Data Analysis of DiaHelp user

Internet of things

December 21, 2022

1 Introduction

For this data analysis we will have to interpret the results for the data we have, and since it is not real data, it may show incoherent results. Here we start at a point where we have a new measurement found on *measurement.json*.

2 Reading the new measurement

Since the measurement we get is received in json format, we use the library `rjson` to import this data to the script. Then we transform this data to a data frame and select the wanted columns.

```
> library(rjson)
> js <- fromJSON(file = "../measurement.json")
> df <- as.data.frame(js)
> df <- df[,c("phone_number", "age", "gender", "full_name", "sensors.name"
+       ,"sensors.data", "sensors.unit_of_measurement"
+       ,"sensors.name.1", "sensors.data.1", "sensors.unit_of_measurement.1"
+       ,"sensors.name.2", "sensors.data.2", "sensors.unit_of_measurement.2"
+       ,"sensors.name.3", "sensors.data.3", "sensors.unit_of_measurement.3")]
```

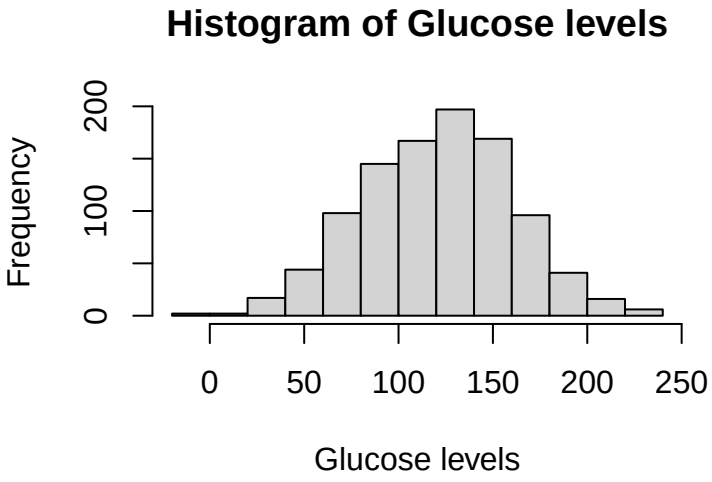
3 Simulating data

Medical data is sensitive and hard to find a good dataset on the Internet, that is the reason why we had to simulate our data, though, and the end of the day we have generated random data.
We needed an historic of the user that is why we have generated 1000 measurements for each sensor. All the different data follows a normal distribution, and it is clearly seen when we plot an histogram of the variable of the measurement for each sensor.

Glucose levels simulation

We have decided to generate random data following a normal distribution with mean of 120 and a standard deviation of 40. This numbers are an estimation done by ourselves after reading information on https://www.healthline.com/health/diabetes/blood-sugar-level-chart#recommended-ranges.

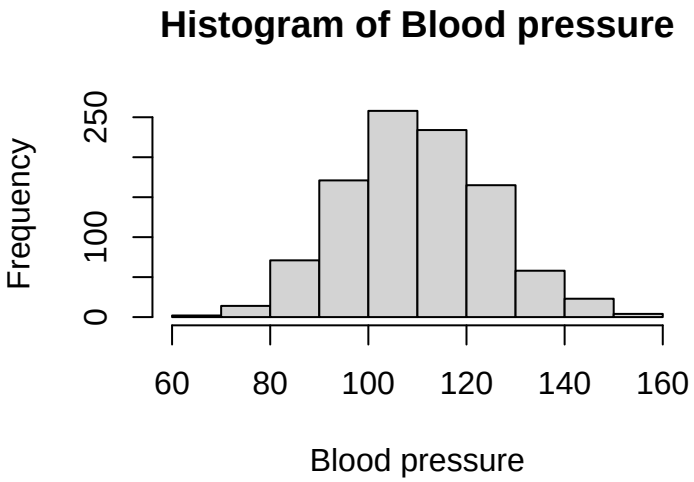
```
> sim_gc <- rnorm(1000, mean = 120, sd = 40)
> hist(sim_gc, xlab = "Glucose levels", main = "Histogram of Glucose levels")
```



Blood pressure simulation

We have decided to generate random data following a normal distribution with mean of 110 and a standard deviation of 15. This numbers are an estimation done by ourselves after reading information on https://www.emedicinehealth.com/what_is_a_normal_blood_pressure_range_by_age/article_em.htm.

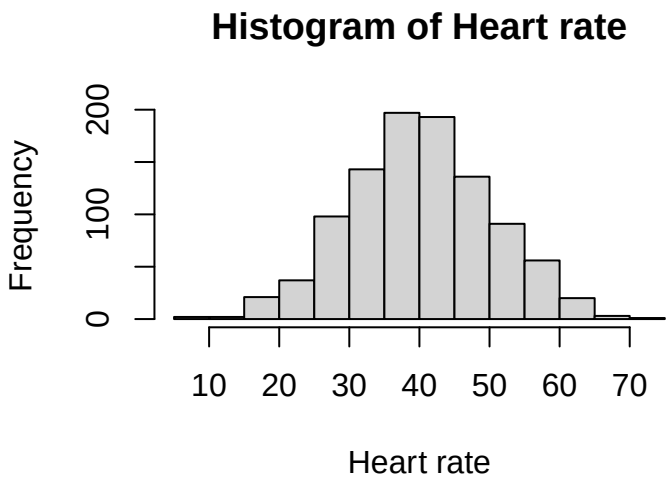
```
> sim_bp <- rnorm(1000, mean = 110, sd = 15)
> hist(sim_bp, xlab = "Blood pressure", main = "Histogram of Blood pressure")
```



Heart rate simulation

We have decided to generate random data following a normal distribution with mean of 40 and a standard deviation of 10. This numbers are an estimation done by ourselves after reading information on https://www.whoop.com/thelocker/normal-hrv-range-age-gender/.

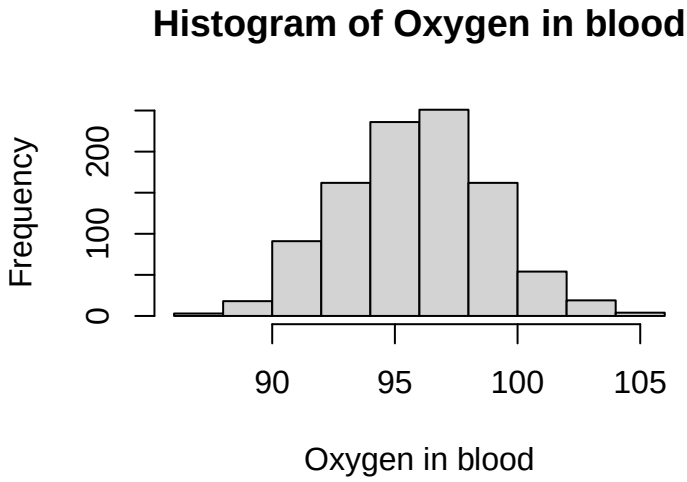
```
> sim_hr <- rnorm(1000, mean = 40, sd = 10)
> hist(sim_hr, xlab = "Heart rate", main = "Histogram of Heart rate")
```



Oxygen in blood simulation

We have decided to generate random data following a normal distribution with mean of 96 and a standard deviation of 3. This numbers are an estimation done by ourselves after reading information on https://www.emedicinehealth.com/what_is_a_good_oxygen_rate_by_age/article_em.htm.

```
> sim_ob <- rnorm(1000, mean = 96, sd = 3)
> hist(sim_ob, xlab = "Oxygen in blood", main = "Histogram of Oxygen in blood")
```



Merging all new data

Now we want to have all this data together as we could have it in our database, so merge all the data in a data frame. This is the replacement of the historic of the user that we don't have.

```
> historic <- data.frame(
+   "glucose levels (mg/dl)" = sim_gc
+   ,"blood pressure (mm/Hg)" = sim_bp
+   ,"heart rate (HRV)" = sim_hr
+   ,"oxygen levels (HRV)" = sim_ob
+   ,check.names=FALSE
+ )
```

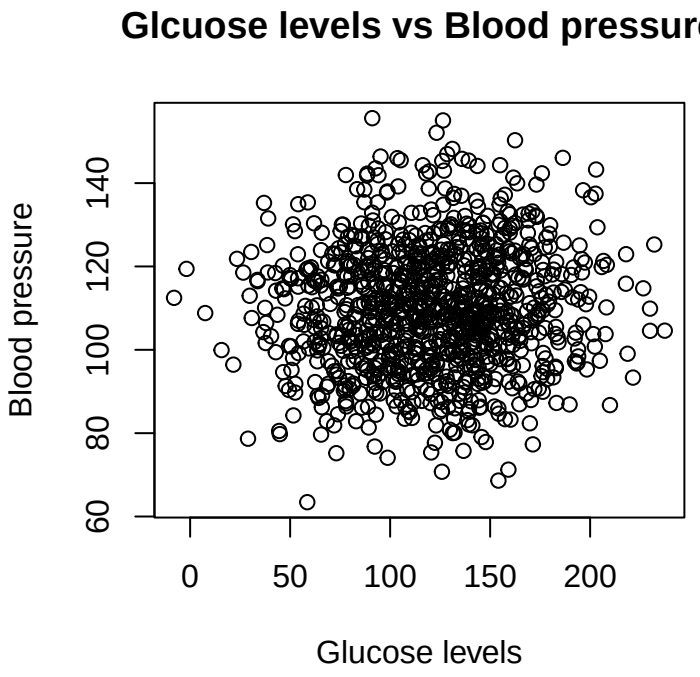
4 Correlation study between variables

Our project is focused on diabetic people, that is why we studied the glucose levels against the other variables to see if there is any correlation between them. This has been done performing a Pearson Hypothesis Test, where the null hypothesis H0 is "there is not correlation", and the alternative hypothesis H1, "there is correlation" between the variables of the measurements.

Glucose levels vs. Blood pressure

First of all a plot, with glucose levels on x-axis and blood pressure on y-axis.

```
> plot(historic$`glucose levels (mg/dl)`, historic$`blood pressure (mm/Hg)`,
+      xlab = "Glucose levels", ylab = "Blood pressure",
+      main = "Glcuose levels vs Blood pressure")
```



From the plot we can already expect there won't be any correlation. With the correlation test we can confirm that.

```
> test_gc <- cor.test(historic$`glucose levels (mg/dl)`, historic$`blood pressure (mm/Hg)`)
> print(test_gc)
```

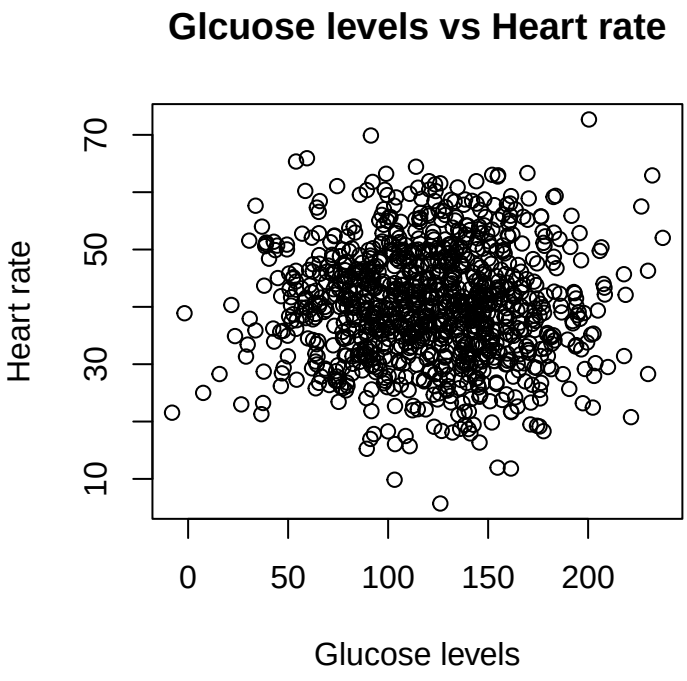
```
##
##  Pearson's product-moment correlation
##
## data:  historic$`glucose levels (mg/dl)` and historic$`blood pressure (mm/Hg)`
## t = 1.8026, df = 998, p-value = 0.07176
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.005043824  0.118541517
## sample estimates:
##        cor
## 0.05696707
```

The correlation coefficient between the two vectors turns out to be 0.0569671. A positive correlation would be near 1, a negative one near -1, and no correlation near 0. The p-value is 0.0717555, higher than 0.05, so we decide to believe that we cannot fail to reject the null hypothesis, hence there is no correlation between the glucose levels and the blood pressure.

Glucose levels vs. Heart rate

First of all a plot, with glucose levels on x-axis and heart rate on y-axis.

```
> plot(historic$`glucose levels (mg/dl)`, historic$`heart rate (HRV)`,
+      xlab = "Glucose levels", ylab = "Heart rate",
+      main = "Glcuose levels vs Heart rate")
```



From the plot we can already expect there won't be any correlation. With the correlation test we can confirm that.

```
> test_hr <- cor.test(historic$`glucose levels (mg/dl)`, historic$`heart rate (HRV)`)
> print(test_hr)

##
```

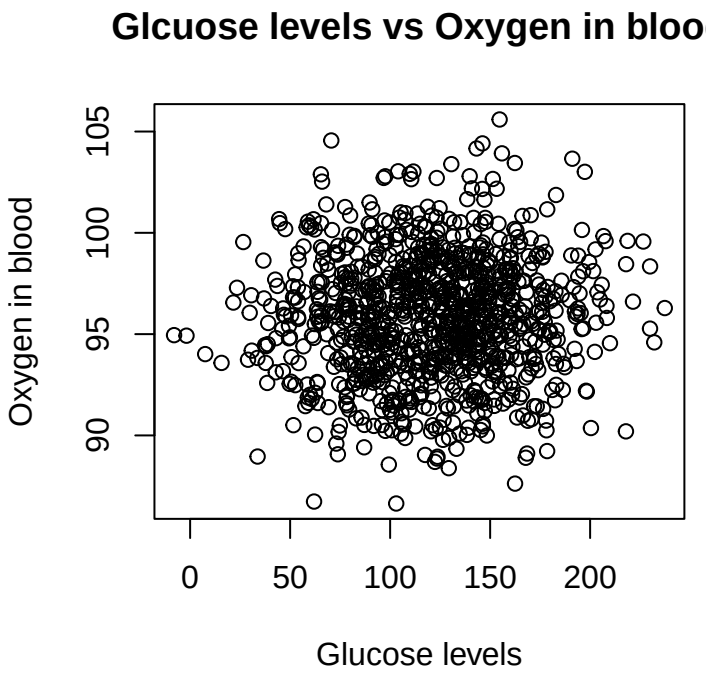
```
##  Pearson's product-moment correlation
##
## data:  historic$`glucose levels (mg/dl)` and historic$`heart rate (HRV)`
## t = -0.40618, df = 998, p-value = 0.6847
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.07478986  0.04917589
## sample estimates:
##         cor
## -0.01285638
```

The correlation coefficient between the two vectors turns out to be -0.0128564. A positive correlation would be near 1, a negative one near -1, and no correlation near 0. The p-value is 0.6846964, higher than 0.05, so we decide to believe that we cannot fail to reject the null hypothesis, hence there is no correlation between the glucose levels and the blood pressure.

Glucose levels vs. Oxygen in blood

First of all a plot, with glucose levels on x-axis and oxygen levels on y-axis.

```
> plot(historic$`glucose levels (mg/dl)`, historic$`oxygen levels (HRV)`,
+      xlab = "Glucose levels", ylab = "Oxygen in blood",
+      main = "Glcuose levels vs Oxygen in blood")
```



From the plot we can already expect there won't be any correlation. With the correlation test we can confirm that.

```
> test_ol <- cor.test(historic$`glucose levels (mg/dl)`, historic$`oxygen levels (HRV)`)
> print(test_ol)

##
##  Pearson's product-moment correlation
```

```
##
## data:  historic$`glucose levels (mg/dl)` and historic$`oxygen levels (HRV)`
## t = 1.2135, df = 998, p-value = 0.2252
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.0236639  0.1001403
## sample estimates:
##       cor
## 0.0383855
```

The correlation coefficient between the two vectors turns out to be 0.0383855. A positive correlation would be near 1, a negative one near -1, and no correlation near 0. The p-value is 0.2252121, higher than 0.05, so we decide to believe that we cannot fail to reject the null hypothesis, hence there is no correlation between the glucose levels and the blood pressure.

5 Abnormal measurements

For the new measurement we have taken from the user we now want to check if the values are normal for they history because in case they are not they could be in a potential risk of an emergency situation and they could need help. So for this section we are going to construct 99% confidence intervals and raise a message in case the new value is out of the bounds of the interval.

First we need to compute the *t-score* for a 99% confidence.

```
> # length of the historic
> n <- length(historic$`glucose levels (mg/dl)`)
>
> # calculate t-value for 99% confidence
> alpha = 0.01
> degrees_of_freedom = n - 1
> t_score = qt(p=alpha/2, df=degrees_of_freedom,lower.tail=F)
```

Glucose levels

Having the *t-score* we now can construct the interval. For the glucose levels:

```
> # confidence interval for glucose levels
> X <- mean(historic$`glucose levels (mg/dl)`)
> sd <- sd(historic$`glucose levels (mg/dl)`)
> std_error <- sd / sqrt(n)
> margin_error <- t_score * std_error
> up_bound_gc <- X + margin_error
> low_bound_gc <- X - margin_error
```

The lower bound of the interval is 117.6988738, the mean is 120.9454389 and the upper bound 124.192004. And with the first data frame we built for the new measurement we can check if the new input value is in the normal values of the user.

```
> if (df$sensors.data < low_bound_gc || df$sensors.data > up_bound_gc)
+   print("You should check your glucose levels!")
```

The new measurement is 120 mg/dl, since it inside the interval then there is nothing to alert.

Blood pressure

For the blood pressure:

```
> # confidence interval for blood pressure
> X <- mean(historic$`blood pressure (mm/Hg)`)
> sd <- sd(historic$`blood pressure (mm/Hg)`)
> std_error <- sd / sqrt(n)
> margin_error <- t_score * std_error
> up_bound_bp <- X + margin_error
> low_bound_bp <- X - margin_error
```

The lower bound of the interval is 108.6629725, the mean is 109.8707757 and the upper bound 111.0785788. And with the first data frame we built for the new measurement we can check if the new input value is in the normal values of the user.

```
> if (df$sensors.data.1 < low_bound_bp || df$sensors.data.1 > up_bound_bp)
+   print("You should check your blood pressure!")

## [1] "You should check your blood pressure!"
```

The new measurement is 105 mm/Hg. In this situation the value is lower than the norm, so we notify the user to check his blood pressure.

Heart rate

For the heart rate:

```
> # confidence interval for heart rate
> X <- mean(historic$`heart rate (HRV)`)
> sd <- sd(historic$`heart rate (HRV)`)
> std_error <- sd / sqrt(n)
> margin_error <- t_score * std_error
> up_bound_hr <- X + margin_error
> low_bound_hr <- X - margin_error
```

The lower bound of the interval is 39.3102688, the mean is 40.1295305 and the upper bound 40.9487923. And with the first data frame we built for the new measurement we can check if the new input value is in the normal values of the user.

```
> if (df$sensors.data.2 < low_bound_hr || df$sensors.data.2 > up_bound_hr)
+   print("You should check your heart rate!")
```

The new measurement is 40 HRV, since it inside the interval then there is nothing to alert.

Oxygen in blood

For the oxygen in blood:

```
> # confidence interval for oxygen levels
> X <- mean(historic$`oxygen levels (HRV)`)
> sd <- sd(historic$`oxygen levels (HRV)`)
> std_error <- sd / sqrt(n)
> margin_error <- t_score * std_error
> up_bound_ol <- X + margin_error
> low_bound_ol <- X - margin_error
```

The lower bound of the interval is 95.5900838, the mean is 95.8344372 and the upper bound 96.0787907. And with the first data frame we built for the new measurement we can check if the new input value is in the normal values of the user.

```
> if (df$sensors.data.3 < low_bound_ol || df$sensors.data.3 > up_bound_ol)
+   print("You should check your oxygen levels!")
```

The new measurement is 96 HRV, since it inside the interval then there is nothing to alert.

6 Improve our data

This new measurement can be used in the future to prevent unwanted situation, that is why we add it to the historic of the user.

```
> new_measurement <- c(df$sensors.data, df$sensors.data.1, df$sensors.data.2, df$sensors.data.3)
> historic <- rbind(historic, new_measurement)
```

7 Results

All the results shown in this analysis are the result of the randomly generated data with Gaussian distributions. That is why they don't reflect reality and we do not take responsibility for any misuse or misinterpretation of these results.
Since the data has been generated using a normal distribution, and independently one from another variable, it makes sense they are not correlated at all, however, we would probably see some if the data was from a real user. Another problem arises with the confidence intervals, which they are quite narrow due to how the data has been accuired.