**Load the data**

df = pd.read\_csv("Comcast\_telecom\_complaints\_data.csv")

**Show the 5 datas**

df.head(5)

**Change the Date\_Month\_Year format and create the column date index**

df["date\_index"] = df["Date\_month\_year"] + " " + df["Time"]

df["date\_index"] = pd.to\_datetime(df["date\_index"])

df["Date\_month\_year"] = pd.to\_datetime(df["Date\_month\_year"])

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

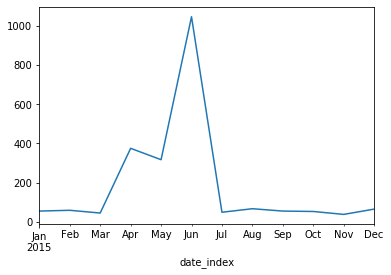
**Set the index to date index and count the Customer Complaint**

df = df.set\_index(df["date\_index"])

**Graph for monthly Customer Complaint**

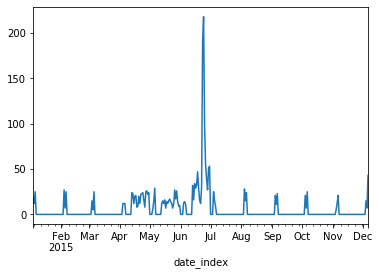
#monthly chart

df.groupby(pd.Grouper(freq="M")).size().plot()



**Graph for monthly Customer Complaint**

f = df.groupby(pd.Grouper(freq="D")).size()



**Frequency of complaint details**

# frequency of complaint types.

df.groupby(["Customer Complaint"]).size().sort\_values(ascending=False).to\_frame().reset\_index().rename({0: "Count"}, axis=1)

**Find Maximum complaints**

# maximum

df.groupby(["Customer Complaint"]).size().sort\_values(ascending=False).to\_frame().reset\_index().rename({0: "Count"}, axis=1).max()

**Create a new categorical variable with value as Open and Closed. Open & Pending is to be categorized as Open and Closed & Solved is to be categorized as Closed.**

df["newStatus"] = ["Open" if Status=="Open" or Status=="Pending" else "Closed" for Status in df["Status"]]

**state wise status of complaints count**

df.groupby(["State"]).size().sort\_values(ascending=False).to\_frame().reset\_index().rename({0: "Count"}, axis=1)

**state wise status of complaints in a stacked bar chart**

Status\_complaints.plot(kind="barh", figsize=(30,50), stacked=True)

**state has the maximum complaints**

df.groupby(["State"]).size().sort\_values(ascending=False).to\_frame().reset\_index().rename({0: "Count"}, axis=1).max()

**state has the maximum complaints and status**

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

clear the puctiation marks and separate the words

stop = set(stopwords.words('english'))

exclude = set(string.punctuation)

lemma = WordNetLemmatizer()

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

dictionary = corpora.Dictionary(doc\_clean)

print(dictionary)

**Provide the percentage of complaints resolved till date, which were received through the Internet and customer care calls.**

Num\_Topic = 9

ldamodel = LdaModel(doc\_term\_matrix, num\_topics= Num\_Topic, id2word= dictionary, passes= 30)

topics = ldamodel.show\_topics()

for topic in topics:

print(topic)

print()

word\_dict = {}

for i in range(Num\_Topic):

words = ldamodel.show\_topic(i,topn =30)

word\_dict["Topic # " + "{}".format(i)] = [i[0] for i in words]

pd.DataFrame(word\_dict)