Wind Statistics Using Pandas

Introduction:

The data have been modified to contain some missing values, identified by NaN.
Using pandas should make this exercise easier, in particular for the bonus question.

You should be able to perform all of these operations without using a for loop or other looping construct.

1. The data in 'wind.data' has the following format:

In [1]:

```
"""
Yr Mo Dy   RPT   VAL   ROS   KIL   SHA   BIR   DUB   CLA   MUL   CLO   BEL   MAL
61  1  1 15.04 14.96 13.17  9.29   NaN  9.87 13.67 10.25 10.83 12.58 18.50 15.04
61  1  2 14.71   NaN 10.83  6.50 12.62  7.67 11.50 10.04  9.79  9.67 17.54 13.83
61  1  3 18.50 16.88 12.33 10.13 11.17  6.17 11.25   NaN  8.50  7.67 12.75 12.71
"""
```

Out[1]:

```
'\nYr Mo Dy   RPT   VAL   ROS   KIL   SHA   BIR   DUB   CLA   MUL   CLO
BEL   MAL\n61  1  1 15.04 14.96 13.17  9.29   NaN  9.87 13.67 10.25 10.83
12.58 18.50 15.04\n61  1  2 14.71   NaN 10.83  6.50 12.62  7.67 11.50 10.0
4  9.79  9.67 17.54 13.83\n61  1  3 18.50 16.88 12.33 10.13 11.17  6.17 1
1.25   NaN  8.50  7.67 12.75 12.71\n'
```

The first three columns are year, month and day. The remaining 12 columns are average windspeeds in knots at 12 locations in Ireland on that day.

Step 1. Import the necessary libraries

In [1]:

```
import pandas as pd
import numpy as np
import datetime as dt
```

Step 2. Import the dataset

In [33]:

```
pd.read_csv("wind_dataset.csv")
```

Out[33]:

	Yr Mo Dy RPT VAL ROS KIL SHA BIR DUB CLA MUL CLO BEL MAL
0	61 1 1 15.04 14.96 13.17 9.29 NaN 9.87 1...
1	61 1 2 14.71 NaN 10.83 6.50 12.62 7.67 1...
2	61 1 3 18.50 16.88 12.33 10.13 11.17 6.17 1...
3	61 1 4 10.58 6.63 11.75 4.58 4.54 2.88 ...
4	61 1 5 13.33 13.25 11.42 6.17 10.71 8.21 1...
...	...
6569	78 12 27 17.58 16.96 17.62 8.08 13.21 11.67 1...
6570	78 12 28 13.21 5.46 13.46 5.00 8.12 9.42 1...
6571	78 12 29 14.00 10.29 14.42 8.71 9.71 10.54 1...
6572	78 12 30 18.50 14.04 21.29 9.13 12.75 9.71 1...
6573	78 12 31 20.33 17.41 27.29 9.59 12.08 10.13 1...

6574 rows × 1 columns

Step 3. Assign it to a variable called data and replace the first 3 columns by a proper datetime index.

In [46]:

```
data = pd.read_csv("wind_dataset.csv",sep ="\s+",parse_dates= {'Date':['Yr','Mo','Dy']})
data
```

Out[46]:

	Date	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA	MUL	CLO	BEL	MAL
0	2061-01-01	15.04	14.96	13.17	9.29	NaN	9.87	13.67	10.25	10.83	12.58	18.50	15.0
1	2061-01-02	14.71	NaN	10.83	6.50	12.62	7.67	11.50	10.04	9.79	9.67	17.54	13.8
2	2061-01-03	18.50	16.88	12.33	10.13	11.17	6.17	11.25	NaN	8.50	7.67	12.75	12.7
3	2061-01-04	10.58	6.63	11.75	4.58	4.54	2.88	8.63	1.79	5.83	5.88	5.46	10.8
4	2061-01-05	13.33	13.25	11.42	6.17	10.71	8.21	11.92	6.54	10.92	10.34	12.92	11.8
...	...	...	...	...	...	...	...	...	...	...	...	...	..
6569	1978-12-27	17.58	16.96	17.62	8.08	13.21	11.67	14.46	15.59	14.04	14.00	17.21	40.0
6570	1978-12-28	13.21	5.46	13.46	5.00	8.12	9.42	14.33	16.25	15.25	18.05	21.79	41.4
6571	1978-12-29	14.00	10.29	14.42	8.71	9.71	10.54	19.17	12.46	14.50	16.42	18.88	29.5
6572	1978-12-30	18.50	14.04	21.29	9.13	12.75	9.71	18.08	12.87	12.46	12.12	14.67	28.7
6573	1978-12-31	20.33	17.41	27.29	9.59	12.08	10.13	19.25	11.63	11.58	11.38	12.08	22.0

6574 rows × 13 columns

Step 4. Year 2061? Do we really have data from this year? Create a function to fix it and apply it.

In [48]:

```
data["Date"] = np.where(pd.DatetimeIndex(data["Date"]).year < 2000,data.Date,data.Date -
data
```

Out[48]:

	Date	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA	MUL	CLO	BEL	MAL
0	1961-01-01	15.04	14.96	13.17	9.29	NaN	9.87	13.67	10.25	10.83	12.58	18.50	15.04
1	1961-01-02	14.71	NaN	10.83	6.50	12.62	7.67	11.50	10.04	9.79	9.67	17.54	13.83
2	1961-01-03	18.50	16.88	12.33	10.13	11.17	6.17	11.25	NaN	8.50	7.67	12.75	12.71
3	1961-01-04	10.58	6.63	11.75	4.58	4.54	2.88	8.63	1.79	5.83	5.88	5.46	10.88
4	1961-01-05	13.33	13.25	11.42	6.17	10.71	8.21	11.92	6.54	10.92	10.34	12.92	11.83
...	...	...	...	...	...	...	...	...	...	...	...	...	..
6569	1978-12-27	17.58	16.96	17.62	8.08	13.21	11.67	14.46	15.59	14.04	14.00	17.21	40.08
6570	1978-12-28	13.21	5.46	13.46	5.00	8.12	9.42	14.33	16.25	15.25	18.05	21.79	41.46
6571	1978-12-29	14.00	10.29	14.42	8.71	9.71	10.54	19.17	12.46	14.50	16.42	18.88	29.58
6572	1978-12-30	18.50	14.04	21.29	9.13	12.75	9.71	18.08	12.87	12.46	12.12	14.67	28.79
6573	1978-12-31	20.33	17.41	27.29	9.59	12.08	10.13	19.25	11.63	11.58	11.38	12.08	22.08

6574 rows × 13 columns

Step 5. Set the right dates as the index. Pay attention at the data type, it should be datetime64[ns].

In [51]:

```
Data = data.set_index("Date")
Data.index.astype("datetime64[ns]")
```

Out[51]:

```
DatetimeIndex(['1961-01-01', '1961-01-02', '1961-01-03', '1961-01-04',
               '1961-01-05', '1961-01-06', '1961-01-07', '1961-01-08',
               '1961-01-09', '1961-01-10',
               ...
               '1978-12-22', '1978-12-23', '1978-12-24', '1978-12-25',
               '1978-12-26', '1978-12-27', '1978-12-28', '1978-12-29',
               '1978-12-30', '1978-12-31'],
              dtype='datetime64[ns]', name='Date', length=6574, freq=None)
```

In [52]:

```
Data
```

Out[52]:

	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA	MUL	CLO	BEL	MAL
Date												
1961-01-01	15.04	14.96	13.17	9.29	NaN	9.87	13.67	10.25	10.83	12.58	18.50	15.04
1961-01-02	14.71	NaN	10.83	6.50	12.62	7.67	11.50	10.04	9.79	9.67	17.54	13.83
1961-01-03	18.50	16.88	12.33	10.13	11.17	6.17	11.25	NaN	8.50	7.67	12.75	12.71
1961-01-04	10.58	6.63	11.75	4.58	4.54	2.88	8.63	1.79	5.83	5.88	5.46	10.88
1961-01-05	13.33	13.25	11.42	6.17	10.71	8.21	11.92	6.54	10.92	10.34	12.92	11.83
...	...	...	...	...	...	...	...	...	...	...	...	...
1978-12-27	17.58	16.96	17.62	8.08	13.21	11.67	14.46	15.59	14.04	14.00	17.21	40.08
1978-12-28	13.21	5.46	13.46	5.00	8.12	9.42	14.33	16.25	15.25	18.05	21.79	41.46
1978-12-29	14.00	10.29	14.42	8.71	9.71	10.54	19.17	12.46	14.50	16.42	18.88	29.58
1978-12-30	18.50	14.04	21.29	9.13	12.75	9.71	18.08	12.87	12.46	12.12	14.67	28.79
1978-12-31	20.33	17.41	27.29	9.59	12.08	10.13	19.25	11.63	11.58	11.38	12.08	22.08

6574 rows × 12 columns

Step 6. Compute how many values are missing for each location over the entire record.

They should be ignored in all calculations below.

In [64]:

```
Data.isnull().values.ravel().sum()
```

Out[64]:

```
31
```

Step 7. Compute how many non-missing values there are in total.

In [65]:

```
x = Data.count()
print(x.sum())
```

```
78857
```

Step 8. Calculate the mean windspeeds of the windspeeds over all the locations and all the times.

A single number for the entire dataset.

In [66]:

```
y = Data.mean()
y.mean()
```

Out[66]:

```
10.227982360836924
```

Step 9. Create a DataFrame called loc_stats and calculate the min, max and mean windspeeds and standard deviations of the windspeeds at each location over all the days

A different set of numbers for each location.

In [67]:

```
def stats(x):
    x = pd.Series(x)
    Min = x.min()
    Max = x.max()
    Mean = x.mean()
    Std = x.std()
    res = [Min,Max,Mean,Std]
    indx = ["Min","Max","Mean","Std"]
    res = pd.Series(res,index =indx)
    return res
loc_stats = Data.apply(stats)
loc_stats
```

Out[67]:

	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CL
Min	0.670000	0.210000	1.500000	0.000000	0.130000	0.000000	0.000000	0.00000
Max	35.800000	33.370000	33.840000	28.460000	37.540000	26.160000	30.370000	31.0800
Mean	12.362987	10.644314	11.660526	6.306468	10.455834	7.092254	9.797343	8.4950
Std	5.618413	5.267356	5.008450	3.605811	4.936125	3.968683	4.977555	4.4994

Step 10. Create a DataFrame called day_stats and calculate the min, max and mean windspeed and standard deviations of the windspeeds across all the locations at each day.

A different set of numbers for each day.

In [68]:

```
day_stats=Data.apply(stats, axis=1)
day_stats.head()
```

Out[68]:

	Min	Max	Mean	Std
Date				
1961-01-01	9.29	18.50	13.018182	2.808875
1961-01-02	6.50	17.54	11.336364	3.188994
1961-01-03	6.17	18.50	11.641818	3.681912
1961-01-04	1.79	11.75	6.619167	3.198126
1961-01-05	6.17	13.33	10.630000	2.445356

Step 11. Find the average windspeed in January for each location.

Treat January 1961 and January 1962 both as January.

In [69]:

```
january_data = Data[Data.index.month == 1]
print("January Windspeeds : ")
print(january_data.mean())
```

```
January Windspeeds :
RPT    14.847325
VAL    12.914560
ROS    13.299624
KIL     7.199498
SHA    11.667734
BIR     8.054839
DUB    11.819355
CLA     9.512047
MUL     9.543208
CLO    10.053566
BEL    14.550520
MAL    18.028763
dtype: float64
```

Step 12. Downsample the record to a yearly frequency for each location.

In [70]:

```
print("Yearly:\n",Data.resample('A').mean())
```

```
Yearly:
                  RPT        VAL        ROS       KIL        SHA       BI
R  \
Date
1961-12-31  12.299583  10.351796  11.362369  6.958227  10.881763  7.729726
1962-12-31  12.246923  10.110438  11.732712  6.960440  10.657918  7.393068
1963-12-31  12.813452  10.836986  12.541151  7.330055  11.724110  8.434712
1964-12-31  12.363661  10.920164  12.104372  6.787787  11.454481  7.570874
1965-12-31  12.451370  11.075534  11.848767  6.858466  11.024795  7.478110
1966-12-31  13.461973  11.557205  12.020630  7.345726  11.805041  7.793671
1967-12-31  12.737151  10.990986  11.739397  7.143425  11.630740  7.368164
1968-12-31  11.835628  10.468197  11.409754  6.477678  10.760765  6.067322
1969-12-31  11.166356   9.723699  10.902000  5.767973   9.873918  6.189973
1970-12-31  12.600329  10.726932  11.730247  6.217178  10.567370  7.609452
1971-12-31  11.273123   9.095178  11.088329  5.241507   9.440329  6.097151
1972-12-31  12.463962  10.561311  12.058333  5.929699   9.430410  6.358825
1973-12-31  11.828466  10.680493  10.680493  5.547863   9.640877  6.548740
1974-12-31  13.643096  11.811781  12.336356  6.427041  11.110986  6.809781
1975-12-31  12.008575  10.293836  11.564712  5.269096   9.190082  5.668521
1976-12-31  11.737842  10.203115  10.761230  5.109426   8.846339  6.311038
1977-12-31  13.099616  11.144493  12.627836  6.073945  10.003836  8.586438
1978-12-31  12.504356  11.044274  11.380000  6.082356  10.167233  7.650658

                  DUB        CLA       MUL        CLO        BEL         MA
L
Date
1961-12-31   9.733923   8.858788  8.647652   9.835577  13.502795  13.68077
3
1962-12-31  11.020712   8.793753  8.316822   9.676247  12.930685  14.32395
6
1963-12-31  11.075699  10.336548  8.903589  10.224438  13.638877  14.99901
4
1964-12-31  10.259153   9.467350  7.789016  10.207951  13.740546  14.91030
1
1965-12-31  10.618712   8.879918  7.907425   9.918082  12.964247  15.59164
4
1966-12-31  10.579808   8.835096  8.514438   9.768959  14.265836  16.30726
0
1967-12-31  10.652027   9.325616  8.645014   9.547425  14.774548  17.13594
5
1968-12-31   8.859180   8.255519  7.224945   7.832978  12.808634  15.01748
6
1969-12-31   8.564493   7.711397  7.924521   7.754384  12.621233  15.76290
4
1970-12-31   9.609890   8.334630  9.297616   8.289808  13.183644  16.45602
7
1971-12-31   8.385890   6.757315  7.915370   7.229753  12.208932  15.02523
3
1972-12-31   9.704508   7.680792  8.357295   7.515273  12.727377  15.02871
6
1973-12-31   8.482110   7.614274  8.245534   7.812411  12.169699  15.44109
6
1974-12-31  10.084603   9.896986  9.331753   8.736356  13.252959  16.94767
1
1975-12-31   8.562603   7.843836  8.797945   7.382822  12.631671  15.30786
3
1976-12-31   9.149126   7.146202  8.883716   7.883087  12.332377  15.47144
8
1977-12-31  11.523205   8.378384  9.098192   8.821616  13.459068  16.59084
9
```

```
1978-12-31   9.489342   8.800466  9.089753   8.301699  12.967397  16.77137
0
```

Step 13. Downsample the record to a monthly frequency for each location.

In [71]:

```
Data.resample('M').mean()
```

Out[71]:

	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA
Date								
1961-01-31	14.841333	11.988333	13.431613	7.736774	11.072759	8.588065	11.184839	9.24533
1961-02-28	16.269286	14.975357	14.441481	9.230741	13.852143	10.937500	11.890714	11.84607
1961-03-31	10.890000	11.296452	10.752903	7.284000	10.509355	8.866774	9.644194	9.82967
1961-04-30	10.722667	9.427667	9.998000	5.830667	8.435000	6.495000	6.925333	7.09466
1961-05-31	9.860968	8.850000	10.818065	5.905333	9.490323	6.574839	7.604000	8.17709
...	...	...	...	...	...	...	...	..
1978-08-31	9.645161	8.259355	9.032258	4.502903	7.368065	5.935161	5.650323	5.41774
1978-09-30	10.913667	10.895000	10.635000	5.725000	10.372000	9.278333	10.790333	9.58300
1978-10-31	9.897742	8.670968	9.295806	4.721290	8.525161	6.774194	8.115484	7.33774
1978-11-30	16.151667	14.802667	13.508000	7.317333	11.475000	8.743000	11.492333	9.65733
1978-12-31	16.175484	13.748065	15.635161	7.094839	11.398710	9.241613	12.077419	10.19483

216 rows × 12 columns

Step 14. Downsample the record to a weekly frequency for each location.

In [72]:

```
Data.resample('W').mean()
```

Out[72]:

	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CL
Date								
1961-01-01	15.040000	14.960000	13.170000	9.290000	NaN	9.870000	13.670000	10.2500
1961-01-08	13.541429	11.486667	10.487143	6.417143	9.474286	6.435714	11.061429	6.6166
1961-01-15	12.468571	8.967143	11.958571	4.630000	7.351429	5.072857	7.535714	6.8200
1961-01-22	13.204286	9.862857	12.982857	6.328571	8.966667	7.417143	9.257143	7.8757
1961-01-29	19.880000	16.141429	18.225714	12.720000	17.432857	14.828571	15.528571	15.1600
...	...	...	...	...	...	...	...	
1978-12-03	14.934286	11.232857	13.941429	5.565714	10.215714	8.618571	9.642857	7.6857
1978-12-10	20.740000	19.190000	17.034286	9.777143	15.287143	12.774286	14.437143	12.4885
1978-12-17	16.758571	14.692857	14.987143	6.917143	11.397143	7.272857	10.208571	7.9671
1978-12-24	11.155714	8.008571	13.172857	4.004286	7.825714	6.290000	7.798571	8.6671
1978-12-31	14.951429	11.801429	16.035714	6.507143	9.660000	8.620000	13.708571	10.4771

940 rows × 12 columns

Step 15. Calculate the min, max and mean windspeeds and standard deviations of the windspeeds across all locations for each week (assume that the first week starts on January 2 1961) for the first 52 weeks.

In [73]:

```
Data = Data.groupby(lambda x: (x.month, x.year))
Data.mean()
```

Out[73]:

	RPT	VAL	ROS	KIL	SHA	BIR	DUB	CLA
(1, 1961)	14.841333	11.988333	13.431613	7.736774	11.072759	8.588065	11.184839	9.24533
(1, 1962)	14.783871	13.160323	12.591935	7.538065	11.779677	8.720000	14.211935	9.60000
(1, 1963)	14.868387	11.112903	15.121613	6.635806	11.080645	7.835484	12.797419	9.84483
(1, 1964)	12.661290	11.818387	11.741290	6.953548	11.400645	6.865806	9.592903	9.68741
(1, 1965)	15.741613	15.546774	15.274194	8.258387	13.588065	9.251290	13.850968	11.26000
...	...	...	...	...	...	...	...	..
(12, 1974)	18.511290	17.805806	14.773871	9.734839	16.944194	10.153871	16.602903	15.03419
(12, 1975)	11.655484	8.686774	11.217742	4.478387	6.628710	4.178065	10.351290	6.17612
(12, 1976)	11.962258	10.086774	10.474516	3.383871	7.645484	6.148387	8.034516	4.50000
(12, 1977)	14.751935	12.744839	13.469677	6.592258	11.247742	9.466774	13.231613	10.70387
(12, 1978)	16.175484	13.748065	15.635161	7.094839	11.398710	9.241613	12.077419	10.19483

216 rows × 12 columns

In [ ]: