

Data Preprocessing

MATLAB

1. Outlier

1.1 Deteksi Outlier

Untuk mengatasi data yang hilang atau outliers pada data menggunakan *function* isoutlier.

```
A = [1 4 17 48 10 7 13 2 3];  
b = isoutlier(A)
```

Hasil :

```
Command Window  
>> hilang  
  
b =  
  
1×9 logical array  
  
0 0 0 1 0 0 0 0 0
```

Nilai 0 = bukan *outliers*, sedangkan nilai 1 = *outliers*.

1.2 Penanganan Outlier

Data cleaning dapat disebut juga dengan *data scrubing* proses ini akan memastikan data yang akan diproses benar dan akurat. Mendeteksi *outliers* dan perubahan yang mendadak akan membantu mengidentifikasi tren atau pola data yang signifikan. Pada MATLAB ada beberapa *function* yang dapat digunakan dalam proses *data cleaning*, berikut penjelasannya:

1. filloutlier

Function ini digunakan untuk mendeteksi *outliers* dan mengganti nilainya sesuai dengan metode yang dipilih.

Metode yang digunakan untuk menangani pada function filloutlier

Metode	Deskripsi
<i>Numeric scalar</i>	Untuk mengganti nilai <i>outliers</i> dengan nilai skalar yang spesifik.
Center	Untuk mengganti nilai <i>outliers</i> dengan nilai pusat.
Clip	Untuk mengganti nilai <i>outliers</i> dengan nilai ambang batas yang lebih rendah untuk elemen yang lebih kecil dan untuk elemen yang lebih besar digunakan nilai ambang atas.
previous	Untuk mengganti nilai <i>outliers</i> dengan nilai yang sama dengan data pada baris sebelumnya.
Next	Untuk mengganti nilai <i>outliers</i> dengan nilai yang sama dengan data pada baris selanjutnya.
Nearest	Untuk mengganti nilai <i>outliers</i> dengan nilai data yang terdekat.

Linear	Untuk mengganti nilai <i>outliers</i> dengan nilai interpolasi linear dari nilai-nilai data terdekat yang tidak hilang. Metode ini digunakan untuk tipe data <i>datetime</i> dan <i>duration</i> .
Spline	Untuk mengganti nilai <i>outliers</i> dengan nilai dari data-data berdekatan yang dihubungkan oleh satu polinom. Metode ini digunakan untuk tipe data <i>datetime</i> dan <i>duration</i> .
Pchip	Untuk mengganti nilai <i>outliers</i> dengan nilai hasil penjumlahan nilai baris sebelumnya dan nilai baris setelahnya lalu dibagi dua.

Metode yang digunakan untuk mendeteksi pada function filloutlier

Metode	Deskripsi
Median	Metode ini mendefinisikan <i>outliers</i> sebagai elemen yang berskala 3 kali lebih dari MAD dari <i>median</i> . Skala MAD didefinisikan sebagai $c * \text{median}(\text{abs}(A - \text{median}(A)))$ dimana $c = -1/(\text{sqrt}(2) * \text{erfcinv}(3/2))$
Mean	Metode ini mendefinisikan <i>outliers</i> sebagai elemen yang berskala 3 kali lebih dari standar deviasinya. Metode ini lebih cepat dari metode <i>median</i> tetapi kurang akurat.
Quartiles	Metode ini digunakan saat data tidak terdistribusi secara normal, dan mendefinisikan <i>outliers</i> sebagai elemen yang bernilai lebih dari 1,5 rentang interkuartil di kuartil atas (75%) atau di kuartil bawah (25%)
Grubbs	Pada metode ini data diasumsikan sebagai data yang berdistribusi normal. Metode grubbs mendeteksi <i>outliers</i> dan menghilangkan satu <i>outliers</i> setiap satu iterasi berdasarkan uji hipotesis.
Gesd	<i>Outliers</i> dideteksi dengan menggunakan uji penyimpangan <i>studentized</i> , metode ini mirip dengan metode grubbs tetapi bekerja lebih baik dari grubbs

Berikut *source code* contoh program penggunaan function filloutlier menggunakan beberapa metode:

```
A = [57 59 65 70 59 58 57 58 350 61 62 60 62 58 57];
C = std(A)
Outlier = 3*C
B = filloutliers(A, 'nearest', 'mean')
```

Hasil :

```
C =

    74.9055

Outlier =

    224.7166

B =

    57    59    65    70    59    58    57    58    61    61    62    60    62    58    57
```

2. rmoutlier

Function ini digunakan untuk mendeteksi dan menghapus *outliers*. *Function* ini mirip dengan *function* filloutlier, bedanya jika pada *function* filloutlier setelah *outliers* dideteksi akan diperbaiki tetapi pada *function* rmoutlier *outliers* akan dihapus. *Function* rmoutlier hanya *competitible* pada MATLAB 2018. *function* rmoutlier terdapat beberapa metode yang digunakan dan mendeteksi *outliers* pada data.

Metode yang digunakan untuk mendeteksi pada function rmoutlier

Metode	Deskripsi
Median	Metode ini mendefinisikan <i>outliers</i> sebagai elemen yang berskala 3 kali lebih dari MAD dari <i>median</i> . Skala MAD didefinisikan sebagai $c * \text{median}(\text{abs}(A - \text{median}(A)))$ dimana $c = -1/(\sqrt{2} * \text{erfcinv}(3/2))$
Mean	Metode ini mendefinisikan <i>outliers</i> sebagai elemen yang berskala 3 kali lebih dari standar deviasinya. Metode ini lebih cepat dari metode <i>median</i> tetapi kurang akurat.
Quartiles	Metode ini digunakan saat data tidak terdistribusi secara normal, dan mendefinisikan <i>outliers</i> sebagai elemen yang bernilai lebih dari 1,5 rentang interkuartil di kuartil atas (75%) atau di kuartil bawah (25%)
Grubbs	Pada metode ini data diasumsikan sebagai data yang berdistribusi normal. Metode grubbs mendeteksi <i>outliers</i> dan menghilangkan satu <i>outliers</i> setiap satu iterasi berdasarkan uji hipotesis.
Gesd	<i>Outliers</i> dideteksi dengan menggunakan uji penyimpangan <i>studentized</i> , metode ini mirip dengan metode grubbs tetapi bekerja lebih baik dari grubbs

Contoh source code :

```
A = [57 59 65 70 59 58 57 58 350 61 62 60 62 58 57];  
[M,N] = rmoutliers(A, 'mean')    % M dan K nilainya sama, yaitu data yang sudah -  
K = rmoutliers(A, 'mean')        % dihilangkan outliernya
```

Hasil :

```
M =  
  
    57    59    65    70    59    58    57    58    61    62    60    62    58    57  
  
N =  
  
1×15 logical array  
  
    0    0    0    0    0    0    0    0    1    0    0    0    0    0    0  
  
K =  
  
    57    59    65    70    59    58    57    58    61    62    60    62    58    57
```

Variabel M dan K berisi data yang telah dihilangkan *outliers*nya. Sedangkan pada variabel N digunakan untuk mendeteksi adanya *outliers* Nilai 0 = bukan *outliers*, sedangkan nilai 1 = *outliers*.

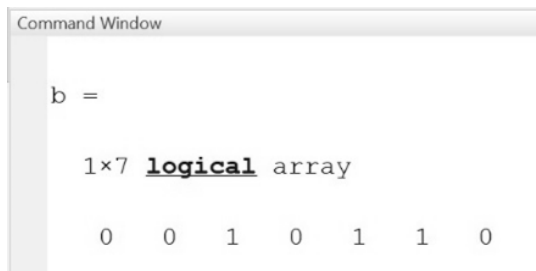
2. Data yang hilang

2.1 Deteksi data hilang

Data disebut data yang hilang jika, data yang seharusnya berisi data numerik tetapi bernilai karakter atau data kosong. Hal ini dapat disebabkan oleh beberapa faktor seperti pengisian data yang salah, data responden yang tidak lengkap, dan lain lain. Deteksi data yang hilang dengan MATLAB menggunakan ismissing. *Function* ini digunakan untuk mencari data yang hilang. Berikut source *code* contoh program penggunaan *function* ismissing:

```
A = [2 4 NaN 6 NaN NaN NaN 9];  
b = ismissing(A)
```

Hasil :



Command Window

```
b =  
  
1×7 logical array  
  
0    0    1    0    1    1    0
```

Nilai 0 = bukan data yang hilang, sedangkan nilai 1 = data hilang.

2.2 Penanganan Data yang Hilang

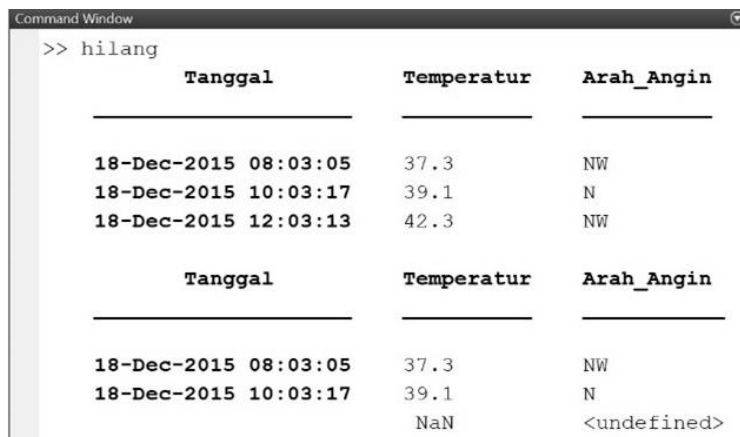
Pada beberapa kasus, data yang akan diolah seringkali terdapat data yang hilang. Data yang hilang dapat mengganggu metode analisis data. Oleh karena itu, data hilang harus ditangani dengan benar. Pada MATLAB ada beberapa *function* yang dapat digunakan dalam penanganan data yang hilang,

1. missing

Function ini digunakan untuk menghilangkan data. *Function* ini memungkinkan pengguna mengosongkan nilai pada data untuk mewakili data yang hilang. Nilai ini selanjutnya secara otomatis dikonversi ke nilai standar sesuai dengan tipe data yang asli. Berikut source *code* contoh program penggunaan *function* missing:

```
Tanggal = datetime({'2015-12-18 08:03:05'; '2015-12-18
10:03:17'; '2015-12-18 12:03:13'});
Temperatur = [37.3;39.1;42.3];
Arah_Angin = categorical({'NW'; 'NW'; 'N'});
TT = timetable(Tanggal, Temperatur, Arah_Angin);
disp(TT)
TT.Tanggal(3) = missing;
TT.Temperatur(3) = missing;
TT.Arah_Angin(3) = missing;
disp(TT)
```

Hasil :



```
>> hilang
```

Tanggal	Temperatur	Arah_Angin
18-Dec-2015 08:03:05	37.3	NW
18-Dec-2015 10:03:17	39.1	N
18-Dec-2015 12:03:13	42.3	NW

Tanggal	Temperatur	Arah_Angin
18-Dec-2015 08:03:05	37.3	NW
18-Dec-2015 10:03:17	39.1	N
	NaN	<undefined>

Contoh program diatas menunjukkan sebuah *time table* dengan data berisi tanggal, temperatur, dan arah angin. Pada tabel kedua dapat dilihat bahwa baris ketiga telah dikosongkan dan tipe data sesuai dengan tipe data tabel pertama.

2. fillmissing

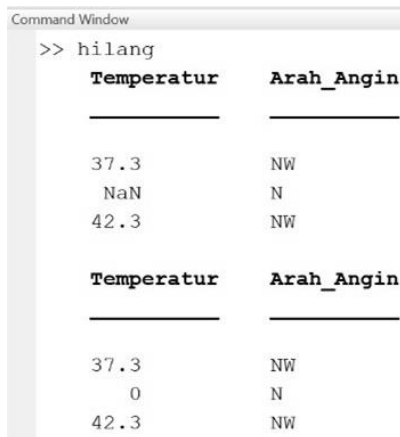
Function ini digunakan untuk mengisi data yang hilang. *Function* ini memungkinkan pengguna mengisi sendiri nilai data yang hilang. Nilai ini selanjutnya secara otomatis dikonversi ke nilai standar sesuai dengan tipe data yang asli. Berikut nilai-nilai tipe data pada data:

- NaN, digunakan untuk mendefinisikan tipe data single, double, duration, dan calenderDuration
- NaT, digunakan untuk mendefinisikan tipe data datetime
- <missing>, digunakan untuk mendefinisikan tipe data string
- <undefined>, digunakan untuk mendefinisikan tipe data categorical
- '', digunakan untuk mendefinisikan tipe data char
- {''}, digunakan untuk mendefinisikan tipe data cell array

Berikut source *code* contoh program penggunaan *function* missing:

```
Temperatur = [37.3;NaN;42.3];  
Arah_Angin = categorical({'NW';'N';'NW'});  
TT = table(Temperatur,Arah_Angin);  
disp(TT)  
F = fillmissing(TT,'constant',0,'DataVariables',@isnumeric);  
disp(F)
```

Hasil :



```
>> hilang  
  
    Temperatur    Arah_Angin  
    _____    _____  
  
    37.3          NW  
    NaN          N  
    42.3          NW  
  
    Temperatur    Arah_Angin  
    _____    _____  
  
    37.3          NW  
    0            N  
    42.3          NW
```

```
F = fillmissing(TT,'constant',0,'DataVariables',@isnumeric);
```

Mengisi nilai data kosong dengan nilai konstan 0 dan sebagai alternatif, digunakan fungsi `@isnumeric` untuk mendefinisikan tipe data numerik.

Dalam pengisian nilai pada data kosong terdapat beberapa metode yang digunakan menangani data hilang.

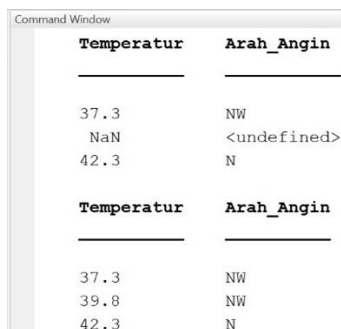
Metode yang digunakan pada function `fillmissing`

Metode	Deskripsi
previous	Untuk mengisi data yang hilang dengan nilai yang sama dengan data pada baris sebelumnya.
Next	Untuk mengisi data yang hilang dengan nilai yang sama dengan data pada baris selanjutnya.
Nearest	Untuk mengisi data yang hilang dengan nilai data yang terdekat.
Linear	Untuk mengisi data yang hilang dengan nilai hasil penjumlahan nilai baris sebelumnya dan nilai baris setelahnya lalu dibagi dua.
Spline	Untuk mengisi data yang hilang dengan nilai dari data-data berdekatan yang dihubungkan oleh satu polinom. Metode ini digunakan untuk tipe data <i>datetime</i> dan <i>duration</i> .
Pchip	Untuk mengisi data yang hilang dengan tipe data numerik, durasi dan <i>datetime</i> .

Berikut source *code* contoh program penggunaan metode *function* fillmissing:

```
Temperatur = [37.3;NaN;42.3];
Arah_Angin = categorical({'NW';'';'N'});
TT = table(Temperatur,Arah_Angin);
disp(TT)
F =
fillmissing(TT,'previous','DataVariables',{'Arah_Angin'})
G = fillmissing(F,'pchip','DataVariables',{'Temperatur'});
disp(G)
```

Hasil :



Temperatur	Arah_Angin
37.3	NW
NaN	<undefined>
42.3	N

Temperatur	Arah_Angin
37.3	NW
39.8	NW
42.3	N

'DataVariable' pada contoh diatas digunakan untuk mengisi data yang hilang pada variabel tertentu.

3. Contoh Source Code

```
Data = readtable('Book1.xlsx'); % Membaca Data
%Deteksi Outlier
Outlier = isoutlier(Data); % Deteksi Outlier
% Penanganan Outlier
B = filloutliers(Data,0); % Mereplace outlier dengan '0'
L = filloutliers(Data,'nearest','DataVariables',{'X2'});% Mereplace dengan data terdekat
K = rmoutliers(Data); % Menghilangkan data outlier

% Deteksi Data Missing
Missing1 = ismissing(B); % Mendeteksi data missing dari variabel B
Missing2 = ismissing(K); % Mendeteksi data missing dari variabel K
% penanganan Data Missing
% Mereplace data missing dengan nilai '0' pada variabel B
X = fillmissing(B,'constant',0,'DataVariables',@isnumeric);
% Mereplace data missing dengan nilai '0' pada variabel K
Y = fillmissing(K,'constant',0,'DataVariables',@isnumeric);
%Mereplace data missing dengan nilai sebelumnya pada variabel B
Z = fillmissing(B,'previous','DataVariables',{'X3'});
% Transformasi Data
Normalisasi = normalize(X,'zscore');% Normalisasi pada data yang sudah OK(data X)
```

Hasil :

Data Awal

Data =

3×3 [table](#)

x1	x2	x3
1	2	3
4	150	6
7	8	NaN

1. Deteksi Outlier

Outlier =

3×3 [logical](#) array

0	0	0
0	1	0
0	0	0

2. Penanganan Outlier

Menggantikan dengan nilai 0

B =

3×3 [table](#)

x1	x2	x3
1	2	3
4	0	6
7	8	NaN

Menggantikan dengan nilai sebelumnya

L =

3×3 [table](#)

x1	x2	x3
1	2	3
4	8	6
7	8	NaN

Menghapus Outlier

K =

2×3 [table](#)

x1	x2	x3
—	—	—
1	2	3
7	8	NaN

Deteksi Data Missing

Missing1 =

3×3 [logical](#) array

0	0	0
0	0	0
0	0	1

Missing2 =

2×3 [logical](#) array

0	0	0
0	0	1

Penanganan Data Missing

Mengisi dengan nilai 0

X =

3×3 [table](#)

x1	x2	x3
—	—	—
1	2	3
4	0	6
7	8	0

Y =

2×3 [table](#)

x1	x2	x3
—	—	—
1	2	3
7	8	0

Mengisi dengan nilai terdekat

Z =

3×3 [table](#)

x1	x2	x3
—	—	—
1	2	3
4	0	6
7	8	6

4. Normalisasi

Normalisasi dilakukan dengan metode z-score

Normalisasi =

3×3 [table](#)

x1	x2	x3
—	—	—
-1	-0.32026	0
0	-0.80064	1
1	1.1209	-1

Phyton

1. Outlier

```
dataset= [10,12,12,13,12,11,14,13,15,10,10,10,100,12,14,13, 12,10,10,11,12,15,12,13,12,11,14,13,15,10,15,12,10,14,13,15,10]
```

```
import numpy as np
import pandas as pd
outliers=[]
def detect_outlier(data_1):

    threshold=3
    mean_1 = np.mean(data_1)
    std_1 =np.std(data_1)

    for y in data_1:
        z_score= (y - mean_1)/std_1
        if np.abs(z_score) > threshold:
            outliers.append(y)
    return outliers
```

```
outlier_datapoints = detect_outlier(dataset)
print(outlier_datapoints)
```

Hasil :

100

Data (df)

	Column_1	Column_2
0	1	1
1	1	1
2	1	1
3	1	1
4	1	10
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1
10	10	1

```

z_scores = stats.zscore(df)

abs_z_scores = np.abs(z_scores)
filtered_entries = (abs_z_scores < 3).all(axis=1)
new_df = df[filtered_entries]

print(new_df)

```

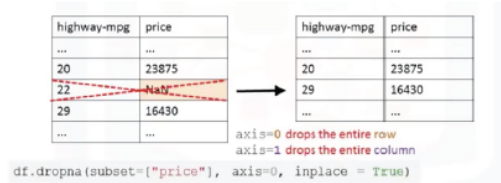
Output :

	Column_1	Column_2
0	1	1
1	1	1
2	1	1
3	1	1
5	1	1
6	1	1
7	1	1
8	1	1
9	1	1

2. Missing Value

Drop Missing Values in Python

- To remove data that contains missing values, Pandas library has a built-in method called **'dropna'**.
- Essentially, with the dropna method, you can choose to drop rows or columns that contain missing values, like NaN.
- So you'll need to specify "axis=0" to drop the rows, or "axis=1" to drop the columns that contain the missing values. "Inplace=True" just writes the result back into the dataframe.



Replace Missing Values in Python

To replace missing values like “NaN” with actual values, pandas library has a built in method called ‘replace’, which can be used to fill in the missing values with the newly calculated values.

```
dataframe.replace(missing_value, new_value)
```

Replace Missing Values in Python

- As an example, assume that we want to replace the missing values of the variable ‘normalized-losses’ by the mean value of the variable. Therefore, the missing value should be replaced by the average of the entries within that column.
- In Python, first we calculate the mean of the column.
- Then we use the method “replace”, to specify the value we would like to be replaced as the first parameter, in this case, NaN.
- The second parameter is the value we would like to replace it with: i.e., the mean, in this example.

normalized-losses	make
...	...
164	audi
164	audi
NaN	audi
158	audi
...	...

normalized-losses	make
...	...
164	audi
164	audi
162	audi
158	audi
...	...

```
mean = df["normalized-losses"].mean()  
df["normalized-losses"].replace(np.nan, mean)
```

2.1 Deteksi Missing Value

```
df.isna().sum()
```

```
PassengerId      0  
Survived          0  
Pclass           0  
Name             0  
Sex              0  
Age             177  
SibSp            0  
Parch            0  
Ticket           0  
Fare             0  
Cabin           687  
Embarked         2  
dtype: int64
```

Dengan bantuan fungsi `isna()` dan `sum()` kita tahu bahwa dalam dataset semua kolom tidak ada nilai yang kosong kecuali kolom Age dengan 177 missing value, Kolom Cabin 687 dan kolom Embarked 2.

2.2 Penanganan Missing Value

Mengganti missing value dengan nilai rata2

```
1 # Langkah 1
2 df_age = df
3 # Langkah 2
4 rata_umur = df_age['Age'].mean()
5 # Langkah 3
6 df_age['Age'] = df_age['Age'].fillna(rata_umur)
7 # Langkah 4
8 df_age['Age'].isna().sum()
```

Menghapus missing value

```
1 # Langkah 1
2 df_cabin = df
3 # Langkah 2
4 df_cabin.dropna()
```

Latihan :

Coba semua source code diatas

Tugas :

1. Carilah data bebas
2. Lakukan PreProcessing dengan menggunakan Matlab dan Phyton.
3. Buatlah laporan yang terdiri dari print screen hasil dan penjelasannya
4. Berilah nama “ laporan Data Preprocessing_kelompok XXX.pdf

Ketentuan :

1. Tugas dikerjakan secara kelompok
2. Satu kelompok terdiri dari 3-4 mhs
3. Tugas terdiri dari Laporan, file matlab, dan file phyton, dan data aslinya dan dikumpulkan dengan nama “Tugas Data PreProcessing_Kelompok XXX.rar
4. Tugas dikumpulkan paling lambat hari Sabtu / 18 Februari 2023 pukul 23.59 Wib