Why deal with missing data?

DEALING WITH MISSING DATA IN PYTHON

Suraj Donthi
Deep Learning & Computer Vision Consultant

Why does missing data exist?

- Real world data is messy data

> Did you know that 72% of organizations believe that data quality issues hinder customer trust and perception?

[1] [Top 9 Benefits of Data Cleansing for Businesses](https://bit.ly/2QwMrab)

Why does missing data exist?

- Values are missed during data acquisition process
  - Faulty weather sensors during weather analysis
  - Incomplete patient information for medical diagnosis etc.
- Values deleted accidentally
  - Data loss
  - Mistakenly deleted due to human error

In this course, you'll learn

- the significance of treating missing values
- to detect missing values in your messy data
- analyze the types for missingness
- treat the missing values appropriately for
  - numerical
  - time-series
  - categorical values

In this course, you'll learn

- to impute(replace) missing values using simple techniques
- to impute using advanced techniques
- to finally evaluate the best method of treating missing values

Workflow for treating missing values

1. Convert all missing values to null values.
2. Analyze the amount and type of missingness in the data.
3. Appropriately delete or impute missing values.
4. Evaluate & compare the performance of the treated/imputed dataset.

NULL value Operations

`None`

```
None or True # Same for False
```

```
True
```

```
None + True # For all operators
```

```
TypeError: unsupported operand
```

```
None / 3 # For all operators
```

```
TypeError: unsupported operand
```

```
type(None)
```

```
NoneType
```

`np.nan`

```
import numpy as np
np.nan or True  # Same for False
```

```
nan
```

```
np.nan * True # For all operators
```

```
nan
```

```
np.nan - 3 # For all operators
```

```
nan
```

```
type(np.nan)
```

```
float
```

NULL value operations

`None`

```
None == None
```

```
True
```

```
np.isnan(None)
```

```
False
```

`np.nan`

```
np.nan == np.nan
```

```
False
```

```
np.isnan(np.nan)
```

```
True
```

Let's practice!

DEALING WITH MISSING DATA IN PYTHON

Handling missing values

DEALING WITH MISSING DATA IN PYTHON

Suraj Donthi
Deep Learning & Computer Vision Consultant

Missing values

- Usually filled with values like `'NA'`, `'-'` or `'.'` etc.

Detect missing values in College dataset

College Dataset

```
college = pd.read_csv('college.csv')

college.head()
```

```
  gradrat            lenroll rmbrd   private stufac     csat   act
0    59.0       5.1761497326  3.75       1.0   10.8        .  21.0
1    52.0       4.7791234931  3.74       1.0   17.7        .  21.0
2    75.0  6.122492809500001     .       1.0   11.4   1052.0  24.0
3    56.0       5.3181199938   4.1       1.0   11.6    940.0  23.0
4    71.0  5.631211781799999     .       1.0   18.3        .  17.0
```

Detect missing values in College dataset

```
college.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 7 columns):
gradrat     200 non-null object
lenroll     200 non-null object
rmbrd       200 non-null object
private     200 non-null float64
stufac      200 non-null object
csat        200 non-null object
act         200 non-null object
dtypes: float64(1), object(6)
```

Detect missing values in College dataset

```
csat_unique = college.csat.unique()
np.sort(csat_unique)
```

```
array(['.', '1000.0', '1006.0', '1010.0', '1013.0', '1020.0', '1024.0',
       '1026.0', '1028.0', '1036.0', '1039.0', '1040.0', '1044.0',
       '1045.0', '1050.0', '1052.0', '1060.0', '1070.0', '1080.0',
       '1092.0', '1096.0', '1109.0', '1111.0', '1120.0', '1139.0',
          ...       ...        ...        ...        ...        ...
          ...       ...        ...        ...        ...        ...
       '940.0', '943.0', '947.0', '950.0', '951.0', '964.0', '970.0',
       '979.0', '980.0', '989.0', '992.0', '994.0', '996.0', '997.0',
       '998.0'], dtype=object)
```

Replace missing values in College dataset

```
college = pd.read_csv('college.csv', na_values='.')

college.head()
```

```
   gradrat   lenroll  rmbrd  private  stufac    csat   act
0     59.0  5.176150   3.75      1.0    10.8     NaN  21.0
1     52.0  4.779123   3.74      1.0    17.7     NaN  21.0
2     75.0  6.122493    NaN      1.0    11.4  1052.0  24.0
3     56.0  5.318120   4.10      1.0    11.6   940.0  23.0
4     71.0  5.631212    NaN      1.0    18.3     NaN  17.0
```

Replace missing values in College dataset

```
college.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 7 columns):
gradrat     187 non-null float64
lenroll     199 non-null float64
rmbrd       114 non-null float64
private     200 non-null float64
stufac      199 non-null float64
csat        105 non-null float64
act         104 non-null float64
dtypes: float64(7)
```

Detect missing values in Diabetes dataset

Pima Indian Diabetes dataset

- contains various clinical diagnostic information of the patients from the Pima community

```
diabetes = pd.read_csv('pima-indians-diabetes.csv')
```

	Pregnant	Glucose	Diastolic_BP	Skin_Fold	Serum_Insulin	BMI	Diabetes_Pedigree	Age	Class
0	6.0	148.0	72.0	35.0	NaN	33.6	0.627	50	1.0
1	1.0	85.0	66.0	29.0	NaN	26.6	0.351	31	0.0
2	8.0	183.0	64.0	NaN	NaN	23.3	0.672	32	1.0
3	1.0	89.0	66.0	23.0	94.0	28.1	0.167	21	0.0
4	0.0	137.0	40.0	35.0	168.0	43.1	2.288	33	1.0

Detect missing values in Diabetes dataset

```
diabetes.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
Pregnant             768 non-null float64
Glucose              763 non-null float64
Diastolic_BP         733 non-null float64
Skin_Fold            541 non-null float64
Serum_Insulin        394 non-null float64
BMI                  768 non-null float64
Diabetes_Pedigree    768 non-null float64
Age                  768 non-null int64
Class                768 non-null float64
dtypes: float64(8), int64(1)
```

Detect missing values in Diabetes dataset

```
diabetes.describe()
```

	Pregnant	Glucose	Diastolic_BP	Skin_Fold	Serum_Insulin	BMI	Diabetes_Pedigree	Age	Class
count	768.000000	763.000000	733.000000	541.000000	394.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	121.686763	72.405184	29.153420	155.548223	31.992578	0.471876	33.240885	0.348958
std	3.369578	30.535641	12.382158	10.476982	118.775855	7.884160	0.331329	11.760232	0.476951
min	0.000000	44.000000	24.000000	7.000000	14.000000	0.000000	0.078000	21.000000	0.000000
25%	1.000000	99.000000	64.000000	22.000000	76.250000	27.300000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	29.000000	125.000000	32.000000	0.372500	29.000000	0.000000
75%	6.000000	141.000000	80.000000	36.000000	190.000000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

Detect missing values in Diabetes dataset

```
diabetes.BMI[diabetes.BMI == 0]
```

```
9  |    0.0
49 |    0.0
60 |    0.0
81 |    0.0
145|    0.0
371|    0.0
426|    0.0
494|    0.0
522|    0.0
684|    0.0
706|    0.0
Name: BMI, dtype: float64
```

Replace missing values with NaN

```
diabetes.BMI[diabetes.BMI == 0] = np.nan
diabetes.BMI[np.isnan(diabetes.BMI)]
```

```
9   |    NaN
49  |    NaN
60  |    NaN
81  |    NaN
145|    NaN
371|    NaN
426|    NaN
494|    NaN
522|    NaN
684|    NaN
706|    NaN
Name: BMI, dtype: float64
```

Summary

- detect missing value characters like '.' etc.
- detect the inherent missing values within the data like '0'.
- replace them values with `NaN`

Let's practice!

DEALING WITH MISSING DATA IN PYTHON

Analyze the amount of missingness

DEALING WITH MISSING DATA IN PYTHON

Suraj Donthi
Deep Learning & Computer Vision Consultant

Load Air Quality dataset

Air Quality dataset

- contains the sensor recordings of Ozone, Solar, Temperature and Wind

```
df_air = pd.read_csv('air-quality.csv',
                     parse_dates=['Date'],
                     index_col='Date')

df_air.head()
```

```
              Ozone  Solar  Wind  Temp
Date
1976-05-01     41.0  190.0   7.4    67
1976-05-02     36.0  118.0   8.0    72
1976-05-03     12.0  149.0  12.6    74
1976-05-04     18.0  313.0  11.5    62
1976-05-05      NaN    NaN  14.3    56
```

Nullity DataFrame

- Use either `.isnull()` or `.isna()` methods on the DataFrame

```
airquality_nullity = airquality.isnull()
airquality_nullity.head()
```

```
            Ozone  Solar   Wind   Temp
Date
1976-05-01  False  False  False  False
1976-05-02  False  False  False  False
1976-05-03  False  False  False  False
1976-05-04  False  False  False  False
1976-05-05   True   True  False  False
```

Total missing values

```
airquality_nullity.sum()
```

```
Ozone     37
Solar      7
Wind       0
Temp       0
dtype: int64
```

Percentage of missingness

```
airquality_nullity.mean() * 100
```

```
Ozone    24.183007
Solar     4.575163
Wind      0.000000
Temp      0.000000
dtype: float64
```

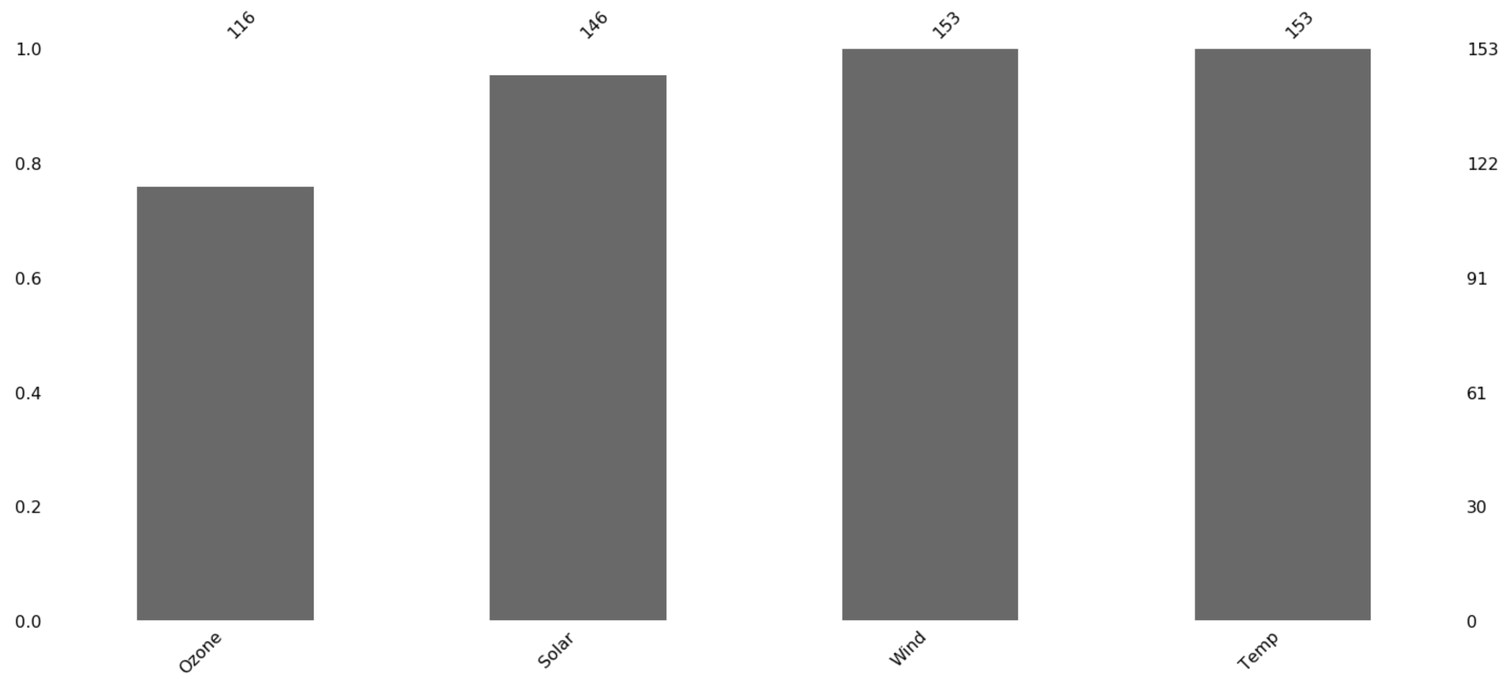
Nullity Bar

Missingno package

- Package for graphical analysis of missing values

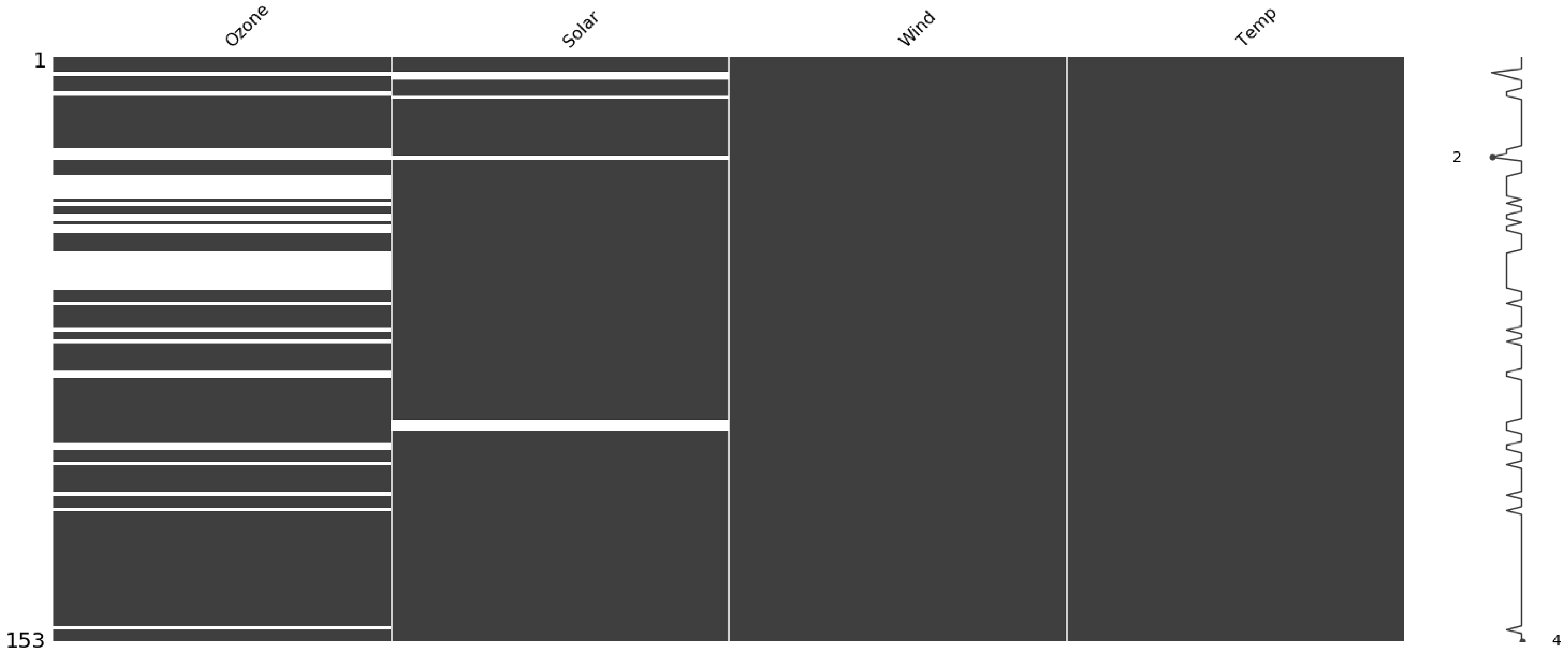
```
import missingno as msno
```

```
msno.bar(airquality)
```



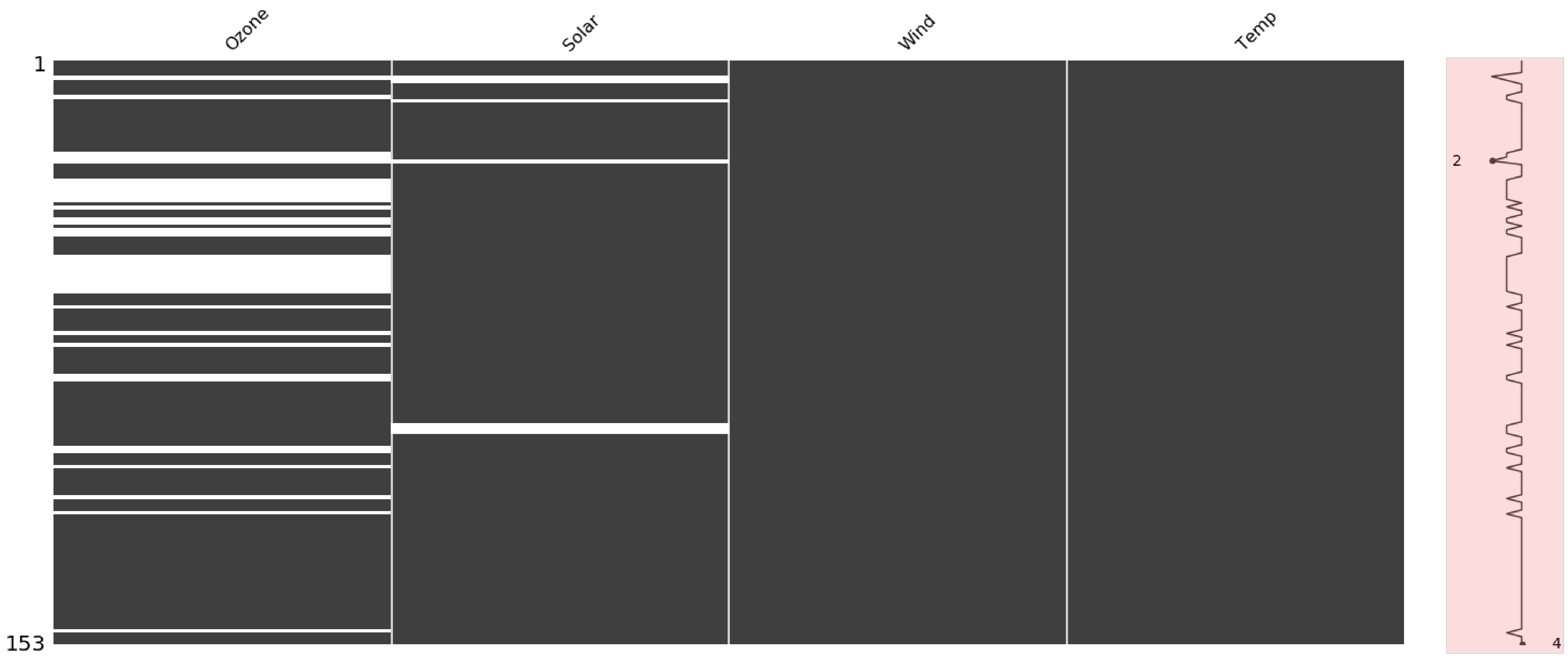
Nullity Matrix

```
msno.matrix(airquality)
```



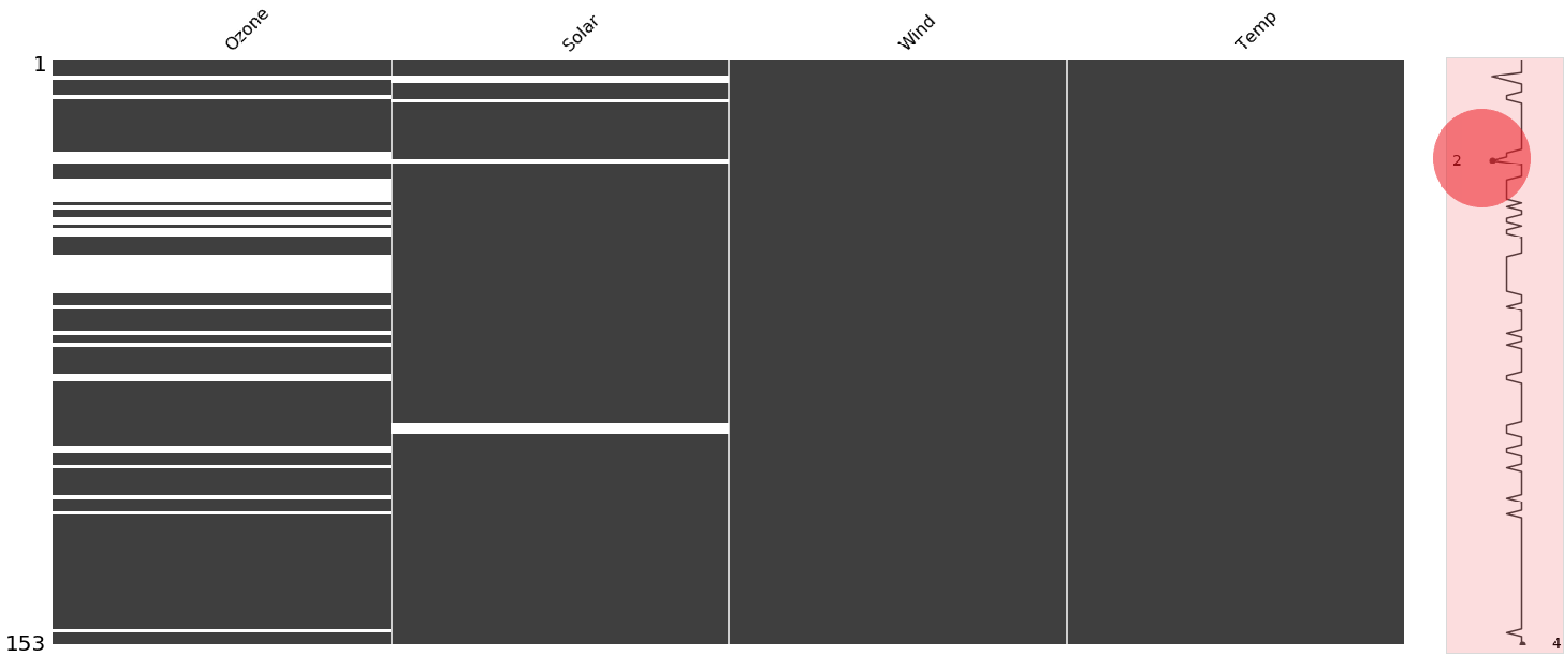
Nullity Matrix

```
msno.matrix(airquality)
```



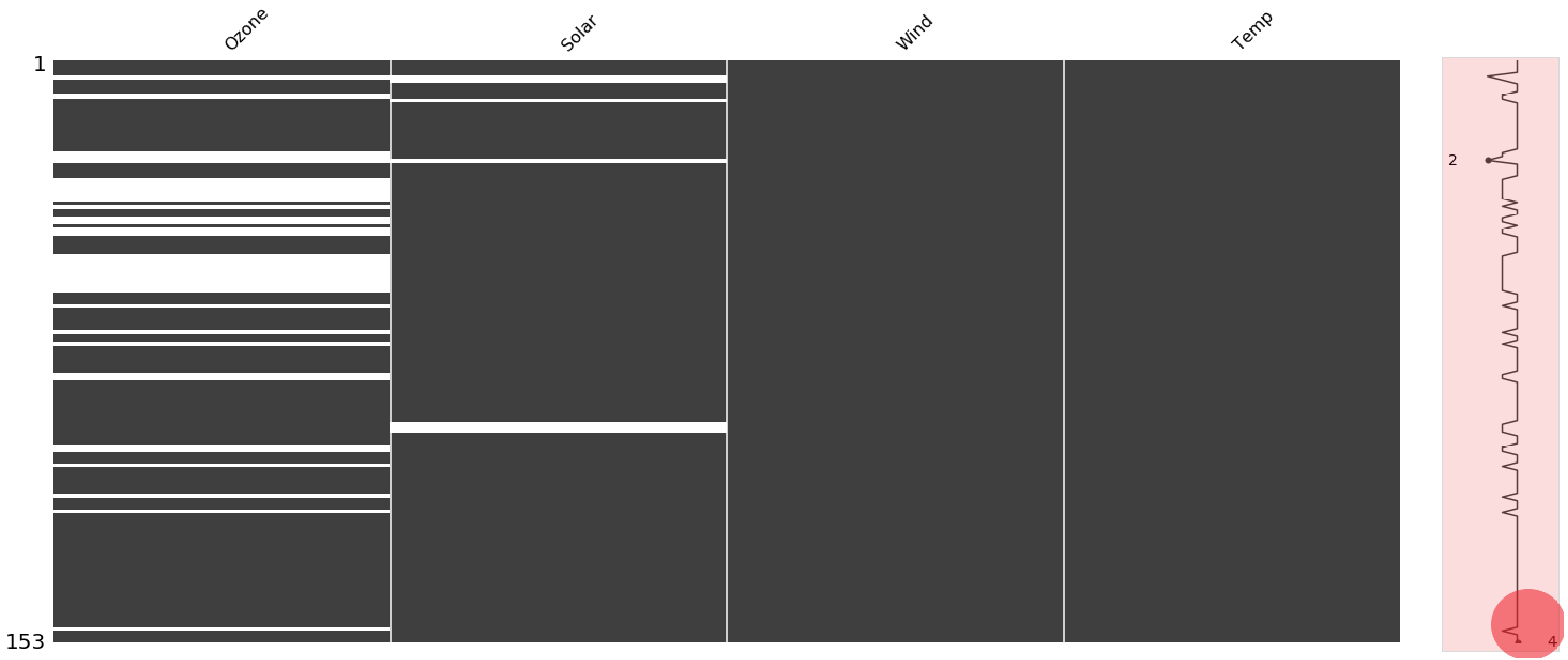
Nullity Matrix

```
msno.matrix(airquality)
```



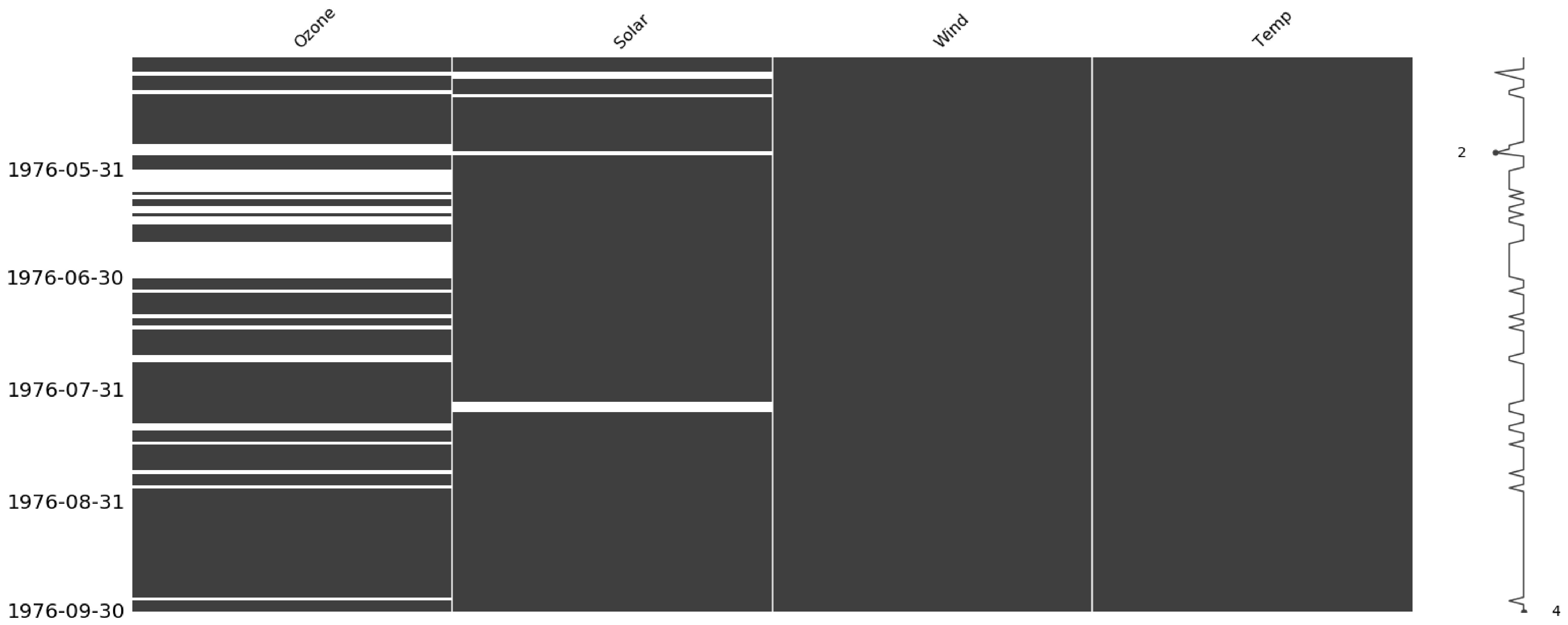
Nullity Matrix

```
msno.matrix(airquality)
```



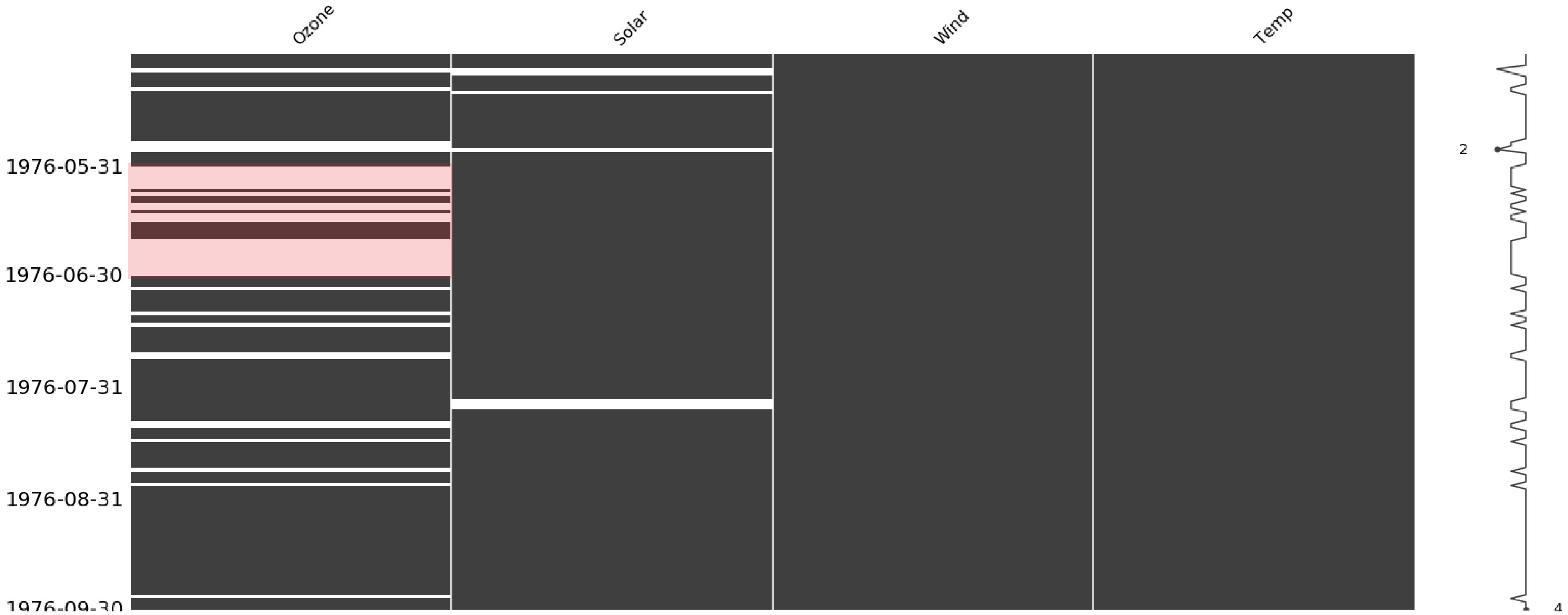
Nullity Matrix for time-series data

```
msno.matrix(airquality, freq='M')
```



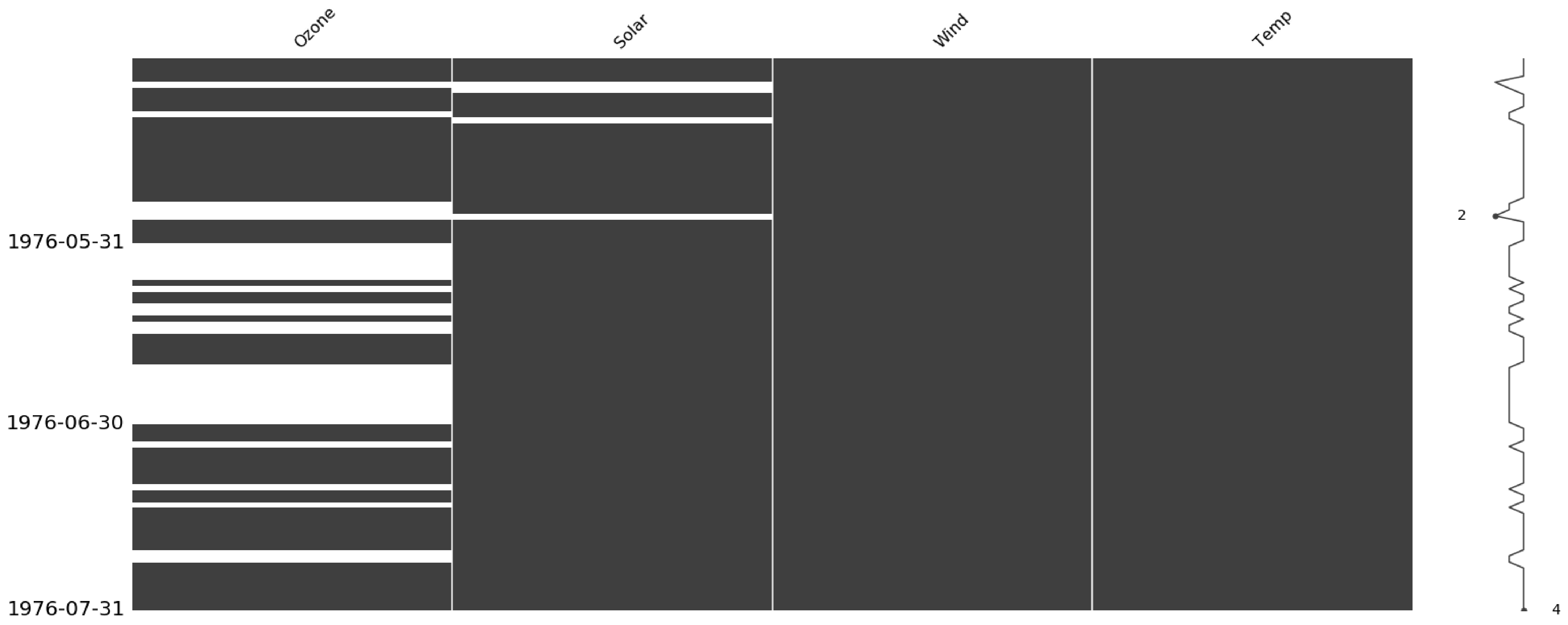
Nullity Matrix for time-series data

```
msno.matrix(airquality, freq='M')
```



Fine tuning the matrix

```
msno.matrix(airquality.loc['May-1976': 'Jul-1976'], freq='M')
```



Summary

In this lesson we learned to analyze

- the amount of missingness numerically
- the amount of missingness graphically
- the percentage of missingness
- the nullity matrix for regular datasets
- the nullity matrix for time-series datasets

Let's practice!

DEALING WITH MISSING DATA IN PYTHON