```
%macro testaus(data,vaste,suppea=0,uusi=0,talleta=0,tied=na,tee3=0,vanha=0,vertailu=0);

*** Virallisten lajikekokeiden analysointi- ja laadunhallintaohjelma ***;
*** Versio 2.2, voimassa 6.9.2001- ***;
*** Tämä toimii SAS-ohjelmiston versioissa 8.0, 8.1 ja 8.2***;

options nodate nonumber;
goptions reset=all ftext=swiss gunit=pct;

%if &suppea=0 %then %do;
proc contents data=&data;
run;
%end;

data koe;
set &data;
if kerranne=. then kerranne=kerr;
if lohko=. then lohko=1;
if lajike=. then lajike=entry;
if ruutu=. then ruutu=plot;
run;

data koe2;set koe;
rename lajike=mmm1;
rename kerranne=mmm2;
rename lohko=mmm3;
rename &vaste=mmm4;
run;

data koe;set koe2;
lajike='1234567890123456';
kerranne=1;
lohko=1;
&vaste=1;
lajike=mmm1;
kerranne=mmm2;
lohko=mmm3;
&vaste=mmm4;
run;

proc sort;
by lajike;
run;

data koe;
set koe;
ed=lag(lajike);
if _n_=1 then ed=-999;
run;

data koe;
retain jlajike 0; *** jlajike = lajikkeen järjestysnumero ***;
set koe;
ero=lajike-ed;
if ero>0 then jlajike=jlajike+1;
run;

proc means noprint;
by lajike;
var jlajike;
output out=jlajike min=jlajike;
run;

data jlajike;
set jlajike;
drop _type_ _freq_;
run;

%if &suppea=0 %then %do;
title 'Lajikekoodit nousevassa järjestyksessä';
proc print noobs;
var jlajike lajike;
run;
%end;

title 'Lajikekokeen tilastollinen analyysi';

proc mixed data=koe method=reml %if &vanha=0 %then %do;ratio covtest%end;;
class lajike lohko kerranne;
model &vaste=lajike / s %if &suppea=0 %then %do;outp=resi%end; ;
random kerranne kerranne*lohko / ;
lsmeans lajike
%if &vertailu=1 %then %do;
/ adjust=tukey
%end;
;
```

```
%if &suppea=0 %then %do;
id lajike jlajike kerranne lohko;
*make predicted out=resi;
%end;
make lsmeans out=koesarja;
make solutionF out=feffe;
run;

%if &talleta ne 0 %then %do;
data koesarja;
set koesarja;
&vaste=estimate; *korjattu 6.9.2001;
keep lajike &vaste;
run;

proc sort;by lajike;run;

*data koesarja;
*set koesarja;
*file 'c:\tilap.dat';
*put lajike &vaste;
*run;

*data koesarja;
*infile 'c:\tilap.dat';
*input lajike &vaste;
*run;

data &tied;
%if &uusi=1 %then %do;set koesarja;%end;
%if &uusi=0 %then %do;merge &tied koesarja;by lajike;%end;
run;

data yleist;
set &data;
run;

proc sort data=yleist;
by lajike;
run;

data yleist;
set yleist;
laglaj=lag(lajike);
run;

data yleist;
set yleist;
jata=1;
if laglaj=lajike and _n_>1 then delete;
drop jata;
run;
%end;

%if &suppea=1 %then %goto loppu;

data resi;
set resi;
if resid=. then resid=_resid_;
if pred=. then pred=_pred_;
run;

proc means data=resi noprint;
var pred;
output out=vv min=minimi max=maksimi;
run;

data lisays;
set vv;
if maksimi>1 then do;
pred=int(minimi); output;
pred=(int(maksimi)+1); output;
end;
run;

title 'Jäännösvirheet';

proc univariate data=resi normal;
var resid;
run;

proc means data=resi noprint;
var resid;
output out=resi2 std=hajonta;
run;
```

```
data resi2;
set resi2;
yhdis=1;
run;

data resi;
set resi;
yhdis=1;
run;

data resi;
merge resi resi2;
by yhdis;
drop yhdis;
run;

******** tulostetaan poikkeavat havainnot;
data ui4;
set resi;
stdres=resid/hajonta;
if abs(stdres)>3;
run;

title 'Poikkeavat havainnot';
proc print data=ui4;
var lajike kerranne lohko stdres;
run;
*****************;

data resi;
set resi lisays;
IF KERRANNE=1 THEN R1=RESID/hajonta;
IF KERRANNE=2 THEN R2=RESID/hajonta;
IF KERRANNE=3 THEN R3=RESID/hajonta;
IF KERRANNE=4 THEN R4=RESID/hajonta;
pre=pred;
if pred>500 then pre=int(pred);
yhdis=1;
run;

proc sort data=koe;
by kerranne;
run;

title h=1 ' ';
goptions reset=all ftext=swiss gunit=pct;

proc gplot data=koe;
plot &vaste*jlajike=kerranne / vaxis=axis2 haxis=axis1 nolegend ;
axis1 value=(h=3) label=(HEIGHT=4 j=r r=0 'Lajike');
axis2 value=(h=3) label=(HEIGHT=4 a=90 r=0 "&vaste");
symbol1 i=none v=1 c=black r=1 h=2.5;* f=swissx;
symbol2 i=none v=2 c=black r=1 h=2.5;* f=swissx;
symbol3 i=none v=3 c=black r=1 h=2.5;* f=swissx;
symbol4 i=none v=4 c=black r=1 h=2.5;* f=swissx;
run;

title h=1 ' ';

proc gplot data=resi;
plot (r1 r2 r3 r4)*pre / vaxis=axis2 haxis=axis1 lvref=2 cvref=grey vref=2 vref=3 vref=-2 vref=-3 overlay nolegend ;
axis1 value=(h=3) label=(HEIGHT=4 r=0 j=r 'Sovitteet');
axis2 value=(h=3) label=(HEIGHT=4 a=90 r=0 'Standardoidut jaannosvirheet');
symbol1 i=none v=1 c=black r=1 h=2.5;
symbol2 i=none v=2 c=black r=1 h=2.5;
symbol3 i=none v=3 c=black r=1 h=2.5;
symbol4 i=none v=4 c=black r=1 h=2.5;
run;


************************************************************************;
%if &tee3=0 %then %goto t3end;

data T3;
set resi;
x=resid;
if x=. then delete;
i=1;
keep x i;
run;

proc means data=t3 noprint;
var x;
output out=calc mean=mean std=std n=n;
run;
```

```
data calc;
set calc;
i=1;
drop _type_ _freq_ ;
run;

data t3;
merge t3 calc;
by i;
drop i;
run;

data lasku;
set t3;
array a(100) a1-a100;
array b(100) b1-b100;
array c(100) c1-c100;
array d(100) d1-d100;
array e(100) e1-e100;
array f(100) f1-f100;
array g(100) g1-g100;
array h(100) h1-h100;
retain a1-a100 b1-b100 c1-c100 d1-d100 0;
do t=1 to 100;
ti=t*(2/99)-1-(2/99);
a(t)=a(t)+exp(ti*(x-mean)/std); e(t)=a(t)/n;
b(t)=b(t)+((x-mean)/std)*exp(ti*(x-mean)/std); f(t)=b(t)/n;
c(t)=c(t)+(((x-mean)/std)**2)*exp(ti*(x-mean)/std); g(t)=c(t)/n;
d(t)=d(t)+(((x-mean)/std)**3)*exp(ti*(x-mean)/std); h(t)=d(t)/n;
end;
output;
drop x a1-a100 b1-b100 c1-c100 d1-d100 t ti mean std;
run;

data lasku;
set lasku;
array e(100) e1-e100;
array f(100) f1-f100;
array g(100) g1-g100;
array h(100) h1-h100;
if _n_=n;
do i=1 to 100;
m0=e(i);
m1=f(i);
m2=g(i);
m3=h(i);
t=i*(2/99)-1-(2/99);
output;
end;
drop i e1-e100 f1-f100 g1-g100 h1-h100;
run;

data final;
set lasku;
t3=sqrt(n)*(1/m0**2)*(m3*m0-3*m2*m1+2*(m1**3)/m0);
run;

*********************************************************************;
title ' ';

data limit;
set calc;
array m(8) m1-m8;
array o(8) o1-o8;
do aika=1 to 100;
t=aika*(2/99)-1-(2/99);
m(1)=0;
m(2)=exp((t**2)/2);
m(3)=t*exp((t**2)/2);
m(4)=(1+t**2)*exp((t**2)/2);
m(5)=(3*t+t**3)*exp((t**2)/2);
m(6)=(3+6*(t**2)+t**4)*exp((t**2)/2);
m(7)=(15*t+10*(t**3)+t**5)*exp((t**2)/2);
m(8)=(15+45*(t**2)+15*(t**4)+t**6)*exp((t**2)/2);
t2=2*t;
o(1)=0;
o(2)=exp((t2**2)/2);
o(3)=t2*exp((t2**2)/2);
o(4)=(1+t2**2)*exp((t2**2)/2);
o(5)=(3*t2+t2**3)*exp((t2**2)/2);
o(6)=(3+6*(t2**2)+t2**4)*exp((t2**2)/2);
o(7)=(15*t2+10*(t2**3)+t2**5)*exp((t2**2)/2);
o(8)=(15+45*(t2**2)+15*(t2**4)+t2**6)*exp((t2**2)/2);
myy0=exp((t**2)/2);
myy1=t*exp((t**2)/2);
myy2=(1+t**2)*exp((t**2)/2);
```

```
myy3=(3*t+t**3)*exp((t**2)/2);
myy4=(3+6*(t**2)+t**4)*exp((t**2)/2);
myy5=(15*t+10*(t**3)+t**5)*exp((t**2)/2);
myy6=(15+45*(t**2)+15*(t**4)+t**6)*exp((t**2)/2);
kits=0;
do j=0 to 3; do l=0 to 3;
if j=0 then aij=(6*myy2*myy1*myy0-myy3*myy0*myy0-6*(myy1**3))/(myy0**4);
if j=1 then aij=(-3*(myy2*myy0-myy1**2))/(myy0**3);
if j=2 then aij=-3*myy1/(myy0**2);
if j=3 then aij=1/myy0;
if l=0 then ail=(6*myy2*myy1*myy0-myy3*myy0*myy0-6*(myy1**3))/(myy0**4);
if l=1 then ail=(-3*(myy2*myy0-myy1**2))/(myy0**3);
if l=2 then ail=-3*myy1/(myy0**2);
if l=3 then ail=1/myy0;
kjltt=o(j+l+2)-m(j+2)*m(l+2)-0.5*(l*m(l+2)+t*m(l+3))*(m(j+4)-m(j+2))
-0.5*(j*m(j+2)+t*m(j+3))*(m(l+4)-m(l+2))-(t*m(l+2)+l*m(l+1))*m(j+3)
-(t*m(j+2)+j*m(j+1))*m(l+3)+0.5*(l*m(l+2)+t*m(l+3))*(j*m(j+2)+t*m(j+3))
+(t*m(l+2)+l*m(l+1))*(t*m(j+2)+j*m(j+1));

kits=kits+aij*ail*kjltt;
* kits=kits+0.5*aij*ail*kjltt;
end;end;
Up5=( 1.65+1.38-(5.92/sqrt(n))+(12.8/n))*sqrt(kits);
Lo5=-(1.65+1.38-(5.92/sqrt(n))+(12.8/n))*sqrt(kits);
Up1=( 2.33+1.38-(5.92/sqrt(n))+(12.8/n))*sqrt(kits);
Lo1=-(2.33+1.38-(5.92/sqrt(n))+(12.8/n))*sqrt(kits);

TT=sqrt(kits);
output;
end;
keep t Up5 Lo5 up1 lo1;
run;

data kuva;
merge limit final;
by t;
keep t t3 up1 up5 lo1 lo5;
run;

proc gplot;
plot (t3 up5 lo5 up1 lo1)*t / vaxis=axis2 haxis=axis1 overlay;
axis1 value=(h=3) label=(HEIGHT=4 r=0 j=r 't');
axis2 value=(h=3) label=(HEIGHT=4 a=90 r=0 'T3-arvo');
symbol1 i=join v=none c=black l=1 r=1 w=2;
symbol2 i=join v=none c=black l=1 r=2;
symbol3 i=join v=none c=black l=2 r=2;
run;

%t3end:

goptions reset=all ftext=swiss gunit=pct;

data pass;
set resi;
_match_=1;
if resid ne . ;
run;

proc univariate noprint;
var resid;
output out=n1 n=nobs median=median std=std;
run;

data n2; set n1;
_match_=1;
run;

data nqplot;
merge pass n2;
drop _match_;
by _match_;
run;

proc sort data=nqplot;
by resid;
run;

data out;
set nqplot;
p=(_n_-.5)/nobs;
z=probit(p);
se=(std/((1/sqrt(2*3.1415926))*exp(-(z**2)/2)))*sqrt(p*(1-p)/nobs);
suora=std*z+median ;
_resid_=resid - suora;
ala=suora-2*se;
yla=suora+2*se;
```

```
label z='Normal Quantile';
run;

proc gplot data=out ;
plot resid*z=1
suora*z=2
ala*z=3
yla*z=3 / overlay vaxis=axis1 haxis=axis2 hminor=1 vminor=1;
symbol1 v=+ h=2 i=none c=black l=1;
symbol2 v=none i=join c=black l=1 w=1.5;
symbol3 v=none i=join c=black l=20 w=1.5;
axis1 label=(a=90 r=0 h=4) value=(h=3);
axis2 label=(h=4) value=(h=3);
run;

%loppu:

data teeuusi;
set koe;
run;

proc sort data=teeuusi;
by kerranne ruutu;
run;

title h=3 'Kokeen sijoittelun tarkastelu';

proc gplot data=teeuusi;
plot &vaste*ruutu / vaxis=axis2 haxis=axis1 nolegend ;
axis1 value=(h=3) label=(HEIGHT=4 j=r r=0 'Ruudun numero');
axis2 value=(h=3) label=(HEIGHT=4 a=90 r=0 "&vaste");
symbol1 i=join v=star c=black r=1 h=2.5;* f=swissx;
by kerranne;
run;

%mend;



%macro testaus8(data,vaste,suppea=0,uusi=0,talleta=0,tied=na,tee3=0,vanha=0,vertailu=0);

*** Virallisten lajikekokeiden analysointi- ja laadunhallintaohjelma ***;
*** Versio 2.2, voimassa 6.9.2001- ***;
*** Tämä toimii SAS-ohjelmiston versioissa 8.0, 8.1 ja 8.2***;

options nodate nonumber;
goptions reset=all ftext=swiss gunit=pct;

%if &suppea=0 %then %do;
proc contents data=&data;
run;
%end;

data koe;
set &data;
if kerranne=. then kerranne=kerr;
if lohko=. then lohko=1;
if lajike=. then lajike=entry;
if ruutu=. then ruutu=plot;
run;

data koe2;set koe;
rename lajike=mmm1;
rename kerranne=mmm2;
rename lohko=mmm3;
rename &vaste=mmm4;
run;

data koe;set koe2;
lajike='1234567890123456';
kerranne=1;
lohko=1;
&vaste=1;
lajike=mmm1;
kerranne=mmm2;
lohko=mmm3;
&vaste=mmm4;
run;

proc sort;
by lajike;
run;

data koe;
set koe;
ed=lag(lajike);
```

```
if _n_=1 then ed=-999;
run;

data koe;
retain jlajike 0; *** jlajike = lajikkeen järjestysnumero ***;
set koe;
ero=lajike-ed;
if ero>0 then jlajike=jlajike+1;
run;

proc means noprint;
by lajike;
var jlajike;
output out=jlajike min=jlajike;
run;

data jlajike;
set jlajike;
drop _type_ _freq_;
run;

%if &suppea=0 %then %do;
title 'Lajikekoodit nousevassa järjestyksessä';
proc print noobs;
var jlajike lajike;
run;
%end;

title 'Lajikekokeen tilastollinen analyysi';

proc mixed data=koe method=reml %if &vanha=0 %then %do;ratio covtest%end;;
class lajike lohko kerranne;
model &vaste=lajike / s %if &suppea=0 %then %do;outp=resi%end; ;
random kerranne kerranne*lohko / ;
lsmeans lajike
%if &vertailu=1 %then %do;
/ adjust=tukey
%end;
;
%if &suppea=0 %then %do;
id lajike jlajike kerranne lohko;
*make predicted out=resi;
%end;
make lsmeans out=koesarja;
make solutionF out=feffe;
run;

%if &talleta ne 0 %then %do;
data koesarja;
set koesarja;
&vaste=estimate; *korjattu 6.9.2001;
keep lajike &vaste;
run;

proc sort;by lajike;run;

*data koesarja;
*set koesarja;
*file 'c:\tilap.dat';
*put lajike &vaste;
*run;

*data koesarja;
*infile 'c:\tilap.dat';
*input lajike &vaste;
*run;

data &tied;
%if &uusi=1 %then %do;set koesarja;%end;
%if &uusi=0 %then %do;merge &tied koesarja;by lajike;%end;
run;

data yleist;
set &data;
run;

proc sort data=yleist;
by lajike;
run;

data yleist;
set yleist;
laglaj=lag(lajike);
run;

data yleist;
```

```
set yleist;
jata=1;
if laglaj=lajike and _n_>1 then delete;
drop jata;
run;
%end;

%if &suppea=1 %then %goto loppu;

data resi;
set resi;
if resid=. then resid=_resid_;
if pred=. then pred=_pred_;
run;

proc means data=resi noprint;
var pred;
output out=vv min=minimi max=maksimi;
run;

data lisays;
set vv;
if maksimi>1 then do;
pred=int(minimi); output;
pred=(int(maksimi)+1); output;
end;
run;

title 'Jäännösvirheet';

proc univariate data=resi normal;
var resid;
run;

proc means data=resi noprint;
var resid;
output out=resi2 std=hajonta;
run;

data resi2;
set resi2;
yhdis=1;
run;

data resi;
set resi;
yhdis=1;
run;

data resi;
merge resi resi2;
by yhdis;
drop yhdis;
run;

******** tulostetaan poikkeavat havainnot;
data ui4;
set resi;
stdres=resid/hajonta;
if abs(stdres)>3;
run;

title 'Poikkeavat havainnot';
proc print data=ui4;
var lajike kerranne lohko &vaste stdres;
run;
*****************;

data resi;
set resi lisays;
IF KERRANNE=1 THEN R1=RESID/hajonta;
IF KERRANNE=2 THEN R2=RESID/hajonta;
IF KERRANNE=3 THEN R3=RESID/hajonta;
IF KERRANNE=4 THEN R4=RESID/hajonta;
pre=pred;
if pred>500 then pre=int(pred);
yhdis=1;
run;

proc sort data=koe;
by kerranne;
run;

title h=1 ' ';
goptions reset=all ftext=swiss gunit=pct;
```

```
proc gplot data=koe;
plot &vaste*jlajike=kerranne / vaxis=axis2 haxis=axis1 nolegend ;
axis1 value=(h=3) label=(HEIGHT=4 j=r r=0 'Lajike');
axis2 value=(h=3) label=(HEIGHT=4 a=90 r=0 "&vaste");
symbol1 i=none v=1 c=black r=1 h=2.5;* f=swissx;
symbol2 i=none v=2 c=black r=1 h=2.5;* f=swissx;
symbol3 i=none v=3 c=black r=1 h=2.5;* f=swissx;
symbol4 i=none v=4 c=black r=1 h=2.5;* f=swissx;
run;

title h=1 ' ';

proc gplot data=resi;
plot (r1 r2 r3 r4)*pre / vaxis=axis2 haxis=axis1 lvref=2 cvref=grey vref=2 vref=3 vref=-2 vref=-3 overlay nolegend ;
axis1 value=(h=3) label=(HEIGHT=4 r=0 j=r 'Sovitteet');
axis2 value=(h=3) label=(HEIGHT=4 a=90 r=0 'Standardoidut jaannosvirheet');
symbol1 i=none v=1 c=black r=1 h=2.5;
symbol2 i=none v=2 c=black r=1 h=2.5;
symbol3 i=none v=3 c=black r=1 h=2.5;
symbol4 i=none v=4 c=black r=1 h=2.5;
run;


************************************************************************;
%if &tee3=0 %then %goto t3end;

data T3;
set resi;
x=resid;
if x=. then delete;
i=1;
keep x i;
run;

proc means data=t3 noprint;
var x;
output out=calc mean=mean std=std n=n;
run;

data calc;
set calc;
i=1;
drop _type_ _freq_ ;
run;

data t3;
merge t3 calc;
by i;
drop i;
run;

data lasku;
set t3;
array a(100) a1-a100;
array b(100) b1-b100;
array c(100) c1-c100;
array d(100) d1-d100;
array e(100) e1-e100;
array f(100) f1-f100;
array g(100) g1-g100;
array h(100) h1-h100;
retain a1-a100 b1-b100 c1-c100 d1-d100 0;
do t=1 to 100;
ti=t*(2/99)-1-(2/99);
a(t)=a(t)+exp(ti*(x-mean)/std); e(t)=a(t)/n;
b(t)=b(t)+((x-mean)/std)*exp(ti*(x-mean)/std); f(t)=b(t)/n;
c(t)=c(t)+(((x-mean)/std)**2)*exp(ti*(x-mean)/std); g(t)=c(t)/n;
d(t)=d(t)+(((x-mean)/std)**3)*exp(ti*(x-mean)/std); h(t)=d(t)/n;
end;
output;
drop x a1-a100 b1-b100 c1-c100 d1-d100 t ti mean std;
run;

data lasku;
set lasku;
array e(100) e1-e100;
array f(100) f1-f100;
array g(100) g1-g100;
array h(100) h1-h100;
if _n_=n;
do i=1 to 100;
m0=e(i);
m1=f(i);
m2=g(i);
m3=h(i);
t=i*(2/99)-1-(2/99);
```

```
output;
end;
drop i e1-e100 f1-f100 g1-g100 h1-h100;
run;

data final;
set lasku;
t3=sqrt(n)*(1/m0**2)*(m3*m0-3*m2*m1+2*(m1**3)/m0);
run;

**********************************************************************;
title ' ';

data limit;
set calc;
array m(8) m1-m8;
array o(8) o1-o8;
do aika=1 to 100;
t=aika*(2/99)-1-(2/99);
m(1)=0;
m(2)=exp((t**2)/2);
m(3)=t*exp((t**2)/2);
m(4)=(1+t**2)*exp((t**2)/2);
m(5)=(3*t+t**3)*exp((t**2)/2);
m(6)=(3+6*(t**2)+t**4)*exp((t**2)/2);
m(7)=(15*t+10*(t**3)+t**5)*exp((t**2)/2);
m(8)=(15+45*(t**2)+15*(t**4)+t**6)*exp((t**2)/2);
t2=2*t;
o(1)=0;
o(2)=exp((t2**2)/2);
o(3)=t2*exp((t2**2)/2);
o(4)=(1+t2**2)*exp((t2**2)/2);
o(5)=(3*t2+t2**3)*exp((t2**2)/2);
o(6)=(3+6*(t2**2)+t2**4)*exp((t2**2)/2);
o(7)=(15*t2+10*(t2**3)+t2**5)*exp((t2**2)/2);
o(8)=(15+45*(t2**2)+15*(t2**4)+t2**6)*exp((t2**2)/2);
myy0=exp((t**2)/2);
myy1=t*exp((t**2)/2);
myy2=(1+t**2)*exp((t**2)/2);
myy3=(3*t+t**3)*exp((t**2)/2);
myy4=(3+6*(t**2)+t**4)*exp((t**2)/2);
myy5=(15*t+10*(t**3)+t**5)*exp((t**2)/2);
myy6=(15+45*(t**2)+15*(t**4)+t**6)*exp((t**2)/2);
kits=0;
do j=0 to 3; do l=0 to 3;
if j=0 then aij=(6*myy2*myy1*myy0-myy3*myy0*myy0-6*(myy1**3))/(myy0**4);
if j=1 then aij=(-3*(myy2*myy0-myy1**2))/(myy0**3);
if j=2 then aij=-3*myy1/(myy0**2);
if j=3 then aij=1/myy0;
if l=0 then ail=(6*myy2*myy1*myy0-myy3*myy0*myy0-6*(myy1**3))/(myy0**4);
if l=1 then ail=(-3*(myy2*myy0-myy1**2))/(myy0**3);
if l=2 then ail=-3*myy1/(myy0**2);
if l=3 then ail=1/myy0;
kjltt=o(j+l+2)-m(j+2)*m(l+2)-0.5*(l*m(l+2)+t*m(l+3))*(m(j+4)-m(j+2))
-0.5*(j*m(j+2)+t*m(j+3))*(m(l+4)-m(l+2))-(t*m(l+2)+l*m(l+1))*m(j+3)
-(t*m(j+2)+j*m(j+1))*m(l+3)+0.5*(l*m(l+2)+t*m(l+3))*(j*m(j+2)+t*m(j+3))
+(t*m(l+2)+l*m(l+1))*(t*m(j+2)+j*m(j+1));

kits=kits+aij*ail*kjltt;
* kits=kits+0.5*aij*ail*kjltt;
end;end;
Up5=( 1.65+1.38-(5.92/sqrt(n))+(12.8/n))*sqrt(kits);
Lo5=-(1.65+1.38-(5.92/sqrt(n))+(12.8/n))*sqrt(kits);
Up1=( 2.33+1.38-(5.92/sqrt(n))+(12.8/n))*sqrt(kits);
Lo1=-(2.33+1.38-(5.92/sqrt(n))+(12.8/n))*sqrt(kits);

TT=sqrt(kits);
output;
end;
keep t Up5 Lo5 up1 lo1;
run;

data kuva;
merge limit final;
by t;
keep t t3 up1 up5 lo1 lo5;
run;

proc gplot;
plot (t3 up5 lo5 up1 lo1)*t / vaxis=axis2 haxis=axis1 overlay;
axis1 value=(h=3) label=(HEIGHT=4 r=0 j=r 't');
axis2 value=(h=3) label=(HEIGHT=4 a=90 r=0 'T3-arvo');
symbol1 i=join v=none c=black l=1 r=1 w=2;
symbol2 i=join v=none c=black l=1 r=2;
symbol3 i=join v=none c=black l=2 r=2;
run;
```

```
%t3end:

goptions reset=all ftext=swiss gunit=pct;

data pass;
set resi;
_match_=1;
if resid ne . ;
run;

proc univariate noprint;
var resid;
output out=n1 n=nobs median=median std=std;
run;

data n2; set n1;
_match_=1;
run;

data nqplot;
merge pass n2;
drop _match_;
by _match_;
run;

proc sort data=nqplot;
by resid;
run;

data out;
set nqplot;
p=(_n_-.5)/nobs;
z=probit(p);
se=(std/((1/sqrt(2*3.1415926))*exp(-(z**2)/2)))*sqrt(p*(1-p)/nobs);
suora=std*z+median ;
_resid_=resid - suora;
ala=suora-2*se;
yla=suora+2*se;
label z='Normal Quantile';
run;

proc gplot data=out ;
plot resid*z=1
suora*z=2
ala*z=3
yla*z=3 / overlay vaxis=axis1 haxis=axis2 hminor=1 vminor=1;
symbol1 v=+ h=2 i=none c=black l=1;
symbol2 v=none i=join c=black l=1 w=1.5;
symbol3 v=none i=join c=black l=20 w=1.5;
axis1 label=(a=90 r=0 h=4) value=(h=3);
axis2 label=(h=4) value=(h=3);
run;

%loppu:

data teeuusi;
set koe;
run;

proc sort data=teeuusi;
by kerranne ruutu;
run;

title h=3 'Kokeen sijoittelun tarkastelu';

proc gplot data=teeuusi;
plot &vaste*ruutu / vaxis=axis2 haxis=axis1 nolegend ;
axis1 value=(h=3) label=(HEIGHT=4 j=r r=0 'Ruudun numero');
axis2 value=(h=3) label=(HEIGHT=4 a=90 r=0 "&vaste");
symbol1 i=join v=star c=black r=1 h=2.5;* f=swissx;
by kerranne;
run;

%mend;
```