Variable Encoding

February 6, 2021

Objective:

Perform preliminary steps for structuring the dataset:

1. The dataset contains 58 variables and over 800k observation points. Remove unwanted variable to reduce the size of the data set. The variables that can be immediately removed were identified in the previous week. Others can be removed along the way as necessary.
2. Provide data encoding for some key variables, such as the survey year, meal name, and time of consumption.

[3]:
```
import pandas as pd

nhanes = pd.read_csv('../../Data/nhanes.csv')


#The following variables have been deemed irrelevant for this analysis, so they␣
 ↪are dropped.


nhanes = nhanes.drop(['DR1IVARA', 'DR1IVB12', 'DR1ICALC', 'DR1IIRON',␣
 ↪'DR1IZINC', 'DR1ISELE', 'DR1IP205',
            'DR1IP226', 'RIDRETH3', 'DR1I_PF_CUREDMEAT', 'DR1I_PF_ORGAN',␣
 ↪'DR1I_PF_POULT', 'DR1I_PF_MPS_TOTAL',
            'DR1I_PF_EGGS', 'DR1I_PF_NUTSDS', 'DR1I_PF_LEGUMES',␣
 ↪'DR1I_PF_TOTAL', 'DR1I_D_TOTAL',
            'DR1I_D_TOTAL', 'DR1I_D_MILK', 'DR1I_D_YOGURT', 'DR1I_D_CHEESE',␣
 ↪'WTDRD1_6YR'], axis=1)



#Map the survey year data, based on the SDDSRVYR encoding key

#Obtain description and value counts
nhanes['SDDSRVYR'].describe()
nhanes['SDDSRVYR'].value_counts()

#Create Survey Year variable based on lookup, mapping from CDC source
survey_year_lookup = {4: '2005-2006', 5: '2009-2010', 6: '2011-2012', 7:␣
 ↪'2013-2014', 8:'2015-2016', 9:'2017-2018'}
```

```
nhanes['Survey_Year'] = nhanes['SDDSRVYR'].map(survey_year_lookup)

#Check for NAs
print("Survey Year NA count is "+str(nhanes['Survey_Year'].isnull().sum()))


#Map the meal occasion data, based on the DR1.030Z encoding key

#Obtain description and value counts
nhanes['DR1.030Z'].describe()
nhanes['DR1.030Z'].value_counts()

#Create Survey Year variable based on lookup, mapping from CDC source
meal_name_lookup = {1: 'Breakfast', 2: 'Lunch', 3: 'Dinner', 4: 'Supper', 5:
 ↪'Brunch', 6:'Snack',
                    7: 'Drink', 8: 'Infant Feeding', 9: 'Extended consumption',␣
 ↪10: 'Desayano',
                    11: 'Almuerzo', 12: 'Comida', 13: 'Merienda', 14: 'Cena',␣
 ↪15: 'Enter comida',
                    16: 'Botana', 17: 'Bocadillo', 18: 'Tentempie', 19:␣
 ↪'Bebida', 91: 'Other'}


nhanes['Meal_Name'] = nhanes['DR1.030Z'].map(meal_name_lookup)

#Check for NAs
print("Meal Name NA count is "+str(nhanes['Meal_Name'].isnull().sum()))
```

```
Survey Year NA count is 0
Meal Name NA count is 0
```

[4]:
```
#Meal Name Counts - Observation Level
nhanes['Meal_Name'].value_counts()
```

[4]:
```
Dinner                  165082
Lunch                   161393
Breakfast               142660
Snack                   136295
Supper                   42739
Drink                    40487
Extended consumption     25242
Infant Feeding           18184
Cena                     18065
Desayano                 16198
Comida                   15428
Almuerzo                 13211
Merienda                  7026
```

```
Brunch                    6602
Bebida                    4958
Botana                    3291
Bocadillo                 2946
Enter comida              2842
Tentempie                  356
Other                        7
Name: Meal_Name, dtype: int64
```

[5]:
```
#Survey Name Counts - Observation Level
nhanes['Survey_Year'].value_counts()
```

[5]:
```
2011-2012    150991
2005-2006    146940
2009-2010    145703
2015-2016    131394
2013-2014    126503
2017-2018    121481
Name: Survey_Year, dtype: int64
```

Meal Time Variable

The time variable can be used for validity checks on meal name, and data grouping of each subject per name. Acccoring the CDC references, the time was collected in the HHMM format. An initial description shows that the time values are in seconds.

[6]:
```
nhanes['DR1.020'].describe()
```

[6]:
```
count    823012.000000
mean      69462.896823
std       17059.701708
min       18000.000000
25%       55800.000000
50%       68400.000000
75%       84600.000000
max      104340.000000
Name: DR1.020, dtype: float64
```

It seems like the data was collected on a 24 hr cycle starting at 5AM and finishing at 4:59AM the next day.

[8]:
```
#Find time minimum and convert seconds to hours
nhanes['DR1.020'].min()/60/60
```

[8]:
```
5.0
```

[9]:
```
#Find time maximum and convert seconds to hours
nhanes['DR1.020'].max()/60/60
```

[9]:
```
28.983333333333334
```

The code below removes the apparent 5AM time collection bias and creates a time variable in a pandas time format.

[10]:
```
#Create a time column, in a pandas time format

#Remove the 5AM bias from the value in seconds
def remove_time_bias(time_in):
    midnight = 24*60*60
    if (time_in >= midnight):
        time_post = time_in - midnight
    else: time_post = time_in
    return round(time_post)

#Create time variable and convert to time formatefrom DR1.020
nhanes['Time'] = nhanes['DR1.020'].apply(remove_time_bias)
nhanes['Time'] = nhanes['Time'].astype(int)
nhanes['Time'] = nhanes['Time'].round().apply(pd.to_timedelta, unit='s')
```

Conclusions

1. Preliminary removal of 23 variables that are identified as not required from previous week. More variables can be removed as the project progresses, to reduce dataset for more complex analysis tasks.
2. Three new variables added as encoders for the following:
   - Survey Year
   - Meal Name
   - Time Conversion: This was done after investigations about time format in the dataset. It seems apparent that the collected time uses a 24H starting at 5AM. So the 5AM time bias is removed and conversion to time format is performed. This operation is a bit time consuming and could be best performed once non-seafood observations are removed.