CSC550 Capstone Project

Big Data Analysis for
Inpatient Prospective Payment System(IPPS)

Changching Chi (90217)
Chi Yang(91070)
Shweta Sahu (90525)

Objective

Inpatient Prospective Payment System records type of diagnosis related group(DRG) for inpatient incident. Taking the advantage of big data analysis, we would like to the analyze Inpatient Prospective Payment System (IPPS) and understand certain the trends from given dataset. We want to understand the highest frequency of Diagnosis-Related Group (DRG) in California, the highest average covered charges, and the average medical cost by state in the dataset. This dataset provided by data.cms.gov for FY2011.

https://data.cms.gov/Medicare-Inpatient/Inpatient-Prospective-Payment-System-IPPS-Provider/97k6-zzx3

Big Data Analysis

What’s the highest frequency of DRG in California?

871- SEPTICEMIA OR SEVERE SEPSIS is the most common one with 268 times

_1	_2
871 - SEPTICEMIA OR SEVERE SEPSIS W/O MV 96+ HOURS W MCC	268
292 - HEART FAILURE & SHOCK W CC	251
690 - KIDNEY & URINARY TRACT INFECTIONS W/O MCC	250
194 - SIMPLE PNEUMONIA & PLEURISY W CC	243
872 - SEPTICEMIA OR SEVERE SEPSIS W/O MV 96+ HOURS W/O MCC	236
291 - HEART FAILURE & SHOCK W MCC	234
603 - CELLULITIS W/O MCC	232
470 - MAJOR JOINT REPLACEMENT OR REATTACHMENT OF LOWER EXTREMITY W/O MCC	229
378 - G.I. HEMORRHAGE W CC	216
190 - CHRONIC OBSTRUCTIVE PULMONARY DISEASE W MCC	212
683 - RENAL FAILURE W CC	211
193 - SIMPLE PNEUMONIA & PLEURISY W MCC	210
191 - CHRONIC OBSTRUCTIVE PULMONARY DISEASE W CC	204
065 - INTRACRANIAL HEMORRHAGE OR CEREBRAL INFARCTION W CC	200
313 - CHEST PAIN	197
309 - CARDIAC ARRHYTHMIA & CONDUCTION DISORDERS W CC	196
682 - RENAL FAILURE W MCC	188
312 - SYNCOPE & COLLAPSE	185
812 - RED BLOOD CELL DISORDERS W/O MCC	183
293 - HEART FAILURE & SHOCK W/O CC/MCC	180

only showing top 20 rows

Big Data Analysis

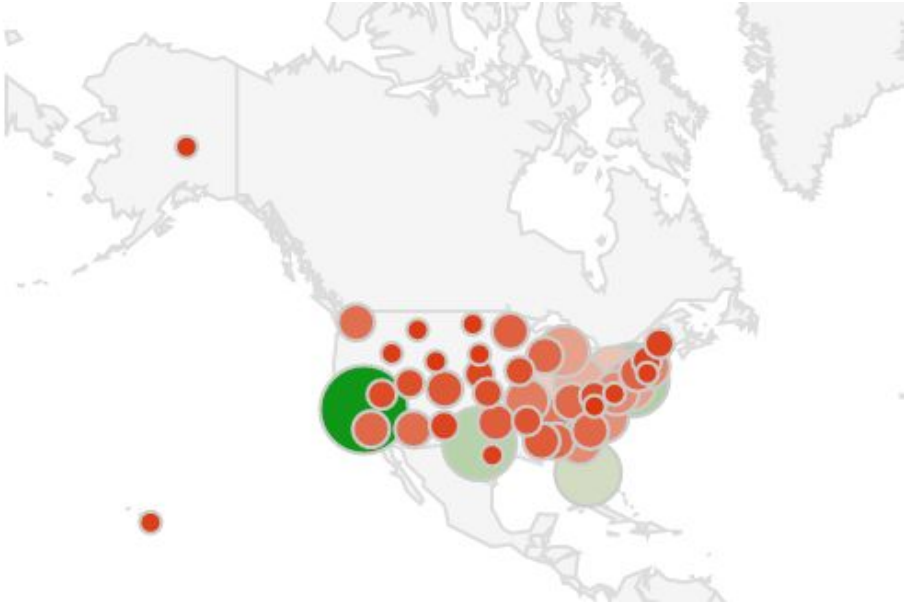
Which state has the highest average covered charges ?

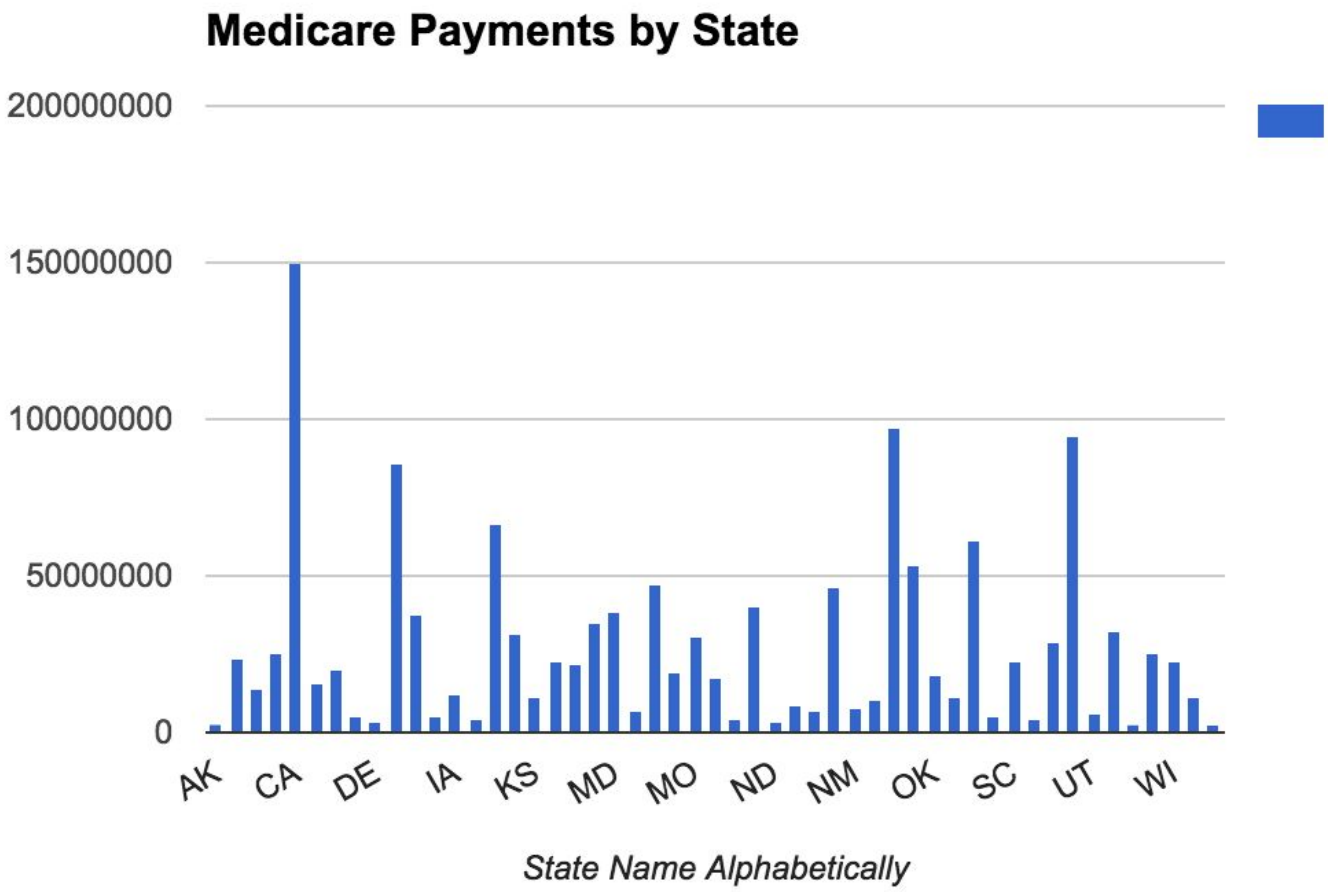
California (CA)

CA : $881932566.42
FL : $513311085.66
TX : $492121014.54
NJ : $319122561.96
PA : $309303421.99

Big Data Analysis

What’s the average medical cost by state?




Medicare Payments by State
200000000
150000000
100000000
50000000
0
AK
CA
DE
IA
KS
MD
MO
ND
NM
OK
SC
UT
WI
State Name Alphabetically

Source Code and Result - Part 1

```
 def parse_state(line):
    line = line.strip().split(",")  # strip out carriage return
    key_in = line[5]  # key is first item in list
    value_in = line[0]  # value is 2nd item
    return (key_in, value_in)
def count_DRG(line):
    #Row(_1=u'CA', _2=u'948 - SIGNS & SYMPTOMS W/O MCC')
    key_in = line[1]  # key is first item in list
    return (key_in, 1)

dataset_raw = sc.textFile("input/capstone.csv")
dataset_1 = dataset_raw.map(parse_state)
df_dataset = dataset_1.toDF();
df_ca = df_dataset.where(df_dataset['_1']=='CA')
dataset_ca = df_ca.rdd;

dataset_ca = dataset_ca.map(count_DRG)
dataset_caMaped = dataset_ca.map(count_DRG)
countsByDRG = dataset_caMaped.reduceByKey(lambda a,b: a+b)
```

```
countsByDRG.collect()
df_DRGcounts = countsByDRG.toDF()
df_DRGcountsSorted = df_DRGcounts.sort("_2",
ascending=False)
df_DRGcountsSorted.show(20,False)
```

Source Code and Result - Part 1

In this questions, I first ran into filter issue, spark operation was not straightforward to me at the point of assignment. In order to filter the dataset by state (California), I had to convert the textFile (RDD) to DataFrame in Spark. In this fashion, i can leverage the filter operation in dataframe. The result shows an interesting finding that during FY2011, the highest frequency of Diagnosis Related Group was Septicemia or severe sepsis.

Source Code and Result - Part 2

```
def parser(line): #parser to extract key, value pair
    line = line.strip().split(",")
    key = line[5] # key is state, col 6
    value = float(line[9]) #avg cov chgs, forcing to float
    return (key,value)

fileIn = sc.textFile("proj/input.csv")
parsed_data = fileIn.map(parser)
parsed_data.take(2)
```

[(u'AL', 32963.07), (u'AL', 15131.85)]

```
summedByStateAvgCovCharges =
parsed_data.reduceByKey(lambda x,y : x + y) #Summing
values for given key using reduceByKey function

summedByStateAvgCovCharges.take(2)
```

[(u'WA', 96436142.259999648), (u'DE', 10666249.659999996)]

```
top5States = summedByStateAvgCovCharges.takeOrdered(5,
lambda(k,v): -v) #Creating an ordered list by sorting with
maximum value

for p in top5States: print " {0} : ${1}".format(p[0],p[1])
```

CA : $881932566.42
FL : $513311085.66
TX : $492121014.54
NJ : $319122561.96
PA : $309303421.99

Source Code and Result - Part 2

The inpatient patient system data file from cms.gov was exported as ‘csv’ format into HDFS system. Later it was found that around 32237 street names had commas within its name. This problem was caught in reduceByKey method much later during summing up values. The file had to be corrected before it can be used for data analysis.

Source Code and Result - Part 3

mapper.py

```
#!/usr/bin/python

import sys

for line in sys.stdin:
    data = line.strip().split("\t")
    if len(data) == 12:
        DRGdef, id, name, address, city, state, zip, ref, discharge, avgCoveredCharge, avgTotalPayment, avgMediPayments = data
        print "{0}\t{1}\t{2}\t{3}\t{4}\t{5}".format(state, DRGdef, discharge, avgCoveredCharge, avgTotalPayment, avgMediPayments)
```

reducer.py

```
#!/usr/bin/python
import sys
dischargeTotal = 0
coveredChargeTotal = 0
totalPaymentTotal = 0
mediPaymentsTotal = 0
oldKey = None
for line in sys.stdin:
    data_mapped = line.strip().split("\t")
    if len(data_mapped) != 6:
        # Something has gone wrong. Skip this line.
        continue
    thisKey, DRG, discharges, coveredCharge, totalPayment, mediPayments = data_mapped
    if oldKey and oldKey != thisKey:
        print oldKey, "\t", dischargeTotal, "\t", coveredChargeTotal, "\t", totalPaymentTotal, "\t", mediPaymentsTotal
        oldKey = thisKey;
        dischargeTotal = 0
```

(con't)

```
        coveredChargeTotal = 0
        totalPaymentTotal = 0
        mediPaymentsTotal = 0


    oldKey = thisKey
    dischargeTotal += float(discharges)
    coveredChargeTotal += float(coveredCharge)
    totalPaymentTotal += float(totalPayment)
    mediPaymentsTotal += float(mediPayments)

if oldKey != None:
    print oldKey, "\t", dischargeTotal, "\t", coveredChargeTotal, "\t",
totalPaymentTotal, "\t", mediPaymentsTotal
```

Source Code and Result - Part 3

AK	6142.0	9320559.71	3366222.49	2993521.94	MN	95666.0	62818102.82	22403429.64	19410472.14
AL	142704.0	113835339.64	27510523.86	23329455.88	MO	178826.0	126578384.36	35413278.33	30804748.99
AR	86769.0	54102745.75	16575787.28	14303062.96	MS	93223.0	73005612.34	19832287.23	17169263.56
AZ	104604.0	117461379.67	28950559.93	25162119.85	MT	15705.0	11471027.71	4681918.2	4038430.56
CA	474979.0	881932566.42	164993988.92	150162602.24	NC	257312.0	126735539.85	45819845.42	40321193.15
CO	61320.0	77669807.25	17960075.69	15405260.33	ND	16425.0	9130764.82	4147263.25	3693364.38
CT	96258.0	62981322.74	22855921.3	20320336.41	NE	39799.0	33704086.35	9910246.84	8488170.14
DC	20307.0	18533898.61	6005089.59	5457129.08	NH	28419.0	22269574.12	7645391.68	6686469.14
DE	26753.0	10666249.66	4081868.53	3530111.27	NJ	251207.0	319122561.96	51536799.21	46266572.71
FL	536859.0	513311085.66	98465078.31	85530724.84	NM	26925.0	27520459.76	8821394.28	7611202.19
GA	191242.0	154489562.36	44343344.17	38092545.14	NV	42600.0	73378632.73	12370645.07	10514618.6
HI	11712.0	14221238.65	5646876.87	4847623.97	NY	435557.0	288516721.91	108259026.05	97477118.66
IA	68784.0	41908598.7	14413999.34	12394840.07	OH	293472.0	198749660.45	61762591.09	53719168.39
ID	18295.0	14086616.42	5414776.23	4662549.61	OK	97292.0	74560689.67	21051175.41	18163419.26
IL	361603.0	285213170.69	77434457.25	66319200.66	OR	39546.0	35579755.32	13556614.53	11736802.69
IN	182573.0	119896475.44	37300911.59	31857542.34	PA	314936.0	309303421.99	71016737.27	61801318.62
KS	61800.0	51728455.5	13850070.38	11833965.5	RI	22289.0	17606308.26	6179625.31	5478948.2
KY	152572.0	79187373.35	26731563.38	23201100.6	SC	121393.0	102100522.02	26000001.9	22423915.85
LA	109876.0	100149423.44	26149231.62	22362581.9	SD	20109.0	14390455.89	4928860.37	4199604.77
MA	193680.0	78891653.79	39495689.06	35506685.64	TN	189065.0	122038755.89	33985667.16	28952967.42
MD	173011.0	44548086.62	41987795.72	38228805.69	TX	479939.0	492121014.54	109670573.65	94561190.98
ME	33114.0	18110722.32	7707835.54	6740364.72	UT	23052.0	18367934.63	7136931.98	5731288.2
MI	295552.0	130729295.63	52859204.18	46940232.88	VA	193399.0	126589706.11	38501742.43	32658285.23

MS	93223.0	73005612.34	19832287.23	17169263.56
MT	15705.0	11471027.71	4681918.2	4038430.56
NC	257312.0	126735539.85	45819845.42	40321193.15
ND	16425.0	9130764.82	4147263.25	3693364.38
NE	39799.0	33704086.35	9910246.84	8488170.14
NH	28419.0	22269574.12	7645391.68	6686469.14
NJ	251207.0	319122561.96	51536799.21	46266572.71
NM	26925.0	27520459.76	8821394.28	7611202.19
NV	42600.0	73378632.73	12370645.07	10514618.6
NY	435557.0	288516721.91	108259026.05	97477118.66
OH	293472.0	198749660.45	61762591.09	53719168.39
OK	97292.0	74560689.67	21051175.41	18163419.26
OR	39546.0	35579755.32	13556614.53	11736802.69
PA	314936.0	309303421.99	71016737.27	61801318.62
RI	22289.0	17606308.26	6179625.31	5478948.2
SC	121393.0	102100522.02	26000001.9	22423915.85
SD	20109.0	14390455.89	4928860.37	4199604.77
TN	189065.0	122038755.89	33985667.16	28952967.42
TX	479939.0	492121014.54	109670573.65	94561190.98
UT	23052.0	18367934.63	7136931.98	5731288.2
VA	193399.0	126589706.11	38501742.43	32658285.23
VT	10071.0	5420238.75	3176902.21	2847681.89
WA	107011.0	96436142.26	29288875.29	25214542.8
WI	100068.0	74107187.99	26273179.72	22679362.48
WV	64968.0	30495307.22	12661915.11	10965454.7
WY	6535.0	7089047.86	2815426.02	2356229.83

VT	10071.0	5420238.75	3176902.21	2847681.89
WA	107011.0	96436142.26	29288875.29	25214542.8
WI	100068.0	74107187.99	26273179.72	22679362.48
WV	64968.0	30495307.22	12661915.11	10965454.7
WY	6535.0	7089047.86	2815426.02	2356229.83