In [2]:
```
import pandas as p
```

In [2]:
```
p.read_csv(r"C:\Users\V14De\Downloads\jamesbond - jamesbond.csv")
```

Out[2]:

	Film	Year	Actor	Director	Box Office	Budget	Bond Actor Salary
0	Dr. No	1962	Sean Connery	Terence Young	448.8	7.0	0.6
1	From Russia with Love	1963	Sean Connery	Terence Young	543.8	12.6	1.6
2	Goldfinger	1964	Sean Connery	Guy Hamilton	820.4	18.6	3.2
3	Thunderball	1965	Sean Connery	Terence Young	848.1	41.9	4.7
4	Casino Royale	1967	David Niven	Ken Hughes	315.0	85.0	NaN
5	You Only Live Twice	1967	Sean Connery	Lewis Gilbert	514.2	59.9	4.4
6	On Her Majesty's Secret Service	1969	George Lazenby	Peter R. Hunt	291.5	37.3	0.6
7	Diamonds Are Forever	1971	Sean Connery	Guy Hamilton	442.5	34.7	5.8
8	Live and Let Die	1973	Roger Moore	Guy Hamilton	460.3	30.8	NaN
9	The Man with the Golden Gun	1974	Roger Moore	Guy Hamilton	334.0	27.7	NaN
10	The Spy Who Loved Me	1977	Roger Moore	Lewis Gilbert	533.0	45.1	NaN
11	Moonraker	1979	Roger Moore	Lewis Gilbert	535.0	91.5	NaN
12	For Your Eyes Only	1981	Roger Moore	John Glen	449.4	60.2	NaN
13	Never Say Never Again	1983	Sean Connery	Irvin Kershner	380.0	86.0	NaN
14	Octopussy	1983	Roger Moore	John Glen	373.8	53.9	7.8
15	A View to a Kill	1985	Roger Moore	John Glen	275.2	54.5	9.1
16	The Living Daylights	1987	Timothy Dalton	John Glen	313.5	68.8	5.2
17	Licence to Kill	1989	Timothy Dalton	John Glen	250.9	56.7	7.9
18	GoldenEye	1995	Pierce Brosnan	Martin Campbell	518.5	76.9	5.1
19	Tomorrow Never Dies	1997	Pierce Brosnan	Roger Spottiswoode	463.2	133.9	10.0
20	The World Is Not Enough	1999	Pierce Brosnan	Michael Apted	439.5	158.3	13.5
21	Die Another Day	2002	Pierce Brosnan	Lee Tamahori	465.4	154.2	17.9
22	Casino Royale	2006	Daniel Craig	Martin Campbell	581.5	145.3	3.3
23	Quantum of Solace	2008	Daniel Craig	Marc Forster	514.2	181.4	8.1

	Film	Year	Actor	Director	Box Office	Budget	Bond Actor Salary
24	Skyfall	2012	Daniel Craig	Sam Mendes	943.5	170.2	14.5
25	Spectre	2015	Daniel Craig	Sam Mendes	726.7	206.3	30.0
26	No Time to Die	2021	Daniel Craig	Cary Joji Fukunaga	774.2	301.0	25.0

In [3]:
```
j=p.read_csv(r"C:\Users\V14De\Downloads\jamesbond - jamesbond.csv")
```

In [9]:
```
j.describe()
```

Out[9]:

	Year	Box Office	Budget	Bond Actor Salary
count	27.000000	27.000000	27.000000	20.000000
mean	1984.962963	502.077778	88.877778	8.915000
std	17.526862	181.640790	70.227615	7.855891
min	1962.000000	250.900000	7.000000	0.600000
25%	1970.000000	376.900000	39.600000	4.125000
50%	1983.000000	463.200000	60.200000	6.800000
75%	1998.000000	539.400000	139.600000	10.875000
max	2021.000000	943.500000	301.000000	30.000000

In [7]:
```
j.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 27 entries, 0 to 26
Data columns (total 7 columns):
 #   Column             Non-Null Count  Dtype  
---  ------             --------------  -----  
 0   Film               27 non-null     object 
 1   Year               27 non-null     int64  
 2   Actor              27 non-null     object 
 3   Director           27 non-null     object 
 4   Box Office         27 non-null     float64
 5   Budget             27 non-null     float64
 6   Bond Actor Salary  20 non-null     float64
dtypes: float64(3), int64(1), object(3)
memory usage: 1.6+ KB
```

In [13]:
```
j.Bond Actor Salary.describe()
```

```
  Cell In[13], line 1
    j.Bond Actor Salary.describe()
           ^
SyntaxError: invalid syntax
```

In [3]:
```
import numpy as n
```

In [5]:
```
d={"Region":["Asia","Asia","Africa","Europe"],"Country":["India","Japan","South Africa
```

In [6]:
```
df=p.DataFrame(d)
```

In [17]:
```
df
```

Out[17]:

	Region	Country	Year	Sales	Profit
0	Asia	India	2020	20000	2000
1	Asia	Japan	2022	30000	3000
2	Africa	South Africa	2023	25000	4000
3	Europe	France	2025	30000	5000

In [18]:
```
df.groupby("Region")["Profit","Year"].mean()
```

```
C:\Users\V14De\AppData\Local\Temp\ipykernel_20644\3214555300.py:1: FutureWarning: Ind
exing with multiple keys (implicitly converted to a tuple of keys) will be deprecate
d, use a list instead.
  df.groupby("Region")["Profit","Year"].mean()
```

Out[18]:

	Profit	Year
Region		
Africa	4000.0	2023.0
Asia	2500.0	2021.0
Europe	5000.0	2025.0

In [7]:
```
df[(df["Sales"]>300)]
```

Out[7]:

	Region	Country	Year	Sales	Profit
0	Asia	India	2020	20000	2000
1	Asia	Japan	2022	30000	3000
2	Africa	South Africa	2023	25000	4000
3	Europe	France	2025	30000	5000

In [8]:
```
ice_cream=["strawberry","chocolate","vanilla","ferrero rocher"]
```

In [9]:
```
p.Series(ice_cream)
```

Out[9]:
```
0        strawberry
1         chocolate
2           vanilla
3    ferrero rocher
dtype: object
```

In [10]:
```
lotteryno=[2345,9876,567,123,4858]
p.Series(lotteryno)
```

Out[10]:
```
0    2345
1    9876
2     567
3     123
4    4858
dtype: int64
```

In [11]:
```
sushi={'tuna':'red','salmon':'orange','prawn':'light orange'}
p.Series(sushi)
```

Out[11]:
```
tuna                 red
salmon            orange
prawn       light orange
dtype: object
```

In [13]:
```
price=p.Series([23.5,45,67,34,86])
```

In [14]:
```
price
```

Out[14]:
```
0    23.5
1    45.0
2    67.0
3    34.0
4    86.0
dtype: float64
```

In [15]:
```
price.sum()
```

Out[15]:
```
255.5
```

In [16]:
```
price.mean()
```

Out[16]:
```
51.1
```

In [17]:
```
price.median()
```

Out[17]:
```
45.0
```

In [18]:
```
price.product()
```

Out[18]:
```
207172710.0
```

In [19]:
```
price.std()
```

Out[19]:
```
25.309089276384483
```

In [20]:
```
a=p.Series(['smart','pretty','smart','happy'])
a
```

Out[20]:
```
0     smart
1    pretty
2     smart
3     happy
dtype: object
```

In [21]:
```
a.values
```

```
Out[21]: array(['smart', 'pretty', 'smart', 'happy'], dtype=object)
```

```
In [22]: a.is_unique
```

```
Out[22]: False
```

```
In [23]: type(a.values)
```

```
Out[23]: numpy.ndarray
```

```
In [24]: wd=['monday','tuesday','wednesday','thursday','friday','saturday','sunday']
         fr=['kiwi','orange','grapes','guava','pineapple','strawberry','mango']
```

```
In [27]: p.Series(fr,wd) #the index comes first even if it is given as the secind parameter
```

```
Out[27]: monday             kiwi
         tuesday          orange
         wednesday        grapes
         thursday          guava
         friday        pineapple
         saturday     strawberry
         sunday            mango
         dtype: object
```

```
In [28]: p.Series(index=wd,data=fr)
```

```
Out[28]: monday             kiwi
         tuesday          orange
         wednesday        grapes
         thursday          guava
         friday        pineapple
         saturday     strawberry
         sunday            mango
         dtype: object
```

```
In [ ]:
```