

UNIVERSITY OF ENGINEERING AND TECHNOLOGY PESHAWAR

Department of Electrical Engineering

Open Ended Lab Report

(OEL)

Course Name	Introduction to AI Programming
Course Code	EE 251
Instructor	Sir Engr. Irshaduullah
Student Name	Mughees ud Din
Registration Number	24JZELE0560

Defining some functions to be used later:

Function 1: Wise_power

```
In [15]: #this function takes a dictionary ,calculate sum of list in values, and return another dictionary having same keys and total
def wise_power(dictionary)->dict:
    power={i:sum(v) for i,v in dictionary.items()}
    return power
```

Function 2: w_days_consumption

```
In [16]: #calculates the sum of consumptions of given days
def w_days_consumption(*days)-> dict:
    return sum([p for d,p in day_wise_power.items() if d in days])
```

Main Program

Loading data

```
In [17]: import csv
month_dict=dict()
day_dict=dict()
hour_dict=dict()
with open(r"Building_2.csv",) as file:
    reader=csv.reader(file)
    for i,r in enumerate(reader):
        if i==0:
            continue
        #month as key and list of power as value
        month=int(float(r[0]))
        power=float(r[7])
        if month not in month_dict:
            month_dict[month]=[]
        month_dict[month].append(power)
        #day as key and list of power as value
        day=int(float(r[2]))
        if day not in day_dict:
            day_dict[day]=[]
        day_dict[day].append(power)
        #hour as key and list of power as value
        hour=int(float(r[1]))
        if hour not in hour_dict:
            hour_dict[hour]=[]
        hour_dict[hour].append(power)
```

Calculating annual consumption

```
In [18]: #annual consumption
total_annual=0
for p in month_dict.values():
    total_annual+=sum(p)
print("Total power consumption of whole year is:",total_annual)
```

Total power consumption of whole year is: 9353.601287571018

Calculating monthly consumption

```
In [19]: # month wise total power consumption
print("Month wise power consumption is : \n")
for m,p in sorted(wise_power(month_dict).items()):
    print(f"Month number {m}: \n Total Power: {p}")
```

Month wise power consumption is :

```
Month number 1:  
    Total Power: 871.7700347239177  
Month number 2:  
    Total Power: 650.1312497212728  
Month number 3:  
    Total Power: 622.765465544637  
Month number 4:  
    Total Power: 507.91130417073566  
Month number 5:  
    Total Power: 602.9855291641236  
Month number 6:  
    Total Power: 982.2664885082245  
Month number 7:  
    Total Power: 1088.248341602707  
Month number 8:  
    Total Power: 980.6947769755682  
Month number 9:  
    Total Power: 728.6168402811686  
Month number 10:  
    Total Power: 662.0097499999999  
Month number 11:  
    Total Power: 569.2347680562337  
Month number 12:  
    Total Power: 1086.9667388224284
```

Calculating daily consumption

```
In [20]: #day wise total power consumption  
print("Day wise power consumption is :\n")  
day_wise_power=wise_power(day_dict)  
for d,p in sorted(day_wise_power.items()):  
    print(f"Day number {d}:\n Total Power: {p}")
```

Day wise power consumption is :

```
Day number 1:  
    Total Power: 1373.1661628652096  
Day number 2:  
    Total Power: 1220.6169038224539  
Day number 3:  
    Total Power: 1260.9006389344534  
Day number 4:  
    Total Power: 1272.1225613459746  
Day number 5:  
    Total Power: 1294.534868103536  
Day number 6:  
    Total Power: 1409.5760747568766  
Day number 7:  
    Total Power: 1522.684077742513
```

Calculating hourly consumption

```
In [21]: #hour wise total consumption  
print("Hour wise power consumption is :\n")  
for m,p in sorted(wise_power(hour_dict).items()):  
    print(f"Hour number {m}:\n Total Power: {p}")
```

Hour wise power consumption is :

```

Hour number 1:
Total Power: 340.32271686503094
Hour number 2:
Total Power: 269.74679075419107
Hour number 3:
Total Power: 232.40176408894857
Hour number 4:
Total Power: 226.79412936607997
Hour number 5:
Total Power: 214.99102515540125
Hour number 6:
Total Power: 264.15851259918213
Hour number 7:
Total Power: 311.67808675664264
Hour number 8:
Total Power: 283.19631004943847
Hour number 9:
Total Power: 276.3138416824341
Hour number 10:
Total Power: 300.6390583333333
Hour number 11:
Total Power: 317.38728056386316
Hour number 12:
Total Power: 345.12650417989096
Hour number 13:
Total Power: 370.4389694613139
Hour number 14:
Total Power: 407.4598972780863
Hour number 15:
Total Power: 448.5564555628458
Hour number 16:
Total Power: 504.1072847147624
Hour number 17:
Total Power: 551.3527653884887
Hour number 18:
Total Power: 516.6789893252214
Hour number 19:
Total Power: 527.9074394690196
Hour number 20:
Total Power: 541.3078669184367
Hour number 21:
Total Power: 578.9661306858063
Hour number 22:
Total Power: 549.6638585316977
Hour number 23:
Total Power: 524.7263863087971
Hour number 24:
Total Power: 449.6792235321045

```

Calculate the consumption of four seasons.

- o Winter: Dec–Feb
 - o Spring: Mar–May
 - o Summer: Jun–Aug
 - o Autumn: Sep–Nov
-

Winter consumption

```
In [22]: winter=[]
total_in_winter=0
for m,p in month_dict.items():
    if m in [12,1,2]:
        winter.append(p)
for i in winter:
    total_in_winter+=sum(i)
print("Winter season total consumption is :",total_in_winter)
```

Winter season total consumption is : 2608.868023267619

Spring consumption

```
In [23]: total_in_spring=sum([sum(p) for m,p in month_dict.items() if m in [3,4,5]])
print("Spring season total consumption is :",total_in_spring)
```

Spring season total consumption is : 1733.6622988794961

Summer consumption

```
In [24]: total_in_summer=sum([sum(p) for m,p in month_dict.items() if m in [6,7,8]])
print("Summer season total consumption is :",total_in_summer)
```

Summer season total consumption is : 3051.2096070864995

Autumn consumption

```
In [25]: total_in_autumn=sum([sum(p) for m,p in month_dict.items() if m in [9,10,11]])
print("Autumn season total consumption is :",total_in_autumn)
```

Autumn season total consumption is : 1959.8613583374022

Calculate daily consumption over the weekdays for the whole year. Compare weekend vs weekday consumption.

Working days consumption

```
In [26]: working_days_consumption=w_days_consumption(1,2,3,4,5)
print("working days consumption is : ",working_days_consumption)
```

working days consumption is : 6421.341135071627

Weekend days consumption

```
In [27]: weekend_days_consumption=w_days_consumption(6,7)
print("weekend days consumption is : ",weekend_days_consumption)
```

weekend days consumption is : 2932.2601524993897

Working days consumption VS Weekend days consumption

```
In [28]: #comparision: weekend consumption Vs working days consumption
print("Average day consumption comparision:")
if weekend_days_consumption/2 > working_days_consumption/5:
    print("weekend days average is more than working days average")
else:
    print("working days average is more than weekend days average")

print("\n\nTotal consumption comparision:")
if weekend_days_consumption > working_days_consumption:
    print("weekend (2)days total is more than working (5)days total")
else:
    print("working (5)days total is more than weekend (2)days total")
```

Average day consumption comparision:
weekend days average is more than working days average

Total consumption comparision:
working (5)days total is more than weekend (2)days total

Conclusion

The analysis of Building 2's electricity consumption from the CityLearn dataset reveals a total annual consumption of approximately 9354 kWh. Monthly consumption peaked in July (1088 kWh) and December (1087 kWh), while the lowest was in April (508 kWh). Daily patterns showed higher average consumption on weekends (1466 kWh/day) compared to weekdays (1284 kWh/day), though total weekday consumption was greater due to more days. Seasonal breakdown indicates highest usage in summer (June–August: ~3051 kWh), followed by winter. Hourly data highlights peak demand in the late afternoon and evening (hours 17–21). These insights into temporal energy usage patterns can inform demand-side management strategies and potential integration of renewable energy or storage systems to reduce peak loads and improve efficiency.