In [46]:

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

In [123]:

```
import plotly.express as px
#from plotly.figure_fctory import create_table
import plotly.offline as py
from plotly.offline import iplot
from plotly.graph_objs import Scatter
py.init_notebook_mode(connected=True)
```

In [39]:

```
df = pd.read_csv('data.csv')
```

In [40]:

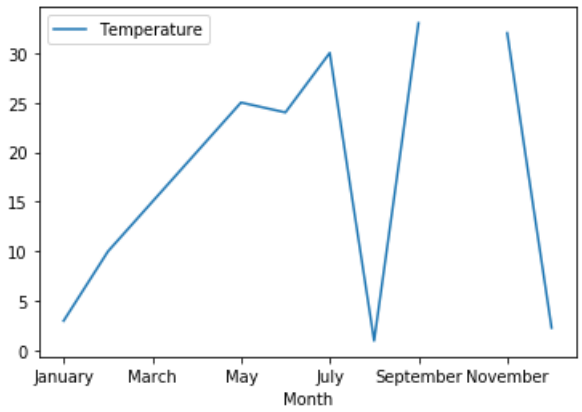
```
print(df)
```

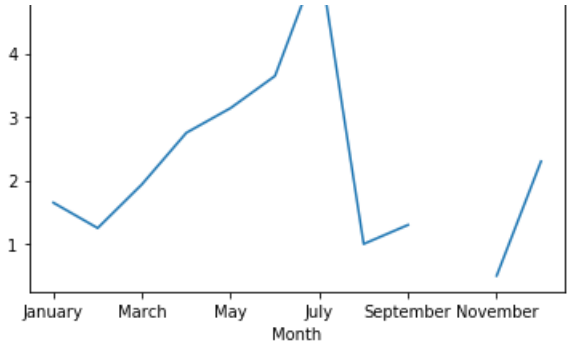
```
        Month  Rainfall  Temperature
0     January     1.650          3.0
1    February     1.250         10.0
2       March     1.940         15.0
3       April     2.750         20.0
4         May     3.140         25.0
5        June     3.645         24.0
6        July     5.500         30.0
7      August     1.000          1.0
8   September     1.300         33.0
9     October       NaN          NaN
10   November     0.500         32.0
11   December     2.300          2.3
```

In [30]:

```
plt.figure(figsize=(50,70))
df.plot(x="Month" , y="Temperature")
df.plot(x="Month", y="Rainfall")
plt.show()
```

```
<Figure size 3600x5040 with 0 Axes>
```

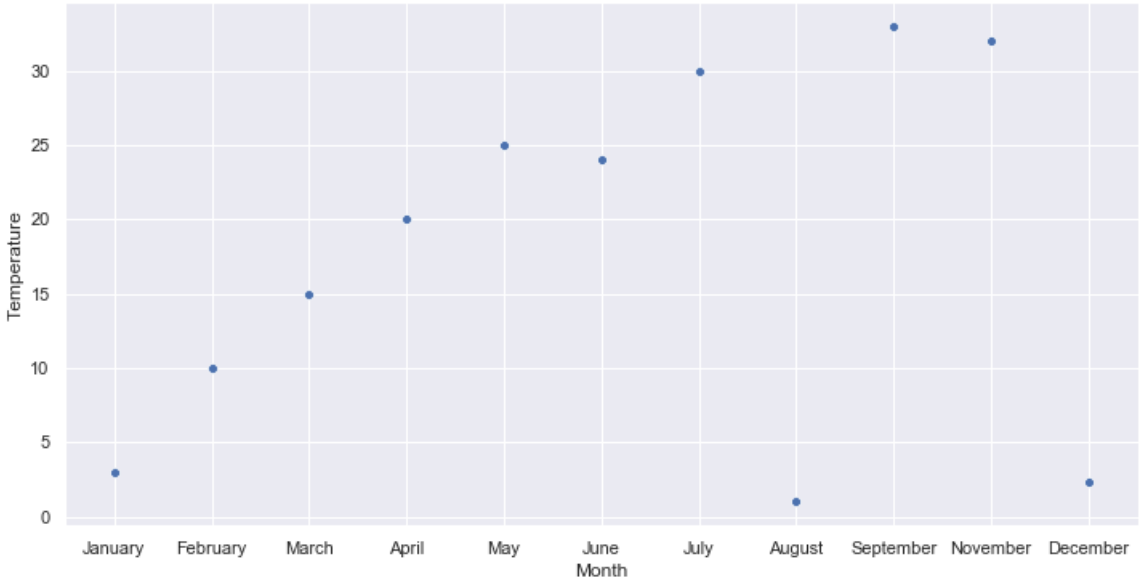




Here scatter plot between month and temp

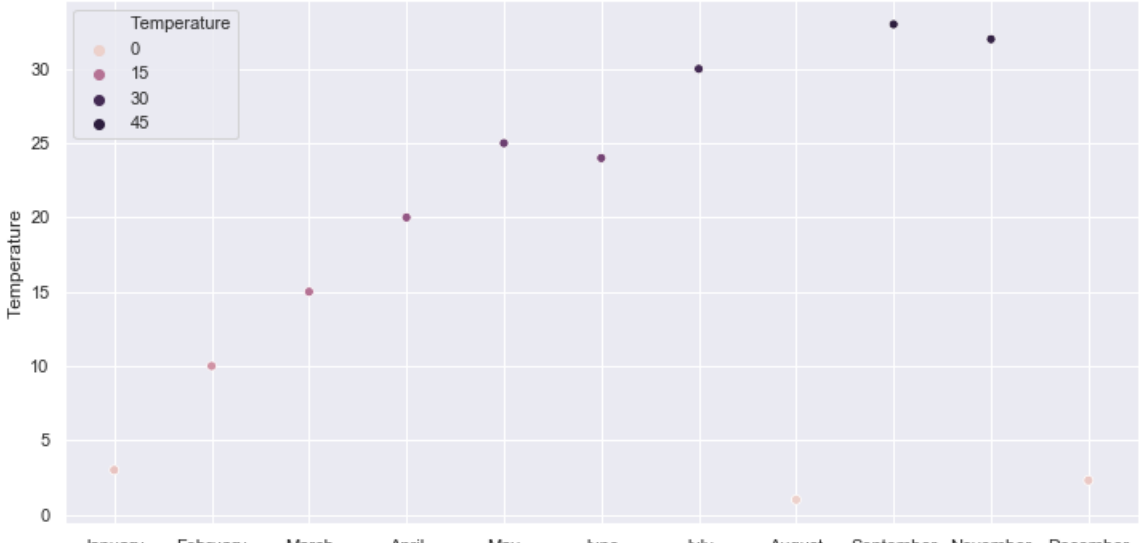
In [7]:

```
sns.set(rc={'figure.figsize':(12,6)})
sns.scatterplot(x='Month' , y='Temperature' ,data= df)
plt.show()
```



In [51]:

```
##HUE:produce points with different colors. Can be either categorical or numeric
sns.set(rc={'figure.figsize':(12,6)})
sns.scatterplot(x='Month' , y='Temperature' ,hue='Temperature',data= df)
plt.show()
```



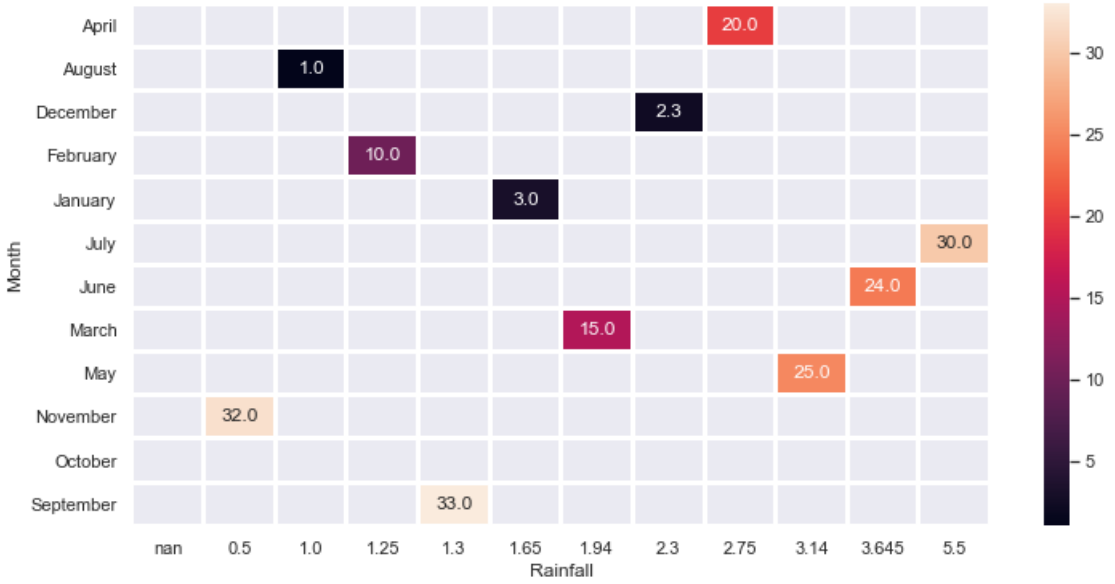
In [60]:

```
dft = df.pivot("Month", "Rainfall", "Temperature")
sns.heatmap(dft, linewidths=2,annot=True,fmt='.1f')

#sns.heatmap(dft)
```

Out[60]:

```
<matplotlib.axes._subplots.AxesSubplot at 0xce968c8>
```



In [99]:

```
df1 = df.query('Rainfall > 5 and Temperature >20')
#df1.head()
fig=px.scatter(df1,x='Rainfall',y='Temperature', color='Month')
fig.show()
```

In [111]:

```
df1=df.pivot("Month","Rainfall","Temperature")
#print(df1)
sns.heatmap(df1,annot=True)
plt.show()
```

In [ ]:

```

```

In [ ]:

```

```