**PAT 2: Class Exercise 4 – Python Practice**

1. Using the spreadsheets that we used in Exercise 2, import the xlsx files into Jupyter Notebooks.

# Package import

import pandas as pd

import numpy as np

from datetime import datetime as dt

# File Import

Comment = pd.read\_excel('Comment2.xlsx')

FallTable = pd.read\_excel('FallTable2.xlsx')

InterviewTable = pd.read\_excel('InterviewTable2.xlsx')

MedicalHistory = pd.read\_excel('MedicalHistory2.xlsx')

SurgicalHistory = pd.read\_excel('SurgicalHistory2.xlsx')

1. Sort each of them by SubjectID

Comment = Comment.sort\_values(by='SubjectID')

FallTable = FallTable.sort\_values(by='SubjectID')

InterviewTable = InterviewTable.sort\_values(by='SubjectID')

MedicalHistory = MedicalHistory.sort\_values(by='SubjectID')

SurgicalHistory = SurgicalHistory.sort\_values(by='SubjectID')

1. Merge all of them together in an **inner join** as a new dataframe

Table\_Inner = pd.merge(pd.merge(pd.merge(pd.merge(SurgicalHistory, FallTable, on = 'SubjectID', how = 'inner'),

MedicalHistory, on = 'SubjectID', how = 'inner'),

InterviewTable, on = 'SubjectID', how = 'inner'),

Comment, on='SubjectID', how = 'inner')

1. Calculate time SINCE the interview date (SIDateiv) - using your "today's date" (whatever it may be) – express the answer in months (assign this to a variable called timesinceint)

Table\_Inner['timesinceint'] = round(((dt.today() - Table\_Inner['SIDateiv'])).dt.days/30.437,2)

1. Subset this dataframe by selecting only cases where timesinceint is >= mean of the value of timesinceint across all participants (e.g. the 50% most recent participants)

Table\_Inner2 = Table\_Inner[Table\_Inner['timesinceint'] >= Table\_Inner['timesinceint'].mean()]

1. Create another subset of this dataframe from 5) selecting only:  **SubjectID**and these variables:

**SDOsteoDiag      SDRheumaDiag SDOADiag            SDLupusDiag      SDHyperthyDiag**

Table\_Inner3 = Table\_Inner2[['SubjectID', 'SDOsteoDiag', 'SDRheumaDiag', 'SDOADiag', 'SDLupusDiag', 'SDHyperthyDiag']]

1. Recode the yes and no's to 1's and 0's, respectively. Also recode missing data and Don't know to NaN (the actual missing data denotation in Python, not the string "NaN"). Challenge yourself and use functions and loops rather than doing one at a time.

#define function

def response(series):

if series == 'Yes':

return 1

elif series == 'No':

return 0

elif series == 'Don\'t Know' or series == 'missing data':

return np.nan

#apply function

columns = list(Table\_Inner3.columns[1:6])

for i in columns:

Table\_Inner3[i] = Table\_Inner3[i].apply(response)

1. Determine the frequencies of the response categories for these same variables that you've recoded

for i in columns:

counts[i] = pd.DataFrame(Table\_Inner3[i].value\_counts())

counts

counts

Out[281]:

|  | **SDHyperthyDiag** | **SDOsteoDiag** | **SDRheumaDiag** | **SDOADiag** | **SDLupusDiag** |
| --- | --- | --- | --- | --- | --- |
| **0.0** | 140 | 98 | 130 | 84 | 142.0 |
| **1.0** | 3 | 44 | 13 | 58 | NaN |