Final Exam
금융 IT 프로그래밍

날짜: 6월 14일, 2023년
시간: 오후 1:00~2:15

이름: _____

서명: _____

주의사항:

1. Cybercampus에 접속하여 기말시험 Portal에 저장된 `20xxxxxxx_FinalExam.py`와 함께 `*.csv` 화일도 내려받아 본인의 적당한 작업 Folder에 저장하시오.

2. 내려받은 `20xxxxxxx_FinalExam.py`의 머리부분, `20xxxxxxx`을 자신의 학번으로 변경하여 저장하시오. 최종 제출한 Python 화일 이름이 위와 다를 경우, **예외없이 무효로 처리되어 0점으로 채점**하니 주의하시오.

3. 부정행위가 적발된 경우, 출석, 숙제 및 중간/기말시험의 결과에 무관하게 학점은 **F**이다.

4. GPT, Google, Kaggle, Naver 등에서 검색된 Code를 참조할 수 있으나, (Rule#1) 반드시 참조 표기를 Comment로 남기고 (Rule#2) **참조한 Code를 재작성하여 구현**해야한다.

   Rule#1 또는 #2을 위배한 경우 부정행위로 간주해 학점은 **F**이다.

   여기서 재작성이라 함은 참조한 Code와 제출한 Code를 Stanford Code Copy 프로그램 MOSS (`http://theory.stanford.edu/~aiken/moss/`)를 실행하여 일치률이 20% 미만이어야 한다 (변수명 바꾸고 위치를 바꾸어도 불일치율을 낮추는 효과가 매우 작다). **MOSS를 사용한 유사도 측성에 의해 부정행위로 판단된 사항은 이의신청의 대상이 아니다**.

5. 시험 문제지는 양면으로 표지를 포함하여 **총 5장**이며, 시작하기 전에 확인하라. 내용을 모두 확인한 후 이름을 기입하고 서명한 후, 퇴실하기 전에 제출하시오. 출석부로 사용되어 제출하지 않을 경우, 결시로 처리되어 학점은 **F**이다.

1 CSV File Description

CSV 화일의 구조에 대한 설명으로 시작한다. CSV 화일은 게이트볼 동호회에 대한 정보–선수 (Players), 팀 (Teams), 다른 동호회와의 친선경기 (Matches) 및 위원회 (Committees)에 대한 정보를 기본적으로 저장한다. 추가로 개별 선수의 징계사항 (Penalties)도 기록으로 저장한다. 단순화를 위하여, 게이트볼 동호회는 많다고 가정하지만 동호회 정보 자체는 제공하지 않는다 [1]. 리그 정보도 단순화를 위하여 생략해버리고 제공하지 않는다.

CSV 화일은 5개이고 각 Column Header의 이름, 의미와 Data Type은 다음과 같다, 단 정수/실수의 길이나 문자열의 길이는 CSV 화일을 참조해 각자 적절히 정하시오.

Players

PNo	Integer	선수 고유번호
PName	String	선수 이름
BDate	String	생일
Sex	String	성별
JDate	String	최초 참여일자
Address	String	주소
PCode	String	우편번호
Mobile	String	전화번호
PType	String	선수 유형

Teams

TNo	Integer	팀 고유번호
CNo	Integer	주장 선수번호
Division	String	소속 리그

Matches

MNo	Integer	경기 고유번호
TNo	Integer	팀번호
PNo	Integer	선수번호
Win	Integer	이긴 세트의 수
Loss	Integer	패한 세트의 수

Penalties

PNo	Integer	선수번호
PId	Integer	징계번호
PDate	String	징계일자
PAmount	Float	벌금

Committees

PNo	Integer	선수번호
Duration	String	재직기간
Position	String	직책

CSV 화일별로 Column Header에 대한 세부 정보를 기술한다:

- Players
  - 동호회에 등록된 선수 유형은 아마추어 (`"A"`)와 준프로 (`"SP"`)로 나뉜다. 아마추어 선수는 다른 동호회에 소속된 팀과 경기하지 못한다.
  - 한글문제로 주소는 의미없는 영문주소로 채워졌으며 전화번호와 우편번호 역시 형식만 맞고 의미는 없다. 혹시 같은 값이 존재한다면 우연에 의한 일치이다.
- Teams
  - 동호회에는 다수의 팀이 있으며, 각 팀은 4명의 선수로 구성된다.
  - 주장 (Captain) 선수번호만 저장하며 한 선수가 여러 팀의 주장을 맡을 수 있다.
  - 아마추어 선수들의 경기는 기록으로 남기지 않는다. 준프로 선수로 구성된 팀은 리그에 소속되어 있고 `"1-st"`, `"2-nd"`, `"3-rd"`중 하나에 속한다.
- Matches
  - 경기는 3세트일 경우, 2세트를 얻으면 이기는 것이고 5세트의 경우 3세트를 얻어야 이긴다. 각 세트를 어떻게 얻었는지는 기록하지 않는다.

  - 무승부는 없다. 모든 경기는 이기든 지든 둘 중 하나이다.

- Committees
  - 직책은 네 가지: `"Chairman"`, `"Treasurer"`, `"Secretary"`, `"Member"`이며 위원장, 경리, 총무, 평위원을 의미한다.
  - 재직기간은 임명일자 `"YYYY-MM-DD"`가 먼저 나타나고 그 뒤에 해임일자 `"YYYY-MM-DD"`가 반드시 뒤따라온다. 그러나, 입력하는 사람에 따라 일자의 앞/뒤에 붙는 수식어의 차이가 있다. 예를 들어, `"From 2008-05-12 until 2010-12-31"` 처럼 입력하는 사람도 있고, `"Start from 2008-05-12 end at 2010-12-31"`으로 입력하는 사람도 있다.
- Penalities
  - 벌금은 원화금액이다. 소수점 이하 금액이 나타날 수 있는데, 지연이자와 이율등에 의하여 만들어진 값으로 대체로 무시해도 된다.

2 The Problem

제 2장은 시험에서 해결해야 하는 문제를 설명한다. 문제는 크게 두 가지로 분류된다: 첫 번째 부류는 주어진 조건을 만족하는 값 또는 값의 집합을 찾는 문제로 구성되며 다른 한 부류는 주어진 조건을 Plot으로 표현하는 문제이다.

문제를 설명하기 전에 `20xxxxxxx_Final_Exam.py` 화일을 먼저 간략히 살펴본다. 필요한 CSV 화일들은 현재 작업 폴더에 저장하였다고 가정한다. 미리 설정이 가능한 변수들은 이미 적절하게 선언하였으니 (예. `players_path` 등) 변경하지 마시오. 편의를 위하여 Line 16–22 사이에 있는 각 DataFrame에 대응되는 Field들의 Data Type도 미리 정의되어 있고, 그 뒤에 CSV 화일을 읽어 DataFrame을 생성하는 Routine도 미리 만들어져 있다. 각 DataFrame Field의 Data Type (예. `np.str_` 등)은 원하는대로 수정하여 사용할 수 있으며, `read_csv()` 함수의 Argument도 추가 및 수정 가능하다.

```
1  def count_amateur_players(params, ...):
2      # your code
3      pass
4
5  # your code
6
7  def main():
8      # data loading
9      # 1.1 file paths
10     players_path    = "./players.csv"
11     teams_path      = "./teams.csv"
12     matches_path    = "./matches.csv"
13     penalities_path = "./penalties.csv"
14     committees_path = "./committees.csv"
15     # 1.2 user-defined data types
16     dt_players      = {...}
17     dt_teams        = {"TNo": np.int16,
18                        "PNo": np.int16,
19                        "Division": np.str_}
20     dt_matches      = {...}
21     dt_penalties    = {...}
```

```
22      dt_committees    = {...}
23      # 1.3 data frames for each CSV files
24      df_players       = pd.read_csv(players_path, ...)
25      df_teams         = pd.read_csv(teams_path, ...)
26      df_matches       = pd.read_csv(matches_path, ...)
27      df_penalties     = pd.read_csv(penalities_path, ...)
28      df_committees    = pd.read_csv(committees_path, ...)
29
30      # problems
31      # 2.1 an answer to Problem #1
32      res = count_amateur_players(args, ...)
33      print("The number of amateur players: ", res)
34      # your code
35
36      return None
```

각 문제별로 대응되는 함수를 구현해야 한다. Line 32의 `count_amateur_players()`은 문제 #1에 대한 답을 구현하는 함수로 구현시 `args, ...` 부분을 수정하여 사용하시오. 예를 들어, 특정 선수의 이름 `"Kim"`을 Argument로 사용하려면 `count_amateur_players("Kim")`으로 변경하면 된다 – 이것은 예제일뿐이다, 실제 구현과 혼동하지 말라. 물론, Line 1의 함수 선언도 수정해야 한다.

나머지 문제도 모두 유사한 방식이다. 즉, 각 문제마다 구현해야 하는 함수 이름이 주어지므로 해당 함수의 이름으로 함수를 정의하고 `main()` 함수에서 적절히 호출하는 구조로 구현하면 된다.

2.1 The Data Query Problems

DataFrame의 Population과 Aggregation에 **`for`**- 또는 **`while`**-loop을 사용할 수 없다. 추가의 **`import`**는 허용되지 않는다.

#1. 아마추어 선수들의 인원수를 구하는 함수를 구현하시오. (함수 이름이 미리 주어졌기 때문에 생략한다.)

#2. 2부 리그 즉, `"2-nd"`에 속한 팀의 준프로 (`"SP"`)인 주장 선수의 이름, 나이와 주소를 출력하는 함수를 구현하시오.

함수이름으로 `search_sp_captain_2nd_league(params, ...)`을 사용하시오.

#3. 2016년 1월 1일 이후 5번 이상의 징계로 인하여 벌금을 낸 선수들 중에서 가장 많은 벌금을 낸 선수의 이름과 벌금의 총액을 계산하는 함수를 구현하시오.

함수의 이름으로 `search_dirtiest_player(aparamsrgs, ...)`을 사용하시오.

#4. 정규식 (Regular Expression)을 이용하여 경리 (`"Treasurer"`) 업무를 역임한 선수의 이름과 재직기간을 계산하는 함수를 구현하시오. 이때 정규식을 이용하여 재직기간에서 임명된 일자와 해임된 일자를 추출해야 하며, 재직기간은 총재직 일수를 의미한다.

함수의 이름으로 `search_treasurer_players(params, ...)`을 사용하시오.

2.2 The Plotting Problems

Object-Oriented (OO)-Style의 Plot Coding을 사용하시오. 즉 matplotlib을 클릭하여 화면에 나타나는 예제와 같이 `fig, ax = plt.subplots(args, ...)` 형식으로 구현하시오.

#5. 년도별 벌금의 총계를 Bar Chart로 Plotting하는 함수를 구현하시오. Title로 “The Total Amount of Penalities per Year”, x-축의 Label로 “Years”, y-축의 Label은 “Amount (Won)”로 설정하시오. Bar의 색상은 각기 다르게 하고 Legend에 표시하시오.

함수의 이름으로 `plot_penalty_amount(params,...)`를 사용하시오.

References

[1] Rick F. van der Lans: *The SQL Guide to Oracle*. Addision-Wesley Publishing, 1992.

– 마지막 장. 멋진 여름방학 보내세요! –