Entrée [24]:

```
import pandas as pd
import numpy as np
#Reunion Island (1st location)
df=pd.read_csv('temperature_dataset.csv')
# France metropolitan (2nd location)
df2=pd.read_csv('temperature_time_serie.csv')
```

Entrée [25]:

```
print(df.head())
```

```
        LAT       LON  YEAR  MO  DY    T2M
0 -21.34449  55.47301  2000   1   1  25.00
1 -21.34449  55.47301  2000   1   2  25.00
2 -21.34449  55.47301  2000   1   3  25.19
3 -21.34449  55.47301  2000   1   4  25.25
4 -21.34449  55.47301  2000   1   5  25.02
```

Entrée [26]:

```
print(df2.head())
```

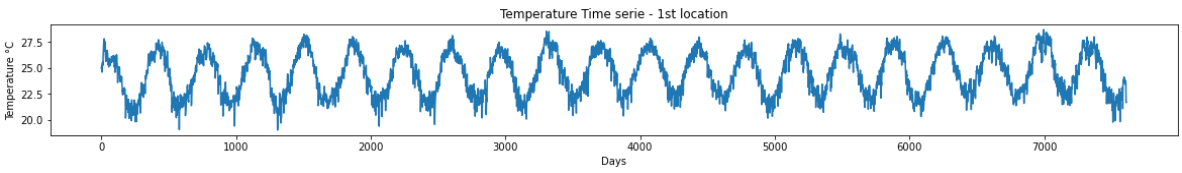
```
        LAT      LON  YEAR  MO  DY     T
0  48.59221  7.49331  2000   1   1  0.23
1  48.59221  7.49331  2000   1   2  0.61
2  48.59221  7.49331  2000   1   3 -1.96
3  48.59221  7.49331  2000   1   4  1.88
4  48.59221  7.49331  2000   1   5  3.36
```

Entrée [27]:

```
import matplotlib.pyplot as plt
plt.figure(figsize=[20,2])
plt.title('Temperature Time serie - 1st location')
plt.xlabel('Days')
plt.ylabel('Temperature °C')
T=df['T2M']
plt.plot(np.arange(0,len(T),1),T)
```

Out[27]:

```
[<matplotlib.lines.Line2D at 0x1f902a6cf08>]
```

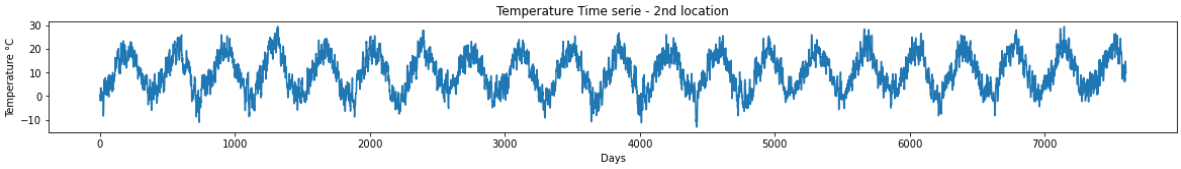


Entrée [28]:

```
df2[df2==-999]=np.nan
df2=df2.fillna(df2.mean())
import matplotlib.pyplot as plt
plt.figure(figsize=[20,2])
plt.title('Temperature Time serie - 2nd location')
plt.xlabel('Days')
plt.ylabel('Temperature °C')
T2=df2['T']
plt.plot(np.arange(0,len(T2),1),T2)
```

Out[28]:

```
[<matplotlib.lines.Line2D at 0x1f902abd688>]
```



Entrée [29]:

```
import datetime
import numpy as np
import scipy as sp
import scipy.fftpack
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
```

Entrée [30]:

```
T = T.values
N = len(T)
T2 = T2.values
N2 = len(T2)
```

Entrée [31]:

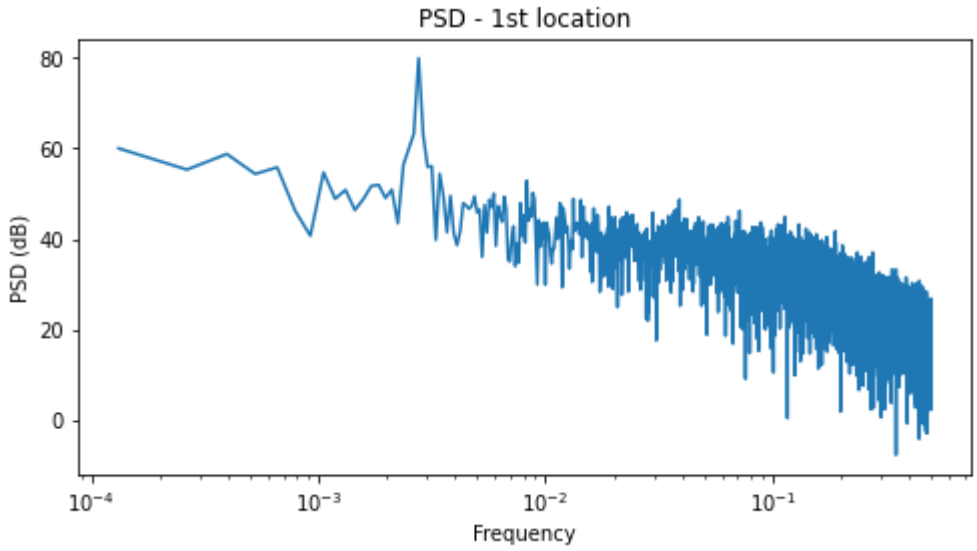
```
T_fft = sp.fftpack.fft(T)
T_psd = np.abs(T_fft) ** 2
fftfreq = sp.fftpack.fftfreq(N, 1)
T2_fft = sp.fftpack.fft(T2)
T2_psd = np.abs(T2_fft) ** 2
fftfreq2 = sp.fftpack.fftfreq(N2, 1)
```

Entrée [32]:

```
#The fftfreq() function returns positive and negative frequencies.
#We are only interested in positive frequencies here, as we have a real signal:
i = fftfreq > 0
i2 = fftfreq2 > 0
```

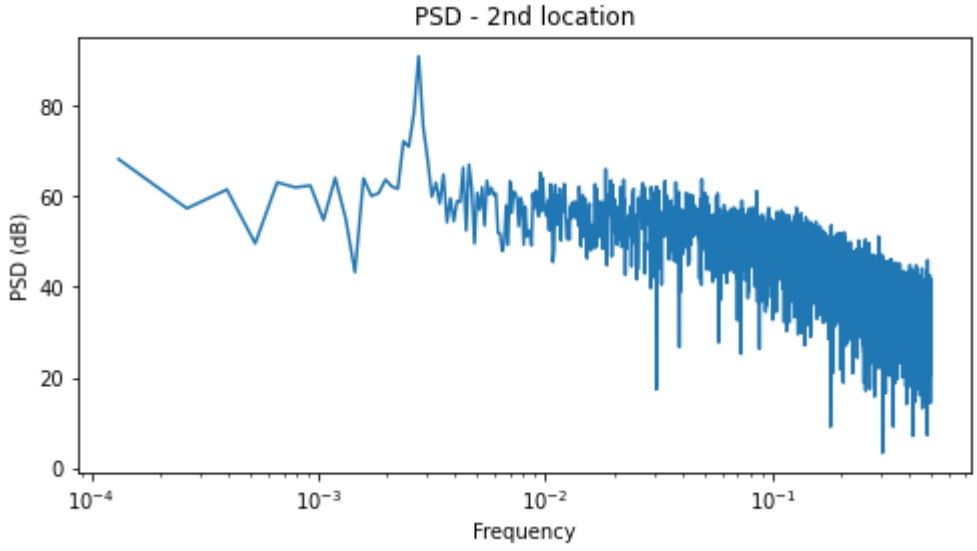
Entrée [33]:

```
plt.figure(figsize=[8, 4])
plt.plot(fftfreq[i], 10 * np.log10(T_psd[i]))
plt.title('PSD - 1st location')
plt.xlabel('Frequency')
plt.ylabel('PSD (dB)')
plt.xscale('log')
```



Entrée [34]:

```
plt.figure(figsize=[8, 4])
plt.plot(fftfreq2[i2], 10 * np.log10(T2_psd[i2]))
plt.title('PSD - 2nd location')
plt.xlabel('Frequency')
plt.ylabel('PSD (dB)')
plt.xscale('log')
```

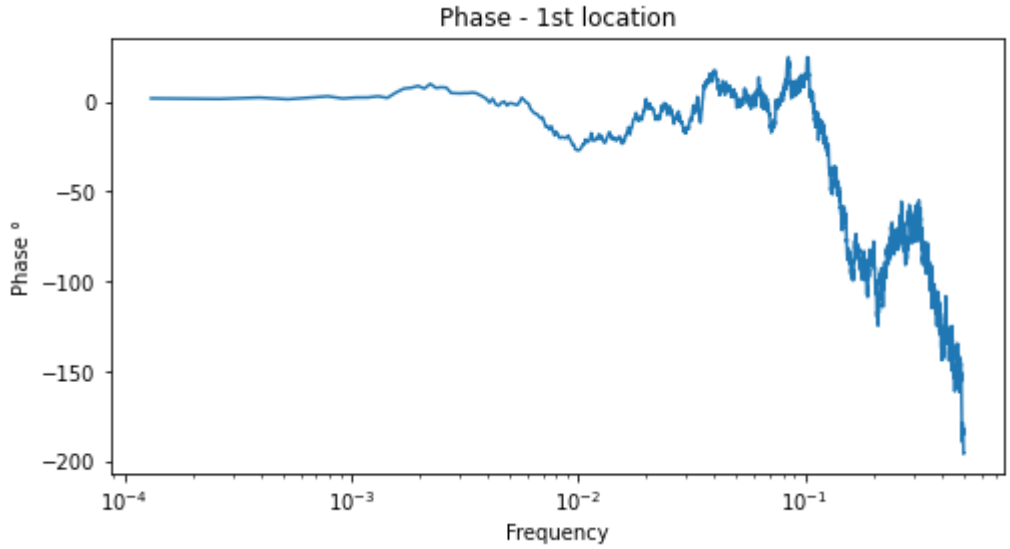


Entrée [35]:

```
phase=np.unwrap(np.angle(T_fft))
phase2=np.unwrap(np.angle(T2_fft))
```

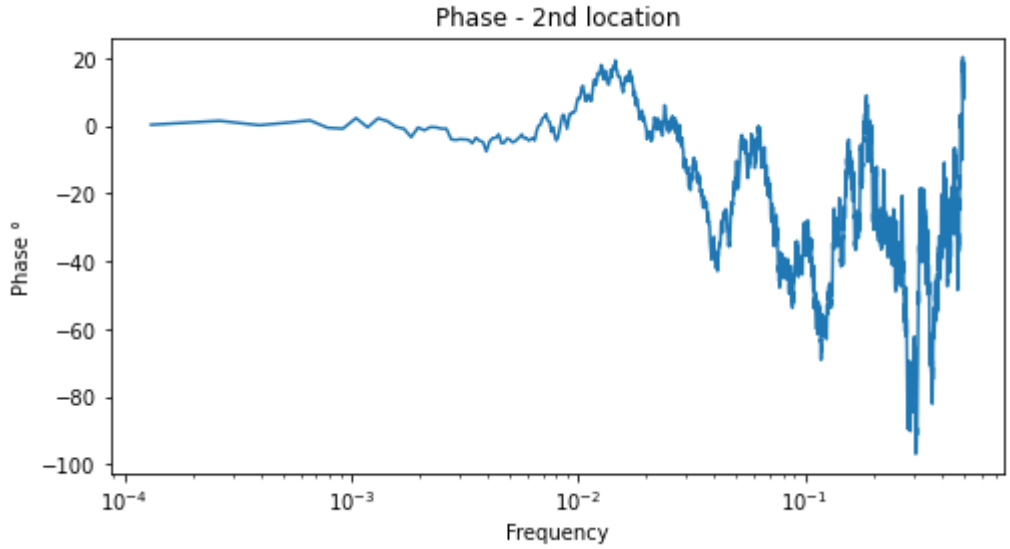
Entrée [36]:

```
plt.figure(figsize=[8, 4])
plt.title('Phase - 1st location')
plt.plot(fftfreq[i],phase[i])
plt.xlabel('Frequency')
plt.ylabel('Phase °')
plt.xscale('log')
```



Entrée [37]:

```
plt.figure(figsize=[8, 4])
plt.title('Phase - 2nd location')
plt.plot(fftfreq2[i2],phase2[i2])
plt.xlabel('Frequency')
plt.ylabel('Phase °')
plt.xscale('log')
```



Entrée [38]:

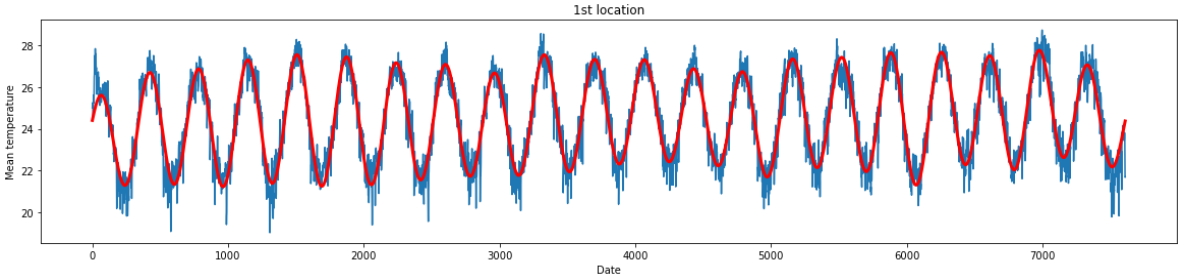
```
# filtrage
temp_fft_bis = T_fft.copy()
temp_fft_bis[np.abs(fftfreq) > 0.003] = 0
# filtrage
temp_fft_bis2 = T2_fft.copy()
temp_fft_bis2[np.abs(fftfreq2) > 0.003] = 0
```

Entrée [46]:

```
temp_slow = np.real(sp.fftpack.ifft(temp_fft_bis))
plt.figure(figsize=[20,4])
plt.plot(np.arange(0,len(T),1),T)
plt.plot(np.arange(0,len(T),1), temp_slow, 'r-',lw=3)
plt.title('1st location')
plt.xlabel('Date')
plt.ylabel('Mean temperature')
```

Out[46]:

```
Text(0, 0.5, 'Mean temperature')
```

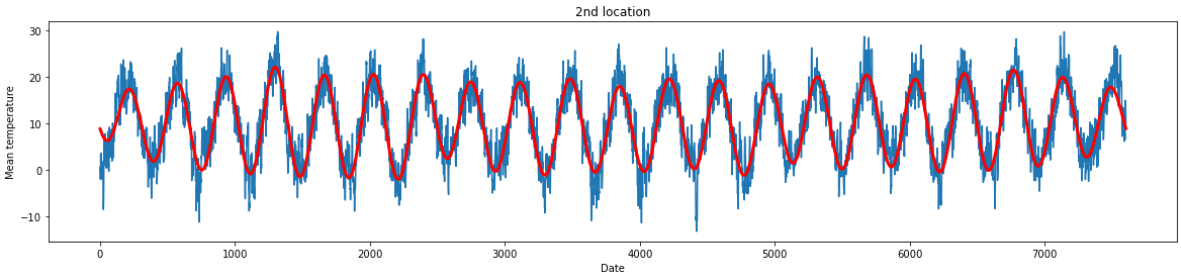


Entrée [45]:

```
temp_slow2 = np.real(sp.fftpack.ifft(temp_fft_bis2))
plt.figure(figsize=[20,4])
plt.title('2nd location')
plt.plot(np.arange(0,len(T2),1),T2)
plt.plot(np.arange(0,len(T2),1), temp_slow2, 'r-',lw=3)
plt.xlabel('Date')
plt.ylabel('Mean temperature')
```

Out[45]:

```
Text(0, 0.5, 'Mean temperature')
```



Entrée [ ]:

```

```