DIPO ANALYTIC

Data Wrangling Dengan Python

Alfazrin Banapon

- Jika ada yang bertanya kepada data analysts, data scientists, atau statisticians tentang tugas apa yang paling sering mereka lakukan, the answer is **Data Wrangling**
- **Data wrangling, data munging, atau data transformation** adalah proses transformasi data **‘mentah’** menjadi format siap pakai dalam analisis.
- Sebagai *data scientist* keterampilan **Data Wrangling** merupakan core yang **harus** dimiliki

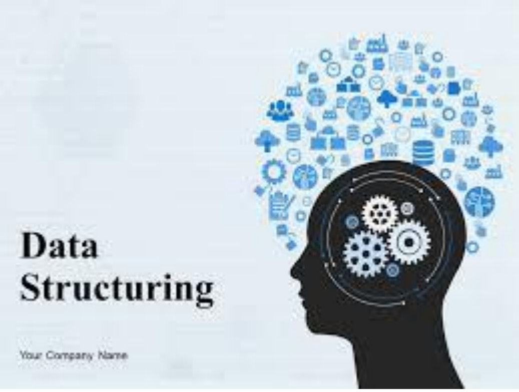
75%

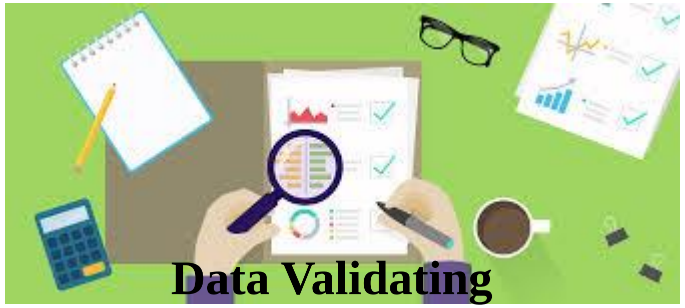
Waktu pekerjaan Data Science di habiskan di tahapan ini

Data Wrangling adalah **Dirty Work** dalam alur kerja analisis data

Menurut Trifacta, terdapat 6 core aktifitas dalam proses Data Wrangling

Data Discovering

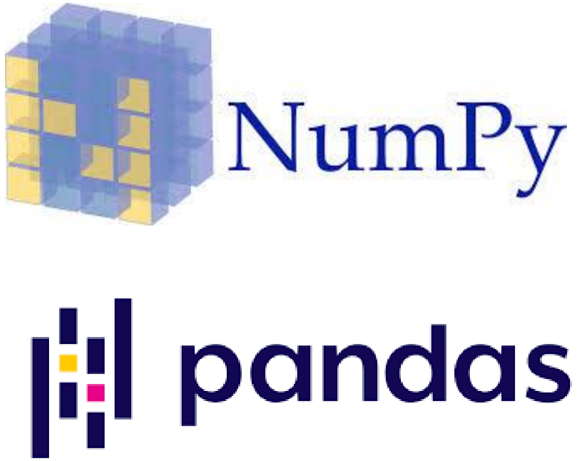



Data Validating

Data Publishing

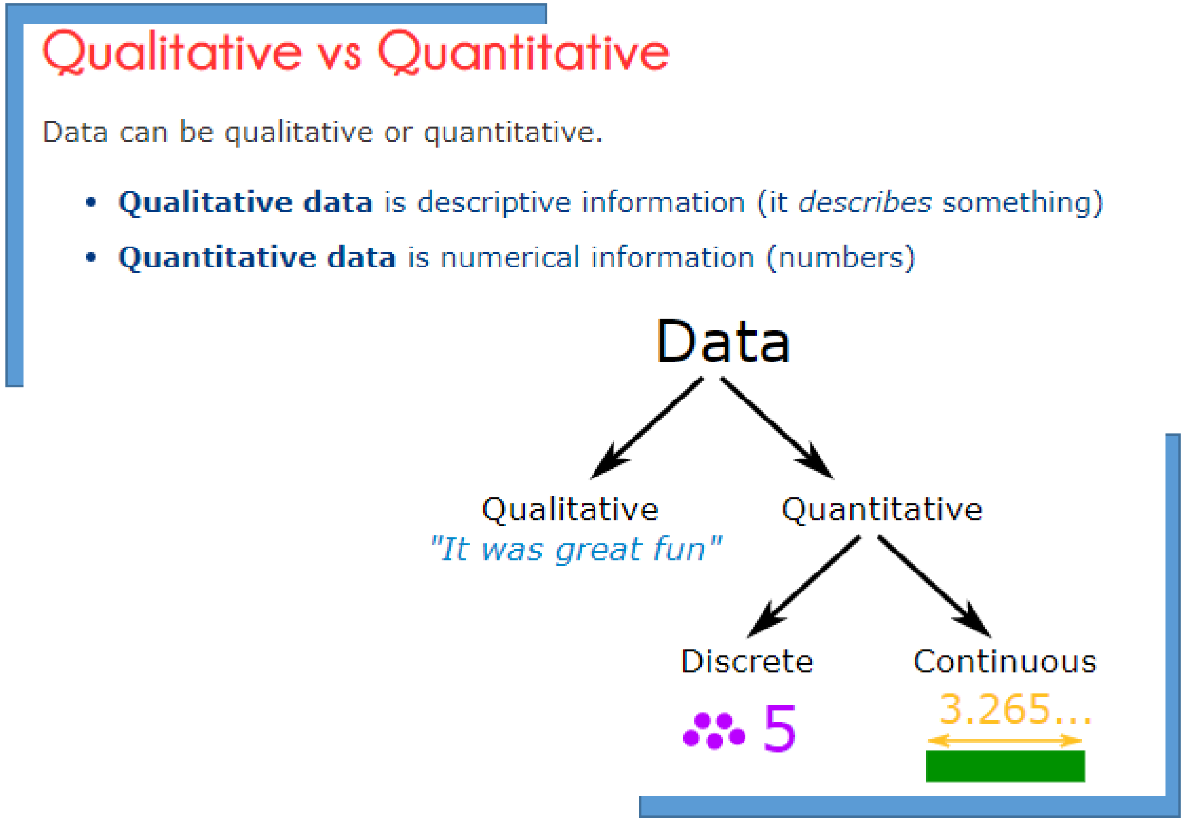
Which Programing Language can we use ?

Which Library in Python can we use, for Data Wrangling ?



Kenalan Dulu Sama
DATA

Data are characteristics or information, usually numerical, that are collected through observation. In a more technical sense, **data** are a set of values of qualitative or quantitative variables about one or more persons or objects, while a datum (singular of **data**) is a single value of a single variable.



Qualitative vs Quantitative

Data can be qualitative or quantitative.

- **Qualitative data** is descriptive information (it *describes* something)
- **Quantitative data** is numerical information (numbers)

Data

Qualitative

"It was great fun"

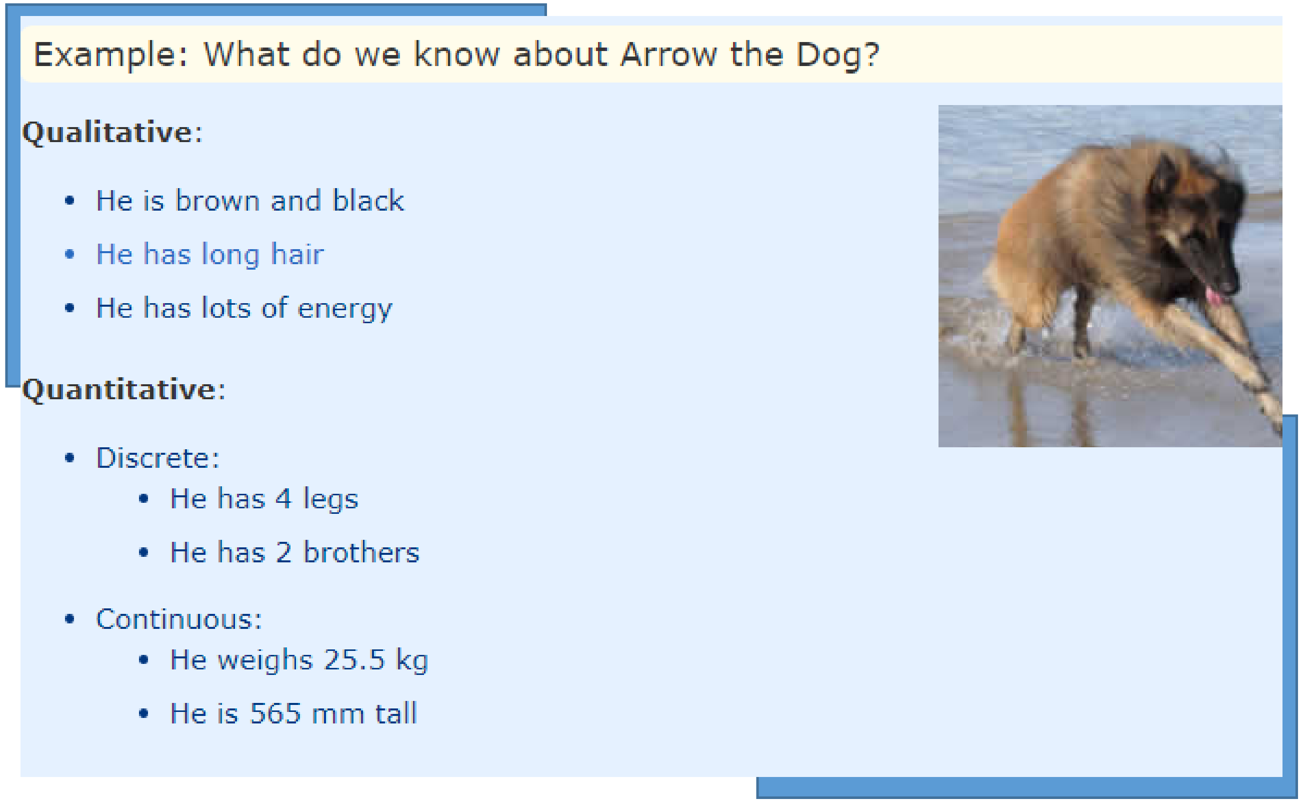
Quantitative

Discrete

5

Continuous

3.265...



Example: What do we know about Arrow the Dog?

Qualitative:

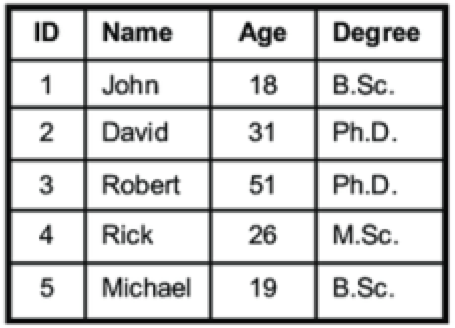
- He is brown and black
- He has long hair
- He has lots of energy

Quantitative:

- Discrete:
  - He has 4 legs
  - He has 2 brothers
- Continuous:
  - He weighs 25.5 kg
  - He is 565 mm tall

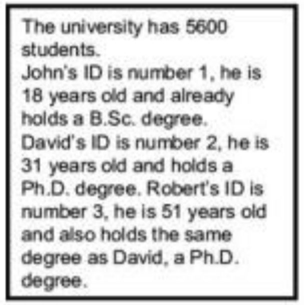
DATA

Structured Data



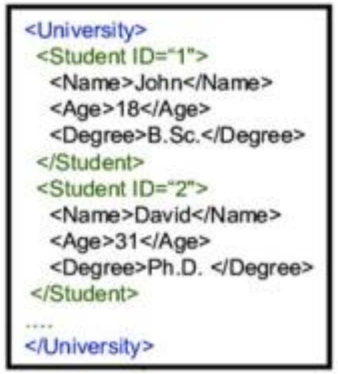
ID	Name	Age	Degree
1	John	18	B.Sc.
2	David	31	Ph.D.
3	Robert	51	Ph.D.
4	Rick	26	M.Sc.
5	Michael	19	B.Sc.

Unstructured Data



The university has 5600 students. John's ID is number 1, he is 18 years old and already holds a B.Sc. degree. David's ID is number 2, he is 31 years old and holds a Ph.D. degree. Robert's ID is number 3, he is 51 years old and also holds the same degree as David, a Ph.D. degree.

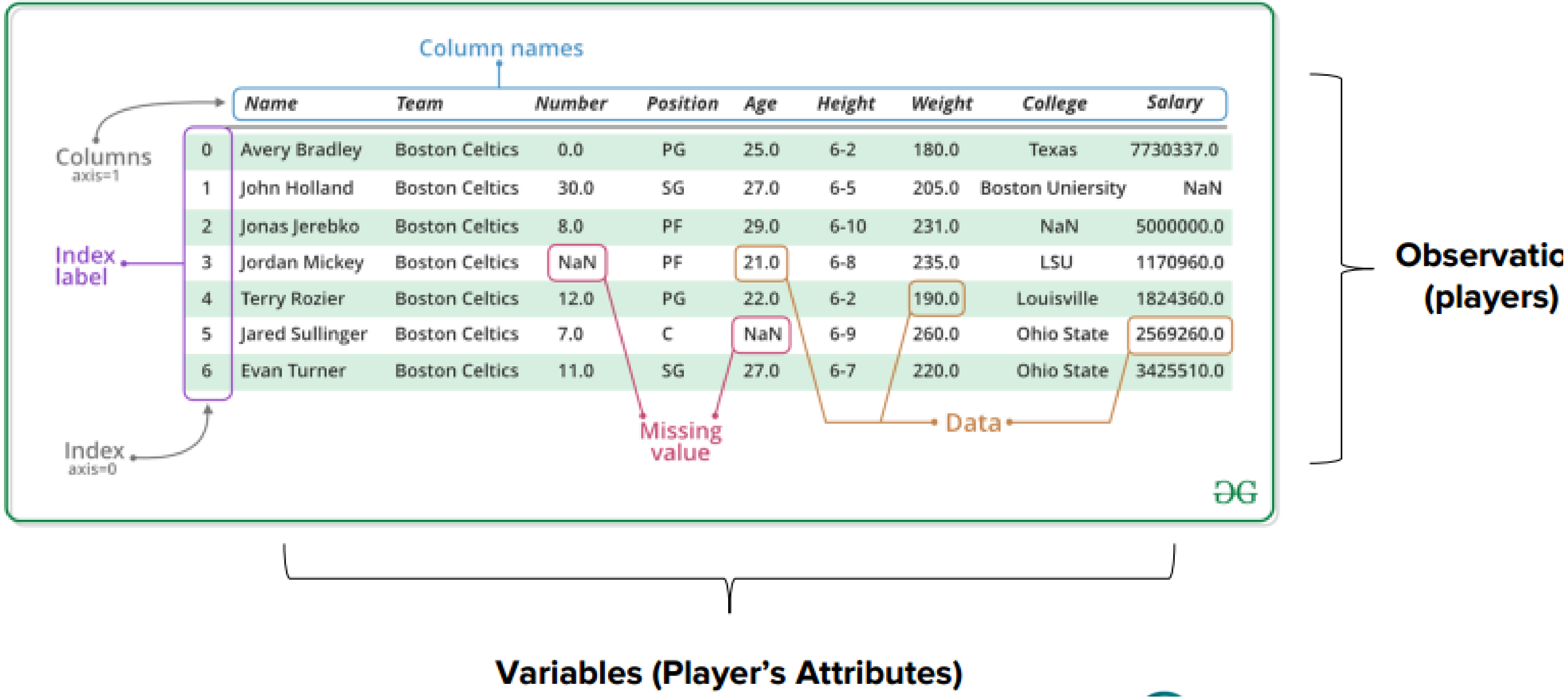
Semi-structured Data



```
<University>
 <Student ID="1">
  <Name>John</Name>
  <Age>18</Age>
  <Degree>B.Sc.</Degree>
 </Student>
 <Student ID="2">
  <Name>David</Name>
  <Age>31</Age>
  <Degree>Ph.D. </Degree>
</Student>
....
</University>
```

STRUCTURED DATA : TABULAR

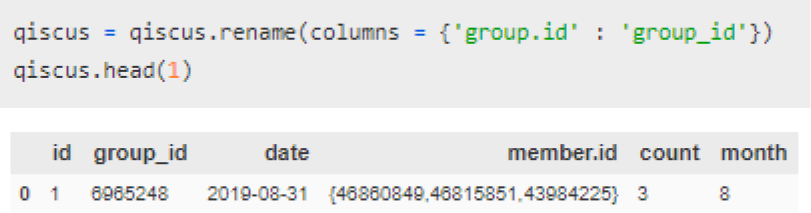
- Data unit individu dari suatu informasi
- Data di organisir pada suatu matriks (seperti numpy)
- Baris merepresentasikan Observasi
- Colum Merepresentasikan Variabel atau Feature



	Name	Team	Number	Position	Age	Height	Weight	College	Salary
0	Avery Bradley	Boston Celtics	0.0	PG	25.0	6-2	180.0	Texas	7730337.0
1	John Holland	Boston Celtics	30.0	SG	27.0	6-5	205.0	Boston Uniersity	NaN
2	Jonas Jerebko	Boston Celtics	8.0	PF	29.0	6-10	231.0	NaN	5000000.0
3	Jordan Mickey	Boston Celtics	NaN	PF	21.0	6-8	235.0	LSU	1170960.0
4	Terry Rozier	Boston Celtics	12.0	PG	22.0	6-2	190.0	Louisville	1824360.0
5	Jared Sullinger	Boston Celtics	7.0	C	NaN	6-9	260.0	Ohio State	2569260.0
6	Evan Turner	Boston Celtics	11.0	SG	27.0	6-7	220.0	Ohio State	3425510.0

Data Cleaning

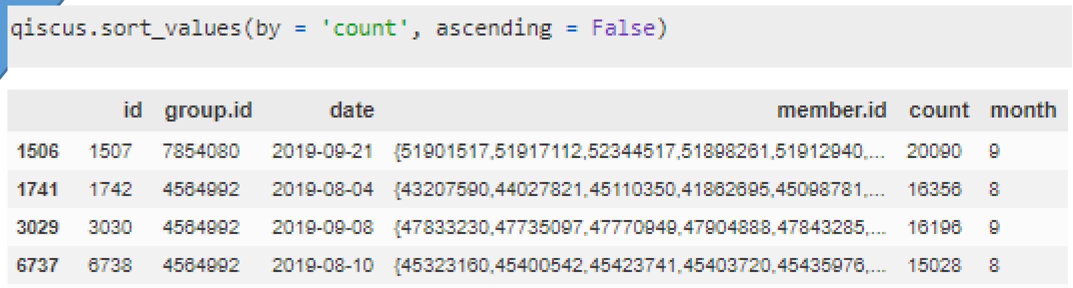
Renaming



```
qiscus = qiscus.rename(columns = {'group.id' : 'group_id'})
qiscus.head(1)
```

	id	group_id	date	member.id	count	month
0	1	6965248	2019-08-31	{46860849,46815851,43984225}	3	8

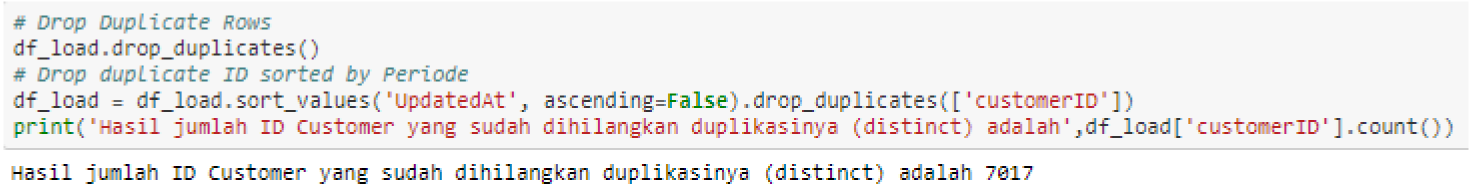
Sorting and Reording



```
qiscus.sort_values(by = 'count', ascending = False)
```

	id	group.id	date	member.id	count	month
1506	1507	7854080	2019-09-21	{51901517,51917112,52344517,51898261,51912940,...	20090	9
1741	1742	4564992	2019-08-04	{43207590,44027821,45110350,41862695,45098781,...	16356	8
3029	3030	4564992	2019-09-08	{47833230,47735097,47770949,47904888,47843285,...	16196	9
6737	6738	4564992	2019-08-10	{45323160,45400542,45423741,45403720,45435976,...	15028	8

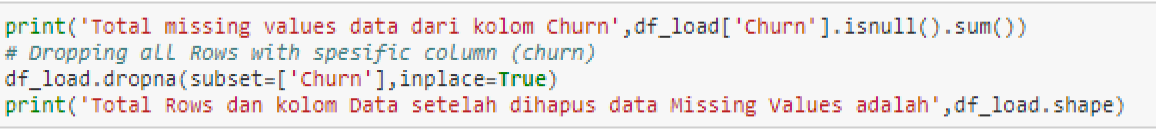
Removing Duplicate Data



```
# Drop Duplicate Rows
df_load.drop_duplicates()
# Drop duplicate ID sorted by Periode
df_load = df_load.sort_values('UpdatedAt', ascending=False).drop_duplicates(['customerID'])
print('Hasil jumlah ID Customer yang sudah dihilangkan duplikasinya (distinct) adalah',df_load['customerID'].count())
```

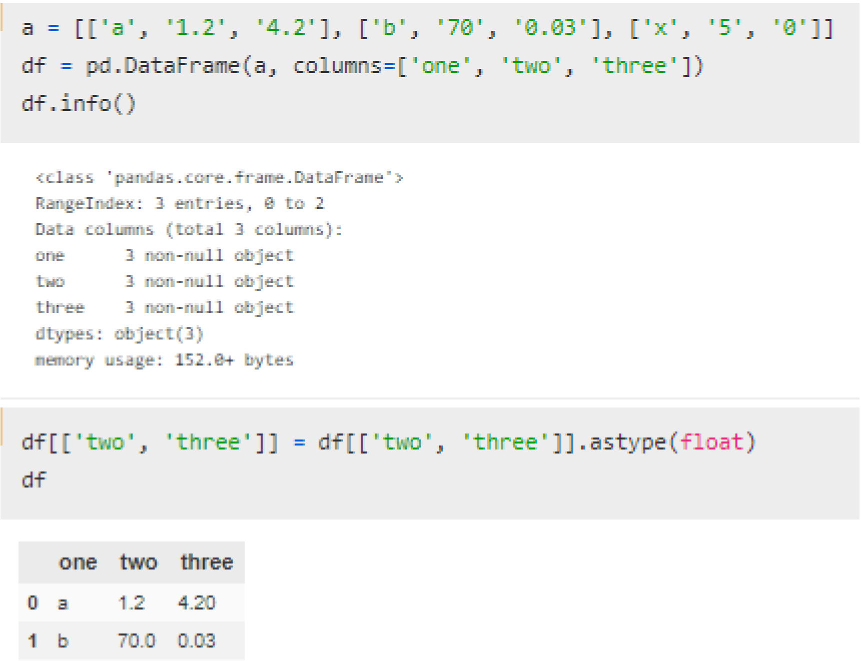
```
Hasil jumlah ID Customer yang sudah dihilangkan duplikasinya (distinct) adalah 7017
```

Handling Missing Value



```
print('Total missing values data dari kolom Churn',df_load['Churn'].isnull().sum())
# Dropping all Rows with spesific column (churn)
df_load.dropna(subset=['Churn'],inplace=True)
print('Total Rows dan kolom Data setelah dihapus data Missing Values adalah',df_load.shape)
```

Data Type Conversion



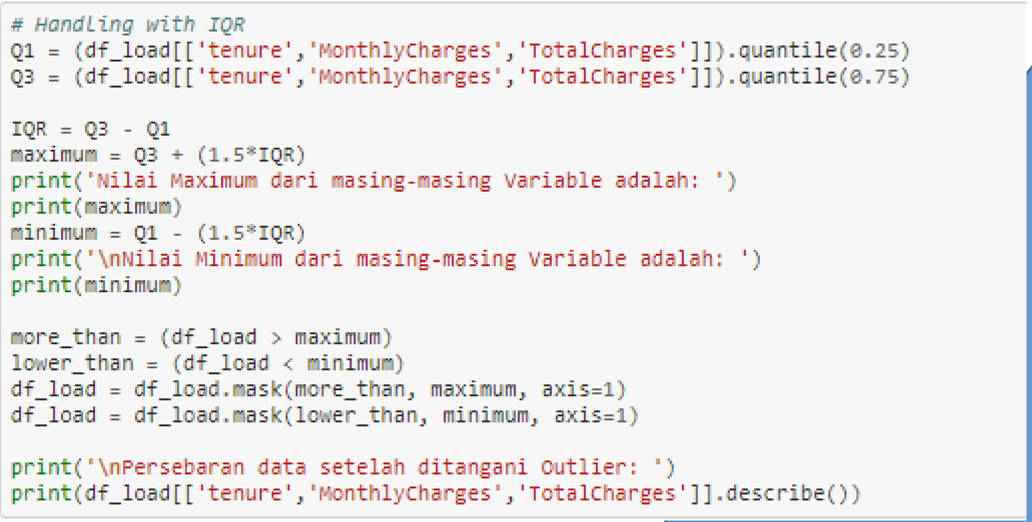
```
a = [['a', '1.2', '4.2'], ['b', '70', '0.03'], ['x', '5', '0']]
df = pd.DataFrame(a, columns=['one', 'two', 'three'])
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3 entries, 0 to 2
Data columns (total 3 columns):
one      3 non-null object
two      3 non-null object
three    3 non-null object
dtypes: object(3)
memory usage: 152.0+ bytes
```

```
df[['two', 'three']] = df[['two', 'three']].astype(float)
df
```

	one	two	three
0	a	1.2	4.20
1	b	70.0	0.03

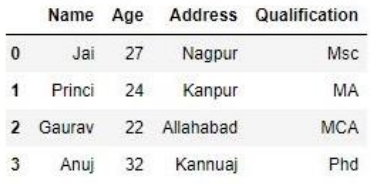
Handling Outlier



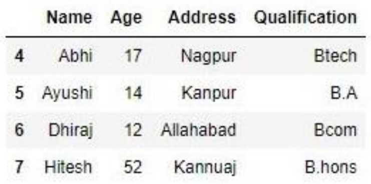
```
# Handling with IQR
Q1 = (df_load[['tenure','MonthlyCharges','TotalCharges']]).quantile(0.25)
Q3 = (df_load[['tenure','MonthlyCharges','TotalCharges']]).quantile(0.75)

IQR = Q3 - Q1
maximum = Q3 + (1.5*IQR)
print('Nilai Maximum dari masing-masing Variable adalah: ')
print(maximum)
minimum = Q1 - (1.5*IQR)
print('\nNilai Minimum dari masing-masing Variable adalah: ')
print(minimum)

more_than = (df_load > maximum)
lower_than = (df_load < minimum)
df_load = df_load.mask(more_than, maximum, axis=1)
df_load = df_load.mask(lower_than, minimum, axis=1)

print('\nPersebaran data setelah ditangani Outlier: ')
print(df_load[['tenure','MonthlyCharges','TotalCharges']].describe())
```

DATA STRUCTURING

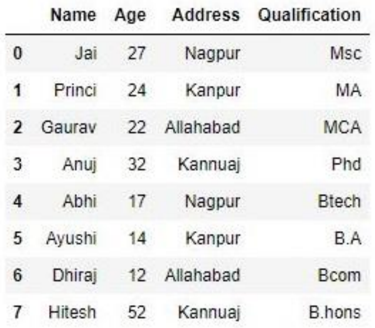
Data tersedia dalam berbagai ukuran dan bentuk, sebagai data scientist pada proses ini dapat dilakukan proses merge, order or reshape data.



	Name	Age	Address	Qualification
0	Jai	27	Nagpur	Msc
1	Princi	24	Kanpur	MA
2	Gaurav	22	Allahabad	MCA
3	Anuj	32	Kannuaj	Phd

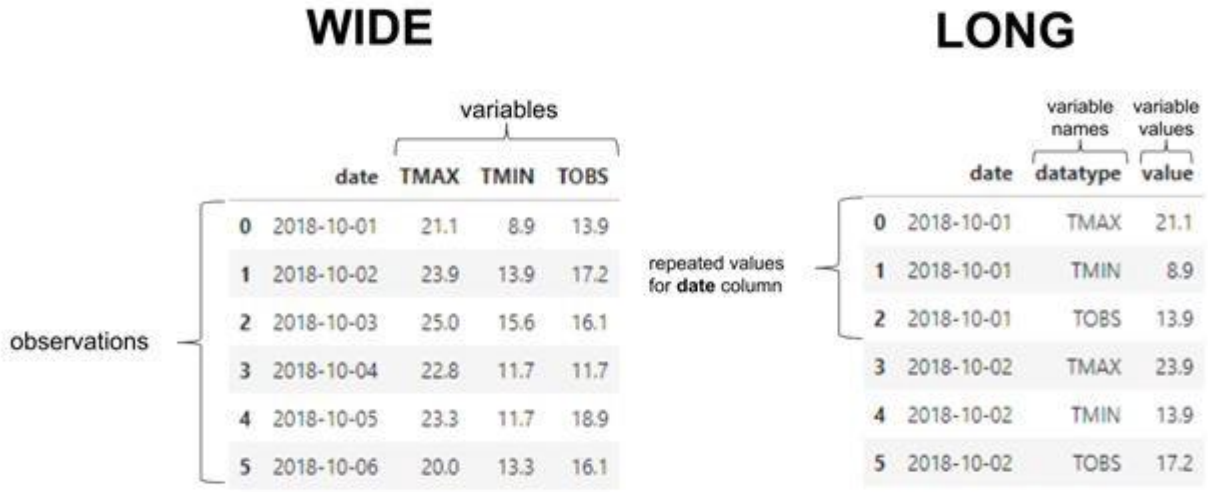


	Name	Age	Address	Qualification
4	Abhi	17	Nagpur	Btech
5	Ayushi	14	Kanpur	B.A
6	Dhiraj	12	Allahabad	Bcom
7	Hitesh	52	Kannuaj	B.hons



	Name	Age	Address	Qualification
0	Jai	27	Nagpur	Msc
1	Princi	24	Kanpur	MA
2	Gaurav	22	Allahabad	MCA
3	Anuj	32	Kannuaj	Phd
4	Abhi	17	Nagpur	Btech
5	Ayushi	14	Kanpur	B.A
6	Dhiraj	12	Allahabad	Bcom
7	Hitesh	52	Kannuaj	B.hons

Atau juga biasanya melakukan perubahan pada struktur data, hal ini biasanya melibatkan switch pada baris dan kolom



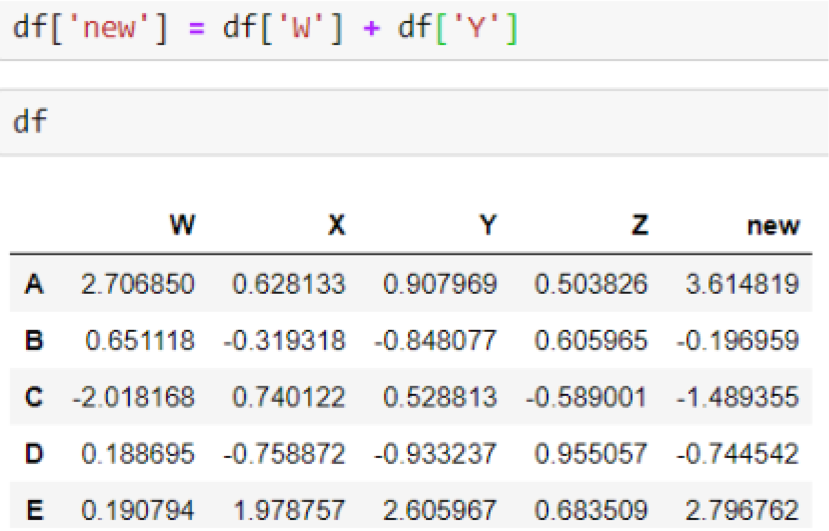
	date	TMAX	TMIN	TOBS
0	2018-10-01	21.1	8.9	13.9
1	2018-10-02	23.9	13.9	17.2
2	2018-10-03	25.0	15.6	16.1
3	2018-10-04	22.8	11.7	11.7
4	2018-10-05	23.3	11.7	18.9
5	2018-10-06	20.0	13.3	16.1

	date	datatype	value
0	2018-10-01	TMAX	21.1
1	2018-10-01	TMIN	8.9
2	2018-10-01	TOBS	13.9
3	2018-10-02	TMAX	23.9
4	2018-10-02	TMIN	13.9
5	2018-10-02	TOBS	17.2

Data Enrichment

Mostly pada bagian ini, digunakan untuk memperkaya data. Dapat digunakan untuk menggabungkan data baru atau membuat kolom baru berdasarkan data yang sudah ada, Beberapa cara untuk Enrichment Data adalah

- **Adding New Column**

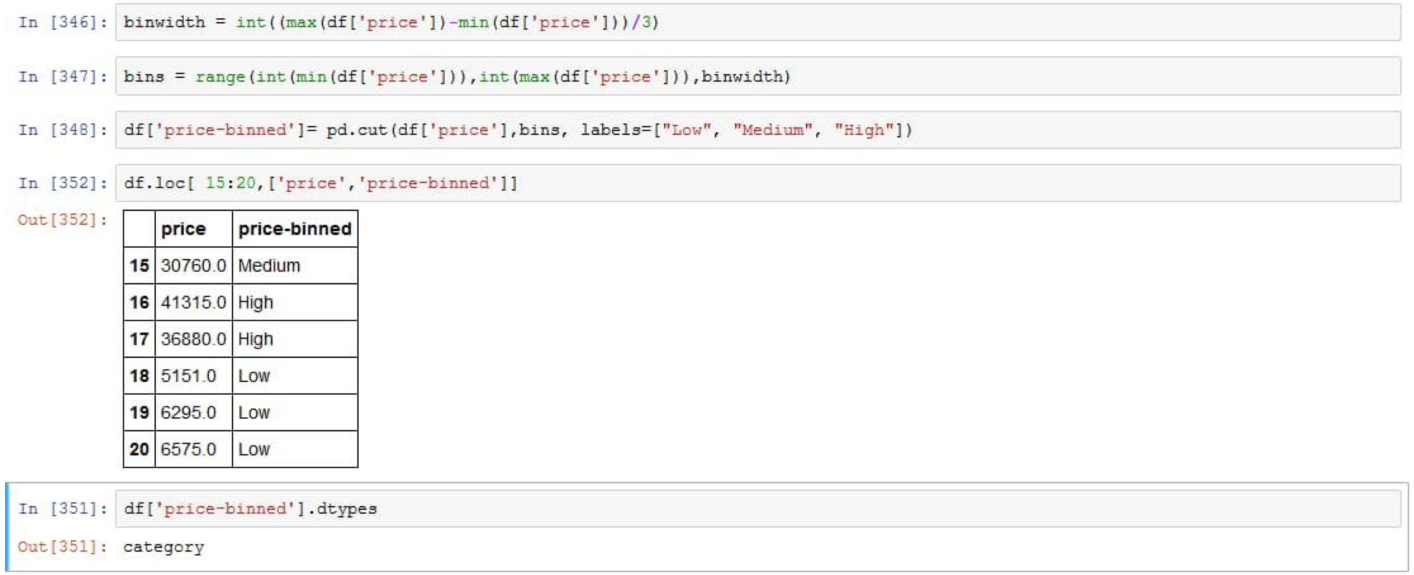


```
df['new'] = df['W'] + df['Y']
```

```
df
```

	W	X	Y	Z	new
A	2.706850	0.628133	0.907969	0.503826	3.614819
B	0.651118	-0.319318	-0.848077	0.605965	-0.196959
C	-2.018168	0.740122	0.528813	-0.589001	-1.489355
D	0.188695	-0.758872	-0.933237	0.955057	-0.744542
E	0.190794	1.978757	2.605967	0.683509	2.796762

- **Binning**



```
In [346]: binwidth = int((max(df['price'])-min(df['price']))/3)

In [347]: bins = range(int(min(df['price'])),int(max(df['price'])),binwidth)

In [348]: df['price-binned']= pd.cut(df['price'],bins, labels=["Low", "Medium", "High"])

In [352]: df.loc[ 15:20,['price','price-binned']]

Out[352]:
```

	price	price-binned
15	30760.0	Medium
16	41315.0	High
17	36880.0	High
18	5151.0	Low
19	6295.0	Low
20	6575.0	Low

```
In [351]: df['price-binned'].dtypes

Out[351]: category
```

ARE YOU READY FOR CODING WITH DIRTY WORK ??