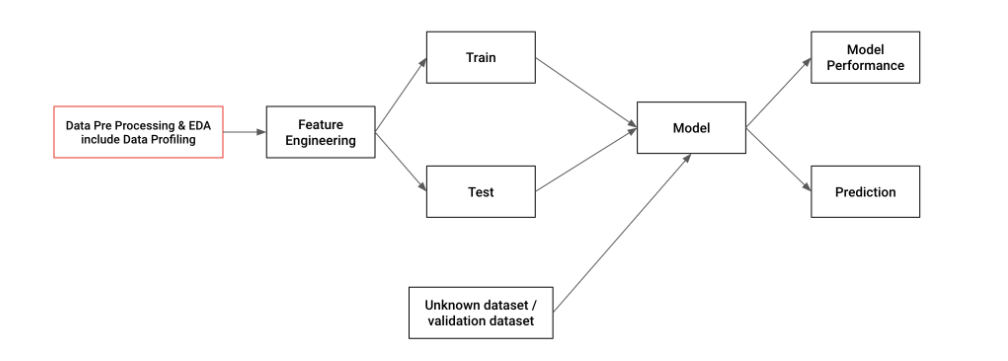
Data Quality with Python for Beginner

Apa itu Data Profiling?

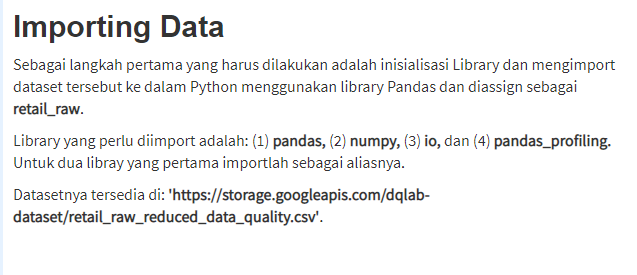
Pada bagian sebelumnya, aku sudah mempelajari mengenai exploratory data analysis, dimana Exploratory Data Analysis (EDA) adalah menggunakan pendekatan statistik yang bertujuan untuk menemukan dan meringkas sebuah dataset, mengetahui struktur dan hubungan antar variabel dalam dataset. EDA merupakan proses pre-analysis baik untuk descriptive analysis maupun predictive analysis.

Dalam bab ini, aku akan fokus pada satu aspek EDA, yaitu: **Data Profiling**!

Data profiling adalah kegiatan merangkum dataset menggunakan statistik deskriptif. Tujuan dari pembuatan data profiling adalah untuk memiliki pemahaman yang kuat tentang data sehingga dapat mulai menyusun framework analisis  dan memvisualisasikan data.



## Importing Data



import pandas as pd

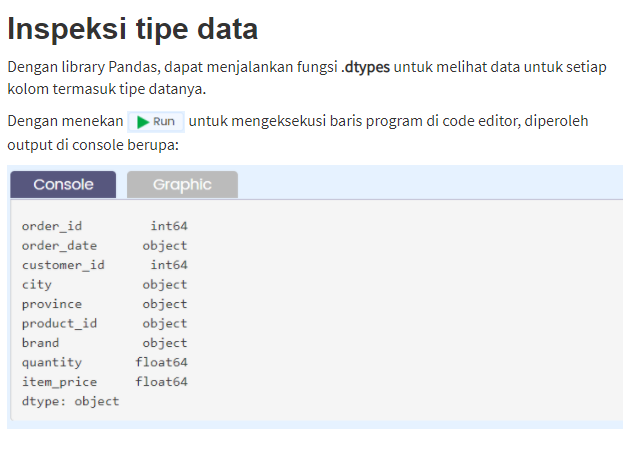
import numpy as np

import io

import pandas\_profiling

retail\_raw = pd.read\_csv('https://storage.googleapis.com/dqlab-dataset/retail\_raw\_reduced\_data\_quality.csv')

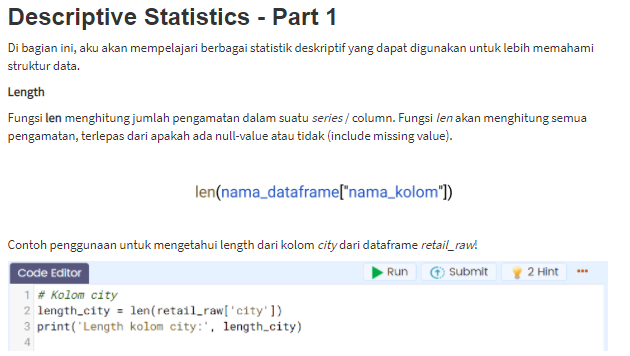
## Inspeksi tipe data

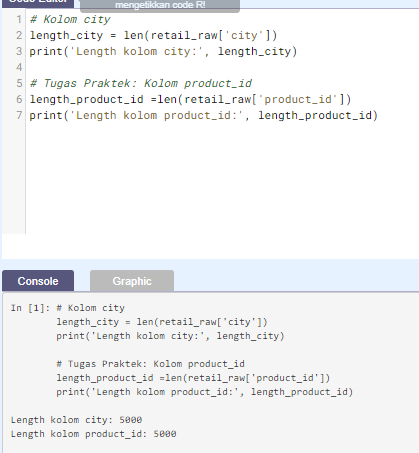
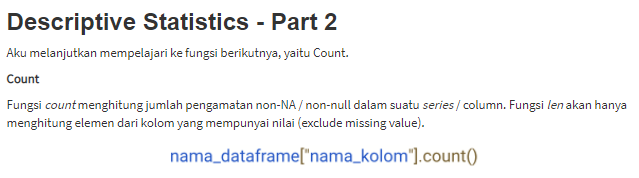
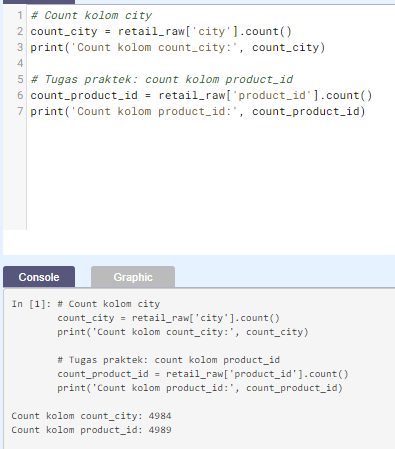


# Cetak tipe data di setiap kolom retail\_raw

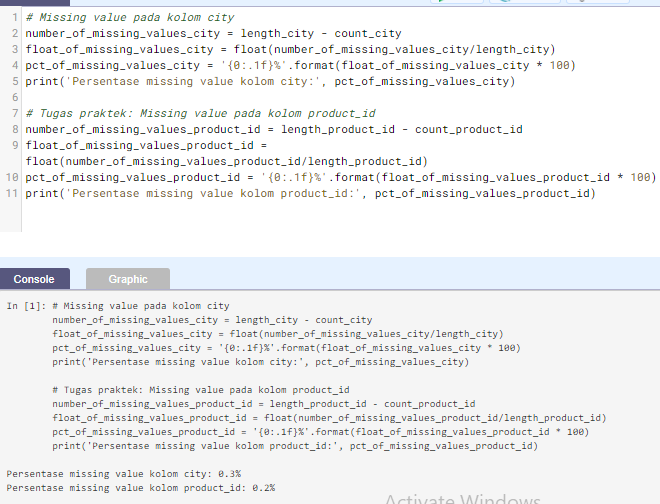
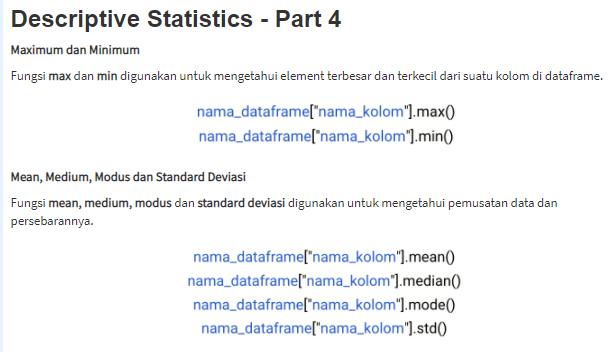
print(retail\_raw.dtypes)

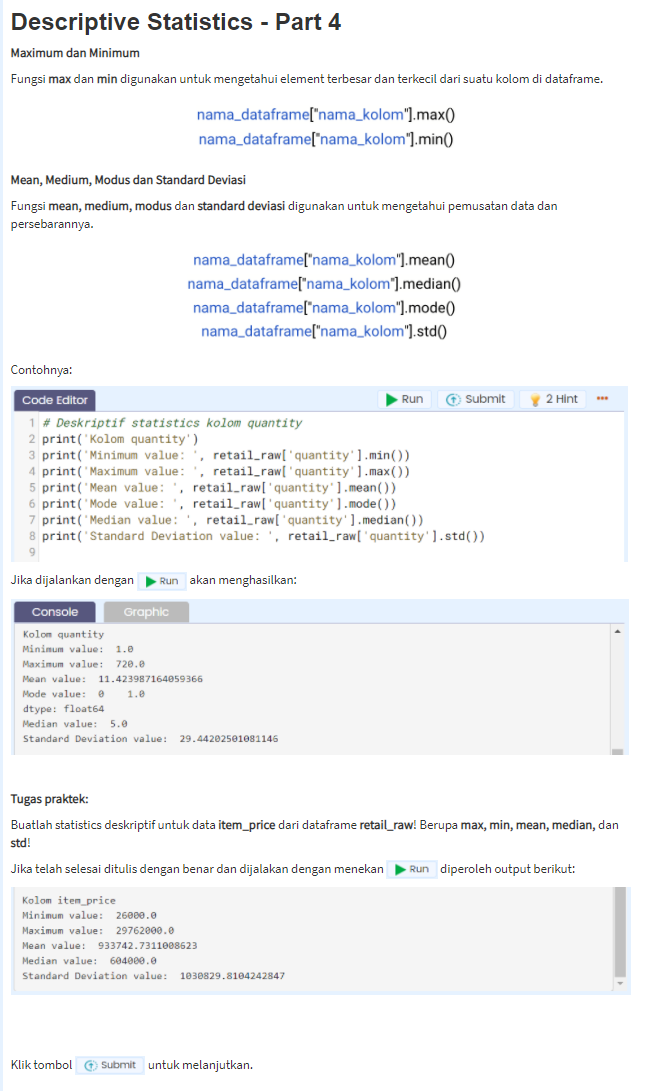
## Descriptive Statistics







# Deskriptif statistics kolom quantity

print('Kolom quantity')

print('Minimum value: ', retail\_raw['quantity'].min())

print('Maximum value: ', retail\_raw['quantity'].max())

print('Mean value: ', retail\_raw['quantity'].mean())

print('Mode value: ', retail\_raw['quantity'].mode())

print('Median value: ', retail\_raw['quantity'].median())

print('Standard Deviation value: ', retail\_raw['quantity'].std())

# Tugas praktek: Deskriptif statistics kolom item\_price

print('')

print('Kolom item\_price')

print('Minimum value: ', retail\_raw['item\_price'].min())

print('Maximum value: ', retail\_raw['item\_price'].max())

print('Mean value: ', retail\_raw['item\_price'].mean())

print('Median value: ', retail\_raw['item\_price'].median())

print('Standard Deviation value: ', retail\_raw['item\_price'].std())

Kolom quantity

Minimum value: 1.0

Maximum value: 720.0

Mean value: 11.423987164059366

Mode value: 0 1.0

dtype: float64

Median value: 5.0

Standard Deviation value: 29.44202501081146

Kolom item\_price

Minimum value: 26000.0

Maximum value: 29762000.0

Mean value: 933742.7311008623

Median value: 604000.0

Standard Deviation value: 1030829.8104242847



# Quantile statistics kolom quantity

print('Kolom quantity:')

print(retail\_raw['quantity'].quantile([0.25, 0.5, 0.75]))

# Tugas praktek: Quantile statistics kolom item\_price

print('')

print('Kolom item\_price:')

print(retail\_raw['item\_price'].quantile([0.25, 0.5, 0.75]))

Kolom quantity:

0.25 2.0

0.50 5.0

0.75 12.0

Name: quantity, dtype: float64

Kolom item\_price:

0.25 450000.0

0.50 604000.0

0.75 1045000.0

Name: item\_price, dtype: float64



print('Korelasi quantity dengan item\_price')

print(retail\_raw[['quantity', 'item\_price']].corr())

Korelasi quantity dengan item\_price

quantity item\_price

quantity 1.000000 -0.133936

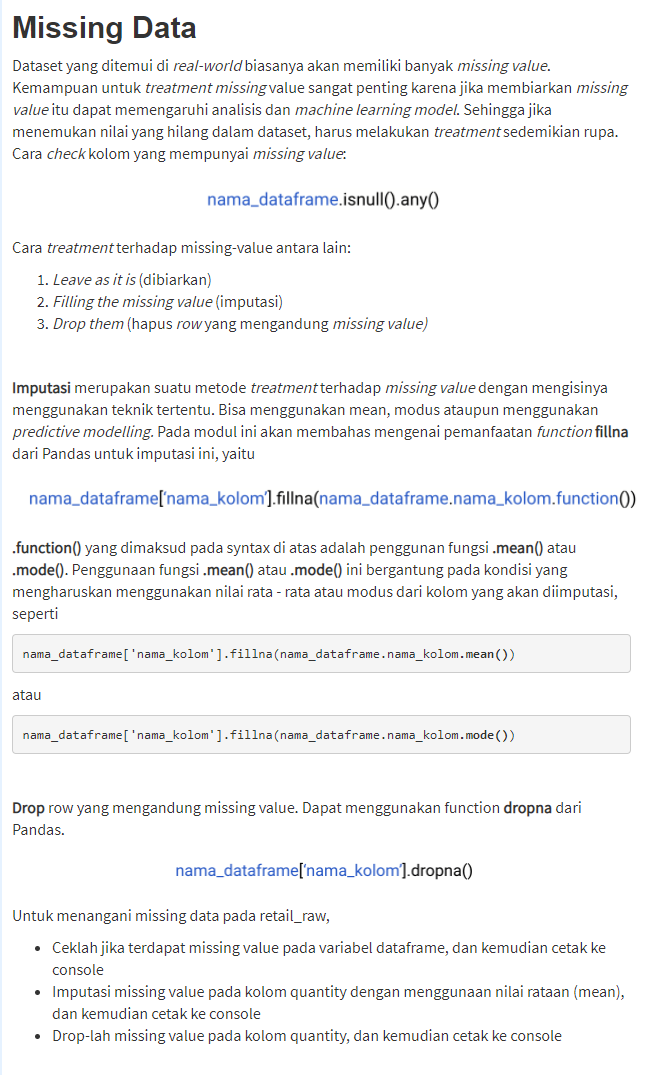
item\_price -0.133936 1.000000

[***Retail\_Profiling.html***](https://storage.googleapis.com/dqlab-image/html/Retail%20Profiling.html) ***>***  <https://storage.googleapis.com/dqlab-image/html/Retail%20Profiling.html>

## Data Cleansing

Data Cleansing berarti proses mengidentifikasi bagian data yang salah, tidak lengkap, tidak akurat, tidak relevan atau hilang dan kemudian memodifikasi, mengganti atau menghapusnya sesuai dengan kebutuhan. Data Cleansing dianggap sebagai elemen dasar dari Data Science

Pada bagian ini, akan membahas data cleansing dari treatment terhadap missing data, treatment outliers, sampai deduplikasi data.



# Check kolom yang memiliki missing data

print('Check kolom yang memiliki missing data:')

print(retail\_raw.isnull().any())

# Filling the missing value (imputasi)

print('\nFilling the missing value (imputasi):')

print(retail\_raw['quantity'].fillna(retail\_raw.quantity.mean()))

# Drop missing value

print('\nDrop missing value:')

print(retail\_raw['quantity'].dropna())

Check kolom yang memiliki missing data:

order\_id False

order\_date False

customer\_id False

city True

province True

product\_id True

brand False

quantity True

item\_price True

dtype: bool

Filling the missing value (imputasi):

0 10.000000

1 2.000000

2 8.000000

3 4.000000

4 2.000000

5 4.000000

6 1.000000

7 2.000000

8 32.000000

9 2.000000

10 6.000000

11 1.000000

12 3.000000

13 4.000000

14 11.423987

15 1.000000

16 10.000000

17 1.000000

18 4.000000

19 6.000000

20 12.000000

21 24.000000

22 10.000000

23 6.000000

24 12.000000

25 12.000000

26 96.000000

27 1.000000

28 10.000000

29 18.000000

...

4970 6.000000

4971 12.000000

4972 1.000000

4973 1.000000

4974 3.000000

4975 72.000000

4976 2.000000

4977 4.000000

4978 12.000000

4979 50.000000

4980 1.000000

4981 6.000000

4982 1.000000

4983 1.000000

4984 3.000000

4985 2.000000

4986 10.000000

4987 25.000000

4988 1.000000

4989 24.000000

4990 10.000000

4991 3.000000

4992 2.000000

4993 3.000000

4994 10.000000

4995 2.000000

4996 3.000000

4997 4.000000

4998 8.000000

4999 1.000000

Name: quantity, Length: 5000, dtype: float64

Drop missing value:

0 10.0

1 2.0

2 8.0

3 4.0

4 2.0

5 4.0

6 1.0

7 2.0

8 32.0

9 2.0

10 6.0

11 1.0

12 3.0

13 4.0

15 1.0

16 10.0

17 1.0

18 4.0

19 6.0

20 12.0

21 24.0

22 10.0

23 6.0

24 12.0

25 12.0

26 96.0

27 1.0

28 10.0

29 18.0

30 36.0

...

4970 6.0

4971 12.0

4972 1.0

4973 1.0

4974 3.0

4975 72.0

4976 2.0

4977 4.0

4978 12.0

4979 50.0

4980 1.0

4981 6.0

4982 1.0

4983 1.0

4984 3.0

4985 2.0

4986 10.0

4987 25.0

4988 1.0

4989 24.0

4990 10.0

4991 3.0

4992 2.0

4993 3.0

4994 10.0

4995 2.0

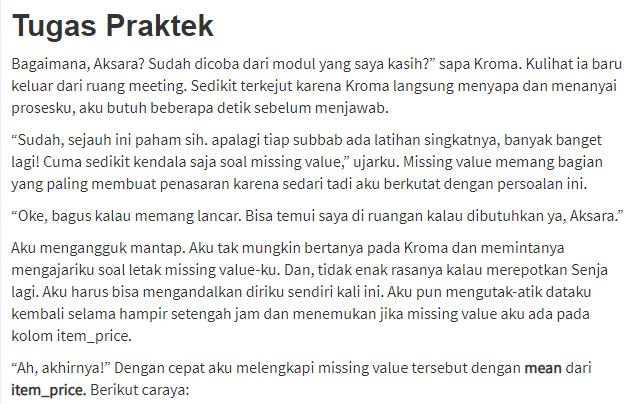
4996 3.0

4997 4.0

4998 8.0

4999 1.0

Name: quantity, Length: 4986, dtype: float64



print(retail\_raw['item\_price'].fillna(retail\_raw['item\_price'].mean()))

0 7.400000e+05

1 6.040000e+05

2 1.045000e+06

3 2.050000e+05

4 9.337427e+05

5 5.200000e+05

6 1.465000e+06

7 2.050000e+05

8 4.500000e+05

9 1.490000e+05

10 1.045000e+06

11 6.040000e+05

12 2.200000e+06

13 1.745000e+06

14 6.950000e+05

15 9.040000e+05

16 7.400000e+05

17 1.748000e+06

18 1.395000e+06

19 7.400000e+05

20 5.200000e+05

21 1.130000e+05

22 7.400000e+05

23 1.045000e+06

24 5.200000e+05

25 3.100000e+05

26 9.337427e+05

27 2.190000e+05

28 1.490000e+05

29 1.150000e+06

...

4970 2.095000e+06

4971 6.950000e+05

4972 3.100000e+05

4973 2.050000e+05

4974 5.945000e+06

4975 1.490000e+05

4976 1.500000e+06

4977 5.900000e+05

4978 1.045000e+06

4979 7.400000e+05

4980 5.945000e+06

4981 2.095000e+06

4982 1.745000e+06

4983 1.045000e+06

4984 5.900000e+05

4985 1.500000e+06

4986 5.900000e+05

4987 1.590000e+05

4988 1.185000e+06

4989 2.400000e+05

4990 7.400000e+05

4991 1.045000e+06

4992 3.100000e+05

4993 3.100000e+05

4994 7.400000e+05

4995 4.500000e+05

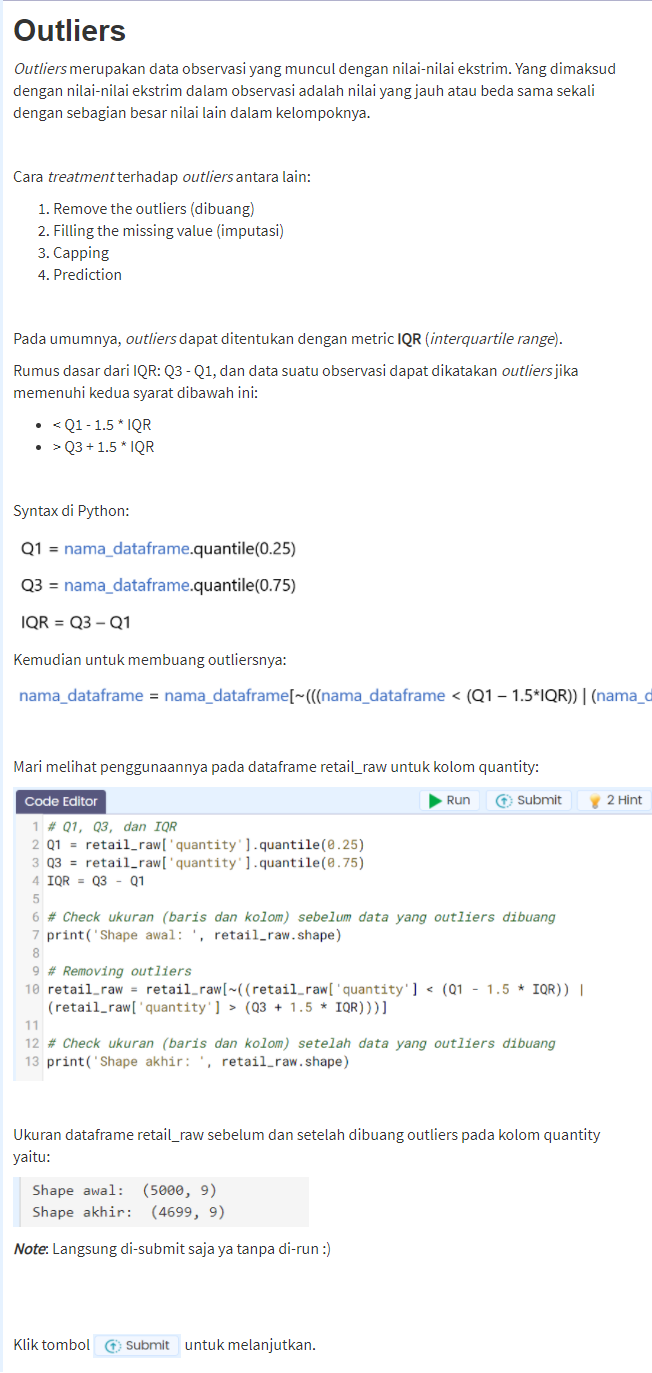
4996 1.465000e+06

4997 7.470000e+05

4998 6.950000e+05

4999 1.045000e+06

Name: item\_price, Length: 5000, dtype: float64



# Q1, Q3, dan IQR

Q1 = retail\_raw['quantity'].quantile(0.25)

Q3 = retail\_raw['quantity'].quantile(0.75)

IQR = Q3 - Q1

# Check ukuran (baris dan kolom) sebelum data yang outliers dibuang

print('Shape awal: ', retail\_raw .shape)

# Removing outliers

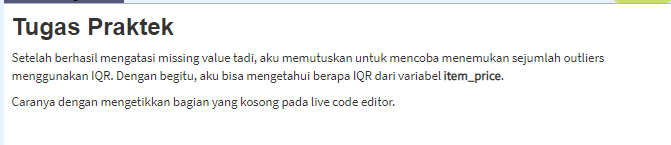
retail\_raw = retail\_raw[~((retail\_raw['quantity'] < (Q1 - 1.5 \* IQR)) | (retail\_raw['quantity'] > (Q3 + 1.5 \* IQR)))]

# Check ukuran (baris dan kolom) setelah data yang outliers dibuang

print('Shape akhir: ', retail\_raw.shape)

Shape awal: (14, 9)

Shape akhir: (14, 9)



# Q1, Q3, dan IQR

Q1 = retail\_raw['item\_price'].quantile(0.25)

Q3 = retail\_raw['item\_price'].quantile(0.75)

IQR = Q3 - Q1

# Check ukuran (baris dan kolom) sebelum data yang outliers dibuang

print('Shape awal: ', retail\_raw.shape)

# Removing outliers

retail\_raw = retail\_raw[~((retail\_raw['item\_price'] < (Q1 - 1.5 \* IQR)) | (retail\_raw['item\_price'] > (Q3 + 1.5 \* IQR)))]

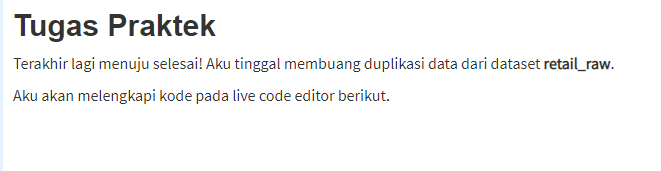
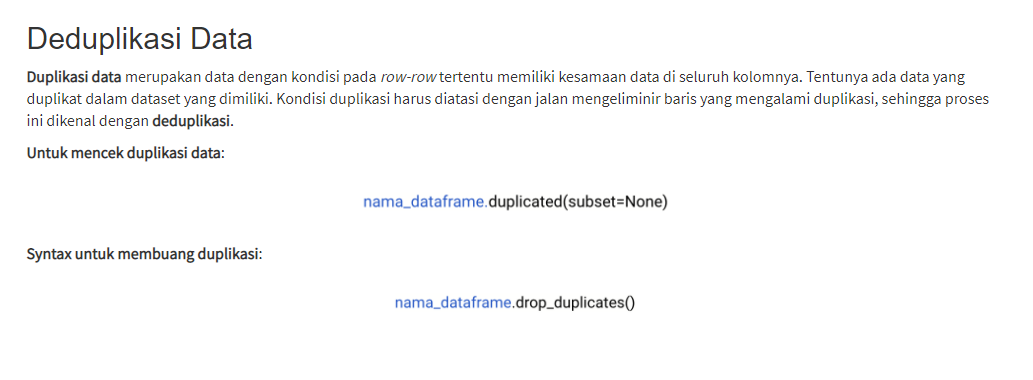
# Check ukuran (baris dan kolom) setelah data yang outliers dibuang

print('Shape akhir: ', retail\_raw.shape)

Shape awal: (14, 9)

Shape akhir: (12, 9)

## Deduplikasi Data



# Check ukuran (baris dan kolom) sebelum data duplikasi dibuang

print('Shape awal: ', retail\_raw.shape)

# Buang data yang terduplikasi

retail\_raw.drop\_duplicates(inplace=True)

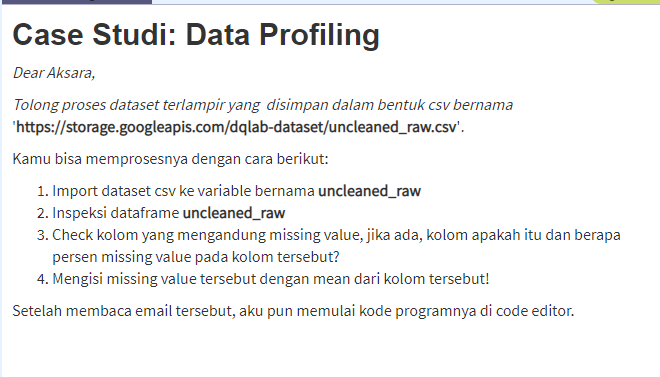
# Check ukuran (baris dan kolom) setelah data duplikasi dibuang

print('Shape akhir: ', retail\_raw.shape)

Shape awal: (12, 9)

Shape akhir: (12, 9)

Mini Project



# Baca dataset uncleaned\_raw.csv

uncleaned\_raw = pd.read\_csv('https://storage.googleapis.com/dqlab-dataset/uncleaned\_raw.csv')

#inspeksi dataframe uncleaned\_raw

print('Lima data teratas:')

print(uncleaned\_raw.head())

#Check kolom yang mengandung missing value

print('\nKolom dengan missing value:')

print(uncleaned\_raw.isnull().any())

#Persentase missing value

length\_qty = len(uncleaned\_raw['Quantity'])

count\_qty = uncleaned\_raw['Quantity'].count()

#mengurangi length dengan count

number\_of\_missing\_values\_qty = length\_qty - count\_qty

#mengubah ke bentuk float

float\_of\_missing\_values\_qty = float(number\_of\_missing\_values\_qty / length\_qty)

#mengubah ke dalam bentuk persen

pct\_of\_missing\_values\_qty = '{0:.1f}%'.format(float\_of\_missing\_values\_qty \* 100)

#print hasil percent dari missing value

print('Persentase missing value kolom Quantity:', pct\_of\_missing\_values\_qty)

#Mengisi missing value tersebut dengan mean dari kolom tersebut

uncleaned\_raw['Quantity'] = uncleaned\_raw['Quantity'].fillna(uncleaned\_raw['Quantity'].mean())

Lima data teratas:

InvoiceNo Description ... CustomerID City

0 536365 WHITE HANGING HEART T-LIGHT HOLDER ... 17850 Surabaya

1 536366 WHITE METAL LANTERN ... 17850 Surabaya

2 536367 CREAM CUPID HEARTS COAT HANGER ... 17850 Surabaya

3 536368 KNITTED UNION FLAG HOT WATER BOTTLE ... 17850 Jakarta

4 536369 RED WOOLLY HOTTIE WHITE HEART. ... 17850 Medan

[5 rows x 7 columns]

Kolom dengan missing value:

InvoiceNo False

Description False

Quantity True

InvoiceDate False

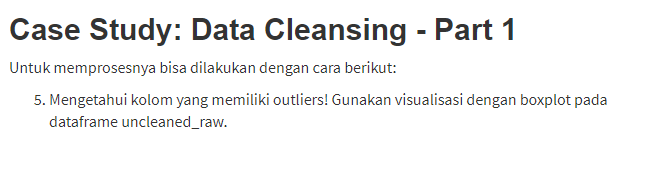
UnitPrice False

CustomerID False

City False

dtype: bool

Persentase missing value kolom Quantity: 4.0%



import matplotlib.pyplot as plt

#Mengetahui kolom yang memiliki outliers!

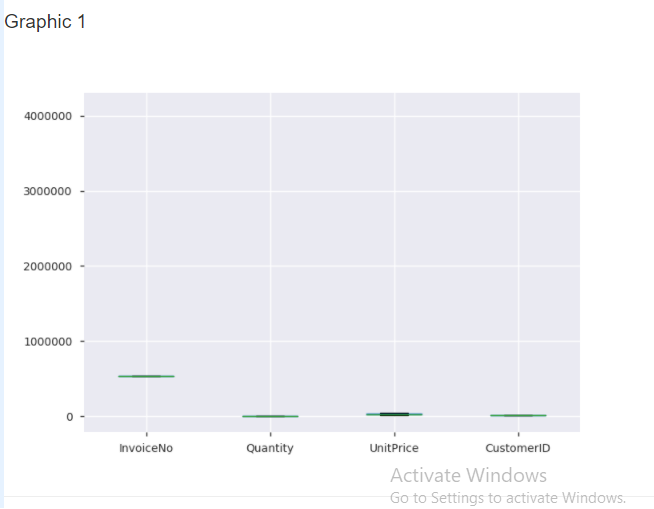
uncleaned\_raw.boxplot()

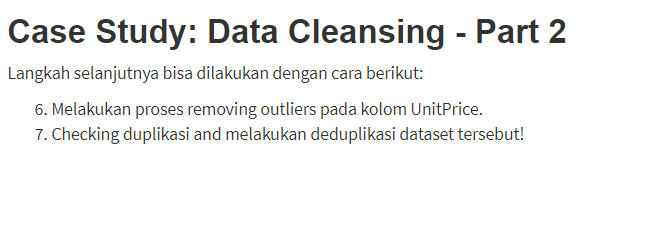
plt.show()

Refrence:

https://matplotlib.org/stable/api/\_as\_gen/matplotlib.pyplot.html

https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.head.html





#Check IQR

Q1 = uncleaned\_raw['UnitPrice'].quantile(0.25)

Q3 = uncleaned\_raw['UnitPrice'].quantile(0.75)

IQR = Q3 - Q1

#removing outliers

uncleaned\_raw = uncleaned\_raw[~((uncleaned\_raw[['UnitPrice']] < (Q1 - 1.5 \* IQR)) | (uncleaned\_raw[['UnitPrice']] > (Q3 + 1.5 \* IQR)))]

#check for duplication

print(uncleaned\_raw.duplicated(subset=None))

#remove duplication

uncleaned\_raw = uncleaned\_raw.drop\_duplicates()

0 False

1 False

2 False

3 False

4 False

5 False

6 False

7 False

8 False

9 False

10 False

11 False

12 False

13 False

14 False

15 True

16 False

17 True

18 True

19 True

20 False

21 False

22 True

23 False

24 False

25 True

26 True

27 False

28 True

29 True

...

475 True

476 True

477 True

478 True

479 True

480 True

481 True

482 True

483 True

484 True

485 True

486 True

487 True

488 True

489 True

490 True

491 True

492 True

493 True

494 True

495 True

496 True

497 True

498 True

499 True

500 True

501 True

502 True

503 True

504 True

Length: 505, dtype: bool

