Competing on Speed - Calibration Section 8

December 2017

- **Instructions**
  To compute the regular outcome
  1) Run model functions
  2) Calibrate exogenous parameters
  3) Compute implicit parameters
  4) Compute equilibrium values

Segmented Duopoly Equilibrium

Model Functions for Oligopoly

```
ClearAll["Global`*"]
 σ[i_]:=r/a  (q[i]-q[i-1])/(s[i]-s[i-1]) ;
demand[i_] := Ne ( G[σ[i+1]]-G[σ[i]]) / (2a);
accessfee[i_] := (a / r Sum[σ[j] (s[j] - s[j - 1]), {j, 1, i}]);
profit[i_]:=accessfee[i] demand[i] -cost[s[i]];
welfare[i_]:=Ne s[i]/(2*r) Integrate[x*G'[x], {x, σ[i], σ[i+1]}]-cost[s[i]];
foc[i_] := D[profit[i], q[i]];
ρ[s_] := (γ + r) s / (1 - s);
cost[s_] := (γ + r) c s / (1 - s);
d[ρ1_, ρ2_, γ_] := ρ2 / ((ρ1 (γ+ρ2+r)) / (γ+ρ1+r));
(* NORMALIZED *)
n = 2.;
σ[3] = σbar;
σ[0] = 0;
q[0] = 0;
a = 1 / 2.;

s[0] = 0;
G[x_] := x / σbar;
g[x_] := G'[x];


fees = Solve[{foc[1] == 0, foc[2] == 0}, {q[1], q[2]}] // Simplify;
redprofit[1] = FullSimplify[profit[1] /. fees[[1]]];
redprofit[2] = FullSimplify[profit[2] /. fees[[1]]];
focFirst[i_] := D[redprofit[i], s[i]];
r = 0.0375 / 252.;
μ = 2.75 / 252;
σbar = μ / 2.;
NN = 28 225.;


fees=Solve[{foc[1]==0,foc[2]==0},{q[1],q[2]}]//Simplify;
redprofit[1]=FullSimplify[profit[1]/.fees[[1]]];
redprofit[2]=FullSimplify[profit[2]/.fees[[1]]];
focFirst[i_]:=D[redprofit[i],s[i]];
```

```
fees=Solve[{foc[1]==0,foc[2]==0},{q[1],q[2]}]//Simplify

focFirst[1]//Simplify
```

- **NORMALIZED AND CALIBRATED PARAMETERS**

Define Number of Venues, Boundaries, Parameters and Distribution

n is the maximum number of venues

```
(* NORMALIZED
n=2.;
σ[3]=σbar;
σ[0]=0;
q[0]=0;
s[0]=0;
G[x_]:=x/σbar;
g[x_]:=G'[x];
a=1/2.;*)

r=0.0375/252;
μ=2.75/252;
σbar=μ/2.;
NN=28225; (* Global number of traders *)
```

- **Check restriction on abar**

```
InverseCDF[UniformDistribution[{0,σbar}],1-2a]
```

- **Check closed-form sigmas**

```
fees=Solve[{foc[1]==0,foc[2]==0},{q[1],q[2]}]//Simplify;

redprofit[1]=FullSimplify[profit[1]/.fees[[1]]];
redprofit[2]=FullSimplify[profit[2]/.fees[[1]]];
focFirst[i_]:=D[redprofit[i],s[i]];


P1=redprofit[1]/.fees/.{c→0,Ne→1,s[2]→dd s[1]}//FullSimplify
P2=redprofit[2]/.fees/.{c→0,Ne→1,s[2]→dd s[1]}//FullSimplify

focFirst[1]/.{s[1]→S1,s[2]→S2}//FullSimplify
focFirst[2]/.{s[1]→S1,s[2]→S2}//FullSimplify
```

Calibrated and Model Implied Parameters

- **Speeds and Gammas**

Check the relationship rho/gamma for different markets

```
Print["ρ2e/γlow γmed γ high"]
Print["Bonds",{ρB2e/γBa,ρB2e/γBb,ρB2e/γBc}]
Print["Equity",{ρE2e/γEa,ρE2e/γEb,ρE2e/γEc}]
Print["Futures",{ρF2e/γFa,ρF2e/γFb,ρF2e/γFc}]
```

■ Asset: CORPORATE BONDS

```
a=1/2.;
ρB1e=1; (*half a day*)
ρB2e=39.; (*10 minutes  *)
ρ1e=ρB1e; ρ2e=ρB2e;
V=1.97; (*95% TRACE sample for 2013Q4*) (*The 95% percentile number is 1.97*)

KB=21723.; (* Number of bonds =21723*)
SB=5; (* # bonds per investor *)
(*Ne=NN/KB SB;*)
Ne=13. ;
```

■ Asset: EQUITIES

```
a=1/2.;
ρE1e=390/2.;(* one per minute *)
ρE2e=23400.; (* one per second *)
ρ1e=ρE1e; ρ2e=ρE2e;
V=3023.4;

KE=2805.;
SE=5;

Ne=92. ;


115*3/4.
```

```
86.25
```

■ Asset: FUTURES

```
a=9/20.;
ρFme=234000/2.; (* 100ms delay *)
ρme=ρFme;
V=1030204; (*May 3-5 2010 CME E-Mini from Kyle et al 2014: 446340*)

KF=1.; (* Number of bonds =21723*)
SF=1.; (* # bonds per investor *)
Ne=NN 3/4. ;
```

Compute implicit N and c, or Implicit γ and c

```
(* ONE venue: For futures *)
γ=8V ρme/(Ne ρme-8 V)

Rationalize[γ,0.01];

(* Speed: one venue. Rationalize ρm and recover c*)
CC=NSolve[ρme/(ρme+r+γ)==1-(8 r (r+γ) ce /(Ne σbar))^.5,ce]; (* From monopoly Uniform *)
c=ce/.CC[[1]]
```

```
(* TWO venues *)

σ1[ρ1_,ρ2_,γ_,σbar_]:=σbar (d[ρ1,ρ2,γ]-1)/(4 d[ρ1,ρ2,γ]-1);
σ2[ρ1_,ρ2_,γ_,σbar_]:=σbar (2 d[ρ1,ρ2,γ]-1)/(4 d[ρ1,ρ2,γ]-1);

(* w1 and w2 Give the fraction of investors per venue. G(σ_i+1)-G(σ_i)  *)

w1[ρ1_,ρ2_,γ_,σbar_]:=(σ2[ρ1,ρ2,γ,σbar]-σ1[ρ1,ρ2,γ,σbar])/σbar;
w2[ρ1_,ρ2_,γ_,σbar_]:=(σbar-σ2[ρ1,ρ2,γ,σbar])/σbar;
(*
Sn[ρ1_,ρ2_,γ_,σbar_]:=NSolve[V==NN/KE  S γ a/4 (ρ1 w1[ρ1,ρ2,γ,σbar]/(γ+ρ1)+ρ2  w2[ρ1,ρ2,γ,σbar]/(γ+ρ2)) && S>0,
Se=S/.First[Sn[ρ1e,ρ2e,γ,σbar]]
*)
γn[ρ1_,ρ2_,Ne_,σbar_]:=NSolve[V==Ne γγ 1/4 (ρ1 w1[ρ1,ρ2,γγ,σbar]/(γγ+ρ1)+ρ2  w2[ρ1,ρ2,γγ,σba
γ=γγ/.First[γn[ρ1e,ρ2e,Ne,σbar]]
Rationalize[γ,0.01]

(*Bonds and Equities: Calibrate c with Two venues *)

(*First venue First order condition *)
S1=ρ1e/(ρ1e+γ+r); S2=ρ2e/(ρ2e+γ+r);

C1=NSolve[{-c (r+γ)/(-1+S1)^2+Ne (7 S1-4 S2) S2^2 σbar/(2 r (S1-4 S2)^3)==0 && c>0},{c},Reals];
c1=c/.First[C1];
C2=NSolve[-c (r+γ)/(-1+S2)^2-2 Ne S2 (2 S1^2-3 S1 S2+4 S2^2) σbar/(r (S1-4 S2)^3)==0 && c>0,{c},Reals];
c2=c/.First[C2];

c=c2
(* Bonds Nk a=.45*)
c=0.036201289983601025;
γ=0.8336289761821931;
Ne=13;
a=9/20.;
speedsB={{s[1]→0.555875963870688,s[2]→0.9791379008865045}};
speeds=speedsB;

(* Stocks Nk, a=.45 *)
c=0.00015697073782393323`;
γ=182.95192671329988`;
Ne=92;
a=9/20.;

speedsE={{s[1]→0.5665438517003876`,s[2]→0.9923582719825069`}}
(* Futures Nk a=.45 *)
c=0.0027502712663807724`;
γ=390.62998446250475`;
Ne=28225.*.75;
a=9/20.;
speedsF={{s[1]→0.5695823104670836`,s[2]→0.996919148832049`}};
```

Equilibrium Computation

```
speeds=NSolve[{focFirst[1]==0 , focFirst[2]==0, 1>=s[2]>s[1]>=0},{s[1],s[2]},Reals,6]
s[1]=s[1]/.speeds;
s[2]=s[2]/.speeds;
ρ1=ρ[s[1]]
ρ2=ρ[s[2]]
sm=1- Sqrt[ 8 c  r (r+γ)/(σbar Ne)];
ρm=ρ[sm]
π1=(profit[1]/.fees/.speeds)[[1]][[1]][[1]][[1]]
π2=(profit[2]/.fees/.speeds)[[1]][[1]][[1]][[1]]
```

- **Case where Venue 2 Reacts to minimum speed**

```
speeds=NSolve[{focFirst[1]==0 , focFirst[2]==0, 1>=s[2]>s[1]>=0},{s[1],s[2]},Reals,10]
ρ1=ρ[s[1]]/.speeds;
ρ2=ρ[s[2]]/.speeds;
ρ1=1.5 ρ1(* minimum speed *)
s[1]=ρ1/(γ+r+ρ1);
speeds2=NSolve[{focFirst[2]==0, 1>=s[2]>s[1]},{s[2]},Reals]
s[2]=s[2]/.First[speeds2];
ρ2=ρ[s[2]];
(*profit[1]/.fees/.speeds2
profit[2]/.fees/.speeds2*)
π1=(profit[1]/.fees)[[1]][[1]][[1]]
π2=(profit[2]/.fees)[[1]][[1]][[1]]
```

Assign Values

```
q[1]=q[1]/.fees/.speeds;
q[2]=q[2]/.fees/.speeds;
(*s[1]=s[1]/.speeds;
s[2]=s[2]/.speeds;
ρ1=ρ[s[1]];

ρ2=ρ[s[2]];*)
σ1=r/a q[1]/s[1];
σ2=r/a (q[2]-q[1])/(s[2]-s[1]);

(*Print["q1 =",q[1]/.fees/.speeds];
Print["q2 =",q[2]/.fees/.speeds];*)
Print["σ1 normalized=",σ1/σbar];
Print["σ2 mornalized=",σ2/σbar];
Print["ρ1 =",ρ1/.speeds];
Print["ρ2 =",ρ2/.speeds];
Print["s1 =",s[1]/.speeds];
Print["s2 =",s[2]/.speeds];
Print["d=",s[2]/s[1]];
```

- ***Planner***

```
W1[q_, s_] :=Ne (s *Integrate[x*g[x], {x, (q*r)/(a* s), σbar}])/(2*r) - cost[s];


SP1=NMaximize[{W1[q, s],(Ne q*(1 - G[q r /(a s)]))/ (2*a) ≥ cost[s]  && 1≥ s ≥0 && q≥ 0}, ·
qP=q/.Last[SP1]; sP=s/.Last[SP1]; σP=qP r /(a sP); ρP=ρ[sP];
```

Duopoly Outcomes

The notation xij means: value of x for venue i out of j venues

```
participationD=(demand[1]+demand[2]) ;


welfareD=welfare[1]+welfare[2];
welfare12=welfare[1];
welfare22=welfare[2];
participation12=demand[1] ;
participation22=demand[2] ;
participationDA=Ne(1-σ1/σbar);
participation12A=Ne (G[σ2]-G[σ1]) ;
participation22A=Ne (1-G[σ2]) ;

volume12 = Ne (σ2-σ1)/σbar (γ*ρ[s[1]]) 1/(4*(γ + ρ[s[1]]));
volume22 = Ne *(1-σ2/σbar)(γ*ρ[s[2]]) 1/(4*(γ + ρ[s[2]]));

volumeD =volume12+volume22;

(* Walrasian*)
σW=zn/.FindRoot[ 1/(2*a)-1 - G[zn] /(1-G[zn]) ==0, {zn,σbar/2}];
priceW=μ/r+σW/r; (* This is for a<.5 *)
volumeW=a/2  γ Ne;
participationW=Ne; (*2a Ne;(1-G[σW])Ne;*)
participationWA=2a Ne; (*Active traders only *)
welfareW= Ne Integrate[x*G'[x], {x, σW,σbar}]/(2*r) ;

(* Planner, one venue *)
welfareP = SP1[[1]];
volumeP=ρP/4  γ/(γ + ρP) (1-G[σP])Ne;
participationP=(1-G[σP]) Ne;
participationAP=2a (1-G[σP]) Ne; (*Active traders only *)

(*Consolidated*)
σC=zn/.FindRoot[ 1 - G[zn] -zn* G'[zn] ==  0, {zn,σbar/2}];
(*sC=1-(8 r c /(Ne σbar))^.5;*)
sC=1- Sqrt[ 8 c  r (r+γ)/(σbar Ne)];
qC=a/r σC  sC;
profitsC=qC Ne ((1-G[σC])/(2 a))-cost[sC];
ρC=sC/(1-sC) (γ+r);
participationC=(1-G[σC]) Ne;
priceC=μ/r+σC/r  (r+γ sC)/(r+γ );

volumeC = ρC/4  γ/(γ + ρC) (1-G[σC])Ne;


welfareC=Ne (sC*Integrate[x*G'[x], {x,σC,σbar}])/(2*r) -cost[sC] ;

(*Normalized*)
(* Consolidated *)
PC=participationC/participationW 100;
VC=volumeC/volumeW 100;
WC=welfareC/welfareW 100;

PCP=participationC/participationP 100;
VCP=volumeC/volumeP 100;
WCP=welfareC/welfareP 100;

(*Duopoly. Simple notation: Walrasian as benchmark. 'P' at the end: Planner as benchmark *)
PD=participationD/participationW 100;
PDA=participationDA/participationWA 100;
```

```
VD=volumeD/volumeW 100;
WD=welfareD/welfareW 100;

PDP=participationD/participationP 100;
PDAP=participationDA/participationAP 100;
VDP=volumeD/volumeP 100;
WDP=welfareD/welfareP 100;

(* Venue 1*)
P12=participation12/participationW 100;
P12A=participation12A/participationWA 100;
V12=volume12/volumeW 100;
W12=welfare[1] /welfareW 100;

P12P=participation12/participationP 100;
P12AP=participation12A/participationAP 100;
V12P=volume12/volumeP 100;
W12P=welfare[1] /welfareP 100;

(* Venue 2*)
P22=participation22/participationW 100;
P22A=participation22A/participationWA 100;
V22=volume22/volumeW 100;
W22=welfare[2]/welfareW 100;

P22P=participation22/participationP 100;
P22AP=participation22A/participationAP 100;
V22P=volume22/volumeP 100;
W22P=welfare[2]/welfareP 100;


profits1=π1 * 1/profitsC 100;
profits2=π2 * 1/profitsC 100;


SFP=Abs[ (welfareP-welfareW)/welfareW 100]

(* Adjusted profits for market intervention tables: Corporate bonds *)
πCBase=56.93138869076862;
πC=profitsC/πCBase 100;
profits1=π1 * 1/πCBase 100;
profits2=π2 * 1/πCBase 100;

(* Adjusted profits for market intervention tables: Equities *)
πCBase=414.7356506287216;
πC=profitsC/πCBase 100;
profits1=π1 * 1/πCBase 100;
profits2=π2 * 1/πCBase 100;

(* Adjusted profits for market intervention tables: Futures *)
πCBase=96378.7996848356;
πC=profitsC/πCBase 100;
profits1=π1 * 1/πCBase 100;
profits2=π2 * 1/πCBase 100;

(* Adjusted profits for comparison with DGP: Corporate bonds *)
πCBase=21.449;
πC=profitsC/πCBase 100;
profits1=π1 * 1/πCBase 100;
profits2=π2 * 1/πCBase 100;
```

Summary: *Walrasian* benchmark

```
Print["** CASE ** :
Equities"]
Print["** Parameters **:
{a,σbar,γ}=",{a,σbar,γ}]
Print["Implicit Parameters: c, N investors",{c,Ne}]
Print["** OUTCOMES **"]
Print["Participation=",NumberForm[{PC,P12,P22,PD},4]];
Print["ParticipationA=",NumberForm[{PC,P12A,P22A,PDA},4]];
Print["Volume=",NumberForm[{VC,V12,V22,VD},4]];
Print["Welfare=",NumberForm[{WC,W12,W22,WD},4]];
Print["Profits=",NumberForm[{profitsC,profits1,profits2,profits1+profits2},4]];
```

Summary: Planner one venue is benchmark

```
Print["** CASE ** :
Equities"]
Print["** Parameters **:
{a,σbar,γ}=",{a,σbar,γ}]
Print["Implicit Parameters: c, N investors",{c,Ne}]
Print["** OUTCOMES **"]
Print["Participation=",NumberForm[{PCP,P12P,P22P,PDP},4]];
Print["ParticipationA=",NumberForm[{PCP,P12AP,P22AP,PDAP},4]];
Print["Volume=",NumberForm[{VCP,V12P,V22P,VDP},4]];
Print["Welfare=",NumberForm[{WCP,W12P,W22P,WDP},4]];
Print["Profits=",NumberForm[{πC,profits1,profits2,profits1+profits2},4]];
```

- **Table misallocation vs. participation losses**

```
TLP=100(1-welfareP/welfareW)

MLP=100(Ne Integrate[x g[x],{x,σP,σbar}](1-sP)/(2*r))/welfareW
PLP=100(Ne  Integrate[x g[x],{x,σW,σP}]/(2*r))/welfareW
CLP=100 (cost[sP]/welfareW)

TLC=(1-welfareC/welfareP) 100
MLC=100(Ne Integrate[x g[x],{x,σC,σbar}](sP-sC)/(2*r))/welfareP
PLC=100(Ne sP Integrate[x g[x],{x,σP,σC}]/(2*r))/welfareP
CLC=100(cost[sC]-cost[sP])/welfareP

s1=First[s[1]]; s2=First[s[2]];
TLD=100(1-welfareD/welfareP)
CLD=100(cost[s1]+cost[s2]-cost[sP])/welfareP
MLD=Ne 100/welfareP * (( Integrate[x g[x],{x,σ2,σbar}](sP-s2)/(2*r))+ Integrate[x g[x],{x,σ
PLD=100(Ne sP Integrate[x g[x],{x,σP,σ1}]/(2*r))/welfareP
```

```
Print["** CASE ** :
Equities"]
Print["** Parameters **:
{a,σbar,γ}=",{a,σbar,γ}]
Print["Implicit Parameters: c, N investors",{c,Ne}]
Print["** OUTCOMES=Misallocation loss, participation loss, total fraction of welfare loss

Print["Planner=",NumberForm[{PLP,MLP,CLP,TLP},3]];
Print["Monopoly=",NumberForm[{PLC,MLC,CLC,TLC},3]];
Print["Duopoly=",NumberForm[{PLD,MLD,CLD,TLD},3]];
```

Integrated Equilibrium

```
ClearAll["Global`*"]
σ[1]=r/a q[1]/s[1];
σ[2]=2r (q[2]-q[1]/(2a) z)/(s[2]-s[1]);
z=1-s[2]/s[1] ((γ s[1]+r)/(γ s[2]+r)) (1-2a);
demand[1]=Ne 1/(2a) (1-2 a+2a G[σ[2]]-G[σ[1]]);
demand[2]=Ne (1-G[σ[2]]);
(*fee1= 1/(2r) s[1] σ[1];*)
fee2=1/(2r)  ((s[2] - s[1])*σ[2] + s[1]*σ[1] *z);
profit[1] =q[1]  * demand[1] - cost[s[1]];
profit[2] = q[2] * demand[2] - cost[s[2]];
welfare[i_]:=Ne s[i]/(2*r) Integrate[x*G'[x], {x, σ[i], σ[i+1]}]-cost[s[i]];

ρ[s_]:=(γ+r) s/(1-s);
cost[s_]:=(γ+r) c s/(1-s);

(* NORMALIZED *)
n=2;
σ[3]=σbar;
σ[0]=0;
q[0]=0;
s[0]=0;
G[x_]:=x/σbar;
g[x_]:=G'[x];

r=0.0375/252;
μ=2.75/252;
σbar=μ/2.;
```

■ Second Stage

```
fees=Solve[{D[profit[1],q[1]]==0,D[profit[2],q[2]]==0},{q[1],q[2]}] //Simplify;
redprofit[1]=(profit[1]/.fees//Simplify)[[1]];
redprofit[2]=(profit[2]/.fees//Simplify)[[1]];

focFirst[1]=D[redprofit[1],s[1]]//Simplify;
focFirst[2]=D[redprofit[2],s[2]]//Simplify;
```

```
(* Bonds Nk a=.45*)
c=0.036201289983601025`;
γ=0.8336289761821931`;
Ne=13;
a=9/20.;
speedsB={{s[1]→0.555875963870688`,s[2]→0.9791379008865045`}};
speeds=speedsB;
profitsC=56.93138869076861;

(* Stocks Nk, a=.45 *)
c=0.00015697073782393323`;
γ=182.95192671329988`;
Ne=92;
a=9/20.;

speedsE={{s[1]→0.5665438517003876`,s[2]→0.9923582719825069`}};
speeds=speedsE;
profitsC=414.7356506287216;
(* Futures Nk a=.45 *)
c=0.0027502712663807724`;
γ=390.62998446250475`;
Ne=28225.*.75;
a=9/20.;
speedsF={{s[1]→0.5695823104670836`,s[2]→0.996919148832049`}};
speeds=speedsF;
profitsC=96378.799684835;
```

- **Planner**

```
W1[q_, s_] :=Ne (s *Integrate[x*g[x], {x, (q*r)/(a* s), σbar}])/(2*r) - cost[s];

SP1=NMaximize[{W1[q, s],(Ne q*(1 - G[q r /(a s)]))/ (2*a) ≥ cost[s]  && 1≥ s ≥0 && q≥ 0}, ·
qP=q/.Last[SP1]; sP=s/.Last[SP1]; σP=qP r /(a sP); ρP=ρ[sP];
```

First Stage Speeds

```
(*speeds=NSolve[{focFirst[1]==0 , focFirst[2]==0,1>=s[2]>s[1]>=0},{s[1],s[2]},Reals]


s[1]=s[1]/.speeds;
s[2]=s[2]/.speeds;
ρ1=ρ[s[1]]
ρ2=ρ[s[2]]*)
```

Assign Values

```
q[1]=q[1]/.fees/.speeds;
q[2]=q[2]/.fees/.speeds;
s[1]=s[1]/.speeds;
s[2]=s[2]/.speeds;
ρ1=ρ[s[1]]/.speeds;
ρ2=ρ[s[2]]/.speeds;
σ1=σ[1]/.fees/.speeds;
σ2=σ[2]/.fees/.speeds;

(*Print["q1 =",q[1]/.fees/.speeds];
Print["q2 =",q[2]/.fees/.speeds];*)
Print["σ1 normalized=",σ[1]/σbar/.fees/.speeds];
Print["σ2 mornalized=",σ[2]/σbar/.fees/.speeds];
Print["ρ1 =",ρ1];
Print["ρ2 =",ρ2];
Print["s1 =",s[1]/.speeds];
Print["s2 =",s[2]/.speeds];
Print["d=",s[2]/s[1]/.speeds];

π1=(profit[1]/.fees/.speeds)[[1]][[1]][[1]][[1]];
π2=(profit[2]/.fees/.speeds)[[1]][[1]][[1]][[1]];


(*Check restriction on abar*)
a=.45;
σ11=First[σ[1][[1]]];
Boole[σ11>InverseCDF[UniformDistribution[{0,σbar}],1-2a]]
(* Same as σ1>σW *)
```

- **Welfare adjustments**

```
σ2t=fee2/s[2] r/a
σ2t s[2]
σ1 s[1]
W2temp=(.5-a)(1-G[σ2])s[2]σ2t/r
W1temp=(.5-a)(1-G[σ2])s[1]σ1/r
welfare11=welfare[1]+W1temp;
welfare22=welfare[2]-W2temp;
welfareD=welfare11+welfare22
welfareDold=welfare[1]+welfare[2]
```

Outcomes

The notation xij means: value of x for venue i out of j venues

```
participationD=(demand[1]+demand[2]) ;
participationDA=Ne (1-G[σ1]) ;
participation12=demand[1] ;
participation22=demand[2] ;
participation12A=Ne (G[σ2]-G[σ1]) ;
participation22A=Ne (1-G[σ2]) ;
volume12 = Ne (σ2-σ1)/σbar (γ*ρ[s[1]]) 1/(4*(γ + ρ[s[1]]));
volume22 = Ne *(1-σ2/σbar)(γ*ρ[s[2]]) 1/(4*(γ + ρ[s[2]]));
volumeD =volume12+volume22;

σ2t=fee2/s[2] r/a;
W1temp=(.5-a)(1-G[σ2])s[1]σ1/r;
W2temp=(.5-a)(1-G[σ2])s[2]σ2t/r;
```

```
welfare11=welfare[1]+W1temp;
welfare22=welfare[2]-W2temp;
welfareD=welfare11+welfare22;


(* Walrasian*)
σW=zn/.FindRoot[ 1/(2*a)-1 - G[zn] /(1-G[zn]) ==0, {zn,σbar/2}];
priceW=μ/r+σW/r; (* This is for a<.5 *)
volumeW=a/2  γ Ne;
participationW=Ne;(*(1-G[σW])Ne;*)
participationWA=2a Ne;(*(1-G[σW])Ne;*)
welfareW= Ne Integrate[x*G'[x], {x, σW,σbar}]/(2*r) ;

(* Planner, one venue *)
welfareP = SP1[[1]];
volumeP=ρP/4  γ/(γ + ρP) (1-G[σP])Ne;
participationP=(1-G[σP]) Ne;
participationAP=2a (1-G[σP]) Ne; (*Active traders only *)

(*Duopoly*)
PD=participationD/participationW 100;
PDA=participationDA/participationWA 100;
VD=volumeD/volumeW 100;
WD=welfareD/welfareW 100;

PDP=participationD/participationP 100;
PDAP=participationDA/participationAP 100;
VDP=volumeD/volumeP 100;
WDP=welfareD/welfareP 100;


(* Venue 1*)
P12=participation12/participationW 100;
P12A=participation12A/participationWA 100;
V12=volume12/volumeW 100;
W12=welfare[1] /welfareW 100;

P12P=participation12/participationP 100;
P12AP=participation12A/participationAP 100;
V12P=volume12/volumeP 100;
W12P=welfare[1] /welfareP 100;

(* Venue 2*)
P22=participation22/participationW 100;
P22A=participation22A/participationWA 100;
V22=volume22/volumeW 100;
W22=welfare[2]/welfareW 100;

P22P=participation22/participationP 100;
P22AP=participation22A/participationAP 100;
V22P=volume22/volumeP 100;
W22P=welfare[2]/welfareP 100;


profits1=π1 * 1/profitsC 100;
profits2=π2 * 1/profitsC 100;
```

Summary Walrasian

```
Print["** CASE ** :
Equities"]
Print["** Parameters **:
{a,σbar,γ}=",{a,σbar,γ}]
Print["Implicit Parameters: c, N investors",{c,Ne}]
Print["** OUTCOMES **"]
Print["Participation=",NumberForm[{P12,P22,PD},4]];
Print["ParticipationA=",NumberForm[{P12A,P22A,PDA},4]];
Print["Volume=",NumberForm[{V12,V22,VD},4]];
Print["Welfare=",NumberForm[{W12,W22,WD},4]];
Print["Profits=",NumberForm[{profitsC,profits1,profits2,profits1+profits2},4]];
```

```
participationW
participationD
participation12
participation22
volume12
volume22
```

Summary: Planner one venue is benchmark

```
Print["** CASE ** :
Equities"]
Print["** Parameters **:
{a,σbar,γ}=",{a,σbar,γ}]
Print["Implicit Parameters: c, N investors",{c,Ne}]
Print["** OUTCOMES **"]
Print["Participation=",NumberForm[{PCP,P12P,P22P,PDP},4]];
Print["ParticipationA=",NumberForm[{PCP,P12AP,P22AP,PDAP},4]];
Print["Volume=",NumberForm[{VCP,V12P,V22P,VDP},4]];
Print["Welfare=",NumberForm[{WCP,W12P,W22P,WDP},4]];
Print["Profits=",NumberForm[{100,profits1,profits2,profits1+profits2},4]];
```

Three Venues Outcomes

- **Functions and Parameters**

```
ClearAll["Global`*"]
 σ[i_]:=r/a (q[i]-q[i-1])/(s[i]-s[i-1]) ;
demand[i_] := Ne ( G[σ[i+1]]-G[σ[i]]) / (2a);
accessfee[i_] := (a / r Sum[σ[j] (s[j] - s[j - 1]), {j, 1, i}]);
profit[i_]:=accessfee[i] demand[i]-cost[s[i]];
welfare[i_]:=Ne s[i]/(2*r) Integrate[x*G'[x], {x, σ[i], σ[i+1]}]-cost[s[i]];
foc[i_] := D[profit[i], q[i]];
ρ[s_] := (γ + r) s / (1 - s);
cost[s_] := (γ + r) c s / (1 - s);
d[ρ1_, ρ2_, γ_] := ρ2 / (ρ1 (γ+ρ2+r) / (γ+ρ1+r));

(* NORMALIZED *)
n = 3;
σ[4] = σbar;
σ[0] = 0;
q[0] = 0;
s[0] = 0;
G[x_] := x / σbar;
g[x_] := G'[x];
(*a=1/2;*)

(* d=s2/s1 *)
```

- **Second Stage**

```
fees=Solve[{foc[1]==0,foc[2]==0,foc[3]==0},{q[1],q[2],q[3]}] //FullSimplify

redprofit[i_]:=redprofit[i]=(profit[i]/.fees//Simplify)[[1]];

focFirst[i_]:=D[redprofit[i],s[i]];

σ[1]/.fees/.{σbar->1,s[1]->1/4,s[2]→1/2,s[3]→1}//FullSimplify
σ[2]/.fees//Simplify
σ[3]/.fees//Simplify

(* Check that we get reasonable numbers*)
σ[1]/.fees/.{s[1]→.2,s[2]→.5,s[3]→.95}

redprofit[1]/.fees/.{Ne→1}//FullSimplify
redprofit[2]/.fees/.{Ne→1}//FullSimplify
redprofit[3]/.fees/.{Ne→1}//FullSimplify
```

- **BR Functions**

```
b1[x_] :=First[(( NSolve[{focFirst[1]==0,s[2]>s[1]>0},s[1],Reals])/.{s[2]→x})];
b2[y_] :=First[(( NSolve[{focFirst[2]==0,1>s[2]>y},s[2],Reals])/.{s[1]→y})];

{0.0535714 σbar}
```

- **Parameters**

```
(* Bonds (a) *)
r=0.0375/252;
μ=2.75/252;
σbar=μ/2;
Ne=555;
c=0.005986573871300805;
γ=0.059772780209332764;
```

- **First stage, restrictions**

```
AbsoluteTiming[speeds2=NSolve[{focFirst[1]==0 && focFirst[2]==0 && focFirst[3]==0 },{s[1],s[2

focFirst[3]/.{s[1]→.15,s[2]→.5,s[3]→.9972}//Simplify
focFirst[2]//Simplify


(* Acceptable Solution with Parameters Equity (a) *)
s[1]=.41779987465023716;
s[2]=.67805807258;
s[3]=.9994389863646;

(* Acceptable parameters with Equity (c) *)
s[1]=0.41388897311191675;
s[2]=0.6730246573311033;
s[3]=0.993346265668684;
```

- **Check FOC first stage**

```
focFirst[1]/.{s[1]→ .41779987465023716,s[2]→ 0.67805807258,s[3]→ 0.9994389863646} (* Shoul
```

- **Equilibrium Profits**

```
redprofit[1]/.{s[1]→ .41779987,s[2]→ 0.678058,s[3]→ 0.999438986}
redprofit[1]/.{s[1]→ .42779987,s[2]→ 0.678058,s[3]→ 0.999438986} (* Check, a little faster
redprofit[1]/.{s[1]→ .40779987,s[2]→ 0.678058,s[3]→ 0.999438986} (* Check, a little slower

redprofit[2]/.{s[1]→ .41779987,s[2]→ 0.678058,s[3]→ 0.999438986}
redprofit[2]/.{s[1]→ .41779987,s[2]→ 0.688058,s[3]→ 0.999438986} (* Check, a little faster
redprofit[2]/.{s[1]→ .41779987,s[2]→ 0.668058,s[3]→ 0.999438986} (* Check, a little slower

redprofit[3]/.{s[1]→ .41779987,s[2]→ 0.678058,s[3]→ 0.999438986}
redprofit[3]/.{s[1]→ .41779987,s[2]→ 0.678058,s[3]→ 0.9999 98986} (* Check, a little faste
redprofit[3]/.{s[1]→ .41779987,s[2]→ 0.678058,s[3]→ 0.999008986} (* Check, a little slower
```

- **Welfare**

```
(welfare[1]+welfare[2]+welfare[3])/.fees
fees

welfare[1]/.fees
welfare[2]/.fees
(Ne s[3]/(2*r) Integrate[x*G'[x], {x,0.002165288944151114 , σbar}]-cost[s[3]])

Welfare3=170.87;
Welfare2=166.29;

(* With Stock a parameters *)
Welfare3-Welfare2


redprofit[1]/.fees

Solve[{foc[1]==0,foc[2]==0,foc[3]==0},{σ[1],σ[2],σ[3]}] //FullSimplify
```

■ *Shortcut: This gives the Best Response of two venues to the speed “x” of the first*

```
x=0.001;
speeds=NSolve[{ focFirst[2]==0/.{s[1]→ x} , focFirst[3]==0/.{s[1]→ x}},{s[2],s[3]},Reals]
(* fees/.speeds[[1]]
symEqu=NSolve[{D[redprofit[1],s[1]]==0,D[redprofit[2],s[2]]==0,
    D[redprofit[3],s[3]]==0},{s[1],s[2],s[3]}]*)
```

```
(Ne s[3]/(2*r) Integrate[x*G'[x], {x,0.002164471958110832 , σbar}]-cost[s[3]])
```

```
%/.σ[3]→0.0021644719581108
```

```
speeds={{s[3]→0.9993042108944853`,s[1]→-1.9072460384586214`,s[2]→1.008635562042798`},{s[3]
```