EASC2410 Lecture 8

# Python Basics: Data on Maps

Dr. Binzheng Zhang
Department of Earth Sciences

# Review of Lecture 7

**In Lecture 7, we learned:**

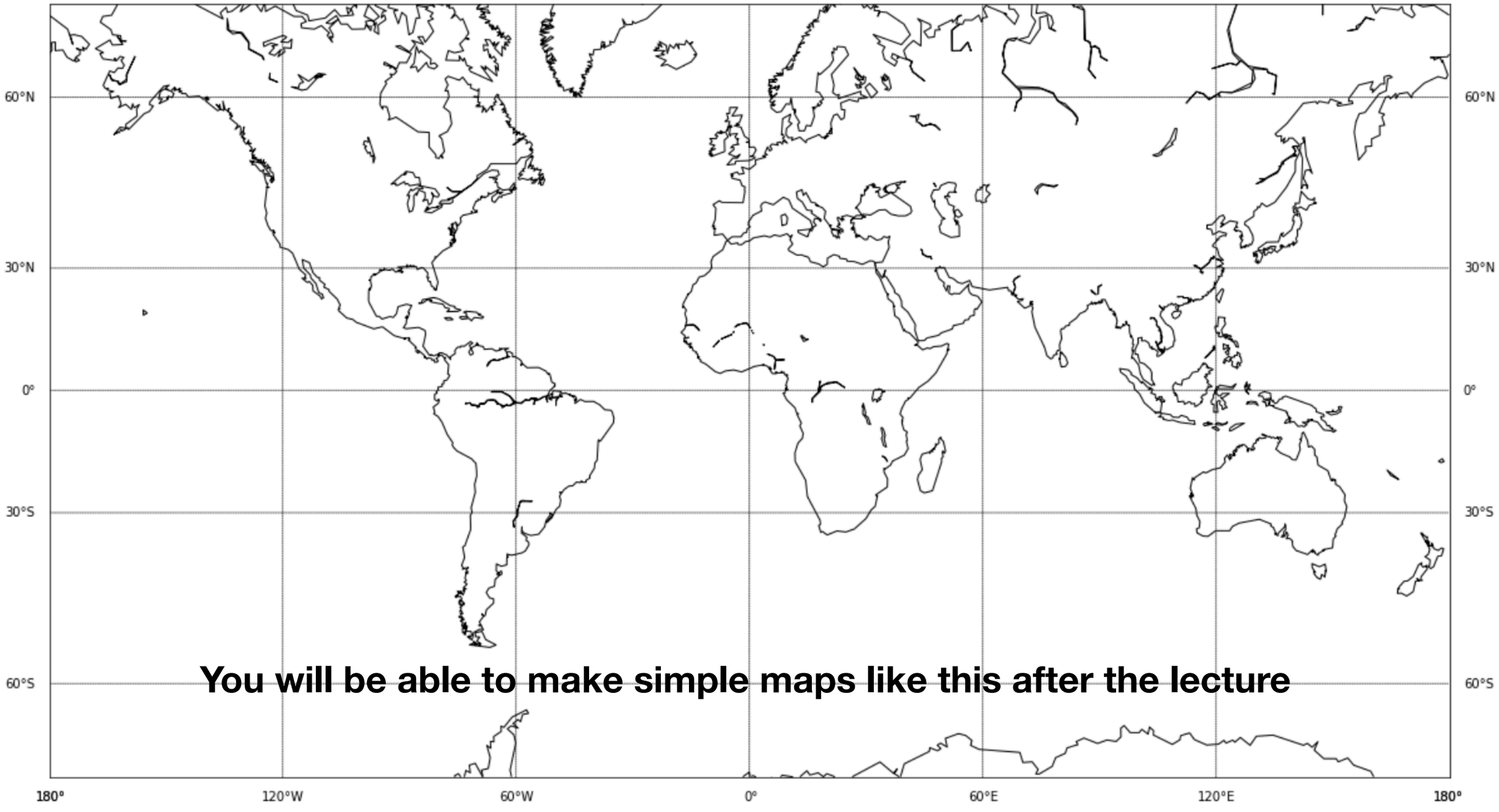
- **2-D NumPy arrays**
- **Index slicing of 2-D arrays**
- **Load data into Python using loadtxt() and genfromtxt()**
- **Visualize your data**

**In Lecture 8, you will learn:**

- **Creating maps using the *basemap* module**
- **plotting spatial data points on your maps**

# What is a map?

A map is basically a **projection** tries to represent something that is essentially 3D (a globe) onto a 2D medium (a paper or a computer screen). So all maps, except those at the smallest scale, will increasingly distort the area as the scale increases because the Earth is not flat.



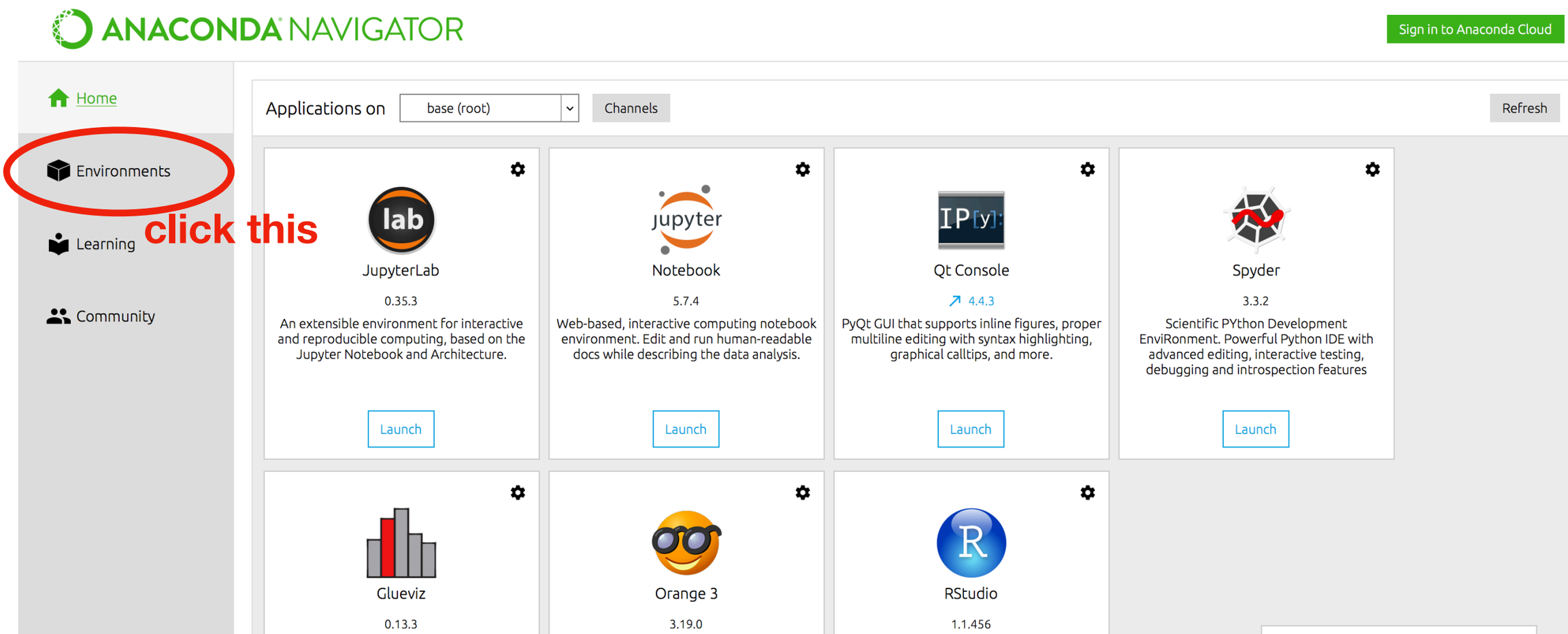

**You will be able to make simple maps like this after the lecture**

# How to make maps in Python?

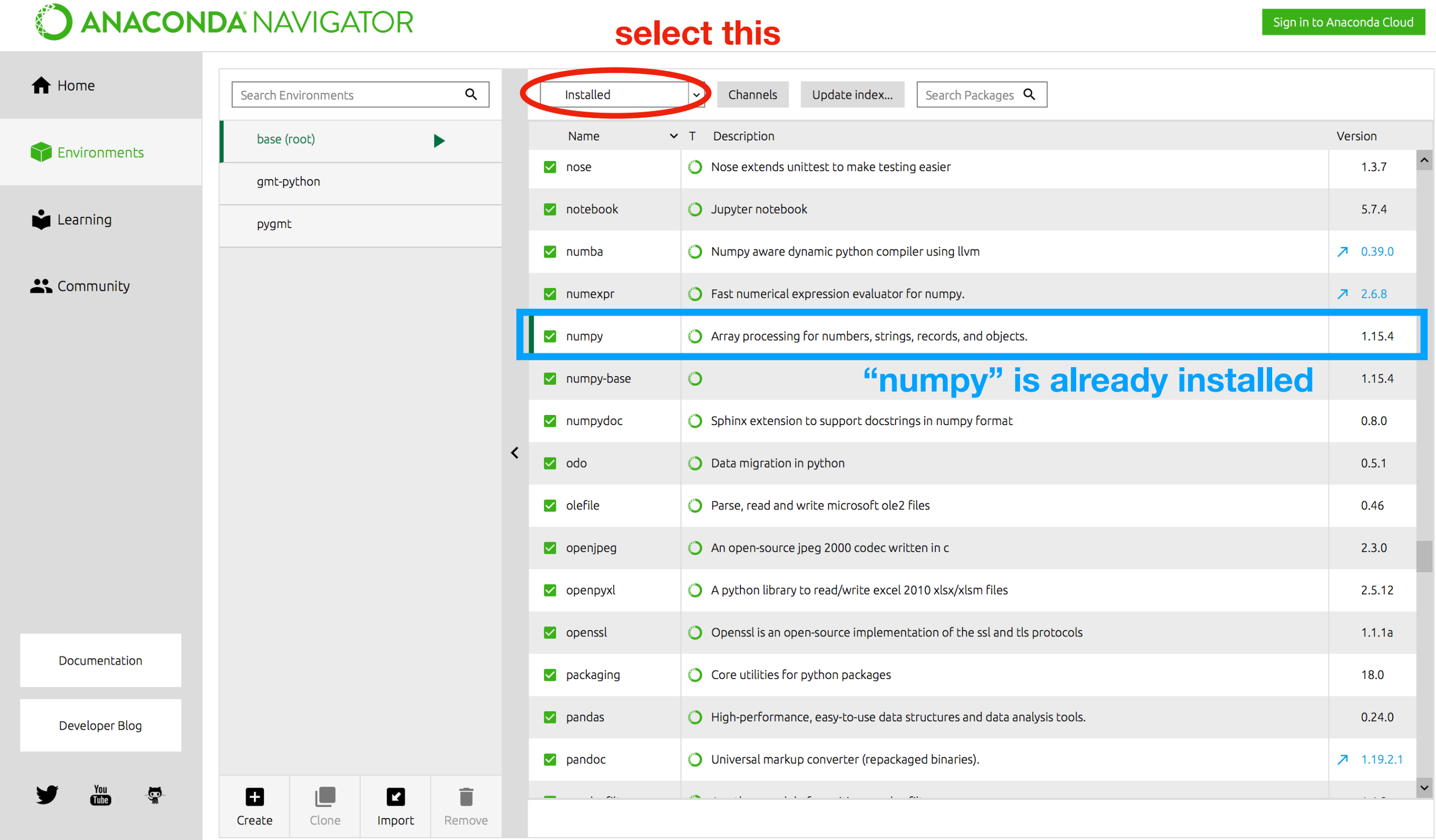
There are many ways to make maps in Python. In this course, our maps in Python are plotted using the tools in **matplotlib**'s **Basemap** module. So the first thing we need to do is install the **basemap** module in *anaconda*:

# Why we need to “install” basemap?

We have used modules in the previous practices and homework assignments, such as **math**, **numpy** and **matplotlib**. Why we didn’t need to install these libraries? **Because Anaconda has already done that for you!**



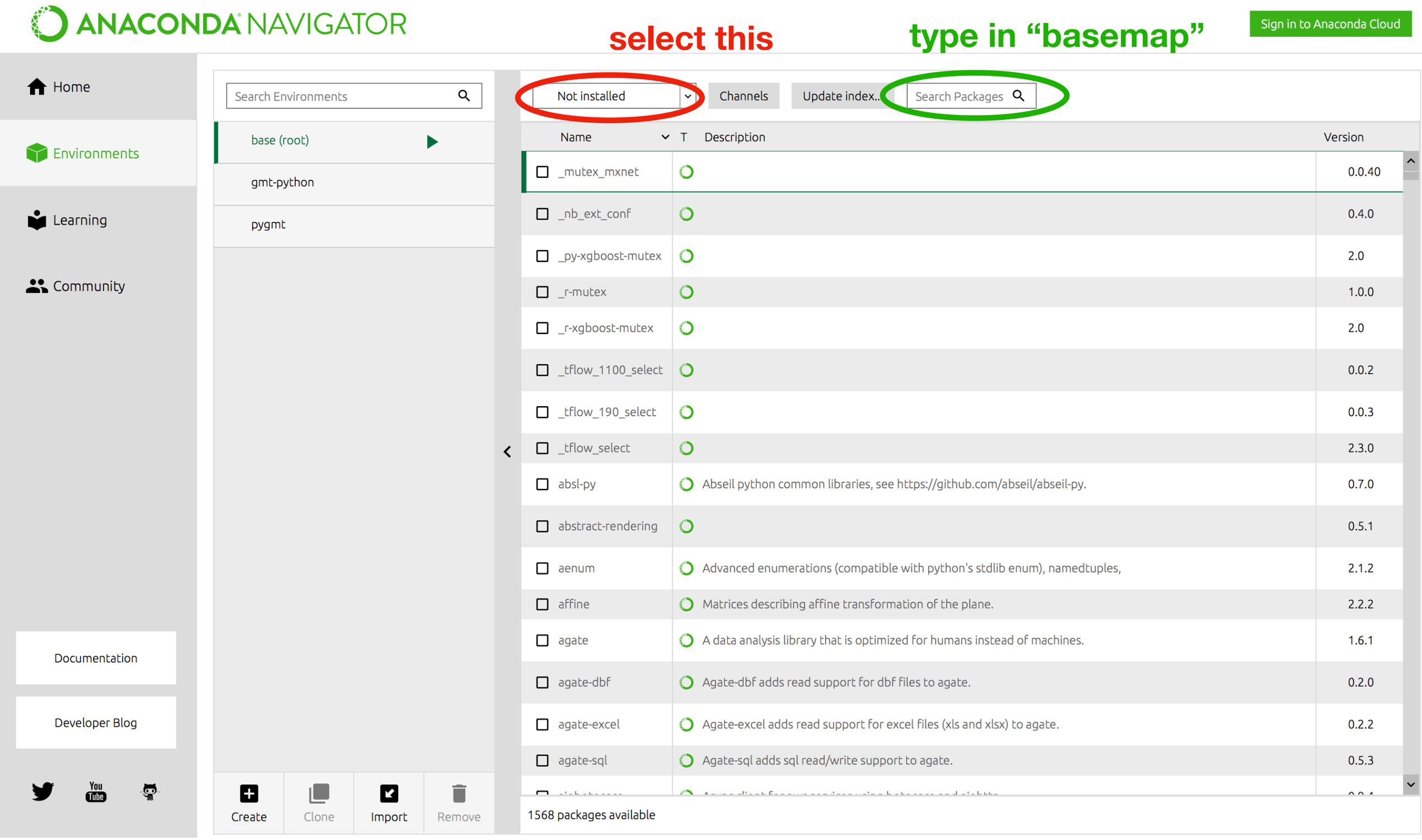
# Check modules that are already installed




You can also find that a fairly large amount of modules are already installed, such as matplotlib, pandas, etc.

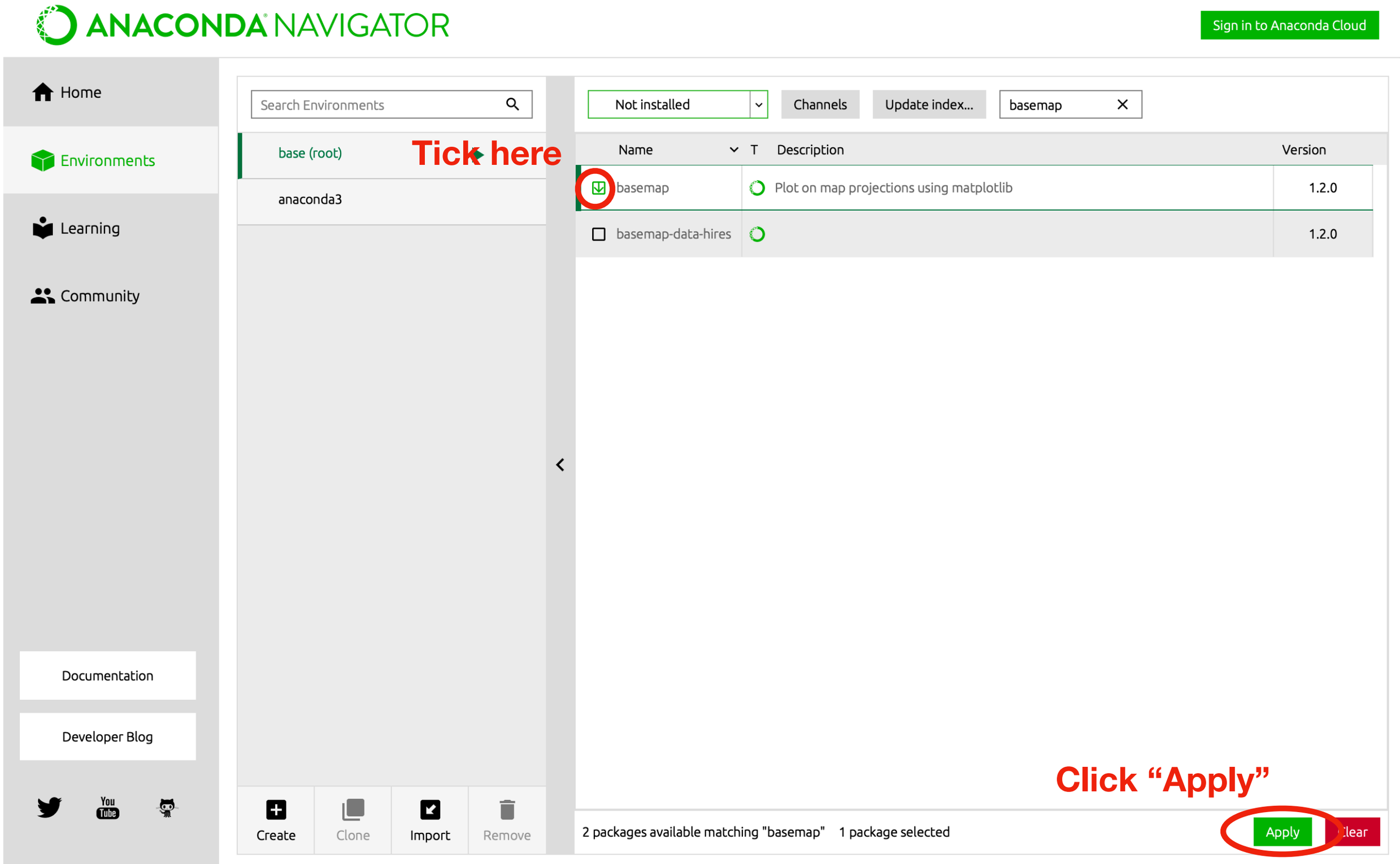
# So, How to “install” basemap? - it’s relatively easy in Anaconda

Step 1: Select the **"Not installed"** option from the first drop-down manu



# So, How to “install” basemap? - it’s relatively easy in Anaconda

Step 2: Tick the **"basemap"** and click apply



## So, How to “install” basemap? - it’s relatively easy in Anaconda

Step 3: Wait and let Anaconda does it’s own job…

## How to use basemap?

For most of the users you should be able to use the basemap module using the following codes (note that numpy and matplotlib is also needed to make maps:

In [ ]:
```
import numpy as np # import numpy
import matplotlib.pyplot as plt # import pyplot
from mpl_toolkits.basemap import Basemap # import basemap module using this syntax!
```

Some users may have problem with the $PATH for the pyproj library, so you might need to use the following code to load the basemap module (it’s not the best way to resolve the problem, but it’s an easy way to get your computer working properly with basemap!)

In [1]:
```
import numpy as np # import numpy
import matplotlib.pyplot as plt # import pyplot

import os # build-in module OperatingSystem
os.environ['PROJ_LIB'] = '<path of your Anaconda>/share/proj' # setting the environment path for basemap

from mpl_toolkits.basemap import Basemap # import the basemap module using this syntax!
```

Now let’s create maps!

## Basic Syntax of creating maps using basemap

```
m = basemap(Projection, [Options])

m.drawcoastlines()
```

m is an object of basemap class (just regard it as a map named “m”)

basemap() is the function that creates a map object

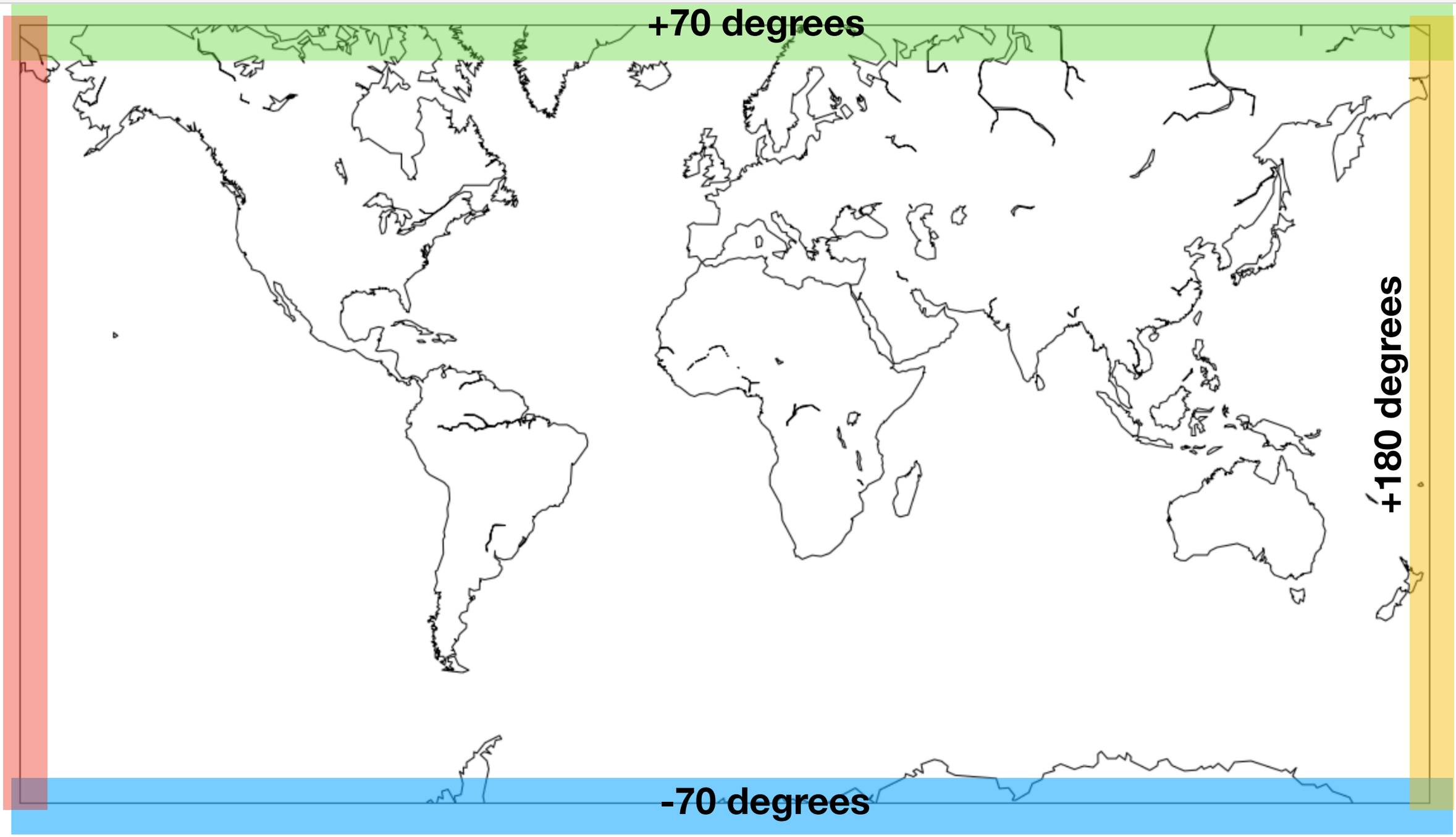
the keyword “Projection” tells basemap what kind of map you want

Various options specify the style, label, colors etc of the map

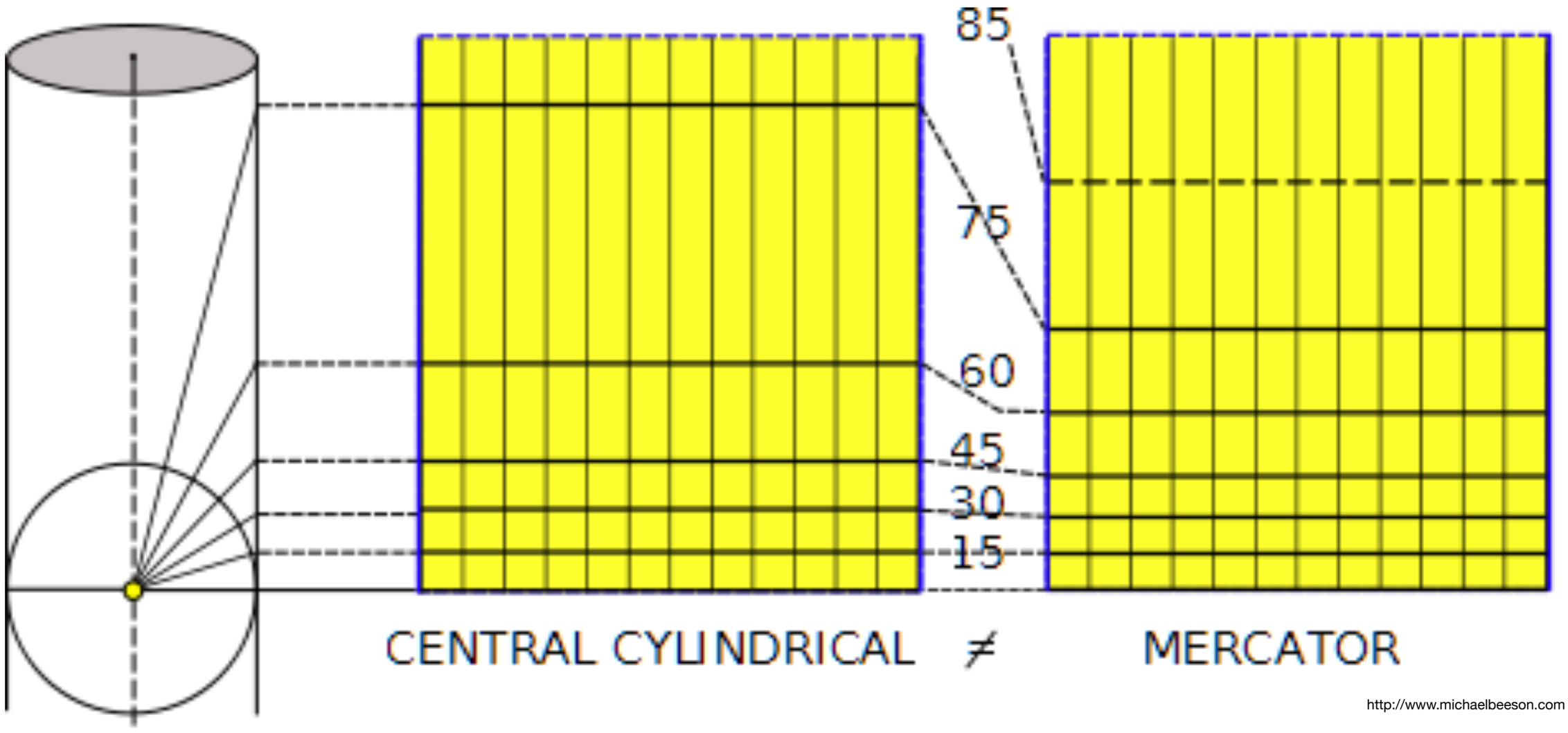
drawcoastlines() function put on the coast lines so you can see a map

# Mercator Projection of the world map (merc)

In [3]:
```
plt.figure(figsize=(20,15)) # create a new figure with size (16,15)

# make a map instance called 'm', pay attention to the boundaries of the map
m = Basemap(projection='merc',llcrnrlat=-70,urcrnrlat=70,llcrnrlon=-180,urcrnrlon=180)
m.drawcoastlines(); # put on the coastlines
plt.show() # show your map!
```



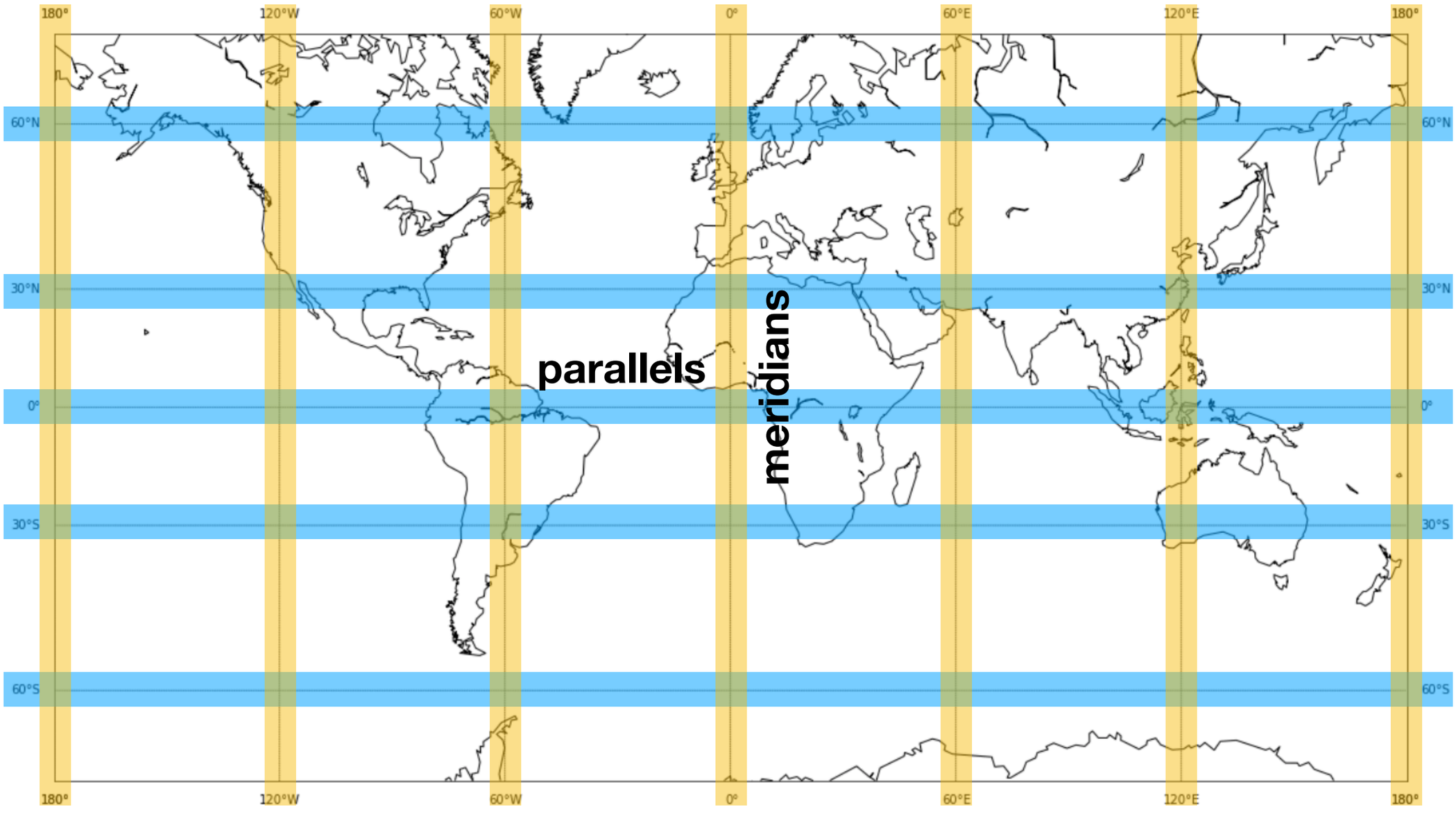
# Mercator Projection of the world map




- **The most popular projection of maps**
- **Less distortion in mid and low latitude regions**
- **Large distortion in high latitude regions (see Iceland)**

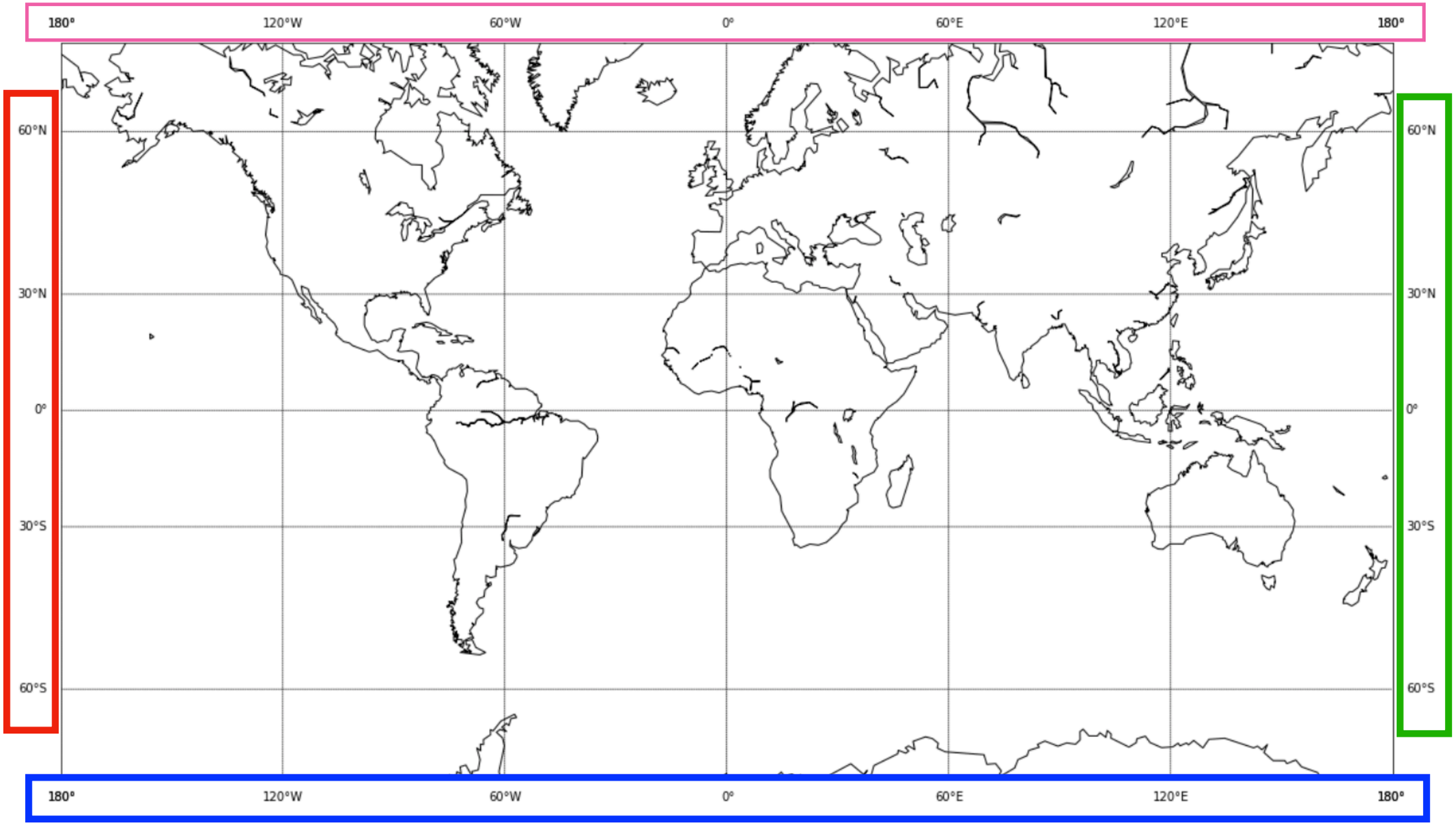
# Label your maps

In [6]:
```
plt.figure(figsize=(20,15)) # create a new figure with size (16,15)

# make a map instance called 'm', pay attention to the boundaries of the map
m = Basemap(projection='merc',llcrnrlat=-70,urcrnrlat=70,llcrnrlon=-180,urcrnrlon=180)
m.drawcoastlines(); # put on the coastlines

m.drawparallels(np.arange(-90.,91.,30.), labels=[True,True,False,False]) # plot the latitudes (
m.drawmeridians(np.arange(-180.,181.,60.),labels=[False,False,True,True]); # same for longitude

plt.show() # show your map!
```



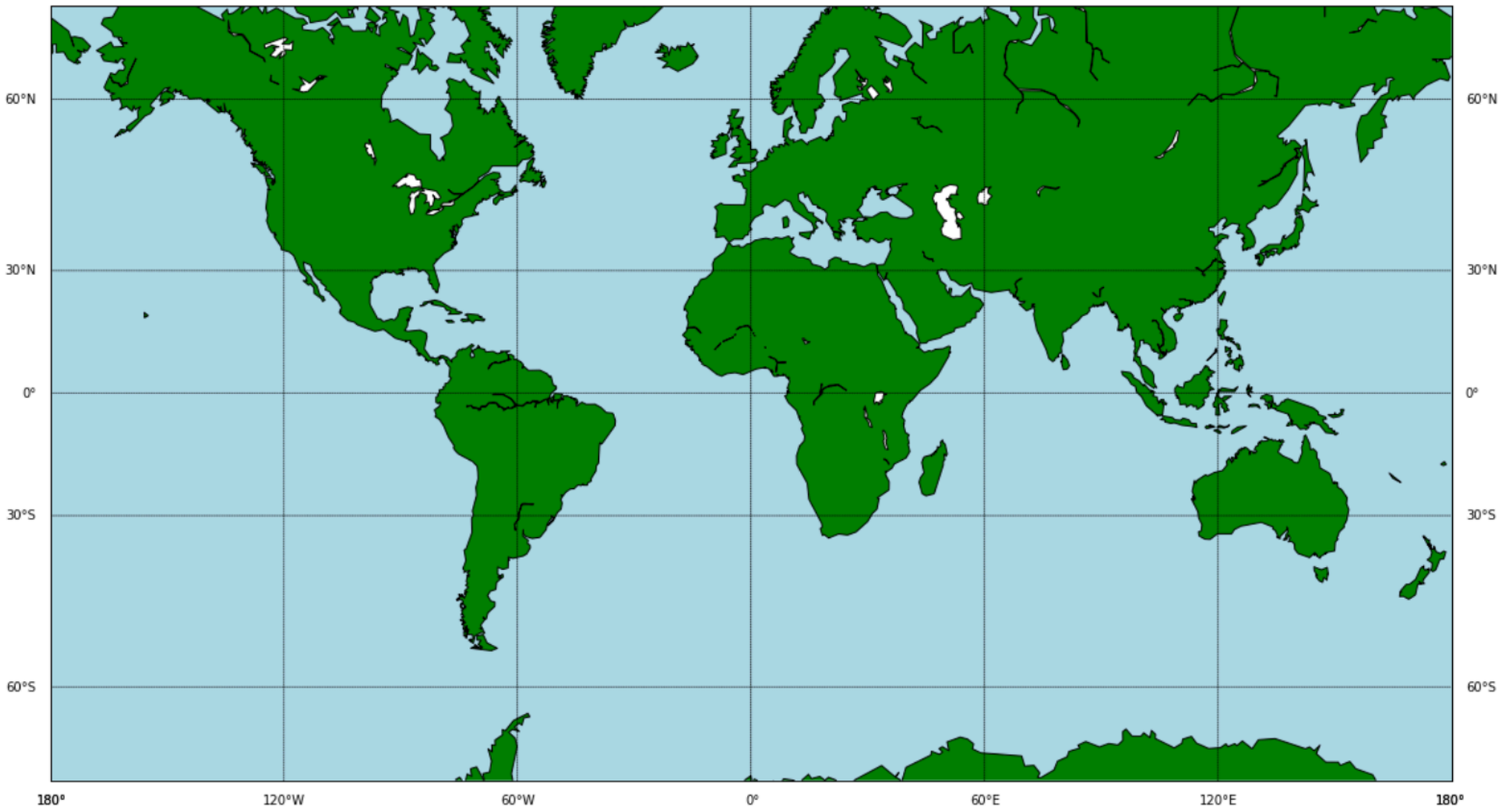
# Label your maps

In [6]:
```
plt.figure(figsize=(20,15)) # create a new figure with size (16,15)

# make a map instance called 'm', pay attention to the boundaries of the map
m = Basemap(projection='merc',llcrnrlat=-70,urcrnrlat=70,llcrnrlon=-180,urcrnrlon=180)
m.drawcoastlines(); # put on the coastlines

m.drawparallels(np.arange(-90.,91.,30.), labels=[True,True,False,False]) # plot the latitudes (
m.drawmeridians(np.arange(-180.,181.,60.),labels=[False,False,True,True]); # same for longitude

plt.show() # show your map!
```



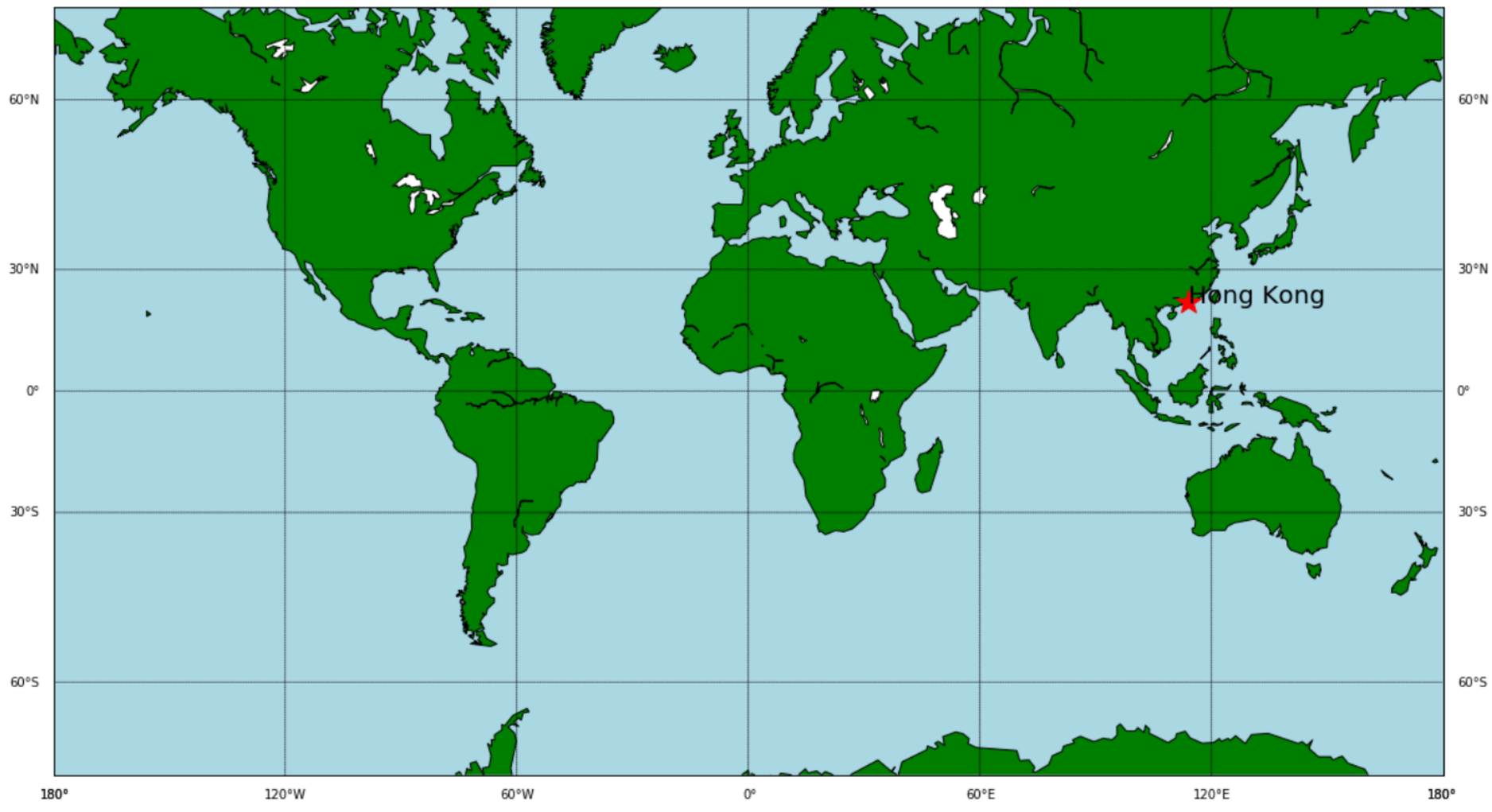
# Color your maps

In [7]:
```
plt.figure(figsize=(20,15)) # create a new figure with size (16,15)

# make a map instance called 'm', pay attention to the boundaries of the map
m = Basemap(projection='merc',llcrnrlat=-70,urcrnrlat=70,llcrnrlon=-180,urcrnrlon=180)
m.drawcoastlines(); # put on the coastlines

m.drawparallels(np.arange(-90.,91.,30.), labels=[True,True,False,False]) # plot the latitudes (
m.drawmeridians(np.arange(-180.,181.,60.),labels=[False,False,False,True]); # same for longituc

m.fillcontinents(color='green')
m.drawmapboundary(fill_color='lightblue');

plt.show() # show your map!
```



# Plot data points on your map

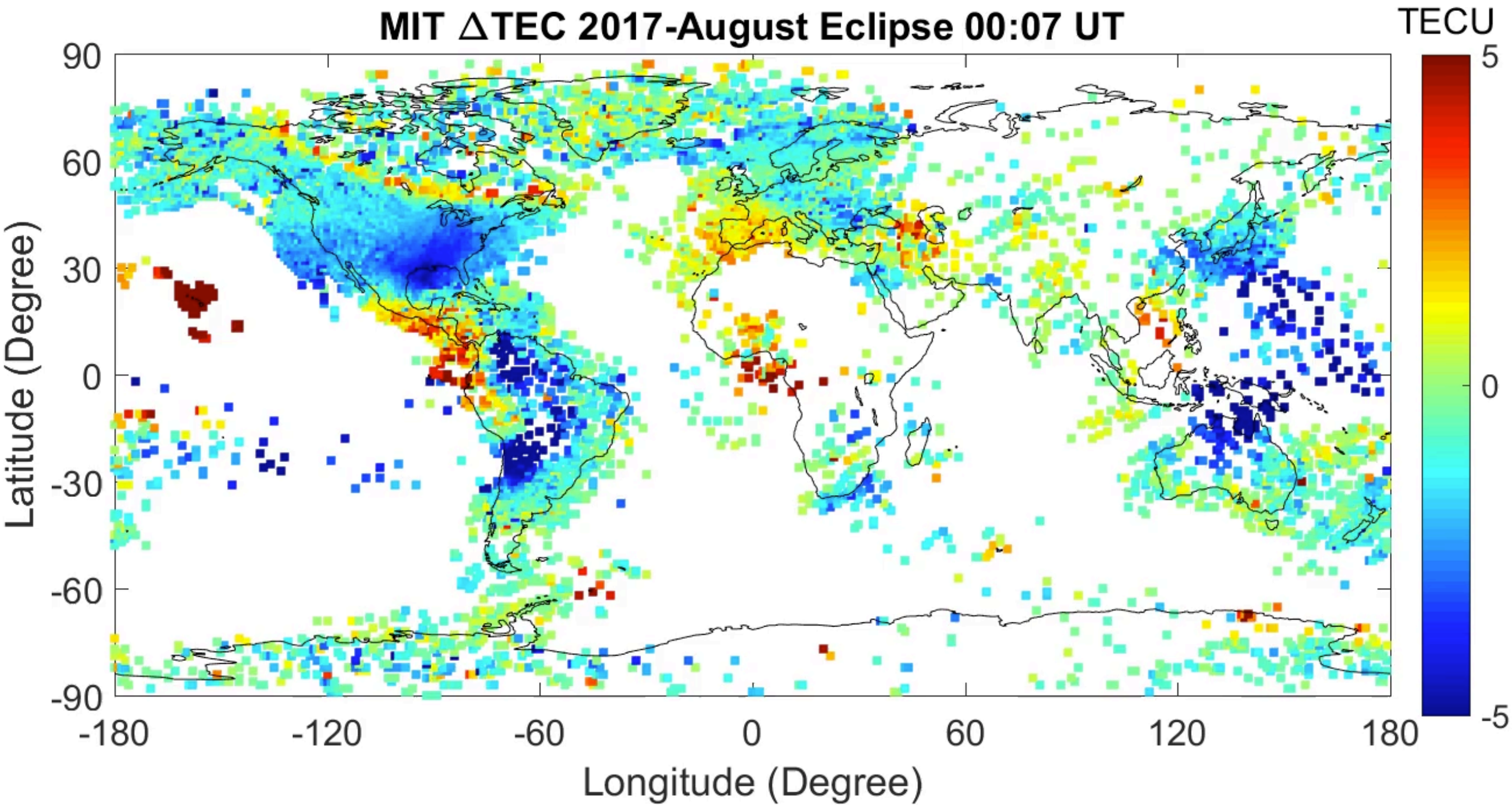
```
# set the lat and long of Hong Kong
HK_lat=22.3964   # latitude
HK_lon=114.1095  # longitude
x,y = m(HK_lon,HK_lat) # this will convert the lat,lon info to map coordinates x,y

# Plot the location of Hong Kong on your map and annotate it
plt.plot(x,y,'r*',markersize=20); # markersize sets the size of the marker
plt.text(x,y,'Hong Kong',fontsize=20) # annotate the marker
plt.show() # show your map!
```




Now you've got your first map created. To make your map more useful, you also want to put something **ON the map**. That's quite straightforward:

- Step 1: take your latitude and longitude (or arrays of them) and **convert them to map coordinates**.
- Step 2: **plot** just like any ordinary matplotlib plot.

# Example of Mercator Projection of Global GPS TEC Maps



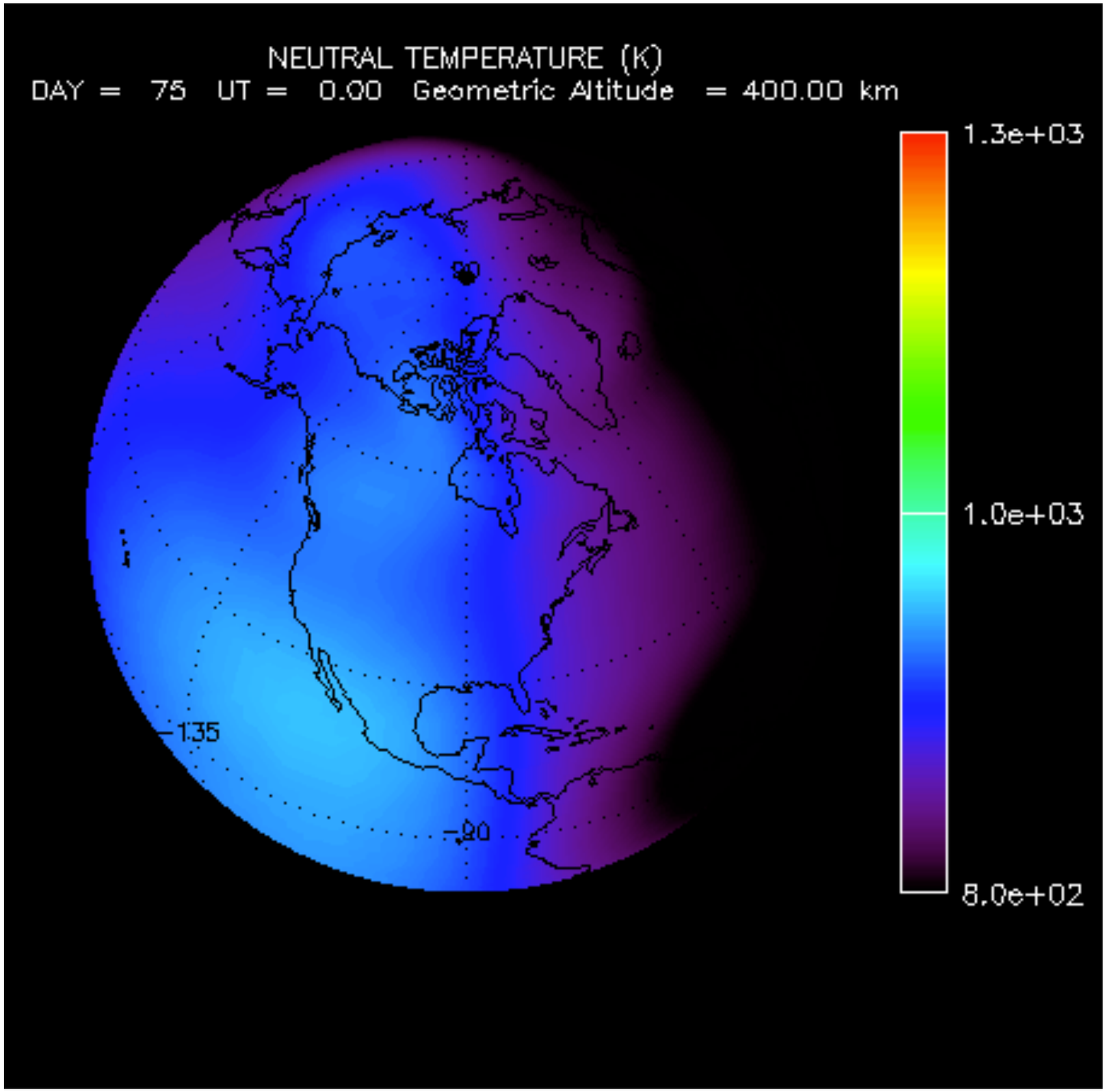
# Orthographic Projection of the world map (ortho)


```
plt.figure(figsize=(10,10)) # create a new figure with size (16,15)

m = Basemap(projection='ortho',lon_0=100,lat_0=30)
m.drawcoastlines();

m.drawparallels(np.arange(-90.,91.,30.))
m.drawmeridians(np.arange(0.,361.,60.))
m.drawmapboundary()

m.fillcontinents(color='orange')
m.drawmapboundary(fill_color='lightblue');

# set the lat and long of Hong Kong
HK_lat=22.3964
HK_lon=114.1095  # takes the longitude and converts to 0=>360
x,y=m(HK_lon,HK_lat) # this will convert the lat,lon info to map coordinat
plt.plot(x,y,'r*',markersize=20); # markersize sets the size of the marker

text_lat = HK_lat
text_lon = HK_lon + 3 # let's shift the text annotation 3 degrees eastward
text_x,text_y=m(text_lon,text_lat) # this will convert the lat,lon info to
plt.text(text_x,text_y,'Hong Kong',fontsize=20) # put on the text annotati

plt.show()
```

The Mercator is a nice classical map, but it sure does distort the map at high latitudes.

Another type of map projection is the orthographic projection which is much less distorted. The downside to this projection is that you cannot see the whole globe at once. To create an orthographic map, you initialize a map instance with the projection '**ortho**' and arguments **lon_0** and **lat_0** - the central longitude and latitude- instead of the lower left/ upper right corner syntax.

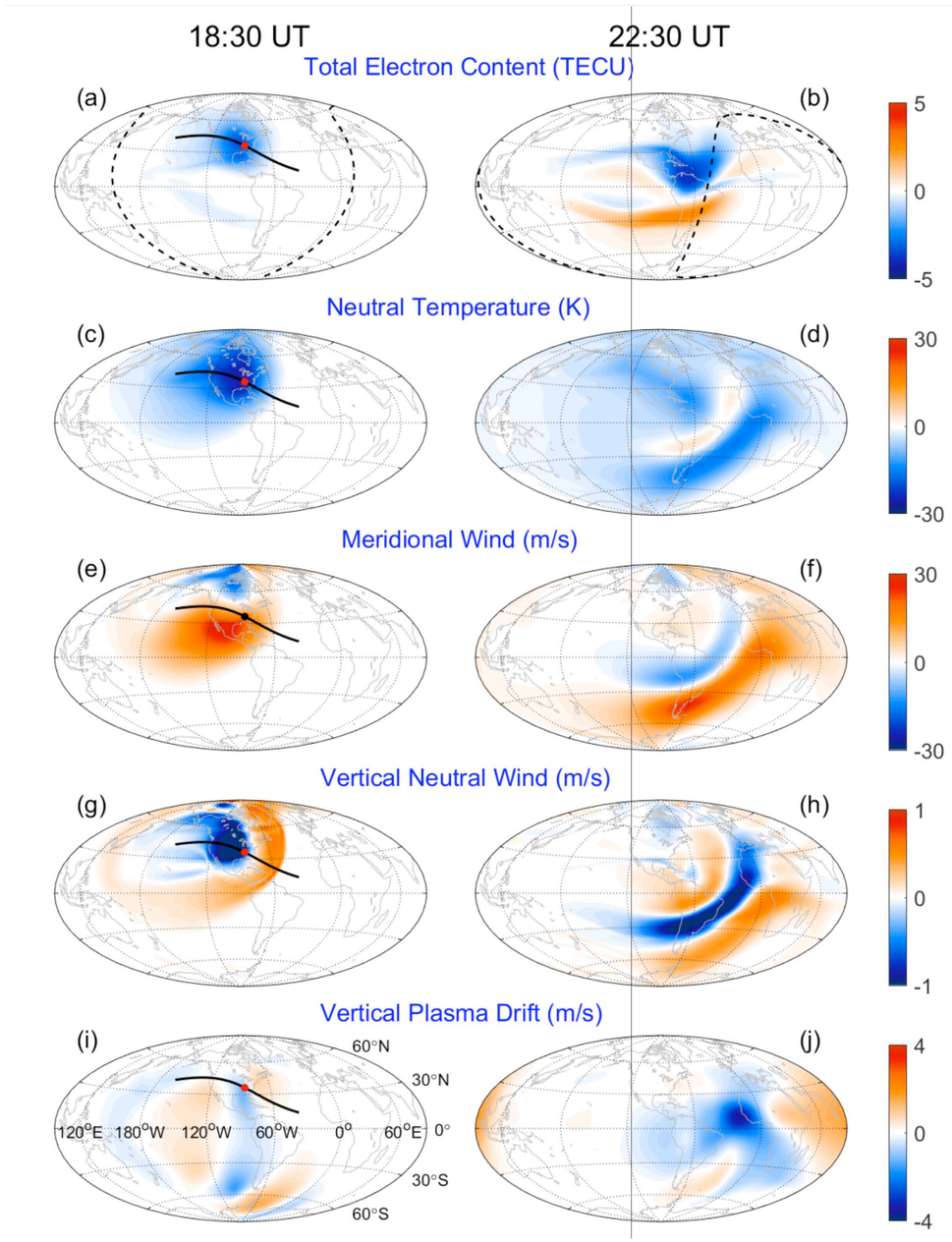
# Example of Orthographic Projection of global upper atmosphere



# Hammer Projection (hammer)

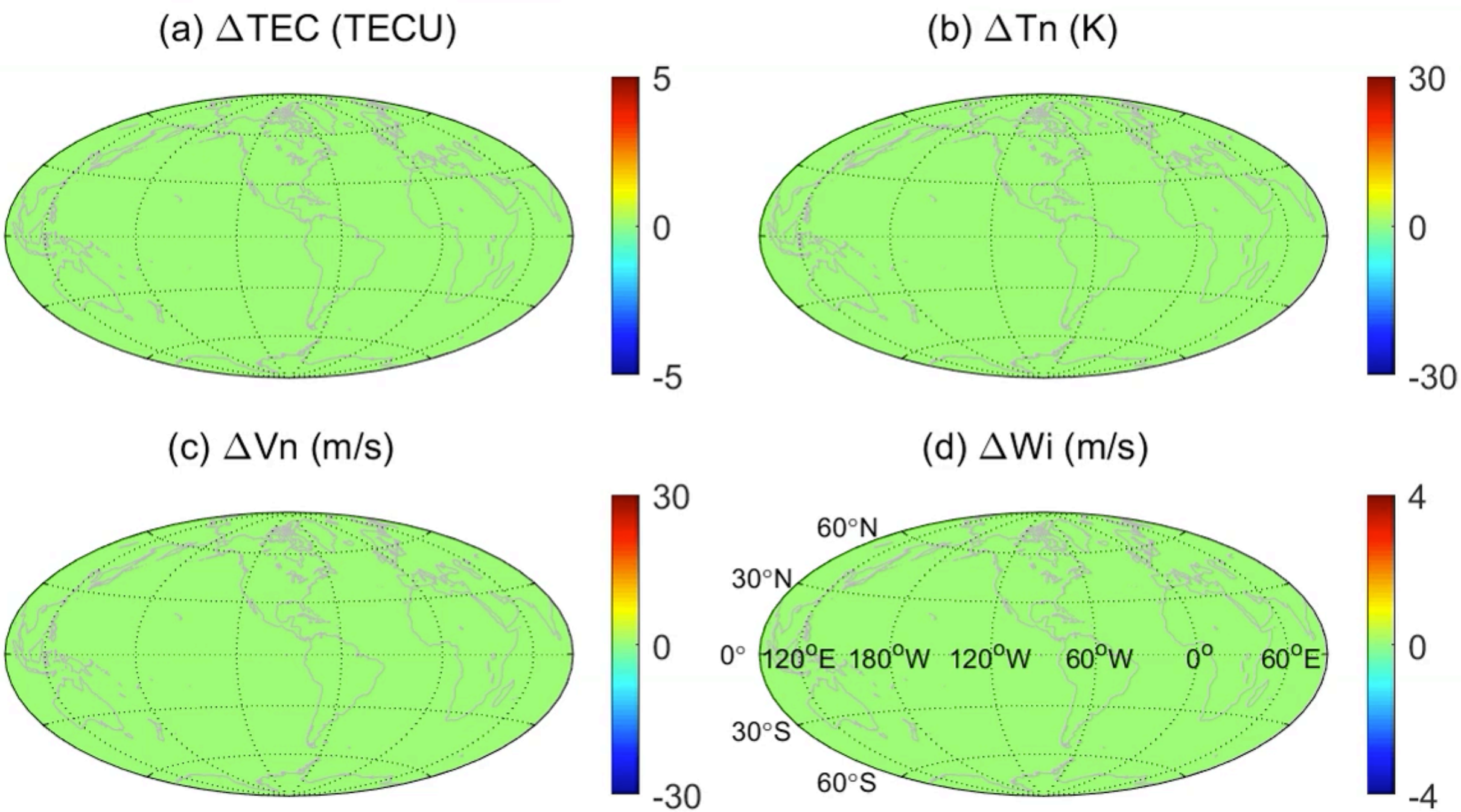
```
plt.figure(figsize=(16,10)) # create a new figure with size (16,15)


m = Basemap(projection='hammer',lon_0=-200) # make a Hammer map object
m.drawcoastlines()
m.fillcontinents(color='yellow',lake_color='lightblue')
m.drawparallels(np.arange(-90.,120.,30.))
m.drawmeridians(np.arange(0.,420.,30.))
m.drawmapboundary(fill_color='lightblue')

# set the lat and long of Hong Kong
HK_lat=22.3964
HK_lon=114.1095  # takes the longitude and converts to 0=>360
x,y=m(HK_lon,HK_lat) # this will convert the lat,lon info to map coordinat
plt.plot(x,y,'r*',markersize=20); # markersize sets the size of the marker

text_lat = HK_lat
text_lon = HK_lon + 3 # let's shift the text annotation 3 degrees eastward
text_x,text_y=m(text_lon,text_lat) # this will convert the lat,lon info to
plt.text(text_x,text_y,'Hong Kong',fontsize=20) # put on the text annotati

plt.show()
```


One more global scale example of a map projection is the Hammer projection (one of the most popular). This is always a global map centered on the equator, so all you need to specify is the central longitude (with the **lon_0** argument).



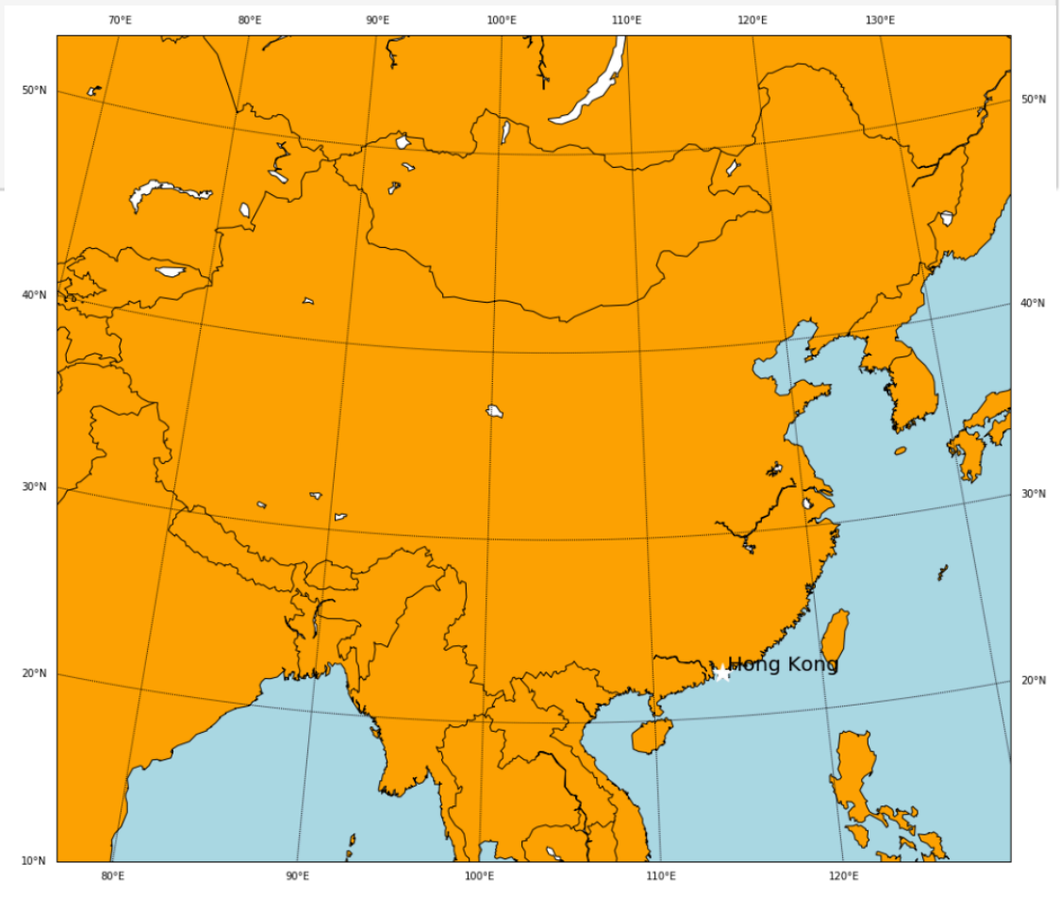

Dang et al., [2018], GRL

# Example of Hammer Projection of global upper atmosphere during eclipse



# Lambert conformal conic projections (lcc)

```
plt.figure(figsize=(20,15)) # create a new figure with size (16,15)

# make a map instance called 'm', pay attention to the boundaries of the map
m = Basemap(projection='lcc',llcrnrlat=10,urcrnrlat=53,llcrnrlon=77,urcrnrlon=140,lat_0=23,lon_0=104,resolution='l')
m.drawcoastlines(); # put on the coastlines

m.drawparallels(np.arange(-90.,91.,10.), labels=[True,True,False,False]) # plot the latitudes (parallels) from -90 to +
m.drawmeridians(np.arange(-180.,181.,10.),labels=[False,False,True,True]); # same for longitudes (meridians) but with 6

m.fillcontinents(color='orange')
m.drawmapboundary(fill_color='lightblue');
m.drawcountries(color='black',linewidth=1)

# set the lat and long of San Diego
HK_lat=22.3964
HK_lon=114.1095  # takes the longitude and converts to 0=>360
x,y=m(HK_lon,HK_lat) # this will convert the lat,lon info to map coordinates x,y
plt.plot(x,y,'w*',markersize=20); # markersize sets the size of the marker
plt.text(x+30000,y+20000,'Hong Kong',fontsize=20,color='black')

plt.show() # show your map!
```




## Function/options used:

`projection = ‘lcc’`

`llcrnrlat, urcrnrlat, llcrnrlon, urcrnrlon`

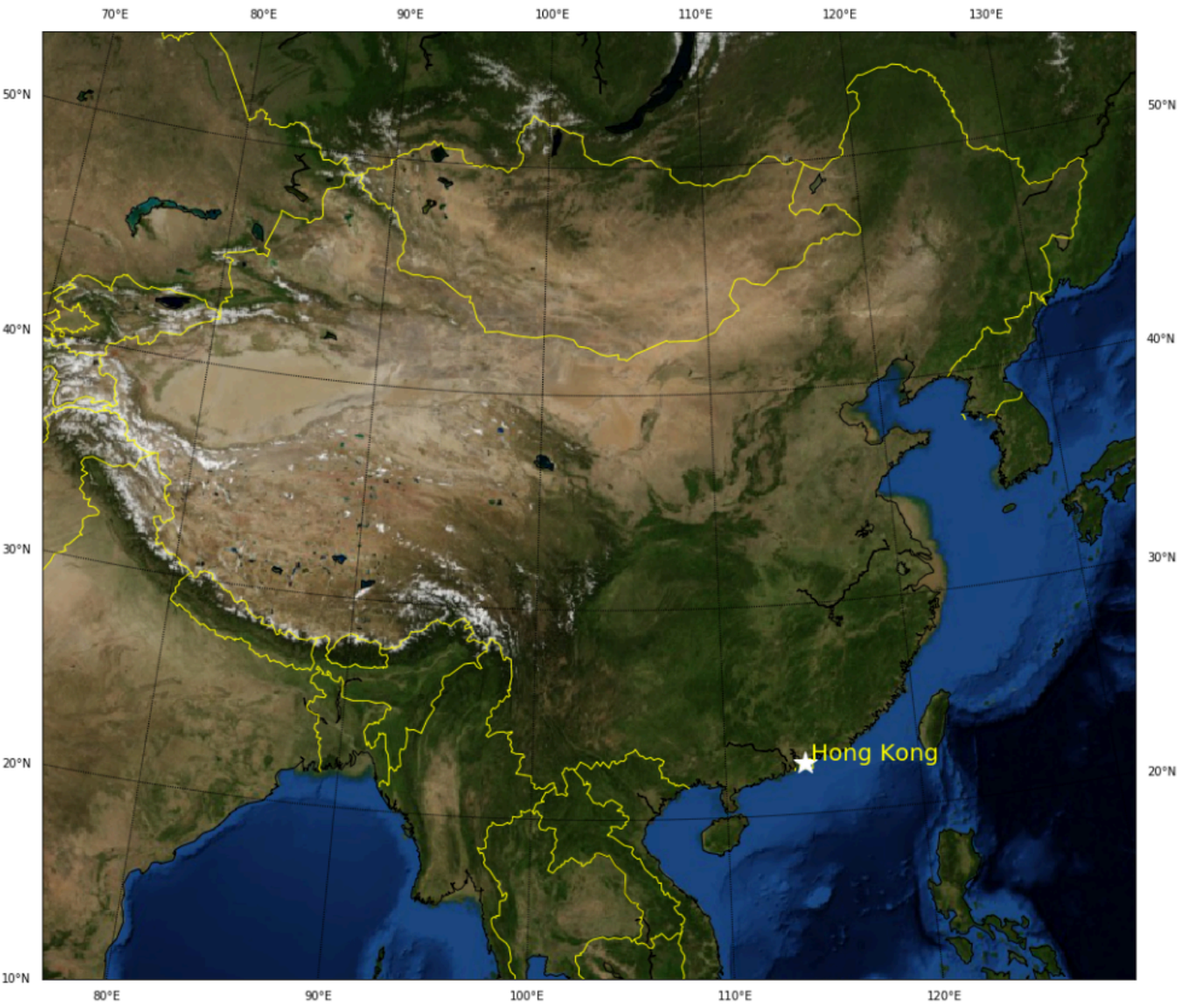
`fillcontinents()`

`drawmapboundary()`

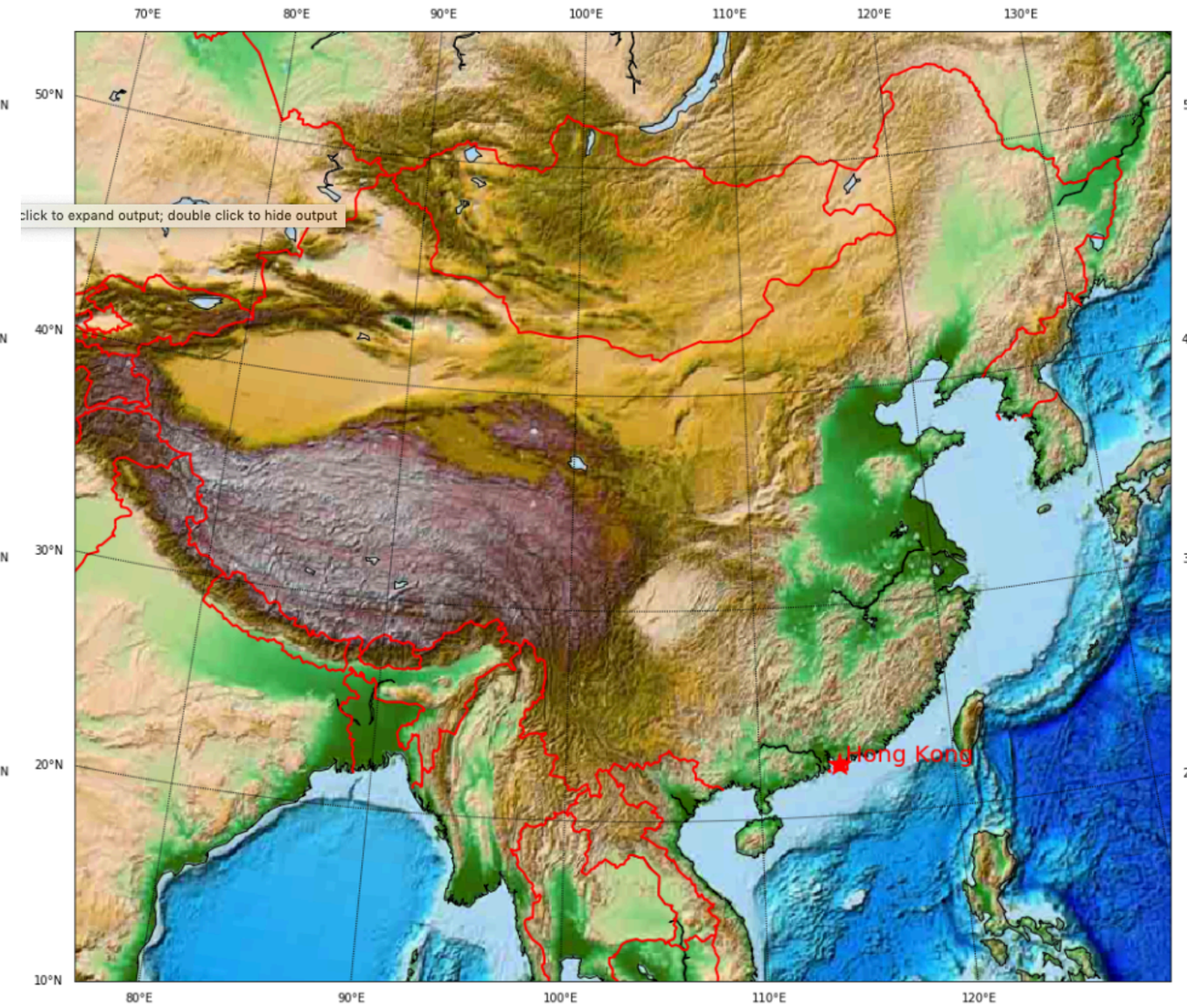
`drawcountries()`

## use the `bluemarble()` and `etopo()` view

```
m.bluemarble() # NASA's blue Marble image
```




```
m.etopo() # NASA's
```

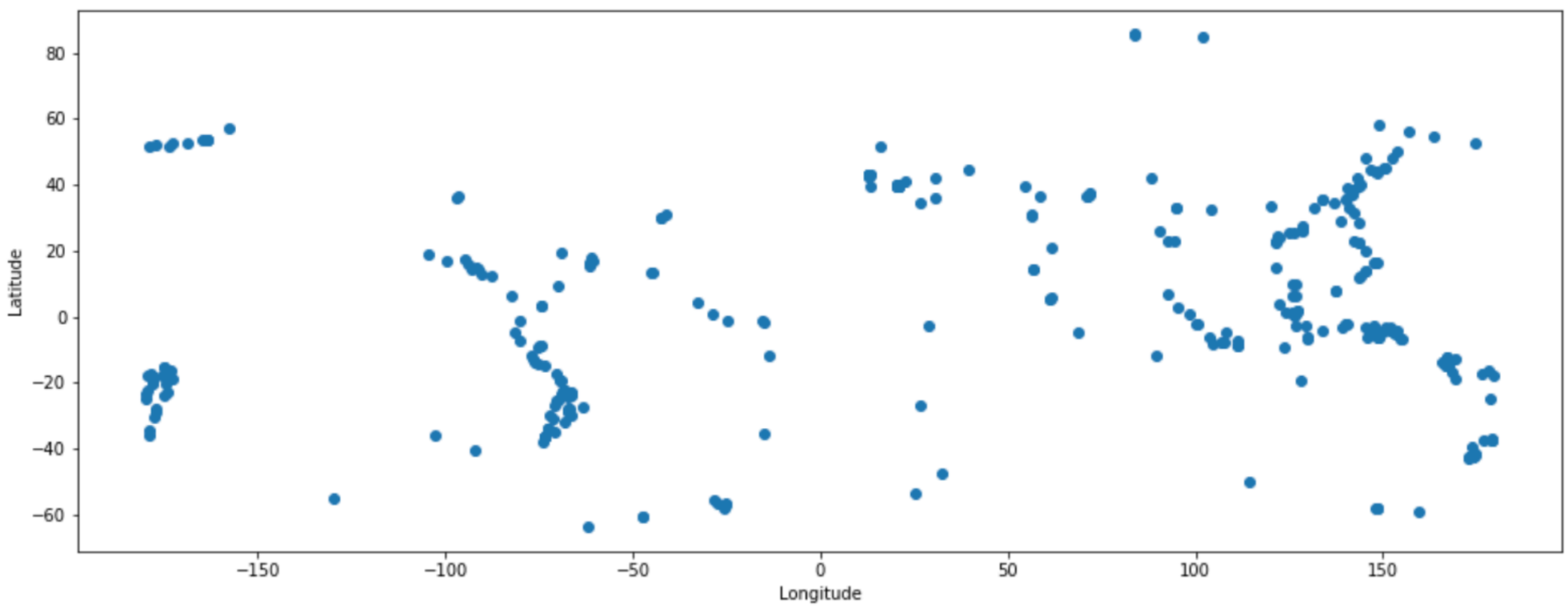



The lcc projection is the most popular projection for **local** maps

# Now let's do science

Recall the earthquake data sets we looked yesterday, plotting the locations of earthquakes on a lon-lat figure gives the following scatter plot:

```
In [21]: EQ = np.loadtxt('datasets/earthquake.csv', delimiter=',')

         lat = EQ[:,0]
         lon = EQ[:,1]
         dep = EQ[:,2]
         mag = EQ[:,3]

         plt.figure(figsize=(16,6))
         plt.scatter(lon,lat)
         plt.xlabel('Longitude'),plt.ylabel('Latitude')
         plt.show()
```




**Everything is correct, but does the plot make sense?**

# Now put everything together

Now let's put the global map (Mercator projection of the globe) together with earthquake data points:

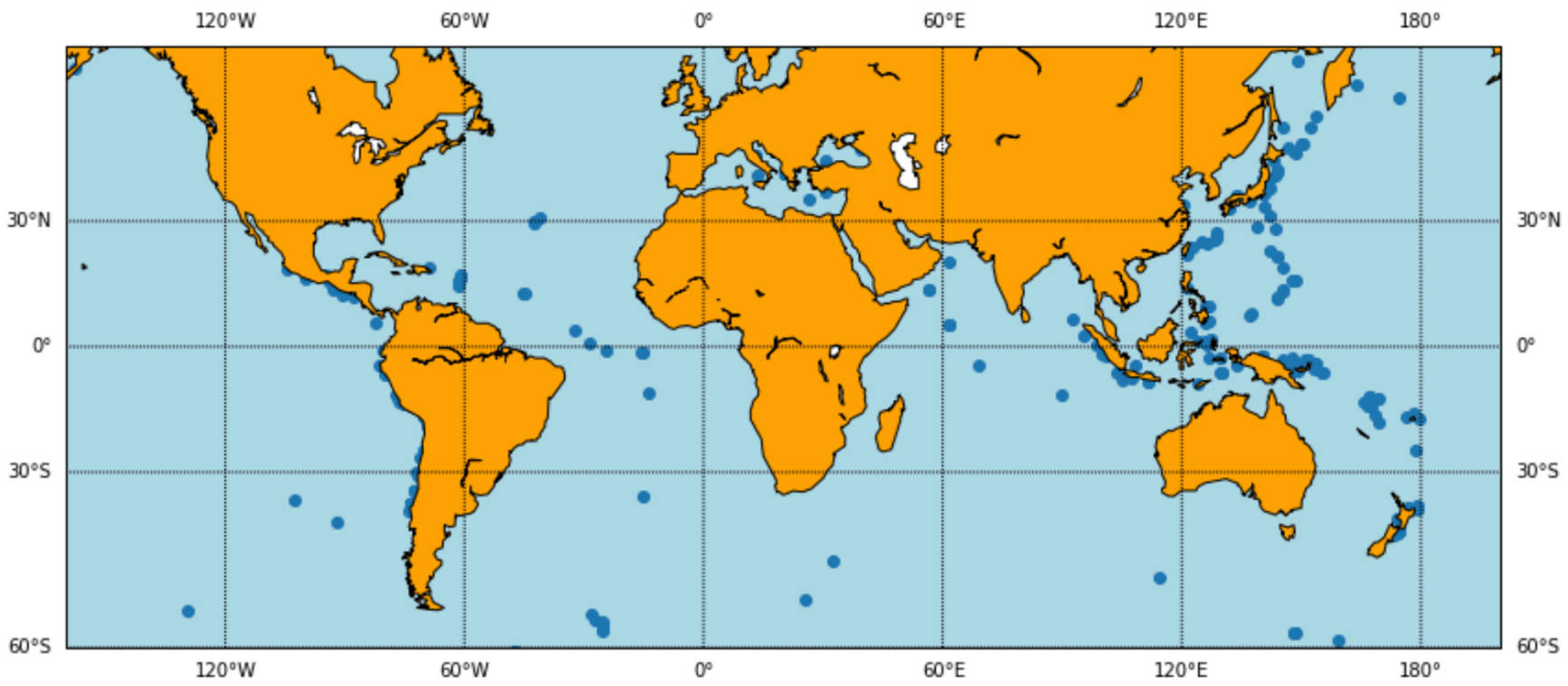
In [26]:
```
EQ = np.loadtxt('datasets/earthquake.csv', delimiter=',')
disa = EQ[:,3]>7.5

lat = EQ[:,0]
lon = EQ[:,1]
dep = EQ[:,2]
mag = EQ[:,3]

plt.figure(figsize=(16,6))
m = Basemap(projection='merc',llcrnrlat=-60,urcrnrlat=60,\
            llcrnrlon=-160,urcrnrlon=200) # make a map instance 'm'
m.drawcoastlines(); # put on the coastlines
m.drawparallels(np.arange(-90.,91.,30.), labels=[True,True,False,False]) # plot the latitudes (
m.drawmeridians(np.arange(-180.,181.,60.),labels=[False,False,True,True]); # same for longitude
m.fillcontinents(color='orange')
m.drawmapboundary(fill_color='lightblue');
x, y = m(lon,lat)
plt.scatter(x,y)
plt.show()
```

load data

Slicing 2-D array

Map!

Plot data



# Now let’s make maps!