Jupyter_Notebook_Wind_Energy_Planning_Analysis_for_Locations_in_Libya

September 1, 2021

0.1 Planning and Analysis for Wind Energy in Libya

0.1.1 Comparison of Capacity Factor and Variability of Wind Enery for Coast, Southern, Western and Eastern Libyan Cities

Data have already been downloaded and stored.
17 Locations in Libya and Berlin in Germany for a purpuse of comparison.
Data are retrieved from NREL's Developer Network: https://developer.nrel.gov/

In [1]:
```
import os  # for getting environment variables
import pathlib  # for finding the example dataset
import pvlib
import pandas as pd  # for data wrangling
import matplotlib.pyplot as plt  # for visualization

from pvlib.iotools import get_pvgis_tmy
from pvlib import clearsky, solarposition, irradiance
import numpy as np
```

0.1.2 Reading a TMY dataset

In [2]:
```
import os
import pathlib  # for finding the example dataset
os.getcwd()
os.chdir("C:/Users/Mhdella/Desktop/TMY Libyan Cities")
```

In [3]:
```
cities=['Tripoli','Zuwara', 'Tarhunah', 'Msallata', 'Ghanima', 'Gharyan', 'Misurata','Sirte', 'Magrun', 'Benghazi',
        'Derna', 'Tobruk','Houn','Gadamis','Sabha','Kufra', 'Jaghbub','Berlin']

df_dt=['df_Trip','df_Zuwara', 'df_Tarhunah', 'df_Msallata', 'df_Ghanima', 'df_Gharyan', 'df_Misurata',
       'df_Sirte','df_Magrun','df_Benghazi', 'df_Derna', 'df_Tobruk', 'df_Houn','df_Gadamis','df_Sabha',
       'df_Kufra','df_Jaghbub', 'df_Berlin']

for i in np.arange(len(cities)):

    data = pd.read_csv(cities[i]+'_get_pvgis_tmy.csv',index_col='time(UTC)')
    vars()[df_dt[i]] = data
    vars()[df_dt[i]].index=pd.to_datetime(vars()[df_dt[i]].index, format='%Y-%m-%d')

    vars()[df_dt[i]]['Year']=vars()[df_dt[i]].index.year
    vars()[df_dt[i]]['Month']=vars()[df_dt[i]].index.month
    vars()[df_dt[i]]['Day']=vars()[df_dt[i]].index.day
    vars()[df_dt[i]]['Hour']=vars()[df_dt[i]].index.hour
    vars()[df_dt[i]]['Year'] = 2021
    vars()[df_dt[i]].index=pd.to_datetime(df_Trip[['Year', 'Month', 'Day', 'Hour']])

# df_Berlin
```

0.1.3 Wind speed at 10 meter (WS10m) for some Libyan CIties

WS10m: 10-m total wind speed (m/s)

In [4]:
```
cities=['Tripoli','Zuwara', 'Tarhunah', 'Msallata', 'Ghanima', 'Gharyan', 'Misurata','Sirte', 'Magrun', 'Benghazi',
        'Derna', 'Tobruk','Houn','Gadamis','Sabha','Kufra', 'Jaghbub','Berlin']

df_dt=['df_Trip','df_Zuwara', 'df_Tarhunah', 'df_Msallata', 'df_Ghanima', 'df_Gharyan', 'df_Misurata',
       'df_Sirte','df_Magrun','df_Benghazi', 'df_Derna', 'df_Tobruk', 'df_Houn','df_Gadamis','df_Sabha',
       'df_Kufra','df_Jaghbub', 'df_Berlin']


df_tem=pd.DataFrame({'Tripoli':df_Trip['WS10m'].resample('M').mean()})
df_tem
for i in np.arange(len(cities)):
    df_tem[cities[i]]=vars()[df_dt[i]]['WS10m'].resample('M').mean()

Wind10m_comparison=df_tem

# Wind10m_comparison.describe()
Wind10m_comparison
```

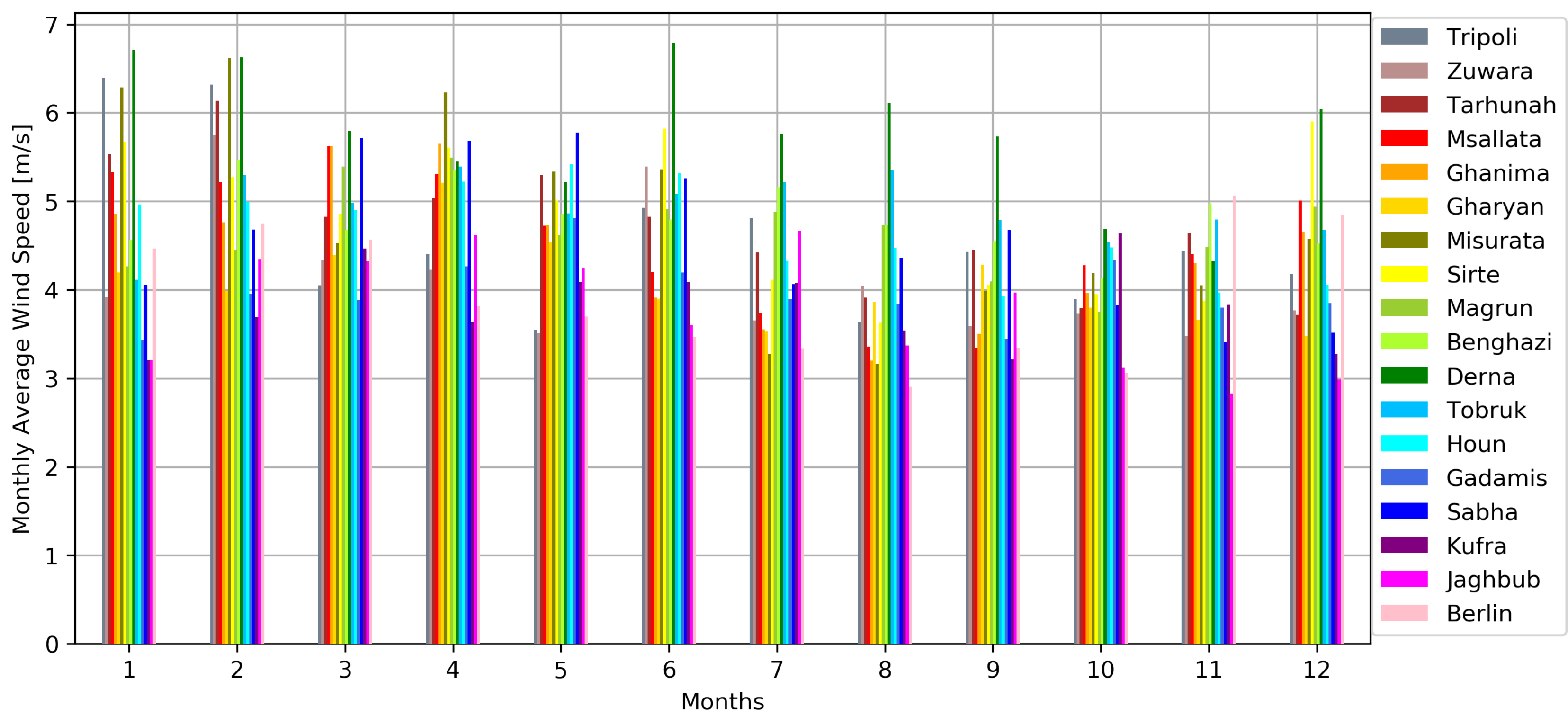
Out[4]:

	Tripoli	Zuwara	Tarhunah	Msallata	Ghanima	Gharyan	\
2021-01-31	6.394113	3.921089	5.527608	5.325591	4.859341	4.193978	
2021-02-28	6.317589	5.745610	6.136042	5.213274	4.760432	3.987083	
2021-03-31	4.051640	4.336747	4.823172	5.626317	5.627164	4.389933	
2021-04-30	4.400028	4.227639	5.032042	5.312806	5.649750	5.206111	
2021-05-31	3.546062	3.509637	5.294825	4.723374	4.729906	4.543293	
2021-06-30	4.923917	5.390097	4.822375	4.201042	3.909625	3.898389	
2021-07-31	4.810202	3.655282	4.421815	3.740202	3.554005	3.525417	
2021-08-31	3.637594	4.040027	3.911075	3.359610	3.198495	3.862513	

```
2021-09-30  4.427556  3.588278  4.452806  3.344556  3.501819  4.281306
2021-10-31  3.890954  3.731263  3.794583  4.274207  3.959368  3.800887
2021-11-30  4.442611  3.477889  4.642319  4.404361  4.301431  3.659153
2021-12-31  4.176116  3.768656  3.718387  5.010067  4.653642  3.474933

            Misurata     Sirte    Magrun  Benghazi     Derna    Tobruk  \
2021-01-31  6.287204  5.667231  4.263737  4.561129  6.709933  4.114530
2021-02-28  6.621682  5.272128  4.450997  5.466920  6.627321  5.297634
2021-03-31  4.527137  4.859247  5.391465  4.673710  5.793602  4.983925
2021-04-30  6.228361  5.606972  5.494014  5.350847  5.448764  5.392069
2021-05-31  5.335954  5.007513  4.620161  4.853925  5.217312  4.864624
2021-06-30  5.360847  5.822528  4.914861  4.790958  6.792208  5.083278
2021-07-31  3.278159  4.112433  4.878602  5.161694  5.760726  5.215296
2021-08-31  3.159798  3.628414  4.731358  4.727876  6.109341  5.350296
2021-09-30  3.988472  4.054931  4.093750  4.547625  5.729458  4.786306
2021-10-31  4.188723  3.947110  3.745901  4.131935  4.683253  4.539570
2021-11-30  4.047472  3.874056  4.486778  4.973750  4.323903  4.793139
2021-12-31  4.575202  5.902581  4.939167  4.521922  6.037876  4.672110

                Houn   Gadamis     Sabha     Kufra   Jaghbub    Berlin
2021-01-31  4.963978  3.436196  4.054597  3.205618  3.205336  4.462917
2021-02-28  4.988631  3.956012  4.683036  3.689836  4.343750  4.750818
2021-03-31  4.901237  3.886949  5.715027  4.465336  4.321828  4.567110
2021-04-30  5.222653  4.264514  5.682056  3.634667  4.617347  3.814361
2021-05-31  5.417298  4.813118  5.777903  4.088078  4.243642  3.698911
2021-06-30  5.319028  4.193597  5.257264  4.089528  3.602681  3.465681
2021-07-31  4.325874  3.893199  4.064704  4.073535  4.665497  3.341317
2021-08-31  4.470927  3.835161  4.361680  3.537876  3.373253  2.901949
2021-09-30  3.924097  3.443347  4.674139  3.215569  3.965444  3.343597
2021-10-31  4.475860  4.334261  3.826505  4.635578  3.118871  3.062392
2021-11-30  3.968806  3.798972  3.406056  3.832472  2.827222  5.062264
2021-12-31  4.057702  3.848804  3.512728  3.276129  2.987567  4.846237
```

In [5]:
```
Wind10m_comparison.index=Wind10m_comparison.index.month
```

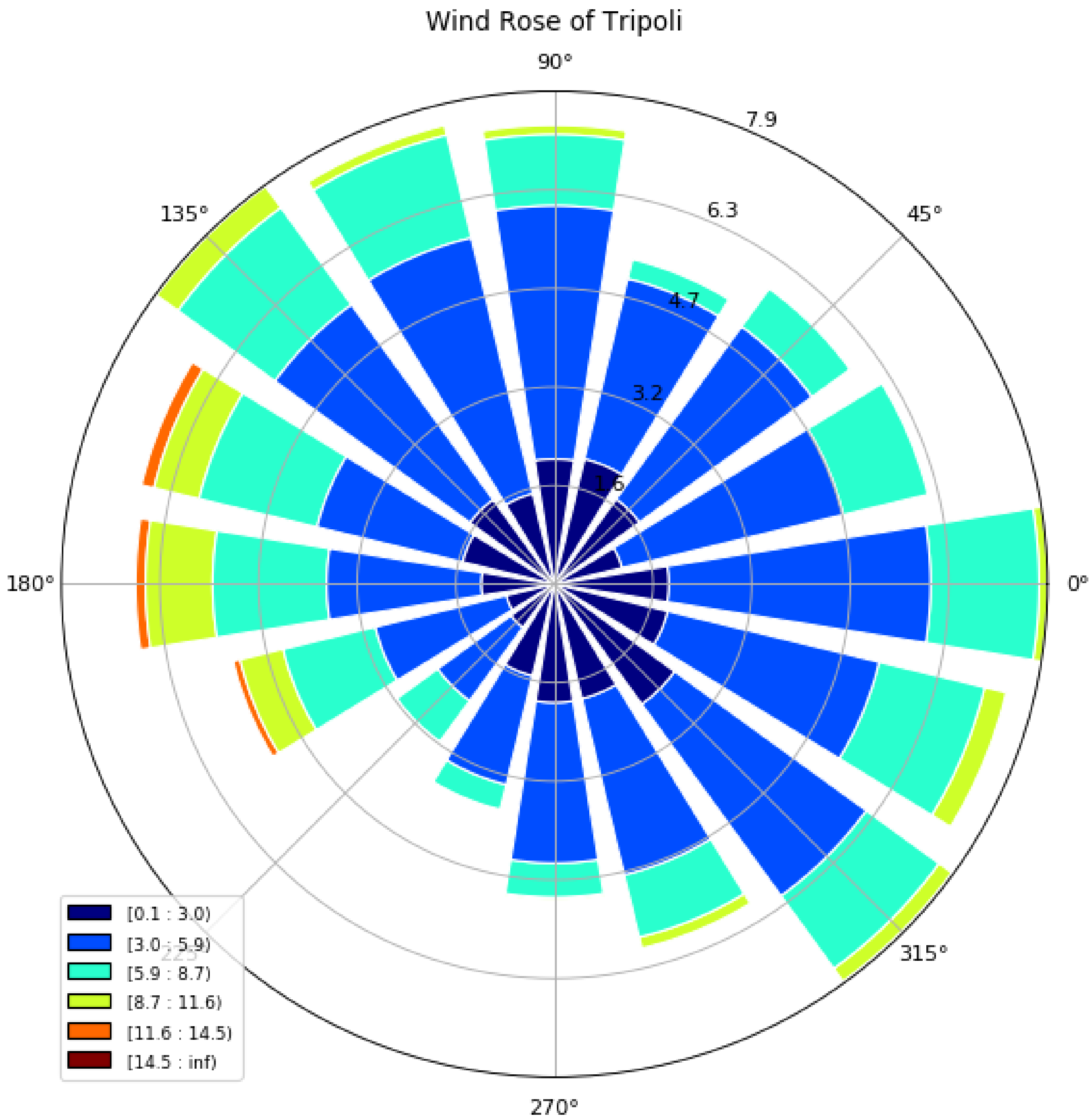
In [6]:
```
plt.rcParams['figure.figsize'] = [10, 5.0]
plt.rcParams['figure.dpi'] = 300

c = ['slategrey','rosybrown','brown','red','orange','gold','olive','yellow','yellowgreen',
     'greenyellow','green','deepskyblue','cyan','royalblue','blue','purple', 'magenta', 'pink']

Wind10m_comparison.plot.bar(zorder=3,color=c)

plt.xticks(rotation=0)
plt.xlabel('Months');
plt.ylabel('Monthly Average Wind Speed [m/s]');
plt.legend(bbox_to_anchor=(1.16, 1.01),loc='upper right')
plt.grid()
```

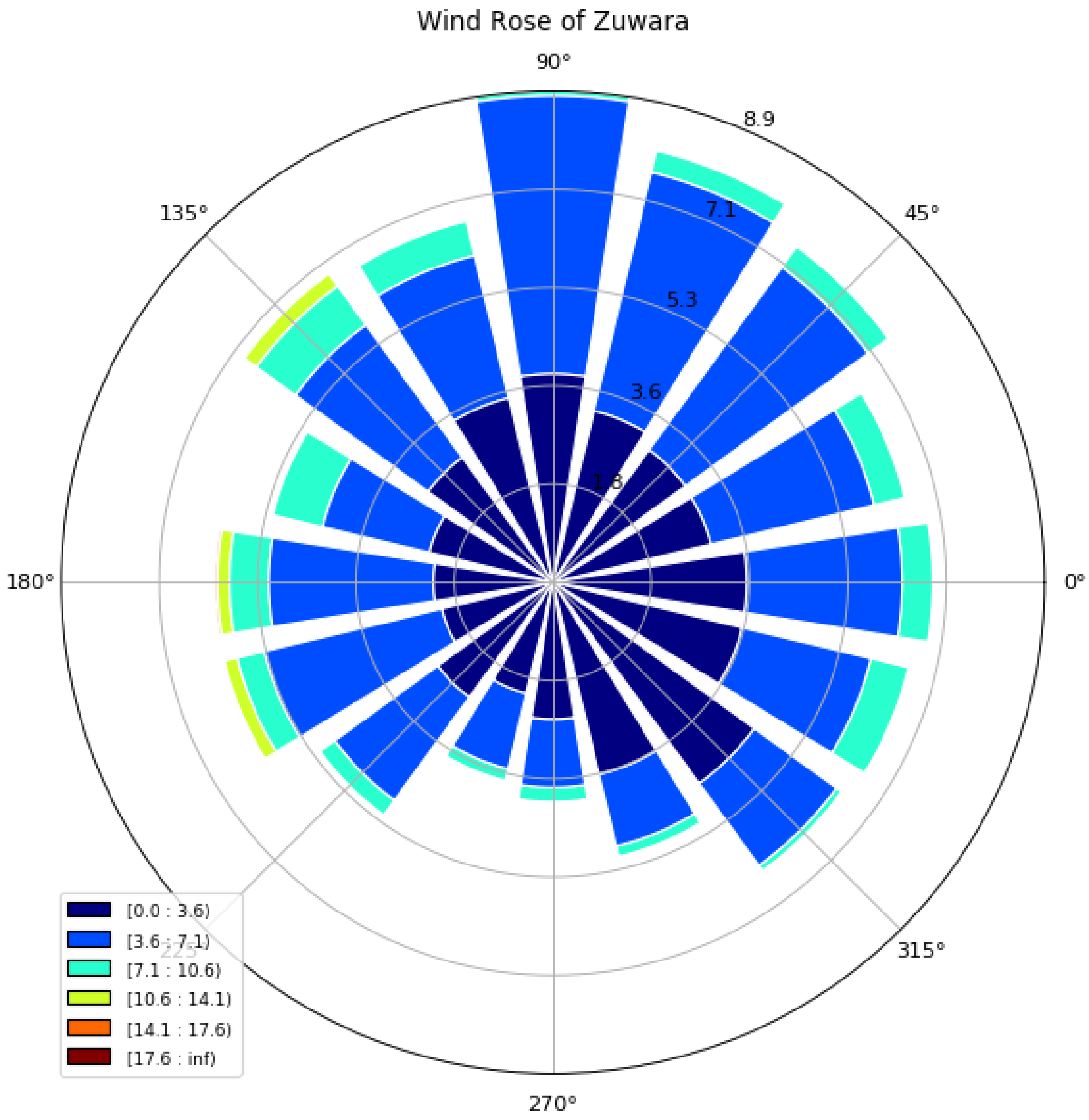


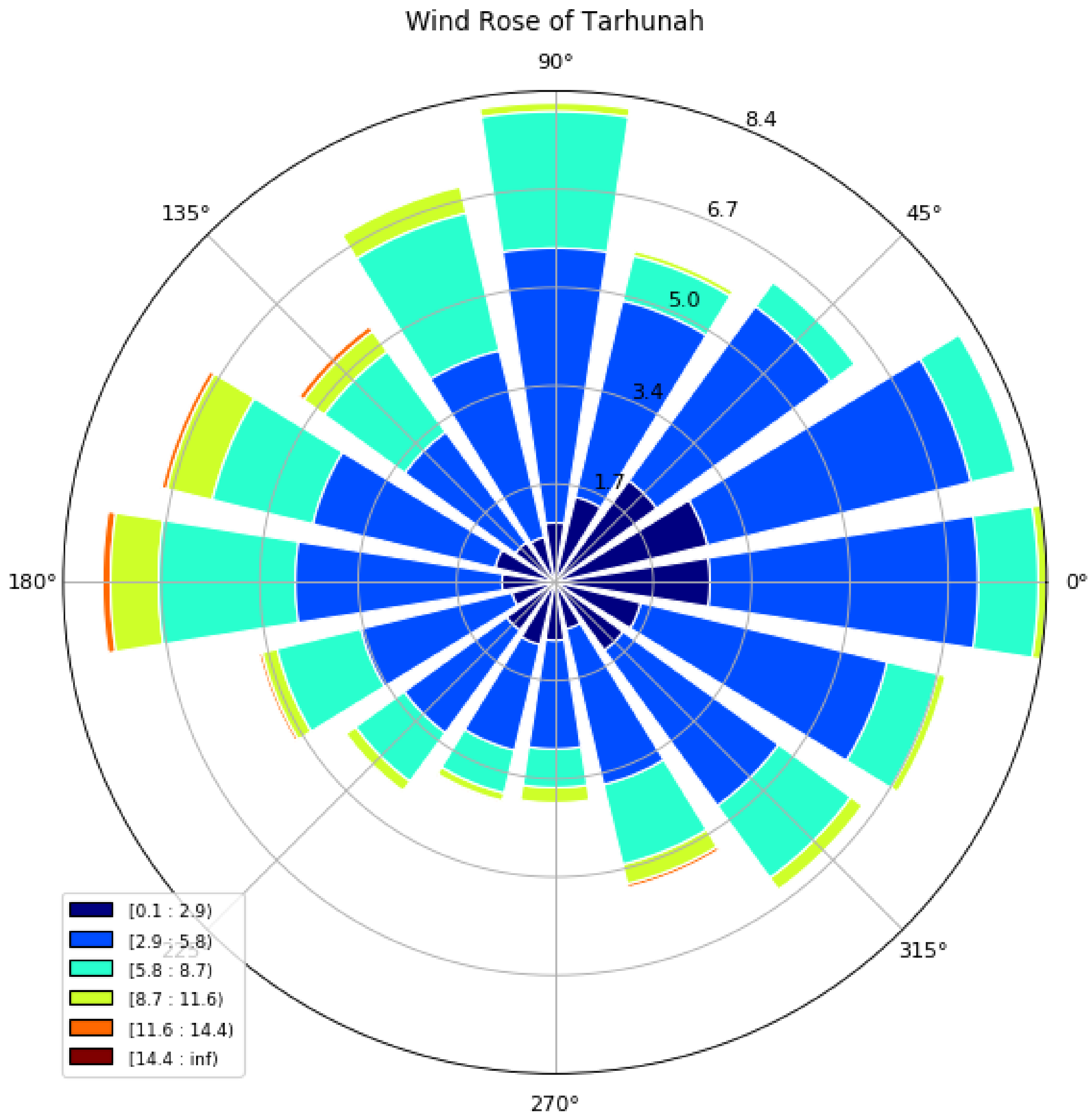
0.1.4 Plot wind roses

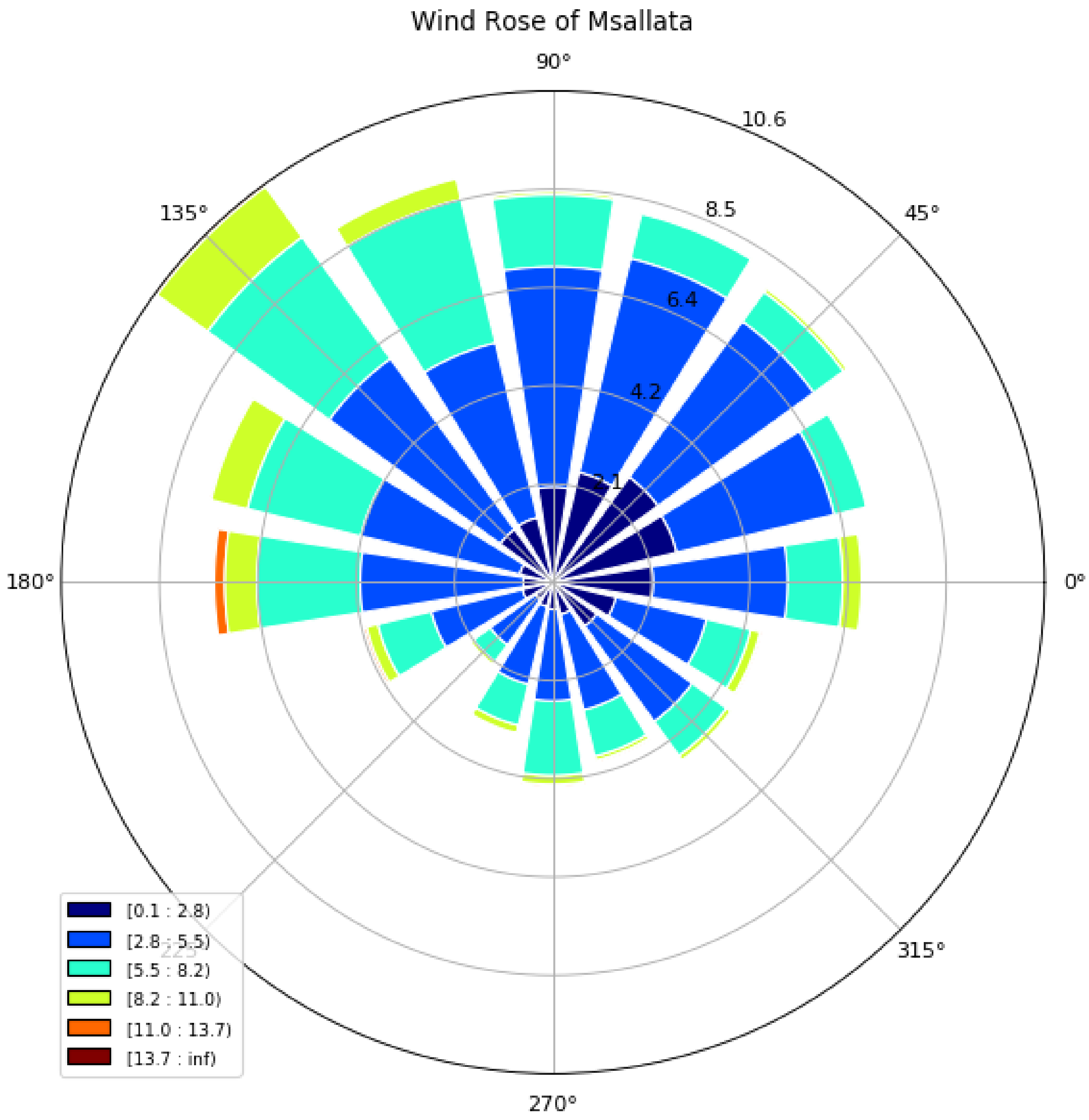
In [7]:
```
import matplotlib.pyplot as plt
from windrose import WindroseAxes
import matplotlib.cm as cm
import numpy as np

cities=['Tripoli','Zuwara', 'Tarhunah', 'Msallata', 'Ghanima', 'Gharyan', 'Misurata','Sirte', 'Magrun', 'Benghazi',
        'Derna', 'Tobruk','Houn','Gadamis','Sabha','Kufra', 'Jaghbub','Berlin']

df_dt=['df_Trip','df_Zuwara', 'df_Tarhunah', 'df_Msallata', 'df_Ghanima', 'df_Gharyan', 'df_Misurata',
```

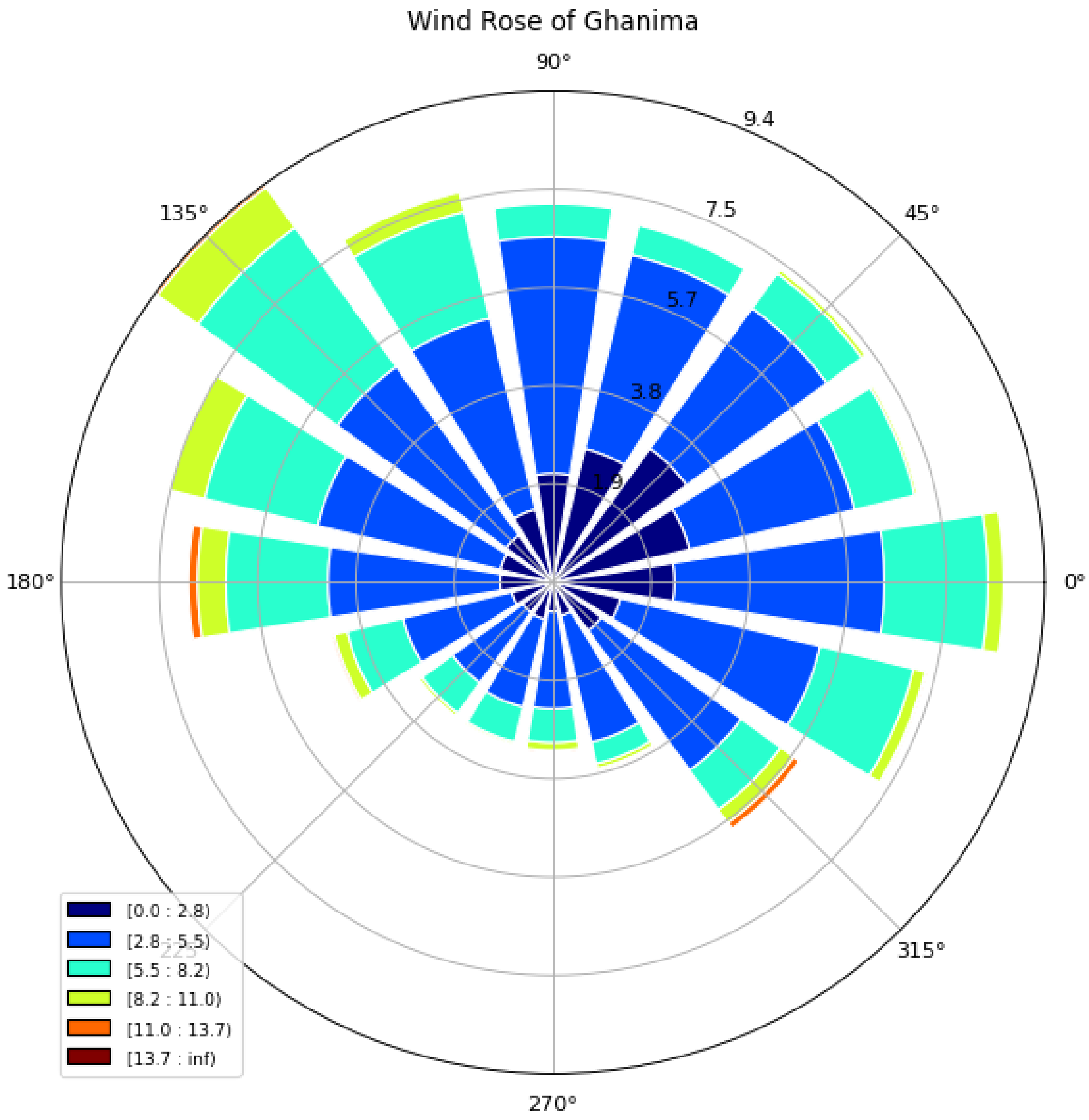
```
        'df_Sirte','df_Magrun','df_Benghazi', 'df_Derna', 'df_Tobruk', 'df_Houn','df_Gadamis','df_Sabha',
        'df_Kufra','df_Jaghbub', 'df_Berlin']

for i in np.arange(len(cities)):
    dt_tem=vars()[df_dt[i]][['WS10m','WD10m']]

    ax = WindroseAxes.from_ax()
    ax.bar(dt_tem['WD10m'], dt_tem['WS10m'], normed=True, opening=0.8, edgecolor='white')
    ax.set_legend()
    ax.set_title('Wind Rose of '+ cities[i])
```

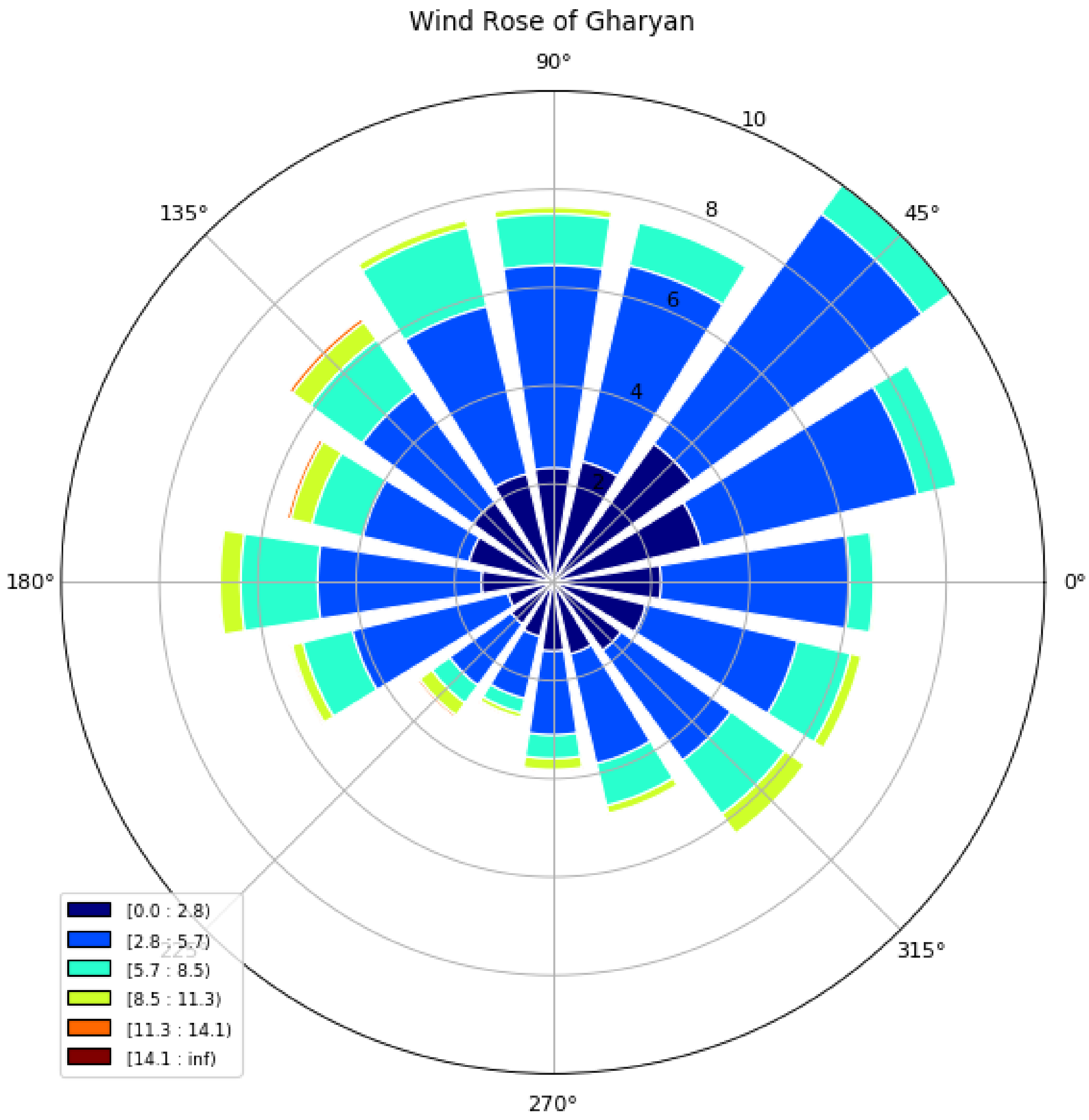


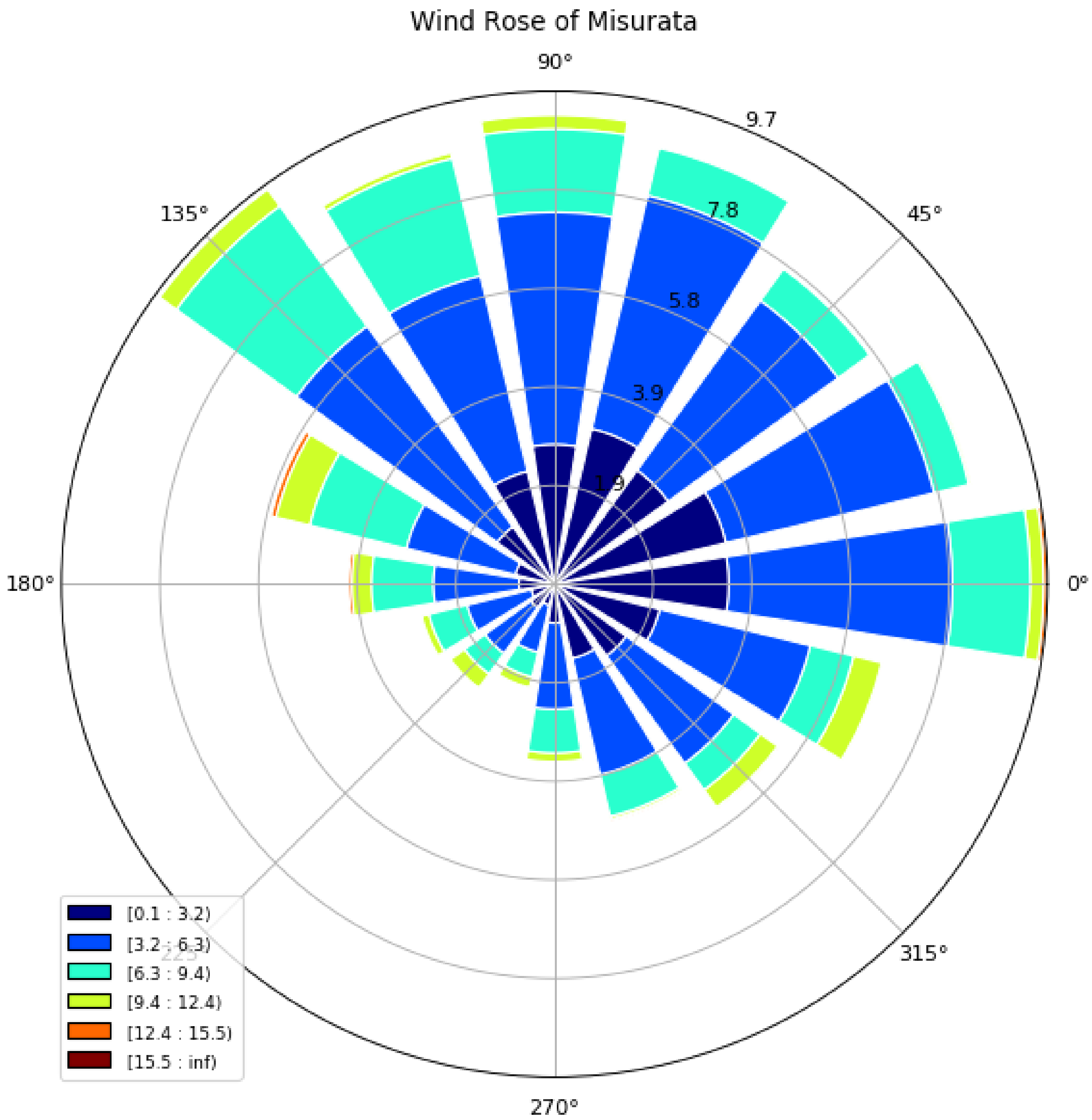

Wind Rose of Zuwara
90°
8.9
7.1
5.3
3.6
1.8
45°
135°
180°
0°
315°
270°
225
[0.0 : 3.6)
[3.6 : 7.1)
[7.1 : 10.6)
[10.6 : 14.1)
[14.1 : 17.6)
[17.6 : inf)


Wind Rose of Tarhunah
90°
135°
45°
180°
0°
315°
270°
225
8.4
6.7
5.0
3.4
1.7
[0.1 : 2.9)
[2.9 : 5.8)
[5.8 : 8.7)
[8.7 : 11.6)
[11.6 : 14.4)
[14.4 : inf)

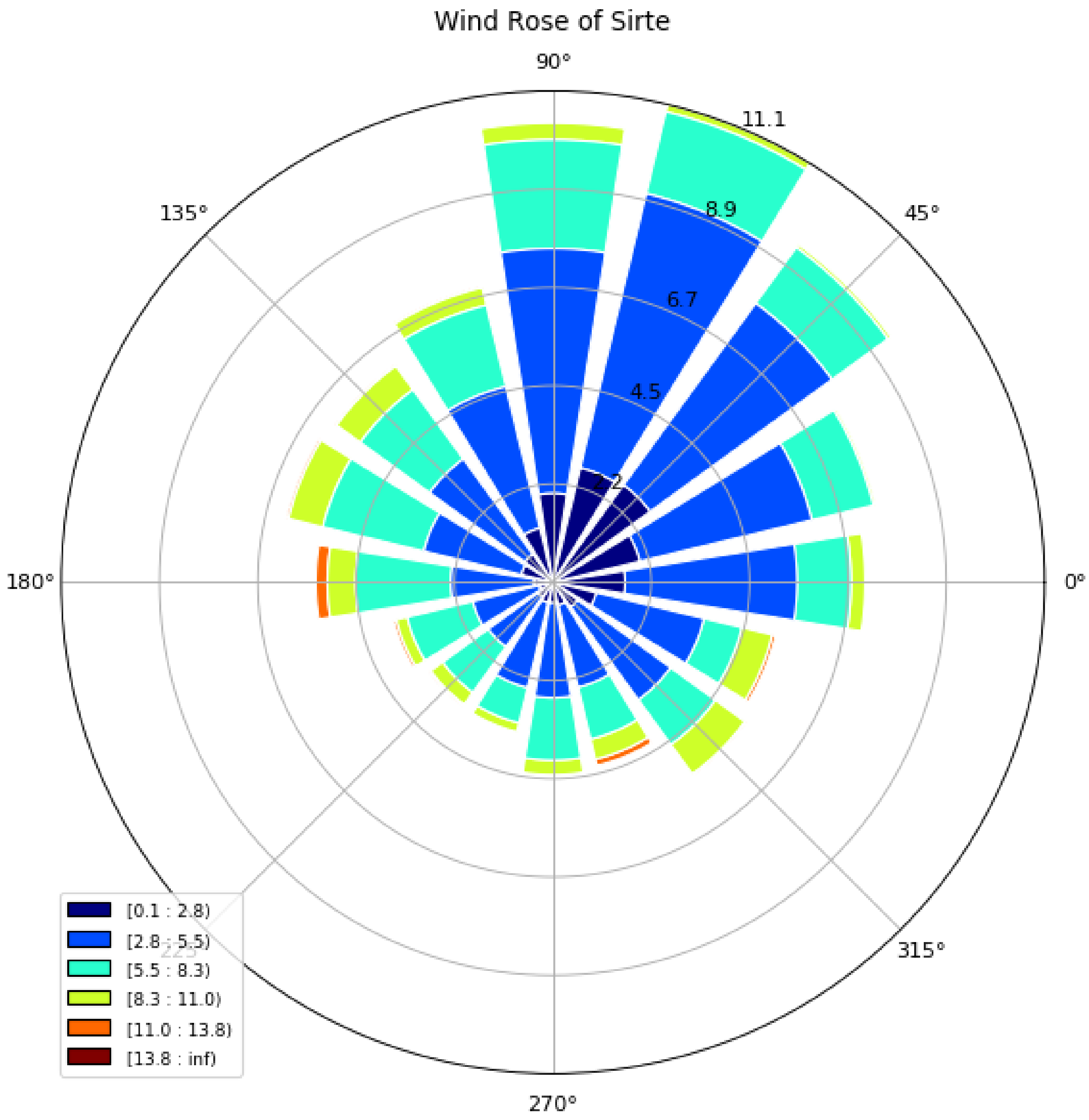

Wind Rose of Msallata
90°
135°
45°
180°
0°
315°
270°
225°
10.6
8.5
6.4
4.2
2.1
[0.1 : 2.8)
[2.8 : 5.5)
[5.5 : 8.2)
[8.2 : 11.0)
[11.0 : 13.7)
[13.7 : inf)

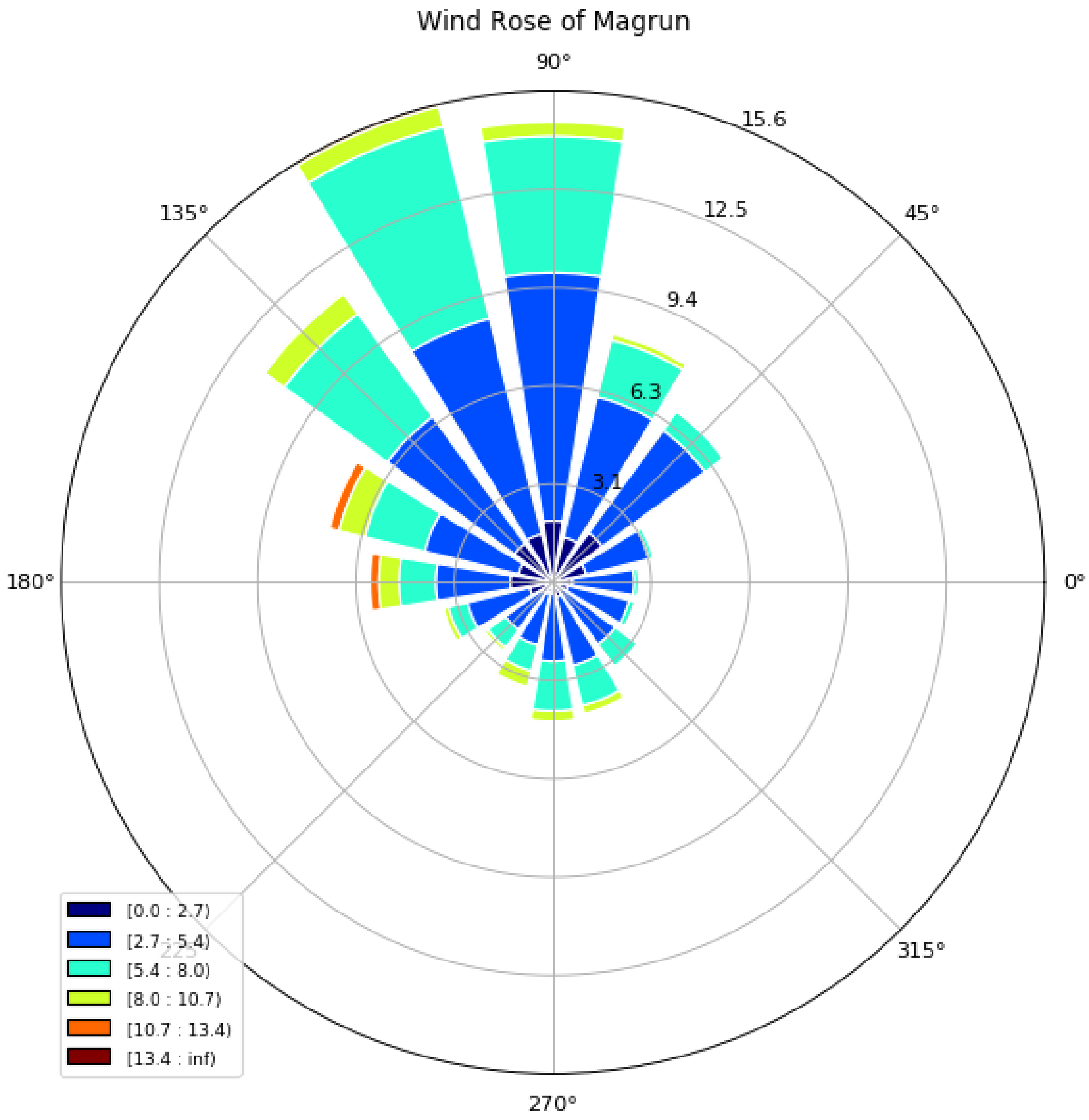

Wind Rose of Ghanima
90°
45°
0°
315°
270°
225°
180°
135°
1.9
3.8
5.7
7.5
9.4
[0.0 : 2.8)
[2.8 : 5.5)
[5.5 : 8.2)
[8.2 : 11.0)
[11.0 : 13.7)
[13.7 : inf)

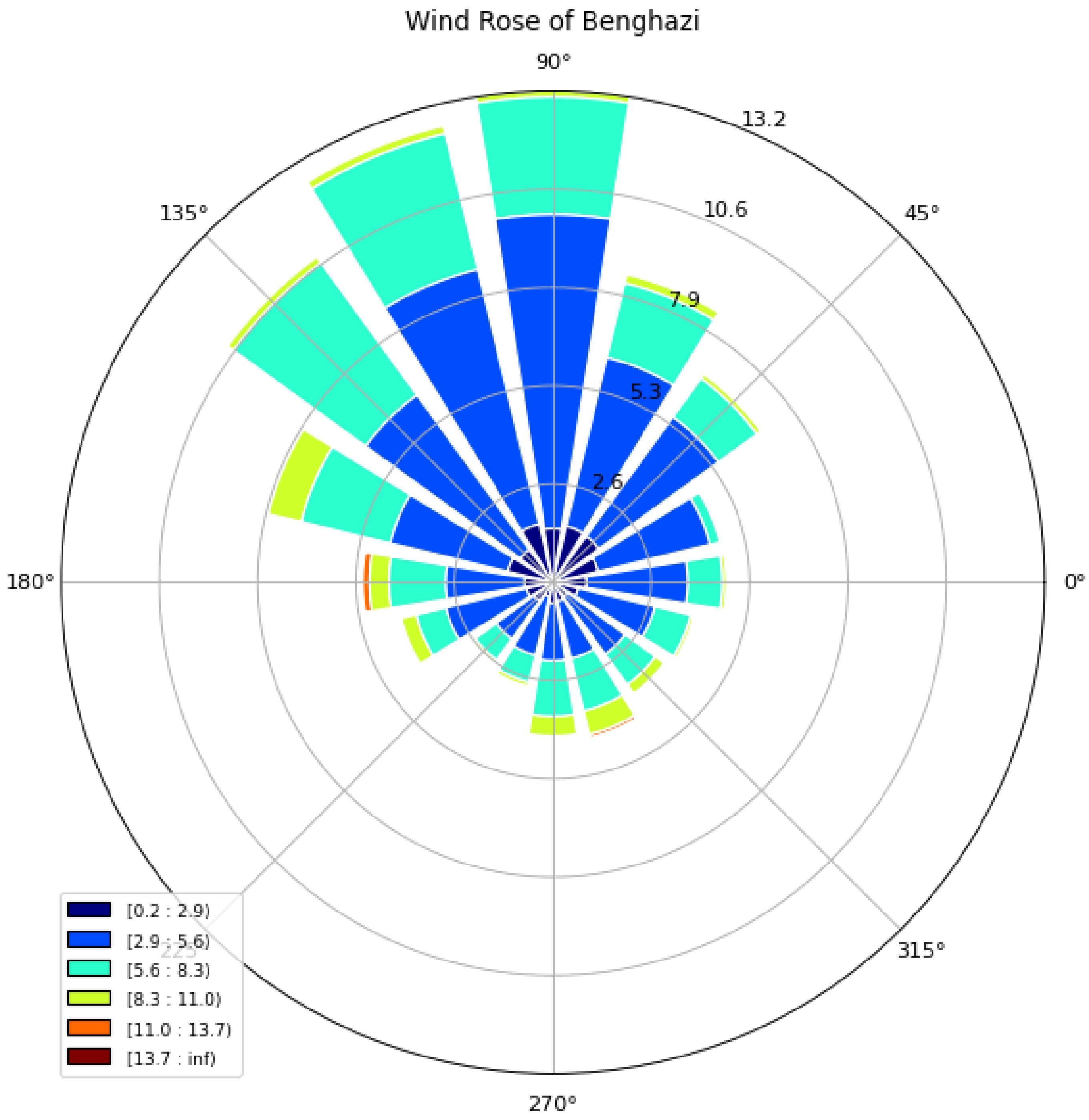

Wind Rose of Gharyan
90°
45°
0°
315°
270°
225°
180°
135°
2
4
6
8
10
[0.0 : 2.8)
[2.8 : 5.7)
[5.7 : 8.5)
[8.5 : 11.3)
[11.3 : 14.1)
[14.1 : inf)

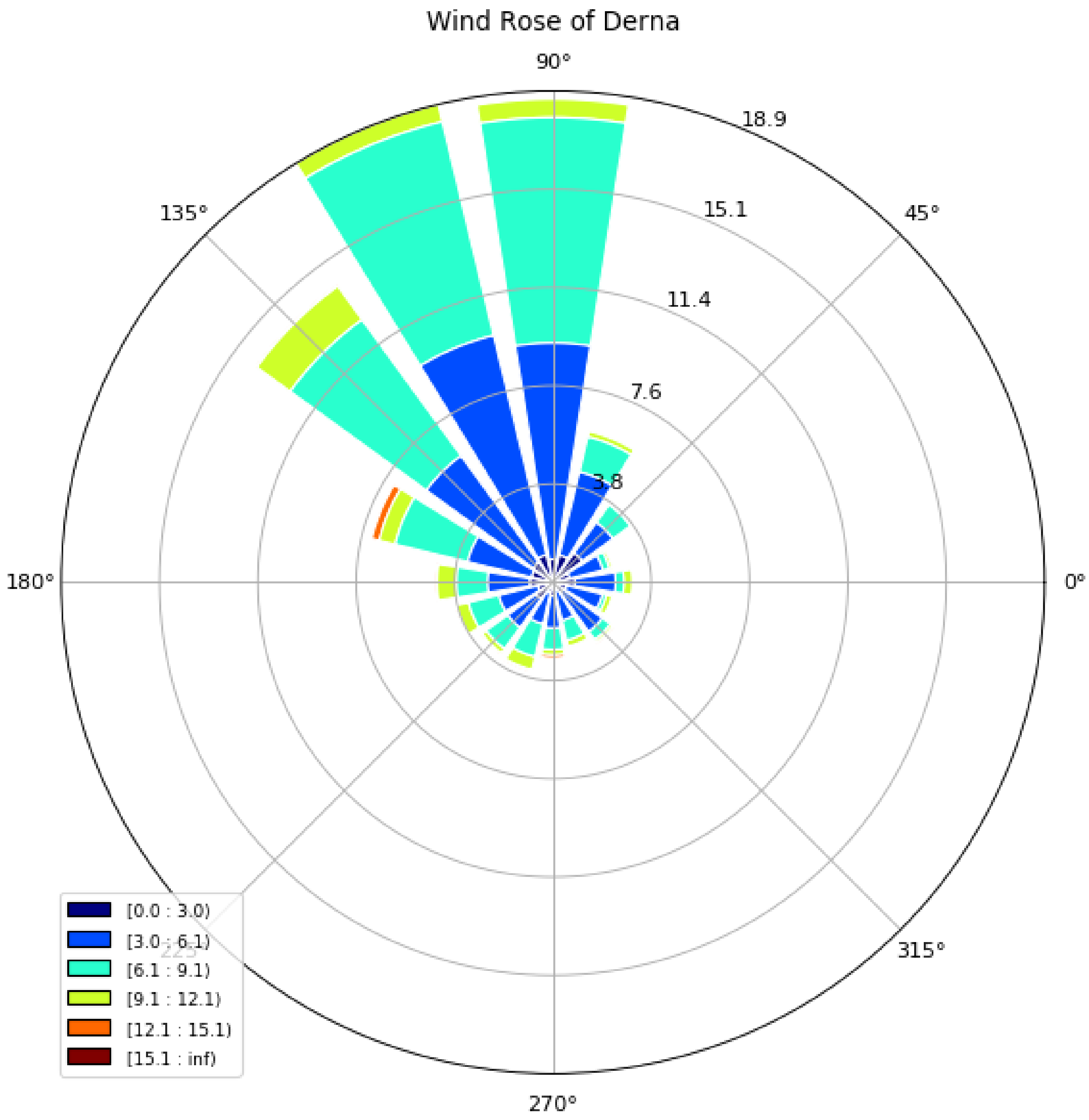


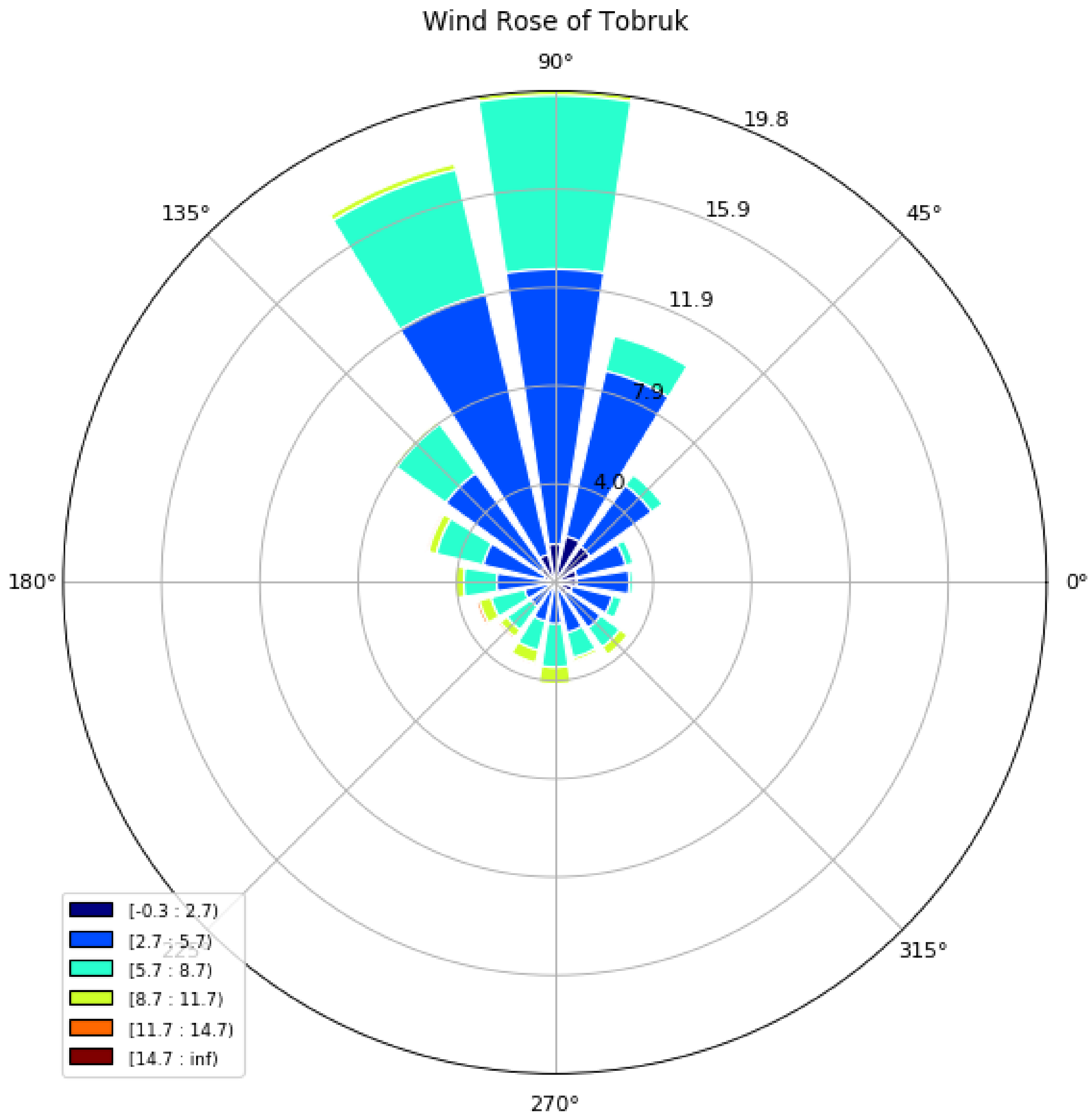
Wind Rose of Misurata
90°
45°
0°
315°
270°
225°
180°
135°
1.9
3.9
5.8
7.8
9.7
[0.1 : 3.2)
[3.2 : 6.3)
[6.3 : 9.4)
[9.4 : 12.4)
[12.4 : 15.5)
[15.5 : inf)

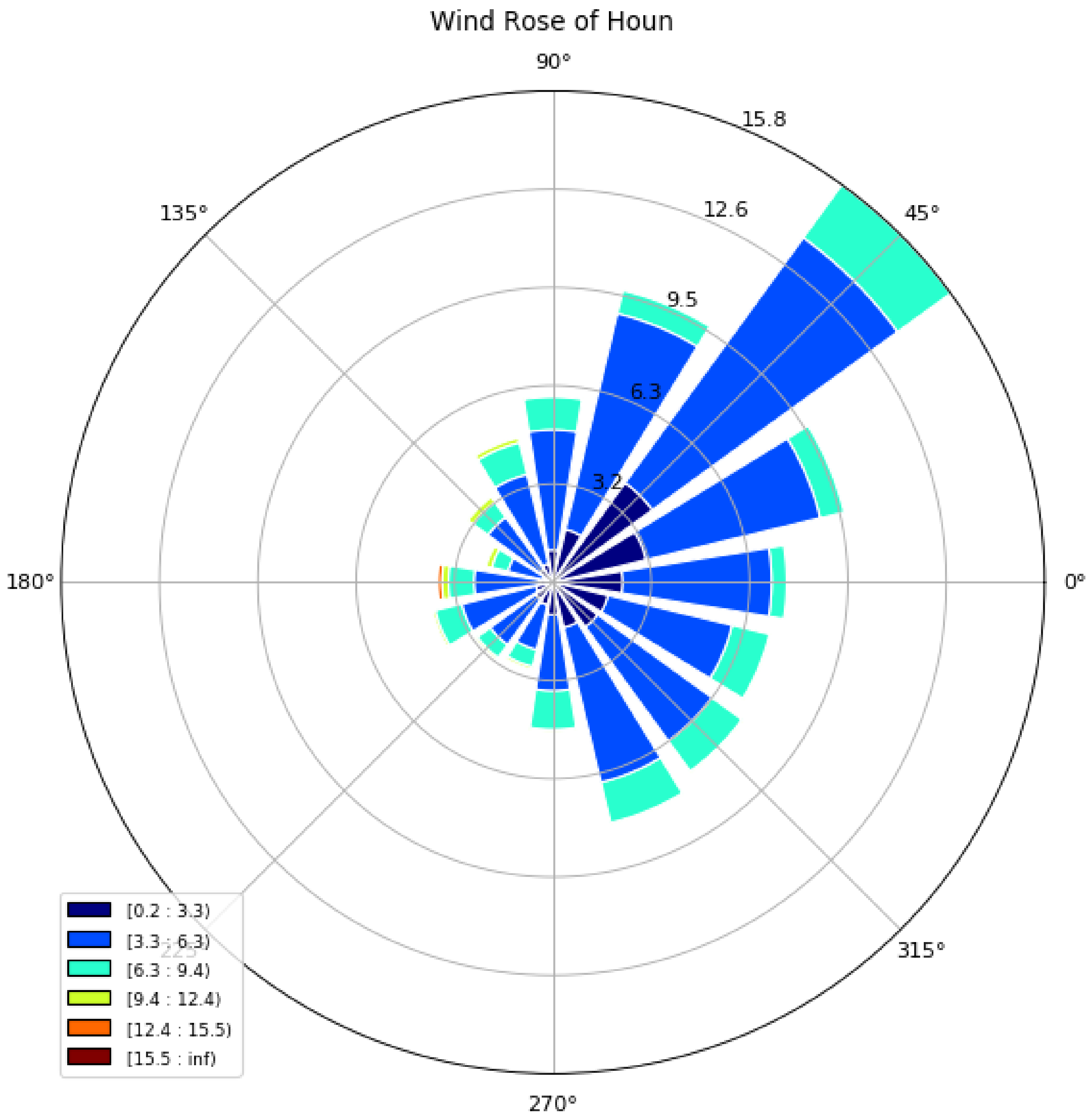

Wind Rose of Sirte
90°
45°
135°
180°
0°
315°
270°
2.2
4.5
6.7
8.9
11.1
[0.1 : 2.8)
[2.8 : 5.5)
[5.5 : 8.3)
[8.3 : 11.0)
[11.0 : 13.8)
[13.8 : inf)

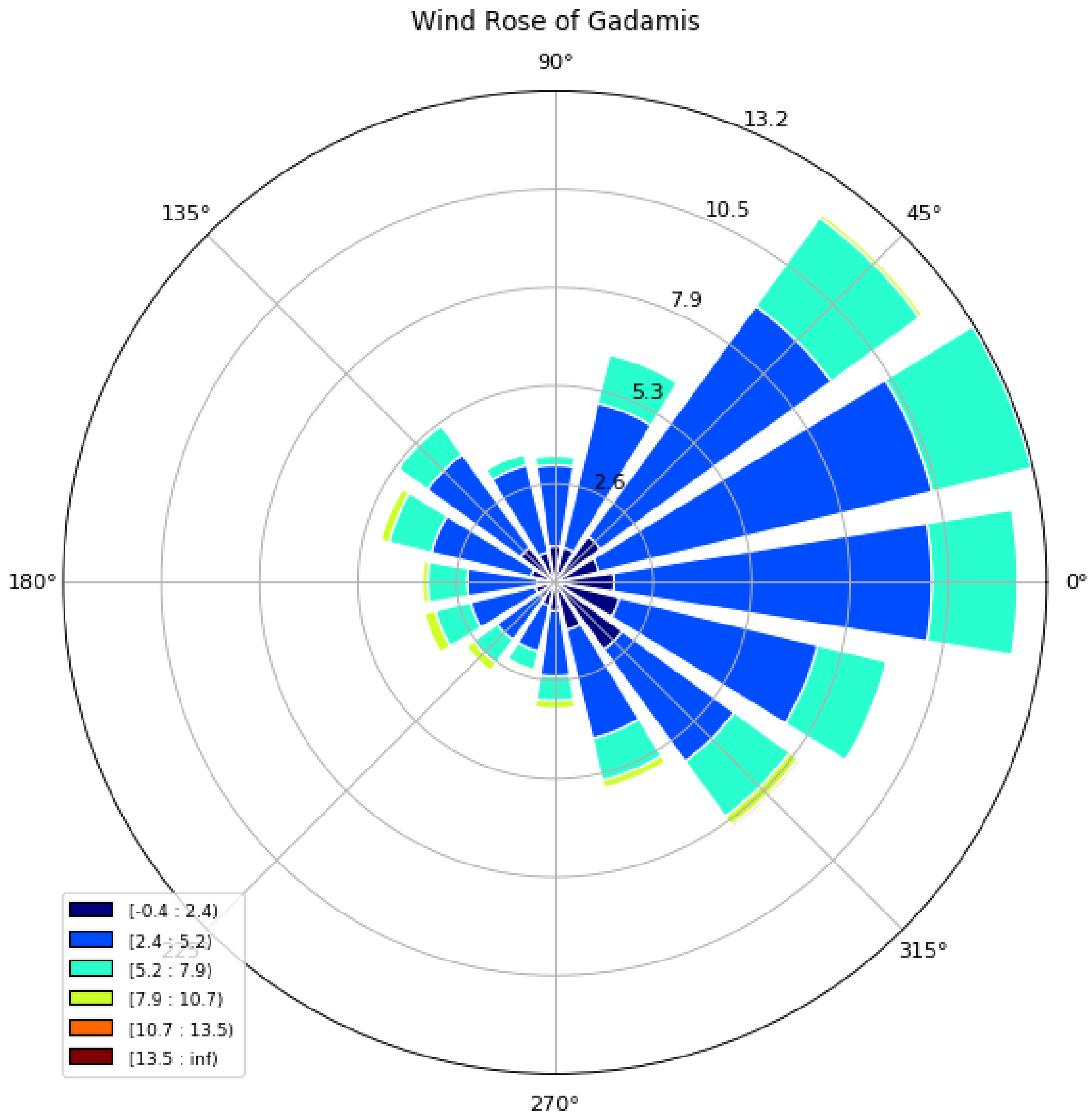

Wind Rose of Magrun
90°
135°
45°
180°
0°
225°
315°
270°
15.6
12.5
9.4
6.3
3.1
[0.0 : 2.7)
[2.7 : 5.4)
[5.4 : 8.0)
[8.0 : 10.7)
[10.7 : 13.4)
[13.4 : inf)

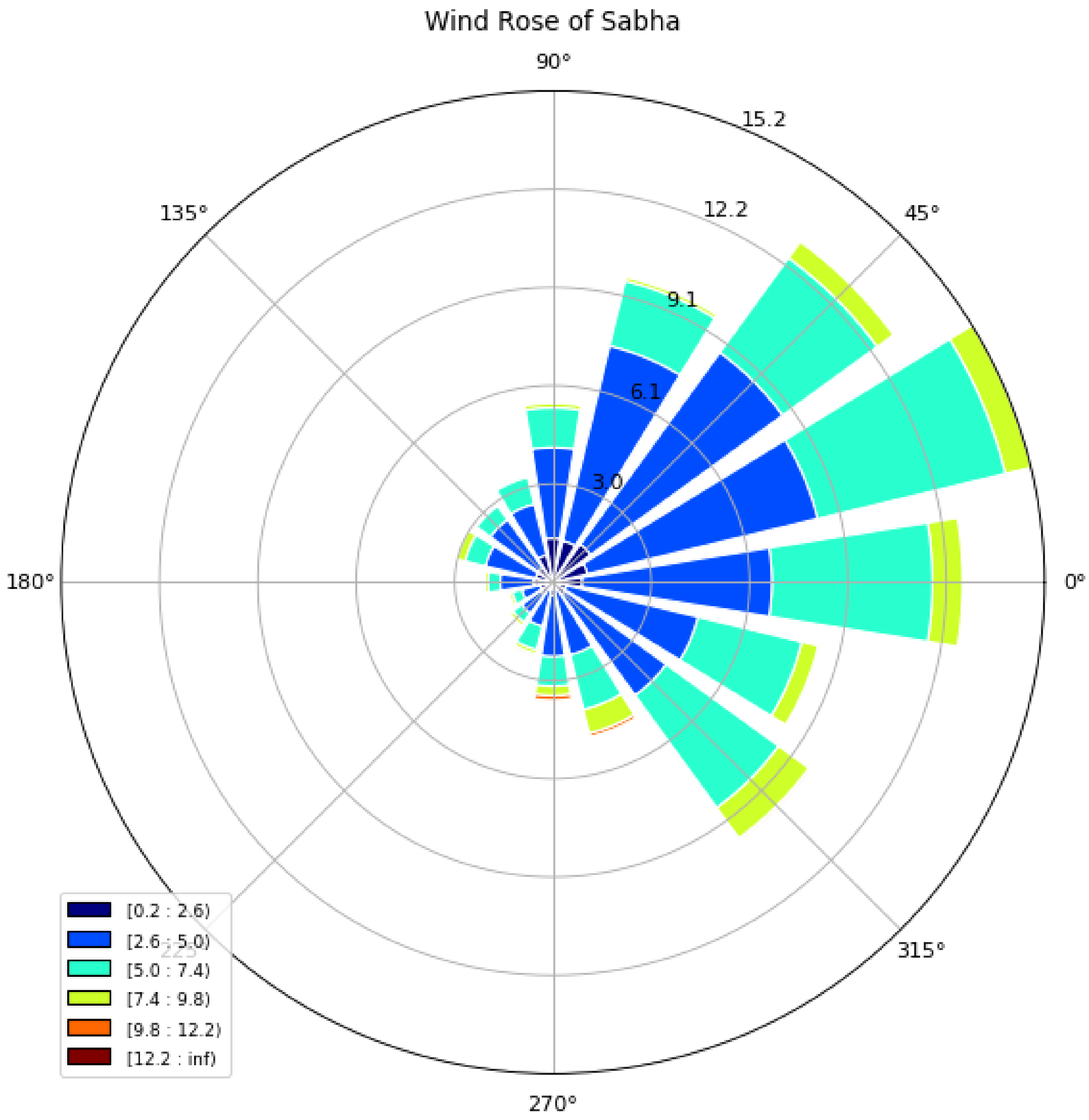

Wind Rose of Benghazi
90°
45°
0°
315°
270°
225°
180°
135°
2.6
5.3
7.9
10.6
13.2
[0.2 : 2.9)
[2.9 : 5.6)
[5.6 : 8.3)
[8.3 : 11.0)
[11.0 : 13.7)
[13.7 : inf)

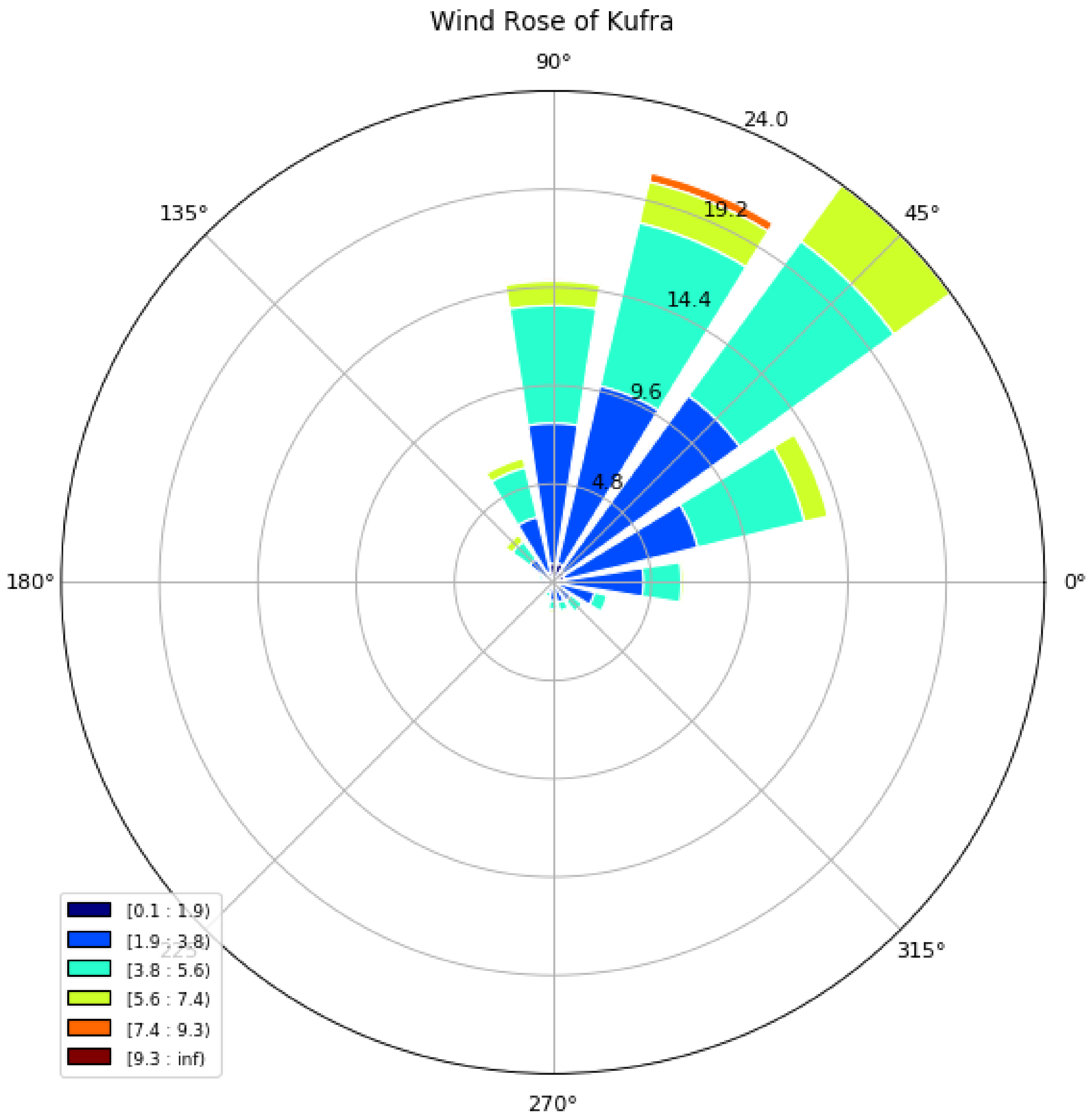

Wind Rose of Derna
90°
135°
45°
180°
0°
315°
270°
3.8
7.6
11.4
15.1
18.9
[0.0 : 3.0)
[3.0 : 6.1)
[6.1 : 9.1)
[9.1 : 12.1)
[12.1 : 15.1)
[15.1 : inf)


Wind Rose of Tobruk
90°
19.8
15.9
11.9
7.9
4.0
135°
45°
180°
0°
225°
315°
270°
[-0.3 : 2.7)
[2.7 : 5.7)
[5.7 : 8.7)
[8.7 : 11.7)
[11.7 : 14.7)
[14.7 : inf)

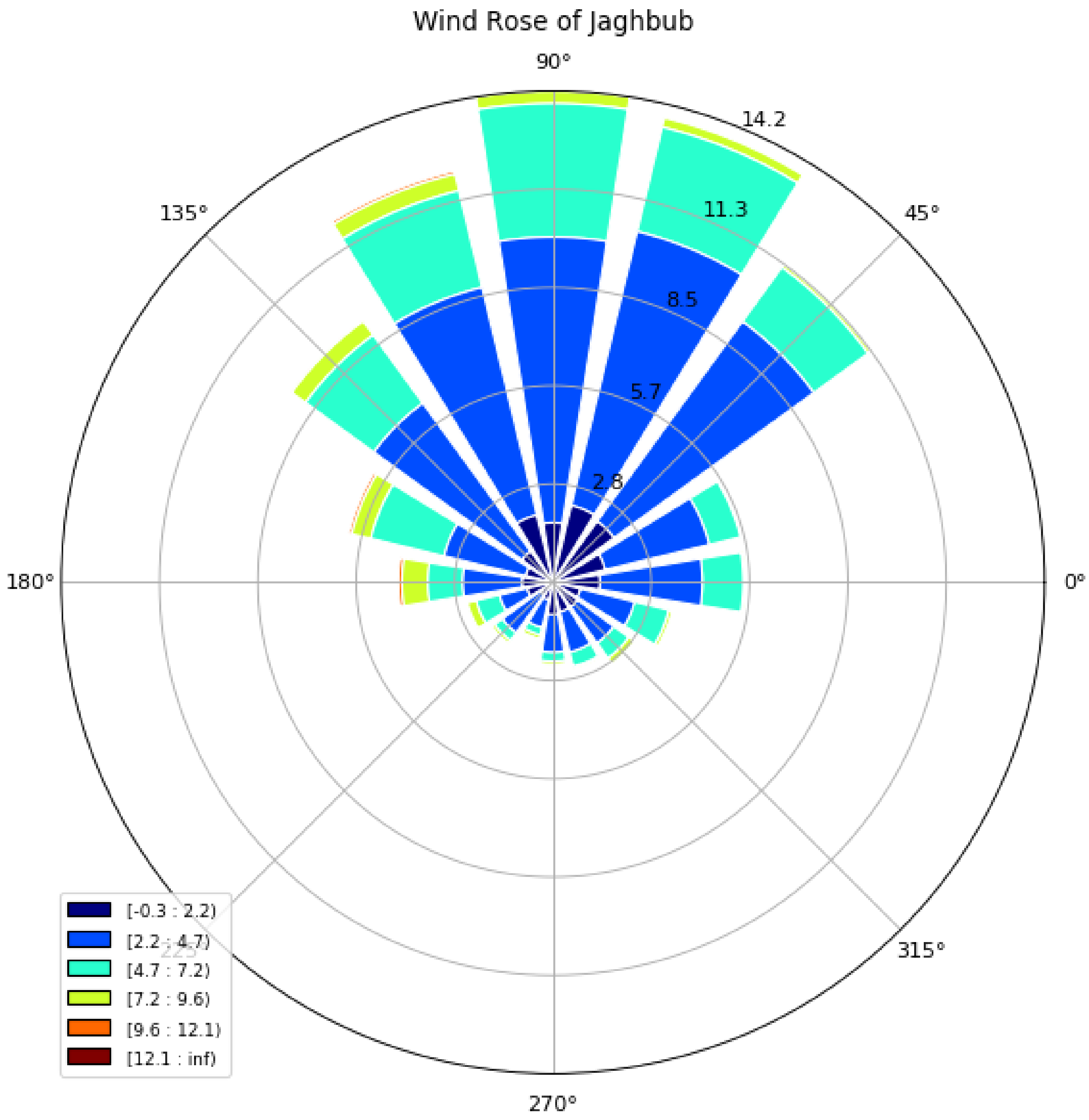

Wind Rose of Houn
90°
45°
0°
315°
270°
225°
180°
135°
3.2
6.3
9.5
12.6
15.8
[0.2 : 3.3)
[3.3 : 6.3)
[6.3 : 9.4)
[9.4 : 12.4)
[12.4 : 15.5)
[15.5 : inf)

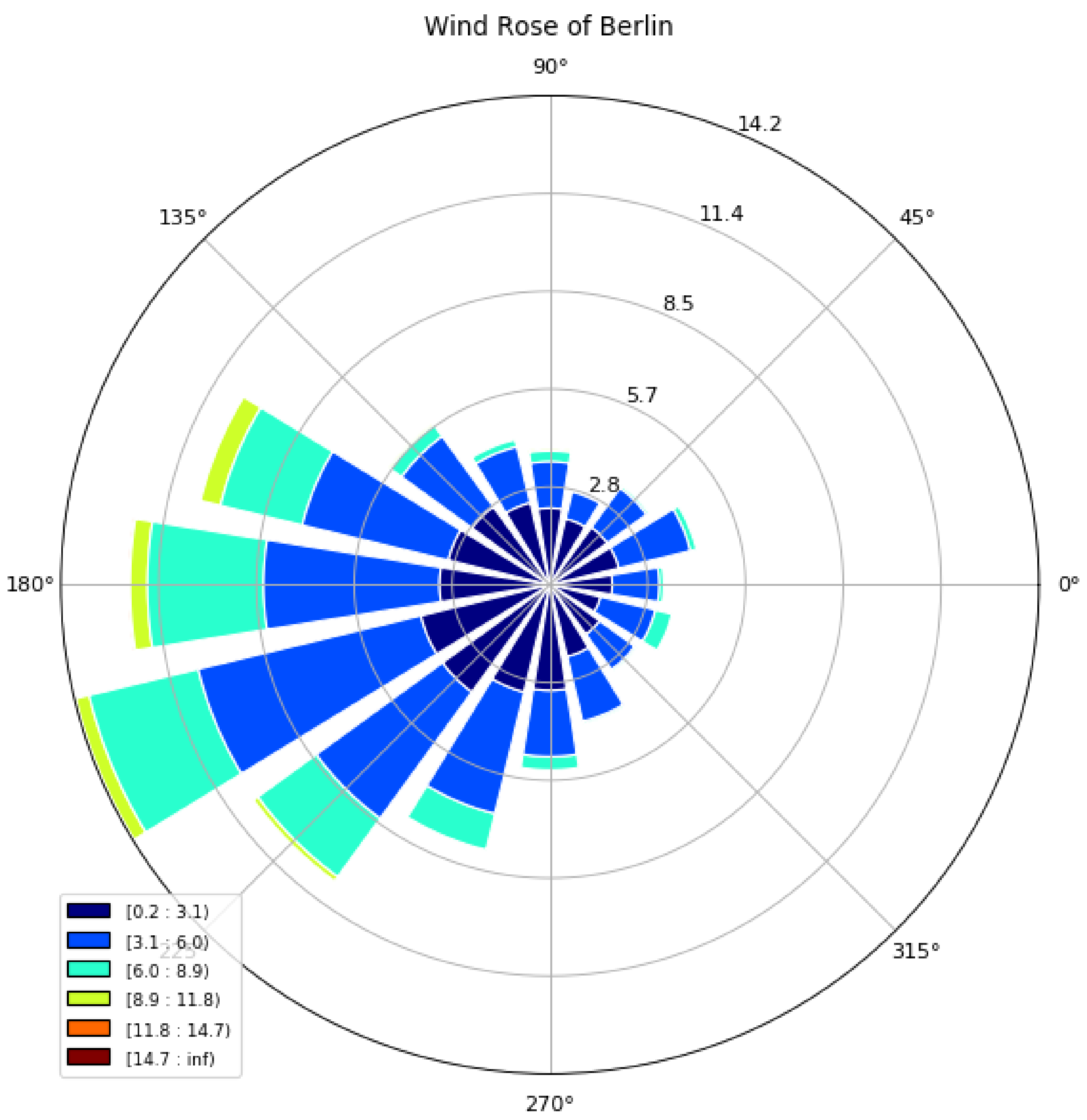

Wind Rose of Gadamis
90°
135°
45°
180°
0°
315°
270°
13.2
10.5
7.9
5.3
2.6
[-0.4 : 2.4)
[2.4 : 5.2)
[5.2 : 7.9)
[7.9 : 10.7)
[10.7 : 13.5)
[13.5 : inf)


Wind Rose of Sabha
90°
45°
0°
315°
270°
225°
180°
135°
15.2
12.2
9.1
6.1
3.0
[0.2 : 2.6)
[2.6 : 5.0)
[5.0 : 7.4)
[7.4 : 9.8)
[9.8 : 12.2)
[12.2 : inf)



Wind Rose of Kufra
90°
24.0
45°
19.2
14.4
135°
9.6
4.8
180°
0°
315°
270°
[0.1 : 1.9)
[1.9 : 3.8)
[3.8 : 5.6)
[5.6 : 7.4)
[7.4 : 9.3)
[9.3 : inf)


Wind Rose of Jaghbub
90°
45°
0°
315°
270°
180°
135°
14.2
11.3
8.5
5.7
2.8
[-0.3 : 2.2)
[2.2 : 4.7)
[4.7 : 7.2)
[7.2 : 9.6)
[9.6 : 12.1)
[12.1 : inf)




```
In [8]: plt.rcParams['figure.figsize'] = [10, 5.0]
        plt.rcParams['figure.dpi'] = 300
```

0.2 Correlation of wind speed and solar irradiance

0.2.1 Correlation without considering night hours, becasue there is no solar irradiance at night

```
In [9]: from sklearn.metrics import r2_score
        # plt.rcParams['figure.figsize']=[5, 4.8]  ### default figure size

        cities=['Tripoli','Zuwara', 'Tarhunah', 'Msallata', 'Ghanima', 'Gharyan', 'Misurata','Sirte', 'Magrun', 'Benghazi',
                'Derna', 'Tobruk','Houn','Gadamis','Sabha','Kufra', 'Jaghbub','Berlin']

        df_dt=['df_Trip','df_Zuwara', 'df_Tarhunah', 'df_Msallata', 'df_Ghanima', 'df_Gharyan', 'df_Misurata',
               'df_Sirte','df_Magrun','df_Benghazi', 'df_Derna', 'df_Tobruk', 'df_Houn','df_Gadamis','df_Sabha',
               'df_Kufra','df_Jaghbub', 'df_Berlin']

        Corr_solar_wind=pd.DataFrame({'tem':[0]},index=['Corr_GHI_WS','Corr_DNI_WS'])

        ##### Since correlation is same for non-nomalized or nomralized no need for nomalized data
        # Corr_solar_wind=pd.DataFrame({'tem':[0]},index=['Corr_GHI_WS','Corr_DNI_WS','Corrnorm_GHI_WS','Corrnorm_DNI_WS'])

        df_windsp=pd.DataFrame({'Tripoli':df_Trip['WS10m']})
        df_ghi=pd.DataFrame({'Tripoli':df_Trip['G(h)']})
        df_dni=pd.DataFrame({'Tripoli':df_Trip['Gb(n)']})

        # for i in np.arange(len(cities)):
        for i in np.arange(len(cities)-1):
            vars()[df_dt[i]]=vars()[df_dt[i]][vars()[df_dt[i]]['G(h)']>0.01]
            df_windsp[cities[i]]=vars()[df_dt[i]]['WS10m']
            df_ghi[cities[i]]=vars()[df_dt[i]]['G(h)']
            df_dni[cities[i]]=vars()[df_dt[i]]['Gb(n)']
        #     day_hrs=df_Sirte[df_Sirte['G(h)']>0.01].count()[0]

            windspeed_comparison=df_windsp
            ghi_comparison=df_ghi
            dni_comparison=df_dni

            df_tem=vars()[df_dt[i]][['WS10m','G(h)','Gb(n)']]
            df_tem_norm=0+(df_tem-df_tem.min())/(df_tem.max()-df_tem.min())*(1-0)
```

```
    Corr_solar_wind.at['Corr_GHI_WS',cities[i]]=df_tem['G(h)'].corr(df_tem['WS10m'])
    Corr_solar_wind.at['Corr_DNI_WS',cities[i]]=df_tem['Gb(n)'].corr(df_tem['WS10m'])

#     Corr_solar_wind.at['Corrnorm_GHI_WS',cities[i]]=df_tem_norm['G(h)'].corr(df_tem_norm['WS10m'])
#     Corr_solar_wind.at['Corrnorm_DNI_WS',cities[i]]=df_tem_norm['Gb(n)'].corr(df_tem_norm['WS10m'])

Corr_solar_wind.drop(['tem'], axis=1, inplace=True)

Corr_solar_wind
```

Out[9]:

```
              Tripoli    Zuwara  Tarhunah  Msallata   Ghanima   Gharyan  \
Corr_GHI_WS -0.018879  0.066374  0.016751 -0.040884 -0.022189  0.030488
Corr_DNI_WS -0.067343 -0.023226 -0.048938 -0.146477 -0.152906 -0.031215

             Misurata     Sirte    Magrun  Benghazi     Derna    Tobruk  \
Corr_GHI_WS -0.056642 -0.017901  0.066411  0.070508  0.038926  0.132381
Corr_DNI_WS -0.117396 -0.056006 -0.007122 -0.024761 -0.062727  0.057894

                 Houn   Gadamis     Sabha     Kufra   Jaghbub
Corr_GHI_WS  0.041430  0.063430  0.076660 -0.001165  0.062920
Corr_DNI_WS -0.024885  0.010074 -0.044828  0.025716 -0.011235
```

In [10]:

```
Corr_solar_wind.loc['Corr_GHI_WS'].describe()
# Corr_solar_wind.loc['Corr_DNI_WS'].describe()
```

Out[10]:

```
count    17.000000
mean      0.029919
std       0.050278
min      -0.056642
25%      -0.017901
50%       0.038926
75%       0.066374
max       0.132381
Name: Corr_GHI_WS, dtype: float64
```

In [11]:

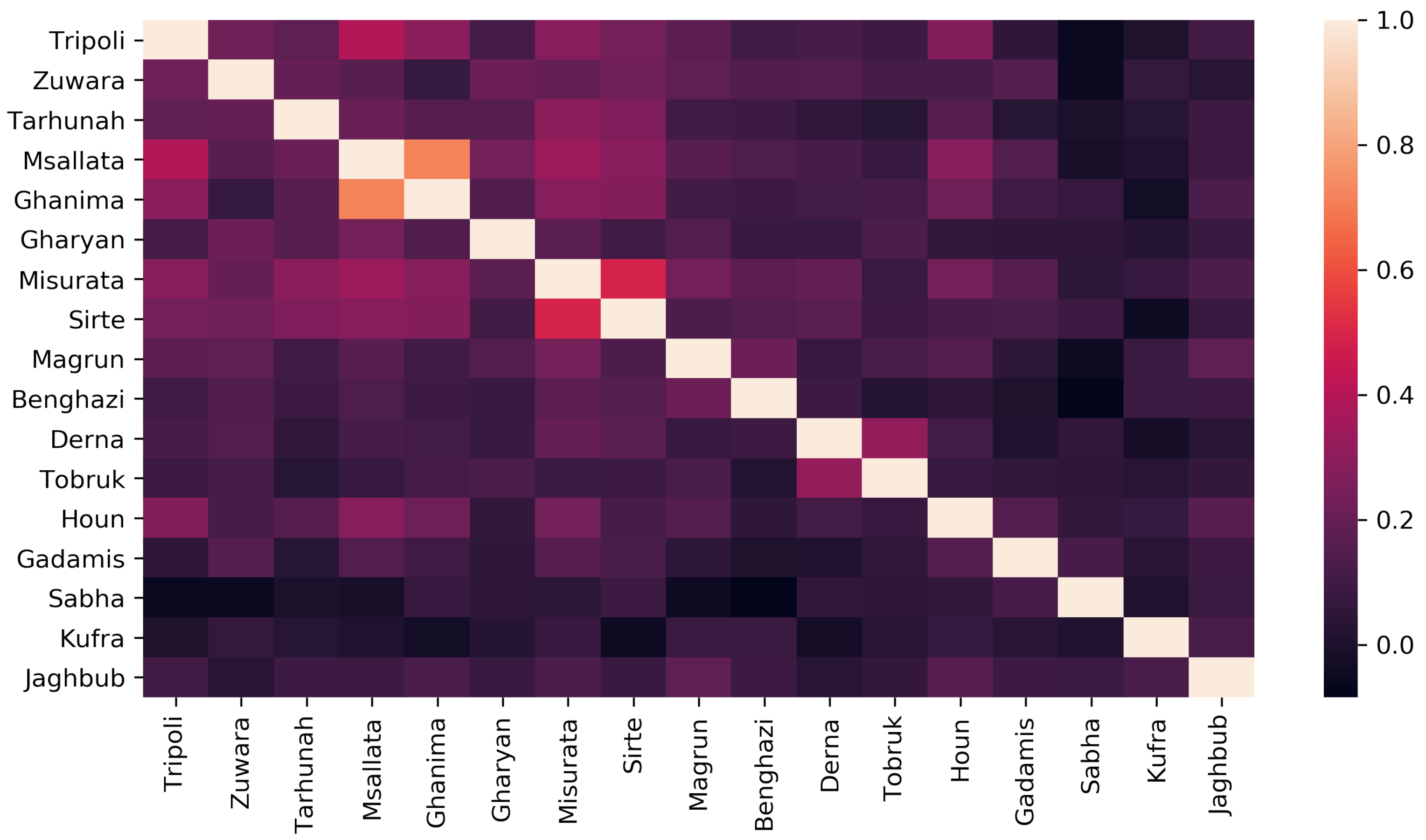
```
# windspeed_comparison
# ghi_comparison
# dni_comparison

windspeed_comparison.shape
# ghi_comparison.shape
# dni_comparison.shape
```

Out[11]: (8760, 17)

In [12]:

```
import seaborn as sns
sns.heatmap(windspeed_comparison.corr())
```

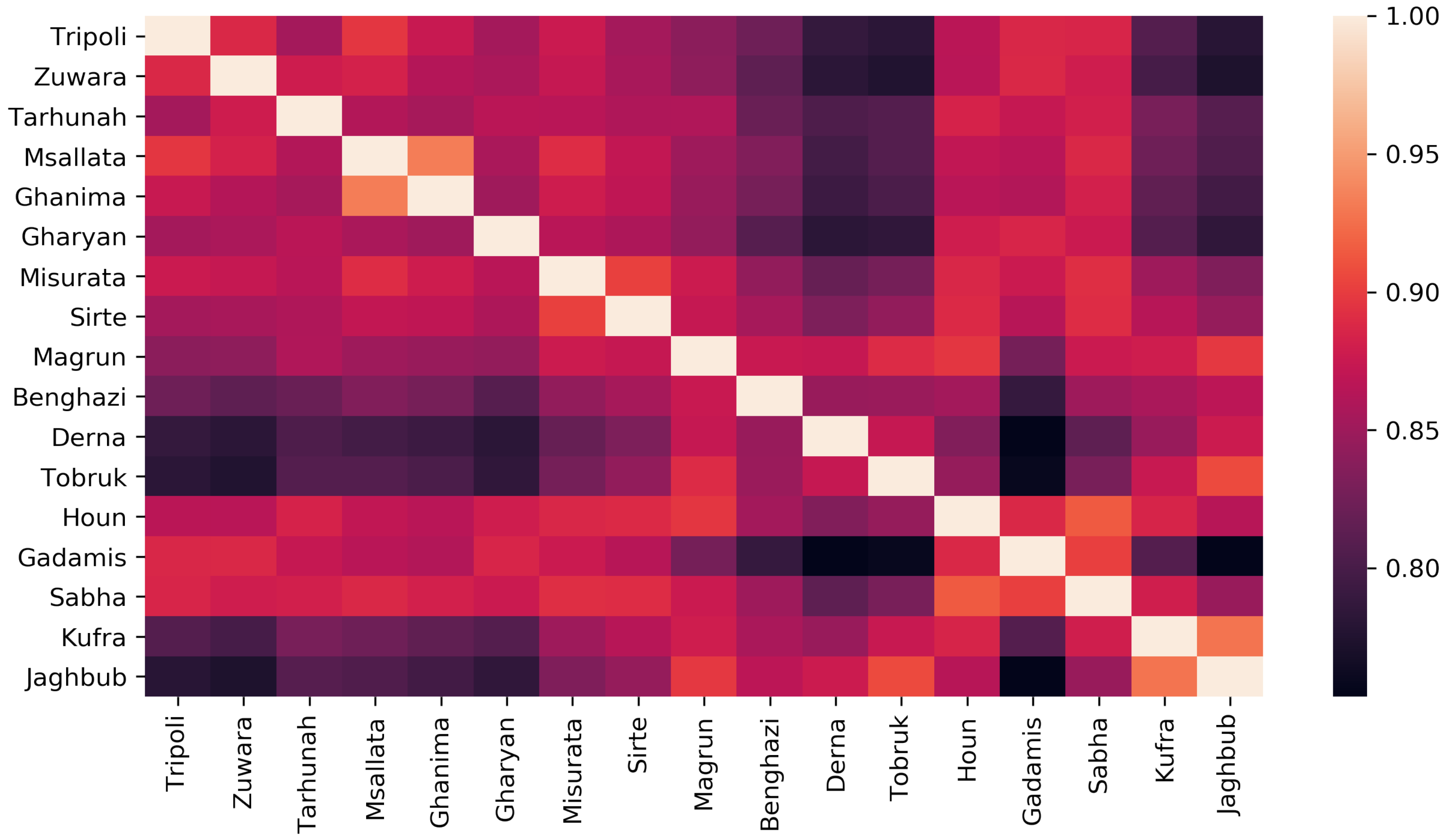
Out[12]: <matplotlib.axes._subplots.AxesSubplot at 0x20f92ae6710>



In [13]:

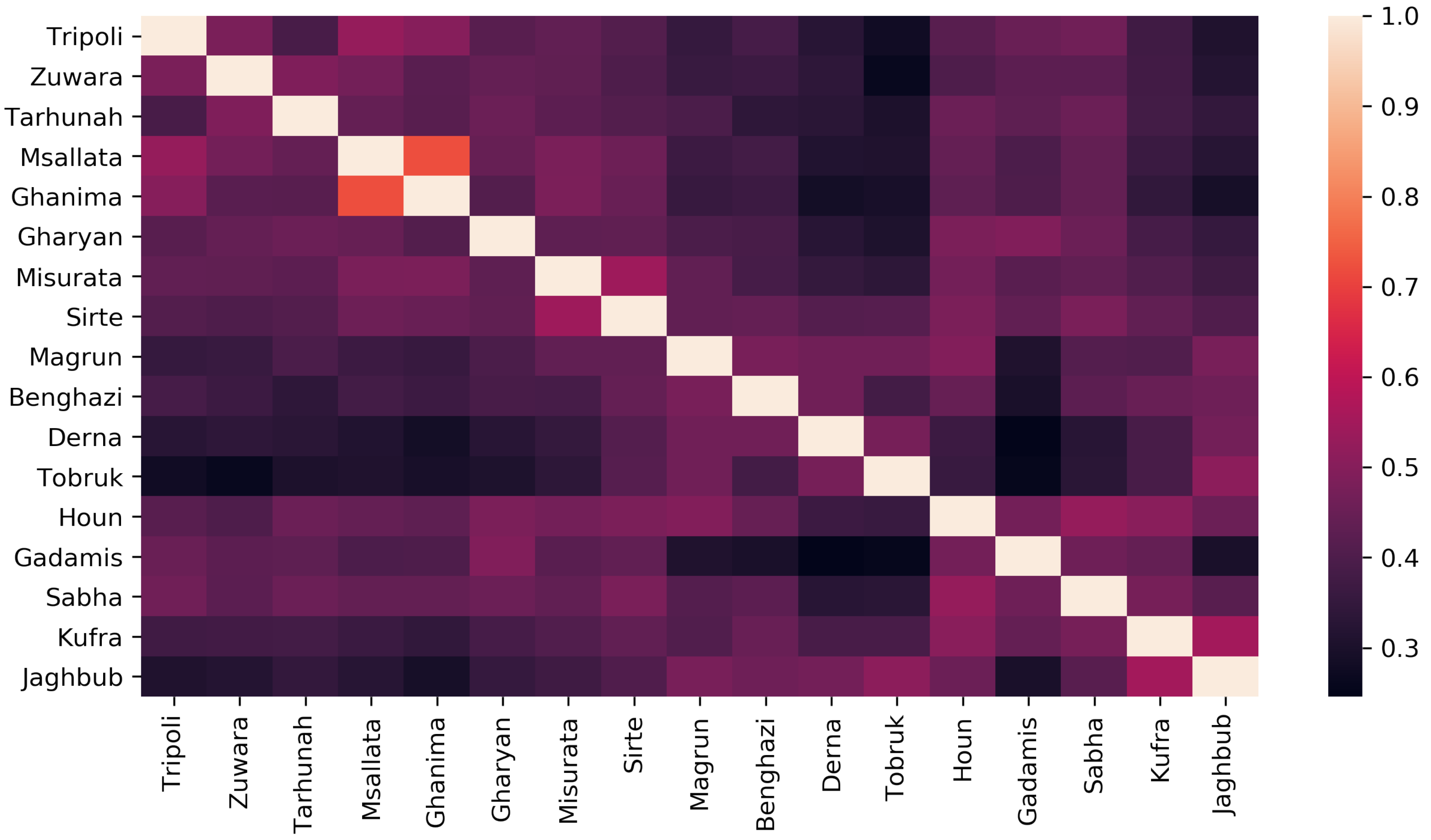
```
sns.heatmap(ghi_comparison.corr())
```

Out[13]: <matplotlib.axes._subplots.AxesSubplot at 0x20f92bbbd30>




```
In [14]: sns.heatmap(dni_comparison.corr())
```

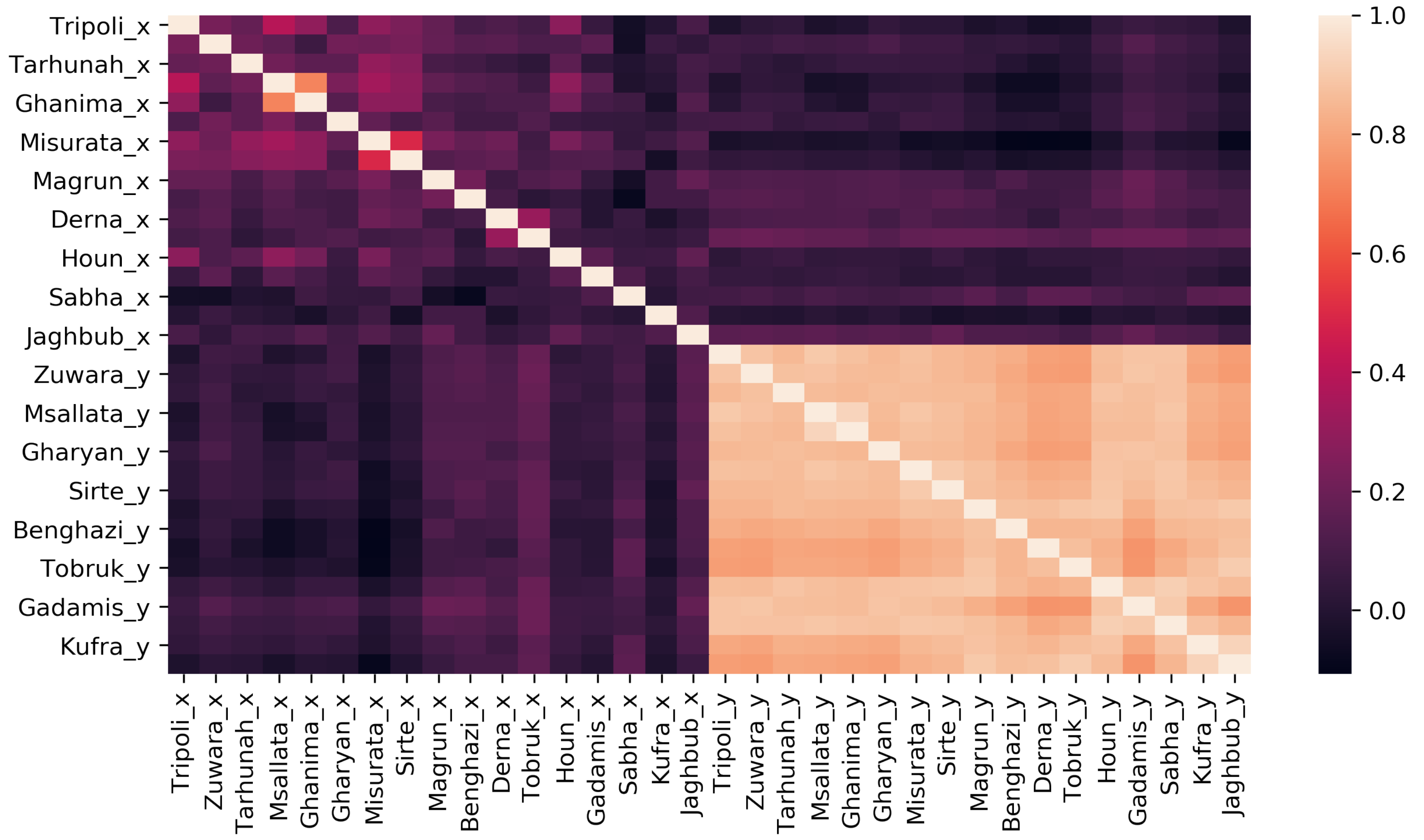
```
Out[14]: <matplotlib.axes._subplots.AxesSubplot at 0x20f92c91668>
```




```
In [15]: df_ws_ghi = pd.merge(windspeed_comparison, ghi_comparison, left_index =True, right_index = True)
         df_ws_dni = pd.merge(windspeed_comparison, dni_comparison, left_index =True, right_index = True)
         # df_ws_ghi
         # df_ws_ghi.describe()
```

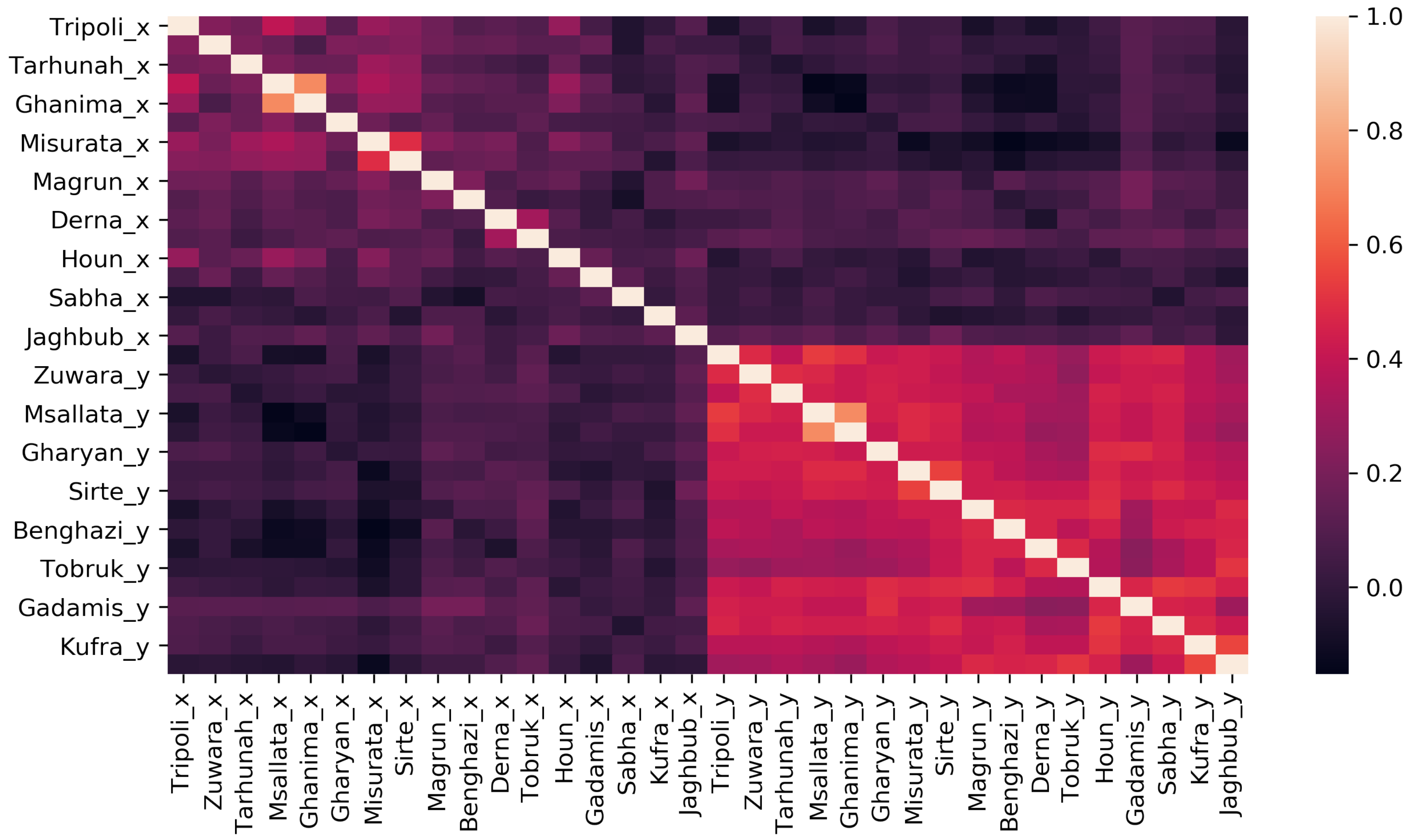
```
In [16]: sns.heatmap(df_ws_ghi.corr())
```

```
Out[16]: <matplotlib.axes._subplots.AxesSubplot at 0x20f8f0a1080>
```




```
In [17]: sns.heatmap(df_ws_dni.corr())
```

```
Out[17]: <matplotlib.axes._subplots.AxesSubplot at 0x20f8ecc4c50>
```



0.3 Modeling Wind Speed to Wind Output Power for a Particular Wind Turbine

Using windpowerlib https://windpowerlib.readthedocs.io/en/stable/
https://github.com/mhdella/windpowerlib

```
In [18]: from windpowerlib import ModelChain, WindTurbine, create_power_curve
         from windpowerlib import data as wt
```

0.3.1 Import weather data

In order to use the windpowerlib you need to at least provide wind speed data for the time frame you want to analyze. The function below imports example weather data from the weather.csv file provided along with the windpowerlib. The data includes wind speed at two different heights in m/s, air temperature in two different heights in K, surface roughness length in m and air pressure in Pa.

To find out which weather data in which units need to be provided to use the ModelChain or other functions of the windpowerlib see the individual function documentation.

In [19]:
```
cities=['Tripoli','Zuwara', 'Tarhunah', 'Msallata', 'Ghanima', 'Gharyan', 'Misurata','Sirte', 'Magrun', 'Benghazi',
        'Derna', 'Tobruk','Houn','Gadamis','Sabha','Kufra', 'Jaghbub','Berlin']

df_dt=['df_Trip','df_Zuwara', 'df_Tarhunah', 'df_Msallata', 'df_Ghanima', 'df_Gharyan', 'df_Misurata',
       'df_Sirte','df_Magrun','df_Benghazi', 'df_Derna', 'df_Tobruk', 'df_Houn','df_Gadamis','df_Sabha',
       'df_Kufra','df_Jaghbub', 'df_Berlin']

for i in np.arange(len(cities)):

    data = pd.read_csv(cities[i]+'_get_pvgis_tmy.csv',index_col='time(UTC)')
    vars()[df_dt[i]] = data
    vars()[df_dt[i]].index=pd.to_datetime(vars()[df_dt[i]].index, format='%Y-%m-%d')

    vars()[df_dt[i]]['Year']=vars()[df_dt[i]].index.year
    vars()[df_dt[i]]['Month']=vars()[df_dt[i]].index.month
    vars()[df_dt[i]]['Day']=vars()[df_dt[i]].index.day
    vars()[df_dt[i]]['Hour']=vars()[df_dt[i]].index.hour
    vars()[df_dt[i]]['Year'] = 2021
    vars()[df_dt[i]].index=pd.to_datetime(df_Trip[['Year', 'Month', 'Day', 'Hour']])

# df_Berlin
```

In [20]:
```
weather=[]
cities=['Tripoli','Zuwara', 'Tarhunah', 'Msallata', 'Ghanima', 'Gharyan', 'Misurata','Sirte', 'Magrun', 'Benghazi',
        'Derna', 'Tobruk','Houn','Gadamis','Sabha','Kufra', 'Jaghbub','Berlin']

df_dt=['df_Trip','df_Zuwara', 'df_Tarhunah', 'df_Msallata', 'df_Ghanima', 'df_Gharyan', 'df_Misurata',
       'df_Sirte','df_Magrun','df_Benghazi', 'df_Derna', 'df_Tobruk', 'df_Houn','df_Gadamis','df_Sabha',
       'df_Kufra','df_Jaghbub', 'df_Berlin']


for i in np.arange(len(cities)):

    weather =vars()[df_dt[i]]
    # weather.index.name = None
    temp=weather['T2m']
    weather=weather.rename(columns={'WS10m':'wind_speed',  'T2m':'temperature', 'SP':'pressure'})
    weather['roughness_length'] = 0.15
    weather['temperature'] =273.15+temp
    weather.drop(['RH','G(h)','Gb(n)','Gd(h)','IR(h)','WD10m','Year', 'Month', 'Day', 'Hour'], axis=1, inplace=True)

    weather.columns=[['temperature', 'wind_speed', 'pressure', 'roughness_length'],[2,10,0,0]]
    vars()[df_dt[i]]=weather

# vars()[df_dt[i]]
# weather
# df_Mis
```

0.3.2 Initialize wind turbine

There are three ways to initialize a WindTurbine object in the windpowerlib. You can either use turbine data from the OpenEnergy Database (oedb) turbine library that is provided along with the windpowerlib, as done for the 'enercon_e126', or specify your own turbine by directly providing a power (coefficient) curve, as done below for 'my_turbine', or provide your own turbine data in csv files, as done for 'my_turbine2'.

You can execute the following to get a table of all wind turbines for which power and/or power coefficient curves are provided.

We will use a wind trubine from GE, which is GE120/2500 https://www.thewindpower.net/turbine_en_592_ge-energy_2.5-120.php

In [21]:
```
# get power curves
# get names of wind turbines for which power curves and/or are provided
# set print_out=True to see the list of all available wind turbines
df = wt.get_turbine_types(print_out=False)

# find all Enercons
# print(df[df["manufacturer"].str.contains("Enercon")])
print(df[df["manufacturer"].str.contains("GE")])
```

```
   manufacturer turbine_type  has_power_curve  has_cp_curve
21      GE Wind   GE100/2500             True         False
22      GE Wind   GE103/2750             True          True
23      GE Wind   GE120/2500             True          True
24      GE Wind   GE120/2750             True          True
25      GE Wind   GE130/3200             True          True
```

In [22]:
```
# specification of wind turbine where power curve is provided in the
# oedb turbine library

ge_120 = {
        'turbine_type': 'GE120/2500',  # turbine type as in oedb turbine library
        'hub_height': 100  # in m
    }
# initialize WindTurbine object
ge120 = WindTurbine(**ge_120)
```

0.3.3 Use the ModelChain to calculate turbine power output

The ModelChain is a class that provides all necessary steps to calculate the power output of a wind turbine. When calling the 'run_model' method, first the wind speed and density (if necessary) at hub height are calculated and then used to calculate the power output.

In [23]:
```python
#### power output calculation for ge120
cities=['Tripoli','Zuwara', 'Tarhunah', 'Msallata', 'Ghanima', 'Gharyan', 'Misurata','Sirte', 'Magrun', 'Benghazi',
        'Derna', 'Tobruk','Houn','Gadamis','Sabha','Kufra', 'Jaghbub','Berlin']

df_dt=['df_Trip','df_Zuwara', 'df_Tarhunah', 'df_Msallata', 'df_Ghanima', 'df_Gharyan', 'df_Misurata',
       'df_Sirte','df_Magrun','df_Benghazi', 'df_Derna', 'df_Tobruk', 'df_Houn','df_Gadamis','df_Sabha',
       'df_Kufra','df_Jaghbub', 'df_Berlin']



for i in np.arange(len(cities)):

    weather =vars()[df_dt[i]]
    #### own specifications for ModelChain setup
    modelchain_data = {
        'wind_speed_model': 'logarithmic',      # 'logarithmic' (default),
                                                # 'hellman' or
                                                # 'interpolation_extrapolation'
        'density_model': 'ideal_gas',           # 'barometric' (default), 'ideal_gas'
                                                #  or 'interpolation_extrapolation'
        'temperature_model': 'linear_gradient', # 'linear_gradient' (def.) or
                                                # 'interpolation_extrapolation'
        'power_output_model':
            'power_coefficient_curve',          # 'power_curve' (default) or
                                                # 'power_coefficient_curve'
        'density_correction': True,             # False (default) or True
        'obstacle_height': 0,                   # default: 0
        'hellman_exp': None}                    # None (default) or None

    # initialize ModelChain with own specifications and use run_model method to
    # calculate power output
    mc_ge120 = ModelChain(ge120, **modelchain_data).run_model(
        weather)
    # write power output time series to WindTurbine object
    ge120.power_output = mc_ge120.power_output

    vars()[df_dt[i]]['PW_Wind']=ge120.power_output
    vars()[df_dt[i]+'_PW_Curve']=ge120.power_curve
    vars()[df_dt[i]+'_PW_Cp']=ge120.power_coefficient_curve

# vars()[df_dt[i]]
# vars()[df_dt[i]]['PW_Wind']
```

In [24]:
```python
cities=['Tripoli','Zuwara', 'Tarhunah', 'Msallata', 'Ghanima', 'Gharyan', 'Misurata','Sirte', 'Magrun', 'Benghazi',
        'Derna', 'Tobruk','Houn','Gadamis','Sabha','Kufra', 'Jaghbub','Berlin']

df_dt=['df_Trip','df_Zuwara', 'df_Tarhunah', 'df_Msallata', 'df_Ghanima', 'df_Gharyan', 'df_Misurata',
       'df_Sirte','df_Magrun','df_Benghazi', 'df_Derna', 'df_Tobruk', 'df_Houn','df_Gadamis','df_Sabha',
       'df_Kufra','df_Jaghbub', 'df_Berlin']



df_tem=pd.DataFrame({'Tripoli':df_Trip['PW_Wind',].resample('M').sum()})
df_tem
for i in np.arange(len(cities)):
    df_tem[cities[i]]=vars()[df_dt[i]]['PW_Wind',].resample('M').sum()

PW_Wind_comparison=df_tem

# # PW_Wind_comparison.describe()
PW_Wind_comparison
```

```
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
```

Out[24]:

	Tripoli	Zuwara	Tarhunah	Msallata
2021-01-31	1.233823e+09	6.563418e+08	1.072380e+09	1.055705e+09
2021-02-28	1.232927e+09	1.065110e+09	1.159623e+09	9.861940e+08
2021-03-31	6.840335e+08	7.589762e+08	8.782608e+08	1.057770e+09
2021-04-30	7.975678e+08	7.148443e+08	9.418069e+08	1.039795e+09
2021-05-31	5.213789e+08	4.840239e+08	9.634585e+08	8.989776e+08
2021-06-30	8.850381e+08	1.096890e+09	8.484234e+08	7.085471e+08
2021-07-31	9.846533e+08	5.610340e+08	7.718120e+08	5.483634e+08
2021-08-31	5.452158e+08	7.112266e+08	5.983949e+08	4.434599e+08
2021-09-30	7.999770e+08	4.926898e+08	7.690518e+08	4.187425e+08
2021-10-31	6.382198e+08	5.594551e+08	5.472171e+08	7.673814e+08
2021-11-30	8.181772e+08	4.881820e+08	8.167281e+08	7.731433e+08
2021-12-31	7.116705e+08	5.167463e+08	5.576381e+08	1.014080e+09

	Ghanima	Gharyan	Misurata	Sirte
2021-01-31	9.472439e+08	6.837305e+08	1.280761e+09	1.136600e+09
2021-02-28	8.682733e+08	5.899248e+08	1.281676e+09	9.244250e+08
2021-03-31	1.079482e+09	7.260579e+08	7.624387e+08	9.548665e+08
2021-04-30	1.181007e+09	9.282684e+08	1.239411e+09	1.096995e+09
2021-05-31	9.236674e+08	7.560305e+08	1.093043e+09	9.858632e+08
2021-06-30	6.139530e+08	5.862362e+08	1.069205e+09	1.199379e+09
2021-07-31	4.868493e+08	4.912442e+08	4.664368e+08	7.120975e+08

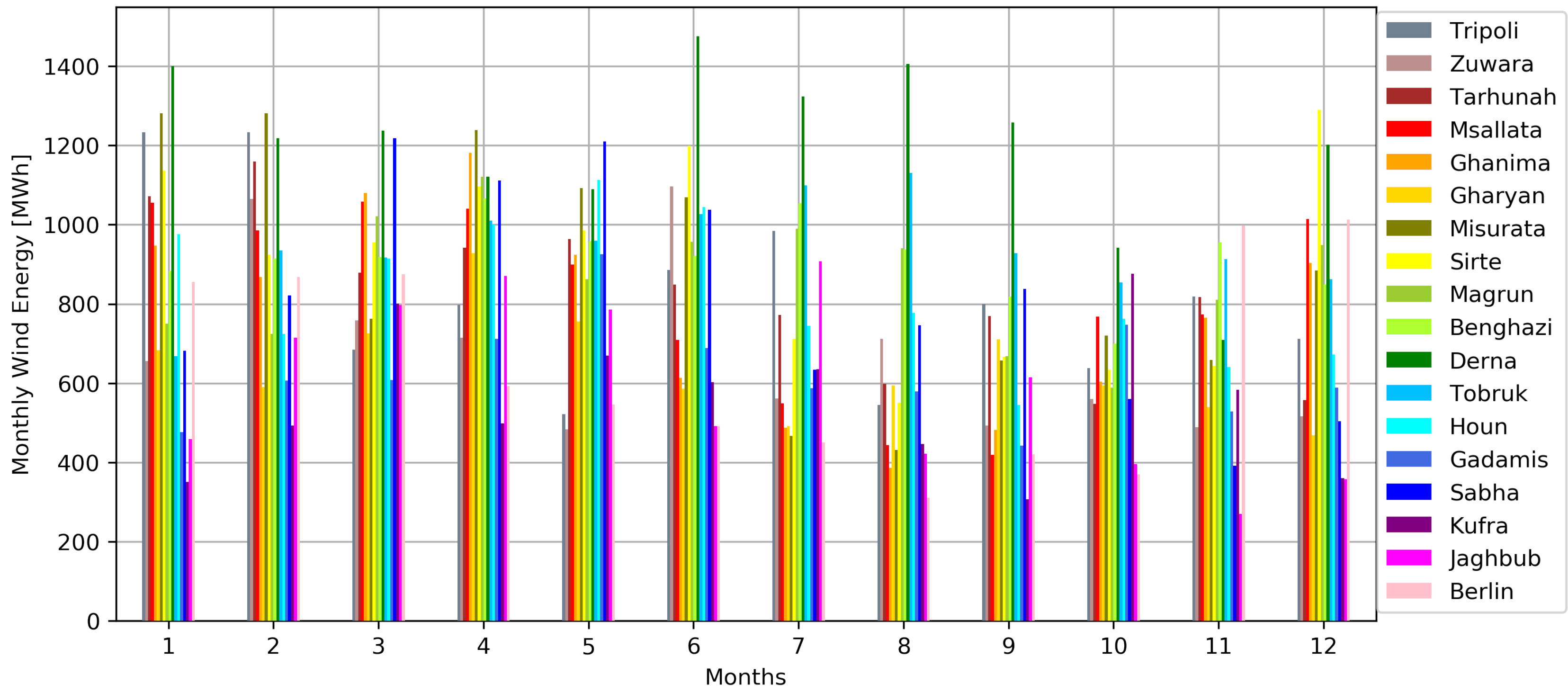
```
2021-08-31  3.857082e+08  5.940871e+08  4.317863e+08  5.497205e+08
2021-09-30  4.821710e+08  7.106109e+08  6.567398e+08  6.667348e+08
2021-10-31  6.044576e+08  5.936234e+08  7.203912e+08  6.335377e+08
2021-11-30  7.656827e+08  5.398451e+08  6.590450e+08  6.430542e+08
2021-12-31  9.041625e+08  4.687127e+08  8.847158e+08  1.288761e+09

                  Magrun      Benghazi         Derna        Tobruk  \
2021-01-31  7.506282e+08  8.829393e+08  1.400028e+09  6.685409e+08
2021-02-28  7.243157e+08  9.150553e+08  1.218404e+09  9.343409e+08
2021-03-31  1.021578e+09  9.190877e+08  1.237225e+09  9.175855e+08
2021-04-30  1.121674e+09  1.066776e+09  1.121666e+09  1.010394e+09
2021-05-31  8.618430e+08  9.577347e+08  1.089353e+09  9.602590e+08
2021-06-30  9.565255e+08  9.207377e+08  1.476126e+09  1.027179e+09
2021-07-31  9.890508e+08  1.054671e+09  1.323571e+09  1.099174e+09
2021-08-31  9.398077e+08  9.380734e+08  1.406133e+09  1.130710e+09
2021-09-30  6.676445e+08  8.184178e+08  1.258601e+09  9.286683e+08
2021-10-31  5.882051e+08  6.997753e+08  9.422476e+08  8.538214e+08
2021-11-30  8.097814e+08  9.558372e+08  7.098323e+08  9.129622e+08
2021-12-31  9.490383e+08  8.493525e+08  1.202426e+09  8.620006e+08

                    Houn       Gadamis         Sabha         Kufra  \
2021-01-31  9.756065e+08  4.770973e+08  6.811497e+08  3.507817e+08
2021-02-28  7.243111e+08  6.068439e+08  8.219009e+08  4.935132e+08
2021-03-31  9.148067e+08  6.080890e+08  1.218285e+09  8.004039e+08
2021-04-30  1.000001e+09  7.116468e+08  1.112154e+09  4.981017e+08
2021-05-31  1.112512e+09  9.250209e+08  1.210722e+09  6.694428e+08
2021-06-30  1.044626e+09  6.879866e+08  1.037125e+09  6.023965e+08
2021-07-31  7.451443e+08  5.867695e+08  6.337102e+08  6.357442e+08
2021-08-31  7.778566e+08  5.789876e+08  7.455438e+08  4.467794e+08
2021-09-30  5.442506e+08  4.422653e+08  8.378133e+08  3.074311e+08
2021-10-31  7.623670e+08  7.472764e+08  5.601609e+08  8.755263e+08
2021-11-30  6.404036e+08  5.282371e+08  3.921635e+08  5.826436e+08
2021-12-31  6.728102e+08  5.888288e+08  5.036680e+08  3.596595e+08

                 Jaghbub        Berlin
2021-01-31  4.580622e+08  8.553083e+08
2021-02-28  7.151638e+08  8.684264e+08
2021-03-31  7.969964e+08  8.750233e+08
2021-04-30  8.710807e+08  5.932219e+08
2021-05-31  7.864301e+08  5.464065e+08
2021-06-30  4.912357e+08  4.918726e+08
2021-07-31  9.071451e+08  4.499309e+08
2021-08-31  4.215692e+08  3.111526e+08
2021-09-30  6.149091e+08  4.203312e+08
2021-10-31  3.956638e+08  3.693392e+08
2021-11-30  2.694907e+08  9.985872e+08
2021-12-31  3.570939e+08  1.012789e+09
```

In [25]:
```
PW_Wind_comparison.index=PW_Wind_comparison.index.month
PW_Wind_MW_comparison=PW_Wind_comparison/1000000
```

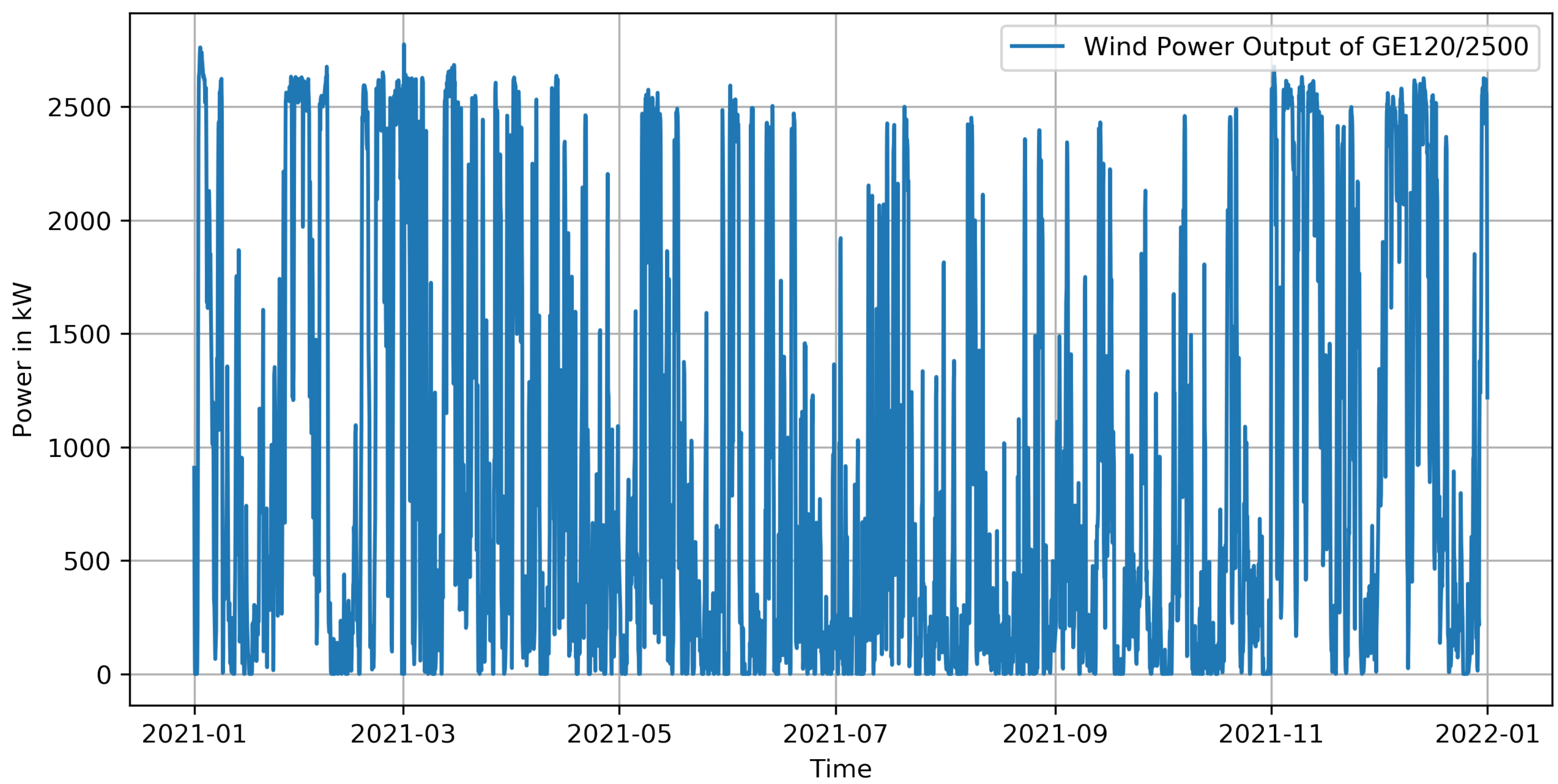
In [26]:
```
plt.rcParams['figure.figsize'] = [10, 5.0]
plt.rcParams['figure.dpi'] = 300

c = ['slategrey','rosybrown','brown','red','orange','gold','olive','yellow','yellowgreen',
     'greenyellow','green','deepskyblue','cyan','royalblue','blue','purple', 'magenta', 'pink']

PW_Wind_MW_comparison.plot.bar(zorder=3,color=c)

plt.xticks(rotation=0)
plt.xlabel('Months');
plt.ylabel('Monthly Wind Energy [MWh]');
plt.legend(bbox_to_anchor=(1.16, 1.01),loc='upper right')
plt.grid()
print(PW_Wind_MW_comparison.describe())
```

```
           Tripoli       Zuwara     Tarhunah     Msallata      Ghanima  \
count    12.000000    12.000000    12.000000    12.000000    12.000000
mean    821.056825   675.459966   827.066221   809.346621   770.221524
std     233.926976   212.446201   194.314965   238.275413   254.033909
min     521.378879   484.023854   547.217134   418.742487   385.708200
25%     672.580098   510.732170   726.387538   668.501164   575.055547
50%     798.772396   608.687876   832.575722   836.060412   816.977995
75%     909.941860   725.877305   947.219776  1020.508948   929.561518
max    1233.823277  1096.890154  1159.622684  1057.769908  1181.007409

           Gharyan     Misurata        Sirte       Magrun     Benghazi  \
count    12.000000    12.000000    12.000000    12.000000    12.000000
mean    639.030969   878.804142   899.336128   865.007682   914.871523
std     128.632750   307.809254   251.533855   159.065176    98.906348
min     468.712677   431.786324   549.720532   588.205146   699.775317
25%     574.638401   658.468730   660.814631   744.050056   874.542593
50%     593.855246   823.577259   939.645751   900.825343   919.912686
75%     714.472663  1129.635029  1106.896059   964.656818   956.311536
max     928.268427  1281.675922  1288.760593  1121.674038  1066.776437

             Derna       Tobruk         Houn      Gadamis        Sabha  \
count    12.000000    12.000000    12.000000    12.000000    12.000000
```

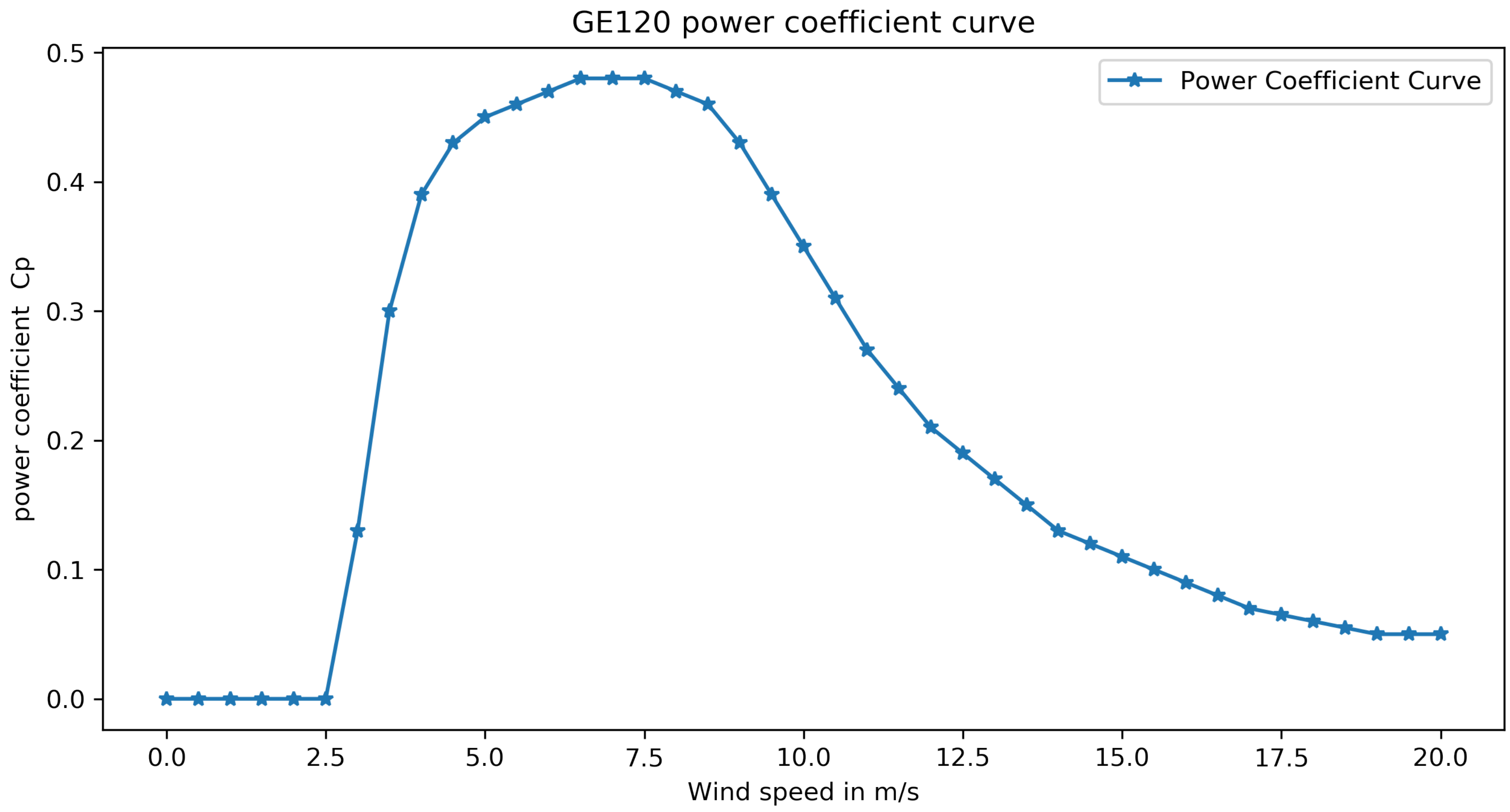
```
mean    1198.800997    942.136276    826.224600   624.087428    812.866307
std      214.103690    121.809220    178.624010   130.231251    278.576773
min      709.832314    668.540870    544.250580   442.265342    392.163546
25%     1113.587365    900.221765    711.435863   566.299973    615.322884
50%     1227.814324    931.504600    770.111790   597.836342    783.722317
75%     1342.685355   1014.589927    981.705039   693.901624   1055.881889
max     1476.126486   1130.709742   1112.512367   925.020885   1218.285181

             Kufra      Jaghbub       Berlin
count    12.000000    12.000000    12.000000
mean    551.868665   590.403390   649.365745
std     177.188393   219.000429   255.653201
min     307.431126   269.490707   311.152557
25%     424.999445   415.092845   442.530979
50%     540.372668   553.072423   569.814212
75%     644.168827   789.071642   870.075631
max     875.526262   907.145077  1012.788923
```

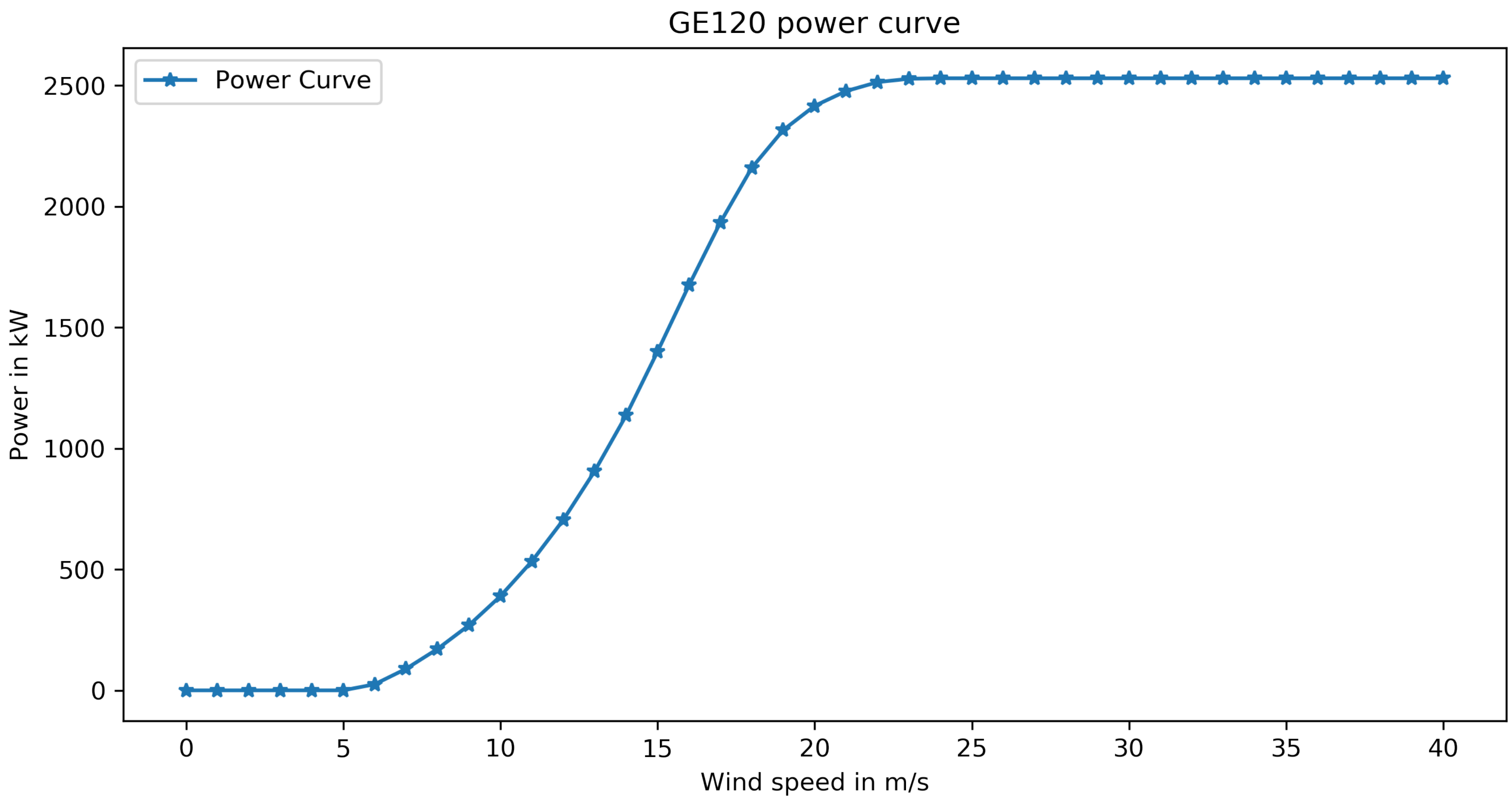


In [27]:
```python
plt.rcParams['figure.figsize'] = [10, 5.0]
plt.rcParams['figure.dpi'] = 300

#### plot turbine power output

plt.plot(ge120.power_output/1000)

# plt.legend(['Enercon E126'], bbox_to_anchor=(1.2, 1.02),loc='upper right')
plt.legend(['Wind Power Output of GE120/2500'],loc='upper right')

plt.xlabel('Time')
plt.ylabel('Power in kW')
plt.grid()
plt.show()
```



In [28]:
```
#### plot power (coefficient) curves
ge120pcrv_kw=ge120.power_curve['value']/1000

if plt:
    if ge120.power_coefficient_curve is not None:
        ge120.power_coefficient_curve.plot(
        x='wind_speed', y='value', style='-*',
        title='GE120 power coefficient curve')
        plt.xlabel('Wind speed in m/s')
        plt.ylabel('power coefficient  Cp')
        plt.legend(['Power Coefficient Curve'],loc='upper right')
        plt.show()

    if ge120.power_curve is not None:
        ge120pcrv_kw.plot(x='wind_speed', y='value', style='-*',
        title='GE120 power curve')
        plt.xlabel('Wind speed in m/s')
        plt.ylabel('Power in kW')
        plt.legend(['Power Curve'],loc='upper left')
        plt.show()
```





0.4 Calculating Net Capacity Factor

For demonstration purposes, we'll assume a 1kW array with a temperature coefficient of -0.4%/°C:

0.4.1 Capacity Factor for all locations

In [29]: PW_Wind_comparison

Out[29]:

```
         Tripoli        Zuwara      Tarhunah      Msallata       Ghanima  \
1   1.233823e+09  6.563418e+08  1.072380e+09  1.055705e+09  9.472439e+08
2   1.232927e+09  1.065110e+09  1.159623e+09  9.861940e+08  8.682733e+08
3   6.840335e+08  7.589762e+08  8.782608e+08  1.057770e+09  1.079482e+09
4   7.975678e+08  7.148443e+08  9.418069e+08  1.039795e+09  1.181007e+09
5   5.213789e+08  4.840239e+08  9.634585e+08  8.989776e+08  9.236674e+08
6   8.850381e+08  1.096890e+09  8.484234e+08  7.085471e+08  6.139530e+08
7   9.846533e+08  5.610340e+08  7.718120e+08  5.483634e+08  4.868493e+08
8   5.452158e+08  7.112266e+08  5.983949e+08  4.434599e+08  3.857082e+08
9   7.999770e+08  4.926898e+08  7.690518e+08  4.187425e+08  4.821710e+08
10  6.382198e+08  5.594551e+08  5.472171e+08  7.673814e+08  6.044576e+08
11  8.181772e+08  4.881820e+08  8.167281e+08  7.731433e+08  7.656827e+08
12  7.116705e+08  5.167463e+08  5.576381e+08  1.014080e+09  9.041625e+08

         Gharyan      Misurata         Sirte        Magrun      Benghazi  \
1   6.837305e+08  1.280761e+09  1.136600e+09  7.506282e+08  8.829393e+08
2   5.899248e+08  1.281676e+09  9.244250e+08  7.243157e+08  9.150553e+08
3   7.260579e+08  7.624387e+08  9.548665e+08  1.021578e+09  9.190877e+08
4   9.282684e+08  1.239411e+09  1.096995e+09  1.121674e+09  1.066776e+09
5   7.560305e+08  1.093043e+09  9.858632e+08  8.618430e+08  9.577347e+08
6   5.862362e+08  1.069205e+09  1.199379e+09  9.565255e+08  9.207377e+08
7   4.912442e+08  4.664368e+08  7.120975e+08  9.890508e+08  1.054671e+09
8   5.940871e+08  4.317863e+08  5.497205e+08  9.398077e+08  9.380734e+08
9   7.106109e+08  6.567398e+08  6.667348e+08  6.676445e+08  8.184178e+08
10  5.936234e+08  7.203912e+08  6.335377e+08  5.882051e+08  6.997753e+08
11  5.398451e+08  6.590450e+08  6.430542e+08  8.097814e+08  9.558372e+08
12  4.687127e+08  8.847158e+08  1.288761e+09  9.490383e+08  8.493525e+08

           Derna        Tobruk          Houn       Gadamis         Sabha  \
1   1.400028e+09  6.685409e+08  9.756065e+08  4.770973e+08  6.811497e+08
2   1.218404e+09  9.343409e+08  7.243111e+08  6.068439e+08  8.219009e+08
3   1.237225e+09  9.175855e+08  9.148067e+08  6.080890e+08  1.218285e+09
4   1.121666e+09  1.010394e+09  1.000001e+09  7.116468e+08  1.112154e+09
5   1.089353e+09  9.602590e+08  1.112512e+09  9.250209e+08  1.210722e+09
6   1.476126e+09  1.027179e+09  1.044626e+09  6.879866e+08  1.037125e+09
7   1.323571e+09  1.099174e+09  7.451443e+08  5.867695e+08  6.337102e+08
8   1.406133e+09  1.130710e+09  7.778566e+08  5.789876e+08  7.455438e+08
9   1.258601e+09  9.286683e+08  5.442506e+08  4.422653e+08  8.378133e+08
10  9.422476e+08  8.538214e+08  7.623670e+08  7.472764e+08  5.601609e+08
11  7.098323e+08  9.129622e+08  6.404036e+08  5.282371e+08  3.921635e+08
12  1.202426e+09  8.620006e+08  6.728102e+08  5.888288e+08  5.036680e+08

           Kufra       Jaghbub        Berlin
1   3.507817e+08  4.580622e+08  8.553083e+08
2   4.935132e+08  7.151638e+08  8.684264e+08
3   8.004039e+08  7.969964e+08  8.750233e+08
4   4.981017e+08  8.710807e+08  5.932219e+08
5   6.694428e+08  7.864301e+08  5.464065e+08
6   6.023965e+08  4.912357e+08  4.918726e+08
7   6.357442e+08  9.071451e+08  4.499309e+08
8   4.467794e+08  4.215692e+08  3.111526e+08
9   3.074311e+08  6.149091e+08  4.203312e+08
10  8.755263e+08  3.956638e+08  3.693392e+08
11  5.826436e+08  2.694907e+08  9.985872e+08
12  3.596595e+08  3.570939e+08  1.012789e+09
```

In [30]:
```
typical_hrs=8760
```

In [31]:
```
# ghi_comparison_and_Berlin.describe()
rating_pw=2500000    ##2,500 kW
day_hrs=typical_hrs
PW_Wind_comparison.sum()
PW_CFs=PW_Wind_comparison.sum()/(day_hrs*rating_pw)

PW_CFs
```

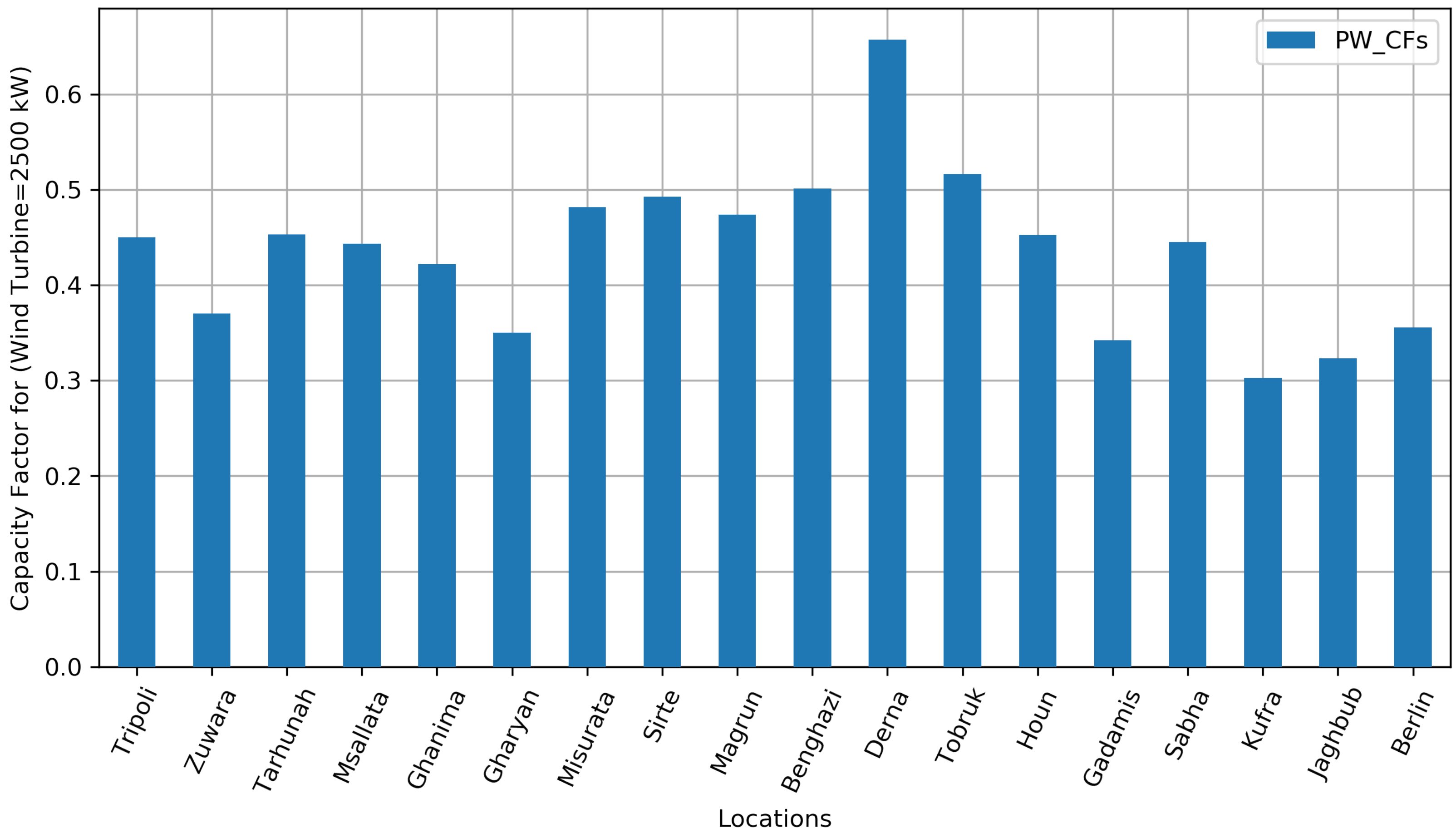
Out[31]:
```
Tripoli     0.449894
Zuwara      0.370115
Tarhunah    0.453187
Msallata    0.443478
Ghanima     0.422039
Gharyan     0.350154
Misurata    0.481537
Sirte       0.492787
Magrun      0.473977
Benghazi    0.501299
Derna       0.656877
Tobruk      0.516239
Houn        0.452726
Gadamis     0.341966
Sabha       0.445406
```

```
Kufra       0.302394
Jaghbub     0.323509
Berlin      0.355817
dtype: float64
```

In [32]:
```
# CFs_dict={'GHI': GHI_CFs, 'DNI':DNI_CFs}
CFs_dict={'PW_CFs': PW_CFs}

df_pw_cfs=pd.DataFrame(CFs_dict)
df_pw_cfs.T
```

Out[32]:
```
         Tripoli    Zuwara  Tarhunah  Msallata   Ghanima   Gharyan  Misurata  \
PW_CFs  0.449894  0.370115  0.453187  0.443478  0.422039  0.350154  0.481537

           Sirte    Magrun  Benghazi     Derna    Tobruk      Houn   Gadamis  \
PW_CFs  0.492787  0.473977  0.501299  0.656877  0.516239  0.452726  0.341966

           Sabha     Kufra   Jaghbub    Berlin
PW_CFs  0.445406  0.302394  0.323509  0.355817
```

In [33]:
```
plt.rcParams['figure.figsize'] = [10, 5.0]
plt.rcParams['figure.dpi'] = 300

df_pw_cfs.plot.bar(zorder=3)
# plt.yticks(np.arange(0, 1, 0.1))
plt.xticks(rotation=0)
# plt.title('Capacity Factors Based on GHI and DNI')
plt.xlabel('Locations');
plt.ylabel('Capacity Factor for (Wind Turbine=2500 kW)');
# plt.legend(bbox_to_anchor=(1.2, 1.02),loc='upper right')
# plt.legend(bbox_to_anchor=(1.08,1.01),loc='upper right')

plt.xticks(rotation=65)
plt.grid()
```



0.5 Wind Power Variability for given locations

In [34]:
```
# ge120.power_output.describe()
# ge120.power_output.max()
```

In [35]:
```
df_pw_variability=[]

cities=['Tripoli','Zuwara', 'Tarhunah', 'Msallata', 'Ghanima', 'Gharyan', 'Misurata','Sirte', 'Magrun', 'Benghazi',
        'Derna', 'Tobruk','Houn','Gadamis','Sabha','Kufra', 'Jaghbub','Berlin']

i=0
dt=[df_Trip,df_Zuwara, df_Tarhunah, df_Msallata, df_Ghanima, df_Gharyan, df_Misurata, df_Sirte, df_Magrun,
    df_Benghazi, df_Derna, df_Tobruk, df_Houn, df_Gadamis, df_Sabha, df_Kufra, df_Jaghbub, df_Berlin]

for dti in dt:
    dti_pw=dti['PW_Wind',]/ge120.power_output.max()
    dti_pw[dti_pw>10]=0
    dti_pw[dti_pw>1]=1
# #    dti_pw_variability=1-dti_pw.describe()[1]#mean
    dti_pw_variability=dti_pw.describe()[2]#std deviation
```

```
        df_pw_variability.append(dti_pw_variability)
        print('Wind Power Variability of', cities[i],'=', round(dti_pw_variability,4))
        i=i+1

    PW_Var_dict= {'Wind Power Variability':df_pw_variability}
    df_pw_var=pd.DataFrame(PW_Var_dict)
    df_pw_var.index=cities
```

```
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)
C:\Users\Mhdella\Anaconda3\lib\site-packages\IPython\core\interactiveshell.py:2899: PerformanceWarning: indexing past lexsort depth may impact perf
  raw_cell, store_history, silent, shell_futures)


Wind Power Variability of Tripoli = 0.3259
Wind Power Variability of Zuwara = 0.3134
Wind Power Variability of Tarhunah = 0.3076
Wind Power Variability of Msallata = 0.315
Wind Power Variability of Ghanima = 0.3163
Wind Power Variability of Gharyan = 0.295
Wind Power Variability of Misurata = 0.3329
Wind Power Variability of Sirte = 0.3259
Wind Power Variability of Magrun = 0.3182
Wind Power Variability of Benghazi = 0.3159
Wind Power Variability of Derna = 0.3144
Wind Power Variability of Tobruk = 0.3136
Wind Power Variability of Houn = 0.2798
Wind Power Variability of Gadamis = 0.2785
Wind Power Variability of Sabha = 0.2867
Wind Power Variability of Kufra = 0.2351
Wind Power Variability of Jaghbub = 0.2863
Wind Power Variability of Berlin = 0.3257
```

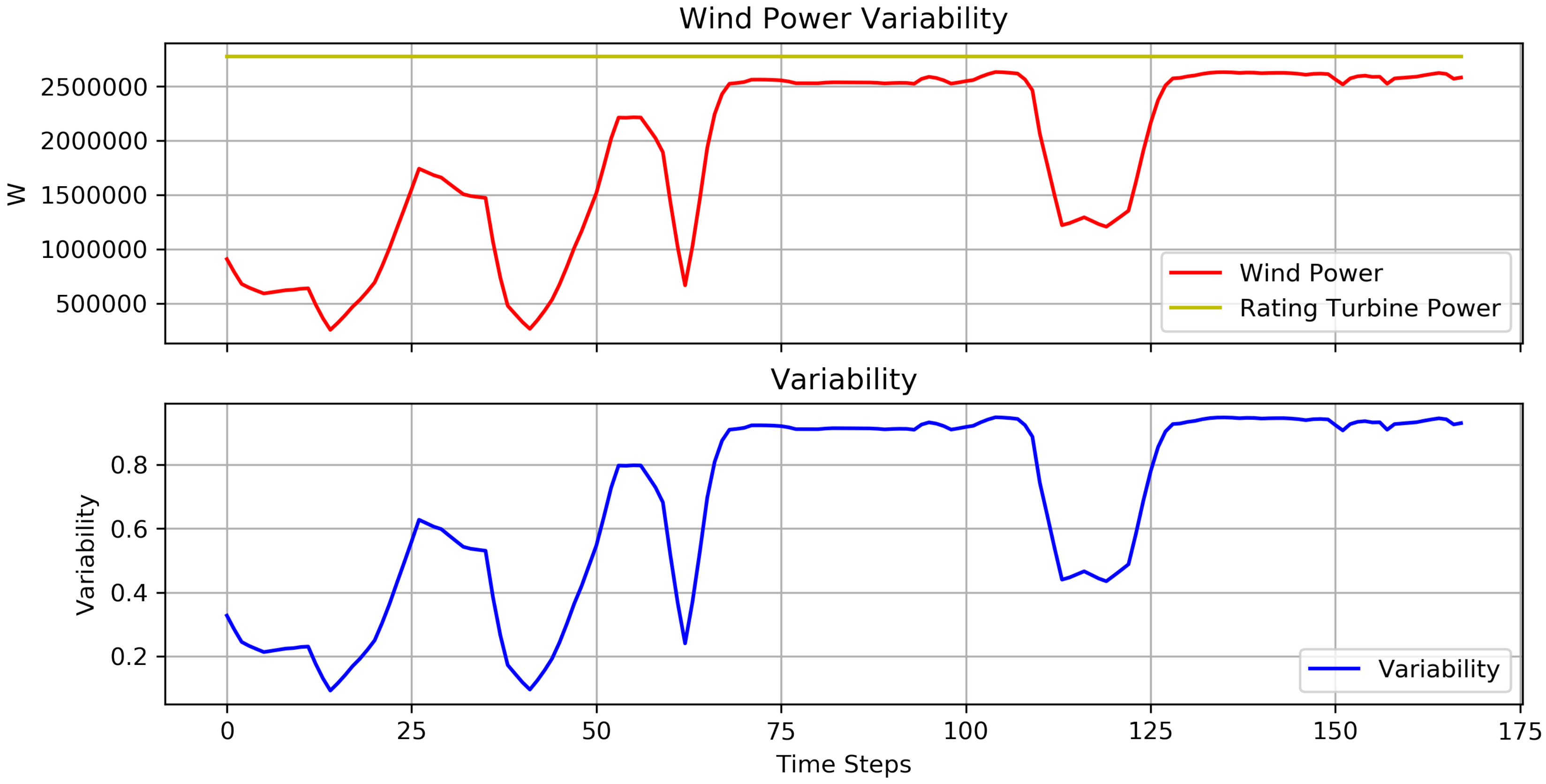
In [36]:
```
df_noise=1-dti_pw
dti_pw.describe()
df_noise.describe()
```

Out[36]:
```
count    8760.000000
mean        0.679462
std         0.325731
min         0.000000
25%         0.434836
50%         0.822675
75%         0.949489
max         1.000000
Name: (PW_Wind,), dtype: float64
```

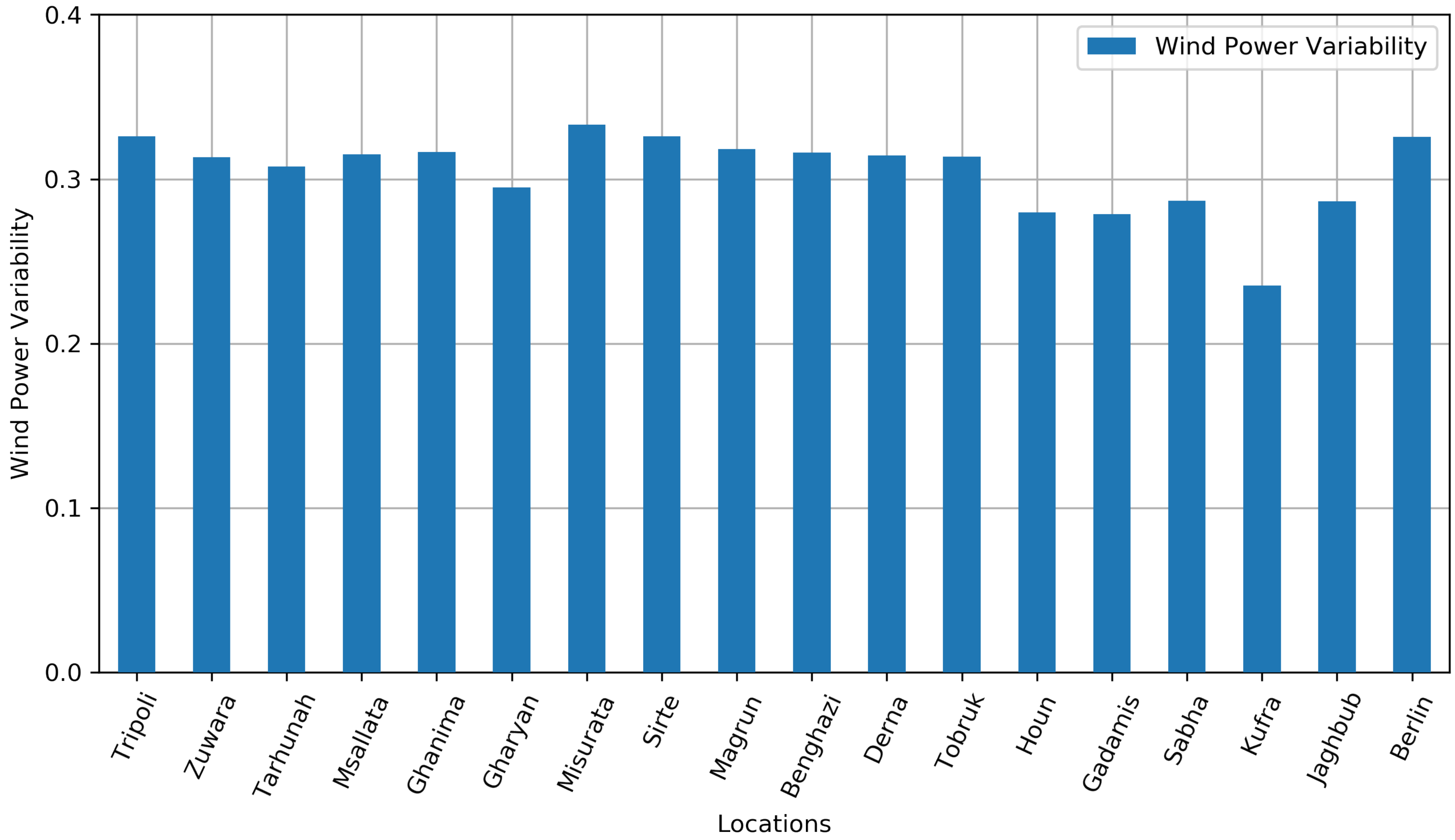
In [37]:
```
# Const_PW=ge120.power_output.max()*np.ones(8760)
dti['Const_PW']=ge120.power_output.max()
```

In [38]:
```
fig, axs = plt.subplots(2)
axs[0].plot(
dti['2021-01-24':'2021-01-30'][['PW_Wind',]].values,'r-',
dti['2021-01-24':'2021-01-30'][['Const_PW']].values,'y-',zorder=3)
axs[0].set_title('Wind Power Variability')
```

```
axs[0].set(xlabel='Time Steps', ylabel='W')
axs[0].legend(['Wind Power', 'Rating Turbine Power'], loc='lower right')
axs[0].label_outer()
axs[0].grid()


axs[1].plot(dti_pw['2021-01-24':'2021-01-30'].values,'b-',zorder=3)
axs[1].set_title('Variability')
axs[1].set(xlabel='Time Steps', ylabel='Variability')
axs[1].legend(['Variability'], loc='lower right')
axs[1].label_outer()
axs[1].grid()


# fig.suptitle('GHI Variability', color='orange')

for ax in axs.flat:
    ax.label_outer()
```




```
In [39]: plt.rcParams['figure.figsize'] = [10, 5.0]
         plt.rcParams['figure.dpi'] = 300

         df_pw_var.plot.bar(zorder=3)
         plt.xticks(rotation=0)
         plt.xlabel('Locations');
         plt.ylabel('Wind Power Variability');
         plt.yticks(np.arange(0, 0.5, 0.1))

         plt.xticks(rotation=65)
         plt.grid()
```



0.6 Aggregation of Some Locations and then Calculate the Variability Factors the Wind Power

In [40]:
```
cities=['Tripoli','Zuwara', 'Tarhunah', 'Msallata', 'Ghanima', 'Gharyan', 'Misurata','Sirte', 'Magrun', 'Benghazi',
         'Derna', 'Tobruk','Houn','Gadamis','Sabha','Kufra', 'Jaghbub','Berlin']

df_dt=['df_Trip','df_Zuwara', 'df_Tarhunah', 'df_Msallata', 'df_Ghanima', 'df_Gharyan', 'df_Misurata',
       'df_Sirte','df_Magrun','df_Benghazi', 'df_Derna', 'df_Tobruk', 'df_Houn','df_Gadamis','df_Sabha',
       'df_Kufra','df_Jaghbub', 'df_Berlin']


df_tem=pd.DataFrame({'Tripoli':df_Trip['PW_Wind']})

for i in np.arange(len(cities)):
    df_tem[cities[i]]=vars()[df_dt[i]]['PW_Wind']

df_pwi=df_tem
df_pwi=df_pwi.drop(df_pwi.columns[-1], axis = 1) ### drop Berlin Column

df_pwi.describe()
```

Out[40]:

	Tripoli	Zuwara	Tarhunah	Msallata	Ghanima
count	8.760000e+03	8.760000e+03	8.760000e+03	8.760000e+03	8.760000e+03
mean	1.124735e+06	9.252876e+05	1.132967e+06	1.108694e+06	1.055098e+06
std	9.045518e+05	8.697813e+05	8.536700e+05	8.740421e+05	8.777576e+05
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	2.849261e+05	1.778221e+05	3.281590e+05	2.773044e+05	2.412286e+05
50%	8.864910e+05	5.692916e+05	9.798336e+05	9.390867e+05	8.290779e+05
75%	2.117460e+06	1.696361e+06	2.045035e+06	2.018223e+06	1.925031e+06
max	2.701884e+06	2.751882e+06	2.572662e+06	2.559644e+06	2.614749e+06

	Gharyan	Misurata	Sirte	Magrun	Benghazi
count	8.760000e+03	8.760000e+03	8.760000e+03	8.760000e+03	8.760000e+03
mean	8.753849e+05	1.203841e+06	1.231967e+06	1.184942e+06	1.253249e+06
std	8.185537e+05	9.237246e+05	9.045546e+05	8.830059e+05	8.767755e+05
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	1.598462e+05	2.872960e+05	3.781087e+05	3.713953e+05	4.411392e+05
50%	5.717048e+05	1.076485e+06	1.101054e+06	1.008379e+06	1.105333e+06
75%	1.543270e+06	2.226730e+06	2.238574e+06	2.151313e+06	2.225949e+06
max	2.505678e+06	2.750912e+06	2.712657e+06	2.693715e+06	2.702251e+06

	Derna	Tobruk	Houn	Gadamis	Sabha
count	8.760000e+03	8.760000e+03	8.760000e+03	8.760000e+03	8.760000e+03
mean	1.642193e+06	1.290598e+06	1.131815e+06	8.549143e+05	1.113515e+06
std	8.726092e+05	8.703866e+05	7.763619e+05	7.729529e+05	7.957098e+05
min	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00	0.000000e+00
25%	8.661029e+05	4.672832e+05	4.291030e+05	2.055079e+05	3.891420e+05
50%	2.026356e+06	1.222763e+06	1.033785e+06	5.869258e+05	9.608268e+05
75%	2.403194e+06	2.235560e+06	1.920503e+06	1.419236e+06	1.928967e+06
max	2.768783e+06	2.664395e+06	2.501903e+06	2.540151e+06	2.487014e+06

	Kufra	Jaghbub
count	8.760000e+03	8.760000e+03
mean	7.559845e+05	8.087718e+05
std	6.524434e+05	7.946478e+05

```
min    0.000000e+00  0.000000e+00
25%    2.384865e+05  1.431595e+05
50%    5.465814e+05  5.113383e+05
75%    1.134275e+06  1.324050e+06
max    2.452718e+06  2.631017e+06
```

In [41]:
```
west_locs=[df_pwi['Tripoli'],df_pwi['Zuwara'], df_pwi['Tarhunah'],df_pwi['Msallata'], df_pwi['Ghanima'],
           df_pwi['Gharyan'],df_pwi['Misurata'], df_pwi['Sirte'], df_pwi['Houn'], df_pwi['Gadamis'],df_pwi['Sabha']]

east_locs=[df_pwi['Magrun'],df_pwi['Benghazi'], df_pwi['Derna'], df_pwi['Tobruk'],df_pwi['Kufra'],df_pwi['Jaghbub']]

df_agg_west_pw=pd.DataFrame(sum(west_locs))
df_agg_east_pw=pd.DataFrame(sum(east_locs))
df_agg_all_pw=pd.DataFrame(df_pwi.sum(axis=1))

df_agg_west_maxpw=len(list(west_locs))*ge120.power_output.max()
df_agg_east_maxpw=len(list(east_locs))*ge120.power_output.max()
df_agg_all_maxpw=len(list(df_pwi))*ge120.power_output.max()

# df_agg_west_pw
# df_agg_east_pw
# df_agg_all_pw
```

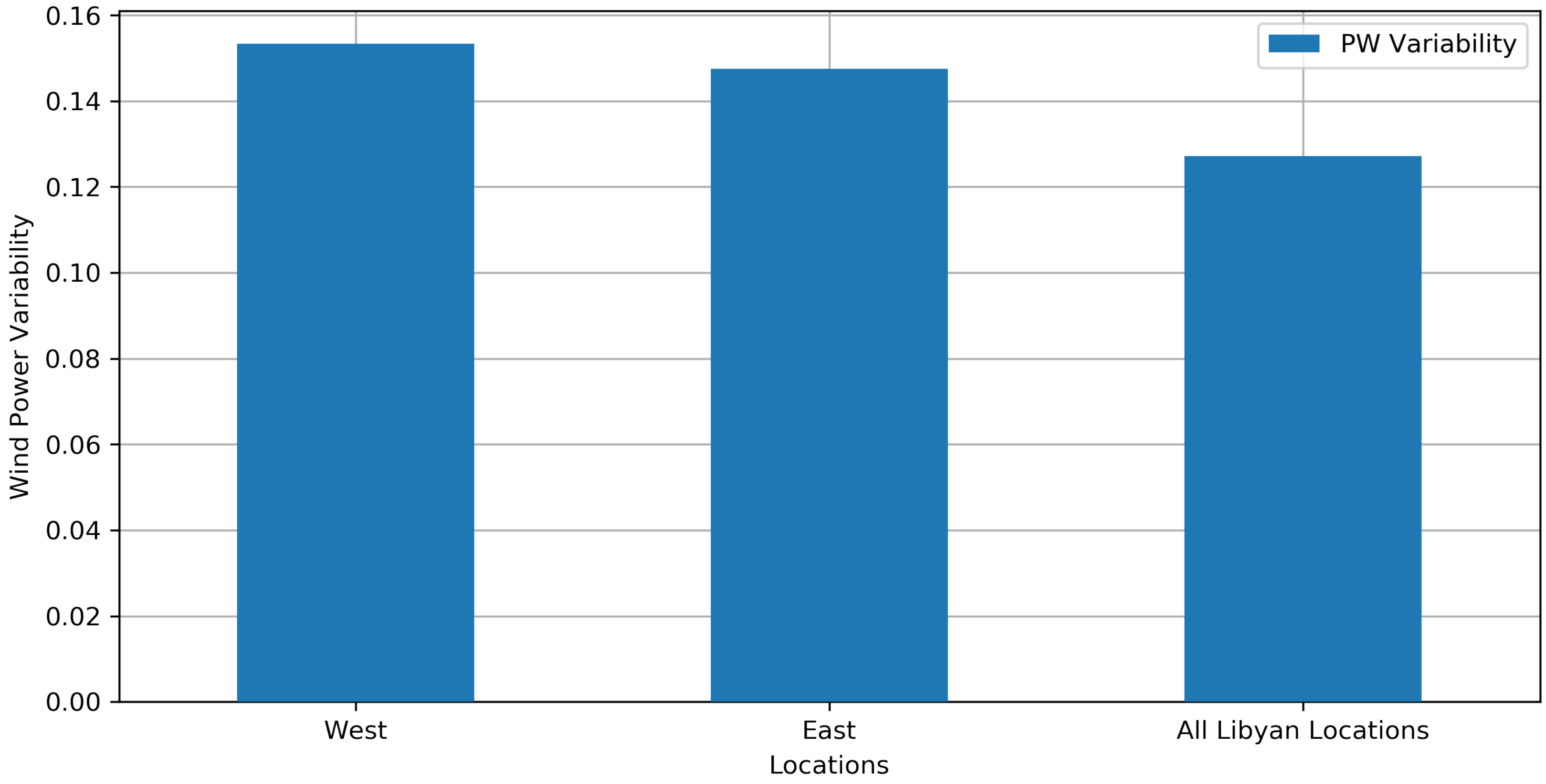
In [42]:
```
df_aggi_pw_variability=[]
cities=['West','East', 'All Libyan Locations']
i=0
dagg=[df_agg_west_pw, df_agg_east_pw, df_agg_all_pw]
dagg_max=[df_agg_west_maxpw, df_agg_east_maxpw, df_agg_all_maxpw]

for daggi in dagg:
    dagg_pw=daggi/dagg_max[i]
    dagg_pw[dagg_pw>10]=0
    dagg_pw[dagg_pw>1]=1
# #     dfagg_pw_variability=1-dagg_pw[0].describe()[1]#mean
    dfagg_pw_variability=dagg_pw[0].describe()[2]#std deviation

    df_aggi_pw_variability.append(dfagg_pw_variability)
    print('PW Variability of', cities[i],'=', round(dfagg_pw_variability,4))
    i=i+1

AggPW_Var_dict= {'PW Variability':df_aggi_pw_variability}
df_aggpw_var=pd.DataFrame(AggPW_Var_dict)
df_aggpw_var.index=cities
df_aggpw_var
```

```
PW Variability of West = 0.1534
PW Variability of East = 0.1475
PW Variability of All Libyan Locations = 0.1272
```

Out[42]:

	PW Variability
West	0.153398
East	0.147481
All Libyan Locations	0.127237

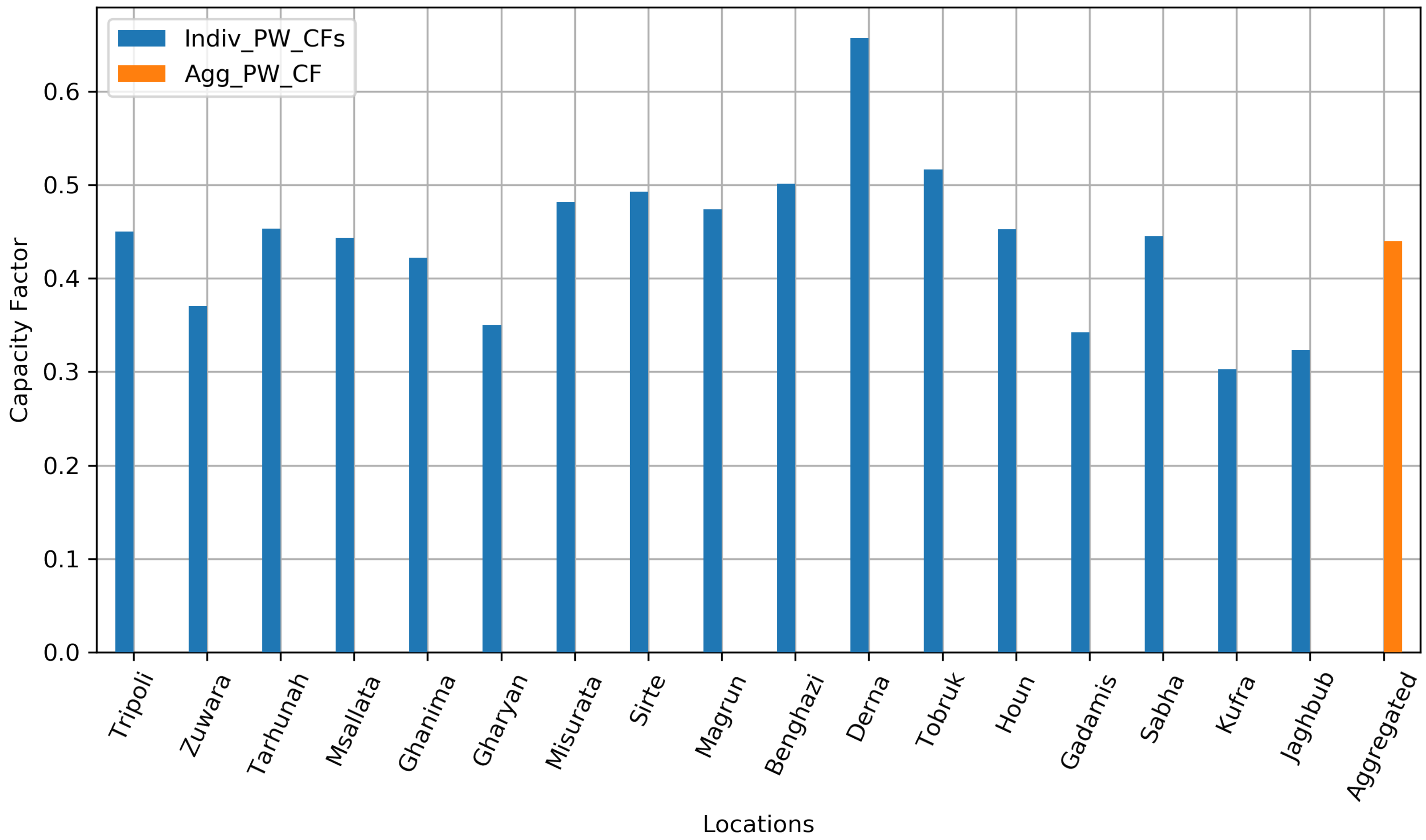
In [43]:
```
plt.rcParams['figure.figsize'] = [10, 5.0]
plt.rcParams['figure.dpi'] = 300

df_aggpw_var.plot.bar(zorder=3)
# plt.yticks(np.arange(0, 1, 0.1))
plt.xticks(rotation=0)
plt.xlabel('Locations');
plt.ylabel('Wind Power Variability');
# plt.legend(bbox_to_anchor=(1.16, 1.01),loc='upper right')
plt.grid()
```



0.6.1 Wind Power Capacity Factor for aggregated locations

```
In [44]: # df_agg_all_pw.describe()

In [45]: typical_hrs=8760
         rating_pw=2500000
         day_hrs=typical_hrs
         # Agg_All_PW_CFs=df_agg_all_pw.sum()/(9*day_hrs*rating_pw)
         # Agg_All_PW_CFs=df_agg_all_pw.sum()/(10*day_hrs*rating_pw)
         Agg_All_PW_CFs=df_agg_all_pw.sum()/(17*day_hrs*rating_pw)

         Agg_All_PW_CFs

Out[45]: 0    0.439858
         dtype: float64

In [46]: # df_pw_cfs.min()
         # df_pw_cfs.max()
         df_pw_cfs.mean()

Out[46]: PW_CFs    0.435189
         dtype: float64

In [47]: PW_AggPW_CFs=pd.DataFrame(df_pw_cfs)
         df2 = pd.DataFrame([Agg_All_PW_CFs], index=['Aggregated'])
         PW_AggPW_CFs=PW_AggPW_CFs.append(df2)
         PW_AggPW_CFs=PW_AggPW_CFs.drop('Berlin')
         PW_AggPW_CFs=PW_AggPW_CFs.rename({0: 'PW_CFs'}, axis=1)
         PW_AggPW_CFs

Out[47]:               PW_CFs    PW_CFs
         Tripoli     0.449894       NaN
         Zuwara      0.370115       NaN
         Tarhunah    0.453187       NaN
         Msallata    0.443478       NaN
         Ghanima     0.422039       NaN
         Gharyan     0.350154       NaN
         Misurata    0.481537       NaN
         Sirte       0.492787       NaN
         Magrun      0.473977       NaN
         Benghazi    0.501299       NaN
         Derna       0.656877       NaN
         Tobruk      0.516239       NaN
         Houn        0.452726       NaN
         Gadamis     0.341966       NaN
         Sabha       0.445406       NaN
         Kufra       0.302394       NaN
         Jaghbub     0.323509       NaN
         Aggregated       NaN  0.439858

In [48]: PW_AggPW_CFs.describe()   #mean of CFs = Agg CF

Out[48]:           PW_CFs    PW_CFs
         count  17.000000  1.000000
         mean    0.439858  0.439858
         std     0.085929       NaN
         min     0.302394  0.439858
         25%     0.370115  0.439858
         50%     0.449894  0.439858
         75%     0.481537  0.439858
         max     0.656877  0.439858
```

In [49]:
```
plt.rcParams['figure.figsize'] = [10, 5.0]
plt.rcParams['figure.dpi'] = 300

PW_AggPW_CFs.plot.bar(zorder=3)
# plt.yticks(np.arange(0, 1, 0.1))
plt.xticks(rotation=0)
# plt.title('Capacity Factors')
plt.xlabel('Locations');
plt.ylabel('Capacity Factor');

plt.legend(['Indiv_PW_CFs','Agg_PW_CF'],loc='upper left')


plt.xticks(rotation=65)
plt.grid()
```



0.6.2 Improvement of aggregation vs. other best variability for each region

In [50]:
```
West_locs=['Tripoli','Zuwara','Tarhunah','Msallata','Ghanima','Gharyan','Misurata','Sirte','Houn','Gadamis',
           'Sabha']

East_locs=['Magrun','Benghazi','Derna','Tobruk','Kufra','Jaghbub']

All_locs=['Tripoli','Zuwara', 'Tarhunah', 'Msallata', 'Ghanima', 'Gharyan', 'Misurata', 'Sirte', 'Magrun', 'Benghazi',
        'Derna', 'Tobruk','Houn','Gadamis','Sabha','Kufra', 'Jaghbub']

min_west=df_pw_var.loc[West_locs].min()
min_east=df_pw_var.loc[East_locs].min()
min_all=df_pw_var.loc[All_locs].min()

pw_improvement_agg_west=(1-(df_aggpw_var/min_west.values))*100
pw_improvement_agg_east=(1-(df_aggpw_var/min_east.values))*100
pw_improvement_agg_all=(1-(df_aggpw_var/min_all.values))*100

df_pw_improv = pd.concat([pw_improvement_agg_west, pw_improvement_agg_east, pw_improvement_agg_all],axis=1, join='inner')
df_pw_improv.columns=['Agg vs. Best West','Agg vs. Best East','Agg vs. Best All']

df_pw_improv
```

Out[50]:

	Agg vs. Best West	Agg vs. Best East	Agg vs. Best All
West	44.925192	34.752601	34.752601
East	47.049281	37.269021	37.269021
All Libyan Locations	54.317646	45.879888	45.879888

In [51]:
```
# df_agg_all_PW=df_agg_all_pw/9
# df_agg_all_PW=df_agg_all_pw/10
df_agg_all_PW=df_agg_all_pw/17


df_indiv_PW=df_Trip['PW_Wind']
print('Avg Aggregated PW:',df_agg_all_PW.mean(),
'Avg individual PW:',df_indiv_PW.mean())

print('Std Aggregated PW:',df_agg_all_PW.std(),
'Std individual PW:',df_indiv_PW.std())
```

```
Avg Aggregated PW: 0    1.099645e+06
dtype: float64 Avg individual PW: 1124735.3772854137
Std Aggregated PW: 0    353103.104608
dtype: float64 Std individual PW: 904551.7877033348
```

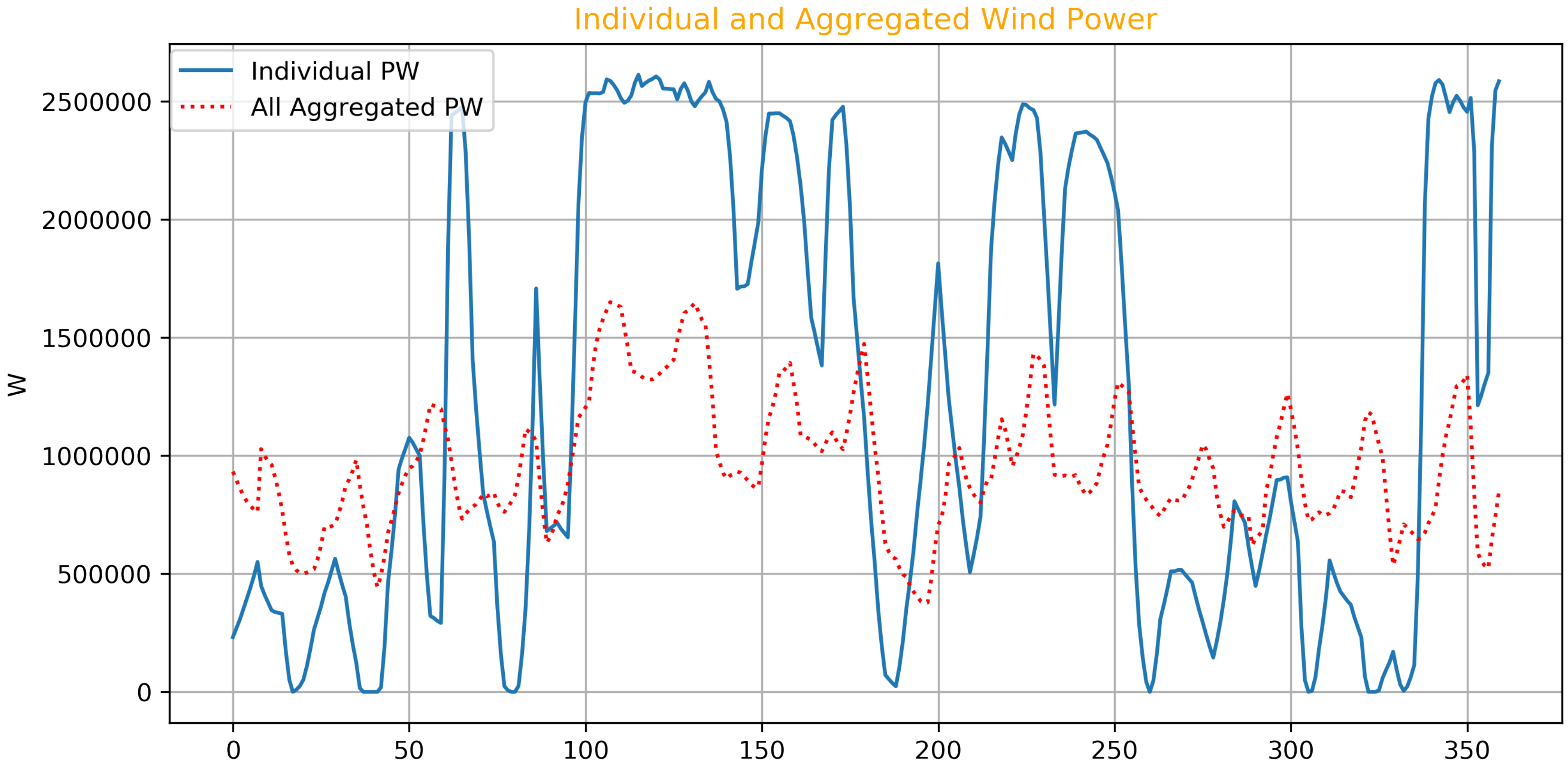
In [52]:
```
df_pwi.describe()
```

Out[52]:
```
            Tripoli        Zuwara      Tarhunah      Msallata       Ghanima  \
count  8.760000e+03  8.760000e+03  8.760000e+03  8.760000e+03  8.760000e+03
mean   1.124735e+06  9.252876e+05  1.132967e+06  1.108694e+06  1.055098e+06
std    9.045518e+05  8.697813e+05  8.536700e+05  8.740421e+05  8.777576e+05
min    0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
25%    2.849261e+05  1.778221e+05  3.281590e+05  2.773044e+05  2.412286e+05
50%    8.864910e+05  5.692916e+05  9.798336e+05  9.390867e+05  8.290779e+05
75%    2.117460e+06  1.696361e+06  2.045035e+06  2.018223e+06  1.925031e+06
max    2.701884e+06  2.751882e+06  2.572662e+06  2.559644e+06  2.614749e+06

            Gharyan      Misurata         Sirte        Magrun     Benghazi  \
count  8.760000e+03  8.760000e+03  8.760000e+03  8.760000e+03  8.760000e+03
mean   8.753849e+05  1.203841e+06  1.231967e+06  1.184942e+06  1.253249e+06
std    8.185537e+05  9.237246e+05  9.045546e+05  8.830059e+05  8.767755e+05
min    0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
25%    1.598462e+05  2.872960e+05  3.781087e+05  3.713953e+05  4.411392e+05
50%    5.717048e+05  1.076485e+06  1.101054e+06  1.008379e+06  1.105333e+06
75%    1.543270e+06  2.226730e+06  2.238574e+06  2.151313e+06  2.225949e+06
max    2.505678e+06  2.750912e+06  2.712657e+06  2.693715e+06  2.702251e+06

              Derna        Tobruk          Houn       Gadamis         Sabha  \
count  8.760000e+03  8.760000e+03  8.760000e+03  8.760000e+03  8.760000e+03
mean   1.642193e+06  1.290598e+06  1.131815e+06  8.549143e+05  1.113515e+06
std    8.726092e+05  8.703866e+05  7.763619e+05  7.729529e+05  7.957098e+05
min    0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
25%    8.661029e+05  4.672832e+05  4.291030e+05  2.055079e+05  3.891420e+05
50%    2.026356e+06  1.222763e+06  1.033785e+06  5.869258e+05  9.608268e+05
75%    2.403194e+06  2.235560e+06  1.920503e+06  1.419236e+06  1.928967e+06
max    2.768783e+06  2.664395e+06  2.501903e+06  2.540151e+06  2.487014e+06

              Kufra       Jaghbub
count  8.760000e+03  8.760000e+03
mean   7.559845e+05  8.087718e+05
std    6.524434e+05  7.946478e+05
min    0.000000e+00  0.000000e+00
25%    2.384865e+05  1.431595e+05
50%    5.465814e+05  5.113383e+05
75%    1.134275e+06  1.324050e+06
max    2.452718e+06  2.631017e+06
```

In [53]:
```
Avg_all_agg=df_agg_all_pw/17
## print(df_pwi.describe())
print('\n')
print('All agg:', Avg_all_agg.describe())
```

```
All agg:                  0
count  8.760000e+03
mean   1.099645e+06
std    3.531031e+05
min    2.788272e+05
25%    8.260719e+05
50%    1.075667e+06
75%    1.356446e+06
max    2.387103e+06
```

In [54]:
```
plt.plot(
    df_indiv_PW['2021-01-01':'2021-01-15'].values,'-',
    df_agg_all_PW['2021-01-01':'2021-01-15'].values,'r:',zorder=3)

plt.legend(['Individual PW', 'All Aggregated PW'],bbox_to_anchor=(-0.008, 1.009),loc='upper left')

plt.ylabel('W')

plt.title('Individual and Aggregated Wind Power', color='orange')

plt.grid()
```


Individual and Aggregated Wind Power
Individual PW
All Aggregated PW
2500000
2000000
1500000
1000000
500000
0
W
0
50
100
150
200
250
300
350

In [ ]:

In [ ]:

In [ ]: