```
require(phydynR)
source('~/git/phydynR/R/phylo.source.attribution.R')
source('model0.R')
#~ require(data.table)
#~ require(doMPI)
#~ require(phangorn)

MH <- 10 # look up to 10 years in past for infector probs

########################


#~ o <- ode(y=y0, times=times_day, func=dydt, parms=list()  , method = 'euler')
o <- ode(y=y0, times=times_day, func=dydt, parms=list()  , method = 'adams')
tfgy <- .tfgy( o )
sampleTimes <- scan( file = 'sampleTimes' )
ss  <- matrix( scan( file = 'sampleStates' ) , byrow=TRUE, ncol = m)
colnames(ss) <- DEMES
# regularise
ss <- ss + 1e-4
ss = sampleStates <- ss / rowSums(ss)

## sim tree
if (F){
print('sim tree')
print(date())
st.tree <- system.time( {
    tree <- sim.co.tree.fgy(tfgy,  sampleTimes, sampleStates)
})
save(tree, file = 'phydynR-testSA0-tree.RData')
print(date())
} else{ # load the tree from file
    load('phydynR-testSA0-tree.RData' )
}

## cd4s & ehis
    # from cori paper
    #~ k =1: CD4>=500
    #~ k = 2 : 350<=CD4<500.
    #~ k = 3: 200<=CD4<350
    #~ k = 4 : CD4<200
cd4s <- setNames( sapply( 1:nrow( tree$sampleStates), function(k){
    deme <- DEMES[ which.max(tree$sampleStates[k,] )  ]
    stage <- strsplit( deme, '.' , fixed=T)[[1]][1]
    stage <- as.numeric( tail( strsplit(stage, '')[[1]], 1 ) )
    if (stage==1) return(1e3)
    if (stage==2) return(750)
    if (stage==3) return(400)
    if (stage==4) return(300)
    if (stage==5) return(100)
}), tree$tip.label)
ehis <-  setNames( sapply( 1:nrow( tree$sampleStates), function(k){
    deme <- DEMES[ which.max(tree$sampleStates[k,] )  ]
```

```
    stage <- strsplit( deme, '.' , fixed=T)[[1]][1]
    stage <- as.numeric( tail( strsplit(stage, '')[[1]], 1 ) )
    ifelse( stage==1, TRUE, FALSE)
}), tree$tip.label)

############################
#~ incidence and prevalence
yfin <- tfgy[[4]][[length(times_day)]]
ffin <- tfgy[[2]][[length(times_day)]]
newinf <- sum(ffin[1:120, 1:120] ) * 365
plwhiv <- sum( yfin[-length(yfin)] )

# rescale tree
sampleTimes <- days2years( tree$sampleTimes )
tree$edge.length <- tree$edge.length / 365
#~ bdt <- DatedTree(  tree, sampleTimes, tree$sampleStates, tol = Inf )
bdt <- DatedTree(  tree, sampleTimes, tree$sampleStates, tol = .1 )

n<- bdt$n
sampleDemes <- setNames( sapply( 1:n, function(u) DEMES[which.max( tree$sampleStates[u,])] ), tree$tip.l
#~ bdt$sampleDemes <- setNames( sapply( 1:n, function(u) DEMES[which.max( bdt$sampleStates[u,])] ), bdt


st.W <- system.time( {
    W <- phylo.source.attribution.hiv( bdt
      , bdt$sampleTimes # must use years
      , cd4s = cd4s[bdt$tip.label] # named numeric vector, cd4 at time of sampling
      , ehi = ehis[bdt$tip.label] # named logical vector, may be NA, TRUE if patient sampled with early
      , numberPeopleLivingWithHIV  = plwhiv# scalar
      , numberNewInfectionsPerYear = newinf # scalar
      , maxHeight = MH
      , res = 1e3
      , treeErrorTol = Inf
    )
})
```

```
## [1] "NOTE : sample times must be in units of years"
## [1] "start source attrib"
## [1] "Mon Mar 21 18:13:44 2016"
## [1] "source attrib complete"
## [1] "Mon Mar 21 18:13:57 2016"
```

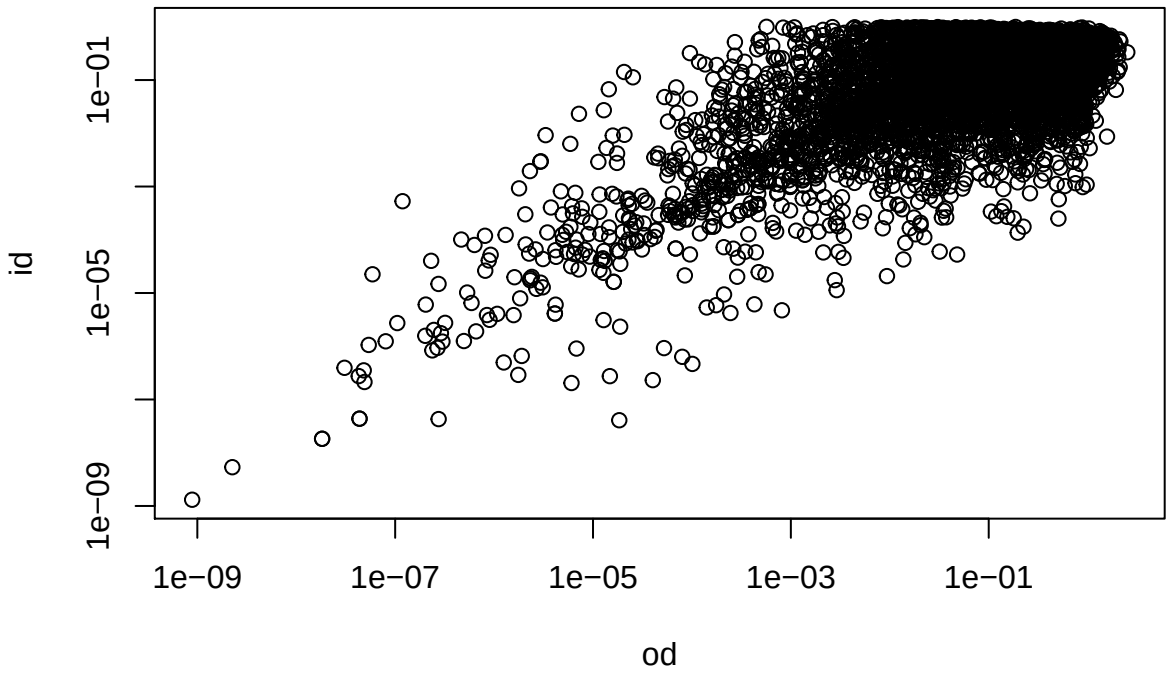
```
Ws <- list(W )


wsids <- unique( W$donor )

wvec <- W$infectorProbability
wvec_o <- order( wvec )
wvec <- wvec[wvec_o]
print ('sum w')
```

```
## [1] "sum w"
```

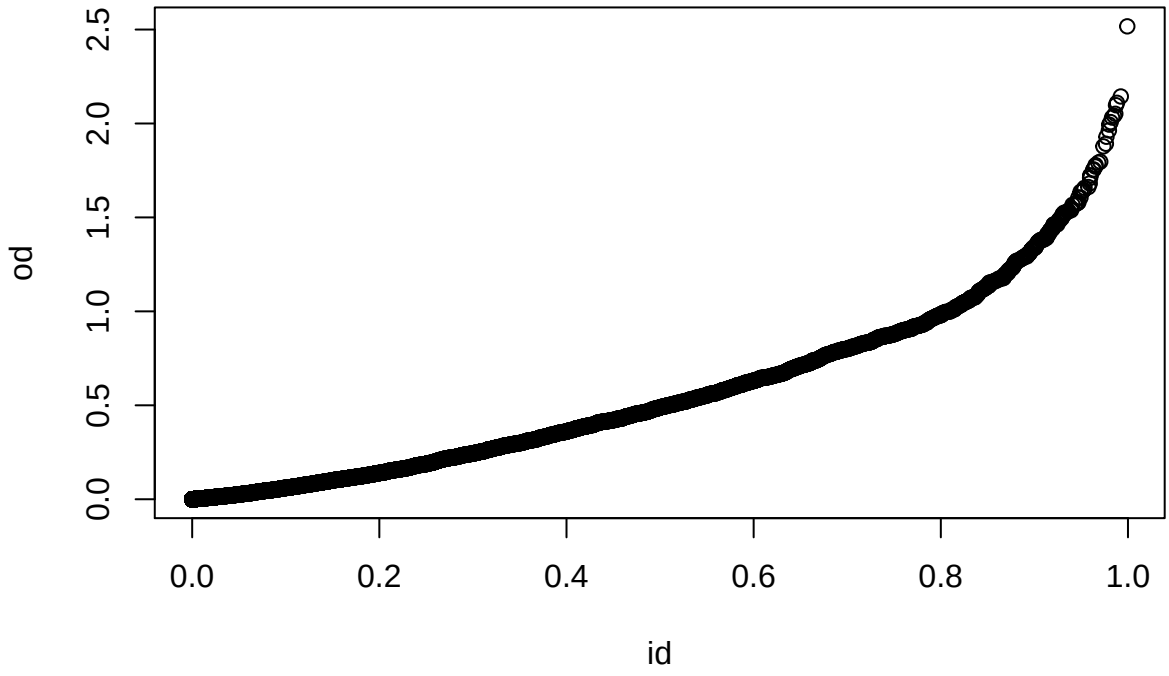
```
print(sum(wvec) / length(Ws))
```

```
## [1] 2041.778
```

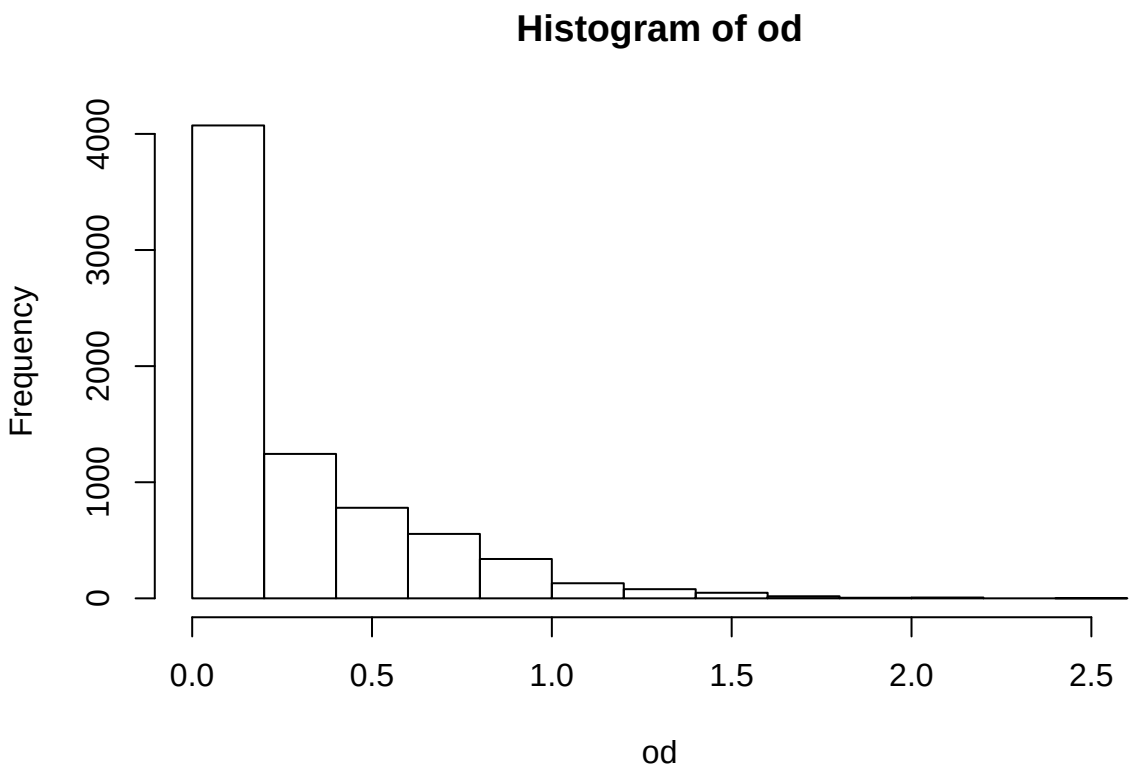
```
## out degree & indegree
od <- sapply( wsids, function(sid) sum( wvec[W$donor[wvec_o]==sid]  ) )
id <- sapply( wsids, function(sid) sum( wvec[W$recip[wvec_o]==sid]  ) )
 plot( od, id, log = 'xy' )
```



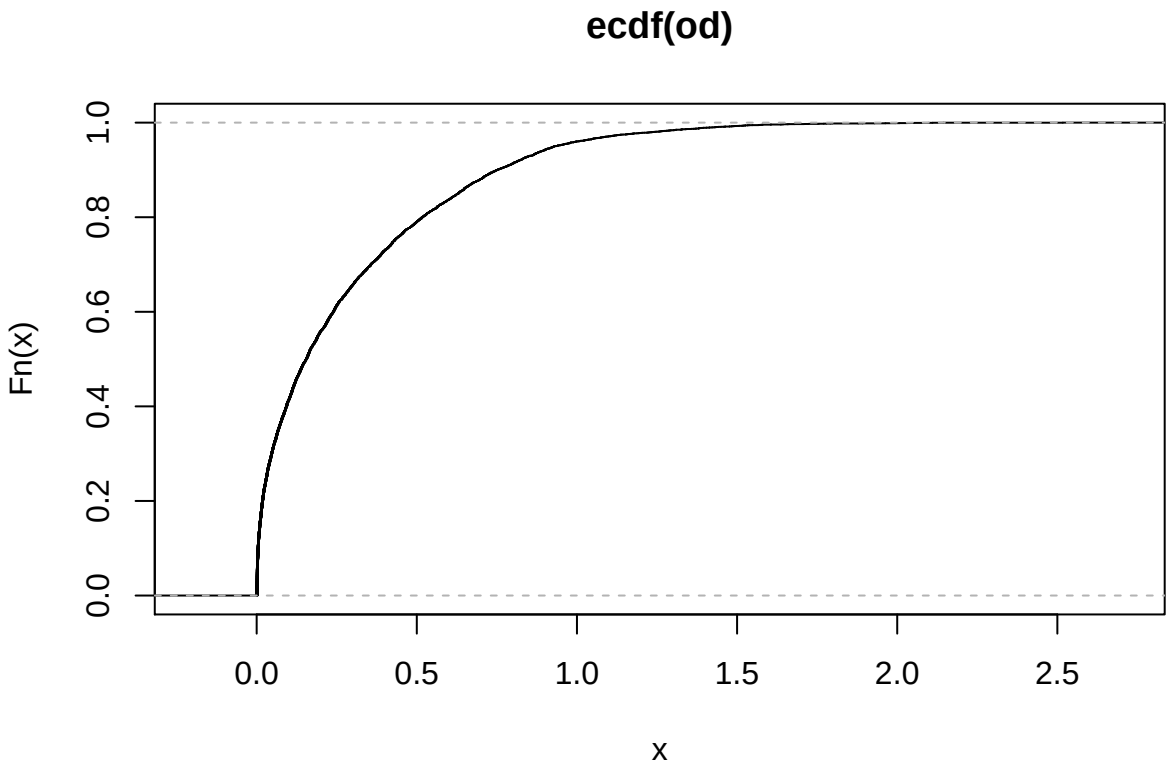

```
qqplot( id, od )
```



```
hist( od )
```



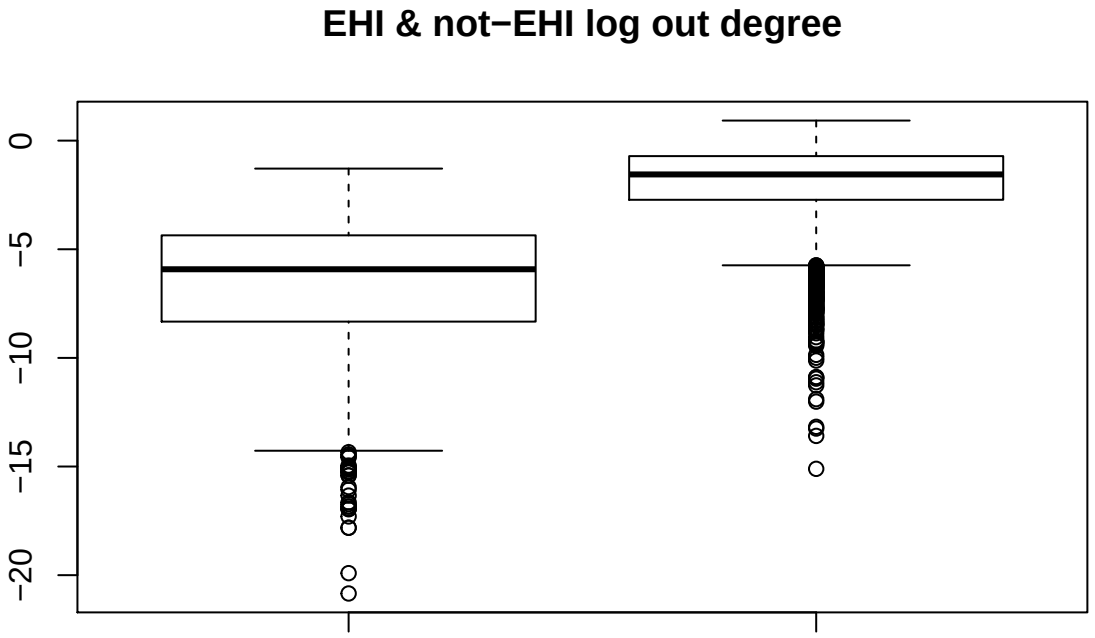

```
plot( ecdf( od ))
```



```
## out degree by stage
acute_sids <- names( ehis )[ which(ehis==TRUE ) ]
nr_sids <-  setdiff( names(ehis), acute_sids )
acute_ods <- od[ wsids %in% acute_sids ]
nr_ods <- od[ wsids %in% nr_sids ]
print( wilcox.test( acute_ods, nr_ods ) )
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  acute_ods and nr_ods
## W = 383390, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
```

```
  boxplot( log( acute_ods), log ( nr_ods ), main = 'EHI & not-EHI log out degree')
```




```
#~   boxplot( ( acute_ods), ( nr_ods ), main = 'EHI & not-EHI out degree')
print ( summary(acute_ods) )
```

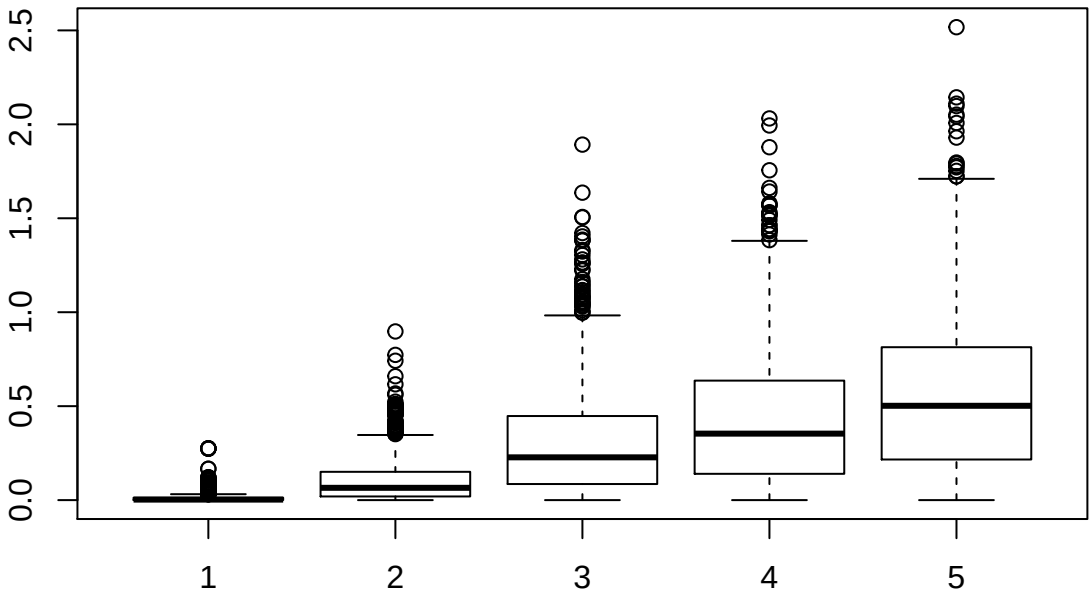
```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000000 0.0002412 0.0026840 0.0126800 0.0127900 0.2761000
```

```
print ( summary(nr_ods) )
```

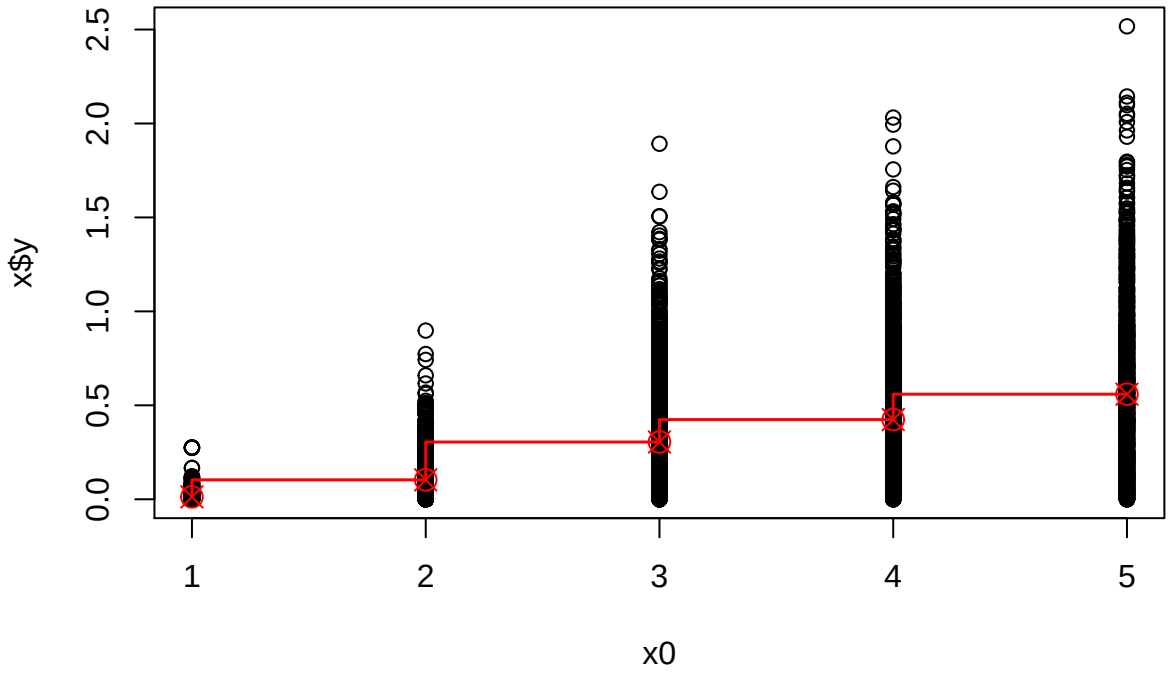
```
##      Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
## 0.0000003 0.0656600 0.2111000 0.3243000 0.4904000 2.5170000
```

```
## isotonic regression stage
.cd42stage <- function(cd4)
{
```

```
    # from cori paper
    #~ k =1: CD4>=500
    #~ k = 2 : 350<=CD4<500.
    #~ k = 3: 200<=CD4<350
    #~ k = 4 : CD4<200
    if (is.na(cd4) ) return (NA)
    if (cd4  >= 500) return (2)
    if (cd4 >= 350) return(3)
    if (cd4 >= 200 ) return(4)
    return(5)
}
cd4s <- cd4s[bdt$tip.label]
ehis <- ehis[bdt$tip.label]
stages <- sapply( cd4s, .cd42stage )
for (i in 1:length(ehis)){
    if (!is.na(ehis[i])){
        if (ehis[i]) stages[i] <- 1
    }
}
od_by_stage <- lapply( 1:5, function(stage) od[names(od) %in% names(stages[stages==stage])] )
boxplot(  od_by_stage , main = 'out degree by stage')
```




```
ir_stage <- isoreg( stages[names(od)] , od )
 plot( ir_stage  )
```




```
print( 'cor( stages[names(od)] , od  )' )
```

```
## [1] "cor( stages[names(od)] , od  )"
```

```
print( cor( stages[names(od)] , od  ) )
```

```
## [1] 0.5543021
```

```
tr0 <- min( ir_stage$yf ) #mean (od_by_stage[[1]] ) / .5
tr1 <- max( ir_stage$yf )
paf0 <- tr0 / tr1
print( summary( lm( scale(od) ~ scale(stages[names(od)]) ) ) )
```

```
##
## Call:
## lm(formula = scale(od) ~ scale(stages[names(od)]))
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7219 -0.4332 -0.0540  0.2821  5.9176
##
## Coefficients:
##                             Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -9.883e-17  9.756e-03    0.00        1
## scale(stages[names(od)])  5.543e-01  9.757e-03   56.81   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 0.8324 on 7277 degrees of freedom
## Multiple R-squared:  0.3073, Adjusted R-squared:  0.3072
## F-statistic:  3228 on 1 and 7277 DF,  p-value: < 2.2e-16
```

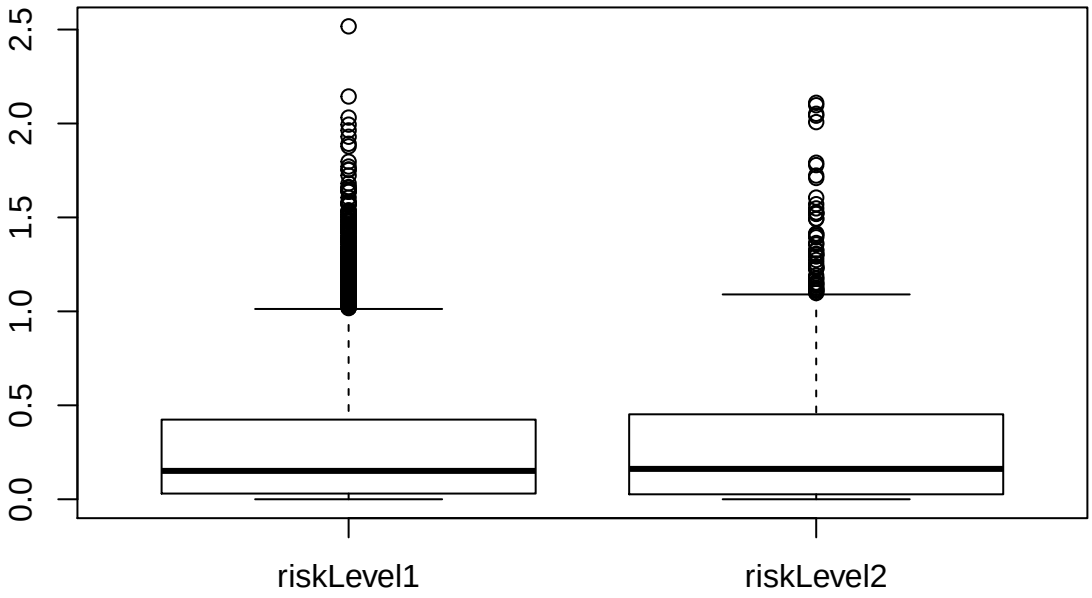
```
truepaf0 <- {
    FF <- tfgy[[2]][[1e3]]
    sum( FF[NH_COORDS$stage1, ] ) / sum(FF[1:120,1:120] )
}

## now see what sort of pattern there is in od defined by threshold distance
if (F)
{
    threshold <- 0.015
    diag(D) <- Inf
    od_distance <- setNames( sapply( 1:nrow(D), function(i) sum( D[i, ] < threshold ) ), rownames(D) )
    od_distance_by_stage <- lapply( 1:5, function(stage) od_distance[names(od_distance) %in% names(stag
    boxplot( od_distance_by_stage )
    od_distance0 <- od_distance[names(od_distance) %in% names(stages)]
    x <- stages[names(od_distance0)]
    y <- od_distance0
    #~ plot( isoreg( x[!is.na(x)] , y[!is.na(x)] ) )
    print(' cor( x[!is.na(x)] , y[!is.na(x)] ) ')
    print( cor( x[!is.na(x)] , y[!is.na(x)] ) )
    print( summary( lm( scale(od_distance0) ~ scale(stages[names(od_distance0)]) ) ) )
}


###################################
#~ assoc of risk level & outdegree
rl_ids <- c( 'riskLevel1', 'riskLevel2' )
rl_sids <- setNames( lapply( rl_ids, function(rlid) grepl( rlid, sampleDemes[ names(od) ] ) ), rl_ids )
od_rl <- setNames( lapply( rl_ids, function(rlid) od[ rl_sids[[rlid]] ] ), rl_ids )

rownames(FF) = colnames(FF) <- DEMES
sum( FF[ grepl( 'riskLevel1', DEMES ), ] )
```

```
## [1] 0.6983875
```

```
sum( FF[ grepl( 'riskLevel2', DEMES ), ] )
```

```
## [1] 1.746063
```

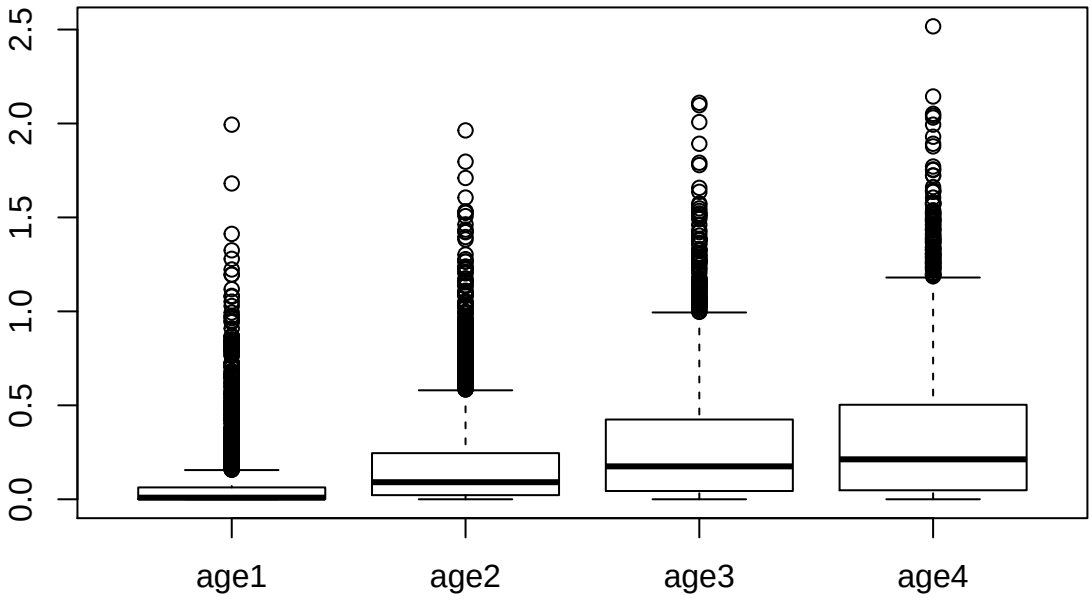
```
  boxplot(  od_rl , main = 'out degree by risk level')
```




```
print ( ( wilcox.test( od_rl[[1]], od_rl[[2]] ) ))
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  od_rl[[1]] and od_rl[[2]]
## W = 4163000, p-value = 0.5648
## alternative hypothesis: true location shift is not equal to 0
```

```
#~ ROC for detecting risk level TODO

#########################
#~ assoc of outdegree & age
age_ids <- paste(sep='', 'age', 1:4)
age_sids <- setNames( lapply( age_ids, function(ageid) grepl(  ageid, sampleDemes[names(od)] ) ) , age_
od_age <- setNames(  lapply( age_ids, function(ageid) od[ age_sids[[ageid]] ] ), age_ids )
  boxplot(  od_age , main = 'out degree by age group')
```




```
print ( ( kruskal.test( od_age ) ))
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  od_age
## Kruskal-Wallis chi-squared = 1385.1, df = 3, p-value < 2.2e-16
```

```
##
save.image( file='phydynR-testSA0.0.RData' )
```