

Developing and deploying Jupyter tools in nanoHUB

In this workshop

- Develop Jupyter notebooks using nanoHUB
- Deploying tools in nanoHUB

Alejandro Strachan

strachan@purdue.edu

School of Materials Engineering & Network for Computational Nanotechnology

Purdue University

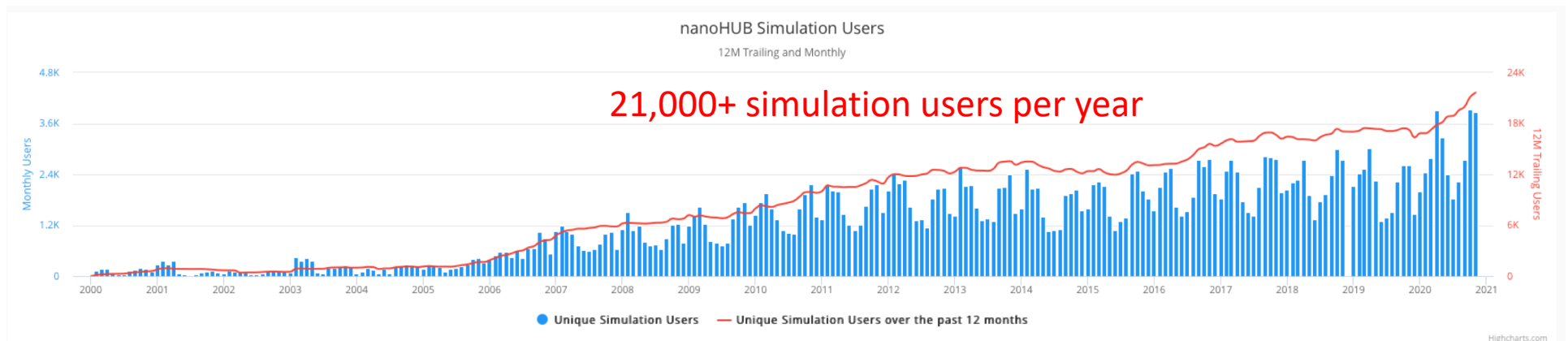
West Lafayette, Indiana USA



PURDUE
UNIVERSITY

Why deploy tools in nanoHUB?

- Tools are publications: indexed by Web of Science and Google Scholar.
- Impact statistics: usage metrics, secondary citations
- Anyone can run your tool without downloading/installing any software
 - Use it in the classroom
 - Document your research and make it reproducible (tools are containerized)
- A full scientific software development environment
 - Libraries for machine learning, statistics, data science, visualization, etc.
- Launch physics-based, research grade codes
 - LAMMPS, Quantum Espresso, NEMO5, abinit, MOOSE, and many more



What is Jupyter?

Popular tool for interactive computing

The standard deviation can be easily calculated:

$$\sigma^2 = \int_{-\infty}^{\infty} (x - \mu)^2 f(x) dx$$

In our case, $\sigma = (2/3 \times 0.5^3)^{1/2} \sim 0.288$

In the following cells we will study how large a sample we need to obtain accurate estimates of the mean and standard deviation of the population.

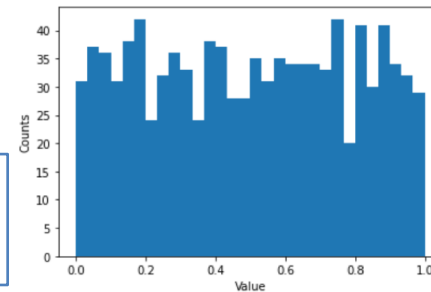
```
In [20]: #Create empty arrays to collect our data
SampleSize = []
SampleMean = []
SampleStd = []
# Loop over increasingly larger sample sizes, compute SAMPLE mean and standard deviation
for size in [25, 50, 75, 100, 150, 500, 1000, 1500, 2000, 5000]:
    sample = np.random.uniform(0,1,size)
    SampleSize = np.append(SampleSize, [size])
    SampleMean = np.append(SampleMean, [np.mean(sample)])
    SampleStd = np.append(SampleStd, [np.std(sample, ddof=1)])
```

Rich text

Live code

```
In [17]: # Let's start by plotting a histogram of the sample values
# The matplotlib command below creates and plots a histogram
plt.hist(sample,30)
plt.ylabel("Counts")
plt.xlabel("Value")
plt.show()
# The histogram you obtain should be relatively consistent with the theoretical distribution
# Fluctuations are expected, if you increase the size of the sample
```

Powerful visualization



Start working with Jupyter on nanoHUB

<https://nanohub.org/tools/jupyter> (you will need a free nanoHUB account to launch Jupyter)

Starts the Jupyter notebook server using the latest installed release of anaconda.

Launch tool → **Launch Tool**

Version 1.7 - published on 27 Jan 2020
doi:10.21981/W6TE-1750 [cite this](#)
This tool is closed source.
[View All Supporting Documents](#)

- 1659 users, detailed usage
- 0 Citation(s)
- 2 questions (Ask a question)
- 0 review(s) (Review this)
- 2 wish(es) (New Wish)

→ Share: [f](#) [t](#) [w](#) ...

1659 users, detailed usage

Inside the Jupyter tool

The screenshot shows the nanoHUB Jupyter interface. At the top, there are logos for nanoHUB and Jupyter, along with buttons for 'Submit a ticket' and 'Terminate Session'. Below the logos are navigation tabs: 'Files', 'Running', 'Formgrader', 'Assignments', and 'Courses'. The main area displays a file directory with a list of folders and files, including '_SugarAidECE202', '_SugarAidME200', 'app-matsimtk', 'app-nmst_dft', 'atomicstructure', 'Aug01-07', 'Aug02-07', 'Aug10-07', 'Aug8', 'bayes', 'bin', 'citrinednn', 'citrinetools', and 'coars'. A dropdown menu is open, showing options for 'Notebook' (Octave 4.2.0, Python (jul262020), Python (tellurium), Python 3, Python2, R) and 'Other' (Text File, Folder, Terminal, noVNC Desktop). Annotations with arrows point to the 'Upload' button, the 'Python 3' option in the dropdown, and the 'Open file utilities' text. A text box at the bottom right provides information about using a Matlab kernel.

Upload files

Launch a notebook running various kernels

Open file utilities

Launch Python 3 notebook

Your home directory in nanoHUB

Want to use Matlab kernel?

- Launch Jupyter with anaconda 5.1
- <https://nanohub.org/tools/jupyter51>

Markdown cells

The screenshot shows the top part of a Jupyter notebook interface. At the top left, there are logos for nanoHUB and Jupyter, followed by the text "Untitled11 Last Checkpoint: 2 minutes ago (autosaved)". Below this is a menu bar with options: File, Edit, View, Insert, Cell, Kernel, Widgets, Help. Underneath the menu bar is a toolbar with icons for saving, adding, undo, redo, copy, paste, up/down arrows, Run, Stop, Refresh, and a dropdown menu currently set to "Markdown". To the right of the dropdown are icons for keyboard shortcuts, Appmode, and a Validate button. The main content area contains a code cell with the following text: `# Title`, `## Subtitle`, `Text`, and `* Bullets`. A blue text annotation "Shift + Enter to execute" is positioned to the right of the code cell.

The screenshot shows the rendered output of the Markdown cell. It displays a large heading "Title", a smaller heading "Subtitle", the word "Text", and a bulleted list with one item "Bullets". Below the rendered content is a text input field with the label "In []:" and an empty text box.

Code cells

Import libraries we will use

- Lots of libraries available
- Submit a Ticket if we don't have what you need

The screenshot shows a Jupyter Notebook interface with the following content:

```
In [1]: # Import various libraries we will use
# Numpy: https://numpy.org/
import numpy as np

#SciPy: https://www.scipy.org/
import scipy.stats as stats

#Matplotlib for plotting: https://matplotlib.org/
import matplotlib.pyplot as plt

In [2]: # First we create an array, called sample, with 1000 random numbers drawn from a uniform distribution from 0 to 1
# This means that all real numbers in the [0:1) interval have equal probability
sample = np.random.uniform(0,1,1000)
```

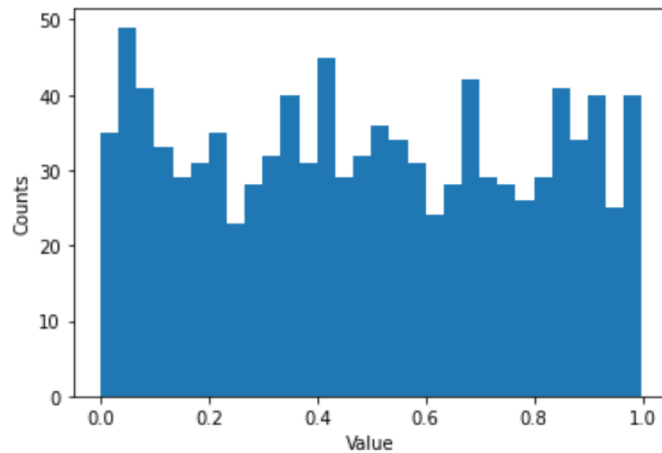
Shift + Enter to execute

Generate an array of random numbers

Plotting

Now let's make a plot

```
In [3]: # Let's start by plotting a histogram of the sample we just obtained.  
#  
# The matplotlib command below creates and plots a histogram with our sample.  
# The input is the number of bins in the histogram  
plt.hist(sample,30)  
plt.ylabel("Counts")  
plt.xlabel("Value")  
plt.show()
```



Shift + Enter to execute

Math in markdown

The *population* average, μ , and standard deviation, σ , are defined as:
`$$ \mu = \frac{\sum_{i=1}^N x_i}{N} $$`
(This is the same as that of the sample, but the sum runs over all members of the population.)
`$$ \sigma^2 = \frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N} $$`
Note the N in the denominator.

Shift + Enter to
execute

The *population* average, μ , and standard deviation, σ , are defined as:

$$\mu = \frac{\sum_{i=1}^N x_i}{N}$$

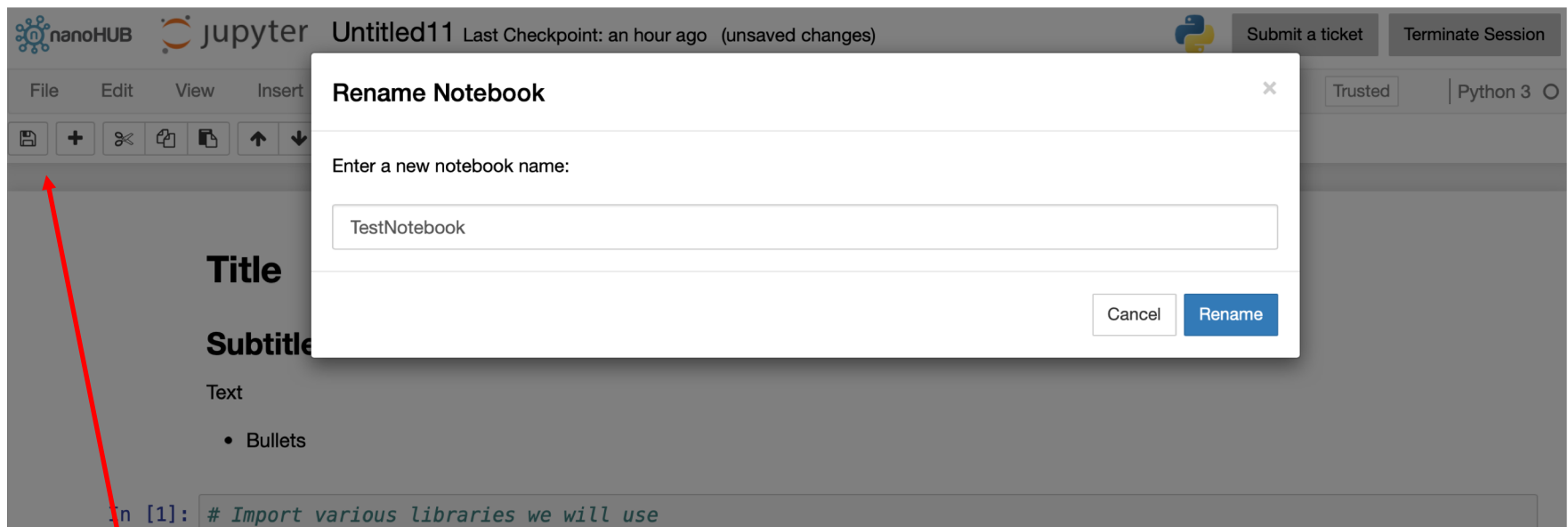
(This is the same as that of the sample, but the sum runs over all members of the population.)

$$\sigma^2 = \frac{\sum_{i=1}^N (x_i - \bar{x})^2}{N}$$

Note the N in the denominator.

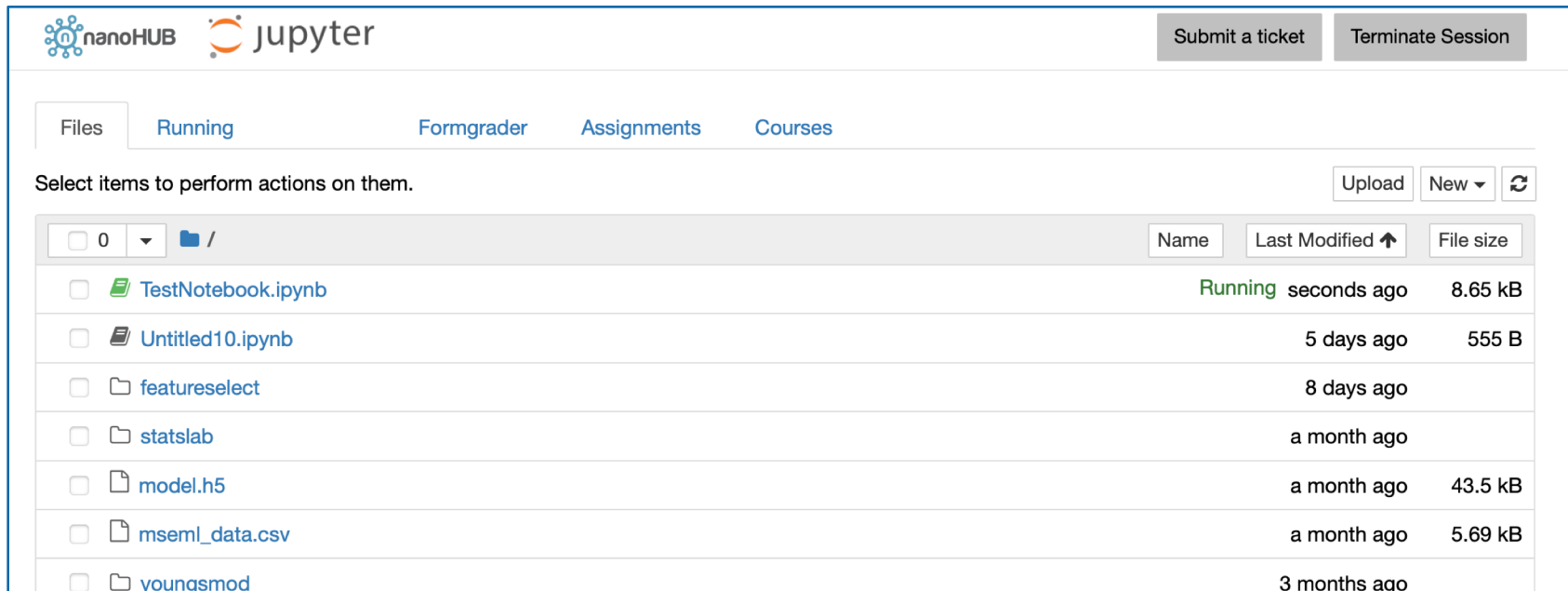
Saving your notebook before publishing

Click on the name to rename file



Save the file

Check your home directory



The screenshot shows the nanoHUB Jupyter interface. At the top, there are logos for nanoHUB and Jupyter, along with buttons for "Submit a ticket" and "Terminate Session". Below the logos, there are navigation tabs: "Files", "Running", "Formgrader", "Assignments", and "Courses". The "Files" tab is active. A message says "Select items to perform actions on them." with buttons for "Upload", "New", and a refresh icon. Below this is a table of files and folders in the home directory. The table has columns for "Name", "Last Modified", and "File size".

	Name	Last Modified	File size
<input type="checkbox"/>	/		
<input type="checkbox"/>	TestNotebook.ipynb	Running seconds ago	8.65 kB
<input type="checkbox"/>	Untitled10.ipynb	5 days ago	555 B
<input type="checkbox"/>	featureselect	8 days ago	
<input type="checkbox"/>	statslab	a month ago	
<input type="checkbox"/>	model.h5	a month ago	43.5 kB
<input type="checkbox"/>	msem1_data.csv	a month ago	5.69 kB
<input type="checkbox"/>	younasm0d	3 months ago	

Deploying a tool in nanoHUB

Tools are publications

Machine Learning for Materials Science: Part 1

By Juan Carlos Verduzco Gastelum¹, Alejandro Strachan¹, Saaketh Desai¹
1. *Purdue University*

Machine learning and data science tools applied to materials science

[Launch Tool](#)

Version 1.3 - published on 01 Apr 2020
doi:10.21981/WGQC-3249 [cite this](#)
[Open source: license](#) | [download](#)
[View All Supporting Documents](#)

1783 users, detailed usage
0 Citation(s)
1 question ([Ask a question](#))
2 review(s) ([Review this](#))
0 wish(es) ([New Wish](#))
Share: [f](#) [t](#) [s](#) ...

With DOIs

Cite this work

Researchers should cite this work as follows:

Juan Carlos Verduzco Gastelum, Alejandro Strachan, Saaketh Desai (2020), "Machine Learning for Materials Science: Part 1," <https://nanohub.org/resources/mseml>. (DOI: 10.21981/WGQC-3249).

[BibTex](#) [EndNote](#)

Indexed by Web of Science and Google Scholar

Google Scholar

strachan Machine Learning for Materials Science

Articles

Any time
Since 2020
Since 2019
Since 2016
Custom range...

Sort by relevance
Sort by date

Machine Learning for Materials Science: Part 1
JCV Gastelum, A Strachan, S Desai - 2019 - nanohub.org
Data science and machine learning are playing increasingly important role in science and engineering and materials science and engineering is not an exception. This online tool provides examples of the use of these tools in the field of materials science using Jupyter notebooks. The notebooks contain step by step explanations of the activities and live code, that can be modified by the users for hands-on learning. The initial set of tutorials focus on: i) data query, organization and visualization, ii) developing a simple model using linear ...

☆ [🔗](#) All 2 versions [🔗](#)

Publish a tool

<https://nanohub.org/tools/create>

Tools: Create New Tool

ABOUT YOUR TOOL:

Tool Name: **REQUIRED**

Short name, used for the directory containing this tool. Example: qdot

Title: **REQUIRED**

Full name for this tool. Example: Quantum Dot Lab

Version:

1.0

Optional version number for this release of the tool. Example: 1.0 or 2.1.5b. Spaces not allowed.

At a glance: **REQUIRED**

A one-line description of your tool. Example: Simulate 3-D confined states in simple quantum dot geometries.

REPOSITORY HOST:

- Host subversion repository on HUB
- Host GIT repository on HUB
- Host GIT repository on GitHub

PUBLISHING OPTION:

- Rapture or Linux-GUI based tool
- Jupyter notebook
- SimTool

ACCESS:

Tool Access: **REQUIRED** What should I choose?
Anyone can run tool

Source Code Access: **REQUIRED** What should I choose?
Open source (anyone can access code)

Project Area Access: **REQUIRED** What should I choose?
Open to public

Development team: **REQUIRED**

strachan

nanohub.org logins for people allowed to modify your code. Example: mylogin, fred, barney, wilma

Tool registered by nanoHUB – ready for you to add code

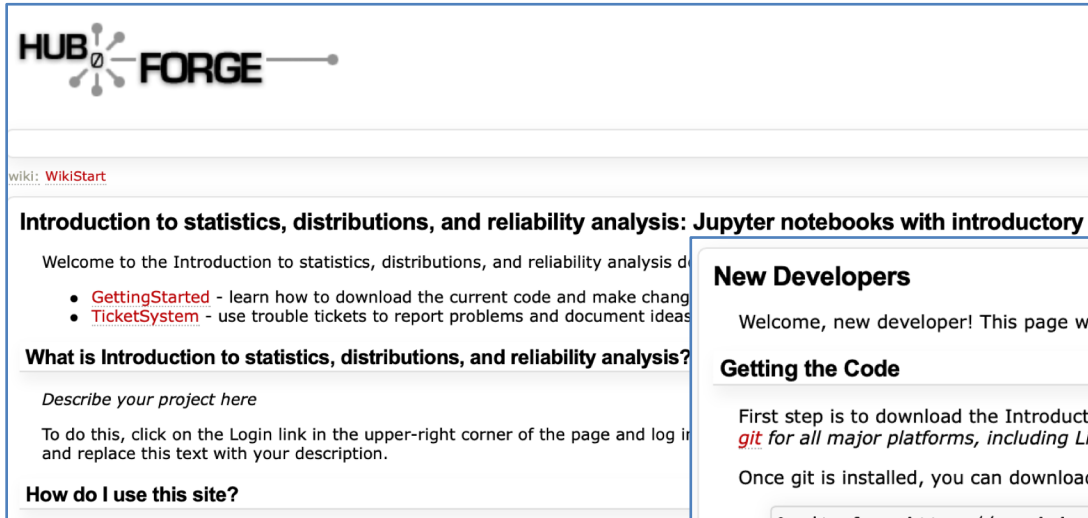
The screenshot shows the nanoHUB website interface. At the top, there is a navigation bar with the nanoHUB logo and various menu items like 'RESOURCES', 'EXPLORE', 'NANOHUB-U', 'PARTNERS', 'COMMUNITY', 'ABOUT', 'SUPPORT', 'DONATE', and 'TAKE A POLL'. On the right side of the navigation bar, it says 'Logged in Help Search'. Below the navigation bar, the main heading is 'Tools: Status for statslab - Created'. There are two buttons: 'All Tools' and 'New Tool'. Below the heading, there is a status filter: 'STATUS: Registered Created Uploaded Installed Approved Published'. The 'Created' status is highlighted. Below the filter, it says 'This tool is one of 674 tools under development on nanoHUB.org.' There is a 'Tool Information' section with an 'edit' link. The tool information table is as follows:

Tool Information edit	
Title	Introduction to statistics, distributions, and reliability analysis (statslab - id #1792)
Version	This version 1.0 (under development)
At a glance	Jupyter notebooks with introductory materials and hands-on exercises
Description	Preview Edit description page
VNC geometry	780x600
Repository Host Type	gitLocal
Tool execution	open to public
Source code	open source
Project area	open to public
Publishing Option	jupyter
Development team	ganeshs, strachan, jverduzc, rapplet, psalek

To the right of the tool information, there is a 'What's next?' section. It says: 'The nanoHUB.org team has created the following project area for your tool on the nanoFORGE: <https://nanohub.org/tools/statslab/wiki>'. Below this, it says 'Follow these steps to start using your project area:' and lists three links: 'Learn more about uploading source code into your project area and how the directories are arranged', 'Learn more About the Rappture toolkit.', and 'Learn about nanoHUB's software development environment'. At the bottom of the 'What's next?' section, it says 'When you are ready, Follow these instructions to access the source code repository for your specific project and upload your code.' Below the 'What's next?' section, there is a 'We are waiting for You' section. It says: 'Once your source code has been uploaded into your project area, click here to let us know:'.

Click link following nanoFORGE

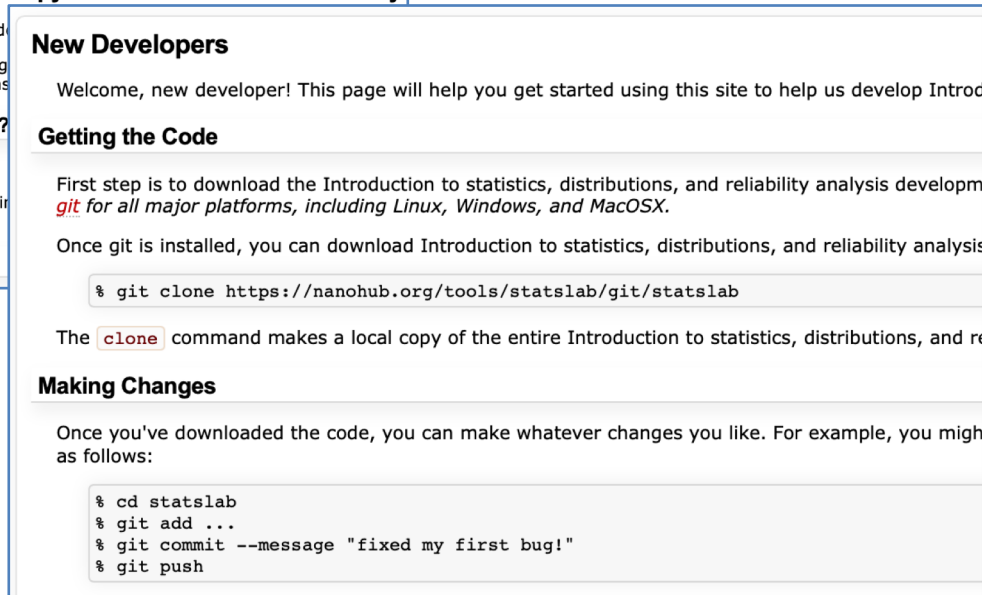
Tool registered by nanoHUB – ready for you to add code



The screenshot shows the nanoHUB Forge interface. At the top is the HUB FORGE logo. Below it, there's a link to 'wiki: WikiStart'. The main heading is 'Introduction to statistics, distributions, and reliability analysis: Jupyter notebooks with introductory'. Underneath, there's a welcome message and a list of links: 'GettingStarted' and 'TicketSystem'. There are sections for 'What is Introduction to statistics, distributions, and reliability analysis?' and 'How do I use this site?'.

Getting started shows you how to check out the code

Simple git commands to develop code



The screenshot shows the 'New Developers' section. It starts with a welcome message: 'Welcome, new developer! This page will help you get started using this site to help us develop Intro'. Below that is the 'Getting the Code' section, which explains that the first step is to download the code using git for all major platforms (Linux, Windows, and MacOSX). It provides a terminal command: `% git clone https://nanohub.org/tools/statslab/git/statslab`. The text explains that the 'clone' command makes a local copy of the entire project. The 'Making Changes' section follows, stating that once the code is downloaded, users can make changes. It provides a terminal command sequence: `% cd statslab`, `% git add ...`, `% git commit --message "fixed my first bug!"`, and `% git push`.



Open a terminal to move the code to the repository

The screenshot shows the nanoHUB Jupyter interface. At the top, there are logos for nanoHUB and Jupyter, along with buttons for 'Submit a ticket' and 'Terminate Session'. Below the logos, there are navigation tabs: 'Files', 'Running', 'Formgrader', 'Assignments', and 'Courses'. The 'Files' tab is active, and the text 'Select items to perform actions on them.' is displayed. A file list is shown with columns for 'Name', 'Status', and 'Size'. The first item is 'TestNotebook.ipynb' with a 'Running' status. A 'New' dropdown menu is open, showing options for 'Notebook' (Octave 4.2.0, Python (jul262020), Python (tellurium), Python 3, Python2, R) and 'Other' (Text File, Folder, Terminal, noVNC Desktop). The 'Terminal' option is highlighted.

Name	Status	Size
TestNotebook.ipynb	Running	
Untitled10.ipynb		
featureselect		
statslab		
model.h5		
msemI_data.csv		
youngsmo		
citrinednn		
matlabdata		
scaledatаре		
data		
Untitled9.ipynb		594 B



Setup your tool

Check out (clone) the empty tool directory structure

  Submit a ticket Terminate Session

```
strachan@nanohub_1748723_5:~$ git clone https://nanohub.org/tools/statslab/git/statslab
```

Check out the directory structure (statslab should be change to your tool's short name)

  Submit a ticket Terminate Session

```
strachan@nanohub_1748723_5:~$ cd statslab/  
strachan@nanohub_1748723_5:~/statslab$ ls  
bin data doc examples middleware rappture README.md src  
strachan@nanohub_1748723_5:~/statslab$
```

Copy your notebook to the bin directory:

```
>cp ../TestNotebook.ipynb bin/
```

The bin & middleware directories



```
strachan@nanohub_1748723_5:~/statslab$ ls -ltr bin/
total 140
-rw-r--r-- 1 strachan public 14997 Dec  9 13:05 Module3.ipynb
-rw-r--r-- 1 strachan public 15314 Dec  9 13:07 LandingPage.ipynb
-rw-r--r-- 1 strachan public 60309 Dec  9 13:22 Module2.ipynb
-rw-r--r-- 1 strachan public 45365 Dec 14 21:09 Module1.ipynb
strachan@nanohub_1748723_5:~/statslab$
```

```
strachan@nanohub_1748723_5:~/statslab$ vi middleware/invoke
strachan@nanohub_1748723_5:~/statslab$
```

```
#!/bin/sh

#
# jupyter tool
#
/usr/bin/invoke_app "$@" -t statslab \
                    -C "start_jupyter -t -T @tool statslab.ipynb" \
                    -u anaconda-6
```

When you are done, commit the code

Making Changes

Once you've downloaded the code, you can make whatever changes you like. For example, you might edit a file to fix a bug. Here's how to do that as follows:

```
% cd statslab
% git add .
% git commit --message "fixed my first bug!"
% git push
```

It's best to commit at the top of the source tree--that's why we said "`cd statslab`" in the example above. When you commit them all at once. Pushing a change makes it permanent. Once pushed, other developers will see the change in the source tree and check it out all over again.

If you want to add a new file or directory to your distribution, you can use the `add` command:

```
% git add README.txt
% git commit --message "made my first addition"
```

Like any other change, the file is not really added until the next `push` operation.

Similarly, if you want to remove a file or directory from your distribution, you can use the `delete` command:

```
% git rm loadInput.c
% git commit --message "removed deprecated code"
```

Once the change is committed, the file will disappear. The file is still kept in the history, so it is not completely gone.

From time to time, you and another developer will modify the same file at the same time. Suppose the other developer has a more up-to-date version. In that case, you need to `pull` before committing. You can do that as follows:

```
% cd statslab
% git pull
```

Add files
Commit changes
Push your changes

Use "pull" to get the most updated version of the code

Let nanoHUB know when you are done

The screenshot shows the nanoHUB interface for a tool named 'statslab'. The status is 'Created'. The page is divided into several sections: 'Tool Information', 'Developer Tools', 'What's next?', and 'We are waiting for You'. A blue arrow points from the text 'Click here' to the link 'My code is committed, working, and ready to be installed'.

Tools: Status for statslab - Created All Tools New Tool

STATUS: Registered **Created** Uploaded Installed Approved Published

This tool is one of 674 tools under development on nanoHUB.org.

Tool Information [edit](#)

Title	Introduction to statistics, distributions, and reliability analysis (statslab - id #1792)
Version	This version 1.0 (under development)
At a glance	Jupyter notebooks with introductory materials and hands-on exercises
Description	Preview Edit description page
VNC geometry	780x600
Repository Host Type	gitLocal
Tool execution	open to public
Source code	open source
Project area	open to public
Publishing Option	jupyter
Development team	ganeshs, strachan, jverduzc, rapplet, psalek

Developer Tools

[History](#) [Wiki](#) [Source](#) [Timeline](#) [Message](#) [Cancel](#)

What's next?

The nanoHUB.org team has created the following project area for your tool on the nanoFORGE:
<https://nanohub.org/tools/statslab/wiki>

Follow these steps to start using your project area:

- [Learn more](#) about uploading source code into your project area and how the directories are arranged
- Learn more About the [Rappature toolkit](#).
- Learn about nanoHUB's [software development environment](#)

When you are ready, [Follow these instructions](#) to access the source code repository for your specific project and upload your code.

We are waiting for You

Once your source code has been uploaded into your project area, [click here](#) to let us know:

- [My code is committed, working, and ready to be installed](#)

Remaining steps before we can publish your tool:

Click here

Try your tool before deploying it

The screenshot shows the nanoHUB interface for a tool named 'featureselect'. The top navigation bar includes 'RESOURCES', 'EXPLORE', 'NANOHUB-U', 'PARTNERS', 'COMMUNITY', 'ABOUT', 'SUPPORT', 'DONATE', and 'TAKE A POLL'. The user is logged in. The breadcrumb trail is 'Home > Tools > Tool Pipeline > Status for featureselect'. The main heading is 'Tools: Status for featureselect - Installed'. Below this, there are buttons for 'All Tools' and 'New Tool'. A status bar shows 'Registered', 'Created', 'Uploaded', 'Installed' (highlighted), 'Approved', and 'Published'. A message states: 'This tool is one of 674 tools under development on nanoHUB.org.' The 'Tool Information' section includes: Title: 'Feature Selection for Machine Learning (featureselect - id #1793)', Version: 'This version 1.0 (under development)', At a glance: 'Assessing feature selection for machine learning models', Description: 'Preview | Edit description page', VNC geometry: '780x600', Repository Host Type: 'gitLocal', Tool execution: 'open to public', Source code: 'open source [change license]', Project area: 'open to public', Publishing Option: 'jupyter', and Development team: 'strachan, mcclurez'. The 'Developer Tools' section is also visible. The 'What's next?' section contains: 'Your latest code is installed and ready on nanohub.org/. Please test your tool by clicking the button below to make sure that everything is working properly, as well as verify that the page describing your tool is created and displays correct information:'. It lists two steps: 'Test your application:' with a 'Launch tool' button, and 'Review the page describing your tool'. The 'We are waiting for You' section says: 'Once you tested your tool and verified that it is working properly, click here to let us know:'. It lists two options: 'My tool is working properly. I approve it.' and 'Need to make changes? Once you've checked in your latest fixes, click here to let us know:'. It lists one option: 'I've committed new code. Please install the latest version for testing and'.

Test your tool

When you are done, click here.

Done!

Check out what is available

Libraries available Jupyter

```
In [4]: import pkg_resources
dists = [str(d) for d in pkg_resources.working_set]
print(dists)

['-illow 5.4.1', 'zstandard 0.11.0', 'zipp 0.6.0', 'zict 0.1.4', 'yarl 1.3.0', 'yamlmagic 0.2.0', 'xmldict 0.12.0', 'xlwt 1.3.0', 'xlsxwriter 1.1.5', 'xlrd 1.2.0', 'xgboost 0.90', 'xarray 0.11.3', 'wurlitzer 1.0.2', 'wra pt 1.11.1', 'wolframalpha 3.0.1', 'widgetsnbextension 3.5.1', 'whichcraft 0.5.2', 'wheel 0.34.2', 'Werkzeug 0.14.1', 'websocket-client 0.56.0', 'webencodings 0.5.1', 'wcwidth 0.1.7', 'watchdog 0.9.0', 'w3lib 1.20.0', 'vtk 8.1.2', 'virtualenv 16.5.0', 'vine 1.3.0', 'urllib3 1.25.9', 'unicodcsv 0.14.1', 'uncertainties 3.0.3', 'typed-ast 1.4.0', 'twine 1.13.0', 'traitlets 0.2.1', 'traitlets 4.3.2', 'tqdm 4.47.0', 'tox 3.9.0', 'tornado 5.1.1', 'toolz 0.9.0', 'toml 0.10.0', 'tinydb 3.15.2', 'threadpoolctl 2.1.0', 'tf-estimator-nightly 1.14.0.dev2019030115', 'textwrap3 0.9.2', 'testpath 0.4.2', 'tesedml 0.4.5.0', 'tesbl 5.18.0', 'terminado 0.8.1', 'termcolor 1.1.0', 'tenuml 1.1.1.2', 'tensorflow 1.13.1', 'tensorflow-estimator 1.13.0', 'tensorboard 1.13.1', 'tenacity 5.0.4', 'tellurium 2.0.3', 'tecombine 0.2.3.0', 'tblib 1.3.2', 'tb-nightly 1.14.0a20190301', 'tabulate 0.8.3', 'ta bles 3.5.1', 'sympy 1.3', 'statsmodels 0.9.0', 'sshunnel 0.1.4', 'sqlparse 0.3.0', 'SQLAlchemy 1.2.1', 'spyder 3.3.3', 'spyder-kernels 0.4.2', 'sphinxcontrib-websupport 1.1.0', 'sphinxcontrib-serializinghtml 1.0.2', 'sphinxcontrib-qthelp 1.0.2', 'sphinxcontrib-jsmath 1.0.1', 'sphinxcontrib-htmlhelp 1.0.2', 'sphinxcontrib-applehelp 1.0.1', 'Sphinx 2.1.2', 'sphinx-py3doc-enhanced-theme 0.5.1', 'slicer 0.0.3', 'six 1.15.0', 'singledispatch 3.4.0.3', 'simtool 0.2.3', 'simplic 0.8.1', 'simpervisor 0.3', 'shiboken2 5.15.0', 'shap 0.37.0', 'setuptools 49.2.0', 'SecretStorage 3.1.1', 'seaborn 0.9.0', 'scrapbook 0.2.0', 'scipy 1.2.1', 's t-rf 0.14.9', 'scikit-optimize 0.5.2', 'scikit-learn 0.23.1', 'scikit-image 0.14.2', 'el-yaml 0.15.89', 'ruamel-yaml 0.15.87', 'rplugins 1.2.2', 'rpy2 2.9.1', 'rope 0.12.0', 'g 1.3.3', 'retry 0.9.2', 'researchpy 0.1.7', 'requests 2.24.0', 'requests-toolbelt 0.10.0', 'querystring-parser 1.2.3', 'QtPy 1.7.0', 'qtconsole 4.4.3', 'QtAwesome 0.5.7',
```

Libraries and codes available to be invoked

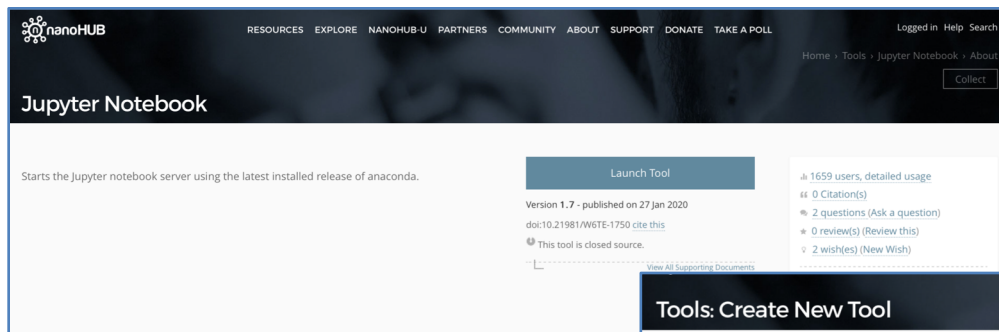
```
strachan@nanohub_1749163_77:~$ . /etc/enviro...
strachan@nanohub_1749163_77:~$ use 2>&1 | grep lammps
lammps-01Feb14: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-03Mar20: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-05Jun19: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-06Apr15: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-07Aug19Curl: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-07Aug19: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-07Jul09: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-09Dec14: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-11Aug17: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-12Feb07: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-15Jan10: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-15May15: * Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-17Feb12: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-22Aug18: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-22Jan08: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-30Oct14: Large-scale Atomic/Molecular Massively Parallel Simulator
lammps-31Mar17: Large-scale Atomic/Molecular Massively Parallel Simulator
strachan@nanohub_1749163_77:~$ man use
strachan@nanohub_1749163_77:~$
```



Jump in!

Start developing your code!

<https://nanohub.org/tools/jupyter> (you will need a free nanoHUB account to launch Jupyter)



Start the publication process

<https://nanohub.org/tools/create>

A screenshot of the 'Tools: Create New Tool' form on nanoHUB. The form is titled 'ABOUT YOUR TOOL:' and contains several input fields. The 'Tool Name' field is marked as 'REQUIRED'. Below it is a 'Short name' field with the example 'qdot'. The 'Title' field is also marked as 'REQUIRED'. Below it is a 'Full name' field with the example 'Quantum Dot Lab'. The 'Version' field contains '1.0' and has a note: 'Optional version number for this release of the tool. Example: 1.0 or 2.1.5b. Spaces not allowed.' The 'At a glance' field is marked as 'REQUIRED' and has a note: 'A one-line description of your tool. Example: Simulate 3-D confined states in simple quantum dot geometries.'

Share your work with your students and colleagues

Why deploy tools in nanoHUB?

- Tools are publications: indexed by Web of Science and Google Scholar.
- Impact statistics: usage metrics, secondary citations
- Anyone can run your tool without downloading/installing any software
 - Use it in the classroom
 - Document your research and make it reproducible (tools are containerized)
- A full scientific software development environment
 - Libraries for machine learning, statistics, data science, visualization, etc.
- Launch physics-based, research grade codes
 - LAMMPS, Quantum Espresso, NEMO5, abinit, MOOSE, and many more

