Comparing Hospital Prices Using Data in Python

Will Silver

Understanding the Data

```
1   Service category,Procedure,Statewide indicator,Hospital,Number of discharges 2018,25th percentile 2018,Median 2018,
2   Diagnostic Imaging and Testing,Bone study,1,Statewide,6907,123,243,390,6725,105,231,387,12,0.052
3   Diagnostic Imaging and Testing,Bone study,0,Adventist Med Ctr,127,241,316,583,137,223,312,440,4,0.013
4   Diagnostic Imaging and Testing,Bone study,0,Adventist Tillamook Reg Med Ctr,35,258,280,289,26,258,260,354,20,0.077
5   Diagnostic Imaging and Testing,Bone study,0,Asante Ashland Comm Hosp,111,324,447,474,132,326,422,516,25,0.059
6   Diagnostic Imaging and Testing,Bone study,0,Asante Rogue Valley Med Ctr,199,351,462,688,207,354,415,654,47,0.113
7   Diagnostic Imaging and Testing,Bone study,0,Asante Three Rivers Med Ctr,134,384,431,663,144,407,431,709,0,0
8   Diagnostic Imaging and Testing,Bone study,0,Blue Mountain Hosp,14,458,470,472,10,458,467,467,3,0.006
9   Diagnostic Imaging and Testing,Bone study,0,Columbia Mem Hosp,44,230,332,596,32,440,440,818,-108,-0.245
10  Diagnostic Imaging and Testing,Bone study,0,Curry Gen Hosp,20,538,568,668,22,472,538,568,30,0.056
```

The data is organized into lines for each procedure offered by every hospital. There are 2751 lines of data. Each line contains the following information in order:

Service Category, Procedure, Statewide indicator, Hospital, Number of discharges 2018, 25th percentile 2018, Median 2018, 75th percentile 2018

Extracting the Data

```
for line in hospital_data_file:
    line = line.strip()
    line = line.split(',')
    if line[0] == "Service category": #skip over the first line
        pass
    else:
        if line[2] == '0': #for all non statewide values
            if line[1] not in hospital_data_dictionary.keys(): #adding a procedure to the data for the first time
                hospital_data_dictionary[line[1]] = [(line[3],line[4], line[5], line[6], line[7])] #stored in a dictionary
                                                                                                   #key = procedure name,
                                                                                                   #vallue = (hsp name, procedure
                                                                                                   #         freq, 25, median, 75)

            else:
                temp_list = hospital_data_dictionary[line[1]] #temporary list of the procedure values
                temp_list.append((line[3],line[4], line[5], line[6], line[7])) #append the data for each hospital
                hospital_data_dictionary[line[1]] = temp_list #replace the original value with the updated list

        elif line[2] == '1': #adding all statewide procedure information to a single list of tuples, formated the same as above
            statewide_data_dictionary[line[1]] = [(line[3], line[4], line[5], line[6], line[7])]


        if line[1] not in procedure_names: #list of all procedure names
            procedure_names.append(line[1])

        if line[3] == "Statewide":
            pass
        elif line[3] not in hospital_names: #list of all hospital names
            hospital_names.append(line[3])
```

Here, we extract the data into easily accessible dictionaries. We also create individual lists of hospital names and procedure names to be referenced later. Details of the process can be found in the code's comments.

Isolating what's Relevant

```
#purpose: normalize hospital median prices for every procedure, relative to eachother
#argument: takes a procedure string as input
#output: creates a dictionary with each hospital as a key, and their respective lists of normalized values
def next_step_analysis(procedure):

    counter = 0
    procedure_median_list = []
    for hospital_info in hospital_data_dictionary[procedure]:
        procedure_median_list.append(int(hospital_info[3])) #creating a list of medians for a selected procedure
        counter+=1

        if hospital_info[0] not in hospital_procedures_info.keys():
            hospital_procedures_info[hospital_info[0]] = [] #creating a dictionary where each hospital name is a key, empty values
```

Here, we isolate all the median prices from all hospitals for a single procedure. With this, we prepare to take the standard deviation of procedure's price range.

Taking the Standard Deviation

```
procedure_median_sum = 0
for procedure_median in procedure_median_list:
    procedure_median_sum+=procedure_median
procedure_median_mean = procedure_median_sum/counter #taking the average of a procedure's medians, a mediann mean


squared_differences_list = []
for procedure_median in procedure_median_list:
    squared_differences_list.append((procedure_median - procedure_median_mean)**2) #subtract procedure mean from median for
                                                                                    # each hospital, add the differences to a list

squared_difference_sum = 0
procedure_median_sd = 0
for squared_difference in squared_differences_list:
    squared_difference_sum += squared_difference
    mean = squared_difference_sum/len(squared_differences_list)
    procedure_median_sd = mean**(1/2) #calculations to arrive at the standard deviation of the procedure's hospital medians
```

Here, we do nothing more than take the standard deviation of the price range in anticipation of normalizing each value.

Normalizing the Values

```
normalized_list = []
for hospital_info in hospital_data_dictionary[procedure]: #accessing list of tupled information for each hospital
    normalized_procedure_value = (int(hospital_info[3])-procedure_median_mean)/procedure_median_sd #now we take the normalized value
                                                                                                    #of each hospital's procedure median.
                                                                                                    #this means the average hospital normalized
                                                                                                    #value for any procedure will be 0.
                                                                                                    #a value such as 1.5 would indicate a higher
                                                                                                    #than average price relative to its alternatives.
    temp_list = hospital_procedures_info[hospital_info[0]] #going back to the dictionary we created before
    temp_list.append(normalized_procedure_value) #adding the normalized value to a list of normalized values of other procedures from the same hospital
    hospital_procedures_info[hospital_info[0]] = temp_list #setting the list back to the dictionary

    normalized_list.append(normalized_procedure_value) #creates a list allowing the viewing of the distribution of the normalized values for a procedure
```

Here, now that we have the standard deviation, we can normalize the list of median prices for a single procedure. This gives us a ranking, of sort, for every hospital's price offering for a certain procedure. If it is above 0, it is above the average median price offering across all hospitals.

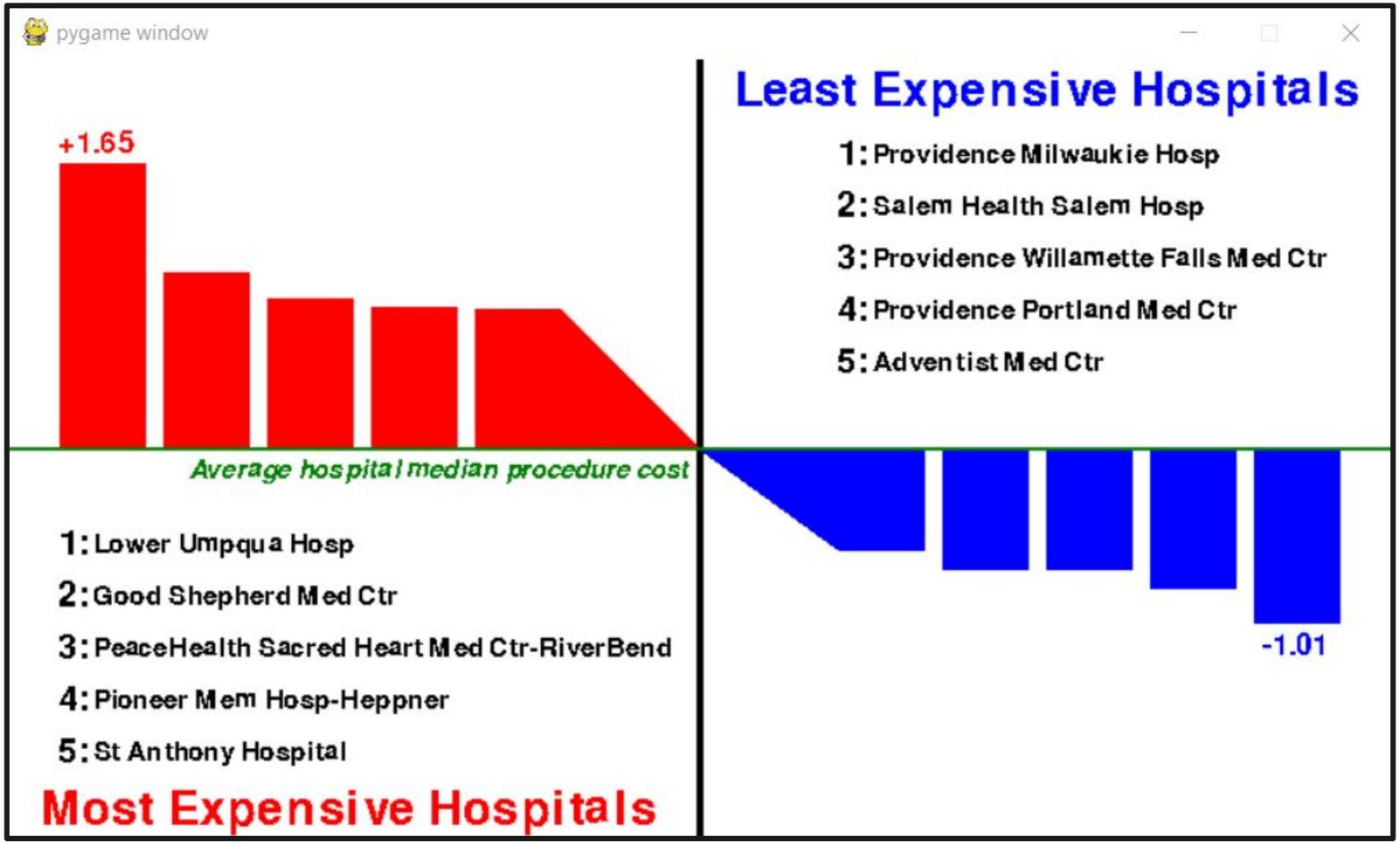
Repeating the Process

```
#purpose: repeat the normalization process for each procedure, making use of every line of data
#argument: none
#result: a final list that includes each hospital's mean normalized value of median procedure price
def normalized_values_for_hospitals():

    for procedure in procedure_names:
        next_step_analysis(procedure) #create full lists of normalized procedures for every hospital in one dictionary

    for hospital in hospital_names:
        normalization_sum = 0
        normalization_list = hospital_procedures_info[hospital]
        for normalization in normalization_list:
            normalization_sum+= normalization
        normalization_mean = normalization_sum/len(normalization_list) #find the mean of all procedure normalizations for a hospital
        final_hospital_data.append((hospital, normalization_mean)) #create a final list of every hospital and its average procedure normalization value
        final_hospital_data.sort(reverse=True, key=lambda x:x[1]) #sort the list based on the values from largest to smallest, worst to best
```

Now, all we have to do is repeat this process for every single procedure and we can obtain a “ranking” of how each hospital’s price offering for a procedure compares to its counterparts. If we take the average of these values for every hospital, we end up with one final value that ranks all of a hospital’s prices in one single value.

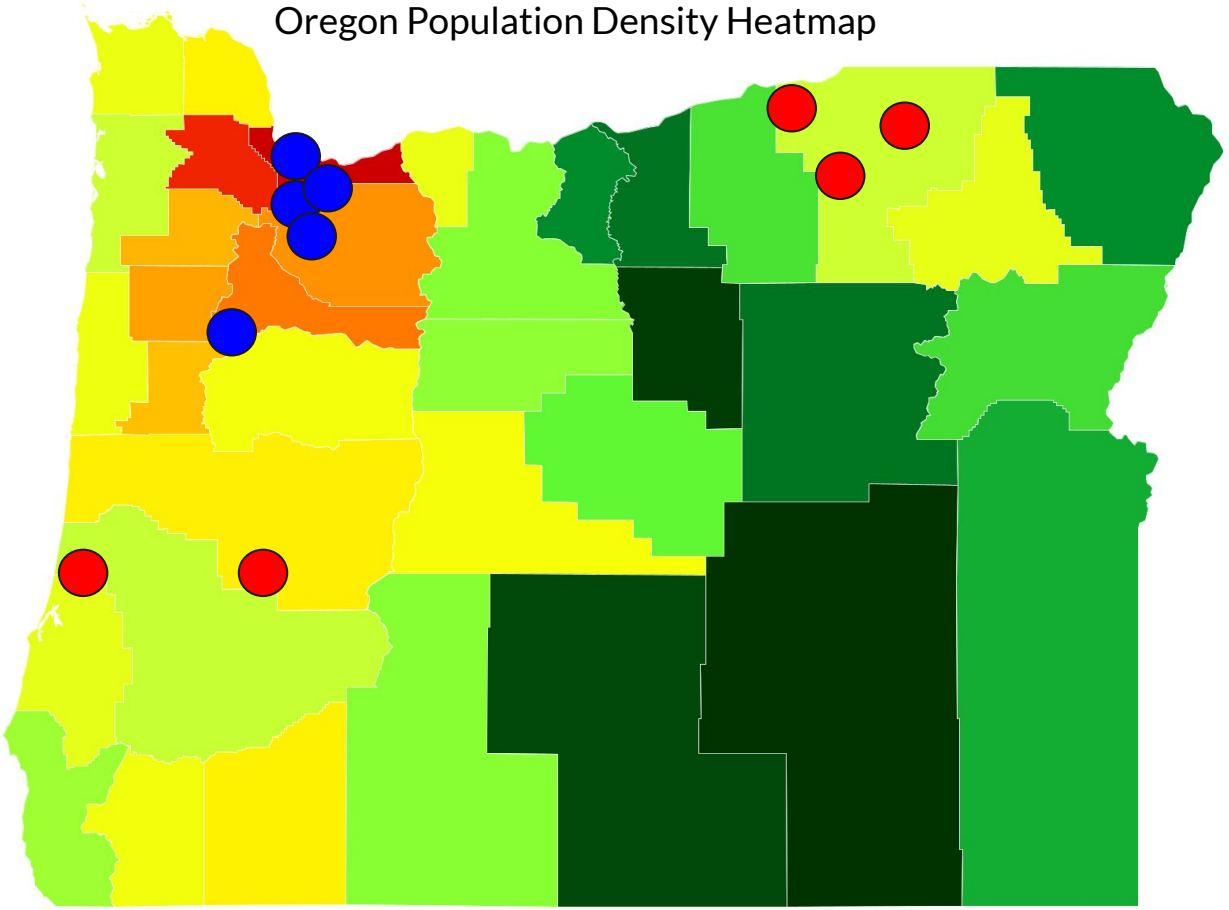


Using PyGame graphics, we can visualize some of the data.

These are the five greatest and five lowest outliers.

Now we need to know why this is the case.

Oregon Population Density Heatmap



1) Providence Milwaukie Hospital
2) Salem Health Hospital
3) Providence Willamette Falls Medical Center
4) Providence Portland Medical Center
5) Adventist Medical Center

1) Lower Umpqua Hospital
2) Good Shepherd Medical Center
3) PeaceHealth Sacred Heart Medical Center-Riverbend
4) Pioneer Memorial Hospital-Heppner
5) St. Anthony Hospital

In Conclusion...

- This data supports the existing theory that lack of localized competition creates an influx in price.
    - In less populated areas where there are less hospitals, prices tend to be on the higher end.
    - Conversely, in more populated areas where more hospitals are present, prices tend to be more affordable.
- This highlights the importance of promoting a populated supply side market to limit distribution in prices across the state.
- Through policy efforts, if the rate of supplier consolidation can be limited, surely this will reflect in a slower increase in medical procedure prices.