3 july 2025

Dataset descriptions

Final dataset: missing SIS Courses from Course Evals.csv

The columns here are merged from both course\_evals (raw data) and SIS Instructor (raw data). Columns with \_x in thename are form course evals and column with \_y are form SIS. Due to these rows being data that is missing from SIS but only in course\_evals, all column with \_y will be empty.

Note: The correct UNIs and fixed\_professor names are not here as I had to use raw Data to merge the two to make sure matching issues wouldn’t occur.

For both course\_evals data and SIS data: Cleaned colums and course number, cleaned professor fullname from course\_evals and SIS, created new\_unique\_course\_number.

Left merged course\_evals and SIS on new\_unique\_course\_id (only those new\_unique\_course\_id not in SIS but in course\_evals\_long are filtered for).--> not\_in\_SIS df.

Converted to csv.

import pandas as pd

import re

def clean\_course\_num(course):

# Make sure it's a string, strip whitespace, and convert to uppercase

course = str(course).strip().upper()

# Extract the first letter and a 4-digit number at the end

match = re.search(r'([A-Z])[A-Z]\*[\s-]?(\d{4})$', course)

if match:

return match.group(1) + match.group(2)

return course # Fallback if pattern doesn't match

'''

actual code

'''

course\_evals\_long = pd.read\_csv("course\_evaluations\_long\_20001\_to\_20243.csv")

SIS\_instructor = pd.read\_excel("SIS Course Instructors.xlsx")

#tcdb = pd.read\_excel("TCDB\_CBS\_Courses.xlsx")

#cleaning columns we are using

course\_evals\_long['professor\_fullname'] = course\_evals\_long['professor\_fullname'].str.strip()

course\_evals\_long['clean\_course\_number'] = course\_evals\_long['course\_number'].apply(clean\_course\_num)

#Some course identifies in course\_evals doesnt have term number and only have year-- so redefining unique course id

course\_evals\_long['new\_unique\_course\_id'] = (

course\_evals\_long['term\_number'].astype(str) +

course\_evals\_long['clean\_course\_number'].astype(str)+

course\_evals\_long['section\_number'].astype(str))

SIS\_instructor['professor\_fullname'] = SIS\_instructor['Instructor\_Name'].str.strip()

SIS\_instructor['course\_number'] = SIS\_instructor['Course\_Identifier'].str[-5:]

SIS\_instructor['clean\_course\_number'] = SIS\_instructor['course\_number'].apply(clean\_course\_num)

SIS\_instructor['new\_unique\_course\_id'] = (

SIS\_instructor['Term\_Identifier'].astype(str) +

SIS\_instructor['clean\_course\_number'].astype(str)+

SIS\_instructor['Section\_Code'].astype(str))

SIS\_instructor['new\_unique\_course\_id'] = SIS\_instructor['new\_unique\_course\_id'].fillna( SIS\_instructor['Unique\_id'])

#Filtering for new\_unique\_course\_id not in SIS but in course\_evals\_long

merged = course\_evals\_long.merge(

SIS\_instructor,

on='new\_unique\_course\_id', # or any other shared column

how='left',

indicator=True

)

not\_in\_SIS = merged[merged['\_merge'] == 'left\_only']

not\_in\_SIS.to\_csv("missing SIS Courses from Course Evals.csv")