BST 223 Data Analysis Project: EDA

Lucas Webster

Overview

In this Exploratory Data Analysis, I plan to prepare my data to be modeled in the future stages. This includes a few steps. First, I will examine missing value counts within my dataset to discover which covariates to keep, drop, or impute. Then I will examine the distribution of my covariates to better understand the data and identify outliers. I will then examine covariance and graphical comparisons of variables in order to grasp the relationship the covariates have with each other before diving into their relationship with the cap scores. Penultimately, I will consider variable transformations that seem appropriate in order to set up a better linear relationship in my model. Finally, I will propose a model building plan that I hope to follow for the remainder of the project.

A NOTE This EDA is 35 pages, FAR LONGER than what I plan to include as EDA in my final report. obviously. I will cherry pick the most interesting and necessary results from this for my final report.

Examining Missing Values

id	0
med_cap	0
iqr_cap	12
alc_ever	732
alc_times_yr	1242
alc_binge_times_yr	3639
cig_smoker_ever	9
sleep_weekday	60
sleep_weekend	62
no_ins_12_mon	525
insulin	2908
bmi	40
sex	0

```
age                       0
race                      0
born_us                   0
edu_lvl                 261
preg                      0
fam_inc_to_pov          763
age_of_diabetes        5137
freq_mod_ex              33
unit_mod_ex            1147
avg_mins_mod_ex        1160
freq_vig_ex              32
unit_vig_ex            3127
avg_mins_vig_ex        3136
mins_seden_daily         43
hep_b                    25
kcal                   1736
carb                   1736
sugar                  1736
fiber                  1736
fat                    1736
vit_e                  1736
chol                   1736
water                  1736
salt                   1736
caff                   1736
mod_ex_per_week          33
vig_ex_per_week          32
cap_cat                   0
diab                     11
ins                      13
mod_ex_mins_per_week   1160
vig_ex_mins_per_week   3137
dtype: int64
```

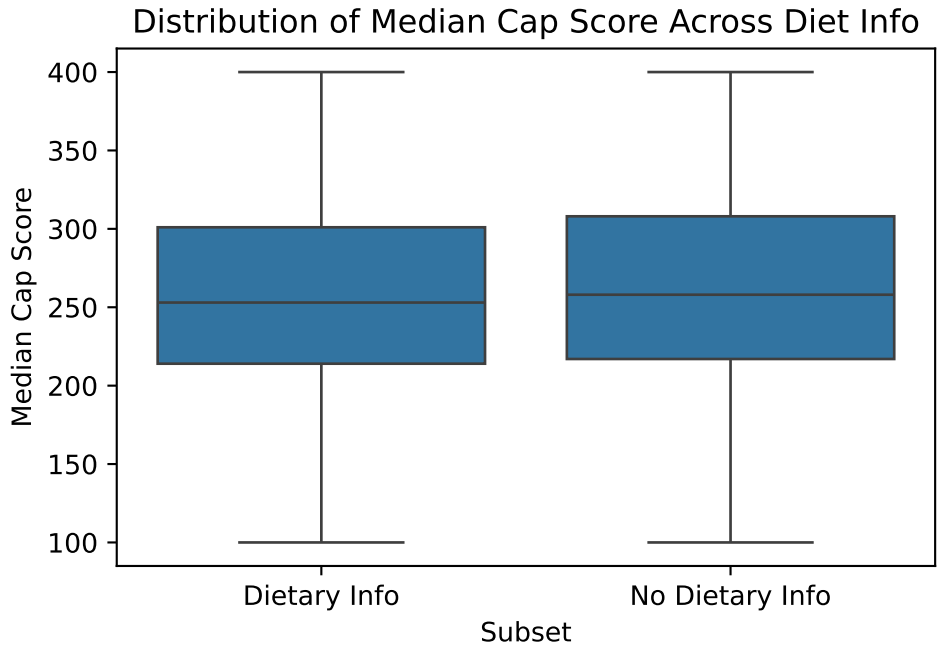
Observe that insulin, alc_binge_times_yr, insulin, age_of_diabetes, and multiple vigorous exercise variables clearly have too many missing values, around half, to impute or model properly, hence we remove them. We also remove many of the exercise variables, as those marked frequency are useless without the unit, and the unit is useless without calculations regarding frequency. The same goes for average minutes, useless without the unit. Hence frequency, unit, and average were removed, but their 'information' retained in 'mod_ex_per_week' and 'mod_ex_mins_per_week'. We also remove the IQR as using a technique like inverse probability weighting to factor in its effect does not apply to what will wind up being a categorical outcome, and is beyond the time and scope of this project.

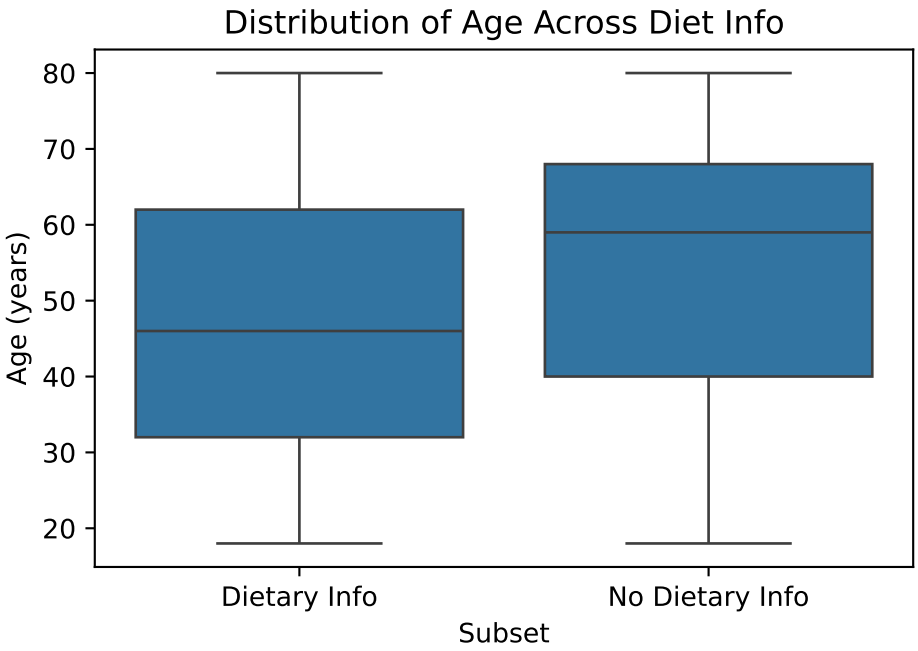
As an aside, those with pregnancy missing were marked as not pregnant, as I feel it is logical that this data would be present but ignored. It also doesn't apply to roughly half of the data (males) so they could autmatically be marked as not pregnant (0).

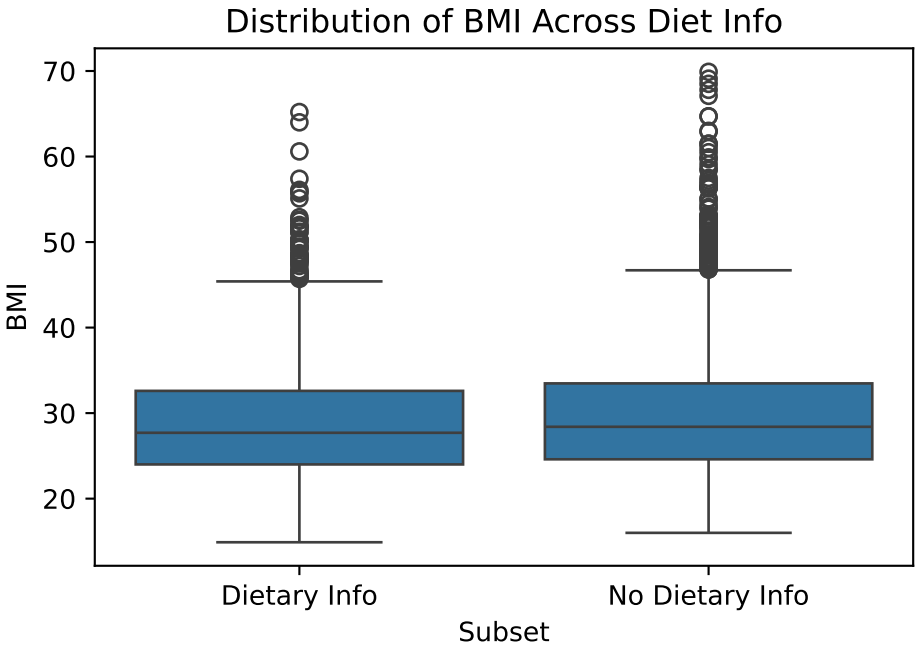
Furthermore, we will drop sleep_weekend from the data. This is because weekend length differs for everyone depending on employment, and hence weekday (which in the survey meant weekday/workday) gives a better view of an individuals average sleep habits.

We also observe the diet data to have a high and equal amount of missing values across the diet data. This is because if an individual is missing one diet entry, they are missing all their diet entries. (This was verifified and can be checked in code appendix) However, removing diet data entirely is likely to severely hurt model accuracy, as diet and liver health and fat production have a strong relationship. Hence, we will briefly analyze the the demographics of the individuals with diet data against those without to see if dropping these individuals from our model would change our population of analysis.

Dietary Information: A Complete Case Analysis?







subset race	Dietary Info	No Dietary Info
Mexican American	0.091590	0.064781

subset race	Dietary Info	No Dietary Info
Other Hispanic	0.114631	0.094513
White	0.493088	0.620981
Black	0.139401	0.112159
Asian	0.087558	0.044960
Other	0.073733	0.062606

subset sex	Dietary Info	No Dietary Info
Female	0.506912	0.55765
Male	0.493088	0.44235

subset diab	Dietary Info	No Dietary Info
No Diabetes	0.773988	0.700387
Prediabetes / Borderline	0.110405	0.166747
Diabetes	0.115607	0.132865

subset hep_b	Dietary Info	No Dietary Info
No	0.991893	0.983499
Yes	0.008107	0.016501

We see from the above boxplots and tables of demographic and baseline health covariates that population of individuals who had their diet recorded does not differ much from those who did not, meaning a complete case analysis of those with complete diet covariates is justified. As a side note, the only twp covariates analyzed here with possible more than a minimal difference are race and age. However, the racial distribution works to our advantage, those with diet data are more diverse than those without. Furthermore, the age distribution of those with diet data is slightly lower than those with it, and as mentioned in my proposal, I plan to examine younger participants more closely, so I don't view this as a concern.

Because we are dropping these 1736 participants missing diet information to perform a complete case analysis on those with dietary information, we will perform no imputation, so as not to give preference to any variable 'based on its missingness.' Hence, in our future model, some

values may just be dropped, but the total will not make up a large proportion of our model (over 20%).

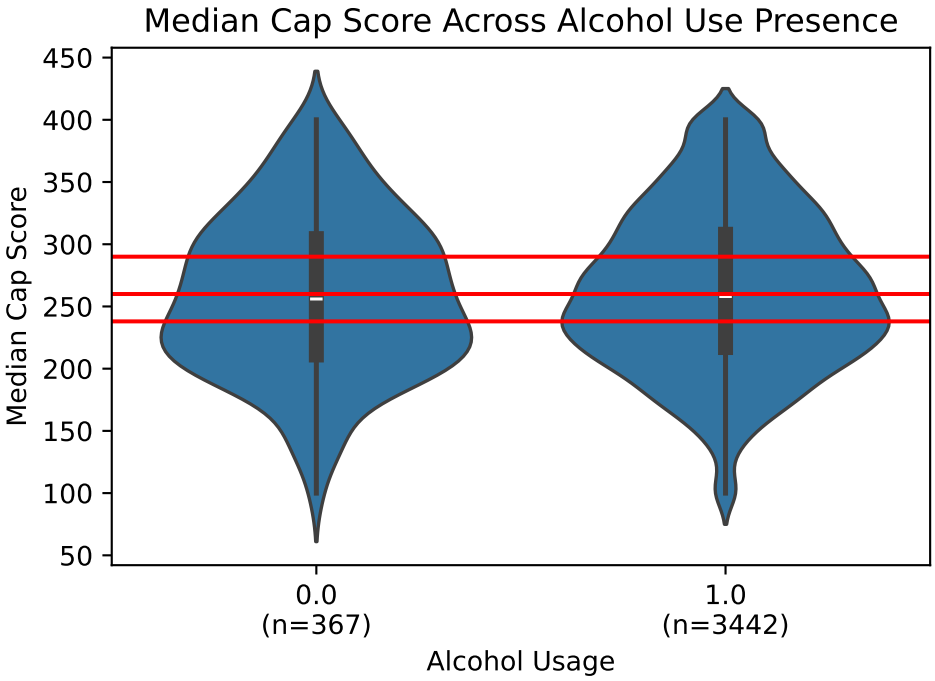
As an aside, this removes the few pregnant individuals, hence we drop the covariate.

```
id                       0
med_cap                  0
alc_ever               328
alc_times_yr           696
cig_smoker_ever          6
sleep_weekday           28
no_ins_12_mon          297
bmi                     19
sex                      0
age                      0
race                     0
born_us                  0
edu_lvl                144
fam_inc_to_pov         460
mins_seden_daily        22
hep_b                   16
kcal                     0
carb                     0
sugar                    0
fiber                    0
fat                      0
vit_e                    0
chol                     0
water                    0
salt                     0
caff                     0
mod_ex_per_week         15
cap_cat                  0
diab                     5
ins                      6
mod_ex_mins_per_week   772
dtype: int64
```

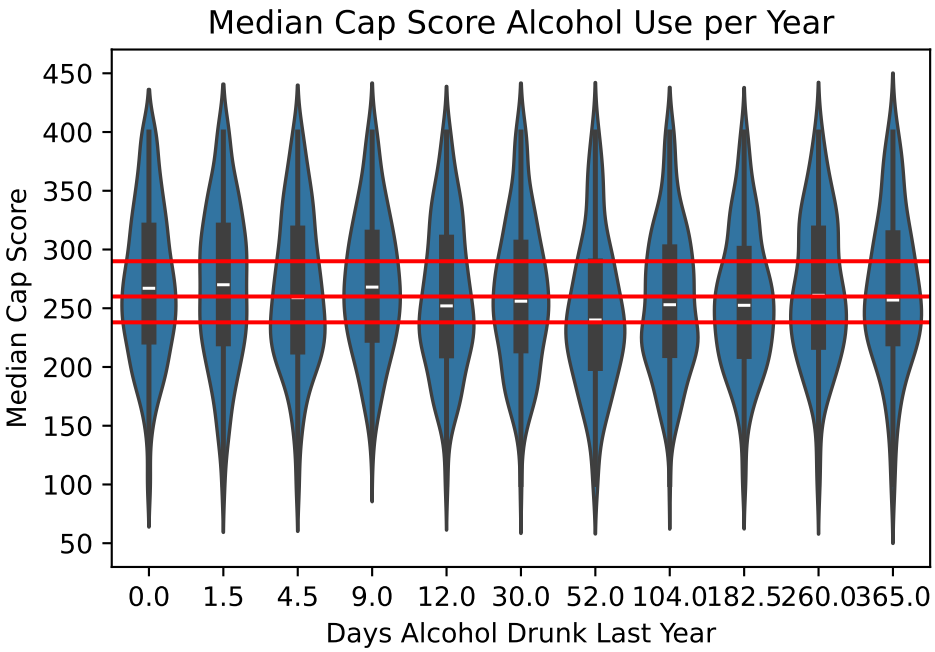
Observe that dropping those individuals who didn't complete the diet data dropped NA's in other column at a faster rate, meaning those who diet data wasn't obtained for were more likely to have less data in general, dropping them again seems like a justified choice. Our dataset now contains 4137 observations.

Data Visualization (Discrete Covariates)

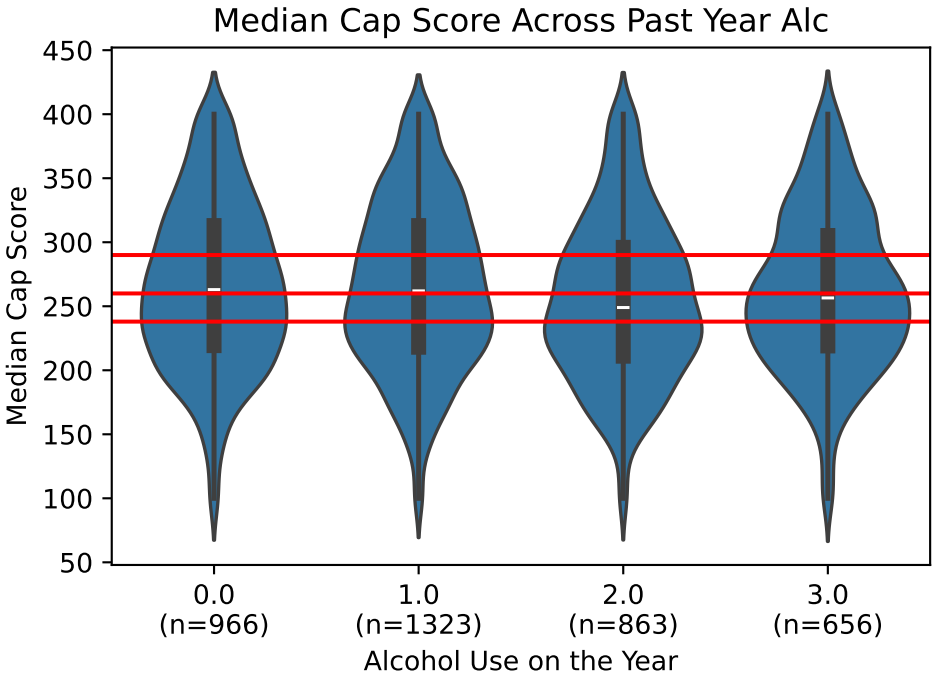
Below are violin plots to help visualize each covariates relationship with median CAP score, with red lines drawn across to seperate the fatty liver disease catgeories based on CAP score which we will be classifying with our multinomial model.



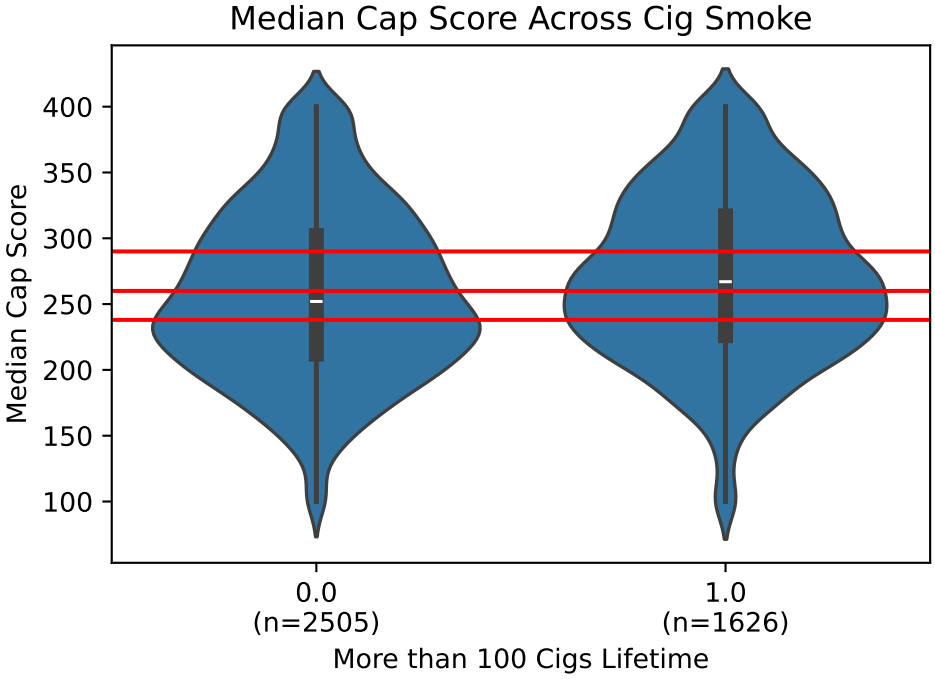
We see similar cap scores, possibly slightly lower, across the indicator of whether the individual has drank alcohol ever or not.



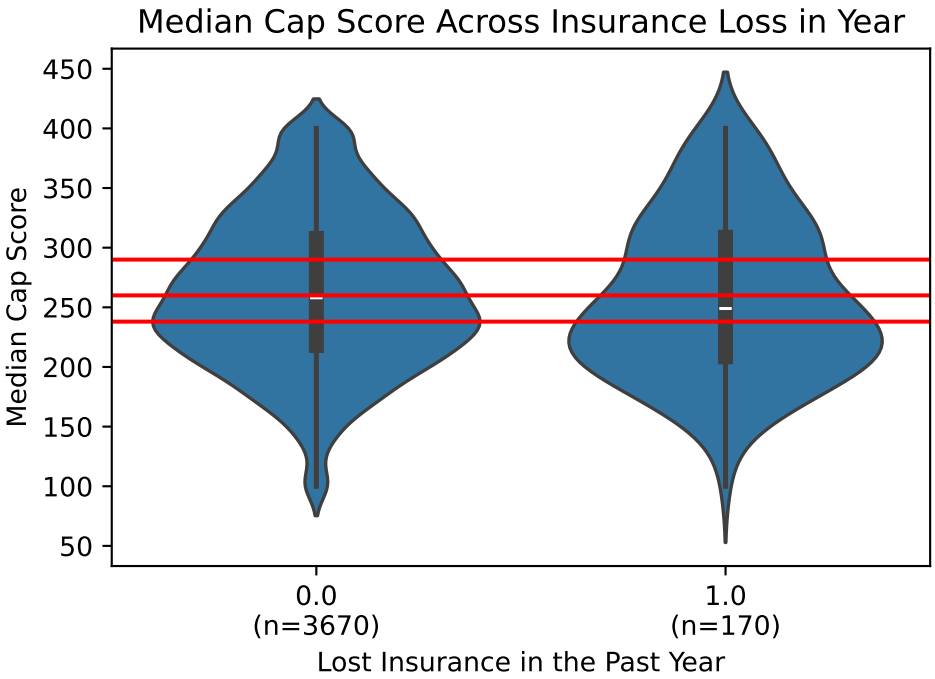
We also need to investigate what do with alc_times_yr variable. Its a categorical variable which represents the amount of days someone has drank in the last year. This was created using information on how many drinks someone had a week on average in the last year. The problem is, there are 11 categories, so we need to find a way to truncate the data, as 11 dummy variables, especially with some at low samples will reduce power. I will recategorize to 0 never, 1 at most monthly, 2 more than monthly to weekly, 3 every other day to daily for a total of four levels. We will also make NAs 0 if the respondent indicated they have never drank in the alc_ever column. We will then revisualize.



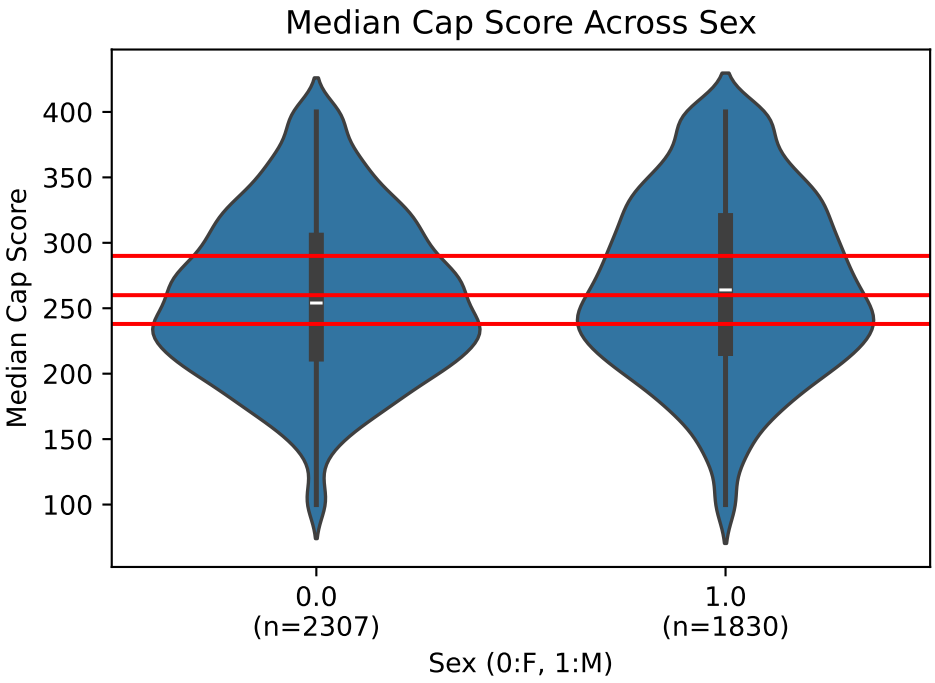
We don't see any obvious pattern in drinking habits and median cap score.



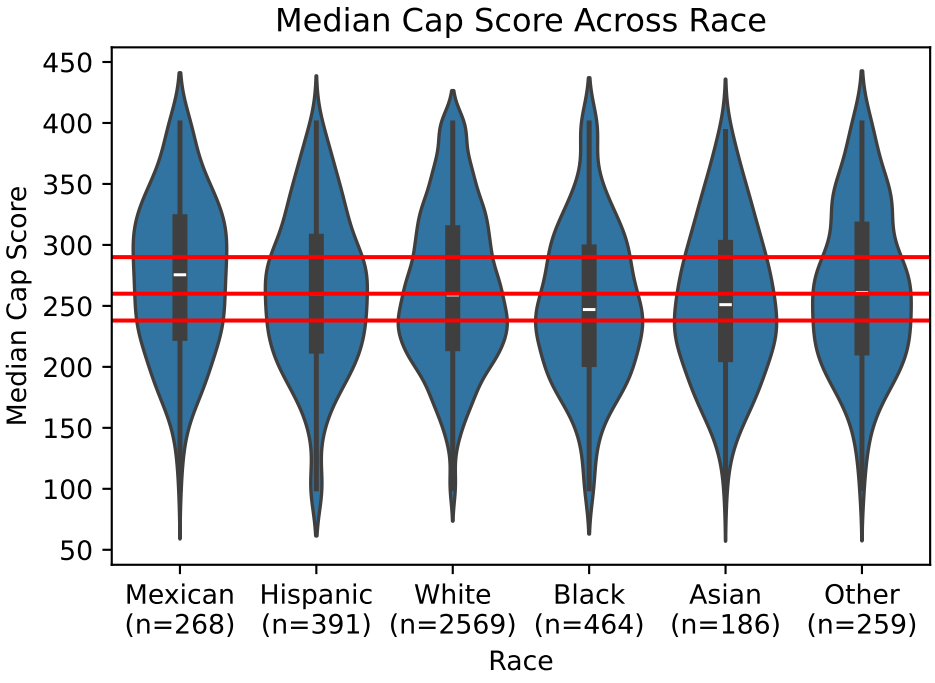
We again see similar trends across those who have smoked less than and more than 100 cigarettes in their lifetime, with less than being slightly lower.



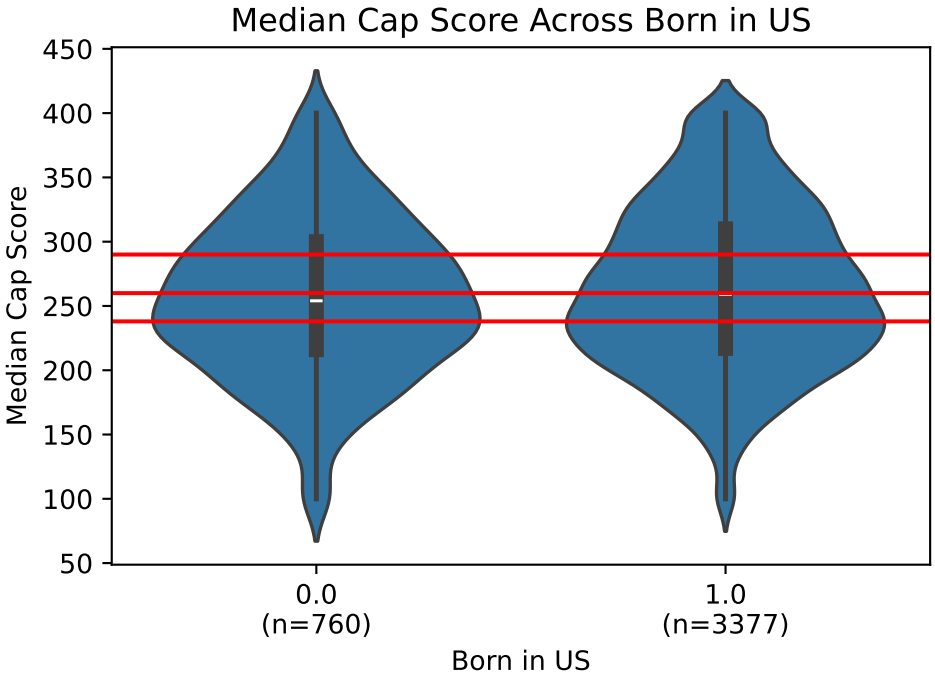
Here we observe oddly that those who have lost insurance in the past 12 months have a lower median cap score, but I attribute this to sample size.



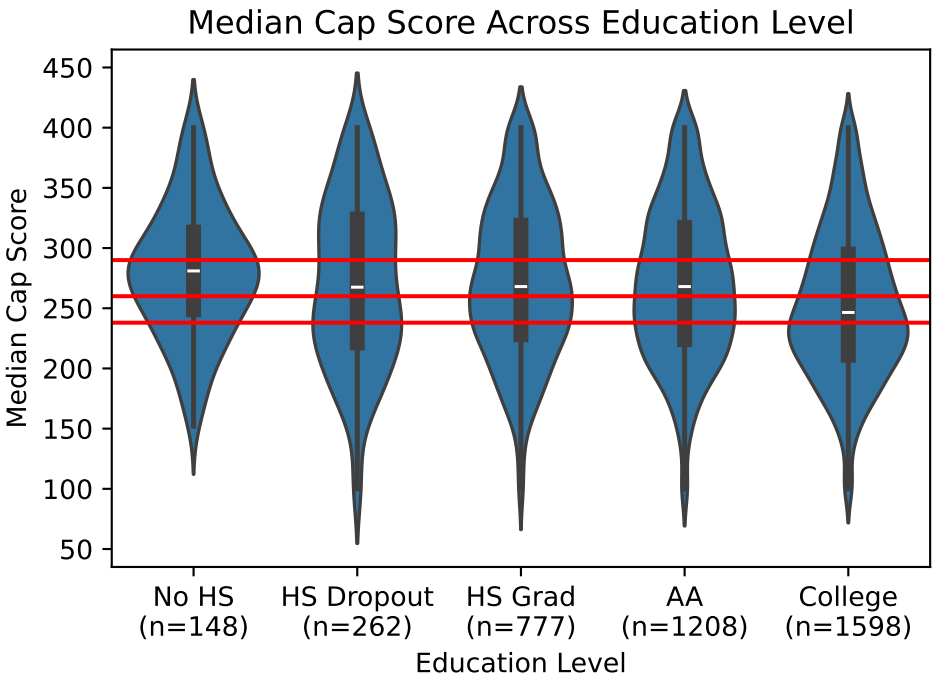
Females show an obviously lower concentration in CAP scores.



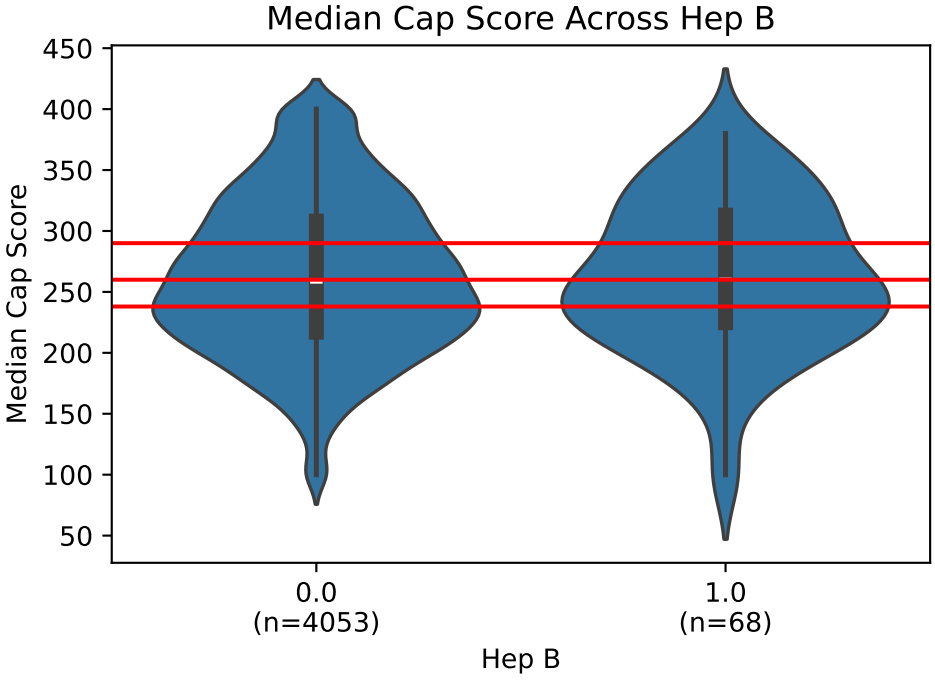
Here we see Mexican Americans to have higher CAP score distribution, while Asians have a lower CAP score distribution



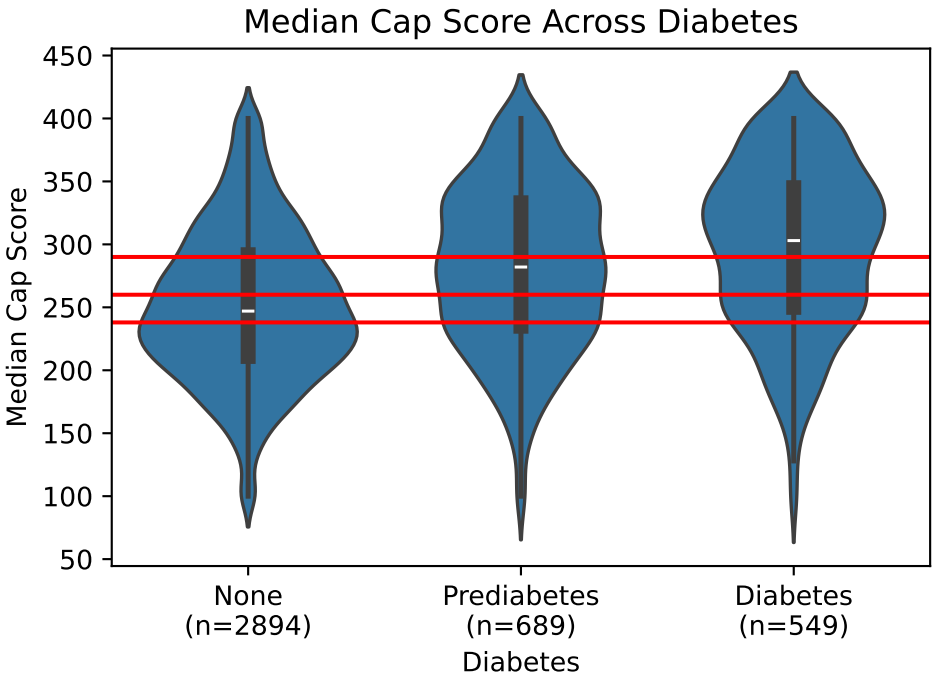
Those born outside of the US seem to have a lower median CAP Score.



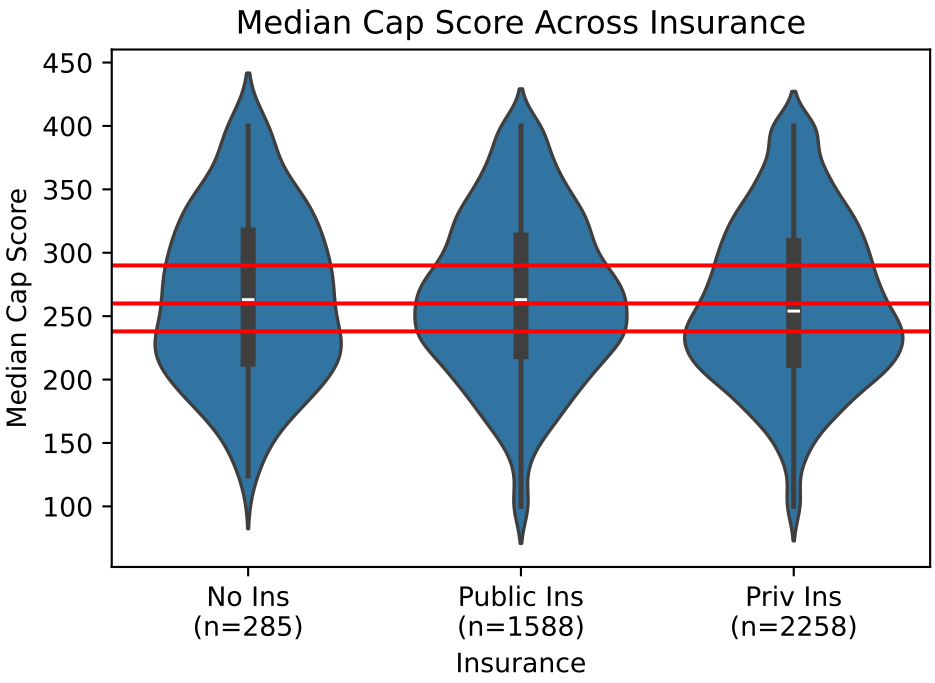
Those with college degrees tend to have lower median CAP scores it seems.



Interstingly, the Hepatitis B distributions appear the same. This may be because those with the disease have advanced warning about Fatty Liver Disease and more doctor monitoring.



Those with diabetes have significantly higher CAP score. This comes from their inability to regulate insulin I suspect. We would have had more insight into this if we were able to keep the insulin variable, but it was missing too many values as discussed.

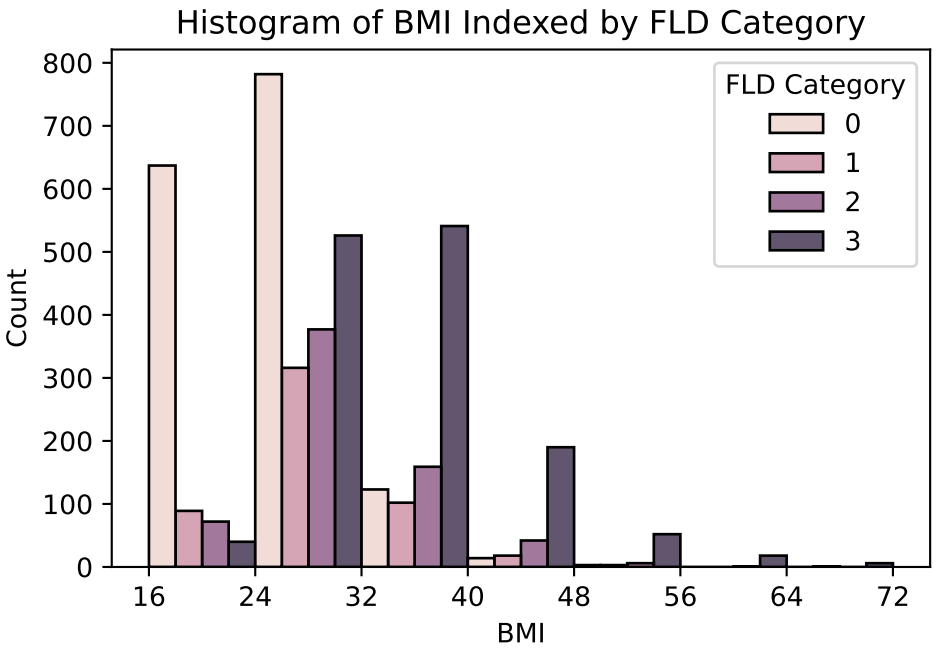


We see those with private insurance have a lower cap score distribution, and I expect this to be correlated with income.

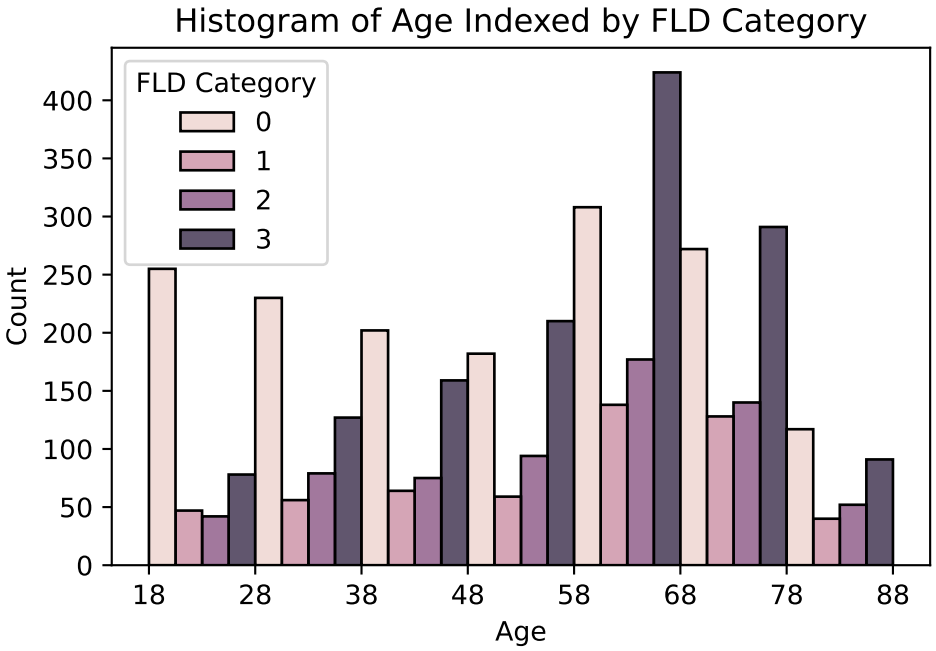
All in all, females with private insurance, no diabetes, non-Mexican race born outside the US seem to likely have the lowest CAP score and hence fatty liver disease classification.

Data Visualization (Continous Covariates)

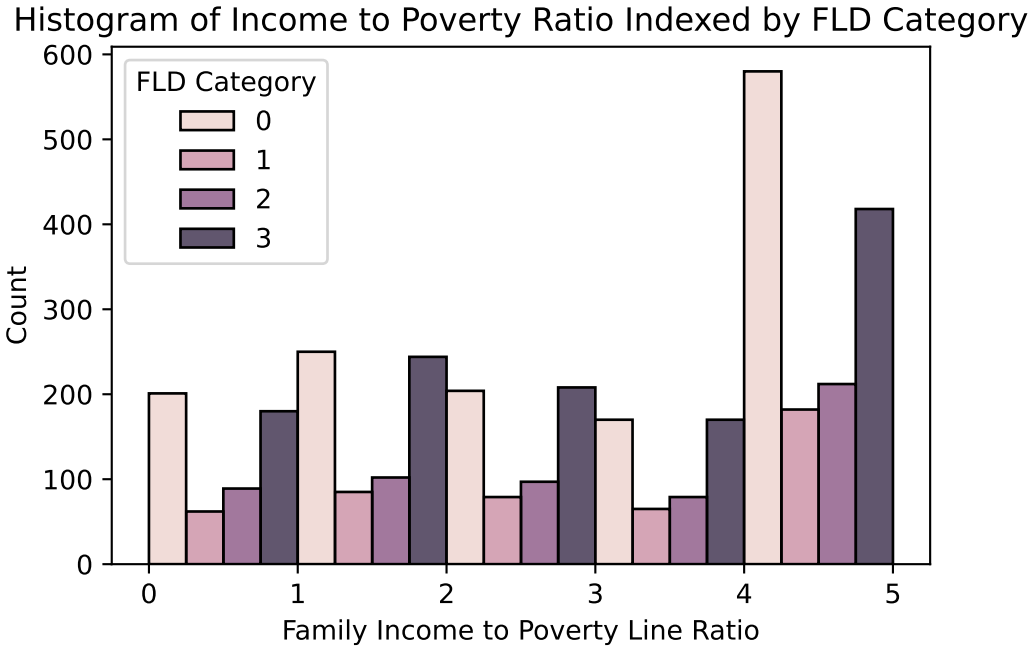
The histograms below are created as follows. Each hisogram bin is split up into four categories, one for each Fatty Liver Disease Category. Hence we can see a trend in total distribution of the continous covariate, and how FLD categories change with the covariate. The lighter color indicates a lower catgeory, and the darker categories indicate higher categories. Each bar between numbers on the x-axis are part of the same age category. I produces these graphs as an economical way to convey the same information as a histogram and a side by side boxplot on CAP category together.



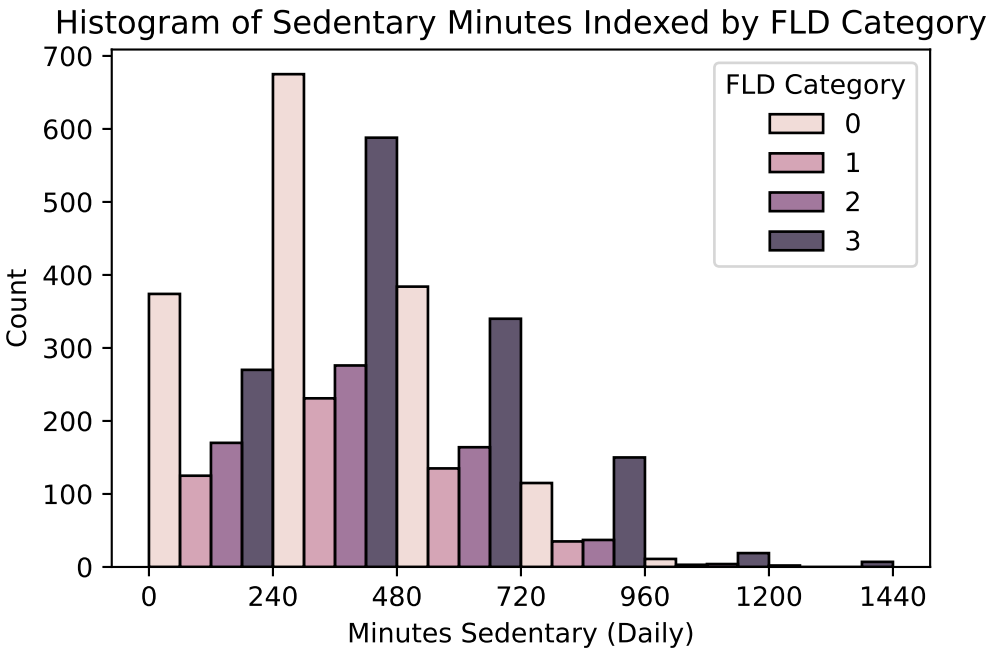
We see above that most individuals are concentrated with a BMI below 30. We also see a trend in FLD though, as BMI goes up, the proportion of people in higher FLD categories also increases. BMI will likely have a positive effect on FLD classification.



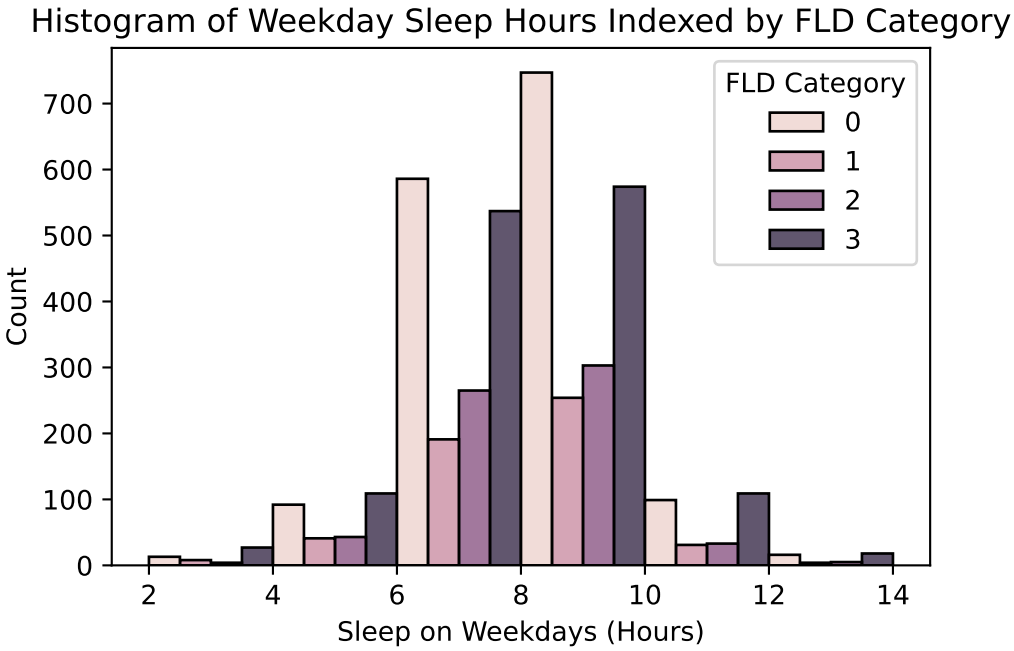
From this age histogram, we see most of our participanst are older, with 50-80 year olds making up a majority of the sample. We also see that at young age bins, the proportion of people with high FLD categorization is low, and it increases with age.



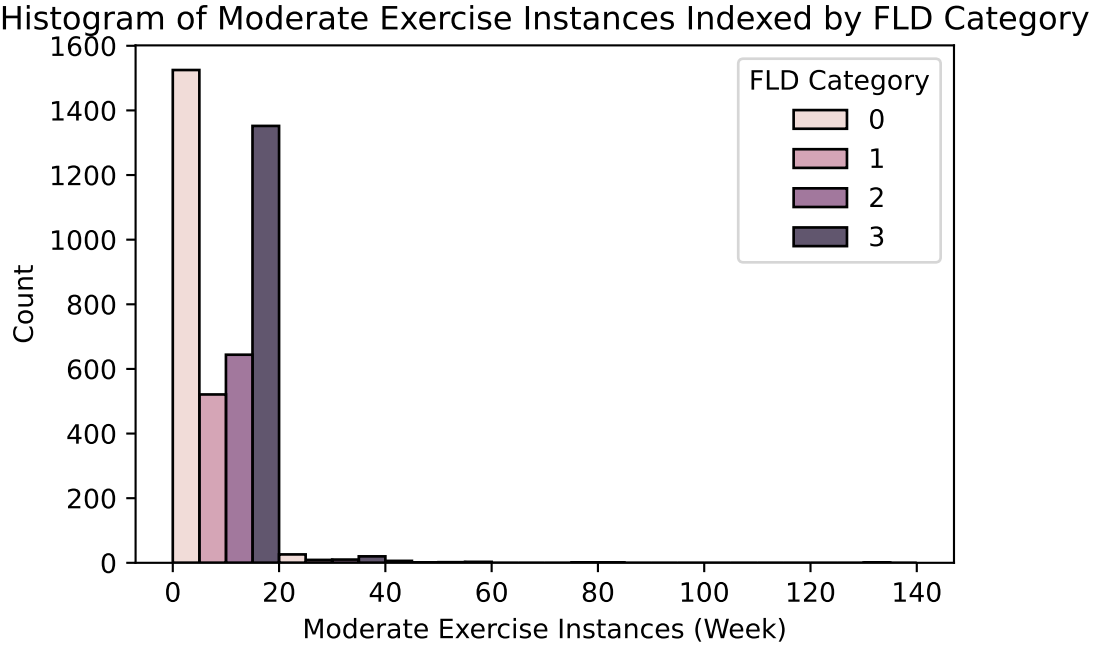
The above x-axis is a respondent's family household income ratio to the poverty line, with any value 5 or greater being rounded down to 5. We see similar FLD rates across categories, with most respondents coming from the 4-5 range, indicating most respondents were financially stable.



We see most respondents to spend less than 8 hours sedentary daily. Those with extreme amounts of time sedentary seem to have higher rates of FLD than those with lower time sedentary. This variable is ripe for standardization we notice as well with a large mean and large spread.



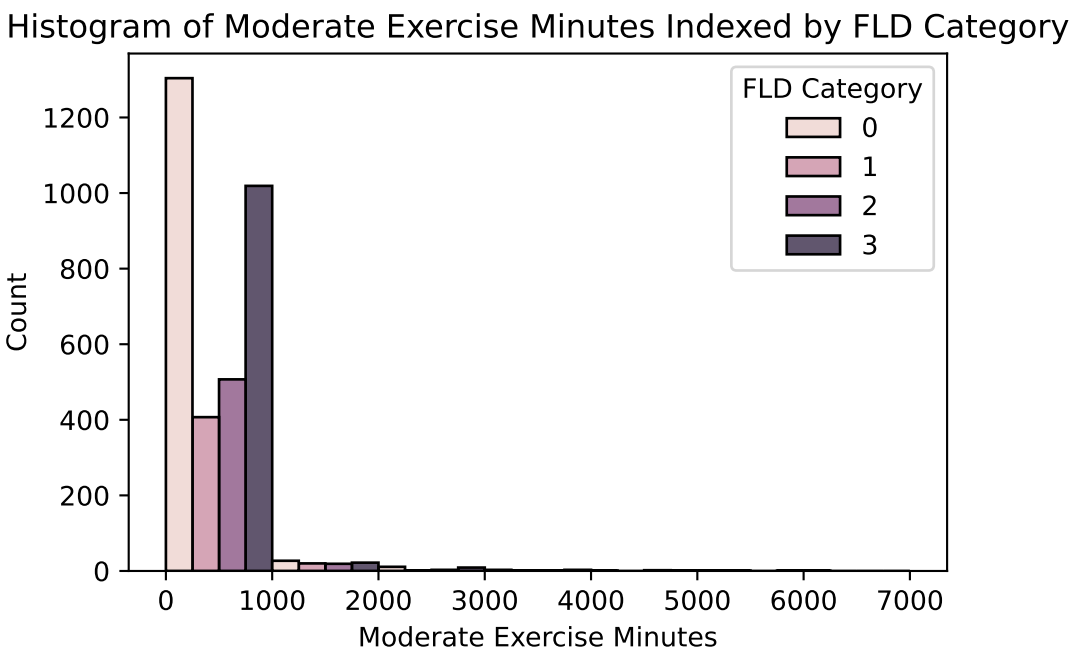
Most individuals get between 6 to 10 hours of sleep, and interestingly, those with 4 to 6 and 10 to 12 have higher rates of FLD. Possibly this extra sleep is an indicator of sedentary lifestyle. This distribution also seems to follow a normal distribution centered at 8.



Observe here that our histogram is skewed by a few outlieing observations above 60. This means they are averaging over 10 instances of exercise per day, which seems unreasonable. Lets investigate this.

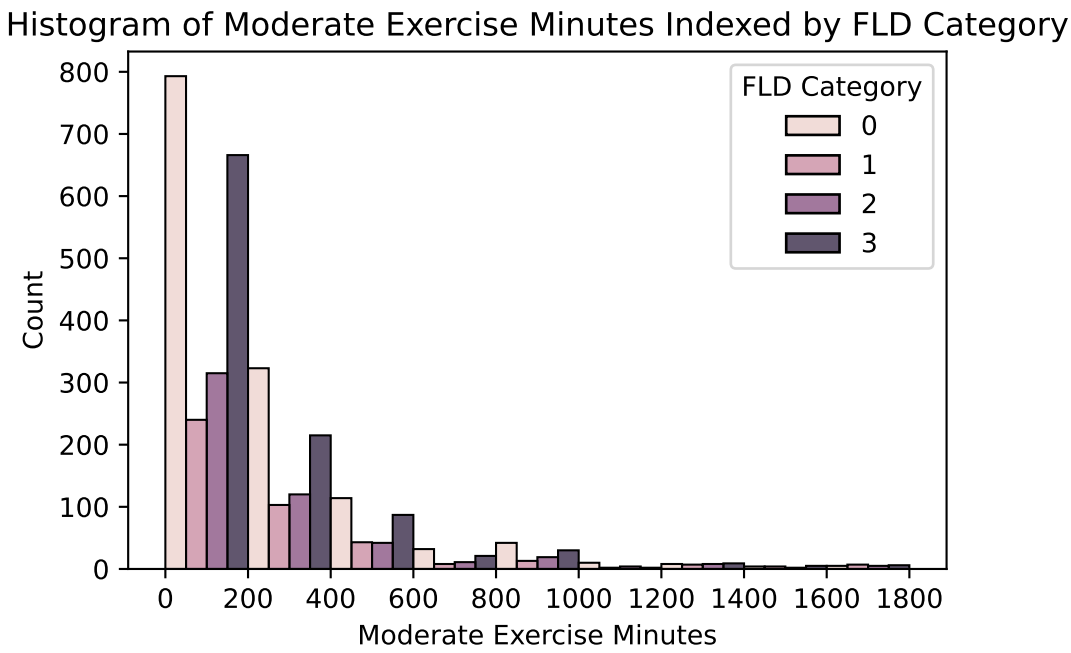
	id	med_cap	alc_ever	alc_times_yr	cig_smoker_ever	sleep_weekday	no_ins_12_mor
2473	135456.0	203.0	1.0	0.0	0.0	9.0	0.0
3284	137082.0	268.0	1.0	4.5	0.0	3.5	0.0
4841	140221.0	296.0	1.0	104.0	0.0	7.5	0.0

There doesn't seem to be anything noticeable about patient 135456, especially considering there minutes of activity per week, its only 168, meaning they average 2 mins per instance. This leads me ot believe that mod_ex_mins_per_week may be a better measure of a person's moderate exercise. So we proceed to investigate this graph.



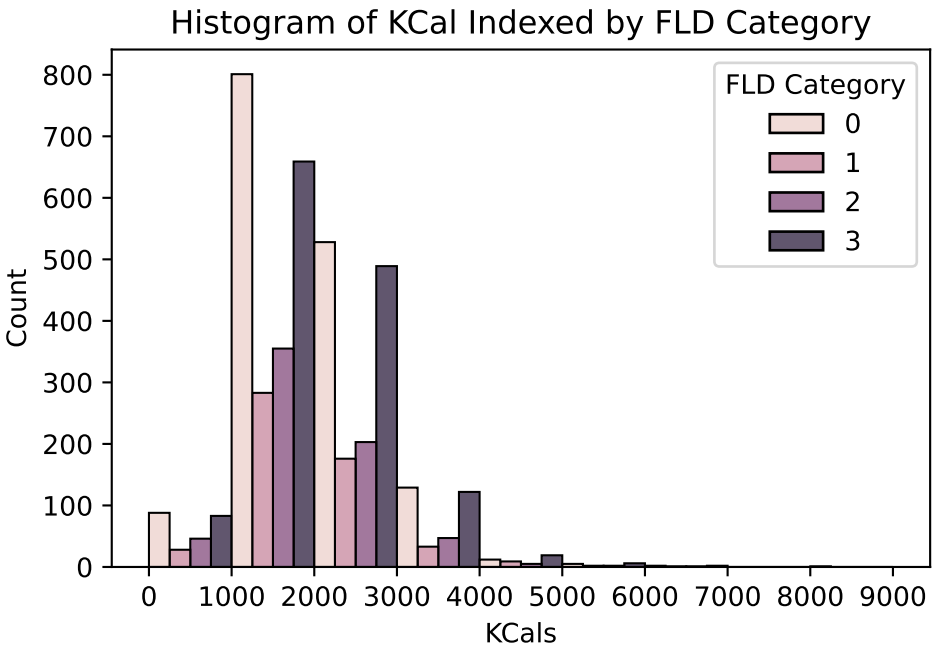
Again, our data is skewed, this time by different individuals who are averaging over 4000 minutes of exercise a week. This is over nine hours of moderate exercise a day on average. While this figure seems unlikely, it comes out to be equivalent to a full workday in a physical field, like construction, so it may not be out of the ordinary.

We also see individuals with over 5000 minutes of moderate activity per day. This figure seems highly unreasonable, about 12 hours a day on average, especailly considering that vigorous activity was recorded, hence professional athletes should be out of the picture for this measure. Hence I choose to remove individuals with over 5000 minutes of weekly moderate activity per day, on the guise of misreporting or misentry. We will then reconstruct our histogram, removing values with over 2000 minutes in order to get a better visual of most of the data

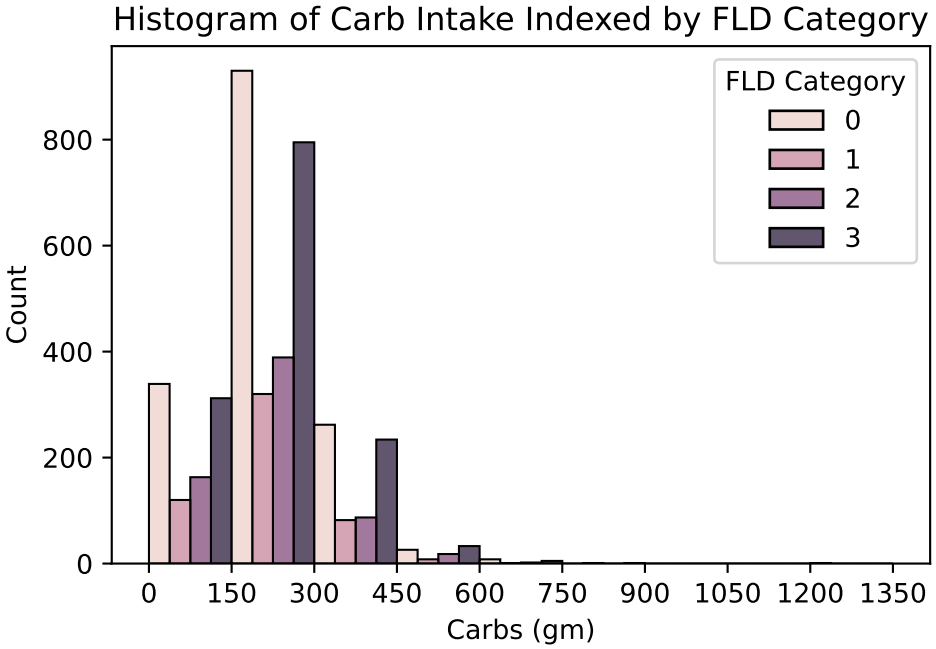


Now we see that for those with under 2000 minutes of of moderate exercise a week, most individuals exercise for about 200 minutes or less. There seems to be no change in FLD pattern across bins. Exercise also appears to take on an exponential distribution. From this investigation, I also remove mod_ex_per_week from the choice of covariates to stick with mod_ex_mins_per_week. This varaible also is likely to need standardization for assist with future model convergence.

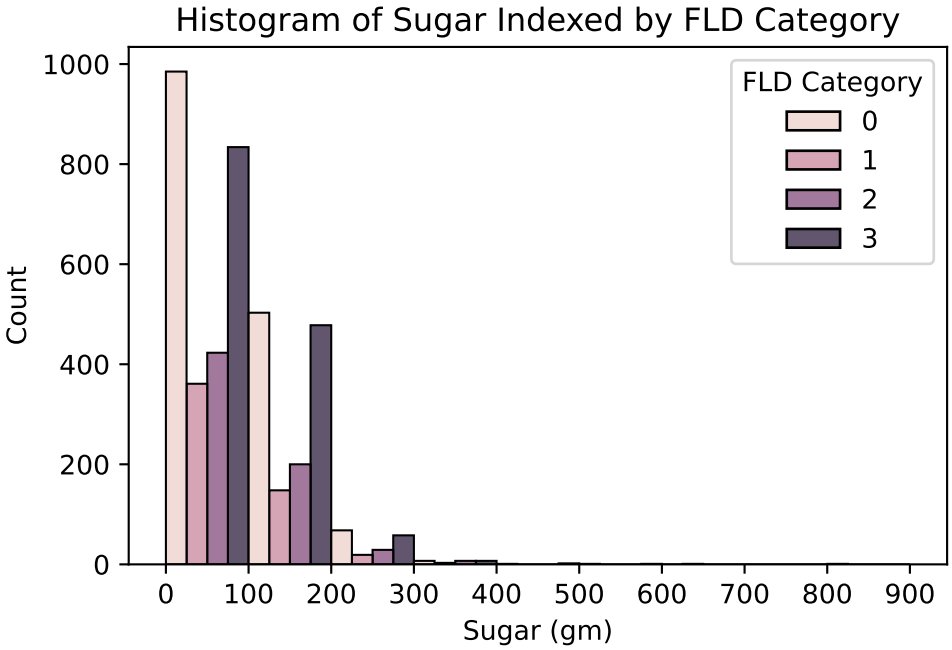
Now we examine diet covariates.



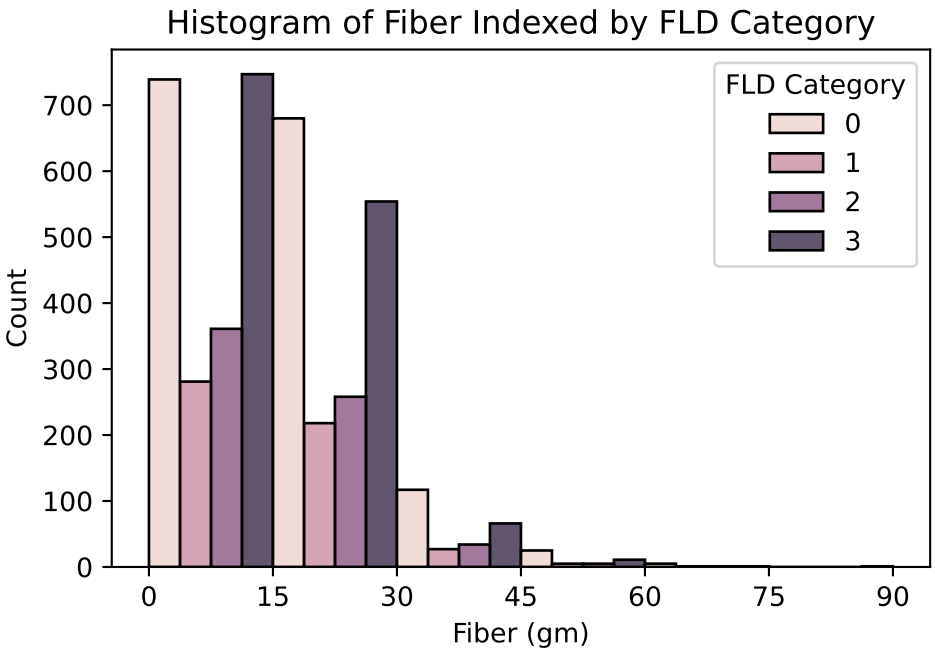
We examine most individuals eat between 1000 and 3000 calories. Anything above that and we notice the propotion of FLD outpaces that of those without. This variable will need standardization.



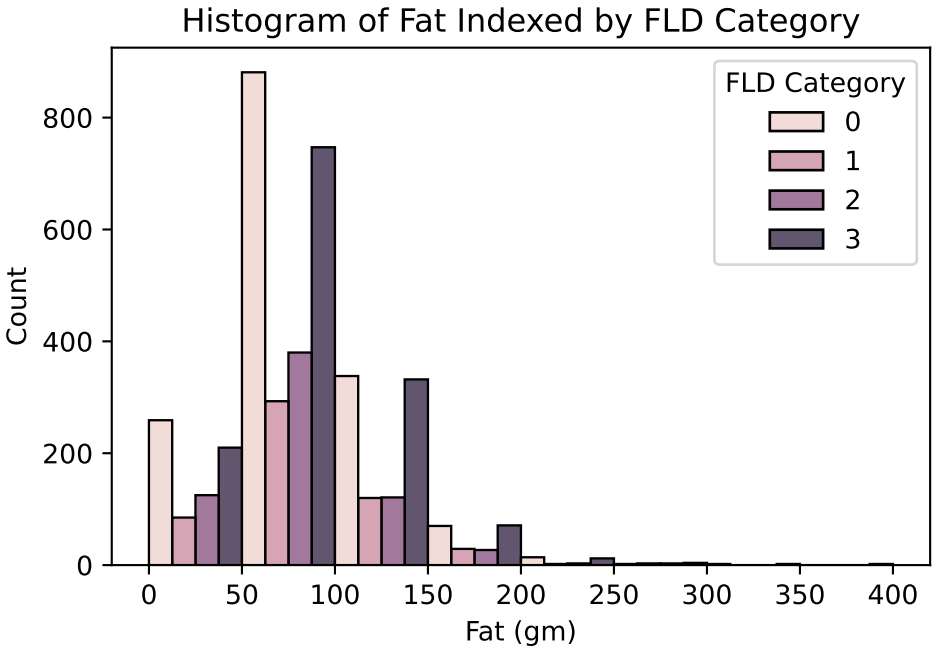
Most individuals eat under 300 grams of carbs per day, and there seems to be no FLD correlation here. This variable will need standardization.



Most people consumed less than 200 grams of sugar per day, well over the recommended amount. Those consuming over 100 grams have higher instances of FLD. This variable will need standardization.

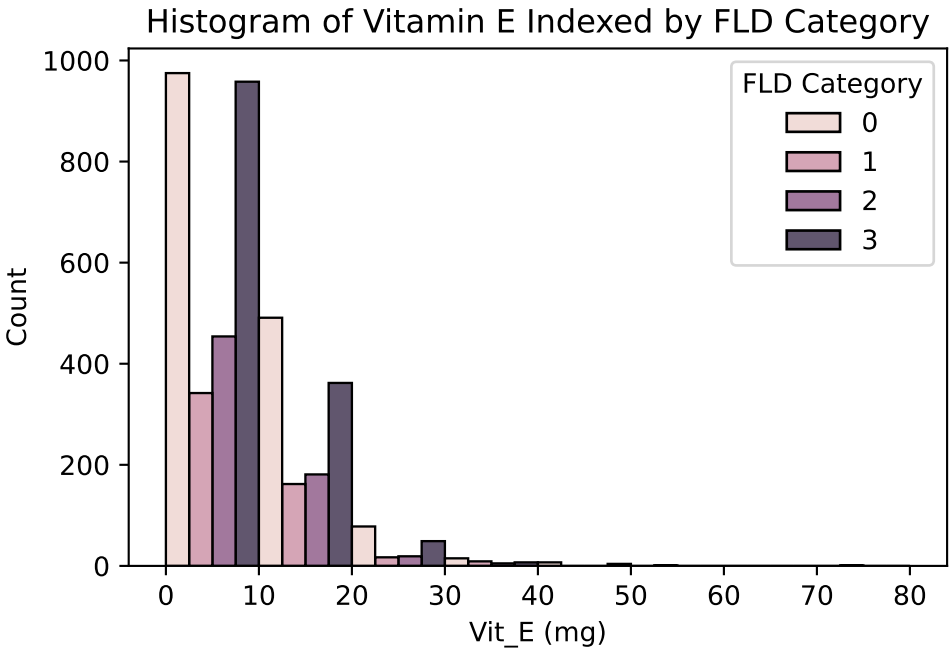


Most individuals consume less than 30 grams of fiber per day, with a stark dropoff after that number. Those who consume less than 15 grams have much higher instance of FLD.

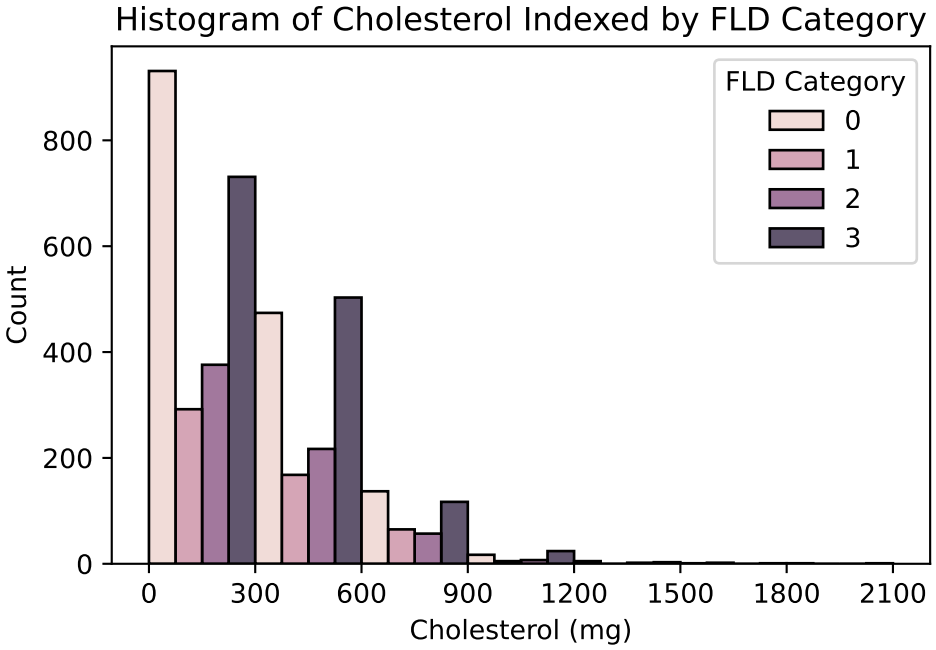


Most individuals consume between 50 to 100 grams of fat per day, with higher instances of

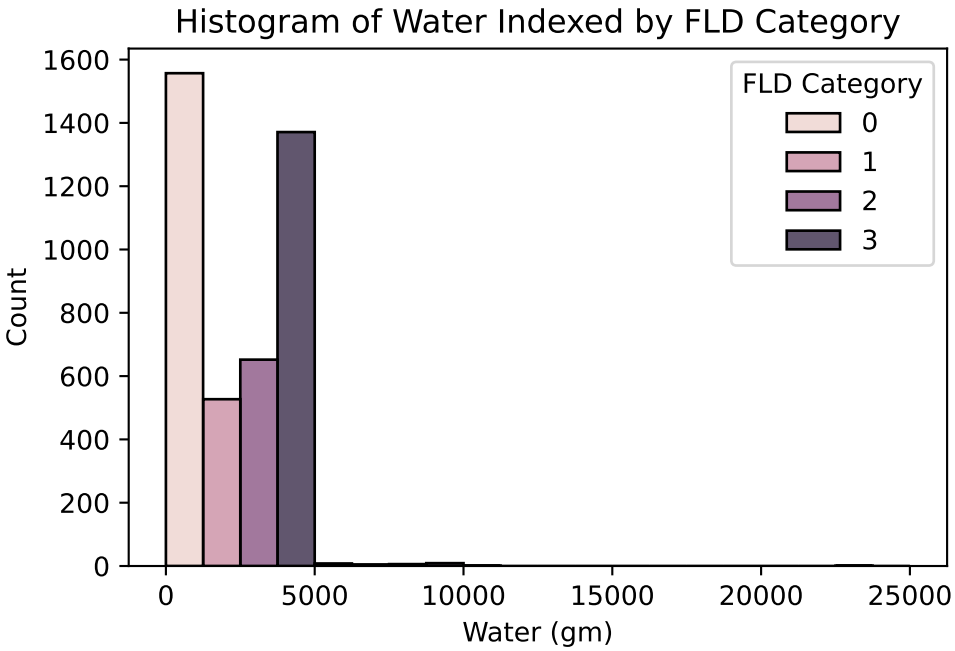
FLD coming over the 100 gram mark. This variable will need standarization.



Most people consume between 10-20 mg of Vitamin E, with a sharp dropoff afterwards. Those with 10-20 mg seem to have lower instances of fatty liver disease than those with 0-10 mg.



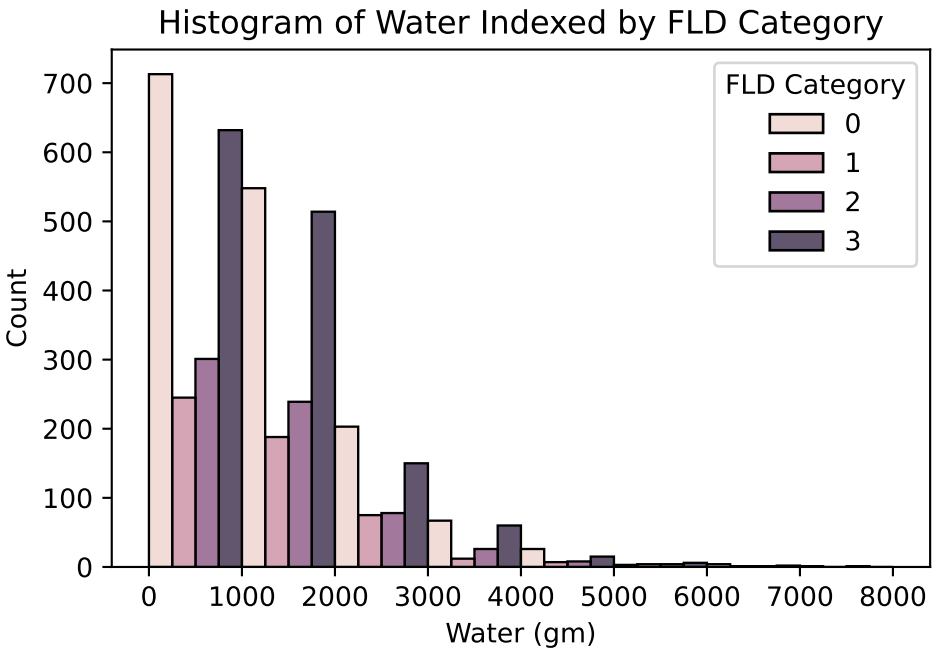
Most people consume 0-300mg of cholesterol, with FLD precense increasing after this cutoff. This data also seems to be exponentially distributed. This variable will need standardization.



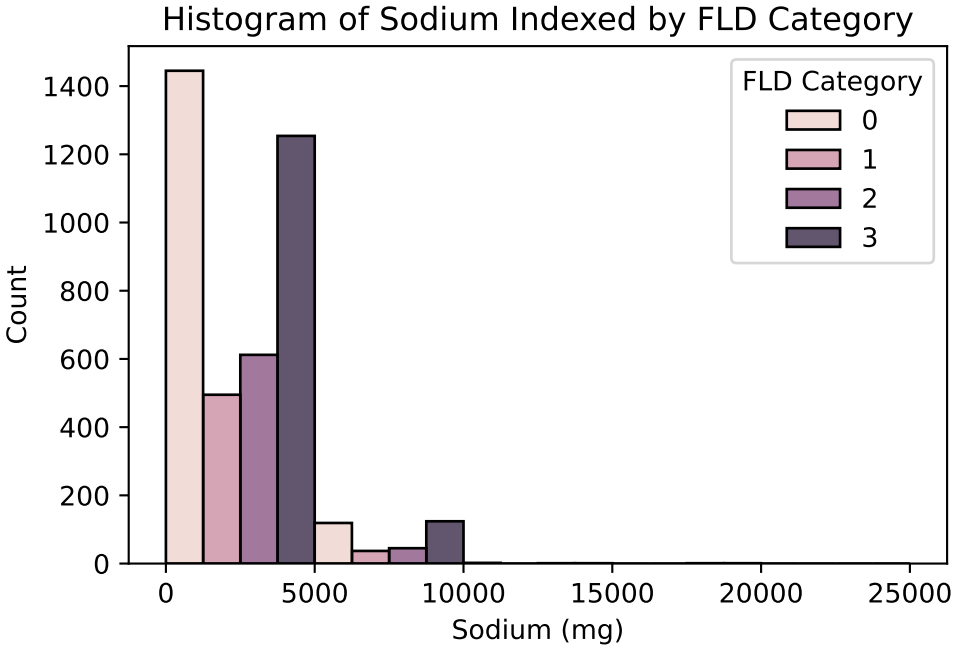
Our water data here is skewed by an individual drinking 20000-25000 gallons. This is abnormal, but possible for a large professional athlete, or someone exercising for a large portion of the day. Lets investigate.

	bmi	water	fat	chol	sugar	carb	salt
3703	26.0	24960.0	34.405	86.5	20.87	114.94	2303.5

We observe this individual to be drinking about 25000 grams of watera day, about 6.5 gallons. This is a lot, and argueably unhealthy, but on the border of kidney capacity and still humanly possible. The rest of their data seems ot be entered properly, and vigorous activity could explain this had we kept it. Hence, we keep the observation, but re-run the graph without this individual (and a few others with high but reasonable water intakes).



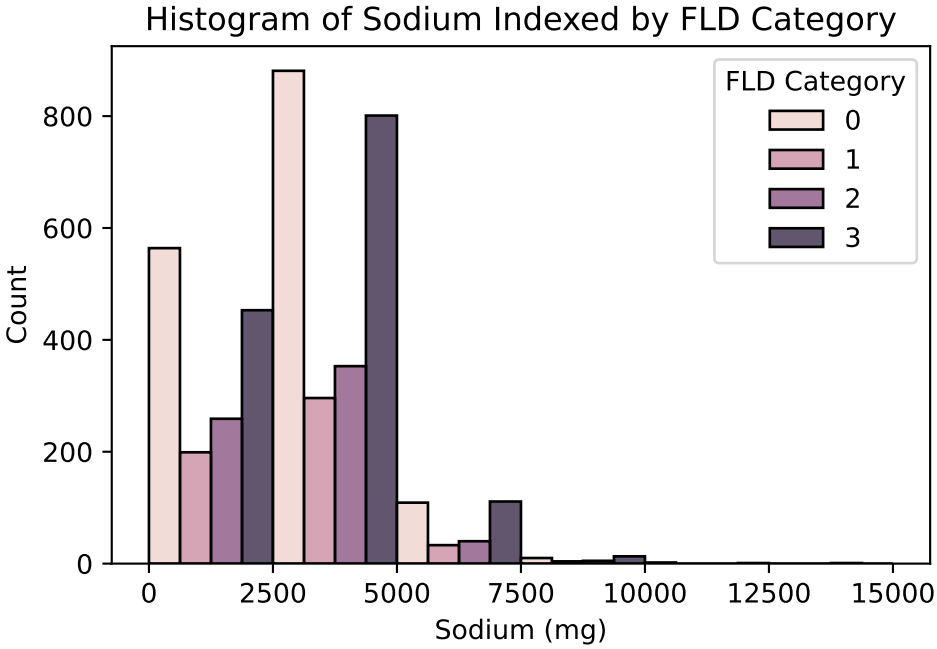
Most individuals consume between 0-2000 grams of water per day. There seems to be no apparent correlation with FLD. This also seems exponentially distributed, and will need standardization.



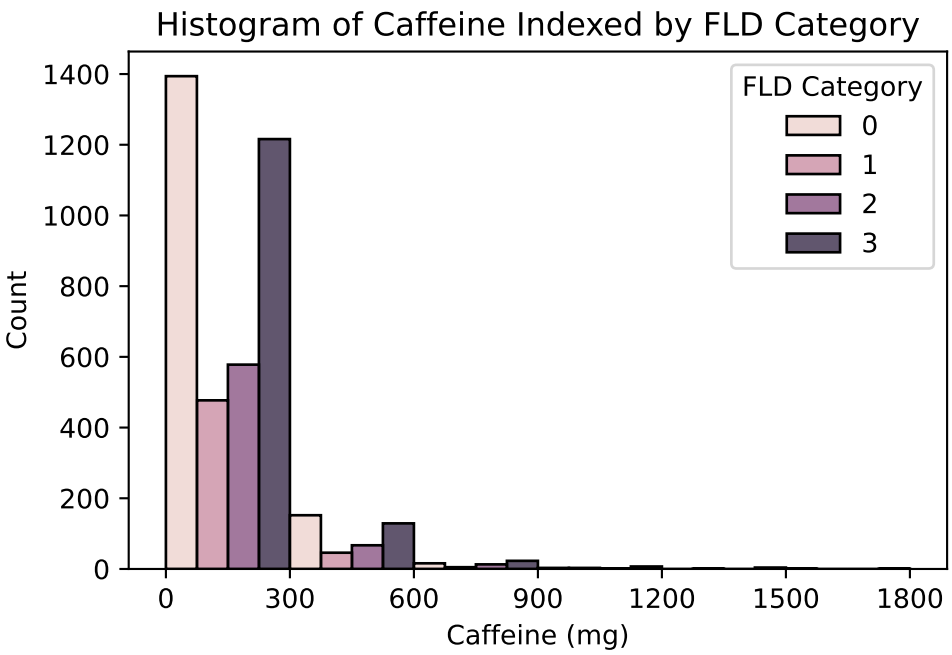
We see an individual consuming almost 20000 milligrams of salt, near lethal levels, and skewing our histogram. Lets investigate.

	bmi	water	fat	chol	sugar	carb	salt
1740	NaN	5.397605e-79	346.415	795.0	158.875	359.22	16300.0
3258	31.7	5.072300e+02	54.425	116.0	202.140	477.28	16573.5

We see both of these individuals consuming on the low end of the 15000 to 20000 sodium bin, around 16000. The first individual in the table has concerning nutrient intake across the board, but their high salt intake lines up with other areas of their diet. The second individual in the table also consumes around 16000, without a diet to match. However, this is still do-able, and while their diet may have been caught on an off day for the interview, there is not enough evidence to justify mis reporting or entry, hence we keep both in the data, but remake the graph without them.



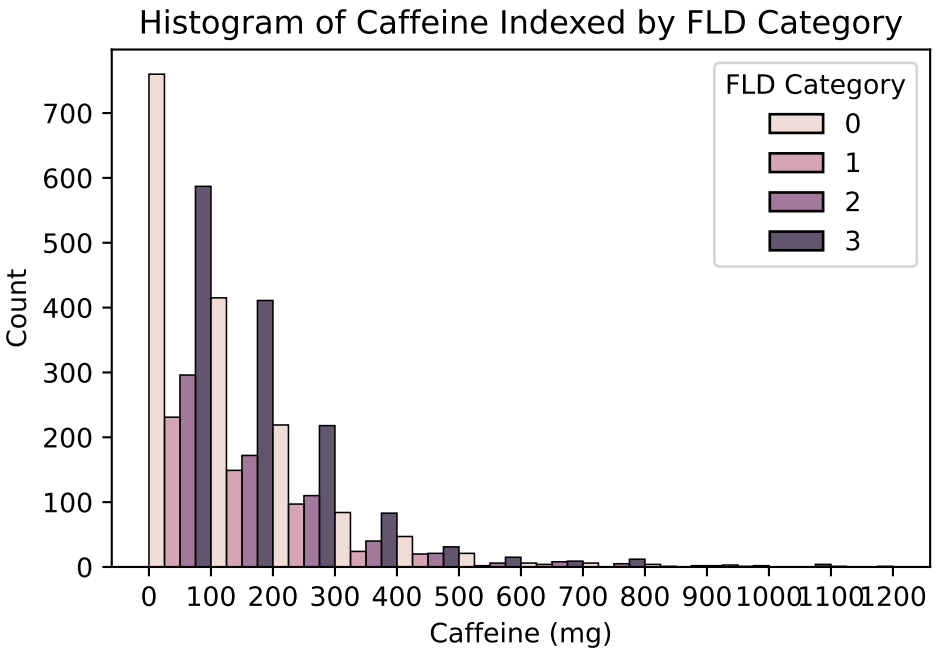
Most people consume well over the daily recommended limit of salt of 2300mg., with the mode between 2500 and 5000, with a sharp dropoff after 5000 mg. However, we see no evident correlation between salt intake and FLD here. This variable will need standardization.



Again, we see some individuals consuming an unhealthy amount of caffiene, some near 1800 mg. This is well over the 1200mg limit at which vomiting and other sicknesses appear. Lets investigate this individual.

	bmi	water	fat	chol	sugar	carb	salt	caff
235	29.1	667.05	114.095	468.5	99.640	298.805	4176.5	1557.5
1354	32.4	1095.00	117.125	335.5	189.080	294.060	4456.0	1221.5
1418	33.5	426.00	132.460	749.5	104.080	460.775	7084.0	1225.5
2529	46.2	240.00	164.610	488.5	104.080	306.665	6779.0	1686.5
4772	34.3	960.00	60.655	290.5	33.955	117.390	2561.5	1485.0
4904	28.2	1200.00	261.020	466.0	327.460	671.585	7708.0	1221.5
5437	31.2	3840.00	84.795	144.0	67.300	234.040	2379.5	1292.5

We notice the none of the other diet entrues to be severely abnormal, and no individual to break 1700mg. While consuming over 1200mg is unhealthy, caffeine is addictive and has a progressively softening affect with gradual increase in consumption. That is to say, 1200mg may look to someone used to 600mg like 600mg looks to someone used to 0mg. (Not an exact scale, just an example to demonstrate the addictive effect of caffiene). Hence while abnormally unhealthy, we have no reason to remove these individuals from observation. Hence, we remake our plot with them truncated out of the graph.



Most individuals consume about 0-200mg of caffine a day, around 1 to 2 cups of coffee. We notice that after 100 mg, the frequency of FLD jumps, and holds at this proportional jump through the rest of the 100mg bins. This variable will also need standardization.

We notice all of our dietary information looks exponential or like a slightly right shifted exponential, which makes sense in relation to food intake, most people consume values relatively around 0, with extreme values pulling the distribution right tailed. However, we have no reason to consider any of this food intake as outliers, as individual diets can vary wildly across culture, individual, and specific diet. Basically, outliers here are in the realm of human possibility, unlike exercising moderately for 12.5 hours a day every week as noted above. Hence we keep all of our diet data. We note inverse relationships with some covariates and FLD, and no relationship at all for others. For example, calories seems to have a positive effect on FLD, and fiber a negative one, whereas carbs has no effect at all at first glance. These covariates noted as having positive or negative effects match up with clinical significance, and should likely remain in the model until significant statistical evidence knocks them out.

Data Transformations

The variables in the following table were standardized to assist with model convergence and possibly future interaction terms, as we have continous data in the '10s' range, and binary covariates, and their data was in the high '100s' to '1000s' range, one or two orders of magnitude higher meaning we may run into convergence issues in the future. Their means and standard deviations are listed for future interpretation reference.

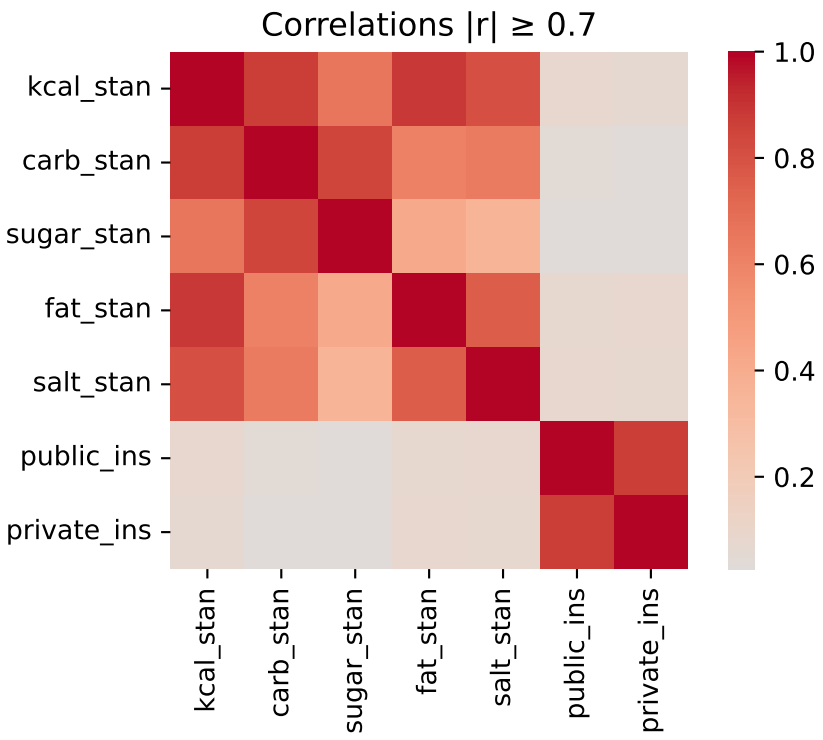
	mins_seden_daily	mod_ex_mins_per_week	kcal	carb	sugar	fat	c
mean	372.219411	265.550504	1987.548875	223.557622	95.275971	84.464123	3
std	203.792536	385.158102	776.527002	97.438838	58.444411	39.223060	2

We will also create dummy variables of each catgorical covariate edu_lvl, diab, and ins and alc_times_yr_recat.

We now have our final dataframe, which we will parse to create our design matrix in the model building stage. It contains id, median cap score and the median cap score categories, all the variables analyzed, their standardization, and dummifications if necessary. This contains all the information needed for any type of model using these variables (standardized or not, dummy relation or linear relation), and the information needed to interpret findings (interpretting standardized covariates).

Covariate Relationships

We will now examine a correlation heatmap for a possible design matrix, basically all the covariates with no_health_ins_12_mon removed due to its obviously high correlation with health insurance measures, and with alc_ever removed, because of its obviously high correlation with alcohol in the past 12 months measures. Furthermore, because we have over 12 variables, we are only going to keep the ones which have high correlation with each other, over 0.7 positive or negative.



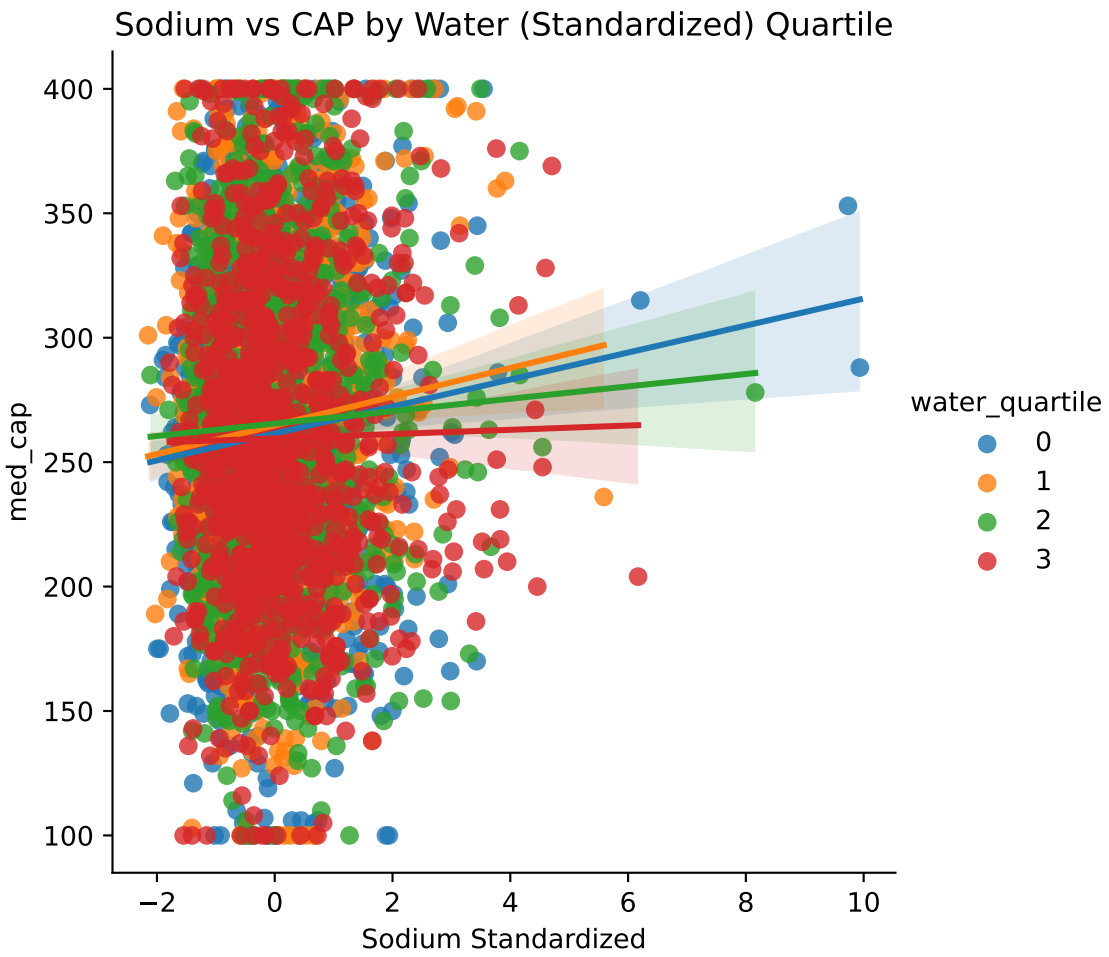
We observe high correlation in insurance, which is expected. One insurance presumes another. We also observe high correlation in som eof our diet variables, particularly carbs, sugar, fat and salt. This also makes sense, as someone who eats a lot is likely to consume a lot of each macronutrient. This is expected in deit survey data however, and our analysis explanation will consider this. Our goal is to try to discern which macronutrients have an effect on FLD.

	Covariate	VIF
0	const	133.944838
1	cig_smoker_ever	1.172163
2	sleep_weekday	1.020674
3	bmi	1.136825
4	sex	1.218884
5	age	1.329975
6	race	1.055249
7	born_us	1.189876
8	fam_inc_to_pov	1.669179
9	hep_b	1.013367
10	mins_seden_daily_stan	1.153800
11	mod_ex_mins_per_week_stan	1.059129
12	kcal_stan	33.940787

	Covariate	VIF
13	carb_stan	15.849751
14	sugar_stan	4.426588
15	fat_stan	10.702515
16	chol_stan	1.740186
17	water_stan	1.117617
18	salt_stan	3.458128
19	caff_stan	1.122174
20	hs_dropout	3.073489
21	hs_grad	7.937333
22	aa	11.608996
23	bach	14.194286
24	prediabetes	1.110836
25	diabetes	1.174878
26	public_ins	4.910514
27	private_ins	5.190577
28	rare_drink	1.847883
29	some_drink	1.821106
30	often_drink	2.058616

We observe high VIF on our dummy variables for education level, which is expected. We also observe high VIF from some of our diet variables, like kcal. This confirms that some diet variables have high multicollinearity, and it may be better to keep only some of them rather than all. KCal for example is likely too high and shouldn't be considered. Hence, in model building, I will need to be careful about which diet covariates to select.

Lastly, I would like to examine a possible interaction term between sodium and water. Sodium and water obviously have an interactive effect on each other in diet. If high sodium intake can be dampened by high water intake, and hence one's effect on CAP score, if there is one, would be effected by the other.



We see no discernable difference in slope between these groups, meaning we don't have evidence to pursue an interaction term between sodium and water.

Model Building Plan

Now that our dataset is finalized and the design matrix can be pulled from it, my model building plan is as follows. Before complete modeling, I will examine whether age needs a log or other transformation, as in my proposal I wanted to give emphasis to younger people. While I don't think I wll have the time or dataset to do this, I will try my best to ensure age is modeled as best as possible. First, I am going to create a proportional odds ordinal logistic model with all the variables I have selected. From there I will remove a few covariates which appear to be obviously insignificant, those with low p-values and little clinical backing. From there, I will test whether there is lack of goodness-of-fit from the proportionality assumption to check whether to continue with my proportional model. Then we will continue to narrow

down the model using significance based and clinical reasoning backwards selection. Once we are considering a few different models, I will then pick my candidate models and compute their BIC, as from a medical perspective an underfit model is likely preferred over a model with false positive covariates. I will also perform LR tests if I am working with nested models. From here I will examine the possibility of linearizing some of my ordinal categorical covariates through a linear or polynomial relationship. From here, I will hopefully be narrowed down to 2-3 models, from which I will perform cross-validation to select my final model, and check my proportionality assumption again. After this, I will perform log-ratio tests on my covariates to confirm their strength, as well as a Pearson Goodness of Fit Test. Finally, I will search for points with high influence and high leverage to consider removal from the model should their information justify it, to hopefully tighten up the model a little more. This plan hopefully covers bases through the rest of the project. (Dr. Li or Muqing, if you are reading this, I would really appreciate feedback on this plan).

Appendix

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import statsmodels.api as sm
from statsmodels.stats.outliers_influence import variance_inflation_factor

nhanes_df_start = pd.read_csv('./data/starting_nhanes_data.csv')
nhanes_df = nhanes_df_start
```

Overview

```
# recoding discrete covariates to be in terms of 0 and 1 instead of 1, 2, and sometimes 3

nhanes_df[['alc_ever', 'cig_smoker_ever', 'health_ins', 'no_ins_12_mon', 'priv_health_ins',

nhanes_df['preg'] = nhanes_df['preg'].replace({3: 0})
nhanes_df['preg'] = nhanes_df['preg'].fillna(0)

# recoding prediabetes and diabetes to be one variable, prediabetes and borderline diabetes t

nhanes_df['diab'] = np.nan

nhanes_df.loc[(nhanes_df['prediabetes'] == 0) & (nhanes_df['diabetes'] == 0), 'diab'] = 0
nhanes_df.loc[(nhanes_df['prediabetes'] == 1) | (nhanes_df['diabetes'] == 3), 'diab'] = 1
nhanes_df.loc[nhanes_df['diabetes'] == 1, 'diab'] = 2
nhanes_df = nhanes_df.drop(columns = ['diabetes', 'prediabetes'])

nhanes_df['ins'] = np.nan

nhanes_df.loc[(nhanes_df['health_ins'] == 0), 'ins'] = 0
nhanes_df.loc[(nhanes_df['health_ins'] == 1), 'ins'] = 1
nhanes_df.loc[(nhanes_df['priv_health_ins'] == 1), 'ins'] = 2
nhanes_df = nhanes_df.drop(columns = ['health_ins', 'priv_health_ins'])

# creating total minutes per week of types of exercise

nhanes_df['mod_ex_mins_per_week'] = nhanes_df['mod_ex_per_week'] * nhanes_df['avg_mins_mod_ex
```

```
nhanes_df['vig_ex_mins_per_week'] = nhanes_df['vig_ex_per_week'] * nhanes_df['avg_mins_vig_ex

# recoding alcohol to be in terms of days of the year, more readable

nhanes_df['alc_times_yr'] = nhanes_df['alc_times_yr'].replace({1: 365, 2: 260, 3: 182.5, 4: 1

nhanes_df['alc_binge_times_yr'] = nhanes_df['alc_binge_times_yr'].replace({1: 365, 2: 260, 3
```

Examining Missing values

```
# observing total amount of missing values in each column

nhanes_df.isna().sum()
```

```
# removing columns

nhanes_df = nhanes_df.drop(columns = ['insulin', 'alc_binge_times_yr', 'age_of_diabetes', 'un

nhanes_df = nhanes_df.drop(columns = ['unit_mod_ex', 'freq_mod_ex', 'freq_vig_ex', 'avg_mins_

nhanes_df = nhanes_df.drop(columns = ['iqr_cap', 'sleep_weekend'])

count = nhanes_df[['kcal', 'carb', 'sugar', 'fiber', 'fat', 'vit_e', 'chol', 'water', 'salt'
```

Dietary Information: A Complete Case Analysis?

```
# examining individuals with and without diet demographics data

no_diet = nhanes_df[nhanes_df[['kcal', 'carb', 'sugar', 'fiber', 'fat', 'vit_e', 'chol', 'wat

diet = nhanes_df[nhanes_df[['kcal', 'carb', 'sugar', 'fiber', 'fat', 'vit_e', 'chol', 'water

no_diet['subset'] = 'Dietary Info'
diet['subset'] = 'No Dietary Info'

diet_compare_df = pd.concat([no_diet, diet])

sns.boxplot(data = diet_compare_df, x = 'subset', y = 'med_cap')
```

```
plt.title('Distribution of Median Cap Score Across Diet Info')
plt.xlabel('Subset')
plt.ylabel('Median Cap Score')
plt.show()

sns.boxplot(data = diet_compare_df, x = 'subset', y = 'age')
plt.title('Distribution of Age Across Diet Info')
plt.xlabel('Subset')
plt.ylabel('Age (years)')
plt.show()

sns.boxplot(data = diet_compare_df, x = 'subset', y = 'bmi')
plt.title('Distribution of BMI Across Diet Info')
plt.xlabel('Subset')
plt.ylabel('BMI')
plt.show()

race_table = pd.crosstab(
    diet_compare_df['race'],
    diet_compare_df['subset'],
    normalize='columns'
)

display(race_table.rename(index = {
    1: 'Mexican American',
    2: 'Other Hispanic',
    3: 'White',
    4: 'Black',
    6: 'Asian',
    7: 'Other'
}))

sex_table = pd.crosstab(
    diet_compare_df['sex'],
    diet_compare_df['subset'],
    normalize='columns'
)

display(sex_table.rename(index = {
    0: 'Female',
    1: 'Male'
}))
```

```
diabetes_table = pd.crosstab(
    diet_compare_df['diab'],
    diet_compare_df['subset'],
    normalize='columns'
)

display(diabetes_table.rename(index = {
    0: 'No Diabetes',
    1: 'Prediabetes / Borderline',
    2: 'Diabetes'
}))

hep_b_table = pd.crosstab(
    diet_compare_df['hep_b'],
    diet_compare_df['subset'],
    normalize='columns'
)

display(hep_b_table.rename(index = {
    0: 'No',
    1: 'Yes'
}))
```

```
# dropping pregnancy and those without diet

nhanes_df = nhanes_df.drop(columns = 'preg')

nhanes_df = nhanes_df.dropna(subset = ['kcal'])

nhanes_df.isna().sum()
```

Data Visualization (Discrete Covariates)

```
alc_ever_plot = sns.violinplot(data = nhanes_df, x = 'alc_ever', y = 'med_cap')
plt.title('Median Cap Score Across Alcohol Use Presence')
plt.xlabel('Alcohol Usage')
plt.ylabel('Median Cap Score')
counts = nhanes_df.groupby("alc_ever")["med_cap"].count()
new_labels = [f"{g}\n(n={counts[g]})" for g in counts.index]
_plot.set_xticklabels(new_labels)
```

```
alc_ever_plot.axhline(238, color="red")
alc_ever_plot.axhline(260, color="red")
alc_ever_plot.axhline(290, color="red")
plt.show()
```

```
alc_times_yr_plot = sns.violinplot(data = nhanes_df, x = 'alc_times_yr', y = 'med_cap')
plt.title('Median Cap Score Alcohol Use per Year')
plt.xlabel('Days Alcohol Drunk Last Year')
plt.ylabel('Median Cap Score')
alc_times_yr_plot.axhline(238, color="red")
alc_times_yr_plot.axhline(260, color="red")
alc_times_yr_plot.axhline(290, color="red")
plt.show()
```

```
nhanes_df.loc[nhanes_df['alc_ever'].isin([0]), 'alc_times_yr'] = 0

nhanes_df['alc_times_yr_recat'] = np.nan
nhanes_df.loc[nhanes_df['alc_times_yr'].isin([0]), 'alc_times_yr_recat'] = 0
nhanes_df.loc[nhanes_df['alc_times_yr'].isin([1.5, 4.5, 7, 9, 12]), 'alc_times_yr_recat'] = 1
nhanes_df.loc[nhanes_df['alc_times_yr'].isin([104, 52, 30]), 'alc_times_yr_recat'] = 2
nhanes_df.loc[nhanes_df['alc_times_yr'].isin([365, 260, 182.5]), 'alc_times_yr_recat'] = 3
```

```
alc_times_yr_recat_plot = sns.violinplot(data = nhanes_df, x = 'alc_times_yr_recat', y = 'med
plt.title('Median Cap Score Across Past Year Alc')
plt.xlabel('Alcohol Use on the Year')
plt.ylabel('Median Cap Score')
counts = nhanes_df.groupby("alc_times_yr_recat")["med_cap"].count()
new_labels = [f"{g}\n(n={counts[g]})" for g in counts.index]
alc_times_yr_recat_plot.set_xticklabels(new_labels)
alc_times_yr_recat_plot.axhline(238, color="red")
alc_times_yr_recat_plot.axhline(260, color="red")
alc_times_yr_recat_plot.axhline(290, color="red")
plt.show()
```

```
cig_smoker_ever_plot = sns.violinplot(data = nhanes_df, x = 'cig_smoker_ever', y = 'med_cap'
plt.title('Median Cap Score Across Cig Smoke')
plt.xlabel('More than 100 Cigs Lifetime')
plt.ylabel('Median Cap Score')
counts = nhanes_df.groupby("cig_smoker_ever")["med_cap"].count()
new_labels = [f"{g}\n(n={counts[g]})" for g in counts.index]
cig_smoker_ever_plot.set_xticklabels(new_labels)
```

```
cig_smoker_ever_plot.axhline(238, color="red")
cig_smoker_ever_plot.axhline(260, color="red")
cig_smoker_ever_plot.axhline(290, color="red")
plt.show()
```

```
no_ins_12_mon_plot = sns.violinplot(data = nhanes_df, x = 'no_ins_12_mon', y = 'med_cap')
plt.title('Median Cap Score Across Insurance Loss in Year')
plt.xlabel('Lost Insurance in the Past Year')
plt.ylabel('Median Cap Score')
counts = nhanes_df.groupby("no_ins_12_mon")["med_cap"].count()
new_labels = [f"{g}\n(n={counts[g]})" for g in counts.index]
no_ins_12_mon_plot.set_xticklabels(new_labels)
no_ins_12_mon_plot.axhline(238, color="red")
no_ins_12_mon_plot.axhline(260, color="red")
no_ins_12_mon_plot.axhline(290, color="red")
plt.show()
```

```
sex_plot = sns.violinplot(data = nhanes_df, x = 'sex', y = 'med_cap')
plt.title('Median Cap Score Across Sex')
plt.xlabel('Sex (0:F, 1:M)')
plt.ylabel('Median Cap Score')
counts = nhanes_df.groupby("sex")["med_cap"].count()
new_labels = [f"{g}\n(n={counts[g]})" for g in counts.index]
sex_plot.set_xticklabels(new_labels)
sex_plot.axhline(238, color="red")
sex_plot.axhline(260, color="red")
sex_plot.axhline(290, color="red")
plt.show()
```

```
race_map = {
    1: "Mexican",
    2: "Hispanic",
    3: "White",
    4: "Black",
    6: "Asian",
    7: "Other"
}

race_plot = sns.violinplot(data = nhanes_df, x = 'race', y = 'med_cap')
plt.title('Median Cap Score Across Race')
plt.xlabel('Race')
```

```
plt.ylabel('Median Cap Score')
counts = nhanes_df.groupby("race")["med_cap"].count()
new_labels = [
    f"{race_map[g]}\n(n={counts[g]})"
    for g in sorted(counts.index)
]
race_plot.set_xticks(race_plot.get_xticks(), labels=new_labels)
race_plot.axhline(238, color="red")
race_plot.axhline(260, color="red")
race_plot.axhline(290, color="red")
plt.show()
```

```
born_us_plot = sns.violinplot(data = nhanes_df, x = 'born_us', y = 'med_cap')
plt.title('Median Cap Score Across Born in US')
plt.xlabel('Born in US')
plt.ylabel('Median Cap Score')
counts = nhanes_df.groupby("born_us")["med_cap"].count()
new_labels = [f"{g}\n(n={counts[g]})" for g in counts.index]
born_us_plot.set_xticklabels(new_labels)
born_us_plot.axhline(238, color="red")
born_us_plot.axhline(260, color="red")
born_us_plot.axhline(290, color="red")
plt.show()
```

```
edu_map = {
    1: "No HS",
    2: "HS Dropout",
    3: "HS Grad",
    4: "AA",
    5: "College",
}

edu_lvl_plot = sns.violinplot(data = nhanes_df, x = 'edu_lvl', y = 'med_cap')
plt.title('Median Cap Score Across Education Level')
plt.xlabel('Education Level')
plt.ylabel('Median Cap Score')
counts = nhanes_df.groupby("edu_lvl")["med_cap"].count()
new_labels = [
    f"{edu_map[g]}\n(n={counts[g]})"
    for g in sorted(counts.index)
]
edu_lvl_plot.set_xticks(edu_lvl_plot.get_xticks(), labels=new_labels)
```

```
edu_lvl_plot.axhline(238, color="red")
edu_lvl_plot.axhline(260, color="red")
edu_lvl_plot.axhline(290, color="red")
plt.show()
```

```
hep_b_plot = sns.violinplot(data = nhanes_df, x = 'hep_b', y = 'med_cap')
plt.title('Median Cap Score Across Hep B')
plt.xlabel('Hep B')
plt.ylabel('Median Cap Score')
counts = nhanes_df.groupby("hep_b")["med_cap"].count()
new_labels = [f"{g}\n(n={counts[g]})" for g in counts.index]
hep_b_plot.set_xticklabels(new_labels)
hep_b_plot.axhline(238, color="red")
hep_b_plot.axhline(260, color="red")
hep_b_plot.axhline(290, color="red")
plt.show()
```

```
diab_map = {
    0: 'None',
    1: 'Prediabetes',
    2: 'Diabetes'
}

diab_plot = sns.violinplot(data = nhanes_df, x = 'diab', y = 'med_cap')
plt.title('Median Cap Score Across Diabetes')
plt.xlabel('Diabetes')
plt.ylabel('Median Cap Score')
counts = nhanes_df.groupby("diab")["med_cap"].count()
new_labels = [
    f"{diab_map[g]}\n(n={counts[g]})"
    for g in sorted(counts.index)
]
diab_plot.set_xticks(diab_plot.get_xticks(), labels=new_labels)
diab_plot.axhline(238, color="red")
diab_plot.axhline(260, color="red")
diab_plot.axhline(290, color="red")
plt.show()
```

```
ins_map = {
    0: 'No Ins',
    1: 'Public Ins',
```

```
    2: 'Priv Ins'
}

ins_plot = sns.violinplot(data = nhanes_df, x = 'ins', y = 'med_cap')
plt.title('Median Cap Score Across Insurance')
plt.xlabel('Insurance')
plt.ylabel('Median Cap Score')
counts = nhanes_df.groupby("ins")["med_cap"].count()
new_labels = [
    f"{ins_map[g]}\n(n={counts[g]})"
    for g in sorted(counts.index)
]
ins_plot.set_xticks(ins_plot.get_xticks(), labels=new_labels)
ins_plot.axhline(238, color="red")
ins_plot.axhline(260, color="red")
ins_plot.axhline(290, color="red")
plt.show()
```

Data Visualization (Continous Covariates)

```
nhanes_df_plot = nhanes_df.rename(columns={'cap_cat': 'FLD Category'})
```

```
binwidth = 8
min_val = nhanes_df["bmi"].min()
max_val = nhanes_df["bmi"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

bmi_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'bmi',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('BMI')
plt.title('Histogram of BMI Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
binwidth = 10
min_val = nhanes_df["age"].min()
max_val = nhanes_df["age"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

age_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'age',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Age')
plt.title('Histogram of Age Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
binwidth = 1
min_val = nhanes_df["fam_inc_to_pov"].min()
max_val = nhanes_df["fam_inc_to_pov"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

fam_inc_to_pov_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'fam_inc_to_pov',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Family Income to Poverty Line Ratio')
plt.title('Histogram of Income to Poverty Ratio Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
binwidth = 240
min_val = nhanes_df["mins_seden_daily"].min()
max_val = nhanes_df["mins_seden_daily"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

mins_seden_daily_plot = sns.histplot(
```

```
    data = nhanes_df_plot,
    x = 'mins_seden_daily',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Minutes Sedentary (Daily)')
plt.title('Histogram of Sedentary Minutes Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
binwidth = 2
min_val = nhanes_df["sleep_weekday"].min()
max_val = nhanes_df["sleep_weekday"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

sleep_weekday_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'sleep_weekday',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Sleep on Weekdays (Hours)')
plt.title('Histogram of Weekday Sleep Hours Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
binwidth = 20
min_val = nhanes_df["mod_ex_per_week"].min()
max_val = nhanes_df["mod_ex_per_week"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

mod_ex_per_week_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'mod_ex_per_week',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)
```

```
plt.xlabel('Moderate Exercise Instances (Week)')
plt.title('Histogram of Moderate Exercise Instances Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
# table of sus mod_ex_per_week values

filtered_df = nhanes_df[nhanes_df["mod_ex_per_week"] > 60]
display(filtered_df)
```

```
binwidth = 1000
min_val = nhanes_df["mod_ex_mins_per_week"].min()
max_val = nhanes_df["mod_ex_mins_per_week"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

mod_ex_mins_per_week_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'mod_ex_mins_per_week',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Moderate Exercise Minutes')
plt.title('Histogram of Moderate Exercise Minutes Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
nhanes_df = nhanes_df[
    (nhanes_df["mod_ex_mins_per_week"] < 5000) |
    (nhanes_df["mod_ex_mins_per_week"].isna())
]
```

```
nhanes_df = nhanes_df[
    (nhanes_df["mod_ex_mins_per_week"] < 5000) |
    (nhanes_df["mod_ex_mins_per_week"].isna())
]
```

```
nhanes_df_plot_avg_mins = nhanes_df.rename(columns={'cap_cat': 'FLD Category'})
```

```
nhanes_df_plot_avg_mins = nhanes_df_plot_avg_mins[nhanes_df_plot_avg_mins['mod_ex_mins_per_w

binwidth = 200
min_val = nhanes_df_plot_avg_mins["mod_ex_mins_per_week"].min()
max_val = nhanes_df_plot_avg_mins["mod_ex_mins_per_week"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

mod_ex_mins_per_week_plot = sns.histplot(
    data = nhanes_df_plot_avg_mins,
    x = 'mod_ex_mins_per_week',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Moderate Exercise Minutes')
plt.title('Histogram of Moderate Exercise Minutes Indexed by FLD Category')
plt.xticks(bins)
plt.show()

nhanes_df = nhanes_df.drop(columns = 'mod_ex_per_week')
```

```
binwidth = 1000
min_val = 0
max_val = 9000
bins = np.arange(min_val, max_val + binwidth, binwidth)

kcal_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'kcal',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('KCals')
plt.title('Histogram of KCal Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
binwidth = 150
min_val = 0
max_val = nhanes_df["carb"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

carb_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'carb',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Carbs (gm)')
plt.title('Histogram of Carb Intake Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
binwidth = 100
min_val = 0
max_val = nhanes_df["sugar"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

sugar_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'sugar',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Sugar (gm)')
plt.title('Histogram of Sugar Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
binwidth = 15
min_val = 0
max_val = nhanes_df["fiber"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

fiber_plot = sns.histplot(
```

```
    data = nhanes_df_plot,
    x = 'fiber',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Fiber (gm)')
plt.title('Histogram of Fiber Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
binwidth = 50
min_val = 0
max_val = nhanes_df["fat"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

fat_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'fat',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Fat (gm)')
plt.title('Histogram of Fat Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
binwidth = 10
min_val = 0
max_val = nhanes_df["vit_e"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

vit_e_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'vit_e',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)
```

```
plt.xlabel('Vit_E (mg)')
plt.title('Histogram of Vitamin E Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
binwidth = 300
min_val = 0
max_val = nhanes_df["chol"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

chol_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'chol',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Cholesterol (mg)')
plt.title('Histogram of Cholesterol Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
binwidth = 5000
min_val = 0
max_val = nhanes_df["water"].max()
bins = np.arange(min_val, max_val + binwidth, binwidth)

water_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'water',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Water (gm)')
plt.title('Histogram of Water Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
# table of sus water values

water_invest = nhanes_df[nhanes_df['water'] > 20000]
display(water_invest[['bmi', 'water', 'fat', 'chol', 'sugar', 'carb', 'salt']])
```

```
binwidth = 1000
min_val = 0
max_val = 8000
bins = np.arange(min_val, max_val + binwidth, binwidth)

water_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'water',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Water (gm)')
plt.title('Histogram of Water Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
binwidth = 5000
min_val = 0
max_val = 25000
bins = np.arange(min_val, max_val + binwidth, binwidth)

salt_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'salt',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Sodium (mg)')
plt.title('Histogram of Sodium Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
# table of sus salt values

salt_invest = nhanes_df[nhanes_df['salt'] > 15000]
display(salt_invest[['bmi', 'water', 'fat', 'chol', 'sugar', 'carb', 'salt']])
```

```
binwidth = 2500
min_val = 0
max_val = 15000
bins = np.arange(min_val, max_val + binwidth, binwidth)

salt_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'salt',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Sodium (mg)')
plt.title('Histogram of Sodium Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
binwidth = 300
min_val = 0
max_val = 1800
bins = np.arange(min_val, max_val + binwidth, binwidth)

caff_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'caff',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Caffeine (mg)')
plt.title('Histogram of Caffeine Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

```
# table of sus caff values

caff_invest = nhanes_df[nhanes_df['caff'] > 1200]
display(caff_invest[['bmi', 'water', 'fat', 'chol', 'sugar', 'carb', 'salt', 'caff']])
```

```
binwidth = 100
min_val = 0
max_val = 1200
bins = np.arange(min_val, max_val + binwidth, binwidth)

caff_plot = sns.histplot(
    data = nhanes_df_plot,
    x = 'caff',
    hue = 'FLD Category',
    multiple = 'dodge',
    bins = bins
)

plt.xlabel('Caffeine (mg)')
plt.title('Histogram of Caffeine Indexed by FLD Category')
plt.xticks(bins)
plt.show()
```

Data Transformations

```
# standardizing necessary variables

to_standardize_list = ['mins_seden_daily', 'mod_ex_mins_per_week', 'kcal', 'carb', 'sugar',

standardize_list = ['mins_seden_daily_stan', 'mod_ex_mins_per_week_stan', 'kcal_stan', 'carb

for i in range(len(standardize_list)):
    nhanes_df[standardize_list[i]] = (nhanes_df[to_standardize_list[i]] - np.mean(nhanes_df[t

display(nhanes_df[to_standardize_list].agg(["mean", "std"]))
```

```
# creating and renaming dummy variables and ensuring na's stay around rather than get replace

cols = ['edu_lvl', 'diab', 'ins', 'alc_times_yr_recat']
na_masks = {col: nhanes_df[col].isna() for col in cols}
```

```
nhanes_df = pd.get_dummies(
    nhanes_df,
    columns = cols,
    prefix = {
        'edu_lvl': 'edu_lvl',
        'diab': 'diab',
        'ins': 'ins',
        'alc_times_yr_recat': 'alc_times_yr_recat'
    },
    drop_first = True,
    dtype = float
)

for col in cols:
    dummy_cols = [c for c in nhanes_df.columns if c.startswith(col + "_")]
    nhanes_df.loc[na_masks[col], dummy_cols] = np.nan

nhanes_df = nhanes_df.rename(columns = {
    'edu_lvl_2.0': 'hs_dropout',
    'edu_lvl_3.0': 'hs_grad',
    'edu_lvl_4.0': 'aa',
    'edu_lvl_5.0': 'bach',
    'diab_1.0': 'prediabetes',
    'diab_2.0': 'diabetes',
    'ins_1.0': 'public_ins',
    'ins_2.0': 'private_ins',
    'alc_times_yr_recat_1.0': 'rare_drink',
    'alc_times_yr_recat_2.0': 'some_drink',
    'alc_times_yr_recat_3.0': 'often_drink'
}
)
```

Covariate Relationships

```
# creating candidate deisgn matrix and making correlation heatmap for those with correlation

cand_list = ['cig_smoker_ever',
       'sleep_weekday', 'bmi', 'sex', 'age', 'race',
       'born_us', 'fam_inc_to_pov', 'hep_b',  'mins_seden_daily_stan',
       'mod_ex_mins_per_week_stan', 'kcal_stan', 'carb_stan', 'sugar_stan', 'fat_stan', 'cho
```

```
design_cand = nhanes_df[cand_list]
corr_design_cand = design_cand.corr().abs()

threshold = 0.7
strong_pairs = (corr_design_cand >= threshold) & (corr_design_cand < 1)
vars_strong = strong_pairs.any(axis = 0)

filt_corr_design_cand = corr_design_cand.loc[vars_strong, vars_strong]

sns.heatmap(
    filt_corr_design_cand,
    cmap = 'coolwarm',
    center=0,
    square=True
)

plt.title(f'Correlations |r|   {threshold}')
plt.show()
```

```
# VIF table

design_cand = sm.add_constant(design_cand)

design_cand = design_cand.dropna() # we must drop NAs for VIF

vif_df = pd.DataFrame({
    "Covariate": design_cand.columns,
    "VIF": [variance_inflation_factor(design_cand.values, i)
            for i in range(design_cand.shape[1])]
})

display(vif_df)
```

```
# plot of sodium against cap score color coated by water quartiles to see if water has an ef

nhanes_df['water_quartile'] = pd.qcut(nhanes_df['water_stan'], 4, labels=False) # getting fo

sns.lmplot(
    data = nhanes_df,
    x = 'salt_stan',
    y = 'med_cap',
```

```
    hue = 'water_quartile',
)

plt.title("Sodium vs CAP by Water (Standardized) Quartile")
plt.xlabel('Sodium Standardized')
plt.show()
```

Model Building plan

```
nhanes_df.to_csv('./data/final_nhanes_df.csv', index = False)
```