CS 101 - Foundation of Data Science and Engineering (Spring 2023)

PSET-4 - Managing Data Excercise-2 (100 pts)

This is an individual assignment. No collaboration is allowed.

Assignment Goal:

Part-1 : Explore Pandas, perform data cleaning using Pandas.

Part-2 : Generate random sample data in SQL

Part-3 : Practice writing SQL queries

Start by reviewing the provided file nj_teachers_salaries_pset4.csv. Examine the column names, data types of this data file. After reviewing this file please provide your solutions for the questions below.

Note: The file has identical columns that you worked on PSET-3, however all the data are not identical

Resources:
https://pandas.pydata.org/docs/reference/frame.html (https://pandas.pydata.org/docs/reference/frame.html)
Module 4 & Module 5 Lectures

Please feel free to create new cells in your notebook for completing the assignment.

Part-1 (60 points)

In this part you will be working with Pandas to explore and clean data. For each of the questions, please make sure that you show your work on what was done in each step. Add comments where necessary.

For Example if you drop rows, be sure to show the how many rows were dropped at each step. You can use df.shape to show before and after count.

For Questions 3-5 that involve modifying your values, you need to show us few rows where the modification was done. As an example you are looking at df['experience_total'] column and you discover that the column has values that are not numerical. You go ahead and set the values as np.NAN. You should show that those values were indeed set as nan

In [ ]:

In [189]:
```
import pandas as pd
import numpy as np
import mysql.connector as sq
```

In [ ]:

Question-1 (1 pts)

Create a dataframe called df using the provided csv file nj_teachers_salaries_pset4.csv. Use df.info() to get the information about the columns, non-null values, and data type inferred by Pandas for each column.

Pandas tries to infer the data type of each column. However if you have a numerical column, with an invalid value (such as a string), it will infer it as an object. String values are inferred as object data type.

In [190]:
```
#your code here
```

In [191]:
```
df = pd.read_csv('nj_teachers_salaries_pset4.csv')
```

```
/var/folders/f7/jtjrhyg16t90k734gwz83gtr0000gn/T/ipykernel_15516/1339891705.py:1: DtypeWarning: Columns (6,12,13,14)
have mixed types. Specify dtype option on import or set low_memory=False.
  df = pd.read_csv('nj_teachers_salaries_pset4.csv')
```

In [192]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 138724 entries, 0 to 138723
Data columns (total 15 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   last_name            138715 non-null  object 
 1   first_name           138715 non-null  object 
 2   county               138715 non-null  object 
 3   district             138714 non-null  object 
 4   school               138714 non-null  object 
 5   primary_job          138714 non-null  object 
 6   fte                  138636 non-null  object 
 7   salary               138715 non-null  float64
 8   certificate          138712 non-null  object 
 9   subcategory          138714 non-null  object 
 10  teaching_route       138713 non-null  object 
 11  highly_qualified     138714 non-null  object 
 12  experience_district  138222 non-null  object 
 13  experience_nj        138092 non-null  object 
 14  experience_total     138082 non-null  object 
dtypes: float64(1), object(14)
memory usage: 15.9+ MB
```

Question-2 (1 pts)

Drop rows that have all values as NaN. (Recall from lecture that you have to set the parameter how='all')

In [193]:
```
#your code here
```

In [194]:
```
df.dropna(how = "all", inplace = True)
```

Question-3 (20 pts)

Numerical Columns/boolean :

Identify numerical/boolean columns, remove any invalid characters from numerical/boolean columns by first setting it to np.NAN , and finally drop rows containing NaN values.

Set the correct data type for each of the numerical columns (i.e. int , float)

At the end of this step your dataframe should not contain any invalid values for numerical/boolean columns.

Please be sure to show your work.

Note: It is not required that you impute the invalid values. But if you choose to do so, it is ok. Just make sure that you are not adding any bias to your data.

In [195]:
```
#your code here
```

In [196]:
```
df[1:2] #viewing columns
```

Out[196]:

	last_name	first_name	county	district	school	primary_job	fte	salary	certificate	subcategory	teaching_route	highly_qualified	experience_district	expe
1	Bird	Kelly	Atlantic	Atlantic City	Atlantic City High School	Coordinator Substance Abuse	1.0	118415.0	Standard certificate	General ed	Traditional	Doesn't need to be highly qualified	16.0	

In [197]:
```
#list of num/bool columns
num = ['fte', 'salary', 'experience_district', 'experience_nj', 'experience_total']

df[num] = df[num].replace('[^0-9]', np.NaN, regex = True)
df = df.dropna()
```

In [198]:
```
#salary and experience are reported as integers
#fte can be reported as decimal
df['fte'] = df['fte'].astype(float)
df[num[1:]] = df[num[1:]].astype(int)

df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 131059 entries, 0 to 138723
Data columns (total 15 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   last_name            131059 non-null  object 
 1   first_name           131059 non-null  object 
 2   county               131059 non-null  object 
 3   district             131059 non-null  object 
 4   school               131059 non-null  object 
 5   primary_job          131059 non-null  object 
 6   fte                  131059 non-null  float64
 7   salary               131059 non-null  int64  
 8   certificate          131059 non-null  object 
 9   subcategory          131059 non-null  object 
 10  teaching_route       131059 non-null  object 
 11  highly_qualified     131059 non-null  object 
 12  experience_district  131059 non-null  int64  
 13  experience_nj        131059 non-null  int64  
 14  experience_total     131059 non-null  int64  
dtypes: float64(1), int64(4), object(10)
memory usage: 16.0+ MB
```

Question-4 (5 pts)

String Columns:

Identify string/object columns. Remove any leading and trailing spaces.

No rows should be dropped.

In [199]:
```
#your code here
```

In [200]:
```
ob = df.select_dtypes(exclude = [int, float]).columns #object columns

df[ob] = df[ob].apply(lambda x: x.str.strip()) #lambda to strip multiple columns in one line
```

Question-5 (20 pts)

Additional Cleaning - String Column :

Perform additional cleaning on string columns. Remove any special/invalid characters from the string columns.

Example :

df['primary_job'] contains a value 'Family & Consumer Sciences â€“ Apparel, Textiles And Interiors'.

The special character should be removed to give the value 'Family & Consumer Sciences Apparel, Textiles And Interiors'

Perform data cleaning on at least 3 string columns.

You should try to avoid setting string columns to np.NAN , and dropping it. However, it is ok if you set some rows to np.NAN and drop it for which values are completely invalid. In the end you should have 100,000 or more rows.

We are not looking for a perfect solution. The data will still consist of invalid values. We are more interested in seeing how you have applied your learning to this assignment.

In all cases please show your work.

In [201]:
```
#your code here
```

In [202]:
```
df['primary_job'] = df['primary_job'].str.replace('[^A-Za-z0-9\s./&-]', '', regex = True) #remove symbols except for so
df['teaching_route'] = df['teaching_route'].str.replace('[^A-Za-z0-9\s./&-]', '', regex = True)
df['last_name'] = df['last_name'].str.replace('[^A-Za-z\s.-]', '', regex = True) #remove digits and symbols which would
df['first_name'] = df['first_name'].str.replace('[^A-Za-z\s.-]', '', regex = True)
df.dropna(thresh = 5) #drop rows with 5+ NA
```

Out[202]:

nty	district	school	primary_job	fte	salary	certificate	subcategory	teaching_route	highly_qualified	experience_district	experience_nj	experience_total
ntic	Atlantic City	Pennsylvania Ave School	Mathematics Grades 5 - 8	1.0	98774	Standard certificate	General ed	Traditional	Not highly qualified	13	13	13
ntic	Atlantic City	Atlantic City High School	Coordinator Substance Abuse	1.0	118415	Standard certificate	General ed	Traditional	Doesn't need to be highly qualified	16	16	16
ntic	Atlantic City	Atlantic City High School	Health & Physical Education	0.8	98774	Standard certificate	General ed	Traditional	Doesn't need to be highly qualified	13	13	15
ntic	Atlantic City	Atlantic City High School	Resource Program In-class	1.0	66184	Standard certificate	Special ed	Alternate	Doesn't need to be highly qualified	16	16	16
ntic	Atlantic City	Atlantic City High School	School Psychologist	1.0	101866	Standard certificate	General ed	Traditional	Doesn't need to be highly qualified	12	12	12
...	...	...	...	...	...	...	...	...	...	...	...	...
rter	The Village Charter School	The Village Charter School	Library Skills Development	1.0	24000	Standard certificate	General ed	Traditional	Highly qualified. Has gradate or undergraduate...	2	13	14
rter	The Village Charter School	The Village Charter School	School Social Worker	1.0	69000	Standard certificate	General ed	Traditional	Highly qualified. Has gradate or undergraduate...	6	7	7
rter	The Village Charter School	The Village Charter School	Basic Skills/remedial English	1.0	63000	Standard certificate	General ed	Traditional	Highly qualified. Has gradate or undergraduate...	14	15	15
rter	The Village Charter School	The Village Charter School	Supervisor Curriculum & Instruction	1.0	84000	Standard certificate	Admin or supervisor	Traditional	Highly qualified. Has gradate or undergraduate...	14	17	17
rter	The Village Charter School	The Village Charter School	Elementary School Teacher K-5	1.0	47000	Standard certificate	General ed	Traditional	Highly qualified. Has gradate or undergraduate...	1	1	8

Question-6 (1 pts)

Drop any duplicate rows. Display df.info() to shows the data types, and Non-Null count.

https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop_duplicates.html
(https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.drop_duplicates.html)

In [208]:
```
df = df.drop_duplicates()
df.info() #dropped 21 dupliciates
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 131038 entries, 0 to 138723
Data columns (total 15 columns):
 #   Column               Non-Null Count   Dtype  
---  ------               --------------   -----  
 0   last_name            131038 non-null  object 
 1   first_name           131038 non-null  object 
 2   county               131038 non-null  object 
 3   district             131038 non-null  object 
 4   school               131038 non-null  object 
 5   primary_job          131038 non-null  object 
 6   fte                  131038 non-null  float64
 7   salary               131038 non-null  int64  
 8   certificate          131038 non-null  object 
 9   subcategory          131038 non-null  object 
 10  teaching_route       131038 non-null  object 
 11  highly_qualified     131038 non-null  object 
 12  experience_district  131038 non-null  int64  
 13  experience_nj        131038 non-null  int64  
 14  experience_total     131038 non-null  int64  
dtypes: float64(1), int64(4), object(10)
memory usage: 16.0+ MB
```

Question -7

Save your cleaned dataframe as cleaned_data.csv. Be sure to set the parameter index = False to avoid saving the index as an extra column

ex: df.to_csv("cleaned_data.csv", index=False)

In [16]:
```
#your code here
```

In [209]:
```
df.to_csv("cleaned_data.csv", index = False)
```

Question -8

Similar to PSET-3

8.1 Connect to your MySql database using your username and password. Name the cursor returned from the mysql connection object as mycursor. (1 pts)

8.2 Use the same database as PSET-3 nj_state_teachers_salaries, or if you have deleted it create a database called nj_state_teachers_salaries

8.3 Create a table called teachers_salaries_pset4 with all the columns in your cleaned_data.csv. For this part ,be sure to use appropriate data type for all the columns. If you are facing difficulty creating a column with Float or bool or int , it is ok to store it as TEXT. (MAX 2 allowed for numerical columns being stored as TEXT) (3 pts)

8.4 Using LOAD DATA statement (as discussed in Module 4 lectures) load the data from cleaned_data.csv to your table created in 8.3. Use of OPTIONALLY ENCLOSED BY clause and TERMINATED by clause is recommended. (3 pts)

In [210]:
```
#your code here 8.1
mydb = sq.connect(
    host="localhost",
    user="root",
    password="mysqlpass")

mycursor = mydb.cursor()
```

In [211]:
```
#your code here 8.2
mycursor.execute('USE nj_state_teachers_salaries')
```

In [212]:
```
#your code here 8.3
mycursor.execute('CREATE TABLE teachers_salaries_pset4 (last_name TEXT,first_name TEXT,county TEXT,district TEXT,school
mydb.commit()
```

In [219]:
```
SQLCMD = "LOAD DATA INFILE '/usr/local/cleaned_data.csv' \
INTO TABLE teachers_salaries_pset4 FIELDS TERMINATED BY ',' \
OPTIONALLY ENCLOSED BY '\"' \
LINES TERMINATED BY '\\n' \
IGNORE 1 ROWS"

mycursor.execute('USE nj_state_teachers_salaries')
mycursor.execute(SQLCMD)
mydb.commit()
```

Question 9 - For this question you are only required to run the cells. To get credit your code from Question 8 must have been successfully run, and executed. No credit will be awarded if data was loaded using MySQL workbench.

Question 9 (5 pts)

Run the 2 cells below. The code checks if all the data rows and columns were stored in the database.

The code below assumes that you named your cursor object as mycursor(As specified in Question-8). If you named it differently, you can rename mycursor to match the variable name.

In [220]:
```
cmd = "select count(*) from \
                nj_state_teachers_salaries.teachers_salaries_pset4 "
mycursor.execute(cmd)
count = mycursor.fetchone()[0]

print(f"Number of rows in teachers_salaries table : {count}")
```

```
Number of rows in teachers_salaries table : 131038
```

In [221]:
```
cmd = """SELECT COUNT(*) \
               FROM INFORMATION_SCHEMA.COLUMNS \
               WHERE table_schema = 'nj_state_teachers_salaries' \
               AND table_name = 'teachers_salaries_pset4'"""
mycursor.execute(cmd)
count = mycursor.fetchone()[0]
print(f"Number of columns in teachers_salaries table : {count}")
```

```
Number of columns in teachers_salaries table : 15
```

End of Part-1

In [ ]:

For both Part-2 and Part-3 you will need to work on MySQL workbench. For both parts you must submit .sql files. More information below.

Part-2 (10 pts)

For this part you will generate a random sample data from the table you created in Part-1 and save it as a csv file. Generating random samples have many use cases in the real world. For example, you are a developer who is working on a software application that requires access to a critical database. Instead you maybe given only a sample of data to work with to develop your application. Another use case is bootstrapping in statistics, or when you test your models with samples of data.

Question 1 (8 pts)

Use a SELECT statement to generate and output a random sample to :

Include all columns

Include field (column) headings

Randomly select 777 records with a seed value of 7

Output results to a csv file named sample.csv

save your sql as output.sql . You will submit this file as a part of this assignment.

You will find module 5 lecture on SQL Random Sample Generation useful

Question 2 (2 pts)

Create a dataframe using sample.csv generated from Question-1. Display the first 5 rows, and last 5 rows. Print the shape of the dataframe.

In [229]:
```
#your code here
sampledf = pd.read_csv('sample.csv')
```

In [232]:
```
sampledf.head() #first 5
```

Out[232]:

first_name	county	district	school	primary_job	fte	salary	certificate	subcategory	teaching_route	highly_qualified	experience_district	experienc
Gina	Bergen	Bergenfield Boro	Lincoln Elementary School	Reading Specialist	1.0	79685	Standard certificate	General ed	Traditional	Highly qualified. Has gradate or undergraduate...	11	
Beth N	Morris	Mine Hill Twp	Canfield Avenue School	Elementary Kindergraten-8 Grade	1.0	52650	Standard certificate	General ed	Traditional	Highly qualified. Has gradate or undergraduate...	4	
Andrea W	Union	Morris-union Jointure Com	Developmental Learning Center Warren	Principal Handicapped School	1.0	108959	Standard certificate	Admin or supervisor	Traditional	Highly qualified. Has gradate or undergraduate...	9	
Isolina	Charter	North Star Academy Charter School	North Star Academy Charter School	Elementary Kindergraten-8 Grade	1.0	63000	Provisional	General ed	Alternate	Highly qualified. Passed the Praxis/NTE	2	
Eileen M	Cape May	Lower Twp	Maud Abrams School	Elementary Kindergraten-8 Grade	1.0	57265	Standard certificate	General ed	Alternate	Highly qualified. Passed the Praxis/NTE	13	

In [231]:
```
sampledf.tail() #last 5
```

Out[231]:

	last_name	first_name	county	district	school	primary_job	fte	salary	certificate	subcategory	teaching_route	highly_qualified	experience_dist
772	Michele	Christopher	Essex	Belleville Town	Belleville Ps5	Physical Education	0.5	52680	Standard certificate	General ed	Traditional	Highly qualified. Has 30 credits in content area.	
773	Inigo	David Dennis	Passaic	Passaic City	Lincoln Middle School # 4	School Counselor	1.0	61443	Standard certificate	General ed	Traditional	Doesn't need to be highly qualified	
774	Cafara	Raymond	Ocean	Jackson Twp	Jackson Memorial High School	Social Studies Non-elementary	1.0	56232	Standard certificate	General ed	Traditional	Highly qualified. Passed the Praxis/NTE	
775	Neilio	Herbert J	Gloucester	Glassboro	Glassboro High School	Health & Physical Education	1.0	82923	Standard certificate	General ed	Traditional	Highly qualified. National Board Certified.	
776	Lynch	Liliana	Burlington	Delran Twp	Delran Intermediate School	Resource Program Pull-out Support	1.0	54360	Standard certificate	Special ed	Traditional	Highly qualified. Passed the Praxis/NTE	

In [235]:
```
print(sampledf.shape) #shape
```

```
(777, 15)
```

Part-3 (30 pts)

For this part you will work on sql queries. You will write your queries for the provided dataset teachersample.csv. We could have asked you to write the queries based on the existing table nj_state_teachers_salaries.teachers_salaries_pset4 , however everyone's data cleaning process will be different resulting in different dataset. We will use a standard dataset to evaluate your queries

All work need to be done in MySQL workbench

Question 1

Create a table called salaries within the nj_state_teachers_salaries database. Load the data in to the table from the provided file teachersample.csv.

You don't need to submit the code for this. This table is intended only for queries in Question-2.

Question 2 (30 pts)

Each query is worth 3 pts

Write the following queries in MySQL workbench, and name the file queries.sql. The file you submit should have the exact name for you to get credit. We will run your query, so you don't need to capture the output. The file should include only the 10 queries. Be sure to test it before submission.

Example Query for your reference:

select count(*) from nj_state_teachers_salaries.salaries

1. Calculate the average salary

In [ ]:
```
#73361
```

2. Calculate the number of people whose salary is more than 150,000.

In [ ]:
```
#9
```

3. Get the last name of the ones who make more than 150,000 but have less than 5 years of total experience

In [ ]:
```
#Cullis, Rafferty
```

4. Get the highest salary for Preschool, School Counselor, Principal (anyone with the word Principal in the title), School Psychologist, and Kindergarten. (These are individual queries. You should have 5 separate queries.)

In [ ]:
```
#Preschool: 102318
#School Counselor: 103318
#Principal: 158327
#School Psychologist: 102030
#Kindergarten: 95890
```

5. Get the last name, first name, and salary of the lowest earner who works in Atlantic City

In [ ]:
```
#Gatti, Gina M 52107
```

6. Get the total number of employees working in Passaic City with more than ten years of total experience.

In [ ]:
```
#2
```

Submission on Gradescope

Gradescope Link:https://www.gradescope.com/courses/501618 (https://www.gradescope.com/courses/501618)

Submission :

Part -1 : This jupyter notebook, and a pdf of this notebook.

Part -2 : output.sql and sample.csv

Part -3 : queries.sql containing all your queries. This file should only include the sql queries. Please don't include the code that created the salaries table.

To create a pdf of this notebook : In your browser open print, and save as pdf. Name the pdf LastNameFirstName.pdf example: DoeJohn.pdf

Name this jupyter notebook with the same format LastNameFirstName.ipynb

Make sure that your notebook has been run before creating pdf. Any outputs from running the code needs to be clearly visible. We need all the files from Part-1, Part-2, and Part-3 to assign you grades.

Drop all the files in gradescope under PSET 4: Managing Data Exercise 2.

In [ ]: