Part 5 : Pandas dan Manipulasi Data

DQLab LiveClass

Outline

- Pandas
- Membaca data dengan pandas
- Konsep dataframe dan series
- Slicing dan filtering data
- Sorting data
- Summarizing data
- Iterasi data
- Export data

Pandas

- Pandas adalah library python yang digunakan untuk manipulasi dan analisis data
- Pandas memiliki struktur data tertentu yaitu DataFrame yang memudahkan proses analisis data
- Untuk dapat menggunakan pandas, import pandas ke dalam program dengan menggunakan

```
import pandas as pd
```

Membaca file dengan pandas

- Gunakan pd.read_csv(<nama_file_csv>) untuk membaca data dari file csv menjadi dataframe
- Contoh:

```
df = pd.read_csv('SuperStore.csv')
```

- Untuk menampilkan 5 baris teratas dari df gunakan df.head()

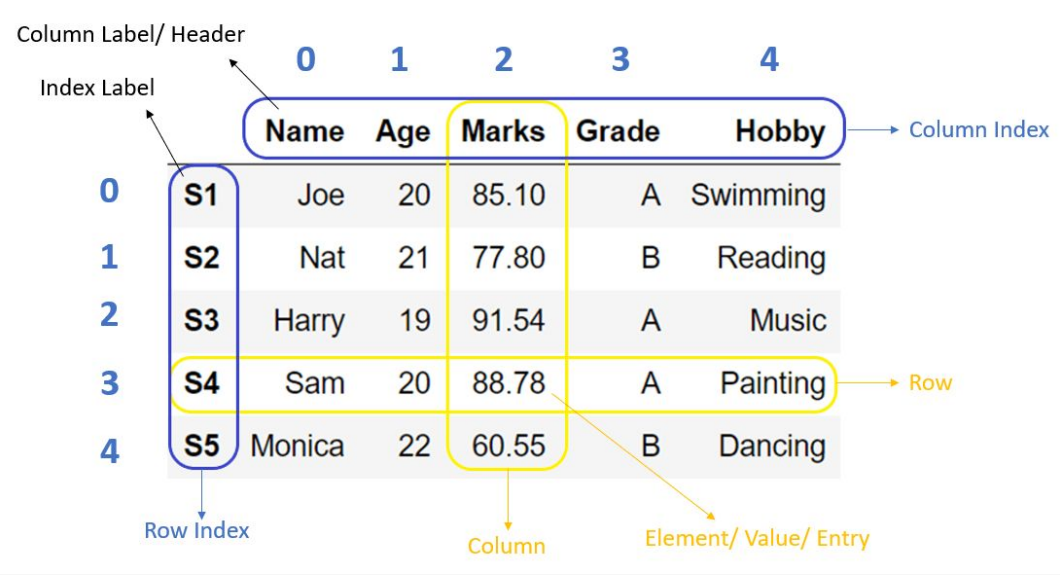
```
[1] import pandas as pd
```

```
[2] df = pd.read_csv('SuperStore.csv')
```

```
[3] df.head()
```

	Order_ID	Customer_ID	Postal_Code	Product_ID	Sales	Quantity	Discount	Profit	Category	Sub-Category	Product_Name	Order_Date	Ship_Date	Ship_Mode	Cust
0	CA-2019-152156	CG-12520	42420	FUR-BO-10001798	261.9600	2	0.00	41.9136	Furniture	Bookcases	Bush Somerset Collection Bookcase	11/8/2019	11/11/2019	Second Class	
1	CA-2019-152156	CG-12520	42420	FUR-CH-10000454	731.9400	3	0.00	219.5820	Furniture	Chairs	Hon Deluxe Fabric Upholstered Stacking Chairs,...	11/8/2019	11/11/2019	Second Class	
2	CA-2019-138688	DV-13045	90036	OFF-LA-10000240	14.6200	2	0.00	6.8714	Office Supplies	Labels	Self-Adhesive Address Labels for Typewriters b...	6/12/2019	6/16/2019	Second Class	Da
3	US-2018-108966	SO-20335	33311	FUR-TA-10000577	957.5775	5	0.45	-383.0310	Furniture	Tables	Bretford CR4500 Series Slim Rectangular Table	10/11/2018	10/18/2018	Standard Class	Se

DataFrame dan Series



Dataframe adalah tabel, sedangkan setiap kolomnya adalah series

- DataFrame adalah struktur data 2 dimensi (tabel) yang terdiri dari baris dan kolom
- Setiap kolom dari dataframe bisa jadi memiliki tipe data yang berbeda namun data pada kolom yang sama memiliki tipe data yang sama
- Series adalah data 1 dimensi yang homogen
- DataFrame terdiri dari berbagai macam series dengan panjang yang sama

```
[5] # menampilkan tipe data dari df yaitu DataFrame
    type(df)

    pandas.core.frame.DataFrame

[6] # kolom kolom dari DataFrame adalah Series
    type(df['Customer_ID'])

    pandas.core.series.Series
```

Selection

- Untuk memilih satu kolom tertentu dari suatu dataframe gunakan

  df[<nama_kolom>]

  contoh: df['alamat']

- Jika kolom yang dipilih lebih dari satu, gunakan

  df[<list_berisi_nama_kolom>]

  contoh: df[['nama','alamat','no_telp']]

Selection dengan loc dan iloc

- Untuk memilih atau mengambil sebagian dari dataframe hingga baris dan kolom tertentu dapat dilakukan dengan dua cara
  - df.loc[<baris>:<nama_kolom>]
  - df.iloc[<baris>:<index_kolom>]
- Pada loc dan iloc, baris yang dimaksud adalah berupa range of index
- Pada loc, untuk memanggil kolom cukup gunakan nama kolom atau list berisi nama kolom (jika kolom lebih dari satu)
- Pada iloc, kolom dipanggil menggunakan index dari kolom
- df.columns berfungsi untuk menampilkan list of columns

loc vs iloc

```
# menampilkan 5 baris teratas dari nama-nama
# kolom yang telah didefinisikan

nama_kolom = ['Category','Customer_ID','City']
df.loc[:5,nama_kolom]
```

	Category	Customer_ID	City
0	Furniture	CG-12520	Henderson
1	Furniture	CG-12520	Henderson
2	Office Supplies	DV-13045	Los Angeles
3	Furniture	SO-20335	Fort Lauderdale
4	Office Supplies	SO-20335	Fort Lauderdale
5	Furniture	BH-11710	Los Angeles

```
# memanggil 10 baris pertama dari
# kolom dengan index 1,3,5,7

df.iloc[:10,[1,3,5,7]]
```

	Customer_ID	Product_ID	Quantity	Profit
0	CG-12520	FUR-BO-10001798	2	41.9136
1	CG-12520	FUR-CH-10000454	3	219.5820
2	DV-13045	OFF-LA-10000240	2	6.8714
3	SO-20335	FUR-TA-10000577	5	-383.0310
4	SO-20335	OFF-ST-10000760	2	2.5164
5	BH-11710	FUR-FU-10001487	7	14.1694
6	BH-11710	OFF-AR-10002833	4	1.9656
7	BH-11710	TEC-PH-10002275	6	90.7152
8	BH-11710	OFF-BI-10003910	3	5.7825
9	BH-11710	OFF-AP-10002892	5	34.4700

Filtering

- Filtering digunakan untuk memilih baris-baris yang memenuhi kondisi tertentu
- Syntax:

df[<kondisi>]

- Jika kondisi lebih dari satu, pisahkan kondisi dengan tanda kurung dan sambungkan dengan operator bitwise seperti
  - & untuk **and**
  - | untuk **or**
  - ~ untuk **not**

Contoh: Satu kondisi

```
# contoh, memfilter sales dengan nilai lebih dari 100

df[df['Sales'] > 100].head()
```

	Order_ID	Customer_ID	Postal_Code	Product_ID	Sales	Quantity	Discount
0	CA-2019-152156	CG-12520	42420	FUR-BO-10001798	261.9600	2	0.00
1	CA-2019-152156	CG-12520	42420	FUR-CH-10000454	731.9400	3	0.00
3	US-2018-108966	SO-20335	33311	FUR-TA-10000577	957.5775	5	0.45
7	CA-2017-115812	BH-11710	90032	TEC-PH-10002275	907.1520	6	0.20
9	CA-2017-115812	BH-11710	90032	OFF-AP-10002892	114.9000	5	0.00

Contoh: Dua kondisi

```
# memfilter transaksi dengan sales > 100 dari kategori Furniture
df[(df['Sales'] > 100) & (df['Category'] == 'Furniture')].head()
```

	Order_ID	Customer_ID	Postal_Code	Product_ID	Sales	Quantity	Discount
0	CA-2019-152156	CG-12520	42420	FUR-BO-10001798	261.9600	2	0.00
1	CA-2019-152156	CG-12520	42420	FUR-CH-10000454	731.9400	3	0.00
3	US-2018-108966	SO-20335	33311	FUR-TA-10000577	957.5775	5	0.45
10	CA-2017-115812	BH-11710	90032	FUR-TA-10001539	1706.1840	9	0.20
24	CA-2018-106320	EB-13870	84057	FUR-TA-10000577	1044.6300	3	0.00

Sorting

- Sorting atau mengurutkan data berdasarkan kolom tertentu dapat menggunakan

  df.sort_values(by=<nama atau list kolom>, ascending=True/False)

- Ascending = True (default) artinya data diurutkan secara menaik, sebaliknya data diurutkan secara menurun
- Jika akan mengurutkan dengan lebih dari 1 kolom dan masing masing kolom memiliki metode pengurutan yang berbeda, maka pada parameter ascending diisi dengan list nilai boolean untuk masing-masing kolom, contoh

  df.sort_values(by=['kolom 1', 'kolom 2'], ascending=[True, False])

Contoh Sorting

```
# sorting berdasarkan tanggal
# secara default, ascending = False
# sehingga diurutkan dari tanggal terkecil ke terbesar

df.sort_values(by='Order_Date')
```

	Order_ID	Customer_ID	Postal_Code	Product_ID	Sales	Quantity	Discount	Profit	Category	Sub-Category	Product_Name	Order_Date	Ship_Date
7980	CA-2017-103800	DP-13000	77095	OFF-PA-10000174	16.448	2	0.2	5.5512	Office Supplies	Paper	Message Book, Wirebound, Four 5 1/2" X 4" Form...	2017-01-03	2017-01-07
739	CA-2017-112326	PO-19195	60540	OFF-LA-10003223	11.784	3	0.2	4.2717	Office Supplies	Labels	Avery 508	2017-01-04	2017-01-08
740	CA-2017-112326	PO-19195	60540	OFF-ST-10002743	272.736	3	0.2	-64.7748	Office Supplies	Storage	SAFCO Boltless Steel Shelving	2017-01-04	2017-01-08
741	CA-2017-112326	PO-19195	60540	OFF-BI-10004094	3.540	2	0.8	-5.4870	Office Supplies	Binders	GBC Standard Plastic Binding Systems Combs	2017-01-04	2017-01-08

Contoh Sorting: Dua kolom

```
# sorting berdasarkan dua kolom

df.sort_values(by=['Category','Sub-Category'])
```

	Order_ID	Customer_ID	Postal_Code	Product_ID	Sales	Quantity	Discount	Profit	Category	Sub-Category	Product_Name	Order_Date
0	CA-2019-152156	CG-12520	42420	FUR-BO-10001798	261.9600	2	0.00	41.9136	Furniture	Bookcases	Bush Somerset Collection Bookcase	2019-11-08
27	US-2018-150630	TB-21520	19140	FUR-BO-10004834	3083.4300	7	0.50	-1665.0522	Furniture	Bookcases	Riverside Palais Royal Lawyers Bookcase, Royal...	2018-09-17
38	CA-2018-117415	SN-20710	77041	FUR-BO-10002545	532.3992	3	0.32	-46.9764	Furniture	Bookcases	Atlantic Metals Mobile 3-Shelf Bookcases, Cust...	2018-12-27
189	CA-2018-102281	MP-17470	10035	FUR-BO-10002613	899.1360	4	0.20	112.3920	Furniture	Bookcases	Atlantic Metals Mobile 4-Shelf Bookcases, Cust...	2018-10-12
192	CA-2018-102281	MP-17470	10035	FUR-BO-10002545	626.3520	3	0.20	46.9764	Furniture	Bookcases	Atlantic Metals Mobile 3-Shelf Bookcases, Cust...	2018-10-12
...	...	...	...	...	...	...	...	...	...	...	...	...

Sorting: Dua kolom dengan arah berbeda

```
# Sorting menggunakan dua kolom
# Category secara menurun
# Sub-Category secara menaik

df.sort_values(by=['Category','Sub-Category'], ascending=[False, True])
```

	Order_ID	Customer_ID	Postal_Code	Product_ID	Sales	Quantity	Discount	Profit	Category	Sub-Category	Product_Name	Order_Date
26	CA-2019-121755	EH-13945	90049	TEC-AC-10003027	90.570	3	0.0	11.7741	Technology	Accessories	Imation 8GB Mini TravelDrive USB 2.0 Flash Drive	2019-01-16
44	CA-2019-118255	ON-18715	55122	TEC-AC-10000171	45.980	2	0.0	19.7714	Technology	Accessories	Verbatim 25 GB 6x Blu-ray Single Layer Recorda...	2019-03-11
47	CA-2019-169194	LH-16900	19901	TEC-AC-10002167	45.000	3	0.0	4.9500	Technology	Accessories	Imation 8gb Micro Traveldrive Usb 2.0 Flash Drive	2019-06-20
59	CA-2019-111682	TB-21055	12180	TEC-AC-10002167	30.000	2	0.0	3.3000	Technology	Accessories	Imation 8gb Micro Traveldrive Usb 2.0 Flash Drive	2019-06-17
62	CA-2018-135545	KM-16720	90004	TEC-AC-10004633	13.980	2	0.0	6.1512	Technology	Accessories	Verbatim 25 GB 6x Blu-ray Single Layer Recorda...	2018-11-24
...	...	...	...	...	...	...	...	...	...	...	...	...

Summarize: Dataset info

- df.info() memuat beberapa informasi dasar dari dataset diantaranya nama kolom beserta tipe datanya

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9994 entries, 0 to 9993
Data columns (total 20 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   Order_ID        9994 non-null   object
 1   Customer_ID     9994 non-null   object
 2   Postal_Code     9994 non-null   int64
 3   Product_ID      9994 non-null   object
 4   Sales           9994 non-null   float64
 5   Quantity        9994 non-null   int64
 6   Discount        9994 non-null   float64
 7   Profit          9994 non-null   float64
 8   Category        9994 non-null   object
 9   Sub-Category    9994 non-null   object
 10  Product_Name    9994 non-null   object
 11  Order_Date      9994 non-null   datetime64[ns]
 12  Ship_Date       9994 non-null   datetime64[ns]
 13  Ship_Mode       9994 non-null   object
 14  Customer_Name   9994 non-null   object
 15  Segment         9994 non-null   object
 16  Country/Region  9994 non-null   object
 17  City            9994 non-null   object
 18  State           9994 non-null   object
 19  Region          9994 non-null   object
dtypes: datetime64[ns](2), float64(3), int64(2), object(13)
memory usage: 1.5+ MB
```

Summarize: Descriptive statistics

- Untuk menampilkan statistik deskriptif dari data seperti menampilkan count, mean, std deviasi, min, max, kuartil 25%, 50%, 75% dapat menggunakan

```
df.describe()
```

- Namun secara default df.describe() akan memunculkan statistik deskriptif untuk kolom kolom bertipe numerik
- Untuk menampilkan statistik deskriptif dari kolom non numerik gunakan

```
df.describe(include='o')
```

Statistik deskriptif: numerik

```
[24] df.describe()
```

	Postal_Code	Sales	Quantity	Discount	Profit
count	9994.000000	9994.000000	9994.000000	9994.000000	9994.000000
mean	55190.371023	229.858001	3.789574	0.156203	28.656896
std	32063.704510	623.245101	2.225110	0.206452	234.260108
min	1040.000000	0.444000	1.000000	0.000000	-6599.978000
25%	23223.000000	17.280000	2.000000	0.000000	1.728750
50%	56430.500000	54.490000	3.000000	0.200000	8.666500
75%	90008.000000	209.940000	5.000000	0.200000	29.364000
max	99301.000000	22638.480000	14.000000	0.800000	8399.976000

Statistik deskriptif: object

```
df.describe(include='O')
```

	Order_ID	Customer_ID	Product_ID	Category	Sub-Category	Product_Name	Ship_Mode	Customer_Name	Segment	Country/Region	City	State	Region
count	9994	9994	9994	9994	9994	9994	9994	9994	9994	9994	9994	9994	9994
unique	5009	793	1862	3	17	1817	4	793	3	1	531	49	4
top	CA-2020-100111	WB-21850	OFF-PA-10001970	Office Supplies	Binders	Staple envelope	Standard Class	William Brown	Consumer	United States	New York City	California	West
freq	14	37	19	6026	1523	48	5968	37	5191	9994	915	2001	3203

Summarize: Value counts

- Value counts digunakan terutama pada kolom non numerik untuk mengetahui banyaknya nilai per item
- Syntax: df[<nama_kolom>].value_counts()

```
df['Region'].value_counts()

West        3203
East        2848
Central     2323
South       1620
Name: Region, dtype: int64
```

Summarize: Group by

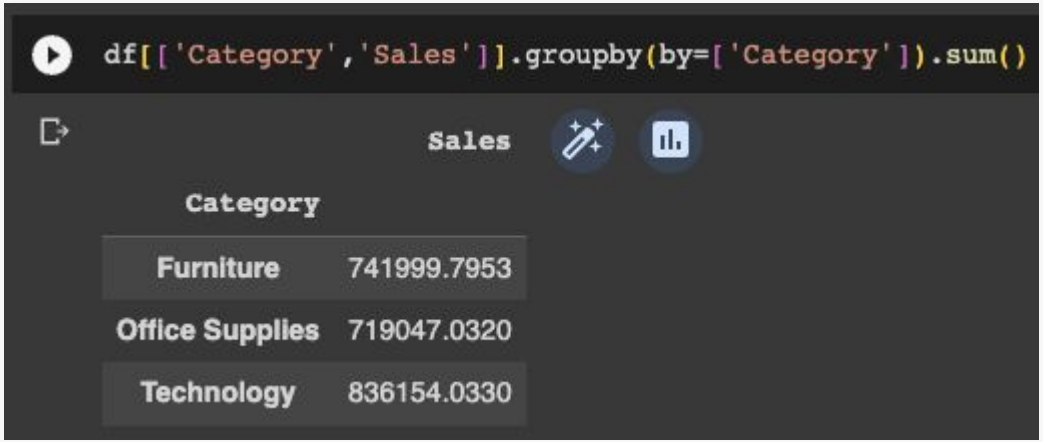
- Group by digunakan untuk melakukan perhitungan agregasi per kelompok pada kolom tertentu
- Syntax

  df.groupby(<kolom atau list kolom>).<fungsi_agregat>()

- Beberapa fungsi agregat antara lain: sum(), min(), max(), mean(), dst
- Nama kolom di dalam parameter groupby adalah nama kolom yang akan menjadi nama kelompok

Contoh: group by

- Buat tabel dengan kolom yang relevan: karena kita ingin menghitung jumlah sales per kategori maka kolom yang akan digunakan hanya ‘Category’ dan ‘Sales’
- Pada fungsi group by, isi parameter dengan ‘Category’ karena kita akan melakukan agregasi berdasarkan nilai-nilai di ‘Category’
- Kolom sisanya (‘Sales’) akan diagregasi dengan fungsi sum()



Group by: multikolom

```
[39] df[['Category','Sub-Category','Sales']].groupby(by=['Category','Sub-Category']).mean()
```

		Sales
Category	Sub-Category	
Furniture	Bookcases	503.859633
	Chairs	532.332420
	Furnishings	95.825668
	Tables	648.794771
Office Supplies	Appliances	230.755710
	Art	34.068834
	Binders	133.560560
	Envelopes	64.867724
	Fasteners	13.936774
	Labels	34.303055
	Paper	57.284092
	Storage	264.590553
	Supplies	245.650200
Technology	Accessories	215.974604
	Copiers	2198.941618
	Machines	1645.553313
	Phones	371.211534

Iteration

- Iterasi menggunakan dataframe dapat dilakukan dengan tiga cara: iterasi dengan kolom, index, dan baris per baris
- Untuk iterasi menggunakan kolom gunakan df.columns
- Untuk iterasi menggunakan index gunakan df.index
- Untuk iterasi menggunakan baris gunakan df.iterrows()

Iterasi dengan kolom

```
# iterasi dengan list of columns

# karena df.columns menghasilkan list of columns
# maka df.columns dapat digunakan sebagai iterator

# iterasi berdasarkan kolom berguna saat kita ingin
#mengaplikasikan operasi berulang terhadap kolom tertentu

for kolom in df.columns:
  print(kolom)
```

```
Order_ID
Customer_ID
Postal_Code
Product_ID
Sales
Quantity
Discount
Profit
Category
Sub-Category
Product_Name
Order_Date
Ship_Date
Ship_Mode
Customer_Name
Segment
Country/Region
City
State
Region
```

Iterasi dengan index

```
# disamping itu, iterasi juga dapat dilakukan terhadap index
df.index
```

```
RangeIndex(start=0, stop=9994, step=1)
```

```
[43] # contoh
     for idx in df.index:
       print(df['Order_ID'][idx], df['Order_Date'][idx])
```

```
CA-2017-106971 2017-09-02 00:00:00
CA-2020-123029 2020-09-30 00:00:00
CA-2019-139409 2019-09-05 00:00:00
US-2020-166688 2020-05-20 00:00:00
US-2020-166688 2020-05-20 00:00:00
US-2020-166688 2020-05-20 00:00:00
CA-2018-126970 2018-09-20 00:00:00
US-2019-165505 2019-01-23 00:00:00
US-2019-165505 2019-01-23 00:00:00
US-2019-165505 2019-01-23 00:00:00
US-2017-157070 2017-06-01 00:00:00
US-2017-157070 2017-06-01 00:00:00
US-2018-106873 2018-09-24 00:00:00
```

Iterasi baris per baris

```
# iterasi baris per baris dengan iterrows()

for index, row in df.iterrows():
  print(row['Order_ID'], row['Order_Date'])
```

```
CA-2019-161746 2019-10-21 00:00:00
CA-2019-161746 2019-10-21 00:00:00
CA-2017-114251 2017-11-05 00:00:00
CA-2017-114251 2017-11-05 00:00:00
CA-2017-114251 2017-11-05 00:00:00
CA-2017-114251 2017-11-05 00:00:00
CA-2017-114251 2017-11-05 00:00:00
CA-2019-116379 2019-11-07 00:00:00
US-2019-144477 2019-08-12 00:00:00
```

Export data

- Misalkan kita memiliki dataframe df_clean yang merupakan hasil processing dari dataframe df, untuk menyimpan df_clean menjadi file csv gunakan df.to_csv(<lokasi serta nama file>), contoh:

  df_clean.to_csv('df_clean.csv')

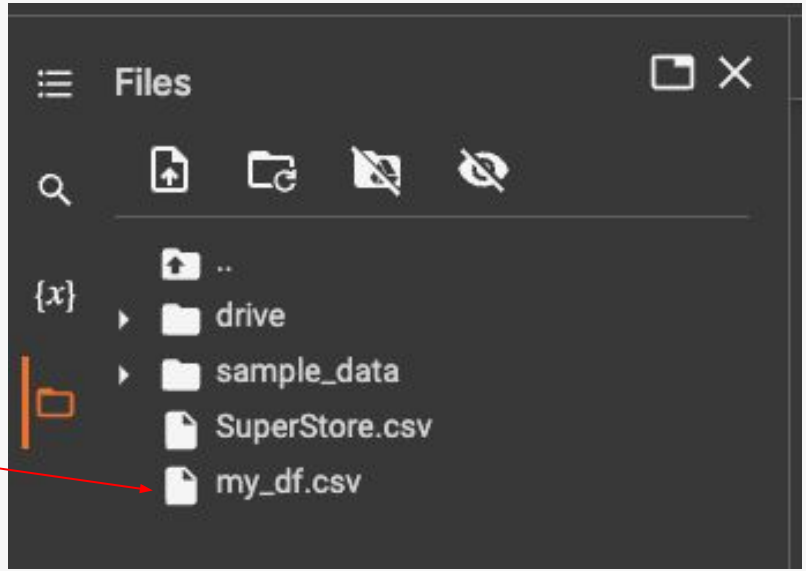
- Potongan kode di atas akan menyimpan df_clean menjadi df_clean.csv pada lokasi yang sama dengan script atau notebook

Membuat dataframe berisi nama konsumen di 10 baris pertama lalu menyimpannya menjadi my_df

```
[46] # export data

     my_df = df.loc[:10,['Customer_Name']]

     my_df.to_csv('my_df.csv')
```



DQLab

live Class

Terimakasih!

dqlab www.dqlab.id