Assigment Games Maximum Score for the C Block PCS Auction

Jeremy T. Fox
March 11, 2005
March 23, 2005: payoffs non-additive across matches
August 11, 2005: exchange two licenses at once
August 19, 2005: oversample adjacent packages
August 20, 2005: put one license exchanges back in
November 3, 2006: jump bids only last rivals
March 1, 2007: HT style bounds using price data
May 20, 2009: Check whether pairwise stable equilibrium exists
June 9, 2010: Make sure winnings always under eligibility

- **Licenses are the agents in a two-sided matching market**

Normalization is coefficient on pop*eligibility is 1.

Opening Stuff

Packages

(L2) In[1]:=
```
SetDirectory[NotebookDirectory[]];
```

(L2) In[2]:=
```
<< MatchEstimation`;
```

```
General::obspkg :
  Histograms` is now obsolete. The legacy version being loaded may conflict with current Mathematica
      functionality. See the Compatibility Guide for updating information. ≫

BarChart3D::shdw : Symbol BarChart3D appears in multiple contexts {BarCharts`,
      System`}; definitions in context BarCharts` may shadow or be shadowed by other definitions. ≫

Histogram3D::shdw : Symbol Histogram3D appears in multiple contexts {Histograms`,
      System`}; definitions in context Histograms` may shadow or be shadowed by other definitions. ≫
```

(L2) In[3]:=
```
<< Developer`;
```

Read in Data - licenses

(L2) In[4]:=
```
rationals = 0.0000000000000000001;
(* SetDirectory["/Users/fox/Documents/Phones/FCC/Talks/SITE July 2004/Empirical"];  *)

SetDirectory[ToFileName[Extract[
    "FileName" /. NotebookInformation[EvaluationNotebook[]], {1}, FrontEnd`FileName]]];
SetDirectory["data"];

(*SetDirectory["Z:\Documents\Phones\FCC\Talks\SITE July 2004\Empirical"]; *)
marketdata = Import["btadata_2004_03_12_1.csv", "CSV"];
marketdata = Delete[marketdata, 1];


bidderdata = Import["biddercblk_03_28_2004_pln.csv", "CSV"];
bidderdata = Delete[bidderdata, 257]; (* FCC *)
bidderdata = Delete[bidderdata, 1];

(* There is a data entry problem where the winner
 of DCR's licenses is falsely labeled as bidder 190, DCC.
  DCC was a small company that won no licenses. So the
 easiest solution is just to call bidder 190 DCR
 instead of DCC. *)

bidderdataDCC = bidderdata[[190]];
bidderdata[[190]] = bidderdata[[234]];
bidderdata[[234]] = bidderdataDCC;


 bidders = Length[bidderdata];
observations = bidders;
eligibilitydata  = Rationalize[bidderdata[[All, 4]], rationals] / 1 000 000;



(* SetDirectory["/Users/fox/Documents/Phones/FCC/Talks/SITE July 2004/Empirical"]; *)
adjacencymatrixpre = Import["btamatrix_merged.csv", "CSV"];
adjacencymatrix = Transpose[Delete[Transpose[adjacencymatrixpre], 1]];

(* bidderelig = bidderdata[[All,4]] /1000; *)

licenses = Length[marketdata];
populationdata = Rationalize[marketdata[[All, 6]], rationals] / 1 000 000;
pricedata = Rationalize[marketdata[[All, 3]], rationals] / 1 000 000;
microwavedata = Rationalize[marketdata[[All , 10]], rationals];
microwavedata = microwavedata / 100;
householdover35thous = Rationalize[marketdata[[All, 8]], rationals];
winnerdata = marketdata[[All, 5]];
householdover35thouspopulationdata = householdover35thous populationdata;
```

```
(* SetDirectory["/Users/fox/Documents/Phones/FCC/Talks/SITE July 2004/Empirical"]; *)
btadistancesdio = Import["distancesmat_dio_perl_fixed.dat", "Table"];
btadistancesdio = Rationalize[btadistancesdio, rationals];
btadistancesdiomiles = btadistancesdio / 1.609;

packages = 2^licenses;
packagesamplenum = 250;
observations = bidders;
fcclimit = 98;

datalicensechoices = Table[Flatten@ Position[winnerdata, i], {i, bidders}];
winningpackages = Select[datalicensechoices, # ≠ {} &];
uniquewinners =
  Flatten@Table[Flatten@Position[datalicensechoices, winningpackages[[p]] ],
    {p, 1, Length[winningpackages] }];
uniquewinnersnum = Length[uniquewinners];

(* extra stats - new this file *)

(* SetDirectory["/Users/fox/Documents/Phones/FCC/Talks/SITE July 2004/Empirical"]; *)
marketstats = Import["btastatsexport.csv", "CSV"];
marketstats[[1]];
marketstats = Delete[marketstats, 1];
densitydata = marketstats[[All, 8]];
```

- **BTA's to use**

(L2) In[43]:=

```
(* gets rid of BTAs with no Todd distance calculation *)

btastouse = Complement[ Range[487], {14, 136, 221, 190, 192, 222, 254}];
btastousenum = Length[btastouse];

winningpackagesusableincnull = Table[ Complement[winningpackages[[p]],
    Complement[Range[licenses], btastouse] ], {p, uniquewinnersnum}];
winningpackagesusable = Select[Union[winningpackagesusableincnull], # ≠ {} &];
winningpackagesusablenum = Length[winningpackagesusable];
winningpackagesusablesizes
  =  Table[Length[winningpackagesusable[[p]] ], {p, winningpackagesusablenum}];




nouse = Complement[Range[licenses], btastouse];
winningpositions =
  Flatten[Table[Last[Position[winningpackagesusable, winningpackagesusable[[p]] ] ],
    {p, winningpackagesusablenum}] ];
datalicensechoicesusable = Table[Complement[datalicensechoices[[i]], nouse],
   {i, observations}];

(* This creates an error, it is not so important *)
bidderpositions = Flatten@
   Table[ If[Position[winningpackagesusable, datalicensechoicesusable[[i ]] ] ≠ {},
     Last[Position[winningpackagesusable, datalicensechoicesusable[[i ]]] ],
     {}], {i, observations}];

uniquewinnersusable =
  Flatten@Table[Flatten@Position[datalicensechoicesusable, winningpackagesusable[[p]] ],
    {p, 1, winningpackagesusablenum }];
uniquewinnersusablenum = Length[uniquewinnersusable];

enclosingpackage =
  Table[Position[uniquewinnersusable, winnerdata[[ btastouse[[b]] ]]][[1, 1]] ,
   {b, btastousenum}];
```

- **Reverse lookup for BTAstouse**

(L2) In[56]:=
```
btatouselookupbta = Table[ Module[ {pos},
     pos = Position[btastouse, b]; If[pos == {}, {}, pos[[1, 1]] ] ] , {b, licenses}] ;
```

Bidder Positions

Key to link up bidders to the dataset I just created

Read in Assset and Revenue Data

Note that these variables have all missing values set to zero.

Fix up "South Central Utah Telephone Association" eligibility data

(L2) In[57]:=
```
(* SetDirectory[
  "/Users/fox/Documents/Phones/FCC/BTAData/C Block Form 175s Ali Manning"]; *)
form175data = Import["fccform175nomiss.csv"];
assetsdata = form175data[[All, 2]] / 1 000 000 // N;
revenuedata = form175data[[All, 3]] / 1 000 000 // N;

bidderdata[[67, 4]] = form175data[[67, 4]];
eligibilitydata[[67]] = Rationalize[form175data[[67, 4]], rationals] / 1 000 000;
```

Read in synergy data

(L2) In[62]:=
```
(* SetDirectory["/Users/fox/Documents/Phones/FCC/BTAData/Pat_Package_2004_10_05"]; *)
geographicdistancematrix =
  Import["geographic distance population weighted centroid.csv"];
geographicdistancematrixkm = geographicdistancematrix / 1000;
airlinetravelmatrix = Import["air travel passengers bta year 1994.csv"];
tripsmatrix = Import["american travel survey 1995 zero.csv"];
adjacencymatrix = Import["adjacency bta.csv"];
```

Eligibility at each round & end of the auction

(L2) In[67]:=
```
(* SetDirectory[
  "/Users/fox/Documents/Phones/FCC/BTAData/Eligibility By Round Fox/"]; *)
eligibilitybyrounddata = Import["cblock eligibility round matrix 1.csv"];
eligibilitybyrounddata = Delete[eligibilitybyrounddata, 1];
eligibilitybyrounddata2 = Transpose[ Delete[Transpose[eligibilitybyrounddata], 1] ];
Dimensions[eligibilitybyrounddata2]
initialeligfromrounds = eligibilitybyrounddata2[[All, 1]];
finalelig = eligibilitybyrounddata2[[All, 181]];
```

(L2) Out[70]= {255, 181}

Look for errors in winners

(L2) In[73]:=
```
dataCheckWinners =
  Table[ {l, marketdata⟦l, 2⟧, {marketdata⟦l, 4⟧} , {bidderdata⟦marketdata⟦l, 5 ⟧, 3 ⟧},
    marketdata⟦l, 5⟧, bidderdata⟦marketdata⟦l, 5 ⟧, 1 ⟧}, {l, 1, 493}];
```

(L2) In[74]:=
```
Union[dataCheckWinners⟦All, {3, 4}⟧ ]
```

(L2) Out[74]=
```
{{{21Centur}, {21st Century Telesis Joint Venture}},
 {{AerForce}, {Aer Force Communications, L.P.}},
 {{Alabama}, {Central Alabama Partnership L.P. 132}},
 {{Alpine}, {Alpine PCS, Inc.}}, {{Americal}, {Americall International, L.L.C.}},
 {{Anishnab}, {Anishnabe Communications Enterprise, Inc.}}, {{BDPCS}, {BDPCS, Inc.}},
 {{BRK}, {BRK WIRELESS CO., INC.}}, {{Brooking}, {Brookings Municipal Utilities}},
 {{Carolina}, {Carolina PCS I Limited Partnership}},
 {{Cellutec}, {CELLUTECH}}, {{ChaseTel}, {Chase Telecommunications L.P.}},
 {{Chillico}, {The Chillicothe Telephone Co.-Communications, Inc.}},
 {{CHPCS}, {CH PCS, Inc.}}, {{Comtel}, {Comtel PCS Mainstreet Limited Partnership}},
 {{ComVent}, {Communications Venture PCS Limited Partnership}},
 {{CookWest}, {Cook Inlet Western Wireless PV/SS PCS, L.P.}},
 {{DCR}, {DCR PCS, Inc.}}, {{Devon}, {Devon Mobile Communications, L.P.}},
 {{Eldorado}, {Eldorado Communications, L.L.C.}},
 {{Enterpri}, {Enterprise Communications Partnership}}, {{FAMS}, {FAMS & ASSOCIATES}},
 {{Fortunet}, {Fortunet Wireless Communications, L.P.}},
 {{Georgia}, {Georgia Independent PCS Corporation}},
 {{Global}, {GLOBAL INFORMATION TECHNOLOGIES, INC.}}, {{GuestMar}, {Mark M. Guest}},
 {{GWI}, {GWI PCS, Inc.}}, {{HighCnty}, {High Country Communications, L.P.}},
 {{Indus}, {Indus, Inc.}}, {{Integrat}, {Integrated Communications Group Corporation}},
 {{Kansas}, {Kansas Personal Communication Services, LTD}},
 {{Kentucky}, {Third Kentucky Cellular Corporation}}, {{KMTel}, {KMTel L.L.C.}},
 {{LongSt}, {Longstreet Communication International, Inc}},
 {{Loralen}, {LORALEN CORP.}}, {{MBO}, {MBO Wireless, Inc.}},
 {{McBride}, {Vincent  D. McBride}}, {{MCG}, {MCG PCS, Inc.}},
 {{MercuryP}, {Mercury PCS, L.L.C.}}, {{Meretel}, {Meretel Communications, LP}},
 {{MFRI}, {MFRI Inc.}}, {{Miccom}, {Miccom Associates, Ltd.}},
 {{MountSol}, {Mountain Solutions, Ltd}}, {{National}, {National Telecom Holdings, Inc.}},
 {{NDakota}, {New Dakota Investment Trust}},
 {{NEngldW}, {New England Wireless Communications, L.P.}},
 {{NextWave}, {NextWave Personal Communications Inc.}},
 {{NMich}, {Northern Michigan PCS Consortium L.L.C.}},
 {{NoverrPb}, {NOVERR PUBLISHING INC.}}, {{NWaveP}, {New Wave PCS, Inc.}},
 {{Omnipoin}, {Omnipoint PCS Entrepreneurs, Inc.}},
 {{OnQue}, {OnQue Communications, Inc.}},
 {{PCS2000}, {PCS 2000, L.P.}}, {{PCSMob}, {PCS Mobile America, Inc.}},
 {{PCSOne}, {PCS One, Inc.}}, {{PCSouth}, {PCSouth, Inc.}},
 {{PCSPlus}, {PCS Plus, LLC An Arizona Limited Liability Company}},
 {{PerComN}, {Personal Communications Network, Inc.}},
 {{PokaLam}, {Poka Lambro PCS, Inc.}},
 {{PokaWire}, {Poka Lambro/PVT Wireless Limited Partnership}},
 {{Polycell}, {POLYCELL COMMUNICATIONS, INC.}},
 {{PVTWire}, {PVT Wireless Limited Partnership}},
 {{Quantum}, {Quantum Communications Group, Inc.}},
 {{Reserve}, {Reserve Telephone Company, Inc.}},
 {{RFW}, {R.F.W. Inc.}}, {{RLV}, {RLV-PCS I PARTNERSHIP}},
 {{Roberts}, {ROBERTS-ROBERTS & ASSOCIATES, LLC}}, {{Rosas}, {Rosas, Inc.}},
 {{R&S}, {R & S PCS, Inc.}}, {{RTCom}, {RT  Communications, Inc.}},
 {{Savannah}, {Savannah Independent PCS Corporation}},
 {{SComSys}, {SOUTHERN COMMUNICATIONS SYSTEMS, INC.}},
 {{SETel}, {SouthEast Telephone Limited Partnership, Ltd.}},
 {{SEWire}, {Southeast Wireless Communications, L.P.}},
 {{SOwega}, {SOWEGA Wireless Communications, L.P.}},
 {{SWireles}, {Southern Wireless, L.P.}},
 {{SWMinn}, {Southwest Minnesota PCS Limited Partnership}},
 {{TriState}, {Mobile Tri-States L.P. 130}}, {{TWS}, {TWS, LLC}},
 {{UrbanC}, {Urban Communicators PCS Limited Partnership}},
 {{UsaMicro}, {USA Micro-Cellular, Inc.}},
 {{Virginia}, {Virginia PCS Alliance Consortium}},
 {{Windkeep}, {Windkeeper Communications, Inc.}},
 {{Wire2000}, {Wireless 2000, Inc.}}, {{WirePCS}, {Wireless PCS, Inc.}},
 {{WireTelC}, {WIRELESS TELECOM, INC.}}, {{WireVent}, {WIRELESS VENTURES, INC.}},
 {{WMinn}, {Western Minnesota PCS Limited Partnership}}}
```

Matching with prices data

All pairwise combinations of one license each, in continental United States

Package combinations

Differemt winning packages

License swaps

(L2) In[75]:=
```
combinationsoflicensesPreRobust = Permutations[ btastouse, {2}];
licenseswapsonewayPreRobust = Select[ combinationsoflicensesPreRobust,
   enclosingpackage[[btatouselookupbta[[ #[[1]]]] ]] ≠
     enclosingpackage[[ btatouselookupbta[[#[[2]] ]] ]] & ];
licenseswapsonewaynumPreRobust = Length[licenseswapsonewayPreRobust];
```

Resulting packages

Take away license 1 and add license 2

(L2) In[78]:=
```
packagesswapsonewayPreRobust =
  Table[ Union[Complement[ winningpackagesusable[[enclosingpackage[[
       btatouselookupbta[[ licenseswapsonewayPreRobust[[l, 1]] ]]] ]] ],
     {licenseswapsonewayPreRobust[[l, 1]]} ], {licenseswapsonewayPreRobust[[l, 2]]} ] ,
   {l, licenseswapsonewaynumPreRobust}];
```

Winners for swaps

Who was the winner of the original package that license 2 is being added to?

(L2) In[79]:=
```
packages1swapwinnerpackPreRobust =
  ToPackedArray[ Table[ enclosingpackage[[btatouselookupbta[[
      licenseswapsonewayPreRobust[[l, 1]] ]] ]] , {l, licenseswapsonewaynumPreRobust}] ];
```

New for the robustness check: pop of package must be under bidder initial eligibility

Key inputs for the calculations

(L2) In[80]:=
```
eligibilitydatareal = N[eligibilitydata] / Total[populationdata[[btastouse]] // N] ;
populationdatareal = N[populationdata] / Total[ populationdata[[btastouse]] ];
```

Population of the new packages

(L2) In[82]:=
```
pack1waypopPreRobust =
  ToPackedArray[ Table [ Total[ populationdatareal[[packagesswapsonewayPreRobust[[l]] ]] ],
    {l, licenseswapsonewaynumPreRobust}] ];
```

- **Compare to eligibility**

(L2) In[83]:=
```
popLessThanEligPreRobust = Table[
   pack1waypopPreRobust[[l]] ≤ eligibilitydatareal[[ packages1swapwinnerpackPreRobust[[ l]] ]],
   {l, licenseswapsonewaynumPreRobust}] ;
Tally[popLessThanEligPreRobust]
```

(L2) Out[84]=
```
{{True, 73 409}, {False, 148 975}}
```

- **Redo key variables**

(L2) In[85]:=
```
licenseswapsoneway =
 ToPackedArray[Pick[ licenseswapsonewayPreRobust, popLessThanEligPreRobust] ];
licenseswapsonewaynum = Length[licenseswapsoneway]
packagesswapsoneway =
  ToPackedArray[Pick[packagesswapsonewayPreRobust, popLessThanEligPreRobust] ];
packages1swapwinnerpack = ToPackedArray[
   Pick[packages1swapwinnerpackPreRobust, popLessThanEligPreRobust] ];
```

(L2) Out[85]=
```
73 409
```

License adds and subtracts (no robustness yet)

- **Combinations of one license and one winning package**

From different winners

(L2) In[88]:=
```
combinationspackageslicenses = Tuples[ {Range[winningpackagesusablenum], btastouse}];
combinationspackageslicensesdiffpack = Select[ combinationspackageslicenses,
   #[[1]] ≠ enclosingpackage[[ btatouselookupbta[[#[[2]] ]] ]] & ];
combinationspackageslicensesdiffpacknum = Length[combinationspackageslicensesdiffpack]
```

(L2) Out[90]=
```
40 320
```

- **Resulting packages, for adds and subtracts**

(L2) In[91]:=
```
packagesadds =
  Table[ Union[ winningpackagesusable[[combinationspackageslicensesdiffpack[[l, 1]] ]],
    {combinationspackageslicensesdiffpack[[l, 2]]} ],
   {l, combinationspackageslicensesdiffpacknum}];
packagesaddsnum = combinationspackageslicensesdiffpacknum;
packagessubtracts =
  Table [ Complement[winningpackagesusable[[enclosingpackage[[b]] ]], {btastouse[[b]]} ],
   {b, btastousenum}];
```

- **Winning packages link for comparison**

(L2) In[94]:=
```
winnerpack1add = combinationspackageslicensesdiffpack[[All, 1]];
winnerpack1sub = enclosingpackage;
```

Pop, price, pop*eligibility & density*pop

Population

(L2) In[96]:=
```
pack1waypop =
  ToPackedArray[ Table [ Total[ populationdatareal[[packagesswapsoneway[[l]] ]] ],
    {l, licenseswapsonewaynum}] ];
packwinpop  =  Table [ Total[ populationdatareal[[winningpackagesusable[[l]] ]] ],
   {l, winningpackagesusablenum}];
pack1diffpop = packwinpop[[packages1swapwinnerpack]] - pack1waypop;

pack1addpop = ToPackedArray[
   Table [ Total[ populationdatareal[[packagesadds[[l]] ]] ], {l, packagesaddsnum}] ];
pack1subpop  = ToPackedArray[ Table [
    Total[ populationdatareal[[packagessubtracts[[l]] ]] ], {l, btastousenum}] ];

pack1adddiffpop = packwinpop[[winnerpack1add]] - pack1addpop;
pack1subdiffpop = packwinpop[[winnerpack1sub]] - pack1subpop;
```

Price

(L2) In[103]:=
```
pricedatareal = N[pricedata];
pack1wayprice = ToPackedArray[ Table [
    Total[ pricedatareal[[packagesswapsoneway[[l]] ]] ], {l, licenseswapsonewaynum}] ];
packwinprice = Table [ Total[ pricedatareal[[winningpackagesusable[[l]] ]] ],
   {l, winningpackagesusablenum}];
pack1diffprice = packwinprice[[packages1swapwinnerpack]] - pack1wayprice;

pack1addprice = ToPackedArray[
   Table [ Total[ pricedatareal[[packagesadds[[l]] ]] ], {l, packagesaddsnum}] ];
pack1subprice = ToPackedArray[ Table [ Total[ pricedatareal[[packagessubtracts[[l]] ]] ],
    {l, btastousenum}] ];

pack1adddiffprice = packwinprice[[winnerpack1add]] - pack1addprice;
pack1subdiffprice = packwinprice[[winnerpack1sub]] - pack1subprice;
```

Pop*eligibility

Note, eligibility applies to all of US, not just continental US

(L2) In[111]:=
```
pack1wayelig = ToPackedArray[
   Table[eligibilitydatareal[[uniquewinnersusable[[packages1swapwinnerpack[[l]]]]],
    {l, licenseswapsonewaynum}]];
pack1waypoptimeselig = pack1waypop pack1wayelig;

packwinelig = ToPackedArray[Table[
    eligibilitydatareal[[uniquewinnersusable[[enclosingpackage[[btatouselookupbta[[
        winningpackagesusable[[l, 1]]]]]]]], {l, winningpackagesusablenum}]];
packwinpoptimeselig = packwinpop packwinelig;

pack1diffpoptimeselig =
  packwinpoptimeselig[[packages1swapwinnerpack]] - pack1waypoptimeselig;


pack1addelig = ToPackedArray[Table[eligibilitydatareal[[uniquewinnersusable[[
      combinationspackageslicensesdiffpack[[l, 1]]]]]], {l, packagesaddsnum}]];
pack1addpoptimeselig = pack1addpop pack1addelig;

pack1subelig = ToPackedArray[
   Table[eligibilitydatareal[[marketdata[[btastouse[[l]], 5]]]], {l, btastousenum}]];
pack1subpoptimeselig = pack1subpop pack1subelig;

pack1adddiffpoptimeselig = packwinpoptimeselig[[winnerpack1add]] - pack1addpoptimeselig;
pack1subdiffpoptimeselig = packwinpoptimeselig[[winnerpack1sub]] - pack1subpoptimeselig;
```

- **Sum (density license * pop license)**

(L2) In[122]:=
```
densityreal = N[densitydata];
pack1waydensitypop =
  ToPackedArray[Table[Total[populationdatareal[[packagesswapsoneway[[l]]]]
      densityreal[[packagesswapsoneway[[l]]]] ], {l, licenseswapsonewaynum}]];

packwindensitypop =
  ToPackedArray[Table[Total[populationdatareal[[winningpackagesusable[[l]]]]
      densityreal[[winningpackagesusable[[l]]]] ], {l, winningpackagesusablenum}]];

pack1diffdensitypop = packwindensitypop[[packages1swapwinnerpack]] - pack1waydensitypop;

pack1adddensitypop = ToPackedArray[Table[
    Total[populationdatareal[[packagesadds[[l]]]] densityreal[[packagesadds[[l]]]] ],
    {l, packagesaddsnum}]];
pack1subdensitypop = ToPackedArray[Table[
    Total[populationdatareal[[packagessubtracts[[l]]]]
      densityreal[[packagessubtracts[[l]]]] ], {l, btastousenum}]];

pack1adddiffdensitypop = packwindensitypop[[winnerpack1add]] - pack1adddensitypop;
pack1subdiffdensitypop = packwindensitypop[[winnerpack1sub]] - pack1subdensitypop;
```

Geographic synergies

- **Gravity equation and geographic distance**

Do not want to divide by zero, so set 0 distances to 10^10 to effectively zero out
Also, make population in 10,000's so that the numbers will come out nicely

Scale of distance function

Divide by maximum synergies for the licenses in a package
We take the power of distances to focus only on local comparisons
Package 80 is GWI/MetroPCS, it is only in Miami, Atlanta, San Francisco

(L2) In[130]:=
```
geodistpower = 2
btadistancesdiomilestenforown =
  (btadistancesdiomiles + DiagonalMatrix[ Diagonal[btadistancesdiomiles] + 10.^10])^
   geodistpower;
```

(L2) Out[130]=
```
2
```

(L2) In[132]:=

■ Gravity equations

(L2) In[133]:=

```
populationdatarealtimes100 = populationdatareal100;

pack1waygeodist =
  ToPackedArray[Table[ Total[populationdatareal[[packagesswapsoneway[[l]]]]
      Total[Transpose[KroneckerProduct[populationdatarealtimes100[[
            packagesswapsoneway[[l]]]], populationdatarealtimes100[[
            packagesswapsoneway[[l]]]] ] / btadistancesdiomilestenforown[[
          packagesswapsoneway[[l]], packagesswapsoneway[[l]]]] ] ] /
       Total[Transpose[KroneckerProduct[populationdatarealtimes100[[
            packagesswapsoneway[[l]]]], populationdatarealtimes100[[btastouse ]]] ] /
         btadistancesdiomilestenforown[[packagesswapsoneway[[l]], btastouse ]] ] ] /
    Total[populationdatareal[[packagesswapsoneway[[l]]]] , {l,
    licenseswapsonewaynum}] ];


packwingeodist = ToPackedArray[
   Table[ Total[populationdatareal[[winningpackagesusable[[l]]]] Total[Transpose[
          KroneckerProduct[populationdatarealtimes100[[winningpackagesusable[[l]]]],
            populationdatarealtimes100[[winningpackagesusable[[l]]]] ] /
          btadistancesdiomilestenforown[[winningpackagesusable[[l]],
            winningpackagesusable[[l]]]] ] / Total[Transpose[
          KroneckerProduct[populationdatarealtimes100[[winningpackagesusable[[l]]]],
            populationdatarealtimes100[[btastouse ]]] / btadistancesdiomilestenforown[[
            winningpackagesusable[[l]], btastouse ]] ] ] /
    Total[populationdatareal[[winningpackagesusable[[l]]]] ] ,
   {l, winningpackagesusablenum}] ];


pack1diffgeodist = packwingeodist[[packages1swapwinnerpack]] - pack1waygeodist;



pack1addgeodist =
  ToPackedArray[Table[ Total[populationdatareal[[packagesadds[[l]]]] Total[
         Transpose[KroneckerProduct[populationdatarealtimes100[[packagesadds[[l]]]],
            populationdatarealtimes100[[packagesadds[[l]]]] ] /
          btadistancesdiomilestenforown[[packagesadds[[l]], packagesadds[[l]]]] ] ] /
       Total[Transpose[KroneckerProduct[populationdatarealtimes100[[
            packagesadds[[l]]]], populationdatarealtimes100[[btastouse ]]] ] /
         btadistancesdiomilestenforown[[packagesadds[[l]], btastouse ]] ] ] /
    Total[populationdatareal[[packagesadds[[l]]]] ] , {l, packagesaddsnum}] ];


(* note quiet shuts down division by 0 errors from empty sets,
remember to check here *)
```

```
pack1subgeodistpre =
  Quiet[ToPackedArray[
    Table [ Total[ populationdatareal[[packagessubtracts[[l]] ]] Total[ Transpose[
            KroneckerProduct[ populationdatarealtimes100[[packagessubtracts[[l]] ]],
              populationdatarealtimes100[[packagessubtracts[[l]] ]] ] /
             btadistancesdiomilestenforown[[ packagessubtracts[[l]],
              packagessubtracts[[l]] ]] ]  ]  /
          Total[ Transpose[ KroneckerProduct[ populationdatarealtimes100[[
               packagessubtracts[[l]] ]], populationdatarealtimes100[[btastouse ]] ] /
             btadistancesdiomilestenforown[[ packagessubtracts[[l]], btastouse ]] ]  ]  ] /
      Total[populationdatareal[[packagessubtracts[[l]] ]] ] , {l,
      btastousenum}] ] ];

pack1subgeodist = pack1subgeodistpre /. Indeterminate → 0.0;


pack1adddiffgeodist = packwingeodist[[winnerpack1add]] - pack1addgeodist;
pack1subdiffgeodist = packwingeodist[[winnerpack1sub]] - pack1subgeodist;
```

- **Airline travel synergies**

Not fixed to not decrease when adding licenses

(L2) In[142]:=
```
airlinetravelmatrixreal = ToPackedArray[ N[airlinetravelmatrix] + 0.00000001 ];


pack1wayairtrav =
  ToPackedArray[Monitor[ Table [Total[ populationdatareal[[packagesswapsoneway[[l]] ]]
        Total[Transpose[airlinetravelmatrixreal[[packagesswapsoneway[[l]],
            packagesswapsoneway[[l]] ]] ] ] / Total[Transpose[
           airlinetravelmatrixreal[[packagesswapsoneway[[l]], btastouse ]] ] ] ] /
      Total[populationdatareal[[packagesswapsoneway[[l]] ]] ] ,
     {l, licenseswapsonewaynum}]  , l] ] ;

packwinairtrav =
  ToPackedArray[Table [Total[ populationdatareal[[winningpackagesusable[[l]] ]]
       Total[Transpose[airlinetravelmatrixreal[[winningpackagesusable[[l]],
           winningpackagesusable[[l]] ]] ] ] / Total[Transpose[
          airlinetravelmatrixreal[[winningpackagesusable[[l]], btastouse ]] ] ] ] /
     Total[populationdatareal[[winningpackagesusable[[l]] ]] ] ,
    {l, winningpackagesusablenum}]  ] ;

pack1diffairtrav = packwinairtrav[[packages1swapwinnerpack]] - pack1wayairtrav;


pack1addairtrav =
  ToPackedArray[Table [Total[ populationdatareal[[packagesadds[[l]] ]]  Total[
         Transpose[airlinetravelmatrixreal[[packagesadds[[l]], packagesadds[[l]] ]] ] ] /
        Total[Transpose[airlinetravelmatrixreal[[packagesadds[[l]], btastouse ]] ] ] ] /
     Total[populationdatareal[[packagesadds[[l]] ]] ] ,   {l, packagesaddsnum}]  ] ;



(* quiet shuts off errors about transposes for empty packages*)

pack1subairtravpre =
  Quiet[ToPackedArray[Table [Total[ populationdatareal[[packagessubtracts[[l]] ]]
        Total[Transpose[airlinetravelmatrixreal[[
            packagessubtracts[[l]], packagessubtracts[[l]] ]] ] ] / Total[
          Transpose[airlinetravelmatrixreal[[packagessubtracts[[l]], btastouse ]] ] ] ] /
      Total[populationdatareal[[packagessubtracts[[l]] ]] ] ,   {l, btastousenum}]  ] ];


pack1subairtrav = pack1subairtravpre /. Indeterminate → 0.0;

pack1adddiffairtrav = packwinairtrav[[winnerpack1add]] - pack1addairtrav;
pack1subdiffairtrav = packwinairtrav[[winnerpack1sub]] - pack1subairtrav;
```

- **ATS travel survey synergies**

Not fixed to not decrease when adding licenses

(L2) In[151]:=
```
tripsmatrixreal = ToPackedArray[ N[tripsmatrix] + 0.00000001 ];


pack1wayatstrav = ToPackedArray[
   Monitor[ Table [Total[ populationdatareal[[packagesswapsoneway[[l]] ]]  Total[Transpose[
            tripsmatrixreal[[packagesswapsoneway[[l]], packagesswapsoneway[[l]] ]] ] ] /
         Total[Transpose[tripsmatrixreal[[packagesswapsoneway[[l]], btastouse ]] ] ] ] /
      Total[populationdatareal[[packagesswapsoneway[[l]] ]] ] ,
     {l, licenseswapsonewaynum}]  , l] ] ;


packwinatstrav =
  ToPackedArray[Table [Total[ populationdatareal[[winningpackagesusable[[l]] ]]
       Total[Transpose[tripsmatrixreal[[winningpackagesusable[[l]],
           winningpackagesusable[[l]] ]] ] ] /
        Total[Transpose[tripsmatrixreal[[winningpackagesusable[[l]], btastouse ]] ] ] ] /
     Total[populationdatareal[[winningpackagesusable[[l]] ]] ] ,
    {l, winningpackagesusablenum}]  ] ;




pack1diffatstrav = packwinatstrav[[packages1swapwinnerpack]] - pack1wayatstrav;

pack1addatstrav = ToPackedArray[Table [Total[ populationdatareal[[packagesadds[[l]] ]]
       Total[Transpose[tripsmatrixreal[[packagesadds[[l]], packagesadds[[l]] ]] ] ] /
        Total[Transpose[tripsmatrixreal[[packagesadds[[l]], btastouse ]] ] ] ] /
     Total[populationdatareal[[packagesadds[[l]] ]] ] ,   {l, packagesaddsnum}]  ] ;



pack1subatstravpre = Quiet[ToPackedArray[
    Table [Total[ populationdatareal[[packagessubtracts[[l]] ]]  Total[Transpose[
            tripsmatrixreal[[packagessubtracts[[l]], packagessubtracts[[l]] ]] ] ] /
         Total[Transpose[tripsmatrixreal[[packagessubtracts[[l]], btastouse ]] ] ] ] /
      Total[populationdatareal[[packagessubtracts[[l]] ]] ] ,   {l, btastousenum}]  ] ];

pack1subatstrav = pack1subatstravpre /. Indeterminate → 0.0;

pack1adddiffatstrav = packwinatstrav[[winnerpack1add]] - pack1addatstrav;
pack1subdiffatstrav = packwinatstrav[[winnerpack1sub]] - pack1subatstrav;
```

■ Eligibility * geographic distance

(L2) In[160]:=
```
packwingeodistelig = packwingeodist packwinelig;

pack1diffgeodistelig = ToPackedArray[ pack1diffgeodist pack1wayelig ];

packaddgeodistelig = pack1addgeodist pack1addelig;
packsubgeodistelig = pack1subgeodist pack1subelig;

pack1adddiffgeodistelig = packwingeodistelig[[winnerpack1add]] - packaddgeodistelig;
pack1subdiffgeodistelig = packwingeodistelig[[winnerpack1sub]] - packsubgeodistelig;
```

Pop * complementarities

Pop * geographic distance

(L2) In[166]:=
```
packwingeodistpop = packwingeodist packwinpop;

pack1diffgeodistpop = ToPackedArray[ pack1diffgeodist pack1waypop ];

packaddgeodistpop = pack1addgeodist pack1addpop;
packsubgeodistpop = pack1subgeodist pack1subpop;

pack1adddiffgeodistpop = packwingeodistpop[[winnerpack1add]] - packaddgeodistpop;
pack1subdiffgeodistpop = packwingeodistpop[[winnerpack1sub]] - packsubgeodistpop;
```

Pop * air travel

(L2) In[172]:=
```
packwinairtravpop = packwinairtrav packwinpop;

pack1diffairtravpop = ToPackedArray[ pack1diffairtrav pack1waypop ];

packaddairtravpop = pack1addairtrav pack1addpop;
packsubairtravpop = pack1subairtrav pack1subpop;

pack1adddiffairtravpop = packwinairtravpop[[winnerpack1add]] - packaddairtravpop;
pack1subdiffairtravpop = packwinairtravpop[[winnerpack1sub]] - packsubairtravpop;
```

Pop * ATS

(L2) In[178]:=
```
packwinatstravpop = packwinatstrav packwinpop;

pack1diffatstravpop = ToPackedArray[ pack1diffatstrav pack1waypop ];

packaddatstravpop = pack1addatstrav pack1addpop;
packsubatstravpop = pack1subatstrav pack1subpop;

pack1adddiffatstravpop = packwinatstravpop[[winnerpack1add]] - packaddatstravpop;
pack1subdiffatstravpop = packwinatstravpop[[winnerpack1sub]] - packsubatstravpop;
```

Put variables into toolkit data array

1 way swaps

(L2) In[184]:=
```
dataarray1way = ToPackedArray[{pack1diffprice, pack1diffpop, pack1diffpoptimeselig,
    pack1diffdensitypop, pack1diffgeodist, pack1diffgeodistelig, pack1diffairtrav,
    pack1diffatstrav, pack1diffgeodistpop, pack1diffairtravpop, pack1diffatstravpop}];
ByteCount[dataarray1way] / 1024.^2
```

(L2) Out[185]=
```
6.16085
```

- **1 license adds**

(L2) In[186]:=
```
dataarray1add =
  ToPackedArray[{pack1adddiffprice, pack1adddiffpop, pack1adddiffpoptimeselig,
    pack1adddiffdensitypop, pack1adddiffgeodist, pack1adddiffgeodistelig,
    pack1adddiffairtrav, pack1adddiffatstrav, pack1adddiffgeodistpop,
    pack1adddiffairtravpop, pack1adddiffatstravpop}];
ByteCount[dataarray1add] / 1024.^2
```

(L2) Out[187]=
```
3.38391
```

- **1 license substractions**

(L2) In[188]:=
```
dataarray1sub =
  ToPackedArray[{pack1subdiffprice, pack1subdiffpop, pack1subdiffpoptimeselig,
    pack1subdiffdensitypop, pack1subdiffgeodist, pack1subdiffgeodistelig,
    pack1subdiffairtrav, pack1subdiffatstrav, pack1subdiffgeodistpop,
    pack1subdiffairtravpop, pack1subdiffatstravpop}];
ByteCount[dataarray1sub] / 1024.^2
```

(L2) Out[189]=
```
0.0404053
```

- **Toolkit map format**

(L2) In[190]:=
```
datamap1way = ToPackedArray[
   Table[ {enclosingpackage[[btatouselookupbta[[ licenseswapsoneway[[l, 1]] ]] ]],
     enclosingpackage[[btatouselookupbta[[ licenseswapsoneway[[l, 2]] ]] ]]} ,
    {l, licenseswapsonewaynum}] ];
```

(L2) In[191]:=
```
datamap1add  = ToPackedArray[ Table[
    {combinationspackageslicensesdiffpack[[l, 1]], enclosingpackage[[btatouselookupbta[[
       combinationspackageslicensesdiffpack[[l, 2]] ]] ]]} , {l, packagesaddsnum}] ];

datamap1sub = ToPackedArray[
   Table[ {enclosingpackage[[l ]], enclosingpackage[[l ]]} , {l, btastousenum}] ];
```

Combine all into one

(L2) In[193]:=
```
dataarray1all = Join[ dataarray1way, dataarray1add, dataarray1sub, 2];
datamap1all = Join[ datamap1way, datamap1add, datamap1sub];
ByteCount[dataarray1all] / 1024.^2
```

(L2) Out[195]=
```
9.58492
```

Export

(L2) In[196]:=
```
SetDirectory[NotebookDirectory[] ];
SetDirectory["DataEst"];
Export["dataarray1wayadd.csv", dataarray1all];
Save["dataarray1wayadd.math", dataarray1all];

Export["datamap1wayadd.csv", datamap1all];
Save["datamap1wayadd.math", datamap1all];
```

■ Mean of variables

(L2) In[202]:=
```
Mean[Transpose[dataarray1all] ]
StandardDeviation[Transpose[dataarray1all] ]
```

(L2) Out[202]=
```
{-11.9982, -0.00110618, -0.000036207, -2.04555, 0.0142385, 0.000460133,
 -0.0000524807, 0.0009238, 0.0000903197, -0.0000113037, -0.000011071}
```

(L2) Out[203]=
```
{51.1588, 0.00392584, 0.00044981, 18.5099, 0.0235929, 0.00167346,
 0.00379413, 0.00998141, 0.000481924, 0.000393824, 0.000491012}
```

■ Swaps

(L2) In[204]:=
```
Mean[Transpose[dataarray1way] ]
StandardDeviation[Transpose[dataarray1way] ]
```

(L2) Out[204]=
```
{-7.50394, -0.00059033, -0.000016974, -1.07839, 0.0182793, 0.000532964,
 0.000247533, 0.00178206, 0.000149769, -6.41669 × 10^-7, 0.000016208}
```

(L2) Out[205]=
```
{35.4209, 0.00272618, 0.00020951, 9.81269, 0.0258036, 0.00124011,
 0.00361362, 0.00932549, 0.000364858, 0.0000630112, 0.000161651}
```

■ Adds, subs

Difference between adds and subs arises from weighting all licenses equally (subs) or weighting all packages equally (adds)

(L2) In[206]:=
```
Mean[Transpose[dataarray1add] ]
StandardDeviation[Transpose[dataarray1add] ]
```

(L2) Out[206]=
```
{-20.5685, -0.00208333, -0.0000814296, -3.87693, 0.00687917, 0.000305282,
 -0.000632615, -0.000704632, -0.0000332225, -0.0000406267, -0.0000737074}
```

(L2) Out[207]=
```
{70.3386, 0.00531308, 0.000581435, 27.9361, 0.0164875,
 0.00207612, 0.00395391, 0.0107026, 0.000437504, 0.000521276, 0.00065305}
```

(L2) In[208]:=
```
Mean[Transpose[dataarray1sub] ]
StandardDeviation[Transpose[dataarray1sub] ]
```

(L2) Out[208]=
```
{20.5685, 0.00208333, 0.000821077, 3.87693, 0.0144353,
 0.00232908, 0.00279607, 0.00645397, 0.00137598, 0.000821232, 0.00107845}
```

(L2) Out[209]=
```
{70.4111, 0.00531856, 0.00347746, 27.9649, 0.0270855,
 0.00793054, 0.00805728, 0.0204561, 0.00392173, 0.00356578, 0.00399345}
```

■ Winning packages

(L2) In[210]:=
```
packindataarrayall = {packwinprice, packwinpop, packwinpoptimeselig,
    packwindensitypop, packwingeodist, packwingeodistelig, packwinairtrav,
    packwinatstrav, packwingeodistpop, packwinairtravpop, packwinatstravpop};
```

(L2) In[211]:=
```
Mean[ Transpose[ packindataarrayall ] ]
StandardDeviation[ Transpose[ packindataarrayall] ]
Map[Max, packindataarrayall ]
Map[Min, packindataarrayall ]
```

(L2) Out[211]=
```
{116.152, 0.0117647, 0.00463667, 21.8933, 0.0677183,
 0.0118288, 0.0147153, 0.0318486, 0.00388512, 0.00232106, 0.00315061}
```

(L2) Out[212]=
```
{496.128, 0.0442504, 0.0299441, 124.931, 0.111326,
 0.0476612, 0.0511259, 0.0778808, 0.020581, 0.0165767, 0.0201134}
```

(L2) Out[213]=
```
{4200.77, 0.379645, 0.270545, 1111.52, 0.478555,
 0.341031, 0.395683, 0.48049, 0.181681, 0.150219, 0.182415}
```

(L2) Out[214]=
$$\{0.102375, 0.000111362, 1.24014\times10^{-8}, 0.00259376, 1.18107\times10^{-19}, 1.21002\times10^{-22},$$
$$3.37919\times10^{-15}, 1.37552\times10^{-11}, 4.57553\times10^{-23}, 2.39575\times10^{-17}, 9.75203\times10^{-14}\}$$

Data for matches only estimator

- **No need to recreate data, use indices into data for prices**

(L2) In[215]:=
```
licenseswapstwowayPreRobust =
  ToPackedArray[ Select[ licenseswapsoneway, #[[1]] < #[[2]] &] ];
licenseswapstwowaynumPreRobust = Length[licenseswapstwowayPreRobust];
```

- **From 2 way list, find 1 way records that correspond**

Some may be 0 entries in licenseswapstwowaydatapos because of eligibility robustness

(L2) In[217]:=
```
licenseswapstwowaydataposmatPreRobust = ToPackedArray[Table[0 , {licenses}, {licenses}] ];
Do[
 licenseswapstwowaydataposmatPreRobust[[
    licenseswapsoneway[[l, 1]], licenseswapsoneway[[l, 2]] ]] = l;
 , {l, licenseswapsonewaynum}]
licenseswapstwowaydataposPreRobust =
  Table[ {licenseswapstwowaydataposmatPreRobust[[ licenseswapstwowayPreRobust[[l, 1]],
     licenseswapstwowayPreRobust[[l, 2]] ]], licenseswapstwowaydataposmatPreRobust[[
     licenseswapstwowayPreRobust[[l, 2]], licenseswapstwowayPreRobust[[l, 1]] ]] } ,
   {l, licenseswapstwowaynumPreRobust}];
```

■ **Delete the 0 entries because of eligibility robustness**

(L2) In[220]:=
```
isAnyOneWayMissingPreRobust = Table[ licenseswapstwowaydataposPreRobust⟦l, 1⟧ ≠ 0 &&
     licenseswapstwowaydataposPreRobust⟦l, 2⟧ ≠ 0, {l, licenseswapstwowaynumPreRobust}];
Tally[isAnyOneWayMissingPreRobust]

licenseswapstwoway = Pick[licenseswapstwowayPreRobust, isAnyOneWayMissingPreRobust];
licenseswapstwowaynum = Length[licenseswapstwoway]
licenseswapstwowaydatapos =
  Pick[licenseswapstwowaydataposPreRobust, isAnyOneWayMissingPreRobust];
```

(L2) Out[221]=
```
{{True, 13 428}, {False, 21 531}}
```

(L2) Out[223]=
```
13 428
```

■ **For additions, find corresponding subtraction**

(L2) In[225]:=
```
addstosubstracts =
  Table[ btatouselookupbta[[combinationspackageslicensesdiffpack[[l, 2]] ]] ,
   {l, combinationspackageslicensesdiffpacknum}];
```

■ **Create data from data for 1 way swap inequalities**

(L2) In[226]:=
```
pack2diffprice = pack1diffprice[[ licenseswapstwowaydatapos[[All, 1]] ]] +
   pack1diffprice[[ licenseswapstwowaydatapos[[All, 2]] ]];
pack2diffpop = pack1diffpop[[ licenseswapstwowaydatapos[[All, 1]] ]] +
   pack1diffpop[[ licenseswapstwowaydatapos[[All, 2]] ]];
pack2diffpoptimeselig = pack1diffpoptimeselig[[ licenseswapstwowaydatapos[[All, 1]] ]] +
   pack1diffpoptimeselig[[ licenseswapstwowaydatapos[[All, 2]] ]];
pack2diffdensitypop = pack1diffdensitypop[[ licenseswapstwowaydatapos[[All, 1]] ]] +
   pack1diffdensitypop[[ licenseswapstwowaydatapos[[All, 2]] ]];
pack2diffgeodist = pack1diffgeodist[[ licenseswapstwowaydatapos[[All, 1]] ]] +
   pack1diffgeodist[[ licenseswapstwowaydatapos[[All, 2]] ]];
pack2diffgeodistelig = pack1diffgeodistelig[[ licenseswapstwowaydatapos[[All, 1]] ]] +
   pack1diffgeodistelig[[ licenseswapstwowaydatapos[[All, 2]] ]];
pack2diffairtrav = pack1diffairtrav[[ licenseswapstwowaydatapos[[All, 1]] ]] +
   pack1diffairtrav[[ licenseswapstwowaydatapos[[All, 2]] ]];
pack2diffatstrav = pack1diffatstrav[[ licenseswapstwowaydatapos[[All, 1]] ]] +
   pack1diffatstrav[[ licenseswapstwowaydatapos[[All, 2]] ]];
pack2diffgeodistpop = pack1diffgeodistpop[[ licenseswapstwowaydatapos[[All, 1]] ]] +
   pack1diffgeodistpop[[ licenseswapstwowaydatapos[[All, 2]] ]];
pack2diffairtravpop = pack1diffairtravpop[[ licenseswapstwowaydatapos[[All, 1]] ]] +
   pack1diffairtravpop[[ licenseswapstwowaydatapos[[All, 2]] ]];
pack2diffatstravpop = pack1diffatstravpop[[ licenseswapstwowaydatapos[[All, 1]] ]] +
   pack1diffatstravpop[[ licenseswapstwowaydatapos[[All, 2]] ]];
```

Data for adds and subtracts

(L2) In[237]:=
```
pack2adddiffprice = pack1adddiffprice + pack1subdiffprice[[addstosubstracts]];
pack2adddiffpop = pack1adddiffpop + pack1subdiffpop[[addstosubstracts]];
pack2adddiffpoptimeselig =
  pack1adddiffpoptimeselig + pack1subdiffpoptimeselig[[addstosubstracts]];
pack2adddiffdensitypop = pack1adddiffdensitypop +
   pack1subdiffdensitypop[[addstosubstracts]];
pack2adddiffgeodist = pack1adddiffgeodist + pack1subdiffgeodist[[addstosubstracts]];
pack2adddiffgeodistelig =
  pack1adddiffgeodistelig + pack1subdiffgeodistelig[[addstosubstracts]];
pack2adddiffairtrav = pack1adddiffairtrav + pack1subdiffairtrav[[addstosubstracts]];
pack2adddiffatstrav = pack1adddiffatstrav + pack1subdiffatstrav[[addstosubstracts]];
pack2adddiffgeodistpop =
  pack1adddiffgeodistpop + pack1subdiffgeodistpop[[addstosubstracts]];
pack2adddiffairtravpop = pack1adddiffairtravpop +
   pack1subdiffairtravpop[[addstosubstracts]];
pack2adddiffatstravpop = pack1adddiffatstravpop +
   pack1subdiffatstravpop[[addstosubstracts]];
```

Put 2 way variables into toolkit data array

Swaps

(L2) In[248]:=
```
dataarray2way = ToPackedArray[{pack2diffprice, pack2diffpop, pack2diffpoptimeselig,
    pack2diffdensitypop, pack2diffgeodist, pack2diffgeodistelig, pack2diffairtrav,
    pack2diffatstrav, pack2diffgeodistpop, pack2diffairtravpop, pack2diffatstravpop}];
ByteCount[dataarray2way] / 1024.^2
```

(L2) Out[249]=
```
1.12704
```

Adds

(L2) In[250]:=
```
dataarray2add =
  ToPackedArray[{pack2adddiffprice, pack2adddiffpop, pack2adddiffpoptimeselig,
    pack2adddiffdensitypop, pack2adddiffgeodist, pack2adddiffgeodistelig,
    pack2adddiffairtrav, pack2adddiffatstrav, pack2adddiffgeodistpop,
    pack2adddiffairtravpop, pack2adddiffatstravpop}];
ByteCount[dataarray2add] / 1024.^2
```

(L2) Out[251]=
```
3.38391
```

Toolkit map format

Swaps

Data map in terms of winning packages

(L2) In[252]:=
```
datamap2way = ToPackedArray[
   Table[ {enclosingpackage[[btatouselookupbta[[licenseswapstwoway[[l, 1]] ]] ]],
     enclosingpackage[[btatouselookupbta[[licenseswapstwoway[[l, 2]] ]] ]]} ,
    {l, licenseswapstwowaynum}] ];
```

Terms of winning licenses

(L2) In[253]:=
```
datamap2waylic = ToPackedArray[licenseswapstwoway];
```

■ Adds

Winning packages

(L2) In[254]:=
```
datamap2add = datamap1add;
```

Winning licenses

(L2) In[255]:=
```
datamap2addlic = Transpose[{combinationspackageslicensesdiffpack[[All, 2]],
    combinationspackageslicensesdiffpack[[All, 2]]}];
```

■ Combine adds and swaps

(L2) In[256]:=
```
dataarray2all = Join[ dataarray2way, dataarray2add, 2];
datamap2all = Join[datamap2way, datamap2add];
```

■ Export

(L2) In[258]:=
```
SetDirectory[NotebookDirectory[] ];
SetDirectory["DataEst"];
Export["dataarray2wayadd.csv", dataarray2all];
Save["dataarray2wayadd.math", dataarray2all];

Export["datamap2wayadd.csv", datamap2all];
Save["datamap2wayadd.math", datamap2all];
```

■ Mean of variables

Keep in mind adds weight by package, more or less, while swaps weight by licenses
So adds overweight small packages compared to swaps

(L2) In[264]:=
```
Mean[Transpose[dataarray2way] ]
StandardDeviation[ Transpose[dataarray2way] ]
```

(L2) Out[264]=
$$\{-1.82167\times10^{-15},\ 5.72364\times10^{-19},\ 8.61364\times10^{-6},\ -1.52644\times10^{-17},\ 0.0313777,\ 0.000775067,\ 0.000310215,\ 0.0025149,\ 0.000190179,\ 9.03657\times10^{-7},\ 0.0000226012\}$$

(L2) Out[265]=
$$\{5.12596\times10^{-15},\ 1.29898\times10^{-18},\ 0.0000527769,\ 5.56963\times10^{-16},\ 0.0344062,\ 0.00132065,\ 0.00148725,\ 0.0109254,\ 0.000264713,\ 0.0000112302,\ 0.000095513\}$$

(L2) In[266]:=
```
Mean[Transpose[dataarray2add] ]
StandardDeviation[ Transpose[dataarray2add] ]
```

(L2) Out[266]=
$$\{3.97325\times10^{-14},\ 3.86828\times10^{-19},\ 0.000739647,\ -1.6815\times10^{-14},\ 0.0213145,\ 0.00263436,\ 0.00216345,\ 0.00574934,\ 0.00134276,\ 0.000780605,\ 0.00100475\}$$

(L2) Out[267]=
$$\{1.59101\times10^{-13},\ 8.95295\times10^{-18},\ 0.00333922,\ 6.30628\times10^{-14},\ 0.0330628,\ 0.00834356,\ 0.00873037,\ 0.0223327,\ 0.00389062,\ 0.00352698,\ 0.00364012\}$$

(L2) In[268]:=
```
Mean[Transpose[dataarray2all] ]
StandardDeviation[ Transpose[dataarray2all] ]
```

(L2) Out[268]=
$$\{2.93509 \times 10^{-14},\ 4.33181 \times 10^{-19},\ 0.000557011,\ -1.26179 \times 10^{-14},\ 0.0238286,\ 0.00216985,\ 0.00170045,\ 0.00494127,\ 0.00105481,\ 0.00058581,\ 0.000759374\}$$

(L2) Out[269]=
$$\{1.38994 \times 10^{-13},\ 7.78187 \times 10^{-18},\ 0.00290955,\ 5.51027 \times 10^{-14},\ 0.033686,\ 0.00730111,\ 0.00764023,\ 0.0201476,\ 0.00340906,\ 0.00307338,\ 0.00318168\}$$

- **Is weighting of adds by packages instead of licenses causing differences in means, especially for elig*pop?**

Answer, no. Found error, it was in elig*pop for swaps calculation
Affects standard deviation of synergy measures a lot

- **Number of licenses (combined) per inequality**

(L2) In[270]:=
```
addpacksizes = Table[
   Total[winningpackagesusablesizes[[datamap2add[[l]] ]] ], {l, Length[datamap2add] } ];
```

- **New means**

(L2) In[271]:=
```
weightmeansadd = Total[addpacksizes Transpose[dataarray2add] ] / Total[addpacksizes]

Sqrt[
 Total[addpacksizes Transpose[(dataarray2add - weightmeansadd)^2] ] / Total[addpacksizes] ]
```

(L2) Out[271]=
$$\{1.13672 \times 10^{-13},\ 1.50883 \times 10^{-18},\ 0.00170667,\ -4.54736 \times 10^{-14},\ 0.0226727,\ 0.00420733,\ 0.00406261,\ 0.00692976,\ 0.00273345,\ 0.00192897,\ 0.00227244\}$$

(L2) Out[272]=
$$\{2.44254 \times 10^{-13},\ 1.4106 \times 10^{-17},\ 0.00522027,\ 9.8542 \times 10^{-14},\ 0.034069,\ 0.00954663,\ 0.0122433,\ 0.0230326,\ 0.0057885,\ 0.00557458,\ 0.00562167\}$$

End of data stuff

No prices

Gets rid of adding licenses reduces complementarities

- **Only geo*pop and elig*pop**

- **Objective function**

(L2) In[273]:=
```
obj2waygeosynergyeligpop[data_, bgeo_] :=
  Module[ {values, onesorzeros},
   values = data[[3]] + bgeo * data[[9]] + 0.000001;
   onesorzeros = values / Abs[values] + 1.0;
   Total[onesorzeros] / 2. / Length[data[[1]] ]
  ];
```

(L2) In[274]:=
```
obj2waygeosynergyeligpopneg[data_, bgeo_] :=
  Module[ {values, onesorzeros},
   values = -data[[3]] + bgeo * data[[9]] + 0.000001;
   onesorzeros = values / Abs[values] + 1.0;
   Total[onesorzeros] / 2. / Length[data[[1]] ]
  ];
```

- **Adds (no robust to elig yet)**

(L2) In[275]:=
```
obj2waygeosynergyeligpop[dataarray2add, 1.] // Timing
```

(L2) Out[275]=
```
{0.002766, 0.879936}
```

(L2) In[276]:=
```
Timing[ans = pairwiseMSE[obj2waygeosynergyeligpop, dataarray2add,
    { bgeodist}, nMaximizeOptions → {Method → {"DifferentialEvolution",
        "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]] } } ] ]
```

(L2) Out[276]=
```
{7.44194, {0.897073, {bgeodist → 9.35226}}}
```

(L2) In[277]:=
```
Do[
 Print[
  AbsoluteTiming[ans = pairwiseMSE[obj2waygeosynergyeligpop, dataarray2add, { bgeodist},
     nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 75,
         "RandomSeed" → Floor[SessionTime[]] } } ] ] ]
 ,
 {k,
  1,
  15}]
```

```
{12.514332, {0.897073, {bgeodist → 9.35026}}}

{12.383349, {0.897073, {bgeodist → 9.34744}}}

{14.186830, {0.897073, {bgeodist → 9.87563}}}

{8.221142, {0.897073, {bgeodist → 9.87188}}}

{13.720280, {0.897073, {bgeodist → 9.34535}}}

{11.569547, {0.897073, {bgeodist → 9.34352}}}

{11.902047, {0.897073, {bgeodist → 9.87617}}}

{12.444316, {0.897073, {bgeodist → 9.34662}}}

{9.211629, {0.897073, {bgeodist → 9.35103}}}

{13.167185, {0.897073, {bgeodist → 9.34851}}}

{9.499285, {0.897073, {bgeodist → 9.8748}}}

{13.230796, {0.897073, {bgeodist → 9.87769}}}

{14.710374, {0.897073, {bgeodist → 9.34456}}}

{9.519295, {0.897073, {bgeodist → 9.87301}}}

{11.412525, {0.897073, {bgeodist → 9.35036}}}
```

- **Negatives (no robust to elig yet)**

(L2) In[278]:=
```
Do[
 Print[AbsoluteTiming[
   ans = pairwiseMSE[obj2waygeosynergyeligpopneg, dataarray2add, {bgeodist},
     nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 75,
          "RandomSeed" → Floor[SessionTime[]] } } ] ] ]
 ,
 {k,
  1,
  15}]
```

```
{14.975304, {0.91131, {bgeodist → 3.80944}}}

{7.676906, {0.91131, {bgeodist → 3.80938}}}

{10.582344, {0.91131, {bgeodist → 3.80936}}}

{13.572473, {0.91131, {bgeodist → 3.80942}}}

{9.254714, {0.91131, {bgeodist → 3.80946}}}

{16.283009, {0.91131, {bgeodist → 3.80894}}}

{12.005050, {0.91131, {bgeodist → 3.80957}}}

{16.288016, {0.91131, {bgeodist → 3.80951}}}

{15.746709, {0.91131, {bgeodist → 3.80952}}}

{8.816011, {0.91131, {bgeodist → 3.80938}}}

{9.452897, {0.91131, {bgeodist → 3.80879}}}

{13.845980, {0.91131, {bgeodist → 3.80922}}}

{10.819950, {0.91131, {bgeodist → 3.8089}}}

{7.622191, {0.91131, {bgeodist → 3.80935}}}

{15.345055, {0.91131, {bgeodist → 3.80861}}}
```

(L2) In[279]:=
```
Do[
 Print[AbsoluteTiming[
   ans = pairwiseMSE[obj2waygeosynergyeligpopneg, dataarray2add, {bgeodist},
     nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 150,
          "RandomSeed" → Floor[SessionTime[]] } } ] ] ]
 ,
 {k,
  1,
  15}]
```

```
{32.145716, {0.91131, {bgeodist → 3.80945}}}

{27.638361, {0.91131, {bgeodist → 3.80943}}}

{30.681827, {0.91131, {bgeodist → 3.80921}}}

{27.891866, {0.91131, {bgeodist → 3.80929}}}

{33.041174, {0.91131, {bgeodist → 3.80922}}}

{27.046455, {0.91131, {bgeodist → 3.80917}}}

{27.732771, {0.91131, {bgeodist → 3.8091}}}

{26.943305, {0.91131, {bgeodist → 3.80887}}}

{29.815729, {0.91131, {bgeodist → 3.80921}}}

{28.582397, {0.91131, {bgeodist → 3.80946}}}

{18.788838, {0.91131, {bgeodist → 3.80919}}}

{28.565031, {0.91131, {bgeodist → 3.80885}}}

{31.262756, {0.91131, {bgeodist → 3.80958}}}

{26.800649, {0.91131, {bgeodist → 3.80949}}}

{30.433512, {0.91131, {bgeodist → 3.80871}}}
```

- **Standard errors for adds (no robust to elig yet)**

- **Swaps**

(L2) In[282]:=
```
obj2waygeosynergyeligpop[dataarray2way, 1.] // Timing
```

(L2) Out[282]=
```
{0.000701, 0.955615}
```

(L2) In[283]:=
```
Timing[ans = pairwiseMSE[obj2waygeosynergyeligpop, dataarray2way,
    {bgeodist}, nMaximizeOptions → {Method → {"DifferentialEvolution",
        "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]]}}]]
```

(L2) Out[283]=
```
{2.93304, {0.955913, {bgeodist → 1.09685}}}
```

(L2) In[284]:=
```
Do[
 Print[
  AbsoluteTiming[ans = pairwiseMSE[obj2waygeosynergyeligpop, dataarray2way, {bgeodist},
     nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 75,
         "RandomSeed" → Floor[SessionTime[]]}}]]]
 ,
 {k,
  1,
  15}]
```

```
{3.212954, {0.955913, {bgeodist → 1.09632}}}

{2.366604, {0.955913, {bgeodist → 1.0956}}}

{3.689563, {0.955913, {bgeodist → 1.09327}}}

{2.480694, {0.955913, {bgeodist → 1.05618}}}

{2.790384, {0.955913, {bgeodist → 1.09431}}}

{2.785870, {0.955913, {bgeodist → 1.09459}}}

{2.381107, {0.955913, {bgeodist → 1.09575}}}

{2.507791, {0.955913, {bgeodist → 1.09472}}}

{2.420787, {0.955913, {bgeodist → 1.09497}}}

{2.894350, {0.955913, {bgeodist → 1.09643}}}

{2.786608, {0.955913, {bgeodist → 1.09415}}}

{2.789246, {0.955913, {bgeodist → 1.09507}}}

{3.257576, {0.955913, {bgeodist → 1.09557}}}

{2.341859, {0.955913, {bgeodist → 1.09688}}}

{2.439927, {0.955913, {bgeodist → 1.09515}}}
```

- **Negative**

(L2) In[285]:=
```
Timing[ans = pairwiseMSE[obj2waygeosynergyeligpopneg, dataarray2way,
    {bgeodist}, nMaximizeOptions → {Method → {"DifferentialEvolution",
        "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]]}}]]
```

(L2) Out[285]=
```
{6.21497, {0.939976, {bgeodist → 30.3227}}}
```

(L2) In[286]:=
```
Timing[ans = pairwiseMSE[obj2waygeosynergyeligpopneg, dataarray2way,
    {bgeodist}, nMaximizeOptions → {Method → {"DifferentialEvolution",
        "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]]}}]]
```

(L2) Out[286]=
```
{3.7734, {0.939976, {bgeodist → 29.988}}}
```

- **Standard errors for swaps**

(L2) In[289]:=
```
estimateway4 = {bgeodist} /. {bgeodist → 1.0561848284265936`}
```

(L2) Out[289]=
```
{1.05618}
```

(L2) In[290]:=
```
Timing[pointcrout1 = pointIdentifiedCR[25, 200, estimateway4,
    obj2waygeosynergyeligpop, {bgeodist}, datamap2way, dataarray2way,
    asymptotics → coalitions, nMaximizeOptions → {Method → {"DifferentialEvolution",
        "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]]}}]]
```

(L2) Out[290]=
```
{431.824,
 {{{0.87303, 1.55774}}, {{-0.144794}, {0.0120099}, {-1.23212}, {0.205249}, {-0.141778},
   {1.05187}, {0.75213}, {0.194077}, {-1.21946}, {0.215212}, {-2.29251}, {-0.146136},
   {1.05947}, {0.199361}, {0.183997}, {0.805952}, {-3.61565}, {-3.37017}, {-0.0286137},
   {-2.29788}, {-2.29236}, {-0.862618}, {0.194771}, {-2.31332}, {0.874775}, {-3.34848},
   {-2.14326}, {-2.30135}, {-0.143206}, {-2.30416}, {-3.33939}, {0.184303}, {-1.21866},
   {-2.29968}, {0.200431}, {-3.29795}, {-0.145172}, {0.193969}, {0.19348}, {0.00524353},
   {1.28281}, {0.634335}, {0.190863}, {-0.145933}, {0.616472}, {1.09575}, {-0.144907},
   {-3.78664}, {1.45774}, {-0.954879}, {-0.141959}, {-1.16639}, {-2.07109}, {-2.31409},
   {0.201751}, {0.200898}, {0.18373}, {1.29069}, {0.195886}, {0.197929}, {0.206391},
   {-2.31434}, {-0.151629}, {-2.29862}, {0.198086}, {0.192808}, {0.197676}, {0.205814},
   {0.188445}, {-2.11667}, {-1.17471}, {-2.02884}, {0.200775}, {-3.34883}, {0.645065},
   {-0.141604}, {-3.42759}, {-1.096}, {1.39792}, {0.618353}, {-2.3072}, {0.197574},
   {-3.78951}, {-0.0139475}, {1.3885}, {0.625819}, {0.197791}, {0.189513}, {-3.8058},
   {-1.71017}, {0.00566357}, {0.197163}, {0.192311}, {0.211108}, {-3.2965}, {0.191315},
   {0.908139}, {1.37582}, {-2.31394}, {-3.80277}, {-3.84758}, {0.201044}, {-2.29093},
   {0.739827}, {-1.21758}, {-1.23983}, {1.06026}, {-2.3024}, {1.42188}, {0.194851},
   {-0.144558}, {-2.31411}, {0.157626}, {-2.29831}, {0.191206}, {-2.12648}, {0.20176},
   {-0.144767}, {0.200576}, {1.0531}, {-2.31379}, {-2.14687}, {0.00692537}, {1.07346},
   {-2.3146}, {0.219941}, {0.883126}, {0.89347}, {1.05353}, {-2.0022}, {-1.23476},
   {0.00579403}, {0.62244}, {-2.29455}, {-0.958694}, {0.203135}, {1.38279}, {0.211866},
   {0.200692}, {0.19222}, {-0.146717}, {-3.36991}, {-3.42947}, {0.205533}, {-2.31393},
   {0.190467}, {-0.144649}, {0.194643}, {-1.16438}, {-0.860035}, {0.604406}, {-2.29693},
   {-2.29783}, {-0.144886}, {1.39754}, {-0.211379}, {0.204452}, {-0.96381}, {0.194507},
   {-2.30152}, {0.00443801}, {0.191512}, {-2.29283}, {-2.29621}, {0.185171}, {0.19927},
   {0.629525}, {1.05142}, {-1.88356}, {-3.42764}, {1.06138}, {-2.29648}, {-3.91648},
   {0.170804}, {-3.34842}, {0.212231}, {-0.125894}, {-0.853646}, {0.196839}, {0.194527},
   {-2.28695}, {-2.30146}, {-0.145033}, {0.00493262}, {0.197086}, {-1.09038},
   {0.0119893}, {-2.27261}, {0.189161}, {-3.29344}, {0.207697}, {-2.29765}, {-3.29867},
   {-2.2947}, {-2.29807}, {0.19635}, {1.28355}, {0.0074516}, {0.753922}, {-2.29974}}}}
```

■ Add air and ATS travel

■ Objective function

(L2) In[291]:=
```
obj2wayallsynergyeligpop[data_, bgeo_, bair_, bats_] :=
  Module[{values, onesorzeros},
   values = data[[3]] + bgeo * data[[9]] + bair * data[[10]] + bats * data[[11]] + 0.000001;
   onesorzeros = values / Abs[values] + 1.0;
   Total[onesorzeros] / 2. / Length[data[[1]]]
  ];
```

(L2) In[292]:=
```
obj2wayallsynergyeligpopneg[data_, bgeo_, bair_, bats_] :=
  Module[{values, onesorzeros},
   values = -data[[3]] + bgeo * data[[9]] + bair * data[[10]] + bats * data[[11]] + 0.000001;
   onesorzeros = values / Abs[values] + 1.0;
   Total[onesorzeros] / 2. / Length[data[[1]]]
  ];
```

■ Adds (no robust)

(L2) In[293]:=
```
obj2wayallsynergyeligpop[dataarray2add, 1., 1., 1.] // Timing
```

(L2) Out[293]=
```
{0.00523, 0.845908}
```

(L2) In[294]:=
```
Timing[ans = pairwiseMSE[obj2wayallsynergyeligpop, dataarray2add,
    {bgeodist, bair, bats}, nMaximizeOptions → {Method → {"DifferentialEvolution",
        "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]]}}]]
```

(L2) Out[294]=
```
{25.419, {0.925992, {bgeodist → 49.3593, bair → -10.5536, bats → -6.76417}}}
```

(L2) In[295]:=
```
Do[Print[Timing[
    ans = pairwiseMSE[obj2wayallsynergyeligpop, dataarray2add, {bgeodist, bair, bats},
      nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 75,
          "RandomSeed" → Floor[SessionTime[]]}}]]]
 ,
 {k,
  1,
  15}]
```

```
{25.6287, {0.924306, {bgeodist → 23.4789, bair → -4.37997, bats → -3.72971}}}

{24.4955, {0.925992, {bgeodist → 45.8363, bair → -9.79007, bats → -6.34549}}}

{25.5736, {0.924529, {bgeodist → 28.0475, bair → -4.80461, bats → -4.43683}}}

{23.6906, {0.925719, {bgeodist → 44.5238, bair → -8.99763, bats → -6.02435}}}

{23.4831, {0.925967, {bgeodist → 43.2716, bair → -9.0609, bats → -6.27912}}}

{25.9809, {0.925223, {bgeodist → 33.4096, bair → -6.7184, bats → -5.07489}}}

{24.3282, {0.92567, {bgeodist → 39.4995, bair → -8.60175, bats → -5.39678}}}

{25.2, {0.925546, {bgeodist → 48.0106, bair → -11.3394, bats → -4.97364}}}

{23.1807, {0.925843, {bgeodist → 47.5411, bair → -10.754, bats → -5.7687}}}

{24.7834, {0.924926, {bgeodist → 28.3178, bair → -5.90425, bats → -3.99968}}}

{24.2442, {0.925521, {bgeodist → 44.1084, bair → -8.58523, bats → -6.04663}}}

{23.0973, {0.925719, {bgeodist → 39.1135, bair → -8.24857, bats → -5.62432}}}

{23.7865, {0.924901, {bgeodist → 26.2547, bair → -5.3824, bats → -3.99711}}}

{23.8314, {0.925099, {bgeodist → 32.1847, bair → -6.61734, bats → -4.4649}}}

{23.1506, {0.925918, {bgeodist → 46.7142, bair → -9.65633, bats → -6.43547}}}
```

(L2) In[296]:=
```
Do[Print[Timing[
    ans = pairwiseMSE[obj2wayallsynergyeligpop, dataarray2add, {bgeodist, bair, bats},
      nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 150,
          "RandomSeed" → Floor[SessionTime[]]}}]]]
 ,
 {k,
  1,
  15}]
```

(L2) Out[296]=
```
$Aborted
```

```
Do[Print[
  Timing[ans = pairwiseMSE[obj2wayallsynergyeligpop, dataarray2add, {bgeodist, bair, bats},
      nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 500,
          "RandomSeed" → Floor[SessionTime[]]}}]]]
 ,
 {k,
  1,
  15}]
```

```
{767.571, {0.939335, {bgeodist → 52.0147, bair → -1.84829, bats → -1.69336}}}

{767.92, {0.93936, {bgeodist → 65.2506, bair → -2.30631, bats → -1.83533}}}

{775.055, {0.939335, {bgeodist → 51.85, bair → -1.85408, bats → -1.7793}}}

{770.254, {0.93936, {bgeodist → 46.874, bair → -1.50386, bats → -1.89645}}}

{771.24, {0.939335, {bgeodist → 53.6229, bair → -1.85234, bats → -1.78913}}}

{769.552, {0.940352, {bgeodist → 156.778, bair → -1.77726, bats → -7.13051}}}

{775.767, {0.93936, {bgeodist → 52.7129, bair → -1.84868, bats → -1.7856}}}

{776.246, {0.939335, {bgeodist → 53.7818, bair → -1.84532, bats → -1.80969}}}

{768.314, {0.939335, {bgeodist → 52.4297, bair → -1.72354, bats → -1.8695}}}

{769.896, {0.939385, {bgeodist → 62.3649, bair → -2.26715, bats → -1.69213}}}

{769.225, {0.939385, {bgeodist → 62.2349, bair → -2.04807, bats → -1.98344}}}

{782.011, {0.939385, {bgeodist → 65.5078, bair → -2.30591, bats → -1.8389}}}

{775.351, {0.939335, {bgeodist → 52.9192, bair → -1.82081, bats → -1.788}}}

{770.982, {0.939385, {bgeodist → 46.5896, bair → -1.47831, bats → -1.91612}}}

{770.957, {0.939335, {bgeodist → 61.3547, bair → -1.96854, bats → -1.99322}}}
```

```
Do[Print[
  Timing[ans = pairwiseMSE[obj2wayallsynergyeligpop, dataarray2add, {bgeodist, bair, bats},
     nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 750,
         "RandomSeed" → Floor[SessionTime[]]}}]]]
 ,
 {k,
  1,
  15}]
```

```
{1160.35, {0.940352, {bgeodist → 136.635, bair → -1.75427, bats → -6.58845}}}

{1165.89, {0.93998, {bgeodist → 102.445, bair → -1.55897, bats → -4.99466}}}

{1163.15, {0.93941, {bgeodist → 52.921, bair → -1.78232, bats → -1.80783}}}

{1165.77, {0.93936, {bgeodist → 64.1681, bair → -2.15689, bats → -1.97393}}}

{1159.45, {0.940005, {bgeodist → 102.983, bair → -0.817116, bats → -5.91782}}}

{1163.53, {0.939385, {bgeodist → 52.3757, bair → -1.81326, bats → -1.73329}}}

{1167.7, {0.939881, {bgeodist → 104.327, bair → -1.41342, bats → -5.16575}}}

{1166.94, {0.939385, {bgeodist → 83.4991, bair → -1.05023, bats → -4.40946}}}

{1165.44, {0.93941, {bgeodist → 63.4142, bair → -2.11244, bats → -1.97934}}}

{1163.38, {0.939459, {bgeodist → 89.2711, bair → -1.38222, bats → -4.12386}}}

{1163.3, {0.939385, {bgeodist → 52.1973, bair → -1.82695, bats → -1.73349}}}

{1162.64, {0.940154, {bgeodist → 107.749, bair → -1.57812, bats → -5.04158}}}

{1157.7, {0.939683, {bgeodist → 96.877, bair → -0.767106, bats → -5.44054}}}

{1156.29, {0.93936, {bgeodist → 52.4839, bair → -1.78508, bats → -1.81587}}}

{1159.35, {0.940079, {bgeodist → 114.201, bair → -1.42771, bats → -5.29076}}}
```

Negative for adds (no robust)

```
Do[Print[Timing[
    ans = pairwiseMSE[obj2wayallsynergyeligpopneg, dataarray2add, {bgeodist, bair, bats},
      nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 75,
           "RandomSeed" → Floor[SessionTime[]]}}]]]
 ,
 {k,
  1,
  2}]
```

```
{131.353, {0.935169, {bgeodist → 55.2751, bair → -0.451361, bats → -1.21992}}}
```

```
{132.264, {0.935169, {bgeodist → 56.8364, bair → -0.679161, bats → -1.03795}}}
```

- **Standard errors for adds (no robust)**

- **Swaps**

(L2) In[297]:=
```
obj2wayallsynergyeligpop[dataarray2way, 1., 1., 1.] // Timing
```

(L2) Out[297]=
```
{0.001125, 0.949136}
```

(L2) In[298]:=
```
Do[Print[Timing[
    ans = pairwiseMSE[obj2wayallsynergyeligpop, dataarray2way, {bgeodist, bair, bats},
      nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 75,
           "RandomSeed" → Floor[SessionTime[]]}}]]]
 ,
 {k,
  25}]
```

```
{8.02957, {0.95986, {bgeodist → 0.862162, bair → -0.26526, bats → -0.616751}}}

{7.86071, {0.95986, {bgeodist → 0.869962, bair → -0.268225, bats → -0.631965}}}

{7.8524, {0.95986, {bgeodist → 0.869923, bair → -0.266428, bats → -0.633848}}}

{7.82699, {0.95986, {bgeodist → 0.870526, bair → -0.267884, bats → -0.630821}}}

{7.8038, {0.95986, {bgeodist → 0.870234, bair → -0.269451, bats → -0.635465}}}

{7.88608, {0.95986, {bgeodist → 0.869864, bair → -0.269324, bats → -0.633586}}}

{7.83434, {0.95986, {bgeodist → 0.86992, bair → -0.268033, bats → -0.631974}}}

{7.83938, {0.95986, {bgeodist → 0.87026, bair → -0.267091, bats → -0.629898}}}

{7.82799, {0.95986, {bgeodist → 0.87038, bair → -0.267845, bats → -0.631451}}}

{7.86623, {0.95986, {bgeodist → 0.870254, bair → -0.266656, bats → -0.630794}}}

{7.86692, {0.959786, {bgeodist → 0.870936, bair → -0.261125, bats → -0.662644}}}

{7.97466, {0.95986, {bgeodist → 0.871074, bair → -0.268918, bats → -0.629528}}}

{7.8618, {0.959786, {bgeodist → 0.868953, bair → -0.263308, bats → -0.630649}}}

{8.00514, {0.95986, {bgeodist → 0.870063, bair → -0.267474, bats → -0.633664}}}

{7.9705, {0.95986, {bgeodist → 0.870039, bair → -0.268084, bats → -0.633253}}}

{7.83797, {0.95986, {bgeodist → 0.870057, bair → -0.269454, bats → -0.63456}}}

{7.83421, {0.95986, {bgeodist → 0.861423, bair → -0.264109, bats → -0.616217}}}

{7.83667, {0.95986, {bgeodist → 0.870515, bair → -0.266964, bats → -0.628709}}}

{7.8278, {0.95986, {bgeodist → 0.87037, bair → -0.26881, bats → -0.632484}}}

{7.84216, {0.95986, {bgeodist → 0.869875, bair → -0.268094, bats → -0.634339}}}

{7.84772, {0.95986, {bgeodist → 0.870193, bair → -0.267737, bats → -0.632934}}}

{7.81795, {0.95986, {bgeodist → 0.870143, bair → -0.267054, bats → -0.631947}}}

{7.79524, {0.959786, {bgeodist → 0.862192, bair → -0.263089, bats → -0.624471}}}

{5.01624, {0.959711, {bgeodist → 0.871093, bair → -0.259692, bats → -0.633304}}}

{7.85539, {0.959786, {bgeodist → 0.873286, bair → -0.260299, bats → -0.662367}}}
```

(L2) In[299]:=

```
Do[Print[ Timing[
    ans = pairwiseMSE[obj2wayallsynergyeligpop, dataarray2way, { bgeodist, bair, bats},
      nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 500,
          "RandomSeed" → Floor[SessionTime[]] } } ] ] ]
,
{k,
 5}]
```

```
{50.1909, {0.95986, {bgeodist → 0.86158, bair → -0.258824, bats → -0.61943}}}

{49.9021, {0.95986, {bgeodist → 0.870146, bair → -0.269028, bats → -0.630008}}}

{49.8714, {0.95986, {bgeodist → 0.870128, bair → -0.267677, bats → -0.633214}}}

{49.8663, {0.95986, {bgeodist → 0.871148, bair → -0.269198, bats → -0.628976}}}
```

(L2) Out[299]=

```
$Aborted
```

Old estimates

- **Negative coefficient on elig*pop**

```
(L2) In[300]:= Do[Print[Timing[
                  ans = pairwiseMSE[obj2wayallsynergyeligpopneg, dataarray2way, { bgeodist, bair, bats},
                    nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 150,
                         "RandomSeed" → Floor[SessionTime[]] } } ] ] ]
               ,
               {k,
                3}]

{16.9061, {0.942508, {bgeodist → 56.9685, bair → 23.9127, bats → 19.576}}}

{17.2966, {0.942657, {bgeodist → 62.1413, bair → 30.5493, bats → 22.8292}}}

{16.7109, {0.942061, {bgeodist → 36.8352, bair → -1.01595, bats → 22.7904}}}
```

- **Standard errors for swaps**

```
(* estimateway3 = { bgeodist,bair,bats} /.
  {bair→0.2335209988622914`,bats→0.32074298170785814`,bgeodist→0.6539897051203899`} *)

(L2) In[304]:= estimateway3 =
               { bgeodist, bair, bats} /. {bgeodist → 0.86158, bair → -0.258824, bats → -0.61943}

(L2) Out[304]=
{0.86158, -0.258824, -0.61943}

(L2) In[305]:= obj2wayallsynergyeligpop[dataarray2way, bgeodist, bair, bats ] /.
               {bgeodist → 0.86, bair → -0.26, bats → -0.62}

(L2) Out[305]=
0.959637

(L2) In[306]:= Timing[pointcrout1 = pointIdentifiedCR[25, 200, estimateway3,
                  obj2wayallsynergyeligpop, { bgeodist, bair, bats}, datamap2way, dataarray2way,
                  asymptotics → coalitions, nMaximizeOptions → {Method → {"DifferentialEvolution",
                       "SearchPoints" → 100, "RandomSeed" → Floor[SessionTime[]] } }] ]

(L2) Out[306]=
{1590.16, {{{0.583184, 1.05502}, {-0.512067, 0.510039}, {-0.955361, -0.271107}},
  {{-0.249062, 0.511598, 0.26674}, {1.12643, -1.96155, -1.21788},
   {0.715579, -1.81429, 2.54325}, {0.197075, -1.45827, -1.05968},
   {-1.28106, -1.46682, 0.700528}, {1.12162, -2.85541, -0.967848},
   {1.20454, -1.7155, -0.541377}, {1.47586, 0.3687, -0.616987}, {1.18265, -5.08487, 0.255549},
   {0.0279207, 0.00623306, 0.4157}, {0.464494, -1.36903, -1.41585},
   {0.306496, -1.40333, -0.171626}, {1.13812, -1.69934, -0.902535},
   {0.0523838, -0.0355835, -0.0793261}, {1.15273, -2.14732, -0.982278},
   {0.750301, -1.87255, 1.1904}, {0.788632, -1.67191, 2.25766},
   {0.0639901, 0.444585, -0.582427}, {1.24171, -0.65752, -1.02752},
   {-0.0804214, -0.00824475, -0.559213}, {0.551835, -2.05642, 5.52738},
```

{0.752685, −1.92591, 2.44253}, {0.00617408, 0.00176998, 0.0616592},
{−1.46044, 0.769504, 1.42398}, {1.61309, −5.80479, −2.83373}, {0.753463, −2.1079, 2.68989},
{0.00979021, −0.0573654, 0.0951044}, {2.10184, 2.98873, −0.876352},
{1.30292, −3.14843, −1.10783}, {−0.563243, 0.159261, 0.367664},
{−0.199778, 1.22728, −0.112648}, {−1.29009, −1.20301, 0.724884},
{−2.44977, 1.88597, 1.26342}, {0.933347, −1.96631, 0.945848},
{1.10738, −5.01488, −0.489208}, {−0.566142, 0.15347, 0.502215},
{−1.30655, −1.62052, 0.969857}, {0.511065, −1.90185, −0.308349},
{1.37355, −4.43937, −2.58806}, {1.41914, −0.702088, −2.0113},
{−0.678132, 0.206016, 0.861576}, {1.04566, −0.263507, −0.616931},
{0.410008, −3.63272, −0.539121}, {−0.357989, 0.409758, −0.816753},
{−0.0560203, −3.10613, −0.40108}, {−1.02388, −1.77263, 0.881405},
{1.12065, −2.87142, −0.937312}, {−1.11709, 1.33039, 0.979778},
{0.0402083, −0.0491537, −0.206195}, {0.39029, −3.91169, −0.312444},
{1.35726, −0.679271, 2.34578}, {−1.26194, −1.54282, 0.864447},
{0.136672, −0.0204039, 1.08501}, {0.394693, −3.70564, −0.419734},
{0.019247, 0.00726576, 0.149889}, {−1.47395, 1.66757, −0.125679},
{−1.49697, 1.03894, 0.270615}, {1.13211, −1.69719, −0.947436},
{−0.0690119, −0.06149, −0.532344}, {1.80583, 1.87833, −2.62978},
{0.964673, −1.72065, 0.793529}, {−1.30676, −1.70834, 0.957781},
{0.0383549, −0.094512, −0.0367446}, {−0.156745, 0.480231, −0.221994},
{0.0409339, −0.0472993, −0.0804761}, {1.93829, 2.36599, −1.17828},
{1.57373, −6.35899, −2.40329}, {0.499651, −3.7648, 0.222463},
{0.263279, −1.32714, −0.344158}, {0.308353, −1.44502, −0.104105},
{−0.536128, 0.108091, 0.93256}, {0.295514, −1.68969, −1.13846},
{3.10542, −5.11808, −2.19004}, {−0.240489, 0.525047, 0.249008},
{−1.0701, −1.84174, 0.527968}, {−0.587179, 0.17705, 0.33103}, {2.23881, −1.00887, 0.10009},
{−0.679623, 0.247981, 0.955135}, {−0.478254, 0.129582, 0.911386},
{0.0437495, 0.00718971, −0.211409}, {−0.287142, −2.72339, −0.958973},
{0.6495, −1.62286, −1.78693}, {−1.50448, 1.0405, 0.30499},
{−0.683209, 0.551786, 1.06116}, {0.410589, −3.73632, −0.401411},
{0.0468551, −0.0285258, −0.0522514}, {−0.530573, 0.130804, 0.361678},
{0.734325, −1.88201, 2.34596}, {2.29588, 1.2687, −0.894736}, {1.00532, −4.84719, 3.02534},
{0.127738, 0.0734468, −0.0210379}, {0.444993, −3.68549, −0.0939083},
{1.11816, −1.7026, −0.948202}, {−0.0761133, −0.263203, −0.12882},
{−0.0148648, −0.0115973, 0.0131769}, {0.277456, −1.41672, −0.407773},
{−0.651293, 0.988605, 0.443807}, {−1.05367, 1.21592, 1.02272},
{−0.289267, −2.7341, −0.195062}, {−0.677397, 1.06687, 0.258069},
{0.120606, −2.45364, −1.31767}, {−0.631235, 0.940769, 0.365036},
{0.284069, −1.4159, −0.416861}, {−1.11944, −2.08895, 0.582876},
{0.553151, −0.0901986, 4.15219}, {−0.652561, 0.848729, 0.449382},
{−0.120234, −0.053959, −0.264851}, {0.0515837, −0.112259, −0.0299365},
{−0.930365, 1.30176, 0.633129}, {1.15046, 1.54697, −1.93715},
{0.531464, −2.03203, −0.594239}, {0.0294399, −0.0456346, −0.1254},
{1.43539, −0.441153, −1.39332}, {1.60058, −7.53151, −2.82084},
{0.139979, −0.037639, 0.0828666}, {−0.10319, −0.166434, −0.199228},
{0.0825926, 0.0365834, −0.632187}, {−1.11696, −1.36879, 0.56954},
{−1.19199, 1.26153, 1.03049}, {1.12167, −4.83502, −0.487334},
{1.15437, −2.14309, −0.970795}, {0.293603, −2.77114, −0.41656},
{1.10788, −4.84845, −0.408717}, {0.764831, −1.65997, 2.23652},
{−0.497481, 0.149362, 0.380331}, {2.2806, 1.28042, −0.575117},

{0.754508, −3.44075, 2.53622}, {−0.720823, 0.235038, 0.821645},
{−0.633861, 0.818322, 0.221547}, {1.48134, −0.857392, −1.42022},
{0.534584, −2.06129, −0.579792}, {1.41307, 1.93746, −0.869709},
{0.107763, −0.0118299, −0.14712}, {−0.273193, 0.131181, 0.608751},
{0.0391484, −0.0433419, 0.131154}, {−1.11458, 0.653759, 1.00964},
{−0.752628, 0.403507, 0.392213}, {1.16023, −2.12751, −0.972241},
{0.082286, 0.214882, −0.227713}, {1.52827, −5.07967, −1.28109},
{−0.332711, 0.508015, −0.517869}, {0.420244, −3.57864, −0.422498},
{−0.0852571, 2.64285, −1.15816}, {1.0419, −0.281804, −0.615275},
{−0.500761, 0.137038, −0.0111878}, {−1.09547, −1.98695, 0.503434},
{0.286764, −1.27756, −0.172005}, {1.53684, 0.72063, −2.6611},
{−0.263233, 0.132124, 0.58887}, {1.29718, −3.09029, −1.15517},
{1.68895, −7.59563, −2.95369}, {−0.0851957, −1.82153, 0.0302617},
{−0.501928, −2.7905, 0.104166}, {0.540582, −2.04578, −0.625638},
{0.234052, −1.39878, −0.571886}, {2.91323, −1.23968, −2.48052},
{−2.62975, 1.91194, 1.66799}, {0.0633031, 0.048102, −0.0222082},
{0.236635, −1.39612, −0.571079}, {−0.0650969, −1.86377, −0.0323736},
{−1.13498, −1.15789, 0.588679}, {−1.08279, −1.44928, −0.037994},
{−1.10422, 1.33243, 0.651743}, {−1.10289, −1.3611, 0.561644}, {1.48763, 1.92021, −0.55332},
{1.71371, −8.14411, −1.6799}, {0.0919959, −0.00832624, −0.141498},
{1.14211, 1.41971, −1.83058}, {0.301795, −1.8342, −0.43736},
{−0.330101, 0.494724, −0.560503}, {−0.274712, −2.75878, −0.730502},
{−0.687222, 0.506245, −0.391674}, {−0.115544, −0.0448117, −0.256208},
{−0.503448, 0.0955496, 0.951471}, {1.21415, −2.77914, −1.58555},
{−1.36355, 1.74685, −0.189211}, {1.17011, −0.525811, −1.63368},
{1.16717, 1.38768, −1.84131}, {0.038968, −0.0135882, 0.328419},
{−0.634005, 1.3264, 0.477589}, {1.11503, −2.7581, −0.92692}, {1.54761, −8.1864, −1.50637},
{0.052751, −0.0126296, −0.0724535}, {0.241033, −1.35879, −0.585042},
{−1.0641, −1.82718, 0.524}, {−1.28109, −1.4699, 0.946714},
{−0.255281, 0.130211, 0.510774}, {1.66129, 1.47368, −2.93158},
{−1.11149, −1.2928, 0.501797}, {0.0444295, −0.03897, −0.210133},
{0.240806, −1.39918, −0.630032}, {1.10579, −4.96827, −0.430571},
{−1.36512, −2.36795, 1.11524}, {−0.105168, −1.7465, −1.13299},
{−0.679966, 0.224595, 0.919355}, {1.38623, 0.349532, −2.53529},
{1.71491, 1.32573, −1.87237}, {0.0118091, −0.00820953, 0.0899846},
{0.29506, −1.43447, −0.115883}, {−1.07296, −1.86395, 0.430368}}}}

Table about counterfactuals

Cannot find the original file with the four regions in the counterfactual.

Read in MTA information

```
SetDirectory[ NotebookDirectory[] ];
SetDirectory["data"];
mtainfofilepre = Import["cntysv2000_census mta may2009.csv" ];
mtainfofilenames = mtainfofilepre[[1]]
mtainfofile = mtainfofilepre[[2 ;; 52]];

mtastouse = Complement[ Range[51], {25, 47, 49, 50, 51}];
mtastousenum = Length[mtastouse]

btainfofilepre = Import["cntysv2000_census bta may2009.csv"];
btainfofilenames = btainfofilepre[[1]]
btainfofile = btainfofilepre[[2 ;; 494]];
```

```
{MTA, Geo. Desc, Market Name, Market Name Abb. (30 Char.),  1990 census ,  2000 census }
```

```
46
```

```
{BTA, Geo. Desc, Market Name, Market Name Abb. (30 Char.),
 MTA, MTA Market Name, MTA Market Name Abb. (30 Char.),  7/1/99 est ,
  7/1/98 est ,  1990 revised ,  1990 census ,  2000 census }
```

Parameters

Set above

```
estimateway3
estimateway3a = Join[{1}, estimateway3]
```

```
{0.32, -0.16, 0.03}
```

```
{1, 0.32, -0.16, 0.03}
```

Actual winning packages in the C block

```
packwincovars = {Total[packwinpoptimeselig],
  Total[packwingeodistpop], Total[packwinairtravpop], Total[packwinatstravpop] }
packwincovarsparams = packwincovars estimateway3a
Total[packwincovarsparams]
```

```
{0.394117, 0.469871, 0.19729, 0.267802}
```

```
{0.394117, 0.150359, -0.0315664, 0.00803406}
```

```
0.520943
```

All licenses won by separate bidders

No complementarities, of course

```
allbtasseparateelig =
  PadRight[Reverse[Sort[eligibilitydatareal] ] , btastousenum, 0.00010893412935493899`];
allbtasseparatepop = Reverse[Sort[populationdatareal[[btastouse]] ] ] ;
allbtasseparatepopelig = allbtasseparatepop allbtasseparateelig;
Total[ allbtasseparatepopelig]
```

```
0.174831
```

MTAs

Packages to look at

```
btainfofilenamesmtapos = Position[btainfofilenames, "MTA"][[1, 1]]
mtapacks = Table[ Intersection[btastouse, Flatten@Position[
      btainfofile[[All, btainfofilenamesmtapos]] , mtastouse[[l]] ] ] , {l, mtastousenum}];
```

```
5
```

Sort by population

Somewhat unnecessary as the MTAs were numbered in order of population

```
mtapop = Table[ Total[populationdatareal[[mtapacks[[m]] ]] ], {m, mtastousenum}];
mtapopordering = Ordering[mtapop];
```

Assign to 47 top winners, by eligibility

```
maxeligwinners = Reverse[Sort[packwinelig][[winningpackagesusablenum - mtastousenum + 1 ;; ]] ];
```

- **Covars**

```
mtapopelig = mtapop maxeligwinners;
mtageodist = Table [ Total[ populationdatareal[[mtapacks[[l]] ]]
       Total[ Transpose[ KroneckerProduct[ populationdatarealtimes100[[mtapacks[[l]] ]],
            populationdatarealtimes100[[mtapacks[[l]] ]] ] /
           btadistancesdiomilestenforown[[ mtapacks[[l]], mtapacks[[l]] ]] ]  ] /
        Total[ Transpose[ KroneckerProduct[ populationdatarealtimes100[[mtapacks[[l]] ]],
            populationdatarealtimes100[[btastouse ]] ] /
           btadistancesdiomilestenforown[[ mtapacks[[l]], btastouse ]] ]  ] ] /
    Total[populationdatareal[[mtapacks[[l]] ]] ] , {l, mtastousenum}];

mtaairtrav = Table [Total[ populationdatareal[[mtapacks[[l]] ]]
       Total[Transpose[airlinetravelmatrixreal[[mtapacks[[l]], mtapacks[[l]] ]] ] ] /
        Total[Transpose[airlinetravelmatrixreal[[mtapacks[[l]], btastouse ]] ] ] ] /
    Total[populationdatareal[[mtapacks[[l]] ]] ] ,   {l, mtastousenum}] ;
mtaatstrav = Table [Total[ populationdatareal[[mtapacks[[l]] ]]
       Total[Transpose[tripsmatrixreal[[mtapacks[[l]], mtapacks[[l]] ]] ] ] /
        Total[Transpose[tripsmatrixreal[[mtapacks[[l]], btastouse ]] ] ] ] /
    Total[populationdatareal[[mtapacks[[l]] ]] ] ,   {l, mtastousenum}] ;

packwincovars = {Total[mtapopelig], Total[mtapop mtageodist],
  Total[mtapop mtaairtrav], Total[mtapop mtaatstrav] }
packwincovarsparams = packwincovars estimateway3a
Total[packwincovarsparams]
```

```
{0.19836, 0.722036, 0.0373545, 0.16777}
```

```
{0.19836, 0.231052, -0.00597671, 0.00503311}
```

```
0.428468
```

Four large regional licenses

- **Manually input groupings**

Old CSV file with four regions seems to be lost
Northeast, South, Midwest, West

Tennessee, Texas, and Oklahoma mostly in South
El Paso is in the West
Pittsburgh is in the Midwest

```
fourgroupsmtas =
 { {1, 8, 9, 10 , 35 }, {6, 7, 11, 13, 14, 15, 17, 23, 26, 28, 29, 33, 37, 40,  41, 44,  48},
  {3, 5, 12 , 16, 18, 19, 20, 21, 31, 32, 34, 38, 45, 46} , {2, 4, 22, 24, 27, 30, 36, 39, 42} }
```

```
{{1, 8, 9, 10, 35}, {6, 7, 11, 13, 14, 15, 17, 23, 26, 28, 29, 33, 37, 40, 41, 44, 48},
 {3, 5, 12, 16, 18, 19, 20, 21, 31, 32, 34, 38, 45, 46}, {2, 4, 22, 24, 27, 30, 36, 39, 42}}
```

- **Do the groupings by state**

Deactivated because of lost file

```
(* btainfofilenamefourregionpos = Position[mtainfofilenames,"Fourregions"][[1,1]]
   fourgroupsmtas =
  Table[ Flatten@Position[ mtainfofile[[All,btainfofilenamefourregionpos]],k],{k,4}] *)
```

Translate to BTAs

```
mtapacksall =
  Table[ Flatten@Position[ btainfofile[[All, btainfofilenamesmtapos]] , l ] , {l, 52}];
btasfourregions = Table[ Union[Flatten@mtapacksall[[fourgroupsmtas[[k]] ]] ], {k, 4}];
```

- **Covars**

```
fourelig = maxeligwinners[[ ;; 4]];
fourpop = Table [ Total[ populationdatareal[[ btasfourregions[[l]] ]] ], {l, 4}];
fourpoporder = Reverse[Ordering[fourpop] ];
fourpopelig = fourpop fourelig[[fourpoporder]];
fourgeodist = Table [
   Total[ populationdatareal[[btasfourregions[[l]] ]] Total[ Transpose[ KroneckerProduct[
            populationdatarealtimes100[[btasfourregions[[l]] ]], populationdatarealtimes100[[
             btasfourregions[[l]] ]] ] / btadistancesdiomilestenforown[[
            btasfourregions[[l]], btasfourregions[[l]] ]] ] ] /
        Total[ Transpose[ KroneckerProduct[ populationdatarealtimes100[[
             btasfourregions[[l]] ]], populationdatarealtimes100[[btastouse ]] ] /
           btadistancesdiomilestenforown[[ btasfourregions[[l]], btastouse ]] ] ] ] /
    Total[populationdatareal[[btasfourregions[[l]] ]] ] , {l,
    4}];

fourairtrav = Table [Total[ populationdatareal[[btasfourregions[[l]] ]]  Total[Transpose[
          airlinetravelmatrixreal[[btasfourregions[[l]], btasfourregions[[l]] ]] ] ] /
        Total[Transpose[airlinetravelmatrixreal[[btasfourregions[[l]], btastouse ]] ] ] ] /
    Total[populationdatareal[[btasfourregions[[l]] ]] ] ,   {l, 4}] ;
fouratstrav = Table [Total[ populationdatareal[[btasfourregions[[l]] ]]
      Total[Transpose[tripsmatrixreal[[btasfourregions[[l]], btasfourregions[[l]] ]] ] ] /
       Total[Transpose[tripsmatrixreal[[btasfourregions[[l]], btastouse ]] ] ] ] /
    Total[populationdatareal[[btasfourregions[[l]] ]] ] ,   {l, 4}] ;

packwincovars = {Total[fourpopelig], Total[fourpop fourgeodist],
  Total[fourpop fourairtrav], Total[fourpop fouratstrav] }
packwincovarsparams = packwincovars estimateway3a
Total[packwincovarsparams]
```

```
{0.504843, 0.959819, 0.373551, 0.579929}
```

```
{0.504843, 0.307142, -0.0597681, 0.0173979}
```

```
0.769615
```

- **One big winner (NextWave)**

```
packwincovars = {packwinelig[[85]] , 1, 1, 1}
packwincovarsparams = packwincovars estimateway3a
Total[packwincovarsparams]
```

```
{0.712628, 1, 1, 1}
```

```
{0.712628, 0.32, -0.16, 0.03}
```

```
0.902628
```

Prices estimator

Normalize price to be in **trillions** manually, in the maximum score functions

Only elig*pop geo*pop

- **Objective function**

```
obj1eliggeo[data_, belig_, bgeo_] :=
  Module[ {values, onesorzeros},
   values = data[[1]] / 1 000 000.0 + belig * data[[3]] + bgeo * data[[9]] + 0.000001;
   onesorzeros = values / Abs[values] + 1.0;
   Total[onesorzeros] / 2. / Length[data[[1]] ]
  ];
```

```
obj1eliggeoneg[data_, belig_, bgeo_] :=
  Module[ {values, onesorzeros},
   values = -data[[1]] / 1 000 000.0 + belig * data[[3]] + bgeo * data[[9]] + 0.000001;
   onesorzeros = values / Abs[values] + 1.0;
   Total[onesorzeros] / 2. / Length[data[[1]] ]
  ];
```

- **Maximize, swaps**

```
Dimensions[dataarray1way]
```

```
{11, 73 409}
```

```
obj1eliggeo[dataarray1way, 1., 1.] // Timing
```

```
{0.011998, 0.830756}
```

```
obj1eliggeo[dataarray1way, 10.^5, 10.^5] // Timing
```

```
{0.024997, 0.824027}
```

```
obj1eliggeo[dataarray1way, 10.^9, 10.^9] // Timing
```

```
{0.024996, 0.824027}
```

```
Do[ Print[Timing[ ans = pairwiseMSE[obj1eliggeo, dataarray1way,
      { belig, bgeodist}, nMaximizeOptions → {Method → {"DifferentialEvolution",
          "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]] } } ] ] ]

 , {k, 1, 15}]
```

```
{193.074, {0.863382, {belig → -0.037774, bgeodist → 53.4407}}}

{195.505, {0.863382, {belig → -0.0376957, bgeodist → 53.5689}}}

{193.302, {0.863382, {belig → -0.0377486, bgeodist → 53.673}}}

{193.572, {0.863382, {belig → -0.0375896, bgeodist → 53.6334}}}

{193.676, {0.863382, {belig → -0.0377759, bgeodist → 50.4842}}}

{196.232, {0.863382, {belig → -0.0379065, bgeodist → 70.6536}}}

{193.206, {0.863382, {belig → -0.0375579, bgeodist → 53.6286}}}

{174.981, {0.863382, {belig → -0.0377324, bgeodist → 70.6884}}}

{195.186, {0.863382, {belig → -0.0380718, bgeodist → 70.1054}}}

{194.119, {0.863382, {belig → -0.0376739, bgeodist → 70.2975}}}

{194.134, {0.863382, {belig → -0.0380255, bgeodist → 70.2452}}}

{195.39, {0.862932, {belig → -0.0223049, bgeodist → 38.5714}}}

{194.31, {0.863382, {belig → -0.0374363, bgeodist → 53.7861}}}

{194.388, {0.83675, {belig → 1.15902, bgeodist → 0.443438}}}

{195.126, {0.863382, {belig → -0.0379286, bgeodist → 53.6602}}}
```

- **Negative, swaps**

```
Do[ Print[Timing[ ans = pairwiseMSE[obj1eliggeoneg, dataarray1way,
      { belig, bgeodist}, nMaximizeOptions → {Method → {"DifferentialEvolution",
          "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]] } } ] ] ]

  , {k, 1, 5}]
```

```
{216.62, {0.913212, {belig → 0.362756, bgeodist → 0.118881}}}

{213.731, {0.913212, {belig → 0.362718, bgeodist → 0.118901}}}

{217.291, {0.913076, {belig → 0.367385, bgeodist → 0.115621}}}

{217.438, {0.913212, {belig → 0.361357, bgeodist → 0.115814}}}

{214.62, {0.913185, {belig → 0.364177, bgeodist → 0.119753}}}
```

```
Do[ Print[Timing[ ans = pairwiseMSE[obj1eliggeoneg, dataarray1way,
      { belig, bgeodist}, nMaximizeOptions → {Method → {"DifferentialEvolution",
          "SearchPoints" → 150, "RandomSeed" → Floor[SessionTime[]] } } ] ] ]

  , {k, 1, 15}]
```

```
{427.818, {0.913185, {belig → 0.363948, bgeodist → 0.119104}}}

{424.857, {0.913199, {belig → 0.385143, bgeodist → 0.127042}}}

{422.717, {0.913199, {belig → 0.362927, bgeodist → 0.119192}}}

{424.581, {0.913171, {belig → 0.360837, bgeodist → 0.117487}}}

{424.237, {0.913199, {belig → 0.362609, bgeodist → 0.116329}}}

{424.129, {0.913171, {belig → 0.361044, bgeodist → 0.117504}}}

{425.577, {0.913185, {belig → 0.386628, bgeodist → 0.127648}}}

{425.44, {0.913171, {belig → 0.363655, bgeodist → 0.116519}}}

{423.019, {0.913199, {belig → 0.36381, bgeodist → 0.116713}}}

{423.166, {0.913185, {belig → 0.362524, bgeodist → 0.119032}}}

{424.694, {0.913199, {belig → 0.362508, bgeodist → 0.118913}}}

{423.257, {0.913199, {belig → 0.360918, bgeodist → 0.115883}}}

{424.125, {0.913212, {belig → 0.36262, bgeodist → 0.118893}}}

{423.848, {0.913171, {belig → 0.386152, bgeodist → 0.127666}}}

{427.839, {0.913171, {belig → 0.361314, bgeodist → 0.117898}}}
```

```
Do[ Print[Timing[ ans = pairwiseMSE[obj1eliggeoneg, dataarray1way,
      { belig, bgeodist}, nMaximizeOptions → {Method → {"DifferentialEvolution",
          "SearchPoints" → 500, "RandomSeed" → Floor[SessionTime[]] } } ] ] ]

 , {k, 1, 3}]
```

```
{1413.81, {0.913212, {belig → 0.362595, bgeodist → 0.118717}}}

{1404.68, {0.913212, {belig → 0.385059, bgeodist → 0.127048}}}

{1405.52, {0.913185, {belig → 0.363819, bgeodist → 0.118971}}}
```

■ Standard errors, geo only

```
estimateway2 = {belig, bgeodist} /. {belig → 0.36, bgeodist → 0.12}
```

```
{0.36, 0.12}
```

```
Timing[pointcrout1 = pointIdentifiedCR[25, 200, estimateway2,
   obj1eliggeoneg, {belig, bgeodist}, datamap1way, dataarray1way,
   asymptotics → coalitions, nMaximizeOptions → {Method → {"DifferentialEvolution",
       "SearchPoints" → 100, "RandomSeed" → Floor[SessionTime[]] }}] ]
```

```
{18075.5, {{{-0.132536, 0.411893}, {-0.22818, 0.151314}},
  {{0.0143215, -0.00377836}, {-0.414974, -0.170881}, {-0.0055361, -0.0241438},
   {4.46586, 1.51396}, {1.2411, 0.370204}, {4.43683, 1.50753}, {1.43376, 0.433462},
   {4.50665, 1.53417}, {0.129362, 0.0210682}, {0.169698, 0.0630507}, {4.43674, 1.50597},
   {2.20386, 0.67375}, {1.27712, 0.380707}, {-0.115983, -0.0765063}, {0.302225, 0.030232},
   {-0.235896, -0.113906}, {0.00409133, -0.0134582}, {4.47765, 1.42587}, {1.632, 0.506291},
   {4.17654, -0.307669}, {4.34157, 1.44862}, {-0.297286, -0.138907}, {4.46355, 1.51364},
   {0.309144, 0.0355494}, {1.91026, -0.456588}, {4.43032, 1.47573}, {4.32348, 1.46824},
   {1.43297, 0.469899}, {0.174713, -4.54309 × 10^-6}, {-0.609708, -0.0842288},
   {0.00377569, -0.010402}, {1.28019, 0.397759}, {0.177084, -0.0134165},
   {-0.376836, -0.14667}, {1.04116, 0.302342}, {-0.125214, -0.0770048},
   {0.00480693, -0.00225659}, {4.29789, 1.39176}, {-0.208425, -0.140571},
   {-0.18422, -0.110942}, {0.0353615, 3.72733}, {-0.045728, -0.0322443}, {4.30529, 1.5542},
   {1.37298, 0.411489}, {6.4157, -0.124173}, {1.40901, 0.425581}, {1.93283, 0.629405},
   {0.0144199, -0.00386974}, {0.0031861, -0.0142486}, {4.48402, 1.52078},
   {4.30272, 1.56918}, {4.46615, 1.51148}, {4.43637, 1.43631}, {-0.217242, -0.0893921},
   {0.0143199, -0.00407813}, {4.44107, 1.50895}, {-0.124816, -0.0749633},
   {-0.403143, -0.288699}, {4.43638, 1.47795}, {0.12553, 0.0368657}, {0.169657, 0.0629704},
   {-0.401345, -0.179308}, {-0.187883, -0.105934}, {5.19525, 1.67142}, {2.02954, 0.655124},
   {-0.478426, 3.35316}, {4.36311, 1.60386}, {1.25261, 0.373673}, {-0.48527, 3.21006},
   {1.39598, 0.438399}, {0.125984, 0.0351608}, {4.58279, 1.52314}, {4.47967, 1.51776},
   {4.34909, 1.47818}, {-0.599513, -0.0490223}, {2.06487, 0.746645}, {0.129306, 13.2182},
   {-0.473836, -0.217817}, {-0.477039, 3.35518}, {-0.0629008, -0.0440105},
   {4.32337, 1.46887}, {1.26464, 0.377377}, {1.65116, 0.525871}, {-0.22704, -0.111199},
   {0.122816, 0.0368565}, {4.62433, 1.57942}, {4.43303, 1.4334}, {1.6312, 0.514929},
   {0.151595, -0.0182434}, {4.43633, 1.51085}, {4.33667, 1.42933}, {0.168478, 0.0511895},
   {4.54095, 1.51366}, {4.43232, 1.50263}, {-0.503074, -0.205574}, {2.41875, -0.398908},
   {0.00487172, -0.0209981}, {-0.217261, -0.0914953}, {4.45024, 1.61547}, {4.47754, 1.52559},
   {1.81963, 0.561395}, {-0.401979, -0.180306}, {2.54412, 0.787322}, {1.59053, 0.488064},
   {1.27289, 0.378876}, {1.91494, 0.64082}, {1.25454, 0.374479}, {-0.401823, -0.179371},
   {4.32883, 1.46364}, {1.64473, 0.50852}, {-0.401354, -0.198579}, {-0.285626, -0.128451},
   {-0.126059, -0.0763857}, {1.32765, 0.380503}, {4.36487, 1.60608}, {1.59042, 0.463738},
   {1.58686, 0.463595}, {1.93382, 0.640007}, {0.313984, 0.0367634}, {4.45999, 1.51709},
   {4.44039, 1.51122}, {4.30513, 1.57228}, {1.42962, 0.427246}, {4.46376, -0.295082},
   {0.125916, 0.0354191}, {1.22942, 0.359501}, {4.46541, 1.51402}, {4.43686, 1.51022},
   {0.00403021, -0.0219367}, {4.53953, 1.51339}, {4.46536, 1.52577}, {1.63463, 0.481316},
   {4.30182, 1.56939}, {-0.43169, -0.174106}, {0.0159351, -0.00287887}, {1.43746, 0.430237},
   {-0.485776, 3.20338}, {1.51164, 0.438055}, {4.47917, 1.49268}, {0.698177, 0.171969},
   {4.43646, 1.50552}, {4.46137, 1.51741}, {1.67367, 0.535222}, {2.25171, 0.688274},
   {0.00417851, -0.062711}, {-0.0382223, -0.0201841}, {1.43369, 0.432906},
   {6.38899, -0.126153}, {1.28082, 0.397889}, {4.4112, 1.61689}, {1.27433, 0.3739},
   {4.45407, 1.43652}, {4.42477, 1.43245}, {2.22841, 0.725078}, {4.31019, 1.46418},
   {4.43328, 1.54102}, {1.27573, 0.379539}, {1.63204, 0.513883}, {1.26584, 0.376207},
   {4.36445, 1.48514}, {4.33064, 1.47903}, {-0.36733, -0.187288}, {0.0212137, -0.0604841},
   {4.46842, 1.51308}, {1.43071, 0.431919}, {2.21274, 0.670892}, {0.0140601, -0.00393629},
   {-0.183115, -0.111046}, {4.2058, -0.339675}, {1.43256, 0.432642}, {0.0244683, -0.0575503},
   {4.44462, 1.51333}, {-0.364221, -0.148342}, {0.317889, 0.0877246},
   {0.0208804, -0.0000879187}, {-0.122393, -0.0769071}, {4.46388, 1.51857},
   {-0.197039, -0.113703}, {-0.401873, -0.179349}, {4.46642, 1.51457}, {0.145994, 0.0422153},
   {0.172238, -0.0110305}, {2.3445, 0.733257}, {-0.213259, -0.145951}, {0.86525, 0.232474},
   {-0.363958, -0.147982}, {1.43105, 0.426335}, {4.4529, 1.51669}, {0.168064, 13.8449},
   {-0.471768, -0.216322}, {-0.405916, -0.197918}, {1.43631, 0.431312}, {4.34143, 1.47731},
   {-0.414772, -0.171001}, {4.44237, 1.60908}, {4.30592, 1.45412}, {0.170338, 0.0583403},
   {4.44304, 1.60121}, {-0.0978039, -0.0513955}, {0.1696, -0.00106088}}}}
```

0.914

Swaps, air and ats

- **Objective function**

```
obj1eliggeotrav[data_, belig_, bgeo_, bair_, bats_] :=
  Module[ {values, onesorzeros},
   values = data[[1]] / 1 000 000. + belig * data[[3]] +
     bgeo * data[[9]] + bair * data[[10]] + bats * data[[11]] + 0.000001;
   onesorzeros = values / Abs[values] + 1.0;
   Total[onesorzeros] / 2. / Length[data[[1]] ]
  ];

obj1eliggeonegtrav[data_, belig_, bgeo_, bair_, bats_] :=
  Module[ {values, onesorzeros},
   values = -data[[1]] / 1 000 000. + belig * data[[3]] +
     bgeo * data[[9]] + bair * data[[10]] + bats * data[[11]] + 0.000001;
   onesorzeros = values / Abs[values] + 1.0;
   Total[onesorzeros] / 2. / Length[data[[1]] ]
  ];
```

- **Swaps**

```
obj1eliggeonegtrav[dataarray1way, 3., 3., 3., 3.] // Timing
```

```
{0.034995, 0.877249}
```

```
Timing[ans = pairwiseMSE[obj1eliggeonegtrav, dataarray1way,
   { belig, bgeodist, bair, bats}, nMaximizeOptions → {Method → {"DifferentialEvolution",
       "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]] } } ] ]
```

```
{306.767,
 {0.913485, {belig → 0.37125, bgeodist → 0.11975, bair → -0.0783443, bats → 0.0392136}}}
```

```
Do[ Print[
  Timing[ ans = pairwiseMSE[obj1eliggeonegtrav, dataarray1way, { belig, bgeodist, bair, bats},
     nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 150,
         "RandomSeed" → Floor[SessionTime[]] } } ] ] ]

 , {k, 1, 15}]
```

```
{602.546, {0.913594, {belig → 0.373281, bgeodist → 0.123893, bair → -0.07054, bats → 0.0354246}}}
{601.734, {0.91343, {belig → 0.370403, bgeodist → 0.121291, bair → -0.102469, bats → 0.0457675}}}
{601.631, {0.913539, {belig → 0.362167, bgeodist → 0.11865, bair → -0.090557, bats → 0.00465784}}}
{605.505, {0.913566, {belig → 0.368075, bgeodist → 0.118932, bair → -0.0765356, bats → 0.0174612}}}
{601.27, {0.913553, {belig → 0.384165, bgeodist → 0.131868, bair → -0.115727, bats → 0.0444309}}}
{603.574, {0.91358, {belig → 0.366857, bgeodist → 0.124657, bair → -0.0863608, bats → 0.0280298}}}
{605.669, {0.913457, {belig → 0.370726, bgeodist → 0.126684, bair → -0.110664, bats → 0.0362624}}}
{609.156, {0.913485, {belig → 0.386771, bgeodist → 0.130304, bair → -0.0563828, bats → 0.0383066}}}
{608.036, {0.91358, {belig → 0.367843, bgeodist → 0.121215, bair → -0.0943255, bats → 0.0419041}}}
{605.047, {0.913485, {belig → 0.374258, bgeodist → 0.119371, bair → -0.069502, bats → 0.0461132}}}
{610.24, {0.913526, {belig → 0.370228, bgeodist → 0.120164, bair → -0.0705257, bats → 0.0466106}}}
{606.504, {0.913457, {belig → 0.366352, bgeodist → 0.124046, bair → -0.0658645, bats → 0.0534366}}}
{601.998, {0.913648, {belig → 0.364589, bgeodist → 0.120027, bair → -0.0932646, bats → 0.0150624}}}
{607.127, {0.91358, {belig → 0.369195, bgeodist → 0.120318, bair → -0.0778146, bats → 0.043809}}}
{609.952, {0.913485, {belig → 0.359544, bgeodist → 0.119936, bair → -0.0866295, bats → 0.00885955}}}
```

```
Do[ Print[
  Timing[ ans = pairwiseMSE[obj1eliggeonegtrav, dataarray1way, { belig, bgeodist, bair, bats},
     nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 500,
         "RandomSeed" → Floor[SessionTime[]] } } ] ] ]

 , {k, 1, 15}]
```

```
{2010.54, {0.913648, {belig → 0.367274, bgeodist → 0.120843, bair → -0.0895706, bats → 0.0210347}}}
{2001.21, {0.913621, {belig → 0.36795, bgeodist → 0.123155, bair → -0.0810312, bats → -0.00659101}}}
{2010.51, {0.913648, {belig → 0.361152, bgeodist → 0.121164, bair → -0.0812036, bats → 0.0266173}}}
{2012.31, {0.913594, {belig → 0.367373, bgeodist → 0.11818, bair → -0.0751573, bats → 0.0142619}}}
{2009.6, {0.913526, {belig → 0.370479, bgeodist → 0.119962, bair → -0.0704632, bats → 0.0260378}}}
{2012.06, {0.913635, {belig → 0.365353, bgeodist → 0.11988, bair → -0.0872374, bats → 0.0269993}}}
{2001.72, {0.913703, {belig → 0.364048, bgeodist → 0.122742, bair → -0.0882209, bats → 0.0299039}}}
{2009.24, {0.913498, {belig → 0.367475, bgeodist → 0.123976, bair → -0.0734892, bats → 0.0344969}}}
{1998.48, {0.913635, {belig → 0.366323, bgeodist → 0.119325, bair → -0.0944618, bats → 0.0265674}}}
{2005.16, {0.913621, {belig → 0.364642, bgeodist → 0.121927, bair → -0.076983, bats → 0.0287531}}}
{2001.22, {0.913566, {belig → 0.369872, bgeodist → 0.118474, bair → -0.0707595, bats → 0.041613}}}
{2003.26, {0.913621, {belig → 0.364424, bgeodist → 0.122065, bair → -0.0826764, bats → -0.00140275}}}
{2006.04, {0.913635, {belig → 0.361632, bgeodist → 0.118391, bair → -0.0927488, bats → 0.0115281}}}
{2014.18, {0.913621, {belig → 0.368499, bgeodist → 0.124142, bair → -0.0795896, bats → 0.0242268}}}
{1993.95, {0.913635, {belig → 0.365154, bgeodist → 0.118187, bair → -0.0918248, bats → 0.0470319}}}
```

- **Positive**

```
Timing[ans = pairwiseMSE[obj1eliggeotrav, dataarray1way,
   {belig, bgeodist, bair, bats}, nMaximizeOptions → {Method → {"DifferentialEvolution",
       "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]]}}]]
```

```
{287.593, {0.874974, {belig → 0.200574, bgeodist → 193.191, bair → 5.343, bats → 1.86974}}}
```

- **Standard errors, geo+ travel**

```
estimateway1 =
 {belig, bgeodist, bair, bats} /. {belig → 0.36, bgeodist → 0.12, bair → -0.09, bats → 0.03}
```

```
{0.36, 0.12, -0.09, 0.03}
```

```
Timing[pointcrout1 = pointIdentifiedCR[25, 200, estimateway1,
   obj1eliggeonegtrav, {belig, bgeodist, bair, bats}, datamap1way, dataarray1way,
   asymptotics → coalitions, nMaximizeOptions → {Method → {"DifferentialEvolution",
       "SearchPoints" → 150, "RandomSeed" → Floor[SessionTime[]]}}]]
```

```
{42002.9,
 {{{-0.153318, 0.416753}, {-4.81772, 0.152841}, {-0.222855, 0.039845}, {-0.805426, 0.189503}},
  {{-0.407302, -0.183029, 0.26408, -0.449391}, {0.352652, 0.0451702, 0.23112, 0.00465794},
   {0.054864, 0.0190027, 0.0777568, 0.500538}, {1.43787, 0.428368, 0.25572, -0.21243},
   {4.31631, 1.45178, -0.115086, -0.5309}, {4.30529, 1.4315, 0.0528796, -0.698078},
   {2.47794, 0.953827, -1.2655, 1.65476}, {4.33346, 1.50933, -0.113388, 0.943856},
   {4.51045, 1.50374, 0.166979, -0.84027}, {1.58564, 0.486709, 0.743816, 0.113868},
   {-0.119422, -0.068416, 0.0994445, -0.0682922}, {7.04691, -0.108335, 1.24184, -0.391151},
   {1.7388, 0.616757, -0.675779, -0.466783}, {4.55953, 1.51146, 0.229073, -0.710636},
   {1.58459, 54.1275, 2.64155, -0.830851}, {1.40742, 0.419954, 0.205278, -0.0634113},
   {0.155493, 0.0268308, 0.420006, -0.649933}, {-0.412752, -0.165416, 0.227556, 0.0527296},
   {-0.0150927, 0.156442, -0.167257, 1.35914}, {1.57844, 1.87636, -1.55802, -1.27647},
   {4.32305, 1.44742, 0.018111, -0.932781}, {1.51826, 0.475983, 0.928828, -0.110973},
   {1.44199, 0.438942, -0.166804, -0.237697}, {1.6159, 0.508256, -0.153417, 0.0598641},
   {4.4099, -0.223484, 1.22486, -0.989679}, {-0.417182, -0.196189, 0.175742, -0.479784},
   {4.21205, 1.39679, 0.409328, -1.23406}, {1.52219, 0.515755, 0.747731, -0.277799},
   {4.33873, 1.46298, 0.16538, -1.39728}, {-0.147383, -0.0334498, -0.0658136, 0.354362},
   {4.48556, 1.5097, 0.0646347, -1.71956}, {1.29579, 0.389017, 0.662061, -0.140872},
   {-0.600514, -0.192026, 0.1357, 0.262381}, {1.56061, 0.489921, 0.691124, -1.04114},
   {4.33176, 1.45909, 0.157378, -0.122542}, {0.0243707, -0.0138399, -0.00227129, 0.436168},
   {-0.126411, -0.034501, -0.0943018, -0.288632}, {-0.411966, -0.146508, 0.106961, 0.146726},
   {-0.0852587, 2.90966, -0.424339, 1.34965}, {4.32221, 1.45467, 0.149503, -0.51483},
   {-0.0919082, -0.0226124, -0.101104, -0.231984},
   {4.30853, 1.45163, 0.0949485, -1.37031}, {1.64575, 0.571169, -0.726187, -0.281289},
   {-0.465277, 3.23655, 0.334352, 0.241304}, {-0.121693, -0.0849372, -0.1416, 0.0372217},
   {4.96632, 2.29884, -0.231812, 17.5906}, {4.30659, 1.39352, 0.140713, -0.0784436},
   {4.4392, 1.43344, 0.131112, 0.259725}, {4.41248, 1.45761, 0.119429, -0.931113},
   {0.274553, 0.160959, -0.668673, 0.617419}, {-0.150884, -0.04771, -0.0567933, 0.0784021},
   {-0.144322, -0.0377289, 0.0207521, 0.336545}, {1.57358, 0.482723, 0.880484, 0.085618},
   {0.0138246, 0.0171869, 0.0406295, 0.00110589}, {1.4956, 0.487669, -0.0871496, -0.791784},
```

{2.09621, 0.660805, −0.223478, −0.0647325}, {1.27373, 2.1075, −1.78222, 5.23462},
{1.26638, 0.376752, 0.514064, −0.231285}, {−0.0314041, −0.0116796, 0.230149, −0.0401466},
{4.43672, 1.48498, 0.0411754, −1.615}, {1.60926, 49.6388, 2.754, −0.307621},
{4.45334, 1.50845, 0.452546, 0.257073}, {1.57541, 0.461274, −0.606775, 0.156388},
{1.4132, 0.429337, 0.631037, −0.842153}, {0.386378, 0.123086, 0.103339, −0.35372},
{1.67537, 0.77548, −0.11876, 6.38472}, {1.57535, 0.483244, 0.539831, −0.818314},
{0.0399521, 0.0273513, −0.303406, 0.11466}, {−0.101981, −0.0730748, 0.0594819, −0.270843},
{1.63506, 0.508726, −0.0471561, 0.165475}, {4.4125, 1.45832, 0.11919, −0.970526},
{4.45338, −0.164509, 0.348228, −0.973117}, {1.62746, 0.505202, 0.259008, −0.441828},
{0.0764776, 3.39814, 0.304573, 1.42654}, {1.63434, 50.946, 2.67202, −0.441219},
{1.82992, 0.821037, −0.874632, 7.15298}, {−0.0679124, −0.0391669, −0.513102, −0.0637414},
{4.44323, 1.63122, 0.508811, −0.0509003}, {0.0556002, 0.00716308, −0.272125, 0.0680498},
{4.45707, 1.51267, 0.405887, 0.358443}, {−0.371186, −0.127642, 0.228041, 0.268968},
{−0.057673, 2.98569, 0.139688, 1.17418}, {1.63297, 0.503982, 0.305581, −0.668364},
{−0.66575, 0.152396, −0.291227, 0.278477}, {−0.399686, 0.242426, −0.514248, 0.802885},
{4.42265, 1.45785, 0.137905, −1.17509}, {1.49509, 0.455243, −0.0896952, 0.0434317},
{4.31003, 1.43144, 0.445894, −1.44233}, {4.45889, −0.325614, 0.653863, −0.141473},
{4.43027, 1.50962, 0.1537, 0.265605}, {0.00366113, 0.0131032, −0.0704925, 0.246807},
{−0.188486, −0.0325085, 0.0644465, 0.940465}, {1.42282, 0.51885, −0.660949, −0.313692},
{2.14954, −0.33496, −0.0882071, −0.739392}, {0.102056, 0.0566908, −0.395132, −0.0414638},
{4.42514, 1.50205, −0.0396055, −1.38956}, {2.31642, 0.745633, 0.254867, −0.538111},
{4.37195, −0.291512, 0.604013, −0.17279}, {4.4293, 1.46205, 0.145284, 0.379579},
{−0.499211, −0.181391, 0.0619166, −0.441011}, {4.47085, 1.47732, −0.00737887, −0.320186},
{1.18436, 0.346026, −0.164087, 0.106099}, {4.38537, 1.46853, −0.00796439, −1.41968},
{2.17816, −0.302777, −0.0735855, −0.996515}, {−0.454797, −0.143704, 0.0399325, 0.481125},
{4.46451, 1.44926, 0.10449, −0.125841}, {−0.0675971, 0.0422442, −0.116828, 1.01088},
{1.4227, 0.431598, 0.485325, −0.219254}, {0.172344, 0.0596894, −0.117291, 0.202177},
{−0.0510201, −0.049044, 0.371692, −0.468984}, {2.20427, 0.705813, 0.241508, −0.829322},
{1.80394, 0.643127, −0.800903, 0.118}, {0.133247, 3.2842, 0.170244, 1.61795},
{6.58922, 0.0344402, 0.108731, −1.01986}, {1.40614, 0.420526, −0.125442, 0.139677},
{2.06112, 51.6767, −0.636636, 3.31565}, {−0.144482, −0.0853388, 0.0635054, 0.390362},
{−0.173313, −0.107014, 0.232003, −0.528295}, {4.46415, 1.46223, 0.117971, 0.0430568},
{4.32383, 1.43233, 0.180002, −1.22719}, {−0.213272, −0.0903979, −0.0210333, −0.431849},
{2.59365, 1.25348, −1.0094, 9.12422}, {0.168194, 0.978404, −1.31372, 2.35016},
{−0.599496, −0.205584, −0.530634, −0.0454282}, {−0.00971221, −0.0365408,
 0.00127594, −0.0465751}, {1.56998, 0.484603, 0.834463, −0.882184},
{4.43019, −0.15736, 0.27112, −0.975164}, {−0.523234, −0.198845, 0.142098, −0.460713},
{−0.573644, −0.236251, −0.519501, −0.0173229},
{4.56136, 1.49805, 0.107076, −1.15912}, {0.00737828, −0.053671, −0.0601231, 0.163379},
{1.94332, 0.971049, −0.480268, 8.34032}, {−0.164132, −0.0865463, 0.199043, −0.576818},
{−0.166569, −0.0938149, 0.249121, −0.578239}, {4.43311, 1.49351, 0.142002, −1.5216},
{1.40568, 0.443456, 0.480841, −0.117741}, {1.57794, 15.935, 0.983862, 2.18896},
{4.53671, 1.49927, 0.176557, −1.16951}, {−0.23658, −0.102313, 0.0199063, −0.516961},
{4.46998, 1.5176, 0.454116, 0.0926292}, {1.53186, 0.477045, 0.672143, −1.06384},
{4.41817, 1.45753, 0.245328, −1.24602}, {0.0339199, 0.0439102, −0.555801, 0.311461},
{0.868868, 0.200762, 0.626515, −0.172193}, {4.46206, 1.5125, 0.447426, 0.329314},
{4.2931, 1.45157, 0.14312, −1.71362}, {1.91169, 0.668277, −0.582986, 0.218633},
{1.94462, −0.428913, 0.828992, −0.595705}, {0.128367, 3.27168, 0.189025, 1.72875},
{4.32225, 1.44134, 0.583597, −1.47055}, {1.93791, 0.900031, −0.602208, 7.26948},
{1.29233, 2.0247, −1.19711, 5.281}, {−0.48458, −0.192407, 0.153363, −0.400357},
{4.32662, 1.45951, 0.0886266, −0.14338}, {2.2013, 0.705962, 0.316815, −0.827208},

{1.56588, 0.484586, 0.725621, −0.838421}, {1.28445, 0.387545, 0.386266, −0.326606},
{4.29668, 1.43217, 0.256536, −0.514281}, {4.22695, 1.39819, 0.0193985, −0.455277},
{−0.477128, −0.145066, 0.105612, 0.487599}, {4.73255, 2.16712, 0.158865, 13.9383},
{−0.114794, −0.0243393, −0.121475, 0.440084}, {0.047901, 0.0805471, 0.0102614, 1.8352},
{0.530901, −0.464623, 0.869952, 1.55}, {−0.0801862, −0.0359589, 0.00840922, −0.328976},
{0.0142303, 2.93682, 0.0336155, 1.27494}, {4.39562, −0.171481, 0.35032, −0.966773},
{−0.185, −0.0799419, 0.288242, −0.503056}, {−0.0101986, 0.00998262, 0.0214192, 0.323224},
{−0.447803, −0.183699, 0.143782, −0.295968}, {1.91671, 69.1892, −0.577457, 0.249745},
{5.38487, −0.0839568, 0.384251, −1.05139}, {4.3287, 1.44695, 0.577103, −1.36449},
{2.25794, 0.724113, 0.27683, −0.835477}, {1.89797, 1.04174, −1.37096, 7.70224},
{4.7528, 2.15475, −0.163187, 15.6286}, {4.42446, 1.4984, 0.0080515, −0.0687516},
{0.0254101, 0.0336381, −0.198897, −0.026316}, {1.41967, 0.433048, 0.678598, −0.841559},
{4.42785, 1.47845, 0.593405, −1.5787}, {−0.102622, −0.0563767, −0.269564, −0.12297},
{−0.0518801, 3.03242, 0.144669, 1.782}, {0.658563, 0.157789, 0.355479, −0.209201},
{−0.113793, 2.92426, −0.173713, 1.15931}, {−0.528735, −0.193085, 0.266189, 0.240889},
{1.57904, 0.549166, −0.623544, 0.133686}, {−0.0422697, 3.00009, 0.118075, 1.21408},
{1.29347, 0.364907, −0.313854, 0.387236}, {4.43361, 1.44018, 0.152222, −0.375391},
{1.96889, −0.448319, 0.543824, −0.372064}, {−0.103244, −0.0748669, 0.21707, −0.527121},
{1.54286, 0.462162, −0.0529618, 0.120343}, {0.65557, 0.154164, 0.410448, −0.192205},
{1.94714, 0.692522, 0.991077, 0.375934}, {2.07921, 0.767024, −0.0941637, 2.42757},
{4.43094, 1.49645, 0.0400789, −1.22274}, {4.4214, 1.50659, −0.0912397, −0.628304},
{1.24655, 0.37555, 0.524463, −0.493163}, {1.59723, 45.5235, 1.54902, 0.288784},
{0.0362492, 0.0479065, −0.340824, 0.410904}}}}