SAS Codes for seemingly unrelated regression analysis

N. Uttam Singh, Kishore K Das , Anjoo Yumnam and Aniruddha Roy, 2017

Code for SUR or SURE estimates.

```
proc syslin data=SEEMING
SUR out=SUR_PRED ;
AR: model ARFoodgrain_Yld =ARMin_Tem    ARRain / DW ;
output predicted=AR_pred residual=AR_res;
AS: model ASFoodgrain_Yld =ASMin_Tem    ASRain / DW ;
output predicted=AS_pred residual=AS_res;
MN: model MNFoodgrain_Yld =MNMin_Tem    MNRain / DW ;
output predicted=MN_pred residual=MN_res;
ML: model MLFoodgrain_Yld =MLMin_Tem    MLRain / DW;
output predicted=ML_pred residual=ML_res;
MZ: model MZFoodgrain_Yld =MZMin_Tem    MZRain / DW;
output predicted=MZ_pred residual=MZ_res;
NL: model NLFoodgrain_Yld =NLMin_Tem    NLRain / DW;
output predicted=NL_pred residual=NL_res;
SK: model SKFoodgrain_Yld =SKMin_Tem    SKRain / DW;
output predicted=SK_pred residual=SK_res;
TR: model TRFoodgrain_Yld =TRMin_Tem    TRRain / DW;
output predicted=TR_pred residual=TR_res;
run;
```

Code to test heteroscedasticity individually.

```
proc model data=SEEMING;
ARFoodgrain_Yld=AR1+AR2*ARMin_Tem+AR3*ARRain;
ASFoodgrain_Yld=AS1+AS2*ASMin_Tem+AS3*ASRain;
MNFoodgrain_Yld=MN1+MN2*MNMin_Tem+MN3*MNRain;
MLFoodgrain_Yld=ML1+ML2*MLMin_Tem+ML3*MLRain ;
MZFoodgrain_Yld=MZ1+MZ2*MZMin_Tem+MZ3*MZRain ;
NLFoodgrain_Yld=NL1+NL2*NLMin_Tem+NL3*NLRain ;
SKFoodgrain_Yld=SK1+SK2*SKMin_Tem+SK3*SKRain ;
TRFoodgrain_Yld=TR1+TR2*TRMin_Tem+TR3*TRRain ;

fit ARFoodgrain_Yld /itsur white breusch=(1 ARMin_Tem
     ARRain);
fit ASFoodgrain_Yld / itsur white breusch=(1 ASMin_Tem
ASRain);
fit  MNFoodgrain_Yld /itsur white breusch=(1MNMin_Tem
MNRain);
fit  MLFoodgrain_Yld / itsur white breusch=(1 MLMin_Tem
MLRain);
fit  MZFoodgrain_Yld /itsur white breusch=(1 MZMin_Tem
MZRain);
```

```
fit  NLFoodgrain_Yld /itsur white breusch=(1 NLMin_Tem
NLRain);
fit SKFoodgrain_Yld/itsur white breusch=(1 SKMin_Tem
     SKRain);
fit TRFoodgrain_Yld/itsur white breusch=(1 TRMin_Tem
     TRRain);
run;
```

Code to test heteroscedasticity simultaneously.

```
proc model data=SEEMING GRARH;
parameters AR1-AR3 AS1-AS3 MN1-MN3 ML1-ML3 MZ1-MZ3 NL1-NL3
SK1-SK3 TR1-TR3;
ARFoodgrain_Yld=AR1+AR2*ARMin_Tem+AR3*ARRain;
ASFoodgrain_Yld=AS1+AS2*ASMin_Tem+AS3*ASRain;
MNFoodgrain_Yld=MN1+MN2*MNMin_Tem+MN3*MNRain;
MLFoodgrain_Yld=ML1+ML2*MLMin_Tem+ML3*MLRain;
MZFoodgrain_Yld=MZ1+MZ2*MZMin_Tem+MZ3*MZRain;
NLFoodgrain_Yld=NL1+NL2*NLMin_Tem+NL3*NLRain;
SKFoodgrain_Yld=SK1+SK2*SKMin_Tem+SK3*SKRain;
TRFoodgrain_Yld=TR1+TR2*TRMin_Tem+TR3*TRRain;
fit ARFoodgrain_Yld ASFoodgrain_Yld MNFoodgrain_Yld
MLFoodgrain_Yld MZFoodgrain_Yld NLFoodgrain_Yld
SKFoodgrain_Yld
TRFoodgrain_Yld/ ITSUR white breusch=(1 ARMin_Tem ARRain
GODFREY=3
DW NORMAL   ;
run;
quit;
```

Code for restrictions and parameter test.

```
proc syslin data=SEEMING   ITSUR
AR: model ARFoodgrain_Yld =ARMin_Tem    ARRain;
AS: model ASFoodgrain_Yld =ASMin_Tem    ASRain;
MN: model MNFoodgrain_Yld =MNMin_Tem    MNRain;
ML: model MLFoodgrain_Yld =MLMin_Tem    MLRain;
MZ: model MZFoodgrain_Yld =MZMin_Tem    MZRain;
NL: model NLFoodgrain_Yld =NLMin_Tem    NLRain;
SK: model SKFoodgrain_Yld =SKMin_Tem    SKRain;
TR: model TRFoodgrain_Yld =TRMin_Tem    TRRain;
stest1: stest
AR.INTERCEPT=AS.INTERCEPT=MN.INTERCEPT=ML.INTERCEPT=MZ.INTE
RCEPT=NL.INTERCEPT=SK.INTERCEPT=TR.INTERCEPT;
stest2: stest
AR.INTERCEPT=AS.INTERCEPT=MN.INTERCEPT=ML.INTERCEPT=MZ.INTE
RCEPT=NL.INTERCEPT=SK.INTERCEPT=TR.INTERCEPT=0;
```

```
stest3: stest
AR.ARMin_Tem=AS.ASMin_Tem=MN.MNMin_Tem=ML.MLMin_Tem=MZ.MZMi
n_Tem=NL.NLMin_Tem=SK.SKMin_Tem=TR.TRMin_Tem;
stest4: stest
AR.ARMin_Tem=AS.ASMin_Tem=MN.MNMin_Tem=ML.MLMin_Tem=MZ.MZMi
n_Tem=NL.NLMin_Tem=SK.SKMin_Tem=TR.TRMin_Tem=0;
/*stest5: stest
AR.ARRain=AS.ASRain=MN.MNRain=ML.MLRain=MZ.MZRain=NL.NLRain
=SK.SKRain=TR.TRRain;*/
/*stest6: stest
AR.ARRain=AS.ASRain=MN.MNRain=ML.MLRain=MZ.MZRain=NL.NLRain
=SK.SKRain=TR.TRRain=0;*/
RESTRICT1: SRESTRICT
AR.ARRain=AS.ASRain=MN.MNRain=ML.MLRain=MZ.MZRain=NL.NLRain
=SK.SKRain=TR.TRRain;
/*RESTRICT2: SRESTRICT
AR.ARRain=AS.ASRain=MN.MNRain=ML.MLRain=MZ.MZRain=NL.NLRain
=SK.SKRain=TR.TRRain;*/
run;
```

Code for SUR estimation with autoregressive error terms.

```
proc model data=SEEMING ;
parameters AR1-AR3 AS1-AS3 MN1-MN3 ML1-ML3 MZ1-MZ3 NL1-NL3
SK1-SK3 TR1-TR3;
ARFoodgrain_Yld=AR1+AR2*ARMin_Tem+AR3*ARRain;
ASFoodgrain_Yld=AS1+AS2*ASMin_Tem+AS3*ASRain;
MNFoodgrain_Yld=MN1+MN2*MNMin_Tem+MN3*MNRain;
MLFoodgrain_Yld=ML1+ML2*MLMin_Tem+ML3*MLRain ;
MZFoodgrain_Yld=MZ1+MZ2*MZMin_Tem+MZ3*MZRain ;
NLFoodgrain_Yld=NL1+NL2*NLMin_Tem+NL3*NLRain ;
SKFoodgrain_Yld=SK1+SK2*SKMin_Tem+SK3*SKRain ;
TRFoodgrain_Yld=TR1+TR2*TRMin_Tem+TR3*TRRain ;
%AR (AR_AR, 2, ARFoodgrain_Yld)
%AR (AR_AS, 2, ASFoodgrain_Yld)
%AR (AR_MN, 2, MNFoodgrain_Yld)
%AR (AR_ML, 2, MLFoodgrain_Yld)
%AR (AR_MZ, 2, MZFoodgrain_Yld)
%AR (AR_NL, 2, NLFoodgrain_Yld)
%AR (AR_SK, 2, SKFoodgrain_Yld)
%AR (AR_TR, 2, TRFoodgrain_Yld)
fit ARFoodgrain_Yld
     ASFoodgrain_Yld
     MNFoodgrain_Yld
     MLFoodgrain_Yld
     MZFoodgrain_Yld
     NLFoodgrain_Yld
     SKFoodgrain_Yld
     TRFoodgrain_Yld
```

```
START= (
AR1=5000 AS1=5000 MN1=5000 ML1=5000 MZ1=5000 NL1=5000
SK1=5000  TR1=5000 AR2=50   AS2=50   MN2=50   ML2=50
MZ2=50   NL2=50   SK2=50    TR2=50   AR3=50   AS3=50
MN3=50   ML3=50   MZ3=50   NL3=50   SK3=50    TR3=50
AR_AR_L1 = 1.5000  AR_AR_L2 = 1.5000  AR_AR_L3 = 1.5000
AR_AS_L1 = 1.5000  AR_AS_L2 = 1.5000  AR_AS_L3 = 1.5000
AR_MN_L1 = 1.5000  AR_MN_L2 = 1.5000  AR_MN_L3 = 1.5000
AR_ML_L1 = 1.5000  AR_ML_L2 = 1.5000  AR_ML_L3 = 1.5000
AR_MZ_L1 = 1.5000  AR_MZ_L2 = 1.5000  AR_MZ_L3 = 1.5000
AR_NL_L1 = 1.5000  AR_NL_L2 = 1.5000  AR_NL_L3 = 1.5000
AR_SK_L1 = 1.5000  AR_SK_L2 = 1.5000  AR_SK_L3 = 1.5000
AR_TR_L1 = 1.5000  AR_TR_L2 = 1.5000  AR_TR_L3 = 1.5000)
/ITSUR     MAXITER=32000 CONVERGE=0.0000001  DW;
RUN;
```

The above code could not converge. Hence we remove an arbitrary equation and do adding up restriction.

```
proc model data=SEEMING ;
parameters AR1-AR3 AS1-AS3 MN1-MN3 ML1-ML3 MZ1-MZ3 NL1-NL3
SK1-SK3 TR1-TR3;
ARFoodgrain_Yld=AR1+AR2*ARMin_Tem+AR3*ARRain;
ASFoodgrain_Yld=AS1+AS2*ASMin_Tem+AS3*ASRain;
MNFoodgrain_Yld=MN1+MN2*MNMin_Tem+MN3*MNRain;
MLFoodgrain_Yld=ML1+ML2*MLMin_Tem+ML3*MLRain ;
MZFoodgrain_Yld=MZ1+MZ2*MZMin_Tem+MZ3*MZRain ;
NLFoodgrain_Yld=NL1+NL2*NLMin_Tem+NL3*NLRain ;
SKFoodgrain_Yld=SK1+SK2*SKMin_Tem+SK3*SKRain ;
TRFoodgrain_Yld=TR1+TR2*TRMin_Tem+TR3*TRRain ;
%AR (AR_AR, 2, ARFoodgrain_Yld)
%AR (AR_AS, 2, ASFoodgrain_Yld)
%AR (AR_MN, 2, MNFoodgrain_Yld)
%AR (AR_ML, 2, MLFoodgrain_Yld)
%AR (AR_MZ, 2, MZFoodgrain_Yld)
%AR (AR_NL, 2, NLFoodgrain_Yld)
%AR (AR_SK, 2, SKFoodgrain_Yld)
%AR (AR_TR, 2, TRFoodgrain_Yld)
fit
     ARFoodgrain_Yld
     ASFoodgrain_Yld
/*   MNFoodgrain_Yld */
     MLFoodgrain_Yld
     MZFoodgrain_Yld
     NLFoodgrain_Yld
     SKFoodgrain_Yld
     TRFoodgrain_Yld
```

```
START= (
AR1=5000 AS1=5000 MN1=5000 ML1=5000 MZ1=5000 NL1=5000
SK1=5000  TR1=5000 AR2=50   AS2=50   MN2=50   ML2=50
MZ2=50   NL2=50   SK2=50    TR2=50   AR3=50   AS3=50
MN3=50   ML3=50   MZ3=50   NL3=50   SK3=50    TR3=50
AR_AR_L1 = 1.5000  AR_AR_L2 = 1.5000  AR_AR_L3 = 1.5000
AR_AS_L1 = 1.5000  AR_AS_L2 = 1.5000  AR_AS_L3 = 1.5000
AR_MN_L1 = 1.5000  AR_MN_L2 = 1.5000  AR_MN_L3 = 1.5000
AR_ML_L1 = 1.5000  AR_ML_L2 = 1.5000  AR_ML_L3 = 1.5000
AR_MZ_L1 = 1.5000  AR_MZ_L2 = 1.5000  AR_MZ_L3 = 1.5000
AR_NL_L1 = 1.5000  AR_NL_L2 = 1.5000  AR_NL_L3 = 1.5000
AR_SK_L1 = 1.5000  AR_SK_L2 = 1.5000  AR_SK_L3 = 1.5000
AR_TR_L1 = 1.5000  AR_TR_L2 = 1.5000  AR_TR_L3 = 1.5000)
/ITSUR  MAXITER=32000 CONVERGE=0.0000001 DW CORR
 breusch=(1 ARMin_Tem    ARRain );
```

Code for adding up restriction.

```
RESTRICT  AR2+AS2+ MN2+ ML2+ MZ2+ NL2+ SK2+ TR2=0,
          AR3+AS3+ MN3+ ML3+ MZ3+ NL3+ SK3+ TR3=0,
          AR1+AS1+ MN1+ ML1+ MZ1+ NL1+ SK1+ TR1=1,
AR_AR_L1 +AR_AS_L1 +AR_MN_L1 +AR_ML_L1 +AR_MZ_L1 +AR_NL_L1
+AR_SK_L1 +AR_TR_L1=0,
AR_AR_L2 +AR_AS_L2 +AR_MN_L2 +AR_ML_L2 +AR_MZ_L2 +AR_NL_L2
+AR_SK_L2 +AR_TR_L2=0;
RUN;
```