# Overview

EMAs were sent to participants in the mornings at 9:00 am and evenings at 7:00 pm four times a week on Mondays, Wednesdays, Fridays, and Sundays. Both surveys consisted of five questions asking participants to rate their contentment, stress, loneliness, and sadness on a four-point Likert scale and their energy level on a similar five-point scale. Morning EMAs included four additional questions regarding sleep quality, asking participants to report their total sleep time (TST) in hours, sleep onset latency (SOL) in minutes, number of awakenings (NAW), and restfulness. Participants could enter any numeric value for TST and SOL but were limited to integer inputs for NAW. Restfulness was rated on a four-point scale similar to mood reports. Participants’ responses to mood and restfulness could be one of “Not at all”, “A little bit”, “Quite a bit”, and “Very much” which we encoded to integer values between 0 and 3, respectively. We included a “Neutral” option for participants rating their energy level which meant encoding energy level responses to integer values between 0 and 4. For the remaining sleep metrics, we used the raw values reported on the EMA. Participants also had the option to leave any question unanswered. For analyses requiring complete EMA submissions, we would exclude these reports but would include responses on other questions where possible.

# Specific Process

The code I created to process the three weekly surveys is publicly available on my UTx000 Github repository. [Here](https://github.com/intelligent-environments-lab/utx000/blob/eda790b607949617775295307157cf770c89be18/src/data/wcwh.py#L356-L485) is a link directly to that code. Responses are stored in individual files per participant in the location. If you are using data from UT Box, the filepath will be: /data/raw/utx000/beiwe/survey\_answers/<beiwe\_id>/survey\_answers/<survey\_id>/<date\_and\_time\_of\_survey\_submission>.csv:

* <beiwe\_id>: as you know, each participant has their own unique identifier
* <survey\_id>: These *should* be unique to one of the three surveys: morning, evening, and weekly. The IDs are just a bunch of letters and numbers. For instance, “eQ2L3J08ChlsdSXXKOoOjyLJ” was the ID for the morning survey from UTx000.
* <date\_and\_time\_of\_survey\_submission>: The filename tells you the date and time the survey was *submitted* which is information that is very handy since sometimes the submission time is *significantly* different than when the survey was administered (by hours and/or days in some cases).

My code loops through each participant and pulls data from their submissions from one survey type. Each row in my DataFrame is indexed by the submission time, includes the responses to each question, and then includes their Beiwe ID.

* I pull the submission time from the filename:

datetime.strptime(file[:-4],'%Y-%m-%d %H\_%M\_%S')

* And then add the Beiwe ID, pid, and the answers from each column:

[pid,df.loc[0,'answer'],df.loc[1,'answer'],df.loc[2,'answer'],df.loc[3,'answer'],df.loc[4,'answer']]

Sometimes there are issues with the file, typically when I participant does not provide a response to a question. In that case, I move the survey answer file to a separate folder so I can review it manually and edit or delete:

print(f'\t\tProblem with evening survey {file} for Participant {pid} - Participant most likely did not answer a question')

self.move\_to\_purgatory(f'{data\_dir}{participant}/survey\_answers/{evening\_survey\_id}/{file}',f'../../data/purgatory/{pid}-survey-evening-{file}-{self.suffix}')

The next step is replacing the string responses with numeric values. I am not sure how this happened, but there are a wide variety of survey response strings in terms of the specific words and capitalization. I think there were differences between iOS and Android surveys, but I have a comprehensive line that replaces any and all strings (but actually still missed one I found out later):

morning\_survey\_df.replace({'Not at all':0,'A little bit':1,'Quite a bit':2,'Very Much':3,'Low energy':0,'Low Energy':0,'Somewhat low energy':1,'Neutral':2,'Somewhat high energy':3,'High energy':4,'High Energy':4,'Not at all restful':0,'Slightly restful':1,'Somewhat restful':2,'Very restful':3, 'NO\_ANSWER\_SELECTED':1, 'NOT\_PRESENTED':-1, 'SKIP QUESTION':-1}, inplace=True)

Then I restrict the responses to the study timeframe because there were a few surveys completed before and after the timeframe I was looking at – June 15th to September 1st. Finally, I save the DataFrames as the CSV files that you are familiar with.

That summarizes the whole process for each of the three surveys. Luckily, the answers for each survey are quite similar so the process is quite similar across all three surveys. Honestly, I think I could have made my code more efficient because the processing is so similar but I think the current code is easier to understand and it works, so I figured why change it?