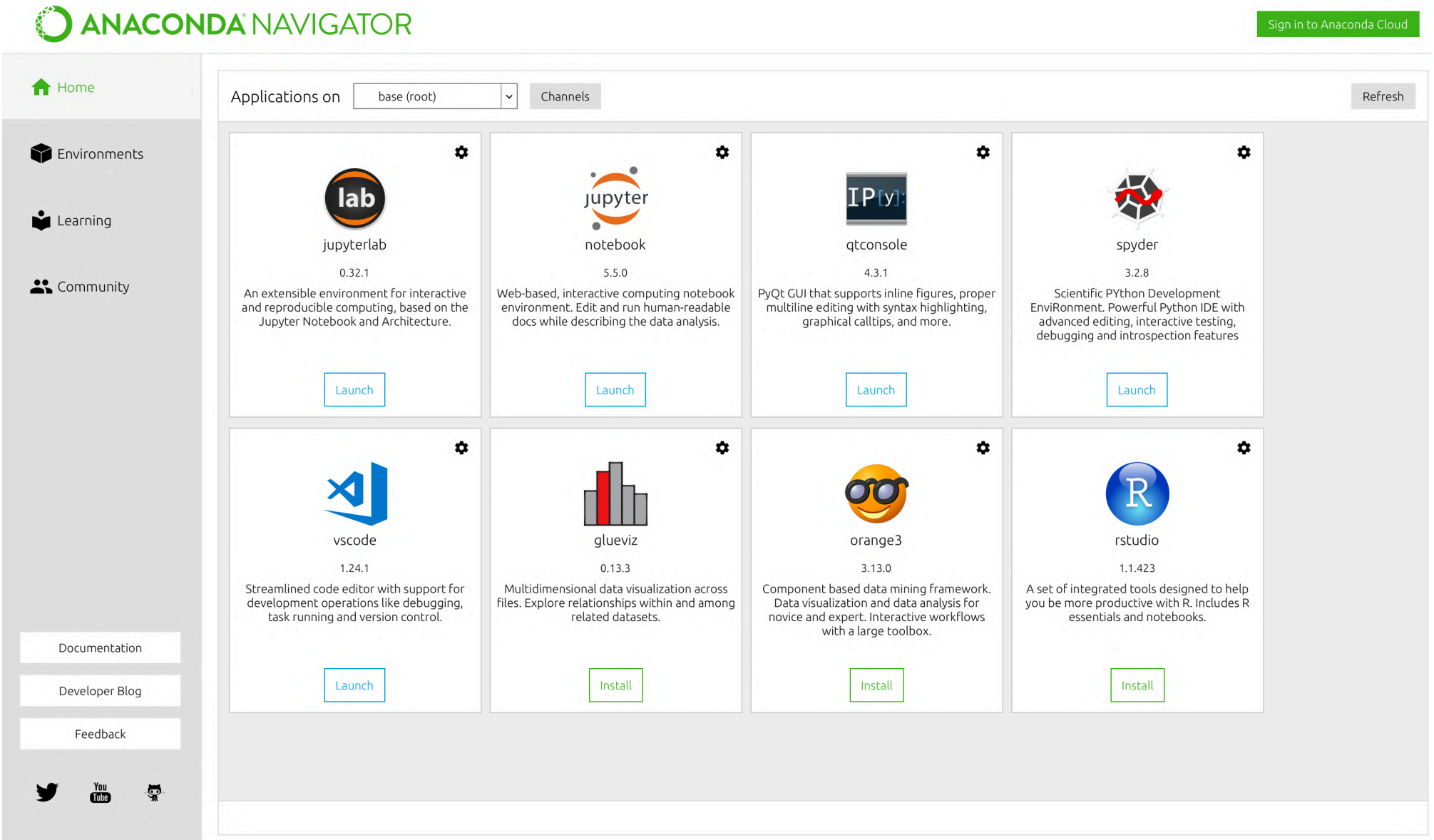
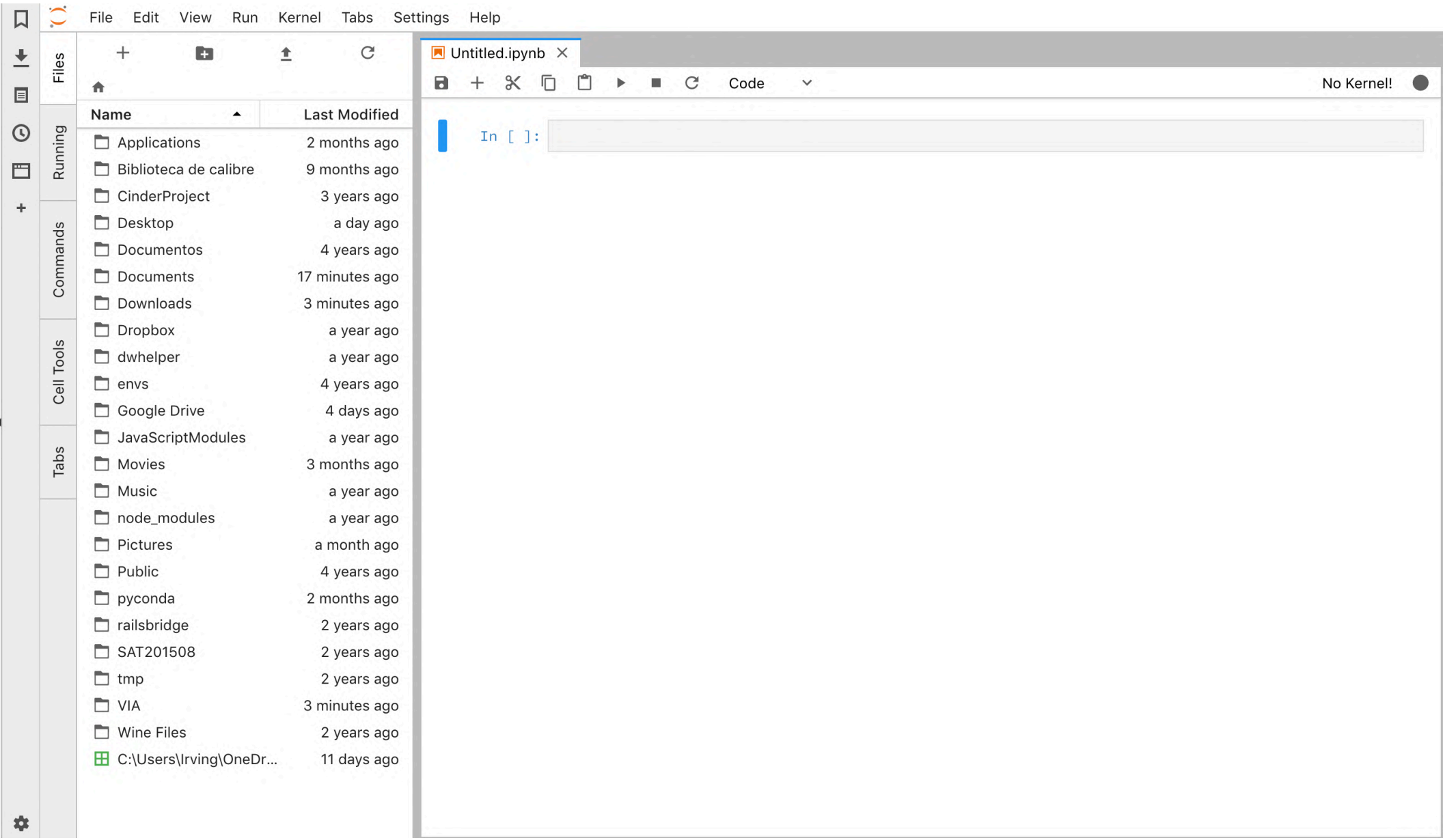
Slack del grupo

miniuri.com/2h8

https://www.anaconda.com/download/

Pandas


ANACONDA NAVIGATOR
Sign in to Anaconda Cloud
Home
Environments
Learning
Community
Documentation
Developer Blog
Feedback
Applications on
base (root)
Channels
Refresh
jupyterlab
0.32.1
An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
Launch
notebook
5.5.0
Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.
Launch
qtconsole
4.3.1
PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.
Launch
spyder
3.2.8
Scientific PYthon Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features
Launch
vscode
1.24.1
Streamlined code editor with support for development operations like debugging, task running and version control.
Launch
glueviz
0.13.3
Multidimensional data visualization across files. Explore relationships within and among related datasets.
Install
orange3
3.13.0
Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows with a large toolbox.
Install
rstudio
1.1.423
A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.
Install


File Edit View Run Kernel Tabs Settings Help
Files
Running
Commands
Cell Tools
Tabs
Name
Last Modified
Applications 2 months ago
Biblioteca de calibre 9 months ago
CinderProject 3 years ago
Desktop a day ago
Documentos 4 years ago
Documents 17 minutes ago
Downloads 3 minutes ago
Dropbox a year ago
dwhelper a year ago
envs 4 years ago
Google Drive 4 days ago
JavaScriptModules a year ago
Movies 3 months ago
Music a year ago
node_modules a year ago
Pictures a month ago
Public 4 years ago
pyconda 2 months ago
railsbridge 2 years ago
SAT201508 2 years ago
tmp 2 years ago
VIA 3 minutes ago
Wine Files 2 years ago
C:\Users\Irving\OneDr... 11 days ago
Untitled.ipynb
Code
No Kernel!
In [ ]:

Listas

```
lista=['hola',10,True,5.763]
```

```
print(lista)
```

```
['hola', 10, True, 5.763]
```

```
lista
```

```
['hola', 10, True, 5.763]
```

Seleccionar valores

seleccionar un valor

```
e0 = lista[1]
```

```
e0
```

```
10
```

seleccionar varios valores

```
e1 = lista[:3]
```

```
e1
```

```
['hola', 10, True]
```

```
print(e1)
```

```
['hola', 10, True]
```

```
e2 = lista[2:]
```

```
e2
```

```
[True, 5.763]
```

https://docs.python.org/3/tutorial/datastructures.html#more-on-lists

5. Data Structures

This chapter describes some things you've learned about already in more detail, and adds some new things as well.

5.1. More on Lists

The list data type has some more methods. Here are all of the methods of list objects:

`list.`**`append`**(*x*)
Add an item to the end of the list. Equivalent to `a[len(a):] = [x]`.

`list.`**`extend`**(*iterable*)
Extend the list by appending all the items from the iterable. Equivalent to `a[len(a):] = iterable`.

`list.`**`insert`**(*i*, *x*)
Insert an item at a given position. The first argument is the index of the element before which to insert, so `a.insert(0, x)` inserts at the front of the list, and `a.insert(len(a), x)` is equivalent to `a.append(x)`.

`list.`**`remove`**(*x*)
Remove the first item from the list whose value is equal to *x*. It raises a `ValueError` if there is no such item.

`list.`**`pop`**([*i*])
Remove the item at the given position in the list, and return it. If no index is specified, `a.pop()` removes and returns the last item in the list. (The square brackets around the *i* in the method signature denote that the parameter is optional, not that you should type square brackets at that position. You will see this notation frequently in the Python Library Reference.)

Tuplas

```
tupla = ("adios",3,False,2.654)
```

```
tupla
```

```
('adios', 3, False, 2.654)
```

```
t = tupla[0]
```

```
t
```

```
'adios'
```

```
tupla[0] = "otra cosa"
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-46-fd1332b33cc7> in <module>()
----> 1 tupla[0] = "otra cosa"

TypeError: 'tuple' object does not support item assignment
```

Diccionarios

```
dic = {}
```

```
dic
```

```
{}
```

```
print(type(dic))
```

```
<class 'dict'>
```

```
dic['nombre']="Irving"
dic['apellido']="Morales"
```

```
dic
```

```
{'nombre': 'Irving', 'apellido': 'Morales'}
```

```
dic["nombre"]
```

```
'Irving'
```

```
dic["edad"]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-54-007d4819c111> in <module>()
----> 1 dic["edad"]

KeyError: 'edad'
```

```
dic["edad"]
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-54-007d4819c111> in <module>()
----> 1 dic["edad"]

KeyError: 'edad'
```

```
dic.get("edad")
```

```
dic.keys()
```

```
dict_keys(['nombre', 'apellido'])
```

```
dic.values()
```

```
dict_values(['Irving', 'Morales'])
```

```
dic.items()
```

```
dict_items([('nombre', 'Irving'), ('apellido', 'Morales')])
```

Slicing

```
lista
```

```
['nuevo valor', 10, True, 5.763, 'un valor más']
```

```
lista[1:3]
```

```
[10, True]
```

```
lista[1:3]
```

```
[10, True]
```

```
lista[1:-1]
```

```
[10, True, 5.763]
```

```
lista[:3]
```

```
['nuevo valor', 10, True]
```

```
lista[3:]
```

```
[5.763, 'un valor más']
```

```
lista[::2]
```

```
['nuevo valor', True, 'un valor más']
```

Importar librerías

```
import pandas as pd
```

Cargar Datos

```
df = pd.read_csv("gapminder.tsv", sep='\t')
```

```
df
```

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
3	Afghanistan	Asia	1967	34.020	11537966	836.197138
4	Afghanistan	Asia	1972	36.088	13079460	739.981106
5	Afghanistan	Asia	1977	38.438	14880372	786.113360
6	Afghanistan	Asia	1982	39.854	12881816	978.011439

```
print(type(df))
```

```
<class 'pandas.core.frame.DataFrame'>
```

```
print(df.head())
```

```
       country continent  year  lifeExp       pop   gdpPercap
0  Afghanistan      Asia  1952   28.801   8425333  779.445314
1  Afghanistan      Asia  1957   30.332   9240934  820.853030
2  Afghanistan      Asia  1962   31.997  10267083  853.100710
3  Afghanistan      Asia  1967   34.020  11537966  836.197138
4  Afghanistan      Asia  1972   36.088  13079460  739.981106
```

```
print(df.head(10))
```

```
       country continent  year  lifeExp       pop   gdpPercap
0  Afghanistan      Asia  1952   28.801   8425333  779.445314
1  Afghanistan      Asia  1957   30.332   9240934  820.853030
2  Afghanistan      Asia  1962   31.997  10267083  853.100710
3  Afghanistan      Asia  1967   34.020  11537966  836.197138
4  Afghanistan      Asia  1972   36.088  13079460  739.981106
5  Afghanistan      Asia  1977   38.438  14880372  786.113360
6  Afghanistan      Asia  1982   39.854  12881816  978.011439
7  Afghanistan      Asia  1987   40.822  13867957  852.395945
8  Afghanistan      Asia  1992   41.674  16317921  649.341395
9  Afghanistan      Asia  1997   41.763  22227415  635.341351
```

```
print(df.tail(3))
```

```
       country continent  year  lifeExp       pop   gdpPercap
1701  Zimbabwe    Africa  1997   46.809  11404948  792.449960
1702  Zimbabwe    Africa  2002   39.989  11926563  672.038623
1703  Zimbabwe    Africa  2007   43.487  12311143  469.709298
```

```
print(df.shape)
```

```
(1704, 6)
```

```
print(df.columns)
```

```
Index(['country', 'continent', 'year', 'lifeExp', 'pop', 'gdpPercap'], dtype='object')
```

```
print(df.dtypes)
```

```
country       object
continent     object
year           int64
lifeExp      float64
pop            int64
gdpPercap    float64
dtype: object
```

```
print(df.info())
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1704 entries, 0 to 1703
Data columns (total 6 columns):
country      1704 non-null object
continent    1704 non-null object
year         1704 non-null int64
lifeExp      1704 non-null float64
pop          1704 non-null int64
gdpPercap    1704 non-null float64
dtypes: float64(2), int64(2), object(2)
memory usage: 80.0+ KB
None
```

Accesando a datos

Accesando a columnas

```
country=df['country']
```

```
print(country.head())
```

```
0    Afghanistan
1    Afghanistan
2    Afghanistan
3    Afghanistan
4    Afghanistan
Name: country, dtype: object
```

```
print(country.tail())
```

```
1699    Zimbabwe
1700    Zimbabwe
1701    Zimbabwe
1702    Zimbabwe
1703    Zimbabwe
Name: country, dtype: object
```

```
subset = df[['country','continent','year']]
```

```
subset.head(3)
```

	country	continent	year
0	Afghanistan	Asia	1952
1	Afghanistan	Asia	1957
2	Afghanistan	Asia	1962

Accesando a filas

```
df.head()
```

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
3	Afghanistan	Asia	1967	34.020	11537966	836.197138
4	Afghanistan	Asia	1972	36.088	13079460	739.981106

```
df.loc[3]
```

```
country      Afghanistan
continent           Asia
year                1967
lifeExp            34.02
pop             11537966
gdpPercap        836.197
Name: 3, dtype: object
```

```
df.loc[[0,99,999]]
```

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
99	Bangladesh	Asia	1967	43.453	62821884	721.186086
999	Mongolia	Asia	1967	51.253	1149500	1226.041130

```
df.head()
```

	country	continent	year	lifeExp	pop	gdpPercap
0	Afghanistan	Asia	1952	28.801	8425333	779.445314
1	Afghanistan	Asia	1957	30.332	9240934	820.853030
2	Afghanistan	Asia	1962	31.997	10267083	853.100710
3	Afghanistan	Asia	1967	34.020	11537966	836.197138
4	Afghanistan	Asia	1972	36.088	13079460	739.981106

```
df.iloc[1]
```

```
country      Afghanistan
continent           Asia
year                1957
lifeExp           30.332
pop              9240934
gdpPercap        820.853
Name: 1, dtype: object
```

```
df.iloc[-1]
```

```
country      Zimbabwe
continent      Africa
year             2007
lifeExp         43.487
pop           12311143
gdpPercap      469.709
Name: 1703, dtype: object
```

```
df.loc[[3,5,7],['country','year']]
```

	country	year
3	Afghanistan	1967
5	Afghanistan	1977
7	Afghanistan	1987

```
df.iloc[3:9,[2,4,-1]]
```

	year	pop	gdpPercap
3	1967	11537966	836.197138
4	1972	13079460	739.981106
5	1977	14880372	786.113360
6	1982	12881816	978.011439
7	1987	13867957	852.395945
8	1992	16317921	649.341395

```
df.iloc[3:9,1:4]
```

	continent	year	lifeExp
3	Asia	1967	34.020
4	Asia	1972	36.088
5	Asia	1977	38.438
6	Asia	1982	39.854
7	Asia	1987	40.822
8	Asia	1992	41.674

Agrupar

```
df.groupby('year')
```

```
<pandas.core.groupby.groupby.DataFrameGroupBy object at 0x10b9a1828>
```

```
df.groupby('year')['lifeExp'].mean()
```

```
year
1952    49.057620
1957    51.507401
1962    53.609249
1967    55.678290
1972    57.647386
1977    59.570157
1982    61.533197
1987    63.212613
1992    64.160338
1997    65.014676
2002    65.694923
2007    67.007423
Name: lifeExp, dtype: float64
```

```
df.groupby('continent')['lifeExp'].mean()
```

```
continent
Africa      48.865330
Americas    64.658737
Asia        60.064903
Europe      71.903686
Oceania     74.326208
Name: lifeExp, dtype: float64
```

```
df.groupby(['year','continent'])[['lifeExp','gdpPercap']].mean()
```

		lifeExp	gdpPercap
year	**continent**		
1952	**Africa**	39.135500	1252.572466
	Americas	53.279840	4079.062552
	Asia	46.314394	5195.484004
	Europe	64.408500	5661.057435
	Oceania	69.255000	10298.085650
1957	**Africa**	41.266346	1385.236062
	Americas	55.960280	4616.043733
	Asia	49.318544	5787.732940
	Europe	66.703067	6963.012816
	Oceania	70.295000	11598.522455
1962	**Africa**	43.319442	1598.078825
	Americas	58.398760	4901.541870
	Asia	51.563223	5729.369625
	Europe	68.539233	8365.486814
	Oceania	71.085000	12696.452430

```
means.head(10)
```

		lifeExp	gdpPercap
year	**continent**		
1952	**Africa**	39.135500	1252.572466
	Americas	53.279840	4079.062552
	Asia	46.314394	5195.484004
	Europe	64.408500	5661.057435
	Oceania	69.255000	10298.085650
1957	**Africa**	41.266346	1385.236062
	Americas	55.960280	4616.043733
	Asia	49.318544	5787.732940
	Europe	66.703067	6963.012816
	Oceania	70.295000	11598.522455

```
means.reset_index().head(10)
```

	year	continent	lifeExp	gdpPercap
0	1952	Africa	39.135500	1252.572466
1	1952	Americas	53.279840	4079.062552
2	1952	Asia	46.314394	5195.484004
3	1952	Europe	64.408500	5661.057435
4	1952	Oceania	69.255000	10298.085650
5	1957	Africa	41.266346	1385.236062

```
global_yearly_life_expectancy = df.groupby('year')['lifeExp'].mean()
```

```
global_yearly_life_expectancy
```

```
year
1952    49.057620
1957    51.507401
1962    53.609249
1967    55.678290
1972    57.647386
1977    59.570157
1982    61.533197
1987    63.212613
1992    64.160338
1997    65.014676
2002    65.694923
2007    67.007423
Name: lifeExp, dtype: float64
```

```
pl=global_yearly_life_expectancy.plot()
```