#!/usr/bin/env python

# coding: utf-8

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

# # Import data into Python environment

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

df.head()

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"])

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

df.head()

df['Date\_month\_year'].value\_counts()

# ## Trend chart for the number of complaints at daily granularity levels

df["Date\_month\_year"].value\_counts().plot(x="Date\_month\_year");

# ## Trend chart for the number of complaints at monthly granularity levels

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

f.plot();

df["Final\_Status"] = df["Status"].apply(lambda status: "Open" if status in ["Open", "Pending"] else "Closed")

df.head(3)

df.groupby(["State"]).size()

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

status\_complaint = df.groupby(["State", "Final\_Status"]).size().unstack().fillna(0)

status\_complaint.plot.bar(stacked=True, figsize=(30, 10))

status\_complaint.max()

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

df.head()

# df.groupby(["Final\_Status"]).mean().unstack().reset\_index().fillna(0)

df.groupby('State').Final\_Status.apply(lambda x: (x == 'Open').mean()).to\_frame().reset\_index()

df.groupby('State').Final\_Status.apply(lambda x: (x == 'Open').mean()).to\_frame().reset\_index().max()

df["isSolved"] = df["Final\_Status"].apply(lambda status: 1 if status == 'Open' else 0 )

df.reset\_index(drop=True, inplace=True)

df.head(2)

percentage\_df = df.groupby(["State", "isSolved"]).size().unstack().reset\_index(drop=True).rename({0:"Total"}, axis=1).rename({1:"openStatus"}, axis=1).fillna(0)

percentage\_df.head(2)

percentage\_df['percentage'] = (percentage\_df["openStatus"]\*100)/percentage\_df["Total"]

percentage\_df.reset\_index(drop=True)

percentage\_df["percentage"].to\_frame().max()

percentage\_df.head()