In [1]:

```
#AYUSH KUMAR
```

In [2]:

```
#importing all the libraries
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
from datetime import datetime
import matplotlib as mpl
import matplotlib.dates as mdates
from matplotlib.lines import Line2D
from matplotlib.patches import Patch
```

In [3]:

```
#reading the dataframe
df=pd.read_csv("Assignment_Dataset.csv")
df.shape
```

Out[3]:

```
(982, 3)
```

In [4]:

```
dfc=df
dfc.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 982 entries, 0 to 981
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Date    982 non-null    object 
 1   GHI     982 non-null    float64
 2   PR      982 non-null    float64
dtypes: float64(2), object(1)
memory usage: 23.1+ KB
```

In [5]:

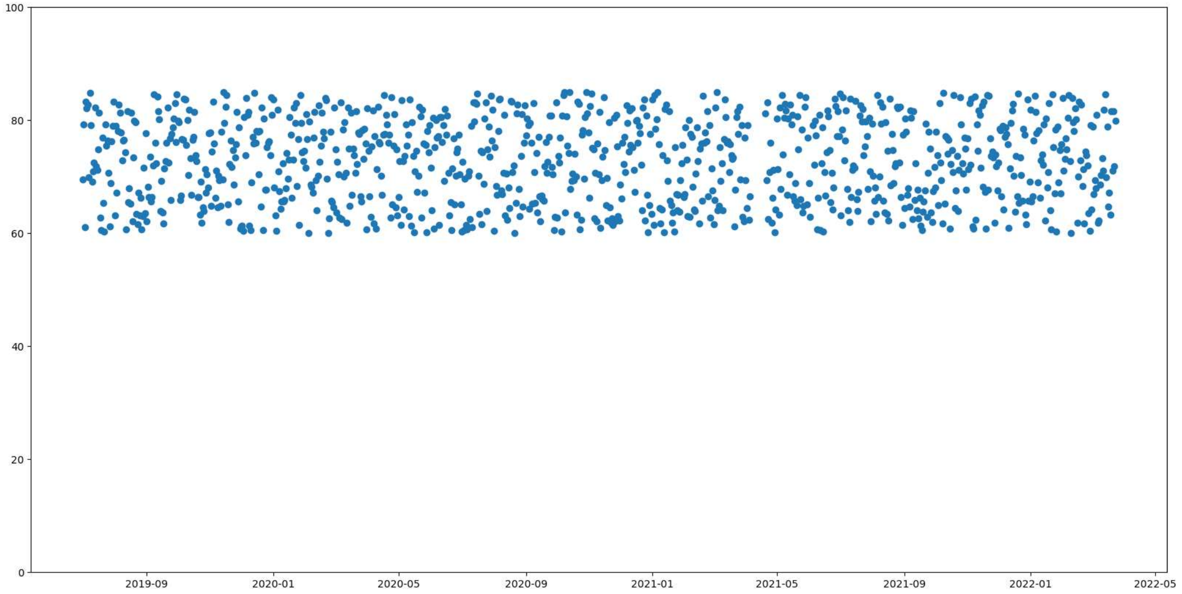
```
#converting to datetime type
dfc['Date'] = pd.to_datetime(dfc['Date'])
dfc = dfc.sort_values(by = 'Date')
dfc.tail()
```

Out[5]:

| | Date | GHI | PR |
|---|---|---|---|
| **977** | 2022-03-20 | 4.226183 | 81.632738 |
| **978** | 2022-03-21 | 3.431675 | 71.057353 |
| **979** | 2022-03-22 | 3.580492 | 81.631509 |
| **980** | 2022-03-23 | 4.997617 | 71.906149 |
| **981** | 2022-03-24 | 5.125050 | 79.911798 |

In [6]:

```
plt.figure(figsize=(20,10))
plt.scatter(dfc['Date'],dfc['PR'])
plt.ylim(0,100)
plt.show()
```



In [7]:

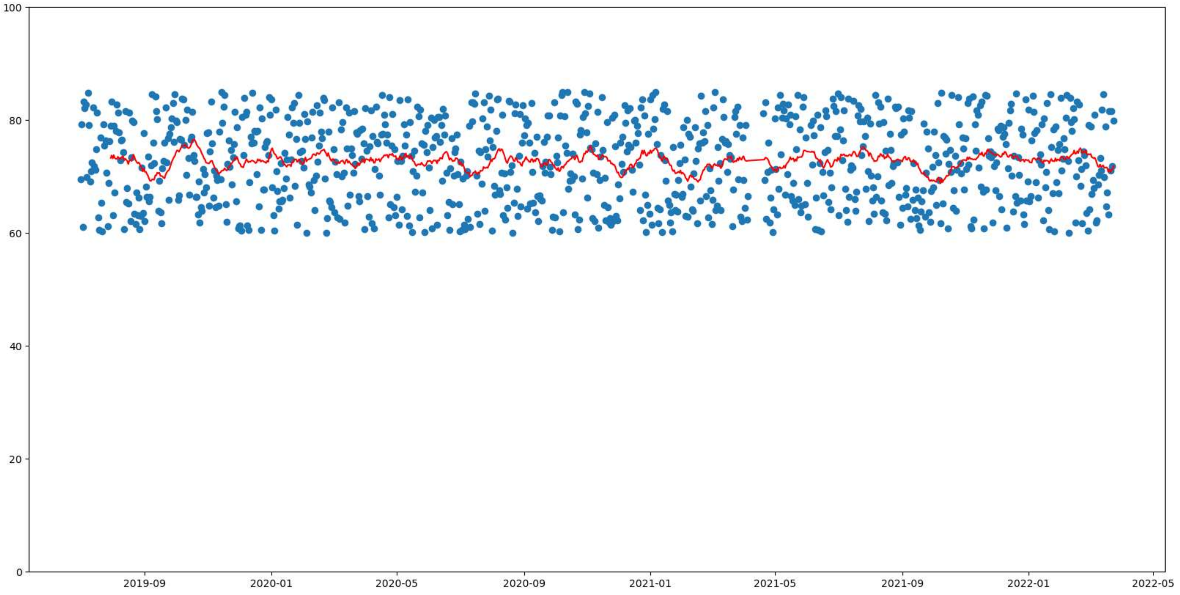
```
dfc['avg'] = dfc['PR'].rolling(30).mean()
dfc.head()
```

Out[7]:

| | Date | GHI | PR | avg |
|---|---|---|---|---|
| **0** | 2019-07-01 | 3.256608 | 69.575676 | NaN |
| **1** | 2019-07-02 | 3.976767 | 79.314411 | NaN |
| **2** | 2019-07-03 | 2.811867 | 61.020006 | NaN |
| **3** | 2019-07-04 | 3.658408 | 83.262576 | NaN |
| **4** | 2019-07-05 | 5.590683 | 82.124440 | NaN |

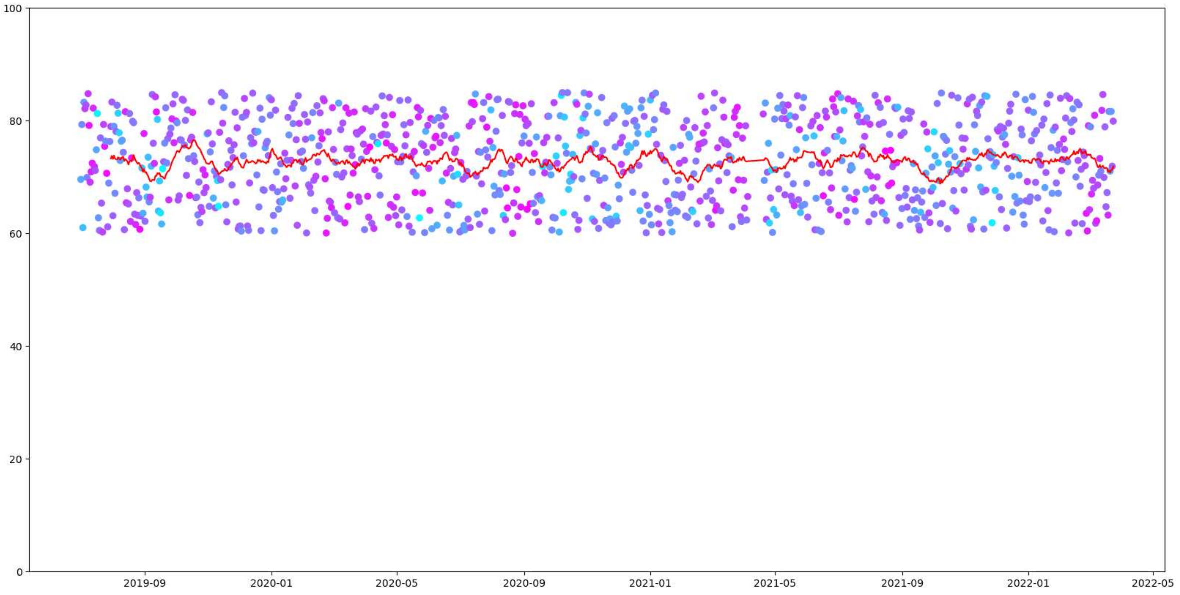
In [8]:

```
plt.figure(figsize=(20,10))
plt.scatter(dfc['Date'],dfc['PR'])
plt.plot(dfc['Date'],dfc['avg'],'r')
plt.ylim(0,100)
plt.show()
```



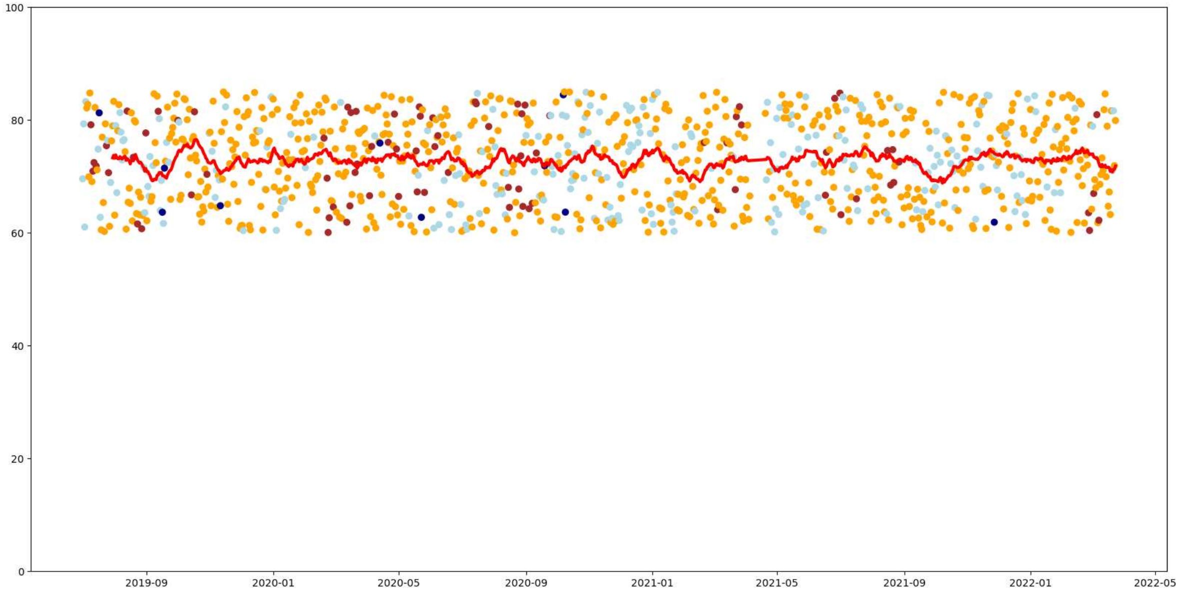
In [9]:

```
plt.figure(figsize=(20,10))
plt.scatter(dfc['Date'],dfc['PR'],c=dfc['GHI'],cmap='cool')
plt.plot(dfc['Date'],dfc['avg'],'r')
plt.ylim(0,100)
plt.show()
```



In [10]:

```
plt.figure(figsize=(20,10))

cmap = (mpl.colors.ListedColormap(['lightblue','orange'])
        .with_extremes(under='navy', over='brown'))
bounds = [2,4,6]
norm = mpl.colors.BoundaryNorm(bounds, cmap.N)

plt.scatter(dfc['Date'],dfc['PR'],c=dfc['GHI'],cmap=cmap,norm=norm)
plt.plot(dfc['Date'],dfc['avg'],'r',linewidth=3)
plt.ylim(0,100)
plt.show()
```



In [11]:

```
datt = dfc['Date']
datt.dtype
```

Out[11]:

```
dtype('<M8[ns]')
```

In [12]:

```
dat = np.datetime_as_string(datt, unit='D')
```

In [13]:

```
dfc.head()
```

Out[13]:

| | Date | GHI | PR | avg |
|---|---|---|---|---|
| **0** | 2019-07-01 | 3.256608 | 69.575676 | NaN |
| **1** | 2019-07-02 | 3.976767 | 79.314411 | NaN |
| **2** | 2019-07-03 | 2.811867 | 61.020006 | NaN |
| **3** | 2019-07-04 | 3.658408 | 83.262576 | NaN |
| **4** | 2019-07-05 | 5.590683 | 82.124440 | NaN |

In [14]:

```
tb = []
for da in dat:
    if(da == '2019-07-01'):
        k = 73.9
        tb.append(k)
    elif(da == '2020-07-01'):
        k = round(k - 0.6 , 1)
        tb.append(k)
    elif(da == '2021-07-01'):
        k = round(k - 0.6 , 1)
        tb.append(k)
    elif(da == '2022-07-01'):
        k = round(k - 0.6 , 1)
        tb.append(k)
    else:
        tb.append(k)
```

In [15]:

```
dfc.tail()
```

Out[15]:

| | Date | GHI | PR | avg |
|---|---|---|---|---|
| **977** | 2022-03-20 | 4.226183 | 81.632738 | 71.181262 |
| **978** | 2022-03-21 | 3.431675 | 71.057353 | 70.789540 |
| **979** | 2022-03-22 | 3.580492 | 81.631509 | 71.231840 |
| **980** | 2022-03-23 | 4.997617 | 71.906149 | 71.249772 |
| **981** | 2022-03-24 | 5.125050 | 79.911798 | 71.855995 |

In [16]:

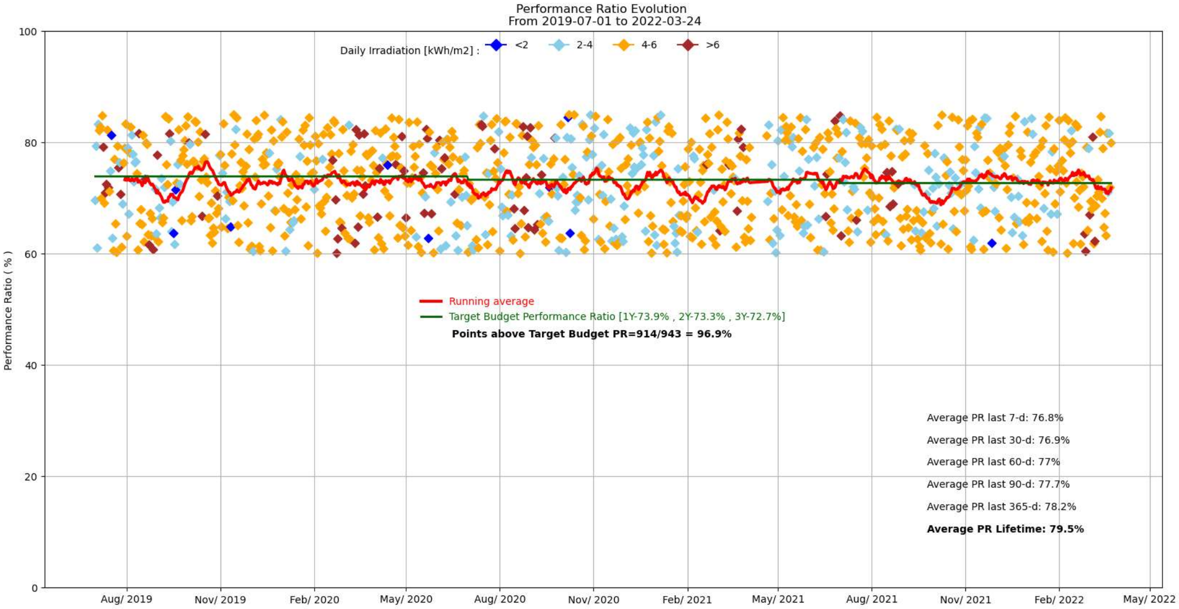
```
ntb = np.array(tb)
```

In [17]:

```
dfc['tbud'] = ntb
```

In [37]:

```
lt.figure(figsize=(20,10))
map = (mpl.colors.ListedColormap(['skyblue','orange']).with_extremes(under='bl
ounds = [2,4,6]

orm = mpl.colors.BoundaryNorm(bounds, cmap.N)
lt.ylabel("Performance Ratio ( % )")
lt.title("Performance Ratio Evolution \n From 2019-07-01 to 2022-03-24")

lt.scatter(dfc['Date'],dfc['PR'],c=dfc['GHI'],cmap=cmap,norm=norm,marker='D')
lt.plot(dfc['Date'],dfc['avg'],'r',linewidth=3,label='Running average')
lt.plot(dfc['Date'],dfc['tbud'],'darkgreen',linewidth=2,label='Target Budget P
plt.plot(dfc['Date'],dfc['avg'],'r',linewidth=0,label='')
lt.ylim(0,100)

egend1=plt.legend(loc='center',frameon=False)
lt.gca().add_artist(legend1)
leg=plt.legend()
or line,text in zip(legend1.get_lines(),legend1.get_texts()):
   text.set_color(line.get_color())

lt.text(dfc['Date'][800],30,"Average PR last 7-d: 76.8%")
lt.text(dfc['Date'][800],26,"Average PR last 30-d: 76.9%")
lt.text(dfc['Date'][800],22,"Average PR last 60-d: 77%")
lt.text(dfc['Date'][800],18,"Average PR last 90-d: 77.7%")
lt.text(dfc['Date'][800],14,"Average PR last 365-d: 78.2%")
lt.text(dfc['Date'][800],10,"Average PR Lifetime: 79.5%",weight='bold')
lt.text(dfc['Date'][350],45,"Points above Target Budget PR=914/943 = 96.9%",we

ate_form = mdates.DateFormatter("%b/ %Y")
lt.gca().xaxis.set_major_formatter(date_form)
lt.gca().xaxis.set_major_locator(mdates.MonthLocator(interval=3))



lt.text(dfc['Date'][240],96,"Daily Irradiation [kWh/m2] :")
egend_elements = [
   Line2D([], [], marker='D', color='blue', label='<2', markerfacecolor='blue'
   Line2D([], [], marker='D', color='skyblue', label='2-4', markerfacecolor='s
   Line2D([], [], marker='D', color='orange', label='4-6', markerfacecolor='or
   Line2D([], [], marker='D', color='brown', label='>6', markerfacecolor='brow

lt.legend(handles=legend_elements, loc='upper center',ncol=len(legend_elements


lt.grid()
lt.show()
```


Performance Ratio Evolution
From 2019-07-01 to 2022-03-24
Daily Irradiation [kWh/m2] :
<2
2-4
4-6
>6
Running average
Target Budget Performance Ratio [1Y-73.9% , 2Y-73.3% , 3Y-72.7%]
Points above Target Budget PR=914/943 = 96.9%
Average PR last 7-d: 76.8%
Average PR last 30-d: 76.9%
Average PR last 60-d: 77%
Average PR last 90-d: 77.7%
Average PR last 365-d: 78.2%
Average PR Lifetime: 79.5%
Performance Ratio ( % )
100
80
60
40
20
0
Aug/ 2019
Nov/ 2019
Feb/ 2020
May/ 2020
Aug/ 2020
Nov/ 2020
Feb/ 2021
May/ 2021
Aug/ 2021
Nov/ 2021
Feb/ 2022
May/ 2022

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: