This code is developed in google colab IDE with python environment.

To run this code, Simply, load the 5 csv files on your google drive, change the folder location accordingly, and copy paste the code and run it.

**Python Source Code**

#Break down to each months

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

from google.colab import drive

drive.mount('/content/drive')

print('\r')

def readcsvfile(folder,filenumber):

  #construct file name and location

  filename=str(folder)+'00'+str(filenumber)+'.csv'

  datafile = pd.read\_csv(filename)

  print('file ',filename,' loaded')

  # check number of rows and fix if needed

  a=len(datafile)

  if a % 3 == 0:

      print('data file has consistent number of rows')

      print('\r')

  else:

      print(f"{a} is not divisible by 3.")

      r = a % 3

      datafile = datafile[:-r]

      print('\r')

  return datafile

#breaking data frame into invidual signals

def BreakByMonth(df):

  data = {'t\_PWAvaialbleChargePower': (df.iloc[2::3,1]/1000).values,

          'PWAvaialbleChargePower': (df.iloc[2::3,3]).values,

          't\_PWRemainingEnergy': (df.iloc[1::3,1]/1000).values,

          'PWRemainingEnergy': (df.iloc[1::3,3]).values,

          't\_PWPackEnergyAvailable': (df.iloc[0::3,1]/1000).values,

          'PWPackEnergyAvailable': (df.iloc[0::3,3]).values,

          }

  dft = pd.DataFrame(data)

  dft['Date'] = pd.to\_datetime(dft['t\_PWAvaialbleChargePower'], unit='s')

  dft.set\_index('Date', inplace=True)

  array = [group for \_, group in dft.groupby(pd.Grouper(freq='M'))] #data broken into monthly based on ACP

  #print(size(monthlyarray))

  #print(monthlyarray[0]['PWAvaialbleChargePower'])

  return array

def monthly\_analysis(arrays):

    months = []

    avl1 = []

    avl2 = []

    for indx, sub\_array in enumerate(arrays):

        if not sub\_array.empty:

            first\_date = sub\_array.index[0]  # Get the first date in the subarray

            month = first\_date.strftime('%b')  # Format the month as

            months.append(month)

            acp=arrays[indx]['PWAvaialbleChargePower']

            er=arrays[indx]['PWRemainingEnergy']

            pea=arrays[indx]['PWPackEnergyAvailable']

            print('Monthly availability analysis  : ',first\_date.strftime('%B %Y'))

            cond1 = np.where(acp >= 3300)

            avlc1 = size(cond1)/len(acp)

            avl1.append(avlc1)

            print("Charge Power Availability                = {:.2%}".format(avlc1))

            cond2 = np.where((acp >= 3300) & (er/pea <= 0.9))

            avlc2 = size(cond2)/len(acp)

            avl2.append(avlc2)

            print("Charge Power Availability while SOE<90%  = {:.2%}".format(avlc2))

            print('---')

    return months, avl1, avl2

def plotmaker(i,xs,y1s,y2s):

  # Create a bar plot

  values1=[(x \* 100 )  for x in y1s]

  values2=[(x \* 100 )  for x in y2s]

  x1 = np.arange(len(xs))

  barWidth = 0.1

  a=['Aug','Sep','Oct','Nov','Dec']

  offset= a.index(xs[0])

  xpos = [x + (i)\*barWidth + offset for x in x1]

  #plt.figure(figsize=(10, 4))

  plt.subplot(1, 2, 1)  # 1 row, 2 columns, first subplot

  plt.bar(xpos, values1, width = barWidth, label ='Pack'+str(i+1))

  plt.xticks([r + barWidth for r in range(5)],

        ['Aug', 'Sep', 'Oct', 'Nov', 'Dec'])

  plt.xlabel("Month")

  plt.ylabel("Percentage %")

  plt.title("Charge Power Availability")

  plt.legend()

  plt.subplot(1, 2, 2)  # 1 row, 2 columns, first subplot

  plt.bar(xpos, values2, width = barWidth, label ='Pack'+str(i+1))

  plt.xticks([r + barWidth for r in range(5)],

        ['Aug', 'Sep', 'Oct', 'Nov', 'Dec'])

  plt.xlabel("Month")

  #plt.ylabel("Percentage %")

  plt.title("Charge Power Availability SOE<90%")

  plt.legend()

  return

from numpy.ma.core import size

from pandas.core.arrays.arrow import array

for i in range(5):

  DFread = readcsvfile('/content/drive/MyDrive/Tesla/',i+1)

  monthlyarray = BreakByMonth(DFread)

  print('Analysis of Battery Serial Number',str(i+1))

  months,avl1,avl2 = monthly\_analysis(monthlyarray)

  print('--'\*100)

  plotmaker(i,months,avl1,avl2)

**Results**

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

file /content/drive/MyDrive/Tesla/001.csv loaded

data file has consistent number of rows

Analysis of Battery Serial Number 1

Monthly availability analysis : August 2017

Charge Power Availability = 85.11%

Charge Power Availability while SOE<90% = 80.57%

---

Monthly availability analysis : September 2017

Charge Power Availability = 90.78%

Charge Power Availability while SOE<90% = 87.41%

---

Monthly availability analysis : October 2017

Charge Power Availability = 96.93%

Charge Power Availability while SOE<90% = 96.55%

---

Monthly availability analysis : November 2017

Charge Power Availability = 73.10%

Charge Power Availability while SOE<90% = 73.10%

---

Monthly availability analysis : December 2017

Charge Power Availability = 41.16%

Charge Power Availability while SOE<90% = 41.16%

---

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

file /content/drive/MyDrive/Tesla/002.csv loaded

29470 is not divisible by 3.

Analysis of Battery Serial Number 2

Monthly availability analysis : September 2017

Charge Power Availability = 87.47%

Charge Power Availability while SOE<90% = 84.39%

---

Monthly availability analysis : October 2017

Charge Power Availability = 94.76%

Charge Power Availability while SOE<90% = 92.82%

---

Monthly availability analysis : November 2017

Charge Power Availability = 100.00%

Charge Power Availability while SOE<90% = 100.00%

---

Monthly availability analysis : December 2017

Charge Power Availability = 86.89%

Charge Power Availability while SOE<90% = 83.57%

---

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

file /content/drive/MyDrive/Tesla/003.csv loaded

data file has consistant number of rows

Analysis of Battery Serial Number 3

Monthly availability analysis : September 2017

Charge Power Availability = 84.64%

Charge Power Availability while SOE<90% = 81.39%

---

Monthly availability analysis : October 2017

Charge Power Availability = 92.92%

Charge Power Availability while SOE<90% = 91.49%

---

Monthly availability analysis : November 2017

Charge Power Availability = 100.00%

Charge Power Availability while SOE<90% = 100.00%

---

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

file /content/drive/MyDrive/Tesla/004.csv loaded

48736 is not divisible by 3.

Analysis of Battery Serial Number 4

Monthly availability analysis : September 2017

Charge Power Availability = 86.24%

Charge Power Availability while SOE<90% = 82.63%

---

Monthly availability analysis : October 2017

Charge Power Availability = 85.33%

Charge Power Availability while SOE<90% = 81.78%

---

Monthly availability analysis : November 2017

Charge Power Availability = 91.94%

Charge Power Availability while SOE<90% = 90.09%

---

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

file /content/drive/MyDrive/Tesla/005.csv loaded

data file has consistant number of rows

Analysis of Battery Serial Number 5

Monthly availability analysis : September 2017

Charge Power Availability = 80.39%

Charge Power Availability while SOE<90% = 78.59%

---

--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

A graph of different colors

Description automatically generated with medium confidence