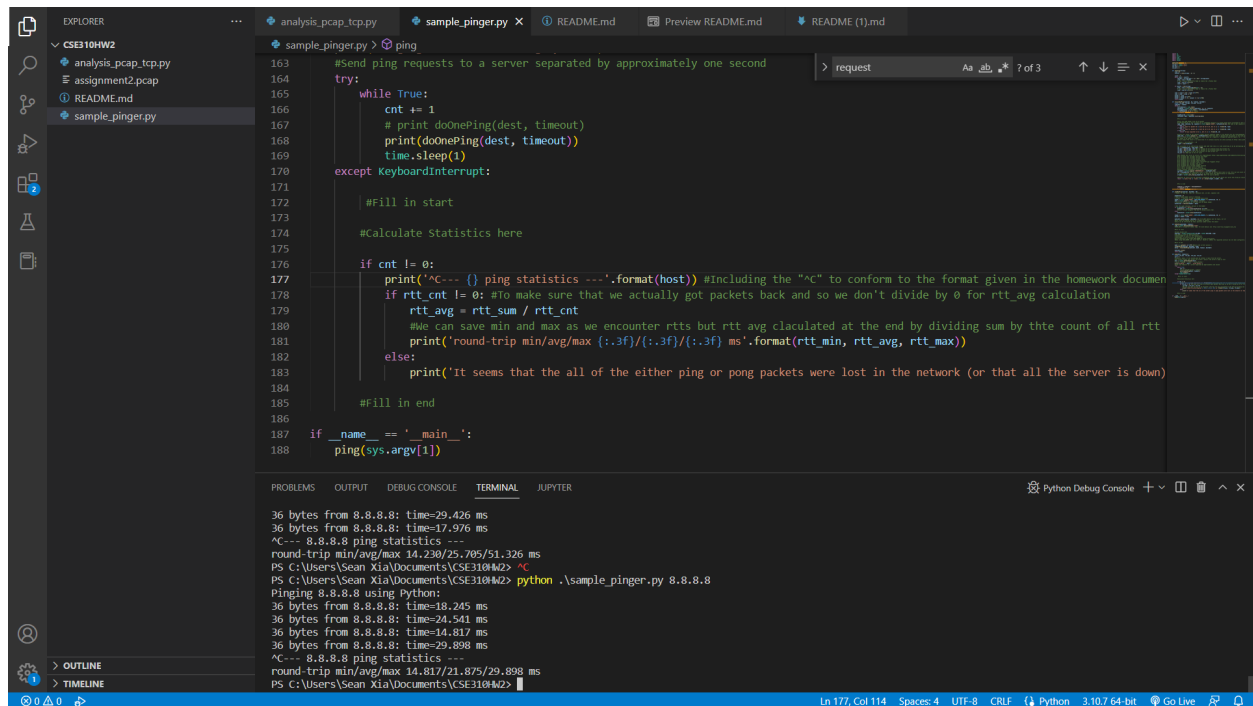


Your program output should look like:

Google DNS Servers 8.8.8.8 (^C printed because won't show on Windows Terminal)

round-trip min/avg/max 14.817/21.875/29.898 ms



The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure: CSE310HW2, analysis\_pcap\_tcp.py, assignment2.pcap, README.md, and sample\_pinger.py. The main editor window shows the sample\_pinger.py script. The script is a Python program that sends ping requests to a server specified by the host argument. It includes a try-except block for KeyboardInterrupt, a while loop for sending pings, and a section for calculating statistics. The output in the terminal shows the results of running the script on 8.8.8.8, including the round-trip min/avg/max times and the ^C signal.

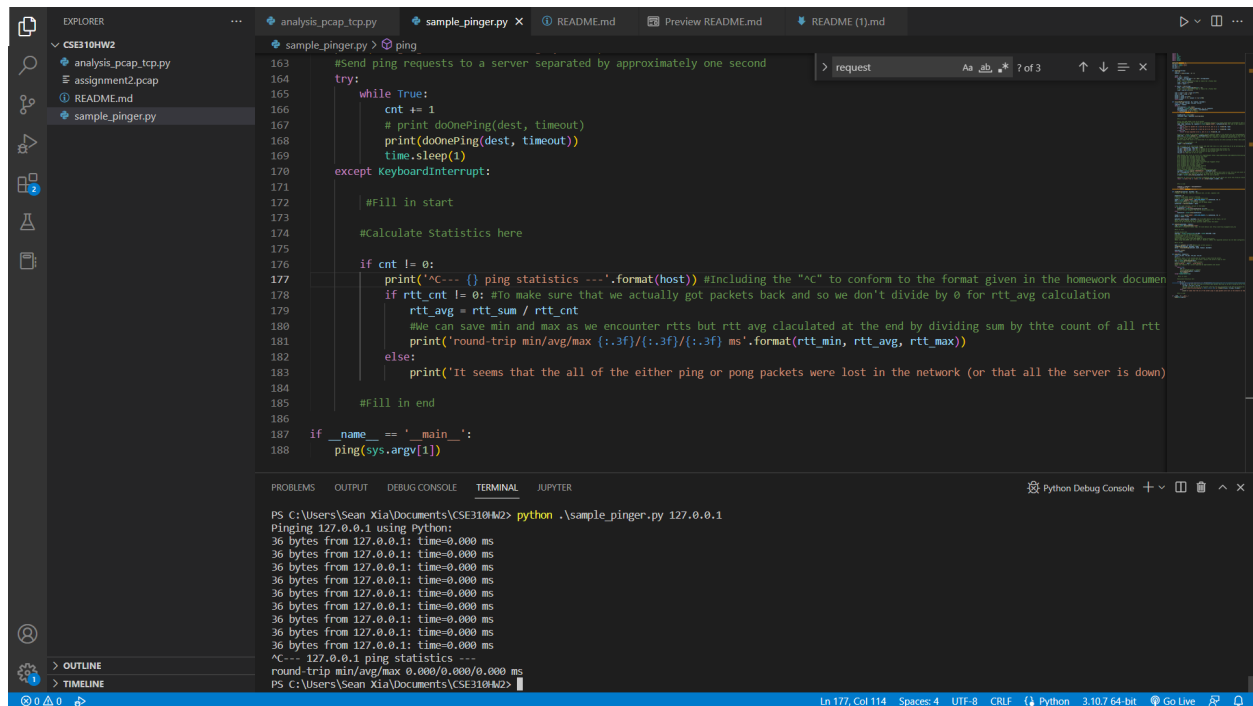
```
163 #send ping requests to a server separated by approximately one second
164 try:
165     while True:
166         cnt += 1
167         # print doOnePing(dest, timeout)
168         print(doOnePing(dest, timeout))
169         time.sleep(1)
170 except KeyboardInterrupt:
171     #Fill in start
172     #Calculate Statistics here
173
174     if cnt != 0:
175         print('^C--- {} ping statistics ---'.format(host)) #including the ^C to conform to the format given in the homework document
176         if rtt_cnt != 0: #to make sure that we actually got packets back and so we don't divide by 0 for rtt_avg calculation
177             rtt_avg = rtt_sum / rtt_cnt
178             #We can save min and max as we encounter rtt's but rtt avg calculated at the end by dividing sum by the count of all rtt
179             print('round-trip min/avg/max {:.3f}/{:.3f}/{:.3f} ms'.format(rtt_min, rtt_avg, rtt_max))
180         else:
181             print('It seems that all of the either ping or pong packets were lost in the network (or that all the server is down)')
182
183     #Fill in end
184
185 if __name__ == '__main__':
186     ping(sys.argv[1])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
36 bytes from 8.8.8.8: time=29.426 ms
36 bytes from 8.8.8.8: time=17.976 ms
^C-- 8.8.8.8 ping statistics ---
round-trip min/avg/max 14.230/25.705/51.326 ms
PS C:\Users\Sean Xia\Documents\CSE310HW2> python .\sample_pinger.py 8.8.8.8
Pinging 8.8.8.8 with: 32 bytes of data:
36 bytes from 8.8.8.8: time=18.245 ms
36 bytes from 8.8.8.8: time=24.541 ms
36 bytes from 8.8.8.8: time=14.817 ms
36 bytes from 8.8.8.8: time=29.898 ms
^C-- 8.8.8.8 ping statistics ---
round-trip min/avg/max 14.817/21.875/29.898 ms
PS C:\Users\Sean Xia\Documents\CSE310HW2>
```

Scenario A:

127.0.0.1 (0.000/0.000/0.000 since testing to same machine)



The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure: CSE310HW2, analysis\_pcap\_tcp.py, assignment2.pcap, README.md, and sample\_pinger.py. The main editor window shows the sample\_pinger.py script. The output in the terminal shows the results of running the script on 127.0.0.1, including the round-trip min/avg/max times and the ^C signal.

```
163 #send ping requests to a server separated by approximately one second
164 try:
165     while True:
166         cnt += 1
167         # print doOnePing(dest, timeout)
168         print(doOnePing(dest, timeout))
169         time.sleep(1)
170 except KeyboardInterrupt:
171     #Fill in start
172     #Calculate Statistics here
173
174     if cnt != 0:
175         print('^C--- {} ping statistics ---'.format(host)) #including the ^C to conform to the format given in the homework document
176         if rtt_cnt != 0: #to make sure that we actually got packets back and so we don't divide by 0 for rtt_avg calculation
177             rtt_avg = rtt_sum / rtt_cnt
178             #We can save min and max as we encounter rtt's but rtt avg calculated at the end by dividing sum by the count of all rtt
179             print('round-trip min/avg/max {:.3f}/{:.3f}/{:.3f} ms'.format(rtt_min, rtt_avg, rtt_max))
180         else:
181             print('It seems that all of the either ping or pong packets were lost in the network (or that all the server is down)')
182
183     #Fill in end
184
185 if __name__ == '__main__':
186     ping(sys.argv[1])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
PS C:\Users\Sean Xia\Documents\CSE310HW2> python .\sample_pinger.py 127.0.0.1
Pinging 127.0.0.1 with: 32 bytes of data:
36 bytes from 127.0.0.1: time=0.000 ms
36 bytes from 127.0.0.1: time=0.000 ms
36 bytes from 127.0.0.1: time=0.000 ms
36 bytes from 127.0.0.1: time=0.000 ms
36 bytes from 127.0.0.1: time=0.000 ms
36 bytes from 127.0.0.1: time=0.000 ms
36 bytes from 127.0.0.1: time=0.000 ms
36 bytes from 127.0.0.1: time=0.000 ms
36 bytes from 127.0.0.1: time=0.000 ms
^C-- 127.0.0.1 ping statistics ---
round-trip min/avg/max 0.000/0.000/0.000 ms
PS C:\Users\Sean Xia\Documents\CSE310HW2>
```

### Scenario B:

stonybrook.edu (Off-campus so it seems like all of the requests timeout since not on campus wifi? That's my guess because cs.stonybrook.edu works)

The screenshot displays a Jupyter Notebook environment with a file explorer on the left showing files like CSE310HW2, analysis\_pcap\_tcp.py, assignment2.pcap, README.md, and sample\_pinger.py. The main editor shows the code for sample\_pinger.py, which uses the socket library to send ping requests. The script includes comments explaining its purpose and logic for calculating round-trip times and handling errors. The terminal output at the bottom shows the command to run the script and the resulting 'Request timed out.' messages.

```
# sample_pinger.py > ping
163 #send ping requests to a server separated by approximately one second
164 try:
165     while True:
166         cnt += 1
167         # print doOnePing(dest, timeout)
168         print(doOnePing(dest, timeout))
169         time.sleep(1)
170 except KeyboardInterrupt:
171 
172     #Fill in start
173 
174     #Calculate Statistics here
175 
176     if cnt != 0:
177         print('^C--- {} ping statistics ---'.format(host)) #Including the "AC" to conform to the format given in the homework document
178         if rtt_cnt != 0: #to make sure that we actually got packets back and so we don't divide by 0 for rtt_avg calculation
179             rtt_avg = rtt_sum / rtt_cnt
180             #we can save min and max as we encounter rtts but rtt avg calculated at the end by dividing sum by the count of all rtt
181             print('round-trip min/avg/max {:.3f}/{:.3f}/{:.3f} ms'.format(rtt_min, rtt_avg, rtt_max))
182         else:
183             print('It seems that the all of the either ping or pong packets were lost in the network (or that all the server is down)')
184 
185     #Fill in end
186 
187 if __name__ == '__main__':
188     ping(sys.argv[1])
```

TERMINAL OUTPUT:

```
PS C:\Users\Sean Xia\Documents\CSE310&Z> python .\sample_pinger.py stonybrook.edu
Pinging 129.49.22.66 using Python:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Request timed out.
^C-- stonybrook.edu ping statistics --
It seems that the all of the either ping or pong packets were lost in the network (or that all the server is down)
PS C:\Users\Sean Xia\Documents\CSE310&Z>
```

cs.stonybrook.edu (seems to work for some reason while regular stonybrook.edu does not work)

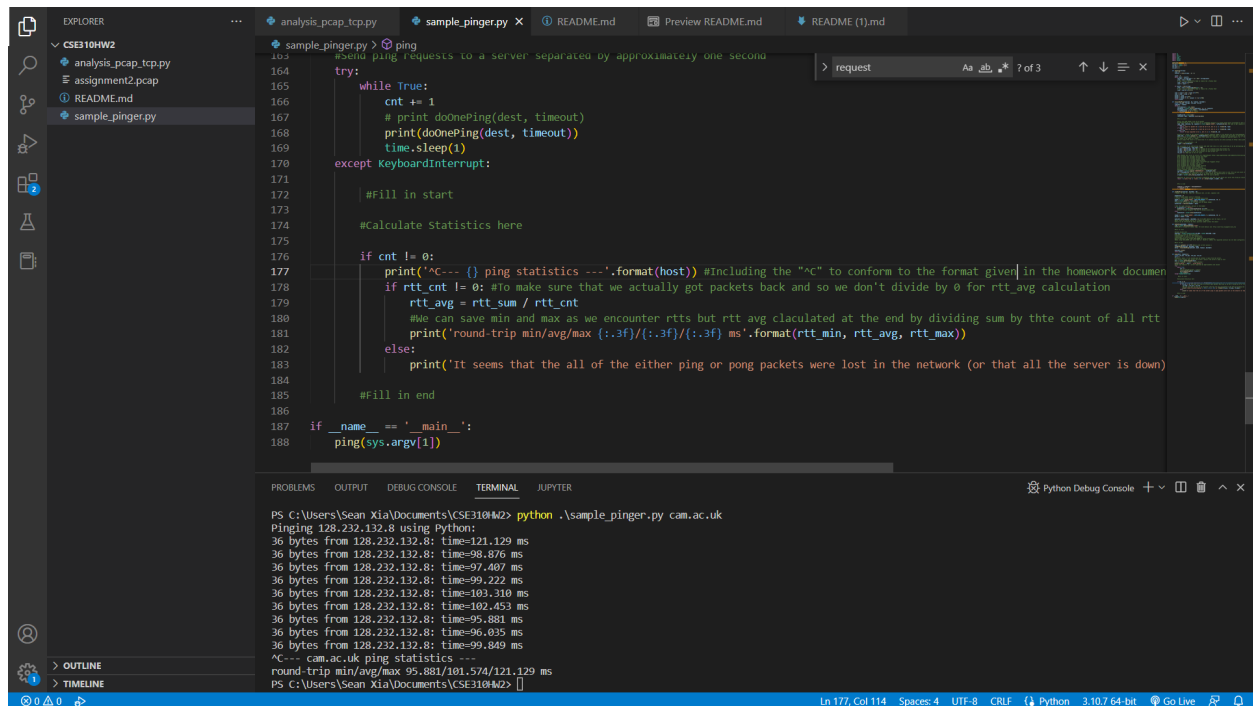
```
EXPLORER
...
CSE310HW2
  analysis_pcap_tcp.py
  assignment2.pcap
  README.md
  sample_pinger.py

sample_pinger.py > ping
163 #send ping requests to a server separated by approximately one second
164 try:
165     while True:
166         cnt += 1
167         # print doOnePing(dest, timeout)
168         print(doOnePing(dest, timeout))
169         time.sleep(1)
170 except KeyboardInterrupt:
171
172     #fill in start
173
174     #Calculate Statistics here
175
176     if cnt != 0:
177         print('^C--- {} ping statistics ---'.format(host)) #Including the "C" to conform to the format given in the homework document
178         if rtt_cnt != 0: #to make sure that we actually got packets back and so we don't divide by 0 for rtt_avg calculation
179             rtt_avg = rtt_sum / rtt_cnt
180             #we can save min and max as we encounter rtt's but rtt_avg calculated at the end by dividing sum by the count of all rtt
181             print('round-trip min/avg/max {:.3f}/{:.3f}/{:.3f} ms'.format(rtt_min, rtt_avg, rtt_max))
182         else:
183             print('It seems that all of the either ping or pong packets were lost in the network (or that all the server is down)')
184
185     #fill in end
186
187 if __name__ == '__main__':
188     ping(sys.argv[1])

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER
Python Debug Console
PS C:\Users\Sean Xia\Documents\CSE310\HW2> python .\sample_pinger.py cs.stonybrook.edu
Ping: 23.185.0.2 using Python:
36 bytes from 23.185.0.2: time=22.017 ms
36 bytes from 23.185.0.2: time=21.633 ms
36 bytes from 23.185.0.2: time=14.532 ms
36 bytes from 23.185.0.2: time=15.543 ms
36 bytes from 23.185.0.2: time=17.545 ms
36 bytes from 23.185.0.2: time=14.123 ms
36 bytes from 23.185.0.2: time=15.932 ms
36 bytes from 23.185.0.2: time=20.988 ms
36 bytes from 23.185.0.2: time=24.281 ms
^C--- cs.stonybrook.edu ping statistics ---
round-trip min/avg/max 14.123/18.510/24.281 ms
PS C:\Users\Sean Xia\Documents\CSE310\HW2>
```

## Scenario C:

Europe I tested the University of Cambridge in Cambridge, England website [cam.ac.uk](http://cam.ac.uk)  
128.232.132.8

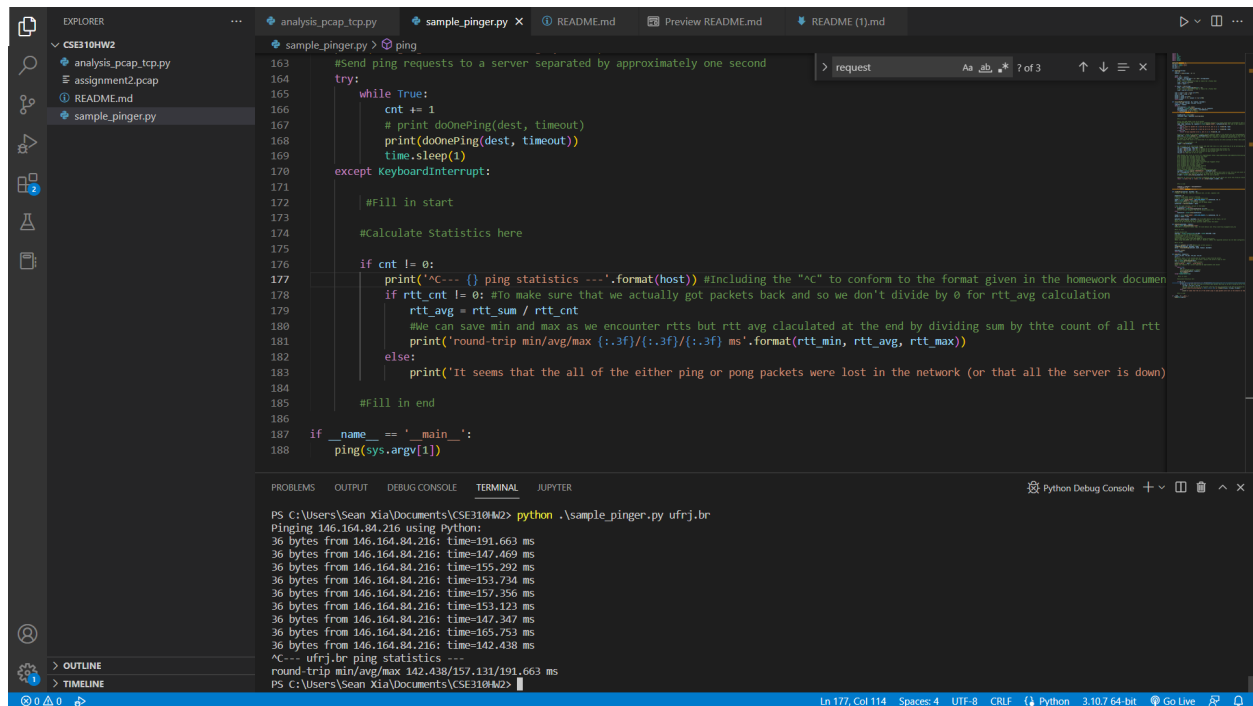


The screenshot shows a VS Code editor with a file explorer on the left containing files like `analysis_pcap_tcp.py`, `assignment2.pcap`, `README.md`, and `sample_pinger.py`. The main editor displays the `sample_pinger.py` script, which is a Python program designed to send ping requests to a specified server and calculate statistics. The script includes comments explaining its functionality, such as sending ping requests at one-second intervals and calculating round-trip time (RTT) statistics. The terminal at the bottom shows the command `python .\sample_pinger.py cam.ac.uk` being executed, resulting in a series of ping responses from 128.232.132.8, followed by a summary of the statistics.

```
163 #Send ping requests to a server separated by approximately one second
164 try:
165     while True:
166         cnt += 1
167         # print doOnePing(dest, timeout)
168         print(doOnePing(dest, timeout))
169         time.sleep(1)
170 except KeyboardInterrupt:
171     #Fill in start
172
173     #Calculate Statistics here
174
175     if cnt != 0:
176         print('\n--- {} ping statistics ---'.format(host)) #including the "\n" to conform to the format given in the homework document
177         if rtt_cnt != 0: #to make sure that we actually got packets back and so we don't divide by 0 for rtt_avg calculation
178             rtt_avg = rtt_sum / rtt_cnt
179             #we can save min and max as we encounter rtt's but rtt avg calculated at the end by dividing sum by the count of all rtt
180             print('round-trip min/avg/max {:.3f}/{:.3f}/{:.3f} ms'.format(rtt_min, rtt_avg, rtt_max))
181         else:
182             print('It seems that all of the either ping or pong packets were lost in the network (or that all the server is down)')
183
184     #Fill in end
185
186 if __name__ == '__main__':
187     ping(sys.argv[1])
```

PS C:\Users\Sean Xia\Documents\CSE310HW2> python .\sample\_pinger.py cam.ac.uk  
Pinging 128.232.132.8 with: 32 bytes of data:  
36 bytes from 128.232.132.8: time=121.129 ms  
36 bytes from 128.232.132.8: time=98.876 ms  
36 bytes from 128.232.132.8: time=97.407 ms  
36 bytes from 128.232.132.8: time=99.222 ms  
36 bytes from 128.232.132.8: time=109.310 ms  
36 bytes from 128.232.132.8: time=102.453 ms  
36 bytes from 128.232.132.8: time=95.881 ms  
36 bytes from 128.232.132.8: time=96.035 ms  
36 bytes from 128.232.132.8: time=99.849 ms  
^C-- cam.ac.uk ping statistics ---  
round-trip min/avg/max = 95.881/101.574/121.129 ms  
PS C:\Users\Sean Xia\Documents\CSE310HW2>

South America I tested the Universidade Federal do Rio de Janeiro in Rio de Janeiro, Brazil  
website [ufrj.br](http://ufrj.br) 146.164.84.216

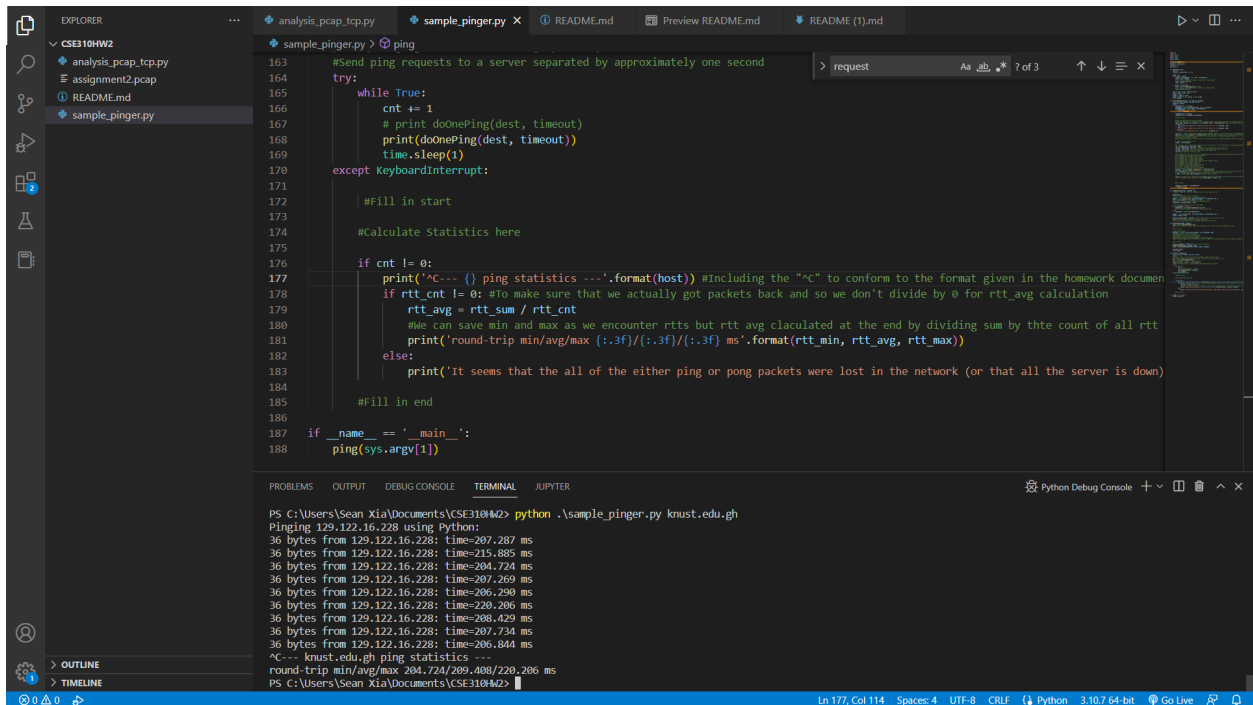


This screenshot is similar to the one above, showing the same `sample_pinger.py` script being executed with the command `python .\sample_pinger.py ufrj.br`. The terminal output displays ping results for the IP address 146.164.84.216, showing a series of round-trip times and a final summary of the statistics.

```
163 #Send ping requests to a server separated by approximately one second
164 try:
165     while True:
166         cnt += 1
167         # print doOnePing(dest, timeout)
168         print(doOnePing(dest, timeout))
169         time.sleep(1)
170 except KeyboardInterrupt:
171     #Fill in start
172
173     #Calculate Statistics here
174
175     if cnt != 0:
176         print('\n--- {} ping statistics ---'.format(host)) #including the "\n" to conform to the format given in the homework document
177         if rtt_cnt != 0: #to make sure that we actually got packets back and so we don't divide by 0 for rtt_avg calculation
178             rtt_avg = rtt_sum / rtt_cnt
179             #we can save min and max as we encounter rtt's but rtt avg calculated at the end by dividing sum by the count of all rtt
180             print('round-trip min/avg/max {:.3f}/{:.3f}/{:.3f} ms'.format(rtt_min, rtt_avg, rtt_max))
181         else:
182             print('It seems that all of the either ping or pong packets were lost in the network (or that all the server is down)')
183
184     #Fill in end
185
186 if __name__ == '__main__':
187     ping(sys.argv[1])
```

PS C:\Users\Sean Xia\Documents\CSE310HW2> python .\sample\_pinger.py ufrj.br  
Pinging 146.164.84.216 with: 32 bytes of data:  
36 bytes from 146.164.84.216: time=191.663 ms  
36 bytes from 146.164.84.216: time=147.469 ms  
36 bytes from 146.164.84.216: time=155.292 ms  
36 bytes from 146.164.84.216: time=153.734 ms  
36 bytes from 146.164.84.216: time=157.356 ms  
36 bytes from 146.164.84.216: time=153.123 ms  
36 bytes from 146.164.84.216: time=147.347 ms  
36 bytes from 146.164.84.216: time=165.753 ms  
36 bytes from 146.164.84.216: time=142.438 ms  
^C-- ufrj.br ping statistics ---  
round-trip min/avg/max = 142.438/157.131/191.663 ms  
PS C:\Users\Sean Xia\Documents\CSE310HW2>

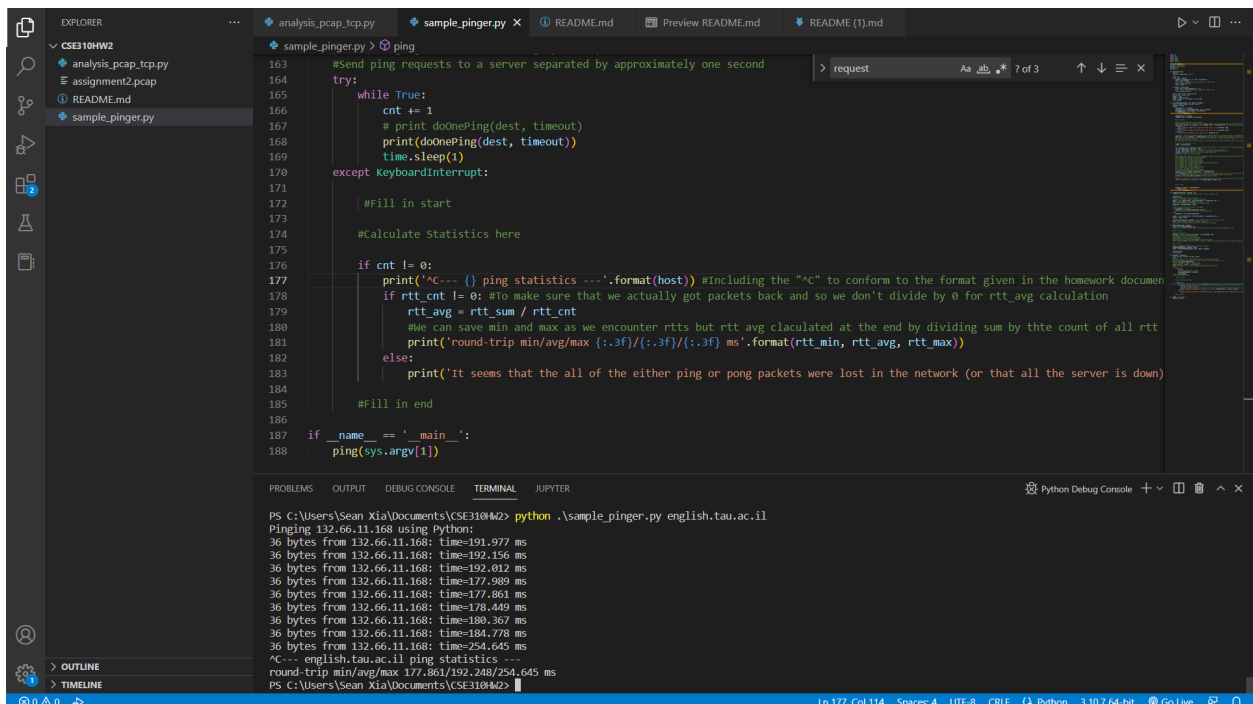
Africa I tested the Kwame Nkrumah University Science & Technology in Kumasi, Ghana website knust.edu.gh 129.122.16.228



```
163 #Send ping requests to a server separated by approximately one second
164 try:
165     while True:
166         cnt += 1
167         # print doOnePing(dest, timeout)
168         print(doOnePing(dest, timeout))
169         time.sleep(1)
170 except KeyboardInterrupt:
171
172     #Fill in start
173
174     #Calculate Statistics here
175
176     if cnt != 0:
177         print('\n--- {} ping statistics ---'.format(host)) #Including the "\n" to conform to the format given in the homework document
178         if rtt_cnt != 0: #to make sure that we actually got packets back and so we don't divide by 0 for rtt_avg calculation
179             rtt_avg = rtt_sum / rtt_cnt
180             #We can save min and max as we encounter rtt's but rtt_avg calculated at the end by dividing sum by the count of all rtt
181             print('round-trip min/avg/max {:.3f}/{:.3f}/{:.3f} ms'.format(rtt_min, rtt_avg, rtt_max))
182         else:
183             print('It seems that all of the either ping or pong packets were lost in the network (or that all the server is down)')
184
185     #Fill in end
186
187 if __name__ == '__main__':
188     ping(sys.argv[1])
```

```
PS C:\Users\Sean Xia\Documents\CSE310HW2> python .\sample_pinger.py knust.edu.gh
Pinging 129.122.16.228 using Python:
36 bytes from 129.122.16.228: time=207.287 ms
36 bytes from 129.122.16.228: time=215.985 ms
36 bytes from 129.122.16.228: time=204.724 ms
36 bytes from 129.122.16.228: time=207.269 ms
36 bytes from 129.122.16.228: time=206.290 ms
36 bytes from 129.122.16.228: time=220.206 ms
36 bytes from 129.122.16.228: time=208.429 ms
36 bytes from 129.122.16.228: time=207.734 ms
36 bytes from 129.122.16.228: time=206.044 ms
^C-- knust.edu.gh ping statistics ---
round-trip min/avg/max 204.724/209.408/220.206 ms
PS C:\Users\Sean Xia\Documents\CSE310HW2>
```

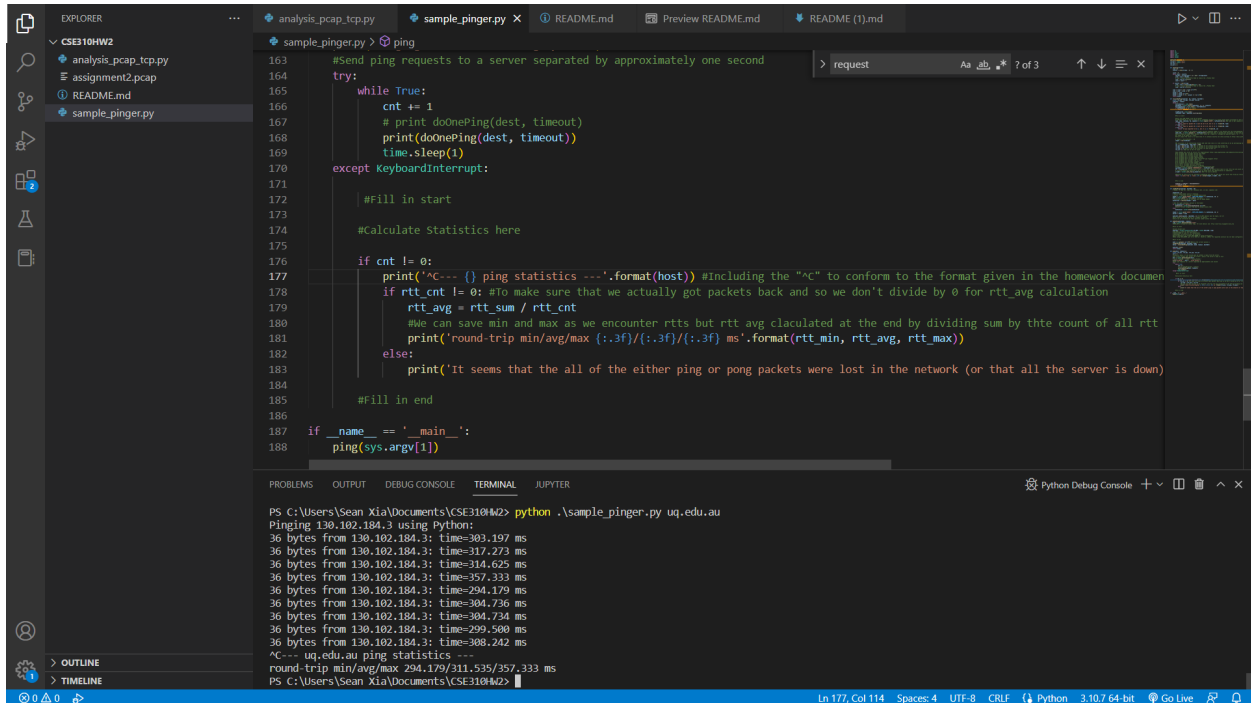
Asia I tested the Tel Aviv University in Tel Aviv, Israel website english.tau.ac.il 132.66.11.168



```
163 #Send ping requests to a server separated by approximately one second
164 try:
165     while True:
166         cnt += 1
167         # print doOnePing(dest, timeout)
168         print(doOnePing(dest, timeout))
169         time.sleep(1)
170 except KeyboardInterrupt:
171
172     #Fill in start
173
174     #Calculate Statistics here
175
176     if cnt != 0:
177         print('\n--- {} ping statistics ---'.format(host)) #Including the "\n" to conform to the format given in the homework document
178         if rtt_cnt != 0: #to make sure that we actually got packets back and so we don't divide by 0 for rtt_avg calculation
179             rtt_avg = rtt_sum / rtt_cnt
180             #We can save min and max as we encounter rtt's but rtt_avg calculated at the end by dividing sum by the count of all rtt
181             print('round-trip min/avg/max {:.3f}/{:.3f}/{:.3f} ms'.format(rtt_min, rtt_avg, rtt_max))
182         else:
183             print('It seems that all of the either ping or pong packets were lost in the network (or that all the server is down)')
184
185     #Fill in end
186
187 if __name__ == '__main__':
188     ping(sys.argv[1])
```

```
PS C:\Users\Sean Xia\Documents\CSE310HW2> python .\sample_pinger.py english.tau.ac.il
Pinging 132.66.11.168 using Python:
36 bytes from 132.66.11.168: time=191.977 ms
36 bytes from 132.66.11.168: time=192.156 ms
36 bytes from 132.66.11.168: time=192.012 ms
36 bytes from 132.66.11.168: time=177.989 ms
36 bytes from 132.66.11.168: time=177.861 ms
36 bytes from 132.66.11.168: time=178.449 ms
36 bytes from 132.66.11.168: time=180.367 ms
36 bytes from 132.66.11.168: time=184.778 ms
36 bytes from 132.66.11.168: time=254.645 ms
^C-- english.tau.ac.il ping statistics ---
round-trip min/avg/max 177.861/192.248/254.645 ms
PS C:\Users\Sean Xia\Documents\CSE310HW2>
```

Australia I tested the University of Queensland in Queensland, Australia website [uq.edu.au](http://uq.edu.au) 130.102.184.3



```
163 #Send ping requests to a server separated by approximately one second
164 try:
165     while True:
166         cnt += 1
167         # print doOnePing(dest, timeout)
168         print(doOnePing(dest, timeout))
169         time.sleep(1)
170 except KeyboardInterrupt:
171
172     #Fill in start
173
174     #Calculate Statistics here
175
176     if cnt != 0:
177         print('%c-- {} ping statistics ---'.format(host)) #Including the "%c" to conform to the format given in the homework document
178         if rtt_cnt != 0: #to make sure that we actually got packets back and so we don't divide by 0 for rtt_avg calculation
179             rtt_avg = rtt_sum / rtt_cnt
180             #ide can save min and max as we encounter rtt's but rtt avg calculated at the end by dividing sum by the count of all rtt
181             print('round-trip min/avg/max {:.3f}/{:.3f}/{:.3f} ms'.format(rtt_min, rtt_avg, rtt_max))
182         else:
183             print('It seems that all of the either ping or pong packets were lost in the network (or that all the server is down)')
184
185     #Fill in end
186
187 if __name__ == '__main__':
188     ping(sys.argv[1])
```

```
PS C:\Users\Sean Xia\Documents\CSE310HW2> python .\sample_pinger.py uq.edu.au
Pinging 130.102.184.3 with Python:
36 bytes from 130.102.184.3: time=303.197 ms
36 bytes from 130.102.184.3: time=317.273 ms
36 bytes from 130.102.184.3: time=314.625 ms
36 bytes from 130.102.184.3: time=357.333 ms
36 bytes from 130.102.184.3: time=294.179 ms
36 bytes from 130.102.184.3: time=304.736 ms
36 bytes from 130.102.184.3: time=304.734 ms
36 bytes from 130.102.184.3: time=299.500 ms
36 bytes from 130.102.184.3: time=308.242 ms
^C-- uq.edu.au ping statistics ---
round-trip min/avg/max 294.179/311.535/357.333 ms
PS C:\Users\Sean Xia\Documents\CSE310HW2>
```

Scenario D is explained in the README as well as attempted to explain here

--- 8.8.8.8 ping statistics ---

round-trip min/avg/max 14.817/21.875/29.898 ms

--- 127.0.0.1 ping statistics ---

round-trip min/avg/max 0.000/0.000/0.000 ms

--- stonybrook.edu ping statistics ---

It seems that the all of the either ping or pong packets were lost in the network (or that all the server is down)

--- cs.stonybrook.edu ping statistics ---

round-trip min/avg/max 14.123/18.510/24.281 ms

--- cam.ac.uk ping statistics ---

round-trip min/avg/max 95.881/101.574/121.129 ms

--- ufrj.br ping statistics ---

round-trip min/avg/max 142.438/157.131/191.663 ms

--- knust.edu.gh ping statistics ---

round-trip min/avg/max 204.724/209.408/220.206 ms

--- english.tau.ac.il ping statistics ---

round-trip min/avg/max 177.861/192.248/254.645 ms

--- uq.edu.au ping statistics ---

round-trip min/avg/max 294.179/311.535/357.333 ms

As expected, the further away the likely location of the server is from New York (location where this program is being tested from), then the higher the general trend of the minimum rtt to these servers is. In general, from what I remember, Australia is the furthest followed by Ghana, Israel, Brazil, England, and then the US, so these website minimum rtt follows the higher distance

means higher minimum rtt trend. I don't want to search up the distance from New York to the respective cities of each university website I used but I believe they likely correspond to distance between New York and the University City. This is likely because the time it takes for packets to travel across the network increases the further away the server is from the client.