Python - Analiza danych z modułem PANDAS

www.udemy.com (http://www.udemy.com) (R)

LAB - S08-L005 - metoda plot 1

1. Zaimportuj **moduł pandas, numpy i matplotlib** i nadaj im standardowe aliasy. Dodaj instrukcję powodującą wyświetlenie wykresu generowanego przez matplotlib w jupyter notebook.
2. Uruchom poniższy fragment kodu, aby przygotować dane do rysowania wykresu (wszystkie zastosowane polecenia powinny Ci już być na tym etapie znane):

```
 marathon = pd.read_csv("./marathon_results_2017.csv", usecols=["Ag
e","M/F","Country","40K"])
 marathon["TimeSeconds"] = pd.to_timedelta(marathon["40K"], errors='coer
ce').dt.total_seconds()
 groupMF = marathon[ marathon["Country"].isin(["USA","CAN"]) ].groupby(b
y=["Country","M/F"])
 groupMF = groupMF.agg({"TimeSeconds": ["mean"]})
 groupMF = groupMF.unstack()
 groupMF.columns = groupMF.columns.droplevel().droplevel()
 groupMF
```

3. Wyświetl wykres kolumnowy prezentujący prezentujący informacje o średnim czasie potrzebnym do przebiegnięcia maratonu z podziałem na kraje i płeć znajdujące się w **groupMF** (parametry domyślne)
4. Zmień wykres tak, aby kolumny dotyczące płci były ustawione jedna na drugiej (ten wykres nie ma dobrego znaczenia biznesowego, bo prezentujemy sumę wartości średnich...)
5. Zmień wykres tak, aby słupki były ułożone poziomo
6. Wykonaj poniższy kod, aby przygotować dane do kolejnego wykresu:

```
age_data = marathon[ marathon["Country"].isin(["USA"]) ]
age_data.head()
```

7. Na podstawie danych z **age_data** dla kolumny **Age** wygeneruj wykres histogramu. Przed wygenerowaniem wykresu możesz spróbować odgadnąć jaki jest wiek biegaczy, a potem porównać go z wynikiem widocznym na wykresie. Ciekawe czy zgadniesz...
8. Ponieważ na poprzednim wykresie słupki nie ilustrują każdego wieku biegacza (są sumaryczne):
   - policz ilość unikalnych wartości w kolumnie **Age**
   - odpowienim parametrem spraw, żeby ilość słupków histogramu była równa wyznaczonej liczbie
9. Wykonaj poniższy kod, aby przygotować dane:

```
 data_USA = marathon[ marathon["Country"].isin(["USA"]) ]
 data_USA.head()
```

10. Narysuj wykres pudełkowy dla **data_USA** ilustrując wartości z kolumny **Age**
11. Wykonaj poniższy kod, aby przygotować dane:

```
USA = marathon[ marathon["Country"].isin(["USA"]) ]["Age"]
CAN = marathon[ marathon["Country"].isin(["CAN"]) ]["Age"]
df = pd.DataFrame({'USA':USA, 'CAN':CAN})
df.head()
```

Rozwiązania:

Poniżej znajdują się propozycje rozwiązań zadań. Prawdopodobnie istnieje wiele dobrych rozwiązań, dlatego jeżeli rozwiązujesz zadania samodzielnie, to najprawdopodobniej zrobisz to inaczej, może nawet lepiej :) Możesz pochwalić się swoimi rozwiązaniami w sekcji Q&A

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib as plt
%matplotlib inline
```

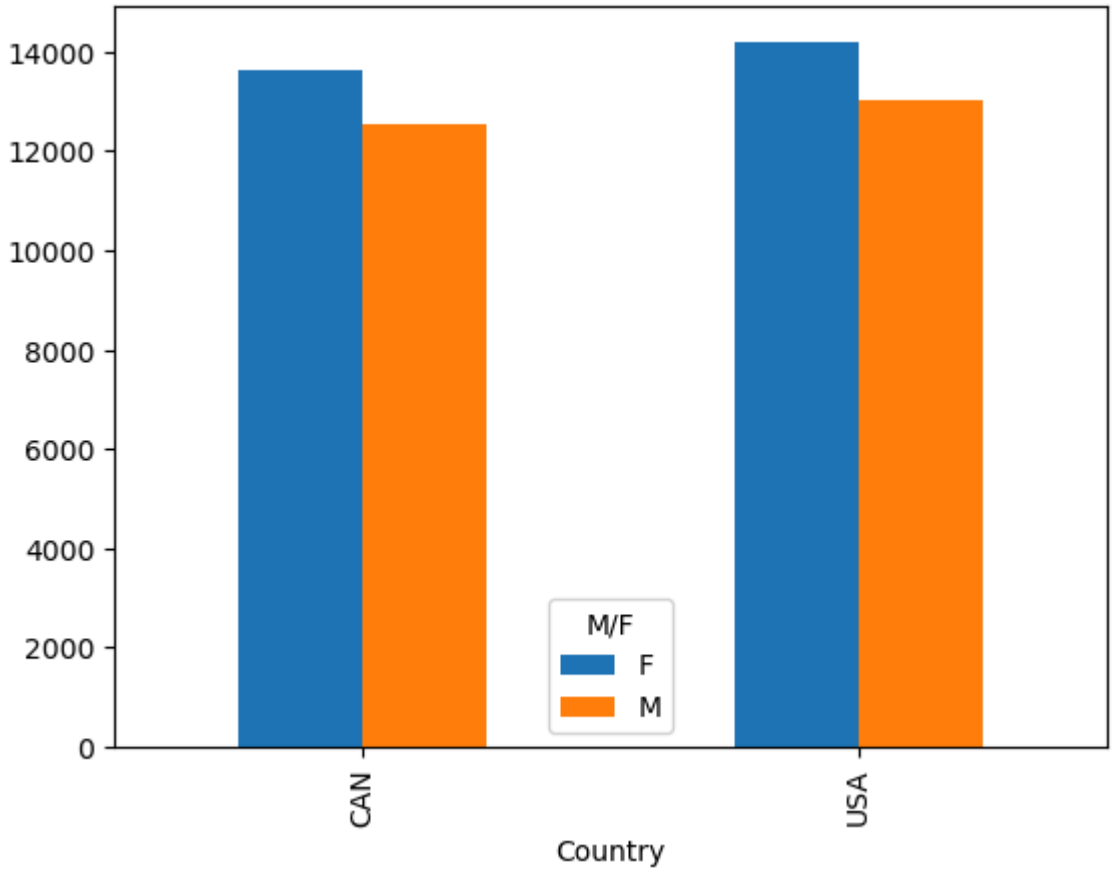
In [2]:
```
marathon = pd.read_csv("./marathon_results_2017.csv", usecols=["Age","M/F","Country"
#marathon["TimeSeconds"] = marathon["40K"].apply(lambda x: pd.Timedelta(x).total_sec
marathon["TimeSeconds"] = pd.to_timedelta(marathon["40K"], errors='coerce').dt.total_
groupMF = marathon[ marathon["Country"].isin(["USA","CAN"]) ].groupby(by=["Country",
groupMF = groupMF.agg({"TimeSeconds": ["mean"]})
groupMF = groupMF.unstack()
groupMF.columns = groupMF.columns.droplevel().droplevel()
groupMF
```

Out[2]:

M/F	F	M
Country		
CAN	13618.346012	12527.160190
USA	14188.759649	13004.157656

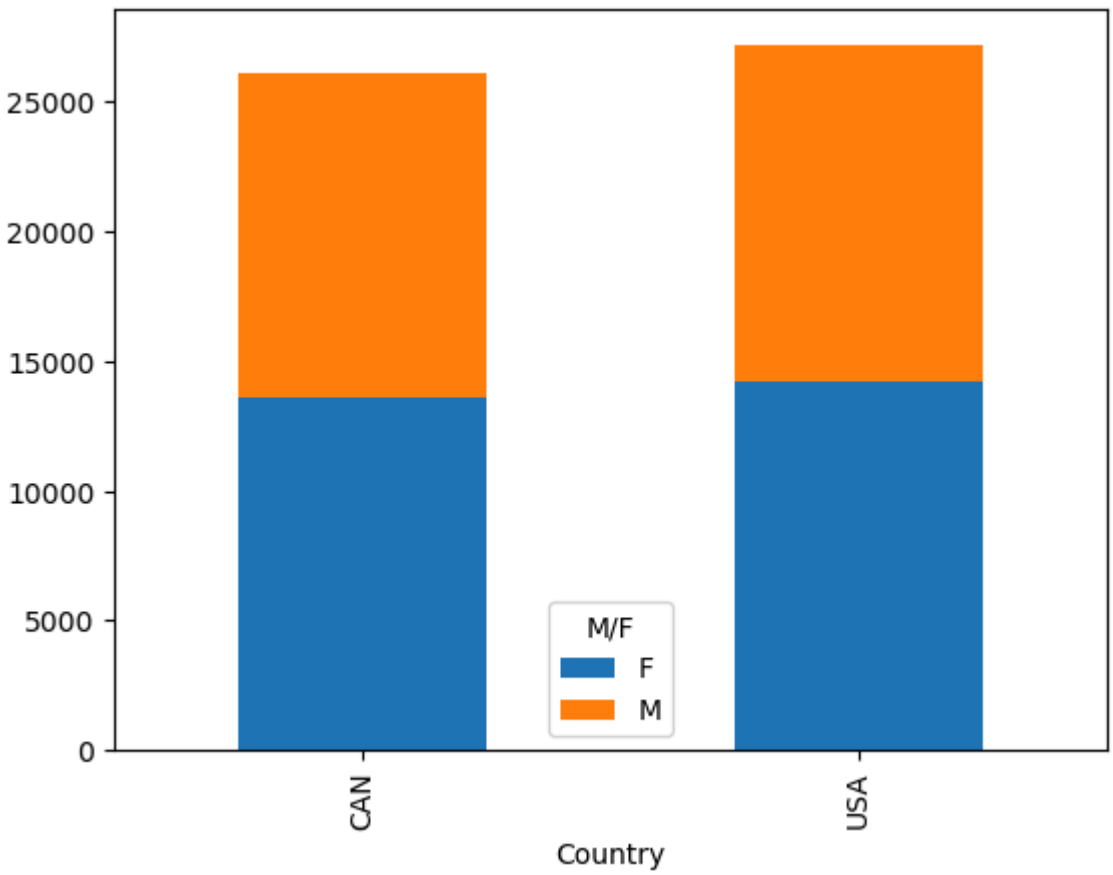
In [3]:
```
groupMF.plot(kind="bar")
```

Out[3]: <Axes: xlabel='Country'>



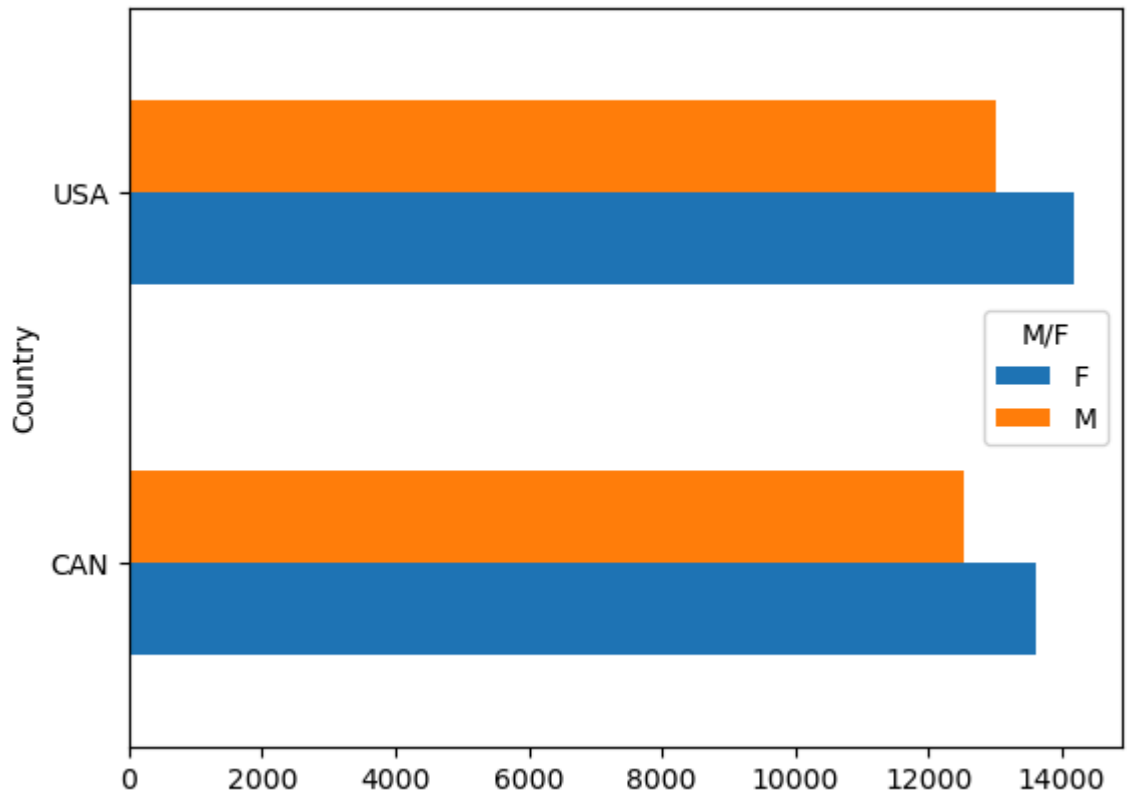
In [4]:
```
groupMF.plot(kind="bar", stacked=True)
```

Out[4]: <Axes: xlabel='Country'>



In [5]:
```
groupMF.plot(kind="barh")
```

Out[5]: <Axes: ylabel='Country'>



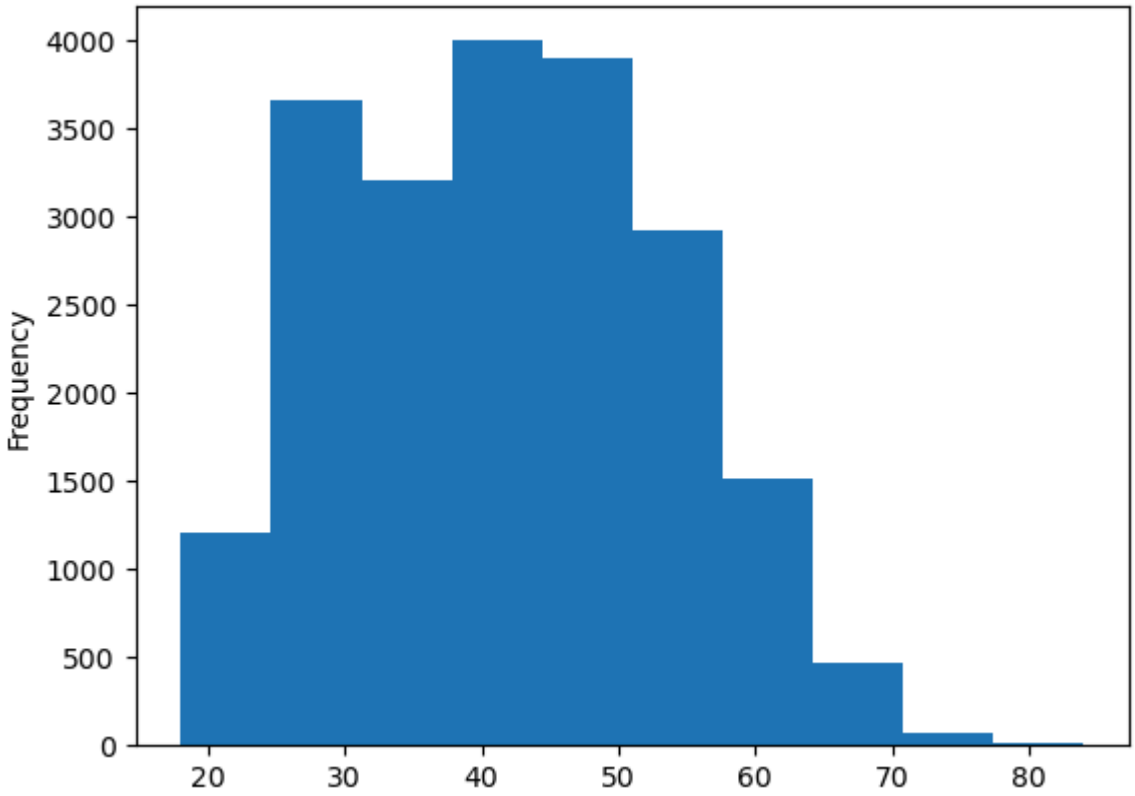
In [6]:
```
age_data = marathon[ marathon["Country"].isin(["USA"]) ]
age_data.head()
```

Out[6]:

	Age	M/F	Country	40K	TimeSeconds
1	30	M	USA	2:03:14	7394.0
3	32	M	USA	2:04:35	7475.0
5	40	M	USA	2:05:21	7521.0
6	33	M	USA	2:05:41	7541.0
8	27	M	USA	2:07:17	7637.0

In [7]:
```
age_data["Age"].plot(kind="hist")
```

Out[7]:
```
<Axes: ylabel='Frequency'>
```

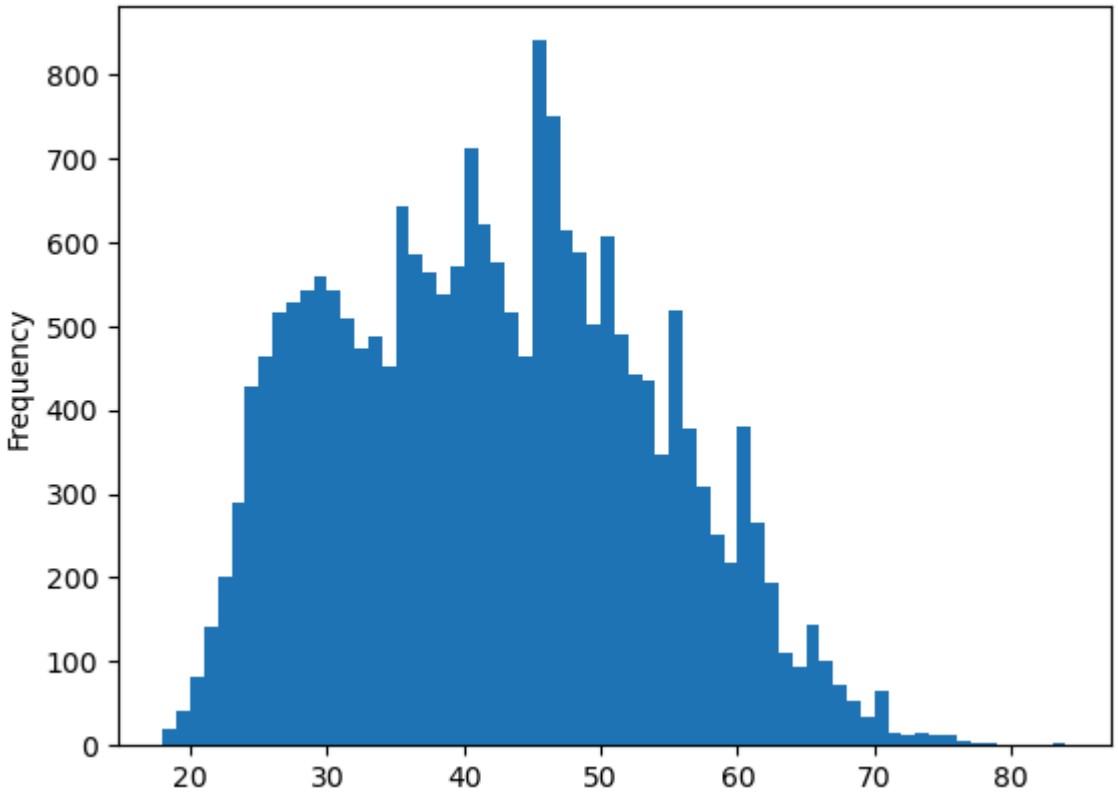


In [8]:
```
age_Count = age_data["Age"].nunique()
age_Count
```

Out[8]:
```
66
```

In [9]:
```
age_data["Age"].plot(kind="hist",bins=age_Count)
```

Out[9]:
```
<Axes: ylabel='Frequency'>
```



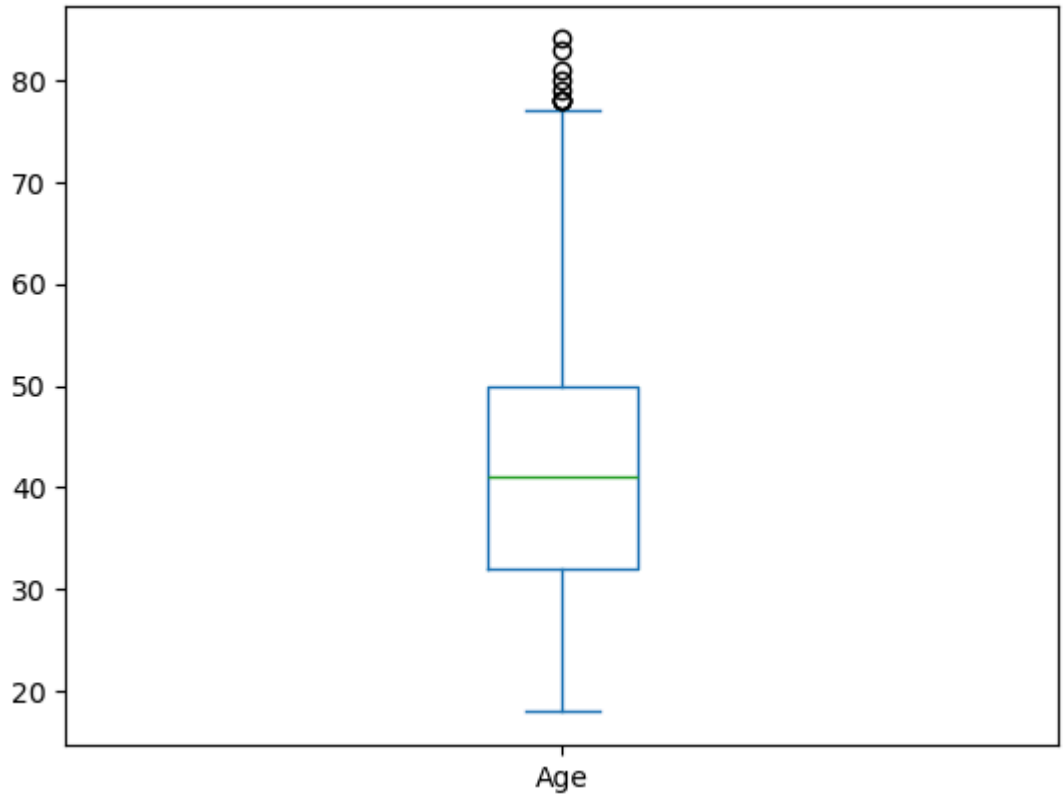
In [10]:
```
data_USA = marathon[ marathon["Country"].isin(["USA"]) ]
data_USA.head()
```

Out[10]:

	Age	M/F	Country	40K	TimeSeconds
1	30	M	USA	2:03:14	7394.0
3	32	M	USA	2:04:35	7475.0
5	40	M	USA	2:05:21	7521.0
6	33	M	USA	2:05:41	7541.0
8	27	M	USA	2:07:17	7637.0

In [11]:
```
data_USA.plot(kind='box',y="Age")
```

Out[11]: <Axes: >



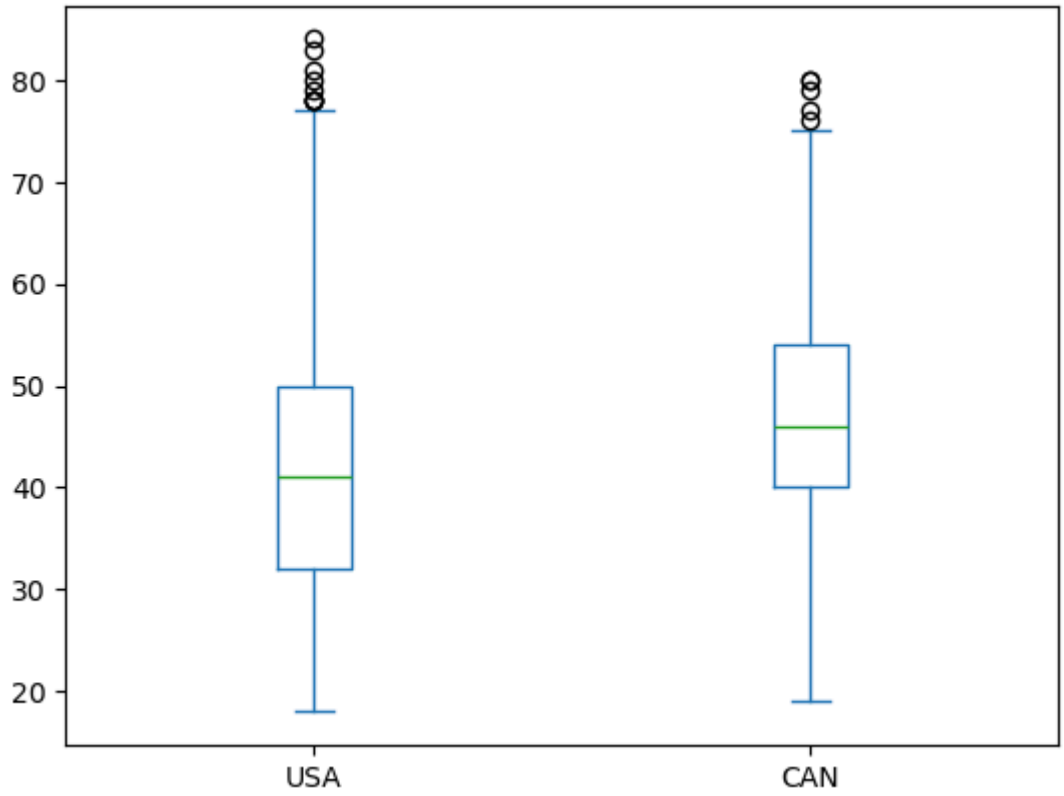
In [12]:
```
USA = marathon[ marathon["Country"].isin(["USA"]) ]["Age"]
CAN = marathon[ marathon["Country"].isin(["CAN"]) ]["Age"]

df = pd.DataFrame({'USA':USA, 'CAN':CAN})

df.head()
```

Out[12]:

	USA	CAN
1	30.0	NaN
3	32.0	NaN
5	40.0	NaN
6	33.0	NaN
8	27.0	NaN

In [13]:
```
df.plot(kind = 'box')
```

Out[13]: <Axes: >



In [ ]: