

COBP PVA info for Tyson

Daniel Laughlin

1/23/2019

load data for flowering plant counts and merge with climate and streamflow data for analysis at segment level

```
### set directory
setwd("~/OneDrive - University of Wyoming/Data/COBP")

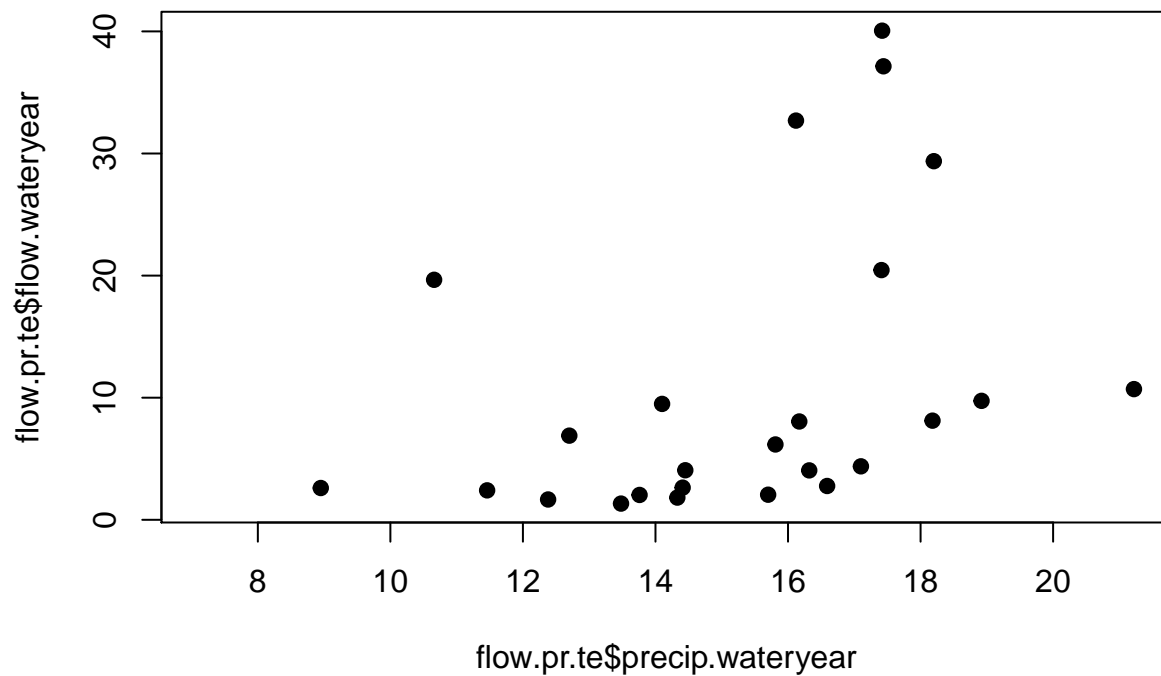
### load data
precip <- read.csv("precip.cheyenne.csv", skip=1, header=TRUE)
temp <- read.csv("temp.cheyenne.csv", skip=1, header=TRUE)
flow <- read.csv("crow.flow.csv", skip=1, header=TRUE)
counts.segment <- read.csv("cobp.count.csv", header=TRUE)

### merge datasets

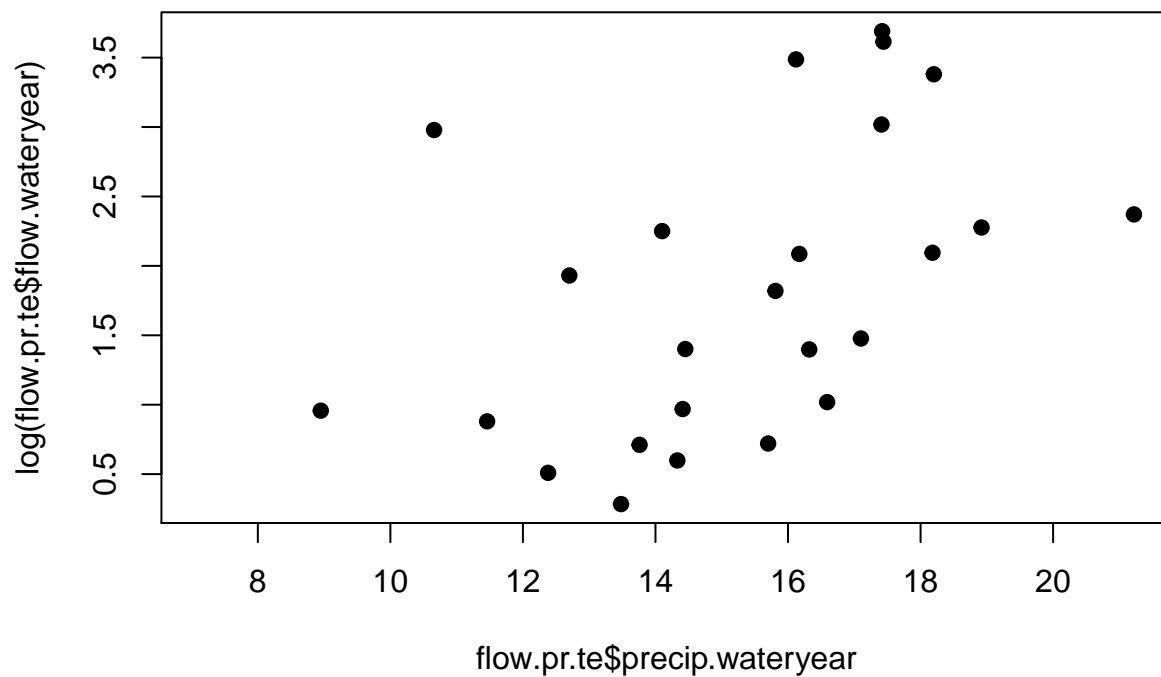
pr.te <- merge(precip[,c(1,14:16)], temp[,c(1,14:16)], by="year") # merge annual summaries for simplicity
flow.pr.te <- merge(pr.te, flow[,c(1,14:16)], all.x=TRUE) # merge annual summaries for simplicity
dat <- merge(counts.segment, flow.pr.te, by="year")
```

stream flow vs water year

```
### there is a poisson-distributed-looking relationship between precip and streamflow
plot(flow.pr.te$flow.wateryear ~ flow.pr.te$precip.wateryear, pch=19)
```



```
plot(log(flow.pr.te$flow.wateryear) ~ flow.pr.te$precip.wateryear, pch=19)
```



```
m1 <- lm(log(flow.pr.te$flow.wateryear) ~ flow.pr.te$precip.wateryear)
summary(m1)
```

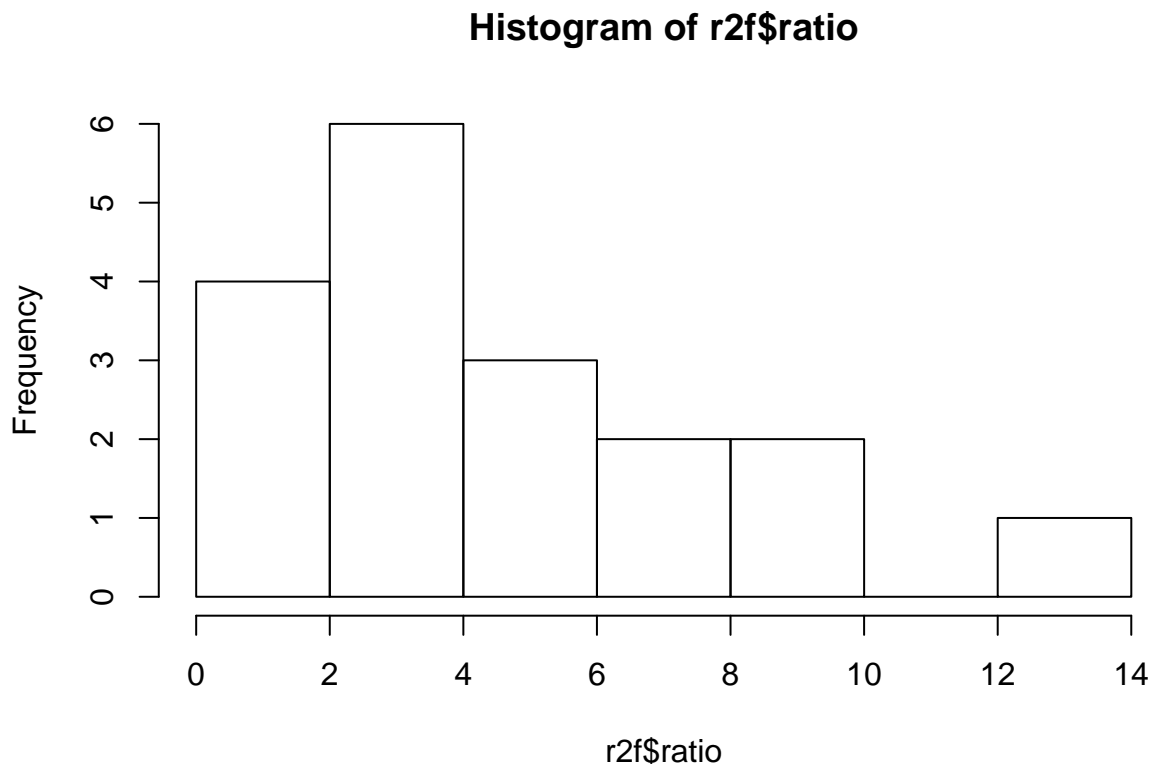
```
##
## Call:
## lm(formula = log(flow.pr.te$flow.wateryear) ~ flow.pr.te$precip.wateryear)
##
## Residuals:
```

##	Min	1Q	Median	3Q	Max

```
## -1.2254 -0.7051 -0.2460  0.6311  1.9673
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -0.87313    1.09083   -0.800   0.432
## flow.pr.te$precip.wateryear  0.17676    0.07004    2.524   0.019 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9594 on 23 degrees of freedom
## (59 observations deleted due to missingness)
## Multiple R-squared:  0.2169, Adjusted R-squared:  0.1828
## F-statistic: 6.369 on 1 and 23 DF,  p-value: 0.01897
#### so either precip is a mediocre substitute for streamflow, or flow will add additional info
#### however, we only have flow data from 1993 onward, which means we could potentially PREDICT
#### stream flow for previous years 1987-1992?
#### I think Bonnie said there was one year where there were releases from a dam,
#### so should check on that if we use it
```

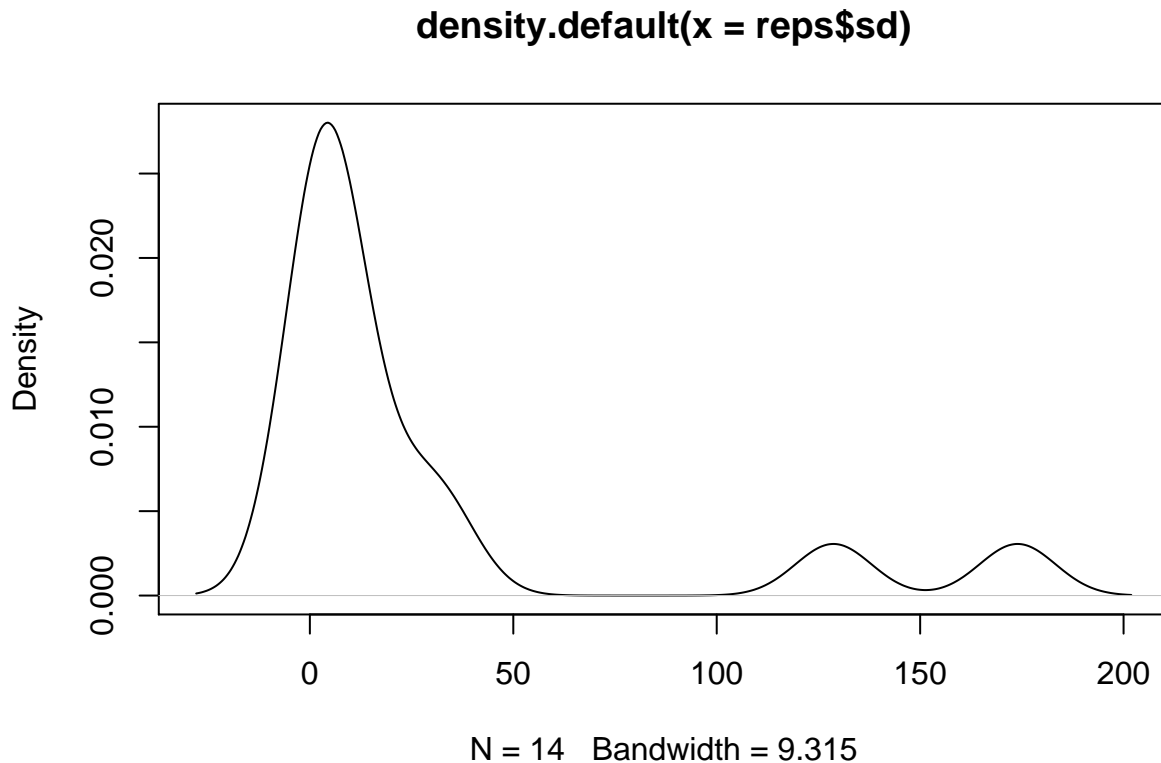
Ratio of flowering stems to rosettes in 2018

```
#### This ratio from 2018 and other years could be used as a prior to constrain estimates of rosette plant
r2f <- read.csv("cobp.rosette.ratio.csv", header=TRUE)
hist(r2f$ratio)
```



Measurement error for flowering stems in 2014

```
### Not sure how we incorporate this into the analysis, but these replicate counts provide information
### about the measurement error in counting flowering stems
reps <- read.csv("count.replication.csv", header=TRUE)
for(i in 1:nrow(reps)) reps$sd[i] <- sd(c(reps$first.count[i], reps$second.count[i]))
plot(density(reps$sd))
```



```
mean(reps$sd)
```

```
## [1] 29.54696
```

```
sd(reps$sd)
```

```
## [1] 53.3329
```

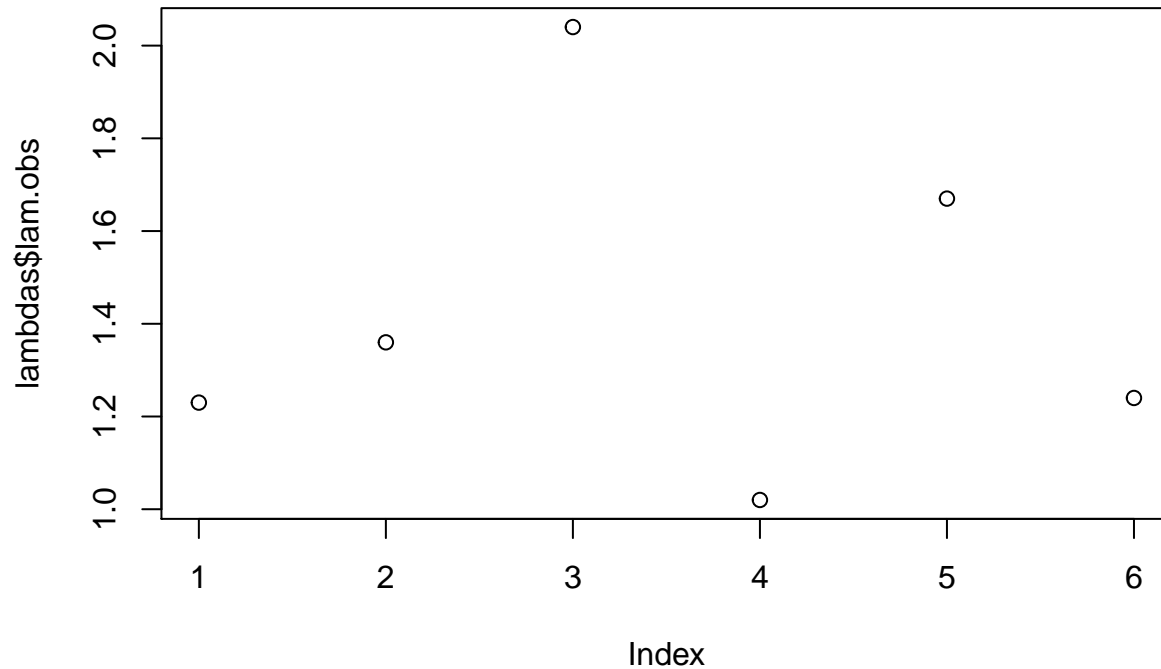
```
#plot(reps$first.count, reps$sd)
```

Comparing lambda estimates using counts at the stream level

```
### lam.floyd are lambdas computed from Floyd's transition matrices (mean matrix of 3 plots per stream)
### lam.flowering.obs are lambdas computed from counts of flowering stems
lambdas <- data.frame(lam.floyd = c(1.23, 1.36, 2.04, 1.02, 1.67, 1.24),
                      lam.flowering.obs = c(0.9378134, 2.1572193, 0.9005392, 0.9268130, 1.2820513, 0.83
rownames(lambdas) <- c("crow.92.93", "crow.93.94", "unnamed.92.93", "unnamed.93.94", "diamond.92.93", "

### These estimates are not correlated!
### This is either because counts of flowering stems is not an accurate assessment of population size
```

```
### or the plots in Floyd's study are not accurate assessments of population growth rates
plot(lambdas$lam.floyd, lambdas$lam.obs)
```

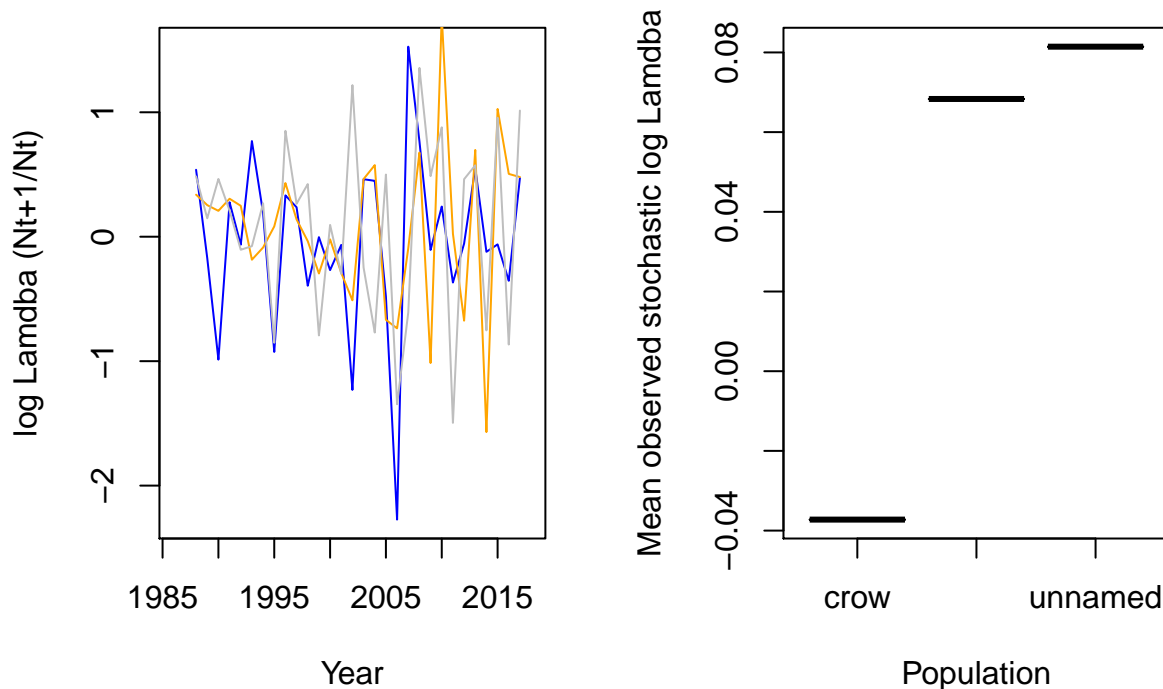


```
### These are counts at the stream level to compute "Lambda"
counts.stream <- read.csv("counts.stream.csv", header=TRUE)
counts.stream$crow.lam <- NA
counts.stream$diam.lam <- NA
counts.stream$un.lam <- NA

### This calculation of lambda only accounts for flowering plants!
### Need to account for rosettes to be useful

for(i in 1: nrow(counts.stream)){
  counts.stream$crow.lam[i] <- counts.stream$crow.total[i+1] / counts.stream$crow.total[i]
  counts.stream$diam.lam[i] <- counts.stream$diamond.total[i+1] / counts.stream$diamond.total[i]
  counts.stream$un.lam[i] <- counts.stream$unnamed.total[i+1] / counts.stream$unnamed.total[i]
}

par(mfrow=c(1,2))
plot(counts.stream$year, log(counts.stream$crow.lam), type="l", xlab="Year", ylab="log Lambda (Nt+1/Nt)")
lines(counts.stream$year, log(counts.stream$diam.lam), col="orange")
lines(counts.stream$year, log(counts.stream$un.lam), col="grey")
boxplot(mean(log(counts.stream$crow.lam[-c(1:2,33)])), mean(log(counts.stream$diam.lam[-c(1:2,33)])), mean(log(counts.stream$un.lam[-c(1:2,33)])),
  xlab="Population", ylab="Mean observed stochastic log Lambda", col=c("blue","orange","grey"), na.rm=TRUE)
```



These functions are needed for the next chunk

Collapsing 4x4 Transition matrices from Floyd into 2x2 transition matrices

```
### If we want to model 2-stages (rosettes and flowering), and use Floyd's data, then we need to collapse
### collapseMatrix does this for us, and it preserves the exact same asymptotic growth rates (eigenvalues)
### Can these 2-stage transition probabilities be used as priors in your analysis?
### After collapsing, the transition probabilities are remarkably similar

### I've done this for the mean transition matrices in each stream, but Floyd published transition matrices

# Mean transition matrix for Crow Creek 1992-1993, broken into survival (matU) and fecundity (matF) matrices
mat.crow.92.93 <- c(0.043, 0.010, 0.000, 7.869,
                  0.348, 0.260, 0.077, 3.829,
                  0.565, 0.410, 0.385, 0.168,
                  0.000, 0.070, 0.253, 0.040)

matU.crow.92.93 <- c(0.043, 0.010, 0.000, 0.000,
                   0.348, 0.260, 0.077, 0.000,
                   0.565, 0.410, 0.385, 0.000,
                   0.000, 0.070, 0.253, 0.040)

matF.crow.92.93 <- c(0.000, 0.000, 0.000, 7.869,
                   0.000, 0.000, 0.000, 3.829,
                   0.000, 0.000, 0.000, 0.168,
                   0.000, 0.000, 0.000, 0.000)

matU <- matrix(matU.crow.92.93, nrow=4, byrow=TRUE)
```

```
matF <- matrix(matF.crow.92.93, nrow=4, byrow=TRUE)
collapse1 <- list(1:3, 4)
matA.crow.92.93 <- collapseMatrix(matU, matF, matC=NULL, collapse1)$matA
matA.crow.92.93
```

```
##           [,1]      [,2]
## [1,] 0.7133300 11.866
## [2,] 0.1010059  0.040
```

```
lifeExpectancy(matU)
```

```
## [1] 3.497024
```

```
colnames(matA.crow.92.93)<-c("rosette","flowering")
rownames(matA.crow.92.93)<-c("rosette","flowering")
```

```
mat.crow.93.94 <- c(0.048, 0.015, 0.000, 1.372,
                   0.455, 0.295, 0.000, 0.823,
                   0.217, 0.515, 0.312, 0.137,
                   0.000, 0.045, 0.581, 0.000)
```

```
matU.crow.93.94 <- c(0.048, 0.015, 0.000, 0.000,
                   0.455, 0.295, 0.000, 0.000,
                   0.217, 0.515, 0.312, 0.000,
                   0.000, 0.045, 0.581, 0.000)
```

```
matF.crow.93.94 <- c(0.000, 0.000, 0.000, 1.372,
                   0.000, 0.000, 0.000, 0.823,
                   0.000, 0.000, 0.000, 0.137,
                   0.000, 0.000, 0.000, 0.000)
```

```
matU <- matrix(matU.crow.93.94, nrow=4, byrow=TRUE)
matF <- matrix(matF.crow.93.94, nrow=4, byrow=TRUE)
collapse1 <- list(1:3, 4)
matA.crow.93.94 <- collapseMatrix(matU, matF, matC=NULL, collapse1)$matA
matA.crow.93.94
```

```
##           [,1]      [,2]
## [1,] 0.6133458  2.332
## [2,] 0.2236279  0.000
```

```
lambda(matA.crow.93.94)
```

```
## [1] 1.091242
```

```
mat.unnamed.92.93 <- c(0.000, 0.023, 0.000, 15.033,
                   0.465, 0.263, 0.039, 6.769,
                   0.465, 0.320, 0.394, 0.937,
                   0.000, 0.051, 0.457, 0.000)
```

```
matU.unnamed.92.93 <- c(0.000, 0.023, 0.000, 0.000,
                   0.465, 0.263, 0.039, 0.000,
                   0.465, 0.320, 0.394, 0.000,
                   0.000, 0.051, 0.457, 0.000)
```

```
matF.unnamed.92.93 <- c(0.000, 0.000, 0.000, 15.033,
                   0.000, 0.000, 0.000, 6.769,
                   0.000, 0.000, 0.000, 0.937,
```

```

0.000, 0.000, 0.000, 0.000)
A <- matrix(mat.unnamed.92.93, nrow=4, byrow=T)
lambda(A)

## [1] 2.038883

matU <- matrix(matU.unnamed.92.93, nrow=4, byrow=TRUE)
matF <- matrix(matF.unnamed.92.93, nrow=4, byrow=TRUE)
collapse1 <- list(1:3, 4)
matA.unnamed.92.93 <- collapseMatrix(matU, matF, matC=NULL, collapse1)$matA
matA.unnamed.92.93

##          [,1] [,2]
## [1,] 0.7084620 22.739
## [2,] 0.1192917  0.000
lambda(matA.unnamed.92.93)

## [1] 2.038883

mat.unnamed.93.94 <- c(0.092, 0.019, 0.007, 1.916,
                      0.441, 0.541, 0.116, 0.421,
                      0.077, 0.344, 0.355, 0.013,
                      0.000, 0.032, 0.471, 0.000)
matU.unnamed.93.94 <- c(0.092, 0.019, 0.007, 0.000,
                      0.441, 0.541, 0.116, 0.000,
                      0.077, 0.344, 0.355, 0.000,
                      0.000, 0.032, 0.471, 0.000)
matF.unnamed.93.94 <- c(0.000, 0.000, 0.000, 1.916,
                      0.000, 0.000, 0.000, 0.421,
                      0.000, 0.000, 0.000, 0.013,
                      0.000, 0.000, 0.000, 0.000)
A <- matrix(mat.unnamed.93.94, nrow=4, byrow=T)
lambda(A)

## [1] 1.024747

matU <- matrix(matU.unnamed.93.94, nrow=4, byrow=TRUE)
matF <- matrix(matF.unnamed.93.94, nrow=4, byrow=TRUE)
collapse1 <- list(1:3, 4)
matA.unnamed.93.94 <- collapseMatrix(matU, matF, matC=NULL, collapse1)$matA
matA.unnamed.93.94

##          [,1] [,2]
## [1,] 0.7060453 2.35
## [2,] 0.1389737 0.00
lambda(matA.unnamed.93.94)

## [1] 1.024747

mat.diamond.92.93 <- c(0.077, 0.007, 0.000, 10.726,
                      0.538, 0.319, 0.060, 2.917,
                      0.308, 0.399, 0.208, 0.713,
                      0.019, 0.036, 0.403, 0.054)
matU.diamond.92.93 <- c(0.077, 0.007, 0.000, 0.000,
                      0.538, 0.319, 0.060, 0.000,
                      0.308, 0.399, 0.208, 0.000,
                      0.019, 0.036, 0.403, 0.054)

```



```
matF.diamond.92.93 <- c(0.000, 0.000, 0.000, 10.726,
                        0.000, 0.000, 0.000, 2.917,
                        0.000, 0.000, 0.000, 0.713,
                        0.000, 0.000, 0.000, 0.000)
```

```
A <- matrix(mat.diamond.92.93, nrow=4, byrow=T)
lambda(A)
```

```
## [1] 1.673656
```

```
matU <- matrix(matU.diamond.92.93, nrow=4, byrow=TRUE)
matF <- matrix(matF.diamond.92.93, nrow=4, byrow=TRUE)
collapse1 <- list(1:3, 4)
matA.diamond.92.93 <- collapseMatrix(matU, matF, matC=NULL, collapse1)$matA
matA.diamond.92.93
```

```
##           [,1] [,2]
## [1,] 0.7147156 14.356
## [2,] 0.1081884  0.054
```

```
lambda(matA.diamond.92.93)
```

```
## [1] 1.673656
```

```
mat.diamond.93.94 <- c(0.078, 0.013, 0.000, 3.782,
                       0.616, 0.416, 0.160, 0.895,
                       0.043, 0.456, 0.395, 0.052,
                       0.016, 0.067, 0.387, 0.000)
```

```
matU.diamond.93.94 <- c(0.078, 0.013, 0.000, 0.000,
                        0.616, 0.416, 0.160, 0.000,
                        0.043, 0.456, 0.395, 0.000,
                        0.016, 0.067, 0.387, 0.000)
```

```
matF.diamond.93.94 <- c(0.000, 0.000, 0.000, 3.782,
                        0.000, 0.000, 0.000, 0.895,
                        0.000, 0.000, 0.000, 0.052,
                        0.000, 0.000, 0.000, 0.000)
```

```
A <- matrix(mat.diamond.93.94, nrow=4, byrow=T)
lambda(A)
```

```
## [1] 1.24222
```

```
matU <- matrix(matU.diamond.93.94, nrow=4, byrow=TRUE)
matF <- matrix(matF.diamond.93.94, nrow=4, byrow=TRUE)
collapse1 <- list(1:3, 4)
matA.diamond.93.94 <- collapseMatrix(matU, matF, matC=NULL, collapse1)$matA
matA.diamond.93.94
```

```
##           [,1] [,2]
## [1,] 0.7533902 4.729
## [2,] 0.1284064 0.000
```

```
lambda(matA.diamond.93.94)
```

```
## [1] 1.24222
```

```
### List of collapsed matrices, that show remarkable similarity!
```

```
two.stage.mats <- list( matA.crow.92.93,
                        matA.crow.93.94,
                        matA.unnamed.92.93,
                        matA.unnamed.93.94,
```

```
matA.diamond.92.93,  
matA.diamond.93.94)
```