* Attached Files:
  + [[File](https://blackboard.andrew.cmu.edu/bbcswebdav/pid-1182639-dt-content-rid-7714658_1/xid-7714658_1) humiditydata.zip](https://blackboard.andrew.cmu.edu/bbcswebdav/pid-1182639-dt-content-rid-7714658_1/xid-7714658_1) (1.298 MB)

This numerical analysis project will focus on performing statistical and visualization analysis on structured sensor humidity and temperature data.

Your source code is to input into your program the structured data attached to this blackboard posting for Sensor A and Sensor B.  Store the data in a sqlite3 database (which requires no database installation), process the data in your program using numpy/pandas to output statistical analysis about the information, and produce a visualization graph using the matplotlib library.

You are to use the following libraries in whatever way you see fit:  sqlite3, numpy/pandas, and matplotlib.

Be sure to format all command line output appropriately.  Label all output appropriately.

Grading for the project is as follows.  You CANNOT skip partial-credit activities.  You work at your own pace, and you can submit project work to this online hyperlink for grading as many times as you like up to, and including, the due date/time — only the last submission is graded (old submissions are zero’ed out, which has no impact on your final grade).  Based on the passing of each partial-credit activity, you earn the following grade — this is your rubric:

* + No activity passed: 0%
  + Activity 1 passed: 65%
  + Activity 1 & 2 passed: 75%
  + Activity 1, 2 & 3 passed: 85%
  + Activity 1, 2 & 3 passed: 90%:  Be sure to document your code and data well.  You will also be graded on how well you can describe how it works.  The comments and documentation required here exceed the typical documentation that you would provide in your code to allow another programmer to modify it.  You need to show that you understand how it is working and why you make the choices that you did.  Always be sure, at the top of any source code that you submit, you include four pieces of information: authors' names, creation date, last modification date, and brief description of your source code.  Your code should also be MODULARIZED -- define your own functions/classes to organize your code and avoid "spaghetti code."
  + Activity 1, 2 & 3 passed: 100%:  You should include a document that describes your understanding of how well the solution works and in what situations you can apply this to other problems.  Filename: **P3Documentation\_LASTNAME\_AndrewID.docx**

Each partial-credit activity builds upon each other, and you may have to rework/rewrite source code to complete the more difficult partial-credit activity.  This should NOT be a GUI program — it it text-based.  
  
Submit your source code (.py files(s) ) for **each** fully completed partial activities through this project hyperlink, any supplemental files your program needs to run, and your MS Word document for a 100%, via **ONE** final submission before the due date/time.  One final submission with all files attached — do **NOT** submit each activity separately or your work will not be graded.

**Final Project Due date/time:  ~~Friday, May 5th~~ Sunday, May 7th @ 11:59pm ET.  No late work will be accepted!**

**YOU MUST USE PYTHON VERSION 3 OR LATER IN THIS CLASS - OTHERWISE YOU EARN A 0% ON THE ASSIGNMENT.**

* https://blackboard.andrew.cmu.edu/images/ci/sets/set01/document_on.gif

**Project #3 Activity #1**

Input into your program all of the sensor data and store the data in a sqlite3 database.

You need to provide comments in your program. You should have a comment block at the start of your code that identifies the author, original creation date, last modification date, and gives a description of the program. You will need to provide comments for each procedure and variable. Comments will be a part of each activity grade.

Submit your source code (.py file(s) ) **[filename = P3A1\_LASTNAME\_AndrewID.py]** through Blackboard. All work must be submitted via Blackboard.

**Please keep in mind that you cannot skip activities. You work at your own pace.**

* https://blackboard.andrew.cmu.edu/images/ci/sets/set01/document_on.gif

**Project #3 Activity #2**

Using some combination of numpy and pandas, create pandas data structure(s) in your program and output to the screen:

* + ~~The average temperature and humidity per day, for each sensor;~~
  + ~~The average temperature and humidity for the entire time period, for each sensor;~~
  + ~~The average temperature and humidity for a user-specified time period, for each sensor;~~
  + ~~The average temperature and humidity for the entire data set~~.

~~For each calculation, print to the screen the total number of data points in each average.~~

2009-04-02 20:00:00 2009-04-17 03:14:34

You need to provide comments in your program. You should have a comment block at the start of your code that identifies the author, original creation date, last modification date, and gives a description of the program. You will need to provide comments for each procedure and variable. Comments will be a part of each activity grade.

Submit your source code (.py file(s) ) **[filename = P3A2\_LASTNAME\_AndrewID.py]** through Blackboard. All work must be submitted via Blackboard.

**Please keep in mind that you cannot skip activities. You work at your own pace.**

* https://blackboard.andrew.cmu.edu/images/ci/sets/set01/document_on.gif

**Project #3 Activity #3**

Using matplotlib, create a visualization plot that represents all data points for a specific sensor for a specific time frame.  This direction is purposely vague; the assignment is for you to be creative in your plot.  There is opportunity for extra credit for submissions that go above and beyond the basic concept of a plot.

You need to provide comments in your program. You should have a comment block at the start of your code that identifies the author, original creation date, last modification date, and gives a description of the program. You will need to provide comments for each procedure and variable. Comments will be a part of each activity grade.

Submit your source code (.py file(s) ) **[filename = P3A3\_LASTNAME\_AndrewID.py]** through Blackboard. All work must be submitted via Blackboard.

**Please keep in mind that you cannot skip activities. You work at your own pace.**

for row in csv.DictReader(f\_input, delimiter=' ', fieldnames=csv\_header[:-1], restkey=csv\_header[-1], skipinitialspace=True):

try:

rows.append([row['user'], row['item'], row['rating'], ' '.join(row['review'])])

except KeyError, e:

rows.append([row['user'], row['item'], row['rating'], ' '])

frame = pd.DataFrame(rows, columns=frame\_header)

print frame

**STEPS:**

**w.r.t marker:**

starting point = receive marker

for 3 its -1 (starting point)

If not participating,

Send marker to everyone else and record state.

(if marker coming from 0-4 and not -1, you block the channel)

Else:

Is this marker form new player? Block the channel

Terminating condition: received marker from all other players

If yes, send state to PITqueue and clear your state

Receive a trade: are you participating in snapshot and channel is allowed?

Add it to the state,

Else:

Don’t do anything

If a player has sent marker

State is hashmap- is the card present in the state, increment the

State hashmap ka first entry is “Player”-‘playernumber’

WARNING: Can not start bundle file:/C:/Users/Rush/Desktop/glassfish4/glassfish/modules/core.jar because it is not contained in the list of installed bundles.

Warning:

Receive of reset acknowledgement time out from PITplayer0

for each in filename:

#filename = 'A'+str(each)+'.csv'

if each == 'A1.csv':

sensors = pd.read\_csv(each, usecols = [0,1,2,3],

names=['#', 'Time, GMT-04:00', 'Temp, °F', 'RH, %', 'End Of File'],header = 0,\

encoding='iso-8859-1', skiprows = 1)

sensors2 = pd.read\_csv(each, usecols = [0,1,2,3],

names=['#', 'Time, GMT-04:00', 'Temp, °F', 'RH, %', 'End Of File'],header = 0,\

encoding='iso-8859-1', skiprows = 1)

combine.append(sensors)

else:

sensors = pd.read\_csv(each, usecols = [0,1,2,3],

header = 1,\

encoding='iso-8859-1', skiprows = 1)

combine.append(sensors)

print('Read data from file ',each)

print(len(combine))

return combine