The first thing you want to do is download Python. The latest (as of this writing, at least) is 3.9.5. You can download it from here:

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

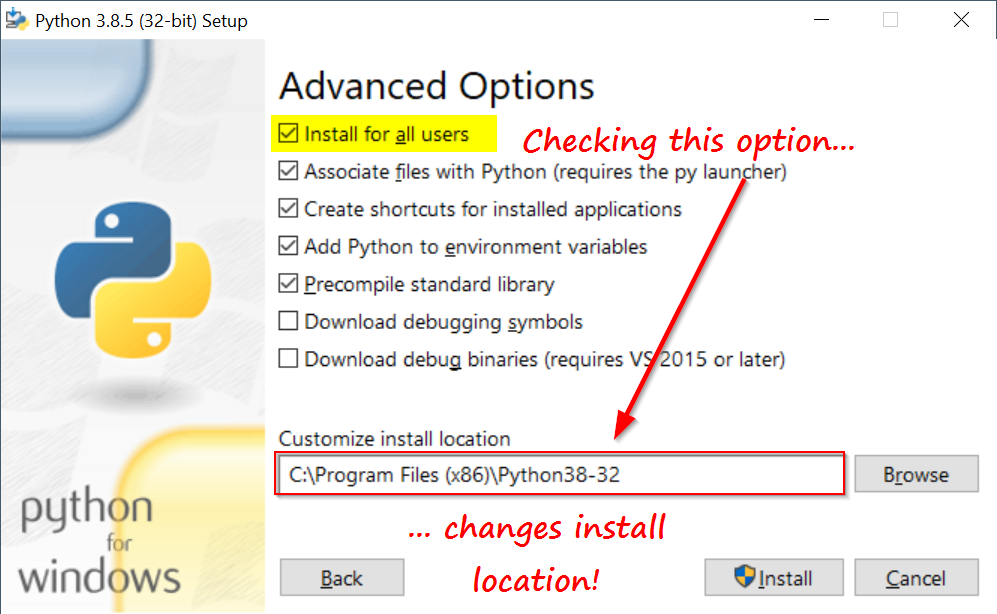
Download Python.

Sample Screenshot:



Check the circled box, and choose “Customize Installation”

Check “Install for all users”



Open a command terminal (DOS prompt)

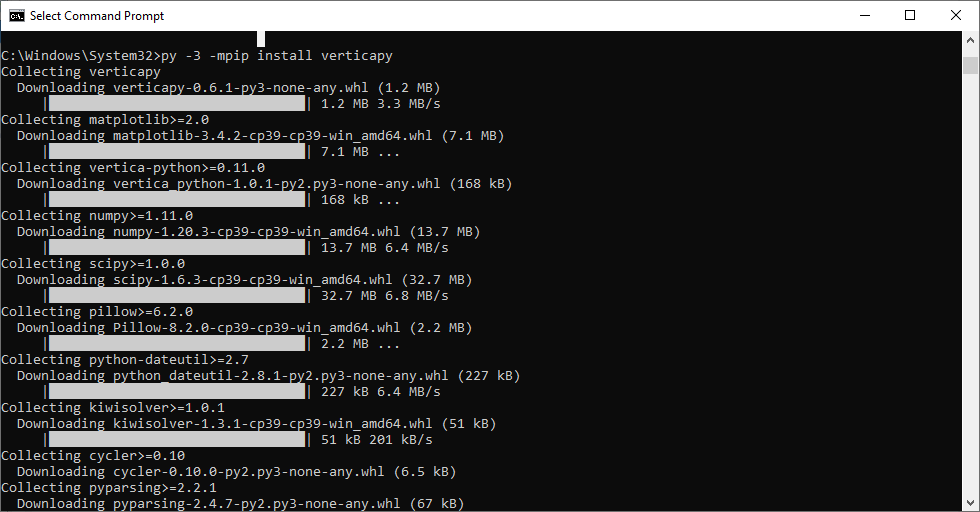
Run:

C:\ pip3 install verticapy

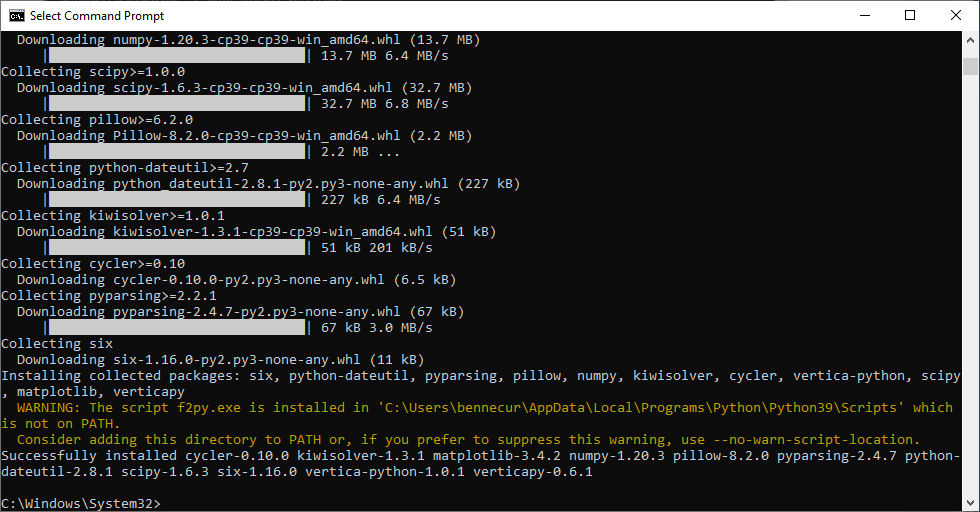
Note: this can fail if it can’t find ‘pip3’ in the PATH. If that happens, this option should work (and accomplishes the same thing):

C:\ py -3 -mpip install verticapy

Sample Screenshot:



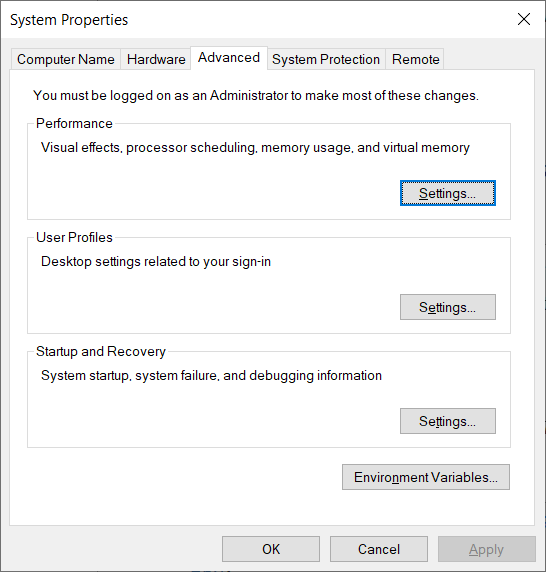
It may be possible to get an error here regarding the “scripts” folder not being in your path, as below:



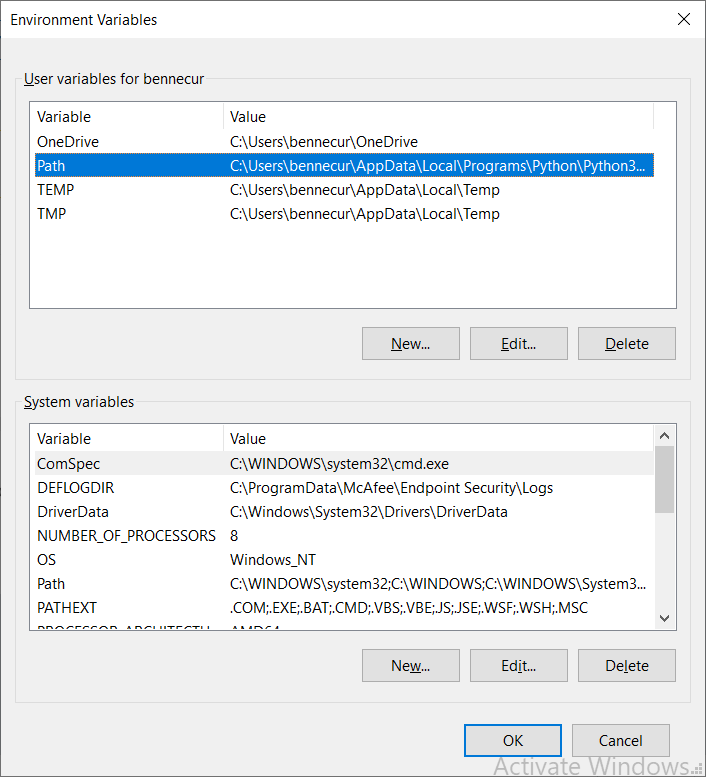
This is apparently due to multiple registry keys that contain trailing nulls in the Windows machine that cause searches to fail. Regardless of the convoluted reason, you can add the scripts folder to your path manually:

Adding Scripts to the PATH:

Go to Start and run, and run “environment variables” or load Control Panel, and search for “environment variables”. It will bring up the “System Properties” dialog.



Click on “Environment Variables” at the bottom.



In the top box, click on “Edit” and then add the necessary path(s).

This shouldn’t require a restart – but you will need to close the instance of the CMD prompt, and open a new one.

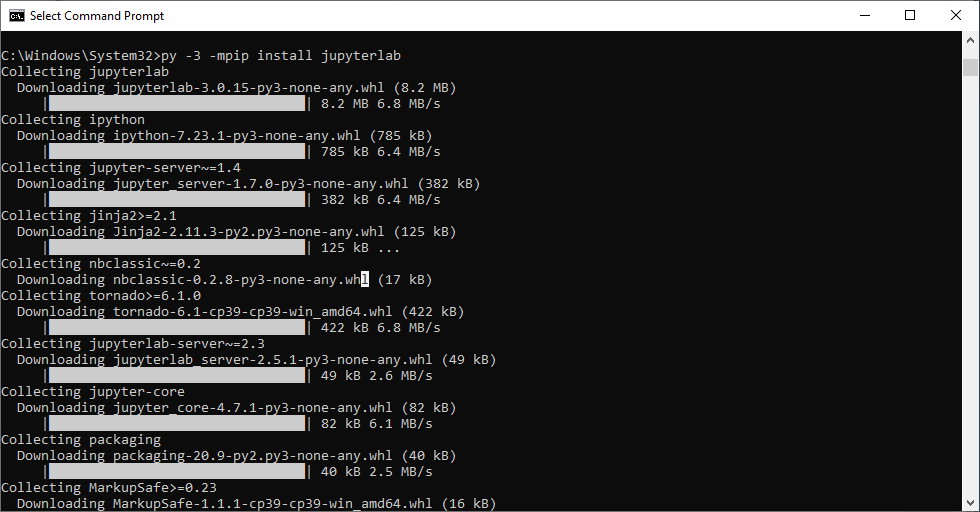
Install JupyterLab. You need not download this – this function works kind of like a “git clone” since it finds the project on Github, and downloads it automatically.

C:\ py -3 -mpip install jupyterlab

Or

C:\ pip3 install jupyterlab

Sample Screenshot:



Once Jupyter is installed, Navigate to a suitable directory (Documents, Desktop, etc) and start a notebook with the following command:

Text

Description automatically generated

The Jupyter hub will open in your browser… If not copy the notebook link from the terminal

Create a new notebook from the dropdown towards the top right

Graphical user interface, application

Description automatically generated

Import the python library into your environment

Graphical user interface, text, application, email

Description automatically generated

Connect to your Vertica Database. You will need the host, port, username, password, and database name to connect. An example of how to make a database connection:

Text

Description automatically generated

Type:  
from verticapy.connect import \*  
conn\_info = {'host': '10.30.30.35',  
'port': '5433',  
'user': 'dbadmin',  
'password': 'dbadmin',  
'database': 'verticadb'}  
new\_auto\_connection(conn\_info, name = 'my\_cluster')  
change\_auto\_connection('my\_cluster')

For more please visit: <https://www.vertica.com/python/workshop/introduction/installation/index>

VerticaPy has many built in datasets to play around with, one of which being the titanic dataset, containing information on the passengers aboard the ship.

A picture containing table

Description automatically generated

Titanic is now a virtual dataframe (learn more: [vDataFrame | VerticaPy](https://www.vertica.com/python/documentation_last/vdataframe/object/index))

Now we can push queries directly to the database. One way to do this is to use the built in SQL Extension: [SQL Magic | VerticaPy](https://www.vertica.com/python/documentation_last/extensions/sql/)

Table

Description automatically generated

Graphical user interface, application, table

Description automatically generated

To calculate the average of a column, one can use both SQL or pandas-like functions in Python. In both cases, VerticaPy pushes the query and computation to the database, and displays the results.

In the first example, the SQL is pushed to the Database. In the second example, the python is converted to SQL first before being pushed to the database for execution. In both cases, the query ran by the database is the same, and the result is the same!

Graphical user interface, application

Description automatically generated

A basic data exploration exercise may be to calculate the correlations between different variables. VerticaPy can plot a heatmap using the corr() function

Chart, waterfall chart, treemap chart

Description automatically generated

VerticaPy can also be used in Python scripts, outside of a Jupyter notebook. An example script is included, titled ‘churn-workflow.py’. This file completes a full example data science pipeline from start to finish, including loading a CSV, preparing and cleaning the data, training a logistic regression model, making predictions, and saving the results to the database. Replace the database connection properties with your personal Vertica instance, and run the script by navigating to the directory of the file, and running ‘churn-workflow.py’.

Here’s what it will look like:

Text

Description automatically generated

For a detailed walkthrough of this example, visit [Telco Churn | VerticaPy](https://www.vertica.com/python/examples/telco_churn/index.php)