

# Assignment 1 Project Report

CSE572 - Data Mining

Fall 2019

October  
05, 2019

Submitted to:

**Professor Ayan Banarjee**

Ira A. Fulton School of Engineering  
Arizona State University

Submitted by:

**Group 28**

Ankush Tale	avtale@asu.edu	1217124012
Keshin Jani	khjani@asu.edu	1217202363
Vaibhav Singhal	vsingha5@asu.edu	1215185400
Manikanta Chintakunta	mchintak@asu.edu	1215146842

# 1. Introduction

The Type 1 Diabetes Patient's CGM and Insulin Data Analysis project is part of the course requirement for Data Mining (CSE 572) for the session of Fall 2019 at Arizona State University. The goal of the project is to attempt to develop a heuristic algorithm to detect meals taken by patients with the help of data from Continuous Glucose Monitor and Insulin Ingestion (Bolus: ingestion of large doses during meals and Basal: regular ingestion of small dosage). The data contains glucose levels for 2.5hrs during multiple meals reported at an interval of every 5 mins.

All the documents including supporting graphs, report and the code are also maintained in a GitHub [Repo](#).

## 2. Team members

Ankush Tale  
Vaibhav Singhal  
Keshin Jani  
Manikanta Chintakunta

## 3. Project Phase 1: Feature Extraction

For this phase of the project, the group went through multiple features and decided to select four features to extend the CGM (Continuous Glucose Monitoring) data. Before extracting the feature, we converted the row-wise time-series for distinct meals for a patient into a column-wise time series for easier processing in Python. The gaps in each time-series have been filled using linear interpolation. Now, we have the CGM data converted to column-wise format with 31 columns.

Selected features:

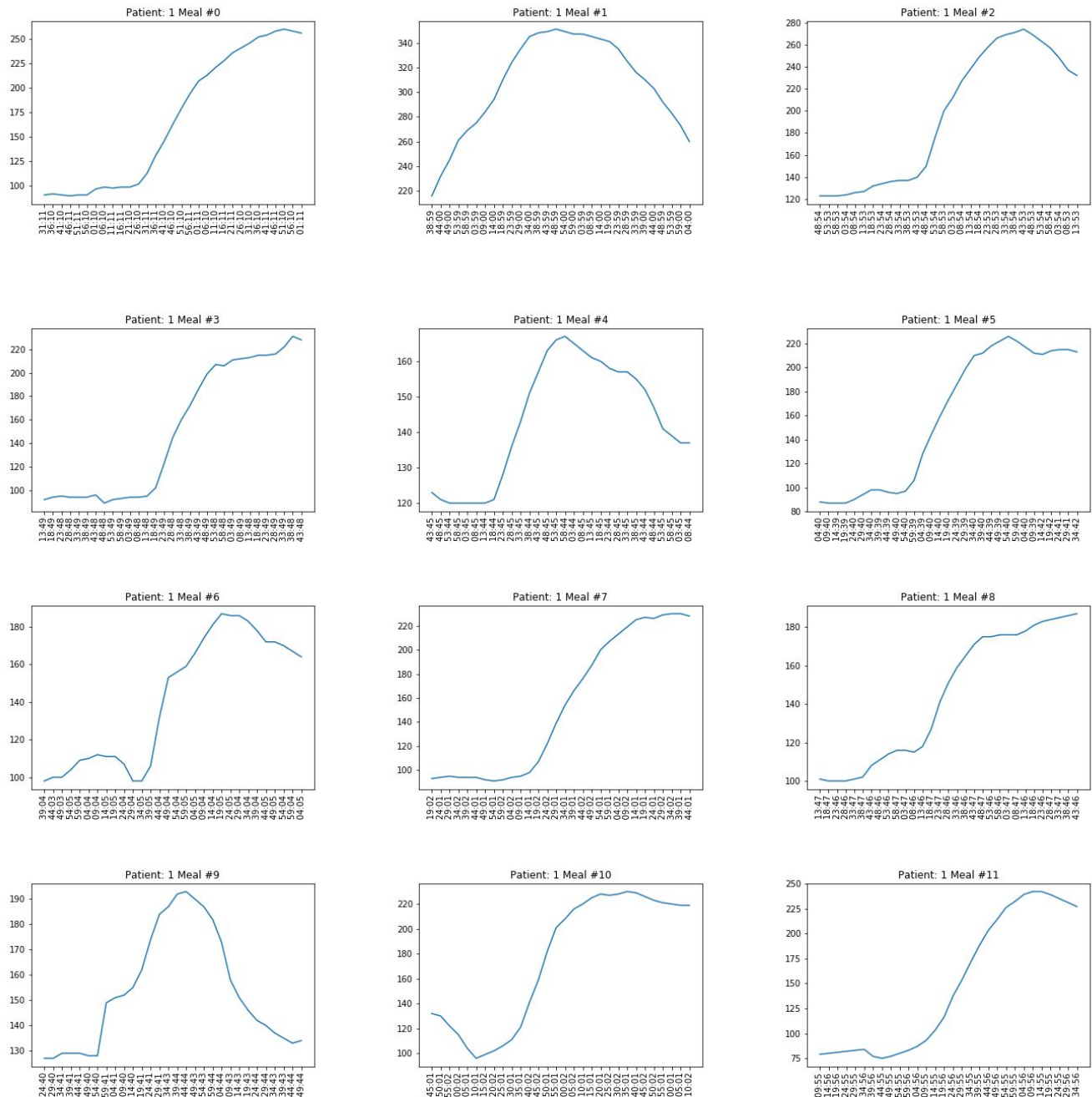
1. Mann-Kendall Test statistics (Ankush)
2. Out of Range readings (Vaibhav)
3. MeanRange (Keshin)
4. Variance of Fast Fourier Transform (Manikanta)

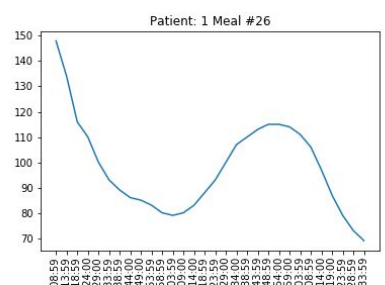
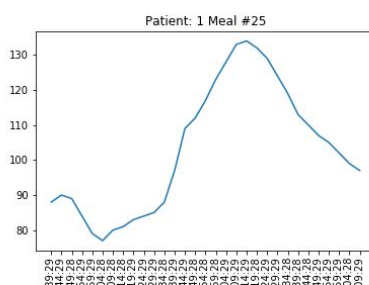
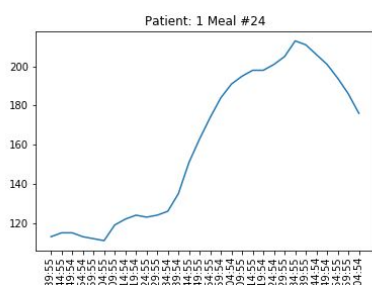
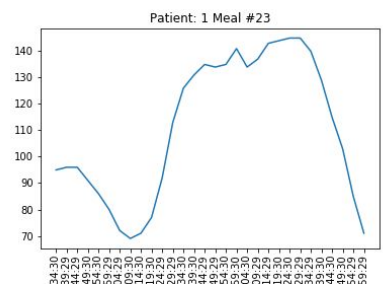
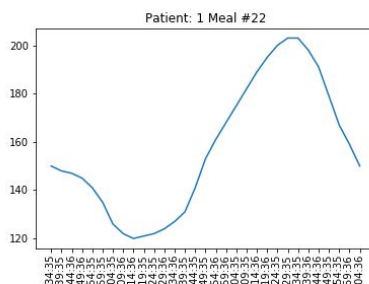
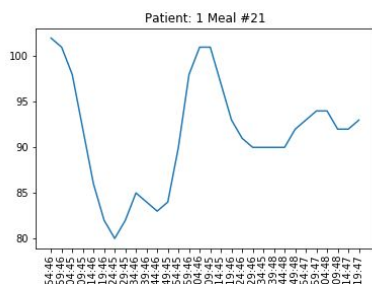
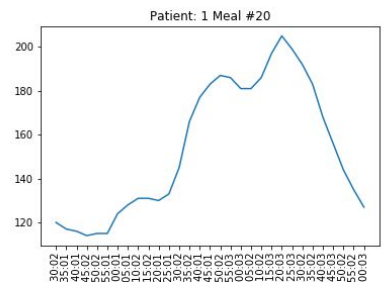
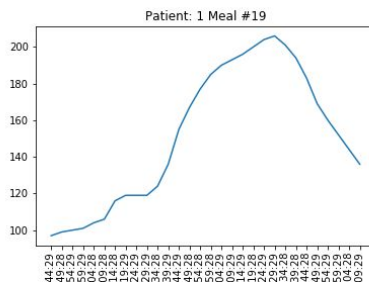
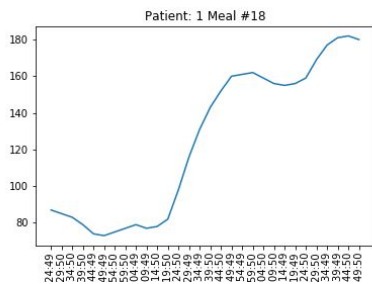
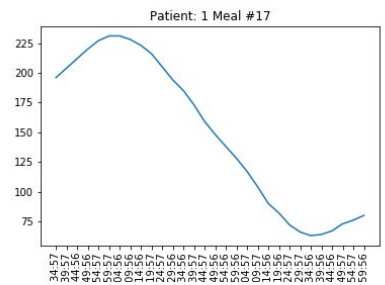
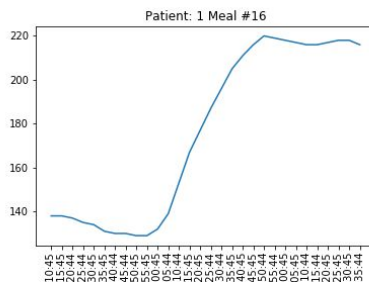
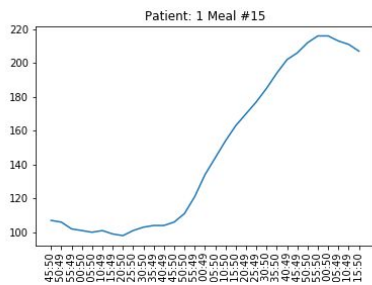
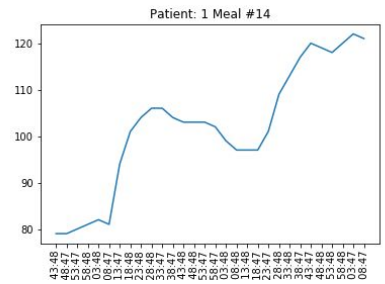
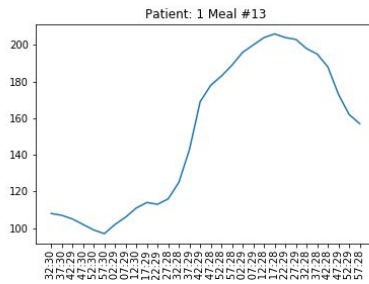
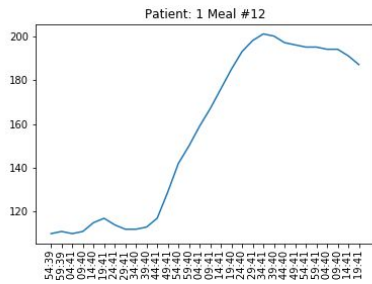
Intuition behind feature selection:

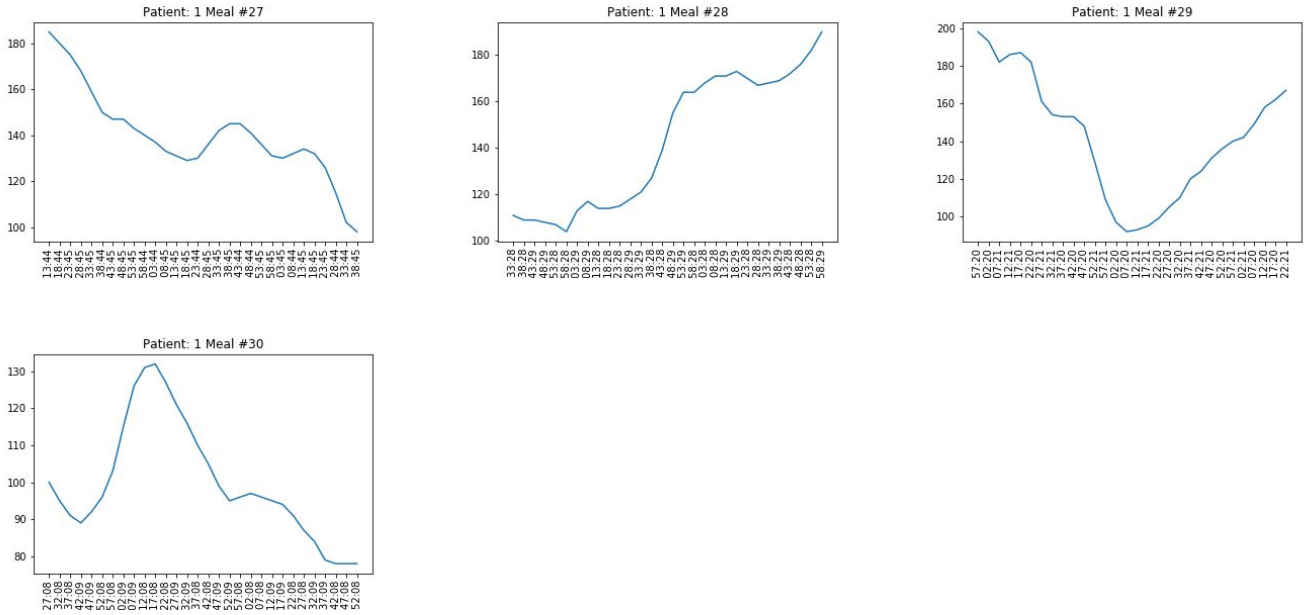
Since the CGM data just had one available feature i.e. CGM values, we decided to plot the values against time to observe any latent features that are easy to interpret. The graphs for all time-series in CGM data for Patient 1 are shown below. It becomes clear that the velocity of values and level gradients are important for noticing how the glucose level has evolved over time. Another general observation is that in majority of time-series the glucose level follows a trend i.e. either it is increasing or decreasing. This can be related to the scenarios where the patient had his meal and then injected with Insulin Bolus which leads to a decrease in glucose levels or when the patient delayed recording a meal after injecting the bolus and hence we see a decrease in glucose levels. There are also cases where you see multiple peaks in values which can be related to a patient having meal in fractions and gaps. With these observations, we tried out many features representative of the above mentioned traits of the time-series. Considering

the common ones, we shortlisted the below mentioned features. A more suitable feature selection would be possible when the CGM data is combined with Insulin injection data which would give a deeper insight. The contribution made by each feature extraction method and how this feature are extracted is explained below.

## Reference Graphs: CGM data for Patient 1







### 3.1 Mann-Kendall Test Statistics

The Mann-Kendall(MK) test is a non-parametric test that helps in locating trends of a given series. MK test recognizes an upward or downward trend in the data despite possible seasonality. MK test can be used in place of parametric linear regression<sup>1</sup>. As mentioned in the paper<sup>2</sup>, The regression analysis requires that the residuals from the fitted regression line be normally distributed; an assumption not required by the MK test, that is, the MK test is a non-parametric (distribution-free) test. Hirsch, Slack, and Smith (1982, page 107)<sup>3</sup> have stated that the MK test can be used as an effective tool in exploratory analysis and is most appropriately used to identify stations where changes are significant or large magnitude and to quantify these findings.

The MK test Statistics is given by:

$$\begin{aligned}
 Z_{MK} &= \frac{S-1}{\sqrt{VAR(S)}} \quad \text{if } S > 0 \\
 &= 0 \quad \text{if } S = 0 \\
 &= \frac{S+1}{\sqrt{VAR(S)}} \quad \text{if } S < 0
 \end{aligned}$$

where VAR(S) is

$$VAR(S) = \frac{1}{18} \left[ n(n-1)(2n+5) - \sum_{p=1}^g t_p(t_p-1)(t_p+5) \right]$$

<sup>1</sup> "The Mann-Kendall test: the need to consider the ... - SciELO." <http://www.scielo.br/pdf/asagr/v35n4/01.pdf>. Accessed 7 Oct. 2019.

<sup>2</sup> "Design Trend Mann-Kendall." [https://vsp.pnnl.gov/help/Vsample/Design\\_Trend\\_Mann\\_Kendall.htm](https://vsp.pnnl.gov/help/Vsample/Design_Trend_Mann_Kendall.htm). Accessed 7 Oct. 2019.

<sup>3</sup> "Techniques of trend analysis for monthly water quality data ...." <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/WR018i001p00107>. Accessed 7 Oct. 2019.

where  $g$  is tied groups,  $t_p$  is number of observations in  $p^{th}$  group

The complete equation can be found in paper<sup>4</sup> and the implementation code is as follows:

```
def mk_test(x, alpha = 0.3):
    """
    Input:
        x: a vector of data
        alpha: significance level (0.05 default)

    Output:
        trend: tells the trend (increasing, decreasing or no trend)
        h: True (if trend is present) or False (if trend is absence)
        p: p value of the significance test
        z: normalized test statistics; proportional to the overall trend

    Examples
    -----
    >>> x = np.random.rand(100)
    >>> trend,h,p,z = mk_test(x,0.05)
    """
    n = len(x)

    # calculate S
    s = 0
    for k in range(n-1):
        for j in range(k+1, n):
            s += np.sign(x[j] - x[k])

    # calculate the unique data
    unique_x = np.unique(x)
    g = len(unique_x)

    # calculate the var(s)
    if n == g: # there is no tie
        var_s = (n*(n-1)*(2*n+5))/18
    else: # there are some ties in data
        tp = np.zeros(unique_x.shape)
        for i in range(len(unique_x)):
            tp[i] = np.sum(x == unique_x[i])
        var_s = (n*(n-1)*(2*n+5) - np.sum(tp*(tp-1)*(2*tp+5)))/18

    if s > 0:
        z = (s - 1)/np.sqrt(var_s)
    elif s < 0:
        z = (s + 1)/np.sqrt(var_s)
    else: # s == 0:
        z = 0

    # calculate the p_value
    p = 2*(1-norm.cdf(abs(z))) # two tail test
    h = abs(z) > norm.ppf(1-alpha/2)

    if (z < 0) and h:
        trend = 'decreasing'
    elif (z > 0) and h:
        trend = 'increasing'
    else:
        trend = 'no trend'

    return trend, h, p, z
```

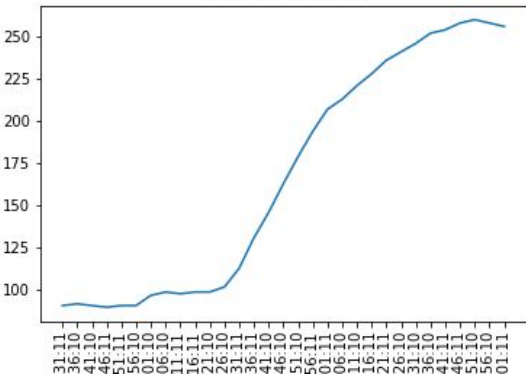
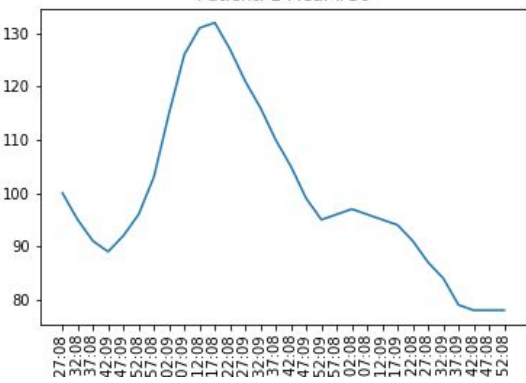
Intuition :

<sup>4</sup> "Design Trend Mann-Kendall." [https://vsp.pnnl.gov/help/Vsample/Design\\_Trend\\_Mann\\_Kendall.htm](https://vsp.pnnl.gov/help/Vsample/Design_Trend_Mann_Kendall.htm). Accessed 7 Oct. 2019.

Since the time-series of glucose values could potentially have variance and won't usually be in one direction, we were searching for a way to statistically extract general trend given a time-series and the confidence level on that trend similar to p-value estimate in null hypothesis testing. This is where we stumbled upon Mann-Kendall Test.

### Observations:

As stated in theory, MK Test finds a general trend in the given data. We could see similar results in the observed values over time series. The first meal mentioned below shows a clear increasing trend which has been backed by the calculated trend from MK Test along with *p value* to back the hypothesis and *z value* which shows the magnitude of the hypothesis.

CGM data for Meals	Trend	<i>p value</i>	<i>z value</i>
<p>Patient: 1 Meal #0</p> 	Increasing	1.885e-13	7.3566
<p>Patient: 1 Meal #30</p> 	Decreasing	0.000702	-3.3886



## 3.2 Out of Range statistics

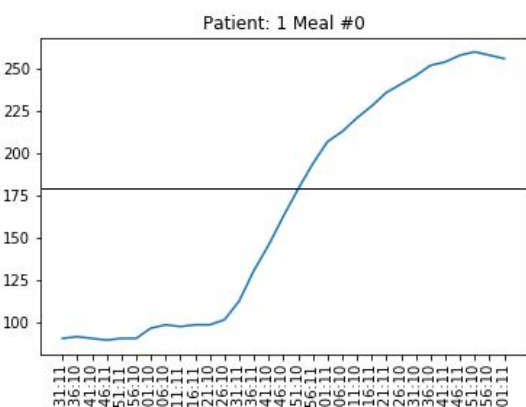
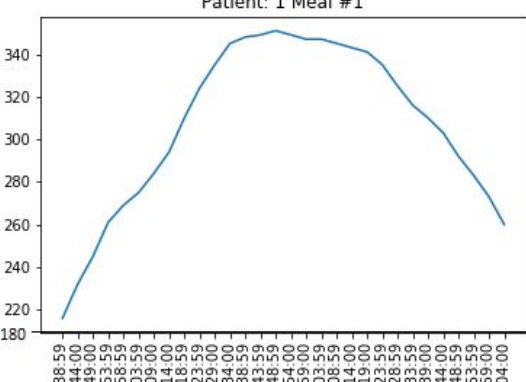
Out of Range: It takes a CGM DataFrame, and evaluate all the instance where the sugar level is over 180<sup>5</sup>, and returns the number of slots where sugar level is above the normal level, each slot is of 5 minute pocket, by which we can observe for how many minutes in one meal time, sugar level was high. We can also check the average high sugar level in one meal time.

### Intuition :

By getting #Out of range and meanOutOfRange values we can observe which meal intake crossed safe level of sugar for diabetes patient. Which can help doctors in scanning logs with better understanding for each time range.

### Observations:

Two mealtime are taken from 2 time ranges, where we can see, in the first plot meanOutOfRange is 25.94 for 14 times, but in second plot meanOutOfRange is 123.97 for 31 times, we can observe that in second case is far more serious than first case, it needs some medical attention as soon as possible because its average is 123.97 above the safe level(180), which can be very dangerous for patients.

CGM data for Meals	#OutOfRange	meanOutOfRange
	14	25.94
	31	123.97

<sup>5</sup> "Hyperglycemia in diabetes - Symptoms and causes - Mayo ...." 3 Nov. 2018, <https://www.mayoclinic.org/diseases-conditions/hyperglycemia/symptoms-causes/syc-20373631>. Accessed 8 Oct. 2019.



### 3.3 Standard Deviation of Fast Fourier Transform

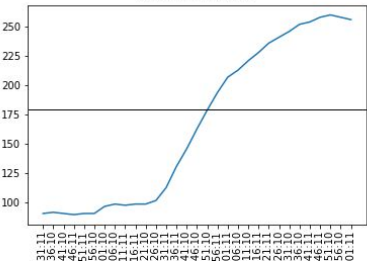
A Fourier Series is used to represent periodic time series data as a sum of sinusoidal components (sine and cosine), Whereas Fast Fourier Transform [FFT] represents time series data in the frequency domain (frequency and power). We are using SD of FFT as it gives more distinction between the two activities based on deviation from the mean. FFT feature was used to segregate the data based on its frequency. We used the fft function from python scipy library. And used numpy variance function to calculate the variance of the fft data. Standard Deviation is the square root of variance.

#### Intuition:

The given time series data is converted to other domain using fourier transform. The key idea behind this is to observe the finer details of the data which is not visible in time series domain. SD of the FFT data helps us to find out how much each value of an activity is deviated from its average sugar level. By using this we can estimate the time interval when the glucose level is one or two standard deviations away from the mean glucose level.

#### Observation:

In meal-0 when you add the values of mean and standard deviation of the result will be closer to the time slot with higher glucose value. Which also infers that during that time meal intake has happened.

CGM data for meals	SD	Mean
	<b>929.9147198118208</b>	<b>385.84791547616663</b>

### 3.4. Difference of Mean Range

Given the dataset of continuous glucose monitor data series and their respective date time series. We have considered a feature to be a difference of the mean ranges in meal. This means that we are finding a range for the mean which is the difference between the maximum and minimum glucose levels of the data series.

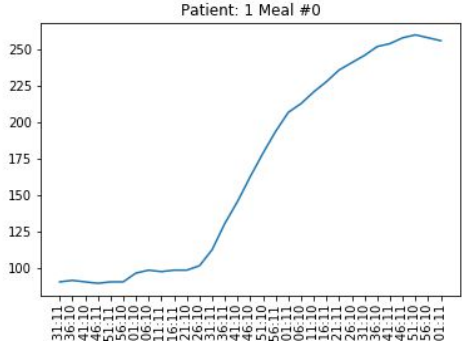
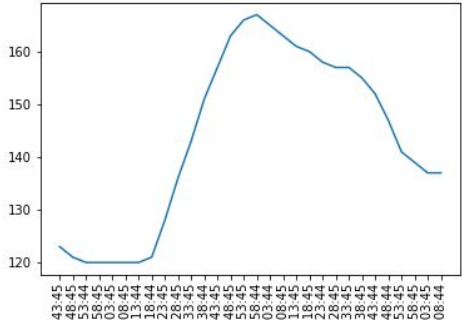
With this we get the range for every meal the patient has and using this range we find a mean over all the meals of every patient. This feature will provide information about how every meal differs from his average meal intake.

#### Intuition:

Here we want to find how much varying his daily meals are over. This we find out using a difference of the mean of meal ranges. This is the difference of the mean meal range and current meal range. This difference in the meal ranges will help us identify the occasions in which the patient has had more or less meal than the usual meal intake of his.

#### Observation:

Given the below two meals for patient 1, we can clearly see that the meal 0 is the meal in which he has had a larger meal intake than his average meal intake while in meal 4 we get that this meal intake is less than his average meal.

CGM data for Meals	Average range for patient	Current meal range	Intuition
	101.51	68.48	Meal intake more than usual
	101.51	-54.51	Meal intake less than usual

## 4. Principal Component Analysis

The feature matrix is of size `#time_series*7` where the first feature “Out of Range” contributes two features, “MK Test” contributes three features, “Variance of FFT” contributes one and “MeanRange” contributes one. Principal Component Analysis (PCA) is generally used to decompose a feature matrix into principal components with data mapped to prominent eigen vectors having highest eigenvalues. We pass this resultant feature matrix to PCA to find best latent semantics which have the highest discrimination power. The outputs of the same can be found in code.

PCA gives us the following:

1. Coeff - matrix representing the coefficients for Principal Components a.k.a Eigenvectors

We get principal component scores which are the transformed representation of the input matrix in the Principal Component. The percentage of variance depicted by each Principal Component.

2. And the Principal Variances i.e. eigenvalues of the input matrix

Principal Components or Eigenvectors are -

Each row in principal component represents the feature from the feature matrix. We pick the highest value of the component and then arrange them in descending order in order to rank them.

OutOfRange: [0.50763523 0.46469441 0.22618434 -0.11072458 0.21180557 0.4987834 0.41146052]

meanOutOfRange: [0.14651226 0.27172214 -0.59990399 0.30315254 -0.64269008 0.19923492 0.01303413]

mk\_test\_trend: [-0.05788929 0.23024768 0.33410788 0.75668486 0.17012612 0.16157205 -0.4520911]

Mk\_test\_p\_value: [0.17261381 -0.29271882 -0.16787171 0.55762197 0.24908462 -0.31858594 0.61794566]

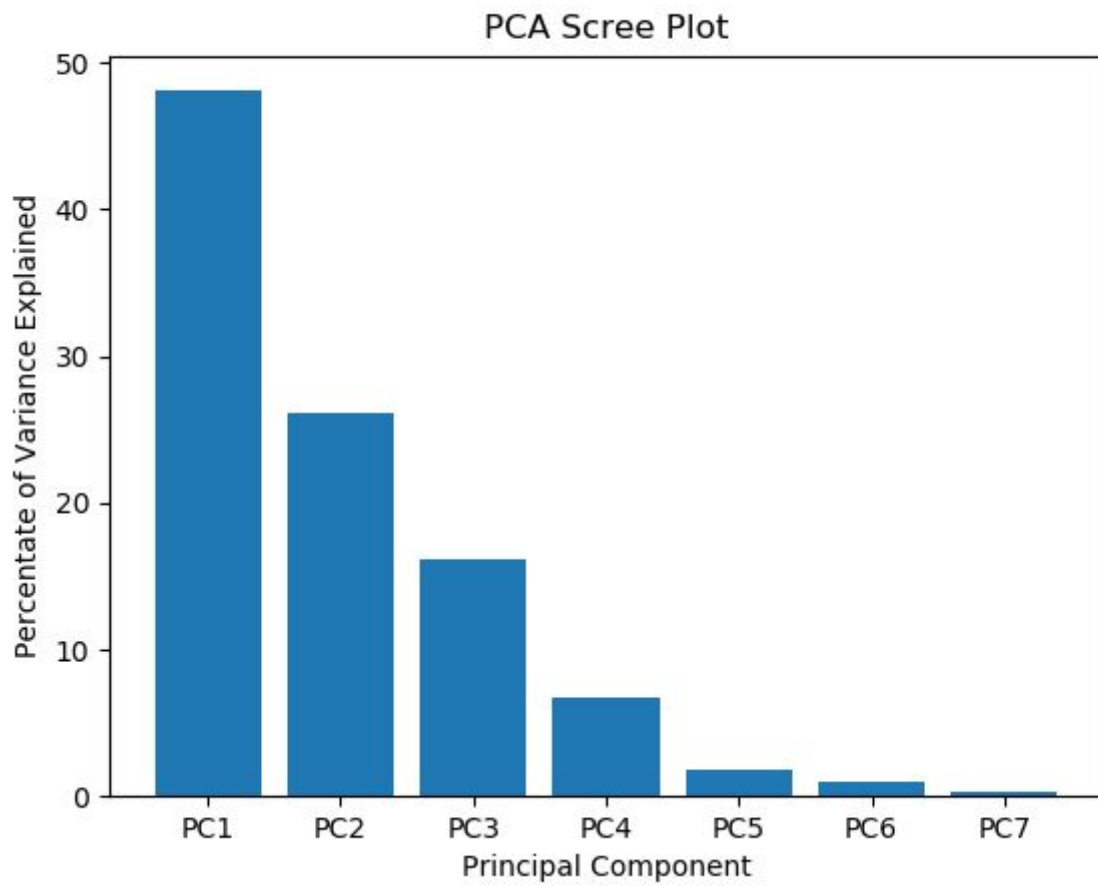
mk\_test\_z\_value: [0.74019068 -0.49665188 0.21939867 0.00084594 -0.26242113 -0.03483148 -0.29536436]

variance\_fft: [0.25238877 -0.00582185 -0.62989411 -0.0830529 0.6175511 0.0692019 -0.38267921]

meanRange: [0.27588331 0.56993938 0.06514981 -0.07366052 -0.02945538 -0.7602077 -0.10297407]

Principal Variance - Plotted below in the bar graph. The values are also mentioned below

[0.48091023      0.26060492      0.16096795      0.06677636      0.01819398      0.0093929      0.00315362]



Using the Principal Components we can rank the features which are -

1	mk_test_trend
2	mk_test_z_value
3	mk_test_p_value
4	variance_fft
5	MeanRange
6	OutOfRange
7	meanOutOfRange

## Reference:

1. Design Trend: Mann-Kendall: [https://vsp.pnnl.gov/help/Vsample/Design\\_Trend\\_Mann\\_Kendall.htm](https://vsp.pnnl.gov/help/Vsample/Design_Trend_Mann_Kendall.htm)
2. Implementation of Mann Kendall Test: <https://github.com/mps9506/Mann-Kendall-Trend>