

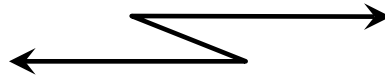
Local, world-class
services for the
pharmaceutical industry

data management, data warehousing, statistics,
information technology and scientific writing

[Beyond Your Data]

Ärrälle ja Sassiin

- R-käyttöliittymä ja SAS/STAT
uudet proseduurit.



❖ Perinteinen vastakkainasettelu R vs. SAS.

- R:n ja SAS:in yhteiskäyttö on ollut vaivalloista, vaikka molemmissa ohjelmistoissa on puolensa ja niitä voisi joissain tilanteissa soveltaa yhteisesti.
- Pääsyynä hankaluuteen on luonnollisesti ohjelmistojen kilpailevat asemat.

❖ SAS on tässä suhteessa nyt parantanut tarjontaansa asiakkailleen ja on rakentanut käyttöliittymän myös R:ään, SAS IML Studion kautta (Julkaistu heinäkuussa 2009): →



+



R-käyttöliittymän idea

- ❖ Tarjota SAS:in käyttäjille mahdollisuus kokeilla uusia algoritmeja, jotka on saatavilla vain nopeasti kehittyvässä R:ssä, olemassa oleviin SAS-datasetteihin ja vertailla tuloksia perinteisempiin SAS:in tuottamiin tuloksiin.
- ❖ Mahdollistaa sofistikoituneen analyysin teon R:llä ja tulosten jatkokäsittelyn tutummin jälleen SAS:lla.
- ❖ Lisäksi tarjotaan R:n käyttäjille mahdollisuutta kokeilla analyysejaan, datan muokkausta ym. SAS:lla, käyttämällä pohjana R-Dataa.

R-käyttöliittymän vaikutus

- ❖ Ennen R-käyttöliittymää SAS:ista R:ään:
 - 1) Tallenna SAS-data(t) formaattiin, jota R pystyy lukemaan.
 - 2) Lue aineisto sisään ja muunna RDataksi.
 - 3) Tee analyysi R:llä.
 - 4) Muodosta tulospaketti ja tallenna se tiedostoksi.
 - 5) Lue tallennettu tulostiedosto takaisin SAS:iin.
 - 6) Jatka aineiston analyysia SAS:issa.
- ❖ SAS IML-studion kautta voidaan kutsua suoraan R-ohjelmaa ja käyttää hyväksi SAS-datoja ja lukea suoraan tulokset takaisin SAS-ympäristöön jatkokäsittelyä varten.

Esimerkki 1

❖ Tehdään analyysi R:ssä ja tuodaan tulokset SAS:in käyttöön.

```
submit / R;  
library(nlme)  
model <- lme(weight~Diet+Time, random=~1|Chick, data=ChickWeight)  
name <- rownames(summary(model)$tTable);  
summar <- cbind(name, summary(model))  
endsubmit;
```

R

```
run ImportDataSetFromR( "WORK.summary", "summar" );
```

IML

```
submit;  
proc report data=summary nowd headline spacing=2 split='#';  
column name value p_value;  
define name /'Variable' width=11;  
define value /'Estimate' width=8 format=6.3;  
define p_value /'p-value' width=7 format=pvalue6.;  
run;endsubmit;
```

SAS

Esimerkki 2

❖ Käytetään SAS-datasettiä ja suoritetaan laskentaa R:ssä.

```
declare DataObject dobj;  
dobj = DataObject.CreateFromServerDataSet("sashelp.class");  
dobj.ExportToR("sashelp.class");
```

IML

```
submit / R;  
malli <- lm( Height ~ Age, data=sashelp.class )  
endsubmit;
```

R

```
dobj.AddAnalysisVarsFromR( {"Fit", "Residual"},  
    "cbind( fitted(malli), residuals(malli) )" );  
dobj.WriteToServerDataSet( "work.result" );
```

IML

```
submit;  
proc univariate normaltest plot data=result;  
var Residual;  
run;  
endsubmit;
```

SAS

Käyttöliittymän ongelmat

- ❖ Yhteys SAS:in ja R:n välillä ei ole vielä kovin suoraviivainen.
 - Useimmat SAS:in käyttäjät käyttävät pääasiassa BASE-SAS:ia
 - Suora yhteys siitä R:ään puuttuu:
 - ✓ SAS -- SAS IML -- R – SAS IML – SAS
 - Näin ollen käyttäjältä vaaditaan muutamien IML-kielen komentojen hallintaa, pystyäkseen soveltamaan R-käyttöliittymää.
- ❖ Rakenne ei ole kovin käyttäjäystävällinen. Esim. R-koodia ei voida ajaa osissa, vaan kaikki täytyy paketoida submit / R; endsubmit;-komentojen sisälle.
 - Sama pätee myös BASE-SAS-komentoihin.
 - Tulostuva LOG on puuttellinen, vaikeuttaa virheen löytämistä.

Käyttöliittymän tulevaisuus

- ❖ IML Studio on vasta ensimmäinen askel. SAS:in tuotekehitys pyrkii parantamaan SAS:in interaktiota R:ään edelleen.
- ❖ Saatavilla SAS/IML Studiossa heinäkuusta 2009 alkaen.
- ❖ Muiden SAS-tuotteiden yhteyteen vuodesta 2010 eteenpäin.
- ❖ Tavoitteena loppujen lopuksi luoda jotakin, joka voisi olla esimerkiksi `proc r;`

SAS/STAT uudet proseduurit versio 9.2

- ❖ Uusia proseduuria SAS/STAT, SAS-versiossa 9.2:
 - MCMC proseduurit on “Markov chain Monte Carlo” (MCMC) perusteinen simulaatio-proseduuri, joka on suunniteltu erilaisten Bayes-mallien sovittamiseen.
 - SEQDESIGN ja SEQTEST proseduurit ovat suunniteltu erityisesti välialalysejä varten. SEQDESIGN-proseduurilla suunnitellaan interim-analysejä kliinisiin kokeisiin ja SEQTEST-proseduurilla suoritetaan itse interim-analyysit.
 - Kokeellinen HPMIXED-proseduuri käyttää useita tehokkaita erikoistekniikoita sovittaakseen lineaarisia sekamalleja, joissa on varianssikomponenttirakenne.
 - Kokeellinen TCALIS-proseduuri päivittää CALIS-proseduurin myös rakenneyhtälö-mallinnukselle.

Proc SEQDESIGN

- ❖ Proc SEQDESIGN:in avulla voidaan laskea rajoja testistatistikan arvoille eri väli-analyysien vaiheissa, joiden perusteella tutkimus joko keskeytetään tai sitä jatketaan.
 - Laskentamenetelmiä on tarjolla runsaasti.
 - ✓ Esim. O'Brien-Fleming, Pocock, Whitehead, Haybittle-Peto, ym.
- ❖ Proseduurilla saadaan laskettua myös tarvittavat otoskoot eri tutkimusvaiheille.
- ❖ Proseduurilla voidaan helposti vertailla myös eri koeasetelmien toimivuutta.

Proc SEQDESIGN -syntaksi

Yksinkertaistettu syntaksi:

```
Proc seqdesing "design parameters" "table output"  
  "graphics output";
```

design method=rajojen laskentamenetelmä

alpha=tyypin I virheen tn beta=tyypin II virheen tn

nstages=tutkimusvaiheiden lkm

stop=koska tutkimus keskeytetään;

samplesize model= (Laskee tarvittun otoskoon, tai
vaihtoehtoisesti "eventtien" lukumäärän);

```
run;
```

Proc SEQTEST

- ❖ SEQTEST-proseduuri vertailee saatua testistatistikan arvoa SEQDESIGN:sta saatuihin raja-arvoihin jokaisessa interim-analyysissä. Tuloksen perusteella tutkimusta voidaan siis joko jatkaa tai se voidaan keskeyttää.
- ❖ SEQTEST:iä voidaan käyttää myös tutkimuksen lopussa parametrien estimaattien, luottamusvälien ja p-arvojen laskentaan.

PROC SEQTEST -syntaksi

Yhteenveto proseduurin koodirakenteesta:

Proc seqtest;

input datasets

boundaries

test variable

p-values and confidence limits

table output

graphics output

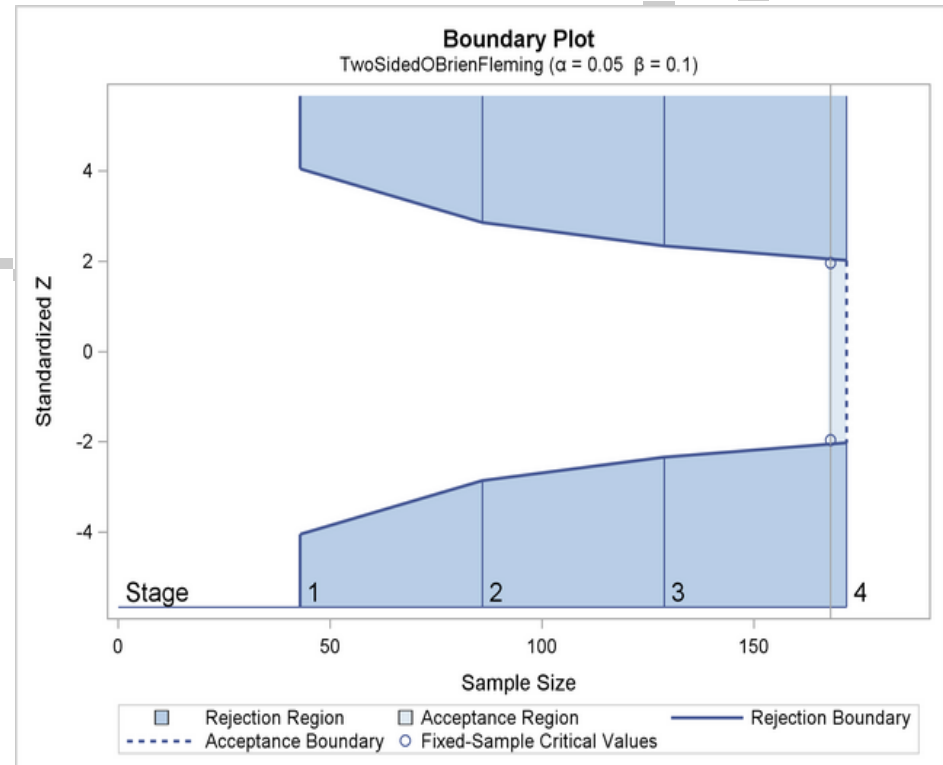
run;

Esimerkki-analyysi

```

proc seqdesign
  altref=-10 boundaryscale=stdz
  plots=boundary
  (hscale=samplesize);
  TwoSidedOBrienFleming: design
  method=obf nstages=4
  alt=twosided stop=reject
  alpha=0.05 beta=0.10;
  samplesize model=twosamplemean(
  stddev=20);
  ods output Boundary=bound_idl;
run;

```



Esimerkki-analyysi (jatkoa)

--Analyysi, esim. Keskiarvojen
vertailu--

ods graphics on;

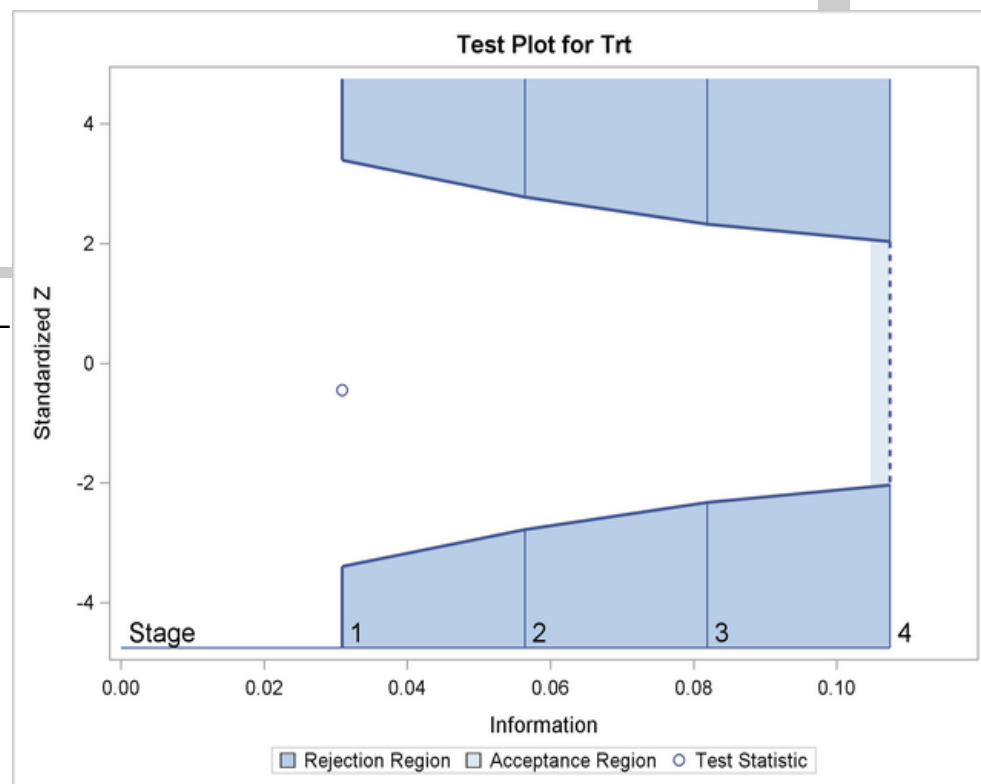
```
proc seqtest boundary=bound_1dl
  parms( testvar=Trt)= 1dlparms_
```

```
  nstages=4 boundaryscale=mle;
```

```
ods output test=test_1dl_1;
```

```
run;
```

ods graphics off;



Kiitos!

❖ Yhteystiedot:

➤ Jouni Junnila, M. Sc

Biostatistikko

Oy 4Pharma Ltd

Mikonkatu 8, 00100 Helsinki

jouni.junnila@4pharma.com

www.4pharma.com