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[306]:= 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[309]:= 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[310]:= 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[311]:= eligibilitydatareal  = N[eligibilitydata] / Total[populationdata[[btastouse]] // N] ;
               populationdatareal = N[populationdata] / Total[ populationdata[[btastouse]] ];
```

Population of the new packages

```
(L2) In[313]:= pack1waypopPreRobust =
                 ToPackedArray[ Table [ Total[ populationdatareal[[packagesswapsonewayPreRobust[[l]] ]] ],
                   {l, licenseswapsonewaynumPreRobust}] ];
```

■ Compare to eligibility

```
(L2) In[314]:= popLessThanEligPreRobust = Table[
                  pack1waypopPreRobust⟦l⟧ ≤ eligibilitydatareal⟦packages1swapwinnerpackPreRobust⟦l⟧⟧,
                  {l, licenseswapsonewaynumPreRobust}] ;
               Tally[popLessThanEligPreRobust]

(L2) Out[315]=
          {{True, 73 409}, {False, 148 975}}
```

■ Redo key variables

```
(L2) In[316]:= licenseswapsoneway =
                 ToPackedArray[Pick[ licenseswapsonewayPreRobust, popLessThanEligPreRobust] ];
               licenseswapsonewaynum = Length[licenseswapsoneway]
               packagesswapsoneway =
                  ToPackedArray[Pick[packagesswapsonewayPreRobust, popLessThanEligPreRobust] ];
               packages1swapwinnerpack = ToPackedArray[
                  Pick[packages1swapwinnerpackPreRobust, popLessThanEligPreRobust] ];

(L2) Out[316]=
          73 409
```

■ License adds and subtracts

■ Combinations of one license and one winning package, pre initial eligibility

From different winners

```
(L2) In[517]:= combinationspackageslicensesPreRobust =
                  Tuples[ {Range[winningpackagesusablenum], btastouse}];
               combinationspackageslicensesdiffpackPreRobust =
                  Select[ combinationspackageslicensesPreRobust,
                    #[[1]] ≠ enclosingpackage[[ btatouselookupbta[[#[[2]] ]] ]] &];
               combinationspackageslicensesdiffpacknumPreRobust =
                 Length[combinationspackageslicensesdiffpackPreRobust]

(L2) Out[519]=
          40 320
```

■ Resulting packages, for adds and subtracts, pre initial eligibility

```
(L2) In[322]:= packagesaddsPreRobust = Table[
                  Union[ winningpackagesusable[[combinationspackageslicensesdiffpackPreRobust[[l, 1]] ]],
                    {combinationspackageslicensesdiffpackPreRobust[[l, 2]]} ],
                  {l, combinationspackageslicensesdiffpacknumPreRobust}];
               packagesaddsnumPreRobust = combinationspackageslicensesdiffpacknumPreRobust;
```

■ Population of new packages

```
(L2) In[324]:= pack1addpopPreRobust =
                 ToPackedArray[ Table [ Total[ populationdatareal[[packagesaddsPreRobust[[l]] ]] ],
                    {l, packagesaddsnumPreRobust}] ];
```

■ Winning packages link for comparison, pre eligibility

```
(L2) In[325]:= winnerpack1addPreRobust = combinationspackageslicensesdiffpackPreRobust[[All, 1]];
```

Which less than population

```
(L2) In[326]:= popLessThanEligPreRobustAdds =
                 Table[ pack1addpopPreRobust⟦l⟧ ≤ eligibilitydatareal⟦ winnerpack1addPreRobust⟦ l⟧ ⟧,
                  {l, packagesaddsnumPreRobust}] ;
               Tally[popLessThanEligPreRobustAdds]

(L2) Out[327]=
          {{True, 16 084}, {False, 24 236}}
```

■ Redo key variables with eligibility constraints

```
(L2) In[328]:= packagesadds =
                 ToPackedArray[Pick[ packagesaddsPreRobust, popLessThanEligPreRobustAdds] ];
               packagesaddsnum = Length[packagesadds]

(L2) Out[329]=
          16 084
```

■ Resulting packages, for subtracts, eligibility not an issue

```
(L2) In[330]:= packagessubtracts =
                 Table [ Complement[winningpackagesusable[[enclosingpackage[[b]] ]], {btastouse[[b]]} ],
                  {b, btastousenum}];
```

■ Winning packages link for comparison

```
(L2) In[331]:= winnerpack1add =
                 Pick[combinationspackageslicensesdiffpack[[All, 1]], popLessThanEligPreRobustAdds];
               winnerpack1sub = enclosingpackage;
```

■ Some variables redone

```
(L2) In[520]:= combinationspackageslicensesdiffpack =
                 Pick[combinationspackageslicensesdiffpackPreRobust, popLessThanEligPreRobustAdds];
               combinationspackageslicensesdiffpacknum = Length[combinationspackageslicensesdiffpack]

(L2) Out[521]=
          16 084
```

Pop, price, pop*eligibility & density*pop

Population

(L2) In[333]:=

```
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[340]:=

```
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[348]:=
        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[359]:= 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[367]:= geodistpower = 4
               btadistancesdiomilestenforown =
                 (btadistancesdiomiles + DiagonalMatrix[ Diagonal[btadistancesdiomiles] + 10.^10])^
                  geodistpower;

(L2) Out[367]=
          4

(L2) In[369]:=
```

- **Gravity equations**

(L2) In[370]:=

```
populationdatarealtimes100 = populationdatareal 100;

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[379]:= 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[388]:= 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[397]:= 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[403]:= 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[409]:= 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[415]:= 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[421]:= dataarray1way = ToPackedArray[ {pack1diffprice, pack1diffpop, pack1diffpoptimeselig,
                    pack1diffdensitypop, pack1diffgeodist, pack1diffgeodistelig, pack1diffairtrav,
                    pack1diffatstrav, pack1diffgeodistpop, pack1diffairtravpop, pack1diffatstravpop}];
               ByteCount[dataarray1way] / 1024.^2

(L2) Out[422]=
          6.16085
```

1 license adds

```
(L2) In[423]:= dataarray1add =
                 ToPackedArray[ {pack1adddiffprice, pack1adddiffpop, pack1adddiffpoptimeselig,
                   pack1adddiffdensitypop, pack1adddiffgeodist, pack1adddiffgeodistelig,
                   pack1adddiffairtrav, pack1adddiffatstrav, pack1adddiffgeodistpop,
                   pack1adddiffairtravpop, pack1adddiffatstravpop}];
               ByteCount[dataarray1add] / 1024.^2

(L2) Out[424]=
     1.34995
```

1 license substractions

```
(L2) In[425]:= dataarray1sub =
                 ToPackedArray[ {pack1subdiffprice, pack1subdiffpop, pack1subdiffpoptimeselig,
                   pack1subdiffdensitypop, pack1subdiffgeodist, pack1subdiffgeodistelig,
                   pack1subdiffairtrav, pack1subdiffatstrav, pack1subdiffgeodistpop,
                   pack1subdiffairtravpop, pack1subdiffatstravpop}];
               ByteCount[dataarray1sub] / 1024.^2

(L2) Out[426]=
     0.0404053
```

Toolkit map format

```
(L2) In[427]:= datamap1way = ToPackedArray[
                  Table[ {enclosingpackage[[btatouselookupbta[[ licenseswapsoneway[[l, 1]] ]] ]],
                    enclosingpackage[[btatouselookupbta[[ licenseswapsoneway[[l, 2]] ]] ]]} ,
                   {l, licenseswapsonewaynum}] ];

(L2) In[428]:= 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[430]:= dataarray1all = Join[ dataarray1way, dataarray1add, dataarray1sub, 2];
               datamap1all = Join[ datamap1way, datamap1add, datamap1sub];
               ByteCount[dataarray1all] / 1024.^2

(L2) Out[432]=
     7.55096
```

Export

```
(L2) In[433]:= 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[439]:= Mean[Transpose[dataarray1all] ]
               StandardDeviation[Transpose[dataarray1all] ]

(L2) Out[439]=
      {-7.67862, -0.000680528, 2.22701 × 10^-7, -1.04243, 0.0592403, 0.00178151,
       0.0000900759, 0.00124151, 0.000402639, 2.34675 × 10^-6, 0.0000162442}

(L2) Out[440]=
      {34.6515, 0.00267036, 0.000340006, 9.56759, 0.0888478,
       0.00530176, 0.00352482, 0.00885784, 0.00104793, 0.00027364, 0.000337946}
```

Swaps

```
(L2) In[441]:= Mean[Transpose[dataarray1way] ]
               StandardDeviation[Transpose[dataarray1way] ]

(L2) Out[441]=
      {-7.50394, -0.00059033, -0.000016974, -1.07839, 0.0674715,
       0.0016694, 0.000247533, 0.00178206, 0.000482511, -6.41669 × 10^-7, 0.000016208}

(L2) Out[442]=
      {35.4209, 0.00272618, 0.00020951, 9.81269, 0.0940868, 0.00421496,
       0.00361362, 0.00932549, 0.00107691, 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[443]:= Mean[Transpose[dataarray1add] ]
               StandardDeviation[Transpose[dataarray1add] ]

(L2) Out[443]=
      {-9.31888, -0.00117468, 0.0000542131, -1.02508, 0.0223342, 0.00223535,
       -0.00070933, -0.00138119, -8.31566 × 10^-6, -8.45208 × 10^-6, -0.0000152902}

(L2) Out[444]=
      {28.5698, 0.0021488, 0.000248663, 6.96963, 0.0432737, 0.00841111,
       0.00264547, 0.0045287, 0.000108629, 0.0000437262, 0.0000999808}

(L2) In[445]:= Mean[Transpose[dataarray1sub] ]
               StandardDeviation[Transpose[dataarray1sub] ]

(L2) Out[445]=
      {20.5685, 0.00208333, 0.000821077, 3.87693, 0.0370482,
       0.00371883, 0.00279607, 0.00645397, 0.00195779, 0.000821232, 0.00107845}

(L2) Out[446]=
      {70.4111, 0.00531856, 0.00347746, 27.9649, 0.0813,
       0.013002, 0.00805728, 0.0204561, 0.00436665, 0.00356578, 0.00399345}
```

Winning packages

```
(L2) In[447]:= packindataarrayall = {packwinprice, packwinpop, packwinpoptimeselig,
                   packwindensitypop, packwingeodist, packwingeodistelig, packwinairtrav,
                   packwinatstrav, packwingeodistpop, packwinairtravpop, packwinatstravpop};
```

```
(L2) In[448]:= Mean[ Transpose[ packindataarrayall ] ]
               StandardDeviation[ Transpose[ packindataarrayall] ]
               Map[Max, packindataarrayall ]
               Map[Min, packindataarrayall ]

(L2) Out[448]=
      {116.152, 0.0117647, 0.00463667, 21.8933, 0.169555,
       0.0190938, 0.0147153, 0.0318486, 0.00552789, 0.00232106, 0.00315061}

(L2) Out[449]=
      {496.128, 0.0442504, 0.0299441, 124.931, 0.228987,
       0.0658945, 0.0511259, 0.0778808, 0.023523, 0.0165767, 0.0201134}

(L2) Out[450]=
      {4200.77, 0.379645, 0.270545, 1111.52, 0.78546,
       0.373639, 0.395683, 0.48049, 0.198444, 0.150219, 0.182415}

(L2) Out[451]=
```

$\{0.102375, 0.000111362, 1.24014\times10^{-8}, 0.00259376, 4.11269\times10^{-36}, 3.37743\times10^{-39},$
$3.37919\times10^{-15}, 1.37552\times10^{-11}, 1.56195\times10^{-39}, 2.39575\times10^{-17}, 9.75203\times10^{-14}\}$

Data for matches only estimator

■ No need to recreate data, use indices into data for prices

```
(L2) In[522]:= 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[524]:= 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[527]:= 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[528]=
          {{True, 13428}, {False, 21531}}

(L2) Out[530]=
          13428
```

- **For additions, find corresponding subtraction**

```
(L2) In[532]:= addstosubstracts =
                 Table[ btatouselookupbta[[combinationspackageslicensesdiffpack[[l, 2]] ]] ,
                  {l, combinationspackageslicensesdiffpacknum}];
```

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

```
(L2) In[533]:= 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[544]:= 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[555]:= dataarray2way = ToPackedArray[ {pack2diffprice, pack2diffpop, pack2diffpoptimeselig,
                   pack2diffdensitypop, pack2diffgeodist, pack2diffgeodistelig, pack2diffairtrav,
                   pack2diffatstrav, pack2diffgeodistpop, pack2diffairtravpop, pack2diffatstravpop}];
               ByteCount[dataarray2way] / 1024.^2

(L2) Out[556]=
      1.12704
```

Adds

```
(L2) In[557]:= dataarray2add =
                 ToPackedArray[ {pack2adddiffprice, pack2adddiffpop, pack2adddiffpoptimeselig,
                   pack2adddiffdensitypop, pack2adddiffgeodist, pack2adddiffgeodistelig,
                   pack2adddiffairtrav, pack2adddiffatstrav, pack2adddiffgeodistpop,
                   pack2adddiffairtravpop, pack2adddiffatstravpop}];
               ByteCount[dataarray2add] / 1024.^2

(L2) Out[558]=
      1.34995
```

Toolkit map format

Swaps

Data map in terms of winning packages

```
(L2) In[559]:= datamap2way = ToPackedArray[
                  Table[ {enclosingpackage[[btatouselookupbta[[ licenseswapstwoway[[l, 1]] ]] ]],
                    enclosingpackage[[btatouselookupbta[[ licenseswapstwoway[[l, 2]] ]] ]]} ,
                   {l, licenseswapstwowaynum}] ];
```

Terms of winning licenses

```
(L2) In[560]:= datamap2waylic = ToPackedArray[licenseswapstwoway ];
```

Adds

Winning packages

```
(L2) In[561]:= datamap2add = datamap1add;
```

Winning licenses

```
(L2) In[562]:= datamap2addlic  =  Transpose[{combinationspackageslicensesdiffpack[[All, 2]],
                    combinationspackageslicensesdiffpack[[All, 2]]}];
```

Combine adds and swaps

```
(L2) In[563]:= dataarray2all = Join[ dataarray2way, dataarray2add, 2];
               datamap2all  = Join[datamap2way, datamap2add];
```

Export

```
(L2) In[565]:= 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[571]:= Mean[Transpose[dataarray2way] ]
               StandardDeviation[ Transpose[dataarray2way] ]

(L2) Out[571]=
        {-1.82167 × 10^-15, 5.72364 × 10^-19, 8.61364 × 10^-6, -1.52644 × 10^-17, 0.119214,
         0.00242131, 0.000310215, 0.0025149, 0.000612981, 9.03657 × 10^-7, 0.0000226012}

(L2) Out[572]=
        {5.12596 × 10^-15, 1.29898 × 10^-18, 0.0000527769, 5.56963 × 10^-16, 0.130166,
         0.00467327, 0.00148725, 0.0109254, 0.000784661, 0.0000112302, 0.000095513}

(L2) In[573]:= Mean[Transpose[dataarray2add] ]
               StandardDeviation[ Transpose[dataarray2add] ]

(L2) Out[573]=
        {2.55452 × 10^-14, 1.80341 × 10^-19, 0.000387052, -9.53748 × 10^-15, 0.0588848,
         0.00506162, 0.00102051, 0.00268248, 0.00121653, 0.000316838, 0.000442888}

(L2) Out[574]=
        {1.30419 × 10^-13, 6.38929 × 10^-18, 0.00125003, 4.8878 × 10^-14, 0.0911461,
         0.013678, 0.00521145, 0.016381, 0.00273614, 0.00148853, 0.00173897}
```

```
(L2) In[575]:= Mean[Transpose[dataarray2all]]
               StandardDeviation[Transpose[dataarray2all]]

(L2) Out[575]=
      {1.30932×10^-14, 3.58712×10^-19, 0.000214862, -5.20486×10^-15, 0.0863345,
       0.00386028, 0.000697326, 0.00260623, 0.000941915, 0.000173087, 0.000251657}

(L2) Out[576]=
      {9.73001×10^-14, 4.80142×10^-18, 0.000942526, 3.63954×10^-14, 0.114625,
       0.0106595, 0.00399159, 0.0141617, 0.00210961, 0.00111011, 0.0013023}
```

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[577]:= addpacksizes = Table[
                  Total[winningpackagesusablesizes[[datamap2add[[l]]]]], {l, Length[datamap2add]}];
```

New means

```
(L2) In[578]:= weightmeansadd = Total[addpacksizes Transpose[dataarray2add]] / Total[addpacksizes]

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

(L2) Out[578]=
      {2.55663×10^-14, 1.53207×10^-19, 0.000385386, -9.10908×10^-15, 0.0591768,
       0.0052742, 0.00102002, 0.00261156, 0.00121892, 0.000314431, 0.000436333}

(L2) Out[579]=
      {1.31144×10^-13, 6.21962×10^-18, 0.00120919, 4.79551×10^-14, 0.0906626,
       0.0142263, 0.00520176, 0.0160125, 0.00278283, 0.00144354, 0.00170974}
```

End of data stuff

No prices

Gets rid of adding licenses reduces complementarities

Only geo*pop and elig*pop

Objective function

```
(L2) In[581]:= 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[582]:= 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**

```
(L2) In[583]:= obj2waygeosynergyeligpop[dataarray2add, 1.] // Timing

(L2) Out[583]=
     {0.001051, 0.942365}

(L2) In[584]:= Timing[ans = pairwiseMSE[obj2waygeosynergyeligpop, dataarray2add,
                  { bgeodist}, nMaximizeOptions → {Method → {"DifferentialEvolution",
                      "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]] } } ] ]

(L2) Out[584]=
     {3.23824, {0.949702, {bgeodist → 6.73757}}}

(L2) In[585]:= Do[
                Print[
                 AbsoluteTiming[ans = pairwiseMSE[obj2waygeosynergyeligpop, dataarray2add, { bgeodist},
                    nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 75,
                        "RandomSeed" → Floor[SessionTime[]] } } ] ] ]
                ,
                {k,
                 1,
                 15}]
{3.094905, {0.949702, {bgeodist → 6.72852}}}

{4.244779, {0.949702, {bgeodist → 6.73944}}}

{4.059370, {0.949702, {bgeodist → 6.81853}}}

{3.379351, {0.949702, {bgeodist → 6.73178}}}

{4.099373, {0.949702, {bgeodist → 6.8077}}}

{3.300406, {0.949702, {bgeodist → 6.75319}}}

{4.239769, {0.949702, {bgeodist → 6.74255}}}

{3.541229, {0.949702, {bgeodist → 6.80831}}}

{3.796696, {0.949702, {bgeodist → 6.75919}}}

{3.772094, {0.949702, {bgeodist → 6.72987}}}

{3.840678, {0.949702, {bgeodist → 6.75831}}}

{3.273642, {0.949702, {bgeodist → 6.82261}}}

{3.273713, {0.949702, {bgeodist → 6.75036}}}

{2.763152, {0.949702, {bgeodist → 6.7478}}}

{3.260874, {0.949702, {bgeodist → 6.73426}}}
```

```
(L2) In[588]:= Do[
          Print[
           AbsoluteTiming[ans = pairwiseMSE[obj2waygeosynergyeligpop, dataarray2add, {bgeodist},
              nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 200,
                  "RandomSeed" → Floor[SessionTime[]] } } ] ] ]
          ,
          {k,
           1,
           15}]
{10.674885, {0.949702, {bgeodist → 6.74068}}}

{8.764432, {0.949702, {bgeodist → 6.81133}}}

{11.641323, {0.949702, {bgeodist → 6.81652}}}

{10.331445, {0.949702, {bgeodist → 6.81239}}}

{11.132177, {0.949702, {bgeodist → 6.81092}}}

(L2) Out[588]=
      $Aborted
```

- **Negatives**

```
(L2) In[586]:= Do[
          Print[AbsoluteTiming[
            ans = pairwiseMSE[obj2waygeosynergyeligpopneg, dataarray2add, {bgeodist},
              nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 75,
                  "RandomSeed" → Floor[SessionTime[]] } } ] ] ]
          ,
          {k,
           1,
           15}]
{6.143468, {0.908356, {bgeodist → 87.1525}}}

{6.111963, {0.913392, {bgeodist → 342.186}}}

{6.104931, {0.909227, {bgeodist → 155.309}}}

{6.161626, {0.909227, {bgeodist → 152.56}}}

{6.171385, {0.909164, {bgeodist → 119.668}}}

{6.157374, {0.909227, {bgeodist → 151.376}}}

{6.116190, {0.909227, {bgeodist → 154.76}}}

{6.108718, {0.909227, {bgeodist → 154.664}}}

{6.153617, {0.908791, {bgeodist → 105.951}}}

{6.115214, {0.909227, {bgeodist → 151.891}}}

{6.157919, {0.908232, {bgeodist → 85.3495}}}

{5.554092, {0.909227, {bgeodist → 151.479}}}

{6.284059, {0.909227, {bgeodist → 152.614}}}

{6.532852, {0.909102, {bgeodist → 119.919}}}

{6.452400, {0.914449, {bgeodist → 870.449}}}
```

```
(L2) In[587]:= Do[
          Print[AbsoluteTiming[
            ans = pairwiseMSE[obj2waygeosynergyeligpopneg, dataarray2add, { bgeodist},
              nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 150,
                   "RandomSeed" → Floor[SessionTime[]] } } ] ] ]
          ,
          {k,
           1,
           15}]

{12.682756, {0.909227, {bgeodist → 152.464}}}

{13.746784, {0.914387, {bgeodist → 740.501}}}

{12.683363, {0.914449, {bgeodist → 869.464}}}

(L2) Out[587]=
       $Aborted
```

- **Standard errors for adds**

```
(L2) In[589]:= estimateadd4 = { bgeodist} /. {bgeodist → 6.7}

(L2) Out[589]=
       {6.7}

(L2) In[590]:= Timing[pointcrout1 = pointIdentifiedCR[25, 150, estimateadd4,
            obj2waygeosynergyeligpop, { bgeodist}, datamap2add, dataarray2add,
            asymptotics → coalitions, nMaximizeOptions → {Method → {"DifferentialEvolution",
                "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]] } }] ]

(L2) Out[590]=
       {376.833,
        {{{-3.00517, 9.19933}}, {{0.2927}, {54.7732}, {-6.16089}, {89.4773}, {14.243}, {32.1581},
          {35.4982}, {1.45204}, {-6.18207}, {9.36409}, {-8.42533}, {17.5678}, {31.421}, {0.143456},
          {-20.1803}, {-13.832}, {-14.1975}, {17.797}, {145.747}, {0.197908}, {52.5399}, {32.1576},
          {-13.5143}, {24.5874}, {-1.10087}, {-26.9061}, {5.89048}, {-13.8926}, {142.802},
          {-15.8679}, {0.302443}, {0.555374}, {1.23225}, {36.5369}, {16.6941}, {89.4601},
          {-20.1346}, {54.858}, {10.4276}, {-13.1167}, {-16.4838}, {-20.1946}, {-13.1921},
          {5.91659}, {-19.2547}, {1.83944}, {-13.5164}, {76.2375}, {-12.7953}, {73.8181},
          {31.2458}, {18.4442}, {-20.2181}, {0.150469}, {-8.43595}, {0.281232}, {0.113381},
          {-8.48757}, {1.23755}, {-19.13}, {-13.1873}, {16.6425}, {89.432}, {-7.60841},
          {0.220313}, {-23.0427}, {-6.26601}, {-13.5951}, {2.15292}, {0.55187}, {0.219186},
          {32.024}, {-20.2405}, {-13.469}, {-9.30692}, {54.7939}, {0.173704}, {31.6298},
          {0.803855}, {32.8584}, {89.4256}, {6.34398}, {30.8662}, {35.5539}, {-13.087}, {55.9826},
          {35.5415}, {0.0437109}, {-4.3603}, {-13.1333}, {-15.9147}, {18.5984}, {-12.9358},
          {-10.2728}, {-20.1699}, {32.2914}, {31.4974}, {30.9851}, {24.5337}, {-13.9155},
          {-13.11}, {0.419541}, {89.7543}, {-13.9543}, {-6.19999}, {0.53882}, {1.90089},
          {-20.191}, {-13.5838}, {34.9225}, {-13.1238}, {35.7536}, {-13.5477}, {-20.2038},
          {8.94721}, {-20.225}, {-13.5424}, {-7.97555}, {16.923}, {-13.155}, {35.515}, {32.1333},
          {-15.9682}, {-26.7058}, {-20.2046}, {-13.7168}, {34.8482}, {30.6872}, {13.114},
          {-20.228}, {0.365125}, {-18.0477}, {0.218339}, {18.0984}, {0.231125}, {18.2125},
          {51.2452}, {89.0913}, {0.521536}, {-27.5471}, {23.4333}, {35.5658}, {1.02415},
          {0.505591}, {52.3168}, {55.1828}, {0.274698}, {6.34563}, {-13.6421}, {0.381969}}}}
```

- **Swaps**

```
       obj2waygeosynergyeligpop[dataarray2way, 1.] // Timing

       {0., 0.932827}
```

```
Timing[ans = pairwiseMSE[obj2waygeosynergyeligpop, dataarray2way,
   { bgeodist}, nMaximizeOptions → {Method → {"DifferentialEvolution",
       "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]] } } ] ]
```

```
{10.1585, {0.943998, {bgeodist → 0.322014}}}
```

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

```
{2.558331, {0.943998, {bgeodist → 0.322104}}}

{2.569005, {0.943998, {bgeodist → 0.32192}}}

{2.569981, {0.943998, {bgeodist → 0.321843}}}

{2.157988, {0.943998, {bgeodist → 0.321963}}}

{2.571210, {0.943998, {bgeodist → 0.321937}}}

{2.987450, {0.943998, {bgeodist → 0.322056}}}

{2.570528, {0.943998, {bgeodist → 0.322114}}}

{2.154265, {0.943998, {bgeodist → 0.322036}}}

{3.806318, {0.943998, {bgeodist → 0.321892}}}

{2.978154, {0.943998, {bgeodist → 0.32186}}}

{3.399081, {0.943998, {bgeodist → 0.321869}}}

{2.152165, {0.943998, {bgeodist → 0.321873}}}

{2.561960, {0.943998, {bgeodist → 0.321782}}}

{3.796738, {0.943998, {bgeodist → 0.32188}}}

{2.974202, {0.943998, {bgeodist → 0.322111}}}
```

- **Negative**

```
Timing[ans = pairwiseMSE[obj2waygeosynergyeligpopneg, dataarray2way,
   { bgeodist}, nMaximizeOptions → {Method → {"DifferentialEvolution",
       "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]] } } ] ]
```

```
{14.6648, {0.92322, {bgeodist → 124.332}}}
```

```
Timing[ans = pairwiseMSE[obj2waygeosynergyeligpopneg, dataarray2way,
   { bgeodist}, nMaximizeOptions → {Method → {"DifferentialEvolution",
       "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]] } } ] ]
```

```
{13.154, {0.92322, {bgeodist → 127.547}}}
```

- **Standard errors for swaps**

```
estimateway4 = { bgeodist} /. {bgeodist → 0.3220728498703993`}
```

```
{0.322073}
```

```
Timing[pointcrout1 = pointIdentifiedCR[25, 150, estimateway4,
   obj2waygeosynergyeligpop, { bgeodist}, datamap2way, dataarray2way,
   asymptotics → coalitions, nMaximizeOptions → {Method → {"DifferentialEvolution",
       "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]] } }] ]
```

```
{666.924, {{{0.314027, 0.504398}},
  {{0.0269416}, {-0.00128945}, {-0.000149422}, {-0.175999}, {0.0269229}, {-0.0000412553},
   {-1.25025}, {-0.00100525}, {0.0252591}, {-0.00153488}, {-0.000900639}, {0.0271093},
   {-0.000143911}, {0.026629}, {-1.25092}, {-1.37652}, {-0.153879}, {-0.000268112},
   {0.0267332}, {-0.000675162}, {-0.0000874521}, {0.0268121}, {-0.000507951}, {-1.37709},
   {0.026974}, {0.0395726}, {-0.000852902}, {-1.25344}, {0.0269048}, {0.0269765},
   {0.0584843}, {0.0268089}, {-0.000808232}, {0.0274393}, {0.000478635}, {0.0307678},
   {-0.0324092}, {0.0301995}, {0.000661262}, {0.000665856}, {-0.000928457}, {0.0269698},
   {-0.00185234}, {0.00539048}, {0.0000586381}, {-0.0011119}, {0.0607454}, {-0.000213212},
   {-0.00630365}, {0.0592355}, {0.0669052}, {-0.736505}, {-0.0163951}, {-1.2513},
   {0.0168758}, {6.24657 × 10^-6}, {0.0000257256}, {-0.82235}, {-1.25297}, {-0.000834551},
   {0.000571041}, {-1.25408}, {-0.00163986}, {0.00825374}, {-0.00659319}, {0.0636734},
   {-0.0299335}, {-1.25406}, {-0.00718956}, {-0.00176233}, {0.0270128}, {-0.000144257},
   {-0.00405876}, {0.0266965}, {-0.000241683}, {-1.25307}, {0.00982927}, {-0.0146222},
   {-0.0264683}, {0.0270121}, {-1.37717}, {-0.00161588}, {-0.0250279}, {-0.000865611},
   {-1.37969}, {-0.00277399}, {0.00980535}, {0.0267534}, {0.0000264717}, {-0.00107973},
   {-0.000369257}, {0.000247059}, {-0.00098141}, {-0.00667465}, {0.009838}, {0.00591188},
   {-0.00152768}, {0.0293721}, {0.0291915}, {-0.000683496}, {0.0597891}, {-0.179424},
   {0.000571041}, {0.0396373}, {-1.25408}, {0.0268148}, {-0.00111595}, {-0.000943383},
   {-0.00104606}, {-0.000281225}, {-0.73545}, {0.0266995}, {0.000121859}, {0.000228163},
   {-0.810342}, {-0.00126608}, {0.0396105}, {0.000368948}, {-0.00187602}, {-0.000358822},
   {0.0266945}, {0.0268954}, {-0.0310586}, {-1.24914}, {-0.0258664}, {-0.001054},
   {0.0268477}, {0.00041193}, {0.000296681}, {-0.00128639}, {-0.173824}, {0.0625004},
   {-0.000163592}, {0.00589541}, {0.00614272}, {0.00213491}, {-0.0281673}, {0.000228163},
   {-0.00106175}, {-0.261348}, {0.0268442}, {-0.00628075}, {-0.00130541}, {0.0270587},
   {0.000621654}, {0.0266466}, {0.0112826}, {-0.194976}, {-1.30416}, {-0.00119991}}}}
```

Add air and ATS travel

Objective function

```
(L2) In[591]:= 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[592]:= 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

```
(L2) In[593]:= obj2wayallsynergyeligpop[dataarray2add, 1., 1., 1.] // Timing
```

```
(L2) Out[593]=
      {0.001229, 0.917558}
```

```
(L2) In[594]:= Timing[ans = pairwiseMSE[obj2wayallsynergyeligpop, dataarray2add,
                  { bgeodist, bair, bats}, nMaximizeOptions → {Method → {"DifferentialEvolution",
                       "SearchPoints" → 75, "RandomSeed" → Floor[SessionTime[]] } } ] ]

(L2) Out[594]=
      {8.45766, {0.952686, {bgeodist → 9.864, bair → -0.366799, bats → -0.115278}}}

(L2) In[595]:= Do[Print[Timing[
                  ans = pairwiseMSE[obj2wayallsynergyeligpop, dataarray2add, { bgeodist, bair, bats},
                    nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 75,
                         "RandomSeed" → Floor[SessionTime[]] } } ] ] ]
                ,
                {k,
                 1,
                 15}]

{9.02681, {0.952686, {bgeodist → 9.13134, bair → -0.393677, bats → -0.0937886}}}

{9.09761, {0.952686, {bgeodist → 9.87676, bair → -0.390991, bats → -0.0975758}}}

{9.0765, {0.952686, {bgeodist → 9.23692, bair → -0.375022, bats → -0.107169}}}

{9.051, {0.952748, {bgeodist → 9.90761, bair → -0.382079, bats → -0.100962}}}

{9.03023, {0.952748, {bgeodist → 9.18189, bair → -0.380876, bats → -0.0998839}}}

{9.01829, {0.952748, {bgeodist → 9.89759, bair → -0.379682, bats → -0.10328}}}

{9.02665, {0.952686, {bgeodist → 9.1547, bair → -0.396214, bats → -0.0916104}}}

{9.03295, {0.952748, {bgeodist → 9.8672, bair → -0.379199, bats → -0.102588}}}

{7.42499, {0.952686, {bgeodist → 9.07828, bair → -0.371996, bats → -0.107247}}}

{9.02159, {0.952748, {bgeodist → 9.86253, bair → -0.378006, bats → -0.104568}}}

{9.02531, {0.952748, {bgeodist → 9.89679, bair → -0.380403, bats → -0.101694}}}

{9.01541, {0.952748, {bgeodist → 9.12546, bair → -0.383774, bats → -0.0982174}}}

{9.00826, {0.952686, {bgeodist → 9.89081, bair → -0.386911, bats → -0.0989992}}}

{5.81291, {0.952686, {bgeodist → 9.87955, bair → -0.356457, bats → -0.121145}}}

{9.0515, {0.952748, {bgeodist → 9.13982, bair → -0.382585, bats → -0.0999755}}}

(L2) In[596]:= Do[Print[Timing[
                  ans = pairwiseMSE[obj2wayallsynergyeligpop, dataarray2add, { bgeodist, bair, bats},
                    nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 150,
                         "RandomSeed" → Floor[SessionTime[]] } } ] ] ]
                ,
                {k,
                 1,
                 15}]
```

```
{17.4436, {0.952748, {bgeodist → 9.13127, bair → -0.380294, bats → -0.101437}}}

{17.5335, {0.952748, {bgeodist → 9.10615, bair → -0.380697, bats → -0.100568}}}

{17.3872, {0.952748, {bgeodist → 9.14821, bair → -0.385614, bats → -0.0971027}}}

{17.3309, {0.952686, {bgeodist → 9.156, bair → -0.383124, bats → -0.101252}}}

{17.2922, {0.952748, {bgeodist → 9.18707, bair → -0.38319, bats → -0.0993436}}}

{17.308, {0.952748, {bgeodist → 9.18128, bair → -0.377368, bats → -0.102654}}}

{17.295, {0.952748, {bgeodist → 9.22979, bair → -0.378461, bats → -0.104372}}}

{17.4119, {0.952748, {bgeodist → 9.11725, bair → -0.382597, bats → -0.099581}}}

{17.3586, {0.952686, {bgeodist → 9.90419, bair → -0.3884, bats → -0.0987902}}}

{17.652, {0.952686, {bgeodist → 9.88258, bair → -0.387802, bats → -0.0960775}}}

{17.3567, {0.952686, {bgeodist → 9.86345, bair → -0.383875, bats → -0.0999717}}}

{15.7334, {0.952748, {bgeodist → 9.14649, bair → -0.380502, bats → -0.101141}}}

{17.3619, {0.952748, {bgeodist → 9.22879, bair → -0.382185, bats → -0.10103}}}

{17.3505, {0.952686, {bgeodist → 10.17, bair → -0.398799, bats → -0.0944816}}}

{17.3628, {0.952748, {bgeodist → 9.1635, bair → -0.381087, bats → -0.101686}}}
```

```
(L2) In[597]:= Do[Print[Timing[
                   ans = pairwiseMSE[obj2wayallsynergyeligpop, dataarray2add, { bgeodist, bair, bats},
                     nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 500,
                          "RandomSeed" → Floor[SessionTime[]] } } ] ] ]
                 ,
                 {k,
                  1,
                  15}]
```

```
{57.3004, {0.952748, {bgeodist → 9.15092, bair → -0.384467, bats → -0.0977595}}}

{57.1094, {0.952748, {bgeodist → 9.85364, bair → -0.381694, bats → -0.100577}}}

{57.0643, {0.952748, {bgeodist → 9.89727, bair → -0.380469, bats → -0.101837}}}

{57.0653, {0.952748, {bgeodist → 9.90303, bair → -0.376104, bats → -0.10557}}}

{57.0076, {0.952748, {bgeodist → 9.89022, bair → -0.374442, bats → -0.106241}}}

{57.1689, {0.952748, {bgeodist → 9.15076, bair → -0.3777, bats → -0.101938}}}

{57.1086, {0.952748, {bgeodist → 9.20909, bair → -0.379048, bats → -0.102003}}}

{57.0286, {0.952748, {bgeodist → 9.89165, bair → -0.380699, bats → -0.0999774}}}

{57.051, {0.952748, {bgeodist → 9.22108, bair → -0.379819, bats → -0.10375}}}

{57.109, {0.952748, {bgeodist → 9.13693, bair → -0.377347, bats → -0.10224}}}

{57.1531, {0.952748, {bgeodist → 9.85883, bair → -0.385246, bats → -0.0969779}}}

{57.3524, {0.952748, {bgeodist → 9.1033, bair → -0.38499, bats → -0.096351}}}

{57.0783, {0.952748, {bgeodist → 9.15929, bair → -0.378198, bats → -0.10306}}}

{57.2206, {0.952748, {bgeodist → 9.19036, bair → -0.379335, bats → -0.103676}}}

{57.1572, {0.952748, {bgeodist → 9.15286, bair → -0.380481, bats → -0.101729}}}
```

```
(L2) In[598]:= Do[Print[Timing[
                  ans = pairwiseMSE[obj2wayallsynergyeligpop, dataarray2add, { bgeodist, bair, bats},
                    nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 750,
                         "RandomSeed" → Floor[SessionTime[]] } } ] ] ]
                ,
                {k,
                 1,
                 15}]
{86.5027, {0.952748, {bgeodist → 9.89381, bair → -0.375891, bats → -0.105561}}}

{87.4181, {0.952748, {bgeodist → 9.09478, bair → -0.383595, bats → -0.0979943}}}

{86.6391, {0.952748, {bgeodist → 9.1297, bair → -0.378593, bats → -0.102206}}}

{86.5277, {0.952748, {bgeodist → 9.89795, bair → -0.384169, bats → -0.0987147}}}

{86.8746, {0.952748, {bgeodist → 9.12296, bair → -0.382156, bats → -0.0995853}}}

{86.5342, {0.952748, {bgeodist → 9.08034, bair → -0.379128, bats → -0.100938}}}

{86.6178, {0.952748, {bgeodist → 9.88088, bair → -0.376453, bats → -0.10514}}}

{86.6125, {0.952748, {bgeodist → 9.86363, bair → -0.384148, bats → -0.0981613}}}

{87.0133, {0.952748, {bgeodist → 9.89527, bair → -0.378178, bats → -0.102267}}}

{86.6441, {0.952748, {bgeodist → 9.13036, bair → -0.380357, bats → -0.101271}}}

{86.6593, {0.952748, {bgeodist → 9.8888, bair → -0.372242, bats → -0.106688}}}

{86.8908, {0.952748, {bgeodist → 9.1397, bair → -0.383375, bats → -0.0985428}}}

{86.8194, {0.952748, {bgeodist → 9.86525, bair → -0.380761, bats → -0.101838}}}

{86.6774, {0.952748, {bgeodist → 9.14514, bair → -0.382456, bats → -0.098412}}}

{86.695, {0.952748, {bgeodist → 9.2316, bair → -0.378456, bats → -0.103842}}}
```

- **Negative for adds**

```
(L2) In[599]:= Do[Print[Timing[
                  ans = pairwiseMSE[obj2wayallsynergyeligpopneg, dataarray2add, { bgeodist, bair, bats},
                    nMaximizeOptions → {Method → {"DifferentialEvolution", "SearchPoints" → 75,
                         "RandomSeed" → Floor[SessionTime[]] } } ] ] ]
                ,
                {k,
                 1,
                 2}]
{9.87527, {0.92993, {bgeodist → 371.069, bair → -31.221, bats → -1.12035}}}

{9.8422, {0.919672, {bgeodist → 90.4099, bair → -8.04329, bats → 2.75788}}}
```

- **Standard errors for adds**

```
(L2) In[600]:= estimateadd5 = { bgeodist, bair, bats} /. {bgeodist → 9.8, bair → -0.37, bats → -0.10}

(L2) Out[600]=
      {9.8, -0.37, -0.1}

(L2) In[601]:= obj2wayallsynergyeligpop[dataarray2add, Sequence @@ estimateadd5]

(L2) Out[601]=
      0.952313
```

```
(L2) In[603]:= Timing[pointcrout1 = pointIdentifiedCR[25, 200, estimateadd5,
                 obj2wayallsynergyeligpop, { bgeodist, bair, bats}, datamap2add, dataarray2add,
                 asymptotics → coalitions, nMaximizeOptions → {Method → {"DifferentialEvolution",
                     "SearchPoints" → 300, "RandomSeed" → Floor[SessionTime[]] } }] ]
```

```
(L2) Out[603]=
      {4133.56, {{{-11.8277, 14.0969}, {-0.497346, 1.22431}, {-0.393607, 0.0559471}},
        {{-38.5645, 0.250261, 0.10874}, {-4.46177, -0.774224, 0.928705},
         {111.209, 0.207427, -0.626924}, {3.3572, -0.161632, 0.0255902},
         {-42.5312, 2.60019, -0.590958}, {-5.24908, -0.333689, 0.463377},
         {-3.61081, -1.01602, 1.01559}, {1.91881, -0.0619079, -0.00810659},
         {205.548, -14.9906, 2.85882}, {2.64685, -0.116951, -0.00471323},
         {-22.0201, -0.902594, -1.03317}, {-3.05863, -0.0840817, -0.00493394},
         {-4.55365, -0.365003, 0.436978}, {-39.6154, 0.646792, -0.0405627},
         {-13.9403, -0.491924, 0.712782}, {-4.81221, 1.07722, -0.92034},
         {-35.3462, 0.252463, 0.804449}, {161.591, -12.7471, 0.859627},
         {-0.355465, -0.669676, 0.791311}, {149.224, -12.7409, 1.80406},
         {-3.64166, 0.690396, -0.634575}, {-25.2785, -0.366683, 0.136266},
         {-35.1319, 0.114792, 0.763947}, {2.9926, -0.169105, 0.0467545},
         {-42.3038, 1.92783, -1.0831}, {75.3899, -0.753398, 0.749752},
         {-3.88221, 0.256031, -0.26895}, {-30.6632, -0.486623, 0.210163},
         {236.248, -14.6988, 2.09954}, {80.9736, -9.66471, 2.70692},
         {-0.315504, -0.50355, 0.618922}, {171.446, -13.3801, 1.33999},
         {-4.1453, 0.32378, -0.262643}, {-5.20807, -1.45852, 1.11446},
         {0.424462, -0.0262858, -0.0234577}, {-4.5431, -0.722325, 0.870485},
         {0.377266, -0.739019, 0.830578}, {-13.6283, -0.49563, 0.721106},
         {-2.46414, -0.201985, 0.111844}, {-2.41826, 0.106526, 0.0270813},
         {-3.08011, -0.672413, 0.634011}, {-39.7126, 0.41861, -0.109066},
         {-2.60212, -0.0710245, 0.0122162}, {-3.19434, -0.0832708, 0.026224},
         {-1.29378, -1.91515, 1.73892}, {-0.440408, 0.0944649, 0.0989388},
         {-4.68231, -0.66824, 0.770705}, {-5.29225, -0.628452, 0.328379},
         {103.291, 0.524608, -0.65781}, {56.9829, -6.28665, -0.339871},
         {0.664106, -0.0402361, 0.0650402}, {1.05119, -0.0692341, 0.0215831},
         {249.256, -17.6089, 3.32437}, {4.08871, -0.115027, 0.0131171},
         {-0.0940555, -1.58706, 1.15669}, {-37.5329, 0.73068, -0.397043},
         {0.139528, -0.671096, 0.752613}, {-3.57569, -0.173365, 0.116225},
         {-17.4244, 0.064407, -0.0659282}, {-32.2184, -0.913765, 1.0486},
         {-32.7485, -0.407127, 0.922771}, {-28.7907, 0.49824, -0.013187},
         {0.329945, -0.0502078, -0.00560126}, {-28.8829, 0.573287, -0.101023},
         {-2.70211, 0.668838, -0.419779}, {-12.7791, 2.07149, -2.11401},
         {-3.32746, -0.0315472, -0.0214603}, {-3.4603, -0.941675, 0.957448},
         {-14.8121, -0.523904, 0.817792}, {18.2036, -3.5451, 0.709625},
         {2.84968, -0.28932, 0.155564}, {-4.05384, -0.894091, 0.90696},
         {1.14723, -2.96202, -0.767635}, {198.318, -14.9074, 2.1811},
         {-33.2668, 0.509932, -0.722468}, {1.12786, 0.65048, -0.561016},
         {-28.6365, -0.117412, -0.0484046}, {-28.7279, 0.537883, -0.0654957},
         {-0.640577, -0.679534, 0.780453}, {-25.1595, 0.733565, 0.794226},
         {-26.9976, -0.105238, -0.0462429}, {-30.7152, -1.42568, 1.16411},
         {2.83266, -0.0853031, -0.0482426}, {-3.98188, 0.285336, -0.260752},
         {-27.0105, -0.349649, 0.148568}, {-13.1033, -0.327241, 0.701696},
         {-33.2378, -0.629332, 0.419179}, {0.323133, -0.0234662, -0.0326995},
         {94.8194, -0.0591737, -0.078052}, {-3.55679, -0.958332, 0.912204},
{-24.219, -0.860164, 0.106194}, {158.261, -11.7976, -0.085737},
{3.38072, -0.11699, 0.396139}, {2.45468, -0.131044, 0.505082},
{81.7594, -6.19751, -1.60316}, {-13.111, 0.0743378, -0.0517254},
{-4.67777, 0.0636025, -0.105169}, {-2.03609, -0.3019, 0.169682},
{-6.04611, 0.268814, -0.0994842}, {248.282, -17.7182, 3.34092},
{-3.70564, 0.671435, -0.576989}, {94.2002, -2.39784, 2.23854},
{-0.794571, -1.30288, 1.1467}, {3.00519, 0.192045, -0.0149945},
{116.653, -9.32458, 0.284559}, {-13.3521, -0.279605, 0.395072},
{82.9366, -6.40252, -1.43776}, {0.318069, -0.0175569, -0.0378133},
{-14.4558, -0.422449, 0.762588}, {2.8028, -1.4734, 1.21854}, {-27.8677, 1.38673, -1.52069},
{-4.24403, -0.766632, 0.813338}, {1.31424, 0.0259241, -0.0872236},
{94.8734, -0.0604212, -0.375323}, {-41.8687, 0.334654, 0.17973},
{75.8052, -0.840966, -0.0319726}, {-1.14511, -0.0224999, 0.243206},
{182.631, -11.7566, 1.33698}, {74.1989, -0.391161, -0.0180997}, {199.397, -13.623, 1.2128},
{-28.6384, 0.202067, 0.669025}, {-28.9151, 0.468221, -0.0539303},
{-12.3785, -0.578815, 0.783273}, {-38.5112, 0.690511, -0.362636},
{0.281529, 0.0428322, -0.0921445}, {0.37486, 0.0828738, -0.119368},
{80.9505, -7.63512, 1.79778}, {8.91183, -0.00487011, -0.0975692},
{0.375113, -0.225966, 0.113762}, {22.5466, -2.61089, -0.340926},
{75.8149, -0.850544, 0.697593}, {95.9481, 0.17642, -0.346709},
{4.72869, -0.0589878, -0.0185273}, {8.23246, -1.63548, 1.37111},
{218.188, -16.8668, 3.05015}, {-23.2817, -1.88453, -0.0879775},
{2.96331, -0.140069, 0.043786}, {1.9568, 0.0727819, -0.129132},
{-38.3122, 1.04992, -0.587795}, {0.347109, -0.0267596, -0.0238521},
{-15.4577, 0.223173, 0.0172991}, {-0.0635923, -0.658058, 0.788554},
{-12.7478, -0.517148, 0.801244}, {-22.6836, -3.39085, 1.79209},
{89.5063, -7.48312, 0.20083}, {109.768, 0.0181491, -0.0368305},
{64.9578, -9.48011, 3.00624}, {-4.44765, 0.308494, -0.258852},
{1.34426, 0.0787273, -0.111595}, {-0.956606, -0.250935, 0.484891},
{111.215, 0.319499, -0.73645}, {-40.9186, 0.533642, -0.169389},
{-1.13308, -0.435862, 0.702743}, {83.0079, -6.48696, -1.50264},
{-13.7179, -0.493511, 0.716171}, {0.531931, -1.96346, 1.8962},
{115.808, -9.86717, 0.396532}, {-3.70154, 0.312005, -0.285485},
{25.9761, -5.39941, -0.642883}, {-30.4537, -0.341298, 0.148916},
{19.6925, -0.936662, 0.774469}, {94.7449, -9.98988, 1.58897},
{-31.4355, 0.230963, 0.00432181}, {-4.20623, 0.962834, -0.788187},
{2.7824, -0.035449, -0.0455234}, {70.8336, -0.422889, -0.00420879},
{-28.4093, 0.357838, -0.0250843}, {-5.01903, -2.08318, 1.73704},
{-27.4413, -0.109964, -0.0940297}, {-2.61069, -1.79059, -0.163595},
{0.210485, 0.0491711, -0.082213}, {110.077, 0.566546, -0.791682},
{72.0735, 0.390789, -0.752072}, {104.523, 2.13985, -2.24322},
{-28.8591, 0.546849, -0.0749482}, {-27.4433, -0.181101, -0.0479014},
{-0.110319, -0.134076, 0.448856}, {-5.05914, -0.361592, 0.676963},
{-3.54193, -0.954733, 0.919917}, {-27.6285, -0.410566, 0.160653},
{-3.1424, 1.10536, -0.881671}, {-28.6638, 0.513361, -0.0459834},
{-27.3724, -0.533178, 0.264961}, {13.6907, 0.154391, -0.280532},
{-25.4793, 0.305181, -0.0988742}, {-32.3811, -0.322298, 0.687798},
{-2.35975, 0.782144, -0.667761}, {-27.1683, -1.37699, 1.17443},
{-4.14152, -0.204178, 0.244828}, {-28.6133, 0.275249, 0.0963842},
{-28.2561, -0.17363, -0.0157258}, {-28.1997, -0.360348, 0.13123},
{-3.21508, 0.0115828, 0.0574461}, {-3.69804, -0.613036, 0.62199},
        {14.4727, -0.901556, 0.814436}, {-28.8104, -0.1857, -0.0278412},
        {4.67884, -0.119615, 0.0000830236}, {-3.35219, 1.17408, -0.979551},
        {0.691935, -0.075312, 0.0105887}, {-27.5642, -0.164974, -0.024552}}}}
```

- **Swaps**

```
obj2wayallsynergyeligpop[dataarray2way, 1., 1., 1.] // Timing

{0.002999, 0.92955}

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

```
{22.1606, {0.944966, {bgeodist → 0.3168, bair → -0.161918, bats → 0.0489038}}}

{18.1822, {0.944966, {bgeodist → 0.317371, bair → -0.166833, bats → 0.05145}}}

{14.1608, {0.944817, {bgeodist → 0.329296, bair → -0.201522, bats → -0.37133}}}

{22.1366, {0.944966, {bgeodist → 0.317038, bair → -0.16136, bats → 0.0335476}}}

{22.1426, {0.944891, {bgeodist → 0.330167, bair → -0.201205, bats → -0.382481}}}

{22.1256, {0.944966, {bgeodist → 0.315749, bair → -0.166035, bats → 0.0299664}}}

{14.1688, {0.944891, {bgeodist → 0.317974, bair → -0.155177, bats → 0.0503006}}}

{22.1316, {0.944966, {bgeodist → 0.317326, bair → -0.166997, bats → 0.0591007}}}

{22.1266, {0.944891, {bgeodist → 0.328766, bair → -0.204247, bats → -0.370445}}}

{18.1312, {0.944966, {bgeodist → 0.317407, bair → -0.161279, bats → 0.0552975}}}

{22.1256, {0.944966, {bgeodist → 0.31591, bair → -0.165569, bats → 0.0333433}}}

{22.1196, {0.944966, {bgeodist → 0.315219, bair → -0.166365, bats → 0.0156413}}}

{16.1405, {0.944966, {bgeodist → 0.317795, bair → -0.163214, bats → 0.0585167}}}

{16.1635, {0.944891, {bgeodist → 0.32961, bair → -0.206056, bats → -0.361158}}}

{16.1355, {0.944966, {bgeodist → 0.315706, bair → -0.162343, bats → 0.0191819}}}

{18.1302, {0.944891, {bgeodist → 0.317743, bair → -0.157323, bats → 0.0576402}}}

{22.1376, {0.944966, {bgeodist → 0.316945, bair → -0.161281, bats → 0.0323839}}}

{22.1296, {0.944966, {bgeodist → 0.31667, bair → -0.162082, bats → 0.0463248}}}

{22.1076, {0.944891, {bgeodist → 0.329621, bair → -0.203455, bats → -0.381848}}}

{22.1576, {0.944891, {bgeodist → 0.31682, bair → -0.163561, bats → 0.0575218}}}

{22.1236, {0.944966, {bgeodist → 0.318114, bair → -0.163222, bats → 0.063743}}}

{22.1236, {0.944966, {bgeodist → 0.316462, bair → -0.164363, bats → 0.032995}}}

{20.1479, {0.944891, {bgeodist → 0.31634, bair → -0.163642, bats → 0.0618777}}}

{22.1206, {0.944966, {bgeodist → 0.316564, bair → -0.160389, bats → 0.0284615}}}

{22.1176, {0.944891, {bgeodist → 0.316311, bair → -0.159326, bats → 0.0615839}}}
```

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

```
{140.689, {0.944966, {bgeodist → 0.31706, bair → -0.163837, bats → 0.0333937}}}

{140.802, {0.944966, {bgeodist → 0.316148, bair → -0.160686, bats → 0.0292544}}}

{141.513, {0.944966, {bgeodist → 0.316406, bair → -0.165265, bats → 0.0248852}}}

{141.481, {0.944966, {bgeodist → 0.317078, bair → -0.163644, bats → 0.0493698}}}

{141.393, {0.944966, {bgeodist → 0.317296, bair → -0.164933, bats → 0.0367157}}}
```

Old estimates

- **Negative coefficient on elig*pop**

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

```
{46.7569, {0.924486, {bgeodist → 99.4193, bair → 0.667778, bats → 7.37048}}}

{46.6779, {0.924486, {bgeodist → 101.93, bair → 0.596787, bats → 7.06099}}}

{46.6759, {0.924486, {bgeodist → 102.296, bair → 0.560428, bats → 7.62884}}}
```

- **Standard errors for swaps**

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

estimateway3 = { bgeodist, bair, bats} /. {bgeodist → 0.32, bair → -0.16, bats → 0.03`}
```

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

```
obj2wayallsynergyeligpop[dataarray2way, bgeodist, bair, bats ] /.
 {bgeodist → 0.32, bair → -0.16, bats → 0.03}
```

```
0.944593
```

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

```
{2191.73, {{{0.303432, 0.466289}, {-0.365694, 0.340656}, {-0.0774637, 0.403099}},
  {{-0.0143851, -0.0260926, 0.0507394}, {-0.0979073, 0.545609, -3.04046},
   {-0.254723, -3.1798, -1.80616}, {-0.23621, -3.27847, -1.73163},
   {-0.213623, -3.25788, -2.1844}, {0.125447, -0.035819, -2.74903},
   {-0.21074, -3.27488, -2.20387}, {0.0569357, 1.27101, -2.67548},
   {-0.0200928, -0.029862, 0.0230784}, {0.116356, -0.030782, -2.70786},
   {-0.230859, -3.1847, -2.16291}, {-0.184491, -3.18852, -0.356978},
   {-1.10956, -0.0892266, -1.11503}, {-1.05635, -1.2896, -1.20825},
   {0.019031, 0.215819, -1.23476}, {-0.166765, -3.16664, -0.636617},
   {0.0280764, -3.34789, -0.935031}, {0.0718212, -0.186628, 4.39122},
   {0.0406717, 0.221574, -1.6396}, {-0.0217163, -0.00981228, 0.183738},
   {-0.623709, -1.55401, -1.31451}, {-0.0601439, 0.713411, -2.80773},
   {-1.10222, -0.459513, -0.983768}, {-0.0087112, 0.263432, -0.325842},
   {-1.06316, -1.28875, -1.22291}, {0.192103, -3.77987, -0.00842442},
   {-1.10445, -0.474273, -1.04754}, {-0.0728046, 0.585979, -2.89059},
   {0.00245283, -3.56774, -0.254017}, {0.0602622, -1.86183, -2.11508},
   {-1.04591, -0.645317, -1.09851}, {-0.0375214, -3.35216, 3.45333},
   {-0.312881, -3.97088, -1.60181}, {-0.02818, 0.208951, 0.0827606},
   {0.0142748, 1.26431, -1.72056}, {0.0412126, 0.393627, -2.26757},
   {-0.0168679, -0.00169699, -0.0736652}, {-0.00713527, -1.64485, -0.335457},
   {-1.10219, -0.633986, -0.968398}, {-0.0240251, -0.0354337, -0.180472},
   {-0.061176, -0.0336877, -1.21706}, {0.0511448, -0.221572, -2.01118},
   {0.0340405, 0.329482, -2.14004}, {0.124969, -0.0182417, -2.66578},
   {-0.00208596, 0.254987, -0.0771606}, {-0.019791, -0.0645608, 0.0283785},
   {0.11774, -0.0374626, -2.72401}, {0.0454628, -0.233462, -2.05653},
   {0.0230807, 0.271398, 0.627536}, {-0.0154407, 0.0290575, 0.0261498},
   {-0.0174191, -0.00400265, 0.00297059}, {-0.00956166, -0.023769, 0.106648},
   {-1.07721, -0.625932, -1.1903}, {-1.0756, -1.13498, -1.21214},
   {-0.0191529, -0.0276746, 0.00128996}, {0.039977, 0.400229, -2.0819},
   {-0.0200888, 0.00184392, -0.0699818}, {0.0404252, 1.63565, -1.045},
   {0.0358801, 1.56001, -1.89308}, {-0.0177835, -0.015205, 0.0262388},
   {-0.626854, -1.54797, -1.3925}, {0.0149631, 1.25761, -1.03695},
   {-0.45835, -1.903, -2.00687}, {0.0242, 1.60802, -0.700031},
   {0.123397, -0.010457, -2.66961}, {0.0619366, -2.62263, 0.0355482},
   {0.0660343, 0.444692, -2.53347}, {0.0355572, -1.89868, -1.5188},
   {-0.00592475, -1.66863, 0.811333}, {-0.247656, -3.08661, -2.09532},
   {-1.10098, -0.48635, -0.981967}, {-1.08107, -0.812319, -1.19541},
   {-0.0230946, -0.0503814, -0.0711048}, {0.034677, -3.34247, -0.538866},
   {0.0397303, 0.224887, -1.85902}, {-0.0211899, -0.0245689, 0.0709653},
   {-0.072024, 0.26222, -2.01563}, {-0.00984882, -0.0208842, 0.0979354},
   {-0.236334, -3.16015, -2.13888}, {-0.17769, -2.72406, -0.823903},
   {-0.216955, -3.19371, -2.181}, {0.0396268, -3.48514, 0.801163},
   {0.00334404, -3.57749, -0.296246}, {-0.0182884, 0.00727273, 0.000851872},
   {0.118197, -0.0353414, -2.68499}, {-0.00523927, 0.232129, -0.0338957},
   {-0.00824486, -0.0164908, 0.135404}, {0.0661002, 0.00891291, -2.53234},
   {-0.0210288, -0.0476004, 0.0197388}, {0.119999, -0.0397776, -2.68704},
   {-0.00797069, -0.000640959, 0.0892941}, {0.0261557, 0.25312, -1.06074},
   {-1.10004, -0.513314, -0.979812}, {-0.0585201, 0.758818, -2.81683},
   {0.127081, -0.0208219, -2.68544}, {-0.00870565, 0.0363538, 0.0526475},
   {-0.0134755, -0.0304942, 0.112251}, {-0.0224531, -0.0256235, -0.0760692},
   {0.0718258, -3.49465, -0.466838}, {0.122294, -0.0361568, -2.67257},
{-1.09785, -0.406992, -1.12361}, {-0.207786, 0.150786, -3.4841},
{0.0708265, -0.0199702, -2.66345}, {-1.10441, -0.492687, -1.05695},
{0.10345, -0.541962, -2.23516}, {-1.07119, -1.30427, -1.00714},
{0.06484, -3.36326, 0.919572}, {-0.0129309, -0.0369158, 0.142747},
{0.0486793, -0.224133, -1.91476}, {-0.0105504, 0.267055, -0.459376},
{0.0328403, 0.277399, -2.1116}, {-0.0634924, -0.0198868, -0.416504},
{-0.0167241, -0.00662128, 0.0367297}, {-1.15657, 0.340212, -0.99669},
{0.00183486, -3.93847, -0.0586487}, {-0.157069, 0.247544, -2.64762},
{-1.10024, -0.471082, -1.07465}, {-0.180867, -1.49081, -2.08518},
{-1.10084, -0.476679, -1.02718}, {-1.10205, -0.494069, -1.05466},
{-0.015097, -0.00555175, 0.00874783}, {-0.015082, -0.0321247, 0.044823},
{-0.0152336, 0.00989636, 0.0691611}, {0.117613, -0.0158831, -2.83625},
{-1.36465, 1.32136, -0.0895931}, {0.0131114, 0.19608, 0.39541},
{-0.027396, -0.643331, 0.514332}, {-0.328195, -3.45299, -1.53754},
{-1.1035, -0.472043, -0.964893}, {0.0179695, -3.01105, 5.1756},
{0.0608797, -3.35425, 0.504953}, {-1.05988, -1.25919, -1.26488},
{0.0197939, -3.33659, -0.986184}, {0.125085, -0.00707688, -2.6727},
{0.125084, -0.0267333, -2.64879}, {-0.0577104, 0.797129, -2.81107},
{-0.221561, -2.67488, -1.93513}, {-0.168757, -3.17807, -0.570714},
{-1.08034, -0.823904, -1.19913}, {-0.0125728, 0.00721396, 0.0797373},
{-1.36264, 1.35393, 0.238381}, {-0.0151718, 0.000725426, 0.0377666},
{-1.08684, -0.580782, -1.06757}, {-0.17535, 0.217645, -2.39293},
{-0.0133067, -0.0222213, 0.0513528}, {-0.0223436, -0.00183722, -0.103435},
{-0.0140006, -0.0281738, 0.118753}, {0.13267, -0.00339898, -2.74985},
{-1.32643, 1.6106, -0.558954}, {-0.0126937, 0.269751, -0.360716},
{-0.0135591, -0.0154642, 0.048376}, {-0.0119634, -0.0264314, 0.0974352},
{-0.0630983, 1.20248, -2.88291}, {-0.0120513, -0.0201062, 0.0721952},
{-0.316455, -2.84843, -1.78886}, {0.0139726, 1.25171, -1.70582},
{-0.215337, -3.21847, -2.18591}, {0.129191, -0.0172388, -2.77085},
{-0.209481, -2.59397, -1.9587}, {-1.05842, -1.2572, -1.23697},
{-1.06273, -1.28808, -1.22144}, {-0.050732, 0.530723, -1.98007},
{-0.14375, 0.273707, -2.64449}, {0.0270658, -3.32474, -0.914978},
{-1.10418, -0.736549, -1.02551}, {-0.200444, -3.12936, -2.1783},
{-1.10265, -0.508761, -1.06034}, {-0.0728755, 0.261009, -1.98556},
{-0.0143958, -0.0236431, 0.062602}, {0.01402, -3.84151, -0.148598},
{0.121374, -0.025754, -2.76716}, {0.0272562, 0.233075, 0.703028},
{0.068417, 0.0337878, -2.11795}, {0.111724, 0.13507, 5.89626},
{0.0246751, -1.95176, 0.108169}, {-0.156736, -2.33949, -1.22484},
{-0.655497, -6.50072, -2.09952}, {0.0348229, 1.5779, -1.92401},
{-0.0143152, 0.245779, -0.471191}, {-0.011449, -0.0259394, 0.145517},
{-0.31186, -2.43166, -1.77546}, {-0.696005, -5.62327, -2.02704},
{-0.0112117, -0.0313118, 0.0974814}, {0.105562, -0.267376, -2.59064},
{0.120789, -0.00767858, -2.73188}, {-0.0167688, 0.00300991, 0.0617272},
{-1.10257, -0.458054, -0.994387}, {0.0323301, 1.55295, -1.95453},
{-1.04327, -1.28617, -1.29215}, {0.0565587, -0.202227, -2.10936},
{-0.012607, -0.0236932, 0.20223}, {-0.168562, 0.221489, -2.54641},
{0.0671051, -0.286472, -2.36756}, {-1.10307, -0.491993, -0.96111},
{-0.0161708, -0.00570229, -0.038934}, {-0.0186094, 0.00418446, -0.0107561},
{-1.07477, -1.14617, -0.969512}, {0.0380913, -3.49026, 0.753639},
{-0.224855, -2.59115, -1.72545}, {-0.0204568, 0.321977, -1.12296}}}}
```

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[]] } }] ]
```

```
{42 002.9,
 {{{-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}}}}
```