ToCatchAThief

c ryan campbell & jenn coughlan

7/23/2018

Welcome to the "To Catch a Thief: With Data!" walkthrough! https://bioconductor.org/packages/devel/bioc/vignettes/SNPRelate/inst/doc/SNPRelateTutorial.html

First we're going to install the required software within R, answer "yes" or "all" to any prompted questions: i ran source successfully!!

In case you're wondering, this is an "R Markdown" document. Everything in the white space is notes, the grey space (between "' symbols) is code to be run.

This makes it easy to take notes, and jot down important things that aren't computer jargon right next to the things that you want to run as computer jargon. For details see: http://rmarkdown.rstudio.com.

Inspect the SNP data then then save them as "genofile" (in R "<-" saves a variable)

```
snpgdsSummary(snpgdsExampleFileName())
```

```
## The file name: /Library/Frameworks/R.framework/Versions/3.4/Resources/library/SNPRelate/extdata/hapma
## The total number of samples: 279
## The total number of SNPs: 9088
## SNP genotypes are stored in SNP-major mode (Sample X SNP).
```

```
genofile <- snpgdsOpen(snpgdsExampleFileName(), allow.duplicate = TRUE)
```

Now let's inspect the data a little more thoroughly

```
head(read.gdsn(index.gdsn(genofile, "snp.rs.id")))
```

```
## [1] "rs1695824"  "rs13328662" "rs4654497"  "rs10915489" "rs12132314"
## [6] "rs12042555"
```

```
pop <- read.gdsn(index.gdsn(genofile, path="sample.annot/pop.group"))
table(pop)
```

```
## pop
## CEU HCB JPT YRI
##  92  47  47  93
```

Questions to answer (feel free to edit this document to include your answers!): 1) How many samples are in this file? 2) How many SNPs are in this file (per sample, not total)? 3) What do these population codes mean (use Google)?

Now we're going to visualize the samples with a Principle Components Analysis (Jenn discussed this method earlier this week). There is a lot going on in the next block of text to run, don't focus on the minor things just the end product.

```
#first we're going to eliminate SNPs in LD with each other
set.seed(1000)
snpset <- snpgdsLDpruning(genofile, ld.threshold=0.2)
```

```
## SNP pruning based on LD:
## Excluding 365 SNPs on non-autosomes
## Excluding 1 SNP (monomorphic: TRUE, MAF: NaN, missing rate: NaN)
## Working space: 279 samples, 8,722 SNPs
##     using 1 (CPU) core
```

```
##     sliding window: 500,000 basepairs, Inf SNPs
##     |LD| threshold: 0.2
##     method: composite
## Chromosome 1: 75.42%, 540/716
## Chromosome 2: 72.24%, 536/742
## Chromosome 3: 74.71%, 455/609
## Chromosome 4: 73.31%, 412/562
## Chromosome 5: 77.03%, 436/566
## Chromosome 6: 75.58%, 427/565
## Chromosome 7: 75.42%, 356/472
## Chromosome 8: 71.31%, 348/488
## Chromosome 9: 77.88%, 324/416
## Chromosome 10: 74.33%, 359/483
## Chromosome 11: 77.40%, 346/447
## Chromosome 12: 76.81%, 328/427
## Chromosome 13: 75.58%, 260/344
## Chromosome 14: 76.95%, 217/282
## Chromosome 15: 76.34%, 200/262
## Chromosome 16: 72.66%, 202/278
## Chromosome 17: 74.40%, 154/207
## Chromosome 18: 73.68%, 196/266
## Chromosome 19: 85.00%, 102/120
## Chromosome 20: 71.62%, 164/229
## Chromosome 21: 76.98%, 97/126
## Chromosome 22: 75.86%, 88/116
## 6,547 markers are selected in total.
```

```
snpset.id <- unlist(snpset)

#runPCA analysis on the smaller set of data, "snpset.id"
pca <- snpgdsPCA(genofile, snp.id=snpset.id, num.thread=2)
```

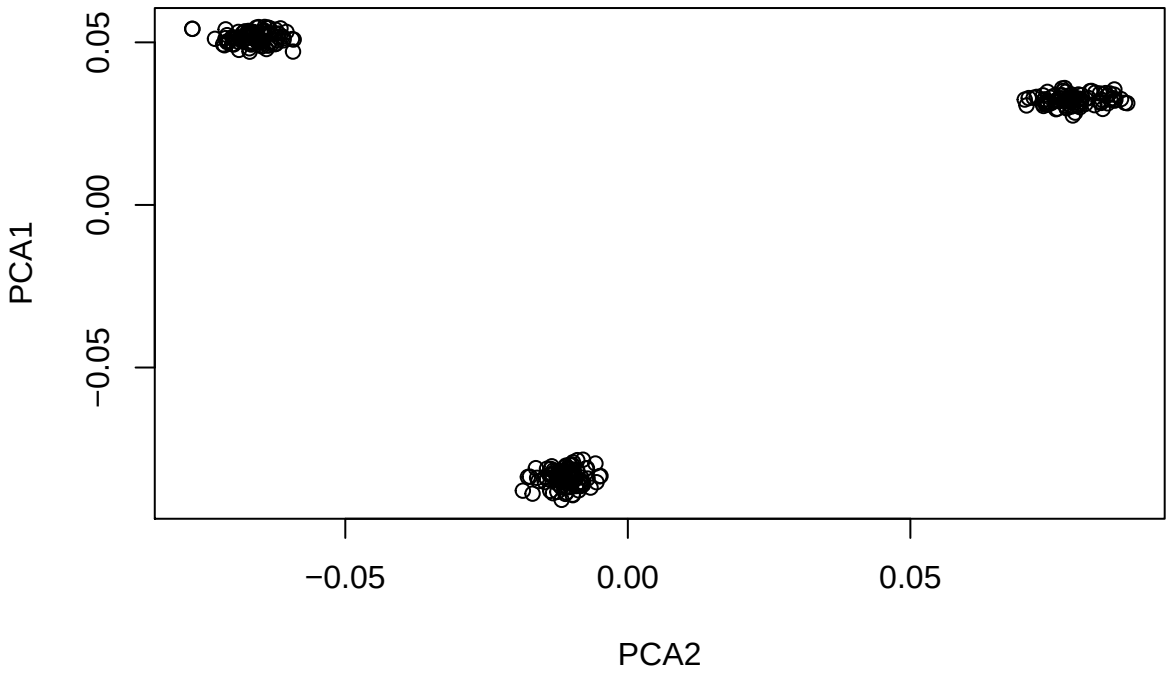
```
## Principal Component Analysis (PCA) on genotypes:
## Excluding 2,541 SNPs (non-autosomes or non-selection)
## Excluding 0 SNP (monomorphic: TRUE, MAF: NaN, missing rate: NaN)
## Working space: 279 samples, 6,547 SNPs
##     using 2 (CPU) cores
## PCA:    the sum of all selected genotypes (0,1,2) = 1826801
## CPU capabilities: Double-Precision SSE2
## Thu Jul 26 12:22:11 2018    (internal increment: 1744)
##
[..................................................]  0%, ETC: ---
[==================================================] 100%, completed in 0s
## Thu Jul 26 12:22:11 2018    Begin (eigenvalues and eigenvectors)
## Thu Jul 26 12:22:11 2018    Done.
```

```
#convert PCA results to a table
tab <- data.frame(sample.id = pca$sample.id,
    EV1 = pca$eigenvect[,1],    # the first eigenvector
    EV2 = pca$eigenvect[,2],    # the second eigenvector
    stringsAsFactors = FALSE)
#visualize that table
head(tab)
```

```
##   sample.id         EV1         EV2
## 1   NA19152 -0.08246488 -0.01006984
```

```
## 2   NA19139 -0.08225226 -0.01051959
## 3   NA18912 -0.08182177 -0.01274934
## 4   NA19160 -0.08793052 -0.01371887
## 5   NA07034  0.03160206  0.07831394
## 6   NA07055  0.03456066  0.08270710
```

```
#plot
plot(tab$EV2, tab$EV1, xlab="PCA2", ylab="PCA1")
```

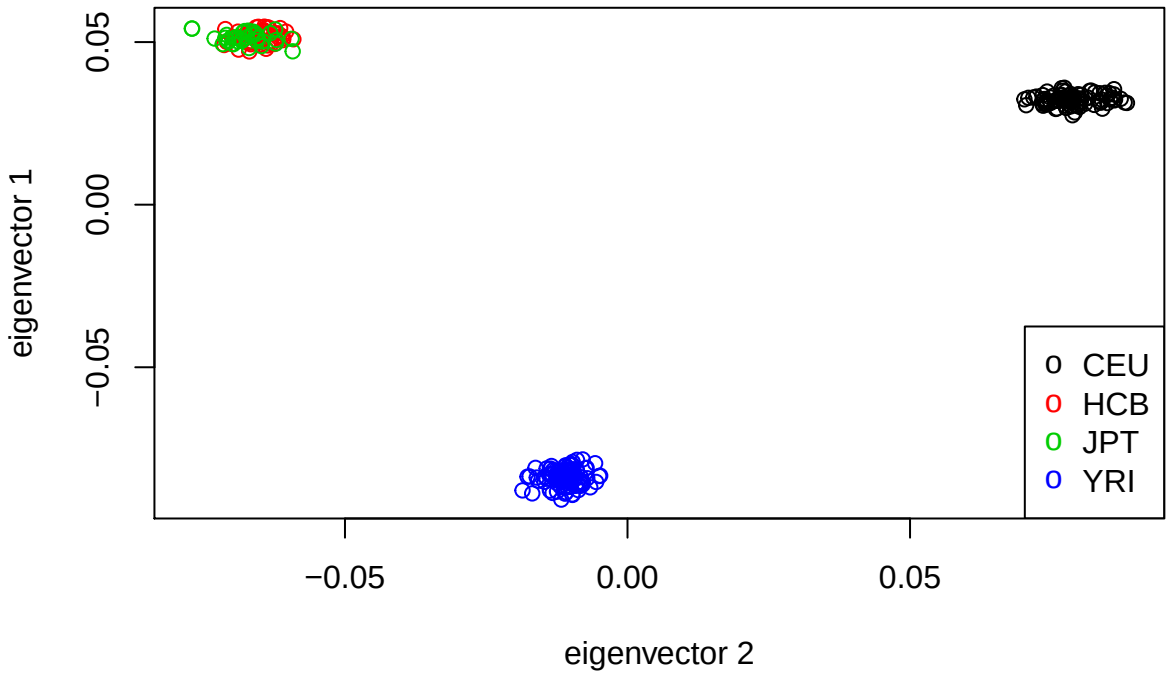



```
# Get sample id and pop info
sample.id <- read.gdsn(index.gdsn(genofile, "sample.id"))
pop_code <- read.gdsn(index.gdsn(genofile, "sample.annot/pop.group"))

#add pop info to table
tab <- data.frame(sample.id = pca$sample.id,
    pop = factor(pop_code)[match(pca$sample.id, sample.id)],
    EV1 = pca$eigenvect[,1],    # the first eigenvector
    EV2 = pca$eigenvect[,2],    # the second eigenvector
    stringsAsFactors = FALSE)
head(tab)
```

```
##   sample.id pop         EV1         EV2
## 1   NA19152 YRI -0.08246488 -0.01006984
## 2   NA19139 YRI -0.08225226 -0.01051959
## 3   NA18912 YRI -0.08182177 -0.01274934
## 4   NA19160 YRI -0.08793052 -0.01371887
## 5   NA07034 CEU  0.03160206  0.07831394
## 6   NA07055 CEU  0.03456066  0.08270710
```

```
#plot with pop info
plot(tab$EV2, tab$EV1, col=as.integer(tab$pop), xlab="eigenvector 2", ylab="eigenvector 1"); legend("bo
```



4)

Why did we eliminate SNPs that are in LD with each other? 5) How many clusters do we have? 6) What does that tell us about our individuals? 7) Which populations cluster together on the PCA?

Now, given a criminal can we find relatives in this database?

We'll need to use a different analysis here, one of "kinship" or the relatedeness between two samples There are 3 outputs from this analysis - k0 - probability that zero genetic segments are shared k1 - probability that at least one genetic segments are shared kinship - numerical relationship between samples (ranging from 0 - .5) relatedness = 2 * kinship (ranges from 0-1)

How much of your DNA do you share with 8) a parent? 9) a sibling? 10) grandparent?

the relatedness values are directly proportional to these answers. If you find a sample of .25 relatedness you've got the criminal's grandparent or aunt/uncle.

Here we will generate a pool of individuals similar to GEDmatch. By recent estimates GEDmatch has 10x as many Caucasian entries as any other human population, so the remainder of this walkthrough will have 89 Caucasian samples and 9 Yoruban, 9 Han Chinese, and 9 Japanese.

```
sample.id <- read.gdsn(index.gdsn(genofile, "sample.id"))

GEDmatch.id <- sample.id[pop_code == "CEU"]
GEDmatch.id <- GEDmatch.id[-37]
GEDmatch.id <- GEDmatch.id[-43]
GEDmatch.id <- GEDmatch.id[-51]
GEDmatch.id <- c(GEDmatch.id,sample.id[pop_code == "YRI"][1:9])
GEDmatch.id <- c(GEDmatch.id,sample.id[pop_code == "HCB"][1:9])
GEDmatch.id <- c(GEDmatch.id,sample.id[pop_code == "JPT"][1:9])
```

The partner differentiation kicks in here SO READ CAREFULLY:

```
#PARTNER A - uncomment (delete the #'s) this section and run
criminal <- GEDmatch.id[sample(1:89,1)]

############
#PARTNER B - uncomment (delete the #'s) this section and run
```

```
#criminal <- GEDmatch.id[sample(90:116,1)]

#############
#THIS SECTION RANDOMLY ASSIGNS YOU A CRIMINAL
#JUST RUN IT ONCE
#IF YOU RUN IT AGAIN YOUR CRIMINAL (AND RESULTS) WILL CHANGE
```

Finally we're going to compare every sample in the database to every other, then isolate those that include our "criminal".

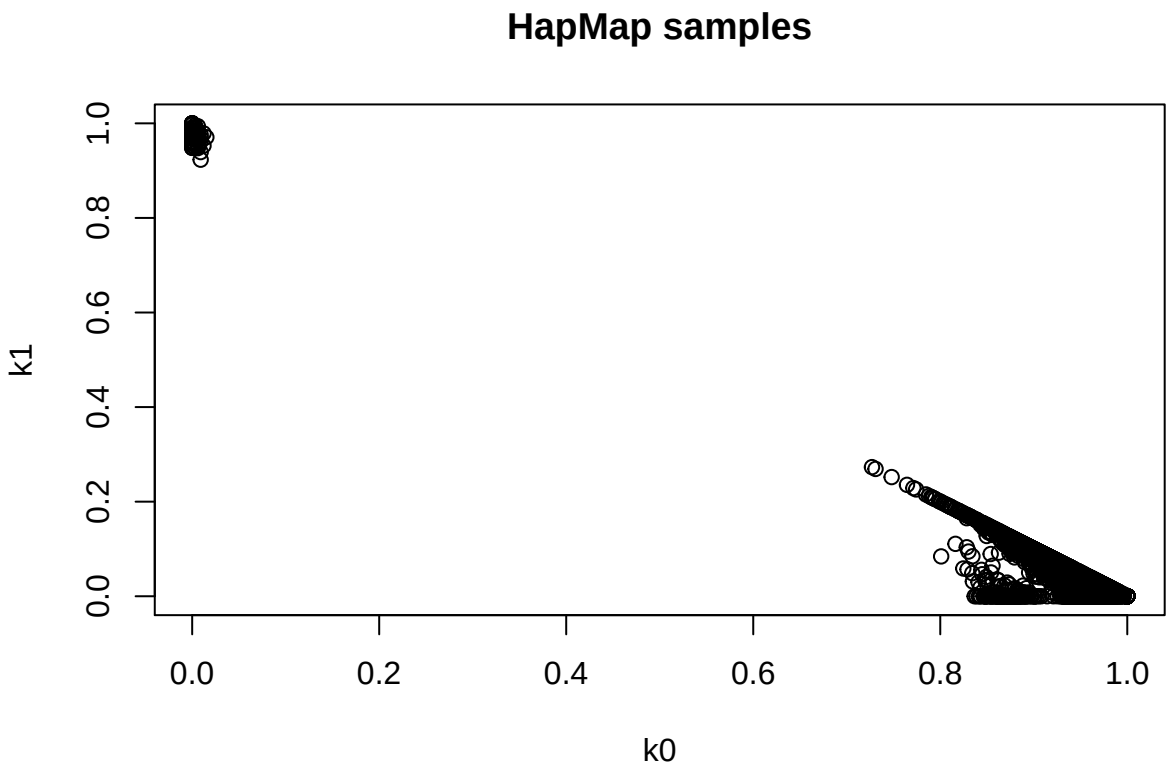
```
# Estimate IBD coefficients
ibd <- snpgdsIBDMoM(genofile, sample.id=GEDmatch.id, snp.id=snpset.id,
    maf=0.05, missing.rate=0.05, num.thread=4)
```

```
## IBD analysis (PLINK method of moment) on genotypes:
## Excluding 2,541 SNPs (non-autosomes or non-selection)
## Excluding 1,534 SNPs (monomorphic: TRUE, MAF: 0.05, missing rate: 0.05)
## Working space: 116 samples, 5,013 SNPs
##     using 4 (CPU) cores
## PLINK IBD:    the sum of all selected genotypes (0,1,2) = 584461
## Thu Jul 26 12:22:12 2018    (internal increment: 65536)
##
[..................................................]  0%, ETC: ---
[==================================================] 100%, completed in 0s
## Thu Jul 26 12:22:12 2018    Done.
```

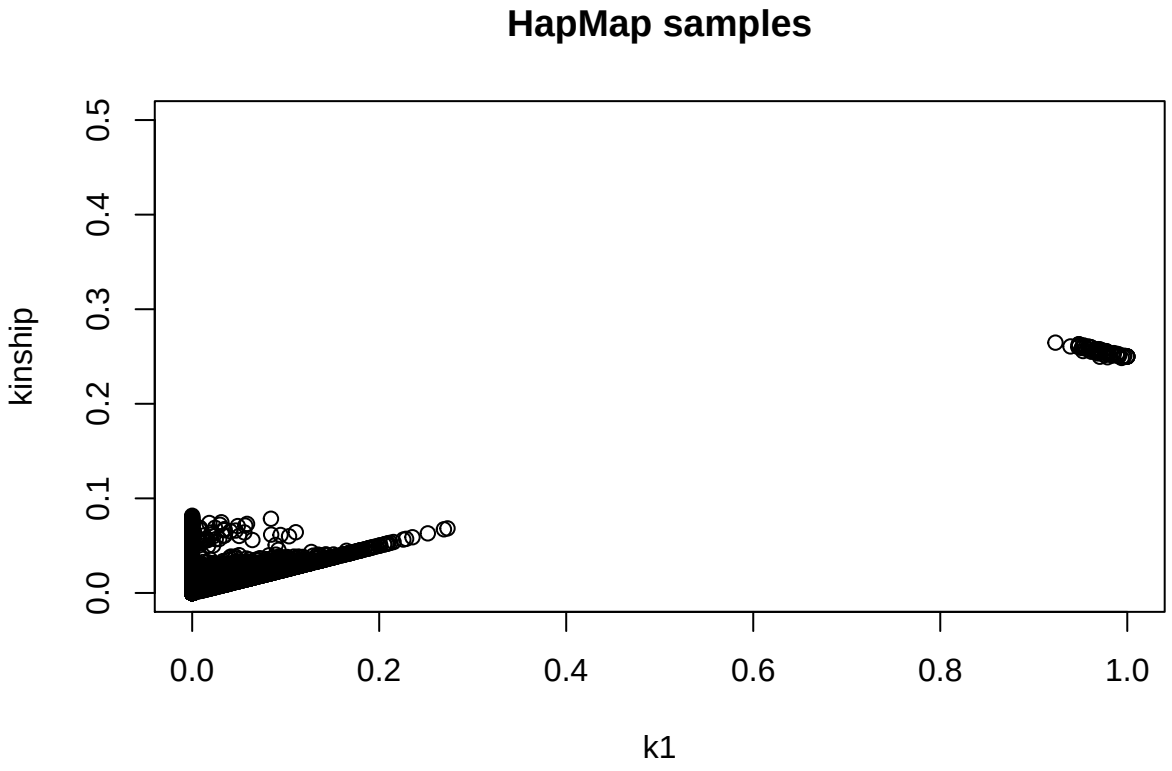
```
#convert the data to a table
ibd.coeff <- snpgdsIBDSelection(ibd)
head(ibd.coeff)
```

```
##       ID1     ID2        k0 k1    kinship
## 1 NA19152 NA19139 0.9079118  0 0.04604409
## 2 NA19152 NA18912 0.9007332  0 0.04963340
## 3 NA19152 NA19160 1.0000000  0 0.00000000
## 4 NA19152 NA07034 1.0000000  0 0.00000000
## 5 NA19152 NA07055 1.0000000  0 0.00000000
## 6 NA19152 NA12814 1.0000000  0 0.00000000
```

```
#plot ALL data
plot(ibd.coeff$k0, ibd.coeff$k1, xlim=c(0,1), ylim=c(0,1),
    xlab="k0", ylab="k1", main="HapMap samples")
```

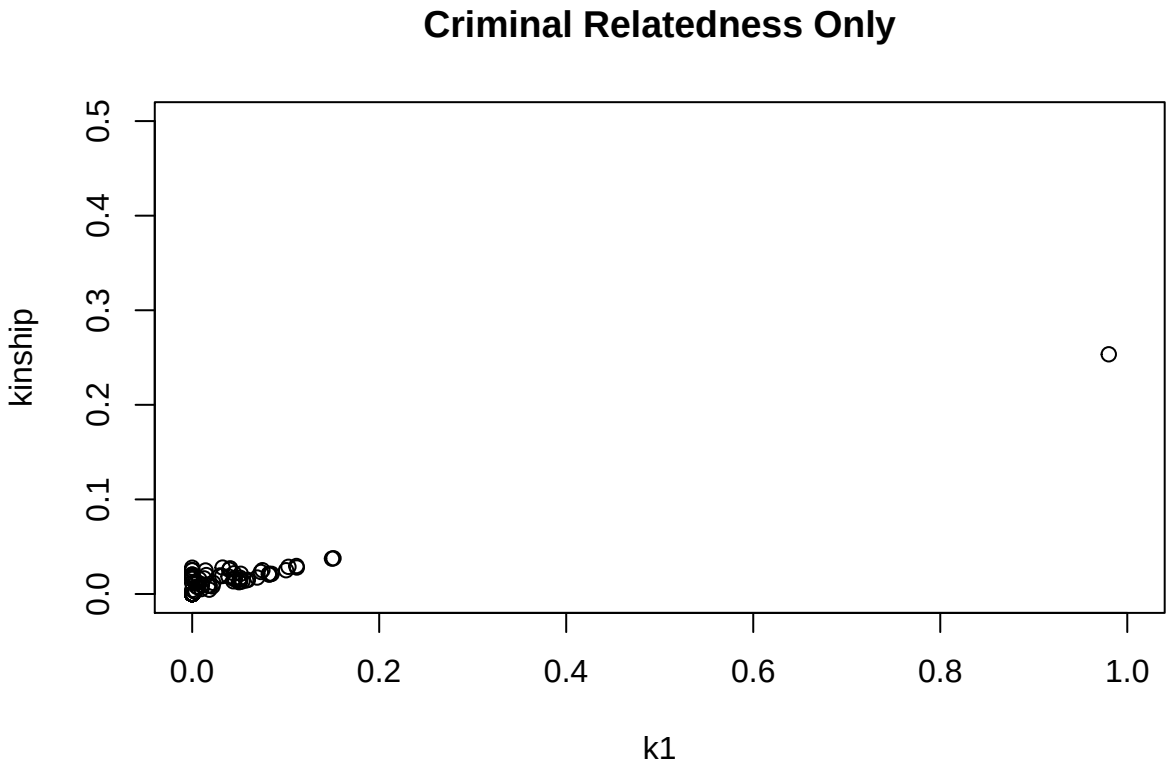



```
plot(ibd.coeff$k1, ibd.coeff$kinship, xlim=c(0,1), ylim=c(0,.5),
    xlab="k1", ylab="kinship", main="HapMap samples")
```

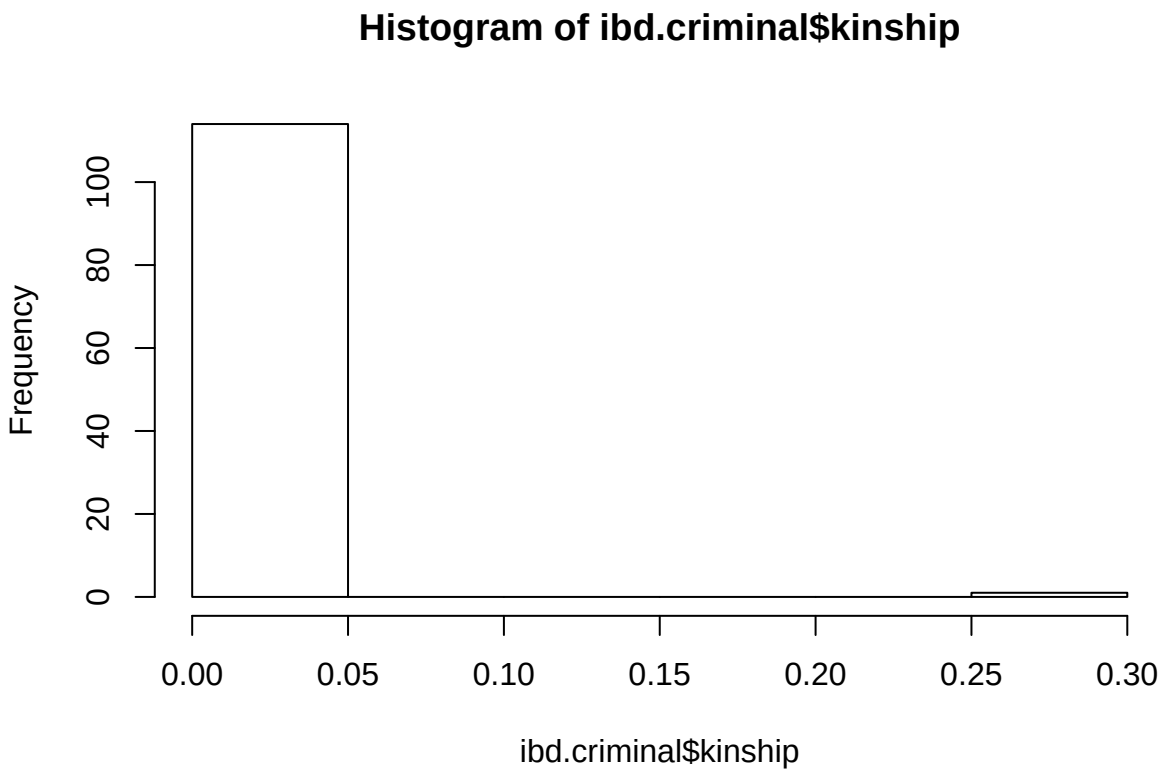



```
#isolate criminal match data
ibd.criminal <- subset(ibd.coeff, ID1 == criminal | ID2 == criminal)
```

```r
#plot criminal data
plot(ibd.criminal$k1, ibd.criminal$kinship, xlim=c(0,1), ylim=c(0,.5),
    xlab="k1", ylab="kinship", main="Criminal Relatedness Only")
```




```r
#plot kinship
hist(ibd.criminal$kinship)
```




```r
#return maximum kinship value
max(ibd.criminal$kinship)
```

```
## [1] 0.2534409
```

```r
#return ID match
ibd.criminal$ID1[which.max(ibd.criminal$kinship)]
```

```
## [1] "NA11995"
```

```r
#return population
tab$pop[tab$sample.id == criminal]
```

```
## [1] CEU
## Levels: CEU HCB JPT YRI
```

11) Are you partner A or B?

12) What is the ID of your criminal? (just run "criminal")

13) Does your criminal have any matches?

14) What kinship level?

15) What relatedness is that?

16) Who is that, as in which family member?

17) Finally compare your result to your partner - how did they differ and do you have a guess as to why?