```python
###### gerekli kutuphaneler ###### - LoadShifting + Winter
import pandas as pd
from pysolar import radiation
from pysolar.solar import *
import openpyxl
import datetime
import scipy
import numpy as np
from matplotlib import pyplot as plt

#ev gunluk tuketimi #12kwh
#batarya kap. 6kwh
###### excel okuma ######

excel = pd.read_excel("degerler.xlsx").iloc[:, 1:]

buzdolabi = excel['buzdolabi']
camasir = excel['camasir']
bulasik = excel['bulasik']
camasirkur = excel['camasirkur']
d_dondurucu = excel['d_dondurucu']
elksup = excel['elksup']
termosifon = excel['termosifon']
tv = excel['tv']
bilgisayar = excel['bilgisayar']
firin = excel['firin']
aydinlatma = excel['aydinlatma']
mikrodalga = excel['mikrodalga']
kettle = excel['kettle']
davlumbaz = excel['davlumbaz']
sackuru = excel['sackuru']
klima = excel['klima']
esarj = excel['esarj']
sarjaleti = excel['sarjaleti']


###### zaman ve yuk tanimlama ######
zaman1 = np.arange(1,25)
# a = np.ones((25), dtype=np.int16)
# b = np.zeros((25), dtype=np.int16)
```

```
41 # x=17
42 # y=20
43 # b[int(x):int(y)]=2
44 # totaltime=a+b
45
46 ########## Solar Radyasyon Verisi Alma - SolarPy
47 latitude_deg = 36.884 # positive in the northern
   hemisphere
48 longitude_deg = -30.704 # negative reckoning west
   from prime meridian in Antalya, Turkey
49 date = datetime.datetime(2019, 12, 1, 10, 30, tzinfo=
   datetime.timezone.utc)  ###20 Aralık saat 10:30 icin
   solar radiation degeri
50 altitude_deg = get_altitude(latitude_deg,
   longitude_deg, date)
51 solarradyasyon= radiation.get_radiation_direct(date,
   altitude_deg)
52 ########## Solar Radyasyon Verisi Alma - SolarPy
53
54 ########## PV Panel Enerji Uretim Hesabi
55 #  E = A * r * H * PR
56
57 #E = Energy (kWh)
58 #A = Total solar panel Area (m2)
59 #r = solar panel yield or efficiency(%)
60 #H = Annual average solar radiation on tilted panels
    (shadings not included)
61 #PR = Performance ratio, coefficient for losses (
   range between 0.5 and 0.9, default value = 0.75)
62 ##########
63 a= 17.94  #Boyut: 1956 × 992 × 40mm # Adet : 10
64 r= 0.15
65 h= solarradyasyon
66 pr= 0.75
67 pv_ye= a*r*h*pr   ###Yıllık Üretim
68 pv_e = pv_ye/365  ###Günlük üretim
69
70 ##Batarya Enerji Depolama
71 bat_time = np.ones(24, dtype = np.float16)
72 bat_time[0]=1
73 bat_time[1:8]=0
```

```
 74 bat_time[8]=pv_e*(0.035)
 75 bat_time[9]= pv_e * 0.0864
 76 bat_time[10]=pv_e*(0.108)
 77 bat_time[11]=pv_e*(0.103)
 78 bat_time[12]=pv_e*(0.126)
 79 bat_time[13]=pv_e*(0.118)
 80 bat_time[14]=pv_e*(0.110)
 81 bat_time[15]=pv_e*(0.0960)
 82 bat_time[16]=pv_e*(0.0930)
 83 bat_time[17]=pv_e*(0.0819)
 84 bat_time[18]=pv_e*(0.0351)
 85 bat_time[18:24]=0
 86 ##Batarya Enerji Depolama
 87 o=0
 88 ##Şebekeden Çekilen Enerji Miktarı
 89 totaltime = np.zeros(24, dtype = np.float32)
 90 totaltime_1 = np.zeros(24, dtype = np.float32)
 91 totaltime_2 = np.zeros(24, dtype = np.float32)
 92 j = np.zeros(24, dtype = np.float32)
 93 for i in range(24):
 94     totaltime_1[o]= buzdolabi[o]+d_donduru cu[o]+
    elksup[o]+termosifon[o]+tv[o]+bilgisayar[o]+firin[o
    ]+aydinlatma[o]+mikrodalga[o]+kettle[o]+davlumbaz[o
    ]+sackuru[o]+klima[o]
 95     totaltime_2[o]= esarj[o]+sarjaleti[o]+camasir[o
    ]+bulasik[o]+camasirkur[o]
 96     j[o] = esarj[o] + sarjaleti[o] + camasir[o] +
    bulasik[o] + camasirkur[o]
 97     o=o+1
 98
 99 totaltime= totaltime_1+totaltime_2
100 threshold=np.sum(totaltime)/12
101 ##Şebekeden Çekilen Enerji Miktarı
102
103 ###toplamyuk
104 pb= input("Tuketimin artmaya basladigi saati giriniz
    : ")
105 p_baslangic=int(pb)
106 p_surec=input("Bu tuketim kac saat suruyor?: ")
107 p_bitis=int(p_baslangic)+int(p_surec)  ###Tuketimin
    azaldigi saat (Uyku saati)
```

```
108 sayac=0
109 battery=1
110 enerjisbt=6
111 enerjisatis=0
112 bsay=8
113 for i in range(10):
114     battery = battery+(bat_time[bsay])
115     bsay= bsay + 1
116     if battery > 6:
117         enerjisatis = battery-enerjisbt
118         battery = 6
119     elif battery > 6 and enerjisatis>0:
120         enerjisatis= enerjisatis+battery-enerjisbt
121     elif battery == 6:
122         battery = 6
123 
124 a = np.ones(24, dtype=int)
125 #index olustur
126 for i in range(24):
127     if totaltime[i] > threshold and i==18 or i==19
    or i==20 or i==21 or i==22:
128         sayac = sayac+1
129 
130 #for i in range(int(p_surec)+1):
131  #   index = totaltime[int(p_baslangic)+i:int(
    p_bitis)]
132     # if not np.all(index == 1):   ##indexten çıkıp
    totaltime_s'imizi düz modüle eşitleyelim ve bir
    sayac ciktisi alalim
133   #  if np.sum(index) != int(p_surec) - i:
134    #     sayac = sayac + 1
135     #    totaltime_s = a
136 
137 ########sayac 24'den büyük olma problemini
    matlabdeki 24den büyük olma durumunda
138 # direk 3 arkaya atması emrini vererek çözdüm #
    değilse de sayac ve 3 saatlik triple tariff arasında
     yazdırdm
139 x = int(p_baslangic)
140 y = int(p_bitis)
141 totaltime_3 = j
```

```
142 if sayac != 0:
143     if y + sayac > 19:
144         totaltime_3[y - sayac - int(p_surec)-1: y -
    sayac] = totaltime_3[x-1 : y]
145         totaltime_3[x:y] = totaltime_2[y - sayac -
    int(p_surec) -1: y - sayac-1]
146
147 shifted=totaltime_3+totaltime_1
148 bat_time[0]=0
149 plt.title("Enerji Uretim Grafigi")
150 plt.plot(zaman1, bat_time, color = 'royalblue',
    label= 'kwH')
151 plt.xlabel("Saat")
152 plt.ylabel("YÜk")
153 plt.legend()
154 plt.show()
155 bat_time[0]=1
156
157
158 aylikkar=enerjisatis*0.26*30
159 toplamtuketim=np.sum(totaltime)
160 shiftedtuketim=np.sum(shifted)
161 pvuretim=np.sum(bat_time)
162
163 notshifted_kwh=totaltime/1000
164 shifted_kwh=shifted/1000
165 notshifted_kwhtotal=notshifted_kwh
166 shifted_kwhtotal=shifted_kwh
167 t=0
168 threshold2=threshold/1000
169
170 for i in range(24):
171     if shifted_kwhtotal[t] > threshold2 and float(
    battery) > 3:
172         shifted_kwhtotal[t] = shifted_kwhtotal[t] -
    float(battery)/3
173         battery= battery-battery/3
174     elif shifted_kwhtotal[t] > threshold2 and float(
    battery) <= 2:
175         shifted_kwhtotal[t] = shifted_kwhtotal[t] -
    float(battery)
```

```
176             battery = 0
177         t=t+1
178 plt.title("LoadShifting Grafigi (GES sistemi ile)")
179 plt.bar(zaman1, shifted_kwhtotal, color = 'tab:red'
    , align = 'center', label= 'Shifted')
180 plt.bar(zaman1, notshifted_kwhtotal, color = 'tab:
    blue', align = 'center', label= 'Not Shifted')
181 plt.ylabel("Yük (kwH)")
182 plt.legend()
183 plt.show()
184
185 plt.title("LoadShiftingden Önce")
186 plt.bar(zaman1, notshifted_kwhtotal, color = 'tab:
    blue', align = 'center', label= 'Not Shifted')
187 plt.ylabel("Yük (kwH)")
188 plt.legend()
189 plt.show()
190
191 plt.title("LoadShiftingden Sonra (GES sistemi ile)")
192 plt.bar(zaman1, shifted_kwhtotal, color = 'tab:red'
    , align = 'center', label= 'Shifted')
193 plt.ylabel("Yük (kwH)")
194 plt.legend()
195 plt.show()
196
197 bat_time[0]=0
198 plt.title("Uretim/Tuketim")
199 plt.plot(zaman1, bat_time, color = 'royalblue',
    label= 'kwH')
200 plt.plot(zaman1, notshifted_kwhtotal, color = 'tab:
    red', label= 'kwH')
201 plt.xlabel("Saat")
202 plt.ylabel("Yük (kwH)")
203 plt.legend()
204 plt.show()
205 bat_time[0]=1
206
207 enerjimaaliyeti= np.ones(24, dtype = np.float16)
208 enerjimaaliyeti[0:18]=enerjimaaliyeti[0:18]*0.92
209 enerjimaaliyeti[18:24]=enerjimaaliyeti[18:24]*2.76
210
```

```
211
212 plt.title("Saatlik Enerji Maaliyeti")
213 plt.plot(zaman1, shifted_kwhtotal[0:24]*
    enerjimaaliyeti[0:24], color = 'royalblue', label= '
    GES + LoadShifting')
214 plt.plot(zaman1, notshifted_kwhtotal[0:24]*
    enerjimaaliyeti[0:24], color = 'tab:red', label= '
    GES YOK SHIFTING YOK')
215 plt.xlabel("Saat")
216 plt.ylabel("Turk Lirasi")
217 plt.legend()
218 plt.show()
219
220 ucret_nshifted=np.sum(notshifted_kwhtotal[0:24]*
    enerjimaaliyeti[0:24]) ## GES YOK SHIFTING YOK
221 ucret_shifted=np.sum(shifted_kwhtotal[0:24]*
    enerjimaaliyeti[0:24])  ## GES ile birlikte
222
223
224 kaydirilanyuk= np.sum(j[0:24])/1000
225 print("LoadShifting + GES Fatura: {}TL" .format(
    ucret_shifted))
226 print("Normal ödenmesi gereken fatura: {}tl" .format
    (ucret_nshifted))
227 print("Toplam kaydırılan yük: {}kwH" .format(
    kaydirilanyuk))
228 print("Loadshifting uygulanmadan önce: {}".format(
    notshifted_kwhtotal))
229 print("Loadshifting uygulandıktan sonra: {}".format(
    shifted_kwhtotal))
230
231
232
233
234
235 #####Koray Göksu 18012116
236
```