Taller de Pandas No 1

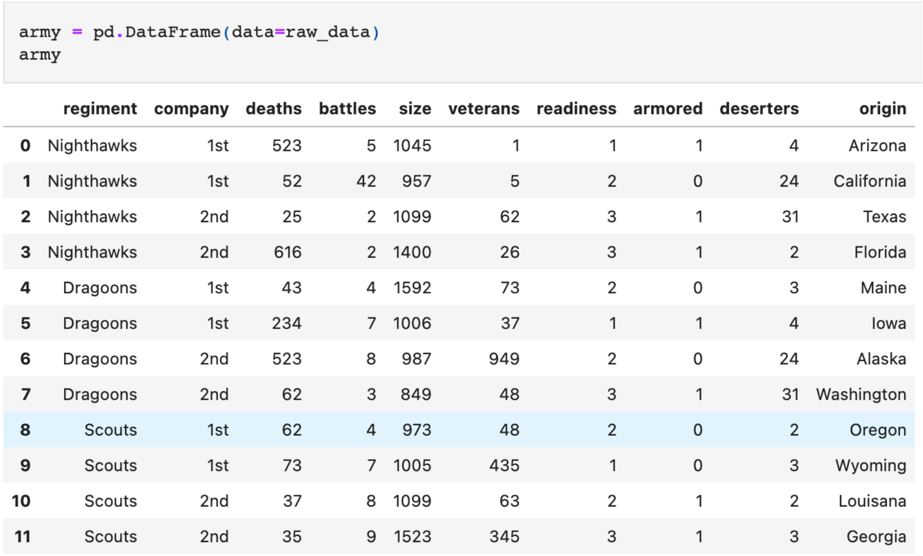
1. Ejercicio básico de manejo de DataFrame con la librería pandas

Creamos el diccionario de datos

```
import pandas as pd

raw_data = {'regiment': ['Nighthawks', 'Nighthawks', 'Nighthawks', 'Nighthawks', 'Dragoons',
'Dragoons', 'Dragoons', 'Dragoons', 'Scouts', 'Scouts', 'Scouts', 'Scouts'],
            'company': ['1st', '1st', '2nd', '2nd', '1st', '1st', '2nd', '2nd','1st', '1st',
'2nd', '2nd'],
            'deaths': [523, 52, 25, 616, 43, 234, 523, 62, 62, 73, 37, 35],
            'battles': [5, 42, 2, 2, 4, 7, 8, 3, 4, 7, 8, 9],
            'size': [1045, 957, 1099, 1400, 1592, 1006, 987, 849, 973, 1005, 1099, 1523],
            'veterans': [1, 5, 62, 26, 73, 37, 949, 48, 48, 435, 63, 345],
            'readiness': [1, 2, 3, 3, 2, 1, 2, 3, 2, 1, 2, 3],
            'armored': [1, 0, 1, 1, 0, 1, 0, 1, 0, 0, 1, 1],
            'deserters': [4, 24, 31, 2, 3, 4, 24, 31, 2, 3, 2, 3],
            'origin': ['Arizona', 'California', 'Texas', 'Florida', 'Maine', 'Iowa', 'Alaska',
'Washington', 'Oregon', 'Wyoming', 'Louisana', 'Georgia']}
```

Cree un marco de datos y asígnelo a una variable llamada ejército. No olvide incluir los nombres de las columnas en el orden presentado en el diccionario ('regimiento', 'compañía', 'muertes'...) para que el orden del índice de las columnas sea consistente con las soluciones. Si se omite, los pandas ordenarán las columnas alfabéticamente.



```
army = pd.DataFrame(data=raw_data)
army
```

	regiment	company	deaths	battles	size	veterans	readiness	armored	deserters	origin
0	Nighthawks	1st	523	5	1045	1	1	1	4	Arizona
1	Nighthawks	1st	52	42	957	5	2	0	24	California
2	Nighthawks	2nd	25	2	1099	62	3	1	31	Texas
3	Nighthawks	2nd	616	2	1400	26	3	1	2	Florida
4	Dragoons	1st	43	4	1592	73	2	0	3	Maine
5	Dragoons	1st	234	7	1006	37	1	1	4	Iowa
6	Dragoons	2nd	523	8	987	949	2	0	24	Alaska
7	Dragoons	2nd	62	3	849	48	3	1	31	Washington
8	Scouts	1st	62	4	973	48	2	0	2	Oregon
9	Scouts	1st	73	7	1005	435	1	0	3	Wyoming
10	Scouts	2nd	37	8	1099	63	2	1	2	Louisana
11	Scouts	2nd	35	9	1523	345	3	1	3	Georgia

Establezca la columna 'origen' como el índice del marco de datos

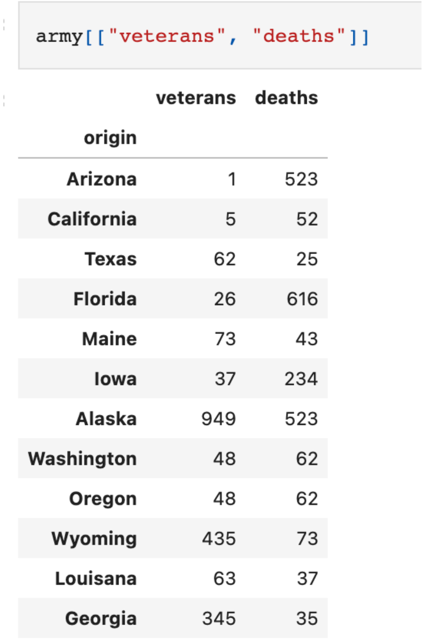
```
army.set_index('origin', inplace=True)
```

Imprime solo la columna veteranos

```
army.veterans
```

```
origin
Arizona          1
California       5
Texas           62
Florida         26
Maine           73
Iowa            37
Alaska         949
Washington      48
Oregon          48
Wyoming        435
Louisana        63
Georgia        345
Name: veterans, dtype: int64
```

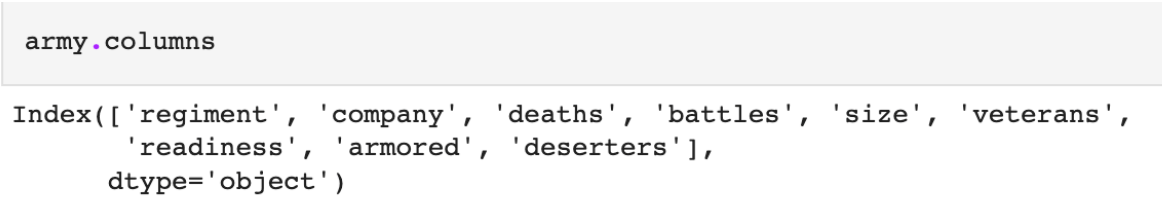
Imprime las columnas 'veteranos' y 'muertes'.



```
army[["veterans", "deaths"]]
```

	veterans	deaths
origin		
Arizona	1	523
California	5	52
Texas	62	25
Florida	26	616
Maine	73	43
Iowa	37	234
Alaska	949	523
Washington	48	62
Oregon	48	62
Wyoming	435	73
Louisana	63	37
Georgia	345	35

Nombres de columnas



```
army.columns
```

```
Index(['regiment', 'company', 'deaths', 'battles', 'size', 'veterans',
       'readiness', 'armored', 'deserters'],
      dtype='object')
```

Seleccione las columnas 'muertes', 'tamaño' y 'desertores' de Maine y Alaska

```
army.loc[["Maine", "Alaska"], ["deaths", "size", "deserters"]]
```

origin	deaths	size	deserters
Maine	43	1592	3
Alaska	523	987	24

Seleccione las filas 3 a 7 y las columnas 3 a 6

```
army.iloc[2:7, 2:6]
```

origin	deaths	battles	size	veterans
Texas	25	2	1099	62
Florida	616	2	1400	26
Maine	43	4	1592	73
Iowa	234	7	1006	37
Alaska	523	8	987	949

Seleccione cada fila después de la cuarta fila y todas las columnas

```
army.iloc[4:, :]
```

origin	regiment	company	deaths	battles	size	veterans	readiness	armored	deserters
Maine	Dragoons	1st	43	4	1592	73	2	0	3
Iowa	Dragoons	1st	234	7	1006	37	1	1	4
Alaska	Dragoons	2nd	523	8	987	949	2	0	24
Washington	Dragoons	2nd	62	3	849	48	3	1	31
Oregon	Scouts	1st	62	4	973	48	2	0	2
Wyoming	Scouts	1st	73	7	1005	435	1	0	3
Louisana	Scouts	2nd	37	8	1099	63	2	1	2
Georgia	Scouts	2nd	35	9	1523	345	3	1	3

Selecciona cada fila hasta la 4ta fila y todas las columnas

```
army.iloc[:4, :]
```

	regiment	company	deaths	battles	size	veterans	readiness	armored	deserters
origin									
Arizona	Nighthawks	1st	523	5	1045	1	1	1	4
California	Nighthawks	1st	52	42	957	5	2	0	24
Texas	Nighthawks	2nd	25	2	1099	62	3	1	31
Florida	Nighthawks	2nd	616	2	1400	26	3	1	2

Seleccionar filas donde df.deaths sea mayor que 50

```
army[army["deaths"] > 50]
```

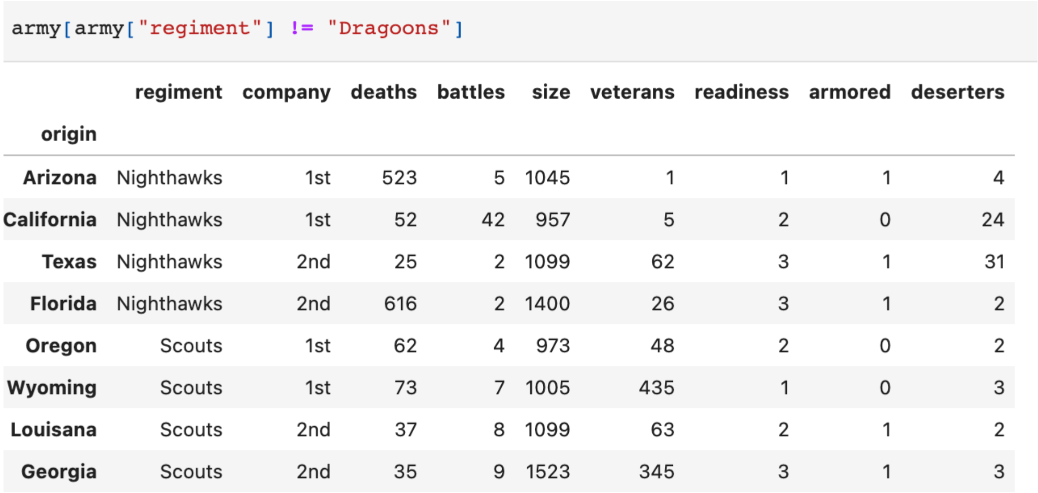
	regiment	company	deaths	battles	size	veterans	readiness	armored	deserters
origin									
Arizona	Nighthawks	1st	523	5	1045	1	1	1	4
California	Nighthawks	1st	52	42	957	5	2	0	24
Florida	Nighthawks	2nd	616	2	1400	26	3	1	2
Iowa	Dragoons	1st	234	7	1006	37	1	1	4
Alaska	Dragoons	2nd	523	8	987	949	2	0	24
Washington	Dragoons	2nd	62	3	849	48	3	1	31
Oregon	Scouts	1st	62	4	973	48	2	0	2
Wyoming	Scouts	1st	73	7	1005	435	1	0	3

Seleccionar filas donde df.deaths sea mayor que 500 o menor que 50

```
army[(army["deaths"] > 500) | (army["deaths"] < 50)]
```

	regiment	company	deaths	battles	size	veterans	readiness	armored	deserters
origin									
Arizona	Nighthawks	1st	523	5	1045	1	1	1	4
Texas	Nighthawks	2nd	25	2	1099	62	3	1	31
Florida	Nighthawks	2nd	616	2	1400	26	3	1	2
Maine	Dragoons	1st	43	4	1592	73	2	0	3
Alaska	Dragoons	2nd	523	8	987	949	2	0	24
Louisana	Scouts	2nd	37	8	1099	63	2	1	2
Georgia	Scouts	2nd	35	9	1523	345	3	1	3

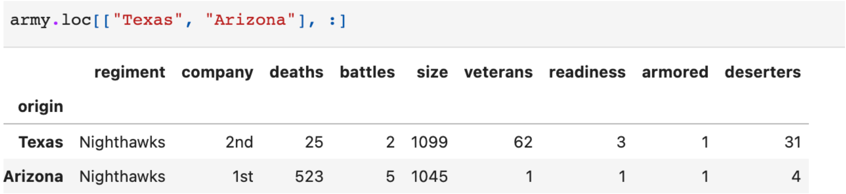
Seleccione todos los regimientos que no se llamen "Dragoons"



```
army[army["regiment"] != "Dragoons"]
```

	regiment	company	deaths	battles	size	veterans	readiness	armored	deserters
origin									
Arizona	Nighthawks	1st	523	5	1045	1	1	1	4
California	Nighthawks	1st	52	42	957	5	2	0	24
Texas	Nighthawks	2nd	25	2	1099	62	3	1	31
Florida	Nighthawks	2nd	616	2	1400	26	3	1	2
Oregon	Scouts	1st	62	4	973	48	2	0	2
Wyoming	Scouts	1st	73	7	1005	435	1	0	3
Louisana	Scouts	2nd	37	8	1099	63	2	1	2
Georgia	Scouts	2nd	35	9	1523	345	3	1	3

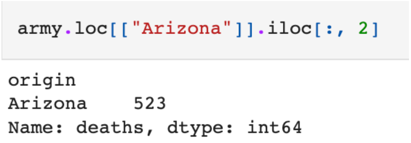
Seleccione las filas llamadas Texas y Arizona



```
army.loc[["Texas", "Arizona"], :]
```

	regiment	company	deaths	battles	size	veterans	readiness	armored	deserters
origin									
Texas	Nighthawks	2nd	25	2	1099	62	3	1	31
Arizona	Nighthawks	1st	523	5	1045	1	1	1	4

Seleccione la tercera celda en la fila llamada Arizona



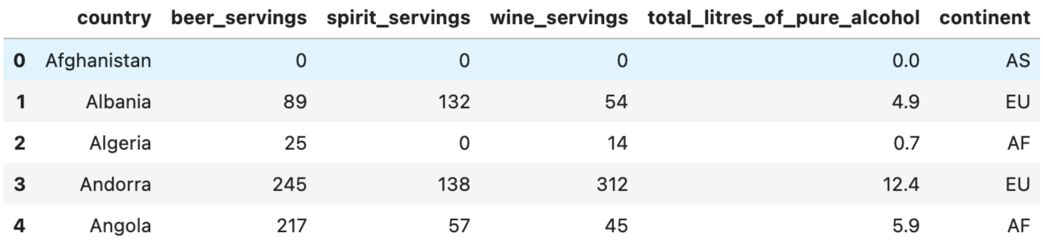
```
army.loc[["Arizona"]].iloc[:, 2]
```

```
origin
Arizona    523
Name: deaths, dtype: int64
```

2. GroupBy en Pandas. Utilización data de consumo de alcohol - https://raw.githubusercontent.com/justmarkham/DAT8/master/data/drinks.csv

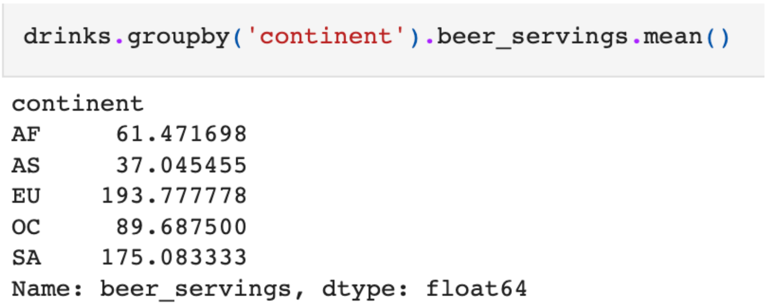
Cargamos la Data:

```
drinks = pd.read_csv('https://raw.githubusercontent.com/justmarkham/DAT8/master/data/drinks.csv')
drinks.head()
```



	country	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol	continent
0	Afghanistan	0	0	0	0.0	AS
1	Albania	89	132	54	4.9	EU
2	Algeria	25	0	14	0.7	AF
3	Andorra	245	138	312	12.4	EU
4	Angola	217	57	45	5.9	AF

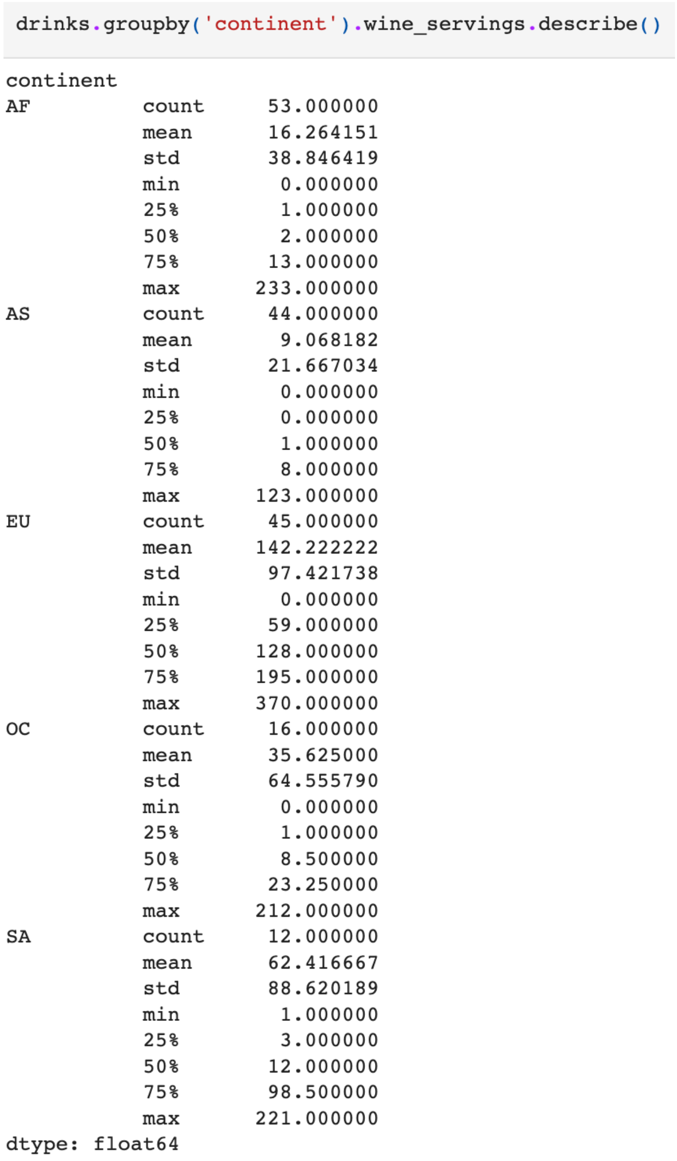
¿Qué continente bebe más cerveza en promedio?



```
drinks.groupby('continent').beer_servings.mean()
```

```
continent
AF     61.471698
AS     37.045455
EU    193.777778
OC     89.687500
SA    175.083333
Name: beer_servings, dtype: float64
```

Para cada continente imprime las estadísticas de consumo de vino.



```
drinks.groupby('continent').wine_servings.describe()
```

```
continent
AF         count     53.000000
           mean      16.264151
           std       38.846419
           min        0.000000
           25%        1.000000
           50%        2.000000
           75%       13.000000
           max      233.000000
AS         count     44.000000
           mean       9.068182
           std       21.667034
           min        0.000000
           25%        0.000000
           50%        1.000000
           75%        8.000000
           max      123.000000
EU         count     45.000000
           mean     142.222222
           std       97.421738
           min        0.000000
           25%       59.000000
           50%      128.000000
           75%      195.000000
           max      370.000000
OC         count     16.000000
           mean      35.625000
           std       64.555790
           min        0.000000
           25%        1.000000
           50%        8.500000
           75%       23.250000
           max      212.000000
SA         count     12.000000
           mean      62.416667
           std       88.620189
           min        1.000000
           25%        3.000000
           50%       12.000000
           75%       98.500000
           max      221.000000
dtype: float64
```

Imprime el consumo medio de alcohol por continente para cada columna

```
drinks.groupby('continent').mean()
```

	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol
continent				
AF	61.471698	16.339623	16.264151	3.007547
AS	37.045455	60.840909	9.068182	2.170455
EU	193.777778	132.555556	142.222222	8.617778
OC	89.687500	58.437500	35.625000	3.381250
SA	175.083333	114.750000	62.416667	6.308333

Imprime la mediana del consumo de alcohol por continente para cada columna

```
drinks.groupby('continent').median()
```

	beer_servings	spirit_servings	wine_servings	total_litres_of_pure_alcohol
continent				
AF	32.0	3.0	2.0	2.30
AS	17.5	16.0	1.0	1.20
EU	219.0	122.0	128.0	10.00
OC	52.5	37.0	8.5	1.75
SA	162.5	108.5	12.0	6.85

Imprime los valores medio, mínimo y máximo para el consumo de bebidas espirituosas.

Esta vez genera un DataFrame

```
drinks.groupby('continent').spirit_servings.agg(['mean', 'min', 'max'])
```

	mean	min	max
continent			
AF	16.339623	0	152
AS	60.840909	0	326
EU	132.555556	0	373
OC	58.437500	0	254
SA	114.750000	25	302