

ProviderScoring

Rohan

```
library(plyr)
library(dplyr)

##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:plyr':
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize
##
## The following objects are masked from 'package:stats':
##
##   filter, lag
##
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union

nyopth <- read.csv("~/cunyMsda2015/nyopth.csv")

nyopth <- nyopth[,c(2,8,12,16,17,18,20,21,22,23,24,25,26,27,28,29)]

tophcpcs <- names((sort(table(nyopth$hcpcs_code),decreasing = TRUE))[1:10])

scores<-vector(mode="list",length = length(unique(nyopth$npi)))
names(scores) <- unique(nyopth$npi)

scoreFunction <- function(rowdiff, q){
  score = 0

  rowscore = rowdiff[4]
  if (rowscore>=q[1] && rowscore<q[2]){
    score = 0
  }
  else if (rowscore>=q[2] && rowscore<q[3]){
    score = 1
  }
  else if (rowscore>=q[3] && rowscore<q[4]){
    score = 2
  }
  else{
    score = 3
  }
  score
}
```

We loop over all the top hcpcs codes and add to the scores based on the spread of the submitted - payment amount and based on the beneficiaries counts.

```

for (i in tophcpcs){
  tmp = nyopth[nyopth$hcpes_code==i,]
  tmp.avgpymnt <- ddply(tmp,~npi,summarise,mean=mean(average_medicare_payment_amt))
  tmp.avgsbmitted <- ddply(tmp,~npi,summarise,mean=mean(average_submitted_chrg_amt))
  tmp.spread <- merge(tmp.avgpymnt,tmp.avgsbmitted,by.x = "npi",by.y = "npi")
  #Here is where we would have to discount based on AGI or COL
  tmp.spread$diff = tmp.spread$mean.y - tmp.spread$mean.x

  #some kind of function that will give out points depending on the quartile that the npi falls under
  q<-quantile(tmp.spread$diff)
  tmp.spread$sc <- apply(tmp.spread, 1, FUN = scoreFunction,q = q)
}

```