

```

#!/usr/bin/env python3
import numpy as np
from sklearn.linear_model import LinearRegression

# Call script as Cubic.py
# Copyleft Ed Egan, 2022.
# Code assumes that file is tab-delimited with header row. Cols are:
# cbsa_id    agg_count    num_obs    r_sq

INPUT_FILENAME = r'E:\projects\agglomeration\MaxR2Data95.txt'
records={}
master={}
with open(INPUT_FILENAME) as f:
    next(f) #loose the header row
    for line in f:
        line=line.rstrip('\r\n')
        (cbsaid,h,n,rsq) = line.split('\t')
        cbsaid=int(cbsaid)
        h=int(h)
        n=int(n)
        rsq=float(rsq)
        master[cbsaid]=1
        if n>=15:
            if cbsaid not in records:
                records[cbsaid]={}
            records[cbsaid][h]=[h,n,rsq]

results={}
for cbsaid in dict.keys(records):
    results[cbsaid]={}
    h_arr = dict.keys(records[cbsaid])
    results[cbsaid]['h_min']=min(h_arr)
    results[cbsaid]['h_max']=max(h_arr)
    results[cbsaid]['h_len']=len(h_arr)

for cbsaid in dict.keys(records):
    dataarray=[]
    for h in dict.keys(records[cbsaid]):
        dataarray.append(records[cbsaid][h])
    data=np.asmatrix(np.array(dataarray))
    a=data[:,0]
    rsq=data[:,2]
    if np.size(rsq) >= 4:
        results[cbsaid]['method']='fit'
        X=np.hstack((a,np.power(a,2),np.power(a,3)))
        reg = LinearRegression().fit(X, rsq)
        yhat=reg.predict(X)
    else:
        results[cbsaid]['method']='grab'
        yhat=rsq
    idx_solve=np.argmax(yhat)
    idx_pick=np.argmax(rsq)
    results[cbsaid]['h_solve']=int(a[idx_solve,0])
    results[cbsaid]['h_pick']=int(a[idx_pick,0])

f = open("cubicresults95.txt", "w")
f.write("cbsaid\th_solve\th_pick\tmethod\tmin\tmax\n")
for cbsaid in dict.keys(master):
    if cbsaid in results:
        f.write(str(cbsaid)+"\t"+str(results[cbsaid]
        ['h_solve'])+"\t"+str(results[cbsaid]
        ['h_pick'])+"\t"+str(results[cbsaid]
        ['method'])+"\t"+str(results[cbsaid]
        ['h_min'])+"\t"+str(results[cbsaid]
        ['h_max']))+"\n")

```

