

ResBaz Perth 2019 Installation

Instructions

- Please note you are **required to bring a laptop computer** with administrator privileges (not a tablet, Chromebook, etc). Please also bring your charger!
- Please ensure you are using an up to date web browser
- Set up eduroam on your laptop for wifi access (guest accounts will be supplied for those not able to access eduroam)
- If you have any questions regarding installation, come and see us at the Registration/Help Desk on Level 1, or ask any ResBaz helpers (look out for stars on name badges)

Links to Stream Installation Instructions:

[Git and Bash Stream Software Requirements](#)

[The Bash Shell](#)

[Windows](#)

[MacOS](#)

[Linux](#)

[Git](#)

[Windows](#)

[MacOS](#)

[Linux](#)

[Text Editor](#)

[Windows](#)

[MacOS](#)

[Python Stream Software Requirements](#)

[Windows](#)

[MacOS](#)

[Linux](#)

[R Stream Software Requirements](#)

[Windows](#)

[MacOS](#)

[Linux](#)

[Structured Data Stream Software Requirements](#)

[Humanities and Arts Stream Software Requirements](#)

[Windows](#)

[MacOS](#)

[Linux](#)

[Docker Stream Software Requirements](#)

[Windows](#)

[MacOS](#)

[Linux](#)

[Genomics Stream Software Requirements](#)

[SSH client](#)

[Windows](#)

[MacOS and Linux](#)

[Machine Learning Stream Software Requirements](#)

[Revision history](#)

[Requirements](#)

[Installation](#)

Git and Bash Stream Software Requirements

The Bash Shell

Windows

1. Download the Git for Windows [installer](#).
2. Run the installer and follow the steps below:
 1. Click on "Next" four times (two times if you've previously installed Git). You don't need to change anything in the Information, location, components, and start menu screens.
 2. **Select "Use the nano editor by default" and click on "Next".**
 3. Keep "Use Git from the Windows Command Prompt" selected and click on "Next". If you forgot to do this programs that you need for the workshop will not work properly. If this happens rerun the installer and select the appropriate option.
 4. Click on "Next".
 5. Keep "Checkout Windows-style, commit Unix-style line endings" selected and click on "Next".
 6. **Select "Use Windows' default console window" and click on "Next".**
 7. Click on "Install".
 8. Click on "Finish".
3. If your "HOME" environment variable is not set (or you don't know what this is):
 1. Open command prompt (Open Start Menu then type `cmd` and press [Enter])
 2. Type the following line into the command prompt window exactly as shown:
`setx HOME "%USERPROFILE%"`
 3. Press [Enter], you should see `SUCCESS: Specified value was saved.`
 4. Quit command prompt by typing `exit` then pressing [Enter]

This will provide you with both Git and Bash in the Git Bash program.

MacOS

The default shell in all versions of macOS is Bash, so no need to install anything. You access Bash from the Terminal (found in `/Applications/Utilities`). See the Git installation [video tutorial](#) for an example on how to open the Terminal.

You may want to keep Terminal in your dock for this workshop.

Linux

The default shell is usually Bash, but if your machine is set up differently you can run it by opening a terminal and typing `bash`. There is no need to install anything.

Git

Git is a version control system that lets you track who made changes to what when and has options for easily updating a shared or public version of your code on github.com. You will need a [supported web browser](#).

You will need an account at github.com for parts of the Git lesson. Basic GitHub accounts are free. We encourage you to create a GitHub account if you don't have one already. Please consider what personal information you'd like to reveal. For example, you may want to review these [instructions for keeping your email address private](#) provided at GitHub.

Windows

Git should be installed on your computer as part of your Bash install (described above).

MacOS

For OS X 10.9 and higher, install Git for Mac by downloading and running the most recent "mavericks" installer from [this list](#). Because this installer is not signed by the developer, you may have to right click (control click) on the .pkg file, click Open, and click Open on the pop up window. After installing Git, there will not be anything in your `/Applications` folder, as Git is a command line program.

For older versions of OS X (10.5-10.8) use the most recent available installer labelled "snow-leopard" [available here](#).

Linux

If Git is not already available on your machine you can try to install it via your distro's package manager. For Debian/Ubuntu run `sudo apt-get install git` and for Fedora run `sudo dnf install git`.

Text Editor

When you're writing code, it's nice to have a text editor that is optimized for writing code, with features like automatic color-coding of key words. The default text editor on macOS and Linux is

usually set to Vim, which is not famous for being intuitive. If you accidentally find yourself stuck in it, hit the `Esc` key, followed by `:+Q+!` (colon, lower-case 'q', exclamation mark), then hitting `Return` to return to the shell.

Windows

nano is a basic editor and the default that instructors use in the workshop. It is installed along with Git. Others editors that you can use are [Notepad++](#) or [Sublime Text](#). **Be aware that you must add its installation directory to your system path.**

Please ask your instructor to help you do this.

MacOS

nano is a basic editor and the default that instructors use in the workshop. See the Git installation [video tutorial](#) for an example on how to open nano. It should be pre-installed. Others editors that you can use are [BBEdit](#) or [Sublime Text](#).

nano is a basic editor and the default that instructors use in the workshop. It should be pre-installed.

Others editors that you can use are [Gedit](#), [Kate](#) or [Sublime Text](#).

Python Stream Software Requirements

[Python](#) is a popular language for research computing, and great for general-purpose programming as well. Installing all of its research packages individually can be a bit difficult, so we recommend [Anaconda](#), an all-in-one installer.

Regardless of how you choose to install it, **please make sure you install Python version 3.x** (e.g., 3.6 is fine).

We will teach Python using the [Jupyter notebook](#), a programming environment that runs in a web browser. For this to work you will need a reasonably up-to-date browser. The current versions of the Chrome, Safari and Firefox browsers are all [supported](#) (some older browsers, including Internet Explorer version 9 and below, are not).

Head over to the python stream git repository for additional information on what we'll be doing on the day, and to get the notebooks and data we'll be using (<https://github.com/darcyabjones/resbaz-perth-2019-python>).

Windows

1. Open <https://www.anaconda.com/download/#windows> with your web browser.
2. Download the Python 3 installer for Windows.
3. Install Python 3 using all of the defaults for installation *except* make sure to check **Add Anaconda to my PATH environment variable**.

MacOS

1. Open <https://www.anaconda.com/download/#macos> with your web browser.
2. Download the Python 3 installer for OS X.
3. Install Python 3 using all of the defaults for installation.

Linux

1. Open <https://www.anaconda.com/download/#linux> with your web browser.
2. Download the Python 3 installer for Linux.
(The installation requires using the shell. If you aren't comfortable doing the installation yourself stop here and request help at the workshop.)
3. Open a terminal window.
4. Type
`bash Anaconda3-`
and then press **Tab**. The name of the file you just downloaded should appear. If it does not, navigate to the folder where you downloaded the file, for example with:

```
cd Downloads
```

Then, try again.

5. Press **Return**. You will follow the text-only prompts. To move through the text, press **Spacebar**. Type **yes** and press enter to approve the license. Press enter to approve the default location for the files. Type **yes** and press enter to prepend Anaconda to your **PATH** (this makes the Anaconda distribution the default Python).
6. Close the terminal window.

R Stream Software Requirements

[R](#) is a programming language that is especially powerful for data exploration, visualization, and statistical analysis. To interact with R, we use [RStudio](#).

Windows

Install R by downloading and running [this .exe file](#) from [CRAN](#). Also, please install the [RStudio IDE](#). Note that if you have separate user and admin accounts, you should run the installers as administrator (right-click on .exe file and select "Run as administrator" instead of double-clicking). Otherwise problems may occur later, for example when installing R packages.

MacOS

Install R by downloading and running [this .pkg file](#) from [CRAN](#). Also, please install the [RStudio IDE](#).

Linux

You can download the binary files for your distribution from [CRAN](#). Or you can use your package manager (e.g. for Debian/Ubuntu run `sudo apt-get install r-base` and for Fedora run `sudo dnf install R`). Also, please install the [RStudio IDE](#).

Structured Data Stream Software Requirements

This lesson requires a working copy of spreadsheet software, such as Microsoft Excel or LibreOffice or OpenOffice.org (see more details in [Setup](#)).

Humanities and Arts Stream Software Requirements

For this stream you will need **OpenRefine** (formerly Google Refine), **jupyter notebook** and a web browser.

The [Jupyter notebook](#) is a programming environment that runs in a web browser. For this to work you will need a reasonably up-to-date browser. The current versions of the Chrome, Safari and Firefox browsers are all [supported](#) (some older browsers, including Internet Explorer version 9 and below, are not). We recommend using [Anaconda](#), an all-in-one installer to install Jupyter notebook.

See the installation instructions for [Python Stream Software Requirements](#) to install Anaconda.

OpenRefine is a Java program that runs on your machine (not in the cloud). It runs inside your browser, but no web connection is needed.

Windows

- If you have Internet Explorer (or Edge) set as your default web browser, check that you have Firefox or Chrome installed and set either of them as your default browser. OpenRefine runs in your default browser, but will not run correctly in Internet Explorer.
- Learn how to set your browser as default by clicking on this link for [Google Chrome](#) and this link for [Firefox](#).
- Download software from [OpenRefine](#)
- Select the most recent version of OpenRefine (do not select beta versions or the release candidates). The version that you should download will be at the top of the page and named OpenRefine 3.1 for example.
- Unzip the downloaded file into a directory by right-clicking and selecting “Extract...”. Name that directory something like OpenRefine.
- Go to your newly created OpenRefine directory.
- Launch OpenRefine by clicking on openrefine.exe (this will launch a command prompt window first; ignore that, and wait for OpenRefine to launch in the web browser, which is where you will interact with the program).
- If you are using a different browser, or OpenRefine does not automatically open for you, point your browser at <http://127.0.0.1:3333/> or <http://localhost:3333> to launch the program.

MacOS

- Check that you have Firefox or Chrome browsers installed and set as your default browser. OpenRefine runs in your default browser. It will not run correctly in Internet Explorer.

- Download software from [OpenRefine](#)
- Select the most recent version of OpenRefine (do not select beta versions or the release candidates). The version that you should download will be at the top of the page and named OpenRefine 3.1 for example.
- Unzip the downloaded file into a directory by double-clicking it. Name that directory something like OpenRefine.
- Go to your newly created OpenRefine directory.
- Launch OpenRefine
- Drag icon into Applications folder, and Ctrl-click/Open... it.
- If you are using a different browser, or OpenRefine does not automatically open for you, point your browser at <http://127.0.0.1:3333/> or <http://localhost:3333> to launch the program.

Linux

- Check that you have Firefox or Chrome browsers installed and set as your default browser. OpenRefine runs in your default browser. It will not run correctly in Internet Explorer.
- Download software from [OpenRefine](#)
- Select the most recent version of OpenRefine (do not select beta versions or the release candidates). The version that you should download will be at the top of the page and named OpenRefine 3.1 for example.
- Unzip the downloaded file into a directory. Name that directory something like OpenRefine.
- Go to your newly created OpenRefine directory.
- Launch OpenRefine
- Type `./refine` into the terminal within the OpenRefine directory

If you are using a different browser, or OpenRefine does not automatically open for you, point your browser at <http://127.0.0.1:3333/> or <http://localhost:3333> to launch the program.

Docker Stream Software Requirements

Participants in the Docker stream should have basic knowledge of Unix and bash (i.e. using commands such as `cd`, `pwd`, `ls`, `mv`, `mkdir`, `rm`).

An SSH client will be required as we will be connecting to virtual machines on Nimbus at the Pawsey Supercomputing Centre. Installation instructions are below.

Windows

Windows users will need to install an SSH client such as [MobXTerm](#) or [PuTTY](#). If you do not have administrator privileges on your Windows laptop, you can download the portable/stand alone versions which do not require installation.

MacOS

MacOS has the `ssh` command installed by default. It can be accessed via the Terminal application.

Linux

Linux distributions will have the `ssh` command installed by default. It can be accessed via the terminal.

Genomics Stream Software Requirements

SSH client

Windows

Please install [MobaXTerm](#) or [Putty](#)

MacOS and Linux

Mac and Linux Users can use the Terminal app- for Mac, the Terminal is found in [/Applications/Utilities](#)

Machine Learning Stream Software Requirements

Requirements

For this workshop, you are required to **bring a computer (not a tablet, Chromebook, etc.) with a working version of Python** installed. It is recommended that you have administrative privileges on your computer — the lessons require specific software packages installed.

We will be using **Scikit Learn** and **fast.ai** for this workshop, please make sure you have installed [Anaconda Distribution](#), and [Jupyter Notebook](#) (a programming environment that runs in a web browser) and also other relevant packages listed as below:

Prerequisites:


- Pytorch \geq v1.0
- fast.ai v1 for PyTorch <https://github.com/fastai/fastai>
- A dataset from Kaggle competition. You can download all of them [here](#).
- **Signup and apply for access Google Colab** <https://colab.research.google.com/> – anyone with a Google Drive account can sign up for Colab by heading to colab website.

Please ensure you have a reasonably up-to-date Web browser. The current versions of Chrome, Safari and Firefox are all [supported](#), but some older browsers, including Internet Explorer version 9 are not.

Windows


1. Install Anaconda Python 3.x distribution <https://www.anaconda.com/distribution/>
2. Make sure you have defined the PATH system variable.

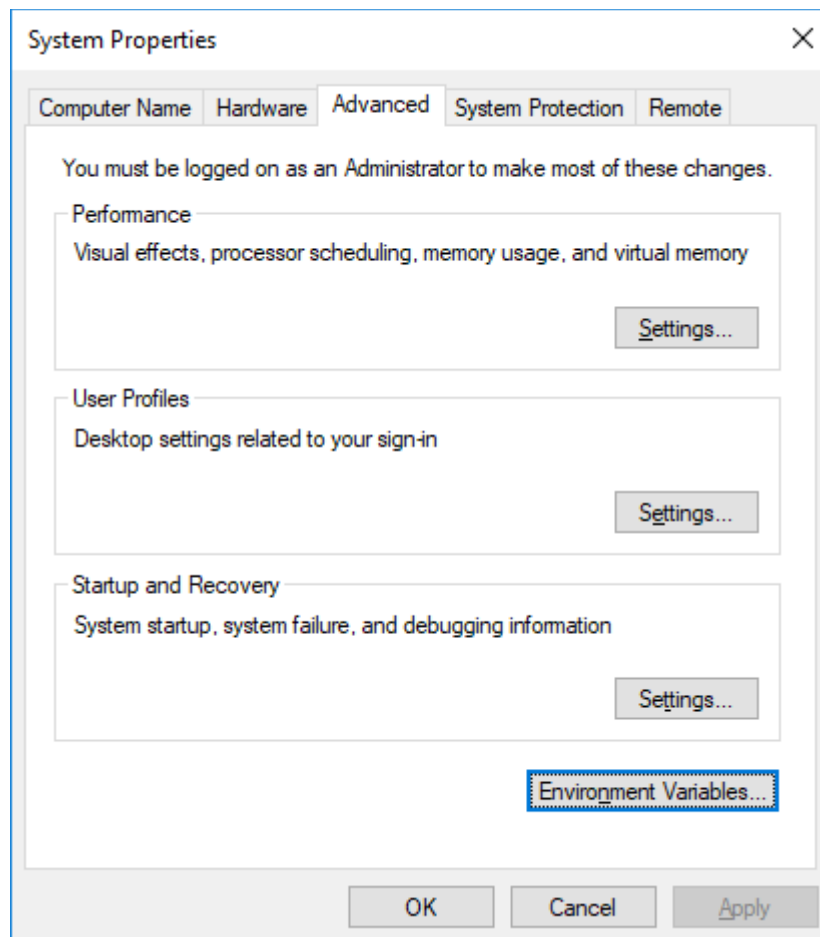
Step 1: Find your current Python and Conda path:

- Open Windows command prompt (Press  and type `cmd`)
- Type `where python` and `where conda`

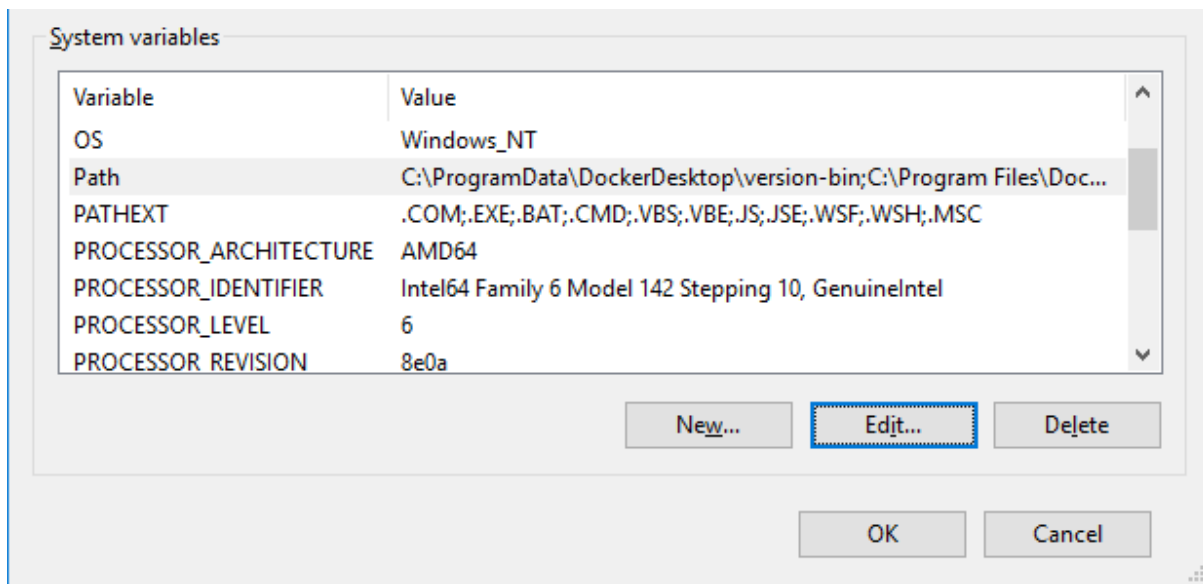
Example: **Note! your path may differ from the instruction**

```
C:\>where python
C:\Users\ResBaz\AppData\Local\Continuum\anaconda3\python.exe
C:\>where conda
C:\Users\ResBaz\AppData\Local\Continuum\anaconda3\condabin\conda.bat
C:\Users\ResBaz\AppData\Local\Continuum\anaconda3\Library\bin\conda.bat
C:\Users\ResBaz\AppData\Local\Continuum\anaconda3\Scripts\conda.exe
```

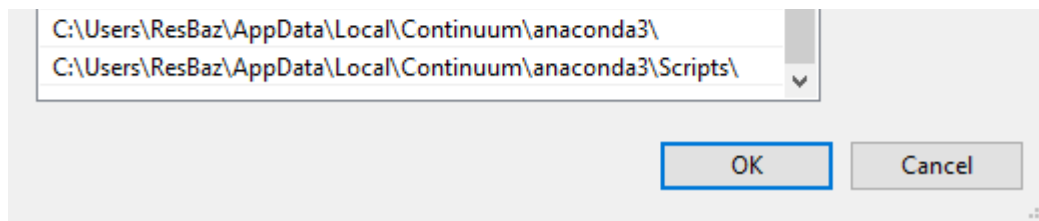
Step 2: Press  + R together to get a command prompt. Type `sysdm.cpl` → go to **Advanced** tab → select **Environment Variables**



In the System variables section, choose Path → select **Edit** → click **New**



Add the path → press **OK**



Step 3: using command line to verify Python has been successfully installed. Open terminal console (i.e. Anaconda Prompt, Git Bash, Cmdr, etc.) → type `python` then press Enter

Example:

```

Anaconda Prompt - python
(base) C:\Users\stippay>python
Python 3.7.3 (default, Mar 27 2019, 17:13:21) [MSC v.1915 64 bit (AMD64)]
:: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>>

```

Type `exit()` then Enter or Ctrl + D to exit.

- Download a dataset from Kaggle competition to your working directory. You can download all of them [here](#).

in terminal, create a directory called 'rossmann'


```
$ mkdir rossmann
# Change to 'rossmann' directory:
$ cd rossmann
# Run the following command to untar rossmann.tgz file
$ tar xvzf rossmann.tgz
```

3. Create a virtual environment called **resbazml**

Note: By creating the virtual environment via Anaconda distribution, you will be working on an isolated working copy of Python with specific versions of libraries or Python itself without affecting the base Anaconda or other projects.

- Make sure the conda will be the latest version (conda install will also do the update)

```
$ conda install conda
```

- Create a virtual environment from YAML file which you can download from [here](#)

Note: download YAML file and place to any directory (i.e. /Desktop). Browse to the folder that YAML file exists before running the following command.

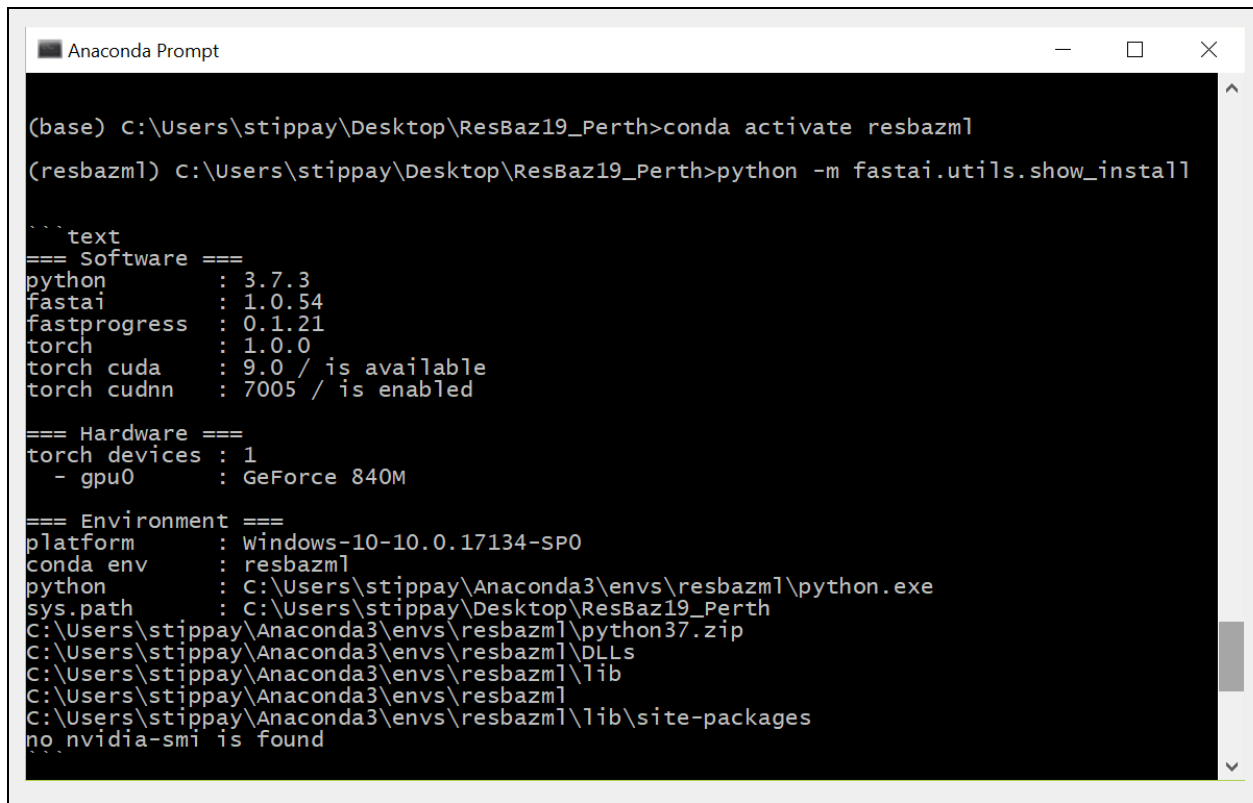
```
$ conda env create -f resbazml_environment.yml
```

- Close your Anaconda Prompt terminal (or others) and relaunch it
- Activate the environment

```
$ conda activate resbazml
```

4. Check the fastai installation and Pytorch version

```
$ python -m fastai.utils.show_install
```



```
(base) C:\Users\stippay\Desktop\ResBaz19_Perth>conda activate resbazml
(resbazml) C:\Users\stippay\Desktop\ResBaz19_Perth>python -m fastai.utils.show_install

...
text
=== Software ===
python      : 3.7.3
fastai      : 1.0.54
fastprogress: 0.1.21
torch       : 1.0.0
torch cuda  : 9.0 / is available
torch cudnn : 7005 / is enabled

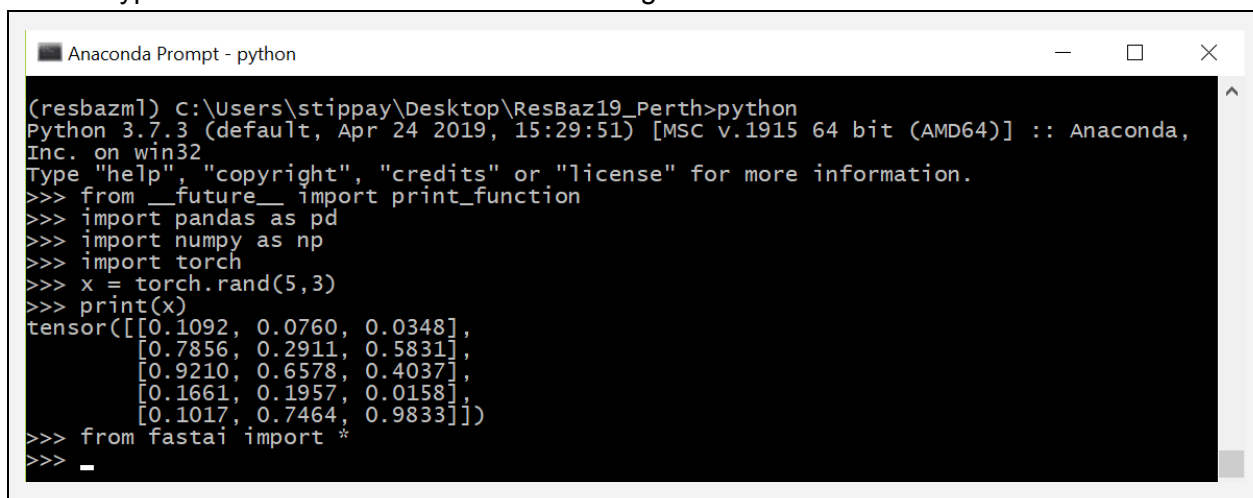
=== Hardware ===
torch devices : 1
- gpu0        : GeForce 840M

=== Environment ===
platform     : Windows-10-10.0.17134-SP0
conda env    : resbazml
python       : C:\Users\stippay\Anaconda3\envs\resbazml\python.exe
sys.path     : C:\Users\stippay\Desktop\ResBaz19_Perth
               C:\Users\stippay\Anaconda3\envs\resbazml\python37.zip
               C:\Users\stippay\Anaconda3\envs\resbazml\DLLs
               C:\Users\stippay\Anaconda3\envs\resbazml\lib
               C:\Users\stippay\Anaconda3\envs\resbazml
               C:\Users\stippay\Anaconda3\envs\resbazml\lib\site-packages
no nvidia-smi is found
```

If you can see the details as example above, your fastai installation should be working fine.

5. Fastai and Pytorch packages verification

- Type the command line as shown in the figure below for the verification



```
(resbazml) C:\Users\stippay\Desktop\ResBaz19_Perth>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] :: Anaconda,
Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from __future__ import print_function
>>> import pandas as pd
>>> import numpy as np
>>> import torch
>>> x = torch.rand(5,3)
>>> print(x)
tensor([[0.1092, 0.0760, 0.0348],
        [0.7856, 0.2911, 0.5831],
        [0.9210, 0.6578, 0.4037],
        [0.1661, 0.1957, 0.0158],
        [0.1017, 0.7464, 0.9833]])
>>> from fastai import *
>>> _
```

- To check if your CUDA is available or not

```
Anaconda Prompt - python
(resbazml) C:\Users\stippay\Desktop\ResBaz19_Perth>python
Python 3.7.3 (default, Apr 24 2019, 15:29:51) [MSC v.1915 64 bit (AMD64)] :: Anaconda, Inc. on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> torch.cuda.is_available()
True
>>>
```

If the result is True, your CUDA is available for use. Otherwise, it will be False.

MacOS and Linux

1. Install the Anaconda Python 3.x distribution (recommended if you're new to conda):
<https://www.anaconda.com/distribution/>
or the Miniconda Python 3.x distribution:
<https://docs.conda.io/en/latest/miniconda.html>
2. Load (or reload) the terminal. Make sure you have defined the PATH system variable and have conda activated by testing the command conda-env list:

```
(base) [cara@Caras-Alien Downloads]$ conda-env list
# conda environments:
#
base * /home/cara/miniconda3
```

If you don't get an output similar to the one above, adjust the following commands to reflect your installation path and either run them interactively or add them to your .bashrc:

```
$ export PATH="$HOME/miniconda3/bin:$PATH"
$ source $HOME/miniconda3/bin/activate
```

3. Create a directory to download the dataset and course content into:

```
$ mkdir resbazml
$ cd resbazml
$ mkdir rossmann
$ cd rossmann
$ wget http://files.fast.ai/part2/lesson14/rossmann.tgz
$ tar xvzf rossmann.tgz
```
4. Create and activate a virtual environment called **resbazml**

```
$ conda create --name resbazml -c conda-forge -c fastai
python=3.7 pytorch fastai jupyter nb_conda_kernels
py-xgboost=0.90 pandas-profiling seaborn plotly python-cufflinks
$ conda activate resbazml
$ pip install isoweek
```

5. Check the fastai and Pytorch installations using steps 4 and 5 in the Windows instructions.

Optional Settings

(Optional for Windows) Create a Jupyter Notebook Kernel for the Python Environment so you can switch to a different kernel. This should have happened automatically for Linux and MacOS.

- Create the Jupyter kernel and install ipywidgets

```
$ python -m ipykernel install --user --name resbazml
--display-name "resbazml"
```
- Sometimes there will be 'autopep8' error once you start Jupyter Notebook. So you can reinstall 'autopep8' package using conda

```
$ conda install -c conda-forge autopep8
```

(Optional) Some extensions would be helpful for using the notebook (i.e. code folding, collapsible heading, etc.) The original GitHub repository which contains a source code is [here](#)

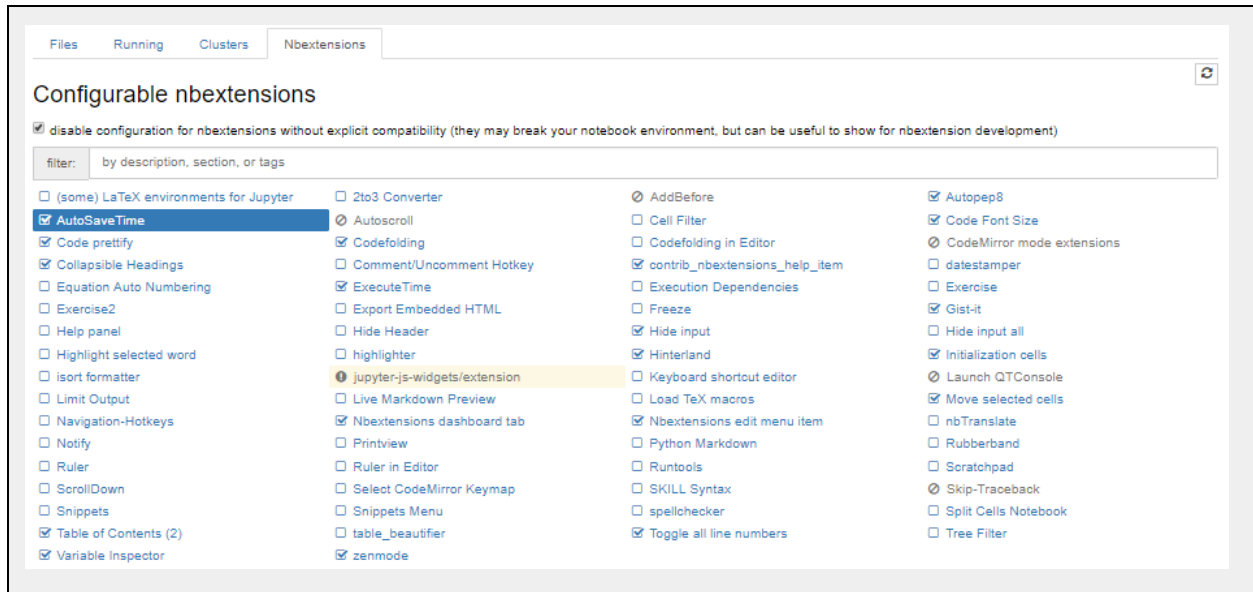
- Install Nbextensions using Conda

```
$ conda install -c conda-forge jupyter_contrib_nbextensions
```
- Install Nbextensions Configurator using Conda

```
$ conda install -c conda-forge jupyter_nbextensions_configurator
```

(Optional) Nbextensions tab will also appear right next to Clusters tab

Example of configuration



(Optional) Increase cell width in a web browser

This solution will not change your default settings. All you need to do is add this following code into any cell of your current notebook and run the cell.

```
from IPython.core.display import display, HTML
display(HTML("<style>.container { width:100% !important; }</style>"))
```