http://www.leapfroggroup.org/media/file/SIRCalc.pdf. .

Import Libraries and Data Set

The analysis relies on several open source libraries, and uses data directly from the healthdata.gov API.

```
# Load Required Libraries
library(dplyr)
library(tidyr)
library(reshape2)
library(ggplot2)
library(googleVis)
op <- options(gvis.plot.tag = "chart")
library(caret)

# Abreviations for the 50 states
stateAbrevs <- data.frame(c("IL", "IN", "IA", "KY", "MI", "MN", "MO", "OH", "WI",
                "CT", "DE", "ME", "MD", "MA", "NH", "NJ", "NY", "PA", "RI", "VT",
                "AK", "AZ", "CA", "HI", "NV", "OR", "UT", "WA",
                "AL", "AR", "FL", "GA", "LA", "MS", "NC", "SC", "TN", "VA", "WV",
                "CO", "ID", "KS", "MT", "NE", "NM", "ND", "OK", "SD", "TX", "WY"))
regions <- cbind(stateAbrevs,
                c("central", "central", "central", "central", "central", "central", "central", "central
                  "northern", "northern", "northern", "northern", "northern", "northern", "northern", "n
                  "pacific", "pacific", "pacific", "pacific", "pacific", "pacific", "pacific", "pacific
                  "southern", "southern", "southern", "southern", "southern", "southern", "southern", "s
                  "western", "western", "western", "western", "western", "western", "western", "western
colnames(regions) <- c("State", "Region")

# Load data from healthdata.gov
states <- read.csv("http://data.medicare.gov/api/views/k2ze-bqvw/rows.csv?accessType=DOWNLOAD")
```

Preprocessing

The dataset containes 7: State, Measure.Name, Measure.ID, Score, Footnote, Measure.Start.Date, Measure.End.Date. Before the analysis, the data can be restructured and some redundant information removed. The data set is reduced to only contain information for the 50 states.

```
# Date Range
startDate <- unique(states$Measure.Start.Date)
endDate <- unique(states$Measure.End.Date)

# Split the Measure.ID column
states <- states %>% mutate("Measure" = substring(text = Measure.ID, first = 1, last = 5),
                            "Type" = substring(text = Measure.ID, first = 7,
                                               last = length(Measure.ID)))

# List of different measures
measures <- unique(filter(states, Type == "SIR")[,c("Measure", "Measure.Name")])

# Then only select needed columns
states <- subset(states, select = c("State", "Measure.Name", "Measure", "Type", "Score"))
states <- dcast(states, State + Measure ~ Type)

# Join the measures so we get measure names with data
states <- left_join(states, measures)

# Remove rows which have NA
states <- states[-which(is.na(states$SIR)),]

# Create data.frame with just SIRs by state
sirs <- dcast(states, State ~ Measure, value.var = "SIR")

# Reduce the list to only contain the 50 states
sirs <- inner_join(sirs, regions, by = c("State" = "State"))
```

```
## Warning in inner_join_impl(x, y, by$x, by$y): joining factors with
## different levels, coercing to character vector
```

```
# Preview Data
head(sirs)
```

```
##   State HAI_1 HAI_2 HAI_3 HAI_4 HAI_5 HAI_6   Region
## 1    AK 0.526 1.319 0.763 0.849 0.405 0.804  pacific
## 2    AL 0.588 0.926 0.721 0.504 1.144 0.660 southern
## 3    AR 0.510 1.090 0.978 1.079 1.118 0.642 southern
## 4    AZ 0.470 1.213 1.090 1.114 0.992 0.912  pacific
## 5    CA 0.409 1.215 1.044 0.904 0.750 1.053  pacific
## 6    CO 0.438 0.857 0.774 1.005 0.553 1.047  western
```

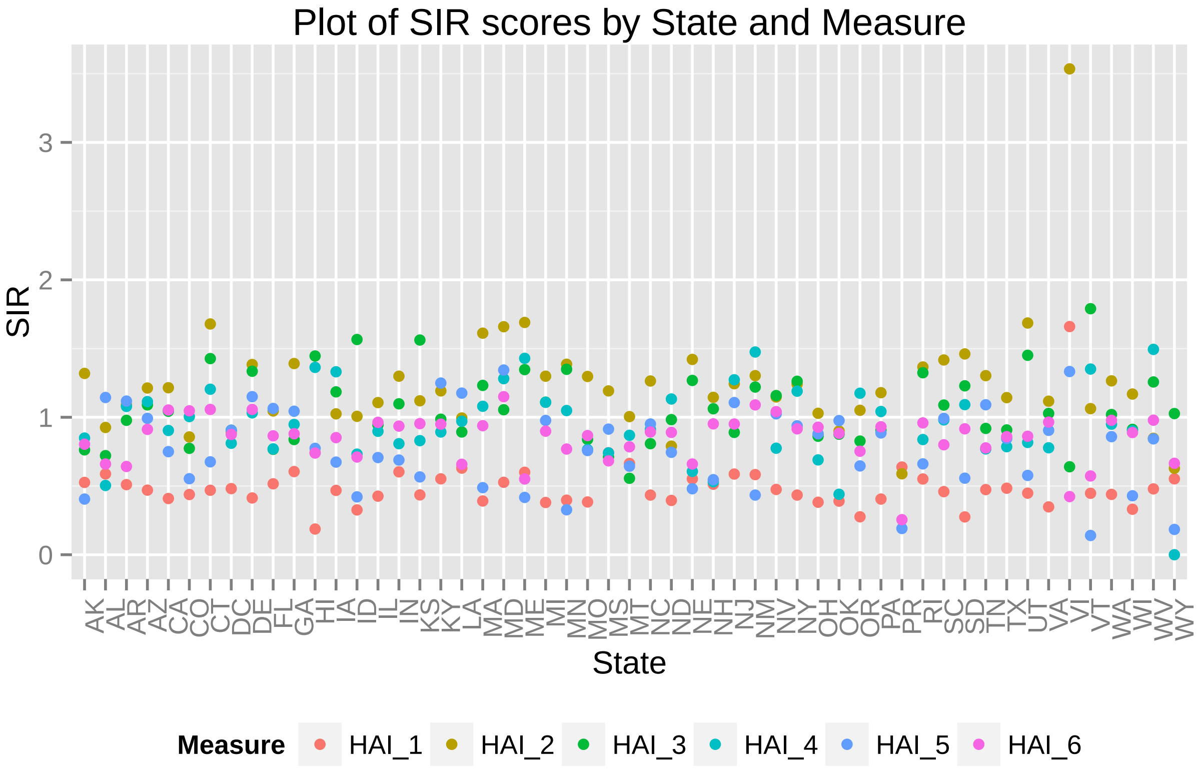
The data for this set was collected from 10/01/2013 to 09/30/2014, with information about 6 different measures collected from 53 states or regions.The 6 measures are various healthcare associated infections:

```
## [1] Catheter-Associated Urinary Tract Infections (CAUTI)
## [2] Clostridium difficile (C.diff.) Laboratory-identified Events (Intestinal infections)
## [3] Methicillin-resistant Staphylococcus Aureus (MRSA) Blood Laboratory-identified Events (Bloodstrea
## [4] Central line-associated blood stream infections (CLABSI)
## [5] Surgical Site Infection from abdominal hysterectomy (SSI: Hysterectomy)
## [6] Surgical Site Infection from colon surgery (SSI: Colon)
## 18 Levels: C.diff Lower Confidence Limit ... Surgical Site Infection from colon surgery (SSI: Colon)
```

Exploration

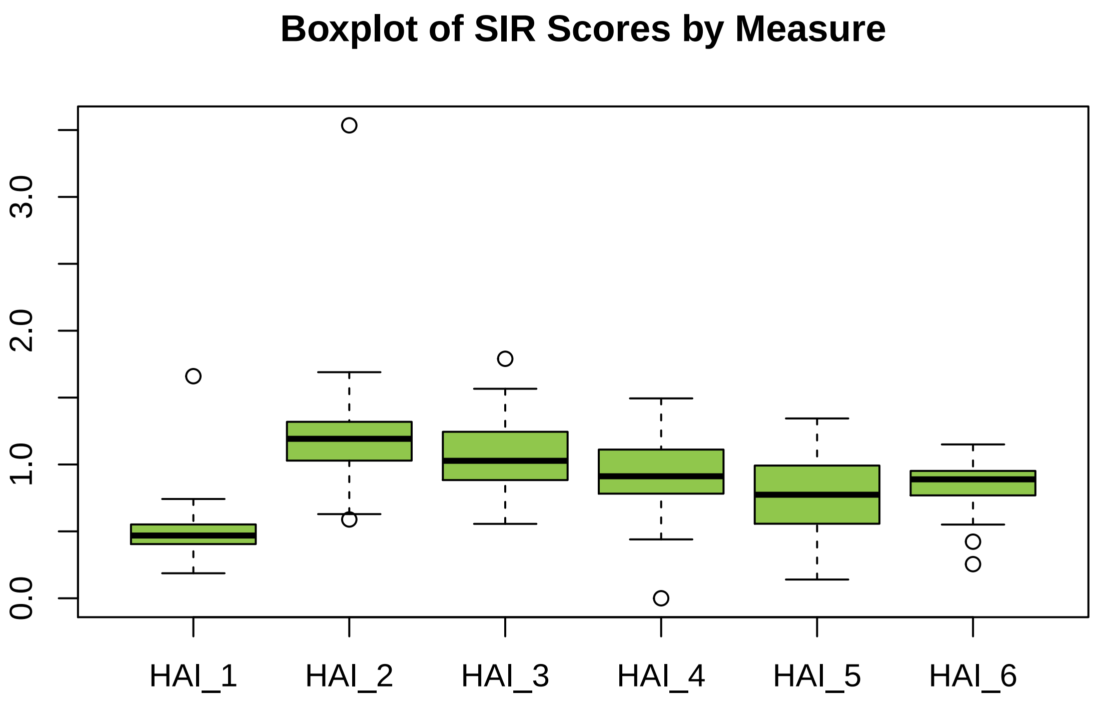
We can plot the SIR scores for each Measure by State.

```
qplot(State, SIR, data = states, col = Measure, main = "Plot of SIR scores by State and Measure") +
  theme(axis.text.x = element_text(angle = 90, hjust = 1), legend.position="bottom")
```



Visually, it looks like the the SIR scores for HAI_2, Catheter-Associated Urinary Tract Infections (CAUTI) are usually the highest, while HAI_1, Central line-associated blood stream infections (CLABSI) are the lowest. Below is the average score and standard deviation grouped by measure, which confirms the visual observation.

```
## Source: local data frame [6 x 3]
##
##   Measure mean(SIR)   sd(SIR)
## 1   HAI_1 0.4933585 0.1942709
## 2   HAI_2 1.2253019 0.4100984
## 3   HAI_3 1.0623462 0.2597521
## 4   HAI_4 0.9484314 0.2809226
## 5   HAI_5 0.7763774 0.2965325
## 6   HAI_6 0.8499434 0.1693796
```



Regional Analysis

```
geo_1 <- gvisGeoChart(data = sirs, colorvar = "HAI_1",
                  options=list(region="US", displayMode="regions",
                               resolution="provinces", width=600, height=400))

plot(geo_1)
```

```
geo_2 <- gvisGeoChart(data = sirs, colorvar = "HAI_2",
                  options=list(region="US", displayMode="regions",
                               resolution="provinces", width=600, height=400))
print(gvisMerge(geo_1, geo_2, horizontal = TRUE), "chart")
```

```
# print(gvisGeoChart(data = sirs, colorvar = "HAI_1",
#                   options=list(region="US", displayMode="regions",
#                                resolution="provinces", width=600, height=400)))
# plot(gvisGeoChart(data = sirs, colorvar = "HAI_2",
#                   options=list(region="US", displayMode="regions",
#                                resolution="provinces", width=600, height=400)))
# plot(gvisGeoChart(data = sirs, colorvar = "HAI_3",
#                   options=list(region="US", displayMode="regions",
#                                resolution="provinces", width=600, height=400)))
```

First, the correlation between the variables should be investigated. This will show how if incidents of Healthcare-Associated Infections are related.

```
cor(sirs[,2:7])
```

```
##            HAI_1      HAI_2       HAI_3       HAI_4       HAI_5
## HAI_1  1.0000000 0.12931414 -0.30863708 -0.15110021  0.16954897
## HAI_2  0.1293141 1.00000000  0.25926305  0.26297632  0.01745694
## HAI_3 -0.3086371 0.25926305  1.00000000  0.34107364 -0.41665337
## HAI_4 -0.1511002 0.26297632  0.34107364  1.00000000  0.03881742
## HAI_5  0.1695490 0.01745694 -0.41665337  0.03881742  1.00000000
## HAI_6 -0.1694669 0.21678285 -0.02889971  0.17031538  0.26476307
##             HAI_6
## HAI_1 -0.16946686
## HAI_2  0.21678285
## HAI_3 -0.02889971
## HAI_4  0.17031538
## HAI_5  0.26476307
## HAI_6  1.00000000
```

Since the variables are not correlated, all of them can be used initially to develop a predictive model.

Confidence Intervals - values that contain 1

```
nrow(filter(states, CI_LOWER < 1, CI_UPPER > 1))
```

```
## [1] 121
```