INF632 Homework 2

Basic Statistical Tests

Assigned: January 29.

Due: February 19 at 11:59pm. Submissions turned in after this time and still within two weeks of this deadline will be automatically docked 50% of the possible points.

Submission: Share a direct link to your repo / folder on canvas.nau.edu by the deadline.

Points: EE499 students, this is worth 15% of your final grade. EE599 students, this is worth 10% of your final grade.

Background:

Wearable devices have seen great utility in behavioral health science. Though they also provide very different data from what is typically collected in such fields, many of the behavioral health statistical methods apply to data from these sources too.

Assignment:

Your assignment will be to write several functions to perform statistical tests on a collection of data sets and then use those functions to evaluate the dataset given. You may use Octave/MATLAB, Python, R, or any of these in Notebooks, but you must write the functions to compute and perform the statistics outlined below from scratch. You shouldn't need any libraries beyond basics for reading CSVs and two statistic value lookups. You can also use pandas, but I don't want you seeking out ActiGraph specific or statistics specific libraries for what you should be writing from scratch.

Functions

Harmonic Mean

Your function should accept N complete datasets (vectors / arrays / dataframes / your choice) and return the harmonic mean of each. Use the following to calculate the harmonic mean(s):

$$\begin{aligned}\bar{H} &= \frac{n}{\frac{1}{x_1} + \frac{1}{x_2} + \ldots \frac{1}{x_n}} \\ &= \frac{n}{\sum_{i=1}^{n} \frac{1}{x_i}}\end{aligned}$$

Pooled Standard Deviation

Remember that pooled standard deviation can be calculated as follows:

$$\sigma_p = \sqrt{\frac{\sum_{i=1}^{k}(n_i - 1)\sigma_i^2}{\sum_{i=1}^{k}(n_i - 1)}}$$
$$= \sqrt{\frac{(n_1 - 1)\sigma_1^2 + (n_2 - 1)\sigma_2^2 + \ldots (n_k - 1)\sigma_k^2}{n_1 + n_2 + \ldots + n_k - k}}$$

Your function should be able to accept the standard deviation and number of samples from each sample set that is to be combined. It should also be able to self identify the k, and should accept any number of pairings (calculated σ and given n).
Because I want you to build your stats and programming knowledge, you should also write a basic standard deviation function. Here's a reminder of the math needed to calculate the standard deviation of a sample.

$$\sigma = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_i - \mu)^2}$$

T-Test

Your t-test function should accept both of the following:

1. two data sets, from which all parameters needed to calculate a $t - value$ can be calculated
2. μ, σ, and n for each of the two data sets, from which the $t - value$ can be calculated

and should return a $p - value$ for the given datasets (or measures of the datasets).

$$df = n_1 + n_2 - 2$$
$$\sigma_p = \sqrt{\frac{(n_1 - 1)\sigma_1^2 + (n_2 - 1)\sigma_2^2}{df}}$$
$$t = \frac{(\mu_1 - \mu_2)}{\sigma_p \cdot \sqrt{(\frac{1}{n_1}) + (\frac{1}{n_2})}}$$

This function is an exception to the requirement that you not use any special libraries. If you are using MATLAB or Octave, you may use the following to translate your $t - value$ into and return a $p - value$:
`tcdf(t, df)`
And if you are using Python:
`from scipy import stats`
`stats.t.cdf`
Or if you have elected to use R:
`pt()`
Consider how you might change this function to *optionally* make use of your harmonic mean function. Can you provide an input flag that allows you to specify to use either a basic arithmetic mean or your harmonic mean? How about a default of one or the other?

ANOVA

Your ANOVA function should accept three or more datasets and calculate the $F-stat$, but then should use the equivalent to `tcdf()` to find the $p-value$. This function is also an exception, in that you can use a library to find the $p-value$ from the $F-stat$, but like above, you can not use a library to calculate the $F-stat$.
Think carefully about how you can accept some undefined number of datasets and still perform the calculation.
See the slides and Wikipedia[1] for details on how to perform the math, though I have provided you with an overview here:

$$
\begin{aligned}
m &= \text{number of groups} \\
N &= \text{number of observations} \\
n_j &= \text{number of observations in group j} \\
x_{ij} &= \text{measure for observation i group j} \\
\bar{x}_j &= \text{mean for jth group} \\
\bar{X} &= \text{overall mean} = \frac{\sum x_{ij}}{N}
\end{aligned}
$$

$$
\begin{aligned}
SS_{total} &= \sum_{i,j}^{N}(x_{ij}-\bar{X})^2 = \sum x_{ij}^2 - \frac{(\sum x_{ij})^2}{N} \\
SS_{between} &= \sum_{j=1}^{m} n_j \left(\bar{x}_j - \bar{X}\right)^2 = \sum_{j=1}^{m}\left(\frac{(\sum_{i=1}^{n_j} x_{ij})^2}{n_j}\right) - \frac{(\sum x_{ij})^2}{N} \\
SS_{within} &= SS_{total} - SS_{between}
\end{aligned}
$$

$$
\begin{aligned}
df_{between} &= m-1 \\
df_{within} &= N-m
\end{aligned}
$$

$$
\begin{aligned}
MS_{between} &= \frac{SS_{between}}{df_{between}} \\
MS_{within} &= \frac{SS_{within}}{df_{within}}
\end{aligned}
$$

$$
F = \frac{MS_{between}}{MS_{within}}
$$

Repeated Measures ANOVA

Finally, you need to implement a repeated measures ANOVA, often referred to as rANOVA or RMANOVA. See Wikipedia[2] for details on the math, though I have provided you with an overview here:

[1] `https://en.wikipedia.org/wiki/Analysis_of_variance`
[2] `https://en.wikipedia.org/wiki/Repeated_measures_design`

$$
\begin{aligned}
n &= \text{number of subjects} \\
k &= \text{number of conditions} \\
x_{ij} &= \text{measure for subject i condition j} \\
\bar{x}_j &= \text{mean for ith condition} \\
\bar{x}_i &= \text{mean for jth subject} \\
\bar{X} &= \text{overall mean}
\end{aligned}
$$

$$
\begin{aligned}
SS_{subjects} &= \sum_{i=1}^{n}\sum_{j=1}^{k}(x_{ij} - \bar{x}_i)^2 \\
SS_{conditions} &= n\sum_{j=1}^{k}(\bar{x}_j - \bar{X})^2 \\
SS_{error} &= SS_{subjects} - SS_{conditions}
\end{aligned}
$$

$$
\begin{aligned}
df_{conditions} &= k - 1 \\
df_{subjects} &= n - 1 \\
df_{error} &= df_{conditions} \cdot df_{subjects}
\end{aligned}
$$

$$
\begin{aligned}
MS_{conditions} &= \frac{SS_{conditions}}{df_{conditions}} \\
MS_{error} &= \frac{SS_{error}}{df_{error}}
\end{aligned}
$$

$$
F = \frac{MS_{conditions}}{MS_{error}}
$$

Application of your Functions

Using the functions you have developed from above, and the data provided in the git repository ('Sample Data' directory), compute the following and answer the questions posed. The first four questions should make use of the concurrently measured data sets ('actigraph and fitbit' directory), while the last question makes use of the Fitbit only data set ('multiyear' directory).

Daily Steps

Pick one of the devices: How many steps per day, on average do the subjects walk? Use the harmonic and arithmetic mean. Are they different? Why?

Group Variance

Pick one of the devices: What is the variance of the group? *(across subjects, pooled standard deviation)*

Comparing the Devices

Use both of the devices: Does the Fitbit report the same step measures as the ActiGraph? *(t-test)*

Weekend Warriors

Pick one of the devices: Are the subjects equally active across each day of the week? *(as determined by daily steps in day of week, ANOVA)*

Seasonality

In the two year data set ('multiyear' directory), you'll find daily step totals. Across the two years, were all months traveled equally? *(repeated measures ANOVA)*

Expectations and Grading Rubric:

Great news, you get a break from IEEE format submissions on this homework! This time I need to see your code, with comments that make it clear what you are doing. If you choose to complete this assignment in a Notebook (e.g. Jupyter Notebook), you can show your answers to the "Application of your Functions" in markdown cells after each step. If you choose not to use a Notebook, prepare a markdown file (.md) for each or your responses to the "Application of your Functions" questions. Your submission will be graded as follows:

Area	Frac. of Points	UG: Meets Expectations	UG:Exceeds, G:Meets Expectations	G:Exceeds Expectations
Code Requirements	40%	The code takes inputs and provides outputs as specified	The code also accepts optional input and makes use of defaults.	The code can self identify characteristics that change how the calculations will be performed.
Code Clarity	40%	Minimal comments, but sufficient to understand the basics of what is happening.	The code make good use of comments, including outlining some assumptions and where errors could happen (or how they are being handled).	The comments make reading the code so easy that one doesn't even have to know the programming language to understand the steps.
Analysis and Response	20%	The analysis is performed correctly and the findings are accurate.	The analysis and findings are correct, and the explanation shows an understanding of what was done and why.	The analysis and findings are correct, the explanation shows depth of thought, and further analysis is suggested in detail.