A data science workflow for the REPLACE-BG clinical trial dataset

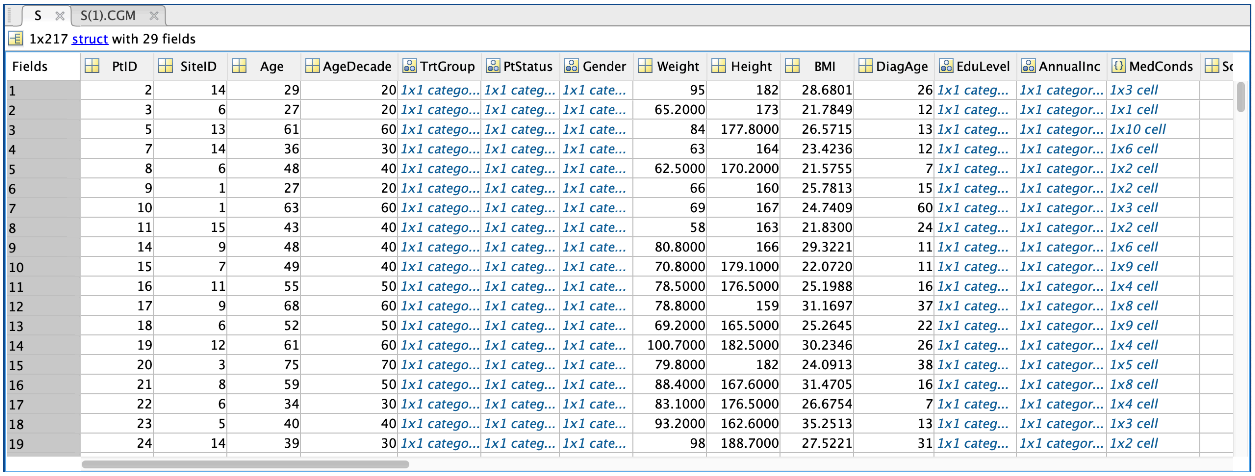
This report references the dataset and code that can be found in this link:
https://drive.google.com/drive/folders/19fOvw-Pb5SRPepv2yQdGeIXpKzaxdAHb?usp=sharing

Introduction:

The most important job of a data scientist or statistician is to look at the data. This is easier said than done when data comes from disparate sources. Relevant data should to be compiled from different sources and processed into logical data structures for flexible utility in an agile workflow. After compiling, data structures can be utilized to generate visualizations to assess if high quality data is being collected and explore relationships between different factors in the dataset. This report describes the compilation and processing steps that could be used to support a workflow during the REPLACE-BG clinical trial data collection phases in order to monitor individual subject data and investigate interesting relationships within the dataset.

Data compilation

As the clinical trial has already concluded, the MATLAB data structure that contains the relevant data to support subsequent analyses is included in the referred material. This structure is named 'S' and it can be loaded by loading 'S.mat'. Here is a snapshot of the format of 'S':



1x217 struct with 29 fields

Fields	PtID	SiteID	Age	AgeDecade	TrtGroup	PtStatus	Gender	Weight	Height	BMI	DiagAge	EduLevel	AnnualInc	MedConds	Sc
1	2	14	29	20	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	95	182	28.6801	26	*1x1 categ...*	*1x1 categor...*	*1x3 cell*	
2	3	6	27	20	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	65.2000	173	21.7849	12	*1x1 categ...*	*1x1 categor...*	*1x1 cell*	
3	5	13	61	60	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	84	177.8000	26.5715	13	*1x1 categ...*	*1x1 categor...*	*1x10 cell*	
4	7	14	36	30	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	63	164	23.4236	12	*1x1 categ...*	*1x1 categor...*	*1x6 cell*	
5	8	6	48	40	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	62.5000	170.2000	21.5755	7	*1x1 categ...*	*1x1 categor...*	*1x2 cell*	
6	9	1	27	20	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	66	160	25.7813	15	*1x1 categ...*	*1x1 categor...*	*1x2 cell*	
7	10	1	63	60	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	69	167	24.7409	60	*1x1 categ...*	*1x1 categor...*	*1x3 cell*	
8	11	15	43	40	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	58	163	21.8300	24	*1x1 categ...*	*1x1 categor...*	*1x2 cell*	
9	14	9	48	40	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	80.8000	166	29.3221	11	*1x1 categ...*	*1x1 categor...*	*1x6 cell*	
10	15	7	49	40	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	70.8000	179.1000	22.0720	11	*1x1 categ...*	*1x1 categor...*	*1x9 cell*	
11	16	11	55	50	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	78.5000	176.5000	25.1988	16	*1x1 categ...*	*1x1 categor...*	*1x4 cell*	
12	17	9	68	60	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	78.8000	159	31.1697	37	*1x1 categ...*	*1x1 categor...*	*1x8 cell*	
13	18	6	52	50	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	69.2000	165.5000	25.2645	22	*1x1 categ...*	*1x1 categor...*	*1x9 cell*	
14	19	12	61	60	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	100.7000	182.5000	30.2346	26	*1x1 categ...*	*1x1 categor...*	*1x4 cell*	
15	20	3	75	70	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	79.8000	182	24.0913	38	*1x1 categ...*	*1x1 categor...*	*1x5 cell*	
16	21	8	59	50	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	88.4000	167.6000	31.4705	16	*1x1 categ...*	*1x1 categor...*	*1x8 cell*	
17	22	6	34	30	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	83.1000	176.5000	26.6754	7	*1x1 categ...*	*1x1 categor...*	*1x4 cell*	
18	23	5	40	40	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	93.2000	162.6000	35.2513	13	*1x1 categ...*	*1x1 categor...*	*1x3 cell*	
19	24	14	39	30	*1x1 catego...*	*1x1 categ...*	*1x1 cate...*	98	188.7000	27.5221	31	*1x1 categ...*	*1x1 categor...*	*1x2 cell*	

Each row contains data related to 1 subject, and each column is a field that can take on one of numerous classes of objects (i.e. cells, tables, categorical, double, etc.). This structure contains 29 fields that were compiled from 9 separate tables or were included as derived variables like BMI and mean hourly [glucose] for each subject (*GlucoseLevelsMean*).

The data structure is already included in the referred material but if the clinical trial data collection was on-going, running '*compileREPLACE_BG_dataset.m*' would update the data structure but due size of the dataset, this may take ~30 minutes.

Graphical User Interface (GUI) to monitor data

Relevant resource: *individualGMDataExplorer.m*

Ensuring that high quality data is collected/captured is a critical aspect of conducting experiments that generate massive amounts of data. In a real-world setting, this necessitates being able visualize the data at interim periods. Subsequently, certain checks can be automatically placed on incoming data to ensure quality data collected during the clinical trial. As such, I created a graphical user interface (GUI) called '*individualGMDataExplorer'*. This GUI (shown below) allows for examining an individual subject's glucose monitoring and bolus data with a few quality check metrics.

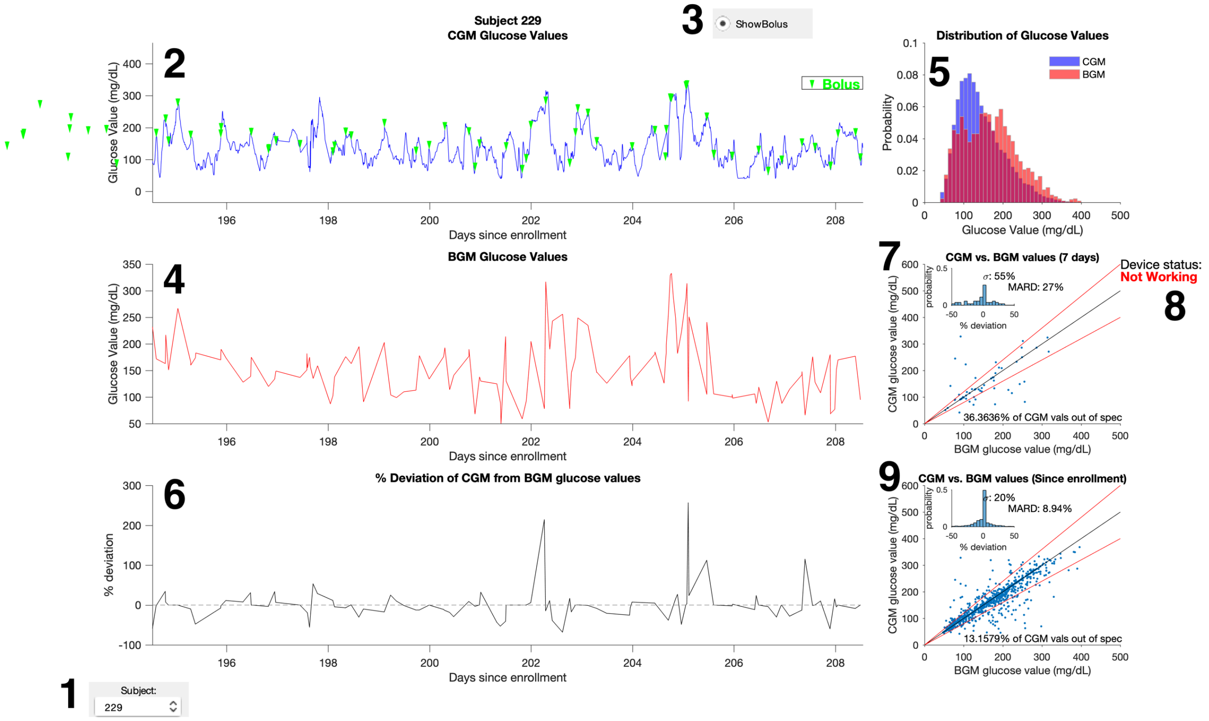


Fig. 1: *individualGMDataExplorer GUI*

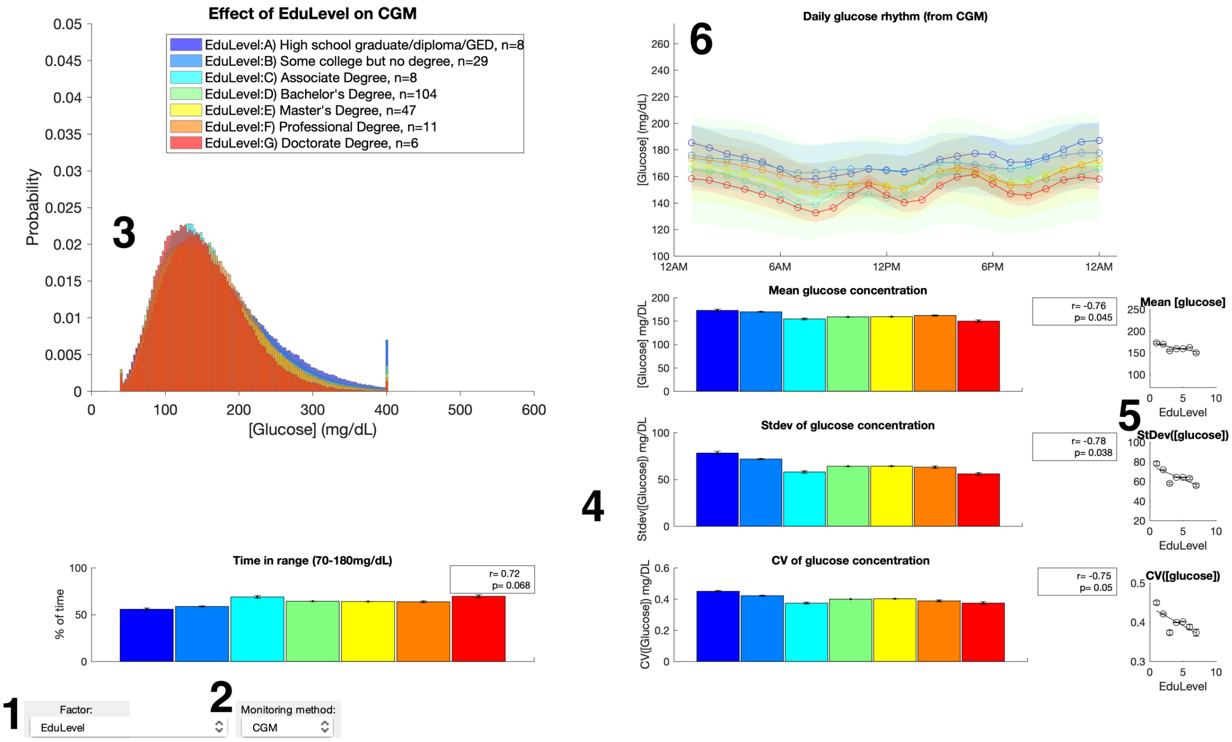
This GUI allows for 1) choosing any subject enrolled in the study and 2) examining CGM-measured glucose concentrations. The x-axis time scale is in days from enrollment, and only the final 2-weeks of data is shown by default; however, the figure can be scrolled back to the date of enrollment. In an effort to improve loading speed, the bolus data is not overlaid in this figure by default. 3) Checking the 'ShowBolus' radio button will show all the boluses administered as downward green arrows. 4) [Glucose] measured using the BGM method is also plotted and 5) the overall distribution of glucose concentrations using the two methods.

Visualizing the device allows for examining if they are working as intended. 6) To visualize sensor accuracy, the relative difference between each BGM-measured [glucose] and the CGM-measured [glucose] within a 5-minute interval preceding it if there was one is plotted. Only the first BGM measurement was analyzed if more than one was performed in a 10-minute interval. 7) Similarly, the GUI allows for a visualization of CGM [glucose] vs. BGM [glucose] in the past 7 days. The red lines represent ± 20% relative deviations of CGM-measured [glucose] from BGM-measured [glucose]. 8) If the mean absolute relative difference exceeds 20% in the last 7 days, a warning appears and it suggests that the device may not be working properly as was the case for subject 229. Lastly, (9) demonstrates all the CGM [glucose] vs. BGM [glucose] measurements from the time of enrollment for each subject.

Graphical User Interface (GUI) to develop insights

Relevant resource: *populationGMDataExplorer.m*

The ultimate reason to the explore large datasets is to discover relationship within the data. When there are many factors a dataset, there can be a multitude of interesting relationships to explore; however, analyzing this can become tedious and difficult to manage if hard-coding analyses to investigate each potentially interesting relationship. In principle, exploring the statistical relationships between variables can be simplified to a small subset data processing steps and statistical tests that depend upon the types of variables in question (e.g. binary, continuous, categorical, etc.). This allows for creating flexible and agile workflows for investigating relationships in datasets. This was the impetus for the *populationGMDataExplorer* GUI to quickly explore how different factors relate to blood glucose levels and fluctuations in the dataset.



Factor: EduLevel

Monitoring method: CGM

Currently, this GUI allows for examining how 15 different factors in the dataset affects glucose levels. 1) The factor to explore can be selected from the drop-down menu. 2) While the CGM data yields cleaner results due to more measurements from the CGM devices, the BGM data can also be examined. Selecting a factor or glucose monitoring method from the drop-down menus populates the subplots in the GUI. 3) The probability distribution of glucose values for each level of the factor can be visualized. If the factor is a continuous or discrete variable with many levels (e.g. Age or HbA1c values), the GUI will discretize the levels into bins. Here, we examine the pooled distribution of glucose levels stratified by education level. 4) Mean, standard deviation, coefficient of variation of [glucose] are calculated for each subject by fitting their glucose concentrations to a log-normal distribution. Also, the Time in Range for each subject in each of the different levels is also computed. Each of these output parameters populate their respective bar charts. 5) Subsequently, linear regressions are computed, and significance values are calculated to asses if there are potentially significant relationships. In the example shown, there appears to be a negative relationship between education and the mean, standard deviation, and coefficient of variation of glucose concentrations, while education tends to increase time in range as well. For the case of a factor with 2 levels such as gender or treatment group, a t-test would be calculated instead to assess significance between groups. 6) Lastly, the average hourly glucose fluctuations were calculated for each subject and then group averages are plotted. The shading error regions represent the pooled standard error of the mean for each trace. This panel qualitatively demonstrates that glucose levels are generally offset to lower levels throughout the day for higher education groups. It is important to keep in mind that correlation does not imply causation, so I would not recommend earning a PhD to manage glucose levels ☺.

Machine Learning to make predictions

predictHbA1c_script.m in this folder contains code that was used to predict HbA1c levels from CGM data using a model based machine learning approach. In the last part of the code, I demonstrated that the glucose concentration-dependent relationship to HbA1c levels change with age.