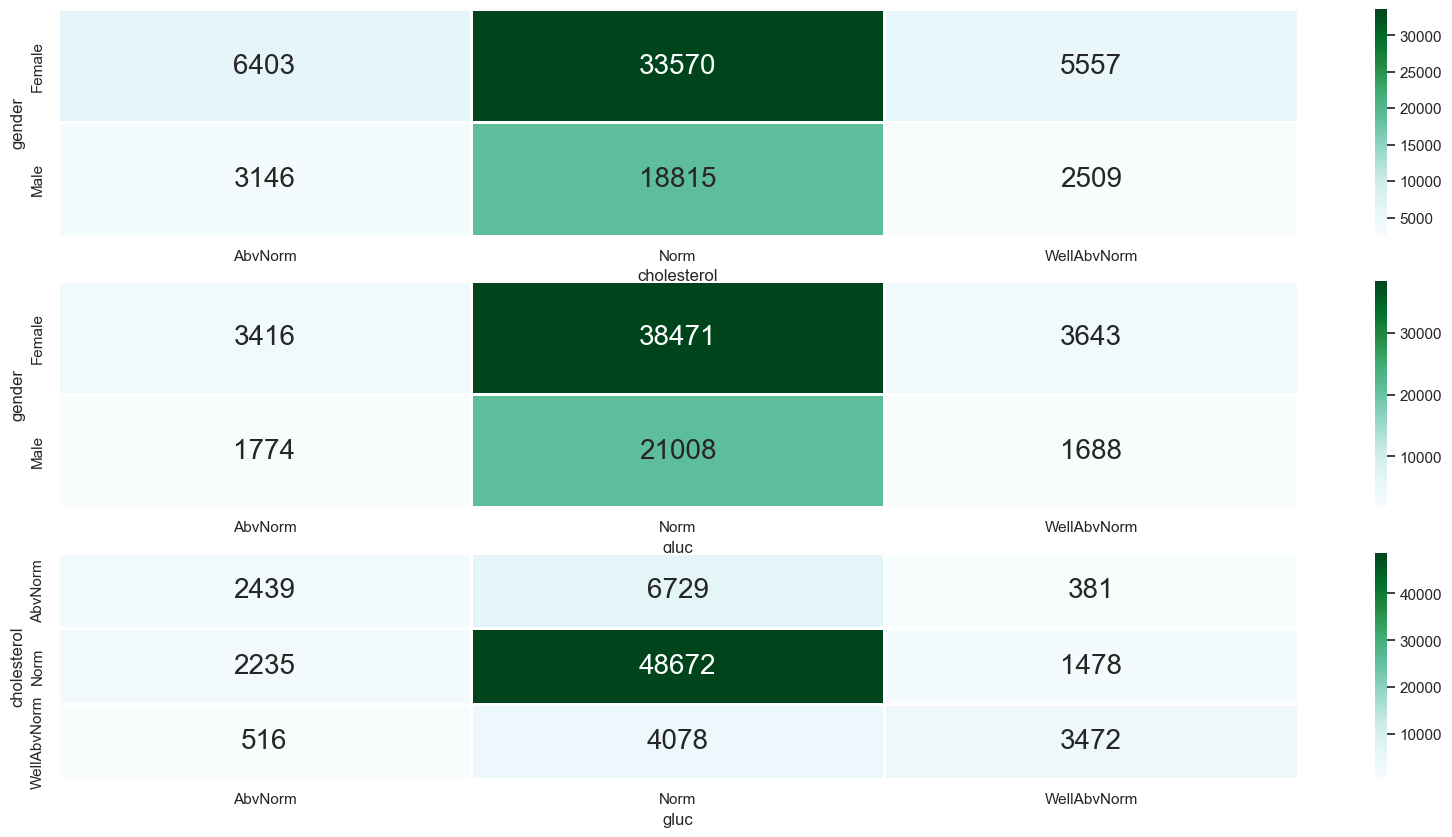
What we/I have done:

1. Motivation
   1. Try to find a dataset that has large datapoint space, and with variety of data variables.
   2. Try to find a dataset that is appropriate[Game sales, Azur Lane group formation]. This needs to be approved by the teaching set, so we decided to just use the preset ones provided in the project paper, to save time and because I do not want to talk to people.
   3. We chose Sales first, since one of our groupmates has an interest in that. But we quickly determined that it is not enough for us. Especially upon doing data analysis between the variables. The results will be incredibly mundane.
   4. The next choice after that is Cardiovascular Disease. Lots of datapoints(>10000) and lots of variables. And hence we stuck with it.
2. Data collection :: Direct from Kaggle

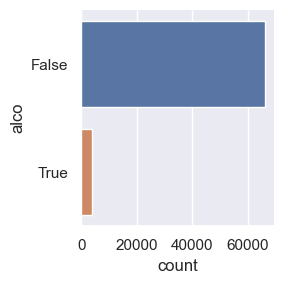
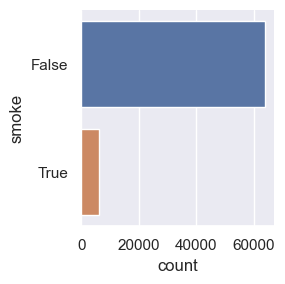
#https://www.kaggle.com/code/bhargavi35/starter-cardiovascular-disease-dataset-d2fc2521-9

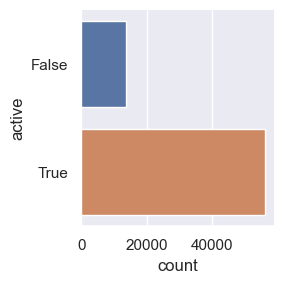
* 1. Data has quite a bit of anomalies, it is not the best dataset. It has 70,000 datapoints before trimming.
  2. Download data into workspace.
  3. Comma-Separated File(CSV), where the values are actually comma separated(ex. 34;54;180;…). Hence, I needed to get rid of the ; delimiters. Fortunately, this is very easy to do due to Pandas.
  4. Import data into Jupyter Notebook(I use vscode because I do not like localhost\_website) using Pandas’ read\_csv method. Do datapull.read(n = some x) for reading x values to double check successful porting.

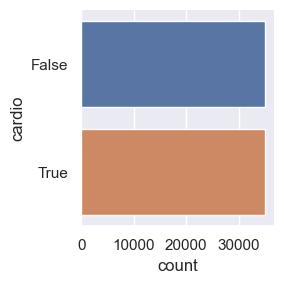
1. Data preparation I
   1. I do info(), and most importantly value\_counts() for numericals to see what we are dealing with.
   2. I assume we look at cardiovascular disease first, since this is what the data was for originally. Also, I don’t care about anything else.
   3. There is an ID column. I do some selective(I used iterated primes(2 \* 2, 3 \* 5, 7 \* 11, 13 \* (next p),…, (p\_n \* p\_n+1)..) because I have that function on hand) and see that ID is literally patient ID and has no bearing on probability of having cardiovascular disease. Makes sense in real life too. So, I throw it away.
   4. Next, I translate [0, 1] ranged data to binaries, to prevent downstream clashes with the math.
   5. Next, I standardise ‘age’ and ‘height’ to proper values of years and meter/res.
   6. Using the guide from the Kaggle page, I map [1, 2] for gender to proper gender terms [‘male’, ‘female’], then map [1, 2, 3] to the relevant states [‘norm’, ‘abvnorm’ , ‘wellabvnorm’]. Again, make my life easier, and be more clearer in data presentation for me and to the viewer. Tier I data preparation is now ready.
2. Data analysis I
   1. I start with categorical variables first.
   2. Info(), describe() whatnot. Get a basic feel of the categorical data first.
   3. Do your favourite catplot.
   4. Plot one against another with heatmap to see what influences what. You see some very generic and obvious trends.



* 1. Then I deal with the True/False stuff. Here, I get a feel of how much proportion of the dataspace is True and False for each binary variable there is.

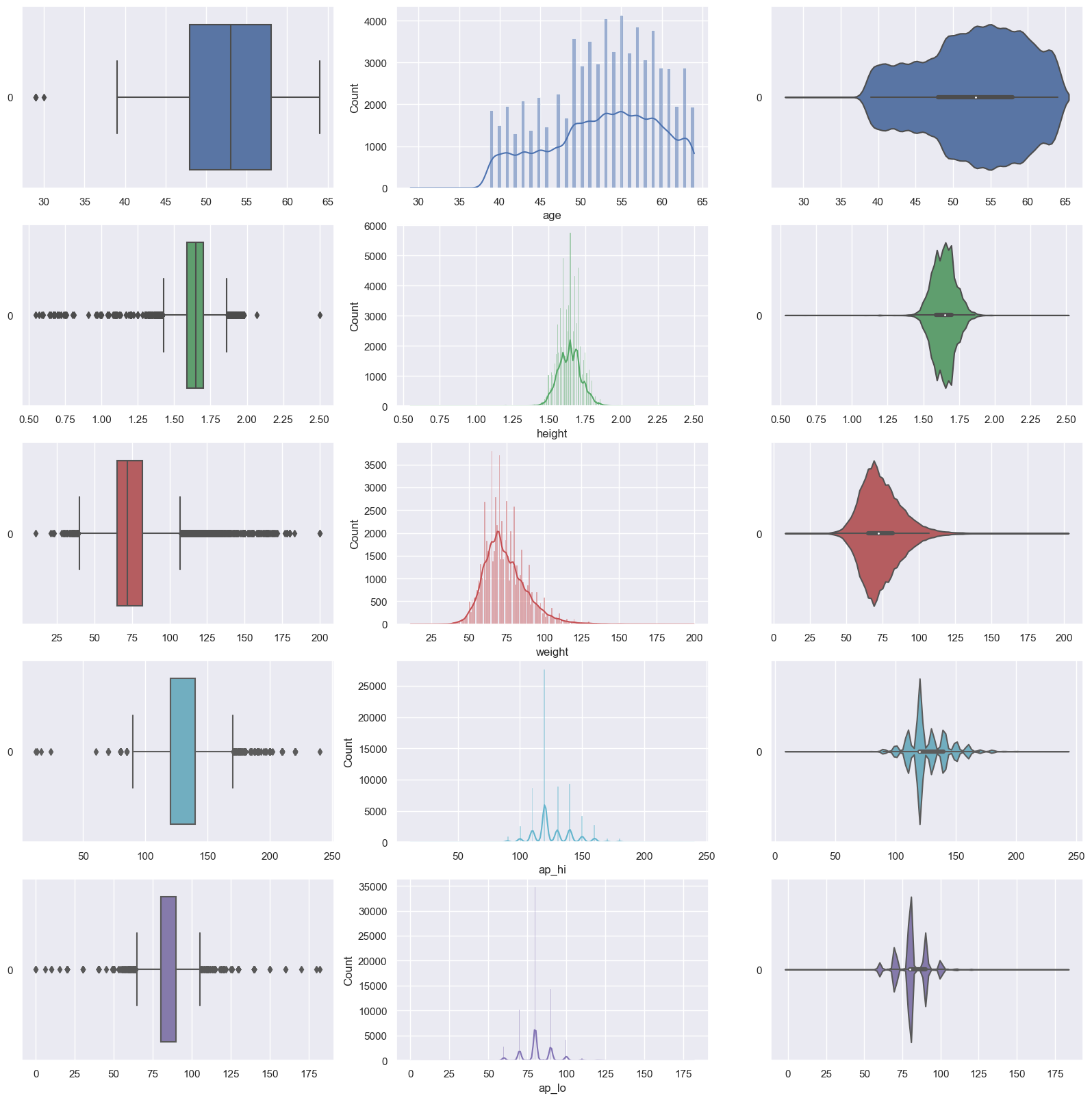




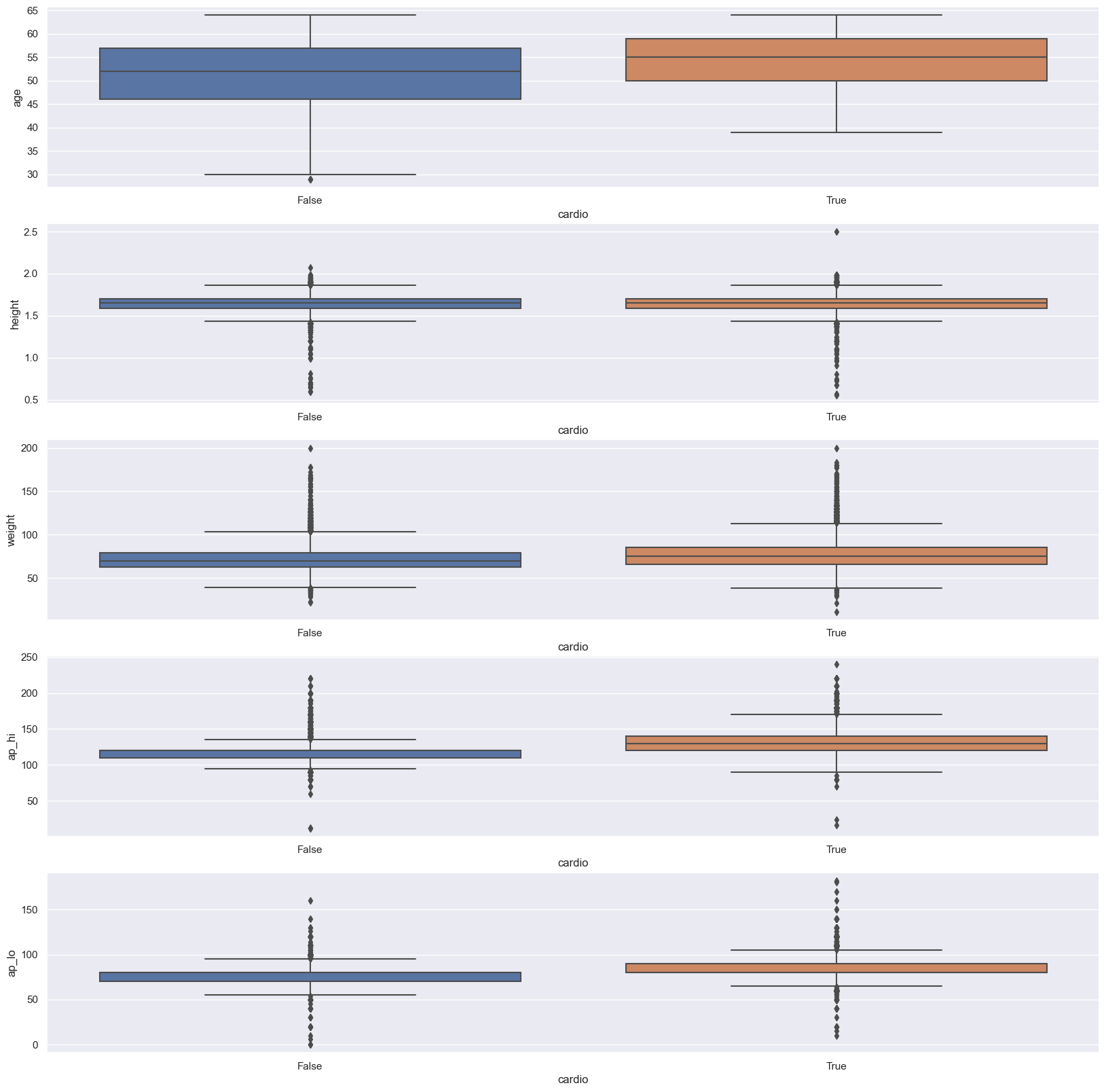


* 1. Here we see that relatively small percentage of people smoke and drink alcohol. A large proportion of people exercise. But cardio seems to be distributed 50-50 among the test subjects. This tells me that either (a) I messed up on data analysis[high probability], (b) The datapoints on binary variables is not enough(for example, “what do they mean by ‘being active?’”), or (c) Those three variables do not contribute to cardio probability, in which case I will throw it away.
  2. We will explore this further below.

1. Data preparation II
   1. Next, I pull out numericals(age, height, weight, ap\_hi and ap\_lo) and combine them with the ‘cardio’ [0 or 100] series, in preparation for seeing how the two relate. I ignore the categoricals.
   2. Running tentative data analysis revealed a massive issue with ap\_hi and ap\_lo. They range from [-11000, 16000] over. Given ap\_hi and ap\_lo represents systolic and diastolic blood pressures respectively, this data range does not make sense. I double check by printing individual value\_counts() to a txt file and confirm the completely absurd values.
   3. I then filter out rows with absurd values, throwing away a total of over 1.3K datapoints. But that is no matter, I have over 68.5K datapoints left. Now, learning my lesson, I ran value\_counts() for every variable, and determine that alles fehr gut.
2. Data analysis II[with end goal = ‘cardio’]
   1. **Numericals : [‘age’, ‘ht’, ‘wt’, ‘sys’, ‘dia’]**
      1. Now I do plotting of the boxplot, histplot and violinplot. Some data are not continuous, but that is expected.



* + 1. For now, this is how the distribution would look like.
    2. I split the individual numericals to False/True cardiovascular disease groups and do a boxplot again.



* + 1. I conjecture ‘height’ has nothing to do with the probability of having cardiovascular disease[from the boxplots]. And so, I do up another function to calculate whether that is the case or not based on the data description for each variable, comparing between False/True values.
    2. And the former showed that I was right. Height has a high chance of not playing a role. And so, I do more checking. We get this:

{'age': 20.661000000000058,

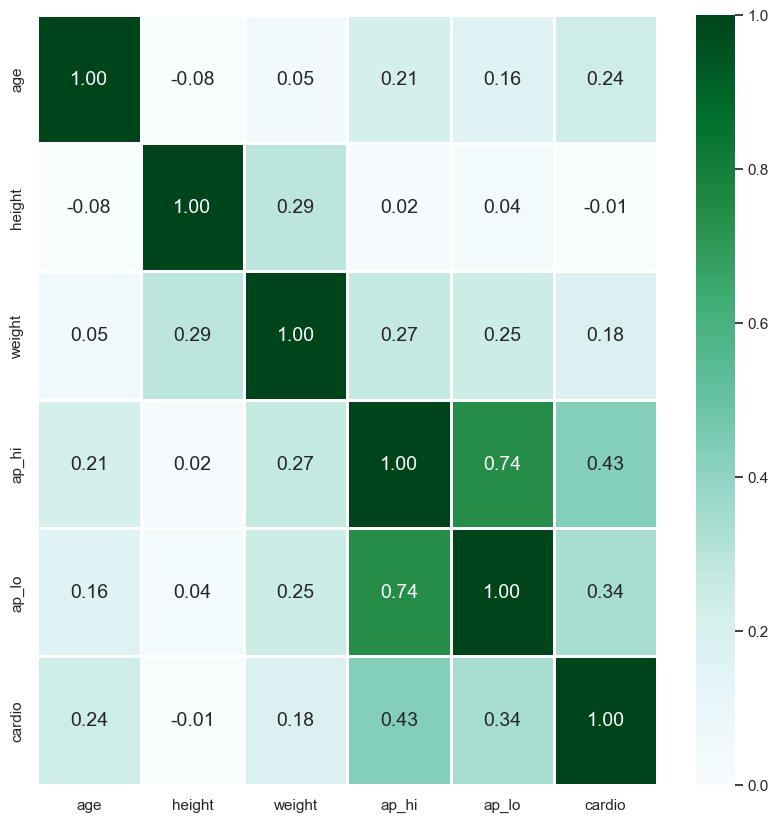
'height': -1.5270000000001573,

'weight': 29.728000000000065,

'ap\_hi': 120.832,

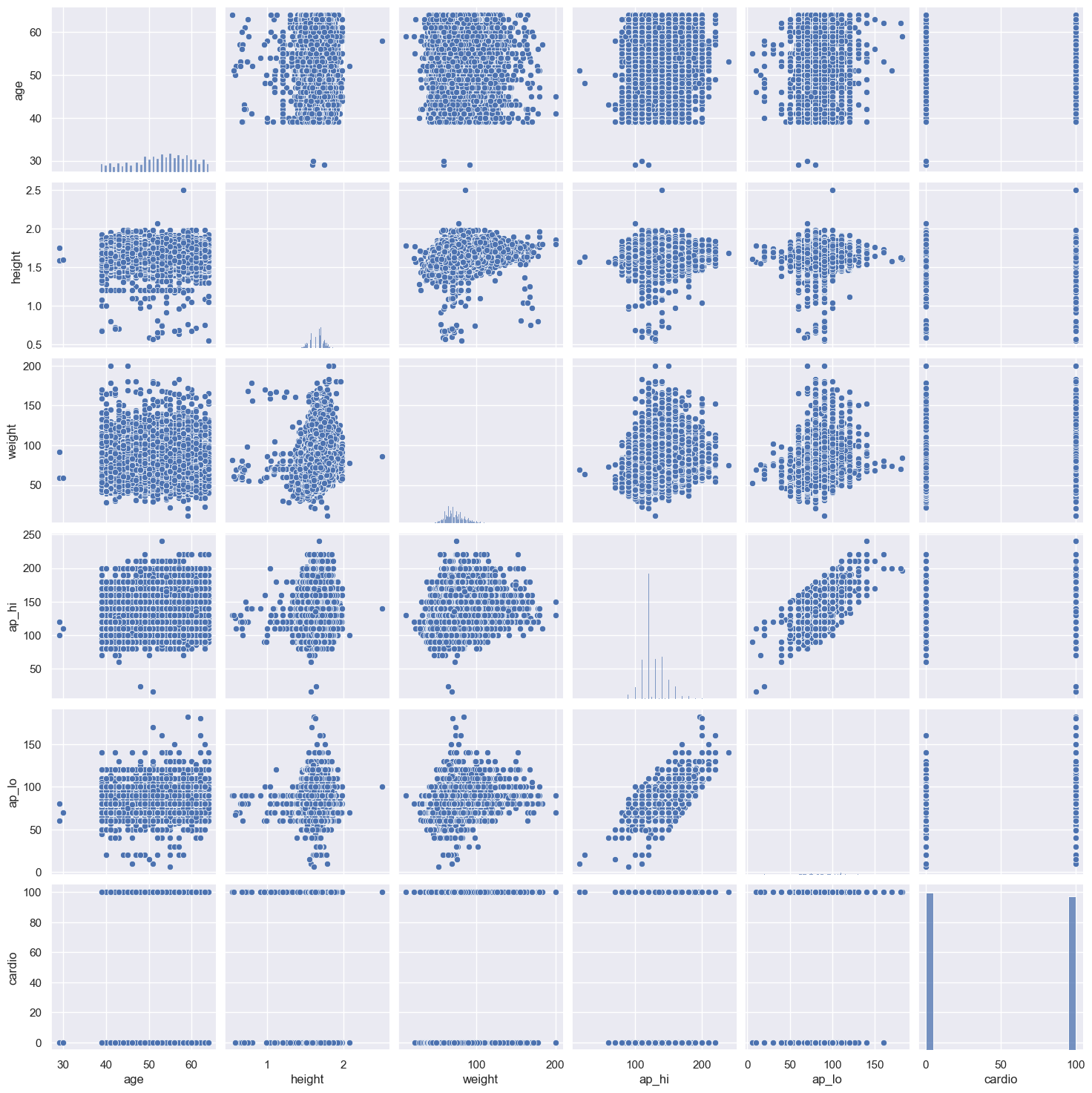
'ap\_lo': 51.82499999999993}

* + 1. And so, we can conclude ‘height’ is not playing a role.
    2. Upon request by group members, the following are plotted too, in relation to numeric variables.
    3. The correlation matrix is a no-brainer for numericals, but remember that ‘cardio’ is a binary ranged variable. However, it seems that pd.corr() knows to calculate it as a point-biserial. And so, we get this:



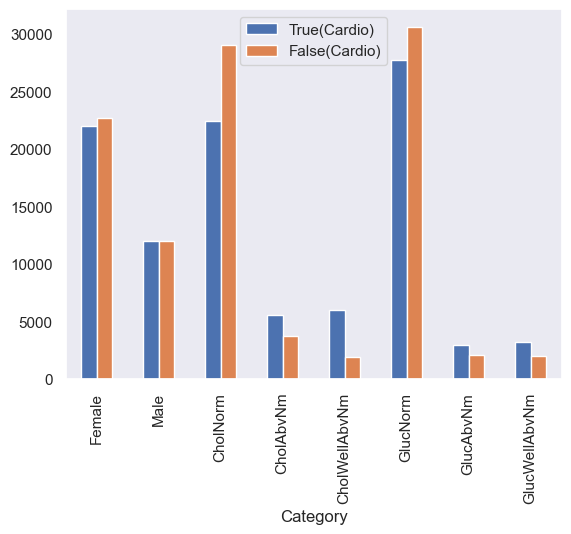
From here, we can see that height has a horrible correlation, justifying our above decision to throw height away. Weight is the next lowest, but to prevent throwing away too much information, I take those |x| >0.10 correlation into account.

* + 1. And translating cardio to [0, 100] integers to prevent Python from interpreting it as Booleans, we get the pairplots. But do not hold your breath.



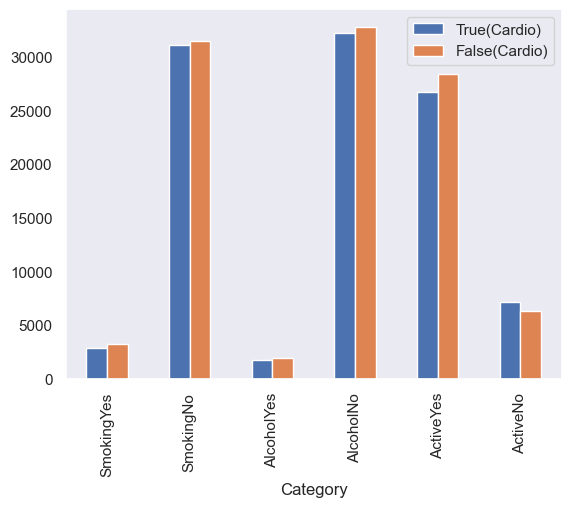
Why does the plots look so messed up?

1. [any variable] -> [cardio, which is a binary] will result in a 0, 100 possibility for cardio, and ‘cont’d’ for the other variables. So you will get two single lines.
2. But the ‘cont’d’ variables are not really continuous too. There are ranges where the values fall within, and ranges where it is very unlikely for values to fall within. Furthermore, there are variables with a range that map to the same exact value in the other variable, and so, you get a lot of regular shaped boundaries for the scatter plot.
   * 1. You can interpret it however you want, this is just my interpretation. It is very likely wrong as I suck at this sort of inference stuff.
   1. **Categoricals : [‘gender’, ‘cholesterol’, ‘gluc’, ‘smoke’, ‘alco’, ‘active’]**
      1. We see the relation between ‘gluc’, ‘cholesterol’ and ‘gender’ to the end state:



We get this. At one go, the gender data series seem to have small correlation by means of difference between male(true/false) and female(true/false). The rest seem to contribute with a higher chance to ‘cardio’. We keep that in mind first.

* + 1. We then see the binary variables relation(possible since I made my own function) to cardio.

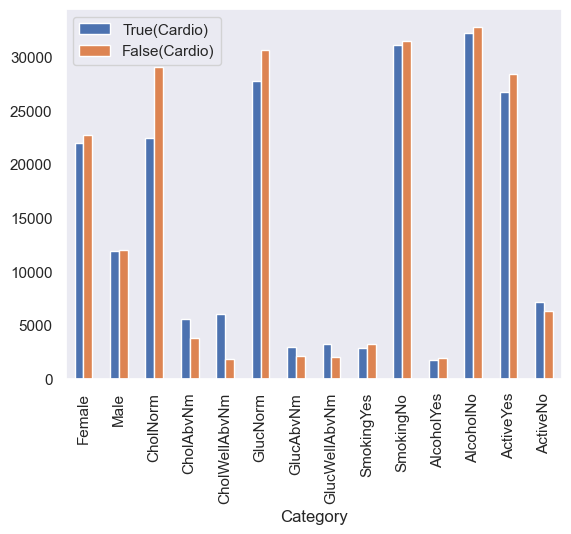


From here, we see two things:

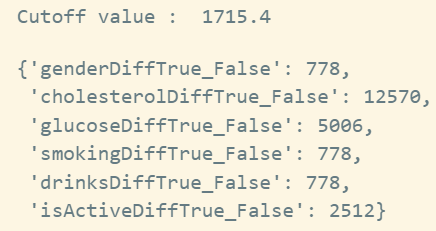
1. Smoking and Alcohol have low correlation.
2. Active may have a high correlation to ‘cardio’.

We keep that in view first.

* + 1. We see all of them at once. At the same time, we also get a new dataframe that contains the True and False separated values.



* + 1. And so, I run another function that spits out the gaps per variable, and we see an obvious few for elimination:



* + 1. The cutoff value is taken from 2.5% of total sample size.

In the end, we see the winners for both numerical and categorical variables to be considered:

1. Age
2. Weight
3. Systolic blood pressure(ap\_hi)
4. Diastolic blood pressure(ap\_lo)
5. Cholesterol
6. Glucose
7. Active binary
8. Formulating an issue
   1. We decide to focus on cardiovascular disease due to the following reasons: <Mainly interest reasons>
      1. That is what the dataset was for at the start.
      2. Smoke, alco and active seem to be quite skewed, and these are observed, not predicted.
      3. Investigating the correlation between the numerical variables seem useless, and even if it shows something interesting that is not what we are interested in.
      4. Cardiovascular disease prediction is the obvious outcome for this dataset, since it is the biggest effect in the dataset as it describes a serious illness(meaning next step is death or prevention).
9. Deciding on final approach to problem