INDEXAR, SELECCIONAR I FILTRAR

Seguim avançant i aprofundint en les bases de dades.

A continuació, crea un nou projecte amb Jupyter Notebook i descarrega't aquesta base de dades des d'aquest **enllaç**. Guarda-la a la mateixa carpeta on tens el projecte nou.

Primer, carreguem la llibreria *Pandas* i li assignem el nom *pd:*

import pandas as pd

Importem la base de dades i la guardem en una variable anomenada *data_base*. Després, veiem les cinc primeres observacions:

data_base = pd.read_csv('insurance.csv')

data_base.head()

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.900	0	yes	southwest	16884.92400
1	18	male	33.770	1	no	southeast	1725.55230
2	28	male	33.000	3	no	southeast	4449.46200
3	33	male	22.705	0	no	northwest	21984.47061
4	32	male	28.880	0	no	northwest	3866.85520

Aleshores, ens apareixen totes les columnes amb les seves cinc primeres observacions, però no sabem com és de gran aquesta base de dades. Si volem imprimir només un determinat nombre de files, però volem, a més, veure el nombre total de files, ho haurem de fer així:

pd.set_option("display.max_rows", 5)

data_base

	age	sex	bmi	children	smoker	region	charges
0	19	female	27.90	0	yes	southwest	16884.9240
1	18	male	33.77	1	no	southeast	1725.5523
...	...	...	...	...	...	...	...
1336	21	female	25.80	0	no	southwest	2007.9450
1337	61	female	29.07	0	yes	northwest	29141.3603

1338 rows × 7 columns

Aquí, ens apareixen els títols de les columnes i quatre files amb les seves dades. A sota de la taula, veiem que està formada per 1.338 files (observacions) i 7 columnes (atributs o variables).

Si volem específicament la informació que conté una variable determinada, només haurem de picar aquest codi:

data_base['region']

Python ens mostrarà aquesta informació:

```
0        southwest
1        southeast
           ...
1336     southwest
1337     northwest
Name: region, Length: 1338, dtype: object
```

Ara, veiem només informació corresponent a la variable *region.*

Si vols veure més observacions, hauràs de canviar el paràmetre de la funció *set_option().* En comptes de cinc, n'hi pots posar deu. Si fas el canvi, hauràs de tornar a executar els codis, per tal de veure més files a la base de dades.

Nota: tu pots escollir una altra variable, com pot ser *smoker, sex, age,* etc.

Suposem, ara, que vols saber el valor més gran que té la columna *bmi*. Això ho pots saber, cridant la funció *describe()*. Te'n recordes? O bé seleccionant només la columna *bmi* i fent ús de la funció *max()*:

data_base['bmi'].max()

En tots dos casos, el resultat és 53,13.

I com ho podem fer per seleccionar només els 30 primers valors d'una columna? Doncs així:

bmi_30 = data_base['bmi'][0:30]

bmi_30

Ara, mirem a veure quin és el valor màxim dels 30 primers valors de la columna *bmi*:

bmi_30.max()

El valor obtingut és 42,13. El valor 53,13, doncs, no està dins dels 30 primers valors.

Anem a indexar, ara, no només valors d'una columna, sinó files i columnes conjuntament. Si volem, per exemple, treballar amb les deu primeres files i les dues primeres columnes, com ho fem? Necessitem treballar amb l'operador *iloc* i ho farem així:

data_base_2 = data_base.iloc[0:10,0:2]

El que hem fet, aquí, és dir que volem les deu primeres files (0:10) i les dues primeres columnes(0:2) de *data_base*. Hem guardat aquesta taula reduïda en un nou espai de memòria, a través d'un nom nou que és *data_base_2*. Això és un objecte nou, una nova base de dades. Si cridem aquesta nova base de dades, obtenim:

data_base_2

	age	sex
0	19	female
1	18	male
2	28	male
3	33	male
4	32	male
5	31	female
6	46	female
7	37	female
8	37	male
9	60	female

Podem, també, en comptes de seleccionar un interval de files o columnes, podem seleccionar específicament unes determinades files o columnes*:*

data_base_3 = data_base.iloc[[0,5,6],[0,4]]

data_base_3

El que hem fet aquí és obtenir les files 0,5 i 6 i les columnes 0 i 4. Després, les hem guardat com una altra base de dades reduïda en un altre espai de memòria.

Per a les columnes, podem fer servir els seus noms propis per seleccionar-les. En aquest cas, hem de fer servir l'operador *loc*. Mirem aquest exemple:

data_base_4 = data_base.loc[0:8,['children','age']]

data_base_4

El que hem fet és seleccionar les vuit primeres files de les columnes *children* i *age*. Les dades que es mostren són les següents:

	children	age
0	0	19
1	1	18
2	3	28
3	0	33
4	0	32
5	0	31
6	1	46
7	3	37
8	2	37

Fem un pas endavant i anem a crear taules reduïdes a partir de la taula mare, no fent ús dels índexs de files o columnes, sinó de criteris condicionals. Imagina que només volem aquelles observacions que es troben a la regió *southwest*. Aquest és el codi que necessitem:

```
data_base_southwest = data_base.loc[data_base.region == 'southwest']
```

I volem treure, per pantalla, només les set primeres files i les dues primeres columnes:

```
data_base_southwest.iloc[0:7,0:2]
```

La taula que es veu per pantalla és aquesta:

	age	sex
0	19	female
12	23	male
15	19	male
18	56	male
19	30	male
21	30	female
29	31	male

Fixa't que la numeració de files fa referència a les files 0, 12, 15, 18, 19, 21 i 29, les quals són les set primeres files que pertanyen a la regió *southwest* de la taula mare *data_base.*

Si volem veure aquelles observacions de la base de dades principal que tinguin un *bmi* entre 35 i 45, haurem de declarar dos condicionals:

```
data_base.loc[(data_base.bmi >= 35) & (data_base.bmi <= 45)].describe()
```

Hem aplicat la funció *describe()*, per veure quins valors màxims i mínims té la columna *bmi* d'aquesta taula reduïda:

	age	bmi	children	charges
count	296.000000	296.000000	296.000000	296.000000
mean	41.679054	38.301976	1.027027	16913.681515
std	14.550498	2.427277	1.149479	15367.757351
min	18.000000	35.090000	0.000000	1141.445100
25%	30.000000	36.297500	0.000000	5745.351188
50%	43.500000	37.707500	1.000000	10979.853800
75%	54.000000	39.840000	2.000000	26781.395215
max	64.000000	44.880000	5.000000	58571.074480

Efectivament, veiem que, en aquesta taula reduïda, només tenim aquelles files que tenen un valor de *bmi* entre 35 i 45.

Per últim, anem a crear una columna nova. Els seus valors dependran del valor de *bmi* de cada fila. El codi és el següent:

```
approved = []
for bmi in data_base['bmi']:
    if bmi < 45:
        approved.append(True)
    else:
        approved.append(False)
data_base['approved'] = approved
data_base.head()
```

El que estem fent és donar un valor a cada fila de la columna *approved* de *True* o *False*, segons si el valor de *bmi* de la mateixa fila és superior o inferior a 45. Vegem els cinc primers registres amb la nova columna:

	age	sex	bmi	children	smoker	region	charges	approved
0	19	female	27.900	0	yes	southwest	16884.92400	True
1	18	male	33.770	1	no	southeast	1725.55230	True
2	28	male	33.000	3	no	southeast	4449.46200	True
3	33	male	22.705	0	no	northwest	21984.47061	True
4	32	male	28.880	0	no	northwest	3866.85520	True

Descobreix tot el que Barcelona Activa pot fer per a tu

Acompanyament durant tot el procés de recerca de feina

barcelonactiva.cat/treball

Suport per posar en marxa la teva idea de negoci

barcelonactiva.cat/emprenedoria

Serveis a les empreses i iniciatives socioempresarials

barcelonactiva.cat/empreses

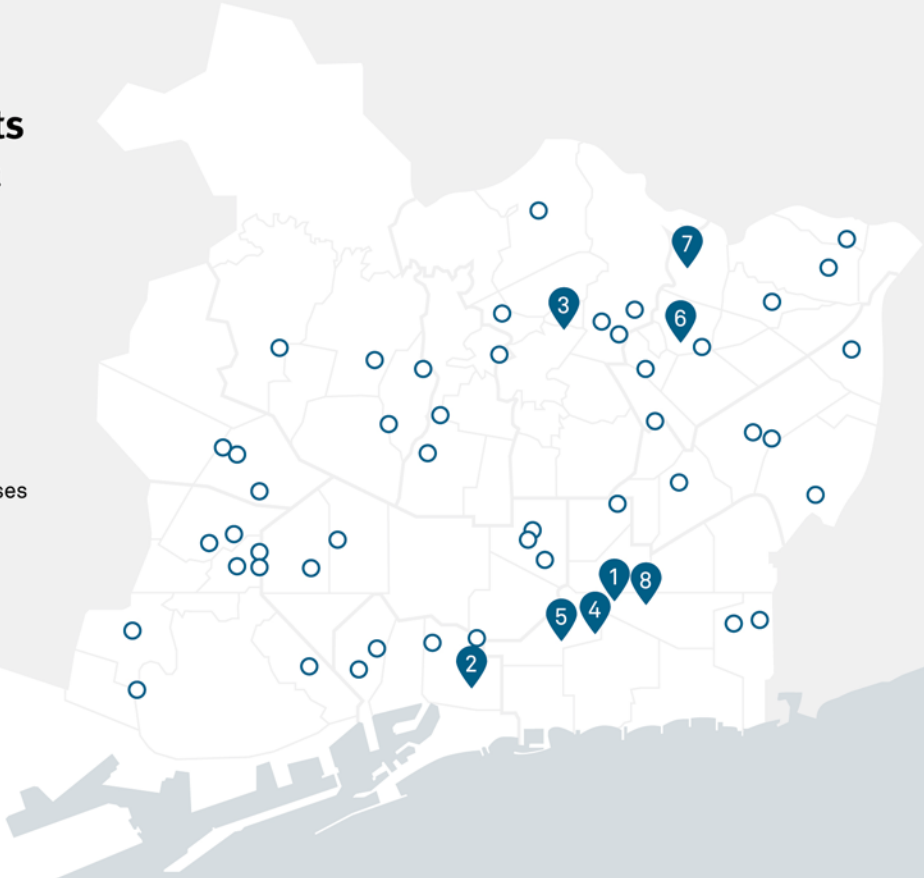
Formació tecnològica i gratuïta per a la ciutadania

barcelonactiva.cat/cibernarium

Xarxa d'equipaments de Barcelona Activa

1. Seu Central Barcelona Activa
   Porta 22
   Centre per a la Iniciativa Emprenedora Glòries
   Incubadora Glòries
2. Convent de Sant Agustí
3. Ca n'Andalet
4. Oficina d'Atenció a les Empreses
   Cibernàrium
   Incubadora MediaTIC
5. Incubadora Almogàvers
6. Parc Tecnològic
7. Nou Barris Activa
8. innoBA

○ Punts d'atenció a la ciutat



Barcelona Activa

Ajuntament de Barcelona

Darrera actualització 2019

Cofinançat per:

UNIÓ EUROPEA
Fons Europeu de Desenvolupament Regional

Segueix-nos a les xarxes socials:

barcelonactiva.cat/cibernarium
barcelonactiva
barcelonactiva
company/barcelona-activa