Resampling worksheet for sensitivity analysis

In[ ]:=
```
(*Get directory of notebook*)
dir = NotebookDirectory[];
(*Set working directory to 'dir'*)
SetDirectory[dir];
(*load posterior samples of movement probs*)
pPSI = Import["psi_bhcp.csv", "CSV", "HeaderLines" → 1];
```

In[ ]:=
```
Dimensions[pPSI]
```

In[ ]:=
```
(*create function to turn sampline into a
 transition matrix*)
getpsi[x_] :=
 ArrayReshape[{0, pPSI[[x, 1 ;; 6]], 0,
    pPSI[[x, 7 ;; 12]], 0, pPSI[[x, 13 ;; 18]], 0,
    pPSI[[x, 19 ;; 24]], 0, pPSI[[x, 25 ;; 30]], 0},
   {6, 6}]ᵀ
```

In[ ]:=
```
μpsi = Median[pPSI]
```

In[ ]:=
```
meanpsi =
 ArrayReshape[{0, μpsi[[1 ;; 6]], 0, μpsi[[7 ;; 12]],
    0, μpsi[[13 ;; 18]], 0, μpsi[[19 ;; 24]], 0,
    μpsi[[25 ;; 30]], 0}, {6, 6}]ᵀ
```

In[ ]:=
```
meanpsi // MatrixForm
```

```
phiests = meanpsi
```

```
(*Create month based matrix*)
```

In[•]:=
```
A1 = Table[ϕ[i, j], {j, 6, 1, -1}, {i, 6, 1, -1}]

(*Replace the diagonal elements with 1-
 off diagonal*)
A1[[1, 1]] = S * (1 - (Total[A1[[2 ;; 6, 1]]]))
A1[[6, 6]] = S * (1 - Total[A1[[1 ;; 5, 6]]])
A1[[2, 2]] =
 S * (1 - Total[Flatten[{A1[[1, 2]], A1[[3 ;; 6, 2]]}]])
A1[[3, 3]] =
 S *
  (1 -
    Total[Flatten[{A1[[1 ;; 2, 3]], A1[[4 ;; 6, 3]]}]])
A1[[4, 4]] =
 S *
  (1 -
    Total[Flatten[{A1[[1 ;; 3, 4]], A1[[5 ;; 6, 4]]}]])
A1[[5, 5]] =
 S * (1 - Total[Flatten[{A1[[1 ;; 4, 5]], A1[[6, 5]]}]])
```

In[•]:=
```
(*Create seasonal matrix*)
A2 = A1.A1.A1;
```

In[•]:=
```
(*Create full matrix*)
A3 =
  A2.A2.(A2 + DiagonalMatrix[{b6, b5, b4, b3, b2, b1}]).
   A2;
vars = Drop[Flatten[A1], {1, 36, 7}];

(*Create function to sample posterior and
 calculate sensitive of each ϕ*)
```

```
(*test*)
ind = 34;
phiests = getpsi[ind];
params =
  Flatten[Table[ϕ[i, j] → phiests[[7 - j, 7 - i]],
    {j, 6, 1, -1}, {i, 6, 1, -1}]];
params2 = Join[Drop[params, {1, 36, 7}], {S → .99}];
A1 /. params2 // MatrixForm

(*Resampling function for transition probs*)
sensResamp = Compile[{{bz, _Real}}, Block[{ind},
   ind = RandomInteger[{1, 30 000}];
   phiests = getpsi[ind];
   params =
    Flatten[Table[ϕ[i, j] → phiests[[7 - j, 7 - i]],
      {j, 6, 1, -1}, {i, 6, 1, -1}]];
   params2 = Join[Drop[params, {1, 36, 7}], {S → .99}];
   Do[
    ev[i] = Eigenvalues[
      A3 /. Dispatch[
        Flatten[{{b3 → 0, b2 → 0, b1 → 0, b4 → bz,
           b5 → bz, b6 → bz}, Drop[params2, {i}]}]]],
    {i, 1, 30}];
   pos = Flatten[Table[Ordering[ev[i] /. params2, -1],
      {i, 1, 30}]];
   sens = Table[D[ev[i][[pos[[i]]]], vars[[i]]] /.
      params2, {i, 1, 30}];
   Return[sens]
  ]]

(*run it for 50 samples becasue soooo slowwww
 about 2 min each*)
```

```
iter = 50;

Do[sphi[i] = sensResamp[.14], {i, 1, iter}]


(*Put results in table form*)

In[•]:= sphitab = Table[sphi[i], {i, 1, iter}];

λ1 =
 Eigenvalues[
   A3 /.
    Dispatch[
     Flatten[{{b3 → 0, b2 → 0, b1 → 0, b4 → .14,
        b5 → .14, b6 → .14}, params2}]]][[1]]

(*Calculate mean and standard error of regional
 elasticity*)

In[•]:= elast = Mean[(sphitab^T * μpsi / λ1)^T]
elastse = StandardDeviation[(sphitab^T * μpsi / λ1)^T] /
  Sqrt[iter]

(*Calculate mean and standard error of regional
 sensitivities*)

In[•]:= sensitivites = Mean[sphitab]
sensitivitesse =
 StandardDeviation[sphitab] / Sqrt[iter]

(*output the results*)
Export["psi_sensi.csv",
 {vars, sensitivites, sensitivitesse, elast, elastse},
 "CSV"]
```

```
(*function for doing sensitivity of recruitment
 rates b*)
sensResampl[bz_] := Block[{ind},
  ind = RandomInteger[{1, 30 000}];
  phiests = getpsi[ind];
  params =
   Flatten[Table[ϕ[i, j] → phiests[[7 - j, 7 - i]],
     {j, 6, 1, -1}, {i, 6, 1, -1}]];
  params2 = Join[Drop[params, {1, 36, 7}], {S → .99}];
  btab = {b6, b5, b4};
  evb = Eigenvalues[
    A3 /. Dispatch[
      Flatten[{{b3 → 0, b2 → 0, b1 → 0}, params2}]]];
  pos =
   Flatten[
    Table[Ordering[evb /. params2 /.
       {b4 → bz, b5 → bz, b6 → bz}, -1], {i, 1, 3}]];
  sensb =
   Table[D[evb[[pos[[i]]]], btab[[i]]] /. params /.
     {b4 → bz, b5 → bz, b6 → bz}, {i, 1, 3}];
  Return[sensb]
 ]

Do[sb[i] = sensResampl[.14], {i, 1, iter}]

(*Put in table form*)
sbtab = Table[sb[i], {i, 1, iter}];
```

In[•]:=
```
Mean[sbtab]
StandardDeviation[sbtab] / Sqrt[iter]
```

In[•]:=
```
Mean[sbtab / λ1]
StandardDeviation[sbtab / λ1] / Sqrt[iter]
```

```
(*output the results*)
Export["b_sensi.csv",
 {btab, Mean[sbtab],
  StandardDeviation[sbtab] / Sqrt[iter],
  Mean[.14 * sbtab / λ1],
  StandardDeviation[.14 * sbtab / λ1] / Sqrt[iter]},
 "CSV"]

(*local *)
```

In[•]:=
```
Length[vars]
```

```
(*Function for calculating local sesitivity*)

spsilocal := Block[{ind},
  ind = RandomInteger[{1, 30 000}];
  phiests = getpsi[ind];
  params =
   Flatten[Table[ϕ[i, j] → phiests[[7 - j, 7 - i]],
     {j, 6, 1, -1}, {i, 6, 1, -1}]];
  params2 = Join[Drop[params, {1, 36, 7}], {S → .99}];
sens1 =
   Table[
    D[A3[[6]] /. Dispatch[
          Flatten[{{b3 → 0, b2 → 0, b1 → 0, b4 → .14,
             b5 → .14, b6 → .14}, Drop[params2, {i}]}],
          {i, 1, 30}], vars[[i]]] /. params2 /.
      {b4 → .14, b5 → .14, b6 → .14}, {i, 1, 30}];
  Return[sens1]]
```

In[•]:=
```
Do[spil[i] = spsilocal, {i, 1, iter}]
```

```
r2 = A3[[6]] /. μparams2 /.
  {b3 → 0, b2 → 0, b1 → 0, b4 → .14, b5 → .14, b6 → .14,
   S → .99}
μparams =
  Flatten[Table[ϕ[i, j] → meanpsi[[7 - j, 7 - i]],
    {j, 6, 1, -1}, {i, 6, 1, -1}]];
μparams2 = Drop[μparams, {1, 36, 7}]

(*Calc Mean and standard error local sensitivity
 and elasticity*)
```

In[•]:=
```
spilmean = Table[Mean[Table[spil[i][[j]], {i, 1, 50}]],
   {j, 1, 30}];
spilse =
  Table[
   StandardDeviation[Table[spil[i][[j]], {i, 1, 50}]] /
    Sqrt[iter], {j, 1, 30}];
```

In[•]:=
```
espilmean =
  Table[
   Mean[Table[spil[i][[j]] / r2 * vars[[j]] /. μparams2,
     {i, 1, 50}]], {j, 1, 30}];
espilse =
  Table[
   StandardDeviation[
     Table[spil[i][[j]] / r2 * vars[[j]] /. μparams2,
      {i, 1, 50}]] / Sqrt[iter], {j, 1, 30}];


(*export the results*)
```

```
Export["phi_sensi_local.csv", spilmean, "CSV"]
Export["phi_sensi_local_se.csv", spilse, "CSV"]
Export["phi_elast_local.csv", espilmean, "CSV"]
Export["phi_eslast_local_se.csv", espilse, "CSV"]
```

Make tables of results for inspection

In[•]:=
```
rnames =
 Drop[Flatten[Table[ϕi,j, {j, 6, 1, -1}, {i, 6, 1, -1}]],
  {1, 36, 7}]
```

In[•]:=
```
TableForm[Round[spilmean, 0.001],
 TableHeadings →
  {rnames, {"r61", "r51", "r41", "r31", "r21", "r11"}}]
```

In[•]:=
```
TableForm[Round[espilmean, .001],
 TableHeadings →
  {rnames, {"r61", "r51", "r41", "r31", "r21", "r11"}}]
```

```
(*Function for sensitivity of recritment on
 local rates*)
```

```
sblocal := Block[{ind},
  ind = RandomInteger[{1, 30 000}];
  phiests = getpsi[ind];
  params =
   Flatten[Table[ϕ[i, j] → phiests[[7 - j, 7 - i]],
     {j, 6, 1, -1}, {i, 6, 1, -1}]];
  params2 = Join[Drop[params, {1, 36, 7}], {S → .99}];
  btab = {b6, b5, b4};
  sensbl =
   Table[
    D[A3[[6]] /. Dispatch[
         Flatten[{{b3 → 0, b2 → 0, b1 → 0}, params2}]],
       btab[[i]]] /. params2 /.
     {b4 → .14, b5 → .14, b6 → .14}, {i, 1, 3}];
  Return[sensbl]]
```

In[•]:=
```
sblocal
```

In[•]:=
```
Do[sbl[i] = sblocal, {i, 1, iter}]
```

In[•]:=
```
Table[Mean[Table[sbl[i][[j]], {i, 1, 50}]],
 {j, 1, 3}]
```

In[•]:=
```
Table[Mean[Table[sbl[i][[j]] / r2, {i, 1, 50}]],
 {j, 1, 3}]
```

```
(*Calculate Mean and SE of sensitivites and
 elasticities on local rates*)
```

In[•]:=
```
sblmean = Table[Mean[Table[sbl[i][[j]], {i, 1, 50}]],
    {j, 1, 3}];
sblse =
  Table[
   StandardDeviation[Table[sbl[i][[j]], {i, 1, 50}]] /
    Sqrt[iter], {j, 1, 3}];

eblmean =
  Table[Mean[Table[.14 * sbl[i][[j]] / r2, {i, 1, 50}]],
   {j, 1, 3}];
eblse =
  Table[
   StandardDeviation[Table[.14 * sbl[i][[j]] / r2,
      {i, 1, 50}]] / Sqrt[iter], {j, 1, 3}];


(*export the results*)

Export["b_sensi_local.csv", sblmean, "CSV"]
Export["b_sensi_local_se.csv", sblse, "CSV"]
Export["b_elast_local.csv", eblmean, "CSV"]
Export["b_eslast_local_se.csv", eblse, "CSV"]
```