PS2

September 18, 2025

1 Problem set 2

1.1 Name: Qiuli Lai

1.2 Link to your PS2 github repo: https://github.com/098pipi/data1030_ps2.git

1.3 Problem 0

-2 points for every missing green OK sign. If you don't run the cell below, that's -16 points.

Make sure you are in the DATA1030 environment.

[1]:
```python
from __future__ import print_function
from packaging.version import parse as Version
from platform import python_version

OK = '\x1b[42m[ OK ]\x1b[0m'
FAIL = '\x1b[41m[FAIL]\x1b[0m'

try:
    import importlib
except ImportError:
    print(FAIL, 'Python version 3.12.10 is required,'
                ' but %s is installed.' % sys.version)

def import_version(pkg, min_ver, fail_msg=''):
    mod = None
    try:
        mod = importlib.import_module(pkg)
        if pkg in {'PIL'}:
            ver = mod.VERSION
        else:
            ver = mod.__version__
        if Version(ver) == Version(min_ver):
            print(OK, '%s version %s is installed.'
                  % (lib, min_ver))
        else:
            print(FAIL, '%s version %s is required, but %s installed.'
                  % (lib, min_ver, ver))
    except ImportError:
        print(FAIL, '%s not installed. %s' % (pkg, fail_msg))
    return mod


# first check the python version
pyversion = Version(python_version())

if pyversion >= Version('3.12.10'):
    print(OK, 'Python version is %s' % pyversion)
elif pyversion < Version('3.12.10'):
    print(FAIL, 'Python version 3.12.10 is required,'
                ' but %s is installed.' % pyversion)
else:
    print(FAIL, 'Unknown Python version: %s' % pyversion)


print()
requirements = {'numpy': '2.2.5', 'matplotlib': '3.10.1','sklearn': '1.6.1',
                'pandas': '2.2.3','xgboost': '3.0.0', 'shap': '0.47.2',
                'polars': '1.27.1', 'seaborn': '0.13.2'}

# now the dependencies
for lib, required_version in list(requirements.items()):
    import_version(lib, required_version)
```

```
[ OK ] Python version is 3.13.5

[ OK ] numpy version 2.2.5 is installed.
[ OK ] matplotlib version 3.10.1 is installed.
[ OK ] sklearn version 1.6.1 is installed.
[ OK ] pandas version 2.2.3 is installed.
[ OK ] xgboost version 3.0.0 is installed.
[ OK ] shap version 0.47.2 is installed.
[ OK ] polars version 1.27.1 is installed.
[ OK ] seaborn version 0.13.2 is installed.
```

1.4 Problem 1 - data collection

Which Rhode Island school has the largest undergraduate student population? And graduate student population? You will collect and analyze data to answer these two questions using the College Scoreboard API of the U.S. Department of Education. An API (application point interface) is a mechanism which allows two software components to communicate with each other using a set of definitions and protocols. The two software components in this case are your jupyter notebook running python and the College Scoreboard server.

APIs are a popular way to share and modify data in an automated, secure, and cost-efficient way. Read more about APIs here.

The documentation of the College Scoreboard API is available here, read it carefully. It is a REST

API which means that we will perform operations using standard HTTP methods. We want to know the name of each RI school, in which city/town it is located, the zipcode, and how many undergrad and gradute students they have based on the most recent data.

You will use python packages like `requests`, `dotenv`, and `pandas` or `polars` to collect and save the data. - The `requests` package is how you will query the API. We will submit only one query to the College Scoreboard API which could in principle be done in a browser. However I want you write python code because often you need to make a large number of API requests which needs to be automated with code. Read more about it here. - One way APIs achieve security is to limit access to authorized users only. Authorized users have an API_KEY which is a secret key specific to each user. This API_KEY needs to be provided when you make a request to the server, it will be part of the HTTP URL. This is how the server knows who makes the request and what level of access the user has. While most users can have read access, only a limited number of users usually have access to modify a dataset. Therefore the API_KEY needs to be kep secret. **Your API_KEY should NEVER be directly copy-pasted into your notebook or pushed to any github repository!** If you do so, that's a security risk even if the repository is private. Also, you will lose points. A pretty popular way to share secrets like the API_KEY with your notebook is done by using the `dotenv` package. Read more about it here. - The API request will be returned in a json format. You will use `pandas` or `polars` to convert the json output to a dataframe, and save it as a csv file.

If some of these terminologies or concepts don't make sense right now, don't worry about it. Read the linked documentations, follow the steps as outlined below, and post on the course forum or come to office hours if you have questions.

[5]:
```
# resolve any error messages you might encounter as you work through the steps

# import pandas/polars, requests, and dotenv packages here
import os
import pandas as pd
import requests as req
import dotenv

### Setup Steps (do these first):
# 1. Go to the college scoreboard documentation (linked above) and request an␣
  ↪API_KEY
# 2. Create a `.env` file in the same folder as this notebook
# 3. Add your API key to the `.env` file: API_KEY=your_key_here
# 4. Add `.env` to your `.gitignore` file
# 5. Test that `load_dotenv()` works before proceeding
dotenv.load_dotenv('DATA1030_PS2.env')
api_key = os.getenv('API_KEY')
# If you encounter any issues, check the documentation for most common errors.


# read the college scoreboard documentation carefully.
# collect info on all Rhode Island schools. We want to know the name of each␣
  ↪school,
# in which city/town it is located, the zipcode, and how many undergrad and␣
↪gradute students they have based on the most recent data
# collect below the fields necessary to collect the data as a python dictionary.
# add your API_KEY to this dictionary (feel free to look up how to access␣
↪environment variables, if you're confused)
request_params = {'api_key': api_key, 'school.state': 'RI',
                  'fields': 'id,school.name,school.city,school.zip,latest.
↪student.enrollment.undergrad,latest.student.enrollment.grad'}


# add the base URL below.
api_url = 'https://api.data.gov/ed/collegescorecard/v1/schools'


# use the request_params and the api_url to make a get request using the␣
↪requests package
# save the response
r = req.get(api_url, request_params)

# print the response below while you are developing the code
# print(r)
# comment out the print statement once you are certain the code works as␣
↪intended
# this is for debugging purposes only because requests are finicky things.
# one missed character in the URL, one small typo in one of the parameters,
# one small error in your code, and the request will return an error code.
# therefore it is important to carefully read the manuals
# and follow them to the letter


# write an if-else statement
# if the response code is 200 (successful request), save the result as a json␣
↪object
# else print our the response code and the error message and raise a valueError
# note: some fatal errors (e.g. if the base URL is totally wrong) will come up␣
↪in the original get call. Here, we only care about calls that return a␣
↪unsuccessful request.
if r.status_code == 200:
    data = r.json()
else:
    print('Error: ', r.status_code)
    raise ValueError(f'Request failed with status {r.status_code}: {r.text}')

# print out the json result below.
print(data['results'])
```

```python
# save the json results into a pandas dataframe
uni_df = pd.DataFrame(data['results'])
# uni_df = pl.DataFrame()

# save uni_df into a csv file, save the file in the same folder w
# here this notebook is located
uni_df.to_csv('rhode_island_schools.csv', index = False)
# You are done with problem 1!
```

```
[{'latest.student.enrollment.undergrad_12_month': 8026,
'latest.student.enrollment.grad_12_month': 3740, 'school.name': 'Brown
University', 'school.city': 'Providence', 'school.zip': '02912', 'id': 217156},
{'latest.student.enrollment.undergrad_12_month': 3306,
'latest.student.enrollment.grad_12_month': 555, 'school.name': 'Bryant
University', 'school.city': 'Smithfield', 'school.zip': '02917-1291', 'id':
217165}, {'latest.student.enrollment.undergrad_12_month': 4469,
'latest.student.enrollment.grad_12_month': 466, 'school.name': 'Johnson & Wales
University-Providence', 'school.city': 'Providence', 'school.zip': '02903-3703',
'id': 217235}, {'latest.student.enrollment.undergrad_12_month': 2279,
'latest.student.enrollment.grad_12_month': 190, 'school.name': 'New England
Institute of Technology', 'school.city': 'East Greenwich', 'school.zip':
'02818-1205', 'id': 217305}, {'latest.student.enrollment.undergrad_12_month':
1327, 'latest.student.enrollment.grad_12_month': None, 'school.name': 'New
England Tractor Trailer Training School of Rhode Island', 'school.city':
'Pawtucket', 'school.zip': '02860', 'id': 217323},
{'latest.student.enrollment.undergrad_12_month': 4447,
'latest.student.enrollment.grad_12_month': 588, 'school.name': 'Providence
College', 'school.city': 'Providence', 'school.zip': '02918-0001', 'id':
217402}, {'latest.student.enrollment.undergrad_12_month': 8659,
'latest.student.enrollment.grad_12_month': 1353, 'school.name': 'Rhode Island
College', 'school.city': 'Providence', 'school.zip': '02908', 'id': 217420},
{'latest.student.enrollment.undergrad_12_month': 16371,
'latest.student.enrollment.grad_12_month': None, 'school.name': 'Community
College of Rhode Island', 'school.city': 'Warwick', 'school.zip': '02886-1807',
'id': 217475}, {'latest.student.enrollment.undergrad_12_month': 17493,
'latest.student.enrollment.grad_12_month': 3160, 'school.name': 'University of
Rhode Island', 'school.city': 'Kingston', 'school.zip': '02881', 'id': 217484},
{'latest.student.enrollment.undergrad_12_month': 2165,
'latest.student.enrollment.grad_12_month': 541, 'school.name': 'Rhode Island
School of Design', 'school.city': 'Providence', 'school.zip': '02903-2784',
'id': 217493}, {'latest.student.enrollment.undergrad_12_month': 4655,
'latest.student.enrollment.grad_12_month': 352, 'school.name': 'Roger Williams
University', 'school.city': 'Bristol', 'school.zip': '02809-2921', 'id':
217518}, {'latest.student.enrollment.undergrad_12_month': 2276,
'latest.student.enrollment.grad_12_month': 856, 'school.name': 'Salve Regina
University', 'school.city': 'Newport', 'school.zip': '02840-4192', 'id':
217536}, {'latest.student.enrollment.undergrad_12_month': 254,
'latest.student.enrollment.grad_12_month': None, 'school.name': 'Empire Beauty
School-Warwick', 'school.city': 'Providence', 'school.zip': '02903', 'id':
217581}, {'latest.student.enrollment.undergrad_12_month': None,
'latest.student.enrollment.grad_12_month': 556, 'school.name': 'Roger Williams
University School of Law', 'school.city': 'Bristol', 'school.zip': '02809-5171',
'id': 409616}, {'latest.student.enrollment.undergrad_12_month': 1201,
'latest.student.enrollment.grad_12_month': None, 'school.name': 'Lincoln
Technical Institute-Lincoln', 'school.city': 'Lincoln', 'school.zip': '02865',
'id': 433101}, {'latest.student.enrollment.undergrad_12_month': 103,
'latest.student.enrollment.grad_12_month': None, 'school.name': 'IYRS School of
Technology & Trades', 'school.city': 'Newport', 'school.zip': '02840', 'id':
437237}, {'latest.student.enrollment.undergrad_12_month': 222,
'latest.student.enrollment.grad_12_month': None, 'school.name': 'Paul Mitchell
the School-Rhode Island', 'school.city': 'Cranston', 'school.zip': '02920',
'id': 443641}, {'latest.student.enrollment.undergrad_12_month': 209,
'latest.student.enrollment.grad_12_month': None, 'school.name': 'Toni & Guy
Hairdressing Academy-Cranston', 'school.city': 'Cranston', 'school.zip':
'02920', 'id': 455965}, {'latest.student.enrollment.undergrad_12_month': 2687,
'latest.student.enrollment.grad_12_month': 921, 'school.name': 'Johnson & Wales
University-Online', 'school.city': 'Providence', 'school.zip': '02903', 'id':
460349}, {'latest.student.enrollment.undergrad_12_month': 204,
'latest.student.enrollment.grad_12_month': None, 'school.name': 'MotoRing
Technical Training Institute', 'school.city': 'East Providence', 'school.zip':
'02914-5022', 'id': 479062}]
```

1.5 Problem 2 - EDA

As mentioned in class, you should approach a new dataset with a healthy set of skepticism. You will study the dataset you collected in problem 1.

1.5.1 Problem 2a

Solve the tasks outlined in the cell below.

```
[6]: # import the necessary packages
import pandas as pd

# Read in the csv file and print out the columns.
display(uni_df)
print(uni_df.columns)
# Do you have all the columns we need? If you don't, go back to problem 1 to␣
 ↪resolve the issue.
# Do you have extra columns we don't need? Drop the unnecessary columns.

# Check for duplicate rows in the dataset. If there are duplicate rows, drop␣
 ↪all but one.
uni_df.duplicated() # there're no duplicated rows, so we don't need to drop
```

```
    latest.student.enrollment.undergrad_12_month  \
0                                         8026.0
```

```
1                                               3306.0
2                                               4469.0
3                                               2279.0
4                                               1327.0
5                                               4447.0
6                                               8659.0
7                                              16371.0
8                                              17493.0
9                                               2165.0
10                                              4655.0
11                                              2276.0
12                                               254.0
13                                                 NaN
14                                              1201.0
15                                               103.0
16                                               222.0
17                                               209.0
18                                              2687.0
19                                               204.0

    latest.student.enrollment.grad_12_month  \
0                                    3740.0
1                                     555.0
2                                     466.0
3                                     190.0
4                                       NaN
5                                     588.0
6                                    1353.0
7                                       NaN
8                                    3160.0
9                                     541.0
10                                    352.0
11                                    856.0
12                                      NaN
13                                    556.0
14                                      NaN
15                                      NaN
16                                      NaN
17                                      NaN
18                                    921.0
19                                      NaN

                                         school.name      school.city  \
0                                   Brown University       Providence
1                                  Bryant University       Smithfield
2             Johnson & Wales University-Providence       Providence
3                 New England Institute of Technology   East Greenwich
4   New England Tractor Trailer Training School of…       Pawtucket
```

```
5                                      Providence College       Providence
6                                     Rhode Island College       Providence
7                          Community College of Rhode Island          Warwick
8                                University of Rhode Island         Kingston
9                             Rhode Island School of Design       Providence
10                                 Roger Williams University          Bristol
11                                   Salve Regina University          Newport
12                             Empire Beauty School-Warwick       Providence
13                     Roger Williams University School of Law          Bristol
14                         Lincoln Technical Institute-Lincoln          Lincoln
15                          IYRS School of Technology & Trades          Newport
16                        Paul Mitchell the School-Rhode Island         Cranston
17                     Toni & Guy Hairdressing Academy-Cranston         Cranston
18                           Johnson & Wales University-Online       Providence
19                        MotoRing Technical Training Institute  East Providence

    school.zip      id
0        02912  217156
1   02917-1291  217165
2   02903-3703  217235
3   02818-1205  217305
4        02860  217323
5   02918-0001  217402
6        02908  217420
7   02886-1807  217475
8        02881  217484
9   02903-2784  217493
10  02809-2921  217518
11  02840-4192  217536
12       02903  217581
13  02809-5171  409616
14       02865  433101
15       02840  437237
16       02920  443641
17       02920  455965
18       02903  460349
19  02914-5022  479062

Index(['latest.student.enrollment.undergrad_12_month',
       'latest.student.enrollment.grad_12_month', 'school.name', 'school.city',
       'school.zip', 'id'],
      dtype='object')
```

```
[6]: 0     False
     1     False
     2     False
     3     False
     4     False
```

```
5      False
6      False
7      False
8      False
9      False
10     False
11     False
12     False
13     False
14     False
15     False
16     False
17     False
18     False
19     False
dtype: bool
```

1.5.2 Problem 2b

You will study the validity of the dataset, look for typos/errors, study the missing values, and finally answer our two original questions.

[ ]:
```
# we have some numerical features like the zipcode and the number of students
# what sort of impossible or incorrect values could you see in these columns?
# use critical thinking skills

# - We might get non numerical values in these columns, for example, strings␣
↪and
#   missing values; the value might be negative, so it's not valid; the value␣
↪may
#   be a fraction rather integer.

# test at least three columns and write at least one test per column to verify␣
↪the data validity
# here is the format of the test:
# if condition == True:
#     raise ValueError('error message')
szip = uni_df['school.zip']
undergrad = uni_df['latest.student.enrollment.undergrad_12_month']
grad = uni_df['latest.student.enrollment.grad_12_month']
# test if missing values
if (szip.isna().any()) == True:
    raise  ValueError(f'school.zip has {szip.isna().sum()} missing values.')
if (undergrad.isna().any()) == True:
    raise  ValueError(f'latest.student.enrollment.undergrad_12_month has␣
↪{undergrad.isna().sum()} missing values.')
if (grad.isna().any()) == True:
    raise  ValueError(f'latest.student.enrollment.grad_12_month has {grad.
 ↪isna().sum()} missing values.')

# consider the number of undergraduate/graduate student features.
# what values would be technically possible/plausible but unrealistic?
# write a test
# test if negative values
undergrad_coerce = pd.to_numeric(uni_df['latest.student.enrollment.
 ↪undergrad_12_month'], errors='coerce')
grad_coerce = pd.to_numeric(uni_df['latest.student.enrollment.grad_12_month'],␣
 ↪errors='coerce')
if (undergrad_coerce < 0).any():
    raise ValueError(f'undergrad_12_month has {undergrad_coerce.isna().sum()}␣
 ↪invalid (non-numeric or missing) values")')
if (grad_coerce < 0).any():
    raise ValueError(f'grad_12_month has {grad_coerce.isna().sum()} invalid␣
 ↪(non-numeric or missing) values")')

# are there missing values in the dataset?
print('missing value in zip column = ', szip.isna().sum())
print('missing value in latest.student.enrollment.undergrad_12_month column =␣
 ↪', undergrad.isna().sum())
print('missing value in latest.student.enrollment.grad_12_month column = ',␣
 ↪grad.isna().sum())

# in which columns?
# - in columns atest.student.enrollment.undergrad_12_month and atest.student.
 ↪enrollment.grad_12_month
# what fraction of the points contain missing values?
# - latest.student.enrollment.undergrad_12_month misses 8/20
#   latest.student.enrollment.grad_12_month misses 1/20

# why could the values be missing?
# write a short description on possible reasons.
# - 1. Maybe the college is just closed permanently .
#   2. Because of some ethical issue they do not provide such information
#   3. The staff who is responsible to the collection of the data didn't update␣
 ↪the
#       info. in time, or maybe there was some tech issue lead to the failure␣
 ↪of the update of this data.


# finally, answer the original questions we set out to investigate
# print out which RI school has the largest undergraduate student population.
print(f'{uni_df.loc[undergrad.idxmax(), 'school.name']} has the largest␣
 ↪undergraduate student population',  undergrad.max())
```

```python
# print out which RI school has the largest graduate student population.
print(f'{uni_df.loc[grad.idxmax(), 'school.name']} has the largest graduate␣
 ↪student population', grad.max())
# the dataset is small enough that you could answer these questions just by␣
 ↪looking at the csv file but that's not accepted.
# use code to determine the answer
```

```
missing value in zip column =  0
missing value in latest.student.enrollment.undergrad_12_month column =  1
missing value in latest.student.enrollment.grad_12_month column =  8
University of Rhode Island has the largest undergraduate student population
17493.0
Brown University has the largest graduate student population 3740.0
```