

SAS-intro

I SAS finns 3 olika fönster; "Log", "Editor" och "Output". I log-fönstret registreras allt som vi skriver och säger åt SAS att göra. Här kan man exempelvis se vilka procedurer man har kört, hur lång tid det tog och om det har blivit något fel. Om texten i "loggen" är blå är allt ok och om texten är grön betyder det att man fått en varning. Om texten är röd är något fel och SAS kan då inte köra koden som vi skrivit förän vi rättat till felet.

I Editor skriver vi all kod. Det är där vi läser in datat och skriver koden till de procedurer vi vill köra. För att tala om för SAS att koden ska köras tycker man "submit". Antingen använder man F8-tangenten eller klickar på den lilla gubben uppe till höger när man är inne i SAS-programmet. Man kan köra all kod man skrivit samtidigt, eller genom att markera bitar av koden, köra endast den markerade koden. I output fönstret får vi resultaten av de procedurer vi kör.

SAS jobbar med olika "steg". Det finns två typer av steg vi kan skriva i Editor; DATA-steg och PROC-steg. När man använder DATA steget läser man in datat i ett format som SAS kan använda. PROC-steget använder procedurer för att göra något med datat; skriva ut det i output eller analysera det på något vis. Varje steg måste börja med DATA eller PROC.

Exempel - Så här läser vi in data:

```
data work.langd;  
input namn$ langd;  
datalines;  
Karin 162  
Daniel 192  
Maria 173  
Anna 155  
Bertil 181  
;  
run;
```

I det här exemplet ser vi att varje rad i ett steg ska avslutas med ett semikolon. (Dock inte efter de radvisa observationerna, då sätter vi ett semikolon efter sista raden.) För att läsa in data behöver vi först och främst "data" kommandot, vi behöver "input" där vi ger våra variabler namn och vi skriver "datalines" för att tala om för SAS att våra observationer kommer att läsas in. Till sist behöver vi också skriva "run" för att SAS ska köra koden som vi skrivit. När vi kör den här koden kommer vi inte att få någon "output" utan SAS bara lagrar våra variabler och observationer.

Ett SAS namn innehåller två delar; den första delen - i detta exemplet "work" - är namnet på biblioteket där jag vill spara datat. Man kan se det som en plats på datorn som man ger ett namn. Efter biblioteksnamnet sätter vi en punkt och

sedan kommer filnamnet Jag kallar den här filen för "langd" som i längd (SAS kan inte läsa å, ä eller ö). Biblioteksnamnet "work" är en tillfällig plats som rensas varje dag. Man kan även skapa egna bibliotek som man själv namnger. Vi kommer dock jobba genomgående med biblioteksnamnet "work".

En annan sak som vi kan lära oss av exemplet ovan är att om vi sätter ett dollar tecken, \$, efter variabelnamnet förstår SAS att det är en kvalitativ variabel och tillåter alltså text istället för siffror.

Om vi nu vill titta på det vi sparat kan vi använda "proc print", det ser ut så här:

```
proc print data=work.langd;  
run;
```

Och vad vi får i vårt "output-fönster" är detta:

Obs	namn	langd
1	Karin	162
2	Daniel	192
3	Maria	173
4	Anna	155
5	Bertil	181

Efter att man gör ett PROC-steg, exempelvis proc print, måste man alltid skriva data = biblioteksnamn.filenames, annars vet inte SAS vilket datamaterial vi vill jobba med.

Om vi till exempel vill beräkna medelvärden kan vi skriva:

```
proc means data=work.langd;  
var langd;  
run;
```

Och vi får i utskriften:

```

The MEANS Procedure

Analysis Variable : langd


```

N	Mean	Std Dev	Minimum	Maximum
5	172.6000000	14.7410990	155.0000000	192.0000000

Här lägger vi även till en extra rad i koden, nämligen "var langd". Det gör vi för att tala om för SAS vilken variabel vi vill räkna ut medelvärde för. Specificerar vi inte det kommer vi få mått på alla våra numeriska variabler. Ett bra sätt att se om man skrivit rätt är att se att vissa ord i koden blir blå. Det betyder att just det kommandot vi har skrivit finns som SAS-kommando. Blir det rött är något fel.

```
proc means data=work.langd;  
vari langd;  
run;
```

Skriver vi "vari" istället för "var" kommer inte SAS förstå vad det är vi vill göra och vi kommer att få en varning i loggen. Man kan alltid gå till loggen och kolla om något har blivit fel.

```
11  proc means data=work.langd;  
12  vari langd;  
    ----  
    1  
WARNING 1-322: Assuming the symbol VAR was misspelled as vari.  
13  run;  
  
NOTE: There were 5 observations read from the data set WORK.LANGD.  
NOTE: PROCEDURE MEANS used (Total process time):  
      real time           0.73 seconds  
      cpu time            0.04 seconds
```

Här kan vi läsa att SAS antar att vi har stavat fel och menar "var" istället för "vari". Det gick alltså att köra koden ändå.

För att kopiera bilder och diagram från SAS till Word ska bilderna och diagrammen skapas i rätt format. För att göra detta lägger vi till två extra rader, som omsluter den ursprungliga koden. Exempelvis:

```
ods rtf;  
proc print data=work.langd;  
run;  
ods rtf close;
```

Vi skriver alltså "ods rtf" före proceduren som skapar bilden vi vill flytta till word och "ods rtf close" sist. (Ods står för output delivery system.)

När vi kör koden får vi frågan om vi ska öppna eller spara filen. Vi väljer **öppna**. (Det är viktigt att inte välja spara här.) Då öppnas ett nytt fönster som heter "results viewer". Härifrån kopierar vi bilden och klistrar in den i ett nytt word-dokument. När vi är klara stänger vi fönstret "results viewer". Först då kan vi skapa andra bilder i rtf format.