

Fictional Army - Filtering and Sorting

Introduction:

This exercise was inspired by [this page](#)

Special thanks to: <https://github.com/chrisaibon> for sharing the dataset and materials.

Step 1. Import the necessary libraries

```
In [1]: import pandas as pd
```

Step 2. This is the data given as a dictionary

```
In [3]: # Create an example dataframe about a fictional army
raw_data = {'regiment': ['Nighthawks', 'Nighthawks', 'Nighthawks', 'Nighthawks', 'Dragoons', 'Dragoons', 'Dragoons', 'Dragoons',
                        'company': [[1st, 1st, 2nd, 2nd, 1st, 1st, 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, 1086, 987, 849, 973, 1805, 1899, 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']
```

Step 3. Create a dataframe and assign it to a variable called army.

Don't forget to include the columns names in the order presented in the dictionary ('regiment', 'company', 'deaths'...) so that the column index order is consistent with the solutions. If omitted, pandas will order the columns alphabetically.

```
In [27]: ar = pd.DataFrame(raw_data, columns=['regiment', 'company', 'deaths', 'battles', 'size', 'veterans', 'readiness', 'armored', 'deserters', 'origin'])
```

	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

```
In [33]: ar.describe(include="all")
```

	regiment	company	deaths	battles	size	veterans	readiness	armored	deserters	origin
count	12	12	12.000000	12.000000	12.000000	12.000000	12.000000	12.000000	12.000000	12
unique	3	2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	Nighthawks	1st	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Arizona
freq	4	6	NaN	NaN	NaN	NaN	NaN	NaN	NaN	1
mean	NaN	NaN	190.416667	8.416667	1127.916667	174.333333	2.083333	0.583333	11.083333	NaN
std	NaN	NaN	227.027615	10.849871	240.241719	280.254214	0.792961	0.514929	12.324833	NaN
min	NaN	NaN	25.000000	2.000000	849.000000	1.000000	1.000000	0.000000	2.000000	NaN
25%	NaN	NaN	41.500000	3.750000	983.500000	34.250000	1.750000	0.000000	2.750000	NaN
50%	NaN	NaN	62.000000	6.000000	1025.500000	55.000000	2.000000	1.000000	3.500000	NaN
75%	NaN	NaN	306.250000	8.000000	1174.250000	141.000000	3.000000	1.000000	24.000000	NaN
max	NaN	NaN	616.000000	42.000000	1592.000000	949.000000	3.000000	1.000000	31.000000	NaN

Step 4. Set the 'origin' colum as the index of the dataframe

```
In [50]: ar = ar.set_index("origin")
ar
```

	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
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
Louisiana	Scouts	2nd	37	8	1099	63	2	1	2
Georgia	Scouts	2nd	35	9	1523	345	3	1	3

Step 5. Print only the column veterans

```
In [5]: ar["veterans"]
```

0	1
1	5
2	62
3	26
4	73
5	37
6	949
7	48
8	48
9	435
10	63
11	345

Name: veterans, dtype: int64

Step 6. Print the columns 'veterans' and 'deaths'

```
In [17]: ar[["veterans", "deaths"]]
```

	veterans	deaths
0	1	523
1	5	52
2	62	25
3	26	616
4	73	43
5	37	234
6	949	523
7	48	62
8	48	62
9	435	73
10	63	37
11	345	35

Step 7. Print the name of all the columns.

```
In [19]: ar.columns
```

```
Out[19]: Index(['regiment', 'company', 'deaths', 'battles', 'size', 'veterans', 'readiness', 'armored', 'deserters', 'origin'], dtype='object')
```

Step 8. Select the 'deaths', 'size' and 'deserters' columns from Maine and Alaska

```
In [26]: ar.loc[["Maine", "Alaska"], ["deaths", "size", "deserters"]]
```

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

Step 9. Select the rows 3 to 7 and the columns 3 to 6

```
In [29]: ar.iloc[3:7, 3:6]
```

	battles	size	veterans
3	2	1400	26
4	4	1592	73
5	7	1006	37
6	8	987	949

Step 10. Select every row after the fourth row and all columns

```
In [30]: ar.loc[3:]
```

	regiment	company	deaths	battles	size	veterans	readiness	armored	deserters	origin
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	Louisiana
11	Scouts	2nd	35	9	1523	345	3	1	3	Georgia

Step 11. Select every row up to the 4th row and all columns

```
In [34]: ar.loc[0:3]
```

	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

Step 12. Select the 3rd column up to the 7th column

```
In [36]: ar.iloc[:, 3:7]
```

	battles	size	veterans	readiness
0	5	1045	1	1
1	42	957	5	2
2	2	1099	62	3
3	2	1400	26	3
4	4	1592	73	2
5	7	1006	37	1
6	8	987	949	2
7	3	849	48	3
8	4	973	48	2
9	7	1005	435	1
10	8	1099	63	2
11	9	1523	345	3

Step 13. Select rows where df.deaths is greater than 50

```
In [39]: ar["deaths"]>50
```

0	True
1	True
2	False
3	True
4	False
5	True
6	True
7	True
8	True
9	True
10	False
11	False

Name: deaths, dtype: bool

Step 14. Select rows where df.deaths is greater than 500 or less than 50

```
In [46]: ar[(ar.deaths<50) | (ar.deaths>500)]
```

	regiment	company	deaths	battles	size	veterans	readiness	armored	deserters	origin
0	Nighthawks	1st	523	5	1045	1	1	1	4	Arizona
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
6	Dragoons	2nd	523	8	987	949	2	0	24	Alaska
10	Scouts	2nd	37	8	1099	63	2	1	2	Louisiana
11	Scouts	2nd	35	9	1523	345	3	1	3	Georgia

Step 15. Select all the regiments not named "Dragoons"

```
In [47]: ar[(ar["regiment"] != "Dragoons")]
```

	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
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	Louisiana
11	Scouts	2nd	35	9	1523	345	3	1	3	Georgia

Step 16. Select the rows called Texas and Arizona

```
In [52]: ar.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

Step 17. Select the third cell in the row named Arizona

```
In [53]: ar.loc[["Arizona"], ["deaths"]]
```

	deaths
origin	
Arizona	523

Step 18. Select the third cell down in the column named deaths

```
In [55]: ar.loc[["Texas", "deaths"]]
```

```
Out[55]: 25
```