BST 223: Project Proposal

Lucas Webster

Proposal

I propose to use National Health and Nutrition Examination Survey (NHANES) data from the NHANES survey conducted from August 2021 to August 2023, found at this link. This data is collected by the CDC every two years, pandemic permitting, on a diverse set of about 12000 Americans young and old, although not every particpant is able to participate in every part of the survey. The survey is anonymized and was conducted partly via a computer questionairre, and partly by in-person questionnaire and medical examinations.

With upwards of 50 different questionairre and exam datasets provided from the study, I have selected 13 to analyze in adults 18 or over. These include the alcohol use, cigarette smoking, demographics, diabetes, health_insurance, insulin, liver ultrasound, nutrient intake day 1, nutrient intake day 2, physical activity, sleep, and body measures, and Hepatitis B datasets.

My outcome of interest from this selection is from the liver ultrasound dataset, specifically median CAP measurement (100 to 400), which is a measure of fat in the liver. Fatty liver disease, which is when 5%-10% or more of the liver's total weight is comprised of fat, is common among older alcohlics, as alcohol damages the liver, but also among the inactive with poor diet. It can lead to fibrosis and eventually cirrhosis of the liver, resulting in transplant or death. Addionally, Non Alcoholic Fatty Liver Disease (NAFLD) is on rise, notably among young people. With FLD changing demographics from that of old alcoholics to the general American, I want to investigate the cause. This survey provides ample information for this investigation, inclduing the CAP measurement (as well as a level of uncertainty in the measurement), alcohol history, smoking history, exercise history, dietary information, diabetes history, and more, all of which are known to be influential on fat accumulation in the liver. In short, I selected promising demographic and lifestyle covariates, and a few clinical covariates with surefire connection to fatty liver disease.

Specifically, I am looking to test which of my covariates (mostly lifestyle and demographic) significantly increase the likelihood of high fat content in the liver, with further investigative focus being directed to young people and non-alcoholics. While the median CAP is provided, I transform it into categories related to clinical cutoffs, with a score of 100-238 being given a 0 (no Fatty Liver Disease), 238-260 being given a 1 (Steatosis Grade 1, Mild Fatty Liver Disease

Present) 260-290 being given a 2 (Steatosis Grade 2, Moderate Fatty Liver Disease Present), and 290-400 given a 3 (Steatosis Grade 3, Severe Fatty Liver Disease Present).

Because of the ordered categorical nature of the classifications of Steatosis Grades associated with median CAP scores, I will use an ordinal logistic regression model (ordered logistic family) for my data, with the link function taking the form of the cumulative logit $g()...$,

$$g(\pi_j) = \log(\frac{\pi_1(x) + ... + \pi_j(x)}{\pi_{j+1}(x) + ... + \pi_3(x)}) = P(Y \leq j|X = x) = \alpha_j + \beta(x), (j = 0, 1, 2)$$

With Y denoting the steatosis category and X denoting the design matrix.

Observe here I assume proportional odds in my model. If I wasn't to assume this, my β would be indexed by a j to become β_j. In the coming analysis, I will be checking this assumption, and then either moving forward with a proportional odds cumulative logistic regression model, or a non-proportional odds cumulative logistic regression model.

Data

```
Number of observations (rows): 5873
Number of variables (columns): 45

Column names:
- id
- med_cap
- iqr_cap
- alc_ever
- alc_times_yr
- alc_binge_times_yr
- cig_smoker_ever
- sleep_weekday
- sleep_weekend
- health_ins
- no_ins_12_mon
- priv_health_ins
- insulin
- bmi
- sex
- age
- race
- born_us
- edu_lvl
- preg
- fam_inc_to_pov
- diabetes
- prediabetes
- age_of_diabetes
- freq_mod_ex
- unit_mod_ex
- avg_mins_mod_ex
- freq_vig_ex
- unit_vig_ex
- avg_mins_vig_ex
- mins_seden_daily
- hep_b
- kcal
- carb
- sugar
- fiber
- fat
- vit_e
- chol
- water
- salt
- caff
- mod_ex_per_week
- vig_ex_per_week
- cap_cat
```

From the above table, we can see there to be well over 20 covariates, though some may be cut down or combined in future stages, and we have 5873 observations, meaning this dataset created from the merging and filtering of 13 NHANES datasets from the 2021-2023 study meets the requirements. This dataset was filtered out of NHANES total to include only individuals over the age of 18 who completed the liver ultrasound to receive a median CAP score.

As far as dataset complexity goes, the covariates above are included for the following reasons. med_cap and iqr_cap are the median CAP score and IQR of the CAP score. We use the median to place individuals into categories, and IQR will be investigated as a possible measure of outcome certainty. The alc covariates represent different alcohol measures, and alcohol is a known driver of fatty liver disease having a direct impact on the CAP score. Smoking as well can effect diet, exercise, and metabolism, effecting CAP. the sleep covariates are measure the amount of sleep an individual averages on weekdays and weekends respectively, a critical health measure, and poor sleep has a myriad of health effects. health_ins measures whether someone has health insurance, priv_health_ins measures whether someone has private health insurance, and no_ins_12_mon measures whether someone has not had health insurance in the past 12 months. These are included as access to healthcare is critical in heatlh outcomes like FLD, and these measure that. insulin measures and individuals insulin levels after fasting. Poor insulin regulation is a surefire driver of FLD. BMI, sex, age, race are all self explanatory and help will help discretize our model across demographics. born_us signals whether someone was born in United States, as culture influence health outcomes. Education level is a mask for healthcare access as well, as those who experience higher education are more likely to understand healthcare and need for it better. preg measures whether an individual is pregnant, which drastically effects other measures in the dataset. fam_inc_to_pov measures the individuals household income relative to the poverty line. diabetes and prediabetes indicate if an individual has either of the two, which effects insulin regulation and directly effects FLD. age_of_diabetes indicates how long someone with diabetes has had the disease, which also puts them at greater risk for FLD. Any covariate including ex is a measure of moderate or vigorous exercise. Exercise helps regulate insulin levels and directly reduces chances of FLD. mins_seden_daily is a measure of how many minutes an individual spends sedentary on average, which is likely to increase FLD risk. hep_b indiciates whether an individual has hepatitis_b, another risk factor for FLD. The remaining covariates are measures of nutrient intake over a two day average, with the two days being from 3-10 days apart. Diet is another critical part of managing insulin levels and FLD risk, and hence, nutrient intake data is included. Our last variable in the list is cap_cat, our category for the CAP score for which we are creating this model to predict.

The above are all hypothesized to have an effect on an individual's CAP score and hence FLD classification, and the complexity is hence justified. Variables will be mutated, scaled, and dropped as necessary in the following stages.

Appendix

```python
import pandas as pd
import numpy as np

alc_use = pd.read_sas('./data/alcohol_use.xpt', format = 'xport')
cig_smoke = pd.read_sas('./data/cigarette_smoking.xpt', format = 'xport')
demographics = pd.read_sas('./data/demographics.xpt', format = 'xport')
diabetes = pd.read_sas('./data/diabetes.xpt', format = 'xport')
health_insurance = pd.read_sas('./data/health_insurance.xpt', format = 'xport')
insulin = pd.read_sas('./data/insulin.xpt', format = 'xport')
liver_ultra = pd.read_sas('./data/liver_ultra.xpt', format = 'xport')
nutrient_day_1 = pd.read_sas('./data/nutrient_intake_day1.xpt', format = 'xport')
nutrient_day_2 = pd.read_sas('./data/nutrient_intake_day2.xpt', format = 'xport')
physical_act_a = pd.read_sas('./data/physical_activity_adults.xpt', format = 'xport')
sleep = pd.read_sas('./data/sleep.xpt', format = 'xport')
weight_height = pd.read_sas('./data/weight_height.xpt', format = 'xport')
hep_b = pd.read_sas('./data/hep_b.xpt', format = 'xport')
```

```python
alc_use = alc_use[['SEQN', 'ALQ111', 'ALQ121', 'ALQ280']]
cig_smoke = cig_smoke[['SEQN', 'SMQ020']]
sleep = sleep[['SEQN', 'SLD012', 'SLD013']]
health_ins = health_insurance[['SEQN', 'HIQ011', 'HIQ210', 'HIQ032A']]
liver_ultra = liver_ultra[['SEQN' ,'LUXCAPM', 'LUXCPIQR']]
insulin = insulin[['SEQN', 'LBXIN']]
bmi = weight_height[['SEQN', 'BMXBMI']]
demographics = demographics[['SEQN', 'RIAGENDR', 'RIDAGEYR', 'RIDRETH3', 'DMDBORN4', 'DMDEDUC
nutrient_day_1 = nutrient_day_1[['SEQN', 'DR1TKCAL', 'DR1TCARB', 'DR1TSUGR', 'DR1TFIBE', 'DR1
nutrient_day_2 = nutrient_day_2[['SEQN', 'DR2TKCAL', 'DR2TCARB', 'DR2TSUGR', 'DR2TFIBE', 'DR2
diabetes = diabetes[['SEQN', 'DIQ010', 'DIQ160', 'DID040']]

nhanes_df = liver_ultra.copy()

for df in [alc_use, cig_smoke, sleep, health_ins, insulin, bmi, demographics, nutrient_day_1
    nhanes_df = nhanes_df.merge(df, on = 'SEQN', how = 'left')
```

```python
# simplifying diet data

day1 = nhanes_df.iloc[:, 21:31]
day2 = nhanes_df.iloc[:, 31:41]
```

```
day_avg = (day1.values + day2.values) / 2
col_names = ['kcal', 'carb', 'sugar', 'fiber', 'fat', 'vit_e', 'chol', 'water', 'salt', 'caf

day_avg_df = pd.DataFrame(
    day_avg,
    columns = col_names)

nhanes_df[col_names] = day_avg_df
nhanes_df = nhanes_df.drop(columns = nhanes_df.columns[21:41])
nhanes_df = nhanes_df.rename(columns = {
    'SEQN': 'id',
    'LUXCAPM': 'med_cap',
    'LUXCPIQR': 'iqr_cap',
    'ALQ111': 'alc_ever',
    'ALQ121': 'alc_times_yr',
    'ALQ280': 'alc_binge_times_yr',
    'SMQ020': 'cig_smoker_ever',
    'SLD012': 'sleep_weekday',
    'SLD013': 'sleep_weekend',
    'HIQ011': 'health_ins',
    'HIQ210': 'no_ins_12_mon',
    'HIQ032A': 'priv_health_ins',
    'LBXIN': 'insulin',
    'BMXBMI': 'bmi',
    'RIAGENDR': 'sex',
    'RIDAGEYR': 'age',
    'RIDRETH3': 'race',
    'DMDBORN4': 'born_us',
    'DMDEDUC2': 'edu_lvl',
    'RIDEXPRG': 'preg',
    'INDFMPIR': 'fam_inc_to_pov',
    'DIQ010': 'diabetes',
    'DIQ160': 'prediabetes',
    'DID040': 'age_of_diabetes',
    'PAD790Q': 'freq_mod_ex',
    'PAD790U': 'unit_mod_ex',
    'PAD800': 'avg_mins_mod_ex',
    'PAD810Q': 'freq_vig_ex',
    'PAD810U': 'unit_vig_ex',
    'PAD820': 'avg_mins_vig_ex',
    'PAD680': 'mins_seden_daily',
    'HEQ010': 'hep_b',})
```

```
for c in ['freq_mod_ex', 'freq_vig_ex', 'mins_seden_daily', 'avg_mins_mod_ex', 'avg_mins_vig
    nhanes_df[c] = (
        nhanes_df[c]
            .replace({7777: np.nan, 9999: np.nan})
            .round(3)
            .astype('Int64')
    )

for c in ['alc_times_yr', 'alc_binge_times_yr', 'priv_health_ins']:
    nhanes_df[c] = (
        nhanes_df[c]
            .replace({77: np.nan, 99: np.nan})
            .round(3)
            .astype('Int64')
    )

for c in ['alc_ever', 'edu_lvl', 'cig_smoker_ever', 'diabetes', 'prediabetes', 'hep_b', 'heal
    nhanes_df[c] = (
        nhanes_df[c]
            .replace({7: np.nan, 9: np.nan})
            .round(3)
            .astype('Int64')
    )

nhanes_df['age_of_diabetes'] = (
    nhanes_df['age_of_diabetes']
        .replace({777: np.nan, 999: np.nan})
        .round(3)
        .astype('Int64')
    )

for c in ['unit_mod_ex', 'unit_vig_ex']:
    nhanes_df[c] = (
        nhanes_df[c]
        .str.decode('utf-8', errors = 'ignore')
        .str.strip()
    )

ex_conversion = {
    "M": 1 / (30/7),
    "W": 1,
    "D": 7,
    "Y": 1 / 52,
    '': 1
}

nhanes_df['mod_ex_per_week'] = nhanes_df['freq_mod_ex'] * nhanes_df['unit_mod_ex'].map(ex_co

nhanes_df['vig_ex_per_week'] = nhanes_df['freq_vig_ex'] * nhanes_df['unit_vig_ex'].map(ex_co

nhanes_df = nhanes_df[nhanes_df['age'] >= 18].reset_index(drop = True)
nhanes_df = nhanes_df.dropna(subset=['med_cap']).reset_index(drop = True)

bins = [100, 238, 260, 290, np.inf]
labels = [0, 1, 2, 3]

nhanes_df['cap_cat'] = pd.cut(
    nhanes_df['med_cap'],
    bins = bins,
    labels = labels,
    right = False
)
```

```
print("Number of observations (rows):", nhanes_df.shape[0])
print("Number of variables (columns):", nhanes_df.shape[1])

print("\nColumn names:")
for col in nhanes_df.columns:
    print("-", col)
```