Online Supplement to
" Heteroskedasticity Robust Specification Testing in Spatial Autoregression "

Jungyoon Lee[a], Peter C. B. Phillips[b] and Francesca Rossi[c]

[a] *Royal Holloway, University of London*

[b] *Yale University, University of Auckland, & Singapore Management University*

[c] *University of Verona*

April 13, 2024

This Online Supplement is a self-contained document presenting Matlab and R codes for the specification test developed in the paper, based on the test statistic $\hat{T}(t, t_Y)$ of (3.18). We also provide a small simulated dataset and the resulting output tables for illustration. Interested readers are directed to the corresponding author's website for downloads of these codes and a larger dataset that replicates columns 2 and 4 of Table 13. Below, we present an outline of the codes followed by Matlab and R codes, the example data and the output tables.

S.1 Outline of codes

Below, we present two different files in both Matlab and R:

- "LPRtest.m" and "LPRtest.R" are function files for implementing the specification test based on (3.18). The codes contain automatic choices of t and t_Y based on the description given in Section 7 and let $p_n = n^{1/3}$. The instrument variable Z is set as spatial IV, i.e. pre-multiplying the first $K1$ columns of regressors by W, as discussed below. There are four inputs required:
  - "Y": The $n \times 1$ vector of dependent variables.
  - "X": Data matrix of independent variables included in the regression. Note that this would include e.g. the vector of ones for the intercept and dummies for fixed effects. Some of these may not need to be used in the exponential of the test statistic (in the first line of equation (3.11)) and generating the spatial instrument used in the 2SLS estimation, necessitating the next input:

  - "K1": a scalar for specifying the first K1 columns of X to be used in the exponential of test and the spatial instrument- hence "X" needs to have its columns ordered accordingly, e.g. in the example dataset discussed below, the last column of "X" is the vector of ones.
  - "W": The $n \times n$ weight matrix.

  There are two output tables produced:

  - Table 1 reports the 2SLS estimates of λ and β, along with the corresponding t-statitics and their p-values.
  - Table 2 reports the test statistic $\hat{T}(t, t_Y)$ of (3.18) and the corresponding p-value.

- "LPRexample.m" and "LPRexample.R" demonstrate the use of the aforementioned function files. They assign inputs and will display the outputs when run. We also include the output windows here for completeness.

S.2 Matlab codes

"LPRtest.m"

```
function [table1,table2] = LPRtest(Y, X, Wn, K1)
n=size(X,1);%sample size
K=size(X,2);%number of regressors
tx=zeros(K1,1);
tp=0;
k= n^(1/3);%p in the paper
X_scaled=zeros(n, K1);
mpreX=zeros(1, K1);
for i=1:K1
 mpreX(:,i)=ones(1,n)*X(1:n,i)/n;
 X_scaled(:,i)=X(1:n,i)-ones(n,1)*mpreX(i);
end
pretau=0;
for j=1:n
 pretau=pretau+ exp(ones(1,K1)*atan(X_scaled(j,:))');
end
tau=log(10*n/pretau);
tp=tau/(3.75);%t_y in the paper
tx=tau*ones(K1,1);%t
funSn = @(t) inv(eye(n)- t*Wn);%Sn^(-1)
funGn= @(t)Wn* funSn(t);
%estimation
```

```
Z=zeros(n,K+K1);
Z(1:n,1:K)= X;
Z(1:n,K+1:K+K1)= Wn*X(1:n,1:K1);
P=zeros(n,n);
P=Z*(inv(Z'*Z))*Z';
Xex=zeros(n,K+1);%includes Wy
Xex=cat(2, Wn*Y,X);
theta2sls=0;
theta2sls=(inv(Xex'*P*Xex))*Xex'*P*Y;
lambda2sls=theta2sls(1);
beta2sls=theta2sls(2:K+1);
ehat_correct=zeros(n,1);%fitted residual
ehat_correct=Y- lambda2sls*Wn*Y- X*beta2sls;
Cov=zeros(K+1,K+1)';
Cov=inv(Xex'*P*Xex)*Xex'*P*diag(diag(ehat_correct*ehat_correct'))*P*Xex
    *inv(Xex'*P*Xex);
t_stat_lambda=0;
t_stat_lambda=lambda2sls/sqrt(Cov(1,1));
tstat_theta=zeros(K+1,1);
tstat_theta=inv(sqrt(diag(diag(Cov))))*theta2sls;
%implementation of specification test
wy0=zeros(n,1);
wy1=zeros(n,1);
prewy0=zeros(n,1);
prewy1=zeros(n,1);
prewy0=Y;
mpre0=0;
mpre0=ones(1,n)*prewy0/n;
wy0=(prewy0-ones(n,1)*mpre0);
e0_1=zeros(n,1);
e0_2=zeros(n,1);
for i=1:n
      e0_1(i)=exp(tx'*atan(X_scaled(i,:))');
      e0_2(i)=exp(tp*wy0(i)/k);
end
ehat_correct_sq= ehat_correct.^2;
Sigma=diag(ehat_correct_sq);
B=zeros(K+K1,K+1);
B=(1/n)*Z'*Xex;
A=zeros(K+K1,K+K1);
A=(1/n)*Z'*Z;
Omega=zeros(K+1,K+1);
Omega= B'*inv(A)*B;
```

```
Q_cc=zeros(n,K+1);
Q_cc=[funGn(lambda2sls)*X*beta2sls,  X];
Q=zeros(n,n);
Q=funSn(lambda2sls)*(eye(n)-(1/n)*Q_cc*inv(Omega)*B'*inv(A)* Z');
Sbar= zeros(1,n);
Sbar= (1/n)* sum(funSn(lambda2sls));
Sd=zeros(n,n);
Sd= funSn(lambda2sls) - ones(n,1)* Sbar;
E=zeros(2,n);
E(1,1:n)= e0_1';
E(2,1:n)= ones(1,n);
psi1= zeros(2,n);
psi1= E*Q;
V_hat=zeros(2,2);
V_hat=(psi1)*Sigma*psi1'/n;
Mhat=zeros(2,1);
Mhat_1=0;
Mhat_2=0;
Mhat_1= (Y-funSn(lambda2sls)*X*beta2sls)'*e0_1/n;
Mhat_2= (Y-funSn(lambda2sls)*X*beta2sls)'*e0_2/n - (tp/(n*(k)))* trace(
    Sd' * Q*Sigma);
Mhat(1,1)= Mhat_1;
Mhat(2,1)= Mhat_2;
That=n* Mhat'*(inv(V_hat))*Mhat;
    %Reporting results
    pvalue=2*normcdf(abs(tstat_theta), 'upper');
    pvalueTsq= chi2cdf(That, 2, 'upper');
    %the first element is lambda
    table1=table(theta2sls,tstat_theta,pvalue)
    table2=table(That, pvalueTsq)
```

"LPRexample.m"

```
table=xlsread('LPRdata');
Wn=xlsread('LPRW');
K1=2;
Y=table(:,1);
X=table(:,2:4);
[table1, table2] = LPRtest(Y, X, Wn, K1);
```

S.3 R codes

"LPRtest.R"

```
setwd("")

LPRtest <- function(Y, X, Wn, K1) {
  n <- nrow(X) #sample size
  K <- ncol(X) #number of regressors
  #automated choice of t
  tx <- rep(0, K1)
  tp <- 0
  k <- n^(1/3) #p in the paper
  X_scaled <- matrix(0, n, K1)
  mpreX <- rep(0, K1)
  for (i in 1:(K1)) {
    mpreX[i] <- mean(X[,i])
    X_scaled[,i] <- X[,i] - mpreX[i]
  }
  pretau <- sum(exp(rowSums(atan(X_scaled))))
  tau <- log(10 * n / pretau)
  tp <- tau / 3.75 #t_y in the paper
  tx <- rep(tau, K1) #t
  funSn <- function(t) solve(diag(n) - t * Wn)
  funGn <- function(t) Wn %*% funSn(t)
  #2SLS estimation
  Z <- matrix(0, n, K+K1)
  Z[,1:K] <- as.matrix(X)
  Z[,(K+1):(K+K1)] <- Wn %*% X[,1:(K1)]
  P <- Z %*% solve(t(Z) %*% Z) %*% t(Z)
  Xex <- cbind(Wn %*% Y, X) #includes Wy
  theta2sls <- solve(t(Xex) %*% P %*% Xex) %*% t(Xex) %*% P %*% Y
  lambda2sls <- theta2sls[1]
  beta2sls <- theta2sls[2:(K+1)]
  ehat_correct <- Y - lambda2sls * Wn %*% Y - X %*% beta2sls
  Cov <- solve(t(Xex) %*% P %*% Xex) %*% t(Xex) %*% P %*%
    diag(diag(ehat_correct %*% t(ehat_correct))) %*% P %*%
    Xex %*% solve(t(Xex) %*% P %*% Xex)
  t_stat_lambda <- lambda2sls / sqrt(Cov[1,1])
  tstat_theta <- solve(sqrt(diag(diag(Cov)))) %*% theta2sls
  wy0 <- rep(0, n)
  wy1 <- rep(0, n)
  prewy0 <- Y
  mpre0 <- mean(prewy0)
```

```
  sigpre0 <- 1
  wy0 <- (prewy0 - mpre0) / sqrt(sigpre0)
  e0_1 <- exp(tx %*% t(atan(X_scaled)))
  e0_2 <- exp(tp * wy0 / k)
  ehat_correct_sq <- ehat_correct^2
  Sigma <- diag(as.vector(ehat_correct_sq))
  B <- matrix(0, K+K1, K+1)
  B <- (1/n) * t(Z) %*% Xex
  A <- matrix(0, K+K1, K+K1)
  A <- (1/n) * t(Z) %*% Z
  Omega <- matrix(0, K+1, K+1)
  Omega <- t(B) %*% solve(A) %*% B
  Q_cc <- cbind(funGn(lambda2sls) %*% X %*% beta2sls, X)
  Q <- funSn(lambda2sls) %*% (diag(n) - (1/n) * Q_cc %*% solve(Omega)
                                %*% t(B) %*% solve(A) %*% t(Z))
  Sbar <- (1/n) * sum(diag(funSn(lambda2sls)))
  Sd <- funSn(lambda2sls) - matrix(rep(Sbar, n), n, n)
  E <- matrix(0, 2, n)
  E[1,] <- e0_1
  E[2,] <- rep(1, n)
  psi1 <- E %*% Q
  V_hat <- psi1 %*% Sigma %*% t(psi1) / n
  Mhat <- matrix(0, 2, 1)
  Mhat_1 <- t(Y - funSn(lambda2sls) %*% X %*% beta2sls) %*% t(e0_1) / n
  Mhat_2 <- t(Y - funSn(lambda2sls) %*% X %*% beta2sls) %*% e0_2 / n -
    (tp / (n * k)) * sum(diag(t(Sd) %*% Q %*% Sigma))
  Mhat[1] <- Mhat_1
  Mhat[2] <- Mhat_2
  That <- n * t(Mhat) %*% solve(V_hat) %*% Mhat
  #Reporting results
  pvalue <- 2 * pnorm(abs(tstat_theta), lower.tail = FALSE)
  pvalueTsq <- pchisq(That, df = 2, lower.tail = FALSE)
  #the first element is lambda
  table1 <-data.frame(theta2sls=theta2sls,tstat_theta=tstat_theta,
     pvalue=pvalue)
  table2 <-data.frame(That=That, pvalueTsq=pvalueTsq)
  list(table1 = table1, table2 = table2)
}
```

“LPRexample.R”

```
setwd("")
source("LPRtest.R")
```

```r
library(readxl)
data <- read_excel("LPRdata.xlsx", col_names = FALSE)
Wn <- read_excel("LPRW.xlsx", col_names = FALSE)
K1<- 2
Y<-as.matrix(data[,1])
X<-as.matrix(data[,2:4])
Wn<-as.matrix(Wn)
LPRtest(Y,X,Wn,K1)
```

S.4 Example Datasets

In the interest of space, we simulated a small dataset with $n = 20$ and $K = 3$. Below we include tables of:

- 20×4 data matrix. Column 1 is the dependent variable Y and Columns 2-3 are two regressors. Column 4 is the vector of ones which is not included in the exponential of the test and the spatial instrument. Hence, we set $K1 = 2$. Save as "LPRdata.xlsx"
- 20×20 weight matrix. Save as "LPRW.xlsx"

6.830	1.134	1.626	1
9.247	2.773	1.876	1
6.733	1.762	1.077	1
5.927	0.627	1.167	1
9.114	2.179	1.831	1
10.973	3.121	3.442	1
7.157	1.225	2.345	1
5.602	0.888	1.134	1
6.077	1.552	1.112	1
9.131	3.746	1.818	1
8.083	3.904	0.822	1
7.333	2.690	0.806	1
10.502	3.611	2.056	1
7.479	3.383	0.349	1
7.895	1.512	1.934	1
6.585	0.369	1.449	1
7.611	2.614	2.831	1
9.185	2.231	2.987	1
7.465	1.446	2.764	1
7.588	0.900	2.757	1

0	0.0787	0	0	0	0	0	0.2222	0.0740	0.1508	0	0	0	0	0.3821	0	0	0	0.0922	0
0.1454	0	0	0	0	0	0	0.4271	0	0	0	0	0	0.3425	0.0851	0	0	0	0	0
0	0	0	0	0.2639	0.0585	0.0423	0	0	0	0.1043	0.0481	0	0	0	0.1172	0.2530	0	0	0.1127
0	0	0	0	0	0	0	0	0.3032	0.1487	0	0	0	0	0.0587	0	0	0.2462	0.2433	0
0	0	0.2724	0	0	0.0569	0.0463	0	0	0	0.1015	0.0526	0	0	0	0.1141	0.2463	0	0	0.1098
0	0	0.0889	0	0.0834	0	0	0	0	0	0.2386	0	0.0574	0	0	0.2123	0.0983	0	0	0.2207
0	0	0.1003	0	0.1063	0	0	0	0	0	0.0374	0.5830	0	0	0	0.0419	0.0907	0	0	0.0404
0.3171	0.3301	0	0	0	0	0	0	0	0.0733	0	0	0	0.0937	0.1857	0	0	0	0	0
0.0549	0	0	0.1740	0	0	0	0	0	0.2376	0	0	0	0	0.0937	0	0	0.0507	0.3888	0
0.1136	0	0	0.0866	0	0	0	0.0387	0.2411	0	0	0	0	0	0.1941	0	0	0.0253	0.3005	0
0	0	0.1005	0	0.0948	0.1513	0.0152	0	0	0	0	0.0173	0.0204	0	0	0.2399	0.1111	0	0	0.2494
0	0	0.1078	0	0.1143	0	0.5516	0	0	0	0.0402	0	0	0	0	0.0451	0.0975	0	0	0.0434
0	0	0	0	0	0.3878	0	0	0	0	0.2175	0	0	0	0	0.1936	0	0	0	0.2012
0	0.7384	0	0	0	0	0	0.2616	0	0	0	0	0	0	0	0	0	0	0	0
0.3337	0.0402	0	0.0396	0	0	0	0.1136	0.1103	0.2249	0	0	0	0	0	0	0	0	0.1375	0
0	0	0.1093	0	0.1031	0.1303	0.0165	0	0	0	0.2323	0.0188	0.0176	0	0	0	0.1209	0	0	0.2511
0	0	0.2511	0	0.2368	0.0642	0.0379	0	0	0	0.1144	0.0432	0	0	0	0.1286	0	0	0	0.1237
0	0	0	0.5484	0	0	0	0	0.1969	0.0966	0	0	0	0	0	0	0	0	0.1580	0
0.0651	0	0	0.1329	0	0	0	0	0.3701	0.2819	0	0	0	0	0.1116	0	0	0.0388	0	0
0	0	0.1051	0	0.0991	0.1353	0.0159	0	0	0	0.2412	0.0181	0.0183	0	0	0.2509	0.1162	0	0	0

S.5 Output tables

Matlab output

```
Command Window
New to MATLAB? See resources for Getting Started.
>> LPRexample

table1 =

  4×3 table

    theta2sls    tstat_theta      pvalue
    _________    ___________    __________

     0.27908        1.0323         0.30191
     0.95476        8.9792      2.7268e-19
      1.0302        5.4002      6.6562e-08
      1.8376       0.81297         0.41624


table2 =

  1×2 table

     That      pvalueTsq
    _______    _________

    0.12896     0.93755

fx >>
```

R output

```
Console   Terminal ×   Background Jobs ×
R 4.3.1 · C:/Users/ucte002/Dropbox/codes/
> LPRtest(Y,X,Wn,K1)
$table1
          ...1    ...1.1        ...1.2
...1 0.2790818 1.032350 3.019082e-01
...2 0.9547612 8.979227 2.726769e-19
...3 1.0302305 5.400213 6.656195e-08
...4 1.8375980 0.812968 4.162364e-01

$table2
       That pvalueTsq
1 0.1034594 0.9495855

>
```