Quiz 2B

CS 119: Big Data

Submitted by: Hezekiah Branch

In [1]:

```
import pyspark
import nltk
import pandas as pd
```

Question 1: TF-IDF analysis of text

Read in dataset of last ten president speeeches from public Google Cloud bucket

In [2]:

```
path = "https://storage.googleapis.com/jsingh-bigdata-public/ten_us_presidents_transcripts.csv"
pres_speeches = pd.read_csv(path)
pres_speeches.drop('Unnamed: 0', axis=1, inplace=True)
pres_speeches.head()
```

Out[2]:

	content	president	year
0	\nSenator Hatfield, Mr. Chief Justice, Mr. Pre...	Ronald Reagan	1981
1	\nSenator Mathias, Chief Justice Burger, Vice ...	Ronald Reagan	1985
2	\nMr. Chief Justice, Mr. President, Vice Presi...	George Bush	1989
3	\nMy fellow citizens, today we celebrate the m...	William J. Clinton	1993
4	\nMy fellow citizens, at this last Presidentia...	William J. Clinton	1997

In [3]:

```
speeches = pres_speeches.copy()
```

In [4]:

```
separate_docs = lambda x: x.split('\n')[1:]
speeches['content'] = speeches['content'].apply(separate_docs)
speeches
```

Out[4]:

	content	president	year
0	[Senator Hatfield, Mr. Chief Justice, Mr. Pres...	Ronald Reagan	1981
1	[Senator Mathias, Chief Justice Burger, Vice P...	Ronald Reagan	1985
2	[Mr. Chief Justice, Mr. President, Vice Presid...	George Bush	1989
3	[My fellow citizens, today we celebrate the my...	William J. Clinton	1993
4	[My fellow citizens, at this last Presidential...	William J. Clinton	1997
5	[Thank you, all. Chief Justice Rehnquist, Pres...	George W. Bush	2001
6	[Vice President Cheney, Mr. Chief Justice, Pre...	George W. Bush	2005
7	[My fellow citizens, I stand here today humble...	Barack Obama	2009
8	[Thank you. Thank you so much., Vice President...	Barack Obama	2013
9	[Chief Justice Roberts, President Carter, Pres...	Donald J. Trump	2017

Add required preprocessing with regex from assignment prompt

In [5]:

```
import re
import string

def clean_text(text):
    text = text.lower()
    text = re.sub('\[.*?\]', '', text)
    text = re.sub('[%s]' % re.escape(string.punctuation), ' ', text)
    text = re.sub('[\d\n]', ' ', text)
    return text

round1 = lambda x: clean_text(x).strip()
```

In [6]:

```
for row in range(len(speeches.content)):
    cleaned = []
    for item in speeches.content.iloc[row]:
        cleaned.append(clean_text(item).strip())
    speeches.at[row, 'content'] = cleaned
```

In [7]:

```
speeches
```

Out[7]:

	content	president	year
0	[senator hatfield mr chief justice mr pres...	Ronald Reagan	1981
1	[senator mathias chief justice burger vice p...	Ronald Reagan	1985
2	[mr chief justice mr president vice presid...	George Bush	1989
3	[my fellow citizens today we celebrate the my...	William J. Clinton	1993
4	[my fellow citizens at this last presidential...	William J. Clinton	1997
5	[thank you all chief justice rehnquist pres...	George W. Bush	2001
6	[vice president cheney mr chief justice pre...	George W. Bush	2005
7	[my fellow citizens i stand here today humble...	Barack Obama	2009
8	[thank you thank you so much, vice president ...	Barack Obama	2013
9	[chief justice roberts president carter pres...	Donald J. Trump	2017

Build TF-IDF Matrix using scikit-learn TfidfVectorizer

In [8]:

```
from sklearn.feature_extraction.text import TfidfVectorizer

corpus = speeches.content.iloc[0]
vectorizer = TfidfVectorizer(analyzer='word', use_idf=True)
tfidf_matrix = vectorizer.fit_transform(corpus)
tfidf_matrix
```

Out[8]:

```
<38x835 sparse matrix of type '<class 'numpy.float64'>'
	with 1728 stored elements in Compressed Sparse Row format>
```

In [9]:

```
tfidf_matrix[0,:].nonzero()[1]
```

Out[9]:

```
array([111, 245, 474,  27, 465, 609, 484, 672,  47, 460,  85, 778, 551,
       388, 105, 470, 313, 635], dtype=int32)
```

In [10]:

```
mapping = pd.DataFrame.from_dict(dict(zip(vectorizer.get_feature_names(), vectorizer.idf_)),
                                 orient='index',
                                 columns=['tf_idf'])

mapping
```

Out[10]:

	tf_idf
above	3.970414
abraham	3.970414
accept	3.970414
achieved	3.970414
achievement	3.970414
...	...
yet	3.564949
you	2.584120
young	3.564949
your	3.277267
yourselves	3.970414

835 rows × 1 columns

In [11]:

```
top_20 = mapping.tf_idf.sort_values(ascending=False)[0:20]
cell_top_20 = []
for name in top_20.index:
    cell_top_20.append((name, top_20[name]))
cell_top_20
```

Out[11]:

```
[('above', 3.970414465569701),
 ('paraphrase', 3.970414465569701),
 ('opening', 3.970414465569701),
 ('opportunities', 3.970414465569701),
 ('order', 3.970414465569701),
 ('orderly', 3.970414465569701),
 ('pace', 3.970414465569701),
 ('paddies', 3.970414465569701),
 ('paid', 3.970414465569701),
 ('parallel', 3.970414465569701),
 ('part', 3.970414465569701),
 ('omaha', 3.970414465569701),
 ('party', 3.970414465569701),
 ('past', 3.970414465569701),
 ('patriotism', 3.970414465569701),
 ('patrol', 3.970414465569701),
 ('perform', 3.970414465569701),
 ('period', 3.970414465569701),
 ('personal', 3.970414465569701),
 ('piled', 3.970414465569701)]
```

Run for entire corpus of presidential speech documents

In [12]:

```python
tf_idf_matrix = []

for doc in speeches.content:
    corpus = doc
    vectorizer = TfidfVectorizer(analyzer='word', use_idf=True)
    tfidf_matrix = vectorizer.fit_transform(corpus)
    tfidf_matrix
    mapping = pd.DataFrame.from_dict(dict(zip(vectorizer.get_feature_names(), vectorizer.idf_)),
                                     orient='index',
                                     columns=['tf_idf'])
    top_20 = mapping.tf_idf.sort_values(ascending=False)[0:20]
    cell_top_20 = []
    for name in top_20.index:
        cell_top_20.append((name, top_20[name]))
    tf_idf_matrix.append(cell_top_20)
```

In [13]:

```python
cols = [str(speeches.iloc[row].president.split()[-1]) + '_' + str(speeches.iloc[row].year) for row in rang
cols = dict(zip(range(10), cols))
tf_idf_matrix = pd.DataFrame(tf_idf_matrix).transpose()
tf_idf_matrix = tf_idf_matrix.rename(columns=cols)
tf_idf_matrix
```

Out[13]:

	Reagan_1981	Reagan_1985	Bush_1989	Clinton_1993	Clinton_1997	Bush_2001
0	(above, 3.970414465569701)	(absent, 4.091042453358316)	(important, 3.70805020110221)	(abiding, 3.2512917986064953)	(abolished, 3.5649493574615367)	(abandonment, 3.772588722239781)
1	(paraphrase, 3.970414465569701)	(out, 4.091042453358316)	(money, 3.70805020110221)	(ocean, 3.2512917986064953)	(opened, 3.5649493574615367)	(others, 3.772588722239781)
2	(opening, 3.970414465569701)	(once, 4.091042453358316)	(merely, 3.70805020110221)	(part, 3.2512917986064953)	(obligations, 3.5649493574615367)	(persistent, 3.772588722239781)
3	(opportunities, 3.970414465569701)	(open, 4.091042453358316)	(michel, 3.70805020110221)	(pain, 3.2512917986064953)	(obsessions, 3.5649493574615367)	(permit, 3.772588722239781)
4	(order, 3.970414465569701)	(opportunities, 4.091042453358316)	(mistrust, 3.70805020110221)	(over, 3.2512917986064953)	(off, 3.5649493574615367)	(people, 3.772588722239781)
5	(orderly, 3.970414465569701)	(oppression, 4.091042453358316)	(mists, 3.70805020110221)	(out, 3.2512917986064953)	(office, 3.5649493574615367)	(peaceful, 3.772588722239781)
6	(pace, 3.970414465569701)	(orderly, 4.091042453358316)	(mitchell, 3.70805020110221)	(order, 3.2512917986064953)	(onto, 3.5649493574615367)	(peace, 3.772588722239781)
7	(paddies, 3.970414465569701)	(origin, 4.091042453358316)	(trumpets, 3.70805020110221)	(opportunities, 3.2512917986064953)	(open, 3.5649493574615367)	(pastor, 3.772588722239781)
8	(paid, 3.970414465569701)	(others, 4.091042453358316)	(truly, 3.70805020110221)	(one, 3.2512917986064953)	(opportunities, 3.5649493574615367)	(passing, 3.772588722239781)
9	(parallel, 3.970414465569701)	(ours, 4.091042453358316)	(member, 3.70805020110221)	(oldest, 3.2512917986064953)	(nuclear, 3.5649493574615367)	(passed, 3.772588722239781)
10	(part, 3.970414465569701)	(overwhelm, 4.091042453358316)	(most, 3.70805020110221)	(older, 3.2512917986064953)	(other, 3.5649493574615367)	(pain, 3.772588722239781)
11	(omaha, 3.970414465569701)	(planter, 4.091042453358316)	(motives, 3.70805020110221)	(old, 3.2512917986064953)	(others, 3.5649493574615367)	(page, 3.772588722239781)
12	(party, 3.970414465569701)	(own, 4.091042453358316)	(move, 3.70805020110221)	(often, 3.2512917986064953)	(outlines, 3.5649493574615367)	(over, 3.772588722239781)

13	(past, 3.970414465569701)	(paces, 4.091042453358316)	(moves, 3.70805020110221)	(office, 3.2512917986064953)	(overcome, 3.5649493574615367)	(out, 3.772588722239781)
14	(patriotism, 3.970414465569701)	(part, 4.091042453358316)	(moving, 3.70805020110221)	(offer, 3.2512917986064953)	(parents, 3.5649493574615367)	(order, 3.772588722239781)
15	(patrol, 3.970414465569701)	(parties, 4.091042453358316)	(triumph, 3.70805020110221)	(oath, 3.2512917986064953)	(part, 3.5649493574615367)	(remains, 3.772588722239781)
16	(perform, 3.970414465569701)	(party, 4.091042453358316)	(try, 3.70805020110221)	(pays, 3.2512917986064953)	(numbers, 3.5649493574615367)	(options, 3.772588722239781)
17	(period, 3.970414465569701)	(pass, 4.091042453358316)	(meet, 3.70805020110221)	(numbers, 3.2512917986064953)	(nothing, 3.5649493574615367)	(opportunity, 3.772588722239781)
18	(personal, 3.970414465569701)	(passing, 4.091042453358316)	(voices, 3.70805020110221)	(nor, 3.2512917986064953)	(partisanship, 3.5649493574615367)	(onward, 3.772588722239781)
19	(piled, 3.970414465569701)	(paying, 4.091042453358316)	(turning, 3.70805020110221)	(news, 3.2512917986064953)	(mystical, 3.5649493574615367)	(one, 3.772588722239781)

Question 1: Retail Data Analysis

In [14]:

```
online_retail = pd.read_csv("online-retail-online_retail_II.csv")
online_retail
```

Out[14]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	12/1/2009 7:45	6.95	13085.0	United Kingdom
1	489434	79323P	PINK CHERRY LIGHTS	12	12/1/2009 7:45	6.75	13085.0	United Kingdom
2	489434	79323W	WHITE CHERRY LIGHTS	12	12/1/2009 7:45	6.75	13085.0	United Kingdom
3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	12/1/2009 7:45	2.10	13085.0	United Kingdom
4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	12/1/2009 7:45	1.25	13085.0	United Kingdom
...	...	...	...	...	...	...	...	...
1067366	581587	22899	CHILDREN'S APRON DOLLY GIRL	6	12/9/2011 12:50	2.10	12680.0	France
1067367	581587	23254	CHILDRENS CUTLERY DOLLY GIRL	4	12/9/2011 12:50	4.15	12680.0	France
1067368	581587	23255	CHILDRENS CUTLERY CIRCUS PARADE	4	12/9/2011 12:50	4.15	12680.0	France
1067369	581587	22138	BAKING SET 9 PIECE RETROSPOT	3	12/9/2011 12:50	4.95	12680.0	France
1067370	581587	POST	POSTAGE	1	12/9/2011 12:50	18.00	12680.0	France

1067371 rows × 8 columns

In [15]:

```
from pyspark.sql import SparkSession

spark = SparkSession \
    .builder \
    .appName("Online Retail") \
    .getOrCreate()
```

```
21/12/29 23:03:34 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using b
uiltin-java classes where applicable
Using Spark's default log4j profile: org/apache/spark/log4j-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
```

In [16]:

```python
path = "online-retail-online_retail_II.csv"

df = spark.read.option("header", True).csv(path, inferSchema=True)
df.show()
```

```
+-------+---------+--------------------+--------+--------------+-----+-----------+--------------+
|Invoice|StockCode|         Description|Quantity|   InvoiceDate|Price|Customer ID|       Country|
+-------+---------+--------------------+--------+--------------+-----+-----------+--------------+
| 489434|    85048|15CM CHRISTMAS GL...|      12|12/1/2009 7:45| 6.95|      13085|United Kingdom|
| 489434|   79323P|  PINK CHERRY LIGHTS|      12|12/1/2009 7:45| 6.75|      13085|United Kingdom|
| 489434|   79323W| WHITE CHERRY LIGHTS|      12|12/1/2009 7:45| 6.75|      13085|United Kingdom|
| 489434|    22041|"RECORD FRAME 7""...|      48|12/1/2009 7:45|  2.1|      13085|United Kingdom|
| 489434|    21232|STRAWBERRY CERAMI...|      24|12/1/2009 7:45| 1.25|      13085|United Kingdom|
| 489434|    22064|PINK DOUGHNUT TRI...|      24|12/1/2009 7:45| 1.65|      13085|United Kingdom|
| 489434|    21871| SAVE THE PLANET MUG|      24|12/1/2009 7:45| 1.25|      13085|United Kingdom|
| 489434|    21523|FANCY FONT HOME S...|      10|12/1/2009 7:45| 5.95|      13085|United Kingdom|
| 489435|    22350|           CAT BOWL |      12|12/1/2009 7:46| 2.55|      13085|United Kingdom|
| 489435|    22349|DOG BOWL , CHASIN...|      12|12/1/2009 7:46| 3.75|      13085|United Kingdom|
| 489435|    22195|HEART MEASURING S...|      24|12/1/2009 7:46| 1.65|      13085|United Kingdom|
| 489435|    22353|LUNCHBOX WITH CUT...|      12|12/1/2009 7:46| 2.55|      13085|United Kingdom|
| 489436|   48173C|DOOR MAT BLACK FL...|      10|12/1/2009 9:06| 5.95|      13078|United Kingdom|
| 489436|    21755|LOVE BUILDING BLO...|      18|12/1/2009 9:06| 5.45|      13078|United Kingdom|
| 489436|    21754|HOME BUILDING BLO...|       3|12/1/2009 9:06| 5.95|      13078|United Kingdom|
| 489436|    84879|ASSORTED COLOUR B...|      16|12/1/2009 9:06| 1.69|      13078|United Kingdom|
| 489436|    22119| PEACE WOODEN BLO...|       3|12/1/2009 9:06| 6.95|      13078|United Kingdom|
| 489436|    22142|CHRISTMAS CRAFT W...|      12|12/1/2009 9:06| 1.45|      13078|United Kingdom|
| 489436|    22296|HEART IVORY TRELL...|      12|12/1/2009 9:06| 1.65|      13078|United Kingdom|
| 489436|    22295|HEART FILIGREE DO...|      12|12/1/2009 9:06| 1.65|      13078|United Kingdom|
+-------+---------+--------------------+--------+--------------+-----+-----------+--------------+
only showing top 20 rows
```

In [17]:

```python
df.dtypes
```

Out[17]:

```
[('Invoice', 'string'),
 ('StockCode', 'string'),
 ('Description', 'string'),
 ('Quantity', 'int'),
 ('InvoiceDate', 'string'),
 ('Price', 'double'),
 ('Customer ID', 'int'),
 ('Country', 'string')]
```

Question: Loading the data into a Dataframe and removing junk records. How many records were removed by doing so?

Response: **When removing junk records, we have found 269486 records to remove.**

In [18]:

```
df_clean = df.dropna()
df_clean = df_clean.dropDuplicates()
num_empty_rows = df.count() - df_clean.count()
num_empty_rows
```

Out[18]:

```
269486
```

Dealing with calculation of monetary value

In [19]:

```
df_clean.show()
```

```
[Stage 8:==============>                                                  (1 + 3) / 4]
+-------+---------+--------------------+--------+---------------+-----+-----------+--------------+
|Invoice|StockCode|         Description|Quantity|    InvoiceDate|Price|Customer ID|       Country|
+-------+---------+--------------------+--------+---------------+-----+-----------+--------------+
| 489514|    21791|VINTAGE HEADS AND...|      72|12/1/2009 11:21| 1.06|      15311|United Kingdom|
| 489520|   72739B|WHITE CHOCOLATE S...|      12|12/1/2009 11:41| 1.25|      14911|          EIRE|
| 489522|    22315|200 RED + WHITE B...|       1|12/1/2009 11:45| 1.25|      15998|United Kingdom|
| 489536|    21611|SET OF 12 LILY BO...|       2|12/1/2009 12:13| 2.95|      16393|United Kingdom|
| 489561|    21816|CHRISTMAS TREE T-...|       6|12/1/2009 12:57| 1.45|      14654|United Kingdom|
| 489562|    35071|ASSORTED SANTA CH...|       1|12/1/2009 13:07| 0.85|      17998|United Kingdom|
| 489576|    22152|PLACE SETTING WHI...|      24|12/1/2009 13:38| 0.42|      15984|United Kingdom|
| 489599|    21239|    PINK  SPOTTY CUP|      16|12/1/2009 14:40| 0.85|      12758|      Portugal|
| 489658|  79323LP|LIGHT PINK CHERRY...|       6|12/1/2009 17:31| 6.75|      15485|United Kingdom|
| 489679|    22086|PAPER CHAIN KIT 5...|       6|12/2/2009 10:00| 2.95|      16163|United Kingdom|
| 489681|   85226C|BLUE PULL BACK RA...|      11|12/2/2009 10:02| 0.55|      17998|United Kingdom|
| 489723|   85231E|STRAWBERRY SCENTE...|      36|12/2/2009 10:58| 0.85|      14299|United Kingdom|
| 489765|    22315|200 RED + WHITE B...|       1|12/2/2009 11:33| 1.25|      15353|United Kingdom|
| 489766|    21975|PACK OF 60 DINOSA...|       1|12/2/2009 11:34| 0.55|      17818|United Kingdom|
| 489791|   84029G|KNITTED UNION FLA...|       1|12/2/2009 12:06| 3.75|      15542|United Kingdom|
| 489797|   72807A|SET/3 ROSE CANDLE...|       5|12/2/2009 12:19| 4.25|      15581|United Kingdom|
| 489814|    21973|SET OF 36 MUSHROO...|       2|12/2/2009 13:06| 1.45|      14669|United Kingdom|
| 489827|    22335|HEART DECORATION ...|      24|12/2/2009 13:51| 0.65|      17412|United Kingdom|
| 489866|    21877| HOME SWEET HOME MUG|       6|12/2/2009 15:04| 1.25|      16200|United Kingdom|
| 489878|    20966|SANDWICH BATH SPONGE|       1|12/2/2009 15:51| 1.25|      15989|United Kingdom|
+-------+---------+--------------------+--------+---------------+-----+-----------+--------------+
only showing top 20 rows
```

Change the Price and InvoiceDate columns to type Double and timestamp, respectively.

The cell ran but I cleared the output due to a large number of observations

In [20]:

```
from pyspark.sql.types import DoubleType, TimestampType, StringType, DateType

df_clean = df_clean.withColumn("Price", df_clean["Price"].cast(DoubleType()))
```

In [22]:

```
from pyspark.sql.functions import to_timestamp

spark.sql("set spark.sql.legacy.timeParserPolicy=LEGACY")

test = df_clean.select('Customer ID', to_timestamp(df_clean.InvoiceDate, 'MM/d/yyyy HH:mm').alias('Invoice
```

In [23]:

```
test.schema['InvoiceDate']
```

Out[23]:

```
StructField(InvoiceDate,TimestampType,true)
```

In [25]:

```
test = test.orderBy('InvoiceDate', ascending=False)
```

In [26]:

```
test.show()
```

```
[Stage 11:==========================================>            (154 + 4) / 200]
+-----------+-------------------+
|Customer ID|        InvoiceDate|
+-----------+-------------------+
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      12680|2011-12-09 12:50:00|
|      13113|2011-12-09 12:49:00|
|      13113|2011-12-09 12:49:00|
|      13113|2011-12-09 12:49:00|
|      13113|2011-12-09 12:49:00|
+-----------+-------------------+
only showing top 20 rows
```

In [27]:

```
test.count()
```

Out[27]:

```
797885
```

In [28]:

```
df_clean.count()
```

Out[28]:

```
797885
```

Order the Spark DataFrame by Price descending to calculate percentiles

In [29]:
```
df_clean.orderBy("Price", ascending=False).show(20)
```

```
[Stage 18:===============>                                          (1 + 3) / 4]
+-------+---------+-----------+--------+---------------+--------+-----------+--------------+
|Invoice|StockCode|Description|Quantity|    InvoiceDate|   Price|Customer ID|       Country|
+-------+---------+-----------+--------+---------------+--------+-----------+--------------+
|C556445|        M|     Manual|      -1|6/10/2011 15:31| 38970.0|      15098|United Kingdom|
|C512770|        M|     Manual|      -1|6/17/2010 16:52|25111.09|      17399|United Kingdom|
|C502262|        M|     Manual|      -1|3/23/2010 15:20| 10953.5|      12918|United Kingdom|
|C502264|        M|     Manual|      -1|3/23/2010 15:24| 10953.5|      12918|United Kingdom|
| 502263|        M|     Manual|       1|3/23/2010 15:22| 10953.5|      12918|United Kingdom|
|C522793|        M|     Manual|      -1|9/16/2010 14:53| 10468.8|      14063|United Kingdom|
| 524159|        M|     Manual|       1|9/27/2010 16:12| 10468.8|      14063|United Kingdom|
|C525398|        M|     Manual|      -1|10/5/2010 11:47| 10468.8|      14063|United Kingdom|
| 496115|        M|     Manual|       1|1/29/2010 11:04|  8985.6|      17949|United Kingdom|
|C496116|        M|     Manual|      -1|1/29/2010 11:05|  8985.6|      17949|United Kingdom|
| 551697|     POST|    POSTAGE|       1| 5/3/2011 13:46| 8142.75|      16029|United Kingdom|
|C551685|     POST|    POSTAGE|      -1| 5/3/2011 12:51| 8142.75|      16029|United Kingdom|
|C525470|        M|     Manual|      -1|10/5/2010 15:12| 7044.79|      15413|United Kingdom|
| 501768|        M|     Manual|       1|3/19/2010 11:45| 6958.17|      15760|        Norway|
|C501751|        M|     Manual|      -1|3/19/2010 11:30| 6958.17|      15760|        Norway|
|C501769|        M|     Manual|      -1|3/19/2010 11:49| 6958.17|      15760|        Norway|
| 501766|        M|     Manual|       1|3/19/2010 11:35| 6958.17|      15760|        Norway|
|C551699|        M|     Manual|      -1| 5/3/2011 14:12|  6930.0|      16029|United Kingdom|
|C505490|        M|     Manual|      -1|4/22/2010 12:55| 5876.34|      15849|United Kingdom|
|C504637|        M|     Manual|      -1|4/15/2010 14:07|  5843.7|      17017|United Kingdom|
+-------+---------+-----------+--------+---------------+--------+-----------+--------------+
only showing top 20 rows
```

In [30]:
```
monetary = df_clean.groupBy('Customer ID').sum('Price')
monetary.show()
```

```
[Stage 21:====================================================> (195 + 4) / 200]
+-----------+------------------+
|Customer ID|        sum(Price)|
+-----------+------------------+
|      15727|2446.9100000000008|
|      16503| 883.9200000000009|
|      17753|198.98999999999998|
|      15957| 356.1499999999999|
|      16386|285.56999999999994|
|      17389| 2438.029999999997|
|      12940|405.89999999999986|
|      16574|155.11999999999995|
|      13623|1051.0200000000002|
|      13832|            148.44|
|      16861|131.48000000000002|
|      13285| 539.3000000000001|
|      17679|            291.41|
|      17420|177.34000000000003|
|      15619|             13.25|
|      15790|117.75000000000001|
|      18051|113.35999999999999|
|      16339| 89.25000000000001|
|      14570|223.49999999999986|
|      13840|114.98000000000002|
+-----------+------------------+
only showing top 20 rows
```

In [31]:

```
from pyspark.sql.functions import percentile_approx

quantiles = monetary.select(percentile_approx("sum(Price)", [0.85, 0.70, 0.40], 1000000).alias("Top 15-30-
```

In [32]:

```
quantiles.show(3, False)
```

```
[Stage 25:====================================================> (193 + 4) / 200]
+-----------------------------------------------------------+
|Top 15-30-60th %                                           |
+-----------------------------------------------------------+
|[734.1300000000003, 355.20999999999987, 118.50999999999999]|
+-----------------------------------------------------------+
```

Use percentiles to assign values to each of the customer records based on the provided chart

In [33]:

```python
from pyspark.sql.functions import when

df_clean = df_clean.withColumn("Monetary", \
              when((df_clean["Price"] >= 734.1300000000006), 1)
             .when((df_clean["Price"] >= 355.21) & (df_clean["Price"] < 734.1300000000006), 2)
             .when((df_clean["Price"] >= 118.51000000000002) & (df_clean["Price"] < 355.21), 3)
             .otherwise(4))

df_clean.show()
```

```
[Stage 27:==============>                                       (1 + 3) / 4]
+-------+---------+--------------------+--------+---------------+-----+-----------+--------------+--------
+
|Invoice|StockCode|         Description|Quantity|    InvoiceDate|Price|Customer ID|       Country|Monetary
|
+-------+---------+--------------------+--------+---------------+-----+-----------+--------------+--------
+
| 489514|    21791|VINTAGE HEADS AND...|      72|12/1/2009 11:21| 1.06|      15311|United Kingdom|       4
|
| 489520|   72739B|WHITE CHOCOLATE S...|      12|12/1/2009 11:41| 1.25|      14911|          EIRE|       4
|
| 489522|    22315|200 RED + WHITE B...|       1|12/1/2009 11:45| 1.25|      15998|United Kingdom|       4
|
| 489536|    21611|SET OF 12 LILY BO...|       2|12/1/2009 12:13| 2.95|      16393|United Kingdom|       4
|
| 489561|    21816|CHRISTMAS TREE T-...|       6|12/1/2009 12:57| 1.45|      14654|United Kingdom|       4
|
| 489562|    35071|ASSORTED SANTA CH...|       1|12/1/2009 13:07| 0.85|      17998|United Kingdom|       4
|
| 489576|    22152|PLACE SETTING WHI...|      24|12/1/2009 13:38| 0.42|      15984|United Kingdom|       4
```

```
|
| 489599|    21239|    PINK  SPOTTY CUP|      16|12/1/2009 14:40| 0.85|      12758|      Portugal|       4
|
| 489658|  79323LP|LIGHT PINK CHERRY...|       6|12/1/2009 17:31| 6.75|      15485|United Kingdom|       4
|
| 489679|    22086|PAPER CHAIN KIT 5...|       6|12/2/2009 10:00| 2.95|      16163|United Kingdom|       4
|
| 489681|   85226C|BLUE PULL BACK RA...|      11|12/2/2009 10:02| 0.55|      17998|United Kingdom|       4
|
| 489723|   85231E|STRAWBERRY SCENTE...|      36|12/2/2009 10:58| 0.85|      14299|United Kingdom|       4
|
| 489765|    22315|200 RED + WHITE B...|       1|12/2/2009 11:33| 1.25|      15353|United Kingdom|       4
|
| 489766|    21975|PACK OF 60 DINOSA...|       1|12/2/2009 11:34| 0.55|      17818|United Kingdom|       4
|
| 489791|   84029G|KNITTED UNION FLA...|       1|12/2/2009 12:06| 3.75|      15542|United Kingdom|       4
|
| 489797|   72807A|SET/3 ROSE CANDLE...|       5|12/2/2009 12:19| 4.25|      15581|United Kingdom|       4
|
| 489814|    21973|SET OF 36 MUSHROO...|       2|12/2/2009 13:06| 1.45|      14669|United Kingdom|       4
|
| 489827|    22335|HEART DECORATION ...|      24|12/2/2009 13:51| 0.65|      17412|United Kingdom|       4
|
| 489866|    21877| HOME SWEET HOME MUG|       6|12/2/2009 15:04| 1.25|      16200|United Kingdom|       4
|
| 489878|    20966|SANDWICH BATH SPONGE|       1|12/2/2009 15:51| 1.25|      15989|United Kingdom|       4
|
+-------+---------+--------------------+--------+---------------+-----+-----------+--------------+--------
+
only showing top 20 rows
```

Repeating steps for frequency

In [34]:

```
df_clean.dtypes
```

```
Out[34]: [('Invoice', 'string'),
          ('StockCode', 'string'),
          ('Description', 'string'),
          ('Quantity', 'int'),
          ('InvoiceDate', 'string'),
          ('Price', 'double'),
          ('Customer ID', 'int'),
          ('Country', 'string'),
          ('Monetary', 'int')]
```

```
In [35]: df_clean = df_clean.withColumnRenamed("Customer ID", "Customer_ID")
```

```
In [36]: frequency = df_clean.withColumn("Quantity", df_clean["Quantity"].cast(DoubleType()))
```

```
In [37]: from pyspark.sql.functions import countDistinct

         freq_view = frequency.groupBy('Customer_ID').agg(countDistinct('Invoice'))
         freq_view.show()
```

```
[Stage 31:===================================================>  (192 + 4) / 200]
+-----------+--------------+
|Customer_ID|count(Invoice)|
+-----------+--------------+
|      16574|             3|
|      15727|            15|
|      17389|            77|
|      15619|             1|
|      15447|             6|
|      18051|             8|
|      13623|            15|
|      12940|             4|
|      14450|             7|
|      16503|            13|
|      15846|             1|
|      14832|             3|
|      15790|             1|
|      13285|             6|
|      17753|             5|
|      14570|             3|
|      13832|             3|
|      17679|            11|
|      16861|             6|
|      15957|             3|
+-----------+--------------+
only showing top 20 rows
```

In [38]:

```
freq_quantiles = freq_view.select(percentile_approx("count(Invoice)", [0.85, 0.70, 0.40], 1000000).alias("
freq_quantiles.show()
```

```
[Stage 36:=================================================>    (185 + 5) / 200]
+----------------+
|Top 15-30-60th %|
+----------------+
|      [13, 7, 3]|
+----------------+
```

In [39]:

```
freq_view = freq_view.withColumn("Frequency", \
                 when((freq_view["count(Invoice)"] >= 13), 1)
                .when((freq_view["count(Invoice)"] >= 7) & (freq_view["count(Invoice)"] < 13), 2)
                .when((freq_view["count(Invoice)"] >= 3) & (freq_view["count(Invoice)"] < 7), 3)
                .otherwise(4))

freq_view.show()
```

```
[Stage 40:===================================================>  (190 + 4) / 200]
```

```
+-----------+--------------+---------+
|Customer_ID|count(Invoice)|Frequency|
+-----------+--------------+---------+
|      16574|             3|        3|
|      15727|            15|        1|
|      17389|            77|        1|
|      15619|             1|        4|
|      15447|             6|        3|
|      18051|             8|        2|
|      13623|            15|        1|
|      12940|             4|        3|
|      14450|             7|        2|
|      16503|            13|        1|
|      15846|             1|        4|
|      14832|             3|        3|
|      15790|             1|        4|
|      13285|             6|        3|
|      17753|             5|        3|
|      14570|             3|        3|
|      13832|             3|        3|
|      17679|            11|        2|
|      16861|             6|        3|
|      15957|             3|        3|
+-----------+--------------+---------+
only showing top 20 rows
```

In [40]:

```
freq_view = freq_view.withColumnRenamed("Customer_ID", "ID")
freq_view = freq_view.drop("count(Invoice)")
```

In [41]:

```
freq_df = df_clean.join(freq_view, df_clean["Customer_ID"] == freq_view["ID"],"left")
```

In [42]:

```
freq_df = freq_df.drop("ID")
freq_df.show()
```

```
[Stage 46:=====================================================>(199 + 1) / 200]
+-------+---------+--------------------+--------+----------------+-----+-----------+--------------+-------
-+---------+
|Invoice|StockCode|         Description|Quantity|     InvoiceDate|Price|Customer_ID|       Country|Monetar
y|Frequency|
+-------+---------+--------------------+--------+----------------+-----+-----------+--------------+-------
-+---------+
| 513796|   85017B|ENVELOPE 50 BLOSS...|      12| 6/28/2010 15:57| 0.85|      12799|         Japan|       
4|        4|
| 513796|    22077|6 RIBBONS RUSTIC ...|      12| 6/28/2010 15:57| 1.65|      12799|         Japan|       
4|        4|
| 513796|    22509|SEWING BOX RETROS...|       1| 6/28/2010 15:57|16.95|      12799|         Japan|       
4|        4|
| 513796|   85032C|CURIOUS IMAGES GI...|       6| 6/28/2010 15:57|  2.1|      12799|         Japan|       
4|        4|
| 513796|   85049G|CHOCOLATE BOX RIB...|      12| 6/28/2010 15:57| 1.25|      12799|         Japan|       
4|        4|
| 513796|   85032A|ROMANTIC IMAGES G...|       6| 6/28/2010 15:57|  2.1|      12799|         Japan|       
4|        4|
| 513796|    22074|6 RIBBONS SHIMMER...|      12| 6/28/2010 15:57| 1.65|      12799|         Japan|       
4|        4|
| 513796|   85032B|BLOSSOM IMAGES GI...|       6| 6/28/2010 15:57|  2.1|      12799|         Japan|       
4|        4|
| 513796|    22078|RIBBON REEL LACE ...|      10| 6/28/2010 15:57|  2.1|      12799|         Japan|       
4|        4|
| 513796|    21259|VICTORIAN SEWING ...|       2| 6/28/2010 15:57| 5.95|      12799|         Japan|       
4|        4|
| 513796|   85017C|ENVELOPE 50 CURIO...|      12| 6/28/2010 15:57| 0.85|      12799|         Japan|       
4|        4|
| 513796|    85178|VICTORIAN SEWING KIT|      12| 6/28/2010 15:57| 1.25|      12799|         Japan|       
4|        4|
| 513796|    85176|SEWING SUSAN 21 N...|      12| 6/28/2010 15:57| 0.85|      12799|         Japan|       
4|        4|
| 513796|   85049D|BRIGHT BLUES RIBB...|      12| 6/28/2010 15:57| 1.25|      12799|         Japan|       
4|        4|
| 513796|    22081|RIBBON REEL FLORA...|      10| 6/28/2010 15:57| 1.65|      12799|         Japan|
4|         4|
| 566488|    22600|CHRISTMAS RETROSP...|      12| 9/13/2011 10:16| 0.85|      12940|United Kingdom|
4|         3|
| 571270|    22696| WICKER WREATH LARGE|       2|10/16/2011 12:09| 1.95|      12940|United Kingdom|
4|         3|
| 571270|    21619|4 VANILLA BOTANIC...|       3|10/16/2011 12:09| 1.25|      12940|United Kingdom|
4|         3|
| 566488|    23174|REGENCY SUGAR BOW...|       4| 9/13/2011 10:16| 4.15|      12940|United Kingdom|
4|         3|
| 571270|    23333|IVORY WICKER HEAR...|       4|10/16/2011 12:09| 1.25|      12940|United Kingdom|
4|         3|
+-------+---------+--------------------+--------+----------------+-----+-----------+--------------+-------
-+----------+
only showing top 20 rows
```

Moving on to the last part of segmentation on recency

In [43]:

```
from pyspark.sql.functions import first

dated = test.groupBy('Customer ID').agg(first('InvoiceDate'))

dated.show()
```

```
[Stage 52:===================================================>  (190 + 4) / 200]
+-----------+-------------------+
|Customer ID| first(InvoiceDate)|
+-----------+-------------------+
|      17389|2011-12-09 09:38:00|
|      15790|2011-11-29 14:53:00|
|      15619|2011-11-29 08:14:00|
|      15727|2011-11-23 12:36:00|
|      13832|2011-11-22 12:31:00|
|      13285|2011-11-16 13:19:00|
|      16386|2011-11-11 12:28:00|
|      13623|2011-11-09 12:00:00|
|      15957|2011-11-08 12:14:00|
|      12940|2011-10-24 14:04:00|
|      17420|2011-10-20 14:52:00|
|      17679|2011-10-18 07:43:00|
|      16861|2011-10-11 09:05:00|
|      16574|2011-09-29 13:39:00|
|      16503|2011-08-25 11:46:00|
|      18024|2011-07-10 12:40:00|
|      14450|2011-06-12 10:46:00|
|      14570|2011-03-04 10:58:00|
|      16339|2011-02-28 13:41:00|
|      15447|2011-01-13 11:26:00|
+-----------+-------------------+
only showing top 20 rows
```

In [44]:

```
recency_view = dated.withColumn("Recency", \
              when((dated["first(InvoiceDate)"] >= "11/15/2011"), 1)
             .when((dated["first(InvoiceDate)"] >= "9/4/2011") & (dated["first(InvoiceDate)"] < "11/14/201
             .when((dated["first(InvoiceDate)"] >= "1/4/2011") & (dated["first(InvoiceDate)"] < "9/4/2011"
             .otherwise(4))

recency_view.show()
```

```
[Stage 58:===================================================>  (190 + 4) / 200]
+-----------+-------------------+-------+
|Customer ID| first(InvoiceDate)|Recency|
+-----------+-------------------+-------+
|      17389|2011-12-09 09:38:00|      4|
|      15790|2011-11-29 14:53:00|      4|
|      15619|2011-11-29 08:14:00|      4|
|      15727|2011-11-23 12:36:00|      4|
|      13832|2011-11-22 12:31:00|      4|
|      13285|2011-11-16 13:19:00|      4|
|      16386|2011-11-11 12:28:00|      4|
|      13623|2011-11-09 12:00:00|      4|
|      15957|2011-11-08 12:14:00|      4|
|      12940|2011-10-24 14:04:00|      4|
|      17420|2011-10-20 14:52:00|      4|
|      17679|2011-10-18 07:43:00|      4|
|      16861|2011-10-11 09:05:00|      4|
|      16574|2011-09-29 13:39:00|      4|
|      16503|2011-08-25 11:46:00|      4|
|      18024|2011-07-10 12:40:00|      4|
|      14450|2011-06-12 10:46:00|      4|
|      14570|2011-03-04 10:58:00|      4|
|      16339|2011-02-28 13:41:00|      4|
|      15447|2011-01-13 11:26:00|      4|
+-----------+-------------------+-------+
only showing top 20 rows
```

In [45]:

```python
recency_df = freq_df.join(recency_view, freq_df["Customer_ID"] == recency_view["Customer ID"], "left")
recency_df.show()
```

```
[Stage 68:====================================================> (195 + 4) / 200]
+-------+---------+--------------------+--------+----------------+-----+-----------+--------------+-------
-+---------+-----------+-------------------+-------+
|Invoice|StockCode|         Description|Quantity|     InvoiceDate|Price|Customer_ID|       Country|Monetar
y|Frequency|Customer ID| first(InvoiceDate)|Recency|
+-------+---------+--------------------+--------+----------------+-----+-----------+--------------+-------
-+---------+-----------+-------------------+-------+
| 513796|   85017B|ENVELOPE 50 BLOSS...|      12| 6/28/2010 15:57| 0.85|      12799|         Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 513796|    22077|6 RIBBONS RUSTIC ...|      12| 6/28/2010 15:57| 1.65|      12799|        Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 513796|    22509|SEWING BOX RETROS...|       1| 6/28/2010 15:57|16.95|      12799|        Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 513796|   85032C|CURIOUS IMAGES GI...|       6| 6/28/2010 15:57|  2.1|      12799|        Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 513796|   85049G|CHOCOLATE BOX RIB...|      12| 6/28/2010 15:57| 1.25|      12799|        Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 513796|   85032A|ROMANTIC IMAGES G...|       6| 6/28/2010 15:57|  2.1|      12799|        Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 513796|    22074|6 RIBBONS SHIMMER...|      12| 6/28/2010 15:57| 1.65|      12799|        Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 513796|   85032B|BLOSSOM IMAGES GI...|       6| 6/28/2010 15:57|  2.1|      12799|        Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 513796|    22078|RIBBON REEL LACE ...|      10| 6/28/2010 15:57|  2.1|      12799|        Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 513796|    21259|VICTORIAN SEWING ...|       2| 6/28/2010 15:57| 5.95|      12799|        Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 513796|   85017C|ENVELOPE 50 CURIO...|      12| 6/28/2010 15:57| 0.85|      12799|        Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 513796|    85178|VICTORIAN SEWING KIT|      12| 6/28/2010 15:57| 1.25|      12799|        Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 513796|    85176|SEWING SUSAN 21 N...|      12| 6/28/2010 15:57| 0.85|      12799|        Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 513796|   85049D|BRIGHT BLUES RIBB...|      12| 6/28/2010 15:57| 1.25|      12799|        Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 513796|    22081|RIBBON REEL FLORA...|      10| 6/28/2010 15:57| 1.65|      12799|        Japan|
4|        4|      12799|2010-06-28 15:57:00|      4|
| 566488|    22600|CHRISTMAS RETROSP...|      12| 9/13/2011 10:16| 0.85|      12940|United Kingdom|
4|        3|      12940|2010-10-24 14:04:00|      4|
| 571270|    22696| WICKER WREATH LARGE|       2|10/16/2011 12:09| 1.95|      12940|United Kingdom|
4|        3|      12940|2011-10-24 14:04:00|      4|
| 571270|    21619|4 VANILLA BOTANIC...|       3|10/16/2011 12:09| 1.25|      12940|United Kingdom|
4|        3|      12940|2011-10-24 14:04:00|      4|
| 566488|    23174|REGENCY SUGAR BOW...|       4| 9/13/2011 10:16| 4.15|      12940|United Kingdom|
4|        3|      12940|2011-10-24 14:04:00|      4|
| 571270|    23333|IVORY WICKER HEAR...|       4|10/16/2011 12:09| 1.25|      12940|United Kingdom|
4|        3|      12940|2011-10-24 14:04:00|      4|
+-------+---------+--------------------+--------+----------------+-----+-----------+-------------+-------
-+---------+-----------+-------------------+-------+
only showing top 20 rows
```

In [46]:

```
recency_df.columns
```

Out[46]:

```
['Invoice',
 'StockCode',
 'Description',
 'Quantity',
 'InvoiceDate',
 'Price',
 'Customer_ID',
 'Country',
 'Monetary',
 'Frequency',
 'Customer ID',
 'first(InvoiceDate)',
 'Recency']
```

In [47]:

```
recency_df = recency_df.drop('first(InvoiceDate)', 'Customer ID')
```

In [48]:

```
RFM_matrix = recency_df.select(['Customer_ID', 'Recency', 'Frequency', 'Monetary'])
RFM_matrix.show()
```

```
[Stage 75:=========================================================> (194 + 4) / 200]
+-----------+-------+---------+--------+
|Customer_ID|Recency|Frequency|Monetary|
+-----------+-------+---------+--------+
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12940|      4|        3|       4|
|      12940|      4|        3|       4|
|      12940|      4|        3|       4|
|      12940|      4|        3|       4|
|      12940|      4|        3|       4|
+-----------+-------+---------+--------+
only showing top 20 rows
```

In [49]:

```
RFM_matrix.count()
```

Out[49]:

```
797885
```

Find the number of customers in each of the 6 categories in the table above

"Best customers" segment with RFM 111

In [50]:

```
RFM_matrix.filter((RFM_matrix['Recency'] == 1) &
                  (RFM_matrix['Frequency'] == 1) &
                  (RFM_matrix['Monetary'] == 1)).show()
```

```
+-----------+-------+---------+--------+
|Customer_ID|Recency|Frequency|Monetary|
+-----------+-------+---------+--------+
+-----------+-------+---------+--------+
```

"Loyal customers" segment with RFM X1X

In [51]:

```
RFM_matrix.filter( (RFM_matrix['Frequency'] == 1)).show()
```

```
[Stage 89:=====================================================>(197 + 3) / 200]
+-----------+-------+---------+--------+
|Customer_ID|Recency|Frequency|Monetary|
+-----------+-------+---------+--------+
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
|      13623|      4|        1|       4|
+-----------+-------+---------+--------+
only showing top 20 rows
```

"Big spenders" segment with RFM XX1

In [52]:

```
RFM_matrix.filter((RFM_matrix['Monetary'] == 1)).show()
```

```
+-----------+-------+---------+--------+
|Customer_ID|Recency|Frequency|Monetary|
+-----------+-------+---------+--------+
|      12757|      4|        1|       1|
|      12757|      4|        1|       1|
|      12757|      4|        1|       1|
|      12757|      4|        1|       1|
|      15202|      4|        2|       1|
|      15202|      4|        2|       1|
|      15202|      4|        2|       1|
|      15202|      4|        2|       1|
|      15202|      4|        2|       1|
|      15202|      4|        2|       1|
|      15202|      4|        2|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
+-----------+-------+---------+--------+
only showing top 20 rows
```

"Almost lost" segment with RFM 311

In [53]:

```
RFM_matrix.filter((RFM_matrix['Recency'] == 3) &
                  (RFM_matrix['Frequency'] == 1) &
                  (RFM_matrix['Monetary'] == 1)).show()
```

```
+-----------+-------+---------+--------+
|Customer_ID|Recency|Frequency|Monetary|
+-----------+-------+---------+--------+
+-----------+-------+---------+--------+
```

"Lost customers" segment with RFM 411

In [54]:

```
RFM_matrix.filter((RFM_matrix['Recency'] == 4) &
                  (RFM_matrix['Frequency'] == 1) &
                  (RFM_matrix['Monetary'] == 1)).show()
```

```
+-----------+-------+---------+--------+
|Customer_ID|Recency|Frequency|Monetary|
+-----------+-------+---------+--------+
|      12757|      4|        1|       1|
|      12757|      4|        1|       1|
|      12757|      4|        1|       1|
|      12757|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      14096|      4|        1|       1|
|      17448|      4|        1|       1|
|      12744|      4|        1|       1|
|      12744|      4|        1|       1|
|      12744|      4|        1|       1|
+-----------+-------+---------+--------+
only showing top 20 rows
```

"Lost cheap customers" segment with RFM 444

In [55]:

```
RFM_matrix.filter((RFM_matrix['Recency'] == 4) &
                  (RFM_matrix['Frequency'] == 4) &
                  (RFM_matrix['Monetary'] == 4)).show()
```

```
[Stage 146:===================================================> (194 + 4) / 200]
+-----------+-------+---------+--------+
|Customer_ID|Recency|Frequency|Monetary|
+-----------+-------+---------+--------+
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      12799|      4|        4|       4|
|      13289|      4|        4|       4|
|      13289|      4|        4|       4|
|      13289|      4|        4|       4|
|      13289|      4|        4|       4|
|      13289|      4|        4|       4|
+-----------+-------+---------+--------+
only showing top 20 rows
```