Pertemuan 2

March 4, 2024

1 Statistika Deskriptif

Statistika Deskriptif merupakan suatu metode untuk mendeskripsikan data dan memberikan gambaran umum tentang data yang dimiliki. Umumnya Statistika Deskriptif digunakan untuk memberikan informasi atau gambaran awal sebelum dilakukannya proses pemodelan atau uji statistik.

Statistika deskriptif terdiri dari

- Ukuran Pusat Data
  - Mean
  - Median
  - Modus
- Ukuran Sebaran Data
  - Range
  - IQR (Interquartile Range)
  - Variansi
  - Standar Deviasi
- Distribusi
- Korelasi

Pada pembahasan kali ini akan dibahas mengenai ukuran pusat data, ukuran sebaran data, dan korelasi. Perihal distribusi akan dibahas lebih lanjut pada bagian visualisasi data.

Pada contoh yang akan dibahas, akan digunakan dataset penjualan berbagai produk dari suatu perusahaan.

```
import pandas as pd

df = pd.read_excel("Data Pertemuan 2.xlsx")
```

```
df
```

Out[3]:

	Date	Payment	Invoice ID	Branch	City	Customer type	Gender	Product line	Unit price	Quantity	Total spent
0	2019-01-05	Ewallet	750-67-8428	A	Yangon	Member	Female	Health and beauty	74.69	7	522.83
1	2019-03-08	Cash	226-31-3081	C	Naypyitaw	Normal	Female	Electronic accessories	15.28	5	76.40
2	2019-03-03	Credit card	631-41-3108	A	Yangon	Normal	Male	Home and lifestyle	46.33	7	324.31
3	2019-01-27	Ewallet	123-19-1176	A	Yangon	Member	Male	Health and beauty	58.22	8	465.76
4	2019-02-08	Ewallet	373-73-7910	A	Yangon	Normal	Male	Sports and travel	86.31	7	604.17
...	...	...	...	...	...	...	...	...	...	...	...
995	2019-01-29	Ewallet	233-67-5758	C	Naypyitaw	Normal	Male	Health and beauty	40.35	1	40.35
996	2019-03-02	Ewallet	303-96-2227	B	Mandalay	Normal	Female	Home and lifestyle	97.38	10	973.80
997	2019-02-09	Cash	727-02-1313	A	Yangon	Member	Male	Food and beverages	31.84	1	31.84
998	2019-02-22	Cash	347-56-2442	A	Yangon	Normal	Male	Home and lifestyle	65.82	1	65.82
999	2019-02-18	Cash	849-09-3807	A	Yangon	Member	Female	Fashion accessories	88.34	7	618.38

1000 rows × 11 columns

1.1 Ukuran Pusat Data

Ukuran pemusatan data terdiri dari Mean, Median, dan Modus. Bahasa Python telah menyediakan metode untuk mendapatkan nilai-nilai tersebut melalui library pandas. Dengan fungsi dalam library pandas, seperti pd.DataFrame.mean(), pd.DataFrame.median(), dan pd.DataFrame.mode(), pandas dapat mengembalikan nilai pemusatan data secara otomatis dari data yang kita miliki. Contoh penulisan syntax adalah sebagai berikut

```
df["Total spent"].mean()
```

Out[4]: 307.58738000000034

```
df.mean(numeric_only = True)
```

Out[5]:
```
Unit price      55.67213
Quantity         5.51000
Total spent    307.58738
dtype: float64
```

```
df["Quantity"].median()
```

Out[6]: 5.0

```
df["Branch"].mode()
```

```
Out[7]: 0    A
        Name: Branch, dtype: object
```

1.2 Ukuran sebaran data

Ukuran sebaran data dapat dilihat melalui ukuran range, IQR, variansi, dan standar deviasi. Pada library pandas telah tersedia fungsi untuk mencari nilai variansi dan standar deviasi, namun tidak untuk range dan IQR. Maka dari itu, perhitungan range dan IQR akan dilakukan secara manual dengan menggunakan fungsi pd.DataFrame.max(), pd.DataFrame.min(), dan pd.DataFrame.quantile(). Sebenarnya perhitungan IQR dapat dilakukan dengan menggunakan library scipy, namun pembahasan kali ini akan lebih berfokus ke library pandas.

```
range = df["Total spent"].max() - df["Total spent"].min()
print(range)
```

```
Out[26]: 982.83
```

```
IQR = df["Total spent"].quantile(0.75) - df["Total spent"].
    quantile(0.25)
print(IQR)
```

```
Out[27]: 330.4075
```

```
df["Quantity"].var()
```

```
Out[10]: 8.546446446446451
```

```
df["Quantity"].std()
```

```
Out[11]: 2.923430595455697
```

1.3 Korelasi

Korelasi menunjukkan nilai hubungan antar 2 variabel dalam rentang nilai -1 sampai 1. Dimana semakin mendekati nilai 1 maka hubungannya akan semakin kuat.

```
df.corr()
```

Out[12]:

	Unit price	Quantity	Total spent
Unit price	1.000000	0.010778	0.633962
Quantity	0.010778	1.000000	0.705510
Total spent	0.633962	0.705510	1.000000

1.4 Deskripsi data

Pada bagian sebelumnya telah ditunjukkan bagaimana mencari nilai-nilai statistika deskriptif secara satu per satu. Namun, pandas telah menyediakan satu fungsi yang dapat mengembalikan nilai-nilai statistika deskriptif di atas secara langsung. Dapat dilihat pada syntax dan output di bawah ini, dengan menggunakan fungsi pd.DataFrame.describe(), akan muncul output berupa beberapa nilai statistika deskriptif. Meskipun tidak selengkap dan sedetil dibandingkan perhitungan satu per satu, namun fungsi ini cukup berguna untuk menggambarkan nilai-nilai tersebut.

```
df.describe()
```

Out[13]:

	Unit price	Quantity	Total spent
count	1000.000000	1000.000000	1000.00000
mean	55.672130	5.510000	307.58738
std	26.494628	2.923431	234.17651
min	10.080000	1.000000	10.17000
25%	32.875000	3.000000	118.49750
50%	55.230000	5.000000	241.76000
75%	77.935000	8.000000	448.90500
max	99.960000	10.000000	993.00000

2 Eksplorasi dan Visualisasi Data

Data Exploration and Visualization umumnya digunakan menggali informasi data lebih dalam secara visual. Selain itu Data Exploration and Visualization juga digunakan untuk mencari kemiripan dalam data, korelasi antar variabel, serta pola dan outlier dalam data. Pada bagian eksplorasi dan

visualisasi data, library yang umum digunakan adalah matplotlib dan seaborn. Selain kedua library tersebut, pandas juga menyediakan beberapa fungsi untuk mengeksplorasi dan memvisualisasikan data, namun fungsi visualisasi pada matplotlib dan seaborn lebih beragam dan fleksibel.

```
import matplotlib.pyplot as plt
import seaborn as sns
```

2.1 Eksplorasi Data

Berikut adalah beberapa fungsi yang biasa digunakan untuk eksplorasi data

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1000 entries, 0 to 999
Data columns (total 11 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   Date           1000 non-null   datetime64[ns]
 1   Payment        1000 non-null   object
 2   Invoice ID     1000 non-null   object
 3   Branch         1000 non-null   object
 4   City           1000 non-null   object
 5   Customer type  1000 non-null   object
 6   Gender         1000 non-null   object
 7   Product line   1000 non-null   object
 8   Unit price     1000 non-null   float64
 9   Quantity       1000 non-null   int64
 10  Total spent    1000 non-null   float64
dtypes: datetime64[ns](1), float64(2), int64(1), object(7)
memory usage: 86.1+ KB
```

```
df.shape()
```

Out[15]: (1000, 11)

```
df["Product line"].value_counts()
```

Out[16]:
```
Fashion accessories       178
Food and beverages        174
Electronic accessories    170
Sports and travel         166
Home and lifestyle        160
Health and beauty         152
Name: Product line, dtype: int64
```

```
df["Gender"].unique()
```

Out[17]: array(['Female', 'Male', 'fml', 'male', 'female'], dtype=object)

```
for col in df.columns:
    if df[col].dtype == "object":
        print(col, df[col].unique())
```

```
Payment ['Ewallet' 'Cash' 'Credit card']
Invoice ID ['750-67-8428' '226-31-3081' '631-41-3108' '123-19-1176' '373-73-7910'
 '699-14-3026' '355-53-5943' '315-22-5665' '665-32-9167' '692-92-5582'
 '351-62-0822' '529-56-3974' '365-64-0515' '252-56-2699' '829-34-3910'
 '299-46-1805' '656-95-9349' '765-26-6951' '329-62-1586' '319-50-3348'
 '300-71-4605' '371-85-5789' '273-16-6619' '636-48-8204' '549-59-1358'
 '227-03-5010' '649-29-6775' '189-17-4241' '145-94-9061' '848-62-7243'
 '871-79-8483' '149-71-6266' '640-49-2076' '595-11-5460' '183-56-6882'
 '232-16-2483' '129-29-8530' '272-65-1806' '333-73-7901' '777-82-7220'
 '280-35-5823' '554-53-8700' '354-25-5821' '228-96-1411' '617-15-4209'
 '132-32-9879' '370-41-7321' '727-46-3608' '669-54-1719' '574-22-5561'
 '326-78-5178' '162-48-8011' '616-24-2851' '778-71-5554' '242-55-6721'
 '399-46-5918' '106-35-6779' '635-40-6220' '817-48-8732' '120-06-4233'
 '285-68-5083' '803-83-5989' '347-34-2234' '199-75-8169' '853-23-2453'
 '877-22-3308' '838-78-4295' '109-28-2512' '232-11-3025' '382-03-4532'
 '393-65-2792' '796-12-2025' '510-95-6347' '841-35-6630' '287-21-9091'
 '732-94-0499' '263-10-3913' '381-20-0914' '829-49-1914' '756-01-7507'
 '870-72-4431' '847-38-7188' '480-63-2856' '787-56-0757' '360-39-5055'
 '730-50-9884' '362-58-8315' '633-44-8566' '504-35-8843' '318-68-5053'
```

```
df.isna().sum()
```

Out[19]:
```
Date            0
Payment         0
Invoice ID      0
Branch          0
City            0
Customer type   0
Gender          0
Product line    0
Unit price      0
Quantity        0
Total spent     0
dtype: int64
```

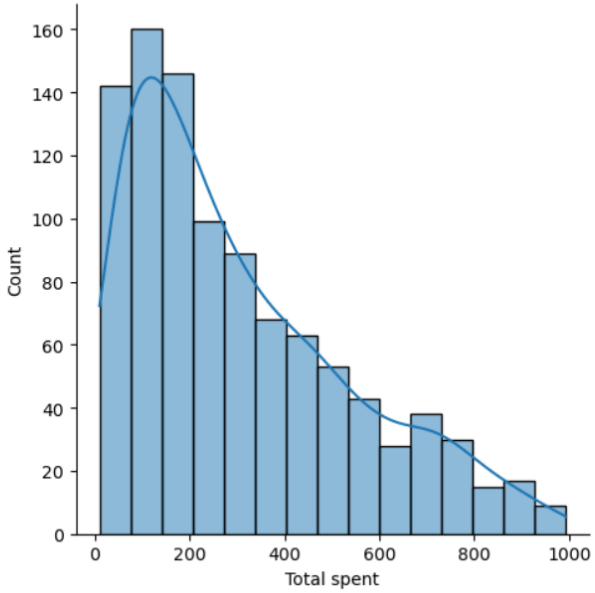
```
df.duplicated().any()
```

Out[20]: False

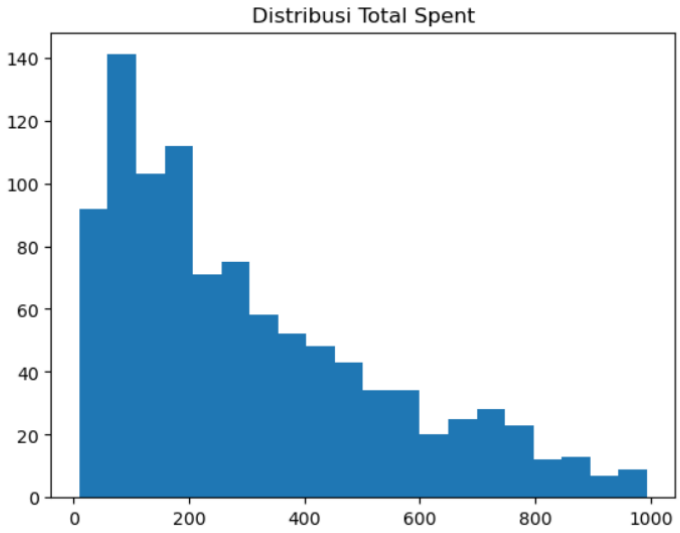
2.2 Visualisasi Data

```
sns.displot(df["Total spent"], kde = True)
```

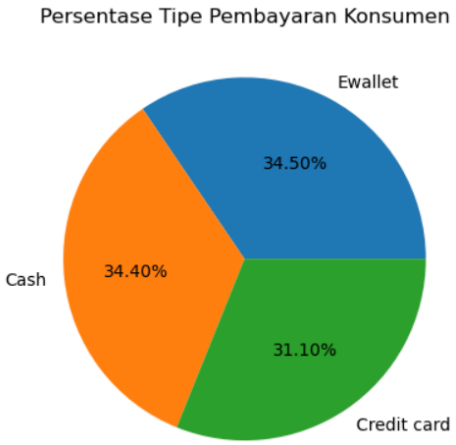
Out[21]: <seaborn.axisgrid.FacetGrid at 0x1cea6f95d90>



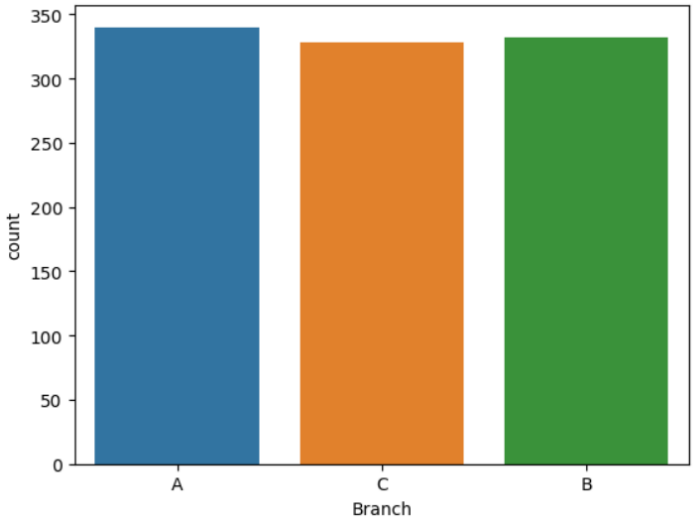

```
plt.hist(df["Total spent"], bins = 20)

plt.title("Distribusi Total spent")
plt.show()
```



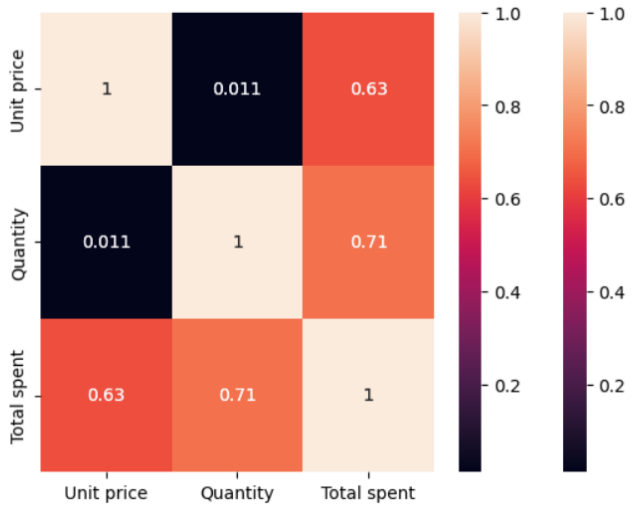

```
label_list = df["Payment"].value_counts().index

plt.pie(x = df["Payment"].value_counts(), labels = label_list,
    autopct = "%.2f%%")
```



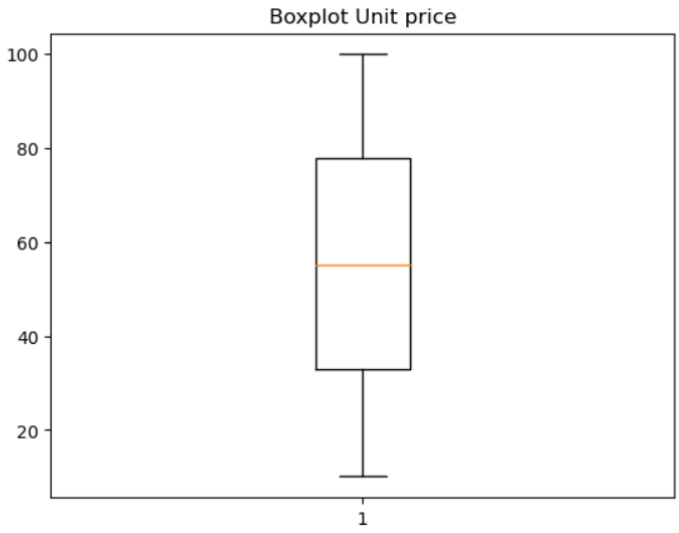

```
sns.countplot(data = df, x = "Branch")

plt.show()
```

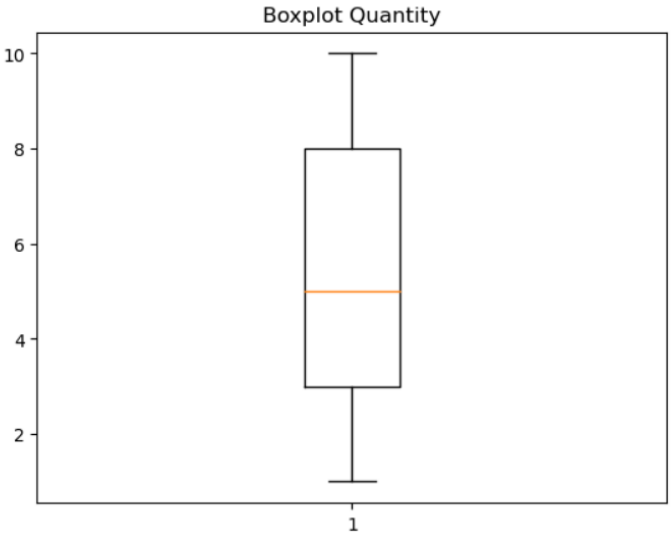


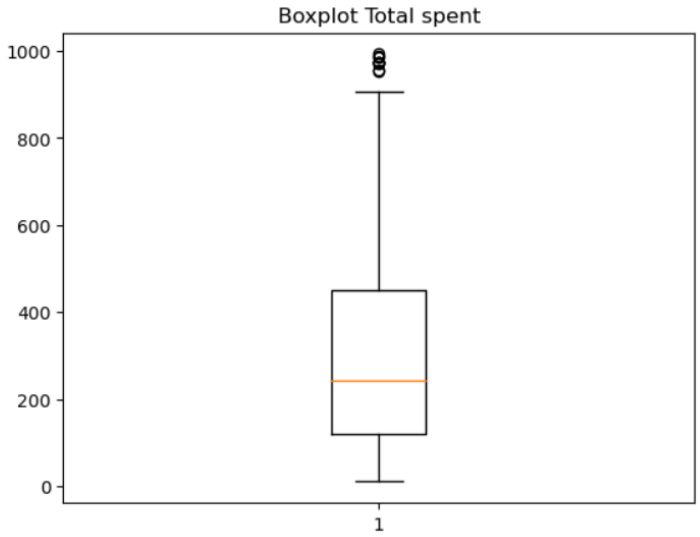

```
sns.heatmap(df.corr(), annot = True)

plt.show()
```



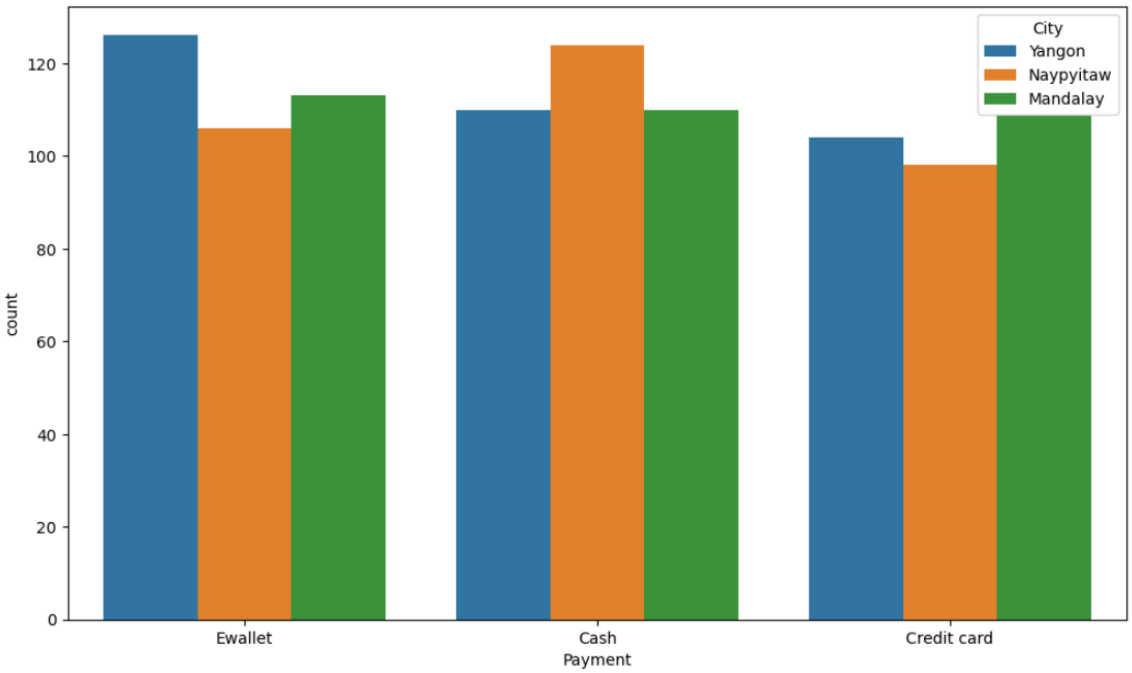

```
for col in ["Unit price","Quantity","Total spent"]:
    plt.boxplot(df[col])
    plt.title("Boxplot " + col)
    plt.show()
```


Boxplot Unit price
100
80
60
40
20
1


Boxplot Quantity
10
8
6
4
2
1




```
plt.figure(figsize = (12,7))    # Mengatur ukuran dalam satuan
    inch

sns.countplot(data = df, x = 'Payment', hue = 'City')

plt.show()
```



2.3 Tambahan Eksplorasi Data

Selain dengan fungsi-fungsi yang sebelumnya telah dikenalkan atau visualisasi di atas, terdapat salah satu metode yang cukup sering digunakan untuk mengeksplor data, terutama pada saat fase *reporting problem*. Metode tersebut adalah dengan *grouping* berdasarkan kategori dalam variabel tertentu. Pada library pandas tersedia fungsi pd.DataFrame.groupby() dan pd.DataFrame.pivot_table() untuk melakukan metode ini.

```
df.groupby(by = "Product line").sum()
```

Out[32]:

Product line	Unit price	Quantity	Total spent
Electronic accessories	9103.77	971	51750.03
Fashion accessories	10173.35	902	51719.90
Food and beverages	9745.54	952	53471.28
Health and beauty	8337.88	854	46851.18
Home and lifestyle	8850.71	911	51297.06
Sports and travel	9460.88	920	52497.93

```
df.groupby(by = ["Product line", "Gender"]).mean()
```

Out[33]:

Product line	Gender	Unit price	Quantity	Total spent
Electronic accessories	Female	52.113012	5.819277	308.153012
	Male	55.041765	5.611765	301.426353
	fml	46.950000	5.000000	234.750000
	male	52.890000	6.000000	317.340000
Fashion accessories	Female	55.874526	5.505263	303.478211
	Male	59.058049	4.536585	277.218293
	female	22.510000	7.000000	157.570000
Food and beverages	Female	59.772697	5.719101	349.994944
	Male	52.152375	5.112500	259.441375
	female	88.360000	5.000000	441.800000
	male	41.305000	7.250000	281.155000
Health and beauty	Female	51.089219	5.359375	276.205156
	Male	57.257126	5.839080	332.340805
	male	86.800000	3.000000	260.400000
Home and lifestyle	Female	57.242532	6.303797	362.108228
	Male	53.683875	5.050000	279.824375
	male	33.840000	9.000000	304.560000
Sports and travel	Female	54.367273	5.636364	309.250227
	Male	59.808684	5.526316	328.625658
	male	65.550000	2.000000	154.180000

```
df.groupby(by = 'Product line').agg({'Unit price': 'sum',
                                     'Quantity': ['max', 'median', '
                                        min'],
                                     'Total spent': ['sum', 'mean']})
```

Out[78]:

	Unit price	Quantity			Total spent	
	sum	max	median	min	sum	mean
Product line						
Electronic accessories	9103.77	10	6.0	1	51750.03	304.411941
Fashion accessories	10173.35	10	5.0	1	51719.90	290.561236
Food and beverages	9745.54	10	5.0	1	53471.28	307.306207
Health and beauty	8337.88	10	6.0	1	46851.18	308.231447
Home and lifestyle	8850.71	10	6.0	1	51297.06	320.606625
Sports and travel	9460.88	10	6.0	1	52497.93	316.252590

```
df.pivot_table(values = 'Total spent', index = ['Product line', '
    Customer type'], aggfunc = 'sum', columns = 'Payment')
```

Out[79]:

	Payment	Cash	Credit card	Ewallet
Product line	Customer type			
Electronic accessories	Member	8005.24	6604.86	8721.80
	Normal	11737.60	8088.49	8592.04
Fashion accessories	Member	6148.85	8625.31	10296.28
	Normal	10626.19	7884.28	8138.99
Food and beverages	Member	10245.71	11199.11	8419.58
	Normal	8050.36	8072.50	7484.02
Health and beauty	Member	8554.04	9145.91	6901.04
	Normal	7816.54	6063.11	8370.54
Home and lifestyle	Member	11013.68	7407.48	8224.58
	Normal	6690.13	5909.73	12051.46
Sports and travel	Member	8090.59	12037.78	6761.44
	Normal	9884.47	4930.08	10793.57

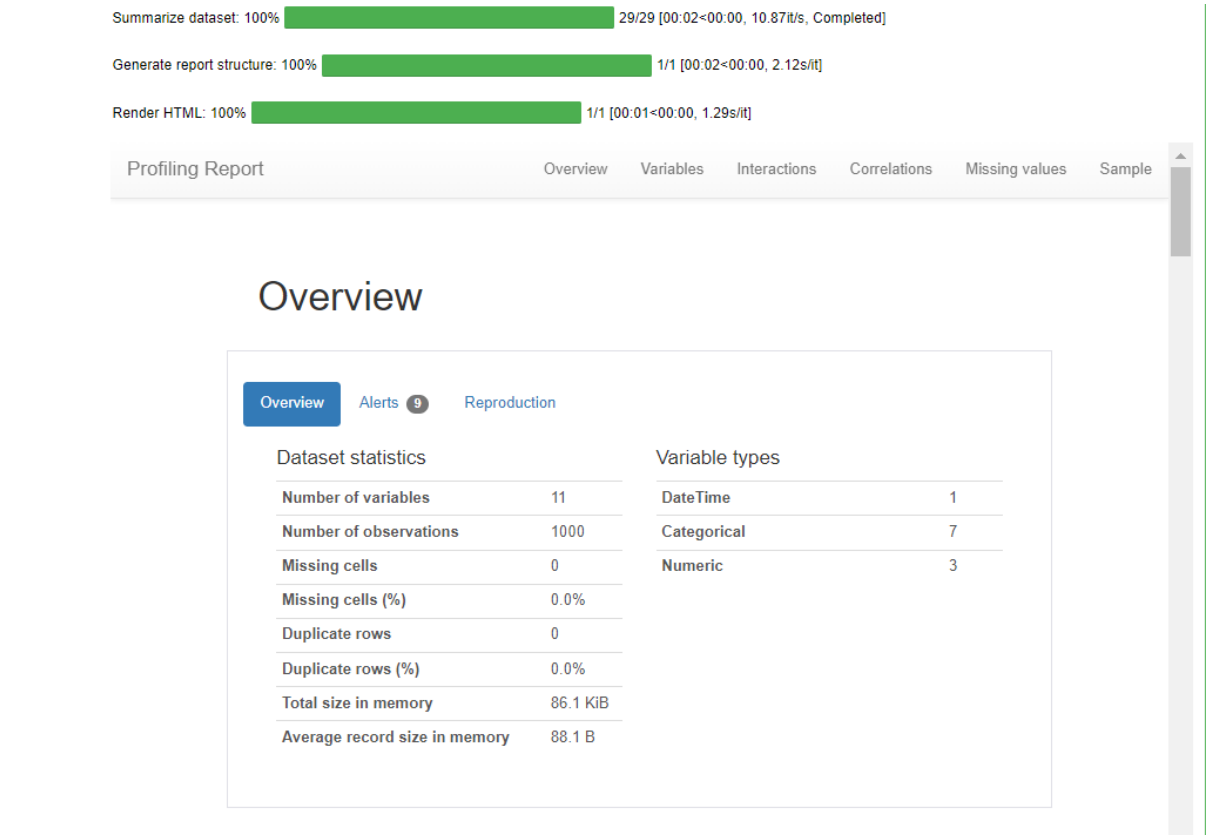
2.4 Tambahan Visualisasi Data

Terkadang ketika kita menemukan suatu permasalahan, kita ingin mengenal dan memberikan gambaran secara umum tentang data yang dimiliki. Untuk memberikan gambaran umum tentang data dan memvisualisasikannya tentunya cukup memakan waktu. Maka dari itu, terdapat satu library yang mampu mengeskplor dan memvisualisasikan data secara otomatis dan cukup lengkap untuk

memberikan gambaran umum terkait data yang dimiliki. Library yang dimaksud merupakan pandas-profiling, yang saat ini berganti nama menjadi ydata-profiling.

ydata-profling merupakan library yang dapat melakukan tahapan EDA (Exploratory Data Analysis) secara otomatis hanya dalam kode yang singkat. Selain memberikan eksplorasi dan visualisasi data, library ini juga memberikan hasil laporan yang interaktif. Namun, dengan segala keuntungan yang dimiliki ydata-profiling di atas, ydata-profiling memiliki satu *downside* yaitu kebutuhan akan RAM / Memory yang cukup besar, terutama ketika data yang dimiliki berdimensi cukup besar. Pada dasarnya library ini hanya disarankan ketika sekiranya gawai yang digunakan oleh analis cukup kuat untuk menjalankan program.

```
import ydata_profiling as pf

full_profile = pf.ProfileReport(df, title="Profiling Report", html
    ={"style":{"full_width":True}})
full_profile
```



3 Data Preprocessing

Data preprocessing merupakan tahapan dalam analisis data guna mempersiapkan data sebelum dilakukan tahap pemodelan. Beberapa tahapan dalam data preprocessing secara umum adalah

- Data Cleaning Membersihkan data dengan cara mengisi nilai yang hilang, memperbaiki data inkonsisten, menghapus duplikat, melengkapi data yang kurang lengkap, dan sebagainya
- Data Transformation Mengubah bentuk data supaya sesusai dengan apa yang diinginkan algoritma pemodelan. Contoh Transformasi data dapat ditemui pada Data Scaling, membuat variabel Dummy, mengubah bentuk data, normalisasi data, Binning, Labeling, dan lain sebagainya
- Data Reduction Tahapan ini merupakan tahapan dimana jumlah data akan dikurangi dengan tujuan untuk meningkatkan akurasi. Data reduction dapat dilakukan dengan Dimensionality reduction atau data compression

3.1 Data Cleaning (Handling Missing Values)

Pada saat kita mencari dan mendapatkan data, tentunya kerap kali kita menemukan adanya data yang tidak lengkap. Ketidaklengkapan data tersebut umumnya disebut sebagai *missing value*. Hal tersebut dapat terjadi karena adanya kesalahan saat menyimpan file maupun kesalahan pada orang yang menginput data.

Beberapa metode untuk mengatasi masalah tersebut adalah

- Menghapus baris atau kolom dengan *missing value*s
- Mengisinya dengan suatu nilai
- Membiarkan dan menggunakan algoritma yang mendukung adanya *missing values*
- Menggunakan Algoritma untuk memprediksi nilai yang hilang

```
df = sns.load_dataset("titanic")
df
```

Out[83]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	NaN	Southampton	no	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	B	Southampton	yes	True
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	NaN	Southampton	no	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	C	Cherbourg	yes	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	NaN	Queenstown	no	True

891 rows × 15 columns

```
df.isna().sum()
```

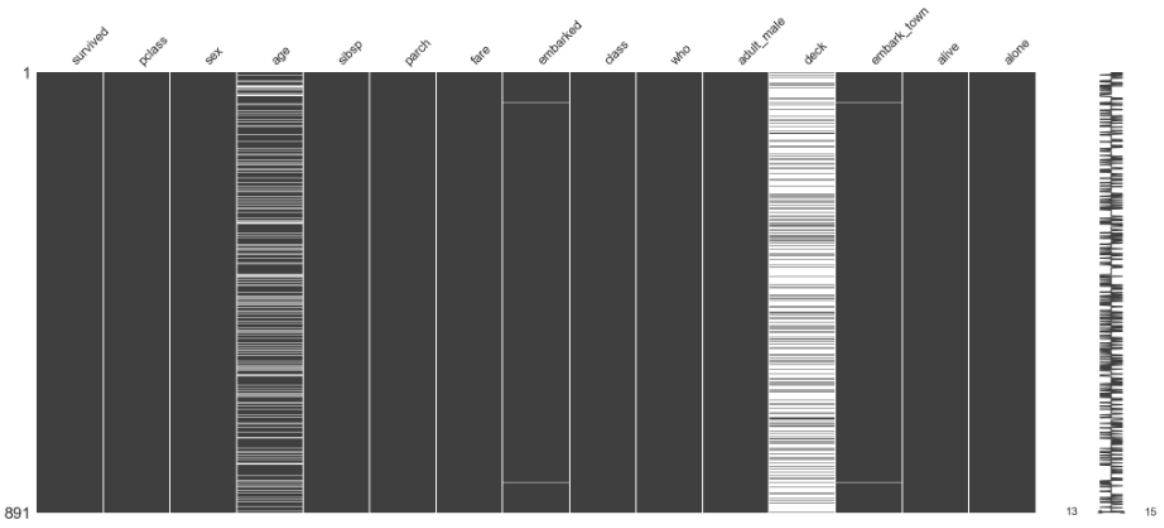
Out[84]:
```
survived         0
pclass           0
sex              0
age            177
sibsp            0
parch            0
fare             0
embarked         2
class            0
who              0
adult_male       0
deck           688
embark_town      2
alive            0
alone            0
dtype: int64
```

```
pd.DataFrame({'Jumlah NA': df.isna().sum(), 'Persentase': (df.isna
    ().sum()/len(df)*100)})
```

Out[85]:

	Jumlah NA	Persentase
survived	0	0.000000
pclass	0	0.000000
sex	0	0.000000
age	177	19.865320
sibsp	0	0.000000
parch	0	0.000000
fare	0	0.000000
embarked	2	0.224467
class	0	0.000000
who	0	0.000000
adult_male	0	0.000000
deck	688	77.216611
embark_town	2	0.224467
alive	0	0.000000
alone	0	0.000000

```
import missingno as mno

mno.matrix(df)
plt.show()
```




```
df.dropna(subset = 'embarked', inplace = True)
df
```

Out[106]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.0	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.0	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.0	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.0	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.0	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.0	0	0	13.0000	S	Second	man	True	NaN	Southampton	no	True
887	1	1	female	19.0	0	0	30.0000	S	First	woman	False	B	Southampton	yes	True
888	0	3	female	NaN	1	2	23.4500	S	Third	woman	False	NaN	Southampton	no	False
889	1	1	male	26.0	0	0	30.0000	C	First	man	True	C	Cherbourg	yes	True
890	0	3	male	32.0	0	0	7.7500	Q	Third	man	True	NaN	Queenstown	no	True

889 rows × 15 columns

```
fill = df['age'].mean()

df['age'].fillna(fill, inplace = True)
df.head(10)
```

Out[107]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	deck	embark_town	alive	alone
0	0	3	male	22.000000	1	0	7.2500	S	Third	man	True	NaN	Southampton	no	False
1	1	1	female	38.000000	1	0	71.2833	C	First	woman	False	C	Cherbourg	yes	False
2	1	3	female	26.000000	0	0	7.9250	S	Third	woman	False	NaN	Southampton	yes	True
3	1	1	female	35.000000	1	0	53.1000	S	First	woman	False	C	Southampton	yes	False
4	0	3	male	35.000000	0	0	8.0500	S	Third	man	True	NaN	Southampton	no	True
5	0	3	male	29.642093	0	0	8.4583	Q	Third	man	True	NaN	Queenstown	no	True
6	0	1	male	54.000000	0	0	51.8625	S	First	man	True	E	Southampton	no	True
7	0	3	male	2.000000	3	1	21.0750	S	Third	child	False	NaN	Southampton	no	False
8	1	3	female	27.000000	0	2	11.1333	S	Third	woman	False	NaN	Southampton	yes	False
9	1	2	female	14.000000	1	0	30.0708	C	Second	child	False	NaN	Cherbourg	yes	False

```
df.drop('deck', axis = 1, inplace = True)
df
```

Out[108]:

	survived	pclass	sex	age	sibsp	parch	fare	embarked	class	who	adult_male	embark_town	alive	alone
0	0	3	male	22.000000	1	0	7.2500	S	Third	man	True	Southampton	no	False
1	1	1	female	38.000000	1	0	71.2833	C	First	woman	False	Cherbourg	yes	False
2	1	3	female	26.000000	0	0	7.9250	S	Third	woman	False	Southampton	yes	True
3	1	1	female	35.000000	1	0	53.1000	S	First	woman	False	Southampton	yes	False
4	0	3	male	35.000000	0	0	8.0500	S	Third	man	True	Southampton	no	True
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
886	0	2	male	27.000000	0	0	13.0000	S	Second	man	True	Southampton	no	True
887	1	1	female	19.000000	0	0	30.0000	S	First	woman	False	Southampton	yes	True
888	0	3	female	29.642093	1	2	23.4500	S	Third	woman	False	Southampton	no	False
889	1	1	male	26.000000	0	0	30.0000	C	First	man	True	Cherbourg	yes	True
890	0	3	male	32.000000	0	0	7.7500	Q	Third	man	True	Queenstown	no	True

889 rows × 14 columns

4 Practice

a) Wakabayashi, seorang Kepala Direktur Penjualan di suatu perusahaan, sedang berencana untuk memberikan hadiah berupa voucher diskon kepada pelanggan yang telah membeli produk perusahaannya. Namun, Wakabayashi memiliki syarat bahwa pelanggan tersebut harus setidaknya menghabiskan biaya sebesar $180 agar mendapatkan diskon. Kalian sebagai rekan kerja andalan Wakabayashi diminta untuk memberikan data yang mendukung kepada Wakabayashi. Dengan menggunakan "Data Pertemuan 2.xlsx", cari dan dapatkanlah data penjualan yang telah membeli barang lebih dari $180!!

b) Seorang peneliti ingin meneliti tentang tingkat survavibilitas penumpang kapal Titanic. Peneliti tersebut menyatakan bahwa jenis kelamin yang paling banyak *survived* adalah wanita. Apakah pernyataan peneliti tersebut benar? Berikan bukti yang mendukung jawabanmu!

c) Berapakah rata-rata usia pria yang selamat dari bencana kapal Titanic?