

# Installation Instructions for PopGen 2.0

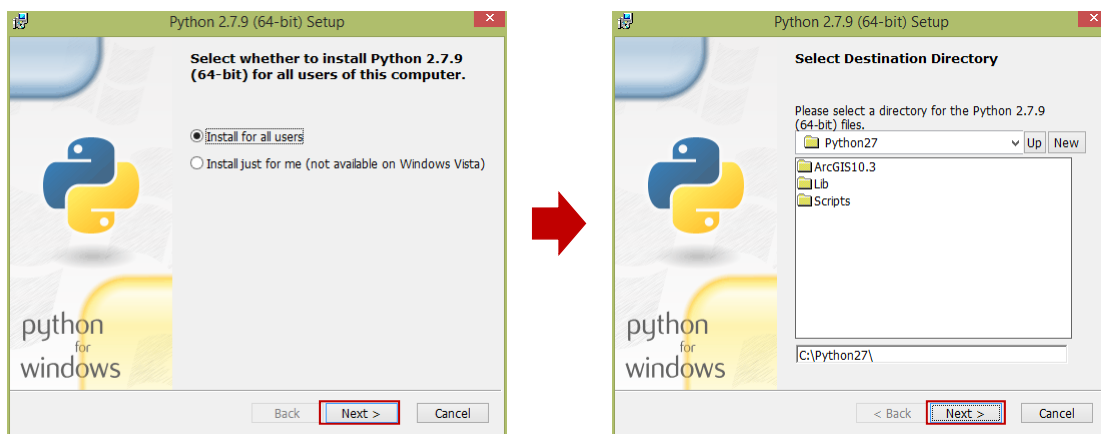
## Downloading PopGen 2.0

- This **Website** includes:
  - popgen-2.0.b2.zip: PopGen Software
  - Folder **Supplementary\_Packages**: All supplements are stored in this folder
  - Folder **User\_Guide**: This folder includes a user's guide for PopGen 2.0
  - Folder **Example\_Input\_Data**: Example input data is located in this folder

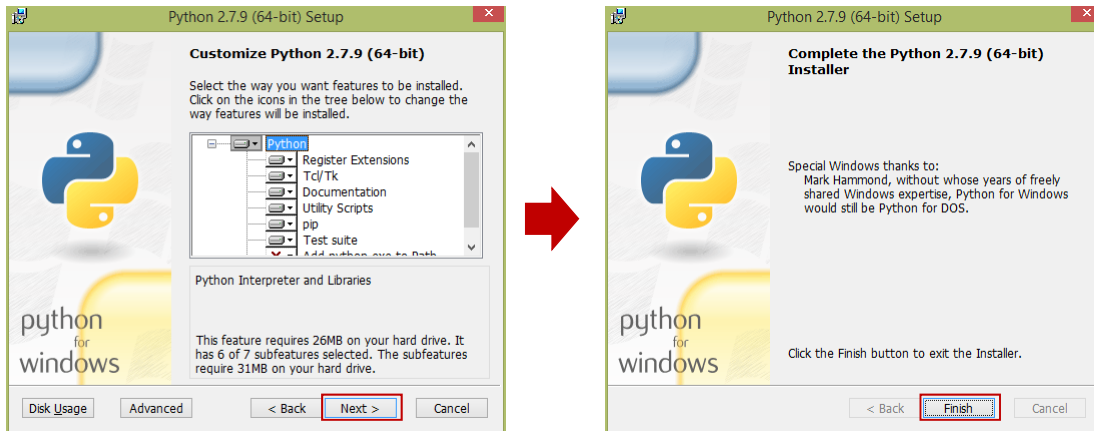
## Installing Python 2.7.9

- To start the installation procedure, double click on the **python-2.7.9.amd64.msi** file in **Supplementary\_Packages** folder. The Python Installation window should open up look similar to this, but may vary based on your operating system.
- Click next to continue installation.
- The default installation path on the next screen should **"C:\Python27\"**.
- Clicking next will take you to the Customize Python 2.7.9 screen, click next again to continue.
- Wait for Python to complete the installation, do not interrupt the installation in any way. After, it is complete press the Finish button to exit the installation.

## Installing Python 2.7.9



## Installing Python 2.7.9



## List of Dependencies Required in PopGen 2.0

- *PyYAML 3.11*
- *NumPy 1.9.2*
- *SciPy 0.16.1*
- *Pandas 0.16.1*
- *Python Dateutil 2.4.2*
- *Pytz 2015.2*

## Installing Dependencies

1. Open the following folders located in **Supplementary\_Packages** folder:
  - numpy-1.9.2+mkl-cp27-none-win amd64
  - scipy-0.16.1-cp27-none-win amd64
  - Pandas-0.16.1-cp27-none-win amd64

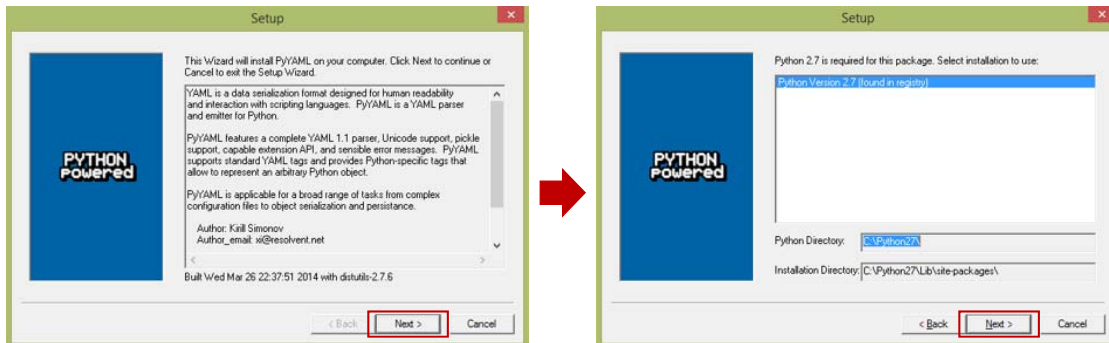
## Installing Dependencies

2. Copy and paste the following sub-folders to **C:/Python27/Lib/site-packages**
  - numpy
  - numpy-1.9.2.data
  - numpy-1.9.2.dist-info
  - scipy
  - scipy-0.16.1.dist-info
  - Pandas
  - Pandas-0.16.1.dist-info

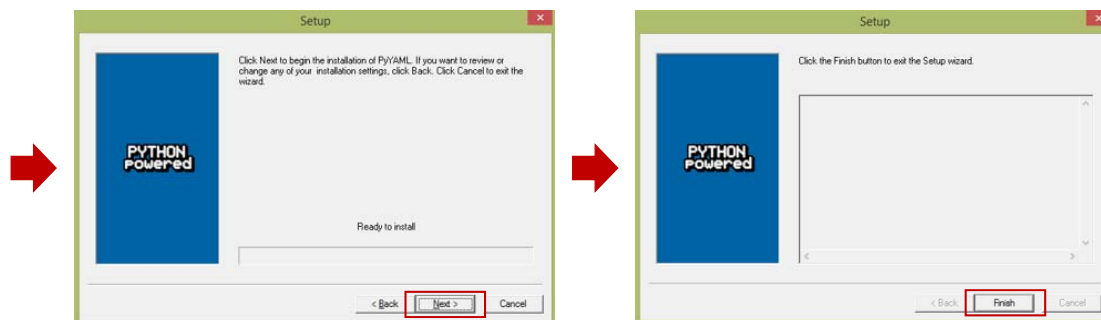
## Installing Dependencies

### 3. Find **PyYAML-3.11.win-amd64-py2.7.exe** in **Supplementary\_Packages** folder

- Double click the executable file and follow the setup wizard to complete the **PyYAML** installation



## Installing Dependencies



## Installing Dependencies

### 4. Install **python-dateutil-2.4.2**

- Open the **command prompt** window
- From the **command prompt** window, change the directory to **python-dateutil-2.4.2**
- Type **python setup.py install** and hit enter

```

C:\Users\ygon11\Downloads\Python_1_0_Installation>cd python-dateutil-2.4.2
C:\Users\ygon11\Downloads\Python_1_0_Installation>python setup.py install
running install
running egg_info
writing python_dateutil.egg-info/PKG-INFO
writing requirements to python_dateutil.egg-info/requirements.txt
writing top-level names to python_dateutil.egg-info/top_level.txt
writing dependency_links to python_dateutil.egg-info/dependency_links.txt
reading manifest file 'python_dateutil.egg-info/SOURCES.txt'
reading manifest template 'MANIFEST.in'
writing manifest file 'python_dateutil.egg-info/SOURCES.txt'
installing library code to build\bdist.win-amd64\egg
running build
creating build\bdist.win-amd64\egg
creating build\bdist.win-amd64\egg\python_dateutil
copying build\lib\dateutil\__init__.py -> build\bdist.win-amd64\egg\python_dateutil
copying build\lib\dateutil\parser.py -> build\bdist.win-amd64\egg\python_dateutil
copying build\lib\dateutil\rrule.py -> build\bdist.win-amd64\egg\python_dateutil
copying build\lib\dateutil\timezone.py -> build\bdist.win-amd64\egg\python_dateutil
copying build\lib\dateutil\zoneinfo\__init__.py -> build\bdist.win-amd64\egg\python_dateutil
copying build\lib\dateutil\zoneinfo\zoneinfo.py -> build\bdist.win-amd64\egg\python_dateutil
copying build\lib\dateutil\zoneinfo\zoneinfo__init__.py -> build\bdist.win-amd64\egg\python_dateutil
copying build\lib\dateutil\zoneinfo\zoneinfo__init__.py to _init_.pyc
python27\lib\site-packages\python_dateutil-2.4.2-py2.7.egg
copying python_dateutil.egg-info/dependency_links.txt -> build\bdist.win-amd64\egg\EGG-INFO
copying python_dateutil.egg-info/requirements.txt -> build\bdist.win-amd64\egg\EGG-INFO
copying python_dateutil.egg-info/top_level.txt -> build\bdist.win-amd64\egg\EGG-INFO
creating python_dateutil.egg-info\safe -> build\bdist.win-amd64\egg\EGG-INFO
removing 'build\bdist.win-amd64\egg' (and everything under it)
moving c:\python27\lib\site-packages\python_dateutil-2.4.2-py2.7.egg
copying python_dateutil-2.4.2-py2.7.egg to c:\python27\lib\site-packages
python-dateutil-2.4.2 is already the active version in easy-install.pth
Installing c:\python27\lib\site-packages\python_dateutil-2.4.2-py2.7.egg
Processing dependencies for python-dateutil-2.4.2
Best match: six 1.9.0
Processing six-1.9.0-py2.7.egg
six 1.9.0 is already the active version in easy-install.pth
Using c:\python27\lib\site-packages\six-1.9.0-py2.7.egg
Finished processing dependencies for python-dateutil-2.4.2

C:\Users\ygon11\Downloads\Python_1_0_Installation>python setup.py install
  
```

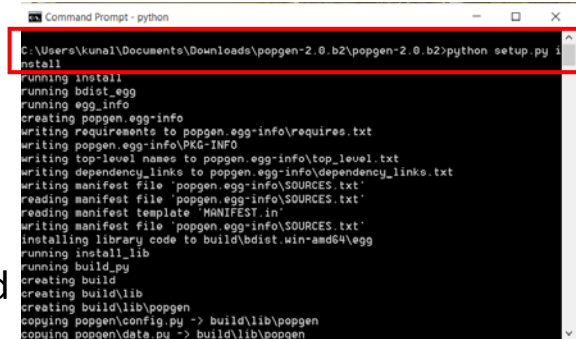
## Installing Dependencies

### 5. Install **pytz-2015.2** (same as step 4)

- Open the **command prompt** window
- From the **command prompt** window, change the directory to **pytz-2015.2**
- Type **python setup.py install** and hit enter

## Setting PopGen 2.0 Software

- Open the **command prompt** window
- From the **command prompt** window, change the directory to **popgen-2.0.b2**
- Type **python setup.py install** and hit enter

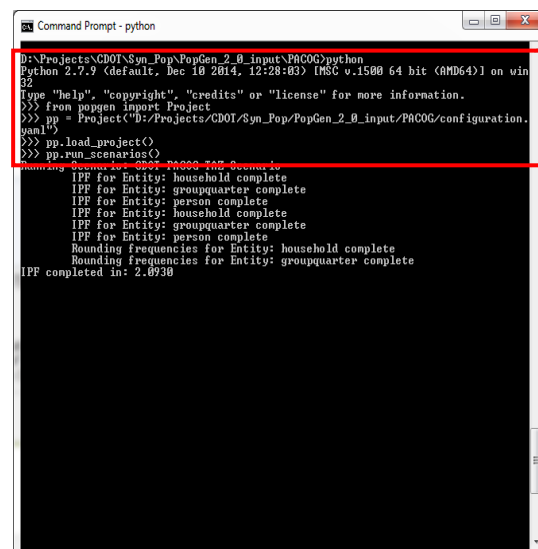


```

C:\Users\kunal\Documents\Downloads\popgen-2.0.b2\popgen-2.0.b2>python setup.py i
install
running install
running bdist_egg
running egg_info
creating popgen.egg-info
writing requirements to popgen.egg-info\requires.txt
writing popgen.egg-info\PKG-INFO
writing top-level names to popgen.egg-info\top_level.txt
writing dependency_links to popgen.egg-info\dependency_links.txt
writing manifest file 'popgen.egg-info\SOURCES.txt'
reading manifest file 'popgen.egg-info\SOURCES.txt'
reading manifest template 'MANIFEST.in'
writing manifest file 'popgen.egg-info\SOURCES.txt'
installing library code to build\bdist.win-amd64\egg
running install_lib
running build_py
creating build
creating build\lib
creating build\lib\popgen
copying popgen\config.py -> build\lib\popgen
copying popgen\data.py -> build\lib\popgen
  
```

## Starting PopGen 2.0

- Open the **command prompt** window
- Type **python** and Enter from the **command prompt**
- Type **from popgen import Project** and Enter
- Type **pp = Project("C:\configuration.yaml")** and Enter
- Type **pp.load\_project()** and Enter
- Type **pp.run\_scenarios()** and Enter



```

D:\Projects\CDOI\Syn_Pop\PopGen_2_0_input\PACOG>python
Python 2.7.9 (default, Dec 10 2014, 12:28:03) [MSC v.1500 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>> from popgen import Project
>>> pp = Project('D:/Projects/CDOI/Syn_Pop/PopGen_2_0_input/PACOG/configuration.
yaml')
>>> pp.load_project()
>>> pp.run_scenarios()
Running Scenario: CDOI-PACOG-102 Scenario
IPF for Entity: household complete
IPF for Entity: groupquarter complete
IPF for Entity: person complete
IPF for Entity: household complete
IPF for Entity: groupquarter complete
IPF for Entity: person complete
Rounding frequencies for Entity: household complete
Rounding frequencies for Entity: groupquarter complete
IPF completed in: 2.0930
  
```