This code can be used to simulate the figures in Demary, Markus (2009), "Transaction Taxes and Traders with Heterogeneous Investment Horizons in an Agent-Based Financial Market Model".

All simulations run under the software R.

1 Program 1

Simulation in the time domain. Constant fundamental value.

T = 3000

```
s = array(dim = c(1,T))
sigc = array(dim = c(1,T))
sigf = array(dim = c(1,T))
shock1a = array(dim = c(1,T))
shock2a = array(dim = c(1,T))
shock1b = array(dim = c(1,T))
shock2b = array(dim = c(1,T))
shock3 = array(dim = c(1,T))
shock4 = array(dim = c(1,T))
ef = array(dim = c(1,T))
ec = array(dim = c(1,T))
pic = array(dim = c(1,T))
pif = array(dim = c(1,T))
w1c = array(dim = c(1,T))
w1f = array(dim = c(1,T))
efN = array(dim = c(1,T))
ecN = array(dim = c(1,T))
picN = array(dim = c(1,T))
pifN = array(dim = c(1,T))
w1cN = array(dim = c(1,T))
w1fN = array(dim = c(1,T))
time = array(dim = c(1,T))
mis = array(dim = c(1,T))
s1 = array(dim = c(1,T))
s2 = array(dim = c(1,T))
fu = array(dim = c(1,T))
ret = array(dim = c(1,T))


fu[1:T] = 0
pic[1:T] = 0
pif[1:T] = 0
picN[1:T] = 0
pifN[1:T] = 0
s1[1:T] = 0
```

s2[1:T] = 1/5

s[1:T] = 0
ec[1:T] = 0
ef[1:T] = 0
ecN[1:T] = 0
efN[1:T] = 0

phi = 0.04
beta = 0.04
gamma = 1000
N = 30
tax = 0.00

shock1a[1:T] = rnorm(T)

shock2a[1:T] = rnorm(T)

shock1b[1:T] = rnorm(T)

shock2b[1:T] = rnorm(T)

shock3[1:T] = rnorm(T)

shock4[1:T] = rnorm(T)

for (t in N+3:T)

time[t] = N+t-2

fu[t] = 0

ef[t] = phi * (fu[t] - s[t]) + 0.005 * shock1a[t]

ec[t] = beta * (s[t] - s[t-1]) + 0.04 * shock2a[t]

efN[t] = (1 - (1-phi)^N) $*$ ($fu[t] - s[t]$) + 0.00 $*$ $shock1b[t]$

ecN[t] = ((1-beta^N)/(1 $-$ $beta$)) $*$ $beta$ $*$ ($s[t] - s[t-1]$) + 0.00 $*$ $shock2b[t]$

pic[t] = (exp(s[t]) - exp(s[t-1])) * ec[t-2] + 0.975 * pic[t-1] - tax * (exp(s[t]) + exp(s[t-1])) * abs(ec[t-2])

pif[t] = (exp(s[t]) - exp(s[t-1])) * ef[t-2] + 0.975 * pif[t-1] - tax * (exp(s[t]) + exp(s[t-1])) * abs(ef[t-2])

picN[t] = (exp(s[t]) - exp(s[t-N])) / N * ecN[t-N-1] + 0.975 * picN[t-1] - tax * (exp(s[t]) + exp(s[t-N])) / N * abs(ecN[t-N-1])

pifN[t] = (exp(s[t]) - exp(s[t-N])) / N * efN[t-N-1] + 0.975 * pifN[t-1] - tax * (exp(s[t]) + exp(s[t-N])) / N * abs(efN[t-N-1])

wc = exp(gamma * (pic[t]))

wf = exp(gamma * (pif[t]))

wcN = exp(gamma * (picN[t]))

wfN = exp(gamma * (pifN[t]))

w1c[t] = wc / (wc + wf + wcN + wfN + 1)

w1f[t] = wf / (wc + wf + wcN + wfN + 1)

w1cN[t] = wcN / (wc + wf + wcN + wfN + 1)

w1fN[t] = wfN / (wc + wf + wcN + wfN + 1)

s[t+1] = s[t] + (w1c[t] * ec[t] + w1f[t] * ef[t] + w1cN[t] * ecN[t] + w1fN[t] * efN[t] + 0.01 * shock3[t])

mis[t] = s[t] - fu[t]

ret[t] = s[t] - s[t-1]

par(mfcol = c(3,3))

plot(time[10:T], s[10:T], ylab = ”Exchange Rate”, xlab = ”Trading Days”, main = ”Exchange Rate”, ylim=c(-2,2), type = ”l”, lwd = 1)

lines(time[10:T], s1[10:T])

plot(time[10:T], abs(mis[10:T]), ylab = ”Distortion”, xlab = ”Trading Days”, main = ”Distortion”, ylim=c(0,0.4), type = ”l”, lwd = 1)

lines(time[10:T], s1[10:T])

plot(time[10:T], ret[10:T], ylab = ”Return”, xlab = ”Trading Days”, main = ”Return”, type = ”l”, ylim=c(-0.08,0.08), lwd = 1)

lines(time[10:T], s1[10:T])

plot(time[10:T], w1c[10:T], ylab = ”Population”, xlab = ”Trading Days”, main = ”Short Term Chartists”, ylim=c(0,1), type = ”l”, lwd = 1)

lines(time[10:T], s2[10:T])

plot(time[10:T], w1cN[10:T], ylab = ”Population”, xlab = ”Trading Days”, main = ”Longer Term Chartists”, ylim=c(0,1), type = ”l”, lwd = 1)

lines(time[10:T], s2[10:T])

plot(time[10:T], w1f[10:T], ylab = ”Population”, xlab = ”Trading Days”, main = ”Short Term Fundamentalists”, ylim=c(0,1), type = ”l”, lwd = 1)

lines(time[10:T], s2[10:T])

plot(time[10:T], w1fN[10:T], ylab = ”Population”, xlab = ”Trading Days”, main = ”Longer Term Fundamentalists”, ylim=c(0,1), type = ”l”, lwd = 1)

lines(time[10:T], s2[10:T])

acf(ret[400:T], lag.max = 80, type=c("correlation"), plot = TRUE, main = "Autocorrelation of Raw Returns")

acf(abs(ret[400:T]), lag.max = 80, type=c("correlation"), plot = TRUE, main = "Autocorrelation of Absolute Returns")

2 Program 2

Simulation in the time domain. Fundamental value follows random walk-

T = 3000

s = array(dim = c(1,T))

sigc = array(dim = c(1,T))

sigf = array(dim = c(1,T))

shock1a = array(dim = c(1,T))

shock2a = array(dim = c(1,T))

shock1b = array(dim = c(1,T))

shock2b = array(dim = c(1,T))

shock3 = array(dim = c(1,T))

shock4 = array(dim = c(1,T))

ef = array(dim = c(1,T))

ec = array(dim = c(1,T))

pic = array(dim = c(1,T))

pif = array(dim = c(1,T))

w1c = array(dim = c(1,T))

w1f = array(dim = c(1,T))

efN = array(dim = c(1,T))

ecN = array(dim = c(1,T))

picN = array(dim = c(1,T))

pifN = array(dim = c(1,T))

w1cN = array(dim = c(1,T))

w1fN = array(dim = c(1,T))

time = array(dim = c(1,T))

mis = array(dim = c(1,T))

```
s1 = array(dim = c(1,T))
s2 = array(dim = c(1,T))
fu = array(dim = c(1,T))
ret = array(dim = c(1,T))
fu[1:T] = 0
pic[1:T] = 0
pif[1:T] = 0
picN[1:T] = 0
pifN[1:T] = 0
s1[1:T] = 0
s2[1:T] = 1/5
s[1:T] = 0
ec[1:T] = 0
ef[1:T] = 0
ecN[1:T] = 0
efN[1:T] = 0
phi = 0.04
beta = 0.04
gamma = 1000
N = 30
tax = 0.00
shock1a[1:T] = rnorm(T)
shock2a[1:T] = rnorm(T)
shock1b[1:T] = rnorm(T)
shock2b[1:T] = rnorm(T)
shock3[1:T] = rnorm(T)
shock4[1:T] = rnorm(T)
for (t in N+3:T)
time[t] = N+t-2
fu[t] = fu[t-1] + 0.01 * shock4[t]
ef[t] = phi * ( fu[t] - s[t]) + 0.005 * shock1a[t]
```

ec[t] = beta * (s[t] - s[t-1]) + 0.04 * shock2a[t]

efN[t] = (1 - (1-phi)^N) $*$ $(fu[t] - s[t])$ + 0.00 $*$ $shock1b[t]$

ecN[t] = ((1-beta^N)/$(1 - beta)$) $*$ $beta$ $*$ $(s[t] - s[t-1])$ + 0.00 $*$ $shock2b[t]$

pic[t] = (exp(s[t]) - exp(s[t-1])) * ec[t-2] + 0.975 * pic[t-1] - tax * (exp(s[t]) + exp(s[t-1])) * abs(ec[t-2])

pif[t] = (exp(s[t]) - exp(s[t-1])) * ef[t-2] + 0.975 * pif[t-1] - tax * (exp(s[t]) + exp(s[t-1])) * abs(ef[t-2])

picN[t] = (exp(s[t]) - exp(s[t-N])) / N * ecN[t-N-1] + 0.975 * picN[t-1] - tax * (exp(s[t]) + exp(s[t-N])) / N * abs(ecN[t-N-1])

pifN[t] = (exp(s[t]) - exp(s[t-N])) / N * efN[t-N-1] + 0.975 * pifN[t-1] - tax * (exp(s[t]) + exp(s[t-N])) / N * abs(efN[t-N-1])

wc = exp(gamma * (pic[t]))

wf = exp(gamma * (pif[t]))

wcN = exp(gamma * (picN[t]))

wfN = exp(gamma * (pifN[t]))

w1c[t] = wc / (wc + wf + wcN + wfN + 1)

w1f[t] = wf / (wc + wf + wcN + wfN + 1)

w1cN[t] = wcN / (wc + wf + wcN + wfN + 1)

w1fN[t] = wfN / (wc + wf + wcN + wfN + 1)

s[t+1] = s[t] + (w1c[t] * ec[t] + w1f[t] * ef[t] + w1cN[t] * ecN[t] + w1fN[t] * efN[t] + 0.01 * shock3[t])

mis[t] = s[t] - fu[t]

ret[t] = s[t] - s[t-1]

par(mfcol = c(3,3))

plot(time[10:T], s[10:T], ylab = ”Exchange Rate”, xlab = ”Trading Days”, main = ”Exchange Rate”, ylim=c(-2,2), type = ”l”, lwd = 1)

lines(time[10:T], s1[10:T])

plot(time[10:T], abs(mis[10:T]), ylab = ”Distortion”, xlab = ”Trading Days”, main = ”Distortion”, ylim=c(0,0.4), type = ”l”, lwd = 1)

lines(time[10:T], s1[10:T])

plot(time[10:T], ret[10:T], ylab = ”Return”, xlab = ”Trading Days”, main = ”Return”, type = ”l”, ylim=c(-0.08,0.08), lwd = 1)

lines(time[10:T], s1[10:T])

plot(time[10:T], w1c[10:T], ylab = ”Population”, xlab = ”Trading Days”, main = ”Short Term Chartists”, ylim=c(0,1), type = ”l”, lwd = 1)

lines(time[10:T], s2[10:T])

plot(time[10:T], w1cN[10:T], ylab = ”Population”, xlab = ”Trading Days”, main = ”Longer Term Chartists”, ylim=c(0,1), type = ”l”, lwd = 1)

lines(time[10:T], s2[10:T])

plot(time[10:T], w1f[10:T], ylab = ”Population”, xlab = ”Trading Days”, main = ”Short Term Fundamentalists”, ylim=c(0,1), type = ”l”, lwd = 1)

lines(time[10:T], s2[10:T])

plot(time[10:T], w1fN[10:T], ylab = ”Population”, xlab = ”Trading Days”, main = ”Longer Term Fundamentalists”, ylim=c(0,1), type = ”l”, lwd = 1)

lines(time[10:T], s2[10:T])

acf(ret[400:T], lag.max = 80, type=c(”correlation”), plot = TRUE, main = ”Autocorrelation of Raw Returns”)

acf(abs(ret[400:T]), lag.max = 80, type=c(”correlation”), plot = TRUE, main = ”Autocorrelation of Absolute Returns”)

3 Program 3

Simulation over 5000 trading days and 100 markets for 10 tax rates. Constant fundamental value.

G = 100

T = 5000

J = 10

s = array(dim = c(1,T))

sigc = array(dim = c(1,T))

sigf = array(dim = c(1,T))

shock1a = array(dim = c(1,T))

shock2a = array(dim = c(1,T))

shock1b = array(dim = c(1,T))

shock2b = array(dim = c(1,T))

shock3 = array(dim = c(1,T))

shock4 = array(dim = c(1,T))

ef = array(dim = c(1,T))

ec = array(dim = c(1,T))

pic = array(dim = c(1,T))

pif = array(dim = c(1,T))

w1c = array(dim = c(1,T))

w1f = array(dim = c(1,T))

efN = array(dim = c(1,T))

ecN = array(dim = c(1,T))

picN = array(dim = c(1,T))

pifN = array(dim = c(1,T))

w1cN = array(dim = c(1,T))

w1fN = array(dim = c(1,T))

time = array(dim = c(1,T))

mis = array(dim = c(1,T))

s1 = array(dim = c(1,T))

s2 = array(dim = c(1,T))

fu = array(dim = c(1,T))

ret = array(dim = c(1,T))

gret = array(dim = c(1,G))

gvol = array(dim = c(1,G))

gk = array(dim = c(1,G))

gwf = array(dim = c(1,G))

gwc = array(dim = c(1,G))

gwfN = array(dim = c(1,G))

gwcN = array(dim = c(1,G))

go = array(dim = c(1,G))

gdis = array(dim = c(1,G))

tax = array(dim = c(1,J))

ms = array(dim = c(1,J))

sds = array(dim = c(1,J))

sks = array(dim = c(1,J))

```
ks = array(dim = c(1,J))
mwf = array(dim = c(1,J))
mwc = array(dim = c(1,J))
mwfN = array(dim = c(1,J))
mwcN = array(dim = c(1,J))
mo = array(dim = c(1,J))
vol = array(dim = c(1,J))
dis = array(dim = c(1,J))
fu[1:T] = 0
pic[1:T] = 0
pif[1:T] = 0
picN[1:T] = 0
pifN[1:T] = 0
s1[1:T] = 0
s2[1:T] = 1/5
s[1:T] = 0
ec[1:T] = 0
ef[1:T] = 0
ecN[1:T] = 0
efN[1:T] = 0
phi = 0.04
beta = 0.04
gamma = 800
N = 30
for (j in 1:J)
tax[j] = 0.01 + (j-1) * 0.003
for (g in 1:G)
shock1a[1:T] = rnorm(T)
shock2a[1:T] = rnorm(T)
shock1b[1:T] = rnorm(T)
shock2b[1:T] = rnorm(T)
```

shock3[1:T] = rnorm(T)

shock4[1:T] = rnorm(T)

for (t in N+3:T)

time[t] = N+t-2

fu[t] = 0.000 * shock4[t]

ef[t] = phi * (fu[t] - s[t]) + 0.005 * shock1a[t]

ec[t] = beta * (s[t] - s[t-1]) + 0.03 * shock2a[t]

efN[t] = (1 - (1-phi)^N) $*$ $(fu[t] - s[t]) + 0.00 * shock1b[t]$

ecN[t] = ((1-beta^N)/$(1 - beta)) * beta * (s[t] - s[t - 1]) + 0.00 * shock2b[t]$

pic[t] = (exp(s[t]) - exp(s[t-1])) * ec[t-2] + 0.975 * pic[t-1] - tax[j] * (exp(s[t]) + exp(s[t-1])) * abs(ec[t-2])

pif[t] = (exp(s[t]) - exp(s[t-1])) * ef[t-2] + 0.975 * pif[t-1] - tax[j] * (exp(s[t]) + exp(s[t-1])) * abs(ef[t-2])

picN[t] = (exp(s[t]) - exp(s[t-N])) / N * ecN[t-N-1] + 0.975 * picN[t-1] - tax[j] * (exp(s[t]) + exp(s[t-N])) / N * abs(ecN[t-N-1])

pifN[t] = (exp(s[t]) - exp(s[t-N])) / N * efN[t-N-1] + 0.975 * pifN[t-1] - tax[j] * (exp(s[t]) + exp(s[t-N])) / N * abs(efN[t-N-1])

wc = exp(gamma * (pic[t]))

wf = exp(gamma * (pif[t]))

wcN = exp(gamma * (picN[t]))

wfN = exp(gamma * (pifN[t]))

w1c[t] = wc / (wc + wf + wcN + wfN + 1)

w1f[t] = wf / (wc + wf + wcN + wfN + 1)

w1cN[t] = wcN / (wc + wf + wcN + wfN + 1)

w1fN[t] = wfN / (wc + wf + wcN + wfN + 1)

s[t+1] = s[t] + (w1c[t] * ec[t] + w1f[t] * ef[t] + w1cN[t] * ecN[t] + w1fN[t] * efN[t] + 0.01 * shock3[t])

mis[t] = s[t] - fu[t]

ret[t] = s[t] - s[t-1]

gret[g] = mean(ret[40:T])

gvol[g] = sqrt(mean(ret[40:T]^2))

gk[g] = mean(ret[40:T]^4)$/gvol[g]^4$

```
gwf[g] = mean(w1f[40:T])

gwc[g] = mean(w1c[40:T])

gwfN[g] = mean(w1fN[40:T])

gwcN[g] = mean(w1cN[40:T])

go[g] = 1 - gwc[g] - gwf[g] - gwcN[g] - gwfN[g]

gdis[g] = mean(abs(mis[40:T]))

ms[j] = mean(gret[1:G])

sds[j] = mean(gvol[1:G])

ks[j] = mean(gk[1:G])

mwf[j] = mean(gwf[1:G])

mwc[j] = mean(gwc[1:G])

mwfN[j] = mean(gwfN[1:G])

mwcN[j] = mean(gwcN[1:G])

mo[j] = mean(go[1:G])

dis[j] = mean(gdis[1:G])

par(mfcol = c(3,3))

plot(tax[1:J], ms[1:J], ylab = "Mean", xlab = "Tax Rate", main = "Mean", ylim=c(-0.015,0.015), type = "o", lwd = 3)

plot(tax[1:J], sds[1:J], ylab = "Standard Deviation", xlab = "Tax Rate", main = "Standard Deviation", ylim=c(0.01,0.013), type = "o", lwd = 3)

plot(tax[1:J], ks[1:J], ylab = "Kurtosis", xlab = "Tax Rate", main = "Kurtosis", ylim=c(2.5,5), type = "o", lwd = 3)

plot(tax[1:J], mwf[1:J], ylab = "Short-Term Fundamentalists", xlab = "Tax Rate", main = "Short-Term Fundamentalists", ylim=c(0,0.2), type = "o", lwd = 3)

plot(tax[1:J], mwc[1:J], ylab = "Short-Term Chartists", xlab = "Tax Rate", main = "Short-Term Chartists", ylim=c(0,0.2), type = "o", lwd = 3)

plot(tax[1:J], mo[1:J], ylab = "Inactive Traders", xlab = "Tax Rate", main = "Inactive Traders", ylim=c(0.2,0.4), type = "o", lwd = 3)

plot(tax[1:J], mwfN[1:J], ylab = "Long-Term Fundamentalists", xlab = "Tax Rate", main = "Long-Term Fundamentalists", ylim=c(0.2,0.4), type = "o", lwd = 3)

plot(tax[1:J], mwcN[1:J], ylab = "Long-Term Chartists", xlab = "Tax Rate", main = "Long-Term Chartists", ylim=c(0.2,0.4), type = "o", lwd = 3)

plot(tax[1:J], dis[1:J], ylab = "Distortion", xlab = "Tax Rate", main = "Distortion", ylim=c(0.011,0.015), type = "o", lwd = 3)
```

4 Program 4

Simulation of 5000 trading days and 100 markets for 10 tax rates. Fundamental value follows random walk.

G = 100

T = 5000

J = 10

s = array(dim = c(1,T))

shock1a = array(dim = c(1,T))

shock2a = array(dim = c(1,T))

shock1b = array(dim = c(1,T))

shock2b = array(dim = c(1,T))

shock3 = array(dim = c(1,T))

shock4 = array(dim = c(1,T))

ef = array(dim = c(1,T))

ec = array(dim = c(1,T))

pic = array(dim = c(1,T))

pif = array(dim = c(1,T))

w1c = array(dim = c(1,T))

w1f = array(dim = c(1,T))

efN = array(dim = c(1,T))

ecN = array(dim = c(1,T))

picN = array(dim = c(1,T))

pifN = array(dim = c(1,T))

w1cN = array(dim = c(1,T))

w1fN = array(dim = c(1,T))

time = array(dim = c(1,T))

mis = array(dim = c(1,T))

s1 = array(dim = c(1,T))

s2 = array(dim = c(1,T))

fu = array(dim = c(1,T))

ret = array(dim = c(1,T))

gret = array(dim = c(1,G))

gvol = array(dim = c(1,G))

gk = array(dim = c(1,G))

gwf = array(dim = c(1,G))

gwc = array(dim = c(1,G))

gwfN = array(dim = c(1,G))

gwcN = array(dim = c(1,G))

go = array(dim = c(1,G))

gdis = array(dim = c(1,G))

tax = array(dim = c(1,J))

ms = array(dim = c(1,J))

sds = array(dim = c(1,J))

sks = array(dim = c(1,J))

ks = array(dim = c(1,J))

mwf = array(dim = c(1,J))

mwc = array(dim = c(1,J))

mwfN = array(dim = c(1,J))

mwcN = array(dim = c(1,J))

mo = array(dim = c(1,J))

vol = array(dim = c(1,J))

dis = array(dim = c(1,J))

fu[1:T] = 0

pic[1:T] = 0

pif[1:T] = 0

picN[1:T] = 0

pifN[1:T] = 0

s1[1:T] = 0

s2[1:T] = 1/5

s[1:T] = 0

ec[1:T] = 0

ef[1:T] = 0

ecN[1:T] = 0

efN[1:T] = 0

phi = 0.04

beta = 0.04

gamma = 800

N = 30

for (j in 1:J)

tax[j] = 0.02 + (j-1) * 0.0025

for (g in 1:G)

shock1a[1:T] = rnorm(T)

shock2a[1:T] = rnorm(T)

shock1b[1:T] = rnorm(T)

shock2b[1:T] = rnorm(T)

shock3[1:T] = rnorm(T)

shock4[1:T] = rnorm(T)

for (t in N+3:T)

time[t] = N+t-2

fu[t] = fu[t-1] + 0.01 * shock4[t]

ef[t] = phi * (fu[t] - s[t]) + 0.005 * shock1a[t]

ec[t] = beta * (s[t] - s[t-1]) + 0.03 * shock2a[t]

efN[t] = (1 - (1-phi)^N) $*$ ($fu[t] - s[t]$) + 0.00 $*$ $shock1b[t]$

ecN[t] = ((1-beta^N)/(1 $-$ $beta$)) $*$ $beta$ $*$ ($s[t] - s[t-1]$) + 0.00 $*$ $shock2b[t]$

pic[t] = (exp(s[t]) - exp(s[t-1])) * ec[t-2] + 0.975 * pic[t-1] - tax[j] * (exp(s[t]) + exp(s[t-1])) * abs(ec[t-2])

pif[t] = (exp(s[t]) - exp(s[t-1])) * ef[t-2] + 0.975 * pif[t-1] - tax[j] * (exp(s[t]) + exp(s[t-1])) * abs(ef[t-2])

picN[t] = (exp(s[t]) - exp(s[t-N])) / N * ecN[t-N-1] + 0.975 * picN[t-1] - tax[j] * (exp(s[t]) + exp(s[t-N])) / N * abs(ecN[t-N-1])

pifN[t] = (exp(s[t]) - exp(s[t-N])) / N * efN[t-N-1] + 0.975 * pifN[t-1] - tax[j] * (exp(s[t]) + exp(s[t-N])) / N * abs(efN[t-N-1])

wc = exp(gamma * (pic[t]))

wf = exp(gamma * (pif[t]))

wcN = exp(gamma * (picN[t]))

wfN = exp(gamma * (pifN[t]))

w1c[t] = wc / (wc + wf + wcN + wfN + 1)

w1f[t] = wf / (wc + wf + wcN + wfN + 1)

w1cN[t] = wcN / (wc + wf + wcN + wfN + 1)

w1fN[t] = wfN / (wc + wf + wcN + wfN + 1)

s[t+1] = s[t] + (w1c[t] * ec[t] + w1f[t] * ef[t] + w1cN[t] * ecN[t] + w1fN[t] * efN[t] + 0.005 * shock3[t])

mis[t] = s[t] - fu[t]

ret[t] = s[t] - s[t-1]

gret[g] = mean(ret[40:T])

gvol[g] = sqrt(mean(ret[40:T]^2))

gk[g] = mean(ret[40:T]^4)/$gvol[g]^4$

gwf[g] = mean(w1f[40:T])

gwc[g] = mean(w1c[40:T])

gwfN[g] = mean(w1fN[40:T])

gwcN[g] = mean(w1cN[40:T])

go[g] = 1 - gwc[g] - gwf[g] - gwcN[g] - gwfN[g]

gdis[g] = mean(abs(mis[40:T]))

ms[j] = mean(gret[1:G])

sds[j] = mean(gvol[1:G])

ks[j] = mean(gk[1:G])

mwf[j] = mean(gwf[1:G])

mwc[j] = mean(gwc[1:G])

mwfN[j] = mean(gwfN[1:G])

mwcN[j] = mean(gwcN[1:G])

mo[j] = mean(go[1:G])

dis[j] = mean(gdis[1:G])

par(mfcol = c(3,3))

plot(tax[1:J], ms[1:J], ylab = ”Mean”, xlab = ”Tax Rate”, main = ”Mean”, ylim=c(-0.015,0.015), type = ”o”, lwd = 3)

plot(tax[1:J], sds[1:J], ylab = ”Standard Deviation”, xlab = ”Tax Rate”, main = ”Standard Deviation”, ylim=c(0.004,0.007), type = ”o”, lwd = 3)

plot(tax[1:J], ks[1:J], ylab = ”Kurtosis”, xlab = ”Tax Rate”, main = ”Kurtosis”, ylim=c(2.5,10.5), type = ”o”, lwd = 3)

plot(tax[1:J], mwf[1:J], ylab = ”Short-Term Fundamentalists”, xlab = ”Tax Rate”, main = ”Short-Term Fundamentalists”, ylim=c(0,0.25), type = ”o”, lwd = 3)

plot(tax[1:J], mwc[1:J], ylab = ”Short-Term Chartists”, xlab = ”Tax Rate”, main = ”Short-Term Chartists”, ylim=c(0,0.25), type = ”o”, lwd = 3)

plot(tax[1:J], mo[1:J], ylab = ”Inactive Traders”, xlab = ”Tax Rate”, main = ”Inactive Traders”, ylim=c(0.15,0.45), type = ”o”, lwd = 3)

plot(tax[1:J], mwfN[1:J], ylab = ”Long-Term Fundamentalists”, xlab = ”Tax Rate”, main = ”Long-Term Fundamentalists”, ylim=c(0.15,0.45), type = ”o”, lwd = 3)

plot(tax[1:J], mwcN[1:J], ylab = ”Long-Term Chartists”, xlab = ”Tax Rate”, main = ”Long-Term Chartists”, ylim=c(0.15,0.45), type = ”o”, lwd = 3)

plot(tax[1:J], dis[1:J], ylab = ”Distortion”, xlab = ”Tax Rate”, main = ”Distortion”, ylim=c(0.00,0.055), type = ”o”, lwd = 3)

References

[Demary, M. (2009)] , ”Transaction Taxes and Traders with Heterogeneous Investment Horizons in an Agent-Based Financial Market Model”, Working Paper.

[R Development Core Team (2009)] , ”R: A Language and Environment for Statistical Computing”, R Foundation for Statistical Computing, Vienna, Austria, http://www.R-project.org.