Dylan Francis

DAAN 682
15FEB2026

<u>Homework_5:</u>

Code:

```
#!/usr/bin/env python
# coding: utf-8

# **DAAN: 682:** Data Analytics Programming in Python\
# **Author:** Dylan Francis\
# **Title: Homework_5:** Data Wranggling\
# **Due Date:** 15FEB2026

# ### 1.) Load Registration.csv Download Registration.csvand Course_info.xlsx Download Course_info.xlsxinto Pandas DataFrames.

# In[55]:


import os
import pandas as pd
import re
from rapidfuzz import process


path = r"C:\Users\dylan\OneDrive\Documents\GRAD_SCHOOL\DAAN_682\HOMEWORK_5"
os.chdir(path)
registration = pd.read_csv("Registration.csv")
course_info = pd.read_excel("Course_info.xlsx")

#print(registration)
#print(course_info)


# ### 2.) Explore and clean Registration data: check for duplicates, missing values, inconsistent formatting, and data type issues. Document all cleaning steps performed.

# In[56]:


#handling missing values by dropping the rows with missing data
print(f"There is missing data in the dataframe which is found:\n {registration.isnull().sum()}")
registration_formatted = registration.dropna()
print(f"These values can be dropped, by dropping the whole row, as seen below:\n {registration_formatted}")
```

```
#handling duplicate data by dropping duplicates
registration_formatted = registration_formatted.drop_duplicates()
print(registration_formatted)


# In[57]:


#handling inconsistent formatting. First fix the column header and then the actual values
registration_formatted.columns = registration_formatted.columns.str.strip() # strip off white spaces
registration_formatted.columns = registration_formatted.columns.str.title() # apply title which capitalizes first letter, lowercases everything else
registration_formatted.columns = registration_formatted.columns.str.replace(" ", "_") #replaces space with underscore

for col in registration_formatted.select_dtypes(include="object").columns:
    registration_formatted[col] = registration_formatted[col].str.strip() # strip off white spaces
    registration_formatted[col] = registration_formatted[col].str.title() # apply title which capitalizes first letter, lowercases everything else
    registration_formatted[col] = registration_formatted[col].str.replace(" ", "_") #replaces space with underscore
    registration_formatted[col] = registration_formatted[col].str.replace(r"[^\w]", "", regex=True) #replaces special characters with "", only keeps letters, numbers, and underscores

print(registration_formatted)


#handling data type issues


# In[58]:


#handling data type issues
registration_formatted.info()
#all of the data types are objects!


# ### 3.) Explore and clean Course info data: check for duplicates, missing values, inconsistent formatting, and data type issues. Document all cleaning steps performed

# In[59]:
```

```
#handling missing values by dropping the rows with missing data
print(f"There is missing data in the dataframe which is found:\n {course_info.isnull().sum()}")
course_info_formatted = course_info.dropna().copy()

course_info_formatted.columns = course_info_formatted.columns.str.strip() # strip off white spaces
course_info_formatted.columns = course_info_formatted.columns.str.title() # apply title which capitalizes first letter, lowercases everything else
course_info_formatted.columns = course_info_formatted.columns.str.replace(" ", "_") # replaces space with underscore

for col in course_info_formatted.select_dtypes(include="object").columns:
    course_info_formatted[col] = course_info_formatted[col].str.strip() # strip off white spaces
    course_info_formatted[col] = course_info_formatted[col].str.title() # apply title which capitalizes first letter, lowercases everything else
    course_info_formatted[col] = course_info_formatted[col].str.replace(" ", "_") # replaces space with underscore
    course_info_formatted[col] = course_info_formatted[col].str.replace(r"[^\w]", "", regex=True) # replaces special characters with "", only keeps letters, numbers, and underscores

print(course_info_formatted)

course_info_formatted.info()


# ### 4.) Which course has the highest registration?

# In[60]:


print(f"The course with the most number of students registered is: {registration_formatted["Coursename"].value_counts().idxmax()}")


# ### 5.) Propose a solution to mitigate the data inconsistency (e.g., naming inconsistency) existing in the two datasets without manually correcting and matching the course names across the datasets. Perform an inner join on the two datasets using the appropriate key column(s).

# In[61]:
```

```
#registration_formatted = registration_formatted.rename(columns={"Coursename":
"Course_Name"})

course_list = course_info_formatted["Course_Name"].unique()

def match_course(name):
    match, score, _ = process.extractOne(name, course_list)
    if score >=75:
        return match
    return None

registration_formatted["matched_course"] =
registration_formatted["Coursename"].apply(match_course)

merged_data = registration_formatted.merge(course_info_formatted,
left_on="matched_course", right_on="Course_Name",how="inner")
print(merged_data)
merged_data.to_excel("merged_data.xlsx", sheet_name="raw_merged_data", index=False)


# ### 6.) Create a pivot table (DataFrame) with student names as the index, course numbers as
columns, and binary values (0 or 1) indicating whether each student registered for each course

# In[62]:


pivot_merged_data = (
    merged_data
        .assign(value=1)
        .pivot_table(
            index="Student_Name",
            columns="Course_Number",
            values="value",
            aggfunc="max",      # ensures 1 if registered
            fill_value=0
        )
)
print(pivot_merged_data)
with pd.ExcelWriter("merged_data.xlsx", mode="a", engine="openpyxl") as writer:
    pivot_merged_data.to_excel(writer, sheet_name="pivot_table", index=True)



#
```

Dylan Francis DAAN 682
15FEB2026

Output from the Terminal

```
%runfile
C:/Users/dylan/OneDrive/Documents/GRAD_SCHOOL/DAAN_682/HOMEWORK_4/Homework_
4.py --wdir
The has: 200 rows, and 6 columns
Here is the number of empty cells for each column:
 one        5
two         3
three       1
four        6
five        4
variable    0
dtype: int64
The toal number of empty cells is: 19
     one   two  three  four  five  A1  A2  B1  B2
0   -92.0 -76.0  -33.0   3.0 -13.0   0   0   0   1
1   -21.0  76.0   38.0  -6.0  80.0   0   0   1   0
2    -2.0 -47.0  -34.0 -86.0 -66.0   1   0   0   0
3   -76.0  43.0    7.0 -40.0 -42.0   1   0   0   0
4    44.0  37.0   -7.0 -14.0  30.0   1   0   0   0
..    ...   ...    ...   ...   ...  ..  ..  ..  ..
195  63.0   3.0  -30.0 -24.0 -59.0   1   0   0   0
196  97.0 -48.0  -61.0 -25.0 -21.0   0   0   1   0
197 -93.0 -75.0  -18.0 -67.0 -58.0   0   0   1   0
198  54.0 -66.0  -80.0  92.0  62.0   1   0   0   0
199  82.0  53.0  -77.0  79.0  97.0   0   0   0   1

[200 rows x 9 columns]
     one   two  three  four  five  A1  A2  B1  B2 one_bin
0   -92.0 -76.0  -33.0   3.0 -13.0   0   0   0   1     low
1   -21.0  76.0   38.0  -6.0  80.0   0   0   1   0     med
2    -2.0 -47.0  -34.0 -86.0 -66.0   1   0   0   0     med
3   -76.0  43.0    7.0 -40.0 -42.0   1   0   0   0     low
4    44.0  37.0   -7.0 -14.0  30.0   1   0   0   0    high
..    ...   ...    ...   ...   ...  ..  ..  ..  ..     ...
195  63.0   3.0  -30.0 -24.0 -59.0   1   0   0   0    high
196  97.0 -48.0  -61.0 -25.0 -21.0   0   0   1   0    high
197 -93.0 -75.0  -18.0 -67.0 -58.0   0   0   1   0     low
198  54.0 -66.0  -80.0  92.0  62.0   1   0   0   0    high
```

```
199  82.0  53.0  -77.0  79.0  97.0   0   0   0   1    high

[200 rows x 10 columns]
     one   two  three  four  five  A1  A2  B1  B2
0     low -76.0  -33.0   3.0 -13.0   0   0   0   1
1     med  76.0   38.0  -6.0  80.0   0   0   1   0
2     med -47.0  -34.0 -86.0 -66.0   1   0   0   0
3     low  43.0    7.0 -40.0 -42.0   1   0   0   0
4    high  37.0   -7.0 -14.0  30.0   1   0   0   0
..    ...   ...    ...   ...   ...  ..  ..  ..  ..
195  high   3.0  -30.0 -24.0 -59.0   1   0   0   0
196  high -48.0  -61.0 -25.0 -21.0   0   0   1   0
197   low -75.0  -18.0 -67.0 -58.0   0   0   1   0
198  high -66.0  -80.0  92.0  62.0   1   0   0   0
199  high  53.0  -77.0  79.0  97.0   0   0   0   1

[200 rows x 9 columns]
The number of unique words in the text is: 109
The word thst appears the most is: 'the' with a quantity of: 14.
The number of words that start with the letter t is: 22

%runfile C:/Users/dylan/Downloads/Homework_5.py --wdir
There is missing data in the dataframe which is found:
 Student name    0
semester new    0
coursename      1
dtype: int64
These values can be dropped, by dropping the whole row, as seen below:
     Student name semester new                                coursename
0       Bill Mumy    Fall 2004                  BEHAVIORAL PHARMACOLOGY
1       Bill Mumy    Fall 2000                  AMERICAN FOREIGN POLICY
2       Bill Mumy    Fall 2003                    DRUGS, BRAIN AND MIND
3       Bill Mumy    Fall 2005               Environmental Case Studies
4       Bill Mumy    Fall 2000                  COMPUTER LINEAR ALGEBRA
          ...          ...                                      ...
4895  Stacy Keach  Summer 2001                   CELL. BIOL. And BIOCHEM.
4896  Ann Landers  Summer 2004                      AMERICAN HEALT POLICY
4897  Ann Landers  Summer 2004                       ANALYTICAL MECHANICS
4898    Tyne Daly  Summer 2004                     COMPUT LINEAR ALGEBRA
4899    Tyne Daly  Summer 2004  EXPERIMENTAL WRITING SEM: The Ecology of Poetry

[4899 rows x 3 columns]
    Student name semester new                                coursename
0       Bill Mumy    Fall 2004                  BEHAVIORAL PHARMACOLOGY
```

```
1      Bill Mumy    Fall 2000                   AMERICAN FOREIGN POLICY
2      Bill Mumy    Fall 2003                     DRUGS, BRAIN AND MIND
3      Bill Mumy    Fall 2005                 Environmental Case Studies
4      Bill Mumy    Fall 2000                   COMPUTER LINEAR ALGEBRA
        ...         ...                          ...
4895  Stacy Keach  Summer 2001                    CELL. BIOL. And BIOCHEM.
4896  Ann Landers  Summer 2004                      AMERICAN HEALT POLICY
4897  Ann Landers  Summer 2004                       ANALYTICAL MECHANICS
4898    Tyne Daly  Summer 2004                     COMPUT LINEAR ALGEBRA
4899    Tyne Daly  Summer 2004  EXPERIMENTAL WRITING SEM: The Ecology of Poetry

[3650 rows x 3 columns]
    Student_Name Semester_New                              Coursename
0      Bill_Mumy    Fall_2004                 Behavioral_Pharmacology
1      Bill_Mumy    Fall_2000                 American_Foreign_Policy
2      Bill_Mumy    Fall_2003                    Drugs_Brain_And_Mind
3      Bill_Mumy    Fall_2005              Environmental_Case_Studies
4      Bill_Mumy    Fall_2000                 Computer_Linear_Algebra
        ...         ...                          ...
4895  Stacy_Keach  Summer_2001                    Cell_Biol_And_Biochem
4896  Ann_Landers  Summer_2004                     American_Healt_Policy
4897  Ann_Landers  Summer_2004                      Analytical_Mechanics
4898    Tyne_Daly  Summer_2004                    Comput_Linear_Algebra
4899    Tyne_Daly  Summer_2004  Experimental_Writing_Sem_The_Ecology_Of_Poetry

[3650 rows x 3 columns]
<class 'pandas.core.frame.DataFrame'>
Index: 3650 entries, 0 to 4899
Data columns (total 3 columns):
 #   Column        Non-Null Count  Dtype 
---  ------        --------------  ----- 
 0   Student_Name  3650 non-null   object
 1   Semester_New  3650 non-null   object
 2   Coursename    3650 non-null   object
dtypes: object(3)
memory usage: 114.1+ KB
There is missing data in the dataframe which is found:
 Course number    0
Course Name      1
Course Type      0
dtype: int64
  Course_Number                                     Course_Name Course_Type
0       Arts400  Experimental_Writing_Sem_The_Ecology_Of_Poetry           C
1       Arts401                             Art_Ancient_To_1945           C
```

```
2       Arts465                     Environmental_Systems_Ii          F
3       Arts486                      Computer_Linear_Algebra          F
4       Arts512                         Analytical_Mechanics          F
5       Arts514                              A_World_At_War          F
6       Arts516                      Behavioral_Pharmacology          F
7       Arts518                     Contemporary_African_Art          F
8       Arts520                      FoodFeast_Arch_Of_Table          F
9       Arts488                          DevilS_Pact_LitFilm          E
10      Arts541                      American_Social_Policy          E
11      Arts543                            Art_And_Religion          E
12      Arts491                      Contemporary_PolThought          E
13      Arts492        AfricanAmerican_Lit_AfricanAmer_LitChange          E
14      Arts493                      American_Health_Policy          E
15      Arts494              Business_German_A_Micro_Perspective          E
16      Arts495                    Comm_And__The_Presidency          E
17      Arts496                      French_Thought_Till_1945          E
18      Arts497                 Contemp_Art__1945_To_Present          E
19      Arts545  20Th_Century_Russian_Literature_Fiction_And_Re...          E
20      Arts547                     Communications_Internshp          E
21      Arts549                           Freshwater_Ecology          E
22      Arts551                                   Aesthetics          E
23      Arts553                     French_Thought_Since_1945          E
24      Arts555                              Becoming_Human          E
25      Arts485          Evidenced_Based_Crime_And_Justice_Policy          E
26      Arts484                       Europe_In_A_Wider_World          E
27      Arts557                 19ThCentury_British_Literature          E
28      Arts559                       American_South_1861Pres          E
29      Arts561                    Augustan_Cultral_Revolution          E
30      Arts565  Environmental_Studies_Research_Seminar_Junior_...          E
32      Arts569                          Cell_Biol__Biochem          E
33      Arts571                       France__The_EuropUnion          E
34      Arts573                       Analyzing_The_Pol_World          E
35      Arts575                 Early_Mesopotam_HistorySociety          E
36      Arts577                       France__The_EuropUnion          E
37      Arts579                          Early_Balcan_HistSoc          E
38      Arts581                         Comparative_Politics          E
39      Arts583                       British_Poetry_16601914          E
40      Arts585                     Contemporary_Socio_Theory          E
41      Arts587                          Elementary_Arabic_Ii          E
<class 'pandas.core.frame.DataFrame'>
Index: 41 entries, 0 to 41
Data columns (total 3 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
```

```
 0   Course_Number  41 non-null     object
 1   Course_Name    41 non-null     object
 2   Course_Type    41 non-null     object
dtypes: object(3)
memory usage: 1.3+ KB
The course with the most number of students registered is: Comput_Linear_Algebra
    Student_Name  ... Course_Type
0      Bill_Mumy  ...           F
1      Bill_Mumy  ...           E
2      Bill_Mumy  ...           E
3      Bill_Mumy  ...           F
4      Bill_Mumy  ...           C
...          ...  ...         ...
3045  Stacy_Keach  ...           E
3046  Ann_Landers  ...           E
3047  Ann_Landers  ...           F
3048    Tyne_Daly  ...           F
3049    Tyne_Daly  ...           C

[3050 rows x 7 columns]
Course_Number   Arts400  Arts401  Arts465  ...  Arts583  Arts585  Arts587
Student_Name                               ...
Abella_Abzug          1        1        0  ...        0        0        1
Al_Gore               0        0        0  ...        0        0        0
Al_Hirt               0        0        1  ...        0        0        1
Al_Roker              1        0        0  ...        0        0        0
Alan_Alda             0        0        0  ...        0        0        0
...                 ...      ...      ...  ...      ...      ...      ...
Willis_Johnson        1        0        0  ...        0        0        0
Winona_Ryder          1        1        0  ...        0        0        0
Wolfgang_Puck         0        0        0  ...        0        0        1
Yogi_Berra            0        0        0  ...        0        0        0
Yoko_Ono              0        0        0  ...        0        0        0

[442 rows x 39 columns]
```