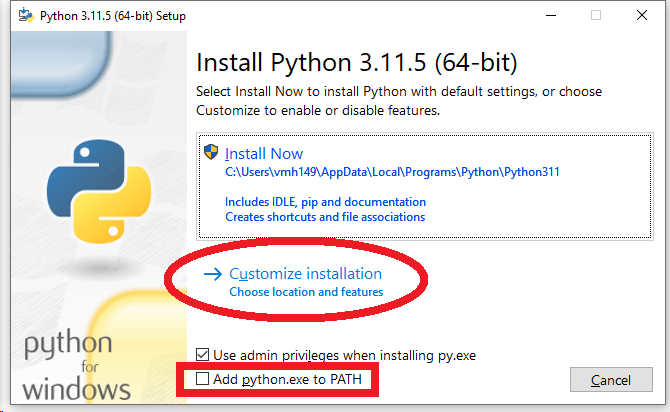
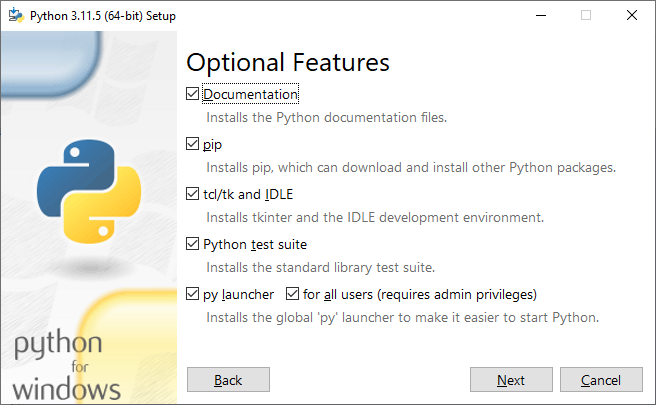
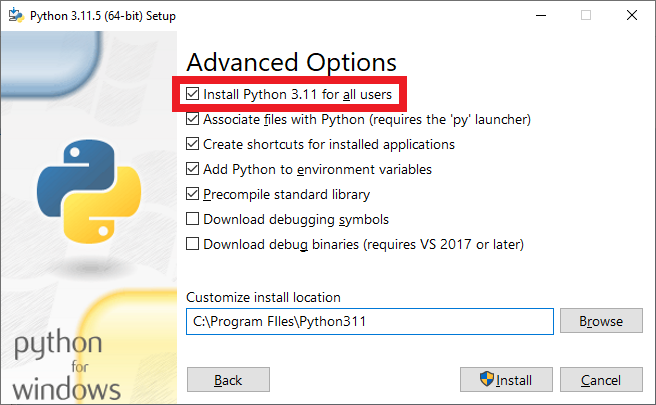
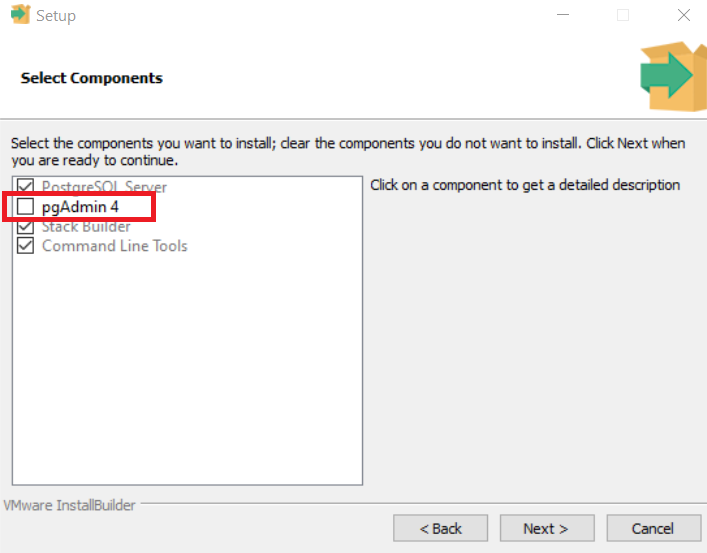
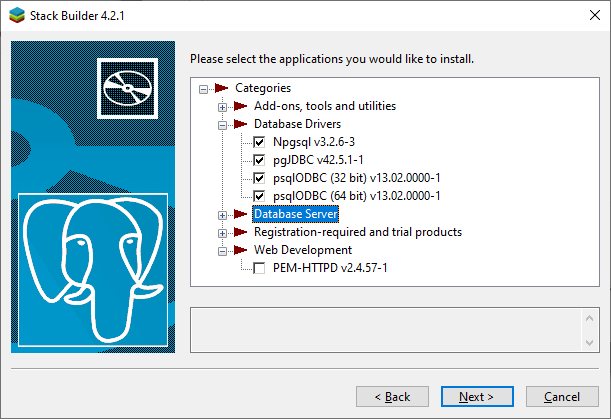
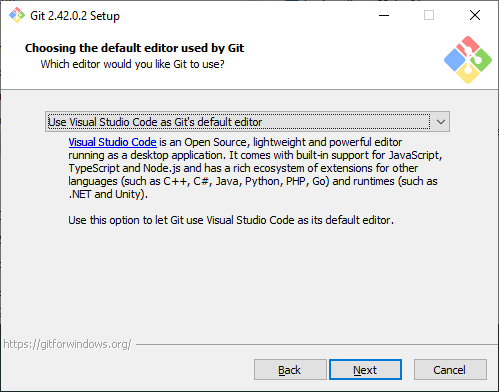
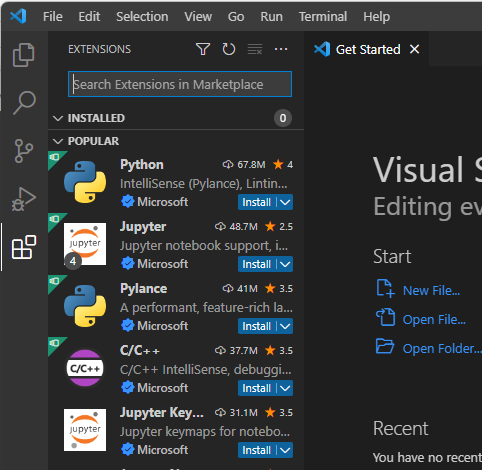
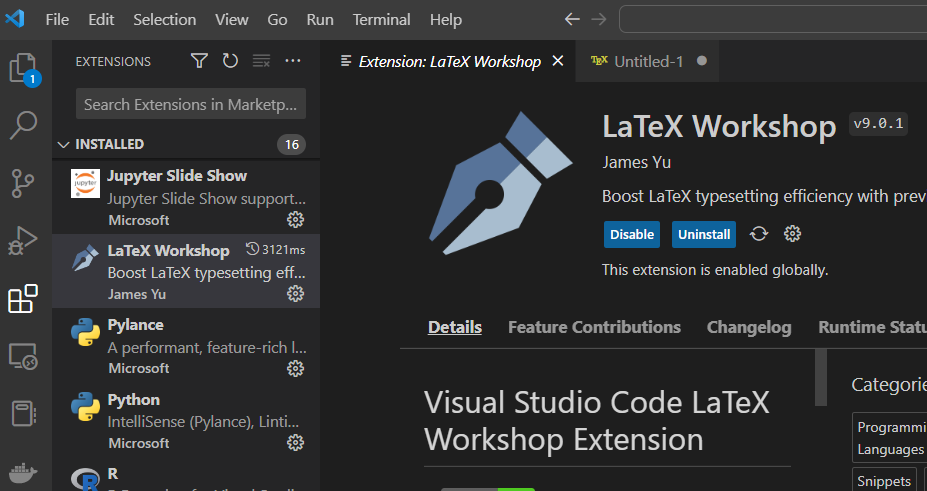
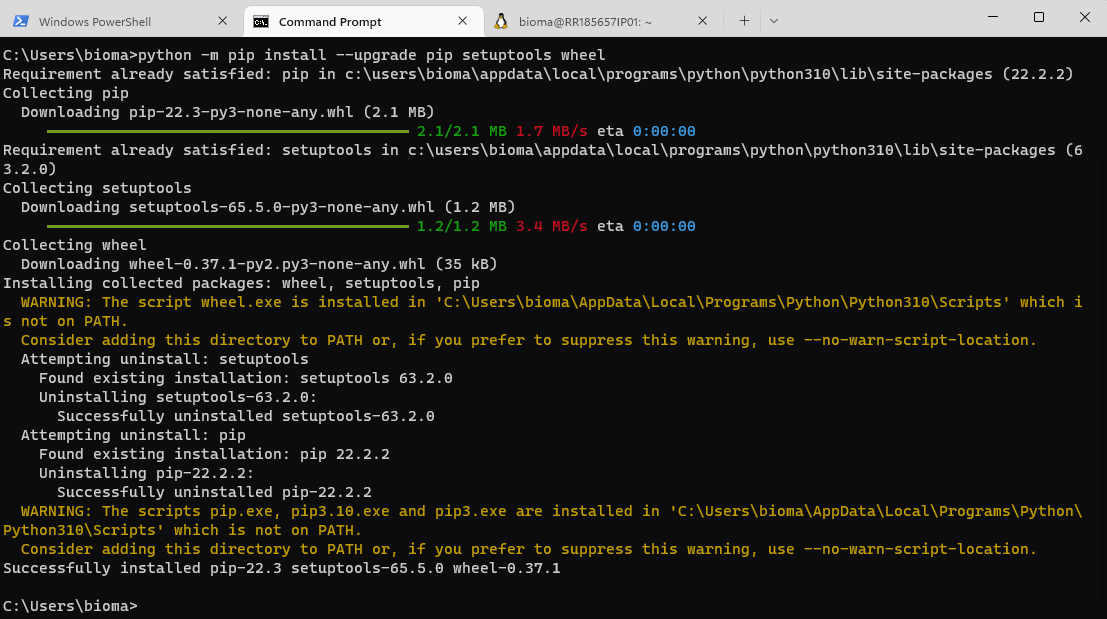
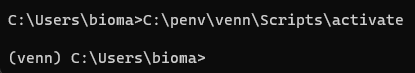
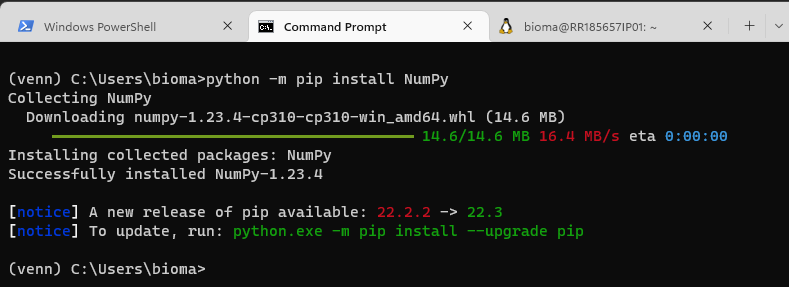
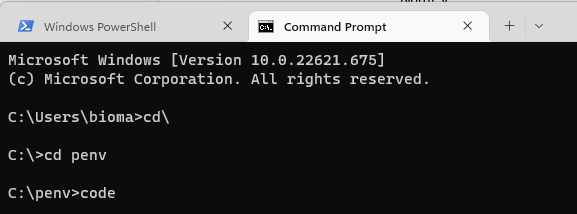
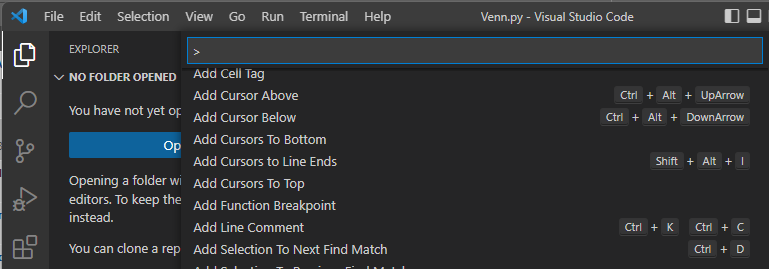
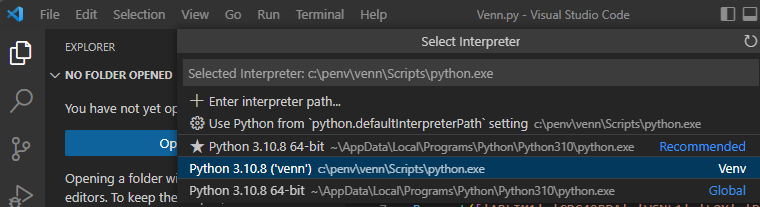
**Environment Setup for Data Analysis under Windows   
(Mostly compatible with Mac & Linux)**

By Juan B. Gutiérrez. [juan.gutierrez3@utsa.edu](mailto:juan.gutierrez3@utsa.edu) - Department of Mathematics. University of Texas at San Antonio

1. **Optional**: Install R: <https://cran.r-project.org/bin/windows/base/>
2. **Required**: Install Python: <https://www.python.org/downloads/>. Install any recent Python version (3.11 is recommended) EXCEPT Python 3.12. 3.12 has some noticeable bugs since it is the newest release. Go to your downloads and double click on the install file to start installation. By default the Python installer for Windows places its executables in the user’s AppData directory, so that it doesn’t require administrative permissions. This works for most scenarios. If you’re the only user on the system, you might want to place Python in a higher-level directory (e.g. C:\Python or /usr/local/bin) to have a shorter path to the binaries (sometimes you will need that). Depending on your preferences, either select “Install Now” or “Customized installation” (my preference). Please make sure you select “*Add python.exe to path*”; this will save you a few headaches later.   
     
     
     
   If you did not add python.exe to PATH at this point, then after the installation finishes, you will have to open File Explorer. Right-click on top of “This PC”. Select properties. In the window that pops up, select “Advanced System Settings”. In the window that pops up, select “Environment variables” in the “Advanced” tab. In System Variables, double click on “path”.   
   For Mac: If you did not add python.exe to PATH during the installation, you will have to open Finder after the installation finishes. Right-click on your Mac’s name and select “Properties”. In the window that pops up, select “Advanced System Settings”. In the window that pops up, select “Environment Variables” in the “Advanced” tab. In System Variables, double-click on “Path”.  
     
   In the window “Optional Features” select all features  
     
     
     
   Select “*Install Python X.XX for all users*” if you can and want. This requires administrative privileges. Select the folder of your choice for the install location.  
     
   
3. **Required**: Install [PostgreSQL](https://www.postgresql.org/). During the initial installation, accept all default options *except* pgAdmin. There is a known bug that will prevent pgAdmin from starting if you select the option pgAdmin in the initial installation. We will install it shortly; instructions below.  
      
     
     
   IMPORTANT: Write down the password you select; you will need to use PostgreSQL. During the Stack Builder step, select all database drivers and ignore other options. An individual installer will be executed. In between the password and the next step is port and locale - leave these as default.  
     
   Stack Builder should launch. Select your local pgsql instance.  
     
     
     
   Once the installation of PostgreSQL Server finishes, install [pgAdmin](https://www.pgadmin.org/) from pgadmin.org; you will need this tool to run and debug SQL queries natively. Open PGAdmin to ensure you can see the server running.
4. **Optional**: Install [SQL Server Developer Edition](https://www.microsoft.com/en-us/sql-server/sql-server-downloads). The basic installation will suffice. Accept all default options and remember the password you will have to select. At the final screen of the basic installation, select “Install SSMS”. SSMS is the SQL Server Management Studio; you will need this tool to run and debug SQL queries natively.
5. **Optional**: Install [MySQL Community Edition](https://dev.mysql.com/downloads/). The full installation will give you all you need. Accept all default options. IMPORTANT: Write down the password you select for user root. You will need it.
6. **Optional**: If you’d like to code in C or C++, you have several options, including [GCC, the GNU Compiler Collection](https://gcc.gnu.org/), but it requires some effort to install the prerequisites. From Microsoft there is [Visual Studio Community Edition](https://visualstudio.microsoft.com/vs/community/) (VSCE), which encapsulates the complexity of installing multiple compilers (including the .NET framework SDK). I recommend you install VSCE with all “Workloads” in “Web & Cloud” (except Python) and “Desktop & Mobile”.
7. **Optional**: If you intend to produce PDF documents using LaTeX, install [MikTeX](https://miktex.org/).
8. **Required**: Install [Visual Studio Code](https://code.visualstudio.com/Download) (VSC). There are two main options: “*User Installer*”and “*System Installer*”. Use the User Installer if you want to install only for the current user. For all users, use System installer (I prefer this choice).   
     
   This little giant has become a darling of coders for many good reasons. It is available for Mac, Linux and Windows. There is an open source identical alternative called VSCodium, available at <https://vscodium.com/> I personally use VSCode.
9. **Required**: Install [Git for Windows](https://gitforwindows.org/). For Mac, install [Git for Mac](https://git-scm.com/download/mac). In the second screen, select Visual Studio Code as your Git editor. Use all other default options.   
   
10. **Required**: Select the Extensions icon on the left and install the following Visual Studio Code libraries
    1. **Required**: Python extension for Visual Studio Code. Also install Jupyter (this also installs Pylance, Jupyter Keymap, Jupyter Notebook Renderers, Jupyter SlideShow). Pay attention to the publisher of the extension; use the ones by Microsoft.   
         
       
    2. **Optional**: Install C/C++ for Visual Studio Code, C/C++ Themes & C/C++ Extension Pack.
    3. **Optional**: Install mssql for Visual Studio Code, SQL Database Projects, and PostgreSQL (by Microsoft).
    4. **Optional**: Install MySQL Shell for VS Code by Oracle Corporation.
    5. **Optional**: Install the extension “**LaTeX Workshop**” by James Yu. You will never use another LaTeX editor :)  
         
       
11. **Optional**: Install [Windows Terminal](https://learn.microsoft.com/en-us/windows/terminal/install). This Microsoft app allows you to use a multi document window with tabs for different command consoles like PowerShell, Ubuntu, DOS, Git, etc. You will need this if you want to complete the steps related to Linux in the next section of this document.
12. **Required**: Open a terminal window by typing “command prompt” in your Windows search bar. Now you are ready to install Python packages. We will start by creating a virtual environment; the reason for this is that sometimes specific versions of different libraries might be incompatible. You should create an environment for each project you will work on.  
    1. All related instructions about packages and virtual environments are at: <https://packaging.python.org/en/latest/tutorials/installing-packages/>
    2. Check your version of pip. It is the most popular tool for installing Python packages, and the one included with modern versions of Python. If the following command returns a version number, you are good to go. Otherwise, reinstall python (or bootstrap it).   
       python -m pip --version
    3. Update the pip and setuptools  
       python -m pip install --upgrade pip setuptools wheel
    4. I like to have an easy-to-access folder for virtual environments, thus I always create a folder like c:\penv for this purpose. To create this folder, run the following command in the terminal: mkdir c:\penv You can name this folder whatever you like. The importance of this will become evident in step 8. Also importantly, the folder in which you create your virtual environments does not need to be the folder in which your code resides.
    5. Now **create** a virtual environment called venn (this is an example for a program about Venn diagrams; you can call it whatever name you prefer):  
       python -m venv c:\penv\venn
    6. To **activate** this environment execute the following command in the terminal in VSC:  
       C:\penv\venn\Scripts\activate  
       Now the command line will show the prompt named as the environment that was activated.  
          
       
    7. To **deactivate** a virtual environment, simply type deactivate in the command prompt.
    8. We will install libraries for this environment. The most important libraries are:  
        NumPy  
        Matplotlib  
        SciPy  
        Jupyter  
        Pandas  
        SciKit-Learn  
        TensorFlow  
        Keras  
        torch  
       For each one of them type the following command: python -m pip install <library name>. For example:  
       python -m pip install NumPy  
       Alternatively, you can simply type

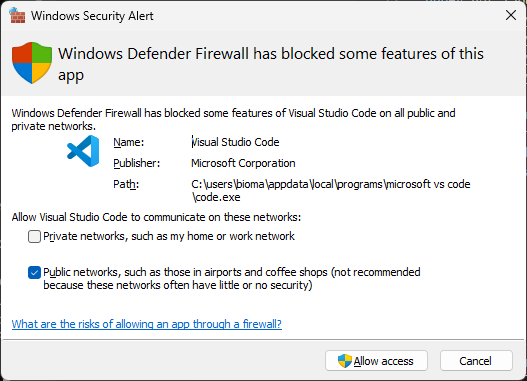
pip install NumPy  
Repeat for the libraries listed above. You should see something like this:  
  


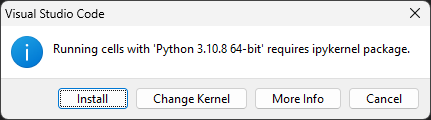
* 1. When you try to install PyTorch, you will encounter an error. The operation will not be successful. Read the error messages. Can you figure out what to do?

1. The reason I recommend having a single directory for Python virtual environments, as described in step 12.d, will become evident now. Open a Terminal window in Command Prompt. Go to the root folder where you installed your virtual environments.
   1. Execute the command code  
        
        
        
      The command code starts VSC. When you start VSC in the root of your virtual environments, all of them become available for code execution. Recall, the folder in which you create your virtual environments does not need to be the folder in which your code resides.
   2. Go to the newly opened windows of VSC. Use keyboard CTRL+SHIFT+P. This will bring up a menu with options.  
        
      
   3. Type python:select interpreter This will show you all the environments available.  
        
      
   4. In the previous screenshot, you see that the virtual environment (Venv) called venn is available. Also an environment called Global, which refers to the base environment. Select the environment in which you want to execute your code. Keep in mind, the difference between environments is the libraries you install. If your code needs a library, for example via the instruction  
      import numpy as np

then the library Numpy must have been installed in that environment

* 1. An important suggestion that comes from experiencing frustration in the past is this: Keep the base environment clean, that is, with minimal libraries. Conflicts between libraries will arise once you install a number of them. This is why it is a good practice to create an environment for each project.

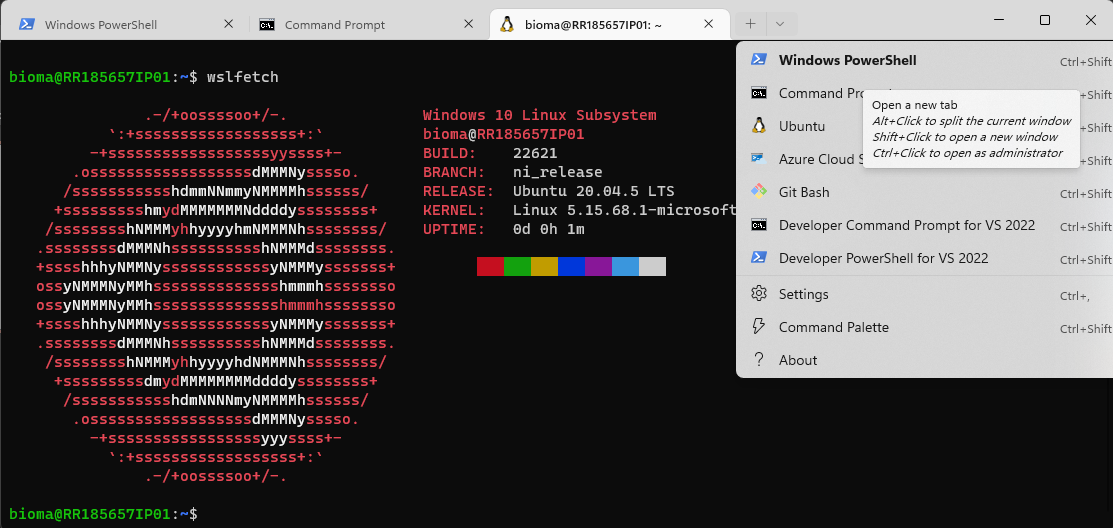
1. Now, create a Jupyter Notebook.
   1. Abundante relevant information regarding Jupyter in VSC is available at  
      <https://code.visualstudio.com/docs/datascience/jupyter-notebooks>
   2. Activate the environment you want to use according to step 12.f
   3. You need to attach the kernel of your virtual environment:  
      python -m ipykernel install --user --name=venn
   4. Execute step 13.a.
   5. In VSC, run the command Create:New Jupyter Notebook from the Command Palette (Ctrl+Shift+P) or by creating a new .ipynb file in your workspace.
   6. You might receive a Security Alert regarding the firewall. Allow access.   
      
   7. After you enter a command, you might see the following message:

  
Install it.

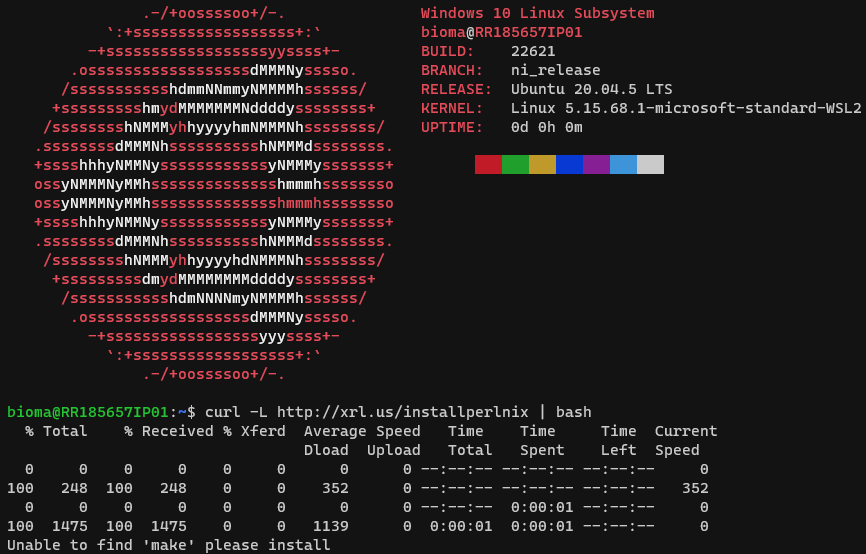
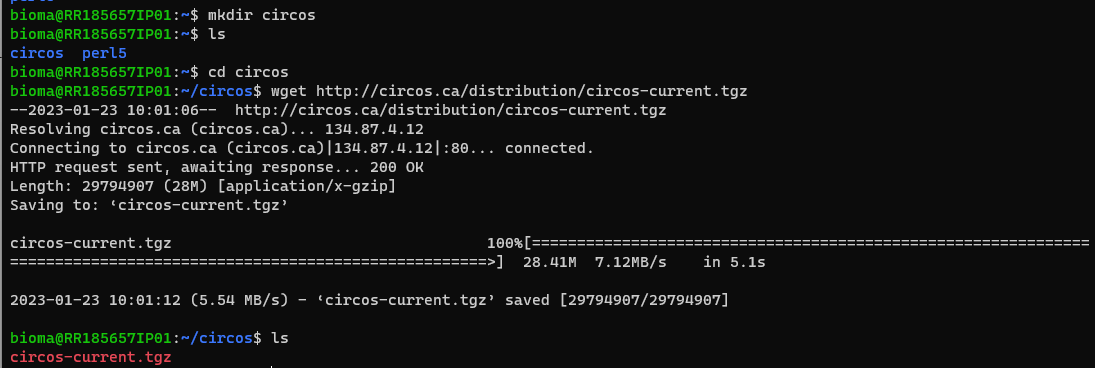
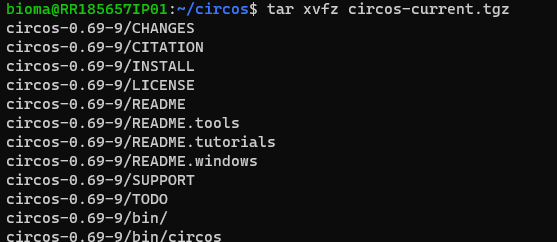
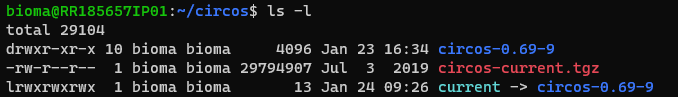
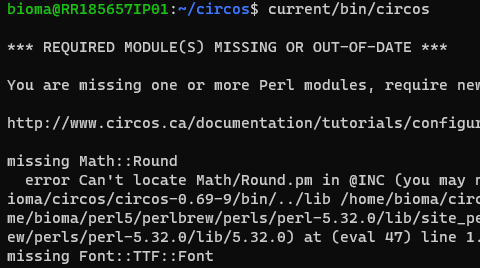
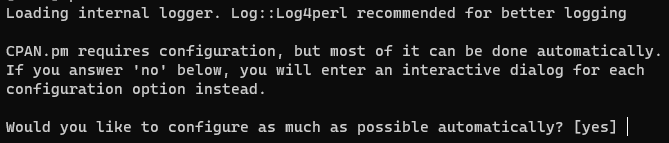
1. Finally, you can simply enter Python commands in the Terminal. Open a Command Prompt and type the following command:  
   python  
   This will allow you to enter python commands

**Optional: Installing a Linux Distribution on Windows & Circos (advanced visualization)**

Circos is an advanced visualization technique. See examples at [circos.ca](http://circos.ca/images/).

Open a Windows Terminal. Install a Linux distribution. Yes, you can (and should) run a fully-functional Linux distribution natively on Windows just like you’d use a command window. I like Ubuntu. Follow the instructions at:   
<https://learn.microsoft.com/en-us/windows/wsl/install> IMPORTANT: Write down the password you select for user root. You will need it.  
  


Step 6 in the previous section pointed to resources to install Ubuntu or other preferred Linux distribution. In this section we will see how to install Perl in Linux. We want to install Perl to have access to libraries such as Circos (advanced visualization techniques).

1. Open a Windows Terminal (step 6 in the previous section).
2. To install Perl, enter the following command:  
   curl -L http://xrl.us/installperlnix | bash  
   Most likely, this will fail.   
     
   The reason it fails is that a utility called make is not present.
3. We will install the GCC compiler, which will include the make utility.
   1. First, update your OS.   
      sudo apt update  
      This will ask you for the password for user root. If you recall, the instructions on step 6 in the previous section asked you to write down that password. This step will show you the libraries that will be updated. Now, enter   
      sudo apt upgrade  
      Accept the installation.
   2. You might see the message   
      The following packages were automatically installed and are no longer required: libfwupdplugin1 libxmlb1  
      You can remove unneeded packages by executing the command   
      sudo apt autoremove
   3. Now, execute again   
      curl -L http://xrl.us/installperlnix | bash  
      This time, it will install.
   4. Once the installation ends, close the terminal window, open a new terminal and test the installation by typing   
      perl -v
4. Now, install circos from [circos.ca](http://circos.ca/).
   1. Make a directory called circos the the following command:   
      mkdir circos
   2. Enter the circos folder with the following command:  
      cd circos
   3. Now, use the utility wget to download the Circos files from the Web:  
      wget <http://circos.ca/distribution/circos-current.tgz>  
      
   4. Now we have a tar file that has been compressed with gzip, indicated by the tgz extension. To decompress it, use the following command:  
      tar xvfz circos-current.tgz  
      x means e**x̲**tract files from the archive.  
      v means print the filenames **v̲**erbosely.  
      f means the following argument is a **f̱**ilename.  
      z means (un)**z̲**ip.  
      
   5. Next, you can create a symbolic link from the folder with the long name circos-0.69-9 to the shortcut current. A symbolic link (also called a symlink) is a file in Linux that points to another file or a folder on your computer. Symlinks are similar to shortcuts in Windows.  
      ln -s circos-0.69-9 current  
      After you create the symlink, get execute ls -l (list files with long format). You will that the symbolic link current points to circos0.69.9  
      
   6. At this point, we can test whether circos has all the requisites installed. Execute the following command:  
      current/bin/circos  
      As you can notice, we used the shortcut current that we created as a symbolic link. You will notice that several components from Perl are missing.   
      
   7. The easiest way to install modules in Perl is to use CPAN, the repository of Perl modules. Execute the following command:  
      sudo cpan  
      If you see   
        
      Hit enter to select [yes]
   8. To install the missing packages, execute the following commands:  
      install Config::General  
      install Graphics::ColorObject  
      install Math::Bezier  
      install Math::VecStat  
      install Readonly  
      install Set::IntSpan  
      To end the CPAN program, enter the command quit
   9. A library called cmake is needed by one of the steps below.   
      sudo apt install cmake
   10. A library called zlib is needed by one of the steps below. So, let’s install it. Note that the flag -O in curl means to write output to a file named as the remote file.  
       curl -O <https://zlib.net/zlib-1.2.13.tar.gz>  
       tar -xzvf zlib-1.2.13.tar.gz  
       cd zlib-1.2.13  
       ./configure  
       make  
       sudo make install  
       cd ..
   11. Now, we have to install GD. Execute the following commands.   
       mkdir srctemp  
       cd srctemp  
       curl -O <http://www.ijg.org/files/jpegsrc.v8d.tar.gz>  
       tar -xzvf jpegsrc.v8d.tar.gz  
       cd jpeg-8d  
       ./configure  
       make  
       sudo make install  
       cd ..  
       Now, we need libpng. But there is a little catch. The web page at <http://www.libpng.org/pub/png/libpng.html> is the official source for this library. The source is located in SourceForge. However, these SourceForge links fail with curl and wget. The trick here is to go the the SourceForge page and riht-click on the DB (Green Download Button). Copy the URL. The use wget as follows (the decreased font is to make it fit in a single line):   
       wget -O libpng-1.6.39.tar.gz <https://sourceforge.net/projects/libpng/files/latest/download>  
       You might be tempted to use the tar command just as we did before. However, if you execute the following command:   
       tar -xzvf libpng-1.6.39.tar.gz  
       You will receive the error: “gzip: stdin has more than one entry--rest ignored” This can happen if the file is actually a ZIP-file rather than a gz-file; it may contain multiple files. Hence, we need to install unzip. Run the following command:   
       sudo apt install unzip  
       Now, we can unzip the file and the compile with the following commands:   
       unzip libpng-1.6.39.tar.gz  
       cd lpng1639  
       cp scripts/pnglibconf.h.prebuilt pnglibconf.h  
       cp scripts/makefile.linux makefile  
       make test  
       cmake . -DCMAKE\_INSTALL\_PREFIX=/.  
       make clean  
       make  
       cd ..
   12. Now, we should be able to install GD  
       wget -O libgd <https://github.com/libgd/libgd/archive/refs/heads/master.zip>  
       unzip libgd.zip  
       mv libgd-master libgd  
       cmake . -DCMAKE\_INSTALL\_PREFIX=/.  
       make clean  
       make  
       sudo make install  
       cd ..
   13. The last step is to install GD in Perl.   
       curl -O http://search.cpan.org/CPAN/authors/id/L/LD/LDS/GD-2.46.tar.gz   
       There is a complication here, and I had to take a shortcut to solve it. The file is nto recognized by tar or unzip. This issue shoudolo be investigatged and solved, but for now, I took the quick path: execute the command  
       explorer.exe .  
       This will open a Windows explorer in the current folder, something along the lines of:  
       \\wsl.localhost\Ubuntu\home\[Windows username]\circos\srctemp  
       Use 7zip in Windows to unpack the file, and paste the folder from the 7zip FIle Manager into the Linux folder. Now, continue  
       cd GD-2.46  
       The circos instructions call for   
       perl Makefile.PL  
       But the following message appears  
       Could not find gdlib-config in the search path. Please install libgd 2.0.28 or higher.  
       Enter the following command instead:  
       perl Makefile.PL --ignore\_missing\_gd  
       Accept all default options.   
       export PATH="/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin:/home/bioma/circos/srctemp/libgd-2.2.5/config:"