IRIS 2022 Seismology Skill Building Workshop OSL

Quiz navigation

1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 18
19 20 21 22 23 24
25 26 27 28 29 30
31 32 33 34 35 36
37

Show one page at a time
Finish review

Started on	Wednesday, August 17, 2022, 10:31 AM
State	Finished
Completed on	Friday, August 19, 2022, 6:59 PM
Time taken	2 days 8 hours
Marks	31.48/37.00
Grade	**85.09** out of 100.00

Question 1
Correct
1.00 points out of 1.00
Flag question

In this activity, you will get a chance to start using ObsPy, a Python library for seismology intended to facilitate the development of seismological software packages and workflows. The goal is to utilize these abilities to provide a bridge for seismology into the larger scientific Python ecosystem.

To get started using ObsPy, go ahead and start python so we can interact with the Python command line. Once you are in Python, which command would load the ObsPy library?

Select one:
- a. obspy import
- b. from obspy import
- c. obspy load
- d. from obspy load
- e. import obspy ✓ Correct. Go ahead and run this command now at the Python command line.
- f. load obspy

Check

Correct
Marks for this submission: 1.00/1.00.

Question 2
Correct
1.00 points out of 1.00
Flag question

During this introduction we will look at some seismic data from April 17, 2012 because there were some sizable earthquakes on that date. ObsPy has many different tools that are helpful to seismologists. We will start with one of the most basic (and most important), handling date and time information. It does not take long as a seismologist to get frustrated with how many datasets and software represent date and time information with different formats. I would guess some of you have been a little bothered with how precise the FetchEvent format is, and then the format for other commands want a different, but still precise, format. ObsPy has created the UTCDateTime() function to help with this. Take a minute to have a look at the manual for this function:

https://docs.obspy.org/packages/autogen/obspy.core.utcdatetime.UTCDateTime.html

As you can quickly see, this function is helpful for converting between many different date and time formats. Let's examine how to use this function. Trying each of these on the Python command prompt, which of the following actually gives you date information instead of an error?

Note: Since this assignment will be working at the command line, you may find it helpful to try running any commands you are considering choosing as an answer to see what happens before selecting them as an answer. "Try before you buy!"

Select one or more:
- a. UTCDateTime(0)
- b. obspy.UTCDateTime("0")
- c. obspy UTCDateTime(0)
- d. obspy.UTCDateTime(0) ✓ This is the only correct answer.
- e. UTCDateTime("0")
- f. obspy UTCDateTime("0")

Check

Correct
Marks for this submission: 1.00/1.00.

Question 3
Correct
0.67 points out of 1.00
Flag question

The correct answer to the previous question is based on how we loaded the obspy library. In essence, question 1 loaded the whole library, so then we need to specify individual functions as part of that library with the *library.function()* format. However, Python also allows you to read individual functions so they can be called by name only. Try the following:

```
from obspy import UTCDateTime
```

Now try each of these on the Python command prompt again, and identify which of the following actually gives you date information instead of an error.

Select one or more:
- a. UTCDateTime(0) ✓ 1 of 2 correct answers.
- b. obspy UTCDateTime(0)
- c. UTCDateTime("0")
- d. obspy UTCDateTime("0")
- e. obspy.UTCDateTime(0) ✓ 1 of 2 correct answers.
- f. obspy.UTCDateTime("0")

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.67/1.00**.

Question 4
Correct
0.67 points out of 1.00
Flag question

Let's take a minute to figure out what is happening when we input the number 0 to the UTCDateTime function. When you enter the command, the interactive Python provides some numbers about the date and time information associated with the input. However, I find that information is clearer when you use the print() function on the UTCDateTime output:

```
print(UTCDateTime(0))
```

What is the date and time interpreted by UTCDateTime when we input this number?

Select one:
- a. midnight on the beginning of January 1, 1970 ✓
- b. midnight at the end of January 1, 1970
- c. 1:01 am on January 1, 1970
- d. noon on January 1, 1970
- e. 1 am on January 1, 1970

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.67/1.00**.

Question 5
Correct
0.67 points out of 1.00
Flag question

The answer to the previous question reveals that when given a number, UTCDateTime interprets it as so-called "epoch time". This is the number of seconds since the beginning of 1970. Although that might seem very arbitrary, it has to do with the history of when the Unix operating system was created (note that Linux is an open source version of Unix). Much like when it was created, the epoch time is a very useful way to do math on date and time information since it is a single number instead of a combination of numbers (year, month, day, hour, minute, second) with special conditions (i.e., minutes only go up to a value of 60 before incrementing the hour category).

For example, what number (instead of zero) can we input into the UTCDateTime function to go exactly one day into the future from the epoch time represented by zero?

Answer: 86400 ✓

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.67/1.00**.

Question 6
Correct
0.40 points out of 1.00
Flag question

I indicated earlier that we will look at earthquakes on April 17, 2012 in this assignment. Which of the following would be acceptable ways to input this date into the UTCDateTime function? You will likely need to review manual link provided in Question 2 again to see which input formats are allowed.

Select one or more:
- a. UTCDateTime("20120417") ✓ 1 of 5 correct answers.
- b. UTCDateTime("April 17, 2012")
- c. UTCDateTime("2012/04/17") ✓ 1 of 5 correct answers.
- d. UTCDateTime("2012-04-17") ✓ 1 of 5 correct answers.
- e. UTCDateTime("04/17/2012")
- f. UTCDateTime("2012/04/17T00:00:00") ✓ 1 of 5 correct answers.
- g. UTCDateTime("04-17-2012T00:00:00")
- h. UTCDateTime("2012-04-17T00:00:00") ✓ 1 of 5 correct answers.

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.40/1.00**.

Question 7
Correct
1.00 points out of 1.00
Flag question

Now let's store the output of the UTCDateTime function in a variable called **dt** and print the date and time 1 day later. Which of the following pairs of commands would accomplish this?

Select one:
- a. dt = UTCDateTime("2012-04-17T00:00:00")
  print (dt+1)
- b. dt = UTCDateTime("2012-04-17T00:00:00")
  print (dt+86400) ✓ Correct. Be sure to run these two commands now.
- c. dt = UTCDateTime("2012-04-17T00:00:00")
  print (dt+24)
- d. dt = UTCDateTime(2012-04-17T00:00:00)
  print (dt+24)
- e. dt = UTCDateTime(2012-04-17T00:00:00)
  print (dt+1)
- f. dt = UTCDateTime(2012-04-17T00:00:00)
  print (dt+86400)

Check

Correct
Marks for this submission: 1.00/1.00.

Question 8
Correct
1.00 points out of 1.00
Flag question

Once you store the date and time as a variable, you can access "attributes" of that variable. Taking a look at the manual for UTCDateTime again, how would you print which day of the week that April 17, 2012 was?

Select one:
- a. print (dt(dayofweek))
- b. print (dt(day))
- c. print (dt.day)
- d. print (dt.week)
- e. print (dt.weekday) ✓ Correct. Be sure to run this command now.
- f. print (dt(week))
- g. print (dt.dayofweek)
- h. print (dt(weekday))

Check

Correct
Marks for this submission: 1.00/1.00.

Question 9
Correct
1.00 points out of 1.00
Flag question

Which day of the week was it? You might need to review the manual page to interpret the output.

Select one:
- a. Monday
- b. Tuesday ✓
- c. Wednesday
- d. Friday
- e. Sunday
- f. Saturday
- g. Thursday

Check

Correct
Marks for this submission: 1.00/1.00.

Question 10
Correct
1.00 points out of 1.00
Flag question

The seismic data for this day is available from the ObsPy website, but to be able to retrieve the seismic data, we need to know the Julian day (the day of the year). The filename will have the year and the Julian day with only a period in between them (no spaces) - How would we print this? You will probably want to try these in Python to see which it allows.

Select one:
- a. print (dt.year,".",dt.julday,sep="") ✓ Correct. Be sure to run this command now.
- b. print (dt.year "." dt.julday)
- c. print (dt.year "." dt.julian)
- d. print (dt.year "." dt.julian sep="")
- e. print (dt.year,".",dt.julian)
- f. print (dt.year,".",dt.julian,sep="")
- g. print (dt.year,".",dt.julday)
- h. print (dt.year "." dt.julday sep="")

Check

Correct
Marks for this submission: 1.00/1.00.

Question 11
Correct
1.00 points out of 1.00
Flag question

Next we will use the read() function to read a seismogram into Python. As we discussed earlier, Python allows you to load individual functions from a library so they can be called by name only using a command like this (but do NOT run it yet!):

```
from obspy import read
```

If you did NOT run this command, how could you still access the ObsPy read() function based on commands we have already run this assignment?

Select one:
- a. from obspy read()
- b. read()
- c. obspy.read() ✓ Correct. Now go ahead and run the `from obspy import read` command.
- d. obspy(read())

Check

Correct
Marks for this submission: 1.00/1.00.

Question 12
Correct
1.00 points out of 1.00
Flag question

Next we will load an example half-day-long seismogram from April 17, 2012 recorded at station BFO. The URL to where you can retrieve this file is:

https://examples.obspy.org/GR.BFO..LHZ.2012.108

ObsPy has 3 main object types: stream, catalog, and inventory. The stream object is designed for collections of seismograms.

How would you use the read() function to load this seismogram into a Python stream object that we will name **st**? Again, you will probably want to try these in Python to see which it allows, but be careful to make sure the command actually does what I am asking for.

Select one:
- a. st = read(https://examples.obspy.org/GR.BFO..LHZ.2012.108)
- b. read(https://examples.obspy.org/GR.BFO..LHZ.2012.108)
- c. st = read("https://examples.obspy.org/GR.BFO..LHZ.2012.108") ✓ Correct. Make sure to run this command now.
- d. read("https://examples.obspy.org/GR.BFO..LHZ.2012.108")
- e. st.read("https://examples.obspy.org/GR.BFO..LHZ.2012.108")
- f. st.read(https://examples.obspy.org/GR.BFO..LHZ.2012.108)

Check

Correct

Marks for this submission: 1.00/1.00.

Question 13
Correct
0.67 points out of 1.00
Flag question

Now that you have created a stream object, take a minute to look at the manual to see some of the "methods" that can be applied to it:
https://docs.obspy.org/packages/autogen/obspy.core.stream.Stream.html
In essence, there are a wide array of actions you can take with a seismogram in ObsPy, each of which can be clicked on that page to learn more about the different methods. To start with, we will simply plot this seismogram, which is very easy using the plot() function available for stream objects in ObsPy. Here is the manual page for it:
https://docs.obspy.org/packages/autogen/obspy.core.stream.Stream.plot.html
How would we perform this on the st object?

Select one:
- a. obspy.plot()
- b. plot(st)
- c. st.obspy.plot()
- d. st.plot() ✓ Correct. Make sure to run this command now.
- e. plot()
- f. obspy.plot(st)
- g. obspy.st.plot()

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.67/1.00**.

Question 14
Correct
0.75 points out of 1.00
Flag question

Hopefully you are able to view the seismogram in a plot window with the command from the previous question. Note that the x and y coordinates (time and amplitude) of the cursor location are shown in the lower right of the plot window, which can allow you to examine the maximum amplitude of the earthquakes in this seismogram. You can also use the magnifying glass button in the plot window to zoom in on parts of the seismogram and then the back arrow button to return to the previous zoom level.
Another method of the stream object to quickly identify the maximum amplitude is the following (see the link in the previous question for more details on this method):
```
st.max()
```
Using this method, what is the largest absolute amplitude in the full seismogram?

Answer: 22651 ✓

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.75/1.00**.

Question 15
Correct
1.00 points out of 1.00
Flag question

Let's make an estimate of the background noise level. To do this, we can create a copy of the seismogram (st2) and then trim it down to just the first hour using the copy() and trim() methods of the Stream object (see link to Stream manual 2 question ago):
```
st2 = st.copy()
st2.trim(dt,dt+3600)
st2.plot()
```
Review how the seismogram looks to make sure it is the first hour and shows background noise. Then exit the plot and use the max() method again to get the maximum amplitude of the background noise. Using this method, what is the largest absolute amplitude in the first hour of the seismogram?

Answer: 2415 ✓

Check

Correct. Note that noise is about a factor of 10 smaller than the earthquake.
Correct
Marks for this submission: 1.00/1.00.

Question 16
Correct
1.00 points out of 1.00
Flag question

Another commonly used function available for stream objects is the filter() function. Here is the manual page for it:
https://docs.obspy.org/packages/autogen/obspy.core.stream.Stream.filter.html
How would we filter the seismogram to only allow frequencies less than 0.1 Hz?

Select one:
- a. st.filter(freq<0.1)
- b. filter(st, freq<0.1)
- c. filter(st, lowpass, freq=0.1)
- d. filter(st, f<0.1)
- e. st.filter(f<0.1)
- f. filter(st, "lowpass", f=0.1)
- g. st.filter("lowpass", freq=0.1) ✓ Correct. Make sure to run this command now. You will need to close the plot window before you can run another command, which can be accomplished by clicking the X button in the upper right of the plot window.
- h. filter(st, "lowpass", freq=0.1)
- i. st.filter("lowpass", f=0.1)
- j. st.filter(lowpass, freq=0.1)

Check

Correct
Marks for this submission: 1.00/1.00.

Question 17
Correct
1.00 points out of 1.00
Flag question

Go ahead and plot the seismogram to see how it has changed after filtering. Then close the plot and use the max() method to identify the maximum amplitude. Using this method, what is the largest absolute amplitude in the filtered full seismogram?

Answer: 22424.37627808705 ✓

Check

Correct
Marks for this submission: 1.00/1.00.

Question 18
Correct
1.00 points out of 1.00
Flag question

Now apply the filter to the seismogram trimmed to the first hour. What is the largest absolute amplitude you observe in the first hour of the seismogram after filtering?

Answer: 835.6201122793359 ✓

Check

Correct
Marks for this submission: 1.00/1.00.

Question 19
Correct
1.00 points out of 1.00
Flag question

Now we will use the plot() function but with the dayplot option selected to let us view up to a full day of data (we only have half a data loaded right now). Reviewing the plot() manual from a few questions ago, which of these would accomplish the dayplot view?

Select one:
- a. st.plot(type=dayplot)
- b. dayplot()
- c. st.plot(type="dayplot") ✓ Correct. Run this command now to see how the plot changes.
- d. st.plot("dayplot")
- e. plot(type=dayplot)
- f. st.dayplot()
- g. plot(type="dayplot")
- h. plot("dayplot")

Check

Correct
Marks for this submission: 1.00/1.00.

Question 20
Correct
0.33 points out of 1.00
Flag question

This plot is a little hard to read because each line is only 15 minutes, so larger/longer earthquake signals show up on multiple lines. You can adjust amount of time plotted per line to 60 minutes. Which of the following would accomplish this?

Select one:
- a. st.plot(type="dayplot", time_interval=3600)
- b. st.plot(type="dayplot", time=3600)
- c. st.plot(type="dayplot", time_offset=3600)
- d. st.plot(type="dayplot", time=60)
- e. st.plot(type="dayplot", time_interval=60)
- f. st.plot(type="dayplot", interval=60) ✓ Correct. Run this command now to see how the plot changes.
- g. st.plot(type="dayplot", time_offset=60)
- h. st.plot(type="dayplot", interval=3600)

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.33/1.00**.

Question 21
Correct
1.00 points out of 1.00
Flag question

At this point you can hopefully see there are some sizable surface waves in this seismogram which are likely from more than one earthquake. A very nice feature of ObsPy is that it can plot the origin times of catalog earthquakes in the seismogram view to help identify what is responsible for the seismic waves visible in the seismogram. Which of the following would accomplish this?

Select one:
- a. st.plot(type="dayplot", interval=60, events={min_mag:6.0})
- b. st.plot(type="dayplot", interval=60, events={"min_magnitude": 6.0}) ✓ Correct. Run this command now to see how the plot changes.
- c. st.plot(type="dayplot", interval=60, min_magnitude=6.0)
- d. st.plot(type="dayplot", interval=60, events={"magnitude": 6.0-10.0})
- e. st.plot(type="dayplot", interval=60, events=6.0)
- f. st.plot(type="dayplot", interval=60, events={min_magnitude=6.0})

Check

Your answer is correct.
Correct
Marks for this submission: 1.00/1.00.

Question 22
Correct
1.00 points out of 1.00
Flag question

Next we should learn more about these events. We can obtain a catalog from the FDSN webservices client for ObsPy. You can read about ObsPy Client library here:
https://docs.obspy.org/packages/obspy.clients.fdsn.html
How would you load the Client library into Python and set it to pull data from the IRIS webservices?

Select one:
- a. from obspy.clients.fdsn import client
  Client = client("IRIS")
- b. from obspy.clients.fdsn import client
  client = Client("IRIS")
- c. from obspy.clients.fdsn import Client
  Client = client(IRIS)
- d. from obspy.Clients.fdsn import Client
  client = Client("IRIS")
- e. from obspy.clients.fdsn import Client
  client = Client("IRIS") ✓ Correct. Run this command now.
- f. from obspy.clients.fdsn import Client
  client = Client(IRIS)
- g. from obspy.clients.fdsn import Client
  Client = client("IRIS")
- h. from obspy.Clients.fdsn import client
  Client = client(IRIS)
- i. from obspy.Clients.fdsn import client
  client = Client(IRIS)
- j. from obspy.Clients.fdsn import Client
  client = Client(IRIS)

Check

Correct
Marks for this submission: 1.00/1.00.

Question 23
Correct
0.00 points out of 1.00
Flag question

Next, you should run these commands to set starttime and endtime variables using UTCDateTime:
```
starttime = UTCDateTime("2012-04-17")
endtime = UTCDateTime("2012-04-18")
```
Then you need to use the `get_events()` function of the client library. You should request from the starttime until the endtime and set a minimum magnitude of 6. The output will be stored in a Catalog object called **cat**. Which of the following would accomplish this?

Select one:
- a. cat = client.get_events(starttime=starttime, endtime=endtime, "min_magnitude":6.0)
- b. cat = get_events(start=starttime, end=endtime, minmagnitude=6)
- c. cat = get_events(start=starttime, end=endtime, "min_magnitude": 6.0)
- d. cat = client.get_events(start=starttime, end=endtime, minmagnitude=6)
- e. cat = get_events(starttime=starttime, endtime=endtime, "min_magnitude": 6.0)
- f. cat = Client.get_events(start=starttime, end=endtime, "min_magnitude": 6.0)
- g. cat = Client.get_events(starttime=starttime, endtime=endtime, minmagnitude=6)
- h. cat = Client.get_events(time=starttime, end=endtime, "min_magnitude": 6.0)
- i. cat = client.get_events(starttime=starttime, endtime=endtime, minmagnitude=6) ✓ Correct. Run this command now.

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.00/1.00**.

Question 24
Correct
1.00 points out of 1.00
Flag question

You can do a variety of things with this Catalog object (**cat**) you just created, as you can see from the manual for the Catalog class:
https://docs.obspy.org/packages/autogen/obspy.core.event.Catalog.html
But first, you should go ahead and `print(cat)` to see the events stored in the cat object. How many events are there at magnitude 6.0 and larger on the day we selected?

Answer: 3 ✓

Check

Correct. We only saw 2 events on the seismogram dayplot because the seismogram only showed the first 12 hours of the day, and the third event occurs in the second half of the day.
Correct
Marks for this submission: 1.00/1.00.

Question 25
Correct
0.67 points out of 1.00
Flag question

Next, you can plot the location of the events using the plot() method of the Catalog object. Here is the manual for that function:
https://docs.obspy.org/packages/autogen/obspy.core.event.Catalog.plot.html
Go ahead and run this command now:
```
cat.plot(projection="local")
```
The dayplot told us earlier where the 2 events occurred that were in the first half of the day. Where did the third event occur? Feel free to make this plot full-screen so it's easier to view.

Select one:
- a. Japan

b. Chile
c. Papua New Guinea
d. Indonesia
e. Near Antarctica ✓

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.67/1.00**.

Question 26
Correct
0.67 points out of 1.00
Flag question

Next we will load a catalog of events in a different way to demonstrate how to read an existing catalog file. This uses the read_events function that you can read about here:
https://docs.obspy.org/packages/autogen/obspy.core.event.read_events.html
How would you load the `read_events` function to use it by name only?

Answer: from obspy import read_events ✓

Check

Go ahead and run this command now.
Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.67/1.00**.

Question 27
Correct
1.00 points out of 1.00
Flag question

Next we should use this function to read a catalog of earthquakes for the Chile mainshock from April 17, 2012 and a week of aftershocks. I have made this catalog available as a QuakeML file in this location:
http://www.users.miamioh.edu/brudzimr/classes/chile.xml
How would we read this file into a catalog object called chile?

Select one:
a. read_events(http://www.users.miamioh.edu/brudzimr/classes/chile.xml)
b. chile = read_events(http://www.users.miamioh.edu/brudzimr/classes/chile.xml)
c. chile.read_events(http://www.users.miamioh.edu/brudzimr/classes/chile.xml)
d. chile.read_events("http://www.users.miamioh.edu/brudzimr/classes/chile.xml")
e. read_events("http://www.users.miamioh.edu/brudzimr/classes/chile.xml")
f. chile = read_events("http://www.users.miamioh.edu/brudzimr/classes/chile.xml") ✓ Correct. Make sure to run this command now.

Check

Correct
Marks for this submission: 1.00/1.00.

Question 28
Correct
1.00 points out of 1.00
Flag question

Using the print() command, how many earthquakes are in this catalog?

Answer: 28 ✓

Check

Correct
Marks for this submission: 1.00/1.00.

Question 29
Correct
1.00 points out of 1.00
Flag question

How would you view a zoomed in map of this catalog?

Select one:
a. chile.plot()
b. plot("chile")
c. plot("chile", projection="local")
d. cat.plot(projection="local")
e. plot(chile, projection="local")
f. chile.plot(projection="local") ✓ Correct. Make sure to run this command now. The plot may take several minutes to load, so do not worry.
g. plot(chile)
h. cat.plot()

Check

Correct
Marks for this submission: 1.00/1.00.

Question 30
Correct
1.00 points out of 1.00
Flag question

How would you describe the pattern of seismicity in this sequence?

Answer: There is correlation between its causes. ✓

Check

Correct
Marks for this submission: 1.00/1.00.

Question 31
Correct
1.00 points out of 1.00
Flag question

Next we should look at which stations in the IRIS Global Seismic Network were recording during the earthquake in Chile. We can use the get_stations() function within the client library to perform the webservices request through IRIS:
https://docs.obspy.org/packages/autogen/obspy.clients.fdsn.client.Client.get_stations.html
The network code for the IRIS Global Seismic Network is IU, and you should use a * for the station code to retrieve all the stations in the network. Since we were looking at the LHZ channel seismogram earlier, we should look for stations recording that channel. Which of the following would store this request in an inventory object named inv?

Select one:
a. inv = client.get_stations(network="IU", station="*", channel="LHZ", level="channel", starttime=starttime, endtime=endtime) ✓ Correct. Make sure to run this command now.
b. inventory = client.get_stations(network="IU", station="*", channel="LHZ", level="channel")
c. inv = client.get_stations(network="IU", station="*", channel="LHZ", level="channel")
d. inv = client.get_stations(net="IU", sta="*", cha="LHZ", level="cha", starttime=starttime, endtime=endtime)
e. inventory = client.get_stations(net="IU", sta="*", cha="LHZ", level="cha", starttime=starttime, endtime=endtime)
f. inv = client.get_stations(net="IU", sta="*", cha="LHZ", level="cha")
g. inventory = client.get_stations(net="IU", sta="*", cha="LHZ", level="cha")
h. inventory = client.get_stations(network="IU", station="*", channel="LHZ", level="channel", starttime=starttime, endtime=endtime)

Check

Correct
Marks for this submission: 1.00/1.00.

Question 32
Correct
1.00 points out of 1.00
Flag question

Now how would you plot the locations of these stations?

Select one:
a. inventory.plot()
b. plot()
c. inv.plot() ✓ Correct. Make sure to run this command now.
d. plot(inventory)
e. client.plot(inv)
f. plot(inv)

Check

Correct
Marks for this submission: 1.00/1.00.

Question 33
Correct
1.00 points out of 1.00
Flag question

Now we want to see how many stations are in the contiguous United States. In order to make it easy to count how many stations there are, we can specify the minimum and maximum coordinates of the Unites States with our get_stations function like this:

```
inv = client.get_stations(network="IU", station="*", channel="LHZ", level="channel", starttime=starttime, endtime=endtime, minlatitude="24.5", maxlatitude="49.5", minlongitude="-124.8", maxlongitude="-66.6")
```

If you have not already, run the command above. Then type `print(inv)` to see how many stations the command produced.

How many stations are in the contiguous United States?

NOTE: If you ran into trouble running `print(inv)` try this:

```
len(inv.get_contents()['stations'])
```

Like the `print(inv)` command, `inv.get_contents()` retrieves the contents of the object inv. `['stations']` specifies that we want the array containing the names of the stations. Encompassing the whole command with the `len()` command tells Python that we want the length of the array of stations, giving us the number of stations.

Answer: 11 ✓

Check

Correct
Marks for this submission: 1.00/1.00.

Question 34
Correct
0.67 points out of 1.00
Flag question

Next I would like you to request data for station ANMO in the US that is one of the quietest and produces some of the best recordings in the world. To see the station details, run the following again and look for the station ANMO in the output:

```
print(inv)
```

Which of the following are network.station.location.channel codes for this station?

NOTE: If you ran into trouble running `print(inv)` try this:

```
inv.get_contents()['channels']
```

Like the `print(inv)` command, `inv.get_contents()` retrieves the contents of the object inv. `['channels']` specifies that we want the array containing the full names of the stations, including the channel code. The resulting array gives the network.station.location.channel codes for each station.

Select one or more:
a. IU.ANMO.00.LHZ ✓ 1 of 2 correct answers.
b. AMNO.IU.00.LHZ
c. IU.ANMO.10.LHZ ✓ 1 of 2 correct answers.
d. AMNO.IU.10.LHZ
e. ANMO.IU..LHZ
f. IU.AMNO..LHZ
g. IU.AMNO.00.LHZ
h. ANMO.IU.10.LHZ
i. IU.ANMO..LHZ
j. ANMO.IU.00.LHZ
k. AMNO.IU..LHZ
l. IU.AMNO.10.LHZ

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.67/1.00**.

Question 35
Correct
0.33 points out of 1.00
Flag question

Next we should request 12 hours of data at station ANMO to examine the seismograms of the Chile earthquakes at this distant station in the United States. We can use the get_waveforms() function within the client library to perform the webservices request through IRIS:
https://docs.obspy.org/packages/autogen/obspy.clients.fdsn.client.Client.get_waveforms.html
Which of the following would request 12 hours of data at station ANMO? Note that your existing **starttime** variable has the correct start time in it, but that **endtime** is 24 hours later, so it's not correct for the end time of your request. Instead, you should use some math to specify the end time relative to the **starttime** variable. Which of the following would accomplish this?

Select one:
a. anmo = client.get_waveforms(IU, ANMO, 00, LHZ, starttime, endtime + 12 * 60 * 60)
b. anmo = client.get_waveforms("IU", "ANMO", "00", "LHZ", starttime, endtime + 12 * 60 * 60)
c. anmo = client.get_waveforms("IU", "ANMO", "00", "LHZ", starttime, starttime + 12 * 60 * 60) ✓ Correct. Make sure to run this command now.
d. anmo = client.get_waveforms(net="IU", sta="ANMO", loc="00", cha="LHZ", starttime, endtime + 12 * 60 * 60)
e. anmo = client.get_waveforms(net=IU, sta=ANMO, loc=00, cha=LHZ, starttime, endtime + 12 * 60 * 60)
f. anmo = client.get_waveforms(net="IU", sta="ANMO", loc="00", cha="LHZ", starttime, starttime + 12 * 60 * 60)
g. anmo = client.get_waveforms(net=IU, sta=ANMO, loc=00, cha=LHZ, starttime, starttime + 12 * 60 * 60)
h. anmo = client.get_waveforms(IU, ANMO, 00, LHZ, starttime, starttime + 12 * 60 * 60)

Check

Correct
Marks for this submission: 1.00/1.00. Accounting for previous tries, this gives **0.33/1.00**.

Question 36
Correct
1.00 points out of 1.00
Flag question

How do we plot it?

Select one:
a. plot(anmo)
b. client.plot(anmo)
c. anmo(plot())
d. anmo.plot() ✓ Correct. Make sure to run this command now.
e. plot()
f. st.plot(anmo)
g. st.plot()

Check

Correct
Marks for this submission: 1.00/1.00.

Question 37
Correct
1.00 points out of 1.00
Flag question

How does this seismogram compare to the BFO seismogram we looked at earlier?

Answer: Same Plots almost with different times ✓

Check

Correct
Marks for this submission: 1.00/1.00.

Finish review