

```

u.pay<-function(p,r,inv,par.setting=NULL,par.behavior=NULL) {

  e0<-par.setting[1]
  ea<-par.setting[2]
  inv.cost<-par.setting[3]
  p0<-par.setting[4]
  p1<-par.setting[5]
  penalty<-par.setting[6]
  old.payment.outcome<-par.setting[7]
  old.attack.outcome<-par.setting[8]
  old.ransom<-par.setting[8]

  # from llk.i   par.setting<-c(e0,ea,inv.cost,p0,p1,penalty,old.payment.outcome,old.attack.outcome,old.ransom)

  gamma.pay<-par.behavior[1]
  gamma.r<-par.behavior[2]
  gamma.a<-par.behavior[3]
  gamma.i<-par.behavior[4]
  ps.bias<-par.behavior[5]
  pay.bias<-par.behavior[6]
  par.social.norm<-par.behavior[7]
  par.inv.ransom<-par.behavior[8]
  par.r.past.pay<-par.behavior[9]

  # (100-r)*p-inv.cost*inv*(1+p*par.behavior[5])-p*penalty+p*par.behavior[6]
  # browser()
  # (100-r)*p-inv.cost*inv-p*penalty+p*pay.bias+alpha*(1-p)*(1-x) # negative social norm model
  (100-r)*p-inv.cost*inv-p*penalty+p*pay.bias+par.social.norm*p*old.payment.outcome # positive social norm model
}

prob.pay<-function(r,inv,par.setting=NULL,par.behavior=NULL) {

  e0<-par.setting[1]
  ea<-par.setting[2]
  inv.cost<-par.setting[3]
  p0<-par.setting[4]
  p1<-par.setting[5]
  penalty<-par.setting[6]
  old.payment.outcome<-par.setting[7]
  old.attack.outcome<-par.setting[8]
  old.ransom<-par.setting[8]

  # from llk.i   par.setting<-c(e0,ea,inv.cost,p0,p1,penalty,old.payment.outcome,old.attack.outcome,old.ransom)

  gamma.pay<-par.behavior[1]
  gamma.r<-par.behavior[2]
  gamma.a<-par.behavior[3]
  gamma.i<-par.behavior[4]
  ps.bias<-par.behavior[5]
  pay.bias<-par.behavior[6]
  par.social.norm<-par.behavior[7]
  par.inv.ransom<-par.behavior[8]
  par.r.past.pay<-par.behavior[9]

  up<-u.pay(1,r,inv,par.setting=par.setting,par.behavior=par.behavior)
  unp<-u.pay(0,r,inv,par.setting=par.setting,par.behavior=par.behavior)

  1/(1+exp(-gamma.pay*(up-unp)))
}

u.r<-function(r,inv,par.setting=NULL,par.behavior=NULL) {

  e0<-par.setting[1]
  ea<-par.setting[2]
  inv.cost<-par.setting[3]
  p0<-par.setting[4]
  p1<-par.setting[5]
  penalty<-par.setting[6]
  old.payment.outcome<-par.setting[7]
  old.attack.outcome<-par.setting[8]
  old.ransom<-par.setting[8]

  # from llk.i   par.setting<-c(e0,ea,inv.cost,p0,p1,penalty,old.payment.outcome,old.attack.outcome,old.ransom)

  gamma.pay<-par.behavior[1]
  gamma.r<-par.behavior[2]
  gamma.a<-par.behavior[3]
  gamma.i<-par.behavior[4]
  ps.bias<-par.behavior[5]
  pay.bias<-par.behavior[6]
  par.social.norm<-par.behavior[7]
  par.inv.ransom<-par.behavior[8]
  par.r.past.pay<-par.behavior[9]

  r*(prob.pay(r,inv,par.setting=par.setting,par.behavior=par.behavior)+par.r.past.pay*old.payment.outcome)
}

gre.r<-function(inv,par.setting=NULL,par.behavior=NULL) {

  e0<-par.setting[1]
  ea<-par.setting[2]
  inv.cost<-par.setting[3]
  p0<-par.setting[4]
  p1<-par.setting[5]
  penalty<-par.setting[6]
  old.payment.outcome<-par.setting[7]
  old.attack.outcome<-par.setting[8]
  old.ransom<-par.setting[8]

  # from llk.i   par.setting<-c(e0,ea,inv.cost,p0,p1,penalty,old.payment.outcome,old.attack.outcome,old.ransom)

  gamma.pay<-par.behavior[1]
  gamma.r<-par.behavior[2]
  gamma.a<-par.behavior[3]
  gamma.i<-par.behavior[4]

```

```

ps.bias<-par.behavior[5]
pay.bias<-par.behavior[6]
par.social.norm<-par.behavior[7]
par.inv.ransom<-par.behavior[8]
par.r.past.pay<-par.behavior[9]

r.list<-c(1:e0)

ur<-u.r(r.list,inv,par.setting=par.setting,par.behavior=par.behavior)

sc<-exp(gamma.r*(ur-max(ur)))
pp<-sc/sum(sc)

# browser()

cbind(r.list,pp)
}

prob.success<-function(inv,par.setting=NULL,par.behavior=NULL) {
  e0<-par.setting[1]
  ea<-par.setting[2]
  inv.cost<-par.setting[3]
  p0<-par.setting[4]
  p1<-par.setting[5]
  penalty<-par.setting[6]
  old.payment.outcome<-par.setting[7]
  old.attack.outcome<-par.setting[8]
  old.ransom<-par.setting[8]

  # from llk.i   par.setting<-c(e0,ea,inv.cost,p0,p1,penalty,old.payment.outcome,old.attack.outcome,old.ransom)

  gamma.pay<-par.behavior[1]
  gamma.r<-par.behavior[2]
  gamma.a<-par.behavior[3]
  gamma.i<-par.behavior[4]
  ps.bias<-par.behavior[5]
  pay.bias<-par.behavior[6]
  par.social.norm<-par.behavior[7]
  par.inv.ransom<-par.behavior[8]
  par.r.past.pay<-par.behavior[9]

  p0-p1*inv
}

u.att<-function(att,inv.list,par.setting=NULL,par.behavior=NULL) {
  e0<-par.setting[1]
  ea<-par.setting[2]
  inv.cost<-par.setting[3]
  p0<-par.setting[4]
  p1<-par.setting[5]
  penalty<-par.setting[6]
  old.payment.outcome<-par.setting[7]
  old.attack.outcome<-par.setting[8]
  old.ransom<-par.setting[8]

  # from llk.i   par.setting<-c(e0,ea,inv.cost,p0,p1,penalty,old.payment.outcome,old.attack.outcome,old.ransom)

  gamma.pay<-par.behavior[1]
  gamma.r<-par.behavior[2]
  gamma.a<-par.behavior[3]
  gamma.i<-par.behavior[4]
  ps.bias<-par.behavior[5]
  pay.bias<-par.behavior[6]
  par.social.norm<-par.behavior[7]
  par.inv.ransom<-par.behavior[8]
  par.r.past.pay<-par.behavior[9]

  result<-ea
  if(att>0) {
    inv<-inv.list[att]

    #   ps<-prob.success(inv,par.setting=par.setting,par.behavior = par.behavior)+prev.att*old.attack.outcome
    ps<-prob.success(inv,par.setting=par.setting,par.behavior = par.behavior)
    qr<-gre.r(inv,par.setting=par.setting,par.behavior=par.behavior)
    pp<-prob.pay(qr[,1],inv,par.setting=par.setting,par.behavior=par.behavior)

    result<-ps*sum(qr[,1]*qr[,2]*pp)
  }
  result
}

gre.att<-function(inv.list,par.setting=NULL,par.behavior=NULL) {

  e0<-par.setting[1]
  ea<-par.setting[2]
  inv.cost<-par.setting[3]
  p0<-par.setting[4]
  p1<-par.setting[5]
  penalty<-par.setting[6]
  old.payment.outcome<-par.setting[7]
  old.attack.outcome<-par.setting[8]
  old.ransom<-par.setting[8]

  # from llk.i   par.setting<-c(e0,ea,inv.cost,p0,p1,penalty,old.payment.outcome,old.attack.outcome,old.ransom)

  gamma.pay<-par.behavior[1]
  gamma.r<-par.behavior[2]
  gamma.a<-par.behavior[3]
  gamma.i<-par.behavior[4]
  ps.bias<-par.behavior[5]
  pay.bias<-par.behavior[6]
  par.social.norm<-par.behavior[7]
  par.inv.ransom<-par.behavior[8]
  par.r.past.pay<-par.behavior[9]

```

```

att.list<-c(0,1,2)
ua<-NULL
for(att in att.list) {
  ua<-c(ua,u.att(att,inv.list,par.setting = par.setting,par.behavior = par.behavior))
}

# flat QRE

sc<-exp(gamma.a*(ua-max(ua)))
pp<-sc/sum(sc)

# nested QRE

# p.d1<-1/(1+exp(-gamma.d*(ua[2]-ua[3])))
# p.d2<-1/(1+exp(-gamma.d*(ua[3]-ua[2])))
#
# u.att<-p.d1*ua[2]+p.d1*ua[3]
# p.att<-1/(1+exp(-gamma.a*(u.att-ua[1])))
#
# pp<-c(1-p.att,p.att*p.d1,p.att*p.d2)

# browser()

cbind(att.list,pp)
}

u.inv<-function(inv,inv.other,par.setting = NULL,par.behavior = NULL) {

  e0<-par.setting[1]
  ea<-par.setting[2]
  inv.cost<-par.setting[3]
  p0<-par.setting[4]
  p1<-par.setting[5]
  penalty<-par.setting[6]
  old.payment.outcome<-par.setting[7]
  old.attack.outcome<-par.setting[8]
  old.ransom<-par.setting[8]

  # from llk.i   par.setting<-c(e0,ea,inv.cost,p0,p1,penalty,old.payment.outcome,old.attack.outcome,old.ransom)

  gamma.pay<-par.behavior[1]
  gamma.r<-par.behavior[2]
  gamma.a<-par.behavior[3]
  gamma.i<-par.behavior[4]
  ps.bias<-par.behavior[5]
  pay.bias<-par.behavior[6]
  par.social.norm<-par.behavior[7]
  par.inv.ransom<-par.behavior[8]
  par.r.past.pay<-par.behavior[9]

  inv.cost<-inv.cost*(1+ps.bias)

  inv.list<-c(inv,inv.other)
  gra<-gre.att(inv.list,par.setting=par.setting,par.behavior=par.behavior)

  # bias of attack prob

  #gra[1,2]<-gra[1,2]*(1+pa.bias)
  #gra[,2]<-gra[,2]/sum(gra[,2])

  # browser()

  ps<-prob.success(inv,par.setting=par.setting,par.behavior = par.behavior)
  # ps<-p0*(1+ps.bias)-p1*inv
  u.noloss<-(e0-inv*inv.cost)*(gra[1,2]+gra[3,2]+gra[2,2]*(1-ps))

  qr<-gre.r(inv,par.setting=par.setting,par.behavior=par.behavior)
  pp<-prob.pay(qr[,1],inv,par.setting=par.setting,par.behavior=par.behavior)

  # u.loss<-gra[2,2]*ps*sum(qr[,2])*(pp*(e0-qr[,1]-inv*inv.cost-penalty)+(1-pp)*(-inv*inv.cost))

  rr<-(1-par.inv.ransom)*qr[,1]+par.inv.ransom*old.ransom
  u.loss<-gra[2,2]*ps*sum(qr[,2])*(pp*(e0-rr-inv*inv.cost-penalty)+(1-pp)*(-inv*inv.cost))

  # browser()

  u.noloss+u.loss
}

llk.i<-function(index,par,par.restrict=NULL,par.restrict.value=NULL,data=NULL) {
  if(!is.null(par.restrict)) {
    final.par<-par.restrict.value
    counter<-1
    for(i in 1:length(par.restrict)) if(par.restrict[i]==1) {
      final.par[i]<-par[counter]
      counter<-counter+1
    }
  } else final.par<-par

  e0<-data[index,"Data1"]
  ea<-data[index,"Outside"]
  inv.cost<-data[index,"Investment1"]
  p0<-0.8
  p1<-0.5
  penalty<-data[index,"Penalty"]

  # par.setting<-c(e0,ea,inv.cost,p0,p1,penalty,data[index,"old_attack_outcome"])

  old.payment.outcome<-ifelse(is.na(data[index,"old_payment_outcome"]),0,data[index,"old_payment_outcome"])
  old.attack.outcome<-ifelse(is.na(data[index,"old_attack_outcome"]),0,data[index,"old_attack_outcome"])
  old.ransom<-ifelse(is.na(data[index,"old_ransom"]),0,data[index,"old_attack_outcome"])

  par.setting<-c(e0,ea,inv.cost,p0,p1,penalty,old.payment.outcome,old.attack.outcome,old.ransom)

```

```

gamma.i<-final.par[4]

u.m1<-matrix(nrow=2,ncol=2)
u.m2<-matrix(nrow=2,ncol=2)

for(i in 0:1) for(j in 0:1) {
  u.m1[(i+1),(j+1)]<-u.inv(i,j,par.setting = par.setting,par.behavior = final.par)
  u.m2[(i+1),(j+1)]<-u.inv(i,j,par.setting = par.setting,par.behavior = final.par)
}

dimnames(u.m1)<-list(c("0","1"),c("0","1"))
dimnames(u.m2)<-list(c("0","1"),c("0","1"))

# browser()

gre.i<-find.gre(c(final.par[4],final.par[4]),list(u.m1,u.m2),tol=1e-8,max.iter=100,mode="m")

gre.i1<-gre.i[[1]]
llk.i1<-(-log(gre.i1[gre.i1[,1]==data[index,"i_1"],2]))

gre.i2<-gre.i[[2]]
llk.i2<-(-log(gre.i2[gre.i2[,1]==data[index,"i_2"],2]))

# browser()

attack.decision<-data[index,"attack.decision"]

gra<-gre.att(c(data[index,"i_1"],data[index,"i_2"]),par.setting=par.setting,par.behavior=final.par)
llk.a<-(-log(gra[gra[,1]==data[index,"attack.decision"],2]))

llk.r<-0
r<-data[index,"ransom"]

if(!is.na(r)) {

  if(r==0) r<-1
  if(attack.decision==1) inv<-data[index,"i_1"]
  if(attack.decision==2) inv<-data[index,"i_2"]

  qr<-gre.r(inv,par.setting=par.setting,par.behavior=final.par)
  llk.r<-(-log(qr[match(r,qr[,1]),2]))
# browser()
}

llk.pay<-0
if(!is.na(data[index,"p_1"])|!(is.na(data[index,"p_2"]))) {
  if(attack.decision==1) inv<-data[index,"i_1"]
  if(attack.decision==2) inv<-data[index,"i_2"]

  if(attack.decision==1) pay<-data[index,"p_1"]
  if(attack.decision==2) pay<-data[index,"p_2"]

  pp<-prob.pay(r,inv,par.setting=par.setting,par.behavior=final.par)

  if(pp==1) llk.pay<-(1-pay)*100
  else llk.pay<-(-pay*log(pp)-(1-pay)*log(1-pp))
}

result<-llk.i1+llk.i2+llk.a+llk.r+llk.pay
if(is.na(result)) browser()
result
}

llk<-function(par,par.restrict=NULL,par.restrict.value=NULL,data=NULL) {

  # weird bug .. parallel gives different results!

  #result<-parLapply(cl=cluster,c(1:nrow(data)),llk.i,par=par,par.restrict=par.restrict,
  # par.restrict.value=par.restrict.value,data=data)
  result<-lapply(c(1:nrow(data)),llk.i,par=par,par.restrict=par.restrict,
    par.restrict.value=par.restrict.value,data=data)

  #browser()
  #sum(result)
  do.call(sum,result)
}

fit.model<-function(start,data.list=NULL,lower=NULL,upper=NULL,scale=NULL,
  par.restrict=NULL,par.restrict.value=NULL) {

  ans<-nlminb(start,scale=scale,lower=lower,upper=upper,llk,par.restrict=par.restrict,
    par.restrict.value=par.restrict.value,data=data.list,
    control=list(trace=2,iter.max=1000,eval.max=5000,rel.tol=1e-8))

  # a<-scale
  # ans<-optim(start,lower=lower,upper=upper,llk,par.restrict=par.restrict,
  # par.restrict.value=par.restrict.value,data=data,method="L-BFGS-B",
  # control=list(trace=6,maxit=1000))

  ans
}

```