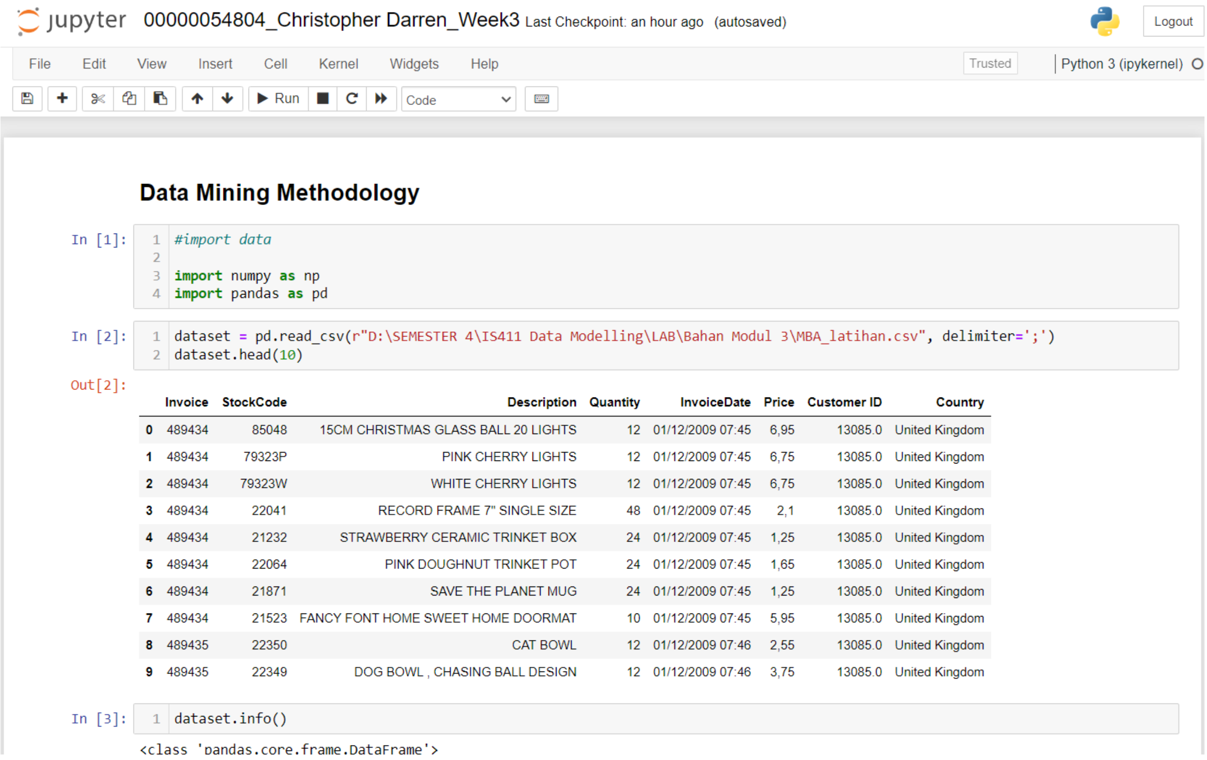
Tugas LAB WEEK 3-ASYNCHRON Christopher Darren



Data Mining Methodology

In [1]:

```
#import data

import numpy as np
import pandas as pd
```

In [2]:

```
dataset = pd.read_csv(r"D:\SEMESTER 4\IS411 Data Modelling\LAB\Bahan Modul 3\MBA_latihan.csv", delimiter=';')
dataset.head(10)
```

Out[2]:

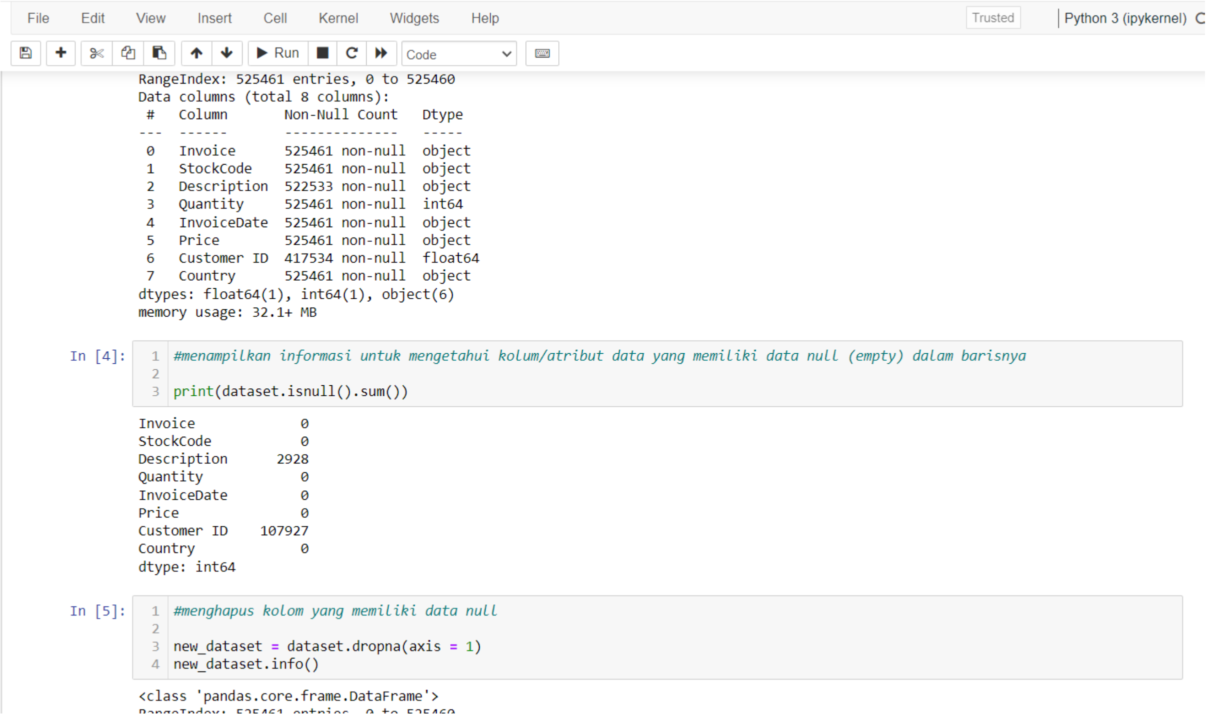
	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	01/12/2009 07:45	6,95	13085.0	United Kingdom
1	489434	79323P	PINK CHERRY LIGHTS	12	01/12/2009 07:45	6,75	13085.0	United Kingdom
2	489434	79323W	WHITE CHERRY LIGHTS	12	01/12/2009 07:45	6,75	13085.0	United Kingdom
3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	01/12/2009 07:45	2,1	13085.0	United Kingdom
4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	01/12/2009 07:45	1,25	13085.0	United Kingdom
5	489434	22064	PINK DOUGHNUT TRINKET POT	24	01/12/2009 07:45	1,65	13085.0	United Kingdom
6	489434	21871	SAVE THE PLANET MUG	24	01/12/2009 07:45	1,25	13085.0	United Kingdom
7	489434	21523	FANCY FONT HOME SWEET HOME DOORMAT	10	01/12/2009 07:45	5,95	13085.0	United Kingdom
8	489435	22350	CAT BOWL	12	01/12/2009 07:46	2,55	13085.0	United Kingdom
9	489435	22349	DOG BOWL , CHASING BALL DESIGN	12	01/12/2009 07:46	3,75	13085.0	United Kingdom

In [3]:

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

Gambar 1.



```
RangeIndex: 525461 entries, 0 to 525460
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   Invoice      525461 non-null  object 
 1   StockCode    525461 non-null  object 
 2   Description  522533 non-null  object 
 3   Quantity     525461 non-null  int64  
 4   InvoiceDate  525461 non-null  object 
 5   Price        525461 non-null  object 
 6   Customer ID  417534 non-null  float64
 7   Country      525461 non-null  object 
dtypes: float64(1), int64(1), object(6)
memory usage: 32.1+ MB
```

In [4]:

```
#menampilkan informasi untuk mengetahui kolum/atribut data yang memiliki data null (empty) dalam barisnya

print(dataset.isnull().sum())
```

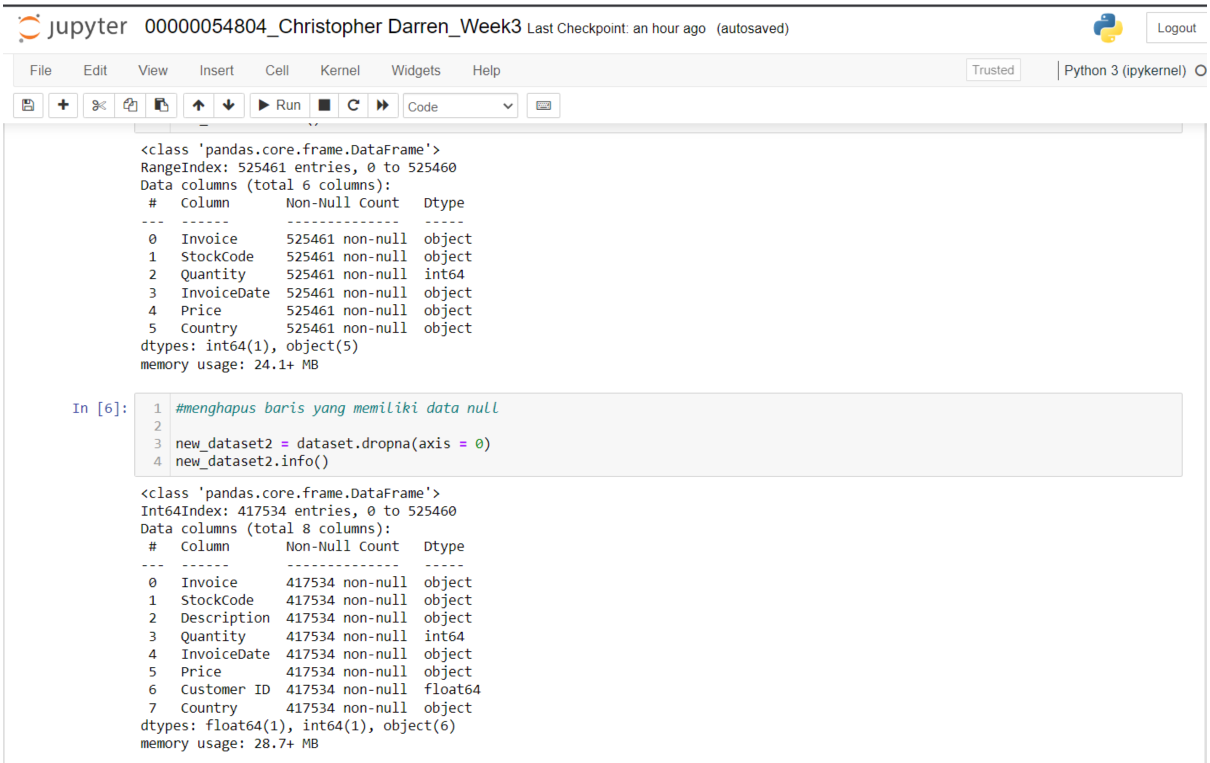
```
Invoice             0
StockCode           0
Description      2928
Quantity            0
InvoiceDate         0
Price               0
Customer ID    107927
Country             0
dtype: int64
```

In [5]:

```
#menghapus kolom yang memiliki data null

new_dataset = dataset.dropna(axis = 1)
new_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525461 entries, 0 to 525460
```

Gambar 2.



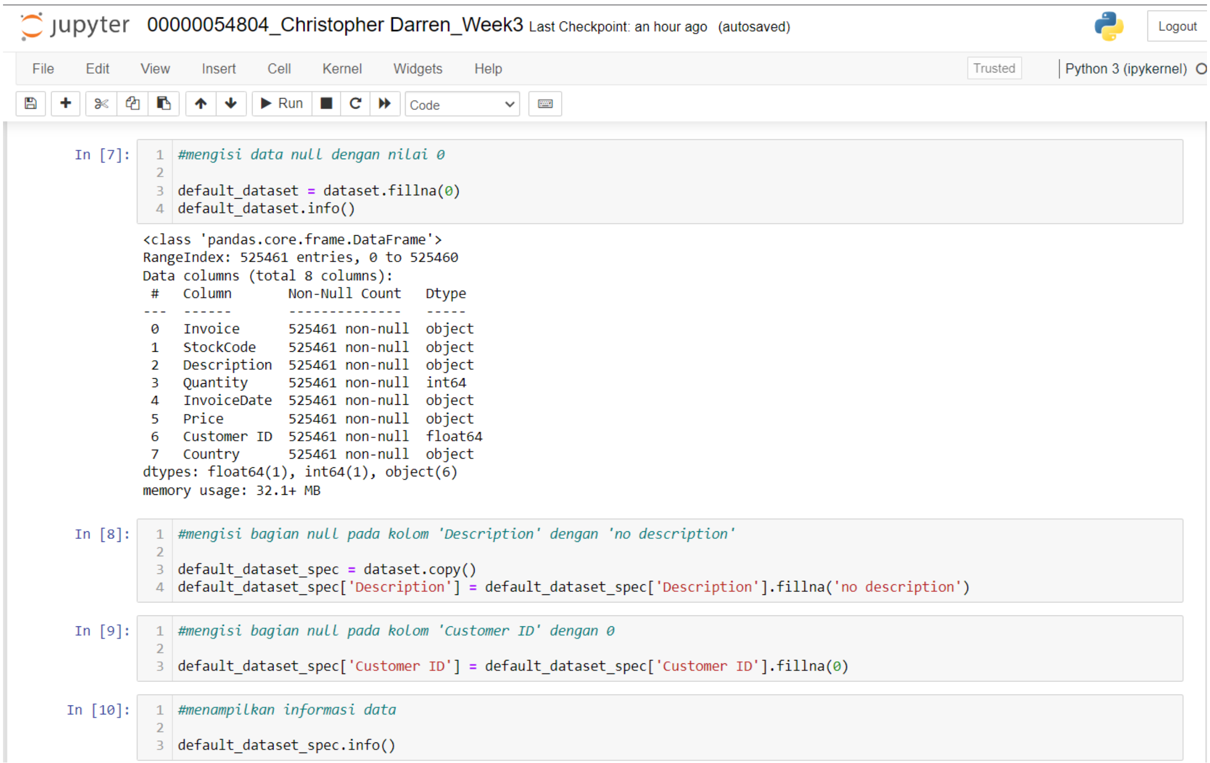
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525461 entries, 0 to 525460
Data columns (total 6 columns):
 #   Column       Non-Null Count   Dtype 
---  ------       --------------   ----- 
 0   Invoice      525461 non-null  object
 1   StockCode    525461 non-null  object
 2   Quantity     525461 non-null  int64 
 3   InvoiceDate  525461 non-null  object
 4   Price        525461 non-null  object
 5   Country      525461 non-null  object
dtypes: int64(1), object(5)
memory usage: 24.1+ MB
```

In [6]:

```
#menghapus baris yang memiliki data null

new_dataset2 = dataset.dropna(axis = 0)
new_dataset2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 417534 entries, 0 to 525460
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   Invoice      417534 non-null  object 
 1   StockCode    417534 non-null  object 
 2   Description  417534 non-null  object 
 3   Quantity     417534 non-null  int64  
 4   InvoiceDate  417534 non-null  object 
 5   Price        417534 non-null  object 
 6   Customer ID  417534 non-null  float64
 7   Country      417534 non-null  object 
dtypes: float64(1), int64(1), object(6)
memory usage: 28.7+ MB
```

Gambar 7.



In [7]:

```
#mengisi data null dengan nilai 0

default_dataset = dataset.fillna(0)
default_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525461 entries, 0 to 525460
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   Invoice      525461 non-null  object 
 1   StockCode    525461 non-null  object 
 2   Description  525461 non-null  object 
 3   Quantity     525461 non-null  int64  
 4   InvoiceDate  525461 non-null  object 
 5   Price        525461 non-null  object 
 6   Customer ID  525461 non-null  float64
 7   Country      525461 non-null  object 
dtypes: float64(1), int64(1), object(6)
memory usage: 32.1+ MB
```

In [8]:

```
#mengisi bagian null pada kolom 'Description' dengan 'no description'

default_dataset_spec = dataset.copy()
default_dataset_spec['Description'] = default_dataset_spec['Description'].fillna('no description')
```

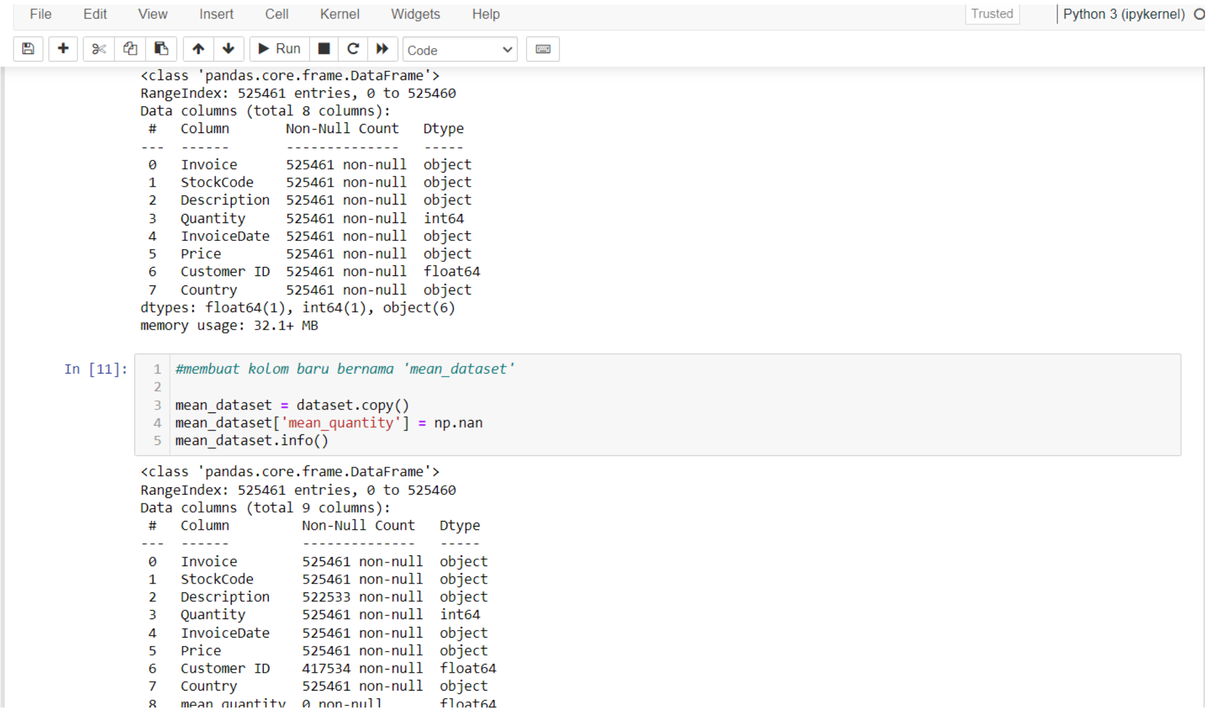
In [9]:

```
#mengisi bagian null pada kolom 'Customer ID' dengan 0

default_dataset_spec['Customer ID'] = default_dataset_spec['Customer ID'].fillna(0)
```

In [10]:

```
#menampilkan informasi data

default_dataset_spec.info()
```

Gambar 8.



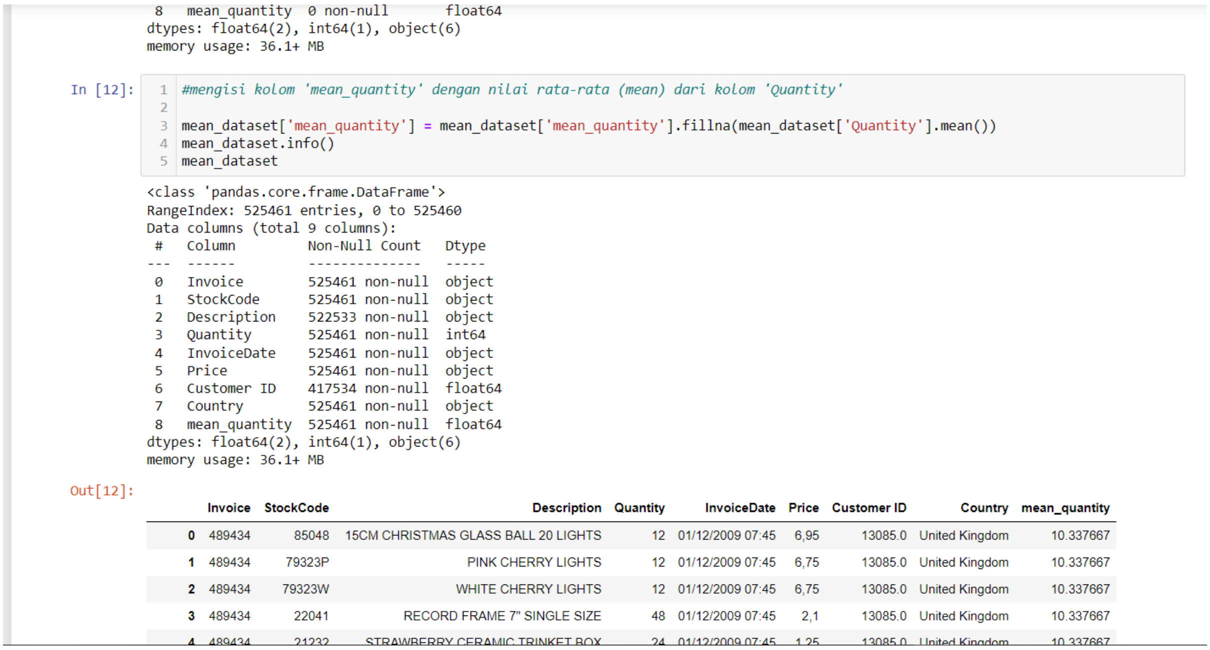
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525461 entries, 0 to 525460
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Invoice      525461 non-null  object
 1   StockCode    525461 non-null  object
 2   Description  525461 non-null  object
 3   Quantity     525461 non-null  int64
 4   InvoiceDate  525461 non-null  object
 5   Price        525461 non-null  object
 6   Customer ID  525461 non-null  float64
 7   Country      525461 non-null  object
dtypes: float64(1), int64(1), object(6)
memory usage: 32.1+ MB
```

In [11]:

```
#membuat kolom baru bernama 'mean_dataset'

mean_dataset = dataset.copy()
mean_dataset['mean_quantity'] = np.nan
mean_dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525461 entries, 0 to 525460
Data columns (total 9 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   Invoice        525461 non-null  object
 1   StockCode      525461 non-null  object
 2   Description    522533 non-null  object
 3   Quantity       525461 non-null  int64
 4   InvoiceDate    525461 non-null  object
 5   Price          525461 non-null  object
 6   Customer ID    417534 non-null  float64
 7   Country        525461 non-null  object
 8   mean_quantity  0 non-null       float64
```

Gambar 9.



```
 8   mean_quantity  0 non-null       float64
dtypes: float64(2), int64(1), object(6)
memory usage: 36.1+ MB
```

In [12]:

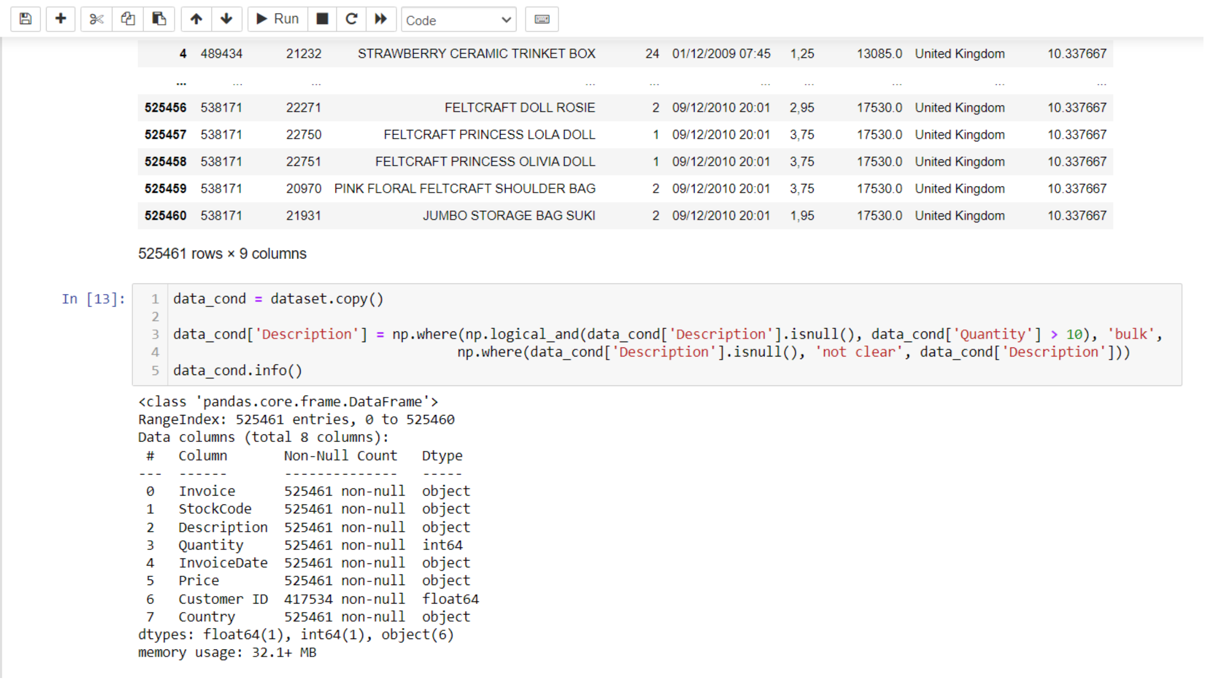
```
#mengisi kolom 'mean_quantity' dengan nilai rata-rata (mean) dari kolom 'Quantity'

mean_dataset['mean_quantity'] = mean_dataset['mean_quantity'].fillna(mean_dataset['Quantity'].mean())
mean_dataset.info()
mean_dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525461 entries, 0 to 525460
Data columns (total 9 columns):
 #   Column         Non-Null Count   Dtype
---  ------         --------------   -----
 0   Invoice        525461 non-null  object
 1   StockCode      525461 non-null  object
 2   Description    522533 non-null  object
 3   Quantity       525461 non-null  int64
 4   InvoiceDate    525461 non-null  object
 5   Price          525461 non-null  object
 6   Customer ID    417534 non-null  float64
 7   Country        525461 non-null  object
 8   mean_quantity  525461 non-null  float64
dtypes: float64(2), int64(1), object(6)
memory usage: 36.1+ MB
```

Out[12]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country	mean_quantity
0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	01/12/2009 07:45	6,95	13085.0	United Kingdom	10.337667
1	489434	79323P	PINK CHERRY LIGHTS	12	01/12/2009 07:45	6,75	13085.0	United Kingdom	10.337667
2	489434	79323W	WHITE CHERRY LIGHTS	12	01/12/2009 07:45	6,75	13085.0	United Kingdom	10.337667
3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	01/12/2009 07:45	2,1	13085.0	United Kingdom	10.337667
4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	01/12/2009 07:45	1,25	13085.0	United Kingdom	10.337667

Gambar 10.



4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	01/12/2009 07:45	1,25	13085.0	United Kingdom	10.337667
...	...	...	...	...	...	...	...	...	...
525456	538171	22271	FELTCRAFT DOLL ROSIE	2	09/12/2010 20:01	2,95	17530.0	United Kingdom	10.337667
525457	538171	22750	FELTCRAFT PRINCESS LOLA DOLL	1	09/12/2010 20:01	3,75	17530.0	United Kingdom	10.337667
525458	538171	22751	FELTCRAFT PRINCESS OLIVIA DOLL	1	09/12/2010 20:01	3,75	17530.0	United Kingdom	10.337667
525459	538171	20970	PINK FLORAL FELTCRAFT SHOULDER BAG	2	09/12/2010 20:01	3,75	17530.0	United Kingdom	10.337667
525460	538171	21931	JUMBO STORAGE BAG SUKI	2	09/12/2010 20:01	1,95	17530.0	United Kingdom	10.337667

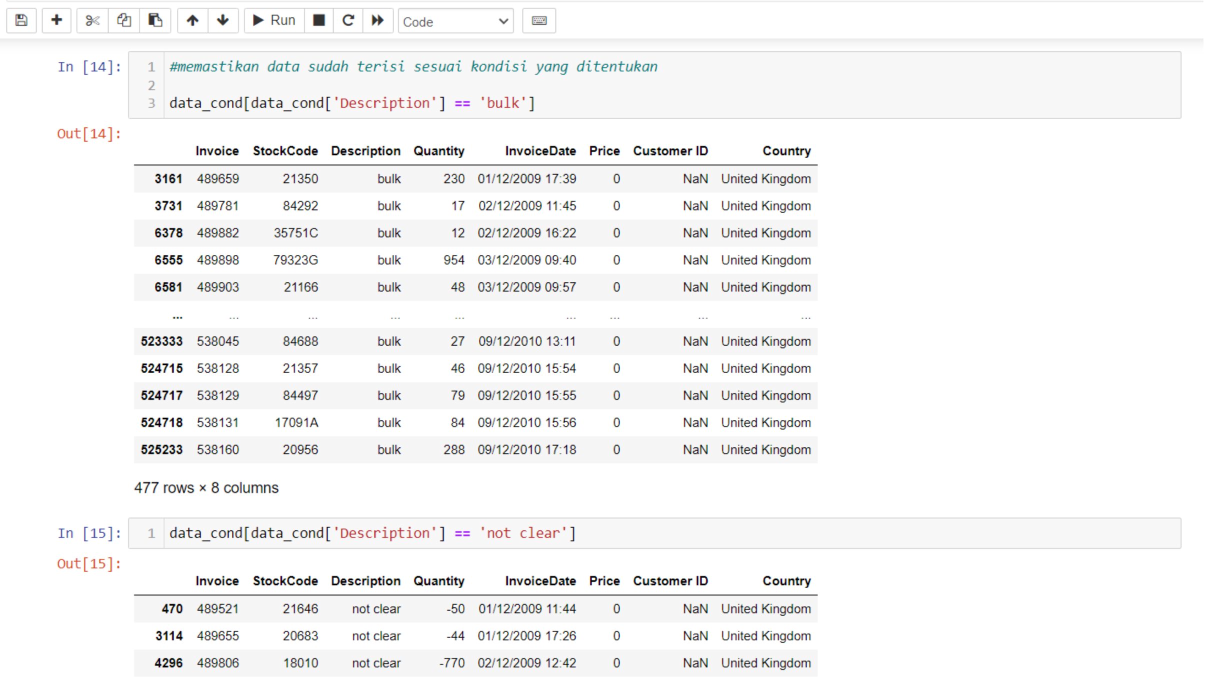
525461 rows × 9 columns

In [13]:

```
data_cond = dataset.copy()

data_cond['Description'] = np.where(np.logical_and(data_cond['Description'].isnull(), data_cond['Quantity'] > 10), 'bulk',
                                    np.where(data_cond['Description'].isnull(), 'not clear', data_cond['Description']))
data_cond.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525461 entries, 0 to 525460
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype  
---  ------       --------------   -----  
 0   Invoice      525461 non-null  object 
 1   StockCode    525461 non-null  object 
 2   Description  525461 non-null  object 
 3   Quantity     525461 non-null  int64  
 4   InvoiceDate  525461 non-null  object 
 5   Price        525461 non-null  object 
 6   Customer ID  417534 non-null  float64
 7   Country      525461 non-null  object 
dtypes: float64(1), int64(1), object(6)
memory usage: 32.1+ MB
```

Gambar 11.



In [14]:

```
#memastikan data sudah terisi sesuai kondisi yang ditentukan

data_cond[data_cond['Description'] == 'bulk']
```

Out[14]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
3161	489659	21350	bulk	230	01/12/2009 17:39	0	NaN	United Kingdom
3731	489781	84292	bulk	17	02/12/2009 11:45	0	NaN	United Kingdom
6378	489882	35751C	bulk	12	02/12/2009 16:22	0	NaN	United Kingdom
6555	489898	79323G	bulk	954	03/12/2009 09:40	0	NaN	United Kingdom
6581	489903	21166	bulk	48	03/12/2009 09:57	0	NaN	United Kingdom
...	...	...	...	...	...	...	...	...
523333	538045	84688	bulk	27	09/12/2010 13:11	0	NaN	United Kingdom
524715	538128	21357	bulk	46	09/12/2010 15:54	0	NaN	United Kingdom
524717	538129	84497	bulk	79	09/12/2010 15:55	0	NaN	United Kingdom
524718	538131	17091A	bulk	84	09/12/2010 15:56	0	NaN	United Kingdom
525233	538160	20956	bulk	288	09/12/2010 17:18	0	NaN	United Kingdom

477 rows × 8 columns

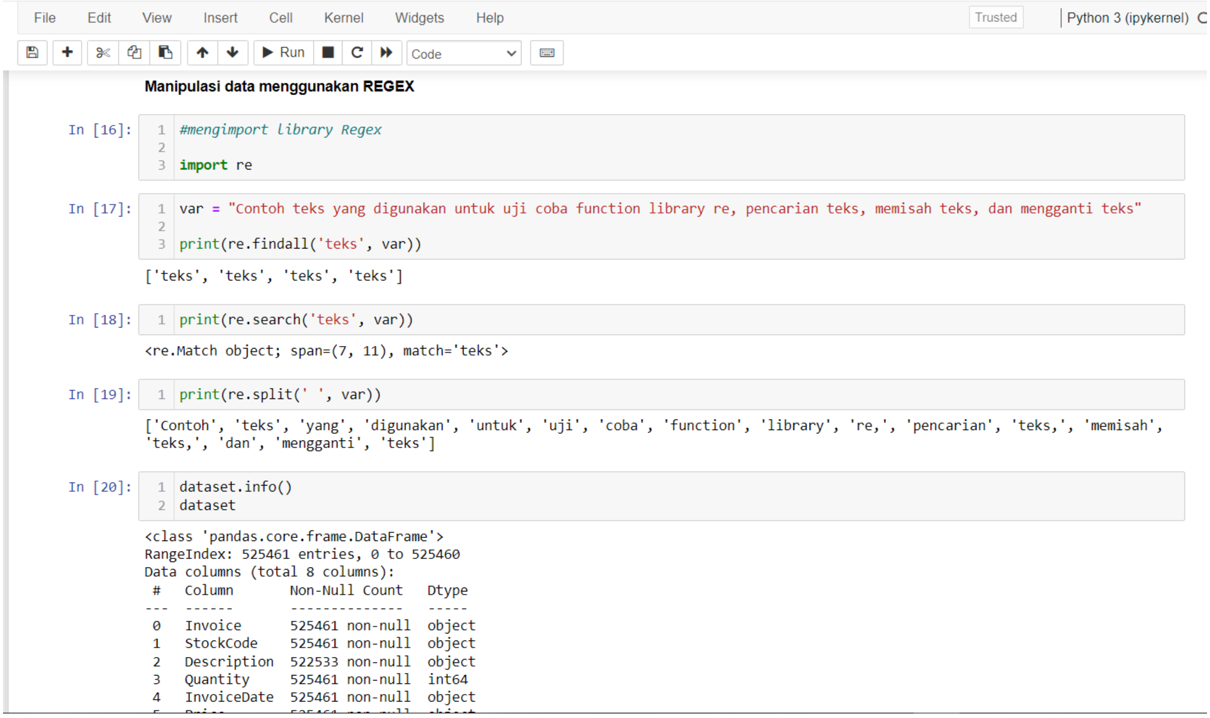
In [15]:

```
data_cond[data_cond['Description'] == 'not clear']
```

Out[15]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
470	489521	21646	not clear	-50	01/12/2009 11:44	0	NaN	United Kingdom
3114	489655	20683	not clear	-44	01/12/2009 17:26	0	NaN	United Kingdom
4296	489806	18010	not clear	-770	02/12/2009 12:42	0	NaN	United Kingdom

Gambar 12.



Manipulasi data menggunakan REGEX

In [16]:
```
#mengimport library Regex

import re
```

In [17]:
```
var = "Contoh teks yang digunakan untuk uji coba function library re, pencarian teks, memisah teks, dan mengganti teks"

print(re.findall('teks', var))
```

```
['teks', 'teks', 'teks', 'teks']
```

In [18]:
```
print(re.search('teks', var))
```

```
<re.Match object; span=(7, 11), match='teks'>
```

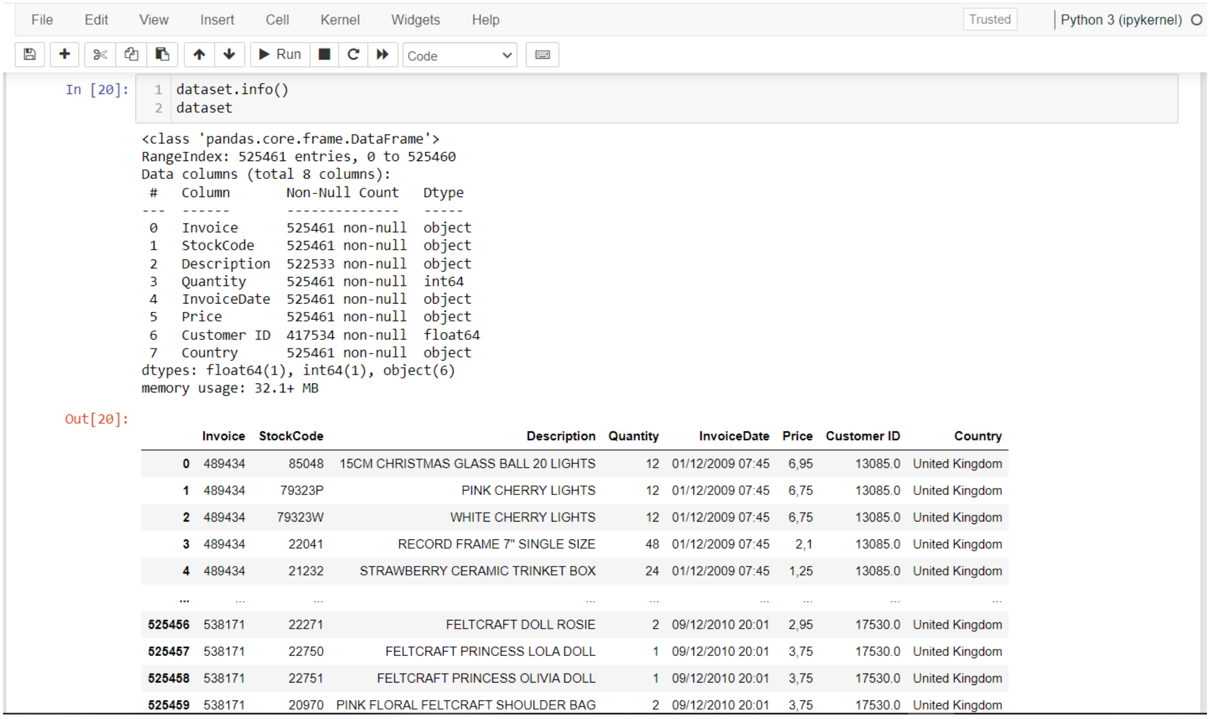
In [19]:
```
print(re.split(' ', var))
```

```
['Contoh', 'teks', 'yang', 'digunakan', 'untuk', 'uji', 'coba', 'function', 'library', 're,', 'pencarian', 'teks,', 'memisah',
'teks,', 'dan', 'mengganti', 'teks']
```

In [20]:
```
dataset.info()
dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525461 entries, 0 to 525460
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Invoice      525461 non-null  object
 1   StockCode    525461 non-null  object
 2   Description  522533 non-null  object
 3   Quantity     525461 non-null  int64
 4   InvoiceDate  525461 non-null  object
```

Gambar 13.



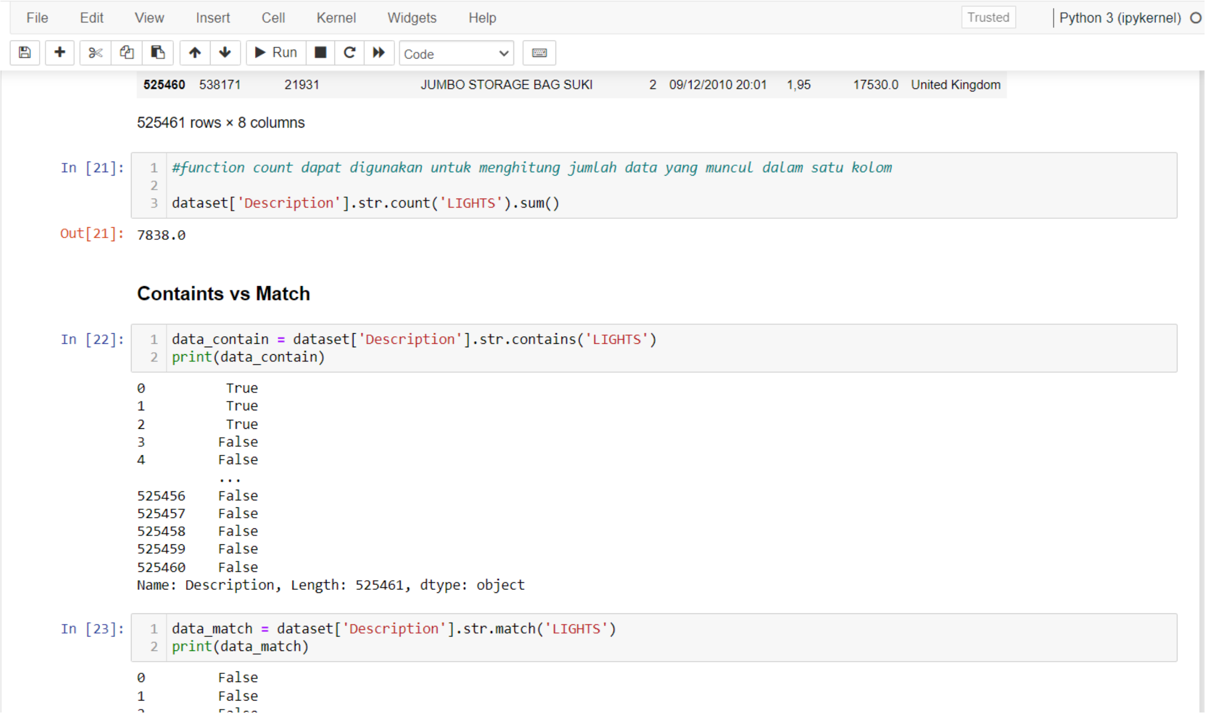
In [20]:
```
dataset.info()
dataset
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 525461 entries, 0 to 525460
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype
---  ------       --------------   -----
 0   Invoice      525461 non-null  object
 1   StockCode    525461 non-null  object
 2   Description  522533 non-null  object
 3   Quantity     525461 non-null  int64
 4   InvoiceDate  525461 non-null  object
 5   Price        525461 non-null  object
 6   Customer ID  417534 non-null  float64
 7   Country      525461 non-null  object
dtypes: float64(1), int64(1), object(6)
memory usage: 32.1+ MB
```

Out[20]:

	Invoice	StockCode	Description	Quantity	InvoiceDate	Price	Customer ID	Country
0	489434	85048	15CM CHRISTMAS GLASS BALL 20 LIGHTS	12	01/12/2009 07:45	6,95	13085.0	United Kingdom
1	489434	79323P	PINK CHERRY LIGHTS	12	01/12/2009 07:45	6,75	13085.0	United Kingdom
2	489434	79323W	WHITE CHERRY LIGHTS	12	01/12/2009 07:45	6,75	13085.0	United Kingdom
3	489434	22041	RECORD FRAME 7" SINGLE SIZE	48	01/12/2009 07:45	2,1	13085.0	United Kingdom
4	489434	21232	STRAWBERRY CERAMIC TRINKET BOX	24	01/12/2009 07:45	1,25	13085.0	United Kingdom
...	...	...	...	...	...	...	...	...
525456	538171	22271	FELTCRAFT DOLL ROSIE	2	09/12/2010 20:01	2,95	17530.0	United Kingdom
525457	538171	22750	FELTCRAFT PRINCESS LOLA DOLL	1	09/12/2010 20:01	3,75	17530.0	United Kingdom
525458	538171	22751	FELTCRAFT PRINCESS OLIVIA DOLL	1	09/12/2010 20:01	3,75	17530.0	United Kingdom
525459	538171	20970	PINK FLORAL FELTCRAFT SHOULDER BAG	2	09/12/2010 20:01	3,75	17530.0	United Kingdom

Gambar 14.



525460	538171	21931	JUMBO STORAGE BAG SUKI	2	09/12/2010 20:01	1,95	17530.0	United Kingdom

525461 rows × 8 columns

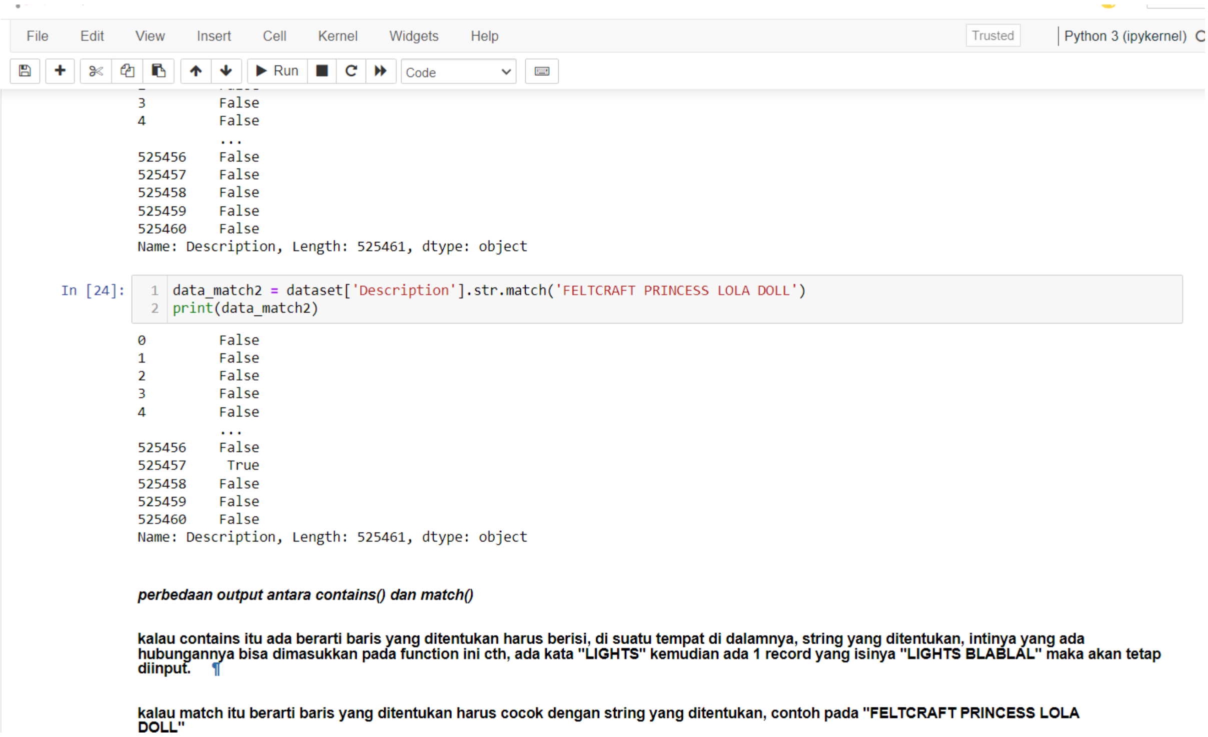
In [21]:
```
#function count dapat digunakan untuk menghitung jumlah data yang muncul dalam satu kolom

dataset['Description'].str.count('LIGHTS').sum()
```
Out[21]:
```
7838.0
```

Containts vs Match

In [22]:
```
data_contain = dataset['Description'].str.contains('LIGHTS')
print(data_contain)
```
```
0          True
1          True
2          True
3         False
4         False
          ...
525456    False
525457    False
525458    False
525459    False
525460    False
Name: Description, Length: 525461, dtype: object
```

In [23]:
```
data_match = dataset['Description'].str.match('LIGHTS')
print(data_match)
```
```
0         False
1         False
```

Gambar 15.



```
3         False
4         False
          ...
525456    False
525457    False
525458    False
525459    False
525460    False
Name: Description, Length: 525461, dtype: object
```

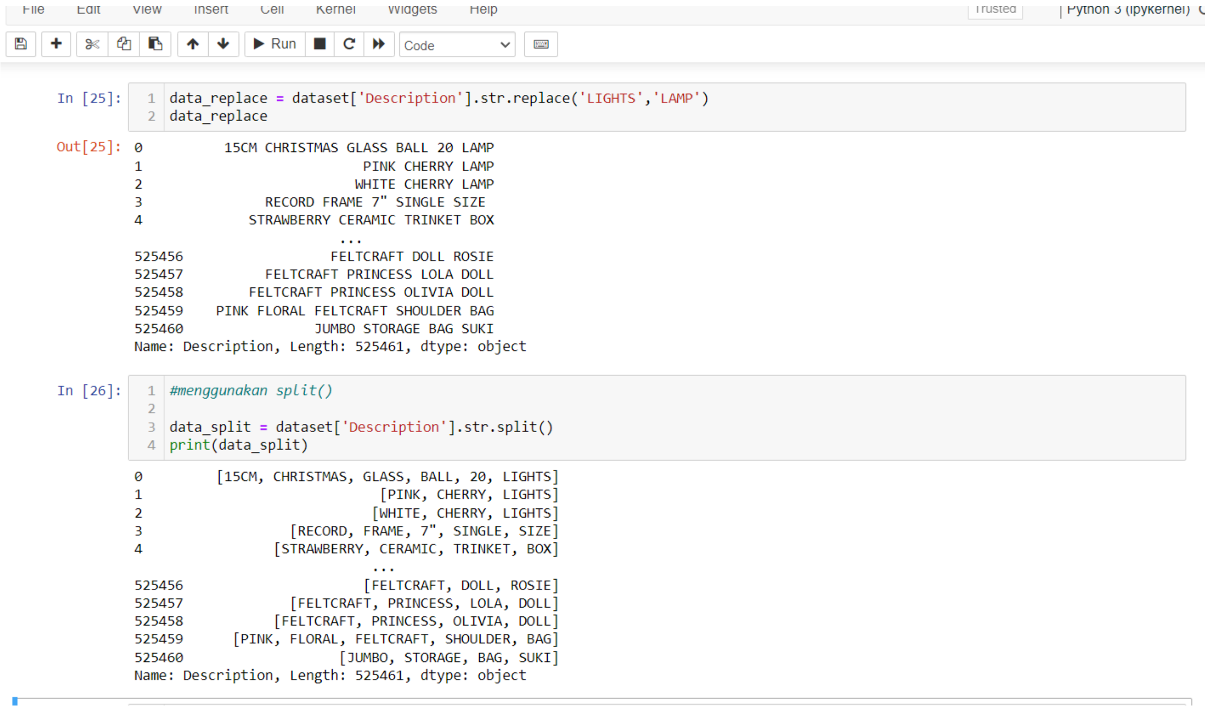
In [24]:
```
data_match2 = dataset['Description'].str.match('FELTCRAFT PRINCESS LOLA DOLL')
print(data_match2)
```
```
0         False
1         False
2         False
3         False
4         False
          ...
525456    False
525457     True
525458    False
525459    False
525460    False
Name: Description, Length: 525461, dtype: object
```

perbedaan output antara contains() dan match()

kalau contains itu ada berarti baris yang ditentukan harus berisi, di suatu tempat di dalamnya, string yang ditentukan, intinya yang ada hubungannya bisa dimasukkan pada function ini cth, ada kata "LIGHTS" kemudian ada 1 record yang isinya "LIGHTS BLABLAL" maka akan tetap diinput. ¶

kalau match itu berarti baris yang ditentukan harus cocok dengan string yang ditentukan, contoh pada "FELTCRAFT PRINCESS LOLA DOLL"

Gambar 16.



In [25]:

```
data_replace = dataset['Description'].str.replace('LIGHTS','LAMP')
data_replace
```

Out[25]:

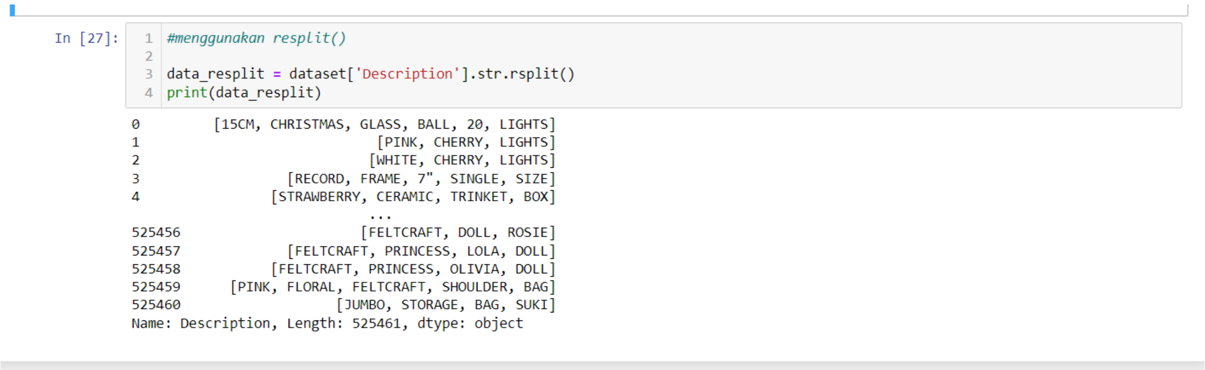
```
0                   15CM CHRISTMAS GLASS BALL 20 LAMP
1                                    PINK CHERRY LAMP
2                                   WHITE CHERRY LAMP
3                        RECORD FRAME 7" SINGLE SIZE
4                      STRAWBERRY CERAMIC TRINKET BOX
                             ...                    
525456                           FELTCRAFT DOLL ROSIE
525457                   FELTCRAFT PRINCESS LOLA DOLL
525458                 FELTCRAFT PRINCESS OLIVIA DOLL
525459             PINK FLORAL FELTCRAFT SHOULDER BAG
525460                         JUMBO STORAGE BAG SUKI
Name: Description, Length: 525461, dtype: object
```

In [26]:

```
#menggunakan split()

data_split = dataset['Description'].str.split()
print(data_split)
```

```
0               [15CM, CHRISTMAS, GLASS, BALL, 20, LIGHTS]
1                                   [PINK, CHERRY, LIGHTS]
2                                  [WHITE, CHERRY, LIGHTS]
3                        [RECORD, FRAME, 7", SINGLE, SIZE]
4                      [STRAWBERRY, CERAMIC, TRINKET, BOX]
                                 ...                    
525456                            [FELTCRAFT, DOLL, ROSIE]
525457                   [FELTCRAFT, PRINCESS, LOLA, DOLL]
525458                 [FELTCRAFT, PRINCESS, OLIVIA, DOLL]
525459            [PINK, FLORAL, FELTCRAFT, SHOULDER, BAG]
525460                         [JUMBO, STORAGE, BAG, SUKI]
Name: Description, Length: 525461, dtype: object
```

Gambar 17.



In [27]:

```
#menggunakan resplit()

data_resplit = dataset['Description'].str.rsplit()
print(data_resplit)
```

```
0               [15CM, CHRISTMAS, GLASS, BALL, 20, LIGHTS]
1                                   [PINK, CHERRY, LIGHTS]
2                                  [WHITE, CHERRY, LIGHTS]
3                        [RECORD, FRAME, 7", SINGLE, SIZE]
4                      [STRAWBERRY, CERAMIC, TRINKET, BOX]
                                 ...                    
525456                            [FELTCRAFT, DOLL, ROSIE]
525457                   [FELTCRAFT, PRINCESS, LOLA, DOLL]
525458                 [FELTCRAFT, PRINCESS, OLIVIA, DOLL]
525459            [PINK, FLORAL, FELTCRAFT, SHOULDER, BAG]
525460                         [JUMBO, STORAGE, BAG, SUKI]
Name: Description, Length: 525461, dtype: object
```

Gambar 18.