

WEEK 3 - Assignment 1.2

```
In [1]: # DSC530-T302
# Week 3
# 1.2 Programming Assignment
# Author: Aarti Ramani
# Created Date: 12/16/2022
# Purpose: Program to match the pregnancy numbers in NSFG pregnancy data and respon
# *****
# Change#:1 (Week 3)
# Change(s) Made: Version 1.0
# Date of Change: 12/16/2022
# Author: Aarti Ramani
# Change Approved by: N/A
# Date Moved to Production: N/A
# *****

import thinkstats2
import numpy as np
from collections import defaultdict

def MakePregMap(df):
    """Make a map from caseid to list of preg indices.

    df: DataFrame

    returns: dict that maps from caseid to list of indices into `preg`
    """
    d = defaultdict(list)
    for index, caseid in df.caseid.iteritems():
        d[caseid].append(index)
    return d

def CleanFemResp(df):
    """Recodes variables from the respondent frame.

    df: DataFrame
    """
    pass

def CleanFemPreg(df):
    """Recodes variables from the pregnancy frame.

    df: DataFrame
    """
    # mother's age is encoded in centiyears; convert to years
    df.agepreg /= 100.0

    # birthwgt_lb contains at least one bogus value (51 lbs)
    # replace with NaN
```

```

df.loc[df.birthwgt_lb > 20, 'birthwgt_lb'] = np.nan

# replace 'not ascertained', 'refused', 'don't know' with NaN
na_vals = [97, 98, 99]
df.birthwgt_lb.replace(na_vals, np.nan, inplace=True)
df.birthwgt_oz.replace(na_vals, np.nan, inplace=True)
df.hpagelb.replace(na_vals, np.nan, inplace=True)

df.babysex.replace([7, 9], np.nan, inplace=True)
df.nbrnaliv.replace([9], np.nan, inplace=True)

# birthweight is stored in two columns, lbs and oz.
# convert to a single column in lb
# NOTE: creating a new column requires dictionary syntax,
# not attribute assignment (like df.totalwgt_lb)
df['totalwgt_lb'] = df.birthwgt_lb + df.birthwgt_oz / 16.0

# due to a bug in ReadStataDct, the last variable gets clipped;
# so for now set it to NaN
df.cmintvw = np.nan

def ReadFemPreg(dct_file='2002FemPreg.dct',
               dat_file='2002FemPreg.dat.gz'):
    """Reads the NSFG pregnancy data.

    dct_file: string file name
    dat_file: string file name

    returns: DataFrame
    """
    dct = thinkstats2.ReadStataDct(dct_file)
    df = dct.ReadFixedWidth(dat_file, compression='gzip')
    CleanFemPreg(df)
    return df

def ReadFemResp(dct_file='2002FemResp.dct',
               dat_file='2002FemResp.dat.gz',
               nrows=None):
    """Reads the NSFG respondent data.

    dct_file: string file name
    dat_file: string file name

    returns: DataFrame
    """
    dct = thinkstats2.ReadStataDct(dct_file)
    df = dct.ReadFixedWidth(dat_file, compression='gzip', nrows=nrows)
    CleanFemResp(df)
    return df

def main():
    preg = ReadFemPreg()
    # print('TEST \n', preg[preg.caseid == 2298].caseid, ' + ', preg[preg.caseid ==

```

```

resp = ReadFemResp()
cnt = 0
# print('2nd \n', resp[resp.caseid == 2298].caseid, ' ', resp[resp.caseid == 22
# print(preg[preg.caseid == 2298])
# print(resp.caseid == 2298)
pregList = MakePregMap(preg)
# print(pregList)
# Match caseid in preg and resp to get pregnum. Preg num in preg df will be the
for index, pregnum in resp.pregnum.iteritems():
    # print('Index : ', index, ' -->', resp.caseid[index], '-->',
    #       resp.pregnum[index], '-->', pregList[resp.caseid[index]],
    #       '-->', len(pregList[resp.caseid[index]]), '-->', preg.pregnum[index]
    if len(pregList[resp.caseid[index]]) != pregnum:
        print('Values don\'t match for : ', 'CASE ID: ', resp.caseid[index], '
            , ' PREG NUM from PREG : ', len(pregList[resp.caseid[index]]))
        cnt = cnt+1
if cnt == 0:
    print('All matched!!')

if __name__ == '__main__':
    main()

```

All matched!!