Data Exploration

Sharad Gurung

November 21, 2015

```
library(plyr)
nyop = read.csv("/home/sharad/Downloads/nyop.txt")
```

Exploring ophthalmologists in New York state.

First, we will see provider distribution in NY by city.

```
ny.city = unique(nyop[,c("npi", "nppes_provider_city")])
ny.city.summ = ddply(ny.city, "nppes_provider_city", function(df){
  total = nrow(df)
})
head(ny.city.summ[order(-ny.city.summ$V1),], 5)
```

```
##     nppes_provider_city  V1
## 127            NEW YORK 397
## 23             BROOKLYN 148
## 20                BRONX  61
## 165           ROCHESTER  57
## 1                ALBANY  32
```

```
tail(ny.city.summ[order(-ny.city.summ$V1),], 5)
```

```
##     nppes_provider_city V1
## 197      VALLEY COTTAGE  1
## 202             WARWICK  1
## 207      WEST HEMPSTEAD  1
## 209          WEST NYACK  1
## 215              WILTON  1
```

The provider frequency distribution above will be used for optimization model and simulation. We will assume that providers are proportionally distributed in areas where patients. Areas such as New York City will have a higher provider density compared to Hudson. Since, our dataset does not have beneficiary information we will use this distribution to spread total beneficiaries to cities. This may not be accurate.

```
sum(nyop$bene_unique_cnt)
```

```
## [1] 2489507
```

We need a rough idea of how many beneficiaries there are in this dataset. The total number of Medicare beneficiaries in NY for 2012 was 3,093,591. http://kff.org/medicare/state-indicator/total-medicare-beneficiaries/

We will make some wild assumptions to estimate this number. Let's assume the beneficiary unique count per procedure can be summed up to get this estimate. This is not accurate because there are overlaps. Second we assume the beneficiaries will stay with their provider and not go to various providers for different services within a specialty. For example, if I am seeing one eye specialist, I will go to that specialist for all my eye related needs.

```
ny.hcpcs = nyop[,c("npi","place_of_service","hcpcs_code","hcpcs_drug_indicator","line_srvc_cnt","bene_u

ny.hcpcs.summ = ddply(ny.hcpcs, "hcpcs_code", function(df){
  data.frame(
    provider.count = nrow(df),
    line.count = sum(df$line_srvc_cnt),
    line.min = min(df$line_srvc_cnt),
    line.avg = round(mean(df$line_srvc_cnt),0),
    line.max = max(df$line_srvc_cnt),
    bene.count = sum(df$bene_unique_cnt),
    bene.min = min(df$bene_unique_cnt),
    bene.avg = round(mean(df$bene_unique_cnt),0),
    bene.max = max(df$bene_unique_cnt)
  )
})
head(ny.hcpcs.summ[order(-ny.hcpcs.summ$line.count),], 10)
```

```
##     hcpcs_code provider.count line.count line.min line.avg line.max
## 175      92014           1467     720995       11      491     5006
## 174      92012           1435     528009       11      368     4970
## 191      92226            794     378785       11      477     5568
## 187      92134            825     271999       11      330     5466
## 183      92083           1203     192709       11      160     2483
## 195      92250            921     182299       11      198     3411
## 256      J2778            114     181890       86     1596    10100
## 186      92133            992     142880       11      144     2176
## 77       67028            259     125307       12      484     2330
## 73       66984            729     112197       12      154     4860
##     bene.count bene.min bene.avg bene.max
## 175     551950       11      376     2294
## 174     285580       11      199     2028
## 191     196691       11      248     1636
## 187     135356       11      164     1767
## 183     159887       11      133     1679
## 195     150631       11      164     1319
## 256       9596       11       84      321
## 186     126173       11      127     1698
## 77       31275       11      121      468
## 73       47475       11       65      368
```

Next, we look at services provided; the table above summarizes procedures. The "line" columns show services provided and "bene" columns show beneficiaries. There is much variation in the number of services provided among the providers.

```
ny.hcpcs.summ[ny.hcpcs.summ$hcpcs_code == "92002"
              | ny.hcpcs.summ$hcpcs_code == "92004"
              | ny.hcpcs.summ$hcpcs_code == "92012"
              | ny.hcpcs.summ$hcpcs_code == "92014", ]
```

```
##     hcpcs_code provider.count line.count line.min line.avg line.max
## 172      92002            232       7079       11       31      191
## 173      92004           1186     106601       11       90     1505
```

```
## 174      92012           1435     528009       11      368     4970
## 175      92014           1467     720995       11      491     5006
##     bene.count bene.min bene.avg bene.max
## 172       7071       11       30      191
## 173     106535       11       90     1505
## 174     285580       11      199     2028
## 175     551950       11      376     2294
```

Let's focus on some examination code: 92002, 92004, 92012, and 92014. First two are for new patients and the last two are for established patients. Most services are for existing patients.

We started with a total of 1614 providers. The provides will bill a mix of these procedures.

```
ny.single = ny.hcpcs[ny.hcpcs$npi == "1003018102", ]
ny.single = ny.single[ny.single$hcpcs_code == "92002" | ny.single$hcpcs_code == "92004" | ny.single$hcp
ny.single[,c(
"hcpcs_code","line_srvc_cnt","bene_unique_cnt","bene_day_srvc_cnt","average_medicare_allowed_amount","a
)]
```

```
##   hcpcs_code line_srvc_cnt bene_unique_cnt bene_day_srvc_cnt
## 6      92002            12              12                12
## 7      92004            89              89                89
## 8      92012           263             226               263
## 9      92014           469             453               469
##   average_medicare_allowed_amount average_submitted_chrg_amt
## 6                           74.76                     78.000
## 7                          138.07                    152.921
## 8                           78.68                     82.000
## 9                          114.17                    115.139
##   average_medicare_payment_amt
## 6                      39.8733
## 7                      97.2629
## 8                      55.8826
## 9                      79.5603
```

For the provider above, there could be 453 to 780 beneficiaries; the accurate number is somewhere in between. The total services are equal to or very close to the number of beneficiaries. We can assume most patients get one eye exam a year.

Interestingly, we see more instances of comprehensive exam compared to intermediate exam. Comprehensive exams pay almost twice but are they twice as medically necessary?

```
ny.exam = ny.hcpcs[ny.hcpcs$hcpcs_code == "92002" | ny.hcpcs$hcpcs_code == "92004" | ny.hcpcs$hcpcs_cod
# ny.exam[,c(
# "hcpcs_code","line_srvc_cnt","bene_unique_cnt","bene_day_srvc_cnt","average_medicare_allowed_amount",
# )]

ny.exam.summ = ddply(ny.exam, "hcpcs_code", function(df){
  data.frame(
    provider.count = nrow(df),
    line.sum = sum(df$line_srvc_cnt),
    bene.sum = sum(df$bene_unique_cnt),
    allow.sum = sum(df$average_medicare_allowed_amount),
    submit.sum = sum(df$average_submitted_chrg_amt),
```

```
    paid.sums = sum(df$average_medicare_payment_amt)
  )
})
ny.exam.summ
```

```
##   hcpcs_code provider.count line.sum bene.sum allow.sum submit.sum
## 1      92002            232     7079     7071  19156.26   31064.49
## 2      92004           1186   106601   106535 183114.16  260257.24
## 3      92012           1435   528009   285580 123125.21  181726.63
## 4      92014           1467   720995   551950 185686.50  261179.42
##   paid.sums
## 1  13233.02
## 2 129664.62
## 3  90681.94
## 4 133194.97
```

NY summary shows a similar pattern.

```
ny.exam.summ$collect = ny.exam.summ$allow.sum - ny.exam.summ$paid.sums
ny.exam.summ$collect_allowed = ny.exam.summ$collect/ny.exam.summ$allow.sum
ny.exam.summ
```

```
##   hcpcs_code provider.count line.sum bene.sum allow.sum submit.sum
## 1      92002            232     7079     7071  19156.26   31064.49
## 2      92004           1186   106601   106535 183114.16  260257.24
## 3      92012           1435   528009   285580 123125.21  181726.63
## 4      92014           1467   720995   551950 185686.50  261179.42
##   paid.sums   collect collect_allowed
## 1  13233.02  5923.241       0.3092066
## 2 129664.62 53449.535       0.2918919
## 3  90681.94 32443.273       0.2634982
## 4 133194.97 52491.529       0.2826890
```

Patient responsibility is slightly lower for established intermediate exams, but this may not be significant enough to make a difference.