

A topographic map of a river system, likely the Fraser River, shown in a dark grey relief. The river channels are highlighted with a vibrant red and orange color gradient, indicating a specific data layer or analysis. The text is overlaid on the left side of the map.

GEOG 450/650

Programming

alex.bevington@unbc.ca

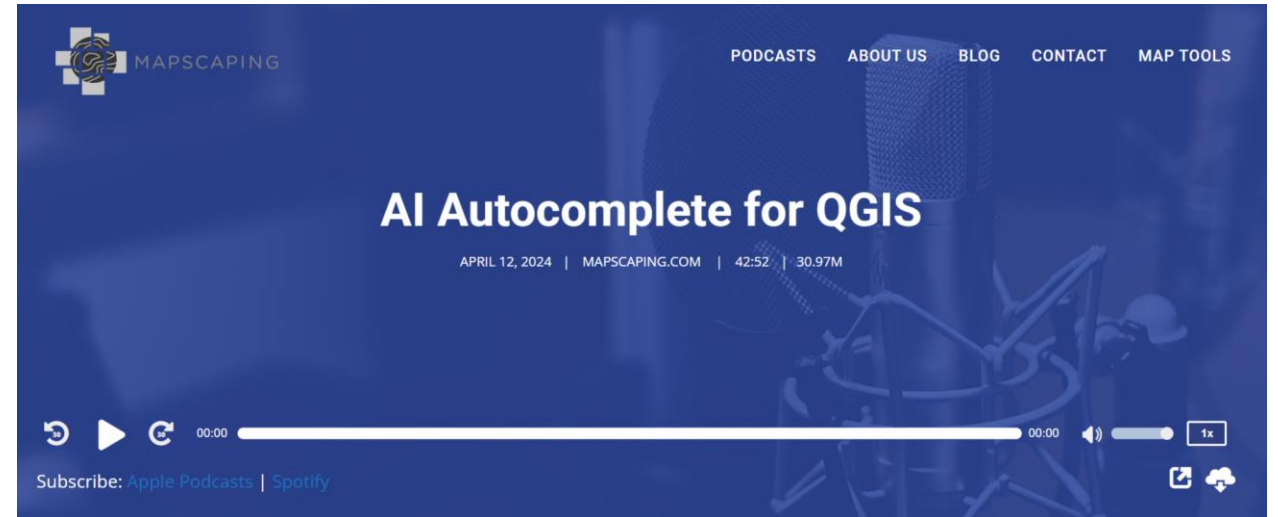
2025-01-14

Housekeeping

- **Syllabus:** updates on course website
- **Reminders:**
 - Create (free) Google Earth Engine account by Thursday
 - Read Coetzee et al. (2020) by Thursday
- **Project** description will be handed out next Tuesday
- **Mid-term** will be based the guest speaker presentations

Podcast discussion

- Main points
- What's exciting
- Implications



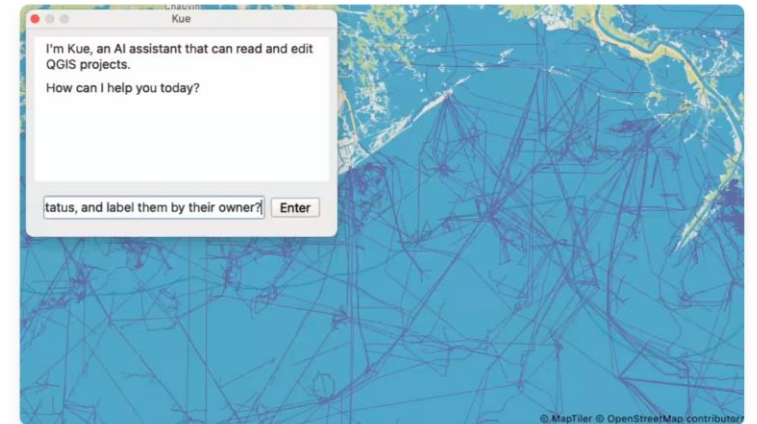
The Suite of AI Tools Built Into QGIS

Kue AI Assistant

AI Vectorizer

AI Georeferencer

Deed Digitizer



Kue can read and edit your QGIS project, integrating a chatbot with your map data.

Learn more →

Is it worth learning to code in 2025?



Reasons to learn coding

- Automate repetitive tasks
- Analyze large datasets
- Create reproducible workflows
- Widely applicable skill
- Many workflows don't exist
- Cutting edge

```
Launcher demo.ipynb Python (ox)
```

```
[1]: import contextily as cx
import geopandas as gpd

Load a shapefile of california census tracts as a GeoDataFrame:
```

```
[2]: gdf = gpd.read_file('./tl_2019_06_tract/')
gdf.shape # output GeoDataFrame count of rows, columns
```

```
[2]: (8057, 13)
```

```
Filter the GeoDataFrame to retain only tracts in Southern California counties:
```

```
[3]: social = ['025', '029', '037', '059', '065', '071', '073', '079', '083', '111']
gdf = gdf[gdf['COUNTYFP'].isin(social)]
gdf.shape
```

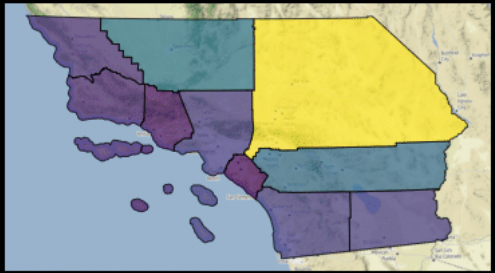
```
[3]: (4879, 13)
```

```
Dissolve the tracts into counties, summing numerical field, then project the counties to the CRS of the California State Plane Coordinate System zone 5:
```

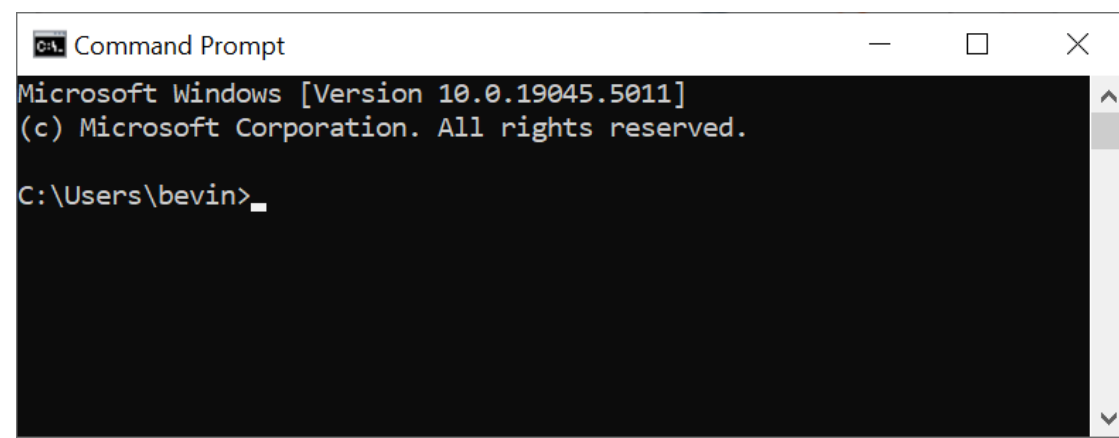
```
[4]: counties = gdf.dissolve(by='COUNTYFP', aggfunc='sum')
counties = counties.to_crs('EPSG:3497')
```

```
Plot a choropleth map of the data, with counties colored by land area (km2):
```

```
[5]: counties['ALAND'] = counties['ALAND'] / 1e6 # convert m2 to km2
ax = counties.plot(column='ALAND', cmap='viridis', edgecolor='k', alpha=0.7, legend=True)
cx.add_basemap(ax, crs=counties.crs.to_string()) # add basemap with contextily
_ = ax.axis('off')
```



Programming 101: Command line



```
Command Prompt
Microsoft Windows [Version 10.0.19045.5011]
(c) Microsoft Corporation. All rights reserved.

C:\Users\bevin>
```

- A text-based interface for interacting with your computer, allowing users to execute commands directly.
 - Prompt: Indicates the system is ready to receive commands.
 - Commands: Text instructions to perform actions.
 - Arguments: Additional options or data passed to commands.
- Advantages:
 - Fast and efficient for repetitive tasks.
 - Enables automation through scripts.
 - Direct access to system functions not always available via GUI.

Programming 101: Command line

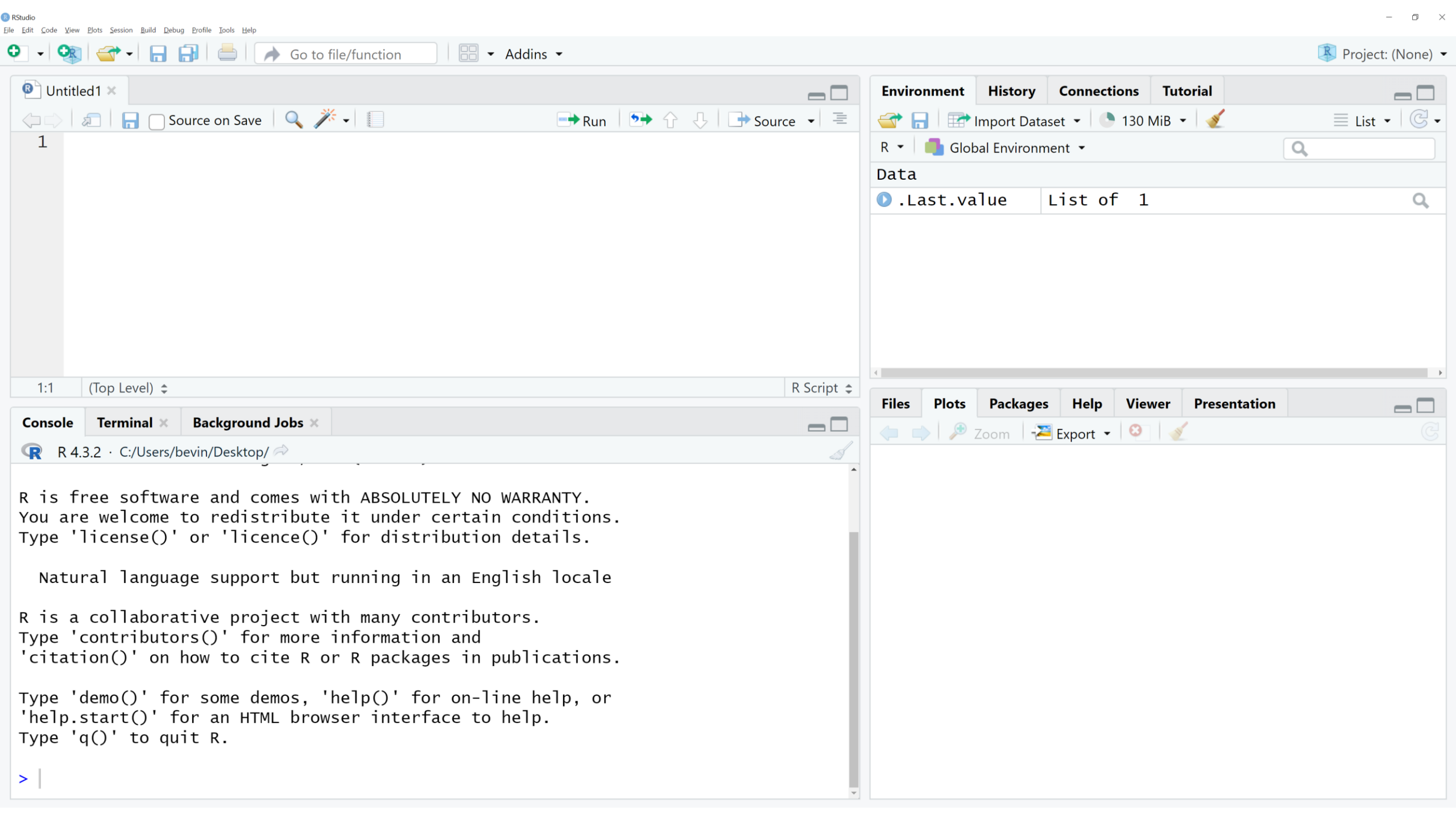
- General Syntax:
 - `command [options] [arguments]`
- Examples:
 - List files and directories: `ls -l`
 - Change directory: `cd /path/to/directory`
 - Create a file: `touch filename.txt`
 - Useful for running software/scripts, etc.
- Tons of learning resources online (e.g. command line ninja!)



Overview of R



- Originally designed for statistics and data visualization
- Very popular with ecologists, hydrologists, biologists, etc.
- Less for geospatial development, but still very powerful.
- Key features:
 - Powerful data manipulation and statistical packages
 - Rich visualization libraries
- R Studio is the most popular IDE



```
1
```

```
R is free software and comes with ABSOLUTELY NO WARRANTY.  
You are welcome to redistribute it under certain conditions.  
Type 'license()' or 'licence()' for distribution details.  
  
Natural language support but running in an English locale  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> |
```

.Last.value	List of 1
-------------	-----------

Python

- General-purpose programming language
- Very popular in geospatial analysis
- Well integrated with QGIS, ArcGIS, Earth Engine, etc.
- Key features:
 - Versatility (data science, web development, etc.)
 - Extensive libraries and frameworks
- Many IDEs (Jupyter, VS Code, PyCharm, Spyder, etc.)

File Edit View Insert Cell Kernel Widgets Help

Trusted Python 3

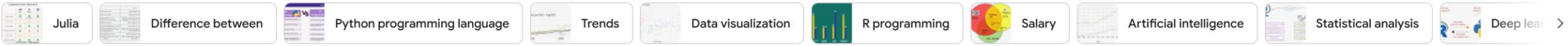
Save + Copy Paste Up Down Run Stop Refresh Run Code Keyboard Voilà

In []:

Other languages, for example...

- **General-Purpose:** Java, C, C++, C#, Rust
- **Web:** JavaScript, TypeScript, PHP, Ruby, HTML & CSS
- **Data Science:** Julia, MATLAB, SAS
- **Database and Query Languages:** SQL, MQL, etc.





Parameter	R	Python
Objective	Data Analysis and Statistical Modeling	Data Science, Web Development, Embedded Systems
Workability	Consists of many apps for mac/linux	Can easily perform public computation as well as optimization
Integration	Locally Run Programs	Programs integrated with web apps for easy deployment
Database Handling Capacity	Proves problems for handling large dataset	Can handle large data easily without any issue
IDE	Rstudio, R GUI	Jupyter (Python, Rupyter Notebook)
Essential Packages and Library	ggplot2, tidyverse, caret	Flumpy, pandas, numpy, web-scrap, tensorflow

DataFlair R Vs Python - The most trending debat...



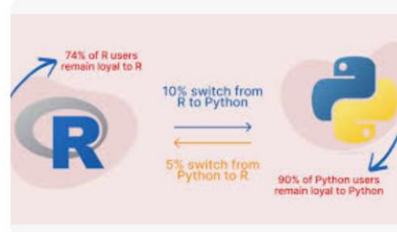
GeeksforGeeks R vs Python - GeeksforGeeks

PYTHON VS R

Python is a generalist "general purpose" programming language. R is a programming language designed for data visualization, data analysis, and statistical computing.

Python	R
Web dev	Clustering data
Mobile game dev	Visualizing data
Machine learning	Analyzing data
Artificial intelligence	Evaluating algorithms
Data visualization	Deep learning

Pathrise Python vs R: Data sc...



Extern Labs R vs Python: Which is Better For Data ...

SUMMARY

R	VS	PYTHON
OVERVIEW		
TYPE OF USER		
SYNTAX		
LEARNING CURVE		
LIBRARIES		
VISUALIZATION		

phoenixNAP R vs. Python - A D...

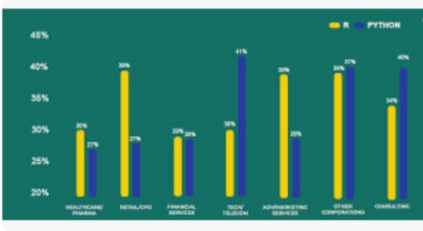


Imaginary Cloud R vs Python: The Data Science langu...

Difference Between R and Python

Features	R	Python
Scope	Used mostly for statistical modeling	Used for a variety of purposes like web-application development and data analysis
Used By	Statisticians, Analyst & Data Scientist	Developer, Data Engineers & Data Scientist
Suitable For	People with no prior experience in programming	Newbies to experienced IT professionals
Package Distribution	CRAN	PyPi
Visualization Tools	ggplot2, plotly, gggraph	Matplotlib, bokeh, seaborn

DataFlair R vs Python for Data Science - And th...



GeeksforGeeks R vs Python - GeeksforGeeks

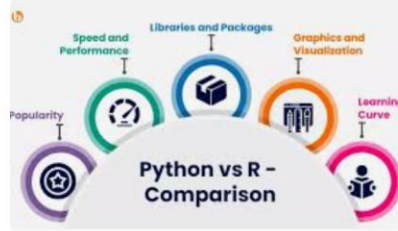
Python Programming Vs R Programming

Python provides a more general approach to data science.	R is mainly used for statistical analysis.
Python users are mostly Programmers and Developers.	R users mainly consists of scholars and R&D professionals.
Python is linear and smooth to learn.	R is difficult to learn at the beginning.
Python provides flexibility to construct new models from scratch.	R provides flexibility to use available libraries.
Python contains packages and libraries like pandas, numpy, scikit-learn, tensorflow, caret.	R contains various packages and libraries like tidyverse, ggplot2, caret, etc.

The Knowledge Academy R Programming Language vs. Python ...

R	Python
Free and open source	Free and open source
Released in the 1990s	Released in the 1990s
Designed by statisticians for statistical/mathematical computing	Designed as a general-purpose programming language with an emphasis on code reusability
Interpreted language	Multi-paradigm language
Used as an interpreter or command-line to run code	Uses an IDE to run code
Additional functionality available online through downloadable packages	Additional functionality available online through downloadable libraries
Large online support community	Large online support community
Used in fields of data science, research, and statistics	Used in multiple fields for web-development and data science

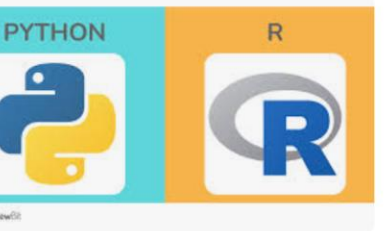
StrataScratch Python vs R for Data Science ...



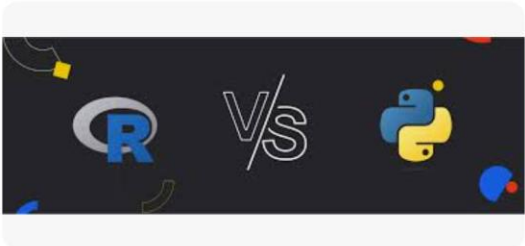
Bacancy Technology Python vs R - The Right Choice For Data ...

	Python	R
General	Python is a general-purpose programming language for data analysis and scientific computing.	R is a functional programming-oriented and language for statistical computing and graphics.
Objective	Python, Python, C/C++, JavaScript, Swift	Data Science & Statistical Modeling
IDE	Python, Python, Jupyter Notebook, Spyder	Can also install on Mac with built-in shell using RStudio which provides a simple way to use R. However, unlike Python (language parity)
Data Collection	It supports CSV files, JSON, XML, and web scraping with BeautifulSoup	Complex data visualization tools make the experience. Data visualization (D3.js) provides much more complex than Python.
Data Analysis	Complex operations with Pandas, Numpy, TensorFlow, etc.	Simple, elegant, ggplot2, Matplotlib, Seaborn, etc.
Essential Packages & Libraries	Numpy, Pandas, Matplotlib, Numpy, Seaborn, TensorFlow	Same, although ggplot2, Matplotlib, Seaborn, etc.
Database Handling Capacity	Can handle large data volumes from one source or multiple sources.	Can handle large data volumes from one source or multiple sources.
Data Visualization	Offers the capabilities of data visualization from the Matplotlib and Seaborn. Python libraries like ggplot2 and plotly are also available.	Developed by and for statisticians. It has complex data visualization features.
System	The use of Python is not restricted to a specific area but can be used in many other areas.	It's functional syntax and easy for beginners, but not challenging for those well-versed in programming. It also offers a few data structures that help to handle large volumes of data.
Learning Curve		

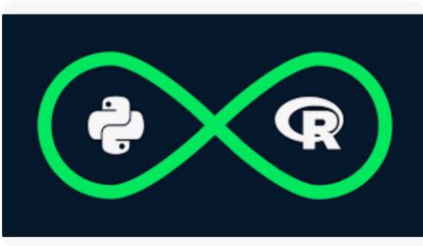
Towards Data Science Python vs R: The Basics. An as...



InterviewBit Python Vs R: Know The Difference ...



Extern Labs



DataCamp



Medium

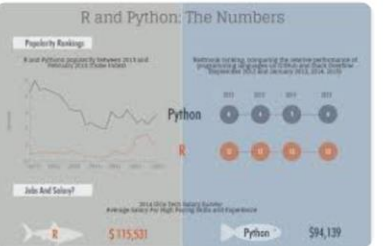
SYNTAX

Python	R
Import Libraries	library(ggplot2) library(Matplotlib)
Import files	read_csv("file.csv")
Assignment	=
Data types	Numeric, Set, Dictionary, Boolean, Sequence Type
Data Structures	Dictionary, Set, Tuple, List
Indexing	starts from [0]
Arithmetic Operators	+, -, *, /, **
Relational Operators	<, >, ==, !=, <=, >=
Logical Operators	AND, OR, NOT

AlmaBetter



Business Science



KDnuggets

We're not so different!

- **R**

```
# Hello, R!  
print("Hello, world!")
```

```
install.packages("dplyr")  
library(dplyr)
```

- **Python**

```
# Hello, Python!  
print("Hello, world!")
```

```
pip install numpy  
import numpy
```

Functionality	Python Package(s)	R Package(s)
Data Manipulation	<i>pandas</i>	<i>dplyr, data.table</i>
Data Visualization	<i>matplotlib, seaborn, plotly</i>	<i>ggplot2, plotly, lattice</i>
Statistical Analysis	<i>statsmodels, scipy.stats</i>	<i>stats, lm, MASS</i>
Machine Learning	<i>scikit-learn, tensorflow, keras</i>	<i>caret, mlr3, xgboost, h2o</i>
Deep Learning	<i>tensorflow, keras, PyTorch</i>	<i>tensorflow, keras, torch</i>
Geospatial Analysis	<i>geopandas, shapely, rasterio</i>	<i>sf, terra, stars</i>
Google Earth Engine	<i>geemap</i>	<i>rgee</i>
Time Series Analysis	<i>statsmodels.tsa, prophet</i>	<i>forecast, zoo, xts</i>
Database Interaction	<i>SQLAlchemy, sqlite3, psycopg2</i>	<i>DBI, RSQLite, Rpostgres</i>
Parallel Computing	<i>dask</i>	<i>parallel, future</i>
Reproducible Reports	<i>jupyter</i>	<i>knitr, rmarkdown</i>
Interactive Dashboards	<i>streamlit</i>	<i>shiny</i>

Category	R Data Types	Python Data Types
Numeric	<i>Integer, numeric</i>	<i>int, float</i>
Character/String	<i>character</i>	<i>str</i>
Logical/Boolean	<i>logical (TRUE, FALSE)</i>	<i>bool (True, False)</i>
Vectors	<i>vector (e.g., c(1, 2, 3))</i>	<i>numpy.array</i>
Lists	<i>List</i>	<i>list</i>
Matrices	<i>Matrix</i>	<i>numpy.array</i>
Data Frames	<i>data.frame, tibble</i>	<i>pandas.DataFrame</i>
Factors/Categorical	<i>factor</i>	<i>pandas.Categorical</i>
Null/Empty	<i>NULL, NA, NaN</i>	<i>None, NaN, pd.NA</i>
Dates and Times	<i>Date, POSIXct, POSIXlt</i>	<i>datetime.date, datetime.datetime</i>

Functions: Example add two numbers

```
R      add_numbers <- function(a, b){  
        return(a + b)}  
add_numbers(5, 3)  
> 8
```

```
Python def add_numbers(a, b):  
        return a + b  
add_numbers(5, 3)  
> 8
```

Logical operators

Operation	R	Python
Equal to	==	==
Not equal to	!=	!=
Greater than	>	>
Less than	<	<
Greater or equal	>=	>=
Less or equal	<=	<=
Logical AND	&	and
Logical OR	`	`
Logical NOT	!	not

R

```
x <- 10  
y <- 20
```

```
if (x > 5 & y < 30) {  
    print("Both are true")  
} else {  
    print("At least one is false") }
```

Python

```
x = 10  
y = 20
```

```
if x > 5 and y < 30:  
    print("Both are true")  
else:  
    print("At least one is false")}
```


Loops

- **R**

```
my_list <- c(1, 2, 3, 4)

for (item in my_list) {
  print(item)}

```
- **Python**

```
my_list = [1, 2, 3, 4]

for item in my_list:
  print(item)

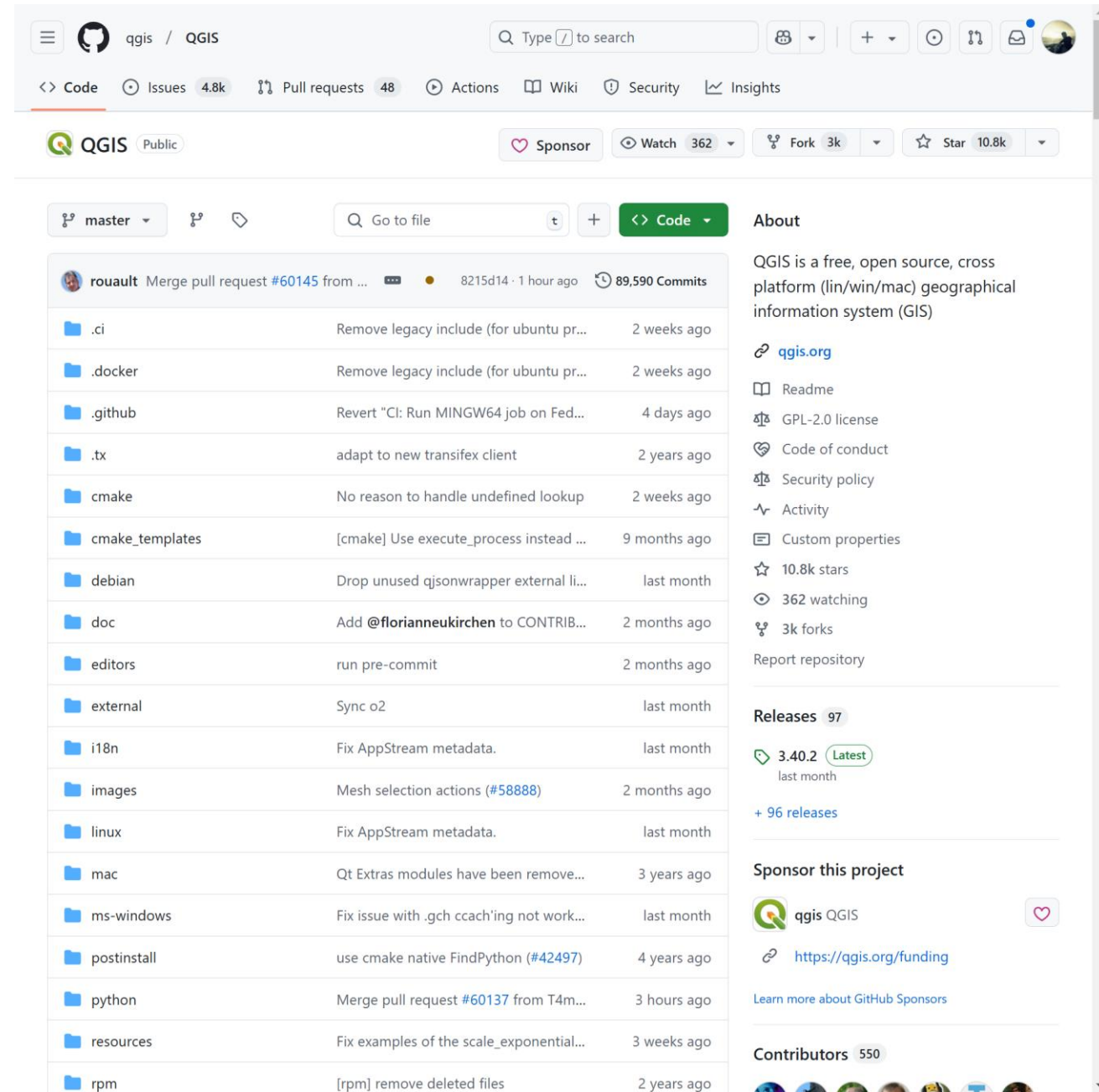
```

Iterating over elements

- In both R and Python, **lapply** and **map** are tools for functional programming that allow you to apply a function to each element in a list or vector.
- They are often used as an alternative to traditional loops because they can be more concise and are often faster.

Version control with git

- Tracks changes in code
- Merge code from many users
- Parallel development without impacting the main repo
- Logs changes over time (good for debugging and review)



The screenshot displays the GitHub interface for the QGIS repository. At the top, the repository name 'qgis / QGIS' is visible, along with search and navigation options. Below the repository name, there are statistics for 'Sponsor', 'Watch' (362), 'Fork' (3k), and 'Star' (10.8k). The main content area shows a file list for the 'master' branch, with columns for file names, commit messages, and commit dates. A 'Merge pull request #60145' notification is visible at the top of the file list. On the right side, the 'About' section provides information about QGIS as a free, open source, cross-platform GIS, including links to the website, license, and other resources. The 'Releases' section shows the latest version '3.40.2' and '+ 96 releases'. The 'Sponsor this project' section includes a link to the QGIS GitHub page and a funding link. The 'Contributors' section shows 550 contributors.

File Name	Commit Message	Commit Date
.ci	Remove legacy include (for ubuntu pr...	2 weeks ago
.docker	Remove legacy include (for ubuntu pr...	2 weeks ago
.github	Revert "CI: Run MINGW64 job on Fed...	4 days ago
.tx	adapt to new transifex client	2 years ago
cmake	No reason to handle undefined lookup	2 weeks ago
cmake_templates	[cmake] Use execute_process instead ...	9 months ago
debian	Drop unused qjsonwrapper external li...	last month
doc	Add @florianneukirchen to CONTRIB...	2 months ago
editors	run pre-commit	2 months ago
external	Sync o2	last month
i18n	Fix AppStream metadata.	last month
images	Mesh selection actions (#58888)	2 months ago
linux	Fix AppStream metadata.	last month
mac	Qt Extras modules have been remove...	3 years ago
ms-windows	Fix issue with .gch ccache'ing not work...	last month
postinstall	use cmake native FindPython (#42497)	4 years ago
python	Merge pull request #60137 from T4m...	3 hours ago
resources	Fix examples of the scale_exponential...	3 weeks ago
rpm	[rpm] remove deleted files	2 years ago

Reproducible workflows demo

- Example from BC ungauged basins project:
 - https://dankovacek.github.io/bcub_demo/notebooks/1_Getting_Started.html
- geemap package in python
 - https://geemap.org/notebooks/18_create_landsat_timelapse/#create-landsat-timeseries