

Bakterie-dataanalyse

20. januar 2014

INDHOLD

1	INTRODUKTION	1
1.1	OVERBLIK OVER KRAV	1
2	PROGRAM-SPECIFIKATIONER	1
2.1	DATA LOAD FUNKTION	3
2.2	DATA STATISTIK FUNKTION	4
2.3	DATA PLOT FUNKTION	5
2.4	HOVED-SCRIPT	6
3	RAPPORT-SPECIFIKATIONER	8
4	EVALUERING	9
5	AFLEVERING	9

1 INTRODUKTION

I dette projekt skal du udvikle, teste og dokumentere et program. Den overordnede ide bag dette projekt er at teste din forståelse af de programmerings-værktøjer og -metoder som du har lært gennem kurset, og at gøre dette på en sjov og berigende måde. Det er dit ansvar, gennem din kode og rapport, at vise at du opfylder læringsmålene (se DTU Kursusbasen). Din kode og rapport vil blive evalueret efter de kriterier der er beskrevet i afsnit 4. Du skal aflevere dit projekt i overensstemmelse med de kriterier der er beskrevet i afsnit 5 inden den deadline der fremgår af CampusNet (Assignments). Afleveringer der modtages efter deadline vil ikke blive accepteret.

1.1 OVERBLIK OVER KRAV

Program Programmet skal implementeres ifølge de specifikationer der er givet i afsnit 2. Det er tilladt at implementere programmet alene eller **i grupper** af to eller tre studerende.

Rapport Rapporten skal skrives ifølge de specifikationer der er givet i afsnit 3. Rapporten skal skrives **individuelt**.

2 PROGRAM-SPECIFIKATIONER

I det følgende vil specifikationerne for programmet blive beskrevet, dvs. de påkrævede funktioner og hoved-script. Bemærk at specifikationerne ikke er udtømmende beskrevet, og at en del af opgaven er at identificere, løse, implementere og dokumentere dele af programmet. Dette er valgt, fordi vi ønsker

at se jer træffe jeres egne valg omkring håndteringen af de problemer i møder. Det er fuldt ud tilladt, at I træffer jeres egne valg når noget ikke er beskrevet i specifikationerne. Bemærk dog at jeres valg skal dokumenteres og jeres argumenter for at træffe disse valg vil blive evalueret.

Overordnede krav

- Alle funktioner skal implementeres ifølge det givne interface og må ikke implementeres som scripts. Det er tilladt at benytte under-funktioner og indbyggede funktioner inden i en funktion. Du må også gerne udvikle “hjælpe”-funktioner som kaldes inde fra en funktion, hvis du mener det giver en mere elegant eller praktisk løsning. Du skal sikre dig at alle designvalg dokumenteres i rapporten.
- Hoved-scriptet skal implementeres ifølge de givne specifikationer. Det er ikke tilladt at implementere hoved-scriptet som en funktion.
- Det er ikke tilladt at kalde et script inde fra et andet script eller fra en funktion. Du må kun implementere ét script, nemlig hoved-scriptet.
- Du må gerne benytte indbyggede funktioner, og du opfordres til at undersøge om der findes indbyggede funktioner som du kan bruge i din egen kode og dokumentere dette i rapporten.
- Alle funktioner skal indeholde en passende hjælpe-tekst som beskriver formålet med funktionen, input og output argumenter, hvordan funktionen benyttes samt anden information der er nødvendig for at forstå og benytte funktionen. Det anbefales at benytte de hjælpe-tekster som findes for de indbyggede funktioner til inspiration.
- Overdreven brug af fejlhåndtering inde i en funktion kan gøre funktionen unødvendigt langsom, særligt hvis funktionen kaldes ofte. Derfor er det ofte den bedste praksis at antage at alle input argumenter til en funktion er gyldige (og angive i funktionens hjælpe-tekst hvilke krav der er til argumenterne.) Bemærk at dette ikke gælder for interaktive bruger-input. Interaktive bruger-input skal fejl-checkes i forbindelse med at de indlæses.

Dine egne udvidelser

- Du opfordres til at udvide funktionaliteten af de specificerede funktioner, hvis det lykkes at færdiggøre de specificerede krav til funktioner, script og rapport inden deadline. På denne måde får du mere programmeringserfaring, og du kan yderligere illustrerer omfanget af din programmeringsforståelse. Bemærk: Vælger du at udvide funktionerne, er det tilladt at udvide de specificerede interfaces, givet at det stadig er muligt at kalde funktionerne som beskrevet. Husk at dokumentere dit ekstraarbejde i rapporten.

2.1 DATA LOAD FUNKTION

Interface	<pre>function data = dataLoad(filename) % Insert your code here</pre>
Input argumenter	filename : En tekst-streng som indeholder filnavnet på en datafil.
Output argumenter	data : En $N \times 3$ matrix.
Bruger-input	Nej.
Skærm-output	Ja (fejl-beskeder, se specifikationerne nedenfor.)
Beskrivelse	Funktionen skal indlæse data fra datafilen filename . Hver linje i datafilen indeholder de følgende felter:

Temperature, **Growth rate**, og **Bacteria**.

Felterne indeholder numeriske værdier og er separeret af et mellemrum. Det følgende illustrerer et eksempel på et uddrag af en sådan datafil:

```
25  0.109  1
20  0.096  2
15  0.517  3
35  1.086  4
40  0.934  2
35  0.109  1
:
```

Feltet **Bacteria** indeholder en numerisk kode som svarer til et af de følgende bakterie-navne:

Bacteria	Navn
1	Salmonella enterica
2	Bacillus cereus
3	Listeria
4	Brochothrix thermosphacta

Data fra filen skal gemmes i en $N \times 3$ matrix kaldet **data**, hvor N er antallet af gyldige rækker i datafilen.

Håndtering af data-fejl Der kan være en eller flere fejlbehæftede (ugyldige) linjer i datafilen, hvilket funktionen skal håndtere. Hvis data load funktionen detekterer en fejlbehæftet linje i datafilen skal den springe denne linje over, skrive en fejlbesked på skærmen og fortsætte til den næste linje. Funktionen skal kun returnere data svarende til de gyldige linjer. Fejlbeskeden skal beskrive i hvilken linje fejlen var og hvad fejlen bestod i. Funktionen skal checke følgende:

- **Temperature** skal være et tal mellem 10 og 60.
- **Growth rate** skal være et positivt tal.
- **Bacteria** skal være en af de fire beskrevet i tabellen ovenfor.

2.2 DATA STATISTIK FUNKTION

Interface `function result = dataStatistics(data, statistic)`
`% Insert your code here`

Input argumenter `data`: En $N \times 3$ matrix med søjlerne Temperature, Growth rate, og Bacteria.
`statistic`: En tekst-streng som specificerer den statistik som skal beregnes.

Output argumenter `result`: En skalar indeholdende den beregnede statistik.

Bruger-input Nej.

Skærm-output Nej.

Beskrivelse Denne funktion skal beregne en af et antal mulige statistikker baseret på data. En “statistik” skal her forstås som et enkelt tal som beskriver et aspekt ved data, som for eksempel en middelværdi (et gennemsnit). Hvilken statistik der skal beregnes afhænger af værdien af tekst-strengen `statistic`. Den følgende tabel viser de forskellige mulige værdier af `statistic` og en beskrivelse af beregningen af de tilsvarende statistikker.

<code>statistic</code>	Beskrivelse
<code>'Mean Temperature'</code>	Middelværdi (gennemsnit) af Temperature.
<code>'Mean Growth rate'</code>	Middelværdi (gennemsnit) af Growth rate.
<code>'Std Temperature'</code>	Standard-afvigelse af Temperature.
<code>'Std Growth rate'</code>	Standard-afvigelse af Growth rate.
<code>'Rows'</code>	Det totale antal rækker i data.
<code>'Mean Cold Growth rate'</code>	Middelværdi (gennemsnit) af Growth rate hvor Temperature er mindre end 20 grader.
<code>'Mean Hot Growth rate'</code>	Middelværdi (gennemsnit) af Growth rate hvor Temperature er større end 50 grader.

Du opfordres til at benytte indbyggede funktioner til at beregne disse statistikker hvor det er muligt.

2.3 DATA PLOT FUNKTION

Interface	<code>function dataPlot(data)</code> <code>% Insert your code here</code>
Input argumenter	<code>data</code> : En $N \times 3$ matrix med søjlerne Temperature, Growth rate, og Bacteria.
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. "Number of bacteria": Et søjlediagram af antallet af hver af de forskellige typer Bacteria i data.2. "Growth rate by temperature": Et diagram med Temperature på x-aksen og Growth rate på y-aksen. X-aksen skal gå fra 10 til 60 grader og y-aksen skal starte ved 0. Plottet skal have ét koordinatsystem med fire grafer, en for hver type Bacteria. De forskellige grafer skal kunne skelnes fra hinanden for eksempel ved brug af forskellige farver, markører eller linjetyper. <p>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.</p>

2.4 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>Brugeren af data-analyse programmet interagerer med programmet gennem hoved-scriptet. Når brugeren kører hoved-scriptet skal han/hun som minimum have følgende muligheder:</p> <ol style="list-style-type: none">1. Indlæs data.2. Filtrer data.3. Vis statistik.4. Generer diagrammer.5. Afslut. <p>Det skal være muligt for brugeren at udføre disse handlinger (se specifikationer nedenfor) i vilkårlig rækkefølge så længe han/hun ønsker det, indtil han/hun 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.</p>
Fejlhåndtering	<p>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.</p> <p>Det må ikke være muligt at vise statistikker eller generere diagrammer før data er indlæst. Hvis brugeren forsøger at gøre dette skal han/hun have passende feedback der forklarer at dette ikke er muligt.</p>
1. Indlæs data	Hvis brugeren vælger at indlæse data, skal du bede brugeren om at indtaste et filnavn på en data-fil. Husk at checke at filnavnet er gyldigt. Indlæs data ved hjælp af <code>dataLoad</code> funktionen, som vil vise information om eventuelle fejlbehæftede linjer i datafilen.
2. Filter data	<p>Hvis brugeren vælger at filtrere data, skal han/hun have mulighed for at specificere en eller flere betingelser som skal være opfyldt for at en data-række inkluderes i beregningen af statistikker og generering af diagrammer. Som minimum skal brugeren kunne specificere to typer af filtre:</p> <ol style="list-style-type: none">1. Et filter for typen af Bacteria, for eksempel <code>Bacteria=Listeria</code>.2. Et interval-filter for Growth rate, for eksempel $0.5 \leq \text{Growth rate} \leq 1$.

Når et sådant filter er specificeret, skal statistikker og diagrammer genereres på baggrund af den delmængde af data-rækker hvor filterets kriterier er opfyldt. Brugeren skal også have mulighed for at deaktivere filterbetingelserne, og når han/hun gør dette skal efterfølgende statistikker og diagrammer igen genereres på baggrund af alle data-rækker. Hvis et filter er aktivt skal information om filteret hele tiden være synligt i programmets brugerinterface.

3. Vis statistik

Hvis brugeren vælger at vise statistikker, skal du spørge brugeren hvilken statistik der skal vises. Brug funktionen `dataStatistics` til at beregne den ønskede statistik og vis den på skærmen sammen med en beskrivelse af statistikken. Hvis et filter er aktivt, skal statistikken beregnes udelukkende baseret på de data-rækker det opfylder filterets betingelser.

4. Generer diagrammer

Hvis brugeren vælger at generere diagrammer, skal funktionen `dataPlot` kaldes for at vise plottene. Hvis et filter er aktivt, skal plottene genereres udelukkende på baseret på de data-rækker der opfylder filterets betingelser.

5. Afslut

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

Data fil

Du opfordres til at lave din egen test-datafil for at kontrollere at dit program fungerer korrekt. Det er vigtigt at du også sikrer og dokumenterer at programmet virker korrekt i tilfælde af fejlbehæftede linjer i datafilen som beskrevet i afsnit 2.1.

3 RAPPORT-SPECIFIKATIONER

Rapportens formål er at forklare hvordan dit program bruges (brugerperspektivet, som kan opfattes som en brugervejledning), og hvorfor det er designet og implementeret, som det er (udviklerperspektivet, hvor du dokumenterer og diskuterer dine design- og implementationsvalg.)

Teksten der følger er tiltænkt som en vejledning i, hvad din rapport skal indeholde. Der bruges en generelt accepteret model for god rapportstruktur, men den er ikke endegyldig. Du skal gøre dit bedste for at skrive en god, præcis, kortfattet og detaljeret rapport i et læsbart format og sprog og samtidig sikre at du har dækket de overordnede opgaver beskrevet ovenfor.

Sprog: Rapporten skal skrives på dansk eller engelsk.

Længde: Rapporten skal fylde 5–10 sider.

Indhold: Rapporten skal indeholde følgende:

1. **En forside** med projektets titel, abstract, kursusnavn og -nummer, dato, dit navn og studienummer. Hvis du arbejder i en gruppe skal navne og studienumre for alle gruppemedlemmer fremgå.
2. **Et abstract** (max. 10 linjer) eller resume skal være på forsiden og skal give læseren et hurtigt overblik over hvad han/hun bliver præsenteret for i din rapport. For flere detaljer om at skrive et abstract, se for eksempel [http://en.wikipedia.org/wiki/Abstract_\(summary\)](http://en.wikipedia.org/wiki/Abstract_(summary)).
3. **En introduktion** (ca. $\frac{1}{2}$ side) hvor du beskriver formålet med og motivationen bag dette projekt i flere detaljer og kort skitserer din tilgang til projektet.
4. **Detaljer vedr. design, implementation og brug af funktioner og script** (ca. 1 side per funktion/script). Dette inkluderer, men er ikke begrænset til:
 - *Funktion interfaces (design):* Funktions navn, hvordan funktionen kan kaldes samt dens input- og outputargumenter. Diskuter såvidt muligt disse interfaces: Kunne du have designet dem anderledes sammenlignet med de givne specifikationer. Hvis ja, hvordan? Hvad ville du have gjort anderledes?
 - *Funktion og hoved-script implementation:* Valg af variable (type, begrundelse for type og formål) kodens flow og struktur (hvordan løser funktionen problemet trin for trin?) Uddyb valg vedrørende vektorisering, løkker og branching.
 - *Brug af hoved-script:* Forklaring af hvordan dit hoved-script benyttes. Tænk på dette som en brugervejledning, og sørg for at dække alle mulige situationer som en bruger kan opleve mens han/hun benytter dit program.
5. **En diskussion** (ca. 1 side) hvor du reflekterer over det udførte arbejde og den løsning du har lavet. Diskuter begrænsninger og simplifikationer i løsningen såvel som mulige ideer til videre udvikling.
6. **En konklusion** (ca. $\frac{1}{2}$ side) hvor du sammenfatter hvad du har opnået i dette projekt. Det anbefales at koble konklusionen sammen med læringsmålene.

4 EVALUERING

Vi benytter de følgende kriterier i bedømmelsen af dit arbejde.

- Opfyldelse af læringsmålene, som specificeret i kursusbasen (<http://www.kurser.dtu.dk>) koblet til pensum.
- Helheden og kvaliteten af jeres program-implementation evalueret i forhold til projektkravene, læringsmålene samt “best practice” programming, som beskrevet i modul-noterne.
- Den individuelle rapport. Læringsmålene evalueres primært baseret på den individuelle rapport, da den fungerer som det individuelle element af eksaminationen.

I bør tolke denne liste som følger: Målet er at alle læringsmål er opfyldt. Det er et krav, at I løser projekt givet projektspecifikationen, men selv om jeres løsning er af top-kvalitet, er I ikke garanteret en høj karakter, hvis rapporten er mangelfuld i indhold og kvalitet. Grundlaget for dette er, at vi skal kunne evaluere jeres individuelle præstation (rapporten) for at give jer en passende karakter. Omvendt, hvis I skriver en fantastisk rapport, men afleverer ingen eller dårlig kode, kan vi ikke vurdere jeres evne til at omsætte teori til praksis (konceptuel programming relateret til aktuel implementation). I bør prioritere jeres tid, og I rådes til at stoppe implementationen i fornuftig tid inden deadline, således at I har tid nok til at skrive en god rapport. Hvis I har tid tilovers efter rapportskrivning, tilrådes I at vende tilbage til koden, rette fejl og udbygge funktionalitet, såvel som at rette eventuelle “kosmetiske” problemer i programmet.

5 AFLEVERING

Før deadline (se CampusNet Assignments) skal du individuelt indsende/uploade:

- Program** Alle implementerede kode-filer (hoved-script og funktioner) samt andre nødvendige filer i en pakket zip-fil. Husk at teste at script og funktioner kører fejlfrit inden aflevering, da en del af vores vurderingsgrundlag er kørsel og test af programmet.
- Rapport** Den individuelle rapport som en enkelt pdf-fil. Rapporter accepteres udelukkende som pdf-filer.