Bahasa Python

Part #3

- Seluruh materi di dalam PPT ini bersumber dari:

Modul PTI-B Python

Tim Materi PTI-B 2018/2019 Institut Teknologi Bandung

2019-03-19

Analisis Data dengan Jupiter Notebook

- *Pandas* adalah pustaka perangkat lunak yang ditulis untuk bahasa pemrograman Python untuk manipulasi dan analisis data.

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

```
import pandas as pd
```

4.2 Membuat Dataframe

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:

```
import pandas as pd

input_data = {}
input_data['A'] = [0 for i in range(5)]
input_data['B'] = [0 for i in range(5)]

for i in range(5):
    input_data['A'][i] = input('Nilai A untuk data ke-' + str(i + 1) + ': ')
for i in range(5):
    input_data['B'][i] = input('Nilai B untuk data ke-' + str(i + 1) + ': ')

df = pd.DataFrame(data=input_data)
print(df)
```

4.3 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

# pandas akan membaca file
# tingkatinflasi20082013.csv
# yang ada di folder yang sama dengan
# tempat kode python ini disimpan
df1 = pd.read_csv("tingkatinflasi20082013.csv")
print(df1)

# pandas akan membaca file data.xlsx
```

```
# yang ada di D:/, lalu meload data
# yang ada di sheet bernama "Sheet 1"
df2 = pd.read_excel("D:/data.xlsx", sheet_name="Sheet 1")
print(df2)

# pandas akan menulis data ke file
df = pd.read_csv("log.txt")
df.to_csv("logggg.csv")

ha = pd.read_csv("logggg.csv")
ha.to_excel("excel_1.xlsx")

writer = pd.ExcelWriter("excel_2.xlsx")
ha.to_excel(writer, "Sheet1")
writer.save()
```

4.4 Mengakses Data

Perhatikan contoh berikut:

```
import pandas as pd

df = pd.read_csv("tingkatinflasi20082013.csv")

# mengambil data ke-5
print(df.loc[4])
# Tahun                            2009
# Cakupan               Prov. Jawa Barat
# Tingkat_Inflasi                    2.02
# Name: 4, dtype: object

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

```python
# mengambil data ke-17 hingga akhir
print(df[16:])
#     Tahun           Cakupan  Tingkat_Inflasi
# 16   2013  Prov. Jawa Barat             9.15
# 17   2013          Nasional             8.38

# mengambil 5 data pertama
print(df[:5])
#    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 data
print(len(df))
# 18

# mengambil kolom "Cakupan" dari data ke-2
print(df.loc[1, "Cakupan"])
# 'Prov. Jawa Barat'
```

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

```
import pandas as pd

df = pd.read_csv("tingkatinflasi20082013.csv")

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

```python
# mengambil data Kota Bandung sebelum tahun 2012
print(df.loc[(df["Cakupan"] == "Kota Bandung") & (df["Tahun"] < 2012)])
#     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)])
#    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
```

4.5 Mengambil Ekstremum

Ekstremum adalah data yang ekstrem: paling tinggi atau paling rendah

```
import pandas as pd

df = pd.read_csv("tingkatinflasi20082013.csv")

# Mengambil data dengan inflasi maksimum
imax = df["Tingkat_Inflasi"].idxmax()
print(df[imax:imax + 1])
#    Tahun           Cakupan  Tingkat_Inflasi
# 1   2008  Prov. Jawa Barat            11.11

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

4.6 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]))
```

Latihan (review)

- **Data (tingkatinflasi20082013.csv)**

- Tampilkan data Provinsi jawa barat dari tahun 2008 sampai 2013 dengan tingkat inflasi lebih dari 5, dan urutkan secara ***Descending*** berdasarkan **Tahun.**

4.7 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())
# 2013    3
# 2012    3
# 2011    3
# 2010    3
# 2009    3
# 2008    3
```

4.8 Menentukan Range

```
import pandas as pd

df = pd.read_csv("tingkatinflasi20082013.csv")

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

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

4.9 Statistik Sederhana

```
import pandas as pd

df = pd.read_csv("tingkatinflasi20082013.csv")

df.describe()
#              Tahun  Tingkat_Inflasi
# 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")

df.mean()
# Tahun                2010.500000
# Tingkat_Inflasi         5.818889

df["Tingkat_Inflasi"].mean() # 5.818888888888889
```

5 Modul 5

5.1 Visualisasi Data

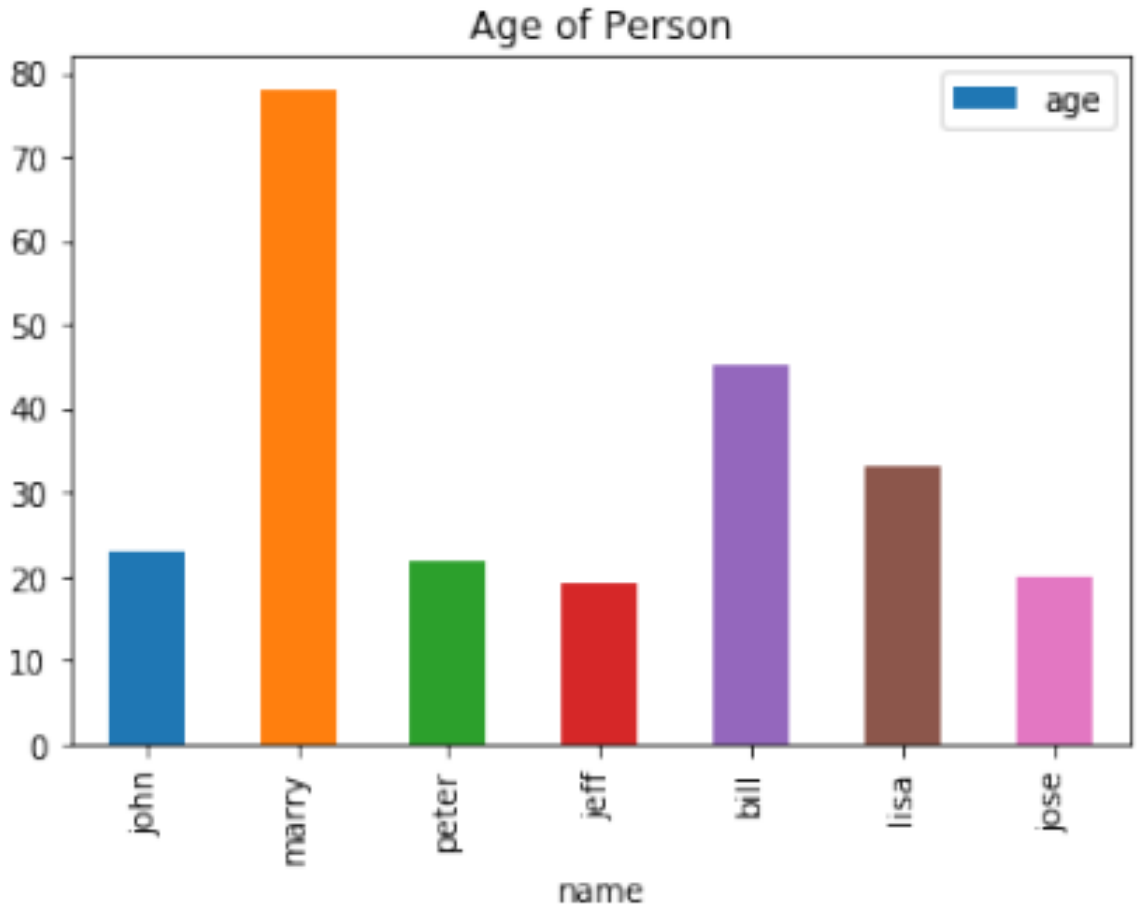
Untuk visualisasi data, kita akan menggunakan matplotlib. Matplotlib harusnya sudah diinstall saat Anda menginstall Anaconda seperti pada modul 4.

Pada modul ini, kita akan menggunakan data yang bisa didownload di https://drive.google.com/drive/folders/1o2Zg_Lc911dsW0Iw37uWgYqM0-dR8Jro?usp=sharing. Contoh notebook juga bisa didownload dari link yang sama. Ada 3 data yang bisa akan digunakan:

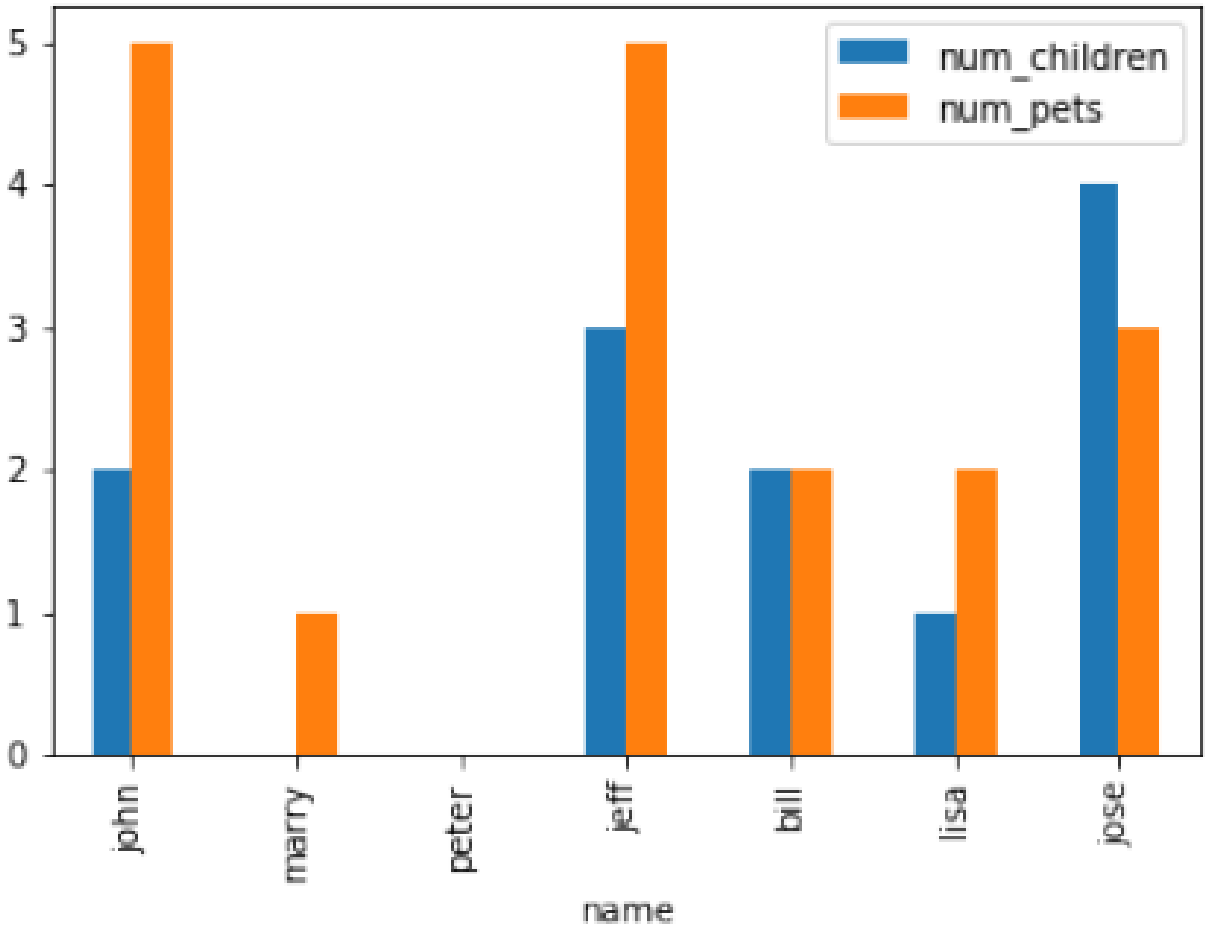
```
import pandas as pd
import matplotlib as plt

data = pd.read_csv("data.csv")
medali = pd.read_csv("medali.csv")
animal = pd.read_csv("animal.csv")
```

5.2 Bar Chart

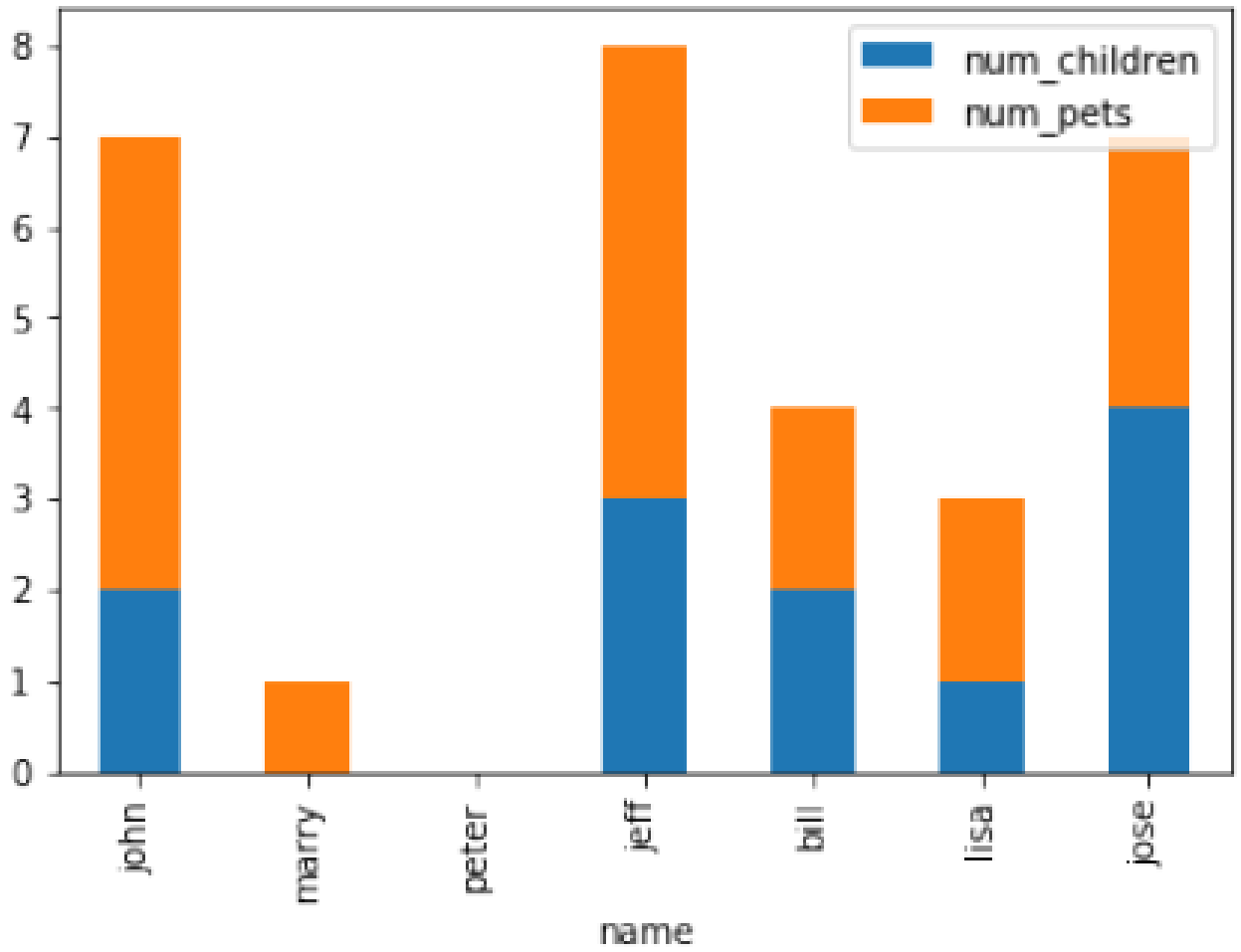
```
# Vertical bar chart untuk menampilkan umur dari setiap orang
data.plot(kind="bar",x="name",y="age",title="Age of Person")
```



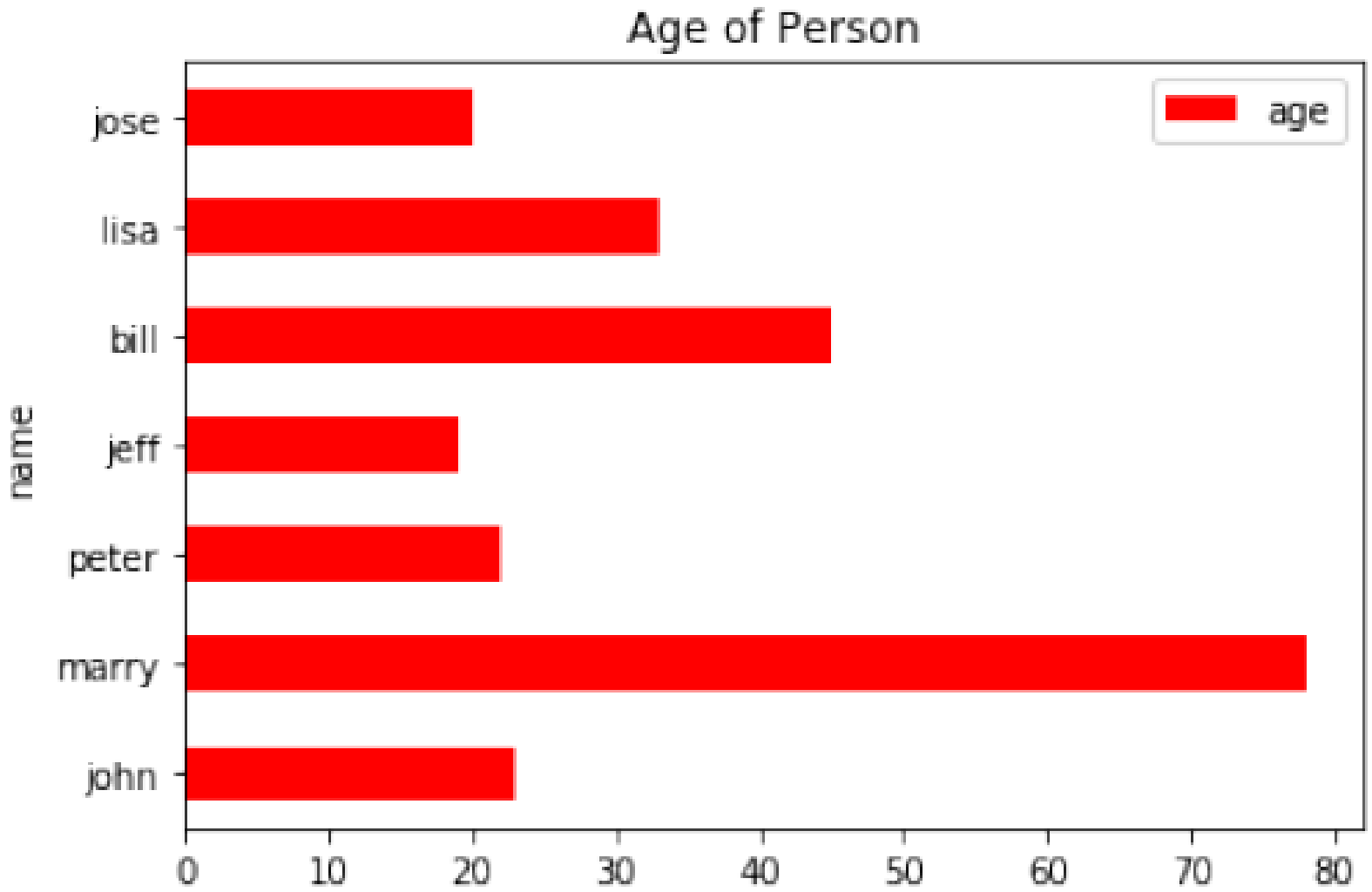
```
# Banyaknya anak (num_children) dan banyaknya piaraan (num_pets) dalam
# 1 grafik vertical bar chart
data.plot(kind="bar",x="name",y=["num_children","num_pets"])
```



```
# Banyaknya anak (num_children) dan banyaknya piaraan (num_pets)
# dalam 1 grafik stacked bar chart
data.plot(kind="bar", x="name", y=["num_children","num_pets"], stacked=True)
```

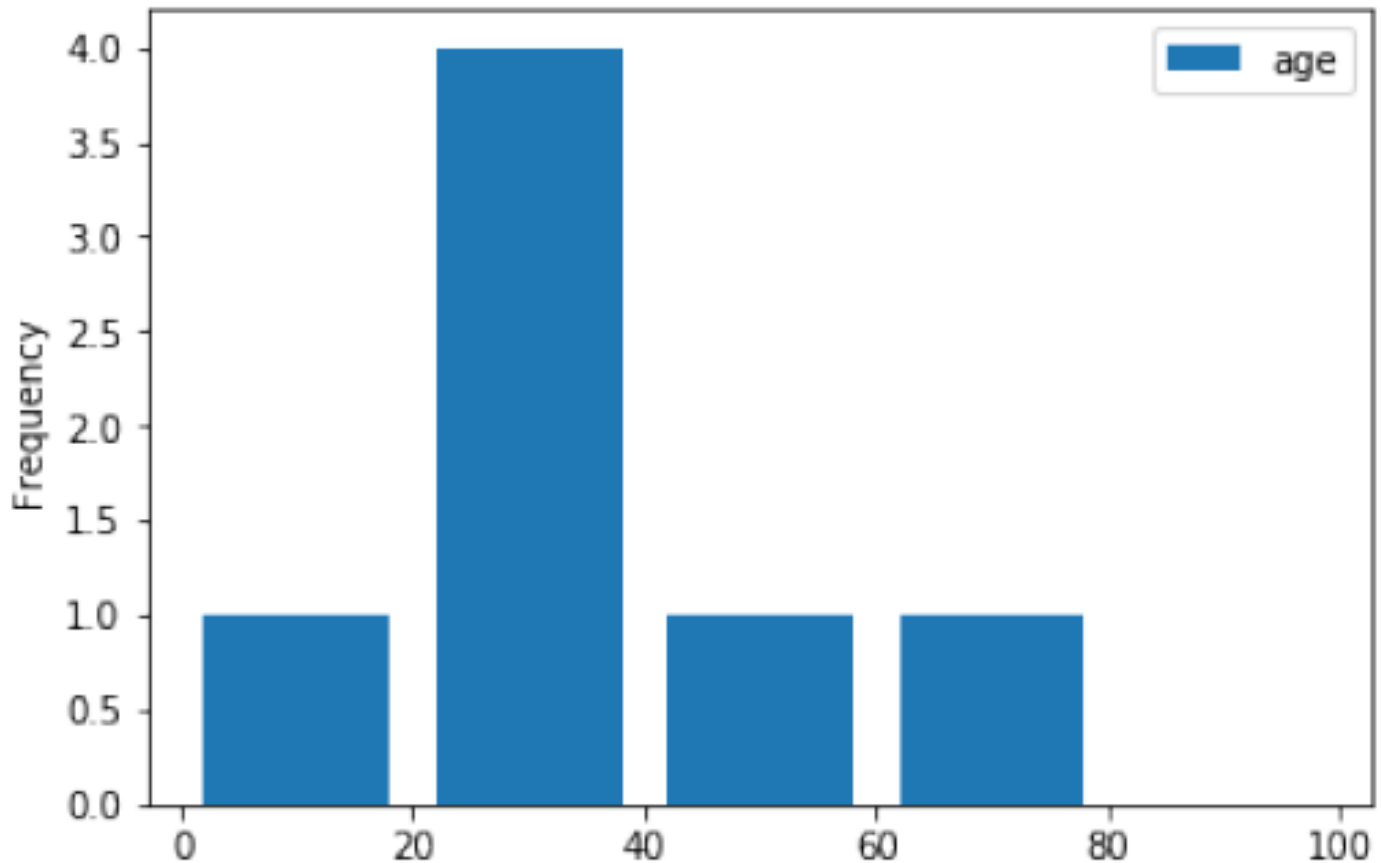


```
data.plot(kind="barh",x="name",y="age",title="Age of Person", color="red")
```



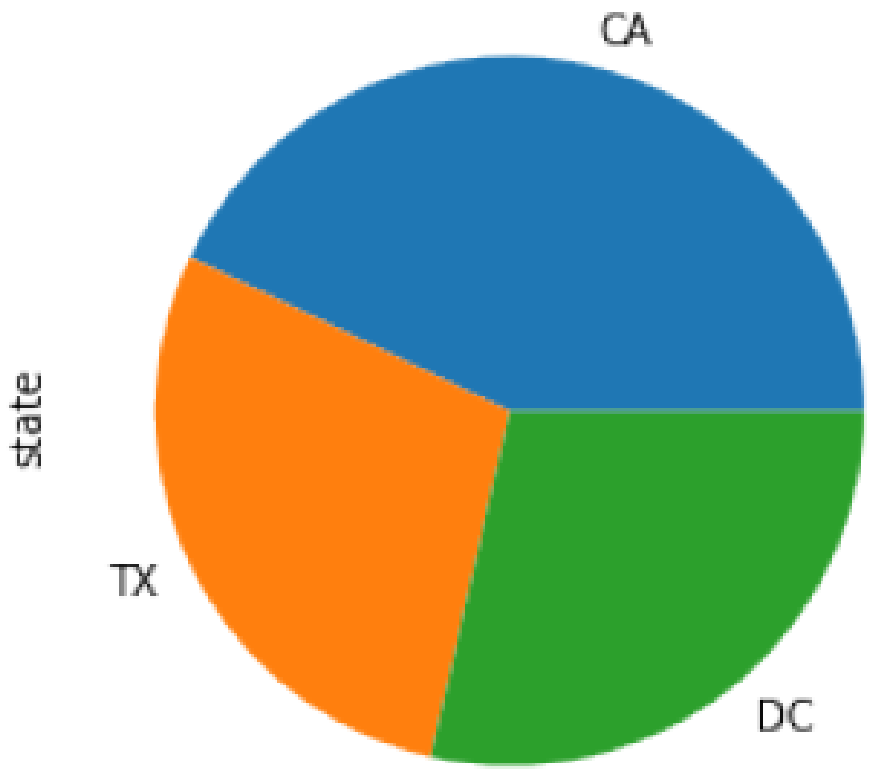
5.3 Histogram

```
# Histogram orang berdasarkan kelompok umur: 0-20; 21-40; 41-60; 61-80; 81-100
data[["age"]].plot(kind="hist",bins=[0,20,40,60,80,100],rwidth=0.8)
```



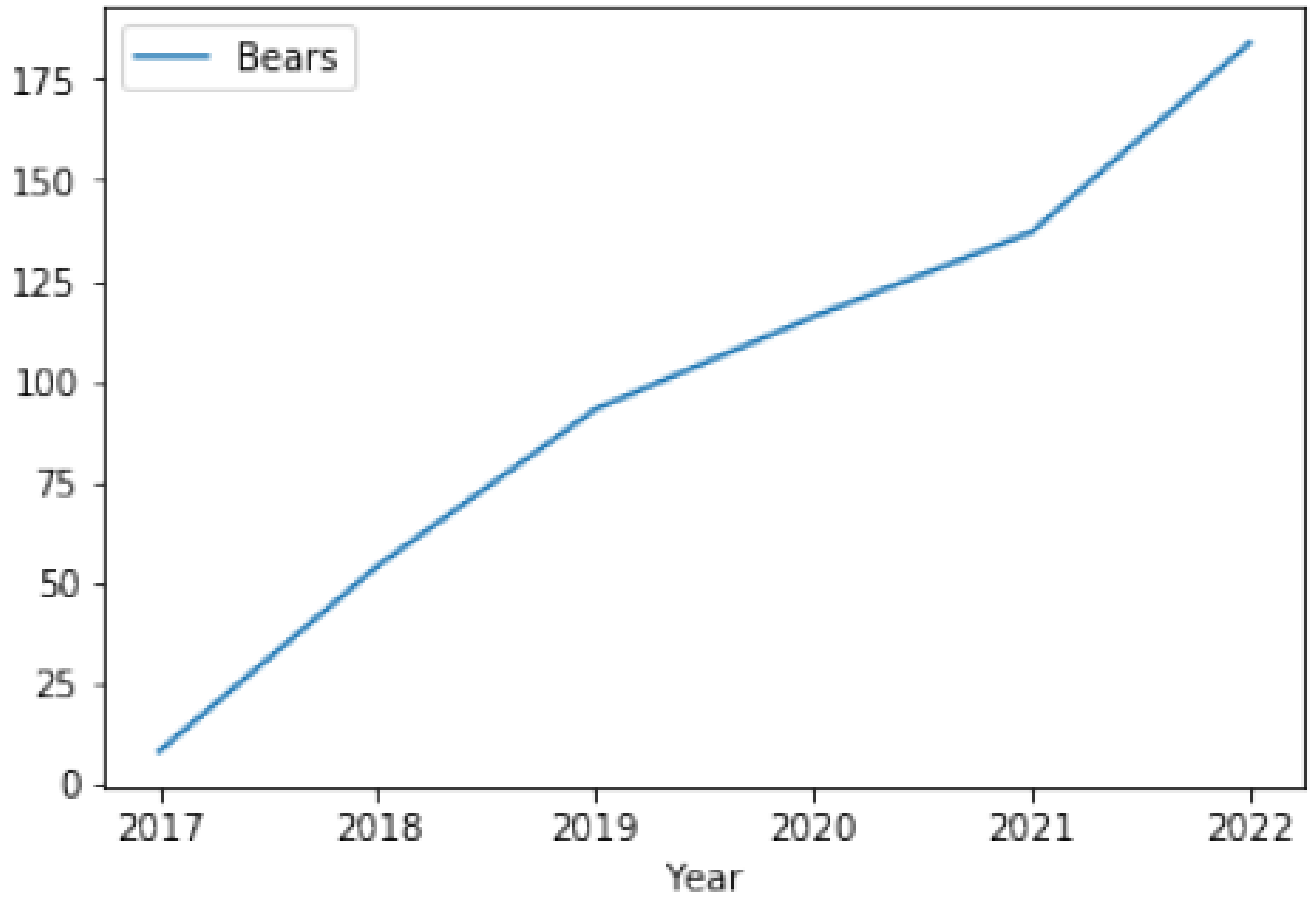
5.4 Pie Chart

```
# Komposisi banyaknya orang berdasarkan negara
data["state"].value_counts().plot(kind = "pie")
```

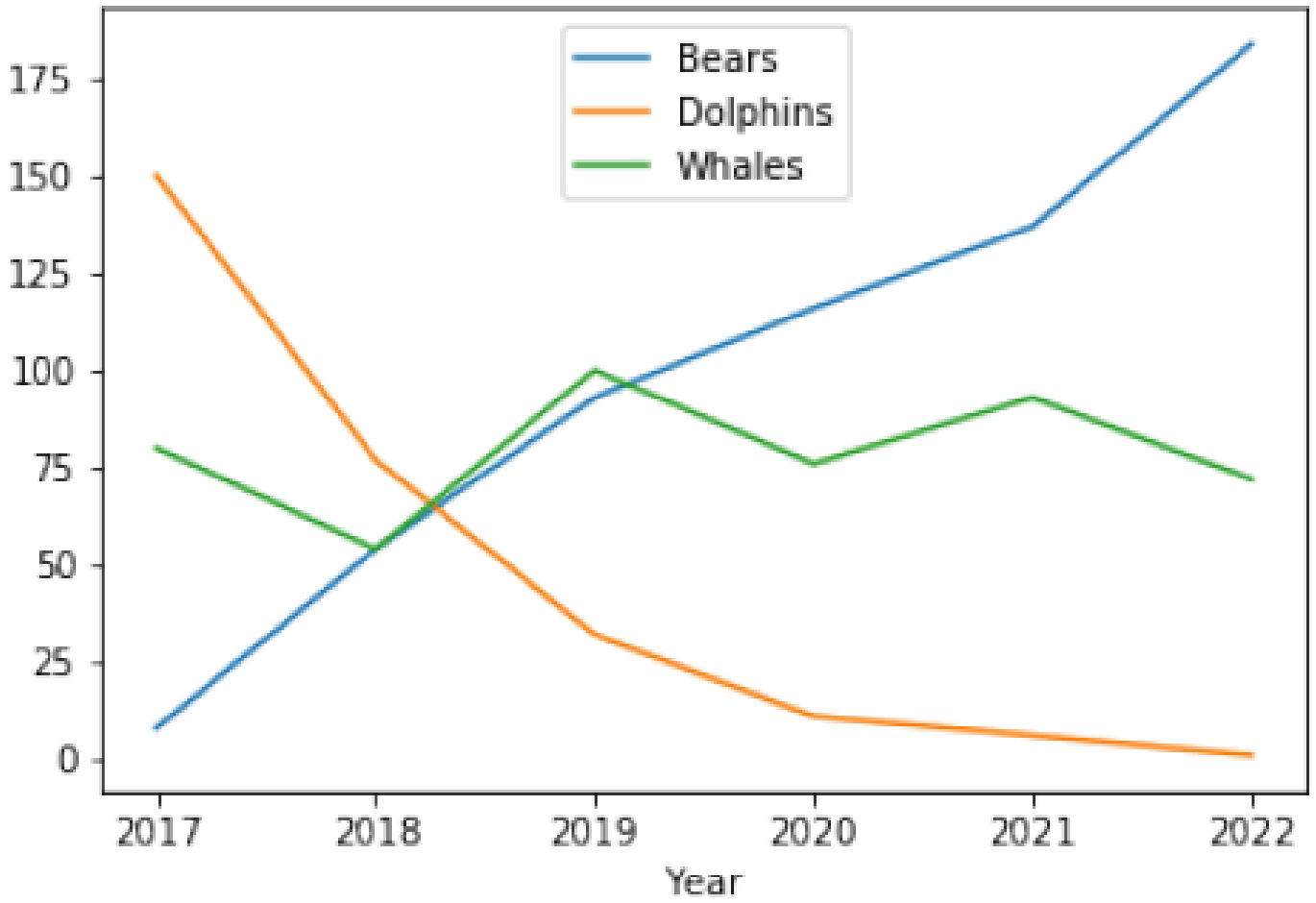


5.6 Line Chart

```
# Pertumbuhan populasi beruang (Bears) dari tahun ke tahun dalam line chart
animal.plot(kind="line",x="Year",y="Bears")
```

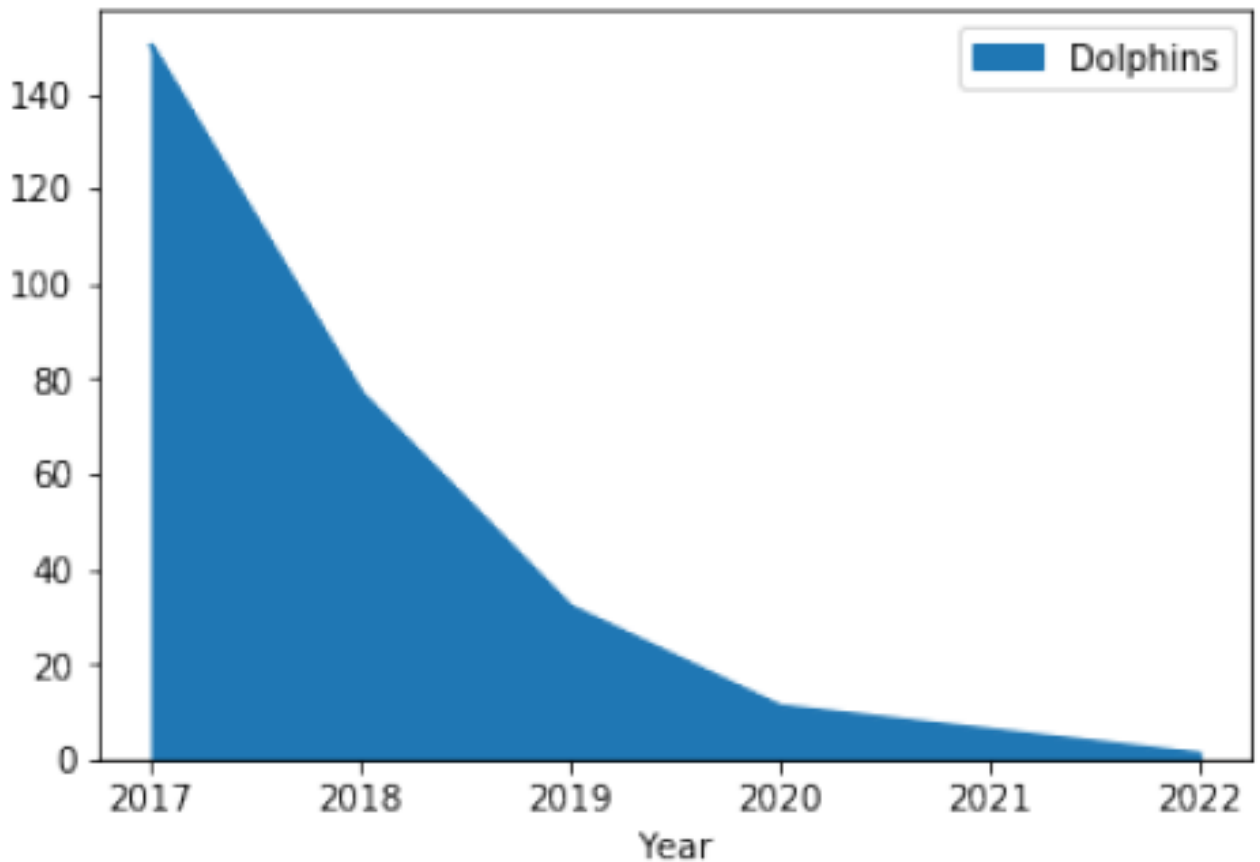


```
# Pertumbuhan populasi beruang (Bears), lumba-lumba (Dolphins), dan ikan paus (
    Whales)
# dari tahun ke tahun dalam 1 line chart
animal.plot(kind="line",x="Year", y=["Bears","Dolphins","Whales"])
```

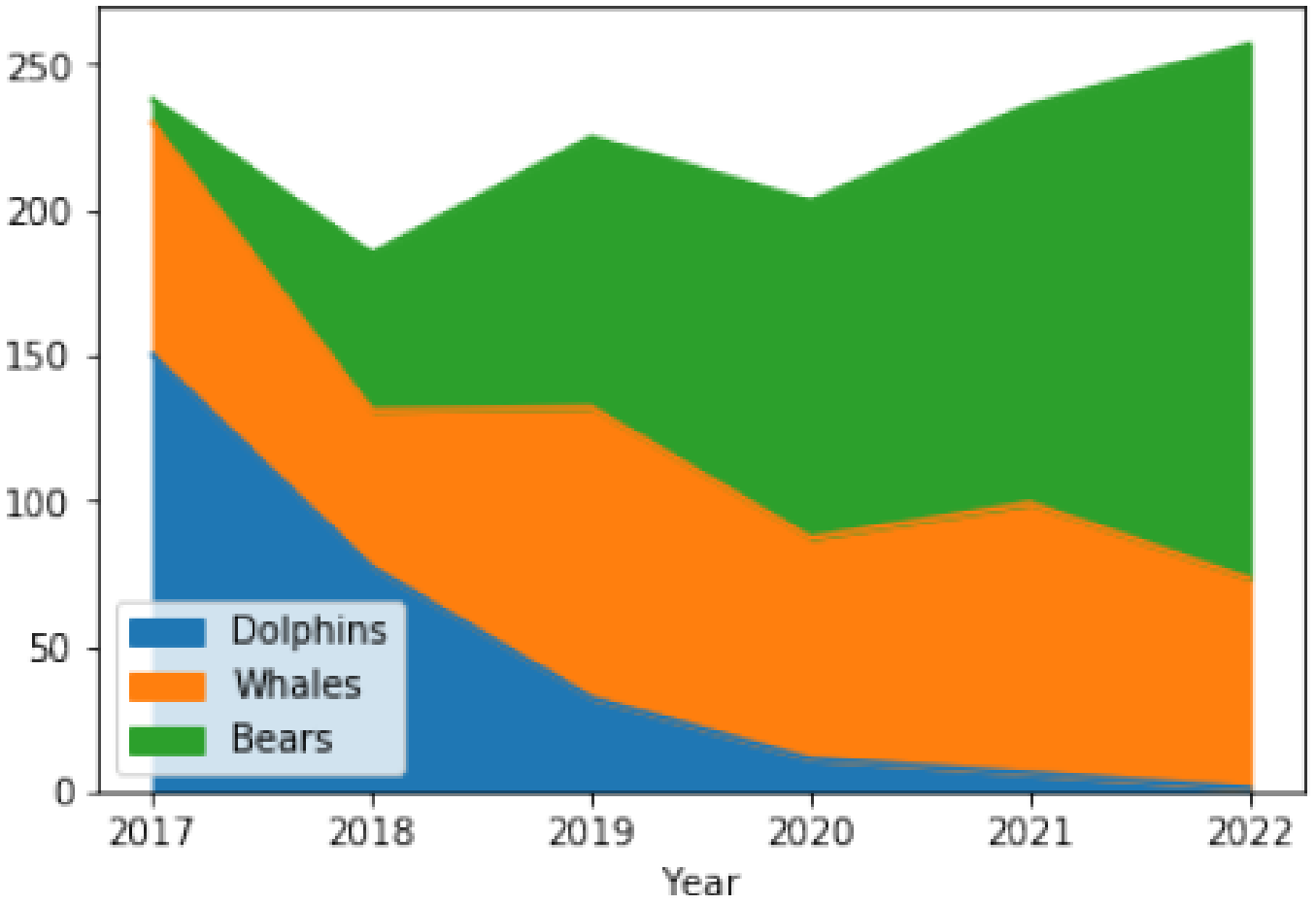


5.7 Area Chart

```
# 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"])
```



5.8 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")
```