

Projekt 2

Eksamensprojekt

Program til karaktergivning af studerende

Introduktion

I dette projekt skal du udvikle, teste og dokumentere et program til behandling af karakterer til studerende. Du skal implementere de funktioner og det hoved-script som er beskrevet i den følgende afsnit.

Karakterafrundingsfunktion

| | |
|-------------------|--|
| Interface | <pre>def roundGrade(grades): # Insert your code here return gradesRounded</pre> |
| Input argumenter | grades : En vektor (hvert element er et tal mellem -3 og 12). |
| Output argumenter | gradesRounded : En vektor (hvert element er et tal på 7-trinsskalaen). |
| Bruger-input | Nej. |
| Skærm-output | Nej. |
| Beskrivelse | Funktionen skal afrunde hvert element i vektoren grades og returnere den nærmeste karakter på 7-trinsskalaen: |

| | | | | | | | |
|----------------------------|----|----|---|---|----|----|------|
| 7-trinsskalaen: Karakterer | 12 | 10 | 7 | 4 | 02 | 00 | -3 |
|----------------------------|----|----|---|---|----|----|------|

For eksempel, hvis funktionen får vektoren $[8.2, -0.5]$ som input, skal den returnere de afrundede karakterer $[7, 0]$, som er de nærmeste tal på karakterskalaen.

Endelig karakter-funktion

| | |
|-------------------|--|
| Interface | <pre>def computeFinalGrades(grades): # Insert your code here return gradesFinal</pre> |
| Input argumenter | grades : En $N \times M$ matrix som indeholder karakterer på 7-trinsskalaen givet til N studerende for M opgaver. |
| Output argumenter | gradesFinal : En vektor med længde N som indeholder den endelige karakter for hver af de N studerende. |
| Bruger-input | Nej. |
| Skærm-output | Nej. |
| Beskrivelse | For hver studerende skal den endelige karakter beregnes på følgende måde: |

1. Hvis der kun er en opgave ($M = 1$) er den endelige karakter lig karakteren for den opgave.
2. Hvis der er to eller flere opgaver ($M > 1$) skal den laveste karakter ikke medregnes. Den endelige karakter beregnes som gennemsnittet af de $M-1$ højeste karakterer, afrundet til nærmeste karakter på skalaen (vha. funktionen **roundGrade**).
3. Uanset ovenstående skal den endelige karakter altid være -3 , hvis den studerende har fået -3 i en eller flere af opgaverne.

Karakter plot-funktion

| | |
|-------------------|---|
| Interface | <pre>def gradesPlot(grades): # Insert your code here</pre> |
| Input argumenter | grades : En $N \times M$ matrix som indeholder karakterer på 7-trinsskalaen givet til N studerende for M opgaver. |
| Output argumenter | Nej. |
| Bruger-input | Nej. |
| Skærm-output | Ja (diagrammer, se specifikationer nedenfor.) |
| Beskrivelse | <p>Denne funktion skal generere og vise to diagrammer:</p> <ol style="list-style-type: none"> 1. "Final grades": Et søjediagram af antallet af studerende som har fået hver af de mulige endelige karakterer på 7-trinsskalaen (beregnet med funktionen <code>computeFinalGrades</code>). 2. "Grades per assignment": Et plot med opgaverne på x-aksen og karaktererne på y-aksen. X-aksen skal vise alle opgaver 1 til M og y-aksen skal vise alle karakterer fra -3 til 12. Plottet skal indeholde: <ol style="list-style-type: none"> 1. Hver af de afgivne karakterer markeret med en prik. Du skal lægge et lille tilfældigt tal (mellem -0.1 og 0.1) til x- og y-koordinatet for hver prik, for at kunne skelne de forskellige prikker, som ellers ville ligge oven i hinanden når mere end en studerende har fået samme karakter i den samme opgave. 2. Den gennemsnitlige karakter for hver opgave vist som en linje. |

Diagrammerne skal indeholde passende titel, beskrivende akse-betegnelser og en signaturforklaring hvor det er hensigtsmæssigt. Du må gerne præsentere diagrammerne i separate figur-vinduer eller som under-plot i et enkelt figur-vindue.

Hoved-script

| | |
|-------------------|--|
| Interface | Skal implementeres som et script. |
| Input argumenter | Nej. |
| Output argumenter | Nej. |
| Bruger-input | Ja (se specifikationer nedenfor.) |
| Skærm-output | Ja (se specifikationer nedenfor.) |
| Beskrivelse | <p>Når brugeren kører hoved-scriptet skal han/hun bedes om at indtaste navnet på en komma-separeret (CSV) datafil som indeholder karakter givet til et antal studerende for et antal opgaver (se beskrivelse af filformat nedenfor). Husk at checke at filnavnet er gyldigt. Efter at have indlæst datafilen, skal du vise information om det indlæste data: Som minimum skal antallet af studerende og antallet af opgaver vises.</p> |

Derefter skal brugeren som minimum have følgende muligheder:

1. Indlæs ny data.
2. Check for datafejl.
3. Generer diagrammer.
4. Vis karakterliste.
5. Afslut.

Det skal være muligt for brugeren at udføre disse handlinger (se specifikationer nedenfor) i vilkårlig rækkefølge så længe han/hun ønsker det, indtil brugeren vælger at afslutte programmet. Detaljerne vedrørende hvordan hoved-scriptet skal implementeres skal du selv bestemme. Det er et krav at programmet er interaktivt, og du skal tilstræbe at gøre programmet brugervenligt ved at give fornuftige valgmuligheder. Tænk på hvad du ville forvente hvis du skulle bruge sådan et script, og på hvad du synes kunne være sjovt. Leg med implementationen og lav et cool script.

Filformat

Den første række i CSV-filen indeholder en overskrift for hver søjle. Hver af de følgende linjer indeholder et studie-id, navn samt et antal karakterer for en studerende. Et eksempel på en datafil med fire studerende og tre opgaver er givet nedenfor:

```
StudentID,Name,Assignment1,Assignment2,Assignment3
s123456,Michael Andersen,7,7,4
s123789,Bettina Petersen,12,10,10
s123468,Thomas Nielsen,-3,7,2
s123579,Marie Hansen,10,12,12
```

Husk at dit program skal fungere for et vilkårligt antal studerende og et vilkårligt antal opgaver. Du opfordres til at lave din egen test-datafil for at kontrollere at dit program fungerer korrekt.

Fejlhåndtering

Du skal teste at alle input givet af brugeren er gyldige. Hvis brugeren giver et ugyldigt input, skal du give informativ feedback til brugeren og gøre det muligt at indtaste et nyt gyldigt input.

1. Indlæs ny data

Hvis brugeren vælger at indlæse ny data, skal brugeren bedes om at indtaste et gyldigt filnavn på en data-fil, og data skal indlæses på samme måde som i starten af scriptet.

2. Check for datafejl

Hvis brugeren vælger at checke for datafejl, skal du vise en rapport over fejl (hvis der er nogen) i den indlæste datafil. Dit program skal som minimum kunne detektere og vise information om følgende mulige fejl:

1. Hvis to studerende i datasættet har samme studie-id.
2. Hvis en karakter i datasættet ikke er en af de mulige karakterer på 7-trinsskalaen.

3. Generer diagrammer

Hvis brugeren vælger at generere diagrammer, skal funktionen `gradesPlot` kaldes for at vise plottene.

4. Vis karakterliste

Hvis brugeren vælger at vise karakterlisten, skal du vise karaktererne for hver opgave samt den endelige karakter for alle de studerende i alfabetisk orden efter deres navn. Den viste liste skal formateres på en måde så den er let at aflæse.

5. Afslut

Hvis brugeren vælger at afslutte programmet, skal hoved-scriptet stoppe.

