In [33]:

```
import pandas as pd
df=pd.read_csv('D:/aiml/SeoulBikeData (1).csv',encoding='ISO-8859-1')
```

In [34]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [35]:

```
df.head()
```

Out[35]:

| | Date | Rented Bike Count | Hour | Temperature(°C) | Humidity(%) | Wind speed (m/s) | Visibility (10m) | Dew point temperature(°C) | R |
|---|---|---|---|---|---|---|---|---|---|
| **0** | 01/12/2017 | 254 | 0 | -5.2 | 37 | 2.2 | 2000 | -17.6 | |
| **1** | 01/12/2017 | 204 | 1 | -5.5 | 38 | 0.8 | 2000 | -17.6 | |
| **2** | 01/12/2017 | 173 | 2 | -6.0 | 39 | 1.0 | 2000 | -17.7 | |
| **3** | 01/12/2017 | 107 | 3 | -6.2 | 40 | 0.9 | 2000 | -17.6 | |
| **4** | 01/12/2017 | 78 | 4 | -6.0 | 36 | 2.3 | 2000 | -18.6 | |

In [36]:

```
df.tail()
```

Out[36]:

| | Date | Rented Bike Count | Hour | Temperature(°C) | Humidity(%) | Wind speed (m/s) | Visibility (10m) | Dew point temperature(°C) |
|---|---|---|---|---|---|---|---|---|
| **8755** | 30/11/2018 | 1003 | 19 | 4.2 | 34 | 2.6 | 1894 | -10.3 |
| **8756** | 30/11/2018 | 764 | 20 | 3.4 | 37 | 2.3 | 2000 | -9.9 |
| **8757** | 30/11/2018 | 694 | 21 | 2.6 | 39 | 0.3 | 1968 | -9.9 |
| **8758** | 30/11/2018 | 712 | 22 | 2.1 | 41 | 1.0 | 1859 | -9.8 |
| **8759** | 30/11/2018 | 584 | 23 | 1.9 | 43 | 1.3 | 1909 | -9.3 |

In [37]:

```
df=df.fillna(0)
```

In [38]:

```
df.describe
```

Out[38]:

```
<bound method NDFrame.describe of             Date  Rented Bike Count  Hour
Temperature(°C)  Humidity(%)  \
0     01/12/2017                254     0             -5.2           37
1     01/12/2017                204     1             -5.5           38
2     01/12/2017                173     2             -6.0           39
3     01/12/2017                107     3             -6.2           40
4     01/12/2017                 78     4             -6.0           36
...          ...                ...   ...              ...          ...
8755  30/11/2018               1003    19              4.2           34
8756  30/11/2018                764    20              3.4           37
8757  30/11/2018                694    21              2.6           39
8758  30/11/2018                712    22              2.1           41
8759  30/11/2018                584    23              1.9           43

      Wind speed (m/s)  Visibility (10m)  Dew point temperature(°C)  \
0                  2.2              2000                      -17.6
1                  0.8              2000                      -17.6
2                  1.0              2000                      -17.7
3                  0.9              2000                      -17.6
4                  2.3              2000                      -18.6
...                ...               ...                        ...
8755               2.6              1894                      -10.3
8756               2.3              2000                       -9.9
8757               0.3              1968                       -9.9
8758               1.0              1859                       -9.8
8759               1.3              1909                       -9.3

      Solar Radiation (MJ/m2)  Rainfall(mm)  Snowfall (cm) Seasons  \
0                         0.0           0.0            0.0  Winter
1                         0.0           0.0            0.0  Winter
2                         0.0           0.0            0.0  Winter
3                         0.0           0.0            0.0  Winter
4                         0.0           0.0            0.0  Winter
...                       ...           ...            ...     ...
8755                      0.0           0.0            0.0  Autumn
8756                      0.0           0.0            0.0  Autumn
8757                      0.0           0.0            0.0  Autumn
8758                      0.0           0.0            0.0  Autumn
8759                      0.0           0.0            0.0  Autumn

         Holiday Functioning Day
0     No Holiday             Yes
1     No Holiday             Yes
2     No Holiday             Yes
3     No Holiday             Yes
4     No Holiday             Yes
...          ...             ...
8755  No Holiday             Yes
8756  No Holiday             Yes
8757  No Holiday             Yes
8758  No Holiday             Yes
8759  No Holiday             Yes

[8760 rows x 14 columns]>
```

In [39]:

```
x1=df['Hour']
x2=df['Humidity(%)']
x3=df['Rainfall(mm)']
x4=df['Dew point temperature(°C)']
x5=df['Snowfall (cm)']
x6=df['Solar Radiation (MJ/m2)']
x7=df['Temperature(°C)']
x8=df['Wind speed (m/s)']
x9=df['Visibility (10m)']
y=df['Rented Bike Count']
size=x1.size
print(x1,x2,x3,x4,x5,x6,x7,x8,x9,y)
```

```
4       2000
        ...
8755    1894
8756    2000
8757    1968
8758    1859
8759    1909
Name: Visibility (10m), Length: 8760, dtype: int64 0        254
1        204
2        173
3        107
4         78
        ...
8755    1003
8756     764
8757     694
8758     712
8759     584
Name: Rented Bike Count, Length: 8760, dtype: int64
```

In [40]:

```
import random
x1_train=[]
x2_train=[]
x3_train=[]
x4_train=[]
x5_train=[]
x6_train=[]
x7_train=[]
x8_train=[]
x9_train=[]
y_train=[]

for j in range(0,1):
  for i in range(0,8760,2):
    x1_train.append(x1[i])
    x2_train.append(x2[i])
    x3_train.append(x3[i])
    x4_train.append(x4[i])
    x5_train.append(x5[i])
    x6_train.append(x6[i])
    x7_train.append(x7[i])
    x8_train.append(x8[i])
    x9_train.append(x8[i])
    y_train.append(y[i])
```

In [41]:

```
import random
x1_train=[]
x2_train=[]
x3_train=[]
x4_train=[]
x5_train=[]
x6_train=[]
x7_train=[]
x8_train=[]
x9_train=[]
y_train=[]

for j in range(1,300):
  for i in range(0,8760,2):
    x1_train.append(x1[i])
    x2_train.append(x2[i])
    x3_train.append(x3[i])
    x4_train.append(x4[i])
    x5_train.append(x5[i])
    x6_train.append(x6[i])
    x7_train.append(x7[i])
    x8_train.append(x8[i])
    x9_train.append(x8[i])
    y_train.append(y[i])
```

In [42]:

```
import random
x1_train=[]
x2_train=[]
x3_train=[]
x4_train=[]
x5_train=[]
x6_train=[]
x7_train=[]
x8_train=[]
x9_train=[]
y_train=[]

for j in range(300,600):
  for i in range(0,8760,2):
    x1_train.append(x1[i])
    x2_train.append(x2[i])
    x3_train.append(x3[i])
    x4_train.append(x4[i])
    x5_train.append(x5[i])
    x6_train.append(x6[i])
    x7_train.append(x7[i])
    x8_train.append(x8[i])
    x9_train.append(x8[i])
    y_train.append(y[i])
```

In [43]:

```
import random
x1_train=[]
x2_train=[]
x3_train=[]
x4_train=[]
x5_train=[]
x6_train=[]
x7_train=[]
x8_train=[]
x9_train=[]
y_train=[]

for j in range(600,1000):
  for i in range(0,8760,2):
    x1_train.append(x1[i])
    x2_train.append(x2[i])
    x3_train.append(x3[i])
    x4_train.append(x4[i])
    x5_train.append(x5[i])
    x6_train.append(x6[i])
    x7_train.append(x7[i])
    x8_train.append(x8[i])
    x9_train.append(x8[i])
    y_train.append(y[i])
```

Mutiple Linear Regression

In [44]:

```
e=[]
ee=[]
def linear(m1,m2,m3,m4,m5,m6,m7,m8,m9,c):
  sum=0
  yp=[]
  for i in range(0,len(x8_train)):
    yp.append(m1*x1_train[i]+m2*x2_train[i]+m3*x3_train[i]+m4*x4_train[i]+m5*x5_train[i]+m6
    g=(y_train[i]-yp[i])**2
    sum=sum+g
  e.append(sum)
  ee.append(np.mod(sum,len(x1_train)))
  print(m1,m2,m3,m4,m5,m6,m7,m8,m9,c,np.mod(sum,len(x9_train)))
  sum=0

for i in range(1,100):
  linear(1+i,2+i,3+i,4+i,5+i,6+i,7+i,8+i,9+i,10+i,)
```

```
2 3 4 5 6 7 8 9 10 11 57676.319091796875
3 4 5 6 7 8 9 10 11 12 1279739.3533935547
4 5 6 7 8 9 10 11 12 13 1642793.9072265625
5 6 7 8 9 10 11 12 13 14 1146839.8051757812
6 7 8 9 10 11 12 13 14 15 1543878.3762207031
7 8 9 10 11 12 13 14 15 16 1081908.9421386719
8 9 10 11 12 13 14 15 16 17 1512931.035522461
9 10 11 12 13 14 15 16 17 18 1084945.5163574219
10 11 12 13 14 15 16 17 18 19 1549951.6552734375
11 12 13 14 15 16 17 18 19 20 1155948.6804199219
12 13 14 15 16 17 18 19 20 21 1654939.0100097656
13 14 15 16 17 18 19 20 21 22 1294920.2055664062
14 15 16 17 18 19 20 21 22 23 75893.77734375
15 16 17 18 19 20 21 22 23 24 1501860.431640625
16 17 18 19 20 21 22 23 24 25 316817.7470703125
17 18 19 20 21 22 23 24 25 26 24767.888671875
18 19 20 21 22 23 24 25 26 27 625708.3413085938
19 20 21 22 23 24 25 26 27 28 367642.5166015625
20 21 22 23 24 25 26 27 28 29 1002564.666015625
21 22 23 24 25 26 27 28 29 30 778485.0625
22 23 24 25 26 27 28 29 30 31 1447388.1748046875
23 24 25 26 27 28 29 30 31 32 1257295.4853515625
24 25 26 27 28 29 30 31 32 33 208183.7060546875
25 26 27 28 29 30 31 32 33 34 52066.3232421875
26 27 28 29 30 31 32 33 34 35 788946.7294921875
27 28 29 30 31 32 33 34 35 36 666821.552734375
28 29 30 31 32 33 34 35 36 37 1437679.521484375
29 30 31 32 33 34 35 36 37 38 1349528.16796875
30 31 32 33 34 35 36 37 38 39 402377.94140625
31 32 33 34 35 36 37 38 39 40 348207.390625
32 33 34 35 36 37 38 39 40 41 1187041.4375
33 34 35 36 37 38 39 40 41 42 1166858.66015625
34 35 36 37 38 39 40 41 42 43 287680.419921875
35 36 37 38 39 40 41 42 43 44 301467.8984375
36 37 38 39 40 41 42 43 44 45 1208264.666015625
37 38 39 40 41 42 43 44 45 46 1256064.8359375
38 39 40 41 42 43 44 45 46 47 444842.7109375
39 40 41 42 43 44 45 46 47 48 526611.140625
40 41 42 43 44 45 46 47 48 49 1501376.06640625
41 42 43 44 45 46 47 48 49 50 1617124.5859375
```

```
42 43 44 45 46 47 48 49 50 51 873885.31640625
43 44 45 46 47 48 49 50 51 52 1023616.484375
44 45 46 47 48 49 50 51 52 53 314344.03125
45 46 47 48 49 50 51 52 53 54 498081.953125
46 47 48 49 50 51 52 53 54 55 1574771.88671875
47 48 49 50 51 52 53 54 55 56 40509.91015625
48 49 50 51 52 53 54 55 56 57 1151183.30078125
49 50 51 52 53 54 55 56 57 58 1402893.9375
50 51 52 53 54 55 56 57 58 59 795588.40625
51 52 53 54 55 56 57 58 59 60 1081206.1328125
52 53 54 55 56 57 58 59 60 61 507887.515625
53 54 55 56 57 58 59 60 61 62 827557.96875
54 55 56 57 58 59 60 61 62 63 288220.6875
55 56 57 58 59 60 61 62 63 64 641868.8984375
56 57 58 59 60 61 62 63 64 65 136476.4375
57 58 59 60 61 62 63 64 65 66 524141.9921875
58 59 60 61 62 63 64 65 66 67 52747.09375
59 60 61 62 63 64 65 66 67 68 474348.9765625
60 61 62 63 64 65 66 67 68 69 36956.9921875
61 62 63 64 65 66 67 68 69 70 492562.015625
62 63 64 65 66 67 68 69 70 71 89136.71875
63 64 65 66 67 68 69 70 71 72 578700.1875
64 65 66 67 68 69 70 71 72 73 209253.7890625
65 66 67 68 69 70 71 72 73 74 732818.8203125
66 67 68 69 70 71 72 73 74 75 397377.5390625
67 68 69 70 71 72 73 74 75 76 954942.6484375
68 69 70 71 72 73 74 75 76 77 653501.21875
69 70 71 72 73 74 75 76 77 78 1245020.6015625
70 71 72 73 74 75 76 77 78 79 977572.515625
71 72 73 74 75 76 77 78 79 80 1603097.921875
72 73 74 75 76 77 78 79 80 81 1369586.390625
73 74 75 76 77 78 79 80 81 82 277025.640625
74 75 76 77 78 79 80 81 82 83 77520.640625
75 76 77 78 79 80 81 82 83 84 771001.53125
76 77 78 79 80 81 82 83 84 85 605452.328125
77 78 79 80 81 82 83 84 85 86 1332924.421875
78 79 80 81 82 83 84 85 86 87 1201406.84375
79 80 81 82 83 84 85 86 87 88 210852.390625
80 81 82 83 84 85 86 87 88 89 113314.546875
81 82 83 84 85 86 87 88 89 90 908738.515625
82 83 84 85 86 87 88 89 90 91 845152.34375
83 84 85 86 87 88 89 90 91 92 1674500.78125
84 85 86 87 88 89 90 91 92 93 1644950.296875
85 86 87 88 89 90 91 92 93 94 756411.28125
86 87 88 89 90 91 92 93 94 95 760830.609375
87 88 89 90 91 92 93 94 95 96 1658093.9375
88 89 90 91 92 93 94 95 96 97 1696529.3125
89 90 91 92 93 94 95 96 97 98 875883.734375
90 91 92 93 94 95 96 97 98 99 948218.9375
91 92 93 94 95 96 97 98 99 100 161677.09375
92 93 94 95 96 97 98 99 100 101 267935.09375
93 94 95 96 97 98 99 100 101 102 1267267.5
94 95 96 97 98 99 100 101 102 103 1407638.828125
95 96 97 98 99 100 101 102 103 104 689030.90625
96 97 98 99 100 101 102 103 104 105 863245.890625
97 98 99 100 101 102 103 104 105 106 178538.59375
98 99 100 101 102 103 104 105 106 107 386784.53125
99 100 101 102 103 104 105 106 107 108 1488104.8125
100 101 102 103 104 105 106 107 108 109 1730388.21875
```

In [45]:

```
print(ee)
```

```
[57676.319091796875, 1279739.3533935547, 1642793.9072265625, 1146839.8051757
812, 1543878.3762207031, 1081908.9421386719, 1512931.035522461, 1084945.5163
574219, 1549951.6552734375, 1155948.6804199219, 1654939.0100097656, 1294920.
2055664062, 75893.77734375, 1501860.431640625, 316817.7470703125, 24767.8886
71875, 625708.3413085938, 367642.5166015625, 1002564.666015625, 778485.0625,
1447388.1748046875, 1257295.4853515625, 208183.7060546875, 52066.3232421875,
788946.7294921875, 666821.552734375, 1437679.521484375, 1349528.16796875, 40
2377.94140625, 348207.390625, 1187041.4375, 1166858.66015625, 287680.4199218
75, 301467.8984375, 1208264.666015625, 1256064.8359375, 444842.7109375, 5266
11.140625, 1501376.06640625, 1617124.5859375, 873885.31640625, 1023616.48437
5, 314344.03125, 498081.953125, 1574771.88671875, 40509.91015625, 1151183.30
078125, 1402893.9375, 795588.40625, 1081206.1328125, 507887.515625, 827557.9
6875, 288220.6875, 641868.8984375, 136476.4375, 524141.9921875, 52747.09375,
474348.9765625, 36956.9921875, 492562.015625, 89136.71875, 578700.1875, 2092
53.7890625, 732818.8203125, 397377.5390625, 954942.6484375, 653501.21875, 12
45020.6015625, 977572.515625, 1603097.921875, 1369586.390625, 277025.640625,
77520.640625, 771001.53125, 605452.328125, 1332924.421875, 1201406.84375, 21
0852.390625, 113314.546875, 908738.515625, 845152.34375, 1674500.78125, 1644
950.296875, 756411.28125, 760830.609375, 1658093.9375, 1696529.3125, 875883.
734375, 948218.9375, 161677.09375, 267935.09375, 1267267.5, 1407638.828125,
689030.90625, 863245.890625, 178538.59375, 386784.53125, 1488104.8125, 17303
88.21875]
```

In [46]:

```
a=np.array(ee).min()
print(a)
c=np.where(ee==a)
```

```
24767.888671875
```

In [47]:

```
print(c)
```

```
(array([15], dtype=int64),)
```

In [48]:

```
ee1=np.arange(0,99)
print(ee1)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47
 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71
 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
 96 97 98]
```

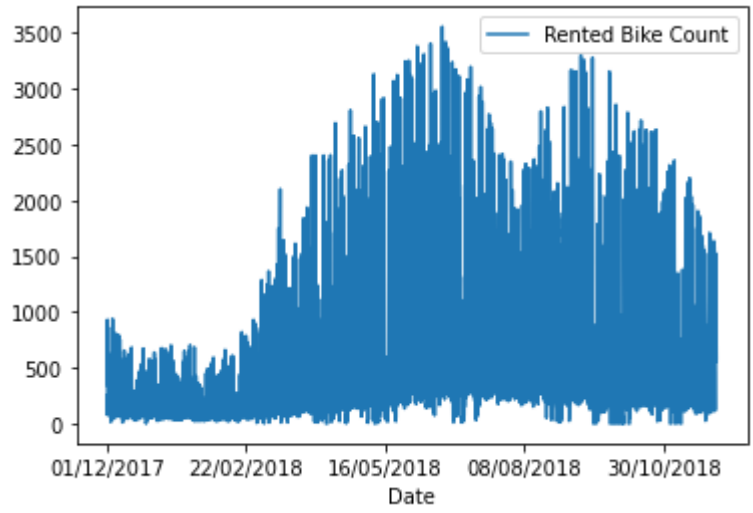
Data Preparation and data Visualisation using Matplotlib

In [49]:

```
df.plot(x='Date',y='Rented Bike Count',kind='line')
```

Out[49]:

```
<AxesSubplot:xlabel='Date'>
```

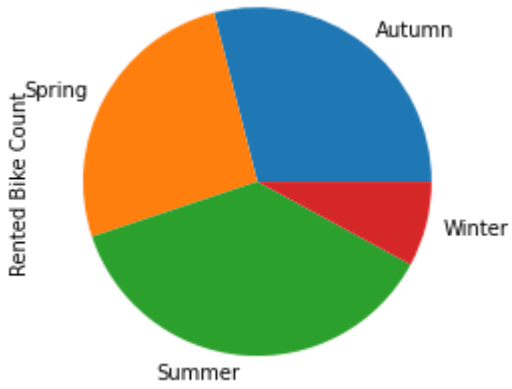



In [50]:

```
df.groupby('Seasons').sum()['Rented Bike Count'].plot.pie()
```

Out[50]:

```
<AxesSubplot:ylabel='Rented Bike Count'>
```

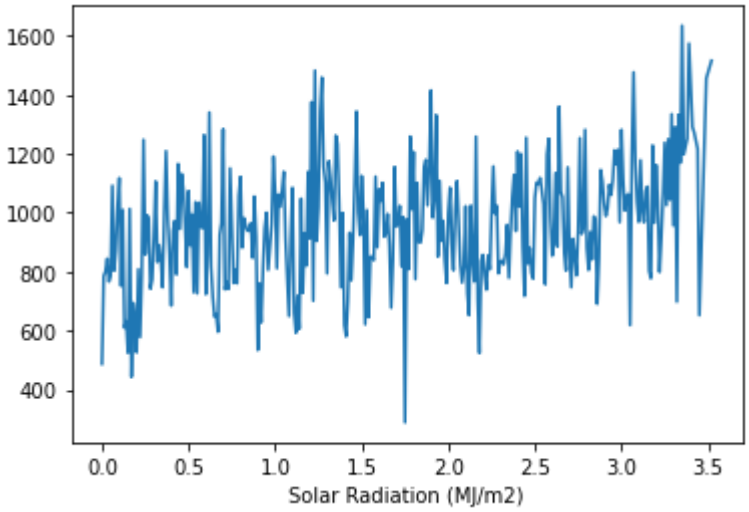


In [51]:

```
df.groupby('Solar Radiation (MJ/m2)').mean()['Rented Bike Count'].plot()
```

Out[51]:

```
<AxesSubplot:xlabel='Solar Radiation (MJ/m2)'>
```

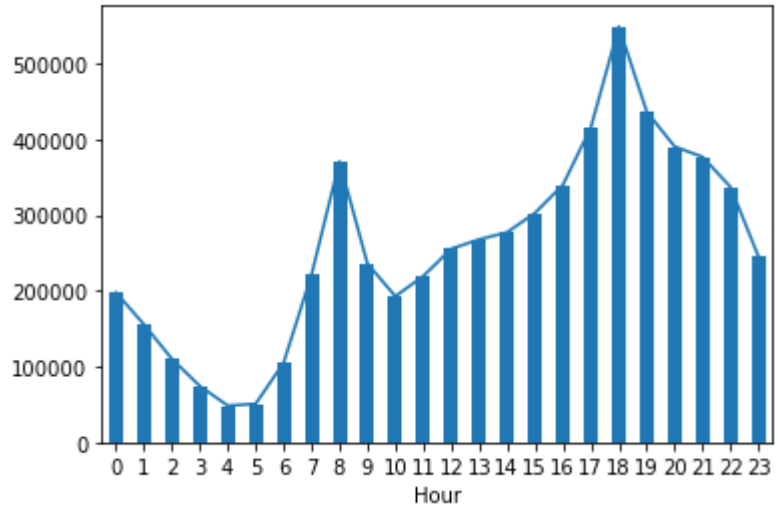


In [52]:

```
df.groupby('Hour').sum()['Rented Bike Count'].plot.bar()
df.groupby('Hour').sum()['Rented Bike Count'].plot()
```

Out[52]:

```
<AxesSubplot:xlabel='Hour'>
```

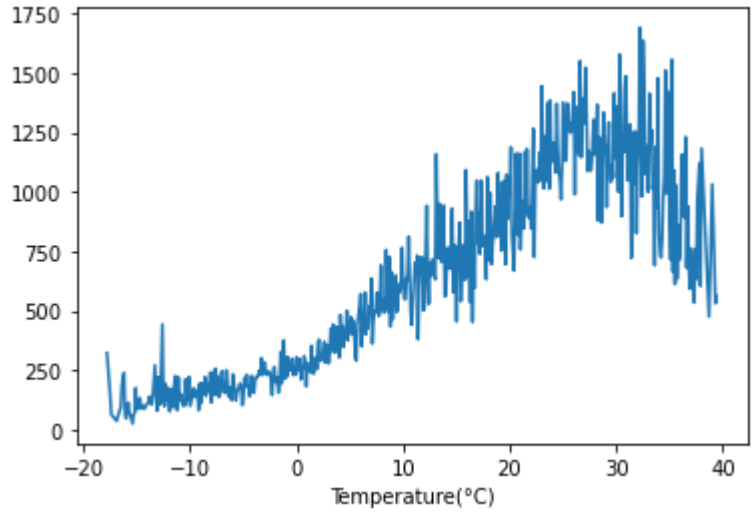



In [53]:

```
df.groupby('Temperature(°C)').mean()['Rented Bike Count'].plot()
```

Out[53]:

```
<AxesSubplot:xlabel='Temperature(°C)'>
```

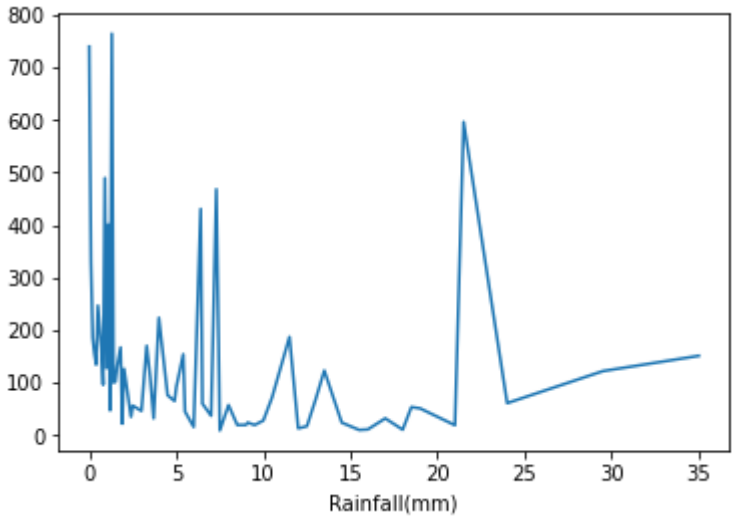


In [54]:

```
df.groupby('Rainfall(mm)').mean()['Rented Bike Count'].plot()
```

Out[54]:

```
<AxesSubplot:xlabel='Rainfall(mm)'>
```




In [55]:

```
df.groupby('Snowfall (cm)').mean()['Rented Bike Count'].plot()
```

Out[55]:

```
<AxesSubplot:xlabel='Snowfall (cm)'>
```