

# pandas sample\_1

July 5, 2022

## 1 Making a dataframe

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

- Use 'Dictionary' syntax to create a dataframe

```
[ ]: df = pd.DataFrame({'a': [1, 2, 3, 4], 'b' : [5, 6, 7, 8], 'c' : [9, 10, 11, 12]})
```

```
[ ]: df
```

```
[ ]:      a  b   c
0  1  5   9
1  2  6  10
2  3  7  11
3  4  8  12
```

- Use 'List' to create a dataframe

```
[ ]: a = [[1, 4, 7], [2, 5, 8], [3, 6, 9]]
```

```
[ ]: df2 = pd.DataFrame(a)
```

```
[ ]: df2
```

```
[ ]:      0  1  2
0  1  4  7
1  2  5  8
2  3  6  9
```

```
[ ]: df2.columns = ['a', 'b', 'c']
```

- change field names

```
[ ]: df.columns = ['d', 'e', 'f']
```

```
[ ]: df
```

```
[ ]:      d  e   f
      0  1  5   9
      1  2  6  10
      2  3  7  11
      3  4  8  12
```

- copy dataframes

```
[ ]: import copy
```

```
[ ]: df3 = copy.deepcopy(df)
```

```
[ ]: df3
```

```
[ ]:      d  e   f
      0  1  5   9
      1  2  6  10
      2  3  7  11
      3  4  8  12
```

- extracting columns

## 2 Extracting columns

- create a series in pandas

```
[ ]: a = pd.Series([1, 2, 3, 1, 2, 3])
```

```
[ ]: a
```

```
[ ]: 0    1
      1    2
      2    3
      3    1
      4    2
      5    3
      dtype: int64
```

- change index

```
[ ]: a = pd.Series([1, 2, 3, 1, 2, 3], index = ['a', 'b', 'c', 'd', 'e', 'f'])
```

```
[ ]: a
```

```
[ ]: a    1
      b    2
      c    3
      d    1
      e    2
```

```
f      3
dtype: int64
```

### 3 Extracting data with conditions

```
[ ]: df = pd.DataFrame({'a': [i for i in range(1, 6)], 'b' : [i for i in range(6, 11)], 'c' : [i for i in range(11, 16)]})
```

```
[ ]: df
```

```
[ ]:      a  b  c
0  1  6  11
1  2  7  12
2  3  8  13
3  4  9  14
4  5  10 15
```

```
[ ]: df[['a', 'c']]
```

```
[ ]:      a  c
0  1  11
1  2  12
2  3  13
3  4  14
4  5  15
```

```
[ ]: df[df['a'] >= 3]
```

```
[ ]:      a  b  c
2  3  8  13
3  4  9  14
4  5  10 15
```

```
[ ]: df[df['a'] >= 3][['a', 'b']]
```

```
[ ]:      a  b
2  3  8
3  4  9
4  5  10
```

```
[ ]: df[(df['a'] >= 3) & (df['b'] < 16)]
```

```
[ ]:      a  b  c
2  3  8  13
3  4  9  14
4  5  10 15
```

```
[ ]: TF = (df['a'] >= 3) & (df['b'] < 16)
```

```
[ ]: TF
```

```
[ ]: 0    False
     1    False
     2     True
     3     True
     4     True
     dtype: bool
```

```
[ ]: df[TF]
```

```
[ ]:   a  b  c
     2  3  8 13
     3  4  9 14
     4  5 10 15
```

```
[ ]: df.drop(df.index[:4])
```

```
[ ]:   a  b  c
     4  5 10 15
```

## 4 Working with a big dataset

```
[ ]: import pandas as pd
     titanic_df = pd.read_csv('train.csv')
```

```
[ ]: type(titanic_df)
```

```
[ ]: pandas.core.frame.DataFrame
```

```
[ ]: titanic_df
```

```
[ ]:   PassengerId  Survived  Pclass  \
     0           1         0       3
     1           2         1       1
     2           3         1       3
     3           4         1       1
     4           5         0       3
     ..          ...       ...     ...
    886          887         0       2
    887          888         1       1
    888          889         0       3
    889          890         1       1
    890          891         0       3
```

	Name	Sex	Age	SibSp	\
0	Braund, Mr. Owen Harris	male	22.0	1	
1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	
2	Heikkinen, Miss. Laina	female	26.0	0	
3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	
4	Allen, Mr. William Henry	male	35.0	0	
..	...	...	...		
886	Montvila, Rev. Juozas	male	27.0	0	
887	Graham, Miss. Margaret Edith	female	19.0	0	
888	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	
889	Behr, Mr. Karl Howell	male	26.0	0	
890	Dooley, Mr. Patrick	male	32.0	0	

	Parch	Ticket	Fare	Cabin	Embarked
0	0	A/5 21171	7.2500	NaN	S
1	0	PC 17599	71.2833	C85	C
2	0	STON/O2. 3101282	7.9250	NaN	S
3	0	113803	53.1000	C123	S
4	0	373450	8.0500	NaN	S
..	...	...	...	...	
886	0	211536	13.0000	NaN	S
887	0	112053	30.0000	B42	S
888	2	W./C. 6607	23.4500	NaN	S
889	0	111369	30.0000	C148	C
890	0	370376	7.7500	NaN	Q

[891 rows x 12 columns]

```
[ ]: live = (titanic_df['Survived'] == 1) & (titanic_df['Sex'] == 'female') &
↳ (titanic_df['Age'] <= 10.0)
```

```
[ ]: live
```

```
[ ]: 0    False
      1    False
      2    False
      3    False
      4    False
      ..
      886  False
      887  False
      888  False
      889  False
      890  False
      Length: 891, dtype: bool
```

```
[ ]: titanic_df[live]
```

```
[ ]: PassengerId Survived Pclass Name \
10      11          1      3      Sandstrom, Miss. Marguerite Rut
43      44          1      2  Laroche, Miss. Simonne Marie Anne Andree
58      59          1      2      West, Miss. Constance Mirium
172     173          1      3      Johnson, Miss. Eleanor Ileen
184     185          1      3  Kink-Heilmann, Miss. Luise Gretchen
233     234          1      3      Asplund, Miss. Lillian Gertrud
237     238          1      2  Collyer, Miss. Marjorie "Lottie"
381     382          1      3      Nakid, Miss. Maria ("Mary")
448     449          1      3  Baclini, Miss. Marie Catherine
469     470          1      3  Baclini, Miss. Helene Barbara
479     480          1      3  Hirvonen, Miss. Hildur E
530     531          1      2      Quick, Miss. Phyllis May
535     536          1      2      Hart, Miss. Eva Miriam
618     619          1      2      Becker, Miss. Marion Louise
644     645          1      3      Baclini, Miss. Eugenie
691     692          1      3      Karun, Miss. Manca
720     721          1      2  Harper, Miss. Annie Jessie "Nina"
750     751          1      2      Wells, Miss. Joan
777     778          1      3  Emanuel, Miss. Virginia Ethel
```

	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
10	female	4.00	1	1	PP 9549	16.7000	G6	S
43	female	3.00	1	2	SC/Paris 2123	41.5792	NaN	C
58	female	5.00	1	2	C.A. 34651	27.7500	NaN	S
172	female	1.00	1	1	347742	11.1333	NaN	S
184	female	4.00	0	2	315153	22.0250	NaN	S
233	female	5.00	4	2	347077	31.3875	NaN	S
237	female	8.00	0	2	C.A. 31921	26.2500	NaN	S
381	female	1.00	0	2	2653	15.7417	NaN	C
448	female	5.00	2	1	2666	19.2583	NaN	C
469	female	0.75	2	1	2666	19.2583	NaN	C
479	female	2.00	0	1	3101298	12.2875	NaN	S
530	female	2.00	1	1	26360	26.0000	NaN	S
535	female	7.00	0	2	F.C.C. 13529	26.2500	NaN	S
618	female	4.00	2	1	230136	39.0000	F4	S
644	female	0.75	2	1	2666	19.2583	NaN	C
691	female	4.00	0	1	349256	13.4167	NaN	C
720	female	6.00	0	1	248727	33.0000	NaN	S
750	female	4.00	1	1	29103	23.0000	NaN	S
777	female	5.00	0	0	364516	12.4750	NaN	S

```
[ ]: len(titanic_df[live])
```

```
[ ]: 19
```

```
[ ]: pclass1 = (titanic_df['Pclass'] == 1)
      pclass3 = (titanic_df['Pclass'] == 3)

      pclass1_survivors = (titanic_df['Pclass'] == 1) & (titanic_df['Survived'] == 1)
      pclass3_survivors = (titanic_df['Pclass'] == 3) & (titanic_df['Survived'] == 1)
```

```
[ ]: print('Passenger class 1 survivors:', len(titanic_df[pclass1_survivors]))
      print('Passenger class 3 survivors:', len(titanic_df[pclass3_survivors]))

      print('Passenger class 1 total members:', len(titanic_df[pclass1]))
      print('Passenger class 3 total members:', len(titanic_df[pclass3]))
```

```
Passenger class 1 survivors: 136
Passenger class 3 survivors: 119
Passenger class 1 total members: 216
Passenger class 3 total members: 491
```

```
[ ]: print('Percentage of class 1 survivors: ', len(titanic_df[pclass1_survivors])/
      ↪len(titanic_df[pclass1]) * 100, '%')
      print('Percentage of class 3 survivors: ', len(titanic_df[pclass3_survivors])/
      ↪len(titanic_df[pclass3]) * 100, '%')
```

```
Percentage of class 1 survivors: 62.96296296296296 %
Percentage of class 3 survivors: 24.236252545824847 %
```