```
1.  windDefault<-function(){
2.
3.    currentSource<-as.matrix(read.table(file="ps3r002a.dat",skip=1))
4.    dim(currentSource)<-c(17,2)
5.    currentCaptor<-as.matrix(read.table(file="ps3r002b.dat",skip=1))
6.    dim(currentCaptor)<-c(17,2)
7.    sourceFert<-as.matrix(read.table(file="f29g2794.dat",skip=1))
8.    dim(sourceFert)<-c(17,1)
9.    captorFert<-as.matrix(read.table(file="f29g2794.dat",skip=1))
10.   dim(captorFert)<-c(17,1)
11.   domestFert<-as.matrix(read.table(file="f29g2444.dat",skip=1))
12.   dim(domestFert)<-c(17,1)
13.   exportFert<-as.matrix(read.table(file="f29g2444.dat",skip=1))
14.   dim(exportFert)<-c(17,1)
15.   newDomFert<-as.matrix(read.table(file="f29g2094.dat",skip=1))
16.   dim(newDomFert)<-c(17,1)
17.   newAfrExpFert<-as.matrix(read.table(file="f29g2094.dat",skip=1))
18.   dim(newAfrExpFert)<-c(17,1)
19.   newAbrExpFert<-as.matrix(read.table(file="f29g2094.dat",skip=1))
20.   dim(newAbrExpFert)<-c(17,1)
21.   sourceSurv<-as.matrix(read.table(file="ss3e110.dat",skip=1))
22.   dim(sourceSurv)<-c(18,2)
23.   captorSurv<-as.matrix(read.table(file="ss3e110.dat",skip=1))
24.   dim(captorSurv)<-c(18,2)
25.   domestSurv<-as.matrix(read.table(file="ss2e110.dat",skip=1))
26.   dim(domestSurv)<-c(18,2)
27.   exportSurv<-as.matrix(read.table(file="ss2e110.dat",skip=1))
28.   dim(exportSurv)<-c(18,2)
29.   capAfrSurv<-as.matrix(read.table(file="sw4e518.dat",skip=1))
30.   dim(capAfrSurv)<-c(18,2)
31.   capAbrSurv<-as.matrix(read.table(file="sw4e518.dat",skip=1))
32.   dim(capAbrSurv)<-c(18,2)
33.   captureSize<-as.matrix(read.table(file="c241p13.dat",skip=1))
34.   dim(captureSize)<-c(17,2)
35.
36.   defaultArray<-numeric(18*2*16)
37.   dim(defaultArray)<-c(18,2,16)
38.   defaultArray[1:17,,1]<-currentSource
39.   defaultArray[1:17,,2]<-currentCaptor
40.   defaultArray[1:17,1,3]<-sourceFert
41.   defaultArray[1:17,1,4]<-captorFert
42.   defaultArray[1:17,1,5]<-domestFert
43.   defaultArray[1:17,1,6]<-exportFert
44.   defaultArray[1:17,1,7]<-newDomFert
45.   defaultArray[1:17,1,8]<-newAfrExpFert
46.   defaultArray[1:17,1,9]<-newAbrExpFert
47.   defaultArray[,,10]<-sourceSurv
48.   defaultArray[,,11]<-captorSurv
49.   defaultArray[,,12]<-domestSurv
50.   defaultArray[,,13]<-exportSurv
51.   defaultArray[,,14]<-capAfrSurv
52.   defaultArray[,,15]<-capAbrSurv
53.   defaultArray[1:17,,16]<-captureSize
54.
55.   return(defaultArray)
56. }
57.
58. terrTypel<-function(defaultArray,partArray,decData,numIter,numDec,firstDec){
59.
60.   ##############################################################################
61.   ##  Initialize aggregate output storage
62.   ##############################################################################
63.   exportArray<-numeric(numDec*17*2)
64.   dim(exportArray)<-c(numDec,17,2)
65.   decPopVect<-numeric(numDec)
66.   Male_pop<-numeric(numDec)
67.   Female_pop<-numeric(numDec)
68.   Dome<-numeric(numDec)
69.   Male_dome<-numeric(numDec)
70.   Female_dome<-numeric(numDec)
71.   NewDom<-numeric(numDec)
72.   Male_NewDom<-numeric(numDec)
73.   Female_NewDom<-numeric(numDec)
```

Add Domestic slaves problem

1

```
NewExp<-numeric(numDec)
Male_NewExp<-numeric(numDec)
Female_NewExp<-numeric(numDec)
New_dome_percent<-numeric(numDec)
Export_percent<-numeric(numDec)

################################################################################
##    Initialize territorial output storage
################################################################################
allRegPopTracker<-numeric(numDec*17*2*numIter)
dim(allRegPopTracker)<-c(numDec,17,2,numIter)
allDomPopTracker<-numeric(numDec*17*2*numIter)
dim(allDomPopTracker)<-c(numDec,17,2,numIter)
allNDPopTracker<-numeric(numDec*17*2*numIter)
dim(allNDPopTracker)<-c(numDec,17,2,numIter)
allNAEPopTracker<-numeric(numDec*17*2*numIter)
dim(allNAEPopTracker)<-c(numDec,17,2,numIter)
allExportTracker<-numeric(numDec*17*2*numIter)
dim(allExportTracker)<-c(numDec,17,2,numIter)
newBirths<-numeric(numDec)
newDeaths<-numeric(numDec)

################################################################################
##    Convert decData to the appropriate format
################################################################################
terr<-as.character(decData[1])
peakDec<-as.numeric(decData[5])
zeroDec<-as.numeric(decData[6])
postPeakCapt<-as.numeric(decData[7])
initPop<-as.numeric(decData[8])
decExports<-as.numeric(decData[9:33])
decs<-seq(firstDec,firstDec-10*(numDec-1),by=-10)


mostRecentPop<-initPop
for(j in 1:numDec){


  ##export driven slaving
  print(decExports[j])
  if(decExports[j]>0){
    currCaptMult<-1
    currPartMult<-1
    precisionRatio<-0
    maxIter<-0

    while((abs(precisionRatio-1)>.05)&(maxIter<5)){
      callToSim<-standardSim2.1(defaultArray,partArray,currCaptMult,currPartMult,postPeakCapt,decExports[j],mostRecentPop,numIter)
      allRegPopTracker[j,,,]<-callToSim$ARPT
      allDomPopTracker[j,,,]<-callToSim$ADPT
      allExportTracker[j,,,]<-callToSim$AEPT
      allNDPopTracker[j,,,]<-callToSim$ANDPT
      allNAEPopTracker[j,,,]<-callToSim$ANAEPT
      birthTracker<-callToSim$BT
      deathTracker<-callToSim$DT
      captiveTracker<-callToSim$CT
      exportTracker<-callToSim$ET
      currCaptMult<-callToSim$CCM
      if (currCaptMult == 0){currCaptMult <- .01}
      precisionRatio<-callToSim$PR
      maxIter<-maxIter+1
    }
    endPopRatio<-mostRecentPop/sum(allRegPopTracker[j,,,numIter])
    allRegPopTracker[j,,,]<-allRegPopTracker[j,,,]*endPopRatio
    allDomPopTracker[j,,,]<-allDomPopTracker[j,,,]*endPopRatio
    allExportTracker[j,,,]<-allExportTracker[j,,,]*endPopRatio
    allNDPopTracker[j,,,]<-allNDPopTracker[j,,,]*endPopRatio
    allNAEPopTracker[j,,,]<-allNAEPopTracker[j,,,]*endPopRatio
    exportArray[j,,]<-allExportTracker[j,,,numIter-1]+allExportTracker[j,,,numIter]
    mostRecentPop<-sum(allRegPopTracker[j,,,numIter-2])
    decPopVect[j]<-mostRecentPop
    newBirths[j]<-(sum(birthTracker[numIter-1,])+sum(birthTracker[numIter,]))*endPopRatio
    newDeaths[j]<-(sum(deathTracker[numIter-1,])+sum(deathTracker[numIter,]))*endPopRatio
  }
```

Create for storage

$ means just this element from overall output

input w/ parameters

simulated final year

'real' final year

CCM

end pop. ratio

compare to "real" data - most recent Pop

adds to 5-yr periods to get decade of exports

scaling

2

case of exports = 0.

```
148.    if(decExports[j]==0){
149.      callToSim<-simpleSim(defaultArray,numIter)
150.      allRegPopTracker[j,,,]<-callToSim$ARPT
151.      birthTracker<-callToSim$BT
152.      deathTracker<-callToSim$DT
153.      captiveTracker<-callToSim$CT
154.      exportTracker<-callToSim$ET
155.      endPopRatio<-mostRecentPop/sum(allRegPopTracker[j,,,numIter])
156.      allRegPopTracker[j,,,]<-allRegPopTracker[j,,,]*endPopRatio
157.      mostRecentPop<-sum(allRegPopTracker[j,,,numIter-2])
158.      decPopVect[j]<-mostRecentPop
159.      newBirths[j]<-(birthTracker[numIter-1]+birthTracker[numIter])*endPopRatio
160.      newDeaths[j]<-(deathTracker[numIter-1]+deathTracker[numIter])*endPopRatio
161.    }
162.  }
163.
164.  for (k in 1:numDec){
165.    Male_pop[k]<-sum(allRegPopTracker[k,,2,numIter-2])
166.    Female_pop[k]<-sum(allRegPopTracker[k,,1,numIter-2])
167.    Dome[k]<-sum(allDomPopTracker[k,,,numIter-2])
168.    Male_dome[k]<-sum(allDomPopTracker[k,,2,numIter-2])
169.    Female_dome[k]<-sum(allDomPopTracker[k,,1,numIter-2])
170.    NewDom[k]<-sum(allNDPopTracker[k,,,numIter-2])
171.    Male_NewDom[k]<-sum(allNDPopTracker[k,,2,numIter-2])
172.    Female_NewDom[k]<-sum(allNDPopTracker[k,,1,numIter-2])
173.    NewExp[k]<-sum(allNAEPopTracker[k,,,numIter-2])
174.    Male_NewExp[k]<-sum(allNAEPopTracker[k,,2,numIter-2])
175.    Female_NewExp[k]<-sum(allNAEPopTracker[k,,1,numIter-2])
176.    New_dome_percent[k]<-(sum(allNDPopTracker[k,,,numIter-2]))/(sum(allRegPopTracker[k,,,numIter-2]))
177.    Export_percent[k]<-(sum(allNAEPopTracker[k,,,numIter-2]))/(sum(allRegPopTracker[k,,,numIter-2]))
178.  }
179.
180.
181.  ################################################################################
182.  ##    Territorial output
183.  ################################################################################
184.  forRates<-cbind(initPop,decPopVect)
185.  midPeriodPops<-(forRates[1:(length(forRates)-1)]+forRates[2:length(forRates)])/2
186.  arpt<-allRegPopTracker[,,,numIter-2]
187.  adpt<-allDomPopTracker[,,,numIter-2]
188.  andpt<-allNDPopTracker[,,,numIter-2]
189.  anaept<-allNAEPopTracker[,,,numIter-2]
190.
191.  temp<-data.frame(Region=terr)
192.  write.table(temp,paste("",terr,".csv",sep=""),row.names = FALSE, col.names = TRUE)
193.  for(z in 1:numDec){
194.    temp<-data.frame(Year=decs[z])
195.    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
196.    temp<-data.frame(AdultSexRatio=c(sum(arpt[z,4:11,2])/sum(arpt[z,3:8,1]),"DomesticSlavePopulation"))
197.    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
198.    temp<-data.frame(Female=adpt[z,,1],Male=adpt[z,,2])
199.    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE, sep=",")
200.    temp<-data.frame(TotalDomesticFemaleSlaves=sum(adpt[z,,1]))
201.    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
202.    temp<-data.frame(TotalDomesticMaleSlaves=sum(adpt[z,,2]))
203.    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
204.    temp<-data.frame(TotalDomesticSlaves=c(sum(adpt[z,,]),"NewDomesticSlavePopulation"))
205.    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
206.    temp<-data.frame(Female=andpt[z,,1],Male=andpt[z,,2])
207.    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE, sep = ",")
208.    temp<-data.frame(TotalNewDomesticFemaleSlaves=sum(andpt[z,,1]))
209.    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
210.    temp<-data.frame(TotalNewDomesticMaleSlaves=sum(andpt[z,,2]))
211.    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
212.    temp<-data.frame(TotalNewDomesticSlaves=c(sum(andpt[z,,]),"NewAfricanExportSlavePopulation"))
213.    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
214.    temp<-data.frame(Female=anaept[z,,1],Male=anaept[z,,2])
215.    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE, sep=",")
216.    temp<-data.frame(TotalNewAfricanExportFemaleSlaves=sum(anaept[z,,1]))
217.    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
218.    temp<-data.frame(TotalNewAfricanExportMaleSlaves=sum(anaept[z,,2]))
219.    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
220.    temp<-data.frame(TotalNewAfricanExportSlaves=sum(anaept[z,,]))
221.    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
```

get populations 10 years earlier

```
    temp<-data.frame(AnnualBirths=newBirths[z]/10)
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(AnnualBirthRate=newBirths[z]/(10*midPeriodPops[z]))
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(AnnualDeaths=newDeaths[z]/10)
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(AnnualDeathRate=c(newDeaths[z]/(10*midPeriodPops[z]),""))
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
  }
  return(list(exportArray=exportArray,decPopVect=decPopVect,Male_pop=Male_pop,Female_pop=Female_pop,Dome=Dome,Male_dome=Male_dome,Female_dome=Female_dome,NewDom=NewDom,Male_NewDom=Ma
}

simpleSim<-function(defaultArray,numIter){

  ARPT<-numeric(17*2*numIter)
  dim(ARPT)<-c(17,2,numIter)
  BT<-numeric(numIter)
  DT<-numeric(numIter)

  defaultArray[1:17,1,4]->captorFert
  defaultArray[,,11]->captorSurv
  defaultArray[1:17,,2]->currentCaptor
  currentCaptor[,2]<-.95109*currentCaptor[,2]

  for (ii in 1:numIter){
    previousCaptor<-currentCaptor
    currentCaptor<-previousCaptor*captorSurv[2:18,]
    DT[ii]<-sum(previousCaptor-currentCaptor)
    midCaptor<-(previousCaptor+currentCaptor)/2
    births<-midCaptor[3:11,1]*5*captorFert[3:11]
    newFemales<-sum(births)
    newMales<-sum(births)*1.03
    currentCaptor[17,]<-currentCaptor[16,]+currentCaptor[17,]
    currentCaptor[2:16,]<-currentCaptor[1:15,]
    currentCaptor[1,]<-c(newFemales,newMales)*captorSurv[1,]
    BT[ii]<-newFemales+newMales
    DT[ii]<-DT[ii]+sum(c(newFemales,newMales)-currentCaptor[1,])

    ARPT[,,ii]<-currentCaptor

  }
  CT<-0
  ET<-0
  return(list(ARPT=ARPT,BT=BT,DT=DT,CT=CT,ET=ET))
}

standardSim1.1<-function(defaultArray,partArray,currCaptMult,currPartMult,postPeakCapt,decExports,mostRecentPop,numIter){

  defaultArray[1:17,,1]->currentSource
  defaultArray[1:17,,2]->currentCaptor
  defaultArray[1:17,1,3]->sourceFert
  defaultArray[1:17,1,4]->captorFert
  defaultArray[1:17,1,5]->domestFert
  defaultArray[1:17,1,6]->exportFert
  defaultArray[1:17,1,7]->newDomFert
  defaultArray[1:17,1,8]->newAfrExpFert
  defaultArray[1:17,1,9]->newAbrExpFert
  defaultArray[,,10]->sourceSurv
  defaultArray[,,11]->captorSurv
  defaultArray[,,12]->domestSurv
  defaultArray[,,13]->exportSurv
  defaultArray[,,14]->capAfrSurv
  defaultArray[,,15]->capAbrSurv
  defaultArray[1:17,,16]->captureSize

  captureSize<-captureSize*currCaptMult
  partition<-partArray*currPartMult

  currentSource[,2]<-.95109*currentSource[,2]
  currentCaptor[,2]<-.95109*currentCaptor[,2]
  currentDomestics<-numeric(34)
  dim(currentDomestics)<-c(17,2)

  ARPT<-numeric(17*2*numIter)
```

4

```
592.
593.      previousNewDomestics<-captives*(1-partition)
594.      currentNewDomestics<-previousNewDomestics*sourceSurv[2:18,]^.5*capAfrSurv[2:18,]*domestSurv[2:18,]^(3/10)
595.      DT[jjj,3]<-sum(previousNewDomestics-currentNewDomestics)
596.      midNewDomestics<-(previousNewDomestics+currentNewDomestics)/2
597.      births<-midNewDomestics[3:11,1]*(2.5*sourceFert[3:11]+1.5*newDomFert[3:11]+1*domestFert[3:11])
598.      newFemales<-sum(births)
599.      newMales<-sum(births)*1.03
600.      currentNewDomestics[17,]<-currentNewDomestics[16,]+currentNewDomestics[17,]
601.      currentNewDomestics[2:16,]<-currentNewDomestics[1:15,]
602.      currentNewDomestics[1,]<-c(newFemales,newMales)*sourceSurv[1,]^.5*capAfrSurv[1,]*domestSurv[1,]^(3/10)
603.      BT[jjj,3]<-newFemales+newMales
604.      DT[jjj,3]<-DT[jjj,3]+sum(c(newFemales,newMales)-currentNewDomestics[1,])
605.
606.      previousDomestics<-currentDomestics
607.      currentDomestics<-previousDomestics*domestSurv[2:18,]
608.      DT[jjj,4]<-sum(previousDomestics-currentDomestics)
609.      midDomestics<-(previousDomestics+currentDomestics)/2
610.      births<-midDomestics[3:11,1]*5*domestFert[3:11]
611.      newFemales<-sum(births)
612.      newMales<-sum(births)*1.03
613.      currentDomestics[17,]<-currentDomestics[16,]+currentDomestics[17,]
614.      currentDomestics[2:16,]<-currentDomestics[1:15,]
615.      currentDomestics[1,]<-c(newFemales,newMales)*domestSurv[1,]
616.      currentDomestics<-currentDomestics+currentNewDomestics
617.      BT[jjj,4]<-newFemales+newMales
618.      DT[jjj,4]<-DT[jjj,4]+sum(c(newFemales,newMales)-currentDomestics[1,])
619.
620.      previousNewAbrExp<-captives*partition
621.      currentNewAbrExp<-previousNewAbrExp*sourceSurv[2:18,]^(1/2)*capAfrSurv[2:18,]*capAbrSurv[2:18,]*exportSurv[2:18,]^(1/10)
622.      DT[jjj,5]<-sum(previousNewAbrExp-currentNewAbrExp)
623.      midNewAbrExp<-(previousNewAbrExp+currentNewAbrExp)/2
624.      births<-midNewAbrExp[3:11,1]*(2.5*sourceFert[3:11]+newAfrExpFert[3:11]+newAbrExpFert[3:11]+.5*exportFert[3:11])
625.      newFemales<-sum(births)
626.      newMales<-sum(births)*1.03
627.      currentNewAbrExp[17,]<-currentNewAbrExp[16,]+currentNewAbrExp[17,]
628.      currentNewAbrExp[2:16,]<-currentNewAbrExp[1:15,]
629.      currentNewAbrExp[1,]<-c(newFemales,newMales)*sourceSurv[1,]^(1/2)*capAfrSurv[1,]*capAbrSurv[1,]*exportSurv[1,]^(1/10)
630.      BT[jjj,5]<-newFemales+newMales
631.      DT[jjj,5]<-DT[jjj,5]+sum(c(newFemales,newMales)-currentNewAbrExp[1,])
632.
633.      ET[jjj]<-sum(currentNewAbrExp)
634.
635.      ARPT[,,jjj]<-currentSource+currentCaptor+currentDomestics
636.      ADPT[,,jjj]<-currentDomestics
637.      AEPT[,,jjj]<-currentNewAbrExp
638.      ANDPT[,,jjj]<-currentNewDomestics
639.      ANAEPT[,,jjj]<-currentNewAbrExp
640.    }
641.
642.    endPopRatio<-mostRecentPop/sum(ARPT[,,numIter])
643.    defCaptives<-(CT[numIter-1]+CT[numIter])*endPopRatio
644.    defExports<-(ET[numIter-1]+ET[numIter])*endPopRatio
645.    CT<-defCaptives
646.    ET<-defExports
647.
648.    return(list(ARPT=ARPT,ADPT=ADPT,AEPT=AEPT,ANDPT=ANDPT,ANAEPT=ANAEPT,BT=BT,CT=CT,DT=DT,ET=ET))
649.  }
650.
651.  standardSim2.1<-function(defaultArray,partArray,currCaptMult,currPartMult,postPeakCapt,decExports,mostRecentPop,numIter){
652.
653.    defaultArray[1:17,,1]->currentSource
654.    defaultArray[1:17,,2]->currentCaptor
655.    defaultArray[1:17,1,3]->sourceFert
656.    defaultArray[1:17,1,4]->captorFert
657.    defaultArray[1:17,1,5]->domestFert
658.    defaultArray[1:17,1,6]->exportFert
659.    defaultArray[1:17,1,7]->newDomFert
660.    defaultArray[1:17,1,8]->newAfrExpFert
661.    defaultArray[1:17,1,9]->newAbrExpFert
662.    defaultArray[,,10]->sourceSurv
663.    defaultArray[,,11]->captorSurv
664.    defaultArray[,,12]->domestSurv
665.    defaultArray[,,13]->exportSurv
```

```
666.  defaultArray[,,14]->capAfrSurv
667.  defaultArray[,,15]->capAbrSurv
668.  defaultArray[1:17,,16]->captureSize
669.
670.  captureSize<-captureSize*currCaptMult
671.  partition<-partArray*currPartMult
672.
673.  currentSource[,2]<-.95109*currentSource[,2]
674.  currentCaptor[,2]<-.95109*currentCaptor[,2]
675.  currentDomestics<-numeric(34)
676.  dim(currentDomestics)<-c(17,2)
677.
678.  ARPT<-numeric(17*2*numIter)
679.  dim(ARPT)<-c(17,2,numIter)
680.  ADPT<-numeric(17*2*numIter)
681.  dim(ADPT)<-c(17,2,numIter)
682.  AEPT<-numeric(17*2*numIter)
683.  dim(AEPT)<-c(17,2,numIter)
684.  ANDPT<-numeric(17*2*numIter)
685.  dim(ANDPT)<-c(17,2,numIter)
686.  ANAEPT<-numeric(17*2*numIter)
687.  dim(ANAEPT)<-c(17,2,numIter)
688.  BT<-numeric(numIter*5)
689.  dim(BT)<-c(numIter,5)
690.  DT<-numeric(numIter*5)
691.  dim(DT)<-c(numIter,5)
692.  CT<-numeric(numIter)
693.  ET<-numeric(numIter)
694.
695.  for (kkk in 1:numIter){
696.    previousSource<-currentSource
697.    currentSource<-previousSource*sourceSurv[2:18,]
698.    DT[kkk,1]<-sum(previousSource-currentSource)
699.    midSource<-(previousSource+currentSource)/2
700.    captives<-midSource*5*captureSize
701.    currentSource<-currentSource-captives
702.    midSource<-(previousSource+currentSource)/2
703.    births<-midSource[3:11,1]*5*sourceFert[3:11]
704.    newFemales<-sum(births)
705.    newMales<-sum(births)*1.03
706.    currentSource[17,]<-currentSource[16,]+currentSource[17,]
707.    currentSource[2:16,]<-currentSource[1:15,]
708.    currentSource[1,]<-c(newFemales,newMales)*sourceSurv[1,]
709.    BT[kkk,1]<-newFemales+newMales
710.    DT[kkk,1]<-DT[kkk,1]+sum(c(newFemales,newMales)-currentSource[1,])
711.
712.    CT[kkk]<-sum(captives)
713.
714.    previousCaptor<-currentCaptor
715.    currentCaptor<-previousCaptor*captorSurv[2:18,]
716.    DT[kkk,2]<-sum(previousCaptor-currentCaptor)
717.    midCaptor<-(previousCaptor+currentCaptor)/2
718.    births<-midCaptor[3:11,1]*5*captorFert[3:11]
719.    newFemales<-sum(births)
720.    newMales<-sum(births)*1.03
721.    currentCaptor[17,]<-currentCaptor[16,]+currentCaptor[17,]
722.    currentCaptor[2:16,]<-currentCaptor[1:15,]
723.    currentCaptor[1,]<-c(newFemales,newMales)*captorSurv[1,]
724.    BT[kkk,2]<-newFemales+newMales
725.    DT[kkk,2]<-DT[kkk,2]+sum(c(newFemales,newMales)-currentCaptor[1,])
726.
727.    previousNewDomestics<-captives*(1-partition)
728.    currentNewDomestics<-previousNewDomestics*sourceSurv[2:18,]^.5*capAfrSurv[2:18,]*domestSurv[2:18,]^(3/10)
729.    DT[kkk,3]<-sum(previousNewDomestics-currentNewDomestics)
730.    midNewDomestics<-(previousNewDomestics+currentNewDomestics)/2
731.    births<-midNewDomestics[3:11,1]*(2.5*sourceFert[3:11]+1.5*newDomFert[3:11]+1*domestFert[3:11])
732.    newFemales<-sum(births)
733.    newMales<-sum(births)*1.03
734.    currentNewDomestics[17,]<-currentNewDomestics[16,]+currentNewDomestics[17,]
735.    currentNewDomestics[2:16,]<-currentNewDomestics[1:15,]
736.    currentNewDomestics[1,]<-c(newFemales,newMales)*sourceSurv[1,]^.5*capAfrSurv[1,]*domestSurv[1,]^(3/10)
737.    BT[kkk,3]<-newFemales+newMales
738.    DT[kkk,3]<-DT[kkk,3]+sum(c(newFemales,newMales)-currentNewDomestics[1,])
739.
```

CCM → default = 1

= 0 to start

to store values

kkk = index for iteration from 1 to 8

```
740.      previousDomestics<-currentDomestics
741.      currentDomestics<-previousDomestics*domestSurv[2:18,]
742.      DT[kkk,4]<-sum(previousDomestics-currentDomestics)
743.      midDomestics<-(previousDomestics+currentDomestics)/2
744.      births<-midDomestics[3:11,1]*5*domestFert[3:11]
745.      newFemales<-sum(births)
746.      newMales<-sum(births)*1.03
747.      currentDomestics[17,]<-currentDomestics[16,]+currentDomestics[17,]
748.      currentDomestics[2:16,]<-currentDomestics[1:15,]
749.      currentDomestics[1,]<-c(newFemales,newMales)*domestSurv[1,]
750.      currentDomestics<-currentDomestics+currentNewDomestics
751.      BT[kkk,4]<-newFemales+newMales
752.      DT[kkk,4]<-DT[kkk,4]+sum(c(newFemales,newMales)-currentDomestics[1,])
753.
754.      previousNewAbrExp<-captives*partition
755.      currentNewAbrExp<-previousNewAbrExp*sourceSurv[2:18,]^(1/2)*capAfrSurv[2:18,]*capAbrSurv[2:18,]*exportSurv[2:18,]^(1/10)
756.      DT[kkk,5]<-sum(previousNewAbrExp-currentNewAbrExp)
757.      midNewAbrExp<-(previousNewAbrExp+currentNewAbrExp)/2
758.      births<-midNewAbrExp[3:11,1]*(2.5*sourceFert[3:11]+newAfrExpFert[3:11]+newAbrExpFert[3:11]+.5*exportFert[3:11])
759.      newFemales<-sum(births)
760.      newMales<-sum(births)*1.03
761.      currentNewAbrExp[17,]<-currentNewAbrExp[16,]+currentNewAbrExp[17,]
762.      currentNewAbrExp[2:16,]<-currentNewAbrExp[1:15,]
763.      currentNewAbrExp[1,]<-c(newFemales,newMales)*sourceSurv[1,]^(1/2)*capAfrSurv[1,]*capAbrSurv[1,]*exportSurv[1,]^(1/10)
764.      BT[kkk,5]<-newFemales+newMales
765.      DT[kkk,5]<-DT[kkk,5]+sum(c(newFemales,newMales)-currentNewAbrExp[1,])
766.
767.      ET[kkk]<-sum(currentNewAbrExp)
768.
769.      ARPT[,,kkk]<-currentSource+currentCaptor+currentDomestics
770.      ADPT[,,kkk]<-currentDomestics
771.      AEPT[,,kkk]<-currentNewAbrExp
772.      ANDPT[,,kkk]<-currentNewDomestics
773.      ANAEPT[,,kkk]<-currentNewAbrExp
774.    }
775.
776.    endPopRatio<-mostRecentPop/sum(ARPT[,,numIter])
777.    defCaptives<-(CT[numIter-1]+CT[numIter])*endPopRatio
778.    defExports<-(ET[numIter-1]+ET[numIter])*endPopRatio
779.    PR<-decExports/defExports
780.    captExpRela<-getCaptExpRela(defaultArray,captureSize,partition,numIter)
781.    CCM<-captExpRela[which.min(abs(decExports/defExports-captExpRela[,2])),1]*currCaptMult
782.    CT<-defCaptives
783.    ET<-defExports
784.
785.    return(list(ARPT=ARPT,ADPT=ADPT,AEPT=AEPT,ANDPT=ANDPT,ANAEPT=ANAEPT,BT=BT,CT=CT,DT=DT,ET=ET,CCM=CCM,PR=PR))
786.  }
787.
788.  getCaptExpRela<-function(defaultArray,capture,partition,numIter){
789.
790.    captRela<-numeric(2*301)
791.    dim(captRela)<-c(301,2)
792.    captRela[,1]<-seq(0,3,by=.01)
793.
794.    defaultArray[1:17,1,3]->sourceFert
795.    defaultArray[1:17,1,4]->captorFert
796.    defaultArray[1:17,1,5]->domestFert
797.    defaultArray[1:17,1,6]->exportFert
798.    defaultArray[1:17,1,7]->newDomFert
799.    defaultArray[1:17,1,8]->newAfrExpFert
800.    defaultArray[1:17,1,9]->newAbrExpFert
801.    defaultArray[,,10]->sourceSurv
802.    defaultArray[,,11]->captorSurv
803.    defaultArray[,,12]->domestSurv
804.    defaultArray[,,13]->exportSurv
805.    defaultArray[,,14]->capAfrSurv
806.    defaultArray[,,15]->capAbrSurv
807.
808.    for(iiii in captRela[,1]){
809.      defaultArray[1:17,,1]->currentSource
810.      defaultArray[1:17,,2]->currentCaptor
811.
812.      captureSize<-iiii*capture
813.
```

Birth, death, export trackers

Definitions
Stores 40 years of simulated data in 5 categories

tot
normalized exports
Minimize difference of $\frac{dec}{def}$ Exp + normalized Exp
This gives best candidate CCM
this lists all the output

function is to get a better number for CCM curr Cap Mult
It tries CCM as ~~300~~ 301 equally spaced numbers from 0 to 3. Try them out, assess them Y or N.

ratio is 301, judgment is 2.

do full simulation per each CCM from 0 to 300.

```
currentSource[,2]<-.95109*currentSource[,2]          ##      modify males according to initial sex ratio = .95109
currentCaptor[,2]<-.95109*currentCaptor[,2]

################################################################################
##  necessary declarations and initializations for the simulation
################################################################################

currentNewDomestics<-numeric(34)
dim(currentNewDomestics)<-c(17,2)
currentDomestics<-numeric(34)
dim(currentDomestics)<-c(17,2)
currentNewAfrExp<-numeric(34)
dim(currentNewAfrExp)<-c(17,2)
currentNewAbrExp<-numeric(34)
dim(currentNewAbrExp)<-c(17,2)
exportTracker<-numeric(2)


################################################################################
##  Begin the first iteration through 5 year periods of time
################################################################################

for (jjjj in 1:numIter){
  previousSource<-currentSource
  currentSource<-previousSource*sourceSurv[2:18,]
  midSource<-(previousSource+currentSource)/2
  captives<-midSource*5*captureSize
  currentSource<-currentSource-captives
  midSource<-(previousSource+currentSource)/2
  births<-midSource[3:11,1]*5*sourceFert[3:11]
  newFemales<-sum(births)
  newMales<-sum(births)*1.03
  currentSource[17,]<-currentSource[16,]+currentSource[17,]
  currentSource[2:16,]<-currentSource[1:15,]
  currentSource[1,]<-c(newFemales,newMales)*sourceSurv[1,]

  previousCaptor<-currentCaptor
  currentCaptor<-previousCaptor*captorSurv[2:18,]
  midCaptor<-(previousCaptor+currentCaptor)/2
  births<-midCaptor[3:11,1]*5*captorFert[3:11]
  newFemales<-sum(births)
  newMales<-sum(births)*1.03
  currentCaptor[17,]<-currentCaptor[16,]+currentCaptor[17,]
  currentCaptor[2:16,]<-currentCaptor[1:15,]
  currentCaptor[1,]<-c(newFemales,newMales)*captorSurv[1,]

  previousNewDomestics<-captives*(1-partition)
  currentNewDomestics<-previousNewDomestics*sourceSurv[2:18,]^.5*capAfrSurv[2:18,]*domestSurv[2:18,]^(3/10)
  midNewDomestics<-(previousNewDomestics+currentNewDomestics)/2
  births<-midNewDomestics[3:11,1]*(2.5*sourceFert[3:11]+1.5*newDomFert[3:11]+1*domestFert[3:11])
  newFemales<-sum(births)
  newMales<-sum(births)*1.03
  currentNewDomestics[17,]<-currentNewDomestics[16,]+currentNewDomestics[17,]
  currentNewDomestics[2:16,]<-currentNewDomestics[1:15,]
  currentNewDomestics[1,]<-c(newFemales,newMales)*sourceSurv[1,]^.5*capAfrSurv[1,]*domestSurv[1,]^(3/10)

  previousDomestics<-currentDomestics
  currentDomestics<-previousDomestics*domestSurv[2:18,]
  midDomestics<-(previousDomestics+currentDomestics)/2
  births<-midDomestics[3:11,1]*5*domestFert[3:11]
  newFemales<-sum(births)
  newMales<-sum(births)*1.03
  currentDomestics[17,]<-currentDomestics[16,]+currentDomestics[17,]
  currentDomestics[2:16,]<-currentDomestics[1:15,]
  currentDomestics[1,]<-c(newFemales,newMales)*domestSurv[1,]
  currentDomestics<-currentDomestics+currentNewDomestics

  previousNewAbrExp<-captives*partition
  currentNewAbrExp<-previousNewAbrExp*sourceSurv[2:18,]^(1/2)*capAfrSurv[2:18,]*capAbrSurv[2:18,]*exportSurv[2:18,]^(1/10)
  midNewAbrExp<-(previousNewAbrExp+currentNewAbrExp)/2
  births<-midNewAbrExp[3:11,1]*(2.5*sourceFert[3:11]+newAfrExpFert[3:11]+newAbrExpFert[3:11]+.5*exportFert[3:11])
  newFemales<-sum(births)
  newMales<-sum(births)*1.03
  currentNewAbrExp[17,]<-currentNewAbrExp[16,]+currentNewAbrExp[17,]
```

jjjj index from jjjj=1 to jjjj=40 for current CCM.

8

```
888.        currentNewAbrExp[2:16,]<-currentNewAbrExp[1:15,]
889.        currentNewAbrExp[1,]<-c(newFemales,newMales)*sourceSurv[1,]^(1/2)*capAfrSurv[1,]*capAbrSurv[1,]*exportSurv[1,]^(1/10)
890.
891.        if(jjjj==(numIter-1)){exportTracker[1]<-sum(currentNewAbrExp)
892.        if(jjjj==(numIter)){exportTracker[2]<-sum(currentNewAbrExp)
893.      }
894.
895.      captRela[which(captRela[,1]==iiii),2]<-sum(exportTracker)
896.    }
897.
898.    captRela[,2]<-captRela[,2]/captRela[which(captRela[,1]==1),2]
899.
900.    maxCapt<-which.max(captRela[,2])
901.    captRela<-captRela[1:maxCapt,]
902.
903.    return(captRela)
904.  }
905.
906.  terrType2<-function(defaultArray,decData,numIter,numDec,firstDec){
907.
908.    ##############################################################################
909.    ##    Initialize aggregate output storage
910.    ##############################################################################
911.    decPopVect<-numeric(numDec)
912.    Male_pop<-numeric(numDec)
913.    Female_pop<-numeric(numDec)
914.    Dome<-numeric(numDec)
915.    Male_dome<-numeric(numDec)
916.    Female_dome<-numeric(numDec)
917.    NewDom<-numeric(numDec)
918.    Male_NewDom<-numeric(numDec)
919.    Female_NewDom<-numeric(numDec)
920.    NewExp<-numeric(numDec)
921.    Male_NewExp<-numeric(numDec)
922.    Female_NewExp<-numeric(numDec)
923.    New_dome_percent<-numeric(numDec)
924.    Export_percent<-numeric(numDec)
925.
926.    ##############################################################################
927.    ##    Initialize territorial output storage
928.    ##############################################################################
929.    allRegPopTracker<-numeric(numDec*17*2*numIter)
930.    dim(allRegPopTracker)<-c(numDec,17,2,numIter)
931.    newBirths<-numeric(numDec)
932.    newDeaths<-numeric(numDec)
933.
934.    ##############################################################################
935.    ##    Convert decData to the appropriate format
936.    ##############################################################################
937.    terr<-as.character(decData[1])
938.    initPop<-as.numeric(decData[8])
939.    decs<-seq(firstDec,firstDec-10*(numDec-1),by=-10)
940.
941.    mostRecentPop<-initPop
942.    for(kkkk in 1:numDec){
943.      callToSim<-simpleSim(defaultArray,numIter)
944.      allRegPopTracker[kkkk,,,]<-callToSim$ARPT
945.      birthTracker<-callToSim$BT
946.      deathTracker<-callToSim$DT
947.      captiveTracker<-callToSim$CT
948.      exportTracker<-callToSim$ET
949.      endPopRatio<-mostRecentPop/sum(allRegPopTracker[kkkk,,,numIter])
950.      allRegPopTracker[kkkk,,,]<-allRegPopTracker[kkkk,,,]*endPopRatio
951.      mostRecentPop<-sum(allRegPopTracker[kkkk,,,numIter-2])
952.      decPopVect[kkkk]<-mostRecentPop
953.      newBirths[kkkk]<-(birthTracker[numIter-1]+birthTracker[numIter])*endPopRatio
954.      newDeaths[kkkk]<-(deathTracker[numIter-1]+deathTracker[numIter])*endPopRatio
955.    }
956.
957.    ##############################################################################
958.    ##    Territorial output
959.    ##############################################################################
960.    forRates<-cbind(initPop,decPopVect)
961.    midPeriodPops<-(forRates[1:(length(forRates)-1)]+forRates[2:length(forRates)])/2
```

second column = sum of all exports (M+F)

this is reference line (assuming CCM=0 i.e. no capture)

second column, all rows.

identify maximum in 2nd column

Then keep from CCM=0 to CCM$_{max}$

yields a smaller matrix

uses Simple Sim

In sum, it's

$$\left(\frac{\text{estimated exports at each CCM}}{\text{reference CCM}}\right)$$

normalized

```
  arpt<-allRegPopTracker[,,,numIter-2]

  for(k in 1:numDec){
    Male_pop[k]<-sum(allRegPopTracker[k,,2,numIter-2])
    Female_pop[k]<-sum(allRegPopTracker[k,,1,numIter-2])

  }

  temp<-data.frame(Region=terr)
  write.table(temp,paste("",terr,".csv",sep=""),row.names = FALSE, col.names = TRUE)
  for(z in 1:numDec){
    temp<-data.frame(Year=decs[z])
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(AdultSexRatio=sum(arpt[z,4:11,2])/sum(arpt[z,3:8,1]))
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(AnnualBirths=newBirths[z]/10)
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(AnnualBirthRate=newBirths[z]/(10*midPeriodPops[z]))
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(AnnualDeaths=newDeaths[z]/10)
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(AnnualDeathRate=c(newDeaths[z]/(10*midPeriodPops[z]),""))
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
  }
  return(list(decPopVect=decPopVect,Male_pop=Male_pop,Female_pop=Female_pop,Dome=Dome,Male_dome=Male_dome,Female_dome=Female_dome,NewDom=NewDom,Male_NewDom=Male_NewDom,Female_NewDom=
}

terrType3<-function(defaultArray,decData,importArray,numIter,numDec,firstDec){

  defaultArray[1:17,1,5]->domestFert
  defaultArray[,,12]->domestSurv

  decPopVect<-numeric(numDec)
  Male_pop<-numeric(numDec)
  Female_pop<-numeric(numDec)
  Dome<-numeric(numDec)
  Male_dome<-numeric(numDec)
  Female_dome<-numeric(numDec)
  NewDom<-numeric(numDec)
  Male_NewDom<-numeric(numDec)
  Female_NewDom<-numeric(numDec)
  NewExp<-numeric(numDec)
  Male_NewExp<-numeric(numDec)
  Female_NewExp<-numeric(numDec)
  New_dome_percent<-numeric(numDec)
  Export_percent<-numeric(numDec)


  ##############################################################################
  ##    Initialize territorial output storage
  ##############################################################################
  allDomPopTracker<-numeric(numDec*17*2)
  dim(allDomPopTracker)<-c(numDec,17,2)
  allRegPopTracker<-numeric(numDec*17*2*numIter)
  dim(allRegPopTracker)<-c(numDec,17,2,numIter)
  newBirths<-numeric(numDec)
  newDeaths<-numeric(numDec)

  ##############################################################################
  ##    Convert decData to the appropriate format
  ##############################################################################
  terr<-as.character(decData[1])
  initPop<-as.numeric(decData[8])
  decs<-seq(firstDec,firstDec-10*(numDec-1),by=-10)

  ##############################################################################
  ##    Calculate Slave Descended Population
  ##############################################################################
  allDomPopTracker[numDec,,]<-importArray[numDec,,]
  for(iiiii in (numDec-1):1){
    for(jjjjj in 1:2){
      if(jjjjj==1){previousDomestics<-allDomPopTracker[iiiii+1,,]}
      if(jjjjj==2){previousDomestics<-currentDomestics}
      currentDomestics<-previousDomestics*domestSurv[2:18,]
```

```
      newDeaths[iiiii+1]<-newDeaths[iiiii+1]+sum(previousDomestics-currentDomestics)
      midDomestics<-(previousDomestics+currentDomestics)/2
      births<-midDomestics[3:11,1]*5*domestFert[3:11]
      newFemales<-sum(births)
      newMales<-sum(births)*1.03
      currentDomestics[17,]<-currentDomestics[16,]+currentDomestics[17,]
      currentDomestics[2:16,]<-currentDomestics[1:15,]
      currentDomestics[1,]<-c(newFemales,newMales)*domestSurv[1,]
      newBirths[iiiii+1]<-newBirths[iiiii+1]+newFemales+newMales
      newDeaths[iiiii+1]<-newDeaths[iiiii+1]+sum(c(newFemales,newMales)-currentDomestics[1,])
    }
    allDomPopTracker[iiiii,,]<-importArray[iiiii,,]+currentDomestics
  }

  ##############################################################################
  ##    Calculate Final Slave Descended Population
  ##############################################################################
  finalDomPop<-numeric(17*2)
  finalDomPop<-c(17,2)
  for(kkkkk in 1:2){
    if(kkkkk==1){previousDomestics<-allDomPopTracker[1,,]}
    if(kkkkk==2){previousDomestics<-currentDomestics}
    currentDomestics<-previousDomestics*domestSurv[2:18,]
    newDeaths[1]<-newDeaths[1]+sum(previousDomestics-currentDomestics)
    midDomestics<-(previousDomestics+currentDomestics)/2
    births<-midDomestics[3:11,1]*5*domestFert[3:11]
    newFemales<-sum(births)
    newMales<-sum(births)*1.03
    currentDomestics[17,]<-currentDomestics[16,]+currentDomestics[17,]
    currentDomestics[2:16,]<-currentDomestics[1:15,]
    currentDomestics[1,]<-c(newFemales,newMales)*domestSurv[1,]
    newBirths[1]<-newBirths[1]+newFemales+newMales
    newDeaths[1]<-newDeaths[1]+newDeaths[1]+sum(c(newFemales,newMales)-currentDomestics[1,])
    finalDomPop<-finalDomPop+currentDomestics
  }

  mostRecentPop<-initPop-sum(finalDomPop)
  for(iiiiii in 1:numDec){
    callToSim<-simpleSim(defaultArray,numIter)
    allRegPopTracker[iiiiii,,,]<-callToSim$ARPT
    birthTracker<-callToSim$BT
    deathTracker<-callToSim$DT
    endPopRatio<-mostRecentPop/sum(allRegPopTracker[iiiiii,,,numIter])
    allRegPopTracker[iiiiii,,,]<-allRegPopTracker[iiiiii,,,]*endPopRatio
    mostRecentPop<-sum(allRegPopTracker[iiiiii,,,numIter-2])
    decPopVect[iiiiii]<-sum(allRegPopTracker[iiiiii,,,numIter-2])+sum(allDomPopTracker[iiiiii,,])
    newBirths[iiiiii]<-newBirths[iiiiii]+(birthTracker[numIter-1]+birthTracker[numIter])*endPopRatio
    newDeaths[iiiiii]<-newDeaths[iiiiii]+(deathTracker[numIter-1]+deathTracker[numIter])*endPopRatio
  }

  forRates<-cbind(initPop,decPopVect)
  midPeriodPops<-(forRates[1:(length(forRates)-1)]+forRates[2:length(forRates)])/2
  arpt<-allRegPopTracker[,,,numIter-2]+allDomPopTracker
  adpt<-allDomPopTracker

  for (k in 1:numDec){
    Male_pop[k]<-sum(arpt[k,,2])
    Female_pop[k]<-sum(arpt[k,,1])
    Dome[k]<-sum(allDomPopTracker[k,,])
    Male_dome[k]<-sum(allDomPopTracker[k,,2])
    Female_dome[k]<-sum(allDomPopTracker[k,,1])


  }

  ##############################################################################
  ##    Territorial output
  ##############################################################################
  forRates<-cbind(initPop,decPopVect)
  midPeriodPops<-(forRates[1:(length(forRates)-1)]+forRates[2:length(forRates)])/2
  arpt<-allRegPopTracker[,,,numIter-2]+allDomPopTracker
  adpt<-allDomPopTracker

  temp<-data.frame(Region=terr)
```

```
  write.table(temp,paste("",terr,".csv",sep=""),row.names = FALSE, col.names = TRUE)
  for(zz in 1:numDec){
    temp<-data.frame(Year=decs[zz])
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(AdultSexRatio=c(sum(arpt[zz,4:11,2])/sum(arpt[zz,3:8,1]),"DomesticSlavePopulation"))
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(Female=adpt[zz,,1],Male=adpt[zz,,2])
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE,sep = ",")
    temp<-data.frame(TotalFemaleSlaves=sum(adpt[zz,,1]))
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(TotalMaleSlaves=sum(adpt[zz,,2]))
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(TotalDomesticSlaves=sum(adpt[zz,,]))
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(AnnualBirths=newBirths[zz]/10)
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(AnnualBirthRate=newBirths[zz]/(10*midPeriodPops[zz]))
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(AnnualDeaths=newDeaths[zz]/10)
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(AnnualDeathRate=c(newDeaths[zz]/(10*midPeriodPops[zz]),""))
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
  }

  return(list(decPopVect=decPopVect,Male_pop=Male_pop,Female_pop=Female_pop,Dome=Dome,Male_dome=Male_dome,Female_dome=Female_dome,NewDom=NewDom,Male_NewDom=Male_NewDom,Female_NewDom=
}

terrType4<-function(defaultArray,importArray,terr, numDec,firstDec){

  defaultArray[1:17,1,5]->domestFert
  defaultArray[,,12]->domestSurv

  ##############################################################################
  ##    Initialize territorial output storage
  ##############################################################################
  allDomPopTracker<-numeric(numDec*17*2)
  dim(allDomPopTracker)<-c(numDec,17,2)

  ##############################################################################
  ##    Calculate Slave Descended Population
  ##############################################################################
  allDomPopTracker[numDec,,]<-importArray[numDec,,]
  for(jjjjjj in (numDec-1):1){
    for(kkkkkk in 1:2){
      if(kkkkkk==1){previousDomestics<-allDomPopTracker[jjjjjj+1,,]}
      if(kkkkkk==2){previousDomestics<-currentDomestics}
      currentDomestics<-previousDomestics*domestSurv[2:18,]
      midDomestics<-(previousDomestics+currentDomestics)/2
      births<-midDomestics[3:11,1]*5*domestFert[3:11]
      newFemales<-sum(births)
      newMales<-sum(births)*1.03
      currentDomestics[17,]<-currentDomestics[16,]+currentDomestics[17,]
      currentDomestics[2:16,]<-currentDomestics[1:15,]
      currentDomestics[1,]<-c(newFemales,newMales)*domestSurv[1,]
    }
    allDomPopTracker[jjjjjj,,]<-importArray[jjjjjj,,]+currentDomestics
  }

  ##############################################################################
  ##    Territorial output
  ##############################################################################
  adpt<-allDomPopTracker
  #terr<-"Americas"
  decs<-seq(firstDec,firstDec-10*(numDec-1),by=-10)
  temp<-data.frame(Region=terr)
  write.table(temp,paste("",terr,".csv",sep=""),row.names = FALSE, col.names = TRUE)
  for(zzz in 1:numDec){
    temp<-data.frame(Year=c(decs[zzz],"DomesticSlavePopulation"))
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(Female=adpt[zzz,,1],Male=adpt[zzz,,2])
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE,sep = ",")
    temp<-data.frame(TotalFemaleSlaves=sum(adpt[zzz,,1]))
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(TotalMaleSlaves=sum(adpt[zzz,,2]))
```

```
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
    temp<-data.frame(TotalDomesticSlaves=sum(adpt[zzz,,]))
    write.table(temp,paste("",terr,".csv",sep=""),append=TRUE, row.names = FALSE, col.names = TRUE)
  }

}

newBPMain_multiple_model1<-function(inFile1,inFile2,outFile1,outFile2,outFile3,outFile4,outFile5,outFile6,outFile7,outFile8,outFile9,outFile10,outFile11,outFile12,outFile13,outFile14

  ##############################################################################
  ##    inFile is a character string naming the(path and) file
  ##    where input is stored, e.g. "fileName.csv"
  ##
  ##    outFile is a character string naming the(path and) file
  ##    where output is stored, e.g. "fileName.csv"
  ##
  ##    numIter is the number of 5 year iterations conducted
  ##    at each stage of the simulation
  ##
  ##    numDec is the number of decades to project back
  ##
  ##    firstDec is the first decade for which we want a back
  ##    projected population
  ##############################################################################

  ##############################################################################
  ##    Read in default fertility, population, partition, capture, and
  ##    survival arrays
  ##    -fertility files are in the usual 17x1 format
  ##    -population, partition, and captureSize files are in 34x1 format
  ##    -survival files are in 36x1 format;
  ##    1st entry = 0-1
  ##    2nd entry = 1-5
  ##    3rd entry = 5-10 ...
  ##    18th entry= 80+ (we are allowing one to gain entry to the next 80+ group either
  ##    by being 80+ and surviving the current period, or by being 75-80 and surviving
  ##    the current period) (note that capAfr and capAbr are 1 year rates, all other
  ##    survival rates are 5 year)
  ##############################################################################

  defaultArray<-windDefault()
  partitionType1<-as.matrix(read.table(file="pcf24m49.dat",skip=1))
  dim(partitionType1)<-c(17,2)
  partitionType2<-as.matrix(read.table(file="pcf27m14.dat",skip=1))
  dim(partitionType2)<-c(17,2)


  ##############################################################################
  ##    Convert data in inFile to a data frame
  ##############################################################################

  decData<-read.csv(inFile1,header=TRUE,as.is=c(1,2))
  decData<-data.frame(decData)
  attach(decData)

  slaveTradeMatrix<-read.csv(inFile2,as.is=c(1))
  slaveTradeMatrix<-as.matrix(slaveTradeMatrix[,2:length(slaveTradeMatrix[1,])])
  slaveTradeMatrix<-data.frame(slaveTradeMatrix)

  ##############################################################################
  ##    Create arrays in which all calculated values are stored
  ##############################################################################

  exportArray<-numeric(sum(TerritoryType==1)*numDec*17*2)
  dim(exportArray)<-c(sum(TerritoryType==1),numDec,17,2)

  decPopArray<-numeric(length(decData[,1])*numDec)
  dim(decPopArray)<-c(length(decData[,1]),numDec)

  Male_pop<-numeric(length(decData[,1])*numDec)
  dim(Male_pop)<-c(length(decData[,1]),numDec)

  Female_pop<-numeric(length(decData[,1])*numDec)
  dim(Female_pop)<-c(length(decData[,1]),numDec)
```

```
1258.
1259.   Dome<-numeric(length(decData[,1])*numDec)
1260.   dim(Dome)<-c(length(decData[,1]),numDec)
1261.
1262.   Male_dome<-numeric(length(decData[,1])*numDec)
1263.   dim(Male_dome)<-c(length(decData[,1]),numDec)
1264.
1265.   Female_dome<-numeric(length(decData[,1])*numDec)
1266.   dim(Female_dome)<-c(length(decData[,1]),numDec)
1267.
1268.   NewDom<-numeric(length(decData[,1])*numDec)
1269.   dim(NewDom)<-c(length(decData[,1]),numDec)
1270.
1271.   Male_NewDom<-numeric(length(decData[,1])*numDec)
1272.   dim(Male_NewDom)<-c(length(decData[,1]),numDec)
1273.
1274.   Female_NewDom<-numeric(length(decData[,1])*numDec)
1275.   dim(Female_NewDom)<-c(length(decData[,1]),numDec)
1276.
1277.   NewExp<-numeric(length(decData[,1])*numDec)
1278.   dim(NewExp)<-c(length(decData[,1]),numDec)
1279.
1280.   Male_NewExp<-numeric(length(decData[,1])*numDec)
1281.   dim(Male_NewExp)<-c(length(decData[,1]),numDec)
1282.
1283.   Female_NewExp<-numeric(length(decData[,1])*numDec)
1284.   dim(Female_NewExp)<-c(length(decData[,1]),numDec)
1285.
1286.   New_dome_percent<-numeric(length(decData[,1])*numDec)
1287.   dim(New_dome_percent)<-c(length(decData[,1]),numDec)
1288.
1289.   Export_percent<-numeric(length(decData[,1])*numDec)
1290.   dim(Export_percent)<-c(length(decData[,1]),numDec)
1291.
1292.
1293.
1294.   ##############################################################################
1295.   ##    Calls to the terrType functions fill the above output arrays and also
1296.   ##    generate individual territorial output files
1297.   ##    terrType1: no imports, exports, source and captor pop (Senegal-Mozambique)
1298.   ##    terrType2: no imports, no exports, captor pop only (Madagascar,sAfrica)
1299.   ##    terrType3: imports, no exports, captor and domestic pop (nAfrica)
1300.   ##    terrType4: imports, no exports, domestic pop only (America)
1301.   ##############################################################################
1302.
1303.   terr1part1<-which((TerritoryType==1)&(PartType==1))
1304.
1305.   for(i in terr1part1){
1306.     temp<-terrType1(defaultArray,partitionType1,decData[i,],numIter,numDec,firstDec)
1307.     exportArray[i,,,]<-temp$exportArray
1308.     decPopArray[i,]<-temp$decPopVect
1309.     Male_pop[i,]<-temp$Male_pop
1310.     Female_pop[i,]<-temp$Female_pop
1311.     Dome[i,]<-temp$Dome
1312.     Male_dome[i,]<-temp$Male_dome
1313.     Female_dome[i,]<-temp$Female_dome
1314.     NewDom[i,]<-temp$NewDom
1315.     Male_NewDom[i,]<-temp$Male_NewDom
1316.     Female_NewDom[i,]<-temp$Female_NewDom
1317.     NewExp[i,]<-temp$NewExp
1318.     Male_NewExp[i,]<-temp$Male_NewExp
1319.     Female_NewExp[i,]<-temp$Female_NewExp
1320.     New_dome_percent[i,]<-temp$New_dome_percent
1321.     Export_percent[i,]<-temp$Export_percent
1322.   }
1323.
1324.   terr1part2<-which((TerritoryType==1)&(PartType==2))
1325.   for(i in terr1part2){
1326.     temp<-terrType1(defaultArray,partitionType2,decData[i,],numIter,numDec,firstDec)
1327.     exportArray[i,,,]<-temp$exportArray
1328.     decPopArray[i,]<-temp$decPopVect
1329.     Male_pop[i,]<-temp$Male_pop
1330.     Female_pop[i,]<-temp$Female_pop
1331.     Dome[i,]<-temp$Dome
```

terr 1 part 1 W Africa

for the regions one by one.

East Africa

```
1332.     Male_dome[i,]<-temp$Male_dome
1333.     Female_dome[i,]<-temp$Female_dome
1334.     NewDom[i,]<-temp$NewDom
1335.     Male_NewDom[i,]<-temp$Male_NewDom
1336.     Female_NewDom[i,]<-temp$Female_NewDom
1337.     NewExp[i,]<-temp$NewExp
1338.     Male_NewExp[i,]<-temp$Male_NewExp
1339.     Female_NewExp[i,]<-temp$Female_NewExp
1340.     New_dome_percent[i,]<-temp$New_dome_percent
1341.     Export_percent[i,]<-temp$Export_percent
1342.   }
1343.
1344.   terr2<-which(TerritoryType==2)
1345.   for(i in terr2){
1346.     temp<-terrType2(defaultArray,decData[i,],numIter,numDec,firstDec)
1347.     decPopArray[i,]<-temp$decPopVect
1348.     Male_pop[i,]<-temp$Male_pop
1349.     Female_pop[i,]<-temp$Female_pop
1350.     Dome[i,]<-temp$Dome
1351.     Male_dome[i,]<-temp$Male_dome
1352.     Female_dome[i,]<-temp$Female_dome
1353.     NewDom[i,]<-temp$NewDom
1354.     Male_NewDom[i,]<-temp$Male_NewDom
1355.     Female_NewDom[i,]<-temp$Female_NewDom
1356.     NewExp[i,]<-temp$NewExp
1357.     Male_NewExp[i,]<-temp$Male_NewExp
1358.     Female_NewExp[i,]<-temp$Female_NewExp
1359.     New_dome_percent[i,]<-temp$New_dome_percent
1360.     Export_percent[i,]<-temp$Export_percent
1361.
1362.
1363.   }
1364.
1365.
1366.   ##############################################################################
1367.   ##    Tally imports by territory
1368.   ##############################################################################
1369.
1370.
1371.   attach(slaveTradeMatrix)
1372.   importArray<-numeric((sum(TerritoryType==3)+2)*numDec*17*2)
1373.   dim(importArray)<-c((sum(TerritoryType==3)+2),numDec,17,2)
1374.   for(i in 1:(sum(TerritoryType==3)+2)){
1375.     importArray[i,,,]<-apply(slaveTradeMatrix[,i]*exportArray,c(2,3,4),sum)
1376.   }
1377.
1378.   terr3<-which(TerritoryType==3)
1379.   for(i in terr3){
1380.     temp<-terrType3(defaultArray,decData[i,],importArray[which(terr3==i),,,],numIter,numDec,firstDec)
1381.     decPopArray[i,]<-temp$decPopVect
1382.     Male_pop[i,]<-temp$Male_pop
1383.     Female_pop[i,]<-temp$Female_pop
1384.     Dome[i,]<-temp$Dome
1385.     Male_dome[i,]<-temp$Male_dome
1386.     Female_dome[i,]<-temp$Female_dome
1387.     NewDom[i,]<-temp$NewDom
1388.     Male_NewDom[i,]<-temp$Male_NewDom
1389.     Female_NewDom[i,]<-temp$Female_NewDom
1390.     NewExp[i,]<-temp$NewExp
1391.     Male_NewExp[i,]<-temp$Male_NewExp
1392.     Female_NewExp[i,]<-temp$Female_NewExp
1393.     New_dome_percent[i,]<-temp$New_dome_percent
1394.     Export_percent[i,]<-temp$Export_percent
1395.
1396.
1397.   }
1398.   k<-length(importArray[,1,1,1])
1399.   terr <- c("Americas","Indian Ocean")
1400.   for(i in 1:2){
1401.     terrType4(defaultArray,importArray[k-2+i,,,],terr[i],numDec,firstDec)
1402.   }
1403.   ##############################################################################
1404.   ##    Begin aggregate output
1405.   ##############################################################################
```

shows which regions are in terrType 1, 2, 3, 4

```
1406.
1407.    decs<-seq(firstDec,firstDec-(10*(numDec-1)),by=-10)
1408.    colnames(decPopArray)<-as.character(decs)
1409.    colnames(Male_pop)<-as.character(decs)
1410.    colnames(Female_pop)<-as.character(decs)
1411.    colnames(Dome)<-as.character(decs)
1412.    colnames(Male_dome)<-as.character(decs)
1413.    colnames(Female_dome)<-as.character(decs)
1414.    colnames(NewDom)<-as.character(decs)
1415.    colnames(Male_NewDom)<-as.character(decs)
1416.    colnames(Female_NewDom)<-as.character(decs)
1417.    colnames(NewExp)<-as.character(decs)
1418.    colnames(Male_NewExp)<-as.character(decs)
1419.    colnames(Female_NewExp)<-as.character(decs)
1420.    colnames(New_dome_percent)<-as.character(decs)
1421.    colnames(Export_percent)<-as.character(decs)
1422.
1423.
1424.    terrMat<-cbind(Territory,Region,decPopArray)
1425.    terrMat1<-cbind(Territory,Region,Male_pop)
1426.    terrMat2<-cbind(Territory,Region,Female_pop)
1427.    terrMat3<-cbind(Territory,Region,Dome)
1428.    terrMat4<-cbind(Territory,Region,Male_dome)
1429.    terrMat5<-cbind(Territory,Region,Female_dome)
1430.    terrMat6<-cbind(Territory,Region,NewDom)
1431.    terrMat7<-cbind(Territory,Region,Male_NewDom)
1432.    terrMat8<-cbind(Territory,Region,Female_NewDom)
1433.    terrMat9<-cbind(Territory,Region,NewExp)
1434.    terrMat10<-cbind(Territory,Region,Male_NewExp)
1435.    terrMat11<-cbind(Territory,Region,Female_NewExp)
1436.    terrMat12<-cbind(Territory,Region,New_dome_percent)
1437.    terrMat13<-cbind(Territory,Region,Export_percent)
1438.
1439.
1440.
1441.
1442.
1443.    write.table(terrMat,outFile1,row.names=FALSE,col.names = TRUE, sep = ",")
1444.
1445.    write.table(terrMat1,outFile2,row.names=FALSE,col.names = TRUE, sep = ",")
1446.
1447.    write.table(terrMat2,outFile3,row.names=FALSE,col.names = TRUE, sep = ",")
1448.
1449.    write.table(terrMat3,outFile4,row.names=FALSE,col.names = TRUE, sep = ",")
1450.
1451.    write.table(terrMat4,outFile5,row.names=FALSE,col.names = TRUE, sep = ",")
1452.
1453.    write.table(terrMat5,outFile6,row.names=FALSE,col.names = TRUE, sep = ",")
1454.
1455.    write.table(terrMat6,outFile7,row.names=FALSE,col.names = TRUE, sep = ",")
1456.
1457.    write.table(terrMat7,outFile8,row.names=FALSE,col.names = TRUE, sep = ",")
1458.
1459.    write.table(terrMat8,outFile9,row.names=FALSE,col.names = TRUE, sep = ",")
1460.
1461.    write.table(terrMat9,outFile10,row.names=FALSE,col.names = TRUE, sep = ",")
1462.
1463.    write.table(terrMat10,outFile11,row.names=FALSE,col.names = TRUE, sep = ",")
1464.
1465.    write.table(terrMat11,outFile12,row.names=FALSE,col.names = TRUE, sep = ",")
1466.
1467.    write.table(terrMat12,outFile13,row.names=FALSE,col.names = TRUE, sep = ",")
1468.
1469.    write.table(terrMat13,outFile14,row.names=FALSE,col.names = TRUE, sep = ",")
1470.
1471.
1472.
1473.
1474.
1475. }
1476.
1477. #Back to 1650
1478.
1479.
```

```
1480. newBPMain_multiple_model1('inFileBeg1890_3.csv','slaveTradeMatrix.csv','Total_pop.csv','Men_pop.csv','Women_pop.csv','Domestic.csv','Men_Domestic.csv','Women_Domestic.csv','New_Domes
1481.
1482.
1483.
1484.
1485.
1486.
```