



Python para universitarios



Python es un entorno de trabajo útil para cálculos científicos, trabajo con base de datos, operaciones financieras y gestión de páginas web

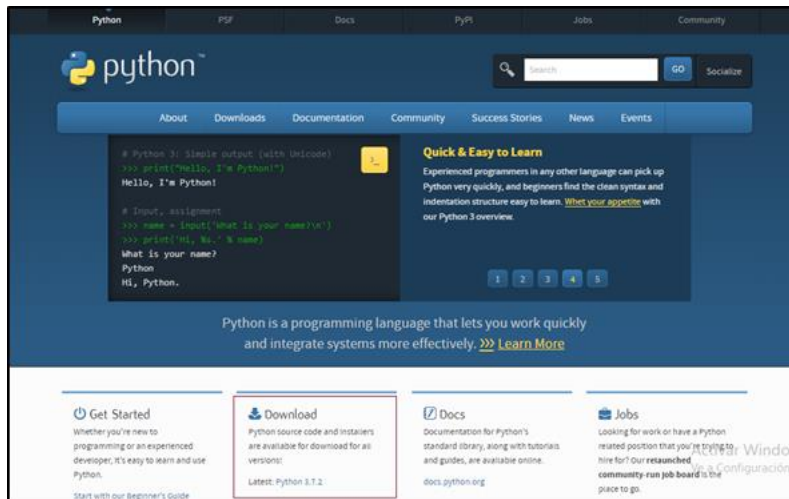
Ventajas de la programación en Python:

- Es Open Source
- Enfatiza la productividad y facilidad en el código
- Esta orientado a objetos.
- Es compatible con equipos Mac, Windows y Linux.

```
4     self.__pineapple_allowed = False
5
6     @property
7     def pineapple_allowed(self):
8         return self.__pineapple_allowed
9
10    @pineapple_allowed.setter
11    def pineapple_allowed(self, value):
12        if value:
13            password = input("Introduzca la contraseña: ")
14            if password == "sw0rdf1sh":
15                self.__pineapple_allowed = value
16            else:
17                raise ValueError("Alerta! Intruso!")
18
```

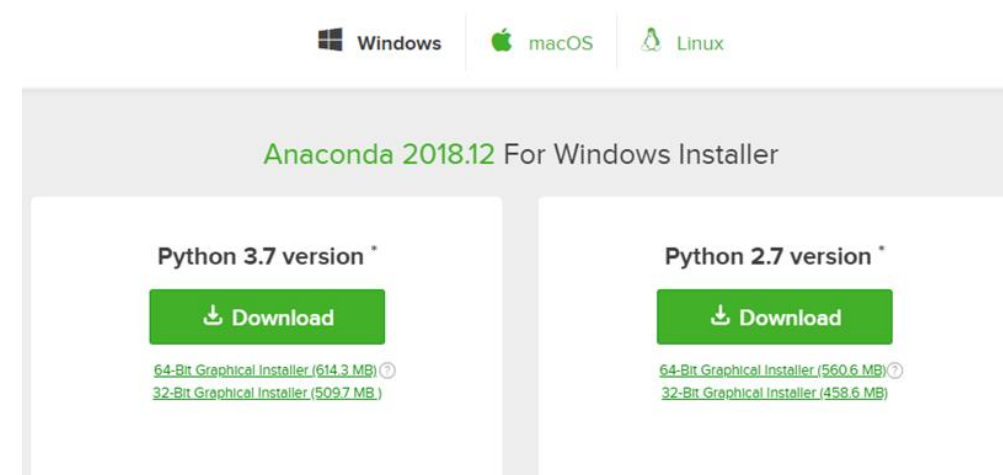
Instalación Python

<https://www.python.org/>



Instalación Anaconda

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



Why?

Who uses Python ?:

- Google
- NASA
- Yahoo
- Youtube
- Linux (RedHat, Ubuntu, ...)
- Lots of researchers
- EVE online (Thousands of online players)
- MIT (Programming Intro. Course)
- etc...





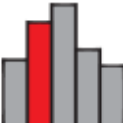



You Tube



Watch this Google I/O video clip, Wesley Chun listing the uses of python and what companies use Python. (from 22:35 to 29:10)



Manejo de entornos de trabajo

 <p>JupyterLab 0.35.3</p> <p>An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.</p> <p>Launch</p>	 <p>Notebook 5.7.4</p> <p>Web-based, interactive computing notebook environment. Edit and run human-readable docs while describing the data analysis.</p> <p>Launch</p>	 <p>Qt Console 4.4.3</p> <p>PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips, and more.</p> <p>Launch</p>	 <p>Spyder 3.3.2</p> <p>Scientific PYTHON Development EnviRonment. Powerful Python IDE with advanced editing, interactive testing, debugging and introspection features</p> <p>Launch</p>
 <p>Glueviz 0.13.3</p> <p>Multidimensional data visualization across files. Explore relationships within and among related datasets.</p>	 <p>Orange 3 3.17.0</p> <p>Component based data mining framework. Data visualization and data analysis for novice and expert. Interactive workflows</p>	 <p>RStudio 1.1.456</p> <p>A set of integrated tools designed to help you be more productive with R. Includes R essentials and notebooks.</p>	 <p>VS Code 1.30.2</p> <p>Streamlined code editor with support for development operations like debugging, task running and version control.</p>

Verificamos la instalación que tenemos

```
import sys
```

```
print('Python version:', sys.version)
```

```
import IPython
```

```
print('IPython:', IPython.__version__)
```

```
import numpy
```

```
print('numpy:', numpy.__version__)
```

```
import pandas
```

```
print('pandas:', pandas.__version__)
```



Programación en Python:

Tipos de estructuras

```
x=3 #numeros  
y=4  
  
a = "gorilas" #cadenas  
  
r= True #booleanos
```

Operaciones básicas

```
x+y  
x-y  
x*y  
x/y  
x//y #parte entera  
x%y #resto
```

Ciclos

```
suma = 0  
for i in 2,4,6,8:  
    suma = suma+i  
    print(suma)  
  
for i in range(2,8):  
    print(i)
```

Funciones matemáticas

```
from math import sqrt  
from math import sin, cos, tan, log, exp, pi, e  
  
sin(pi/1)
```

Programación en Python:

Algebra de matrices

```
import numpy as np #cargmos el modulo numpy con el alias np
```

```
array1 = np.arange(10) #creamos un array de 0 a 9  
type(array1)
```

```
array2 = np.arange(20)
```

```
array2.ndim #Numero de dimensiones  
array2.shape #Consulta la dimension  
array2.size #Consulta la dimension  
array2.dtype #Consulltamos el tipo de elementos
```

```
from numpy import matrix #Importamos matrix del modulo numpy  
a = matrix([[2,3,4],[5,5,5],[6,7,8]])  
print(a)
```

```
b = matrix([[2],[4],[6]])  
print(b)
```

```
a*b #multiplicacion de matrices
```

```
#Practicando el algebra lineal de matrices
```

```
b.T #Transpuesta
```

```
b.H #Hermitica (transpuesta y conjugada)
```


Programación en Python:

Creación de Funciones

```
def fun_sum(n):  
    s = n*(n+1)/2  
    print(s)
```

```
fun_sum(20)
```

```
def mensaje():  
    print("Aprendiendo Python")  
    print("Aprendiendo Funciones")  
    print("Avanzando poco a poco")
```

```
mensaje()
```

```
def producto(num1,num2):  
    prod = num1*10+num2  
    if prod == 20:  
        print("galleta")  
    elif prod == 23:  
        print("gaseosa")  
    else:  
        print("producto no encontrado")
```

```
producto(2,3)
```



Programación en Python:

Manejo de archivos de datos

Para cargar la carpeta de trabajo

```
import pandas as pd #Importamos el paquete pandas con alias pd  
#Leemos nuestra data  
datos = pd.read_csv('C:/Users/Lenovo User/Desktop/Base Credito/Base Credit Card.csv')  
  
datos["EDAD"].describe()
```

Para guardar el archivo de trabajo

```
datos2.to_csv('C:/Users/Lenovo User/Desktop/Base Credito/Base Credit Card2.csv')
```

¿Qué es un valor perdido?

Valores que para una variable determinada no constan en algunas filas o patrones

Consecuencias:

- La pérdida de información que producen
- Pueden introducir un sesgo considerable
- Hacen el manejo del análisis más complicado



Tratamiento:

- Eliminar los registros que tengan datos faltantes
- Imputar los valores perdidos con estimaciones

Programación en Python:

Manejo de datos perdidos

```
import pandas as pd #Importamos el paquete pandas con alias pd
import numpy as np #Importamos el paquete numpy con alias np

datos.dropna()#identificar los valores NaN
datos.dropna(subset=["EDAD"],axis = 0, inplace = True)#borramos las filas perdidas
```

Imputando con el promedio

```
media = datos["EDAD"].mean() #Calculamos la media
datos["EDAD2"].replace(np.nan,media) #Reemplazamos los NAs por la media
```

```
In [23]: datos["EDAD"].head(10)
Out[23]:
0    65.0
1    67.0
2    93.0
3     NaN
4     NaN
5     NaN
6     NaN
7    57.0
8    65.0
9    51.0
Name: EDAD, dtype: float64
```

Programación en Python:

Manejo de formas estadísticas

```
datos["LINEA_CREDITO"].mean() #Obtenemos el promedio de la variable Linea de crédito  
datos["LINEA_CREDITO"].min() #Obtenemos el mínimo de la variable Linea de crédito  
datos["LINEA_CREDITO"].max() #Obtenemos el máximo de la variable Linea de crédito  
datos["LINEA_CREDITO"].std() #Obtenemos la desviación estandar de la variable Linea de crédito
```

Ejemplo: estandarización de variables

$$z = \frac{x - \bar{x}}{s}$$

```
datos["LINEA_CREDITO_N"] = (datos["LINEA_CREDITO"]-datos["LINEA_CREDITO"].mean())/datos["LINEA_CREDITO"].std()
```

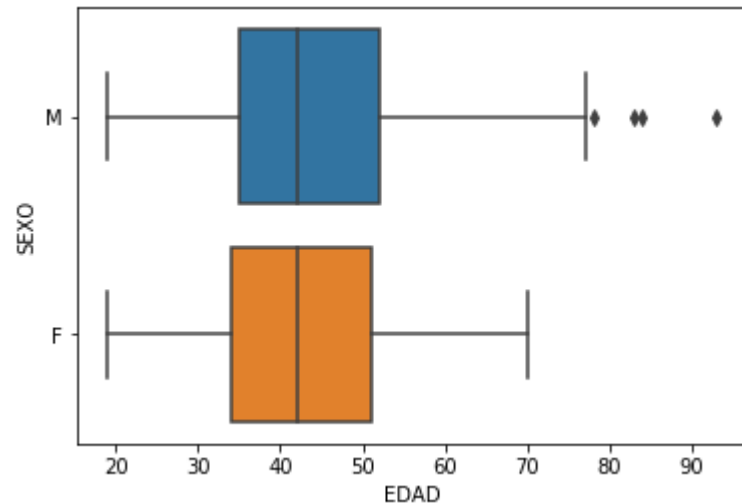
Programación en Python:

Manejo de formas gráficas

```
import seaborn as sns #Cargamos el paquete seaborn con alias sns
import matplotlib.pyplot as plt #Cargamos el paquete matplotlib con alias plt
```

Gráficas más utilizadas

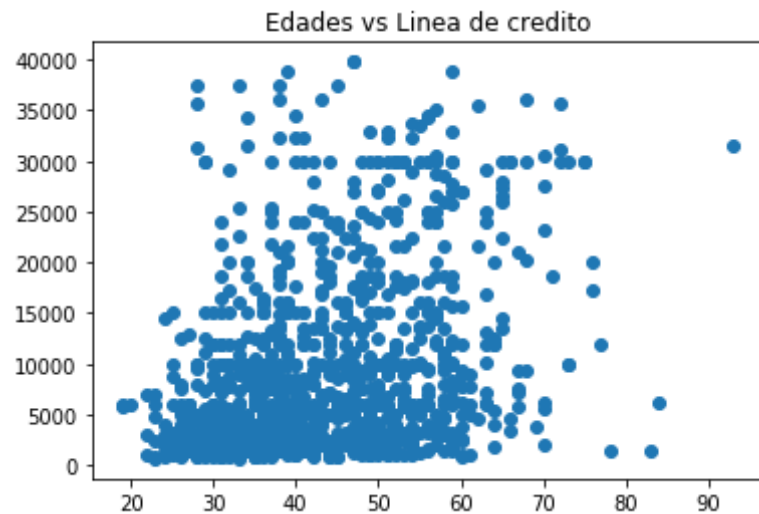
```
sns.boxplot(x="EDAD",y = "SEXO",data = datos) #diagrama de cajas
```



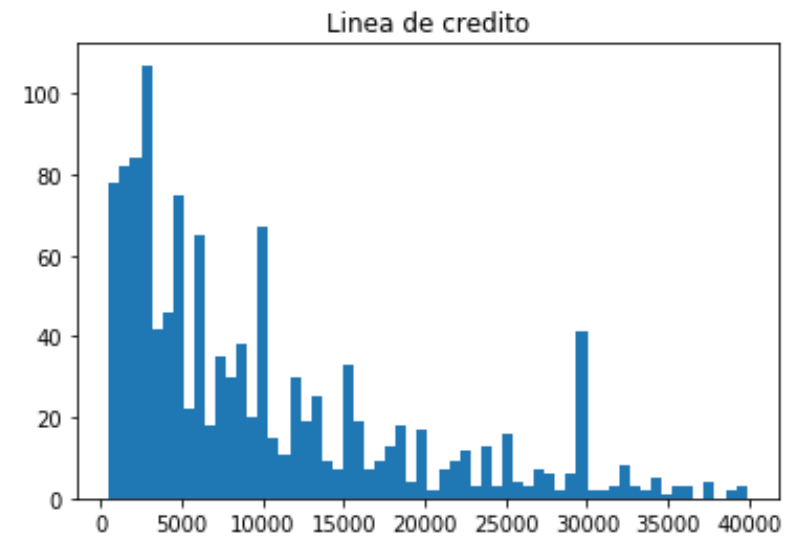
Programación en Python:

Manejo de formas gráficas

```
#diagrama de dispersión
plt.title("Edades vs Linea de credito")
plt.scatter(datos["EDAD"],datos["LINEA_CREDITO"])
```



```
#Histograma de Las Edades
plt.title("Linea de credito")
plt.hist(datos["LINEA_CREDITO"], bins = 60)
show()
```



Programación en Python:

Análisis de kmeans:

Importamos los módulos necesarios

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
from sklearn.cluster import KMeans
from sklearn.metrics import pairwise_distances_argmin_min
```

```
#Ontenemos el cluster
kmeans = KMeans(n_clusters=3).fit(x_train)
centroids = kmeans.cluster_centers_
print(centroids)
centroids.shape
```

Cargamos los datos de entrenamiento

```
namenum = ["LINEA_CREDITO", "RNG_LINEA", "DEUDA_TARJETA", "Total_facturacion"]

x_train = np.array(datos[namenum])
#x.dtypes
y_train = np.array(datos["Grupo"])
```

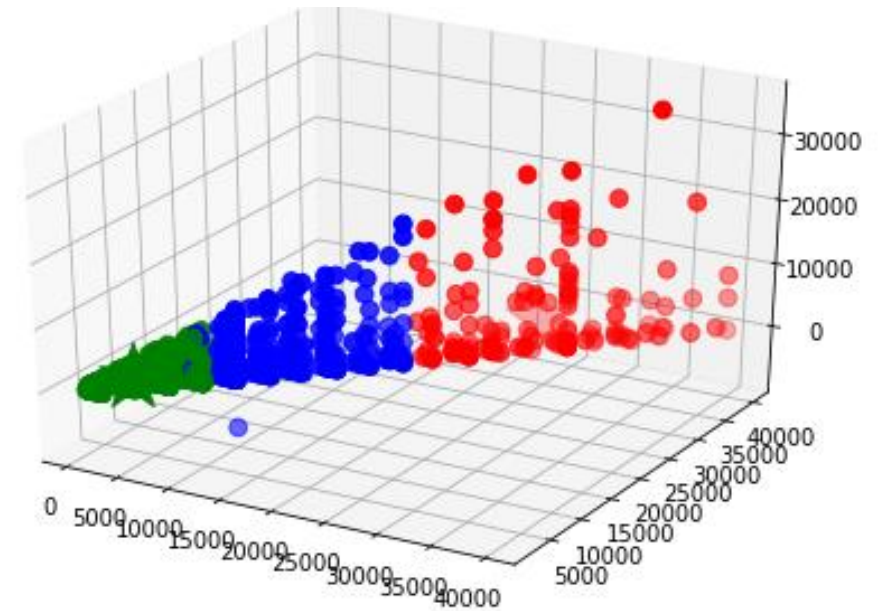

Programación en Python:

Análisis de kmeans:

```
#Visualizamos los cluster
# Predecimos los cluster
labels = kmeans.predict(x_train)

# Getting the cluster centers
C = kmeans.cluster_centers_
colores=['red','green','blue']
asignar=[]
for row in labels:
    asignar.append(colores[row])

fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(x_train[:, 0], x_train[:, 1], x_train[:, 2], c=asignar, s=60)
ax.scatter(C[:, 0], C[:, 1], C[:, 2], marker='*', c=colores, s=1000)
```



Programación en Python:

Arboles de decisión:

Importamos los módulos necesarios

```
import pandas as pd
import numpy as np
import seaborn as sb
import matplotlib as plt

from sklearn import tree
from sklearn.metrics import accuracy_score
from sklearn.tree import export_graphviz
from pydotplus import graph_from_dot_data
```

Generamos el árbol:

```
namenum = ["LINEA_CREDITO", "RNG_LINEA", "DEUDA_TARJETA", "Total_facturacion"]

x_train = datos[namenum]
#x.dtypes
y_train = datos["Grupo"]

tree_model = tree.DecisionTreeClassifier(criterion = 'entropy',
                                         max_depth = 3)

model01 = tree_model.fit(x_train,y_train)
print(model01)
```

Programación en Python:

Regresión Lineal:

```
from sklearn.linear_model import LinearRegression #Importamos el modelo lineal de sklearn  
lm = LinearRegression() #Creamos el objeto lm en base a la regresión
```

Regresión lineal simple

```
x = datos[["LINEA_CREDITO"]] #definimos los predictores  
y = datos["EDAD"] #definimos la variable Y  
  
lm.fit(x,y)#modelo de regresion lineal simple  
  
lm.coef_ #pedimos el b0 del regresor  
lm.intercept_ #pedimos el intercepto
```

Regresión lineal múltiple

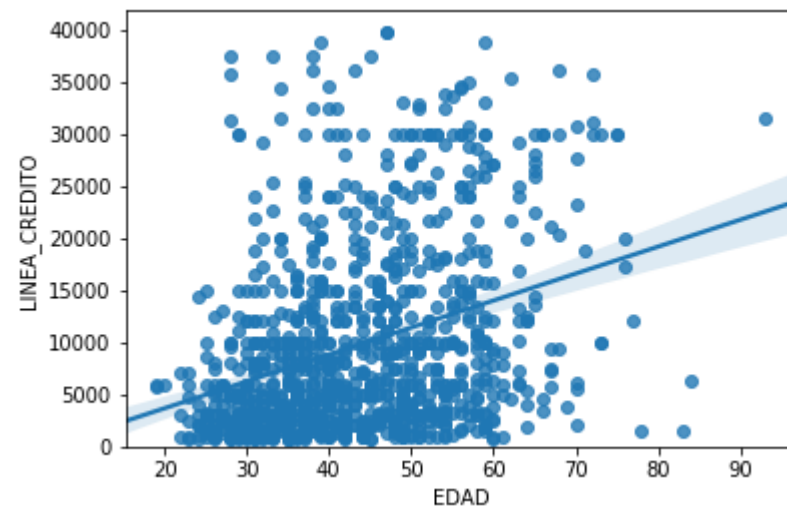
```
x = datos[["LINEA_CREDITO","Total_facturacion"]]  
y = datos["EDAD"]  
  
lm.fit(x,y)  
  
lm.coef_  
lm.intercept_
```

Programación en Python:

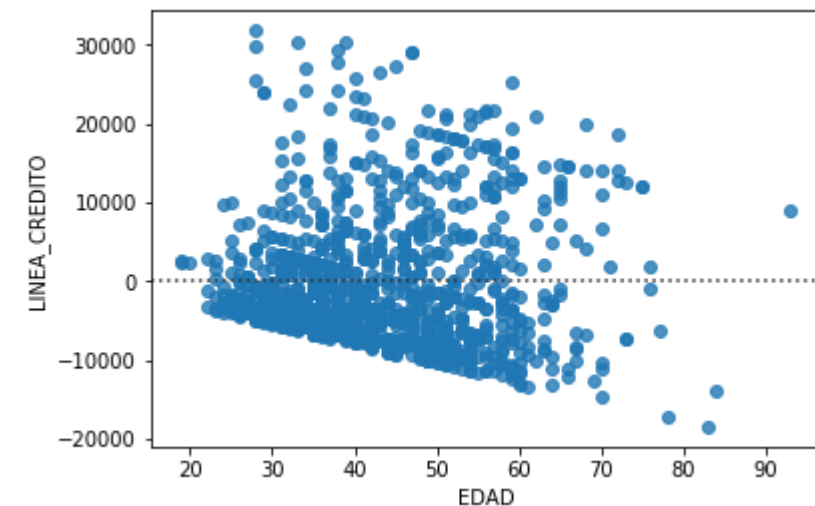
Regresión lineal simple

```
import seaborn as sns #Cargamos el paquete seaborn con alias sns
```

```
#Correlación grafica a partir de regresión
sns.regplot(x = "EDAD",y = "LINEA_CREDITO",data = datos)
plt.ylim(0,)
```



```
#Gráfico de residuos
sns.residplot(datos["EDAD"],datos["LINEA_CREDITO"])
```



Gracias!

Soy [César Quezada]

**Apasionado del mundo de los datos
Especialista en Analítica en DMC**

Me puedes encontrar como



cesar.quezada@dmc.pe



[www.linkedin.com/in/quezada7ba19382]



[cesar.quezada19]