

ALLISON BLACK
IE MBD 2020
SPARK



PHYSICIAN COMPARE 2017

AN ANALYSIS ON MEDICARE PROVIDERS IN THE USA

WHAT IS MEDICARE?

- 65 years and older (mostly)
- Hospital Insurance
- Medical Insurance
- Prescription Drug Coverage



Physician_Compare_2017_Group_Public_Reportin_-_Measures

Hospital	Group PAC ID	State	ACO PC ID 1	Measure Code	Measure Title	Measure Performance Rate	Denominator Count	Star Value	Five Star Benchmark	Reported on PC Live Site
ALASKA CENTER FOR EAR NOSE AND THROAT INC	7012901952	AK		MIPS_GRP_093_overall	Acute Otitis Externa (AOE): Systemic Antimicrobial Therapy - Avoidance of Inappropriate Use	100	24			N
ALASKA CENTER FOR EAR NOSE AND THROAT INC	7012901952	AK		MIPS_GRP_091_overall	Acute Otitis Externa (AOE): Topical Therapy	87	23			N
ALASKA CENTER FOR EAR NOSE AND THROAT INC	7012901952	AK		MIPS_GRP_331_overall	Adult Sinusitis: Antibiotic Prescribed for Acute Sinusitis (Overuse)	63	41			N
ALASKA CENTER FOR EAR NOSE AND THROAT INC	7012901952	AK		MIPS_GRP_332_overall	Adult Sinusitis: Appropriate Choice of Antibiotic: Amoxicillin With or Without Clavulanate Presc	100	22			N
ALASKA CENTER FOR EAR NOSE AND THROAT INC	7012901952	AK		ACI_GRP_TRANS_PPHI	Security Risk Analysis					N
ALASKA CENTER FOR EAR NOSE AND THROAT INC	7012901952	AK		ACI_GRP_TRANS_PHCD	Specialized Registry Reporting					N
ALASKA CENTER FOR EAR NOSE AND THROAT INC	7012901952	AK		MIPS_GRP_402_overall	Tobacco Use and Help with Quitting Among Adolescents	95	65			N
ALASKA EMERGENCY MEDICINE ASSOCIATES	5193639912	AK		MIPS_GRP_116_overall	Avoidance of Antibiotic Treatment in Adults With Acute Bronchitis	53	602			N
ALASKA EMERGENCY MEDICINE ASSOCIATES	5193639912	AK		QCDR_GRP_ACEP21_overall	Coagulation Studies in Patients Presenting with Chest Pain with No Coagulopathy or Bleeding	2	2218			Y
ALASKA EMERGENCY MEDICINE ASSOCIATES	5193639912	AK		MIPS_GRP_076_overall	Prevention of Central Venous Catheter (CVC) - Related Bloodstream Infections	27	396			N
ALASKA HEART INSTITUTE	2668371469	AK		MIPS_GRP_326_overall	Atrial Fibrillation and Atrial Flutter: Chronic Anticoagulation Therapy	85	3016			N
ALASKA HEART INSTITUTE	2668371469	AK		MIPS_GRP_047_overall	Care Plan	61	8776	3	100	Y
ALASKA HEART INSTITUTE	2668371469	AK		MIPS_GRP_006_overall	Coronary Artery Disease (CAD): Antiplatelet Therapy	95	7184	4	100	Y
ALASKA HEART INSTITUTE	2668371469	AK		ACI_GRP_TRANS_PPHI	Security Risk Analysis					N
ALASKA NATIVE TRIBAL HEALTH CONSORTIUM	6709780265	AK		MIPS_GRP_006_overall	Coronary Artery Disease (CAD): Antiplatelet Therapy	100	1259	5	100	Y
ALASKA NATIVE TRIBAL HEALTH CONSORTIUM	6709780265	AK		MIPS_GRP_130_overall	Documentation of Current Medications in the Medical Record	45	86444			N
ALASKA NATIVE TRIBAL HEALTH CONSORTIUM	6709780265	AK		ACI_GRP_TRANS_EP_1	e-Prescribing	99	80505			N
ALASKA NATIVE TRIBAL HEALTH CONSORTIUM	6709780265	AK		ACI_GRP_TRANS_PHCD	Immunization Registry Reporting					N
ALASKA NATIVE TRIBAL HEALTH CONSORTIUM	6709780265	AK		MIPS_GRP_204_overall	Ischemic Vascular Disease (IVD): Use of Aspirin or Another Antiplatelet	100	1185			N
ALASKA NATIVE TRIBAL HEALTH CONSORTIUM	6709780265	AK		ACI_GRP_TRANS_MR_1	Medication Reconciliation	81	52391			N
ALASKA NATIVE TRIBAL HEALTH CONSORTIUM	6709780265	AK		ACI_GRP_TRANS_PSE_1	Patient-Specific Education	50	23568			N
ALASKA NATIVE TRIBAL HEALTH CONSORTIUM	6709780265	AK		ACI_GRP_TRANS_PEA_1	Provide Patient Access					
ALASKA NATIVE TRIBAL HEALTH CONSORTIUM	6709780265	AK		ACI_GRP_TRANS_SM_1	Secure Messaging					
ALASKA NATIVE TRIBAL HEALTH CONSORTIUM	6709780265	AK		ACI_GRP_TRANS_PPHI	Security Risk Analysis					
ALASKA NATIVE TRIBAL HEALTH CONSORTIUM	6709780265	AK		ACI_GRP_TRANS_PHCD	Specialized Registry Reporting					
ALASKA NATIVE TRIBAL HEALTH CONSORTIUM	6709780265	AK		ACI_GRP_TRANS_PEA_1	View, Download, or Transmit (VDT)					
ALASKA NEUROLOGY CENTER LLC	2163512328	AK		MIPS_GRP_286_overall	Dementia: Counseling Regarding Safety Concerns					
ALASKA NEUROLOGY CENTER LLC	2163512328	AK		MIPS_GRP_283_overall	Dementia: Neuropsychiatric Symptom Assessment					
ALASKA NEUROLOGY CENTER LLC	2163512328	AK		MIPS_GRP_131_overall	Pain Assessment and Follow-Up					
ALASKA NEUROLOGY CENTER LLC	2163512328	AK		ACI_GRP_TRANS_PPHI	Security Risk Analysis					
ALASKA NEUROLOGY CENTER LLC	2163512328	AK		ACI_GRP_TRANS_PHCD	Specialized Registry Reporting					
ALASKA RADIOLOGY ASSOCIATES INC	648182725	AK		MIPS_GRP_147_overall	Nuclear Medicine: Correlation with Existing Imaging Studies for All Pa					
ALASKA RADIOLOGY ASSOCIATES INC	648182725	AK		MIPS_GRP_076_overall	Prevention of Central Venous Catheter (CVC) - Related Bloodstream					

All QCDR measures that met the statistical reporting standards are included in the downloadable file regardless of where they were publicly reported. QCDRs could report these measures on their own websites or they could elect to have the measures publicly reported on Physician Compare.

NOTE: Inverse measures that were published on the public-facing profile pages on Physician Compare were reverse-scored, so that a higher score is better to ensure accurate website user interpretation. The original score is what is included in the downloadable file for these measures.

Variable Name	Variable Label	Description	Length	Values
Org_nm	Organization legal name or 'doing business as' name	Name of the group, as it appears on Physician Compare	70	string
Org_PAC_ID	Group PAC ID	Unique Group ID assigned by PECOS to the group	10	digits
ST	State	State with all or a majority of the group's locations	2	string
ACO_ID_1	ACO PC ID 1	ACO ID used on Physician Compare	6	string
ACO_nm_1	ACO Name 1	Name of the ACO, as it appears on Physician Compare	70	string
ACO_ID_2	ACO PC ID 2	ACO ID used on Physician Compare	6	string
ACO_nm_2	ACO Name 2	Name of the ACO, as it appears on Physician Compare	70	string
PQRS	Participating in PQRS	This represents satisfactory participation in the Physician Quality Reporting System (PQRS). This is a voluntary Medicare program encouraging groups to report information on the recommended care they provide to patients. Medicare uses this information to give clinicians and groups performance scores. Those who participated in this program in 2016 may have performance scores available on their Physician Compare profile page.	1	Y/blank

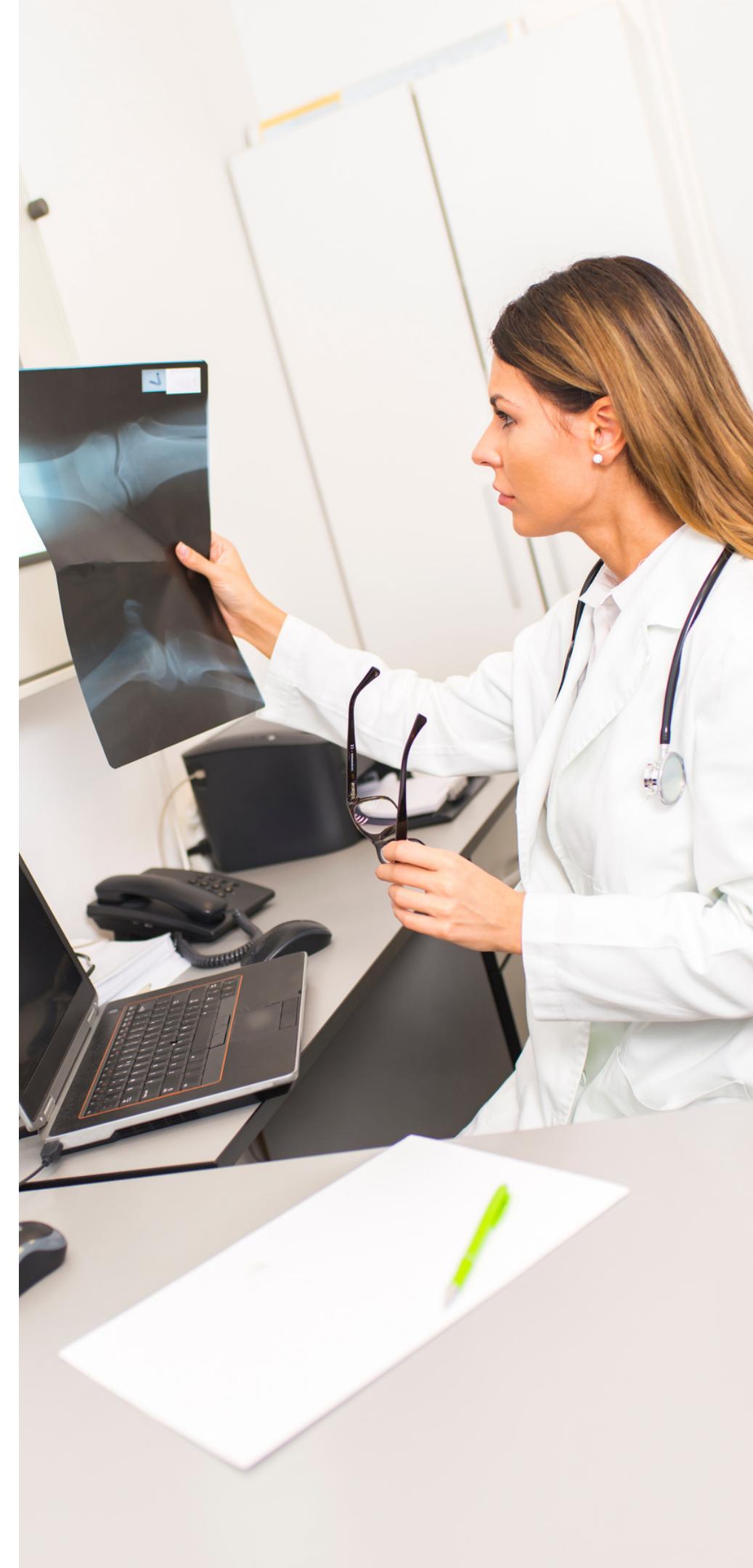
Dataset

Dictionary

Goal of Analysis:

Answer business questions and gain insights!

- Hospital Measure Performance Rate by Grade
Based on patient surveys
- Statistics by type of Measure
E-Prescribing, Secure Messaging, Pain Assessment and Follow-Up
- Best and worst hospitals
States where the worst hospitals are located



What is Spark?



Spark is a unified, in-memory, parallel computing framework for big data.

- Unified: We can use Spark for multiple use cases (batch processing, machine learning, advanced analytics, etc. Spark provides the user with built-in APIs.
- In-memory: Spark leverages the server's memory to speed up computation.
- Parallel: Spark works on big data clusters and can deal with any amount of data.



Pysician Compare Analysis

First steps

1. PySpark environment setup

```
import findspark
findspark.init()

from pyspark.context import SparkContext
from pyspark.sql import SparkSession

sc = SparkContext.getOrCreate()
spark = SparkSession(sc)

# Refer to https://spark.apache.org/docs/latest/api/python/index.html for PySpark documentation
```

2. Data source and Spark data abstraction (DataFrame) setup

```
healthDF = spark.read \
    .option("inferSchema", "true") \
    .option("header", "true") \
    .csv("Physician_Compare_2017_Group_Public_Reportin_--_Measures.csv")
healthDF.show()

Row(Hospital='ALASKA CENTER FOR EAR NOSE AND THROAT INC', Group PAC ID=7012901952, State='AK', ACO PC ID=1, Measure Code='MIPS_GRP_093_overall', Measure Title='Acute Otitis Extern a (AOE): Systemic Antimicrobial Therapy - Avoidance of Inappropriate Use', Measure Performance Rate=100, Denominator Count=24, Star Value=None, Five Star Benchmark=None, Reported on PC Live Site='N')
```

3. Data set metadata analysis

A. Display schema and size of the DataFrame

```
from IPython.display import display, Markdown
healthDF.printSchema()
display(Markdown("This DataFrame has **%d rows**." % healthDF.count()))

root
|-- Hospital: string (nullable = true)
|-- Group PAC ID: long (nullable = true)
|-- State: string (nullable = true)
|-- ACO PC ID: string (nullable = true)
|-- Measure Code: string (nullable = true)
|-- Measure Title: string (nullable = true)
|-- Measure Performance Rate: integer (nullable = true)
|-- Denominator Count: integer (nullable = true)
|-- Star Value: integer (nullable = true)
|-- Five Star Benchmark: integer (nullable = true)
|-- Reported on PC Live Site: string (nullable = true)
```

This DataFrame has 73321 rows.

B. Get two random samples from the data set

```
healthDF.cache() # optimization to make the processing faster
healthDF.sample(False, 0.8).take(2)

[Row(Hospital='MAT-SU VALLEY MEDICAL CENTER LLC', Group PAC ID=5698686475, State=None, Measure Code='MIPS_GRP_093_overall', Measure Title='Acute Otitis Extern a (AOE): Systemic Antimicrobial Therapy - Avoidance of Inappropriate Use', Measure Performance Rate=101, Denominator Count=97, Star Value=None, Five Star Benchmark=None, Reported on PC Live Site='N'),
 Row(Hospital='MAMANA VALLEY CLINIC LLC', Group PAC ID=5698686471, State='AK', Measure Code='MIPS_GRP_093_overall', Measure Title='Acute Otitis Extern a (AOE): Systemic Antimicrobial Therapy - Avoidance of Inappropriate Use', Measure Performance Rate=101, Denominator Count=97, Star Value=None, Five Star Benchmark=None, Reported on PC Live Site='N')]
```

C. Data entities, metrics and dimensions

I've identified the following elements:

- **Entities:** Hospital (main one which is measured - facts), State (dimension), Measure Title (dimension)
- **Metrics:** Measure Performance Rate, Star Value, Five Star Benchmark
- **Dimensions:** Location, measure code, collection type

D. Column categorization

The following could be a potential column categorization:

- **Performance related columns:** Measure title, Measure Performance Rating, Star Value, Five Star Benchmark
- **Location related columns:** Hospital, State, Group PAC ID

4. Columns groups basic profiling to better understand the data set

A. Performance related columns basic profiling

```
from IPython.display import display, Markdown
from pyspark.sql.functions import when, count, col, countDistinct, desc, first, lit

print("Summary of columns Measure Performance Rate, Star Value, Five Star Benchmark:")
healthDF.select("Measure Performance Rate", "Star Value", "Five Star Benchmark").summary().show()

print("Checking for nulls on columns Measure Performance Rate, Star Value, Five Star Benchmark:")
healthDF.select([count(when(col(c).isNull(), c)).alias(c) for c in ["Measure Performance Rate", "Star Value", "Five Star Benchmark"]]).show()

print("Checking amount of distinct values in columns Measure Performance Rate, Star Value, Five Star Benchmark:")
healthDF.select([countDistinct(c).alias(c) for c in ["Measure Performance Rate", "Star Value", "Five Star Benchmark"]]).show()
```

	Measure Performance Rate	Star Value	Five Star Benchmark
count	52828	3196	3196
mean	62.528621185734835	3.5259699624530665	98.15425531914893
stddev	38.124502934221155	1.2387641492890453	4.051891581973628
min	0	1	84
25%	22	3	100
50%	80	4	100
75%	98	4	100
max	100	5	100

```
Summary of columns Measure Performance Rate, Star Value, Five Star Benchmark:
+-----+-----+-----+
|summary|Measure Performance Rate|Star Value|Five Star Benchmark|
+-----+-----+-----+
| count | 52828 | 3196 | 3196 |
| mean | 62.528621185734835 | 3.5259699624530665 | 98.15425531914893 |
| stddev | 38.124502934221155 | 1.2387641492890453 | 4.051891581973628 |
| min | 0 | 1 | 84 |
| 25% | 22 | 3 | 100 |
| 50% | 80 | 4 | 100 |
| 75% | 98 | 4 | 100 |
| max | 100 | 5 | 100 |
+-----+-----+-----+
```

Checking for nulls on columns Measure Performance Rate, Star Value, Five Star Benchmark:

	Measure Performance Rate	Star Value	Five Star Benchmark
	20493	70125	70125

Checking amount of distinct values in columns Measure Performance Rate, Star Value, Five Star Benchmark:

	Measure Performance Rate	Star Value	Five Star Benchmark
	101	5	5



Physician Compare Analysis

First steps

B. Location related columns basic profiling

```
from IPython.display import display, Markdown
from pyspark.sql.functions import when, count, col, countDistinct, desc, first

print ("Summary of columns Hospital, Group PAC ID, State, Reported on PC Live Site:")
healthDF.select("Hospital", "Group PAC ID", "State", "Reported on PC Live Site").summary().show()

print("Checking for nulls on columns Hospital, Group PAC ID, State, Reported on PC Live Site:")
healthDF.select([count(when(col(c).isNull(), c)).alias(c) for c in ["Hospital", "Group PAC ID", "State", "Reported on PC Live Site"]]).show()

print("Checking amount of distinct values in columns Hospital, Group PAC ID, State, Reported on PC Live Site:")
healthDF.select([countDistinct(c).alias(c) for c in ["Hospital", "Group PAC ID", "State", "Reported on PC Live Site"]]).show()

print ("Most and least frequent occurrences for Hospital, Group PAC ID, State, Reported on PC Live Site:")
HospitalDF = healthDF.groupBy("Hospital").agg(count(lit(1)).alias("Total"))
Group_PAC_DF = healthDF.groupBy("Group PAC ID").agg(count(lit(1)).alias("Total"))
StateDF = healthDF.groupBy("State").agg(count(lit(1)).alias("Total"))
Reported_Live_DF = healthDF.groupBy("Reported on PC Live Site").agg(count(lit(1)).alias("Total"))

leastFreqHospital = HospitalDF.orderBy(col("Total").asc()).first()
mostFreqHospital = HospitalDF.orderBy(col("Total").desc()).first()
leastFreqGroupPAC = Group_PAC_DF.orderBy(col("Total").asc()).first()
mostFreqGroupPAC = Group_PAC_DF.orderBy(col("Total").desc()).first()
leastFreqState = StateDF.orderBy(col("Total").asc()).first()
mostFreqState = StateDF.orderBy(col("Total").desc()).first()
leastFreqReportedLive = Reported_Live_DF.orderBy(col("Total").asc()).first()
mostFreqReportedLive = Reported_Live_DF.orderBy(col("Total").desc()).first()

display(Markdown("""
| %s | %s | %s | %s |
| --- | --- | --- | --- |
| %s | %s | %s | %s |
""") % ("leastFreqHospital", "mostFreqHospital", "leastFreqGroupPAC", "mostFreqGroupPAC", \
        "%s (%s occurrences)" % (leastFreqHospital["Hospital"], leastFreqHospital["Total"]), \
        "%s (%s occurrences)" % (mostFreqHospital["Hospital"], mostFreqHospital["Total"]), \
        "%s (%d occurrences)" % (leastFreqGroupPAC["Group PAC ID"], leastFreqGroupPAC["Group PAC ID"]), \
        "%s (%d occurrences)" % (mostFreqGroupPAC["Group PAC ID"], mostFreqGroupPAC["Group PAC ID"])))

display(Markdown("""
| %s | %s | %s | %s |
| --- | --- | --- | --- |
| %s | %s | %s | %s |
""") % ("leastFreqState", "mostFreqState", "leastFreqReportedLive", "mostFreqDest", \
        "%s (%d occurrences)" % (leastFreqState["State"], leastFreqState["Total"]), \
        "%s (%d occurrences)" % (mostFreqState["State"], mostFreqState["Total"]), \
        "%s (%d occurrences)" % (leastFreqReportedLive["Reported on PC Live Site"], leastFreqReportedLive["Reported on PC Live Site"]), \
        "%s (%d occurrences)" % (mostFreqReportedLive["Reported on PC Live Site"], mostFreqReportedLive["Reported on PC Live Site"]))))

Summary of columns Hospital, Group PAC ID, State, Reported on PC Live Site:
+-----+-----+-----+-----+
|summary| Hospital| Group PAC ID|State|Reported on PC Live Site|
+-----+-----+-----+-----+
| count | 73321 | 73321 | 73321 | 73321 |
| mean | null | 4.983921843809727E9 | null | null |
| stddev | null | 2.8811260416445284E9 | null | null |
| min | 1450 ASSOCIATES, LLC | 42101099 | AK | N |
| 25% | null | 2466545041 | null | null |
| 50% | null | 4981731312 | null | null |
| 75% | null | 7416960265 | null | null |
```

Checking for nulls on columns Hospital, Group PAC ID, State, Reported on PC Live Site:
+-----+-----+-----+
|Hospital|Group PAC ID|State|Reported on PC Live Site|
+-----+-----+-----+
| 0 | 0 | 0 | 0 |
+-----+-----+-----+

Checking amount of distinct values in columns Hospital, Group PAC ID, State, Reported on PC Live Site:
+-----+-----+-----+
|Hospital|Group PAC ID|State|Reported on PC Live Site|
+-----+-----+-----+
| 12944 | 13082 | 54 | 2 |
+-----+-----+-----+

Most and least frequent occurrences for Hospital, Group PAC ID, State, Reported on PC Live Site:

leastFreqHospital	mostFreqHospital	leastFreqGroupPAC	mostFreqGroupPAC
ST JOSEPH HOSPITALIST MEDICAL GROUP INC (1 occurrences)	REGENTS OF THE UNIVERSITY OF CALIFORNIA (160 occurrences)	3476443524 (3476443524 occurrences)	9436062296 (9436062296 occurrences)

leastFreqState mostFreqState leastFreqReportedLive mostFreqDest
GU (12 occurrences) FL (5665 occurrences) Y (3552 occurrences) N (N occurrences)

```
# Drop nulls
healthDF = healthDF.dropna(subset=["Measure Performance Rate"])

print("Checking for nulls on columns Measure Performance Rate:")
healthDF.select([count(when(col(c).isNull(), c)).alias(c) for c in ["Measure Performance Rate"]]).show()
```

Checking for nulls on columns Measure Performance Rate:
+-----+
|Measure Performance Rate|
+-----+
| 0 |
+-----+



Pysician Compare Analysis

Business Question #1: Ratio of Measure Performance Rating by Grade

```
from pyspark.sql.functions import count, round

# Grade is going to be categorized as follows:
#
#   "excellent"      =(90,100]
#   "good"           =(80,89]
#   "average"        =(70,79]
#   "poor"            =(0,69]

# 1. Enrich the DF with grades based on our categorization
totalHealth = healthDF.count()

gradeDF = healthDF\
    .where(col("Measure Performance Rate")!=101)\ 
    .withColumn("Grade", when(col("Measure Performance Rate")>=90,"1. Excellent")\ 
        .when((col("Measure Performance Rate")>79) & (col("Measure Performance Rate")<=89),"2. Good")\ 
        .when((col("Measure Performance Rate")>69) & (col("Measure Performance Rate")<=79),"3. Average")\ 
        .when((col("Measure Performance Rate")>49) & (col("Measure Performance Rate")<=69),"4. Poor")\ 
        .otherwise("5. Unacceptable"))
gradeDF.cache() # optimization to make the processing faster

# 2. Ready to answer to this business question
gradeDF.where(col("Measure Performance Rate")!=101)\ 
    .select("Grade", "Measure Performance Rate")\ 
    .groupBy("Grade")\ 
    .agg(count("Grade").alias("Instances"), \
        (count("Grade")/totalHealth*100).alias("Ratio"))\ 
    .orderBy("Grade")\ 
    .select("Grade", "Instances",round("Ratio",2).alias("RoundedRatio")).show()

+-----+-----+-----+
|     Grade | Instances | RoundedRatio |
+-----+-----+-----+
| 1. Excellent | 21803 | 41.27 |
| 2. Good | 4679 | 8.86 |
| 3. Average | 3107 | 5.88 |
| 4. Poor | 4311 | 8.16 |
| 5. Unacceptable | 18928 | 35.83 |
+-----+-----+-----+
```



Pysician Compare Analysis

Business Question #2: Measure Statistics by Type of Measure

```
measuresDF = healthDF.select("Measure Performance Rate")

display(Markdown(" **E-Prescribing Statistics:**"))
e_rxDF = measuresDF.where(col("Measure Title")==( "e-Prescribing" ))
e_rxDF.summary().show()

display(Markdown(" **Secure Messaging Statistics:**"))
secureDF = measuresDF.where(col("Measure Title")==( "Secure Messaging" ))
secureDF.summary().show()

display(Markdown(" **Pain Assessment and Follow-Up Statistics:**"))
painDF = measuresDF.where(col("Measure Title")==( "Pain Assessment and Follow-Up" ))
painDF.summary().show()
```

E-Prescribing Statistics:

summary	Measure Performance Rate
count	4873
mean	84.49538272111636
stddev	18.271657786469472
min	0
25%	79
50%	91
75%	96
max	100

Secure Messaging Statistics:

summary	Measure Performance Rate
count	4383
mean	18.11088295687885
stddev	21.721727249606282
min	0
25%	1
50%	10
75%	28
max	100

Pain Assessment and Follow-Up Statistics:

summary	Measure Performance Rate
count	425
mean	69.04
stddev	37.373002565417686
min	0
25%	38
50%	90
75%	100
max	100



Pysician Compare Analysis

Business Question #3: Best and Worst Hospitals

```
display(Markdown("**Top 100 Hospitals by average Measure Performance Rate:**"))

hospitalState = healthDF.groupBy("Hospital", "State").mean("Measure Performance Rate")
Top100DF = hospitalState.orderBy(avg("Measure Performance Rate"), ascending = False).limit(100)
Top100DF.show(100)
```

Top 100 Hospitals by average Measure Performance Rate:

Hospital	State	avg(Measure Performance Rate)
SUMMIT PATHOLOGY ...	OH	100.0
AMSON ANESTHETIST...	IL	100.0
PATHOLOGY LAB ASS...	AL	100.0
MID MICHIGAN INTE...	MI	100.0
PATHOLOGY ASSOCIA...	FL	100.0
RADIOLOGY REGIONA...	FL	100.0
WEST RIVER ANESTH...	SD	100.0
JELLICO COMMUNITY...	TN	100.0
SOLANO ANESTHESIA...	CA	100.0
PIKEVILLE MEDICAL...	KY	100.0
SOUTHERN WESTCHES...	NY	100.0
MARIA NINA CONCEP...	FL	100.0
FLOWERS PATHOLOGY...	AL	100.0

State	count
FL	11
CA	8
NY	6
GA	6
IL	6
TX	5
CO	5
WA	4
PA	4
AZ	4
MA	4
WI	4
MO	3
MI	3
MS	3
OH	2
IN	2
NC	2
SC	2
MD	2

```
display(Markdown("**Bottom 100 Hospitals by average Measure Performance Rate:**"))

Bottom100DF = hospitalState.orderBy(avg("Measure Performance Rate"), ascending = True).limit(100)
Bottom100DF.show(100)
```

Bottom 100 Hospitals by average Measure Performance Rate:

Hospital	State	avg(Measure Performance Rate)
MAGNOLIA EXPRESS ...	MS	0.0
PATRIOT URGENT CA...	MA	0.0
GARY L CURSON PA	FL	0.0
PAJARO VALLEY NEU...	CA	0.0
ALLERGY and ASTHM...	OH	0.0
VALLEY ENT, P.C.	PA	0.0
LELWICA CHIROPRACTIC	MN	0.0
SYRACUSE ENT SURG...	NY	0.0
ORTHOPEDIC INSTIT...	CA	0.0
WESTERN INFECTIOU...	CO	0.0
PACIFICA EMERGENC...	CA	0.0
KENT E DOBBINS OD...	KS	0.0
VALLEY PULMONARY ...	TX	0.0
ASCENSION OUR LAD...	WI	0.0
PULMONARY SPECIAL...	IN	0.0
TEXAS SPINE CONSU...	TX	0.0



Pysician Compare Analysis

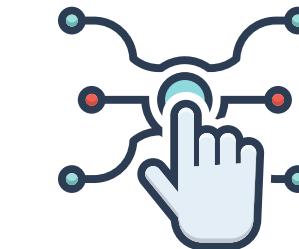
Insights

Pysician Compare Analysis



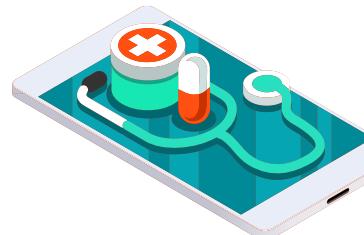
Distribution of Measure Performance Ratings

1/3 are "Unacceptable"



Use of technology

E-Prescribing ratings are high, but secure messaging ratings are low



Pain Assessment and Follow-Up

Mean score suggests room for improvement



Performance by Hospital and State

Florida (high percentage of Medicare patients) is home to 11 hospitals in the bottom 100



Physician Compare 2017: An Analysis

Thank you!

Link to dataset:

<https://data.medicare.gov/data/physician-compare>

