

MODUL PEMROGRAMAN LANJUT

Dosen Pengampu: Sevi Nurafni

Praktikum 7

Visualisasi Data

Asisten Laboratorium : Catherine V. Pang



— Modul ini diperuntukkan bagi Mata Kuliah Pemrograman Lanjut Program Studi S1 Sains Data semester III

Catatan

1. Modul ini dirancang sebagai pegangan untuk mata kuliah Pemrograman Lanjut.
2. Anda dapat membuka modul ini saat latihan praktikum.
3. Sangat disarankan untuk mencoba menjalankan semua program dalam modul ini di komputer Anda untuk memahami keluaran dari program yang ada.
4. Bereksperimen dengan program-program dalam modul ini sangat dianjurkan agar Anda mendapatkan gambaran lebih jelas mengenai apa yang dilakukan oleh program tersebut.
5. Membaca tutorial dari sumber lain dan mengeksplorasi bahasa pemrograman yang digunakan sangat disarankan.

1. Menggunakan Jupyter Notebook

Untuk modul 4 dan 5, kita akan menggunakan Jupyter Notebook. Untuk menginstall Jupyter Notebook, silakan buka halaman <https://docs.anaconda.com/anaconda/install/>.

Untuk menggunakan pandas, kita akan menambahkan line berikut di kernel kita, tepat sesudah header:

```
import pandas as pd
```

Pada modul ini, kita akan menggunakan data yang bisa didownload di https://drive.google.com/drive/folders/1o2Zg_Lc911dsW0lw37uWgYqM0-dR8Jro?usp=sharing. Contoh notebook juga bisa didownload dari link yang sama.

2. Analisa Data

2.1. Membuat Dataframe

```
import pandas as pd # Mengimpor library pandas untuk pengelolaan data

# Inisialisasi dictionary untuk menyimpan data
input_data = {}
input_data["A"] = [0 for i in range(5)] # Membuat kolom A dengan nilai awal 0
input_data["B"] = [0 for i in range(5)] # Membuat kolom B dengan nilai awal 0

# Loop untuk meminta input dari pengguna untuk kolom A
for i in range(5):
    input_data["A"][i] = input(" Nilai A untuk data ke-" + str(i + 1) + ": ")

# Loop untuk meminta input dari pengguna untuk kolom B
for i in range(5):
    input_data["B"][i] = input(" Nilai B untuk data ke-" + str(i + 1) + ": ")

# Membuat DataFrame dari data yang telah diinputkan
df = pd.DataFrame(data=input_data)

# Menampilkan DataFrame
print(df)
```

Untuk membuat data frame, kita perlu membuat dictionary dengan key berupa nama kolom dan berisi array dari data yang ada. Sebagai contoh, perhatikan potongan kode berikut:

2.2. Membaca dan Menulis Data

Untuk membaca data csv, kita dapat menggunakan method `read_csv`. Untuk membaca data excel, kita dapat menggunakan method `read_excel`. Untuk menulis data, baik ke csv maupun excel, perhatikan contoh berikut:

```

import pandas as pd # Mengimpor library pandas untuk pengelolaan data

# Membaca file CSV bernama "tingkatinflasi20082013.csv"
# File ini harus berada di folder yang sama dengan file Python ini
df1 = pd.read_csv("tingkatinflasi20082013.csv")
print(df1) # Menampilkan isi DataFrame df1

# Membaca file Excel bernama "data.xlsx" yang berada di direktori D:/,
# dan meload data dari sheet bernama "Sheet1"
df2 = pd.read_excel("D:/data.xlsx", sheet_name="Sheet1")
print(df2) # Menampilkan isi DataFrame df2

# Menulis data dari DataFrame df1 ke file CSV bernama "data_out.csv"
# File akan disimpan di folder yang sama dengan file Python ini
df1.to_csv("data_out.csv", index=False)

# Menulis data dari DataFrame df2 ke file Excel di direktori D:/ dengan nama "data_out.xlsx"
# Data akan ditulis pada dua sheet bernama "Sheet1" dan "Sheet2"
writer = pd.ExcelWriter("D:/data_out.xlsx", engine='xlsxwriter') # Membuat writer untuk menulis ke Excel
df2.to_excel(writer, sheet_name="Sheet1", index=False) # Menulis df2 ke sheet "Sheet1"
df2.to_excel(writer, sheet_name="Sheet2", index=False) # Menulis df2 ke sheet "Sheet2"
writer.save() # Menyimpan file Excel

```

2.3. Mengakses Data

Perhatikan contoh berikut:

```

import pandas as pd # Mengimpor library pandas

# Membaca file CSV bernama "tingkatinflasi20082013.csv"
df = pd.read_csv("tingkatinflasi20082013.csv")

# Mengambil data baris ke-5 (index 4, karena indexing dimulai dari 0)
print(df.loc[4])
# Output:
# Tahun      2009
# Cakupan   Prov. Jawa Barat
# Tingkat_Inflasi    2.02
# Name: 4, dtype: object

# Mengambil data dari baris ke-5 hingga ke-7 (index 4 hingga 6)
print(df[4:7])
# Output:
#   Tahun      Cakupan Tingkat_Inflasi
# 4  2009  Prov. Jawa Barat      2.02
# 5  2009      Nasional      2.78
# 6  2010   Kota Bandung      4.53

# Mengambil data dari baris ke-17 (index 16) hingga akhir
print(df[16:])
# Output:
#   Tahun      Cakupan Tingkat_Inflasi
# 16  2013  Prov. Jawa Barat      9.15
# 17  2013      Nasional      8.38

# Mengambil 5 data pertama

```

```

print(df[:5])
# Output:
#   Tahun    Cakupan Tingkat_Inflasi
# 0  2008   Kota Bandung    10.23
# 1  2008  Prov. Jawa Barat    11.11
# 2  2008    Nasional    11.06
# 3  2009   Kota Bandung     2.11
# 4  2009  Prov. Jawa Barat     2.02

# Melihat panjang (jumlah baris) dari DataFrame
print(len(df))
# Output: 18

# Mengambil nilai kolom "Cakupan" dari data baris ke-2 (index 1)
print(df.loc[1, "Cakupan"])
# Output:
# 'Prov. Jawa Barat'

```

Selain itu, kita bisa mengakses data berdasar kriteria. Perhatikan contoh berikut:

```

import pandas as pd # Mengimpor library pandas

# Membaca file CSV bernama "tingkatinflasi20082013.csv"
df = pd.read_csv("tingkatinflasi20082013.csv")

# Mengambil data dengan tahun 2012
print(df.loc[df["Tahun"] == 2012])
# Output:
#   Tahun    Cakupan Tingkat_Inflasi
# 12  2012   Kota Bandung     4.02
# 13  2012  Prov. Jawa Barat     3.86
# 14  2012    Nasional     4.30

# Mengambil data Kota Bandung sebelum tahun 2012
print(df.loc[(df["Cakupan"] == "Kota Bandung") & (df["Tahun"] < 2012)])
# Output:
#   Tahun    Cakupan Tingkat_Inflasi
# 0  2008   Kota Bandung    10.23
# 3  2009   Kota Bandung     2.11
# 6  2010   Kota Bandung     4.53
# 9  2011   Kota Bandung     2.75

# Mengambil data dengan tingkat inflasi di atas 10 atau di bawah 3
print(df.loc[(df["Tingkat_Inflasi"] > 10) | (df["Tingkat_Inflasi"] < 3)])
# Output:
#   Tahun    Cakupan Tingkat_Inflasi
# 0  2008   Kota Bandung    10.23
# 1  2008  Prov. Jawa Barat    11.11
# 2  2008    Nasional    11.06
# 3  2009   Kota Bandung     2.11
# 4  2009  Prov. Jawa Barat     2.02
# 5  2009    Nasional     2.78
# 9  2011   Kota Bandung     2.75

```

2.4. Mengambil Ekstremum

Ekstremum adalah data yang ekstrem (paling tinggi atau paling rendah).

```
import pandas as pd # Mengimpor library pandas

# Membaca file CSV bernama "tingkatinflasi20082013.csv"
df = pd.read_csv("tingkatinflasi20082013.csv")

# Mengambil data dengan tahun 2012
print(df.loc[df["Tahun"] == 2012])
# Output:
#   Tahun    Cakupan Tingkat_Inflasi
# 12  2012    Kota Bandung         4.02
# 13  2012  Prov. Jawa Barat         3.86
# 14  2012      Nasional         4.30

# Mengambil data Kota Bandung sebelum tahun 2012
print(df.loc[(df["Cakupan"] == "Kota Bandung") & (df["Tahun"] < 2012)])
# Output:
#   Tahun    Cakupan Tingkat_Inflasi
# 0  2008    Kota Bandung         10.23
# 3  2009    Kota Bandung          2.11
# 6  2010    Kota Bandung          4.53
# 9  2011    Kota Bandung          2.75

# Mengambil data dengan tingkat inflasi di atas 10 atau di bawah 3
print(df.loc[(df["Tingkat_Inflasi"] > 10) | (df["Tingkat_Inflasi"] < 3)])
# Output:
#   Tahun    Cakupan Tingkat_Inflasi
# 0  2008    Kota Bandung         10.23
# 1  2008  Prov. Jawa Barat         11.11
# 2  2008      Nasional         11.06
# 3  2009    Kota Bandung          2.11
# 4  2009  Prov. Jawa Barat          2.02
# 5  2009      Nasional          2.78
# 9  2011    Kota Bandung          2.75

# Mengambil data dengan inflasi minimum
imin = df["Tingkat_Inflasi"].idxmin()
print(df[imin: imin + 1])
# Output:
#   Tahun    Cakupan Tingkat_Inflasi
# 4  2009  Prov. Jawa Barat          2.02
```

2.5. Mengurutkan Data

Data dapat diurutkan secara tidak menurun (ascending) tidak menaik (descending).

```
import pandas as pd
df = pd.read_csv("tingkatinflasi20082013.csv")

# Mengurutkan data berdasar tingkat inflasi, ascending
print(df.sort_values(["Tingkat_Inflasi"], ascending=[1]))

# Mengurutkan data berdasar tahun ascending, lalu tingkat inflasi descending
print(df.sort_values(["Tahun", "Tingkat_Inflasi"], ascending=[1, 0]))
```

2.6. Tabel Frekuensi

Kita dapat membuat tabel frekuensi. Tabel frekuensi berdasar kolom X artinya kita mendaftar semua kemungkinan nilai di kolom X secara unik, lalu menghitung berapa kali nilai itu muncul.

```
import pandas as pd
df = pd.read_csv("tingkatinflasi20082013.csv")

# Mendaftar kemunculan tiap tahun pada data
print(df["Tahun"].value_counts())
# Output yang diharapkan:
# 2013  3
# 2012  3
# 2011  3
# 2010  3
# 2009  3
# 2008  3
```

2.7. Menentukan Range

```
import pandas as pd
df = pd.read_csv("tingkatinflasi20082013.csv")

# Mengambil nilai minimum dan maksimum tiap kolom
minimum = df.min()
maximum = df.max()

# Menuliskan range tingkat inflasi
print(maximum["Tingkat_Inflasi"]) # 11.11
print(minimum["Tingkat_Inflasi"]) # 2.02
```

2.8. Statistik Sederhana

```
import pandas as pd
df = pd.read_csv("tingkatinflasi20082013.csv")

# Menampilkan deskripsi statistik dari data
df.describe()

#output
#   count    18.000000    18.000000
#   mean    2010.500000    5.818889
#   std      1.757338     3.148673
#   min      2008.000000    2.020000
#   25%      2009.000000    3.272500
#   50%      2010.500000    4.415000
#   75%      2012.000000    8.277500
#   max      2013.000000   11.110000
```

Dari data di atas, kita bisa melihat ada 18 data. Rata-rata tingkat inflasi adalah 5.8189. Standar deviasi dari tingkat inflasi adalah 3.1487. Tingkat inflasi minimum adalah 2.02 dan maksimumnya 11.11. Tingkat inflasi juga memiliki kuartil bawah 3.273, kuartil tengah 4.4150, dan kuartil atas 8.2775.

Kita dapat juga mengambil statistik satu per satu:

```
import pandas as pd
df = pd.read_csv("tingkatinflasi20082013.csv")

# Menghitung mean dari seluruh kolom
df.mean()
# Output:
# Tahun          2010.500000
# Tingkat_Inflasi  5.818889

# Menghitung mean untuk kolom "Tingkat_Inflasi"
df["Tingkat_Inflasi"].mean() # 5.818888888888889

# Menghitung standard deviation (std) untuk kolom "Tingkat_Inflasi"
df["Tingkat_Inflasi"].std() # 3.1486727154915681
```

2.9. Koefisien Korelasi

Dua kolom pada data yang sama bisa memiliki korelasi. Tingkat korelasi ini kita sebut sebagai koefisien korelasi. Cara memaknai koefisien korelasi adalah sebagai berikut:

Semakin mendekati 0, semakin dua kolom tidak berkorelasi.
Semakin mendekati 1, semakin dua kolom berbanding lurus.
Semakin mendekati -1, semakin dua kolom berbanding terbalik.

Berikut ini adalah contoh bila kita ingin melihat korelasi tahun dengan tingkat inflasi:

```
correlation = df["Tingkat_Inflasi"].corr(df["Tahun"])
# Menghitung korelasi antara kolom "Tingkat_Inflasi" dan "Tahun"
```



```
print(correlation)
```

3. Visualisasi Data

Untuk visualisasi data, kita akan menggunakan matplotlib. Matplotlib harusnya sudah diinstall saat Anda menginstall Anaconda. Pada subbab ini, kita akan menggunakan data yang bisa didownload di https://drive.google.com/drive/folders/1o2Zg_Lc911dsW0lw37uWgYqM0-dR8Jro?usp=sharing.

Contoh notebook juga bisa didownload dari link yang sama. Ada 3 data yang bisa akan digunakan:

3.1. Membuat Data Frame

```
import pandas as pd
import matplotlib.pyplot as plt

# Membaca file CSV
data = pd.read_csv("data.csv")
medali = pd.read_csv("medali.csv")
animal = pd.read_csv("animal.csv")
```

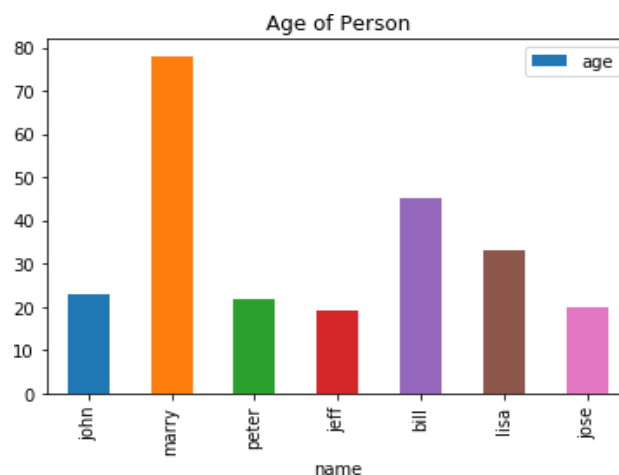
3.2. Bar Chart

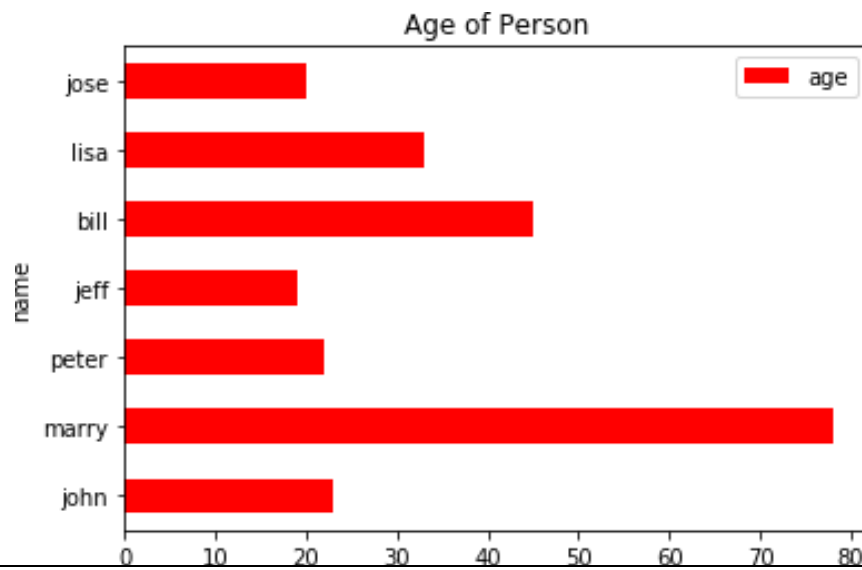
```
import matplotlib.pyplot as plt

# Membaca file CSV
data = pd.read_csv("data.csv")

# Membuat vertical bar chart untuk menampilkan umur dari setiap orang
data.plot(kind="bar", x="name", y="age", title="Age of Person")

# Menampilkan plot
plt.show()
```

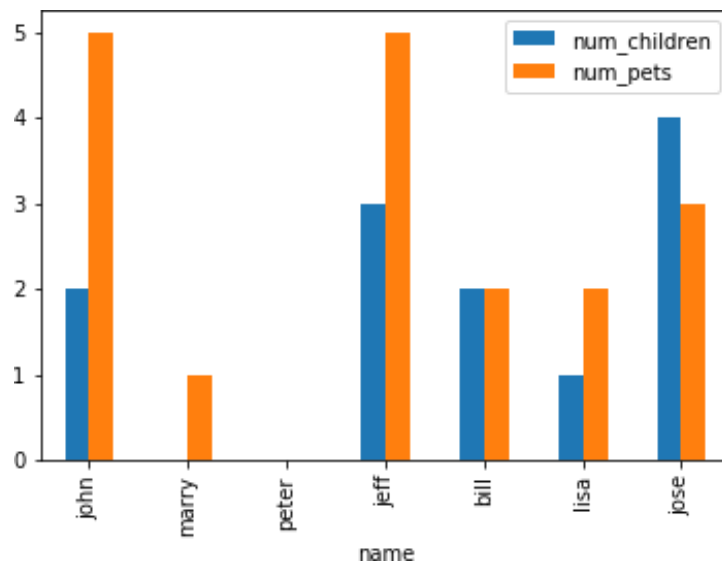




```
# Membaca file CSV
data = pd.read_csv("data.csv")

# Membuat vertical bar chart untuk menampilkan num_children dan num_pets
data.plot(kind="bar", x="name", y=["num_children", "num_pets"], title="Number of Children and Pets")

# Menampilkan plot
plt.show()
```



```
import pandas as pd
import matplotlib.pyplot as plt

# Membaca file CSV
data = pd.read_csv("data.csv")

# Membuat horizontal bar chart untuk menampilkan umur dari setiap orang
data.plot(kind="barh", x="name", y="age", title="Age of Person", color="red")

# Menampilkan plot
plt.show()
```

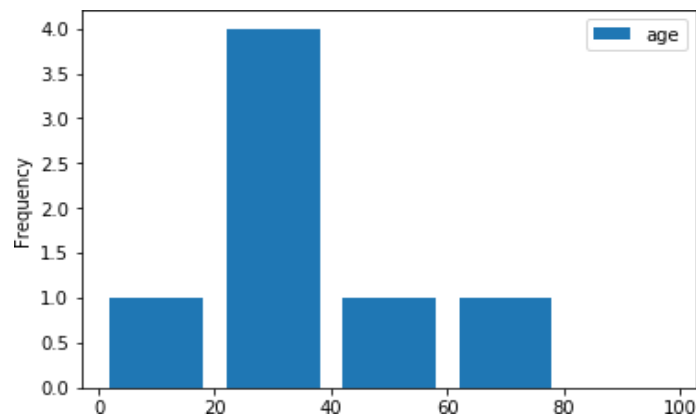
3.3. Histogram

```
import pandas as pd
import matplotlib.pyplot as plt

# Membaca file CSV
data = pd.read_csv("data.csv")

# Membuat histogram orang berdasarkan kelompok umur
data[["age"]].plot(kind="hist", bins=[0, 20, 40, 60, 80, 100], rwidth=0.8, title="Age Groups Distribution")

# Menampilkan plot
plt.show()
```



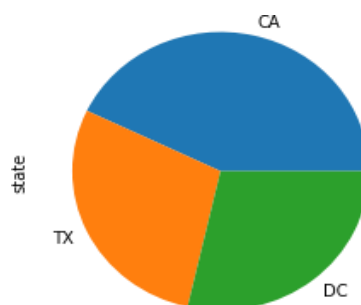
3.4. Pie Chart

```
import pandas as pd
import matplotlib.pyplot as plt

# Membaca file CSV
data = pd.read_csv("data.csv")

# Membuat pie chart berdasarkan jumlah orang per negara
data["state"].value_counts().plot(kind="pie", autopct='%1.1f%%', title="Composition of People by State")

# Menampilkan plot
plt.show()
```



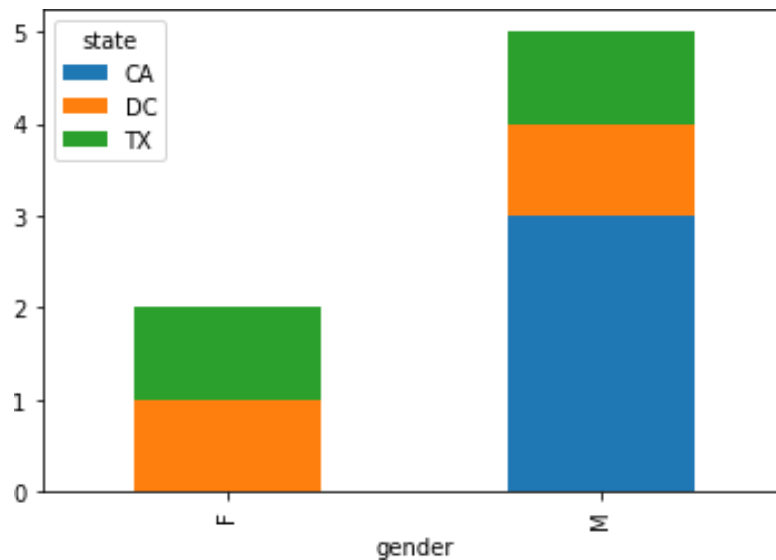
3.5. Stacked Bar Chart

```
import pandas as pd
import matplotlib.pyplot as plt

# Membaca file CSV
data = pd.read_csv("data.csv")

# Membuat stacked bar chart berdasarkan gender per state
data.groupby(["gender", "state"])["name"].size().unstack().plot(
    kind="bar", stacked=True, title="Data Count by Gender and State"
)

# Menampilkan plot
plt.show()
```

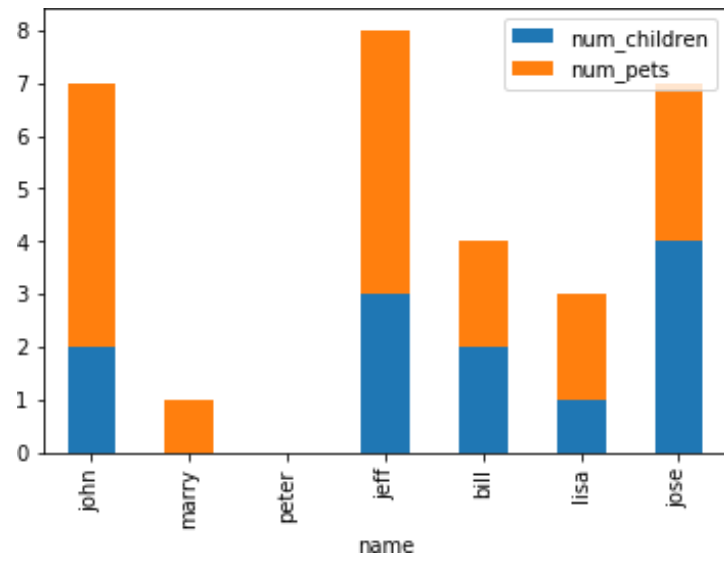


```
import pandas as pd
import matplotlib.pyplot as plt

# Membaca file CSV
data = pd.read_csv("data.csv")

# Membuat stacked bar chart untuk num_children dan num_pets
data.plot(
    kind="bar",
    x="name",
    y=["num_children", "num_pets"],
    stacked=True,
    title="Number of Children and Pets"
)

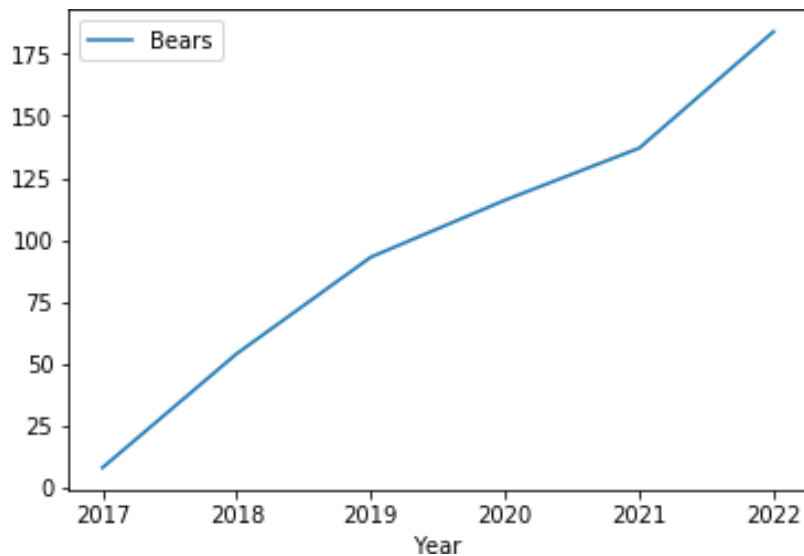
# Menampilkan plot
plt.show()
```



3.6. Line Chart

```
import pandas as pd
import matplotlib.pyplot as plt

animal = pd.read_csv("animal.csv")
animal.plot( kind=" line",x=" Year",y=" Bears")
plt.show()
```

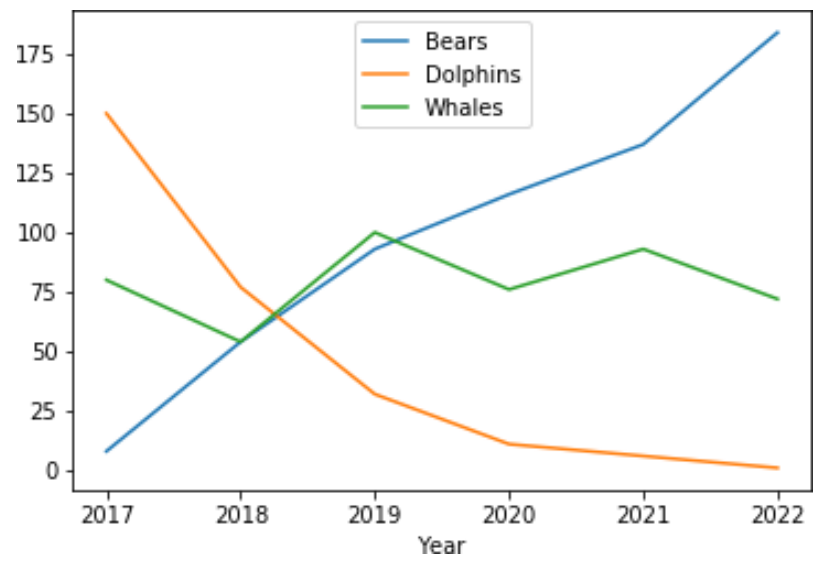


```
# Import library yang diperlukan
import pandas as pd
import matplotlib.pyplot as plt

# Membaca file CSV yang berisi data populasi hewan
animal = pd.read_csv("animal.csv")

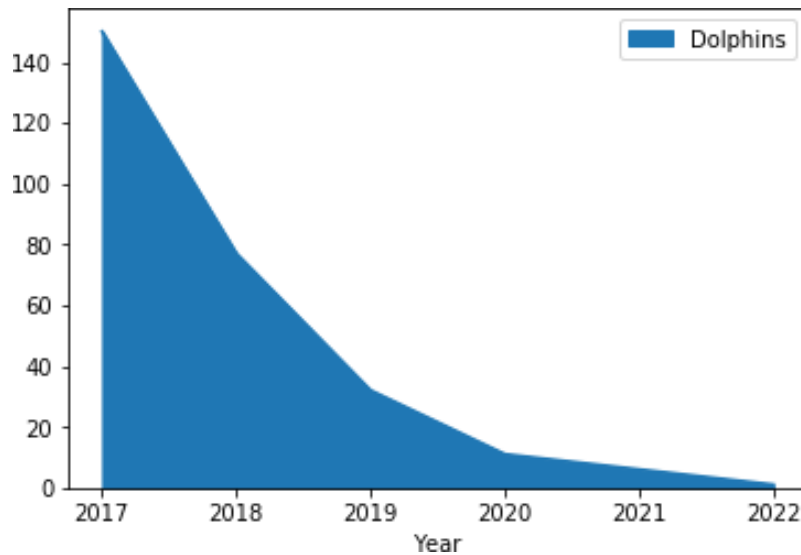
# Membuat line chart untuk menampilkan pertumbuhan populasi beruang, lumba-lumba, dan ikan paus dari tahun ke tahun
# kind="line" -> Membuat grafik garis
# x="Year" -> Menggunakan kolom "Year" sebagai sumbu x
# y=["Bears", "Dolphins", "Whales"] -> Menggunakan kolom "Bears", "Dolphins", dan "Whales" sebagai data pada sumbu y
animal.plot(kind="line", x="Year", y=["Bears", "Dolphins", "Whales"])

# Menampilkan grafik
plt.show()
```



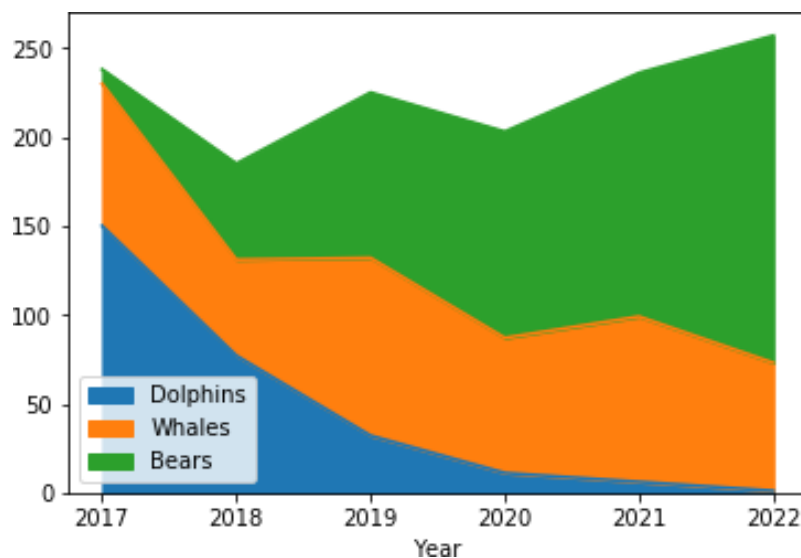
3.7. Area Chart

```
# Pertumbuhan populasi lumba - lumba ( Dolphins) dari tahun ke tahun # dalam area chart  
animal. plot( kind=" area",x=" Year", y=" Dolphins")
```



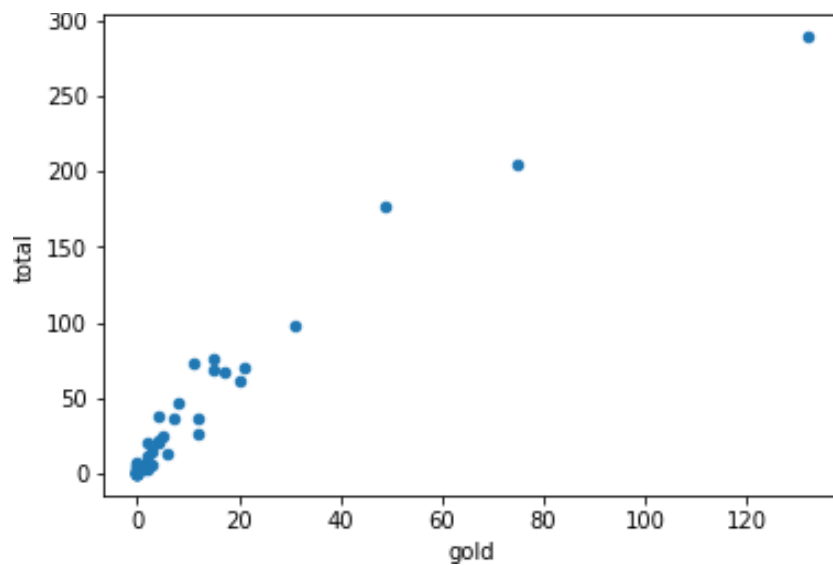
```
# Pertumbuhan populasi lumba - lumba ( Dolphins), ikan paus ( Whales), dan beruang ( Bears),  
# dari tahun ke tahun dalam stacked area
```

```
chart animal. plot( kind=" area",x=" Year", y=[" Dolphins"," Whales"," Bears"])
```

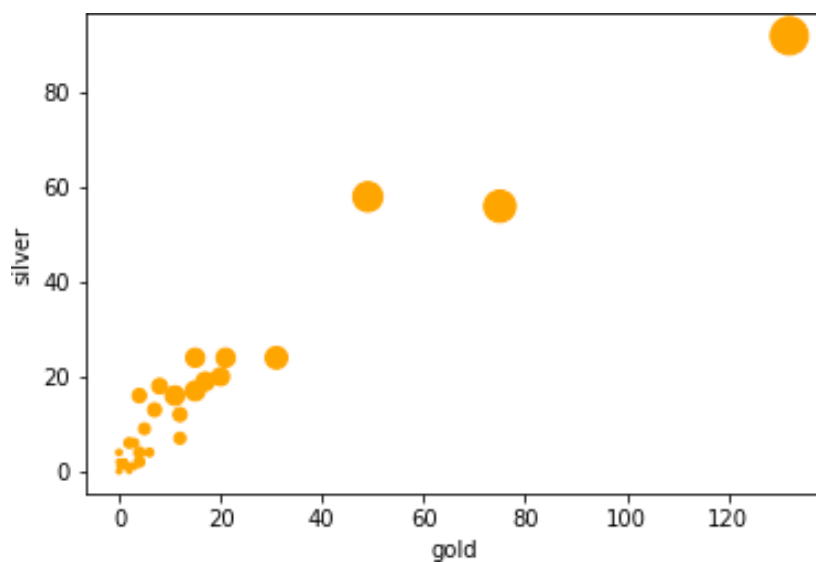


3.8. Scatter dan Bubble Plot

Relationship antara variable gold dan total dalam grafik scatter plot # dan tunjukkan adanya korelasi positif
medali. plot(kind=" scatter", x=" gold", y=" total")



Banyaknya total medali dikaitkan dengan perolehan nilai medali emas (gold)
pada sumbu x dan perolehan medali perak (silver) pada sumbu y
dalam grafik bubble plot
medali. plot(kind=" scatter",x=" gold", y=" silver", sizes=medali[" total"], color=" orange")



Diberikan data mahasiswa tentukan/visualisasikan (Gunakan dataset H05.csv dari pertemuan sebelumnya):

1. Histogram distribusi nilai matematika.
2. Diagram batang horizontal banyaknya mahasiswa masing-masing fakultas.
3. Diagram pie banyaknya mahasiswa masing-masing fakultas
4. Berdasarkan diagram pie dan batang horizontal manakah fakultas dengan mahasiswa terbanyak? manakan diagram yang lebih baik dalam menampilkan fakultas dengan mahasiswa terbanyak dan mengapa?
5. Scatter plot dengan nilai kimia sebagai x dan fisika sebagai y.