

Opg1.1

```
data a; *danner et datasæt kaldet a i det midlertidige bibliotek work;
array hovsa(3) d1 e1 f1; *array laver 3 variable med navnene d1, e1 og f1, som
kan kaldes til ved fx at skrive hovsa(1), så kaldes på obs=i, d1;
do i=1 to 3; *laver en do løkke hvor "i" løber fra 1 til 3, samtidig med at der
laves en ny variabel, i, med tallet i=1,2,3 som observation alt efter hvor i do
løkken koden er;
hovsa(i)=ranuni(12345); *hovsa kalder én af variabelnavnene i arrayen alt efter
hvor "i" er i do løkken.
Dvs hvis i=1, så skriver linjen et random tal mellem 0 og 1 i d1. Dette tal
løber ned gennem observationerne i d1
Tallet 12345 er et såkaldt seed. I dette tilfælde opnås amme resultater hver gang koden afvikles. Hvis dette
seed havde været nul ville tilfældigheds generatoren nulstilles efter et ur,
således at der kommer nye resultater ved hver kørsel;
put e1; *skriver værdien af variabelen e1 (den er dannet via ranuni gennem arrayet) i logvinduet;
x=hovsa(i)**hovsa(1); *laver en ny variabel x, som er lig tallet fra kodelinje
4, "hovsa(i)...", opløftet med hvad der står i d1;
output; *alle variabelværdier outputtes til en ny observation;
if i=3 then put d1; *skriver talværdien for d1 i logvinduet, men kun hvis i=3;
if i=2 then output; *når i=2, så outputtes alle talværdier i en ny observation. Der dannes kun nye
observationer, når der er en output-statement;
end; *ender løkken;
run; *kører programmet;
proc print; *printer resultatet af programmet;
run; *kører programmet;
```

Opg1.2

*en af outputsætningerne skal fjernes, så datasættet ikke får for mange observationer;

```
data a;
array hovsa(3) d1 e1 f1;
do i=1 to 3;
hovsa(i)=ranuni(12345);
put e1;
x=hovsa(i)**hovsa(1);
output;
end;
drop d1 e1 f1; *dette fjerner mellemregningsvariablene d1 e1 og f1 fra datasættet;
run;
proc print;
run;
```

Opg 2.1

```
proc sort data=sasprog.sasprog_olympics out=olympics;
by land;
run;
data a;
set olympics;
drop h gren sex;
run;
proc means data=olympics mean;
var v;
By land;
run;
```

opg 2.2

```
proc sort data=olympics;
by gren sex;
run;
data a;
set olympics;
drop land h;
run;
proc means data=olympics mean;
var v;
by gren sex;
run;
```

opg 3.1

```
data a;
input ord : $20. @@;*her antages at alle ord højst er 20 karakterer lange;
cards;
Københavns Universitet inviterer dig helt tæt på forskningens
forunderlige verden når både Rådhuspladsen og Frue Plads i
to dage omdannes til Forskertorv med masser af boder og aktiviteter
for nysgerrige børn og voksne.
;
run;
proc print;
run;
```

opg 3.2

```
data b;
set a;
if index(lowercase(ord),'s')>0 then tal2+1;
*her testes på om ordet indeholder et lille s - vel at mærke efter at ordet er konverteret til små bogstaver,
så alle s-er fremhæves. Funktionen index angiver et nul hvis der ikke er et s og ellers pladsnummeret for den
første forekomst af et s. Hvis der er et stort eller lille s, tillægges tælleren tal
et et-tal. for den sidste observation får derfor tal=10, dvs at der 10 ord i teksten med mindst et s.
run;
```

En proc print giver for de sidste tre obs:

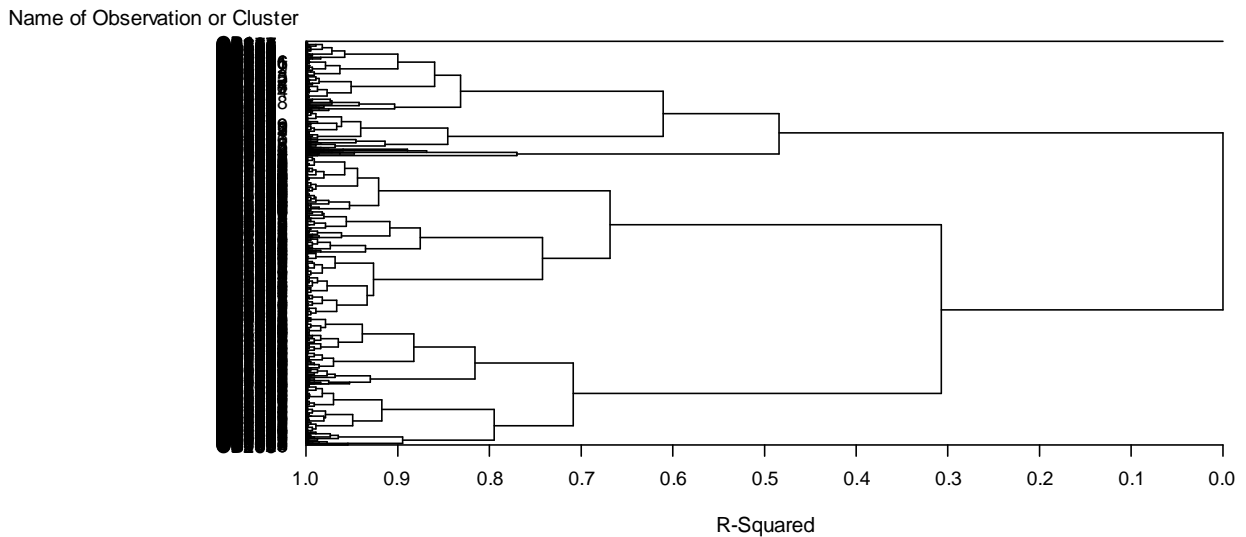
31	børn	9
32	og	9
33	voksne.	10

Opg 4

***denne procedure give ren fornuftig standardisering
men den er nu ikke vigtig for dette datasæt;**

```
proc aceclus data=Olympics out=Ace p=.03 noprint;
var h v;
run;
proc cluster data=Ace method=ward ccc pseudo print=15 outtree=Tree;
var can1 can2;
run;
proc tree data=Tree out=New nclusters=3
haxis=axis1 horizontal;
height _rsq_;
copy can1 can2;
```

run;



Plottet viser at der er et meget lille cluster allerøverst. Går man lodret op på grafen ud for $R^2=0.4$ ser man at den krydser tre streger (plus altså den øverste). Dvs at der er tre stort set lige store klustre, udover det meget lille øverst

Opg 5

```
*Fremgangsmåden kopierer en fremgangsmåde i et eksempel i SAS-  
hjælpen;  
proc standard data=sasprog.SASProg_Olympics out=stand mean=0  
std=1;  
var h v;  
run;  
*dette kald af fastclus fastlæger mange potentielle centrer for  
clustre;  
proc fastclus data=stand outseed=mean1 maxc=20 maxiter=0;  
var h v;  
run;  
*her udvælges clustre med mere end 100 medlemmer  
i SAS hjælpen sættes grænsen ved 100, men det er for få i et  
datasæt med så mange observationer;  
data seed;  
set mean1;
```

```

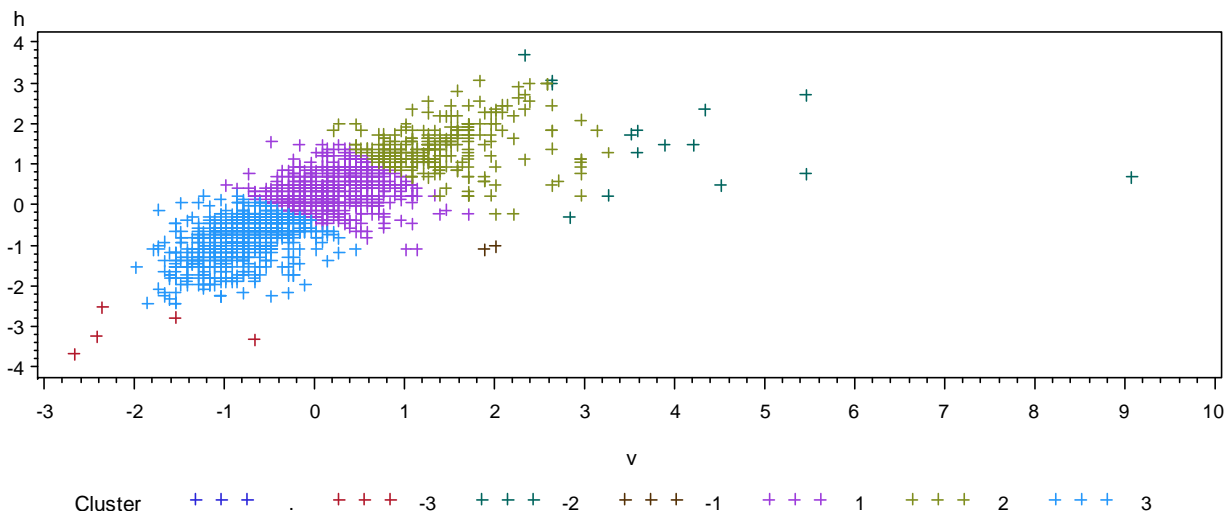
    if _freq_>100;
run;
*nu laves tre clustrer ud fra de potentielle centrer. Optionen
least=1 sikrer, at kun observationer nærmere end 1 (husk at der er
standardiseret) fra centret tilordnes centret. Da der forlanges
tre clustre vil visse punkter ikke opfylde dette krav. Disse
punkter karakteriseres som outliers og får et negativt
klusternummer, hvor tallet ellers viser, hvilket kluster det
ellers ligger tættest på.;
proc fastclus data=stand seed=seed strict=2 maxc=3 out=ny;
var h v;
run;
proc sort data=ny;
by cluster;
run;
proc means;
var h v;
by cluster;
run;
proc gplot;
plot h*v=cluster;
run;
proc print;
var cluster gren h v;
where cluster<0;
run;

```

Det es at der især er outliers med meget høj vægt i klusteret for de tungeste. Det er vægtløftere, kuglestødere etc.

Obs	CLUSTER	gren	h	v
1	.		.	.
2	.		.	.
3	.		.	.
4	-3	Men's 62kg	-3.31788	-0.66543
5	-3	Women's Qualifications	-2.52055	-2.35189
6	-3	Women's Qualifications	-3.22929	-2.41436
7	-3	Women's 48kg	-2.78633	-1.53989
8	-3	Women's Qualifications	-3.67225	-2.66420
9	-2	Men's Shot Put	1.28891	3.58196
10	-2	Men's Basketball	3.06076	2.64503
11	-2	Men's Shot Put	1.82047	3.58196
12	-2	Men's +105kg	0.49158	4.51888
13	-2	Men's Volleyball	2.97216	2.64503
14	-2	Men's Trap	-0.30575	2.83242
15	-2	Men's +105kg	0.75736	5.45581
16	-2	Men's Shot Put	2.35202	4.33150
17	-2	Men's +100kg	1.46610	4.20657
18	-2	Men's Discus Throw	1.73187	3.51950
19	-2	Men's +100kg	0.22581	3.26965
20	-2	Men's +100kg	1.46610	3.89427
21	-2	Men's Discus Throw	2.70639	5.45581
22	-2	Men's Basketball	3.68090	2.33273

23	-2	Men's +100kg	0.66877	9.07858
24	-1	Women's +75kg	-1.10308	1.89550
25	-1	Women's +78kg	-1.01448	2.02042



Opg 6

```
proc discrim data=sasprog.SASProg_Olympics_female outstat=stat
method=normal pool=yes
list crossvalidate;
class gren;
var h v;
run;
```

Dets ses at man er 100% sikker på hvor diskoskasterne hører til, mens der er en lidt større tvivl om højdespringere og maratonløbere. da der er mange maratonløbere er det klogest at klassificere flest muligt i den gren

Posterior Probability of Membership in gren					
Obs	From gren	Classified into gren	Women's Discus Throw	Women's High Jump	Women's Marathon
1	Women's Marathon	Women's Marathon	0.0000	0.0474	0.9526
2	Women's Marathon	Women's Marathon	0.0000	0.0684	0.9316
3	Women's Discus Throw	Women's Discus Throw	1.0000	0.0000	0.0000
4	Women's Marathon	Women's Marathon	0.0000	0.1704	0.8296
5	Women's Marathon	Women's Marathon	0.0000	0.1436	0.8564
6	Women's Marathon	Women's High Jump *	0.0000	0.9451	0.0549
7	Women's Marathon	Women's High Jump *	0.0000	0.6467	0.3533
8	Women's Discus Throw	Women's Discus Throw *	1.0000	0.0000	0.0000
9	Women's Marathon	Women's Marathon	0.0000	0.2700	0.7300
10	Women's Marathon	Women's Marathon	0.0000	0.1866	0.8134
11	Women's Marathon	Women's Marathon	0.0000	0.1554	0.8446
12	Women's High Jump	Women's High Jump	0.0000	0.7146	0.2854
13	Women's High Jump	Women's High Jump	0.0000	0.7981	0.2019

14	Women's Marathon	Women's Marathon	0.0000	0.0289	0.9711
15	Women's Marathon	Women's High Jump *	0.0000	0.6548	0.3452
16	Women's Discus Throw	Women's Discus Throw	1.0000	0.0000	0.0000
17	Women's Discus Throw	Women's Discus Throw	1.0000	0.0000	0.0000
18	Women's Marathon	Women's Marathon	0.0000	0.1626	0.8374
19	Women's Marathon	Women's Marathon	0.0000	0.1554	0.8446
20	Women's Marathon	Women's Marathon	0.0000	0.3440	0.6560
21	Women's Marathon	Women's Marathon	0.0000	0.1531	0.8469
22	Women's Marathon	Women's Marathon	0.0000	0.4625	0.5375
23	Women's Marathon	Women's Marathon	0.0000	0.1286	0.8714
24	Women's Marathon	Women's Marathon	0.0000	0.4220	0.5780

* Misclassified observation

From gren	Women's Discus Throw	Women's High Jump	Women's Marathon	Total
Women's Discus Throw	4 100.00	0 0.00	0 0.00	4 100.00
Women's High Jump	0 0.00	0 0.00	2 100.00	2 100.00
Women's Marathon	0 0.00	3 16.67	15 83.33	18 100.00
Total	4 16.67	3 12.50	17 70.83	24 100.00

```
proc discrim data=stat testdata=sasprog.SASProg_Olympics_female_football
testlist pool=no;
class gren;
var h v;
run;
```

Det ses at alle fodboldspillere har kroppe som højdespringere - måske kunne man have forventet at enkelte ville ligne maratonløbere, men det er ikke så overraskende, at de ikke ligner diskoskastere.

The DISCRIM Procedure
Classification Summary for Test Data:
SASPROG.SASPROG_OLYMPICS_FEMALE_FOOTBALL
Classification Summary using Linear Discriminant Function

Number of Observations and Percent Classified into gren

From gren	Women's Discus Throw	Women's High Jump	Women's Marathon	Total
Total	0 0.00	39 100.00	0 0.00	39 100.00