

Data

What is Data?

- Collection of data objects and their attributes
- An attribute is a property or characteristic of an object
 - Examples: eye color of a person, temperature, etc.
 - Attribute is also known as variable, field, characteristic, or feature
- A collection of attributes describe an object
 - Object is also known as record, point, case, sample, entity, or instance

Attributes



| <i>Tid</i> | Refund | Marital Status | Taxable Income | Cheat |
|------------|--------|----------------|----------------|-------|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

Objects

Attribute Values

- Attribute values are numbers or symbols assigned to an attribute
- Distinction between attributes and attribute values
 - Same attribute can be mapped to different attribute values
 - Example: height can be measured in feet or meters
 - Different attributes can be mapped to the same set of values
 - Example: Attribute values for ID and age are integers
 - But properties of attribute values can be different
 - ID may have no limit but age may have a maximum and minimum value

Types of Attributes

- There are different types of attributes
 - Categorical (Qualitative):
 - Nominal
 - Ordinal
 - Numeric (Quantitative):
 - Interval
 - Ratio

Properties of Attribute Values / 1

- The type of an attribute depends on which of the following properties it possesses:

- Distinctness: $=$ and \neq
- Order: $<$, \leq , $>$, and \geq
- Addition: $+$ and $-$
- Multiplication: $*$ and $/$

- Nominal attribute: distinctness
- Ordinal attribute: distinctness & order
- Interval attribute: distinctness, order & addition
- Ratio attribute: all 4 properties

Properties of Attribute Values / 2

| Attribute Type | Description | Examples | Operations |
|-----------------|---|---|--|
| Nominal | The values of a nominal attribute are just different names, i.e., nominal attributes provide only enough information to distinguish one object from another. ($=$, \neq) | zip codes, employee ID numbers, eye color, sex: { <i>male</i> , <i>female</i> } | mode, entropy, contingency correlation, χ^2 test |
| Ordinal | The values of an ordinal attribute provide enough information to order objects. ($<$, $>$) | hardness of minerals, { <i>good</i> , <i>better</i> , <i>best</i> }, grades, street numbers | median, percentiles, rank correlation, run tests, sign tests |
| Interval | For interval attributes, the differences between values are meaningful, i.e., a unit of measurement exists. ($+$, $-$) | calendar dates, temperature in Celsius or Fahrenheit | mean, standard deviation, Pearson's correlation, t and F tests |
| Ratio | For ratio variables, both differences and ratios are meaningful. ($*$, $/$) | temperature in Kelvin, monetary quantities, counts, age, mass, length, electrical current | geometric mean, harmonic mean, percent variation |

Discrete and Continuous Attributes

● Discrete Attribute

- Has only a finite or countably infinite set of values
- Examples: zip codes, counts, or the set of words in a collection of documents
- Often represented as integer variables.
- Note: binary attributes are a special case of discrete attributes

● Continuous Attribute

- Has real numbers as attribute values
- Examples: temperature, height, or weight.
- Practically, real values can only be measured and represented using a finite number of digits.
- Continuous attributes are typically represented as floating-point variables.

Record Data

- Data that consists of a collection of records, each of which consists of a fixed set of attributes

| <i>Tid</i> | Refund | Marital Status | Taxable Income | Cheat |
|------------|--------|----------------|----------------|-------|
| 1 | Yes | Single | 125K | No |
| 2 | No | Married | 100K | No |
| 3 | No | Single | 70K | No |
| 4 | Yes | Married | 120K | No |
| 5 | No | Divorced | 95K | Yes |
| 6 | No | Married | 60K | No |
| 7 | Yes | Divorced | 220K | No |
| 8 | No | Single | 85K | Yes |
| 9 | No | Married | 75K | No |
| 10 | No | Single | 90K | Yes |

Data Matrix

- If data objects have the same fixed set of numeric attributes, then the data objects can be thought of as points in a multi-dimensional space, where each dimension represents a distinct attribute
- Such data set can be represented by an m by n matrix, where there are m rows, one for each object, and n columns, one for each attribute

| Projection of x Load | Projection of y load | Distance | Load | Thickness |
|-------------------------|-------------------------|----------|------|-----------|
| 10.23 | 5.27 | 15.22 | 2.7 | 1.2 |
| 12.65 | 6.25 | 16.22 | 2.2 | 1.1 |

Document Data

- Each document becomes a 'term' vector,
 - each term is a component (attribute) of the vector,
 - the value of each component is the number of times the corresponding term occurs in the document.

| | team | coach | player | ball | score | game | win | lost | timeout | season |
|------------|------|-------|--------|------|-------|------|-----|------|---------|--------|
| Document 1 | 3 | 0 | 5 | 0 | 2 | 6 | 0 | 2 | 0 | 2 |
| Document 2 | 0 | 7 | 0 | 2 | 1 | 0 | 0 | 3 | 0 | 0 |
| Document 3 | 0 | 1 | 0 | 0 | 1 | 2 | 2 | 0 | 3 | 0 |

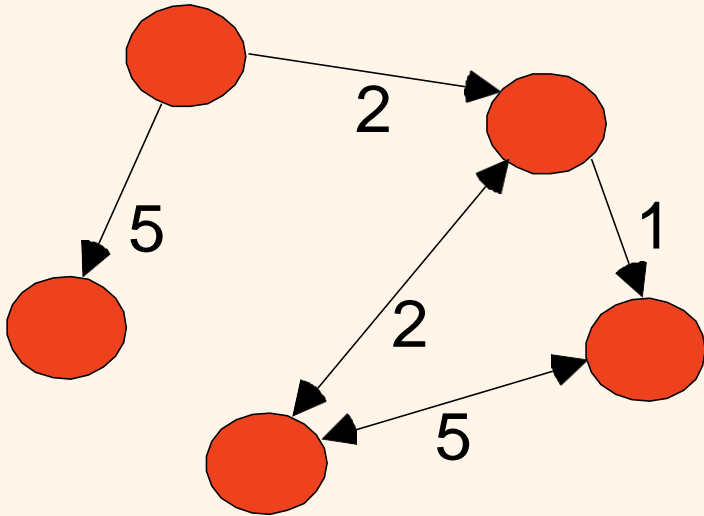
Transaction or Market Basket Data

- A special type of record data, where
 - each record (transaction) involves a set of items.
 - For example, consider a grocery store. The set of products purchased by a customer during one shopping trip constitute a transaction, while the individual products that were purchased are the items.

| <i>TID</i> | <i>Items</i> |
|------------|---------------------------|
| 1 | Bread, Coke, Milk |
| 2 | Beer, Bread |
| 3 | Beer, Coke, Diaper, Milk |
| 4 | Beer, Bread, Diaper, Milk |
| 5 | Coke, Diaper, Milk |

Graph Data

● Examples: Generic graph and HTML Links



```
<a href="papers/papers.html#bbbb">
```

```
Data Mining </a>
```

```
<li>
```

```
<a href="papers/papers.html#aaaa">
```

```
Graph Partitioning </a>
```

```
<li>
```

```
<a href="papers/papers.html#aaaa">
```

```
Parallel Solution of Sparse Linear System of Equations </a>
```

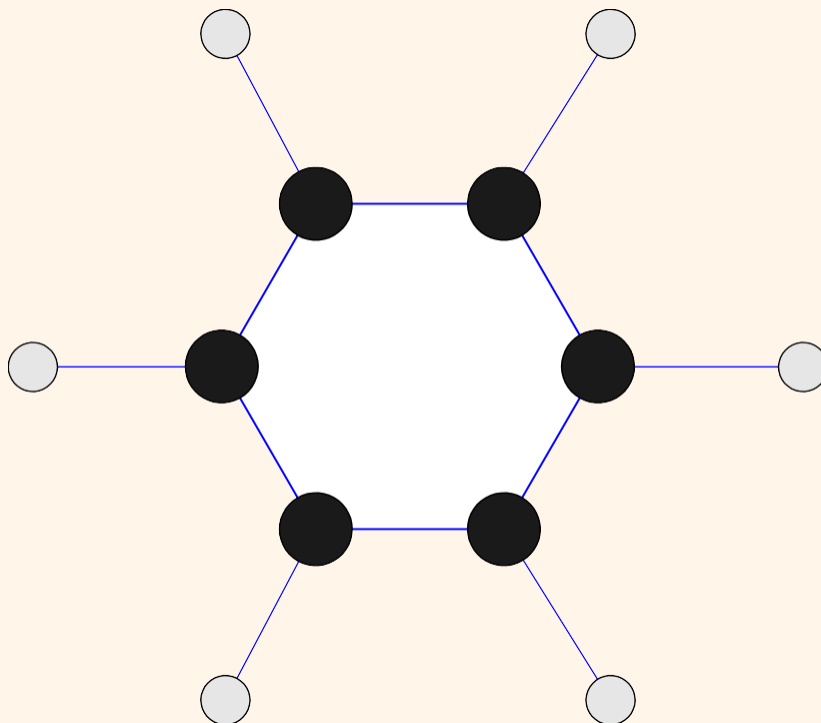
```
<li>
```

```
<a href="papers/papers.html#ffff">
```

```
N-Body Computation and Dense Linear System Solvers
```

Chemical Data

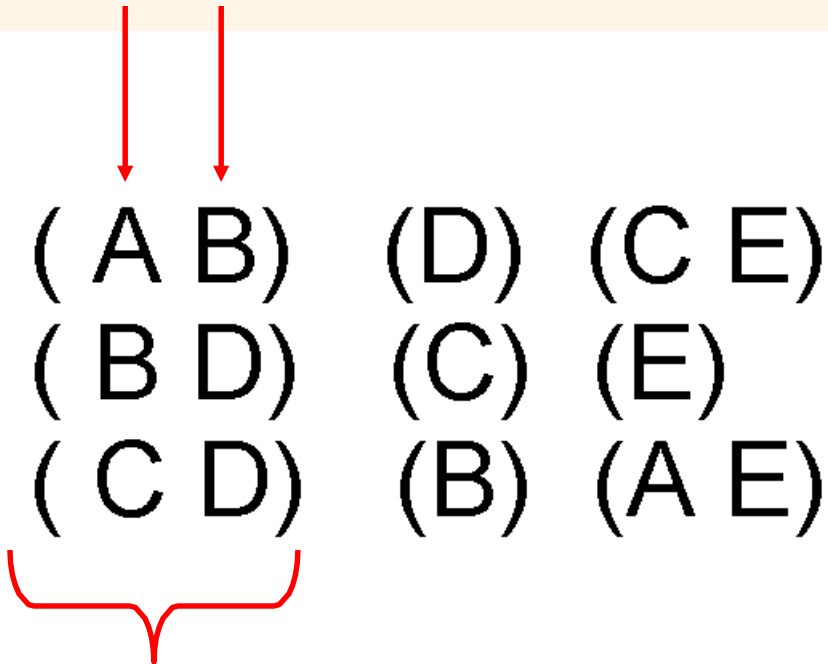
- Benzene Molecule: C_6H_6



Ordered Data / 1

- Sequences of transactions

Items/Events



| | | |
|---------|-------|---------|
| (A B) | (D) | (C E) |
| (B D) | (C) | (E) |
| (C D) | (B) | (A E) |

An element of
the sequence

Ordered Data / 2

- Genomic sequence data

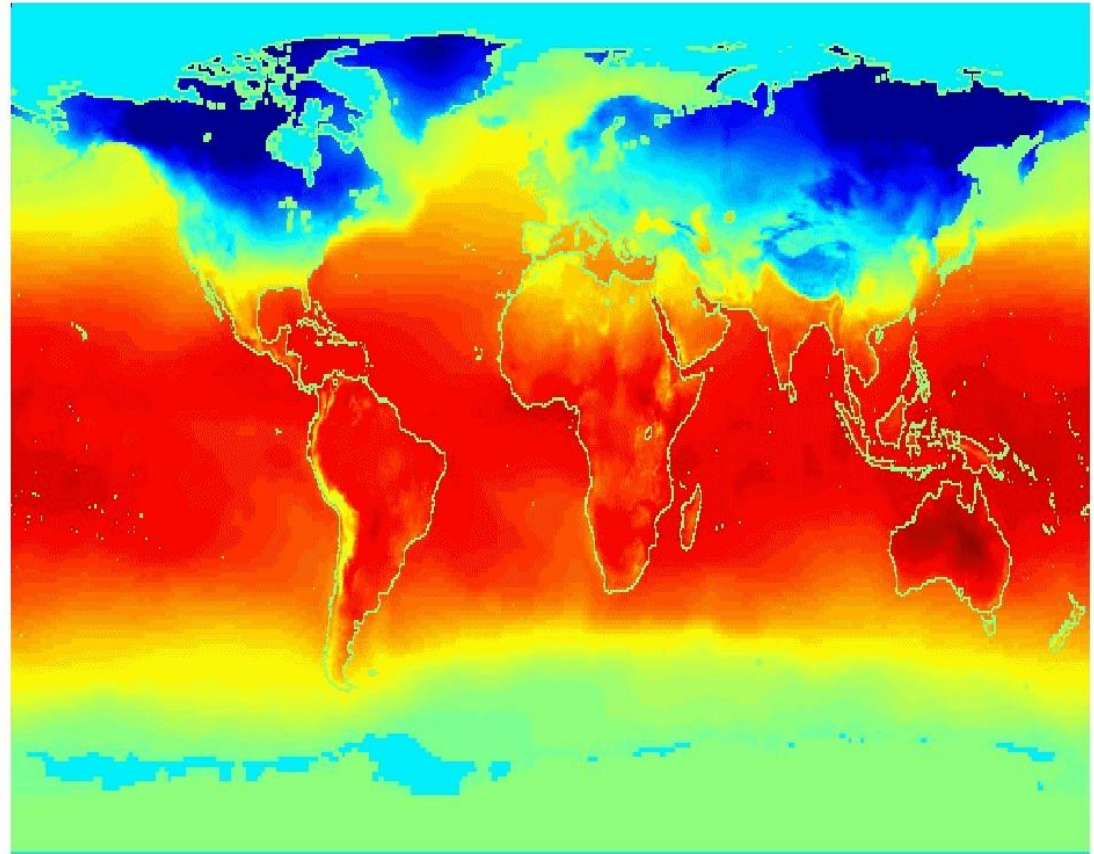
GGTTCCGCCTTCAGCCCCGCGCC
CGCAGGGCCCGCCCCGCGCCGTC
GAGAAGGGCCCGCCTGGCGGGCG
GGGGGAGGCGGGGCCGCCCGAGC
CCAACCGAGTCCGACCAGGTGCC
CCCTCTGCTCGGCCTAGACCTGA
GCTCATTAGGCGGCAGCGGACAG
GCCAAGTAGAACACGCGAAGCGC
TGGGCTGCCTGCTGCGACCAGGG

Ordered Data / 3

- Spatio-Temporal Data

Jan

**Average Monthly
Temperature of
land and ocean**



Terima Kasih

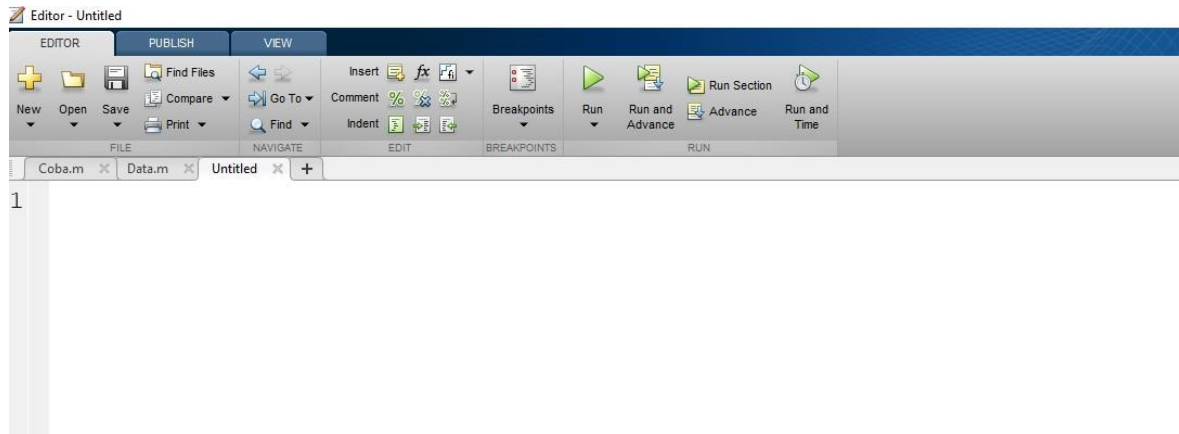
Create Data

1. Matlab

1.1 Membuat File editor (.m file)

Langkah awal membuat file editor (.m file)

1. Pilih New Script



2. Ketik source code di file editor

3. Setelah source code sudah benar, klik tombol Run (Segitiga berwarna Hijau) pada Toolbar di file editor.

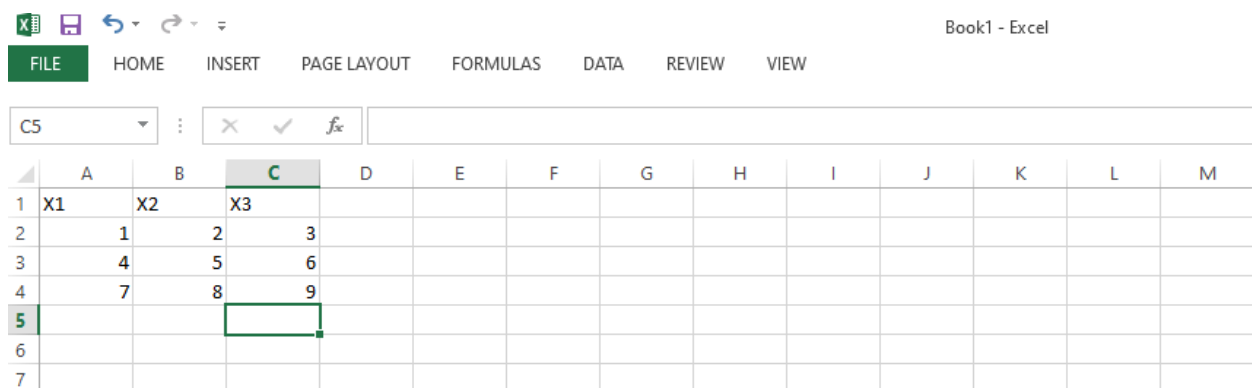
4. Hasil akan muncul

1.2 Membaca data dari Excel

Langkah – Langkah

1. Buatlah data di excel dengan namakan Book 1

Data = 'Book 1'.xls



2. Syntax yang digunakan di matlab untuk membaca data adalah

A. xlsread

xlsread digunakan untuk membaca semua data yang ada di excelnya kecuali variabelnya

```
nama_variabel = xlsread('nama_file')
```

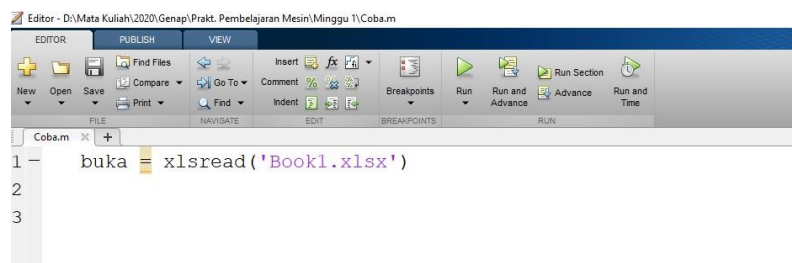
B. readtable

readtable digunakan untuk membaca semua file yang ada di excel beserta variabelnya. Disamping itu juga readtable dapat digunakan untuk file CSV.

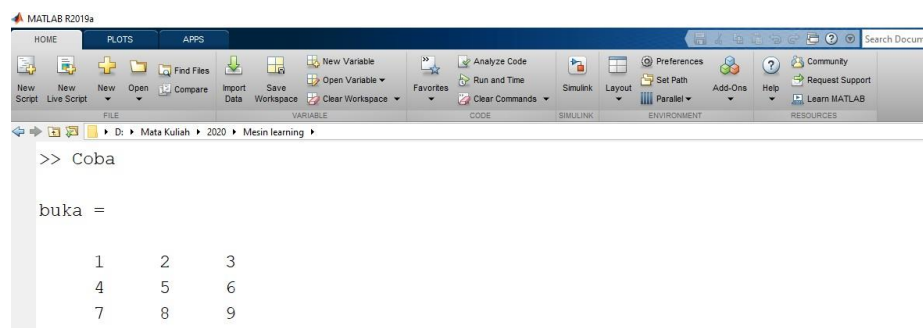
```
nama_variabel = readtable('nama_file')
```

Contoh

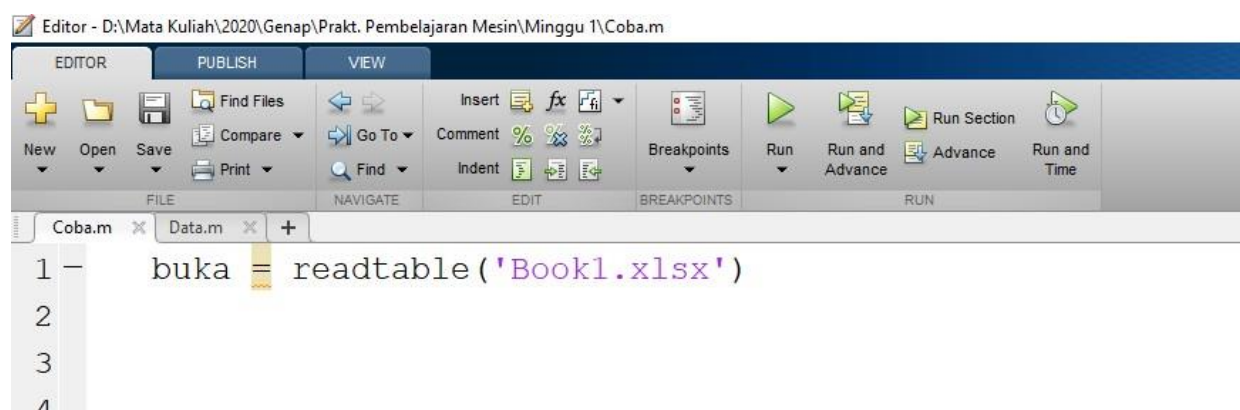
A. Menggunakan xlsread



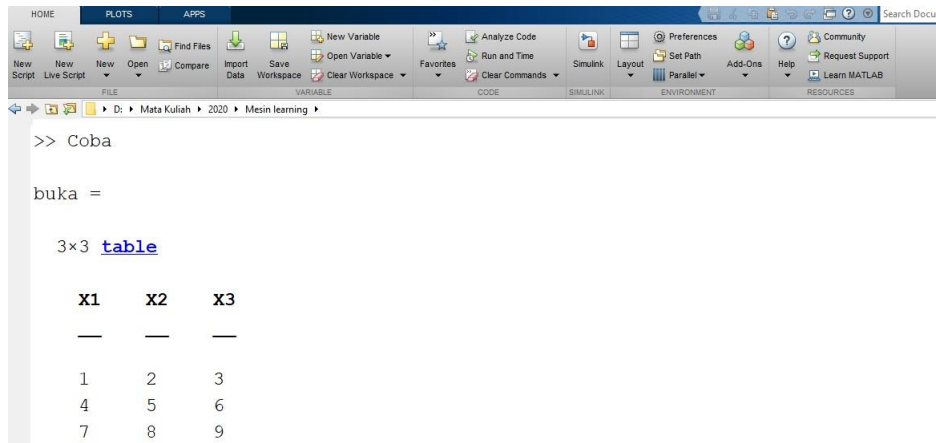
Hasil :



B. Menggunakan readtable



Hasil :



```
>> Coba

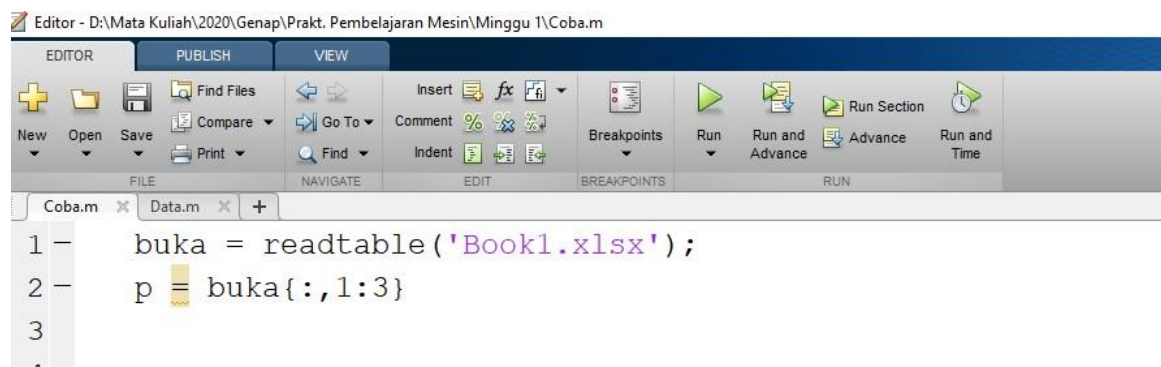
buka =

3x3 table

    X1    X2    X3
    ---    ---    ---
    1     2     3
    4     5     6
    7     8     9
```

1.2. Mengambil Nilai Variabel

1. Mengambil Nilai dari 3 Variabel, yaitu : X1, X2, dan X3

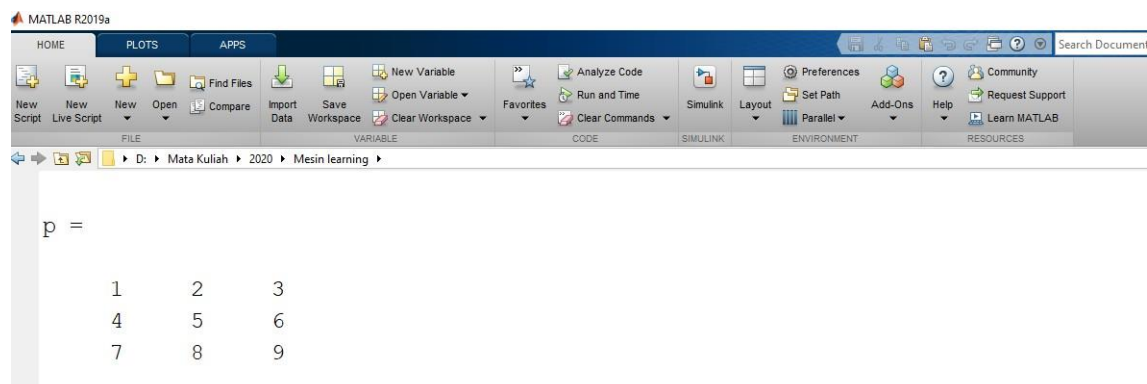


```
Editor - D:\Mata Kuliah\2020\Genap\Prakt. Pembelajaran Mesin\Minggu 1\Coba.m

EDITOR PUBLISH VIEW
+ New Open Save Find Files Compare Print Go To Comment Indent Breakpoints Run Run and Advance Run Section Run and Time
FILE NAVIGATE EDIT BREAKPOINTS RUN

Coba.m x Data.m x +
1 - buka = readtable('Book1.xlsx');
2 - p = buka{:,1:3}
3
4
```

Hasil :



```
MATLAB R2019a

HOME PLOTS APPS
+ New Live Script New Open Find Files Import Save New Variable Open Variable Favorites Analyze Code Run and Time Simulink Layout Set Path Preferences Add-Ons Help Community
FILE VARIABLE CODE SIMULINK ENVIRONMENT RESOURCES

D:\Mata Kuliah\2020\Mesin learning

p =

    1     2     3
    4     5     6
    7     8     9
```

2. Mengambil Nilai dari salah satu variabel

Editor - D:\Mata Kuliah\2020\Genap\Prakt. Pembelajaran Mesin\Minggu 1\Coba.m

```
EDITOR PUBLISH VIEW
+ Find Files Insert fx Breakpoints Run Run and Advance Run and Time
New Open Save Compare Go To Comment % Indent Find Run Run and Advance Run and Time
FILE NAVIGATE EDIT BREAKPOINTS RUN

Coba.m Data.m +
1 % Membaca Data
2 buka = readtable('Book1.xlsx');
3 % Mengambil nilai dari 3 variabel, yaitu X1,X2, dan X3
4 p = buka(:,1:3);
5 % Mengambil nilai dari variabel ke 2, yaitu X2
6 q = buka(:,2);
7 % Mengambil nilai dari variabel ke 1, yaitu X1
8 m = buka(:,1);
9 % Mengambil nilai dari variabel ke 3, yaitu X3
10 n = buka(:,3);
11
```

MATLAB R2019a

HOME PLOTS APPS Search Documentation

New Script New Live Script New Open Find Files Import Data Save Workspace New Variable Open Variable Favorites Analyze Code Run and Time Simulink Layout Set Path Add-Ons Help

FILE VARIABLE CODE SIMULINK ENVIRONMENT

D: \ Mata Kuliah \ 2020 \ Mesin learning \

```
>> Coba
q =

     2
     5
     8

m =

     1
     4
     7
```

fx

MATLAB R2019a

HOME PLOTS APPS Search Documentation

New Script New Live Script New Open Find Files Import Data Save Workspace New Variable Open Variable Favorites Analyze Code Run and Time Simulink Layout Set Path Add-Ons Help

FILE VARIABLE CODE SIMULINK ENVIRONMENT

D: \ Mata Kuliah \ 2020 \ Mesin learning \

```
>> Coba
n =

     3
     6
     9
```

PHYTON

1. Mengimpor function yang ada di library

```
import pandas as pd
```

2. Pembacaan data

```
data = pd.read_csv("Data_Kung_People.csv")  
df = pd.DataFrame(data)  
print(df)
```

Hasil :

| | People | Height | Weight | Age | Male |
|------------------------|--------|--------|--------|-----|------|
| 0 | 1 | 151 | 47 | 63 | 1 |
| 1 | 2 | 139 | 36 | 63 | 0 |
| 2 | 3 | 136 | 31 | 65 | 0 |
| 3 | 4 | 156 | 53 | 41 | 1 |
| 4 | 5 | 145 | 41 | 51 | 0 |
| .. | ... | ... | ... | ... | ... |
| 101 | 102 | 152 | 51 | 34 | 0 |
| 102 | 103 | 160 | 47 | 44 | 1 |
| 103 | 104 | 149 | 40 | 43 | 0 |
| 104 | 105 | 142 | 32 | 73 | 0 |
| 105 | 106 | 167 | 57 | 38 | 1 |
| [106 rows x 5 columns] | | | | | |

3. Mengambil variabel

```
height = df['Height']  
print(height)
```

Hasil :

| | Height |
|-----|--------|
| 0 | 151 |
| 1 | 139 |
| 2 | 136 |
| 3 | 156 |
| 4 | 145 |
| .. | ... |
| 101 | 152 |
| 102 | 160 |
| 103 | 149 |
| 104 | 142 |
| 105 | 167 |

Data Preprocessing

A. Outlier dan Missing Value

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 $c = -1/(\sqrt{2}) * \text{erfcinv}(3/2)$ |
| 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 =  
1x15 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 =  
1x7 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

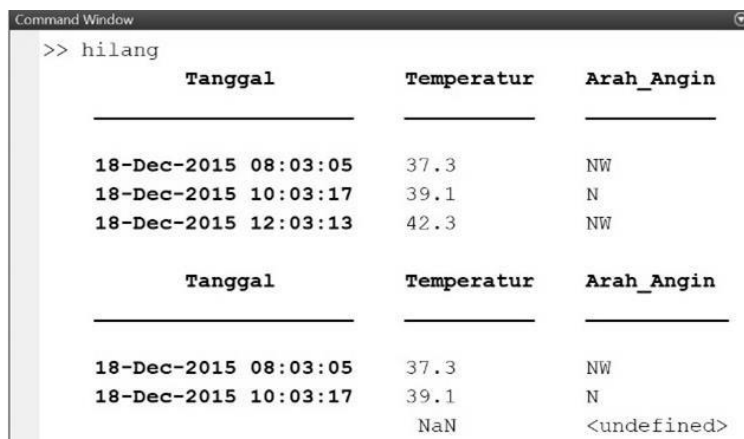
Pada beberapa kasus, data yang akan diolah seringkali terdapat data yang hilang. Data yang hilang dapat mengganggu metode analisis data. Oleh karena itu, data hilang harus ditangani dengan benar. Pada MATLAB ada beberapa function yang dapat digunakan dalam penanganan data yang hilang,

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 :



| 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

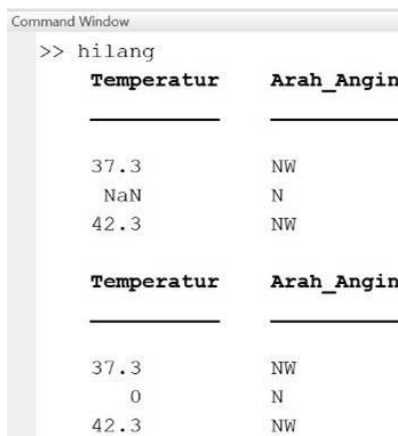
Function ini digunakan untuk mengisi data yang hilang. *Function* ini memungkinkan pengguna mengisi sendiri nilai data yang hilang. Nilai ini selanjutnya secara otomatis dikonversi ke nilai standar sesuai dengan tipe data yang asli. Berikut nilai-nilai tipe data pada data:

- 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.

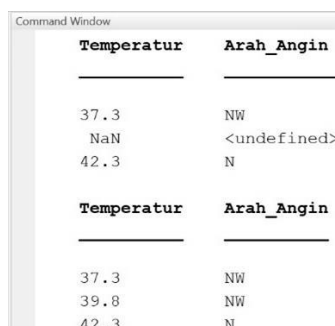
Metode yang digunakan pada function `fillmissing`

| 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 <code>datetime</code> dan <code>duration</code> . |
| Pchip | Untuk mengisi data yang hilang dengan tipe data numerik, durasi dan <code>datetime</code> . |

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)

```

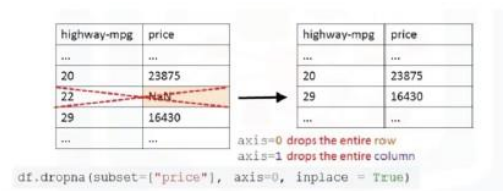
Output :

| | 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.



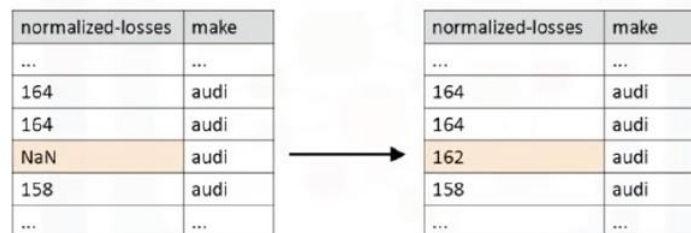
Replace Missing Values in Python

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.



The diagram illustrates the process of replacing a missing value (NaN) in a DataFrame. It shows two side-by-side tables. The left table has columns 'normalized-losses' and 'make'. The 'normalized-losses' column contains values 164, 164, NaN, and 158. The 'make' column contains 'audi' for all rows. The row with NaN is highlighted in orange. An arrow points to the right table, which is identical except the NaN value has been replaced by 162, and this row is also highlighted in orange.

| 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()
```

Data Preprocessing

B. Data Transformation

Pada beberapa kasus, variabel cenderung memiliki nilai rentang yang sangat besar. Nilai rentang yang sangat besar ini akan mempengaruhi hasil pengolahan data. Untuk mengatasi masalah ini, data harus ditransformasi terlebih dahulu. Normalisasi sering digunakan. Nilai rentang yang besar akan menjadi rentang yang tidak terlalu besar. Normalisasi terdapat beberapa metode.. Metode normalisasi data yang paling sering digunakan yaitu:

1. Min-Max Normalization

Min-max merupakan metode normalisasi data dengan menskalakan data diantara 0 dan 1. Metode ini menggunakan rumus:

$$X'_i = \frac{X_i - \text{Min}(X)}{\text{Max}(X) - \text{Min}(X)}$$

Misalnya data $X = [7 \ 10 \ 15 \ 20 \ 25]$

- $X'_1 = \frac{7-7}{25-7} = 0$
- $X'_2 = \frac{10-7}{25-7} = 0,1667$
- $X'_3 = \frac{15-7}{25-7} = 0,4444$
- $X'_4 = \frac{20-7}{25-7} = 0,7222$
- $X'_5 = \frac{25-7}{25-7} = 1$

Perbandingan data sebelum dan sesudah di normalisasi ditunjukkan pada Tabel 1. X adalah data sebelum dinormalisasi dan X' adalah data setelah dinormalisasi. Rentang data X berada diantara 7 dan 25 sedangkan setelah dinormalisasi rentang data menjadi diantara 0 dan 1.

Tabel 1. Perbandingan data sebelum dan sesudah dinormalisasi

| X | X' |
|----|--------|
| 7 | 0 |
| 10 | 0,1667 |
| 15 | 0,4444 |
| 20 | 0,7222 |
| 25 | 1 |

Untuk mempermudah perhitungan, perhitungan dapat dilakukan pada MATLAB. Berikut contoh source codenya

```
v = [7 10 15 20 25];
for i = 1:length(v)
    nor(i) = (v(i)-min(v))/(max(v)-min(v));
end
disp(nor)
```

Hasil :

| Command Window | | | | | |
|----------------|---|--------|--------|--------|--------|
| | 7 | 10 | 15 | 20 | 25 |
| | 0 | 0.1667 | 0.4444 | 0.7222 | 1.0000 |

2. Z-Score Standardization

Metode Z-Score Standardization merupakan metode yang menskalakan selisih antara nilai pada data dan rata-ratanya dengan nilai standar deviasinya. Metode ini menggunakan rumus:

$$X'_i = \frac{X_i - \text{Mean}(X)}{\sigma}$$

Dengan

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (X_i - \text{Mean}(X))^2}{n}}$$

Keterangan: n = Banyak data

σ = standar deviasi

Misalnya data X = [7 10 15 20 25]

- Mencari nilai rata-rata

$$\begin{aligned} \text{Mean}(X) &= \frac{7 + 10 + 15 + 20 + 25}{5} \\ &= \frac{77}{5} \\ &= 15,4 \end{aligned}$$

- Mencari standar Deviasi

$$\sigma = \sqrt{\frac{(7 - 15,4)^2 + (10 - 15,4)^2 + (15 - 15,4)^2 + (20 - 15,4)^2 + (25 - 15,4)^2}{5}}$$

$$\sigma = \sqrt{\frac{70,56 + 29,16 + 0,16 + 21,16 + 92,16}{5}}$$

$$\sigma = \sqrt{\frac{213,2}{5}}$$

$$\sigma = 6,5299$$

- $X'_1 = \frac{7-15,4}{6,5299} = -1,2864$
- $X'_i = \frac{10-15,4}{6,5299} = -0,827$
- $X'_i = \frac{15-15,4}{6,5299} = -0,0613$
- $X'_i = \frac{20-15,4}{6,5299} = 0,7044$
- $X'_i = \frac{25-15,4}{6,5299} = 1,4720$

Perbandingan data sebelum dan sesudah di normalisasi ditunjukkan pada Tabel 2. X adalah data sebelum dinormalisasi dan X' adalah data setelah dinormalisasi. Rentang data X berada diantara 7 dan 25 sedangkan setelah dinormalisasi rentang data menjadi diantara 1,4720 dan -1,2864.

Tabel 2. Perbandingan Data Sebelum dan Sesudah dinormalisasi menggunakan Z-Score

| X | X' |
|----|---------|
| 7 | -1,2864 |
| 10 | -0,827 |
| 15 | -0,0613 |
| 20 | 0,7044 |
| 25 | 1,4720 |

Untuk mempermudah perhitungan dapat dilakukan pada MATLAB dengan source code sebagai berikut:

```
A = [7 10 15 20 25];
rata2 = mean(A);
c = 0;
for i = 1:length(A)
    d(i) = (A(i)-rata2)^2;
    c = c+d(i)
    sd = sqrt(c/length(A));
end
fprintf('Standar Deviasi = %.4f\n Data Baru = ',sd);
for i = 1:length(A)
    X(i) = (A(i)-rata2)/sd;
end
disp(X)
```

Hasil :

```
Command Window
Standar Deviasi = 6.5299
Data Baru =    -1.2864    -0.8270    -0.0613     0.7044     1.4702
```

Normalisasi Data pada MATLAB

MATLAB menyediakan *function* yang dapat digunakan untuk memudahkan pengguna dalam menormalisasi data dengan beberapa metode. *Function* ini adalah `normalize`.

```
Normalisasi = normalize(A)
Normalisasi = normalize(A,method)
```

Pada syntax diatas A merupakan data yang akan dinormalisasi, secara default metode yang digunakan untuk menormalisasi data pada MATLAB adalah metode Z-Score Standardization. method yang digunakan dalam function `normalize` akan dijelaskan pada Tabel 3.

Tabel 3. Metode normalisasi pada MATLAB

| Metode | Penjelasan |
|----------|---|
| 'zscore' | Digunakan untuk data dengan rata-rata = 0 dan standar deviasi = 1 |
| 'scale' | Menskalakan data berdasarkan standar deviasi |
| 'range' | Menskalakan data pada jarak anatar 1 dan 0. |

Sebagai contoh penggunaan function normalisasi dilakukan dengan source code sebagai berikut:

```
Editor - D:\Mata Kuliah\2020\Genap\Prakt. Pembelajaran Mesin\Minggu 2\Transformasi.m*
EDITOR PUBLISH VIEW
+ New Open Save Find Files Compare Print Go To Find Comment Indent Insert % % % Breakpoints Run Run and Advance Run and Time
FILE NAVIGATE EDIT BREAKPOINTS RUN
Transformasi.m* x +
1 - A = readtable('Book1.xlsx');
2 - Normalisasi1 = normalize(A);
3 - Normalisasi2 = normalize(A, 'zscore');
4 - Normalisasi3 = normalize (A, 'scale');
5 - Normalisasi4 = normalize (A, 'range')
6
7
8
```

Keterangan dan hasil source code diatas sebagai berikut :

1. Membaca data

Hasil :

```
>> Transformasi
```

```
A =
```

```
3×3 table
```

| x1 | x2 | x3 |
|-----------|-----------|-----------|
| — | — | — |
| 1 | 2 | 3 |
| 4 | 5 | 6 |
| 7 | 8 | 9 |

2. Normlaisasi

Hasil

```
>> Transformasi
```

```
Normalisasi1 =
```

```
3×3 table
```

| x1 | x2 | x3 |
|-----------|-----------|-----------|
| — | — | — |
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

3. Normalisas dengan method ‘zscore’

Hasil :

```
>> Transformasi
```

```
Normalisasi2 =
```

```
3×3 table
```

| x1 | x2 | x3 |
|-----------|-----------|-----------|
| — | — | — |
| -1 | -1 | -1 |
| 0 | 0 | 0 |
| 1 | 1 | 1 |

4. Normalisas dengan method 'scale'

```
>> Transformasi
```

```
Normalisasi3 =
```

```
3×3 table
```

| x1 | x2 | x3 |
|-----------|-----------|-----------|
| — | — | — |
| 0.33333 | 0.66667 | 1 |
| 1.3333 | 1.6667 | 2 |
| 2.3333 | 2.6667 | 3 |

5. Normalisas dengan method 'range'

```
>> Transformasi
```

```
Normalisasi4 =
```

```
3×3 table
```

| x1 | x2 | x3 |
|-----------|-----------|-----------|
| — | — | — |
| 0 | 0 | 0 |
| 0.5 | 0.5 | 0.5 |
| 1 | 1 | 1 |

Normalisasi Data pada Phyton

Ada 3 macam normalisasi yang menggunakan Phyton, yaitu :

1. Simple Feature Scaling

Simple Feature Scaling merupakan metode normalisasi data dengan menskalakan data diantara 0 dan 1. Metode ini menggunakan rumus:

$$X'_i = \frac{X_i}{Max(X)}$$

Misalnya data X = [7 10 15 20 25]

$$\circ X'_1 = \frac{7}{25} = 0.28$$

$$\circ X'_2 = \frac{10}{25} = 0,4$$

- $X'_3 = \frac{15}{25} = 0,6$
- $X'_4 = \frac{20}{25} = 0,8$
- $X'_5 = \frac{25}{25} = 1$

Normalisasi setiap atribut dapat menerapkan kode berikut. Nilai atribut yang akan dinormalisasi atribut 'umur' dan 'gaji'.

```
df["Umur"] = df["Umur"] / df["Umur"].max()
df["Gaji"] = df["Gaji"] / df["Gaji"].max()
```

2. Min-Max

Penjelasan sama seperti bagian diatas.

```
df["Umur"] = (df["Umur"] - df["Umur"].min()) / (df["Umur"].max() - df["Umur"].min())
df["Gaji"] = (df["Gaji"] - df["Gaji"].min()) / (df["Gaji"].max() - df["Gaji"].min())
```

3. Z score

Penjelasan sama seperti bagian diatas.

```
df["Umur"] = (df["Umur"] - df["Umur"].mean()) / df["Umur"].std()
df["Gaji"] = (df["Gaji"] - df["Gaji"].mean()) / df["Gaji"].std()
```

Normalisasi :

1. Langkah pertama adalah import library terlebih dahulu, library yang digunakan adalah dan Pandas.

```
import pandas as pd
from sklearn import preprocessing
```

2. Baca data

Misalkan data yang digunakan dengan nama "shopping_data.csv". Untuk nilai dan atribut data ini buatlah sendiri (bebas)

```
data = pd.read_csv("shopping_data.csv")
df = pd.DataFrame(data)
```

3. Normalisasi

```
min_max_scaler = preprocessing.MinMaxScaler()
np_scaled = min_max_scaler.fit_transform(df)
df_normalized = pd.DataFrame(np_scaled)
```

Data dan Preprocessing

Soal :

Waktu : 60 menit

1. Carilah data bebas
2. Create data – data tersebut menggunakan matlab dan phyton
 - ❖ Inputkan 3 record dari data tersebut ke dalam matlab dan phyton.
3. Pilihlah beberapa atribut dari data tersebut
4. Lakukan PreProcessing dari data dengan atribut yang dipilih menggunakan Matlab dan Phyton.
 - ❖ Cek apakah data-data tersebut memiliki outlier dan missing value
 - ❖ Jika terdapat outlier dan missing value maka lakukan penanganannya
 - ❖ Normalisasikan data – data tersebut dengan 1 metode matlab dan phyton (bebas)

Tugas :

1. Buat laporan soal diatas
2. Laporan terdiri dari Source code, print screen hasil, dan penjelasannya
3. Penamaan file : “Laporan Data dan Preprocessing NIM.pdf”
4. Paling lambat pengumpulan hari Jumat / 8 September 2023 pukul 23.59
5. Tugas dikerjakan secara individu