Spring 2019

BUAN 6346: Big Data

Project Report

THE UNIVERSITY
OF TEXAS AT DALLAS

Submitted by

Akanksha Guruprasad - AXG180053

Introduction and problem description:

The dataset consists of complaints logged by consumers about financial products and services to companies for response. Data from those complaints help us understand the financial marketplace and protect consumers.
Other than the Consumer complaint dataset, other datasets that could give useful information are:
(1) Census Data for US population

Census Data can be useful in getting the information about number of complaints logged in each state vs the population or the area of the state.

The list of tasks that can be performed are:
(1)No of Complaints logged each Year.
(2) No of Complaints logged in each Region/State
(3) Does population of a state affect the no of complaints logged.
(4) Which financial company has most no of complaints?
(5) Which product is the most complained about?
(6) Was a Timely response given by the companies against which complaint was logged?
(7) How many complaints are logged in the start of the month/year or at the end of the month/year. Is there a significant difference in both these values?

Also, all these data are segregated according to year. So, a trend can be observed whether the no of complaints every year is increasing or decreasing. Similarly, this trend can also be looked for the most/least complaints lodged against company in each State/Region.
Most of the above problem statements can be answered by performing aggregation queries on a single or merged dataset. But as these datasets consists of huge data, pre-processing of these data will be required as we wish to observe this trend based on year-wise data.
For this report, consumer complaints logged from the 01/01/2011 to 04/21/2019 is used for analysis.

Performed work:

Dataset description:

The main dataset of this project is:

(i) Complaints Dataset

The features of this dataset are:
1. **Company**: Company against whom the complaint is registered.
2. **Company public response**: This field has information about the company's response to a consumer's complaint.
3. **Company response to consumer**: This is how the company responded. For example, "Closed with explanation."
4. **Complaint ID**: The unique identification number for a complaint.
5. **Consumer consent provided**: This field identifies whether the consumer opted in to publish their complaint narrative.
6. **Consumer disputed**: This field identifies whether the consumer disputed the company's response.
7. **Date received**: The date the CFPB received the complaint.
8. **Date sent to company**: The date the CFPB sent the complaint to the company.
9. **Issue**: The issue the consumer identified in the complaint.
10. **Product**: The type of product the consumer identified in the complaint.

11. **State**: The state of the mailing address provided by the consumer.
12. **Sub-issue**: The sub-issue the consumer identified in the complaint.
13. **Sub-product**: The type of sub-product the consumer identified in the complaint.
14. **Submitted via**: How the complaint was submitted to the CFPB.
15. **Timely response**: Whether the company gave a timely response.
16. **Tags**: Data that supports easier searching and sorting of complaints submitted by or on behalf of consumers.
17. **ZIP code**: The mailing ZIP code provided by the consumer.
18. **ZIP**: It consists of the 3-digit mailing ZIP code of the consumer.
19. **Region**: The first digit of the ZIP Code dividing the Country into 10 different regions:
    - 0 = Connecticut (CT), Massachusetts (MA), Maine (ME), New Hampshire (NH), New Jersey (NJ), New York (NY, Fishers Island only), Puerto Rico (PR), Rhode Island (RI), Vermont (VT), Virgin Islands (VI), Army Post Office Europe, Central Asia and the Middle East (APO AE); Fleet Post Office Europe and the Middle East (FPO AE)
    - 1 = Delaware (DE), New York (NY), Pennsylvania (PA)
    - 2 = District of Columbia (DC), Maryland (MD), North Carolina (NC), South Carolina (SC), Virginia (VA), West Virginia (WV)
    - 3 = Alabama (AL), Florida (FL), Georgia (GA), Mississippi (MS), Tennessee (TN), Army Post Office Americas (APO AA), Fleet Post Office Americas (FPO AA)
    - 4 = Indiana (IN), Kentucky (KY), Michigan (MI), Ohio (OH)
    - 5 = Iowa (IA), Minnesota (MN), Montana (MT), NorthDakota (ND), South Dakota (SD), Wisconsin (WI)
    - 6 = Illinois (IL), Kansas (KS), Missouri (MO), Nebraska (NE)
    - 7 = Arkansas (AR), Louisiana (LA), Oklahoma (OK), Texas (TX)
    - 8 = Arizona (AZ), Colorado (CO), Idaho (ID), New Mexico (NM), Nevada (NV), Utah (UT), Wyoming (WY)
    - 9 = Alaska (AK), American Samoa (AS), California (CA), Guam (GU), Hawaii (HI), Marshall Islands (MH), Federated States of Micronesia (FM), Northern Mariana Islands (MP), Oregon (OR), Palau (PW), Washington (WA), Army Post Office Pacific (APO AP), Fleet Post Office Pacific (FPO AP)
20. **YearRecieved**: Contains information about the year in which complaint was received.
21. **MonthRecieved**: Contains information about the month in which complaint was received.

The columns that are important in my analysis are: Company, Issue, Product, Sub-issue, Sub-product, Submitted via, Timely response. These variables can be aggregated by State, Region and Year/Month for further analysis when merged with the other two datasets.

(ii) **Census Dataset:**

The features of this dataset are:

1. **State:** Contains full name of the State.
2. **Region:** 1st digit of the ZIP code, dividing the United States into 10 different regions
3. **Abv:** Contains abbreviated form of the State
4. **2010** : Population for the year 2010
5. **2011** : Estimated population for the year 2011.
6. **2012** : Estimated population for the year 2012.
7. **2013**: Estimated population for the year 2013.
8. **2014**: Estimated population for the year 2014.
9. **2015**: Estimated population for the year 2015.
10. **2016**: Estimated population for the year 2016.

11. **2017**: Estimated population for the year 2017.
12. **2018**: Estimated population for the year 2018.

The features that are of importance are Region, Abv and 2018.

Related Work:

(i) Run hive scripts to create hive tables/ Create Spark Dataframes for Complaint Dataset:

```
hive> Create database customer_complaints;
OK
Time taken: 0.826 seconds
hive>
    > CREATE  EXTERNAL TABLE customer_complaints.cust_complaints_all_raw
    > (date_received      string,
    > product             string,
    > sub_product         string,
    > issue               string,
    > sub_issue           string,
    > consumer_complaint_narrative string,
    > company_public_response string,
    > company             string,
    > state               string,
    > zip_code            string,
    > tags                string,
    > consumer_consent_provided string,
    > submitted_via       string,
    > date_sent_to_company string,
    > company_response    string,
    > timely_response     string,
    > consumer_disputed   string,
    > complaint_id        string)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY '\t'
    > STORED AS
    > INPUTFORMAT
    >   'com.amazonaws.emr.s3select.hive.S3SelectableTextInputFormat'
    > OUTPUTFORMAT
    >   'org.apache.hadoop.hive.ql.io.HiveIgnoreKeyTextOutputFormat'
    > LOCATION 's3://finalcomplaintdataset/Complaints'
    > TBLPROPERTIES (
    >   "s3select.format" = "csv",
    >   "s3select.headerInfo" = "ignore"
    > );
OK
```

The data that will be loaded in this table must be tab delimited. Also, the data that is present at l ocation *"s3://finalcomplaintdataset/Complaints"* will be loaded into this table.

```
hive> CREATE TABLE customer_complaints.cust_complaints_all
    > (date_received      string,
    > product             string,
    > sub_product         string,
    > issue               string,
    > sub_issue           string,
    > consumer_complaint_narrative string,
    > company_public_response string,
    > company             string,
    > state               string,
    > zip_code            string,
    > tags                string,
    > consumer_consent_provided string,
    > submitted_via       string,
    > date_sent_to_company string,
    > company_response    string,
    > timely_response     string,
    > consumer_disputed   string,
    > complaint_id        string)
    > ROW FORMAT DELIMITED
    > FIELDS TERMINATED BY ','
    > STORED AS PARQUET
    > TBLPROPERTIES ("parquet.compression"="SNAPPY");
OK
Time taken: 0.956 seconds
```

Create a table *"cust_complaints_all" in Parquet* format and insert data in it from table cust_complaints_all_raw as shown below:

```
hive> INSERT INTO customer_complaints.cust_complaints_all
    > SELECT * FROM customer_complaints.cust_complaints_all_raw;
Query ID = hadoop_20190506034924_faadde25-5b14-464c-9853-b0688fe86d22
Total jobs = 1
Launching Job 1 out of 1
Tez session was closed. Reopening...
Session re-established.
Status: Running (Executing on YARN cluster with App id application_1557112854453_0003)

----------------------------------------------------------------------------------------------
        VERTICES      MODE        STATUS  TOTAL  COMPLETED  RUNNING  PENDING  FAILED  KILLED
----------------------------------------------------------------------------------------------
Map 1 .......... container     SUCCEEDED     11         11        0        0       0       0
----------------------------------------------------------------------------------------------
VERTICES: 01/01  [==========================>>] 100%  ELAPSED TIME: 51.58 s
----------------------------------------------------------------------------------------------
Loading data to table customer_complaints.cust_complaints_all
OK
Time taken: 63.976 seconds
```

Checking for these tables created in HIVE database : customer_complaints using PySpark3:

```
In [2]: spark.sql("show tables in customer_complaints").show(5,False)

+-------------------+----------------------+-----------+
|database           |tableName             |isTemporary|
+-------------------+----------------------+-----------+
|customer_complaints|cust_complaints_all   |false      |
|customer_complaints|cust_complaints_all_raw|false      |
+-------------------+----------------------+-----------+
```

Create Spark Dataframe:

```
In [3]: df = spark.sql("select * from customer_complaints.cust_complaints_all")

In [4]: df.cache()
        df.count()

        1274208

In [5]: df.printSchema()

        root
         |-- date_received: string (nullable = true)
         |-- product: string (nullable = true)
         |-- sub_product: string (nullable = true)
         |-- issue: string (nullable = true)
         |-- sub_issue: string (nullable = true)
         |-- consumer_complaint_narrative: string (nullable = true)
         |-- company_public_response: string (nullable = true)
         |-- company: string (nullable = true)
         |-- state: string (nullable = true)
         |-- zip_code: string (nullable = true)
         |-- tags: string (nullable = true)
         |-- consumer_consent_provided: string (nullable = true)
         |-- submitted_via: string (nullable = true)
         |-- date_sent_to_company: string (nullable = true)
         |-- company_response: string (nullable = true)
         |-- timely_response: string (nullable = true)
         |-- consumer_disputed: string (nullable = true)
         |-- complaint_id: string (nullable = true)
```

Thus, from the above screenshot, we can say that, we have 1,274,208 records that needs to be analyzed.

Pre-Processing:

From the schema, we can say that the column "**date_received**" is not having datatype 'Date' but is of datatype 'String'. So, it would be difficult to extract Year/Month from it. Thus, using Spark-SQL function to_date, Year and Month from the date will be extracted.

Also, column Region is created using zip-code. But the existing zip-codes consisted of some special characters that need to be removed. It is done as given below:

In [6]:
```
dateFormat = "MM/dd/yy"
df1 = df.withColumn("DateRecieved",to_date(col("date_received"), dateFormat))
```

In [7]:
```
dfYear = df1.withColumn(
  "YearRecieved",
  year(col("DateRecieved")))
```

In [9]:
```
dfMonth = dfYear.withColumn(
  "MonthRecieved",
  month(col("DateRecieved")))
```

In [10]:
```
dfzip = dfMonth.withColumn('Zip', df.zip_code.substr(1, 3))
```

In [11]:
```
df = dfzip.withColumn('Region', df.zip_code.substr(1, 1))
```

In [15]:
```
dfcomplaints = df.where(col("Region") != "N")
dfcomplaints = dfcomplaints.where(col("Region") != "(")
dfcomplaints = dfcomplaints.where(col("Region") != "-")
dfcomplaints = dfcomplaints.where(col("Region") != "*")
```

In [16]:
```
dfcomplaints.createOrReplaceTempView("ComplaintsData")
```

A view “ComplaintsData” is created which will be useful for analysis purpose.

(ii) Create Spark Dataframe for Census Dataset:

In [14]:
```
censusDF = spark.read.format("csv")\
.option("header", "true")\
.option("inferSchema", "true")\
.load("s3://finalcomplaintdataset/Census Data.csv")
```

In [17]:
```
censusDF.createOrReplaceTempView("CensusData")
```

In [18]:
```
print(censusDF.show(5))
print(censusDF.printSchema())
```

```
+-----------+------+---+----------+--------------+--------+--------+--------+--------+--------+--------+--------+--------+-----
---+
|      State|Region|Abv|    Census|Estimates Base|    2010|    2011|    2012|    2013|    2014|    2015|    2016|    2017|    2
018|
+-----------+------+---+----------+--------------+--------+--------+--------+--------+--------+--------+--------+--------+-----
---+
|   .Alabama|     3| AL| 4,779,736|     4,780,138| 4785448| 4798834| 4815564| 4830460| 4842481| 4853160| 4864745| 4875120| 4887
871|
|    .Alaska|     9| AK|   710,231|       710,249|  713906|  722038|  730399|  737045|  736307|  737547|  741504|  739786|  737
438|
|   .Arizona|     8| AZ| 6,392,017|     6,392,288| 6407774| 6473497| 6556629| 6634999| 6733840| 6833596| 6945452| 7048876| 7171
646|
|  .Arkansas|     7| AR| 2,915,918|     2,916,028| 2921978| 2940407| 2952109| 2959549| 2967726| 2978407| 2990410| 3002997| 3013
825|
|.California|     9| CA|37,253,956|    37,254,523|37320903|37641823|37960782|38280824|38625139|38953142|39209127|39399349|39557
045|
+-----------+------+---+----------+--------------+--------+--------+--------+--------+--------+--------+--------+--------+-----
---+
only showing top 5 rows
```

The Census dataset is present at lacation “s3://finalcomplaintdataset/Census Data.csv” from where the data is loaded as shown above.

Analysis on the datasets:

Now, that the data is loaded and pre-processed, let us perform some analysis:

(i) **No of Complaints logged each Year:**

The Complaint dataset consists of complaints logged from year 2011 to year 2019.

In [8]:
```
dfYear.groupBy("YearRecieved")\
.count()\
.sort("YearRecieved", ascending= True)\
.show()
```

```
+------------+------+
|YearRecieved| count|
+------------+------+
|        2011|  2536|
|        2012| 72373|
|        2013|108218|
|        2014|153047|
|        2015|168487|
|        2016|191473|
|        2017|242975|
|        2018|257378|
|        2019| 77721|
+------------+------+
```

From the above results, we can say, the no of consumer complaints against financial companies has been constantly increasing. As only 4 months have been taken into account for the year 2019, so it has a lower no of counts.

(ii) **No of Complaints in each Region/State:**

In [22]:
```
%%sql -q -n 100 -o RegionCount
select Region, count(*) as Count
from complaintsdata group by Region
```

In [23]:
```
%%local RegionCount.head(10)
```

Type: Table Pie Scatter Line Area Bar

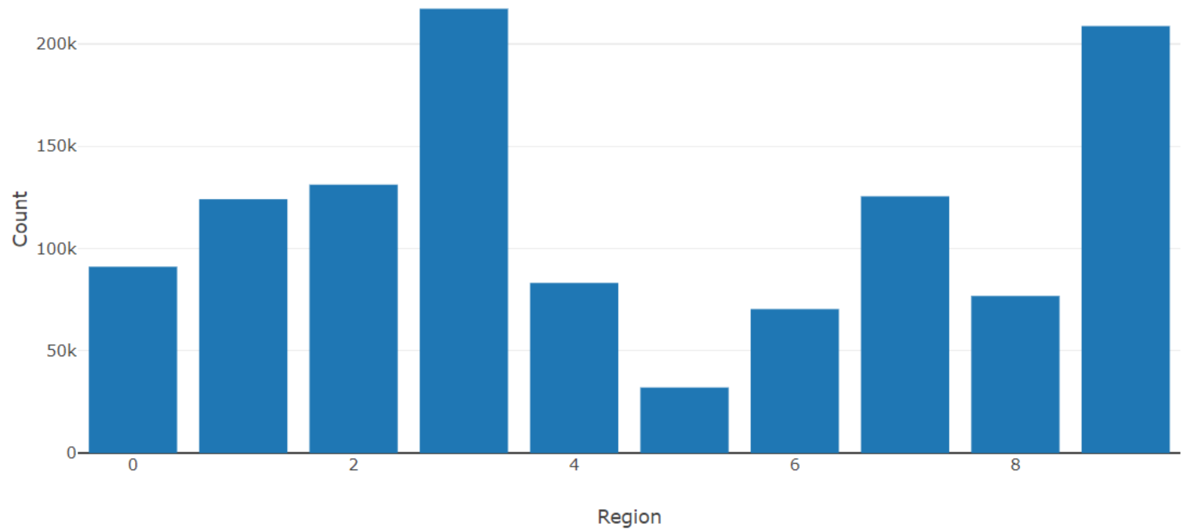
Encoding:

X Region

Y Count Func. -

Log scale X

Log scale Y



We can see that most no of complaints are being logged in Region 3 and Region 9 with more than 200K complaints logged till now.

Region 3 consists of the following States: Alabama (AL), Florida (FL), Georgia (GA), Mississippi (MS), Tennessee (TN), Army Post Office Americas (APO AA), Fleet Post Office Americas (FPO AA)

Region 9 consists of the following States: Alaska (AK), American Samoa (AS), California (CA), Guam (GU), Hawaii (HI), Marshall Islands (MH), Federated States of Micronesia (FM), Northern Mariana Islands (MP), Oregon (OR), Palau (PW), Washington (WA), Army Post Office Pacific (APO AP), Fleet Post Office Pacific (FPO AP)

Also, Region 5 consists of least no of complaints logged till now with less that 50K complaints. It consists of the following States: Iowa (IA), Minnesota (MN), Montana (MT), NorthDakota (ND), South Dakota (SD), Wisconsin (WI).

Let us see which states of Region 3/9 are responsible for the high no complaints logged.

In [55]:
```
%%sql -q -n 100 -o StateCount
select State, count(*) as Count
from complaintsdata group by State order by Count desc limit 10
```

In [56]:
```
%%local StateCount.head(10)
```

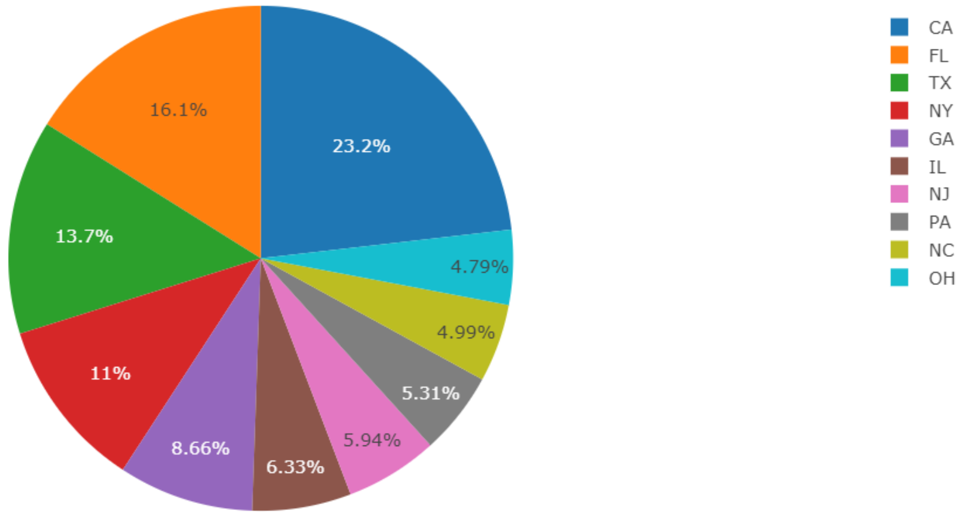
Type: Table | Pie | Scatter | Line | Area | Bar

Encoding:

X: State

Y: Count

Func.: -



From the above results we can say that, most of the states having most no of complaints are from region 3 and 9. However, states from other regions are also having a high no complaints logged such as Texas from region 7, New York. From region 1.

Let us look at the Yearly trends of the complaints logged in these regions:

```
In [20]: spark.sql("select Region, YearRecieved, count(*) as Count,  count(*) * 100.0/ sum(count(*)) over () as percentage \
                                   from complaintsdata group by Region, YearRecieved order by Region asc, YearRecieved asc").show(100)
```

Region	YearRecieved	Count	percentage
0	2011	261	0.02249747873083
0	2012	7414	0.63906631153405
0	2013	9852	0.84921517416152
0	2014	13344	1.15021592407747
0	2015	12568	1.08332686854059
0	2016	13258	1.14280296173705
0	2017	14996	1.29261375880289
0	2018	15080	1.29985432667029
0	2019	4300	0.37064811702137
1	2011	310	0.02672114332015
1	2012	8297	0.71517847137821
1	2013	11927	1.02807443993346
1	2014	16517	1.42371975554464
1	2015	16853	1.45268202701421
1	2016	18429	1.58852887176437
1	2017	20999	1.81005576961203
1	2018	23659	2.03934041874617
1	2019	7098	0.61182798479481
2	2011	267	0.02301466214993
2	2012	8030	0.69216380922828
2	2013	12320	1.06194995388448
2	2014	17551	1.51284769810280
2	2015	17659	1.52215699964659
2	2016	19227	1.65731426650462
2	2017	24427	2.10553989639092
2	2018	24808	2.13838104350375
2	2019	6906	0.59527811538362
3	2011	406	0.03499607802574
3	2012	12131	1.04565867618284
3	2013	18530	1.59723479265255
3	2014	25593	2.20604587416927
3	2015	26817	2.31155129166559
3	2016	31490	2.71435097790765
3	2017	41252	3.55580840078267
3	2018	45893	3.95584977545620
3	2019	15084	1.30019911561635
4	2011	172	0.01482592468085
4	2012	5769	0.49727185746425
4	2013	8370	0.72147086964392
4	2014	11539	0.99462991216502
4	2015	11215	0.96670200753364
4	2016	11385	0.98135553774146
4	2017	15135	1.30459517467870
4	2018	15150	1.30588813322645
4	2019	4425	0.38142277158594
5	2011	97	0.00836113194211
5	2012	2321	0.20006378595502
5	2013	3258	0.28083059657107
5	2014	4726	0.40736813977744
5	2015	4392	0.37857826278090
5	2016	4751	0.40952307069035
5	2017	5859	0.50502960875074
5	2018	5138	0.44288140122228
5	2019	1508	0.12998543266703
6	2011	134	0.01155042969322
6	2012	4116	0.35478782550231
6	2013	5830	0.50252988889176
6	2014	8702	0.75008835216743
6	2015	8798	0.75836328687302
6	2016	10280	0.88610759139062
6	2017	13476	1.16159395929766
6	2018	15077	1.29959573496074
6	2019	3955	0.34091007042314

7	2011	178	0.01534310809995
7	2012	4796	0.41340194633360
7	2013	9128	0.78680837492350
7	2014	16549	1.42647806711317
7	2015	15708	1.35398619120271
7	2016	17971	1.54905053743977
7	2017	25501	2.19811572840975
7	2018	27434	2.36473498659633
7	2019	8227	0.70914466482205
8	2011	146	0.01258479653142
8	2012	4601	0.39659348521286
8	2013	6965	0.60036375233810
8	2014	10127	0.87291941420358
8	2015	10782	0.92937860412195
8	2016	12172	1.04919276288002
8	2017	13662	1.17762664528975
8	2018	13937	1.20133088533182
8	2019	4398	0.37909544619999
9	2011	550	0.04740848008413
9	2012	14283	1.23115512916656
9	2013	20983	1.80867661382776
9	2014	27394	2.36128709713567
9	2015	28680	2.47213674329601
9	2016	31873	2.74736451949351
9	2017	37095	3.19748648858318
9	2018	37055	3.19403859912251
9	2019	10804	0.93127494332532

From the above results, we can say that with each year, the no of complaints are increases against the financial companies.

(iii) Does population of a state affect the no of complaints logged:

```
In [59]: RegionDF = dfnew3.groupBy("Region").agg(count("complaint_id").alias("No-of-Complaints"))
RegCenDF = censusDF.groupBy("Region").agg(sum("2018").alias("Estimated-Population"))
joinType = "inner"
joinExpression = RegionDF["Region"] == censusDF['Region']
RegionDF.join(RegCenDF, joinExpression, joinType).sort("Estimated-Population").show()
```

Region	No-of-Complaints	Region	Estimated-Population
5	32050	5	17285509
8	76790	8	23490080
6	70368	6	23708305
0	91073	0	23761810
2	131195	2	32536437
4	83160	4	32845637
1	124089	1	33316440
7	125492	7	40318727
3	217196	3	46463211
9	208717	9	53441278

From the above results we can say that, region 5 has the least population, so the no of complaints logged are less.

Similarly, region 3 and 9 have more population, so the no of complaints logged are more.

Also, the population in region 4 is more than region 2 but the no of complaints logged by region 4 is less compared to region 2.

In [70]:
```
StateDF = dfnew3.groupBy("State").agg(count("complaint_id").alias("No-of-Complaints"))
stateCensDF = censusDF.withColumn("CA", col("Abv") == "CA")
stateCensDF = stateCensDF.withColumn("FL",col("Abv") == "FL")
stateCensDF = stateCensDF.withColumn("TX",col("Abv") == "TX")
stateCensDF = stateCensDF.withColumn("NY",col("Abv") == "NY")
stateCensDF = stateCensDF.withColumn("GA",col("Abv") == "GA")
stateCensDF = stateCensDF.withColumn("IL",col("Abv") == "IL")
stateCensDF = stateCensDF.withColumn("NJ",col("Abv") == "NJ")
stateCensDF = stateCensDF.withColumn("PA",col("Abv") == "PA")
stateCensDF = stateCensDF.withColumn("NC",col("Abv") == "NC")
stateCensDF = stateCensDF.withColumn("OH",col("Abv") == "OH")
StateCenDF = stateCensDF.select("Abv", "2018").where("CA or FL or TX or NY or GA or NJ or PA or NC or OH")
```

In [73]:
```
joinType = "inner"
joinExpression = StateDF["State"] == StateCenDF['Abv']
StateDF.join(StateCenDF, joinExpression, joinType).sort("2018").show()
```

```
+-----+----------------+---+--------+
|State|No-of-Complaints|Abv|    2018|
+-----+----------------+---+--------+
|   NJ|           43214| NJ| 8908520|
|   NC|           36349| NC|10383620|
|   GA|           63036| GA|10519475|
|   OH|           34829| OH|11689442|
|   PA|           38621| PA|12807060|
|   NY|           79812| NY|19542209|
|   FL|          117108| FL|21299325|
|   TX|           99956| TX|28701845|
|   CA|          168694| CA|39557045|
+-----+----------------+---+--------+
```

Considering the top 10 states in which most no of complaints were logged, the population vs no of complaints results are as above.

From the results we can say that if the population is more, complaints is more. However, there are some exceptions as well such as states like OHIO and PA is not the case.

(iv) Company against which most no of complaints were logged:

In [74]:
```
spark.sql("select company, count(*) as Count, round(count(*) * 100.0/ sum(count(*)) over (),2) as percentage \
from complaintsdata \
group by company \
order by Count desc limit 10").show(10,False)
```

```
+------------------------------------+------+----------+
|company                             |Count |percentage|
+------------------------------------+------+----------+
|EQUIFAX, INC.                       |104879|9.04      |
|Experian Information Solutions Inc. |94296 |8.13      |
|TRANSUNION INTERMEDIATE HOLDINGS, INC.|87084 |7.51      |
|BANK OF AMERICA, NATIONAL ASSOCIATION |77420 |6.67      |
|WELLS FARGO & COMPANY               |65817 |5.67      |
|JPMORGAN CHASE & CO.                |55759 |4.81      |
|CITIBANK, N.A.                      |44860 |3.87      |
|CAPITAL ONE FINANCIAL CORPORATION   |31361 |2.70      |
|OCWEN LOAN SERVICING LLC            |26266 |2.26      |
|Navient Solutions, LLC.             |25304 |2.18      |
+------------------------------------+------+----------+
```

The company against which most no of complaints were logged are “Equifax, INC”.

The issues that have been most complained about this company are:

```
In [20]: spark.sql("select issue, count(*) as Count, round(count(*) * 100.0/ sum(count(*)) over (),2) as percentage \
from complaintsdata \
where company = 'EQUIFAX, INC.' \
group by issue \
order by Count desc limit 10").show(10, False)
```

```
+------------------------------------------------------------------------------+-----+----------+
|issue                                                                         |Count|percentage|
+------------------------------------------------------------------------------+-----+----------+
|Incorrect information on credit report                                        |32521|31.01     |
|Incorrect information on your report                                          |31155|29.71     |
|Problem with a credit reporting company's investigation into an existing problem|12401|11.82     |
|Improper use of your report                                                   |10082|9.61      |
|Credit reporting company's investigation                                      |5733 |5.47      |
|Unable to get credit report/credit score                                      |4192 |4.00      |
|Unable to get your credit report or credit score                              |1652 |1.58      |
|Problem with fraud alerts or security freezes                                 |1573 |1.50      |
|Improper use of my credit report                                              |1572 |1.50      |
|Credit monitoring or identity protection                                      |1295 |1.23      |
+------------------------------------------------------------------------------+-----+----------+
```

(v) Product against most no of complaints were logged:

```
In [23]: %%sql -q -n 100 -o ProductCount
select product, count(*) as Count
from complaintsdata group by product limit 10
```

```
In [24]: %%local ProductCount.head(10)
```

Type: Table Pie Scatter Line Area Bar

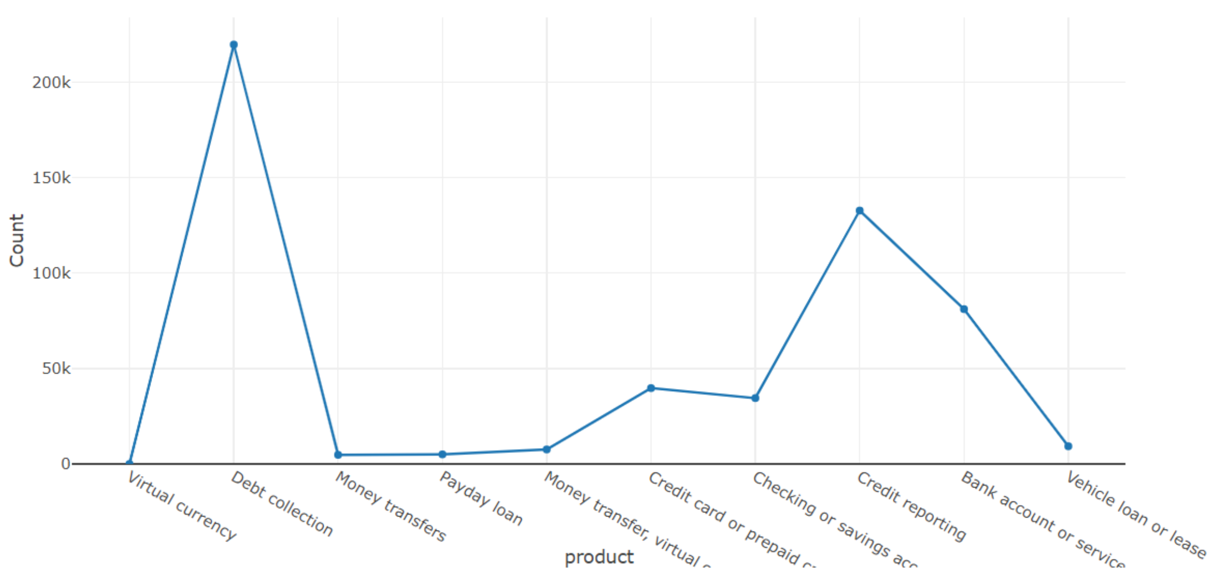
Encoding:

X product

Y Count Func. -

Log scale X

Log scale Y



From the above results, we can say that most of the complaints are related to Debt Collection and Credit reporting.

(vi) Was a Timely response given by the companies against which complaint was logged?

```
In [25]: %%sql -q -n 100 -o TimelyCount
         select timely_response, count(*) as Count
         from complaintsdata group by timely_response order by Count desc
```

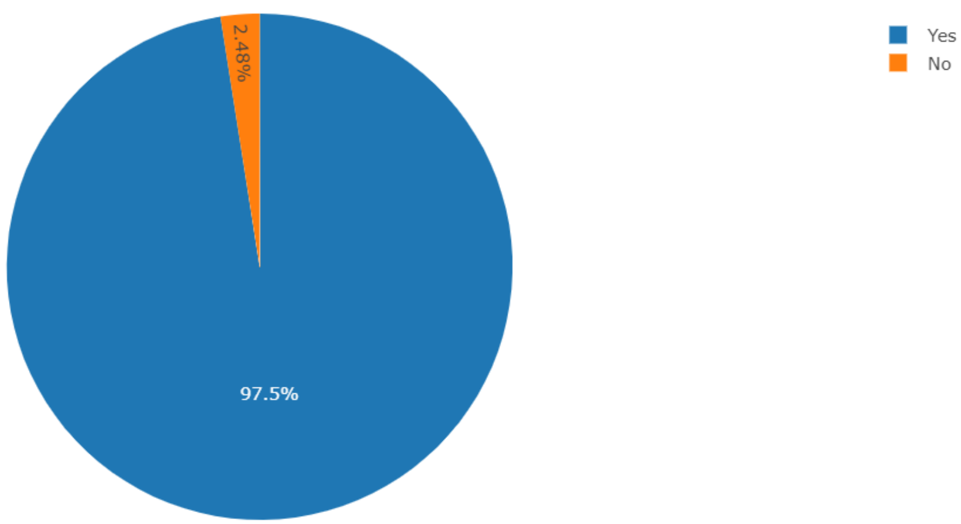
```
In [26]: %%local TimelyCount.head(10)
```

Type: Table Pie Scatter Line Area Bar

Encoding:

X timely_respon:

Y Count Func. -



From the above results we can say that most of the companies give a timely response.
Let us check for the companies that have not given a timely response:

```
In [27]: spark.sql("select company, count(*) as Count \
         from complaintsdata \
         where timely_response = 'No' \
         group by company \
         order by Count desc limit 10").show(10,False)
```

```
+------------------------------------+-----+
|company                             |Count|
+------------------------------------+-----+
|WELLS FARGO & COMPANY               |2894 |
|BANK OF AMERICA, NATIONAL ASSOCIATION|1569 |
|EQUIFAX, INC.                       |1464 |
|OCWEN LOAN SERVICING LLC            |525  |
|Colony Brands, Inc.                 |359  |
|CITIBANK, N.A.                      |352  |
|Mobiloans, LLC                      |337  |
|Southwest Credit Systems, L.P.      |271  |
|Midwest Recovery Systems            |207  |
|Residential Credit Solutions, Inc.  |171  |
+------------------------------------+-----+
```

From the above table, we can say that “WELLS FARGO & COMPANY” is one of the companies that do not give a timely esponse to the customers complaints.

Let us check the top products for which timely response wasn’t given by “WELLS FARGO & COMPANY”

```
In [29]: spark.sql("select company, product, count(*) as Count \
         from complaintsdata \
         where timely_response = 'No' and company  = 'WELLS FARGO & COMPANY'\
         group by company, product \
         order by Count desc limit 10").show(10,False)
```

```
+---------------------+--------------------------------------------------------------------------+-----+
|company              |product                                                                   |Count|
+---------------------+--------------------------------------------------------------------------+-----+
|WELLS FARGO & COMPANY|Bank account or service                                                   |1379 |
|WELLS FARGO & COMPANY|Consumer Loan                                                             |369  |
|WELLS FARGO & COMPANY|Credit card                                                               |293  |
|WELLS FARGO & COMPANY|Checking or savings account                                               |177  |
|WELLS FARGO & COMPANY|Debt collection                                                           |153  |
|WELLS FARGO & COMPANY|Vehicle loan or lease                                                     |143  |
|WELLS FARGO & COMPANY|Mortgage                                                                  |104  |
|WELLS FARGO & COMPANY|Credit reporting, credit repair services, or other personal consumer reports|84   |
|WELLS FARGO & COMPANY|Credit card or prepaid card                                               |58   |
|WELLS FARGO & COMPANY|Money transfers                                                           |37   |
+---------------------+--------------------------------------------------------------------------+-----+
```

(vii) No of ways complaints submitted to CFPB

```
In [34]: %%sql -q -n 100 -o SubmitCount
         select submitted_via, count(*) as Count
         from complaintsdata group by submitted_via order by Count desc
```

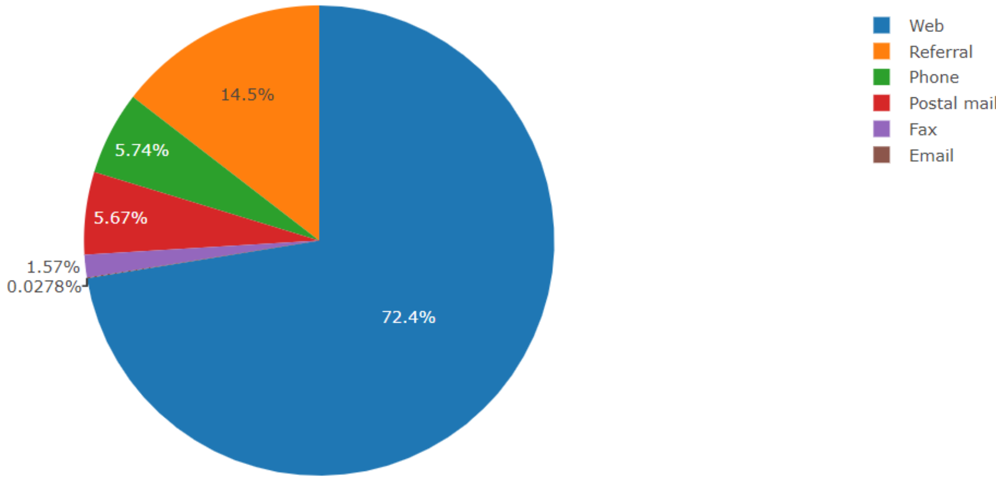
```
In [35]: %%local SubmitCount.head(10)
```

Type: Table Pie Scatter Line Area Bar

Encoding:

X submitted_via

Y Count Func. -



From the results we can say that most of the complaints were filed via Web followed by Referral.

(viii) Dates on which most no of complaints were logged:

```
In [36]: spark.sql("select DateRecieved, count(*) as Count, round(count(*) * 100.0/ sum(count(*)) over (),2) as percentage \
from complaintsdata \
group by DateRecieved \
order by Count desc limit 10").show(10)
```

```
+------------+-----+----------+
|DateRecieved|Count|percentage|
+------------+-----+----------+
|  2017-09-08| 3118|      0.27|
|  2017-09-09| 2397|      0.21|
|  2017-01-19| 1808|      0.16|
|  2017-01-20| 1432|      0.12|
|  2017-09-13| 1365|      0.12|
|  2018-04-05| 1154|      0.10|
|  2017-09-12| 1072|      0.09|
|  2018-04-10| 1041|      0.09|
|  2018-04-24| 1017|      0.09|
|  2017-09-14|  988|      0.09|
+------------+-----+----------+
```

From the year 2011 to 2019, the dates when most no of complaints were logged are given above. From these results we can say that, no many complaints were logged in the starting or end of the month.

(ix) Year-Month when most no of complaints were logged:

In [37]:
```
spark.sql("select concat(YearRecieved,'/', MonthRecieved) as YearMonth, count(*) as Count, round(count(*) * 100.0/ sum(count(*))
from complaintsdata \
group by YearRecieved, MonthRecieved \
order by Count desc limit 10").show(10)
```

```
+---------+-----+----------+
|YearMonth|Count|percentage|
+---------+-----+----------+
|   2017/9|23754|      2.05|
|   2018/4|21466|      1.85|
|   2018/3|20714|      1.79|
|   2018/1|20493|      1.77|
|   2019/3|19957|      1.72|
|   2018/5|19444|      1.68|
|   2018/2|19180|      1.65|
|  2018/10|18801|      1.62|
|   2018/8|18765|      1.62|
|   2017/8|18678|      1.61|
+---------+-----+----------+
```

Of all the years having complaints, most no of complaints were logged in the year 2018. However, most no of complaints were logged in the moth of September in the year 2017.
Also, most of the complaints were logged in the mid-quarter months and not in the beginning or end of the quarter.

Conclusion:

Thus, it can be said that the no of complaints being logged against companies is increases with each year.
Also, the population in a region/state affects the no of complaints logged in each region/state. More the population, more the complaints.

The companies that have been most complained about are:

(i) Equifax INC.
(ii) Experian Informtion Solutions Inc.
(iii) Transunion Intermediate Holdings Inc.

Most of the companies against which complaints were logged have given a timely response
Also, a lot of complaints were logged in the mid-months and mid-quarters.

References:

Consumer Complaint dataset: https://www.consumerfinance.gov/complaint/data-use/
Zip Codes Information: https://en.wikipedia.org/wiki/ZIP_Code
Census dataset: https://www.census.gov/data/datasets/time-series/demo/popest/2010s-state-total.html