Project Sweet

How glucose test can help everybody to stable energy rate in his body

Yoni G'etahun

2022-11-11

Intro

Lately I have been reading Prof Segal and Prof Elinav's book about personalized nutrition.

I discovered that beyond the general nutritional laws that apply to the majority of the population, there are personal laws. These laws depend, among other things, on our lifestyle, our genetics and the microbiome in our body.

Therefore, on the recommendation of their book, I set out on a journey to discover the best diet for me. This is the travel diary of my experience using sugar measurements as a tool for better health.

Although this is a personal experiment, you can do it on yourself and with this code to analyze your glucose result.

Goal

In this project, I had a special self goal: Improve my nutrition. Using the method from the book, I decided to take a glucose test and test my glucose and see what meals are good for me, given my lifestyle.

Method

For 2 weeks, I have been using the glucose meter kit 'FreeStyle' to test my meals (including some snacks), and also my morning and night result. I used sheets to record my meals, tests, sleep and walk and create a small data set.

Ideal meal check has 5 tests: One with the first bite, and 4 later with half hour gap. Later, in order to save test and focus on the major points I started testing twice after each meal with 1 hour gap. All along I did test before and after my sleep.

Then, I tried to find any correlation or effect which needed more data (a quite similar meal).

This is a BI project, not a statistical test. The Data is too small, inconsistent and not precociously measured for a formal experiment.

Structure

The data is made of 2 data sheets I made:

- **sweets**: data per test
- **day_score**: data per day ingredients

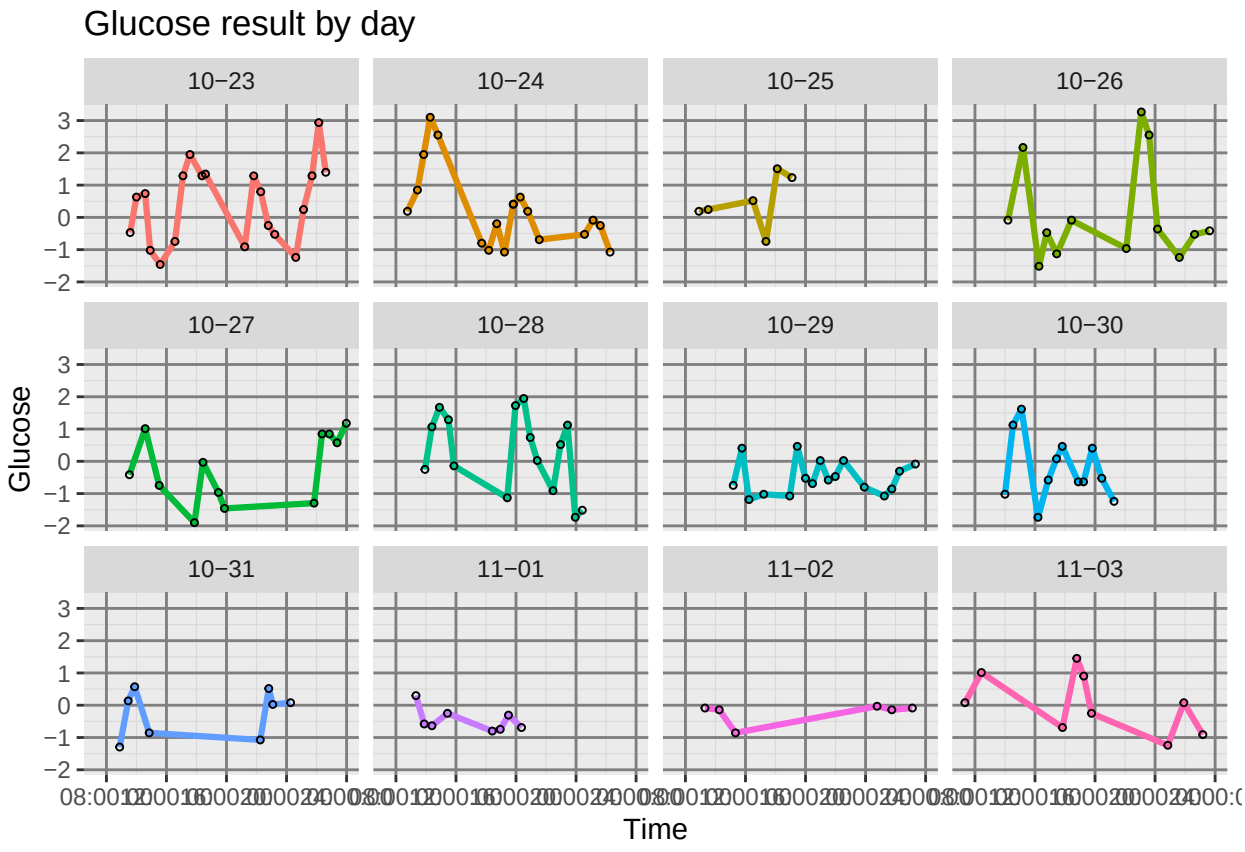
Later you will see:

- **main_df**: combined data before splitting the ingredients
- **Ingredients_df**: combined data 'before splitting the ingredients with split column by ingredients

Average day

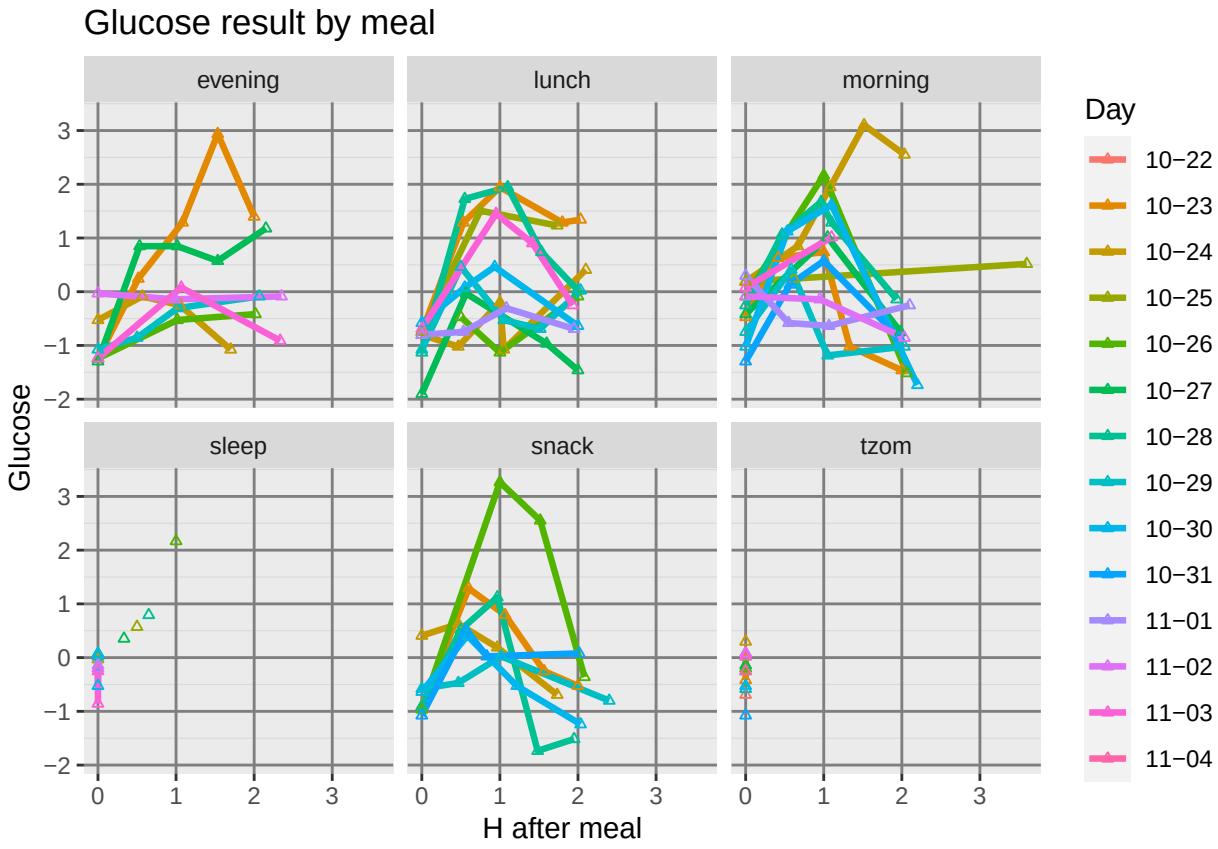
Here we can see an average day. From this graph, I can choose which meal and day to look for looking for a good or bad example

```
sweets %>%
  filter(!is.null(Glucose), !Dinner %in% c('tzom', 'sleep'),) %>%
  mutate(Date = format(Date, "%m-%d")) %>%
  mutate(Date= as.factor(Date)) %>%
  ggplot(aes(x= Time, y= Glucose, color= Date))+
  geom_line(size= 1.1)+ geom_point(size= 0.9, shape= 1, color= "Black")+
  facet_wrap(~Date,3)+
  guides(color=guide_legend(title="Day"))+ xlab("Time")+
  background_grid(minor = "yx", color.major= "gray50")+
    theme(legend.position = "none")+ggtitle("Glucose result by day")
```




```
sweets %>%
  filter(!is.null(Glucose),!Date %in% c('2022-10-22','2022-11-04')) %>%
  mutate(Date = format(Date, "%m-%d")) %>%
  mutate(Date= as.factor(Date),
         Glucose= scale(Glucose)) %>% ##,
  ggplot(aes(x= Time_after_meal, y= Glucose, color= Date))+
  geom_line(size= 1.15)+ geom_point(size= 1.1, shape= 2)+facet_wrap(~Dinner)+
```

```
  guides(color=guide_legend(title="Day"))+ xlab("H after meal")+
  background_grid(minor = "y", color.major= "gray50")+
  ggtitle("Glucose result by meal")
```



From this data, I can see which meal is my "to go" and which us a big no.

For example, worst snacks, or best breakfast:

```
main_df%>%
  filter(Dinner== 'snack', Type != 'zero') %>%
  mutate(Time_after_meal= round(Time_after_meal,2)) %>%
  select(ID,Date, Glucose, Type, Ingredients, Time_after_meal) %>%
  arrange(desc(Glucose)) %>% slice(1:5)
```

```
##   ID       Date   Glucose Type Ingredients Time_after_meal
## 1 61 2022-10-26 3.2665187  1st      shalva            1.00
## 2 62 2022-10-26 2.5519477  2nd      shalva            1.52
## 3 15 2022-10-23 1.2877066  1st   ice coffe            0.60
## 4 95 2022-10-28 1.1228055  2nd   ice cream            0.97
## 5 16 2022-10-23 0.7930035  2nd   ice coffe            1.07
```

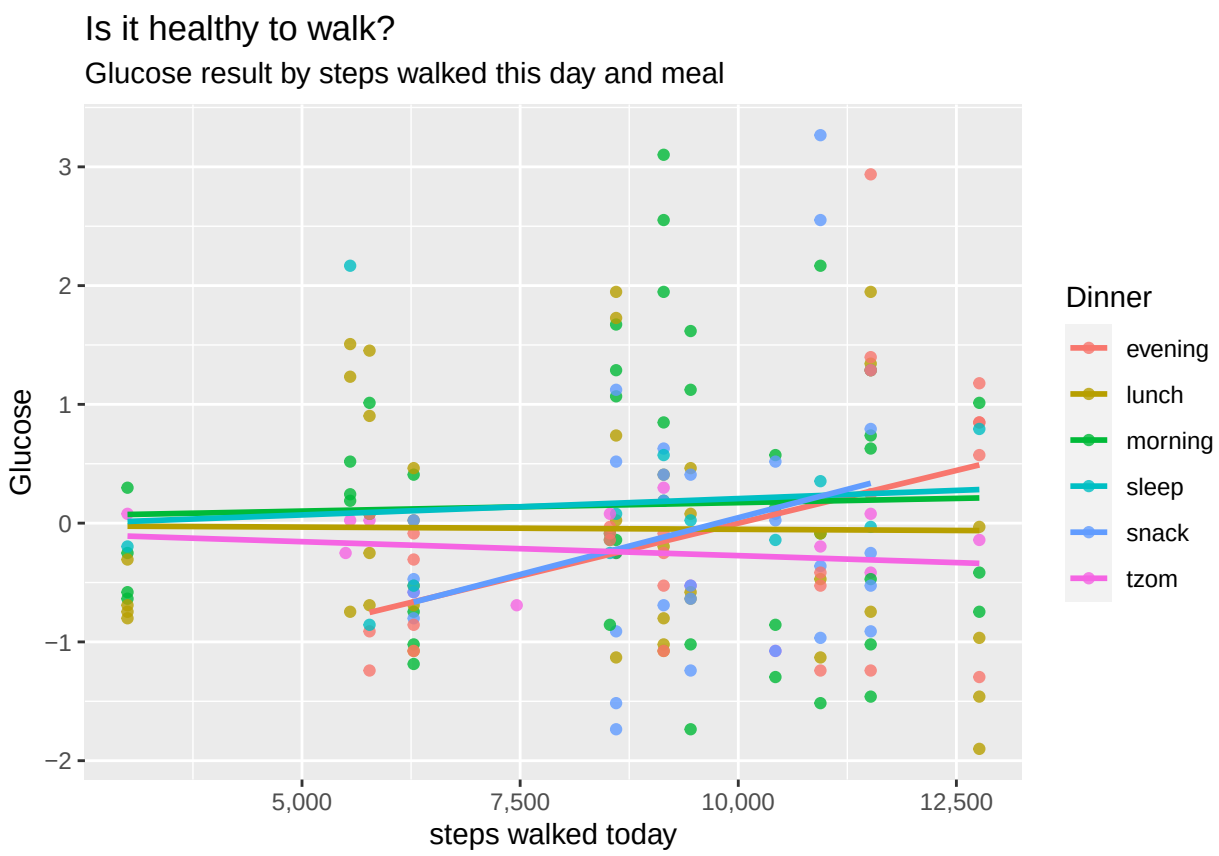
```
main_df%>%
  filter(Dinner== 'morning', Type != 'zero') %>%
  mutate(Time_after_meal= round(Time_after_meal,2)) %>%
  select(ID,Date, Glucose, Type, Ingredients, Time_after_meal) %>%
  arrange(Glucose) %>% slice(1:5)
```

```
##    ID       Date   Glucose Type
```

```
## 1 122 2022-10-30 -1.735479  4rs
## 2  56 2022-10-26 -1.515611  4rs
## 3   8 2022-10-23 -1.460644  4rs
## 4 102 2022-10-29 -1.185809  2nd
## 5   7 2022-10-23 -1.020908  3rd
##                                          Ingredients Time_after_meal
## 1 oatmeal, jam, peanut butter, walnut, popcorn, coffe            2.20
## 2                  Oatmeal, jam, peanut butter, coffe            2.07
## 3                              Oatmeal, coffe, tehini            2.00
## 4                salad, tofu, olive oil, mushroom,tea            1.05
## 5                        Oatmeal, coffe, tehini, jam             1.33
```

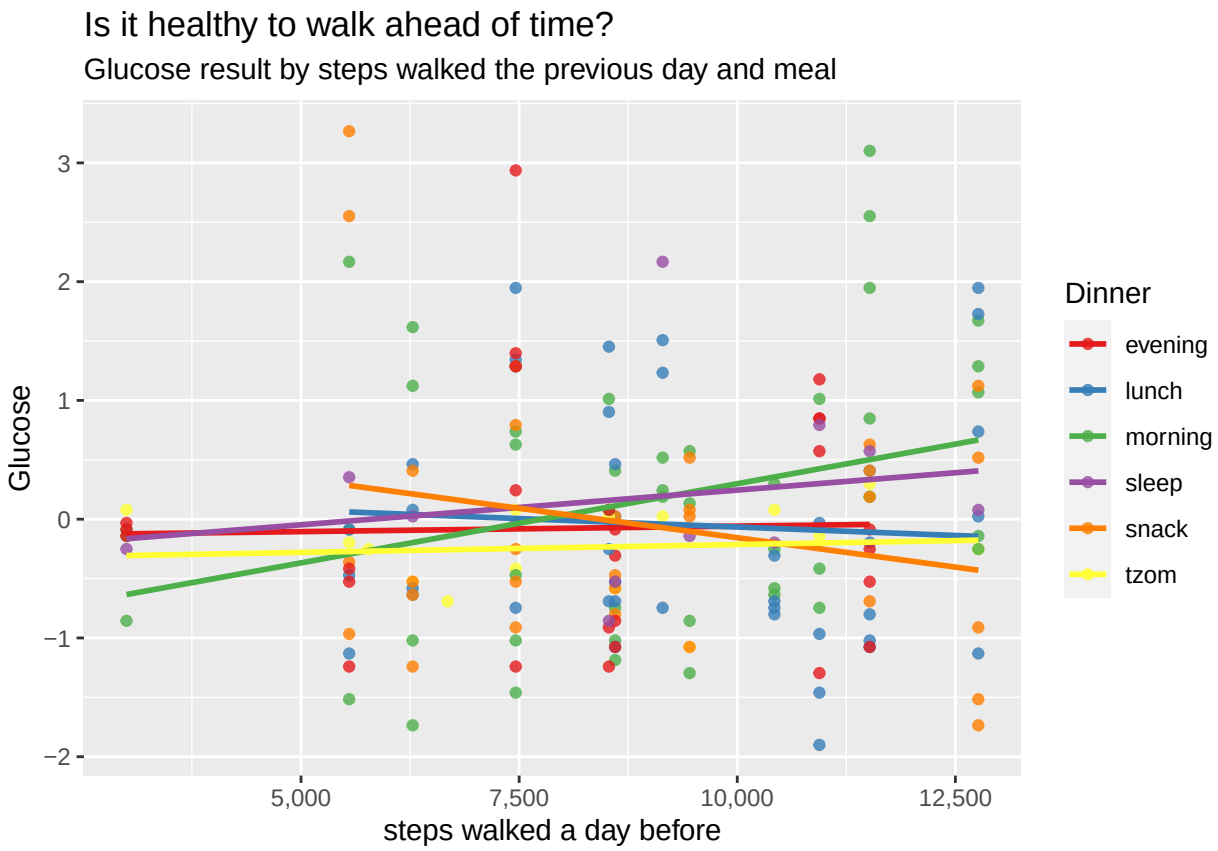
Here I try to find a general effect of walkin/ sleaping on the glucose

```
main_df%>%
  mutate(active= walking_before== "T"| walking_while== "T" ) %>%
  ggplot(aes(y= Glucose, x= Walk_today, color= Dinner))+
  geom_point(alpha= 0.8)+
  geom_smooth(aes(x= Walk_today, y= Glucose), se= F, method = "lm")+
  xlab("steps walked today")+ggtitle("Is it healthy to walk?",
  subtitle = "Glucose result by steps walked this day and meal")+
  scale_x_continuous(labels= scales::comma)
```




```
main_df%>%
  mutate(active= walking_before== "T"| walking_while== "T" ) %>%
  ggplot(aes(y= Glucose, x= Walke_d_before, color= Dinner))+
  geom_point(alpha= 0.8)+
  geom_smooth( se= F, method = "lm")+xlab("steps walked a day before")+
```

```
  ggtitle("Is it healthy to walk ahead of time?",
  subtitle = "Glucose result by steps walked the previous day and meal")+
  scale_color_brewer(palette="Set1")+scale_x_continuous(labels= scales::comma)
```



As far as I can see with no model creations, there is no correlation.

Ingredients effect

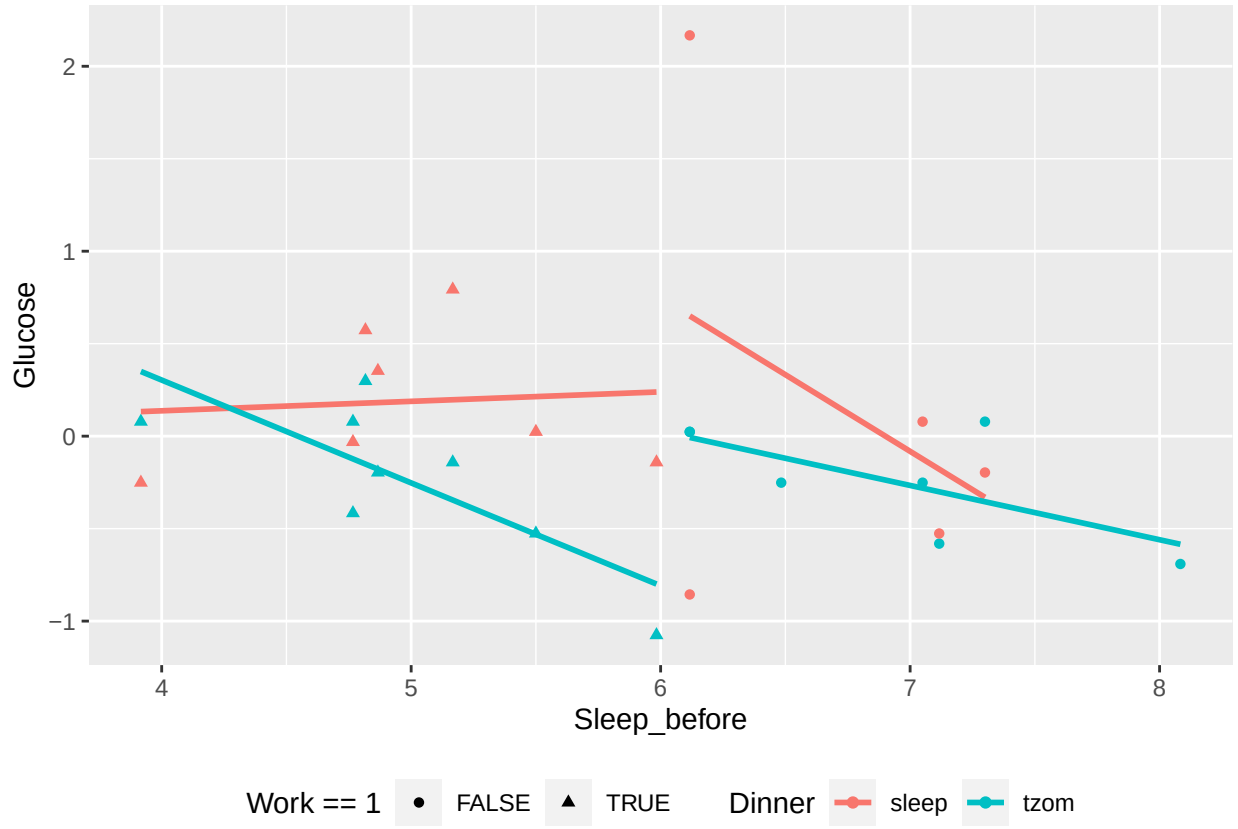
Daily effect

```
Ingredients_df %>%  select(c(1:7,9:17)) %>% sample_n(5)
```

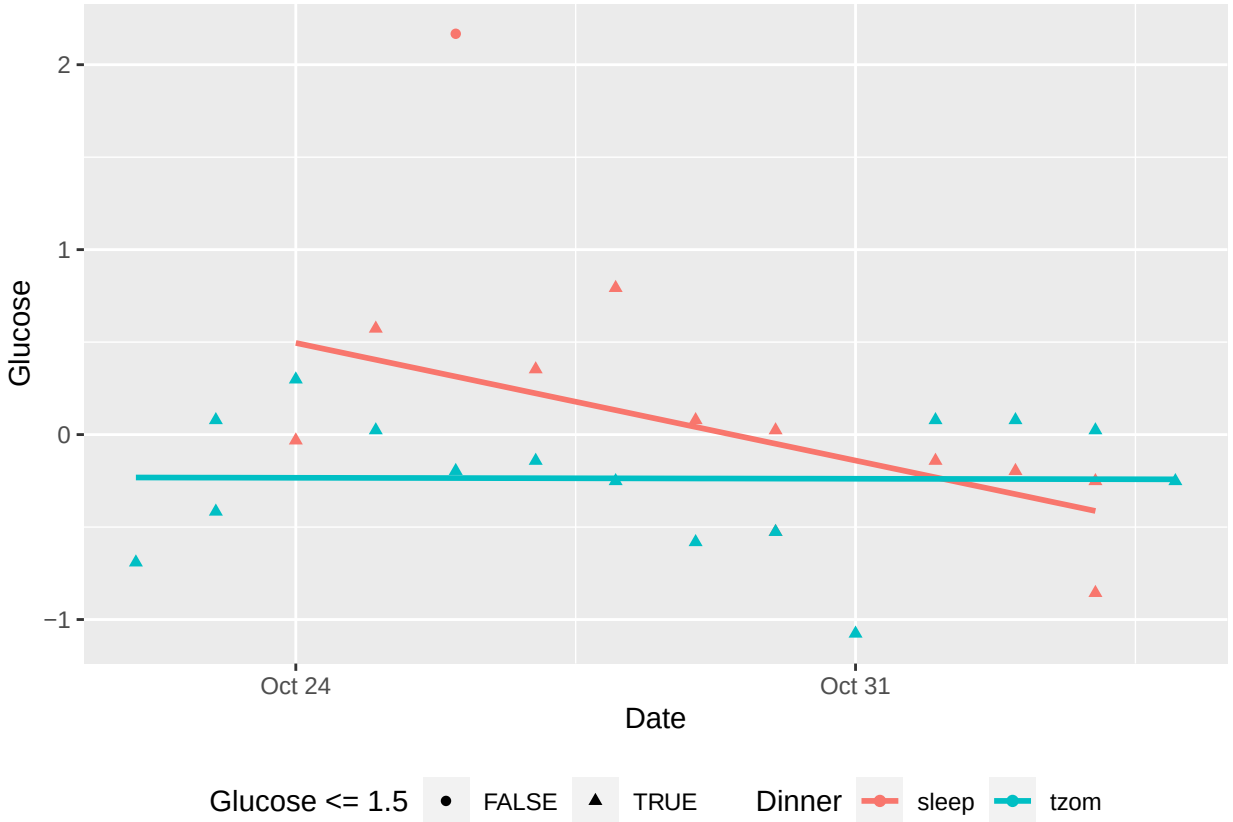
```
##    ID Day_list_date       Date     Time walking_before walking_while
## 1 143    2022-11-01 2022-11-01 09:20:00              F             F
## 2 111    2022-10-29 2022-10-29 18:32:00              F             F
## 3 104    2022-10-29 2022-10-29 14:57:00              F             F
## 4 154    2022-11-02 2022-11-02 10:15:00              F             F
## 5  67    2022-10-26 2022-10-27 00:20:00              F             F
##   Time_after_meal      Type     Glucose Walk_today Walke_d_before Sport Work
## 1       0.0000000      zero  0.29830047       3000          10426     0    0
## 2       1.0166667       2nd  0.02346545       6280           8600     0    0
## 3       0.0000000      zero -1.07587465       6280           8600     0    0
## 4       0.9500000       2nd -0.14143557       8529           3000     0    1
## 5       0.3333333 day_scale  0.35326747      10942           5552     1    1
##   Sleep_before Sleep_score has_5_ckookies
## 1     7.300000          86              0
```

```
## 2     7.116667          64              0
## 3     7.116667          64              0
## 4     3.916667          52              0
## 5     4.866667          58              0
```

```
main_df %>%
  filter(Dinner %in% c('sleep', 'tzom')) %>%
  ggplot(aes(y= Glucose, x= Sleep_before, shape= Work==1,color= Dinner))+
  geom_point()+geom_smooth(method = "lm", se= F)+
  theme(legend.position="bottom"
)
```

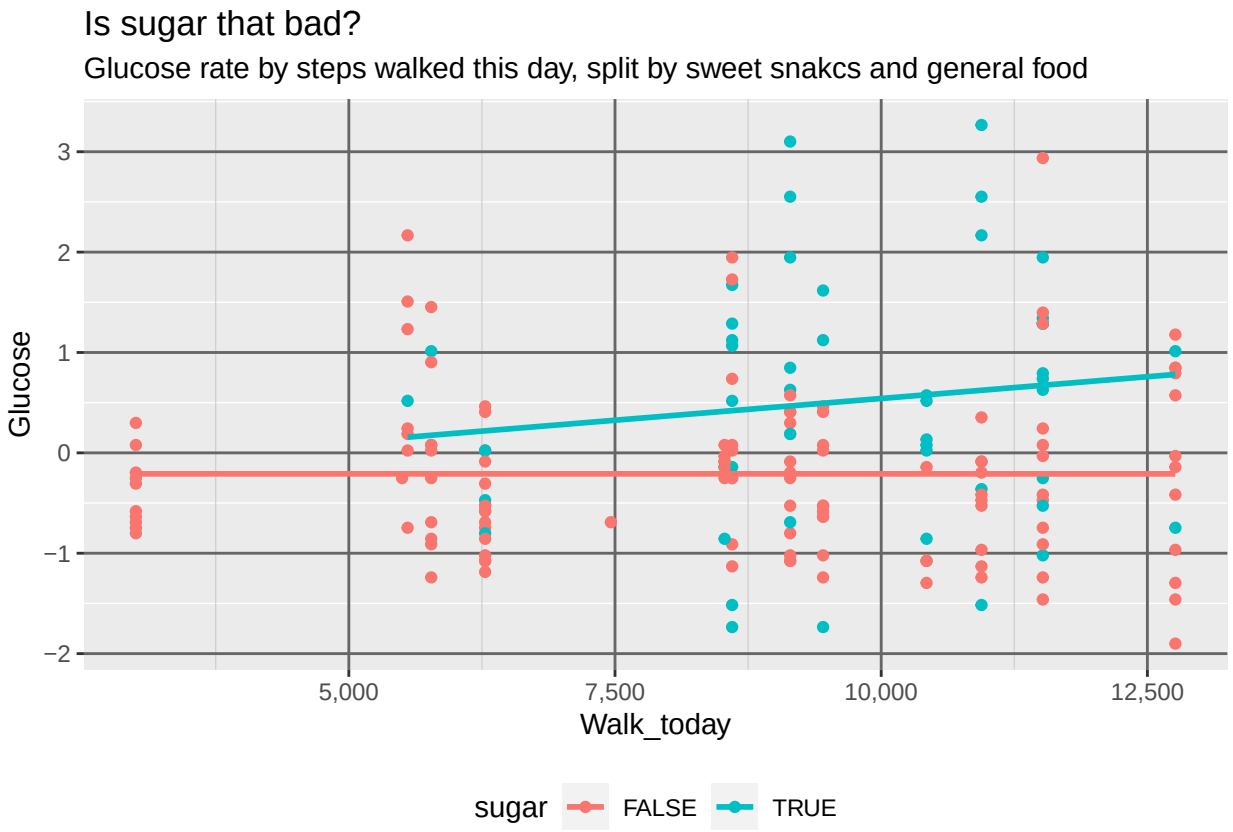



```
main_df %>%
  filter(Dinner %in% c('sleep', 'tzom')) %>%
  ggplot(aes(y= Glucose, x= Date, color= Dinner, shape= Glucose<= 1.5))+
  geom_point()+geom_smooth(se= F, method = "lm")+
theme(legend.position="bottom"
)
```



I can see a clue that sleeping better might help reduce morning glucose rate, and also that in general, my study are of reducing night glucose rate is fine, even removing one elevating point.

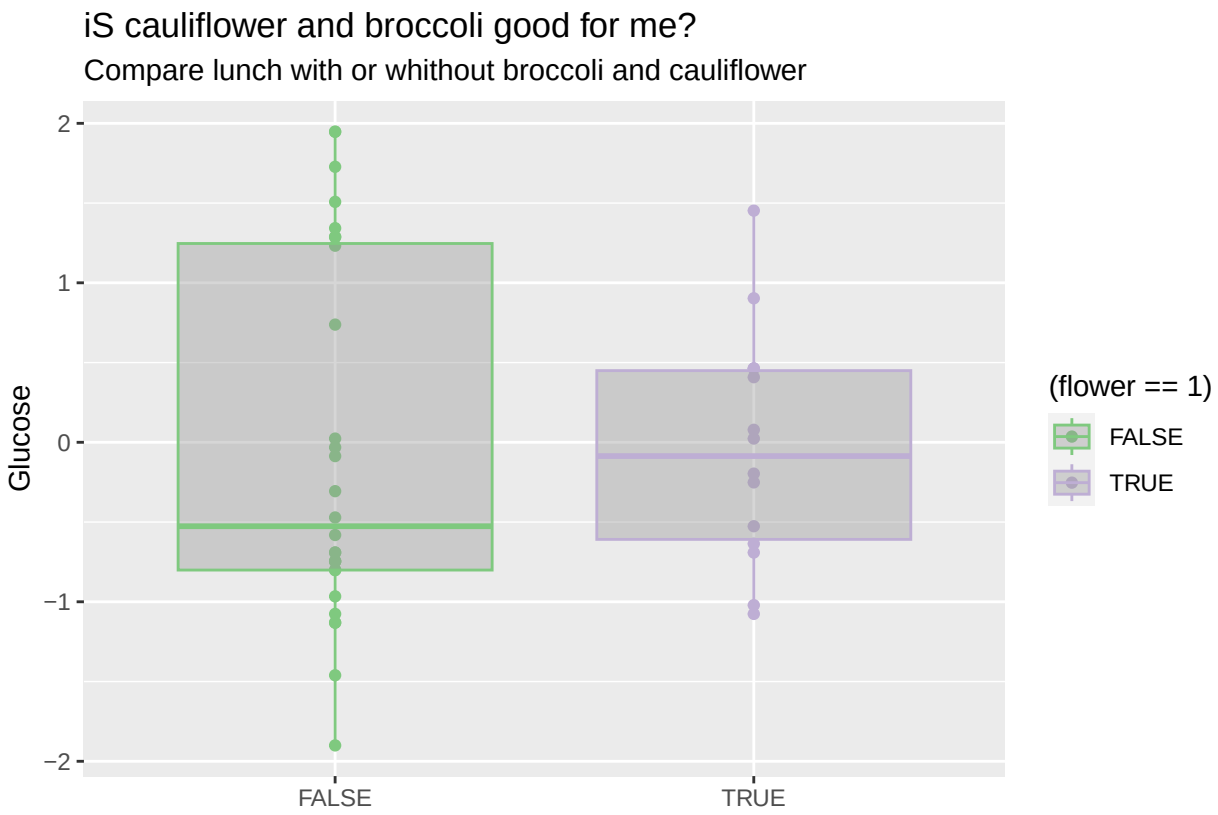
```
Ingredients_df %>%
  mutate(sugar= has_ice_coffe |has_5_ckookies|has_waffels| has_ice_cream| has_jam|
           has_banana|has_apple| has_shalva| has_snack)%>%
  ggplot(aes(y= Glucose, x=Walk_today, color= sugar))+
  geom_point()+
  geom_smooth(alpha= 0.2, method = "lm", se= F)+
  theme(panel.grid.major = element_line(color = "gray40",
                                        size = 0.5,
                                        linetype = 1),
        panel.grid.minor.x = element_line(color = "gray80",
                                          size = 0.20,
                                          linetype = 1),
        legend.position="bottom"
        )+scale_x_continuous(labels= scales::comma)+
  ggtitle("Is sugar that bad?",subtitle ="Glucose rate by steps walked this day, split by sweet snakcs a
```



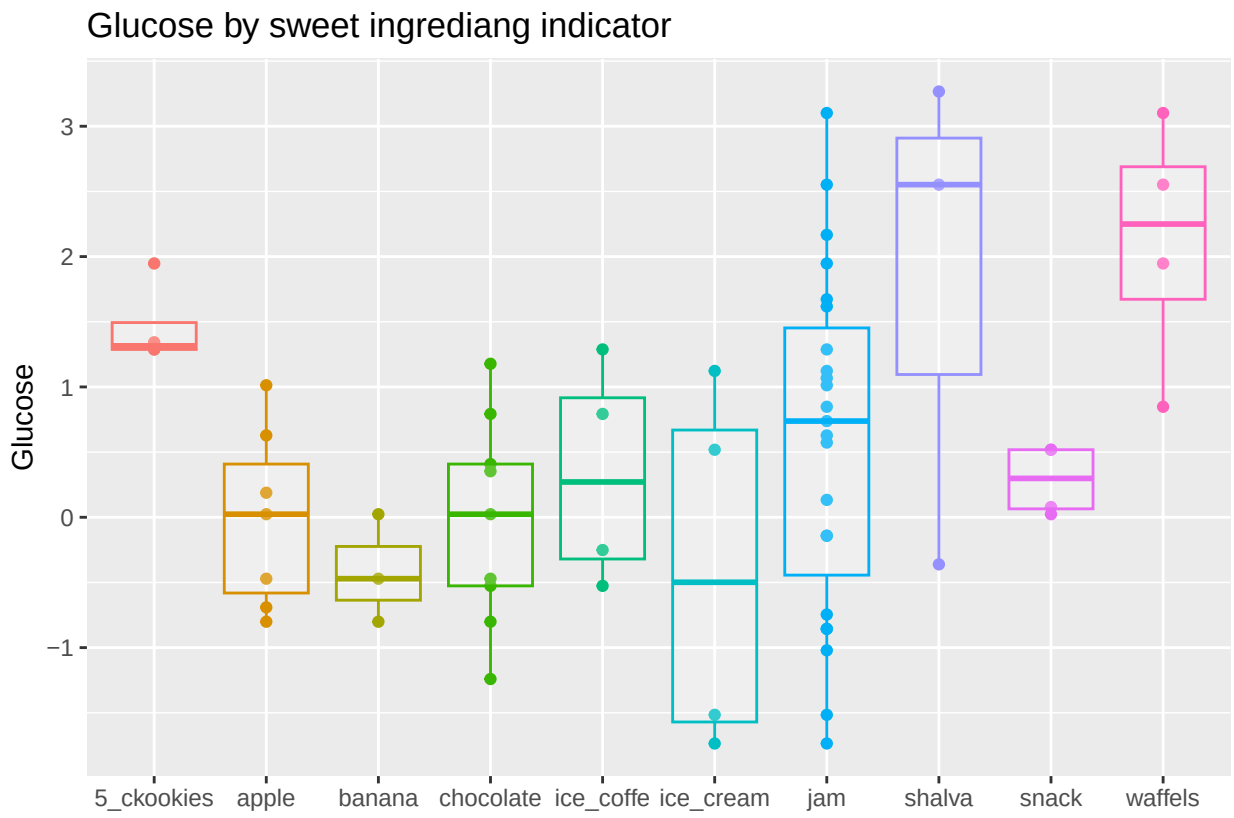
Meal effect

We can also create example of compering lunch with and without cauliflower and broccoli:

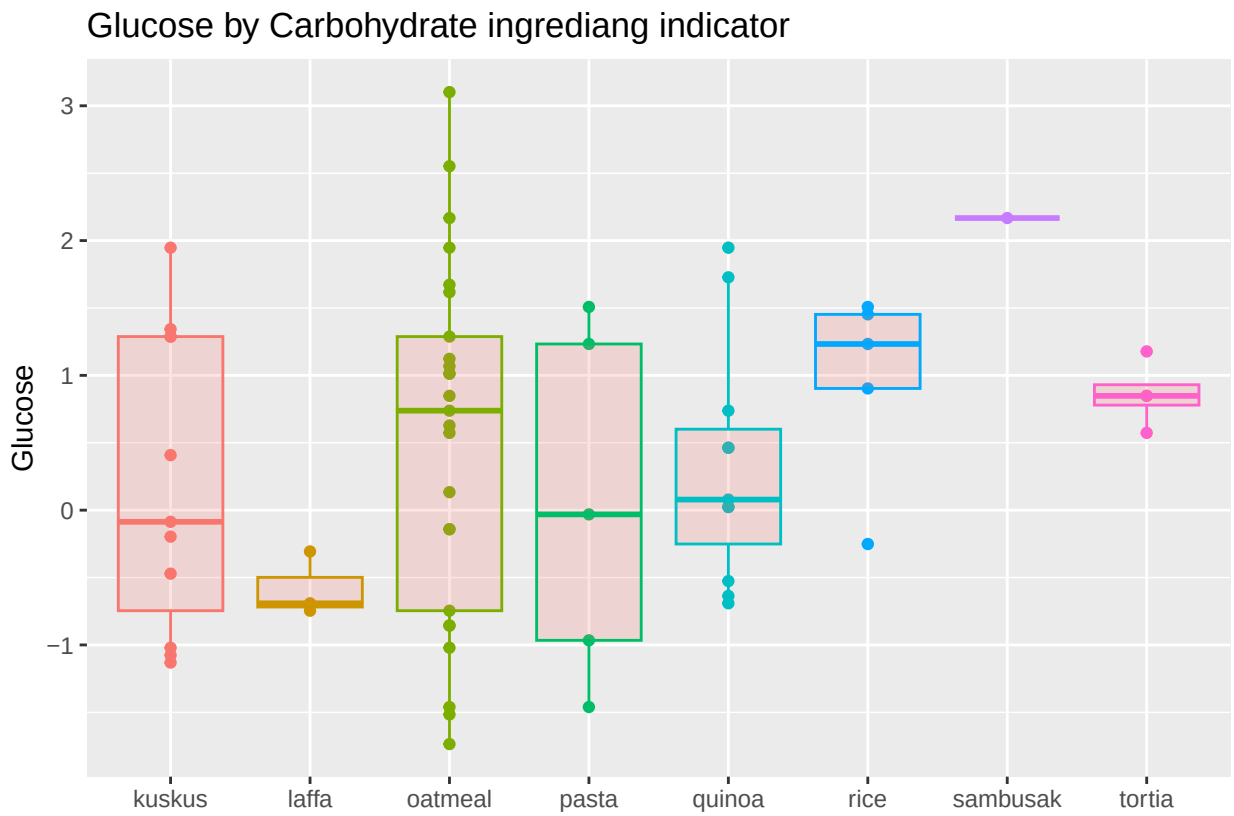
```
Ingredients_df %>%
  filter(Dinner %in% 'lunch') %>%
  mutate(fisher= has_fish| has_salmon,
         flower= has_cauliflower | has_broccoli) %>%
  ggplot(aes(y= Glucose, x= as.factor(flower),color= (flower==1)))+
  geom_point()+geom_boxplot(alpha= 0.4, fill= "Gray60")+
  ggtitle("iS cauliflower and broccoli good for me?", subtitle = "Compare lunch with or whithout broccol
  scale_color_brewer(palette="Accent")
```

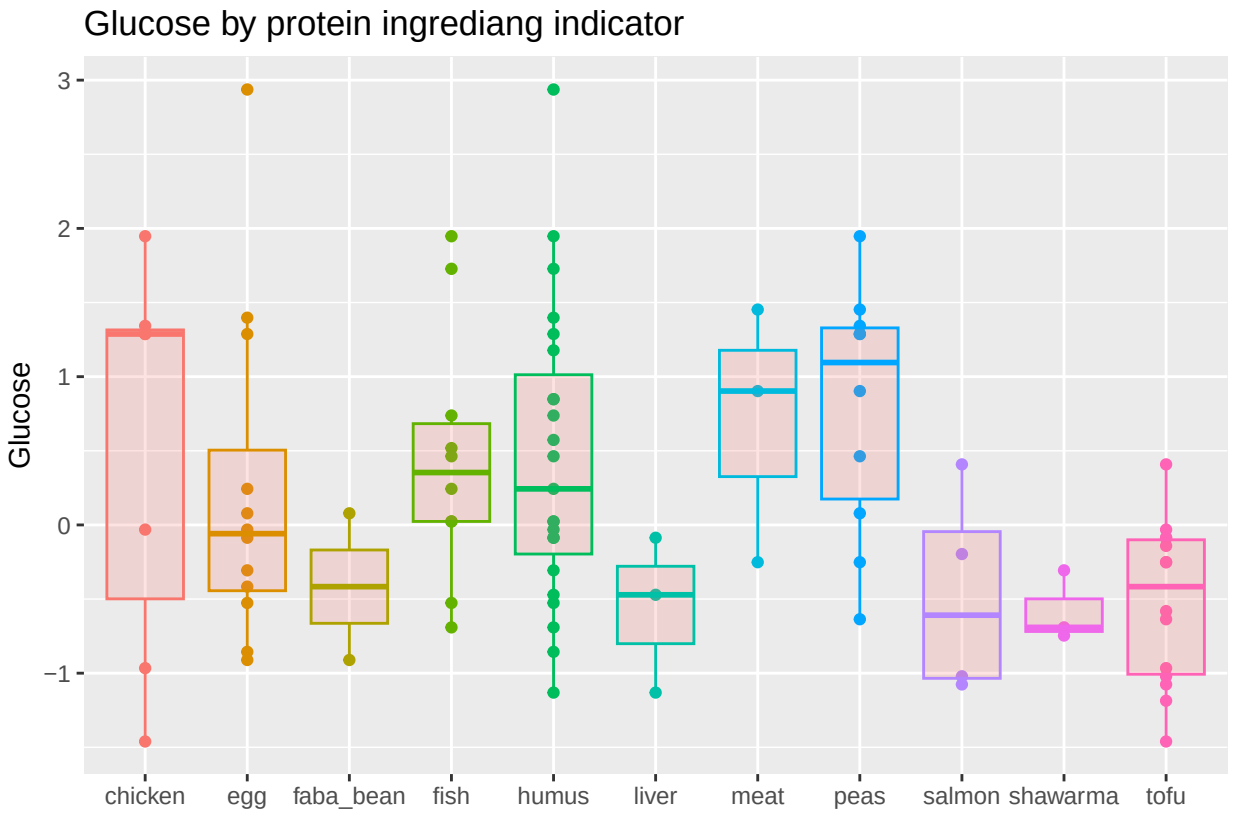


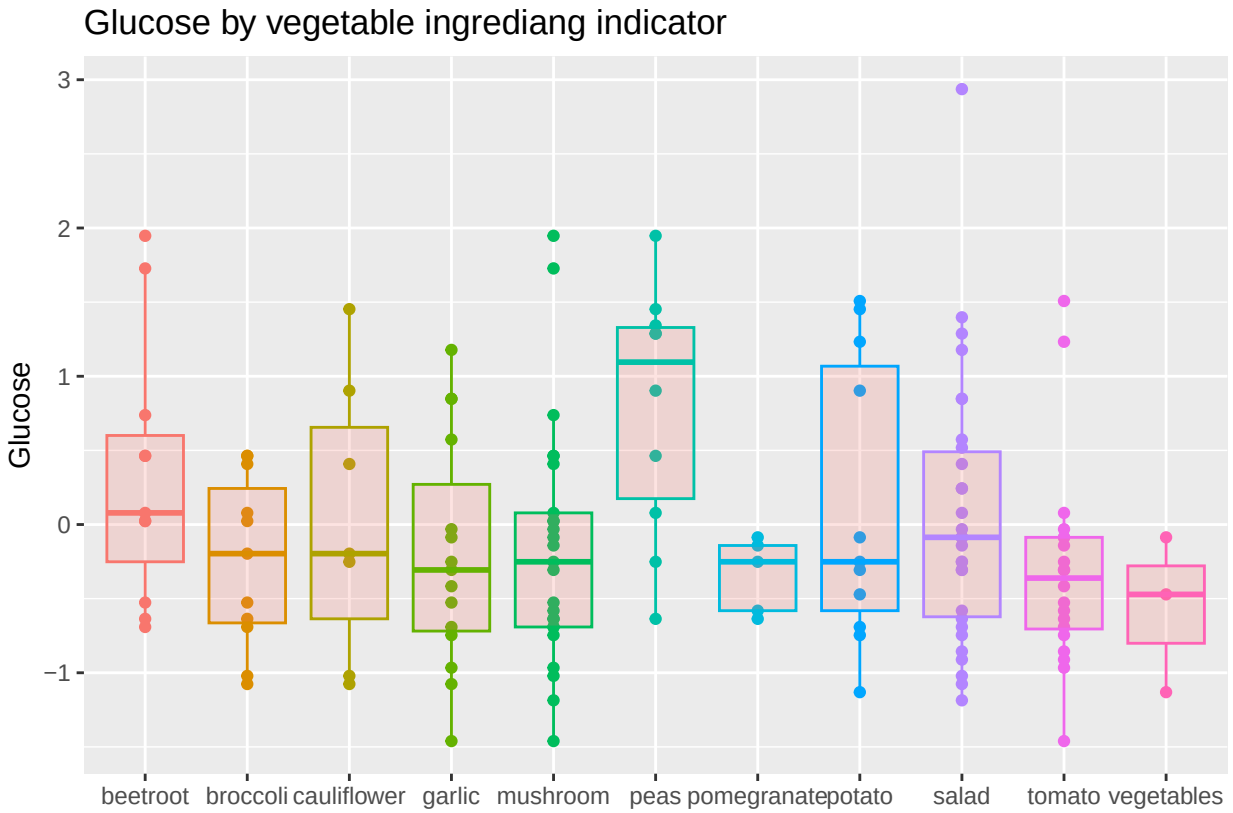
In the following 4 graphs, you will see with me how different ingredients contained in different meal's glucose result:



The same goes for those graphs:




Glucose by protein ingrediang indicator
Glucose
3
2
1
0
−1
chicken
egg
faba_bean
fish
humus
liver
meat
peas
salmon
shawarma
tofu


Glucose by vegetable ingrediang indicator
Glucose
3
2
1
0
−1
beetroot
broccoli
cauliflower
garlic
mushroom
peas
pomegranate
potato
salad
tomato
vegetables

Ingredients linear model

Ideally, I would use a statistical model in order to understand which food is better for my balance, and create a noun recommendation. Unfortunately, this is not a big enough data nor the variables noun recommendation enough. Even though, I made linear regression for myself in order to have some clue for how my body works.

```
options(scipen = 10)

baking_food<- Ingredients_df %>%
  recipe(Glucose~.) %>%
  update_role(ID,Date,Type,Time, new_role = "sider") %>%
  step_rm(has_role("sider"),Day_list_date) %>%
  step_mutate(Time_after_meal2= Time_after_meal^2,
              has_fish= has_fish|has_salmon,
              has_chicken= has_chicken| has_shawarma,
              has_sweet_snack= has_5_ckookies| has_waffels) %>%
  step_rm(has_salmon,has_vegetables, has_beetroot, has_hot,has_shawarma,has_5_ckookies,has_waffels)

baked_food<- baking_food %>% prep(Ingredients_df)%>% bake(Ingredients_df)
lm_par <- linear_reg() %>% set_mode('regression') %>% set_engine("lm")
lm_fit <- lm_par %>% fit(Glucose ~ . , baked_food)
```

Now, I can show a pick to which foods should I eat more / less

```
lm_beta<- summary(lm_fit$fit)
lm_beta
```

```
##
## Call:
## stats::lm(formula = Glucose ~ ., data = data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.71285 -0.31796 -0.01601  0.28003  1.72549
##
## Coefficients: (14 not defined because of singularities)
##                        Estimate  Std. Error t value   Pr(>|t|)
## (Intercept)          0.86605419  1.47176428   0.588   0.557397
## walking_beforeT     -0.66523582  0.24230397  -2.745   0.007023 **
## walking_whileT       0.08648980  0.42588931   0.203   0.839434
## Time_after_meal      1.23024413  0.49020825   2.510   0.013490 *
## Dinnerlunch          0.01568148  0.32186889   0.049   0.961228
## Dinnermorning        0.56728453  0.29470500   1.925   0.056731 .
## Dinnersleep          0.42930724  0.31819291   1.349   0.179945
## Dinnersnack          0.40257203  0.33925215   1.187   0.237835
## Dinnertzom           0.63165667  0.28376248   2.226   0.027980 *
## Walk_today          -0.00011246  0.00006489  -1.733   0.085782 .
## Walke_d_before       0.00004269  0.00004622   0.924   0.357624
## Sport                0.17938041  0.25770133   0.696   0.487796
## Work                -0.01168508  0.41774648  -0.028   0.977734
## Sleep_before        -0.28120193  0.16266445  -1.729   0.086566 .
## Sleep_score         -0.00377654  0.01349656  -0.280   0.780127
## has_apple            0.78045484  0.68606852   1.138   0.257683
## has_banana          -0.85628406  1.13436307  -0.755   0.451890
## has_beans            0.10537376  1.38508641   0.076   0.939491
## has_bread            1.78492156  1.79856150   0.992   0.323096
```

```
## has_broccoli          0.58004502  0.82990087   0.699   0.486017
## has_butter            3.17921842  1.72441057   1.844   0.067831 .
## has_cauliflower       1.37911119  2.16082768   0.638   0.524602
## has_cheese            1.09245637  0.70866217   1.542   0.125948
## has_chickenTRUE       1.72534551  0.53067678   3.251   0.001511 **
## has_chocolate         1.03465730  0.69551524   1.488   0.139614
## has_coffe            -1.40477222  0.31640268  -4.440 0.00002092 ***
## has_egg              -1.90079587  1.73232651  -1.097   0.274844
## has_faba_bean                NA          NA      NA         NA
## has_falafel          -0.45185760  1.54321118  -0.293   0.770205
## has_fishTRUE         -0.16249850  0.68814857  -0.236   0.813749
## has_full_bread        1.65639994  0.72395176   2.288   0.023981 *
## has_garlic           -0.48217677  0.45621122  -1.057   0.292785
## has_honey_cake        0.94762615  0.92207668   1.028   0.306263
## has_humus             2.76546478  1.42686867   1.938   0.055079 .
## has_ice_coffe         1.38413908  0.75877907   1.824   0.070746 .
## has_ice_cream         0.95334804  0.77946639   1.223   0.223823
## has_jam               0.52169407  0.60641701   0.860   0.391435
## has_kuskus           -2.14360947  1.41764095  -1.512   0.133277
## has_laffa            -0.31866940  0.68808908  -0.463   0.644160
## has_liver                    NA          NA      NA         NA
## has_mango                    NA          NA      NA         NA
## has_meat             -0.48595589  3.24841944  -0.150   0.881346
## has_milk              0.74945073  0.71346157   1.050   0.295736
## has_mushroom         -0.72382593  0.49161924  -1.472   0.143687
## has_na                0.73730386  0.78658192   0.937   0.350559
## has_oatmeal           1.37230007  0.72871762   1.883   0.062226 .
## has_olive_oil         0.57433882  0.60765577   0.945   0.346570
## has_pasta                    NA          NA      NA         NA
## has_peanut_butter    -0.04463064  0.44963854  -0.099   0.921107
## has_peas              1.31297080  0.98159805   1.338   0.183694
## has_pomegranate       0.18504273  0.68094283   0.272   0.786310
## has_popcorn          -0.14885979  0.78872799  -0.189   0.850637
## has_potato                   NA          NA      NA         NA
## has_quinoa                   NA          NA      NA         NA
## has_rice                     NA          NA      NA         NA
## has_salad                    NA          NA      NA         NA
## has_sambusak          2.49315754  0.91912316   2.713   0.007712 **
## has_shalva            2.63250106  0.80424298   3.273   0.001408 **
## has_snack             4.28795238  0.86583920   4.952 0.00000256 ***
## has_snyders          -0.82378666  0.74636680  -1.104   0.272035
## has_sugar                    NA          NA      NA         NA
## has_tea               0.35120826  0.34748997   1.011   0.314300
## has_tehini                   NA          NA      NA         NA
## has_tofu                     NA          NA      NA         NA
## has_tomato                   NA          NA      NA         NA
## has_tortia                   NA          NA      NA         NA
## has_veg_hamburger            NA          NA      NA         NA
## has_walnut           -0.20072526  0.75200378  -0.267   0.790013
## has_yogurt           -1.14931695  0.41173571  -2.791   0.006155 **
## Time_after_meal2     -0.62188752  0.17481613  -3.557   0.000547 ***
## has_sweet_snackTRUE   1.94106030  0.51217247   3.790   0.000243 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.624 on 114 degrees of freedom
```

```
## Multiple R-squared:  0.7389, Adjusted R-squared:  0.6106
## F-statistic:  5.76 on 56 and 114 DF,  p-value: 1.465e-15
```

```
lm_beta$coefficients %>%
  as.data.frame() %>% #colnames()
  rename(Pr_value= 4) %>%
  filter( Pr_value<= 0.35, Estimate<= 0) %>%
  arrange(Estimate)%>%  slice(1:7) %>% mutate(across(everything(), ~round(.,4)))%>%
  select(-'t value')
```

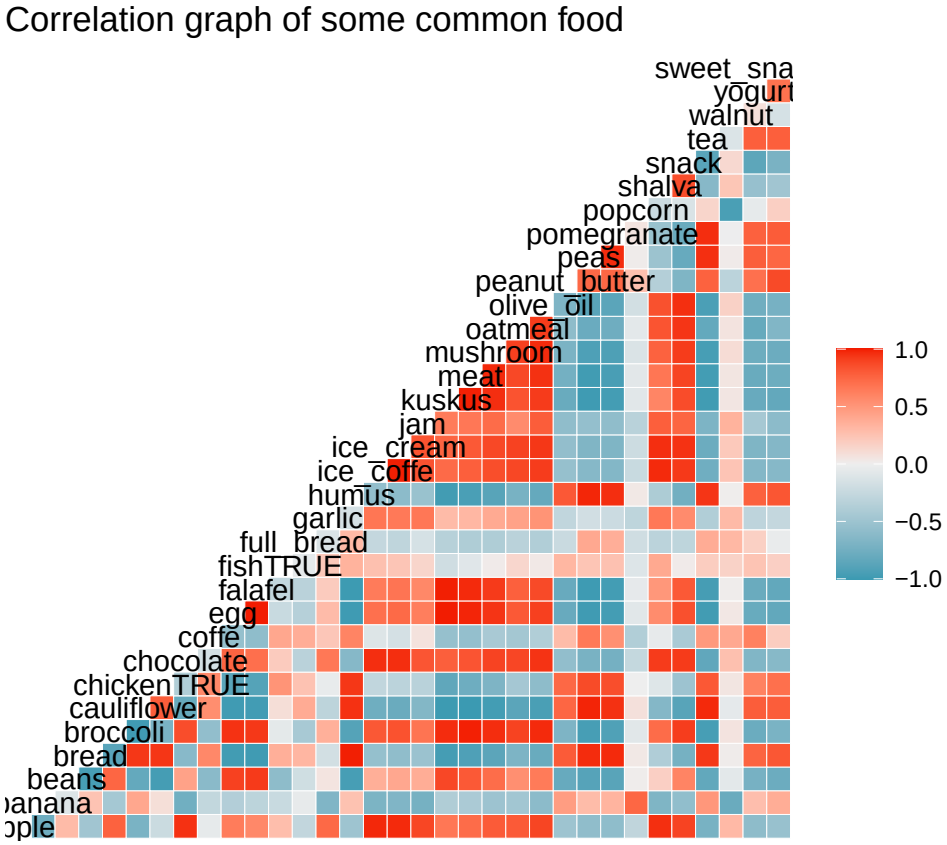
```
##                 Estimate Std. Error Pr_value
## has_kuskus       -2.1436     1.4176   0.1333
## has_egg          -1.9008     1.7323   0.2748
## has_coffe        -1.4048     0.3164   0.0000
## has_yogurt       -1.1493     0.4117   0.0062
## has_snyders      -0.8238     0.7464   0.2720
## has_mushroom     -0.7238     0.4916   0.1437
## walking_beforeT  -0.6652     0.2423   0.0070
```

```
lm_beta$coefficients %>%
  as.data.frame() %>% #colnames()
  rename(Pr_value= 4) %>%
  filter( Pr_value<= 0.35, Estimate>= 0,
          row.names(lm_beta$coefficients)!= '(Intercept)') %>%
  arrange(desc(Estimate)) %>%  slice(1:6) %>% mutate(across(everything(), ~round(.,4))) %>%
  select(-'t value')
```

```
##                     Estimate Std. Error Pr_value
## has_snack             4.2880     0.8658   0.0000
## has_butter            3.1792     1.7244   0.0678
## has_humus             2.7655     1.4269   0.0551
## has_shalva            2.6325     0.8042   0.0014
## has_sambusak          2.4932     0.9191   0.0077
## has_sweet_snackTRUE   1.9411     0.5122   0.0002
```

In a glance, with $pr(> |t|) < 0.35$ I should avoid snacks(energy snack), sambusak, shalva and sweet snacks, and use more coffee, kuskus, yogurt and egg

As said before, there is some correlations that could affect this model, and therefore the liability is quite fragile. Here are some of the corelations:



Conclusions:

After a few days in the project, I felt my energy was more balanced. I became less tired during the afternoon, evening and night.

The experiment's method forced me to eat fewer snacks, and as a result I found out how dessertless meals make me feel more full, I refrained from afternoon snacks and was more energetic later on my way home.

Moreover, I realize how adding a balanced amount of more fat and protein to my meals can stabalize my sugar rate and cause me to eat fewer calories without worrying about excess calories or carbohydrates.

Looking for a better body energy balance, I would that consider sleeping slightly more might help me maintain balance. A walk might help as well.

Now here are some food based self advice:

- Adding fat yogurt to the oatmeal can balance the healthy carbohydrates with oil to stabalize sugar.
- Moderate quantity of carbohydrates like couscous or quinoa in lunch can be fine as long as the meal contain protein & some fat.
- A fish is a good protein to add to my meals which I hadn't thought of till now.
- Ice cream is the best snack for me, which might be due to the fat and water covering the sugar. Nuts with fresh fruit are also excellent, and ice coffee is ok.
- Broccoli and cauliflower are potentially good for balancing lunch.
- After an unhealthy meal, coffee can lower my glucose, and this might be relevant to tea as well.
- The last 2 hours before sleep shouldn't contain too big meals (no sambusak before bed...)

Note!

This is a self experiment.

There is some path that might be good for most people like mixing fat in carbohydrates and spacing out meals, but might not be good for others. I hope that my self diary can help you create your own diary!

Hope for better health and energy balance for all of you!