**Performance Assessment Tasks**

**PAT5: Assignment 3: Multi-centre study data harmonization and visualization – Python (20%)**

**Directions**

This assignment will test your practical knowledge and analysis skills developed through weeks 9 through 12. The assignment will introduce you to working with multi-centre study data, harmonizing variables, and give you practice visualizing trends in the data across sites.

The data source will be fictitious data from the Canadian Multicentre Osteoporosis Bone Quality Study. High resolution CT imaging data derived from different manufacturers will be provided.

Complete the following questions in Python

### Q1. Read and create a subset of the xtrfxnd dataframe by selecting only these variables of interest:

['id', 't\_ttbmd', 't\_ctbmd', 'r\_ttbmd', 'r\_ctbmd', 't\_moart', 'r\_moart', 'hicnum', 'centre', 'age', 'osteomed', 'newosteofx', 'prevfx\_any']  
  
Remember to turn all variables to lower case.

# import packages

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sb

# Reading in the Data

xtrfxnd = pd.read\_sas('xtrfxnd.sas7bdat')

# Changing columns to lowercase

xtrfxnd.columns = xtrfxnd.columns.str.lower()

# Sub-setting the Dataframe

coltosubset = ['id','t\_ttbmd','t\_ctbmd','r\_ttbmd','r\_ctbmd','t\_moart','r\_moart','hicnum','centre','age','osteomed','newosteofx','prevfx\_any']

xtrfxnd = xtrfxnd[coltosubset]

### Q2. Do a frequency count for 'hicnum' and 'centre' - these are the study sites

xtrfxnd[['hicnum','centre']].value\_counts()

### Q3. Create a new variable called 'sitename', which serves as a proper label for the study site names. Base this off the 2 variables: CENTRE and HICNUM (according to the key below).

### Run another frequency procedure on the new 'sitename' variable to check if you did this correctly.

1 or CA = "Calgary"  
4 or TO = "Toronto - Referent"  
5 or VR = "Vancouver"  
2 or HA = "Hamilton"  
3 or SK = "Saskatoon"

(If it fits none of these sites, just let it be missing)

# Creating easier references to specific hicnum strings

CA = xtrfxnd['hicnum'].value\_counts().index[0]

TO = xtrfxnd['hicnum'].value\_counts().index[1]

VR = xtrfxnd['hicnum'].value\_counts().index[2]

HA = xtrfxnd['hicnum'].value\_counts().index[3]

SK = xtrfxnd['hicnum'].value\_counts().index[4]

# Creating sitename column

xtrfxnd['sitename'] = np.where((xtrfxnd['hicnum'] == CA) | (xtrfxnd['centre'] == 1), 'Calgary',

np.where((xtrfxnd['hicnum'] == TO) | (xtrfxnd['centre'] == 4), 'Toronto - Referent',

np.where((xtrfxnd['hicnum'] == VR) | (xtrfxnd['centre'] == 5), 'Vancouver',

np.where((xtrfxnd['hicnum'] == HA) | (xtrfxnd['centre'] == 2), 'Hamilton',

np.where((xtrfxnd['hicnum'] == SK) | (xtrfxnd['centre'] == 3), 'Saskatoon',

np.nan)))))

# Checking value counts on sitename

xtrfxnd['sitename'].value\_counts()

Calgary 352

Toronto - Referent 292

Vancouver 225

Hamilton 200

Saskatoon 161

nan 1

Name: sitename, dtype: int64

### Q4. a) Create a boxplot of how t\_ttbmd varies by site

Add a title and axis labels - change font size to be bigger  
Also make the figure larger: 12 x 6  
Compare side by side those with and without a prevalent fracture (prevfx\_any)

# boxplot for t\_ttbmd variation by site

plt.figure(figsize=(12,6))

sb.boxplot(data=xtrfxnd, x='sitename', y='t\_ttbmd', hue='prevfx\_any', color='gray')

plt.title('Tibia Total Bone Mineral Density Across Sites With or Without a Prevalent Fracture', fontsize=16)

plt.xlabel('Clinical Site Name', fontsize=12)

plt.ylabel('Tibia Total Bone Mineral Density', fontsize=12)

plt.legend(title='Prevalent Fracture')

plt.show()

Chart, box and whisker chart

Description automatically generated

### Q4. b) Determine the distribution (frequencies) of motion grades (t\_moart and r\_moart)

### # Distribution of motion grades

### fig, (ax1, ax2) = plt.subplots(1,2)

### fig.set\_size\_inches(10,5)

### ax1.set\_title('Motion Grade Distribution of the Tibia')

### ax1.hist(data=xtrfxnd, x='t\_moart', bins=20, color='black')

### plt.setp(ax1, xlabel = 'Tibia Motion Grade')

### plt.setp(ax1, ylabel = 'Frequency')

### ax2.set\_title('Motion Grade Distribution of the Radius')

### ax2.hist(data=xtrfxnd, x='r\_moart', bins=20, color='grey')

### plt.setp(ax2, xlabel = 'Radius Motion Grade')

### plt.setp(ax2, ylabel = 'Frequency')

### fig.show()

A picture containing text

Description automatically generated

### Q4.c) Repeat the boxplot exercise from Q4 a) but using a subsetted dataframe looking only at motion artifact (t\_moart) grades 3 and above

### #Subsetting Dataframe

### xtrfxnd\_subset = xtrfxnd[xtrfxnd['t\_moart'] >= 3]

### # boxplot for t\_ttbmd variation by site for motion artifact of grade 3 and above

### plt.figure(figsize=(12,6))

### sb.boxplot(data=xtrfxnd\_subset, x='sitename', y='t\_ttbmd', hue='prevfx\_any', color='gray')

### plt.title('Tibia Total Bone Mineral Density Across Sites With or Without a Prevalent Fracture', fontsize=16)

### plt.xlabel('Clinical Site Name', fontsize=12)

### plt.ylabel('Tibia Total Bone Mineral Density', fontsize=12)

### plt.legend(title='Prevalent Fracture')

### plt.show()

Chart, box and whisker chart

Description automatically generated

### Q5. Create a scatterplot of how t\_ttbmd relates to age

Add a title and axis labels - change font size to be bigger.  
Also make the figure 12 x 6.  
Label the points by whether they had a new osteoporotic fracture or not (newosteofx)

plt.figure(figsize=(12,6))

sb.scatterplot(data=xtrfxnd, x='age', y='t\_ttbmd', hue='newosteofx', color='black')

plt.title('Comparing Tibia Total Bone Mineral Density with Age for Those With/Without An Osteoporotic Fracture', fontsize=14)

plt.ylabel('Tibia Total Bone Mineral Density', fontsize=12)

plt.xlabel('Age (years)', fontsize=12)

plt.legend(title='New Osteoporotic Fracture')

plt.show()

Chart, scatter chart

Description automatically generated

### Q6. Data harmonization

### When the study was completed, the CT scanners were cross-calibrated

* meaning an object was scanned at each of the CT scanners from every site. This was done to check for comparability and to allow any future adjustments.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Vancouver** |  | **Calgary** |  | **Saskatoon** |  |
| **Bone variable** | Slope (95%CI) | Intercept(95%CI) | Slope (95%CI) | Intercept(95%CI) | Slope (95%CI) | Intercept (95%CI) |
| **vBMDi(mg/cm3)**  **(t\_ttbmd or r\_ttbmd)** | 0.99 | -3.93 | 0.97 | 19.06 | 1.03 | -23.26 |
| **vBMDc (mg/cm3)**  **(t\_ctbmd or r\_ctbmd)** | 1.04 | -0.05 | 1.05 | -6.82 | 0.99 | -0.03 |

Toronto was the reference site. Hamilton used the same scanner as in Toronto.

All other sites' bone values had to be adjusted according to Toronto's values. To do this, slopes and intercepts represent **multipliers** and **additions** of values needed, respectively. (note, some are subtraction not addition: those with negative signs).

For example, all bone density values in Vancouver need to first be multiplied by 0.99 then - 3.93.

#### **Q6. a) Create new cross-calibrated values for all sites' 't\_ttbmd', 't\_ctbmd', 'r\_ttbmd', 'r\_ctbmd' values according to this table.**

#### NOTE: Use the vBMDi row values for ttbmd and vBMDc row values for ctbmd

#### Toronto and Hamilton should just take on its original value, unchanged.

As an example: if site is Vancouver, then: t\_ttbmd\_new = 0.99\*t\_ttbmd - 3.93

# Create calibration dataframe

sites = ['Toronto - Referent','Hamilton','Vancouver','Calgary','Saskatoon']

columns = ['Slope1','Intercept1','Slope2','Intercept2']

values = [1.00,0,1.00,0],[1.00,0,1.00,0],[0.99,-3.93,1.04,-0.05],[0.97,19.06,1.05,-6.82],[1.03,-23.26,0.99,-0.03]

calib = pd.DataFrame(values)

calib.columns = columns

calib.set\_axis(sites).rename\_axis('sitename')

#Merge with original dataframe and set new columns

xtrfxnd\_cal = xtrfxnd.merge(calib, on='sitename')

xtrfxnd\_cal['t\_ttbmd\_new'] = xtrfxnd\_cal['t\_ttbmd']\*xtrfxnd\_cal['Slope1'] - xtrfxnd\_cal['Intercept1']

xtrfxnd\_cal['r\_ttbmd\_new'] = xtrfxnd\_cal['r\_ttbmd']\*xtrfxnd\_cal['Slope1'] - xtrfxnd\_cal['Intercept1']

xtrfxnd\_cal['t\_ctbmd\_new'] = xtrfxnd\_cal['t\_ctbmd']\*xtrfxnd\_cal['Slope2'] - xtrfxnd\_cal['Intercept2']

xtrfxnd\_cal['r\_ctbmd\_new'] = xtrfxnd\_cal['r\_ctbmd']\*xtrfxnd\_cal['Slope2'] - xtrfxnd\_cal['Intercept2']

### Q6. b) Create a 1 x 2 subplot containing 2 boxplots: 1 for before and 1 after calibration

### # Transposing columns required

### xtrfxndmelt = pd.melt(xtrfxnd, id\_vars='sitename', value\_vars=['t\_ttbmd','r\_ttbmd','t\_ctbmd','r\_ctbmd'])

### xtrfxnd\_calmelt = pd.melt(xtrfxnd\_cal, id\_vars='sitename', value\_vars=['t\_ttbmd\_new','r\_ttbmd\_new','t\_ctbmd\_new','r\_ctbmd\_new'])

### # Plotting

### fig, axes = plt.subplots(1,2, figsize=(18,8), sharey=True)

### fig.suptitle('Comparing Bone Density Measurements Before and After Calibration', fontsize=14)

### sb.boxplot(ax=axes[0], data=xtrfxndmelt, x='variable', y='value', hue='sitename')

### sb.boxplot(ax=axes[1], data=xtrfxnd\_calmelt, x='variable', y='value', hue='sitename')

### axes[0].set\_title('Before Calibration')

### axes[1].set\_title('After Calibration')

### plt.setp(axes[0], ylabel='Bone Density')

### plt.setp(axes[1], ylabel='Bone Density')

### plt.setp(axes[0], xlabel='Bone Density Parameter')

### plt.setp(axes[1], xlabel='Bone Density Parameter')

### plt.show()

Chart, box and whisker chart

Description automatically generated

**Target Course Competencies**

This exercise targets course competencies 1, 2 and 3.

**Rating Scale**

|  |  |
| --- | --- |
| **Value** | **Description** |
| **3** | **Fully meets or exceeds the outcome** |
| **2** | **Meets most of the outcome** |
| **1** | **Meets some of the outcome** |
| **0** | **Does not meet the outcome** |

**Scoring Standard**

This assignment is designed to give the student experience working with clinical data, harmonizing multi-centre study variables, and plotting data using different styles of graphs. The questions in this assignment help prepare them to assist investigators with data presentation and transformations, while considering inconsistencies across sites. They are designed to detail specific aspects of variables, data collection, and procedures to enable comparability.

A score of 60% on this assignment is required.

**Scoring Guide**

|  |  |  |
| --- | --- | --- |
| **Competency** | **Criteria** | **Ratings** |
| Q1. | Ability to subset a dataframe | 3 2 1 0 |
| Q2. | Ability to complete a frequency check | 3 2 1 0 |
| Q3. | Ability to recode a variable | 3 2 1 0 |
| Q4.a | Ability to create a basic boxplot with pairwise comparisons | 3 2 1 0 |
| Q4.b | Ability to complete a frequency check | 3 2 1 0 |
| Q4.c | Ability to create customized boxplots and apply multiple Boolean operators | 3 2 1 0 |
| Q5. | Ability to generate a basic scatterplot with labeled dots | 3 2 1 0 |
| Q6.a | Ability to transform variables | 3 2 1 0 |
| Q6.b | Ability to create side-by-side subplots | 3 2 1 0 |
|  |  |  |
|  |  | / 27 |

**Deadline**

December 4th 11:59 pm.