Monte Carlo Simulations

Natural Amenities, Neighborhood Dynamics, and Persistence in the Spatial Distribution of Income by Sanghoon Lee and Jeffrey Lin

---

Functions to Generate Data

In[1]:=
```
nextR[prevR_, ia_, re_, numNeighborhoods_, {beta0_, beta1_, beta2_, sdShock_}] :=
  Module[{ones, shocks},
   ones = ConstantArray[1, numNeighborhoods];
   shocks = RandomReal[NormalDistribution[0, sdShock], {1, numNeighborhoods}];
   Flatten[{{beta0, beta1, (1 + beta2)}}.{ones, ia, prevR} + {re} + shocks]];
```

In[2]:=
```
genData[{numNeighborhoods_, numPeriods_, numPeriodsUsed_},
  {probNf_, probIaNf_, reSD_}, {beta0_, beta1_, beta2_, sdShock_}, thresR_] :=
 Module[{nf, ia, re, initR, ones, shocks, rHistory, dRHistory, oneColumn},
  nf = RandomInteger[BernoulliDistribution[probNf], numNeighborhoods];
  ia = nf * RandomInteger[BernoulliDistribution[probIaNf], numNeighborhoods];
  re = If[reSD == 0, ConstantArray[0, numNeighborhoods],
    RandomReal[NormalDistribution[0, reSD], numNeighborhoods]];
  initR = RandomReal[UniformDistribution[{0, 1}], numNeighborhoods];
  rHistory = NestList[nextR[#, ia, re, numNeighborhoods, {beta0, beta1, beta2, sdShock}] &,
    initR, numPeriods];
  dRHistory = rHistory[[2 ;; -1, All]] - rHistory[[1 ;; -2, All]];
  oneColumn = ConstantArray[1., {numPeriods, 1}];
  Transpose[Flatten[Transpose[#[[-numPeriodsUsed ;; -1]]]] & /@
    {oneColumn.{Range[numNeighborhoods]}, Transpose[
      ConstantArray[1., {numNeighborhoods, 1}].{Range[numPeriods]}], oneColumn.{nf},
     oneColumn.{ia}, rHistory[[1 ;; -2, All]], UnitStep[rHistory[[1 ;; -2, All]] -
        (Quantile[#, thresR] & /@ rHistory[[1 ;; -2, All]])] // N, dRHistory}]]
```

Functions to Run Simulations

In[3]:=
```
OneSim[{numNeighborhoods_, numPeriods_, numPeriodsUsed_},
  {probNf_, probIaNf_, reSD_}, {beta0_, beta1_, beta2_, sdShock_}, thresR_] :=
 Module[{temp, data, y, xIa, bIa, xNf, bNf, xNfTop, bNfTop, corrNfIa, corrNfIaTop},
  temp = Transpose[genData[{numNeighborhoods, numPeriods, numPeriodsUsed},
     {probNf, probIaNf, reSD}, {beta0, beta1, beta2, sdShock}, thresR]];
  data = (temp // Insert[#, ConstantArray[1., Dimensions[temp][[2]] ], 1] & //
      Insert[#, temp[[3]] * temp[[6]], -2] & // Transpose);
  y = data[[All, -1]];
  xIa = data[[All, {1, 5, 6}]];
  bIa = Inverse[Transpose[xIa].xIa].Transpose[xIa].y;
  xNf = data[[All, {1, 4, 6}]];
  bNf = Inverse[Transpose[xNf].xNf].Transpose[xNf].y;
  xNfTop = data[[All, {1, -2, 6}]];
  bNfTop = Inverse[Transpose[xNfTop].xNfTop].Transpose[xNfTop].y;
  corrNfIa = Correlation @@ Transpose[data[[All, {4, 5}]]];
  corrNfIaTop = Correlation @@ Transpose[Select[data, #[[-3]] == 1 &][[All, {4, 5}]]];
  {{{numNeighborhoods, numPeriods, numPeriodsUsed}, {probNf, probIaNf},
    {beta0, beta1, beta2, sdShock}, thresR}, {corrNfIa, corrNfIaTop}, {bIa, bNf, bNfTop},
   {Dimensions[xIa][[1]], Dimensions[xNf][[1]], Dimensions[xNfTop][[1]]}}]
```

In[4]:=
```
runSim[{{numNeighborhoods_, numPeriods_, numPeriodsUsed_},
   {probNf_, probIaNf_, reSD_}, {beta0_, beta1_, beta2_, sdShock_}, thresR_}, rep_] :=
 Module[{nObs, corrMeans, bMeans, corrStds, bStds, simResults},
  simResults = ParallelTable[OneSim[{numNeighborhoods, numPeriods, numPeriodsUsed},
     {probNf, probIaNf, reSD}, {beta0, beta1, beta2, sdShock}, thresR], {rep}];
  corrMeans = simResults[[All, 2]] // Mean;
  bMeans = simResults[[All, 3]] // Mean;
  corrStds = simResults[[All, 2]] // StandardDeviation;
  bStds = simResults[[All, 3]] // StandardDeviation;
  nObs = simResults[[All, 4]] // Mean;
  {{{numNeighborhoods, numPeriods, numPeriodsUsed, rep},
    {probNf, probIaNf, reSD}, {beta0, beta1, beta2, sdShock}, thresR},
   {{corrMeans, corrStds}, {bMeans, bStds, nObs}}}
 ]
```

In[5]:=
```
formatSim[runSimOutput_] := Map[Round[#, 0.0001] &,
  {runSimOutput[[1, -1]], runSimOutput[[1, 2, 3]], Grid@{runSimOutput[[2, 1, 1]]},
   Grid@{runSimOutput[[2, 2, 1]][[All, 2]], runSimOutput[[2, 2, 2]][[All, 2]]}}, {2}]
```

Running Simulations

In[6]:=
```
RandomSeed[0];
```

In[7]:=
```
With[{rep = 1000},
 output1 =
  MapThread[runSim[{{50000, 20, 5}, {0.05, 0.5, #1}, {0, 0.1, -0.1, 0.05}, #2}, rep] &,
   {{0, 0.01, 0.02}, {0, 0, 0}}];
 output2 = MapThread[runSim[{{50000, 20, 5}, {0.05, 0.5, #1}, {0, 0.1, -0.1, 0.05}, #2},
     rep] &, {{0, 0, 0, 0}, {0.9, 0.7, 0.5, 0}}];
 output3 = MapThread[runSim[{{50000, 20, 5}, {0.05, 0.5, #1}, {0, 0.1, -0.1, 0.05}, #2},
     rep] &, {{0, 0.01, 0.02, 0, 0.01, 0.02, 0, 0.01, 0.02},
    {0.9, 0.9, 0.9, 0.7, 0.7, 0.7, 0.5, 0.5, 0.5}}];]
```

Generating Tables

In[8]:=
```
incCutoff[output_] := output[[1, -1]]
reSD[output_] := output[[1, 2, 3]]
corr[output_] := Round[#, 0.001] &@output[[2, 1, 1]]
corrSE[output_] := Round[#, 0.001] &@output[[2, 1, 2]]
beta[output_] := Round[#, 0.001] &@output[[2, 2, 1]][[All, 2]]
betaSE[output_] := Round[#, 0.001] &@output[[2, 2, 2]][[All, 2]]
```

In[14]:=
```
({incCutoff[#], ToString[beta[#][[3]]] <> "(" <> ToString[betaSE[#][[3]]] <> ")",
     ToString[corr[#][[2]]] <> "(" <> ToString[corrSE[#][[2]]] <> ")",} & /@
   Reverse[output2]) // Prepend[#, {"θ̃H", "β̂1", "ρAF|H"}] & // Grid
```

Out[14]=

$\tilde{\theta}_H$	$\hat{\beta}_1$	$\rho_{AF\|H}$
0	0.032(0.001)	0.698(0.007)
0.5	0.049(0.001)	0.809(0.006)
0.7	0.064(0.001)	0.872(0.006)
0.9	0.087(0.001)	0.951(0.004)

In[15]:=
```
({reSD[#], ToString[beta[#][[1]]] <> "(" <> ToString[betaSE[#][[1]]] <> ")"} & /@
   output1) // Prepend[#, {"σu", "β̂1"}] & // Grid
```

Out[15]=

σ_u	$\hat{\beta}_1$
0	0.1(0.001)
0.01	0.069(0.001)
0.02	0.036(0.001)

In[16]:=
```
listIncCutoff = Union[incCutoff /@ output3];
listReSD = Union[reSD /@ output3];
betaH[idxInc_, idxRe_] :=
 (ToString[beta[#][[3]]] <> "(" <> ToString[betaSE[#][[3]]] <> ")" & /@ Select[output3,
     (incCutoff[#] == listIncCutoff[[idxInc]] && reSD[#] == listReSD[[idxRe]]) &])[[1]]
Array[betaH, {3, 3}] // MapThread[Prepend, {#, listIncCutoff}] & //
  Prepend[#, Prepend[listReSD, "θ̂H\\σui"]] & // Grid
```

Out[19]=

$\hat{\theta}_H \backslash \sigma_{u_i}$	0	0.01	0.02
0.5	0.049(0.001)	0.037(0.001)	0.021(0.001)
0.7	0.064(0.001)	0.047(0.001)	0.026(0.001)
0.9	0.088(0.001)	0.062(0.001)	0.033(0.001)