

```
[1]: # -*- coding: utf-8 -*-
"""
Created on Sun Sep 18 10:35:56 2022

@author: Brandon Botzer - btb5103
"""

"""
Recall the datasets you used for SWENG 545 term project. (I do not as I have never taken that class... great)
All typos and inconsistency in course names have been cleaned. This time you will use Python to perform data ur

Perform the following actions:

1. Upload Registration.csv Download Registration.csv and Course_info.xlsx Download Course_info.xlsx into Pandas. (5 points)
2. Explore and clean Registration data. (30 points)
3. Explore and clean Course info data. (10 points)
4. Which course has the highest registration? (15 points)
5. Inner join two datasets. (20 points)
6. Create a data frame with student name as the index, course numbers as columns, and if the student registered a course as values(0, 1). ( 20 points)

"""

print("""
Perform the following actions:

1. Upload Registration.csv Download Registration.csv and Course_info.xlsx Download Course_info.xlsx into Pandas. (5 points)
2. Explore and clean Registration data. (30 points)
3. Explore and clean Course info data. (10 points)
4. Which course has the highest registration? (15 points)
5. Inner join two datasets. (20 points)
6. Create a data frame with student name as the index, course numbers as columns, and if the student registered a course as values(0, 1). ( 20 points)

""")

#Imports (may not need all of these but better safe than sorry later)
import os
from pandas import Series, DataFrame
import pandas as pd
import numpy as np
import csv
from numpy import NaN as NA

#regular expressions
import re
```

```
In [9]: #1. Upload Registration.csv Download Registration.csv and Course_info.xlsx Download Course_info.xlsx into Pandas. (5 points)
print("""\n\n1. Upload Registration.csv Download Registration.csv and Course_info.xlsx Download Course_info.xlsx into Pandas. (5 points)
""")

#Set the readpath
readPath = "J:\DSDegree\PennState\DAAN_862\Week 5\Homework"

#Change the directory
os.chdir(readPath)

#Read in the registration table
reglist = pd.read_csv("Registration.csv")

#Read in the course info
#I turned this into a CSV as the 'openpyxl' dependency was
#giving me problems when I ran this on different machines
#It was a pathing issue...
clist = pd.read_excel('Course_info.xlsx')
#courselist = pd.read_csv('Course_info.csv')

print("\n\nDOWNLOADS ACCOMPLISHED!\n\nRegistration List:\n")
print(reglist.describe())
print("\n\nCourse List:\n")
print(clist.describe())
```

```
1. Upload Registration.csv Download Registration.csv and Course_info.xlsx Download Course_info.xlsx into Pandas. (5 points)
DOWNLOADS ACCOMPLISHED!

Registration List:
```

	Student name	semester	new	course name
count	4900	4900	4899	
unique	448	16	168	
top	Ed McMahon	Spring 2002	COMPUT LINEAR ALGEBRA	
freq	52	486	411	

```
Course List:
```

	Course number	Course Name	Course Type
count	42	41	42
unique	42	40	3
top	ARTS565	FRANCE & THE EUROP. UNION	E
freq	1	2	33

```
In [10]: #2. Explore and clean Registration data. (30 points)
print("""\n\n2. Explore and clean Registration data. (30 points)\n""")
```

```
#Rename the data frame columns
reglist.columns = ['student_name', 'semester', 'course_name']

#convert all course names to upper and strip possible edge whitespace
reglist['course_name'] = reglist['course_name'].str.upper()
reglist['course_name'] = reglist['course_name'].str.strip()

#Sort the students alphabetically
reglist = reglist.sort_values('student_name')

#Drop duplicates within the data frame
reglist = reglist.drop_duplicates()

#Reindex and drop the 'index' column
reglist = reglist.reset_index(drop = True)

print("""The Registration Data has been explored and cleaned. \nNote: We are currently not cleaning up the student names. This would need to be done using REGEX listings.
""")
print(reglist)
print("\n\nCleaned Registration List:\n")
print(reglist.describe())
```

```
2. Explore and clean Registration data. (30 points)
```

The Registration Data has been explored and cleaned.

Note: We are currently not cleaning up the student names. This would need to be done using REGEX listings.

	student_name	semester	course_name
0	Abella Abzug	Spring 2001	ART: ANCIENT TO 1945
1	Abella Abzug	Fall 2004	EXPERIMENTAL WRITING SEM
2	Abella Abzug	Fall 2003	A WORLD AT WAR
3	Abella Abzug	Spring 2002	CONTEMPORARY AFRICAN ART
4	Abella Abzug	Spring 2002	20TH CENTURY RUSSIAN LITERATURE: FICTION AND R...
...
3646	state representative	Spring 2002	ANALYTICAL MECHANICS
3647	state representative	Spring 2004	COMMUNICATIONS INTERNSHIP
3648	state representative	Spring 2003	CONTEMPORARY AFRICAN ART
3649	state representative	Fall 2003	A WORLD AT WAR
3650	state representative	Spring 2004	CELL AND BIO AND BIOCHEMISTRY

[3651 rows x 3 columns]

Cleaned Registration List:

	student_name	semester	course_name
count	3651	3651	3650
unique	448	16	168
top	LCheryl Ladd	Spring 2002	COMPUT LINEAR ALGEBRA
freq	25	334	303

```
In [4]: #3. Explore and clean Course_info data. (10 points)
print("""\n\n3. Explore and clean Course_info data. (10 points)\n""")
```

```
#strip the spaces off the course number
clist['Course number'] = clist['Course number'].str.strip()

#sort the list by the course number in ascending order
clist = clist.sort_values('Course number')

#Drop the NaN 'unlisted course'
clist = clist.dropna(axis = 0)

#reset the list index (drop the 'index' column)
clist = clist.reset_index(drop = True)

#Rename the data frame columns (note to self, do this first next time...)
clist.rename(columns = {'Course number':'course_number',
                        'Course Name ':'course_name',
                        'Course Type':'course_type'}, inplace = True)

#Convert all course names to uppercase and strip possible edge whitespace if it is there...
clist['course_name'] = clist['course_name'].str.upper()
clist['course_name'] = clist['course_name'].str.strip()

print("""The Course Info data has been explored and cleaned. We removed the non-existent course listing.""")
print(clist)
```

```
3. Explore and clean Course_info data. (10 points)
```

The Course Info data has been explored and cleaned. We removed the non-existent course listing.

	course_number	course_name
0	ARTS400	EXPERIMENTAL WRITING SEM: THE ECOLOGY OF POETRY
1	ARTS401	ART: ANCIENT TO 1945
2	ARTS403	ENVIRONMENTAL SYSTEMS II
3	ARTS484	EUROPE IN A WIDER WORLD
4	ARTS485	EVIDENCED BASED CRIME AND JUSTICE POLICY
5	ARTS486	COMPUTER LINEAR ALGEBRA
6	ARTS488	DEVIL'S PACT LIT/FILM
7	ARTS491	CONTEMPORARY POL.THUGHT
8	ARTS492	AFRICAN-AMERICAN LIT: AFRICAN-AMER LIT:CHANGE
9	ARTS493	AMERICAN HEALTH POLICY
10	ARTS494	BUSINESS GERMAN: A MICRO PERSPECTIVE
11	ARTS495	COMM AND THE PRESIDENCY
12	ARTS496	FRENCH THOUGHT TILL 1945
13	ARTS497	CONTEMP ART - 1945 TO PRESENT
14	ARTS512	ANALYTICAL MECHANICS
15	ARTS514	A WORLD AT WAR
16	ARTS516	BEHAVIORAL PHARMACOLOGY
17	ARTS518	CONTEMPORARY AFRICAN ART
18	ARTS520	FOOD/FEAST ARCH OF TABLE
19	ARTS541	AMERICAN SOCIAL POLICY
20	ARTS543	ART AND RELIGION
21	ARTS545	20TH CENTURY RUSSIAN LITERATURE: FICTION AND R...
22	ARTS547	COMMUNICATIONS INTERNSHP
23	ARTS549	FRESHWATER ECOLOGY
24	ARTS551	ENVIRONMENTAL AESTHETICS
25	ARTS553	FRENCH THOUGHT SINCE 1945
26	ARTS555	BECOMING HUMAN
27	ARTS557	19TH-CENTURY BRITISH LITERATURE
28	ARTS559	AMERICAN SOUTH 1861-PRES
29	ARTS561	AUGUSTAN CULTURAL REVOLUTION
30	ARTS565	ENVIRONMENTAL STUDIES RESEARCH SEMINAR JUNIOR ...
31	ARTS569	CELL, BIOL. & BIOCHEM.
32	ARTS571	FRANCE & THE EUROP. UNION
33	ARTS573	ANALYZING THE POL WORLD
34	ARTS575	EARLY MESOPOTAM HISTORY/SOCIETY
35	ARTS577	FRANCE & THE EUROP. UNION
36	ARTS579	EARLY BALCAN HIST/SOC
37	ARTS581	COMPARATIVE POLITICS
38	ARTS583	BRITISH POETRY 1660-1914
39	ARTS585	CONTEMPORARY SOCIO THEORY
40	ARTS587	ELEMENTARY ARABIC II

	course_type
0	C
1	C
2	F
3	F
4	E
5	F
6	E
7	E
8	E
9	E
10	E
11	E
12	E
13	E
14	F
15	F
16	F
17	F
18	E
19	E
20	E
21	E
22	E
23	E
24	E
25	E
26	E
27	E
28	E
29	E
30	E
31	E
32	E
33	E
34	E
35	E
36	E
37	E
38	E
39	E
40	E

```
In [5]: #4. Which course has the highest registration? (15 points)
print("""\n\n4. Which course has the highest registration? (15 points)\n""")
```

```
#Gather all of the courses
courses = clist.course_name.unique()

#Create an empty array to store the unique courses counts
courseCounts = np.zeros(len(courses))

#Count the occurrences of each unique course
for i in range(0, len(courses)):
    courseCounts[i] = np.count_nonzero(reglist.course_name == courses[i])

#Create Data frame of courses and the counts
courseData = pd.DataFrame({"courses":courses,
                           "courseCounts":courseCounts})

#Note to self: I tried to do this by creating the DF first
#and then iterating over each name in courses
def for name in courses:
    #courseData.course_counts = np.count_nonzero(reglist.coursename == name)
#but I had an issue trying to put the count into the correct column location
#Counting and then assigning the DF proved to be the easier method but
#it perplexes me I do not know how to do this from the DF itself.

#Get the largest count id location and pass it to the courses to find the course name
popularCourse = courseData.courses[courseData.course_counts.idxmax()]

print("""The most popular course is: " + str(popularCourse) + ".")
print("""It is taken by " + str(int(courseData.course_counts.max())) + " students.\n""")
```

```
4. Which course has the highest registration? (15 points)
```

The most popular course is: A WORLD AT WAR.

It is taken by 269 students.

```
In [6]: #5. Inner join two datasets. (20 points)
print("""\n\n5. Inner join two datasets. (20 points)\n""")
```

```
#inner join the data sets (merge does inner by default)
joinData = pd.merge(reglist, clist)

#sort by student name
joinData = joinData.sort_values('student_name')

#reindex
joinData = joinData.reset_index(drop = True)

print("""The two data frames have been inner joined.\n""")
print(joinData)
```

```
5. Inner join two datasets. (20 points)
```

The two data frames have been inner joined.

	student name	semester	course_name
0	Abella Abzug	Fall 2003	ARTS514
1	Abella Abzug	Spring 2002	ARTS545
2	Abella Abzug	Fall 2004	ARTS512
3	Abella Abzug	Spring 2002	ARTS518
4	Abella Abzug	Fall 2005	ARTS581
...
1745	state representative	Spring 2001	ARTS520
1746	state representative	Fall 2003	ARTS514
1747	state representative	Fall 2002	ARTS493
1748	state representative	Fall 2004	ARTS401
1749	state representative	Spring 2001	ARTS516

	course_type
0	F
1	E
2	F
3	F
4	E
...	...
1745	F
1746	F
1747	E
1748	C
1749	F

[1750 rows x 5 columns]

```
In [7]: #6. Create a data frame with student name as the index, course numbers as columns, and if the student registered a course as values(0, 1). ( 20 points)
```

Here is the final data frame for which students are taking which classes.

course_number	ARTS400	ARTS401	ARTS465	ARTS484	ARTS485	ARTS486	
student_name							
Abella Abzug	0	0	0	0	0	0	0
Al Gore	0	0	0	0	0	0	0
Al Hirt	0	0	1	0	0	0	1
Al Roker	1	0	0	0	0	0	0
Alan Bates	0	0	0	0	0	0	0
...
Winona Ryder	0	0	0	0	0	0	0
Wolfgang Puck	0	0	0	0	0	0	0
Yogi Berra	0	0	0	0	0	0	0
Yoko Ono	0	0	0	0	0	0	0
state representative	0	1	0	0	0	0	0

course_number	ARTS488	ARTS491	ARTS492	ARTS493	...	ARTS555	
student_name					...		
Abella Abzug	0	0	0	0	...	0	0
Al Gore	0	0	0	0	...	0	0
Al Hirt	0	0	0	0	...	0	0
Al Roker	0	0	0	0	...	0	0
Alan Bates	0	0	0	0	...	1	0
...
Winona Ryder	0	0	1	0	0	0	0
Wolfgang Puck	0	0	0	0	0	0	0
Yogi Berra	0	0	0	0	1	...	0
Yoko Ono	0	0	0	0	0	...	0
state representative	0	0	0	1	...	0	0

course_number	ARTS559	ARTS565	ARTS569	ARTS571	ARTS573	ARTS577	
student_name							
Abella Abzug	0	0	0	0	0	0	0
Al Gore	0	0	0	0	0	0	0
Al Hirt	0	0	0	0	0	0	0
Al Roker	0	0	0	0	0	0	0
Alan Bates	0	0	0	0	0	0	0
...
Winona Ryder	0	0	1	0	0	0	0
Wolfgang Puck	0	0	0	0	0	0	0
Yogi Berra	0	0	0	0	0	0	0
Yoko Ono	0	0	0	0	0	0	0
state representative	0	0	0	0	0	0	0

course_number	ARTS581	ARTS583	ARTS587	
student_name				
Abella Abzug	1	0	0	
Al Gore	0	0	0	
Al Hirt	0	0	0	
Al Roker	0	0	0	
Alan Bates	0	0	0	
...	
Winona Ryder	0	0	0	
Wolfgang Puck	0	0	1	
Yogi Berra	0	0	0	
Yoko Ono	0	0	0	
state representative	0	0	0	

[408 rows x 34 columns]