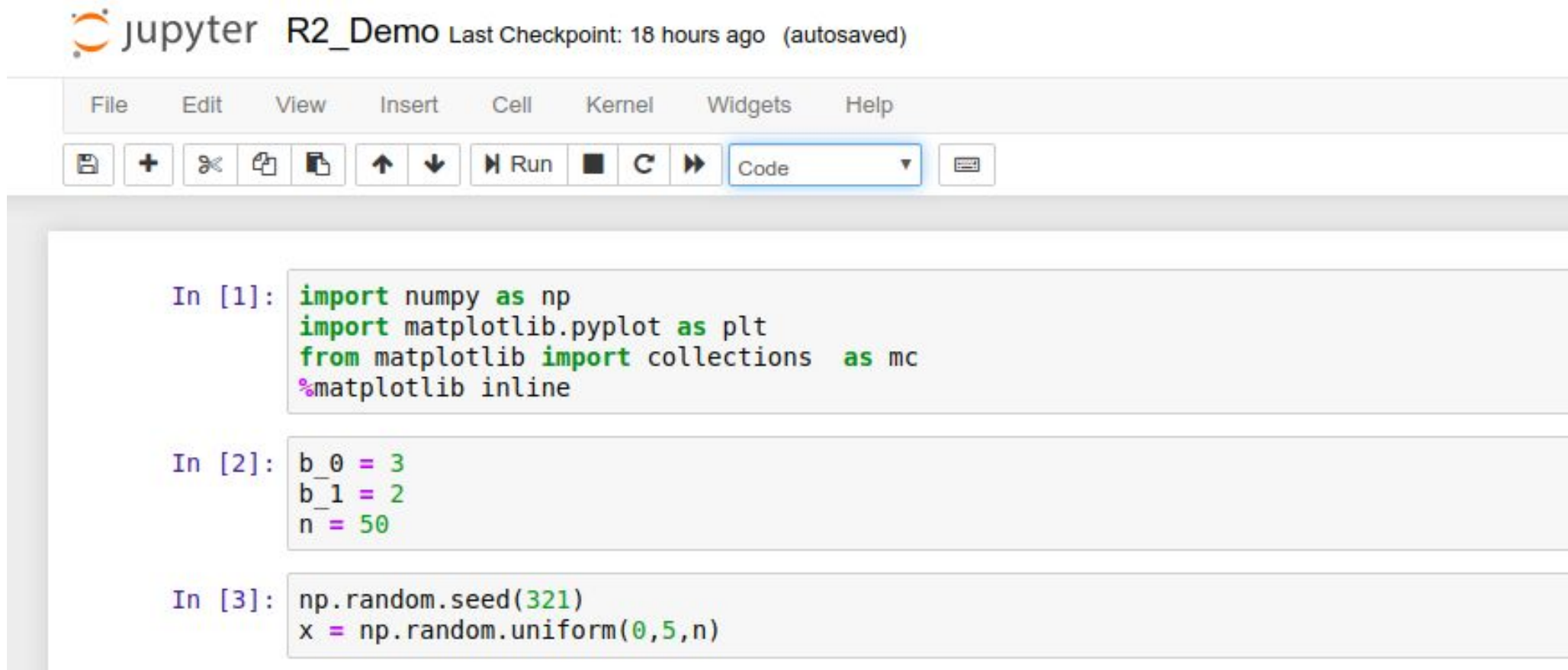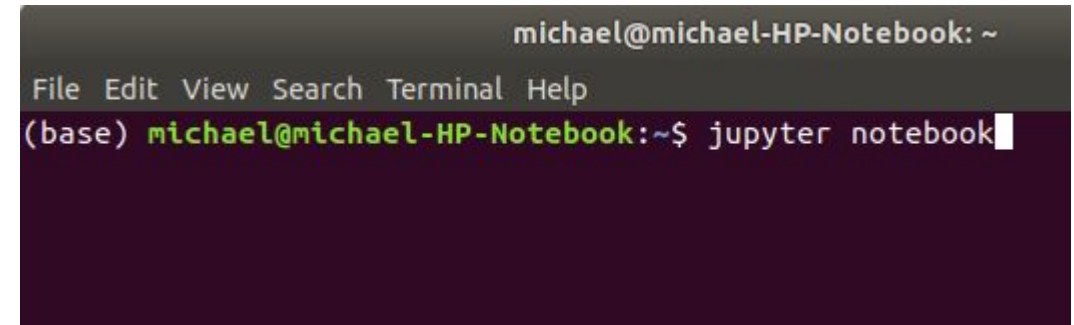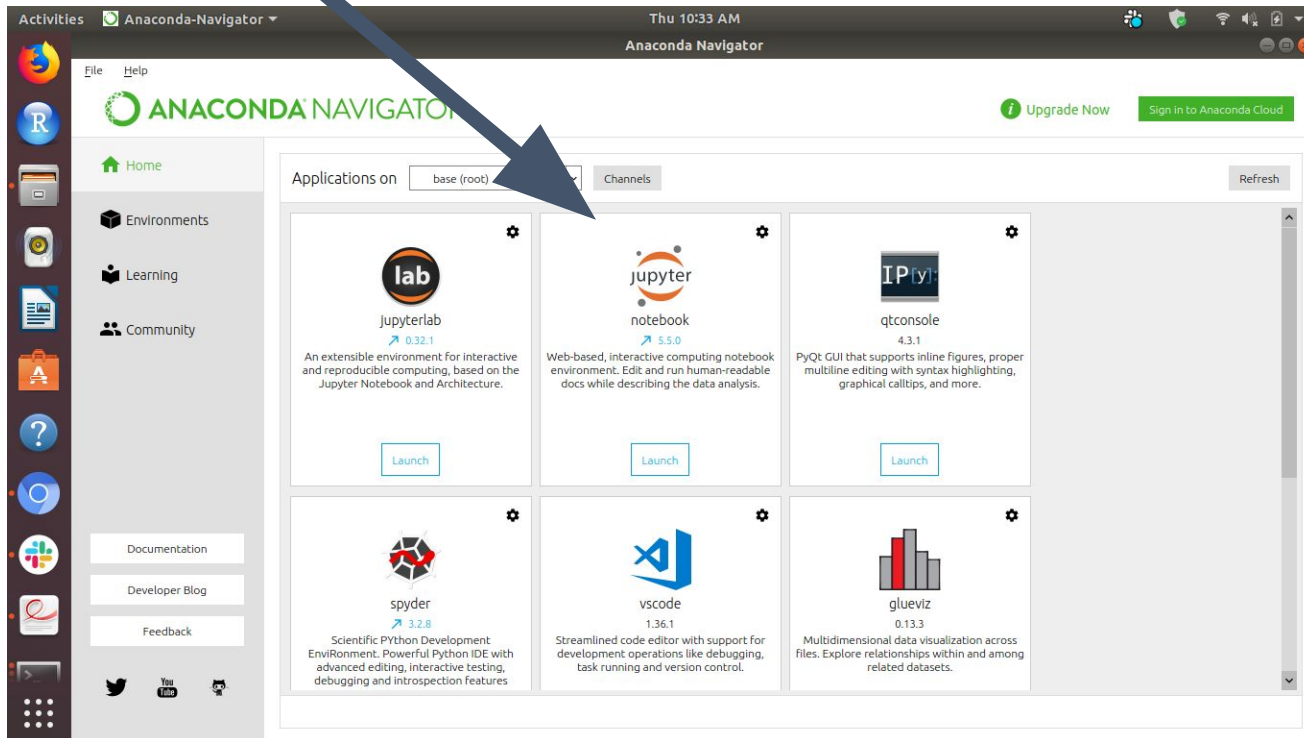# Getting Started with Jupyter Notebooks

# Jupyter Notebooks

A web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text.
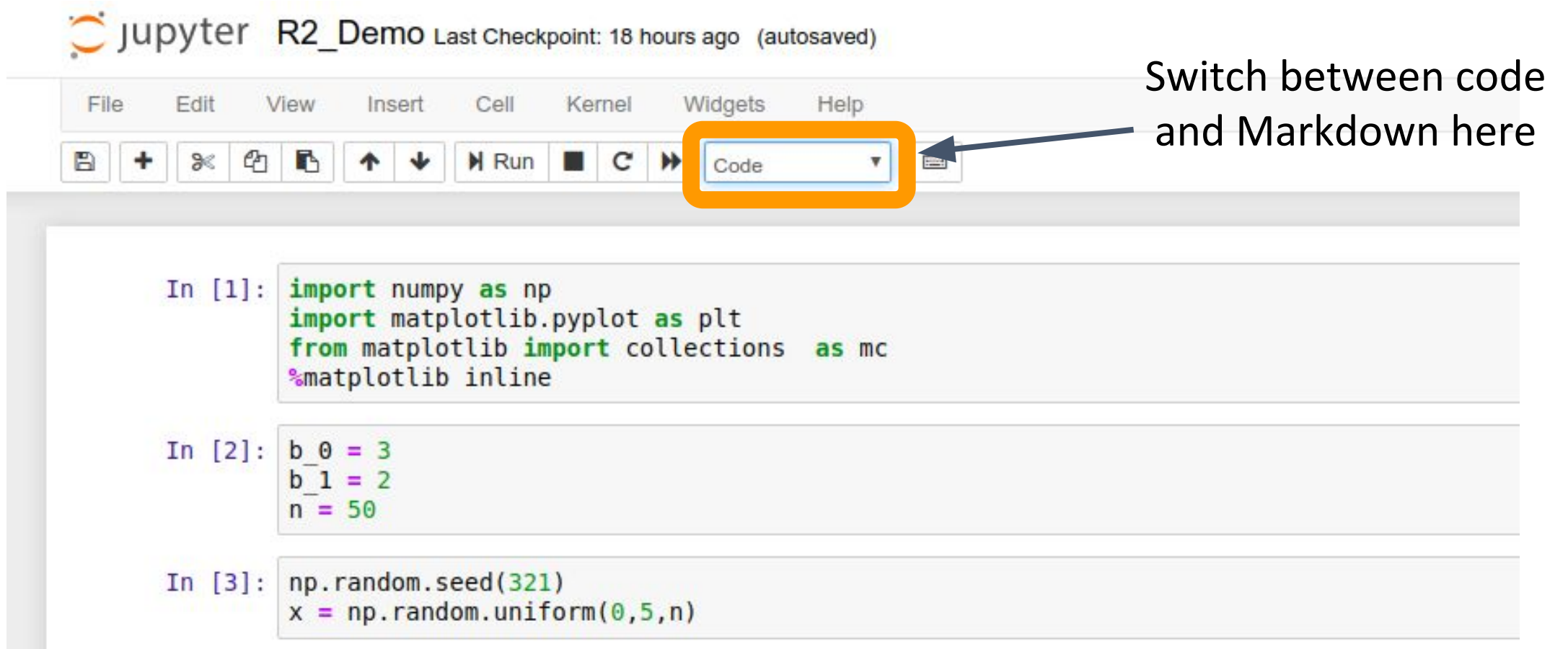
# Jupyter Notebooks

Can be launched either from Anaconda Navigator or by using the 'jupyter notebook' command from the terminal.

## Jupyter Notebooks

Notebooks are divided into cells, which contain either code or Markdown, which can be used to annotate your work.



Switch between code and Markdown here

# Jupyter Notebooks

To run code:
- Ctrl + Enter : Run cell and remain in cell
- Shift + Enter : Run cell and move to next
- Alt + Enter: Run cell and insert new cell below

Notebooks have lots of shortcuts and ways to make you more efficient (tab-completion is your friend).

For example, Shift + Tab inside a function to bring up docstring

**Resources:**
https://www.cheatography.com/weidadeyue/cheat-sheets/jupyter-notebook/
https://www.dataquest.io/blog/jupyter-notebook-tips-tricks-shortcuts/

# Errors happen!!

- Opportunity to gain knowledge
- Read the output (stack trace) from the bottom up to see the error and what caused it
- If you have no idea what is causing the error, try copying it, and pasting to a Google search

```
In [2]: x + 2

    -------------------------------------------------------------------------------------
    NameError                                    Traceback (most recent call last)
    <ipython-input-2-2c7a9192c558> in <module>()
    ----> 1 x + 2

    NameError: name 'x' is not defined
```
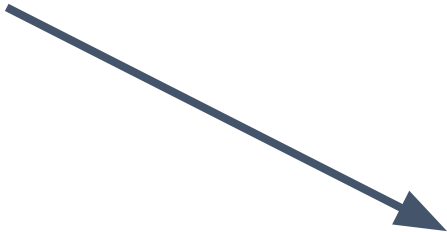
Often when you get an error, you will get a long stack trace.

Start looking down here

```
In [34]: df.rename(lambda x: x.upper)
```

```
-------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-34-18dc4f7801aa> in <module>()
----> 1 df.rename(lambda x: x.upper)

~/anaconda3/lib/python3.6/site-packages/pandas/util/_decorators.py in wrapper(*args, **kwargs)
    195         @wraps(func)
    196         def wrapper(*args, **kwargs):
--> 197             return func(*args, **kwargs)
    198
    199         if not PY2:

~/anaconda3/lib/python3.6/site-packages/pandas/core/frame.py in rename(self, *args, **kwargs)
   4023         kwargs.pop('axis', None)
   4024         kwargs.pop('mapper', None)
-> 4025         return super(DataFrame, self).rename(**kwargs)
   4026
   4027     @Substitution('**_shared_doc_kwargs)

~/anaconda3/lib/python3.6/site-packages/pandas/core/generic.py in rename(self, *args, **kwargs)
   1089                 level = self.axes[axis]._get_level_number(level)
   1090             result._data = result._data.rename_axis(f, axis=baxis, copy=copy,
-> 1091                                                     level=level)
   1092             result._clear_item_cache()
   1093

~/anaconda3/lib/python3.6/site-packages/pandas/core/internals/managers.py in rename_axis(self, mapper, axis, copy, level)
    169         ...
    170         obj = self.copy(deep=copy)
--> 171         obj.set_axis(axis, _transform_index(self.axes[axis], mapper, level))
    172         return obj
    173

~/anaconda3/lib/python3.6/site-packages/pandas/core/internals/managers.py in _transform_index(index, func, level)
   2002         return MultiIndex.from_tuples(items, names=index.names)
   2003     else:
-> 2004         items = [func(x) for x in index]
   2005         return Index(items, name=index.name, tupleize_cols=False)
   2006

~/anaconda3/lib/python3.6/site-packages/pandas/core/internals/managers.py in <listcomp>(.0)
   2002         return MultiIndex.from_tuples(items, names=index.names)
   2003     else:
-> 2004         items = [func(x) for x in index]
   2005         return Index(items, name=index.name, tupleize_cols=False)
   2006

<ipython-input-34-18dc4f7801aa> in <lambda>(x)
----> 1 df.rename(lambda x: x.upper)

AttributeError: 'int' object has no attribute 'upper'
```

# Stack Overflow

- **Question is at the top**
- **Answers underneath**
- **Pay attention to upvotes and the selected answer**

**Example:**

https://stackoverflow.com/questions/11346283/renaming-columns-in-pandas

# Package API Reference

- Can be harder to read when you are first learning
- Get into all of the nitty-gritty. You can even view the source code if you want. This can be powerful as your knowledge of Python grows.

**Example:**
https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.rename.html#pandas.DataFrame.rename