

CARTOGRAFÍA GEOTÉCNICA

Edier V. Aristizábal G.

evaristizabalga@unal.edu.co

Version:August 25, 2020

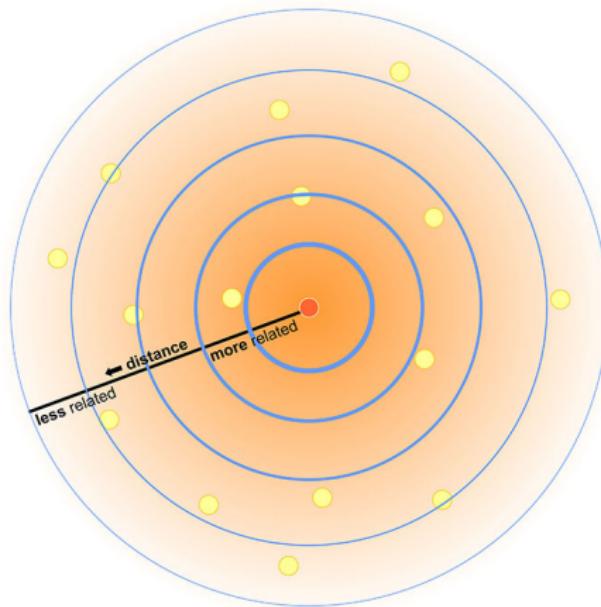




EVERYTHING
HAPPENS
SOMEWHERE

Geospatial Data Science

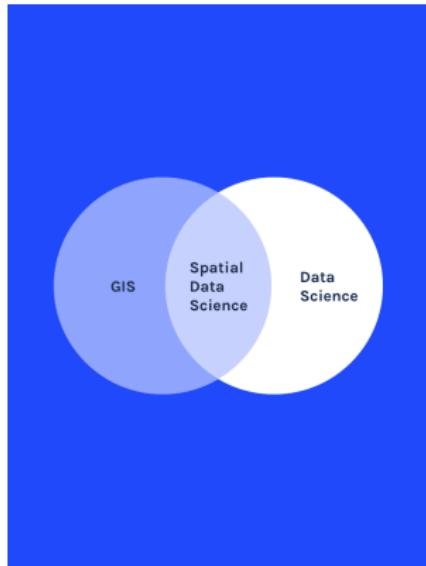
Waldo Tobler's First Law of Geography: *Everything is related to everything else, but near things are more related than distant things.*



<https://carto.com/what-is-spatial-data-science/>

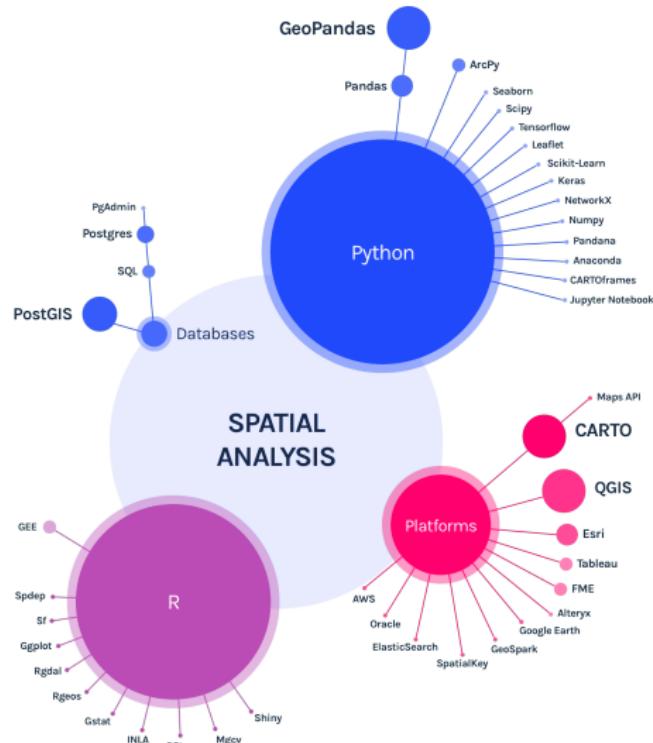
Geospatial Data Science

Geospatial data science (GDS) is a subset of Data Science that focuses on the unique characteristics of spatial data, moving beyond simply looking at **where things happen to understand why they happen there**.

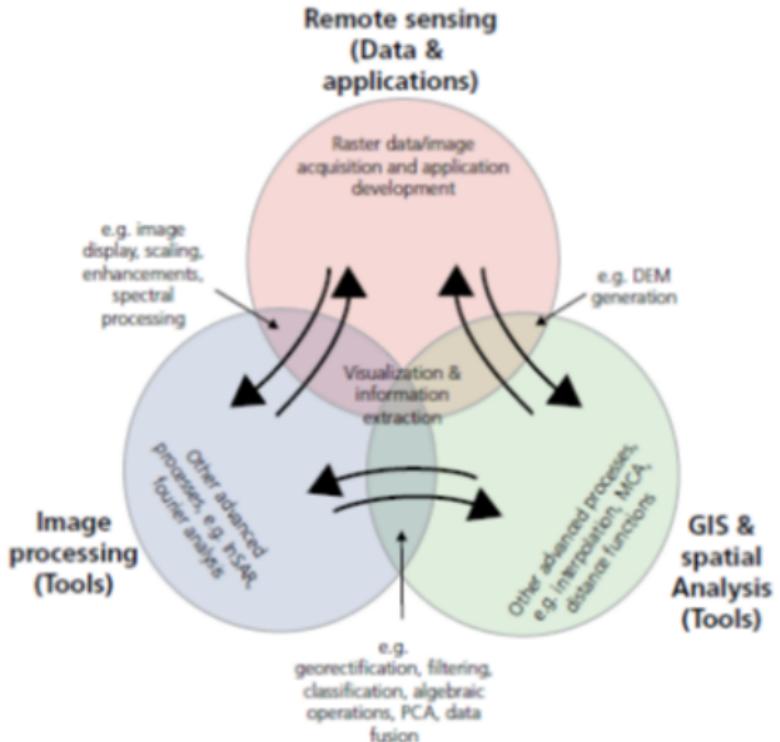


<https://carto.com/what-is-spatial-data-science/>

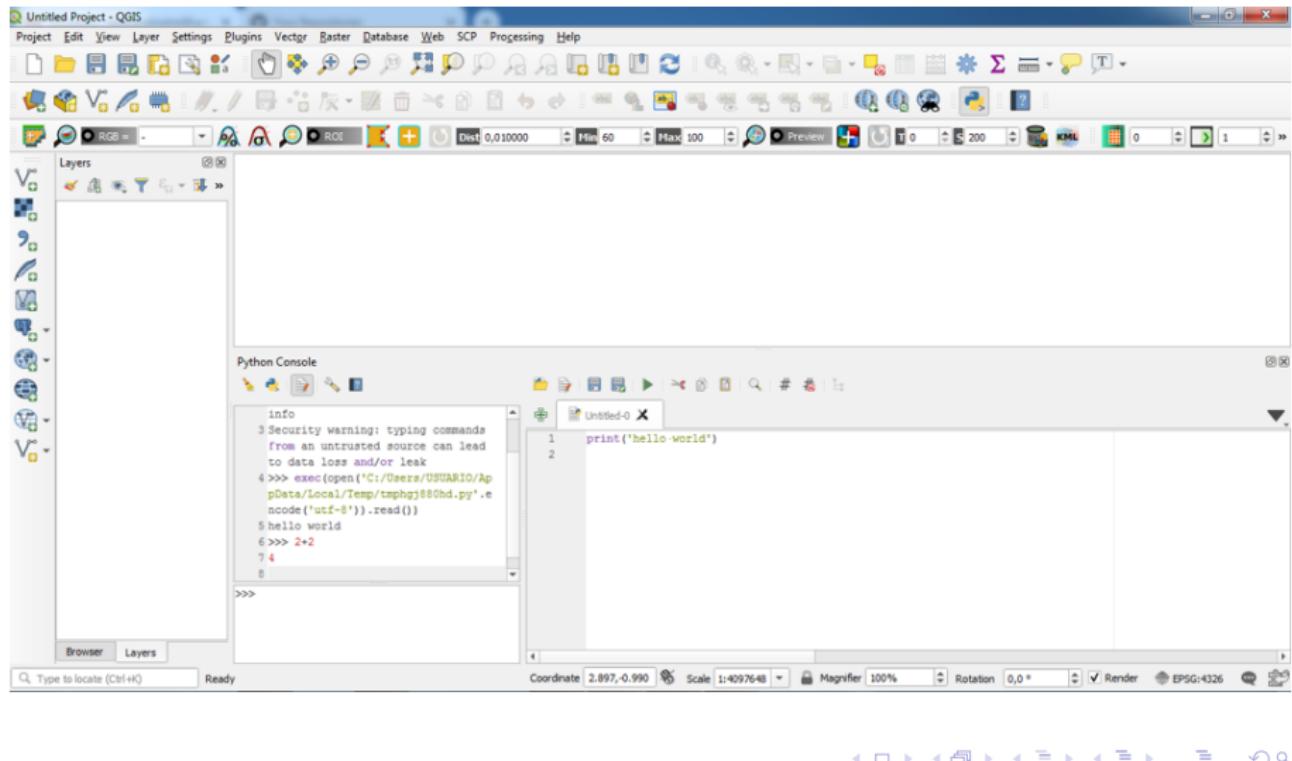
Geospatial Data Science



<https://carto.com/what-is-spatial-data-science/>



QGIS





OSGeo

News Wiki Contact Sign in

Projects Resources About OSGeo Initiatives Community

[Home](#) » [Projects](#) » [OSGeo4W](#)

OSGeo4W

OSGeo4W is a binary distribution of a broad set of open source geospatial software for Windows environments (Windows 10 down to XP).

« Back to projects



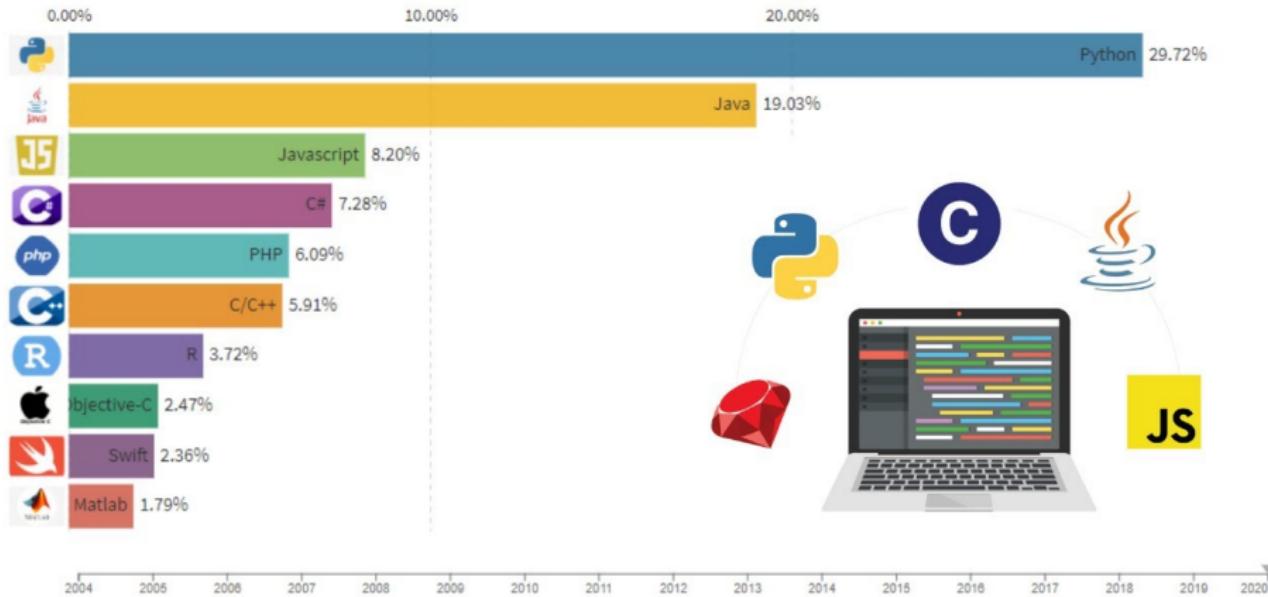
OSGeo4W
FOSSGIS for Windows

Visit our website

OSGeo4W includes open source GIS desktop applications ([QGIS](#), [GRASS GIS](#)), geospatial libraries ([PROJ](#), [GDAL/OGR](#), [GEOS](#), [SpatiaLite](#), [SAGA GIS](#)), scripting languages ([Python](#)) as well as [many other geospatial related packages](#).

Top 10 Most Popular Programming Languages

2004 to 2020



Python code is fast to develop: As the code is not required to be compiled and built, Python code can be much readily changed and executed. This makes for a fast development cycle.

Python code is not as fast in execution: Since the code is not directly compiled and executed and an additional layer of the Python virtual machine is responsible for execution, Python code runs a little slow as compared to conventional languages like C, C++, etc.

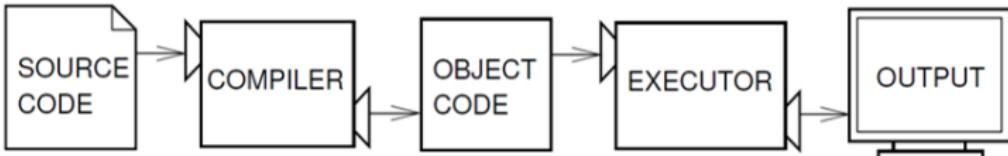
It is interpreted: Many programming languages require that a program be converted from the source language, such as C++ or Visual Basic, into binary code that the computer can understand. This requires a compiler with various options. Python is an interpreted language, which means it does not need compilation to binary code before it can be run. You simply run the program directly from the source code, which makes Python easier to work with and much more portable than other programming languages.

It is object oriented: Python is an object-oriented programming language. An object-oriented program involves a collection of interacting objects, as opposed to the conventional list of tasks. Many modern programming languages support object-oriented programming. ArcGIS and QGIS is designed to work with object-oriented languages, and Python qualifies in this respect.

An **interpreter** reads a high-level program and executes it, meaning that it does what the program says. It processes the program a little at a time, alternately reading lines and performing computations.



A **compiler** reads the program and translates it completely before the program starts running. In this context, the high-level program is called the source code, and the translated program is called the object code or the executable. Once a program is compiled, you can execute it repeatedly without further translation.





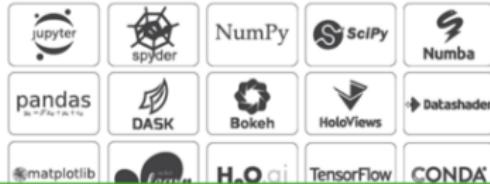
Anaconda Distribution

The World's Most Popular Python/R Data Science Platform



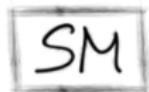
The open-source [Anaconda Distribution](#) is the easiest way to perform Python/R data science and machine learning on Linux, Windows, and Mac OS X. With over 15 million users worldwide, it is the industry standard for developing, testing, and training on a single machine, enabling *individual data scientists* to:

- Quickly download 1,500+ Python/R data science packages
 - Manage libraries, dependencies, and environments with [Conda](#)
 - Develop and train machine learning and deep learning models with [scikit-learn](#), [TensorFlow](#), and [Theano](#)

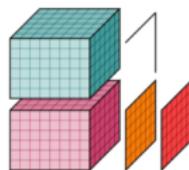


ACCEPT

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



StatsModels
Statistics in Python



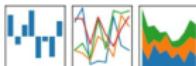
xarray



And many,
many more...



pandas
 $y_{it} = \beta^t x_{it} + \mu_i + \epsilon_{it}$



bokeh



NumPy



IP[y]:
IPython



Anaconda Navigator

Sign in to Anaconda Cloud

Refresh

Home Environments Learning Community Documentation Developer Blog Feedback

Twitter YouTube GitHub

Applications on base融境 Channels

jupyterlab 0.32.1 An extensible environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture. Launch

jupyter 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 1.27.1 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

Anaconda Navigator

File Help

ANACONDA NAVIGATOR

Upgrade Now Sign in to Anaconda Cloud

Home Environments Projects (beta) Learning Community Documentation Developer Blog Feedback

Create Clone Import Remove

Search Environments root

Installed Channels Update index... Search Packages

Name	T	Description	Version
_ipyw_jlab_nb_ext...			0.1.0
alabaster		Configurable, python 2+3 compatible sphinx theme	0.7.10
anaconda			5.0.1
anaconda-client		Anaconda.org command line client library	1.6.5
anaconda-project		Reproducible, executable project directories	0.8.0
asn1crypto		Asn.1 parser and serializer	0.22.0
astroid		Abstract syntax tree for python with inference support	1.5.3
astropy		Community-developed python library for astronomy	2.0.2
babel		Utilities to internationalize and localize python applications	2.5.0
backports			1.0
backports.shutil_...			1.0.0
beautifulsoup4		Python library designed for screen-scraping	4.6.0

217 packages available

```
cmd Command Prompt
Microsoft Windows [Version 10.0.10240]
(c) 2015 Microsoft Corporation. All rights reserved.

C:\Users\Brennan>_
conda create xxxx
conda activate xxxx
deactivate
conda intall xxxx
conda uninstall xxx
conda update xxx
```

Python Packaging Index

The screenshot shows the PyPI project page for 'pip'. At the top, there's a navigation bar with links for 'Help', 'Donate', 'Log in', and 'Register'. Below the header, the project name 'pip 19.1.1' is displayed, along with a 'Latest version' button. A 'pip install pip' button is also present. The page notes that it was last released on May 6, 2019. A descriptive text at the bottom states: 'The PyPA recommended tool for installing Python packages.'

pip 19.1.1

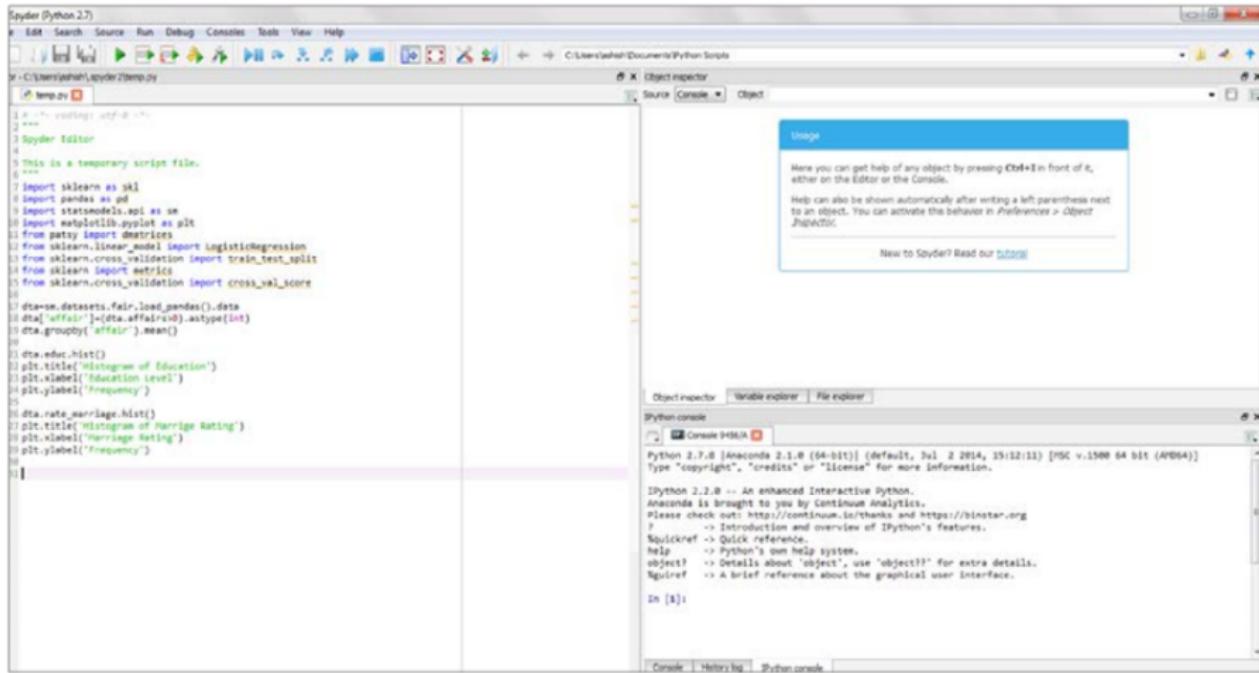
pip install pip

Last released: May 6, 2019

The PyPA recommended tool for installing Python packages.

<https://pypi.org/project/pip/>

Spyder



Jupyter Lab

Notebook .ipynb

The screenshot shows the Jupyter Lab interface. On the left, there is a file tree under 'MACHINE LEARNING > NOTEBOOKS' containing various Jupyter notebooks (ipynb files) from 01_Ambiente_de_trabajo to 20_Ensembles.ipynb, along with a 'tree.dot' file. The main area displays a notebook titled '15_Regresion_lineal.ipynb'. The code cell [1]: contains the following Python code:

```
from sklearn.model_selection import KFold
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import train_test_split
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
import matplotlib.pyplot as plt
import numpy as np
```

The text cell [2]: contains the following explanatory text:

La Regresión Lineal (RL) asume que las variables predictoras tienen una distribución Gaussiana y que no son correlacionables; por lo tanto es importante en el análisis de datos explorar estas condiciones. Adicionalmente, hay que tener en cuenta que los modelos de RL son modelos geométricos, es decir que se basan en calcular las distancias entre la predicción y los datos reales; por lo tanto es importante normalizar o estandarizar los datos.

The code cell [3]: contains the following Python code:

```
from sklearn.datasets import load_boston
boston=load_boston()
X=boston.data
y=boston.target
X_train, X_test, y_train, y_test=train_test_split(X, y, test_size=0.3)
```

The text cell [4]: contains the following explanatory text:

En este Taller utilizaremos la base de datos Boston disponible en la librería Sklearn para problemas de regresión. Por lo tanto la cargamos, sepáramos en datos de entrenamiento y datos de validación con un 30%, y definimos como variables predictoras X^* , y variable dependiente "y".

The title of the notebook is '15_Regresion_lineal.ipynb'.