BIODS 253 Final Design Doc

# Overview

This program is being created as part of the Stanford class BIODS 253: Software Engineering for Scientists by Stanford Bioengineering graduate student Scott Piper ([sjpiper@stanford.edu](mailto:sjpiper@stanford.edu)) in Winter quarter 2022. This class has covered the basics of software engineering and how to write clean, usable code in a collaborative environment. Highlighted topics include testing, source control, variable naming, defining functions, and writing descriptive code.

The content from this program is inspired by a problem set from the Stanford class BIOE 217: Translational Bioinformatics. This problem uses data from a digital health paper to analyze airline travel’s effect on sleeping patterns using wearables. The functions in this repository read in wearable data, extract sleeping events, categorize them as affected by a flight or not, and then compare the two groups. The original problem set was for one participant in the study, but by adapting the code from the Jupyter Notebook problem set into a python program, it will be more easily applied to more participants.

# Background

Wearables are becoming increasingly common in society and have a wide range of applications. One of these is tracking the amount and quality of sleep. Modern wearables can detect when the user is sleeping and keep a log of the total time spent sleeping on a certain day. Airline flights are known to alter sleep patterns, potentially due to a decrease in air pressure and oxygen. A recent study (Li, X., Dunn, J., Salins, D., Zhou, G., Zhou, W., Rose, S. M. S. F., ... & Sonecha, R. (2017). [Digital health: tracking physiomes and activity using wearable biosensors reveals useful health-related information.](https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.2001402) PLoS biology, 15(1), e2001402) has examined the correlation between flights and sleep patterns using data from wearables.

Most wearable data comes in the form of a time series. The data in this study was collected using a Basis Watch. This time series data includes when a sleep event started and how long it lasted. It also tracks the start, duration, and distance traveled for events including air travel. Both sleep and flight events were extracted, and the sleep events were sorted based on if they closely followed a flight. The two groups are then compared, and a t-test is used to determine if there is a significant change and Cohen’s d is found to quantify the effect size. Cohen’s d was chosen for effect size because the flight and non-flight groups are different sizes, and Pearson’s relies on datasets of the same size. Histograms are used to visualize all the sleeps, all the flights, and the comparison between the flight-affected and non-flight-affected sleeps.

# Current Goals

The goal of this project is to take code written for a Jupyter notebook assignment and transform it into a clean, easy to understand python program. This program should be able to be downloaded from a GitHub repository and used by someone who has never seen it before. The code will include comments, functions, unit tests, and variable names. The output will be statistical tests and histograms for visualization.

# Non-Goals

Non-goals include:

* Analyzing large amounts of data using the program
* Answering questions about sleep and air travel themselves
* Contributing towards academic research

# Future Goals

Future goals could include:

* Testing on a larger dataset to make sure code is transferable
* Expanding to other wearables data
* Adding functions to look at other activity types, not just air travel

# Detailed Design

The program consists of a library of functions. First, we define a few basic functions to read in the input data, calculate basic statistics, plot a histogram, and calculate Cohen’s d.

After this initial material comes the first main function. Function one takes the sleep data and determines how many hours of sleep the subject got on each day. A histogram is returned and basic stats are calculated.

The second main function takes activity data and determines the date and duration of flights taken by the subject. This is done using activities labeled as ‘airplane’ or those with speeds over 100 mph, less than 700 mph, and lasting more than 30 minutes. A histogram is returned here as well.

The third main function determines which days were affected by airline travel and daily sleep is separated into either the flight-affected sleep or non-flight-affected sleep. A histogram and t-test are delivered to compare the two groups. Additionally, Cohen’s d is found to analyze the effect size of airline travel on sleep.

## A command line interface is also established so the user can run the functions and input data from the terminal.

## User Requirements

The intended users of this program are other students interested in analyzing multiple research subjects’ data. They expect to be able to input data from the same research paper and visualize that data as well as have statistical results delivered. They are moderately technically savvy. They will use this program on a laptop. There are no major external user requirements. A basic Python package is sufficient. Data sets are not overly large as to require cloud computing.

## Data Validation/Potential Error States

The biggest risk for error is if column names or orders are not the same between input datasets. This formatting must be consistent for the program to run correctly. A file will be included to instruct users on the necessary columns and what they should be named before running the program.

## Privacy

All data used is from an anonymous study. There may be privacy concerns around tracking someone’s movement/travel for their location data. To mitigate this, only time and distance data is required, not actual geographic position.

## Security

There are no major security concerns with this program.

## Testing

Tests were written for the main functions in the sleep analysis library in the file sleep\_analysis\_unittests.py. They include the following:

test\_basic\_stats: This test takes a dataframe of sample data (one column of arbitrary numbers) and runs them through the basic\_stats function. The expected answers were calculated by hand and the actual output is compared to the expected values.

test\_cohens: this test takes two lists of arbitrary numbers and runs them through the cohend function. The actual Cohen’s d was calculated by hand and the output is compared to this expected value. The output eff\_string, a string that tells the size of the effect based on the calculated cohen’s d, is also tested.

test\_sleep\_processing: this test takes a csv file (sleep\_test\_data\_in.csv) containing a small sample of subject 1’s data and runs it through the sleep\_processing function. The test makes sure that the dates are parsed and grouped correctly and the sum of sleep on each day is accurate. The expected output was hand calculated in sleep\_test\_data\_out.csv and is compared to the actual values from the function.

test\_activity\_processing: this test takes a csv file (activity\_test\_data\_in.csv) and runs it through the activity\_processing function. The input test data contains activities to test all of the filtering done in the function, including an activity labeled ‘airplane’ that should be kept, activities labeled ‘walking’ and ‘cycling’ that should be dropped, an activity labeled ‘transport’ with the correct speed that should be kept, activities labeled ‘transport’ with speeds too fast or slow that should be dropped, and an activity labeled ‘airplane’ that should be dropped because the duration is too short. The actual list of dates and duration of flights were calculated by hand (activity\_test\_data\_out.csv) and compared to the output.

test\_fligh\_effect\_sleep: this test runs the sleep\_processing and activity\_processing functions with the test data and then uses those outputs to test the flight\_effect\_sleep function. This function tests if days were sorted properly into being affected by a flight or not. The expected values were found by hand (flight\_effect\_test\_data\_out.csv) and compared to the actual function outputs.

The function read\_data was not tested because it is just reading in data from csv files. The function histogram was not tested because it is just using matplotlib to draw a histogram and add labels.

# Third Party Dependencies

Third party library and function dependencies include:

* Numpy
  + Mean
  + Var
* Pandas
* Matplotlib.pyplot
* Logging
* Argparse
* Datetime
  + Timedelta
* Scipy
  + Stats
* IPython
  + Display
* Math
  + Sqrt
* Unittest
* Os
* Sys

# Work Estimates

The entirety of this project is being completed by Scott Piper. There were three main phases. The first phase was writing the bulk of code necessary to process the data in a Jupyter notebook. The next phase was converting that code to a more generalizable python program. The final phase is writing tests for the functions. Each phase is estimated at 10-15 hours.

# Related Work

Data and motivation for this program come from this study:

Li, X., Dunn, J., Salins, D., Zhou, G., Zhou, W., Rose, S. M. S. F., ... & Sonecha, R. (2017). [Digital health: tracking physiomes and activity using wearable biosensors reveals useful health-related information.](https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.2001402) PLoS biology, 15(1), e2001402d