In [36]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

In [37]:
```
df = pd.read_csv("Comcast_telecom_complaints_data.csv")
```

In [38]:
```
df.head(3)
```

Out[38]:

	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State	Zip code	St
0	250635	Comcast Cable Internet Speeds	22-04-15	22-Apr-15	3:53:50 PM	Customer Care Call	Abingdon	Maryland	21009	Cl
1	223441	Payment disappear - service got disconnected	04-08-15	04-Aug-15	10:22:56 AM	Internet	Acworth	Georgia	30102	Cl
2	242732	Speed and Service	18-04-15	18-Apr-15	9:55:47 AM	Internet	Acworth	Georgia	30101	Cl

In [39]:
```
df["date_index"] = df["Date_month_year"] + " " + df["Time"]
```

In [40]:
```
df["date_index"] = pd.to_datetime(df["date_index"])
df["Date_month_year"] = pd.to_datetime(df["Date_month_year"])
```

In [41]:
```
df.dtypes
```

Out[41]:
```
Ticket #                           object
Customer Complaint                 object
Date                               object
Date_month_year            datetime64[ns]
Time                               object
Received Via                       object
City                               object
State                              object
Zip code                            int64
Status                             object
Filing on Behalf of Someone        object
date_index                 datetime64[ns]
dtype: object
```

In [42]:
```
df = df.set_index(df["date_index"])
```

In [43]:
```
df.head(3)
```

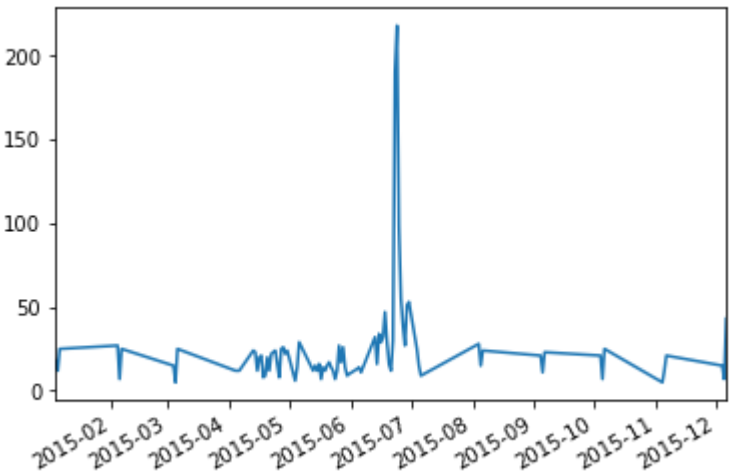
Out[43]:

date_index	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State
2015-04-22 15:53:50	250635	Comcast Cable Internet Speeds	22-04-15	2015-04-22	3:53:50 PM	Customer Care Call	Abingdon	Maryland
2015-08-04 10:22:56	223441	Payment disappear - service got disconnected	04-08-15	2015-08-04	10:22:56 AM	Internet	Acworth	Georgia
2015-04-18 09:55:47	242732	Speed and Service	18-04-15	2015-04-18	9:55:47 AM	Internet	Acworth	Georgia

In [44]:
```
df["Date_month_year"].value_counts()[:3]
```

Out[44]:
```
2015-06-24    218
2015-06-23    190
2015-06-25     98
Name: Date_month_year, dtype: int64
```

In [45]:
```
df["Date_month_year"].value_counts().plot();
```



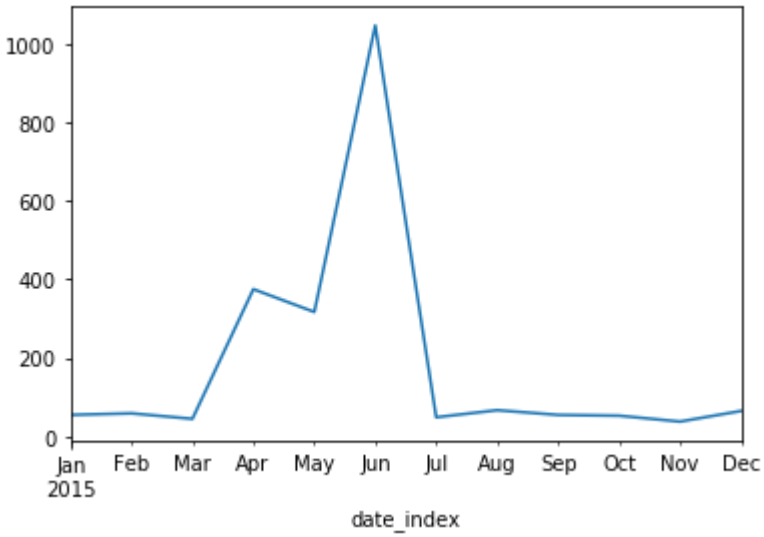
In [46]:
```
f = df.groupby(pd.Grouper(freq="M")).size()
```

In [47]:
```
f.head()
```

Out[47]:
```
date_index
2015-01-31     55
2015-02-28     59
2015-03-31     45
2015-04-30    375
2015-05-31    317
Freq: M, dtype: int64
```

In [48]:
```
df.groupby(pd.Grouper(freq="M")).size().plot()
```

Out[48]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f183083c6d8>
```



In [49]:
```
df.Status.unique()
```

Out[49]:
```
array(['Closed', 'Open', 'Solved', 'Pending'], dtype=object)
```

In [50]:
```
df["newStatus"] = ["Open" if Status=="Open" or Status=="Pending" else "Closed" fo
```

In [51]:
```
df.head(3)
```

Out[51]:

date_index	Ticket #	Customer Complaint	Date	Date_month_year	Time	Received Via	City	State
2015-04-22 15:53:50	250635	Comcast Cable Internet Speeds	22-04-15	2015-04-22	3:53:50 PM	Customer Care Call	Abingdon	Maryland
2015-08-04 10:22:56	223441	Payment disappear - service got disconnected	04-08-15	2015-08-04	10:22:56 AM	Internet	Acworth	Georgia
2015-04-18 09:55:47	242732	Speed and Service	18-04-15	2015-04-18	9:55:47 AM	Internet	Acworth	Georgia

In [52]:
```
df.groupby(["State"]).size().sort_values(ascending=False).to_frame().reset_index(
```

Out[52]:

	State	Count
0	Georgia	288
1	Florida	240
2	California	220
3	Illinois	164
4	Tennessee	143

In [53]:
```
Status_complaints = df.groupby(["State","newStatus"]).size().unstack().fillna(0)
Status_complaints
```

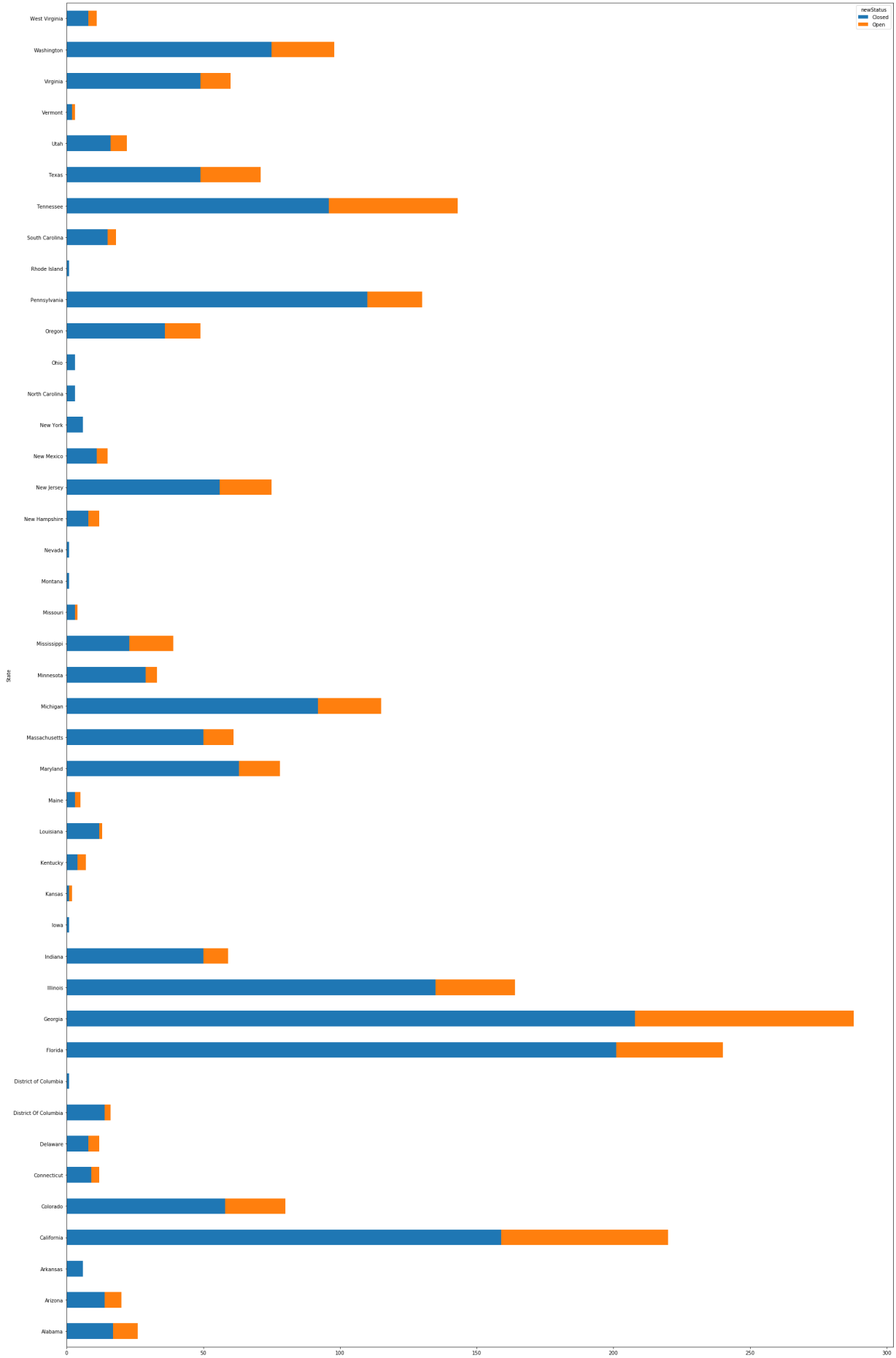
Out[53]:

State	Closed	Open
Alabama	17.0	9.0
Arizona	14.0	6.0
Arkansas	6.0	0.0
California	159.0	61.0
Colorado	58.0	22.0
Connecticut	9.0	3.0
Delaware	8.0	4.0
District Of Columbia	14.0	2.0
District of Columbia	1.0	0.0
Florida	201.0	39.0
Georgia	208.0	80.0
Illinois	135.0	29.0
Indiana	50.0	9.0
Iowa	1.0	0.0
Kansas	1.0	1.0
Kentucky	4.0	3.0
Louisiana	12.0	1.0
Maine	3.0	2.0
Maryland	63.0	15.0
Massachusetts	50.0	11.0
Michigan	92.0	23.0
Minnesota	29.0	4.0
Mississippi	23.0	16.0
Missouri	3.0	1.0
Montana	1.0	0.0
Nevada	1.0	0.0
New Hampshire	8.0	4.0
New Jersey	56.0	19.0
New Mexico	11.0	4.0
New York	6.0	0.0
North Carolina	3.0	0.0
Ohio	3.0	0.0
Oregon	36.0	13.0

newStatus	Closed	Open
State		
Pennsylvania	110.0	20.0
Rhode Island	1.0	0.0
South Carolina	15.0	3.0
Tennessee	96.0	47.0
Texas	49.0	22.0
Utah	16.0	6.0
Vermont	2.0	1.0
Virginia	49.0	11.0
Washington	75.0	23.0
West Virginia	8.0	3.0

In [54]:
```
Status_complaints.plot(kind="barh", figsize=(30,50), stacked=True)
```

Out[54]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7f18307f7940>
```


newStatus
Closed
Open
State
West Virginia
Washington
Virginia
Vermont
Utah
Texas
Tennessee
South Carolina
Rhode Island
Pennsylvania
Oregon
Ohio
North Carolina
New York
New Mexico
New Jersey
New Hampshire
Nevada
Montana
Missouri
Mississippi
Minnesota
Michigan
Massachusetts
Maryland
Maine
Louisiana
Kentucky
Kansas
Iowa
Indiana
Illinois
Georgia
Florida
District of Columbia
District Of Columbia
Delaware
Connecticut
Colorado
California
Arkansas
Arizona
Alabama
0
50
100
150
200
250
300

In [55]:
```
df.groupby(["State"]).size().sort_values(ascending=False).to_frame().reset_index(
```

Out[55]:
```
State    West Virginia
Count              288
dtype: object
```

In [56]:
```
df.groupby(["State","newStatus"]).size().unstack().fillna(0).max()
```

Out[56]:
```
newStatus
Closed    208.0
Open       80.0
dtype: float64
```

In [57]:
```
!pip install wordcloud
```

```
Requirement already satisfied: wordcloud in /opt/anaconda3/lib/python3.7/site-p
ackages (1.5.0)
Requirement already satisfied: numpy>=1.6.1 in /opt/anaconda3/lib/python3.7/sit
e-packages (from wordcloud) (1.16.3)
Requirement already satisfied: pillow in /opt/anaconda3/lib/python3.7/site-pack
ages (from wordcloud) (6.0.0)
```

In [58]:
```
from nltk.corpus import stopwords
from nltk.stem.wordnet import WordNetLemmatizer
import string

stop = set(stopwords.words('english'))
exclude = set(string.punctuation)
lemma = WordNetLemmatizer()
```

In [59]:
```
def clean(doc):
    stop_free = " ".join([i for i in doc.lower().split() if i not in stop])
    punc_free = "".join([ch for ch in stop_free if ch not in exclude])
    normalised = " ".join(lemma.lemmatize(word) for word in punc_free.split())
    return normalised
```

In [60]:
```
doc_complete = df["Customer Complaint"].tolist()
doc_clean = [clean(doc).split() for doc in doc_complete]
```

In [61]:
```
import gensim
from gensim import corpora
```

In [62]:
```
dictionary = corpora.Dictionary(doc_clean)
print(dictionary)
```

```
Dictionary(1416 unique tokens: ['cable', 'comcast', 'internet', 'speed', 'disap
pear']...)
```

In [63]:
```
doc_term_matrix = [dictionary.doc2bow(doc) for doc in doc_clean]
doc_term_matrix
```

Out[63]:
```
[[(0, 1), (1, 1), (2, 1), (3, 1)],
 [(4, 1), (5, 1), (6, 1), (7, 1), (8, 1)],
 [(3, 1), (8, 1)],
 [(1, 1), (9, 1), (10, 1), (11, 1), (12, 1), (13, 1), (14, 1), (15, 1)],
 [(1, 1), (8, 1), (16, 1), (17, 1)],
 [(18, 1), (19, 1), (20, 1), (21, 1), (22, 1), (23, 1), (24, 1)],
 [(8, 1), (10, 1), (20, 1), (25, 1), (26, 1)],
 [(1, 1), (8, 1), (27, 1), (28, 1), (29, 1), (30, 1)],
 [(1, 1), (31, 1), (32, 1)],
 [(1, 1), (33, 1), (34, 1), (35, 1), (36, 1)],
 [(5, 1), (8, 1), (37, 1), (38, 1)],
 [(39, 1), (40, 1), (41, 1), (42, 1), (43, 1), (44, 1)],
 [(1, 1),
  (2, 1),
  (45, 1),
  (46, 1),
  (47, 1),
  (48, 1),
  (49, 1),
```

In [64]:
```
from gensim.models import LdaModel
```

In [65]:
```
Num_Topic = 9
ldamodel = LdaModel(doc_term_matrix, num_topics= Num_Topic, id2word= dictionary,
```

In [66]:
```
topics = ldamodel.show_topics()
for topic in topics:
    print(topic)
    print()
```

```
(0, '0.144*"billing" + 0.084*"service" + 0.074*"practice" + 0.066*"unfair" + 0.
053*"internet" + 0.050*"pricing" + 0.047*"poor" + 0.024*"outage" + 0.022*"monop
olistic" + 0.019*"incorrect"')

(1, '0.069*"fee" + 0.037*"equipment" + 0.036*"comcast" + 0.029*"xfinitycomcast"
+ 0.026*"charge" + 0.024*"asking" + 0.019*"throttle" + 0.018*"bandwidth" + 0.01
8*"broadband" + 0.018*"day"')

(2, '0.106*"comcast" + 0.041*"service" + 0.026*"bill" + 0.025*"month" + 0.022
*"sale" + 0.021*"deceptive" + 0.021*"access" + 0.020*"account" + 0.019*"chargin
g" + 0.017*"without"')

(3, '0.087*"price" + 0.058*"false" + 0.045*"connection" + 0.040*"paying" + 0.03
5*"switch" + 0.024*"bait" + 0.024*"unreliable" + 0.022*"low" + 0.020*"home" +
0.019*"high"')

(4, '0.041*"comcast" + 0.040*"speed" + 0.029*"credit" + 0.024*"payment" + 0.023
*"promised" + 0.023*"service" + 0.021*"bill" + 0.021*"charge" + 0.020*"charged"
+ 0.020*"slowing"')

(5, '0.275*"comcast" + 0.125*"service" + 0.078*"internet" + 0.063*"billing" +
0.047*"issue" + 0.023*"customer" + 0.020*"xfinity" + 0.018*"charge" + 0.011*"fr
audulent" + 0.010*"failure"')

(6, '0.193*"internet" + 0.143*"speed" + 0.055*"slow" + 0.053*"comcast" + 0.019
*"connectivity" + 0.014*"issue" + 0.013*"business" + 0.012*"call" + 0.010*"adve
rtised" + 0.010*"charge"')

(7, '0.140*"comcast" + 0.126*"data" + 0.102*"cap" + 0.045*"complaint" + 0.033
*"service" + 0.030*"internet" + 0.024*"usage" + 0.016*"customer" + 0.012*"charg
e" + 0.012*"help"')

(8, '0.124*"comcast" + 0.063*"service" + 0.061*"internet" + 0.042*"bill" + 0.03
7*"throttling" + 0.036*"cable" + 0.023*"problem" + 0.022*"without" + 0.022*"com
castxfinity" + 0.014*"cramming"')
```

In [67]:
```
word_dict = {}
for i in range(Num_Topic):
    words = ldamodel.show_topic(i, topn =20)
    word_dict["Topic # " + "{}".format(i)] = [i[0] for i in words]
```

In [68]:
```
pd.DataFrame(word_dict)
```

Out[68]:

	Topic # 0	Topic # 1	Topic # 2	Topic # 3	Topic # 4	Topic # 5	Topic # 6	Topic #
0	billing	fee	comcast	price	comcast	comcast	internet	comca
1	service	equipment	service	false	speed	service	speed	da
2	practice	comcast	bill	connection	credit	internet	slow	ca
3	unfair	xfinitycomcast	month	paying	payment	billing	comcast	complai
4	internet	charge	sale	switch	promised	issue	connectivity	servic
5	pricing	asking	deceptive	bait	service	customer	issue	intern
6	poor	throttle	access	unreliable	bill	xfinity	business	usag
7	outage	bandwidth	account	low	charge	charge	call	custom
8	monopolistic	broadband	charging	home	charged	fraudulent	advertised	charg
9	incorrect	day	without	high	slowing	failure	charge	he
10	complaint	improper	refund	service	unauthorized	contract	scam	
11	option	last	wont	speed	throttled	terrible	disconnection	lim
12	quality	deceptive	email	monopoly	change	provide	mbps	ye
13	claim	sold	back	xfinity	billed	shitty	promotion	fe
14	provided	violation	12	advertising	hbogo	300gb	time	mode
15	provider	extortion	mb	system	week	monopoly	overage	month
16	inability	comcasts	pay	contract	download	refusal	much	xfini
17	xfinity	cable	one	security	device	regarding	complaint	intermitte
18	get	advertising	practice	information	inconsistent	lack	consistently	contra
19	signal	unreturned	10	supervisor	every	still	refusing	da

In [ ]:

In [ ]:

In [ ]: