Dependence analysis of four weights of abalone shells (Application to Lecture 2)

Claudia Czado,Technical University Munich

10 August, 2018

1.The abalone data set

Source and data description

The `abalone` dataset is available from the University of California Irvine (UCI) machine learning repository. Metadata can be obtained from http://archive.ics.uci.edu/ml/datasets/Abalone

It is also available in the library PivotalR.

- Sex / nominal / – / M, F, and I (infant)
- Length / continuous / mm / Longest shell measurement
- Diameter / continuous / mm / perpendicular to length
- Height / continuous / mm / with meat in shell
- Whole weight / continuous / grams / whole abalone
- Shucked weight / continuous / grams / weight of meat
- Viscera weight / continuous / grams / gut weight (after bleeding)
- Shell weight / continuous / grams / after being dried
- Rings / integer / – / +1.5 gives the age in years

2. Setup

Load packages

```
library(VineCopula)
library(PivotalR)
library(rafalib)
library(kdevine)
```

Load data and name columns

The dataset contains 10 variables and 4177 observations. Most of the variables are numeric. The only exception is the `sex` variable. The `rings` variable is slightly different from the other numeric variables because it assumes discrete, integer values.

```
data("abalone")
abalone.cols = c( "sex", "len", "dia", "h", "whole",
                  "shuck", "vis", "shell", "rings")
abalone1=abalone[,-1]
colnames(abalone1)=abalone.cols
sex1=abalone1[,1]
sex.num=rep(0,4177)
sex.num[sex1=="M"]=1
sex.num[sex1=="F"]=0
sex.num[sex1=="I"]=2
abalone1[,1]=sex.num
```

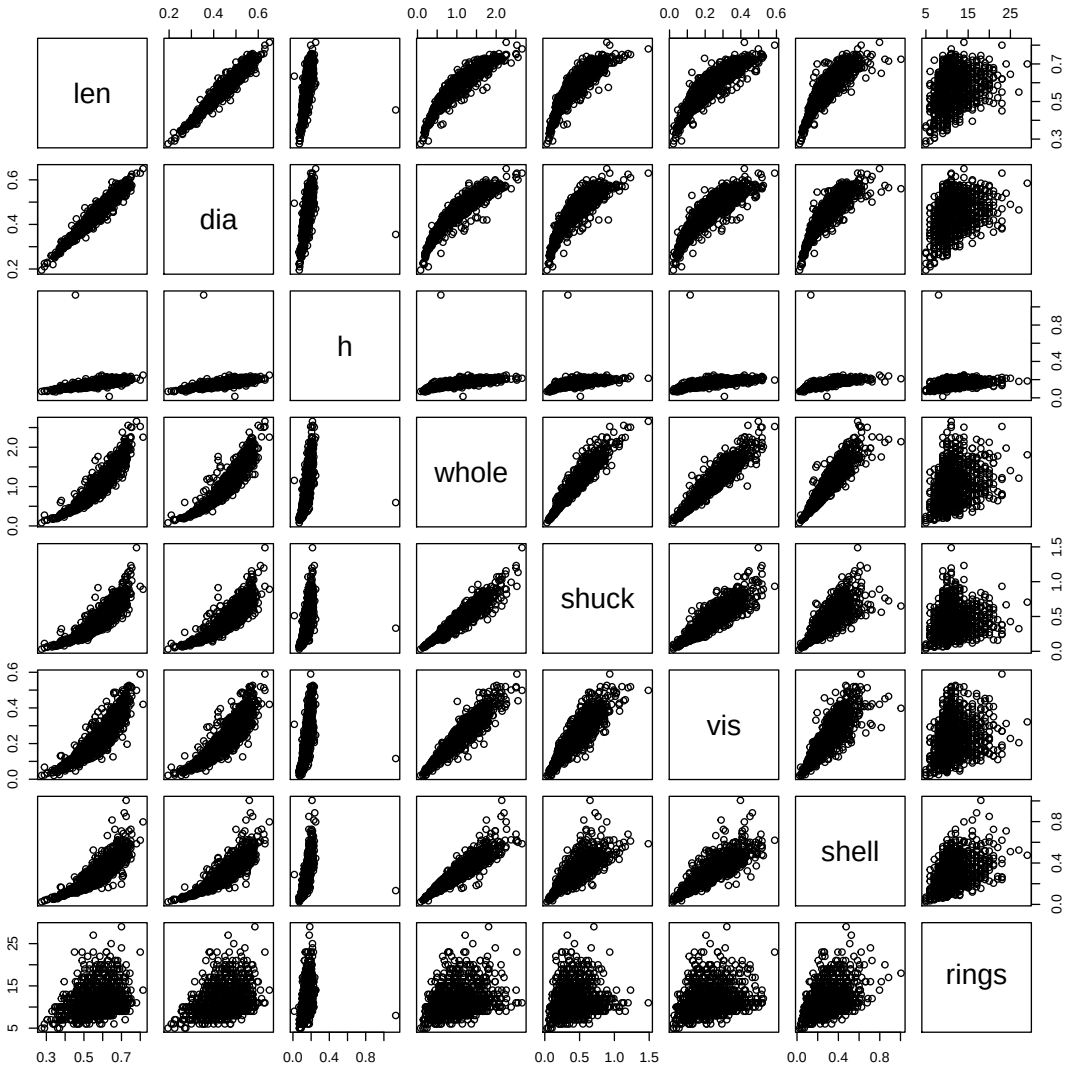
Create datasets for male, female and junvenile separately

```
attach(abalone)
abalone.f<-abalone1[sex=="F",-1]
abalone.m<-abalone1[sex=="M",-1]
abalone.i<-abalone1[sex=="I",-1]
detach(abalone)
```

3. Female abalone data set

Raw data of female abalone shells

```
pairs(abalone.f)
```



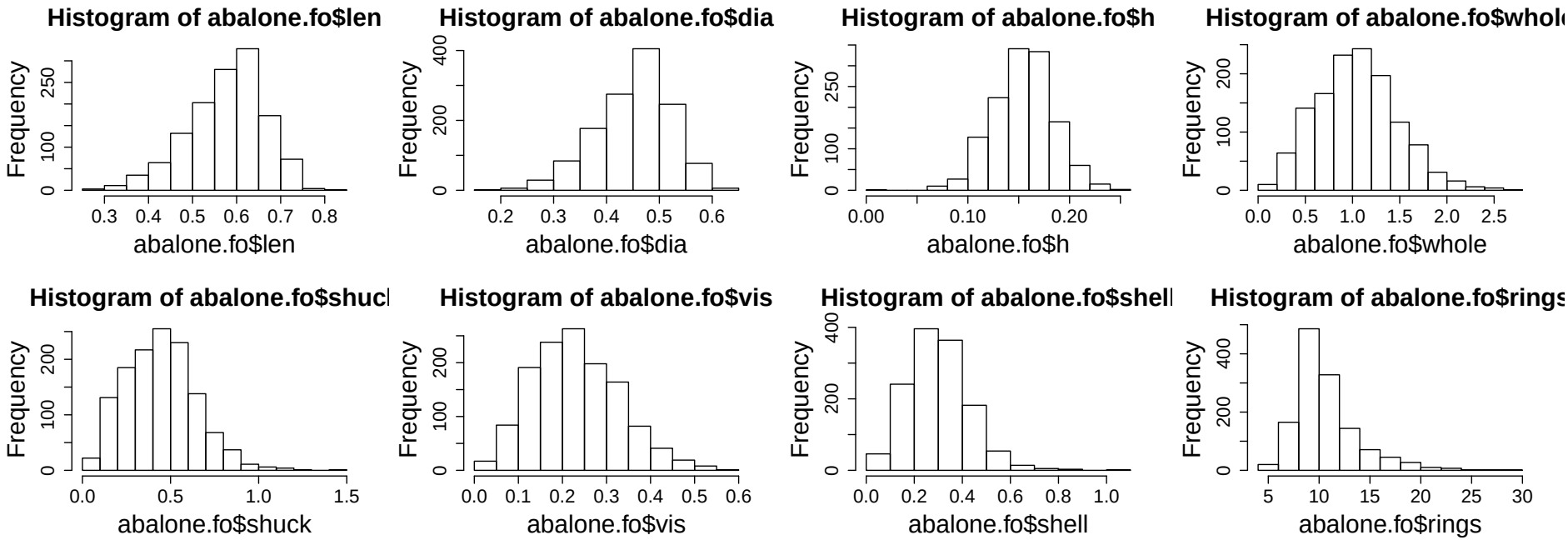
Remove outlier in height

```
temp<-max(abalone.f$h)
ind<-(1:length(abalone.f$h))[abalone.f$h==temp]
abalone.fo<-abalone.f[-ind,]
summary(abalone.fo)
```

```
##       len               dia               h               whole
##  Min.   :0.2750   Min.   :0.1950   Min.   :0.0150   Min.   :0.0800
##  1st Qu.:0.5250   1st Qu.:0.4100   1st Qu.:0.1400   1st Qu.:0.7315
##  Median :0.5900   Median :0.4650   Median :0.1600   Median :1.0385
##  Mean   :0.5792   Mean   :0.4548   Mean   :0.1573   Mean   :1.0469
##  3rd Qu.:0.6400   3rd Qu.:0.5050   3rd Qu.:0.1750   3rd Qu.:1.3204
##  Max.   :0.8150   Max.   :0.6500   Max.   :0.2500   Max.   :2.6570
##      shuck              vis              shell             rings
##  Min.   :0.0310   Min.   :0.0210   Min.   :0.0250   Min.   : 5.00
##  1st Qu.:0.2950   1st Qu.:0.1590   1st Qu.:0.2142   1st Qu.: 9.00
##  Median :0.4405   Median :0.2240   Median :0.2950   Median :10.00
##  Mean   :0.4463   Mean   :0.2308   Mean   :0.3021   Mean   :11.13
##  3rd Qu.:0.5734   3rd Qu.:0.2974   3rd Qu.:0.3750   3rd Qu.:12.00
##  Max.   :1.4880   Max.   :0.5900   Max.   :1.0050   Max.   :29.00
```

Marginal histograms

```
bigpar(2,4)
hist(abalone.fo$len)
hist(abalone.fo$dia)
hist(abalone.fo$h)
hist(abalone.fo$whole)
hist(abalone.fo$shuck)
hist(abalone.fo$vis)
hist(abalone.fo$shell)
hist(abalone.fo$rings)
```



Check for discreteness

```
out.unique<-c(length(unique(abalone.fo$len)),
length(unique(abalone.fo$dia)),
length(unique(abalone.fo$h)),
length(unique(abalone.fo$whole)),
length(unique(abalone.fo$shuck)),
length(unique(abalone.fo$vis)),
length(unique(abalone.fo$shell)),
length(unique(abalone.fo$rings)))
names(out.unique)<-c("len","dia","h","whole","shuck","vis","shell","rings")
out.unique
```

```
##   len   dia     h whole shuck   vis shell rings
##    91    81    38  1072   854   627   482    23
```

The variables h and rings are very discrete, therefore we also consider models where h and rings are considered as ordered

Include ordered versions of rings and h to the data set

```
rings.ord<-ordered(abalone.fo$rings,levels=sort(unique(abalone.fo$rings)))
h.ord<-ordered(abalone.fo$h,levels=sort(unique(abalone.fo$h)))
abalone.fo1<-data.frame(abalone.fo,rings.ord,h.ord)
summary(abalone.fo1[,c("rings.ord","h.ord")])
```

```
##    rings.ord       h.ord
##  10      :248   0.15   :113
##  9       :238   0.175  : 96
##  11      :200   0.16   : 91
##  12      :128   0.165  : 90
##  8       :121   0.17   : 86
##  13      : 88   0.155  : 82
##  (Other):283   (Other):748
```

```
rm(rings.ord)
rm(h.ord)
```

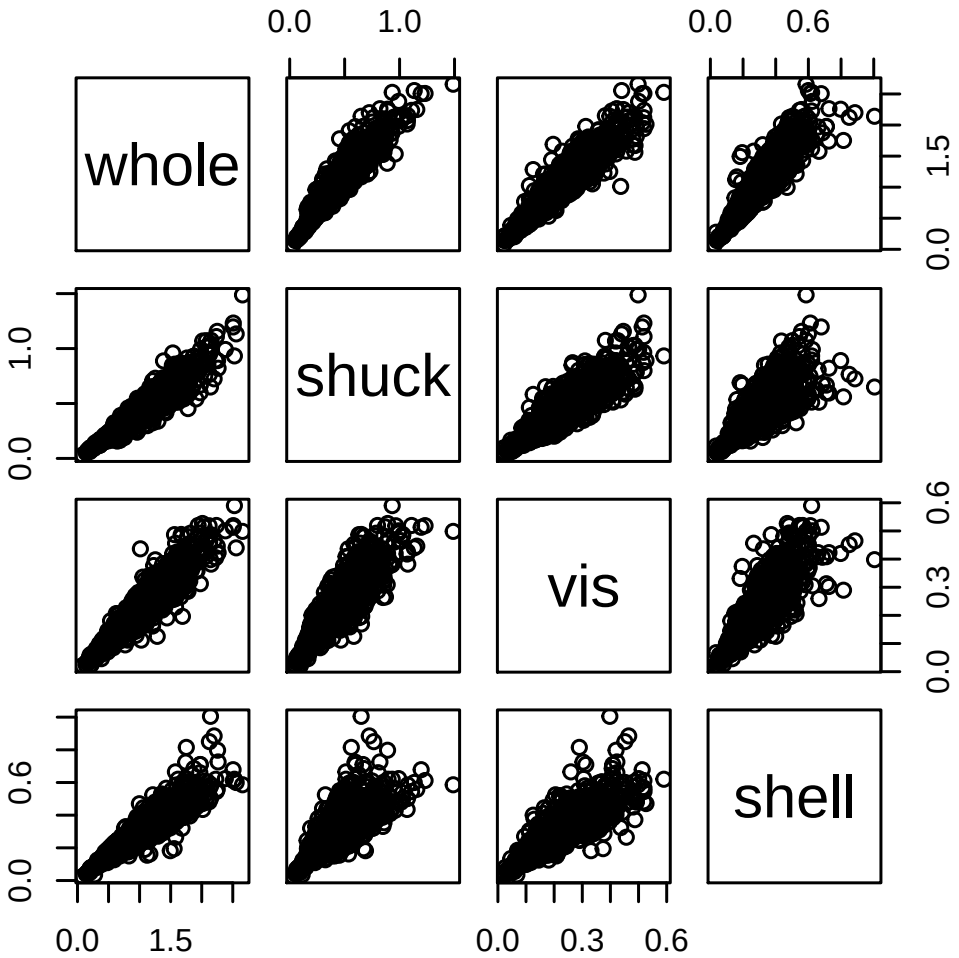
Use emprical cdfs to transform to copula data

```
n<-nrow(abalone.fo)
fak<-n/(n+1)
temp<-ecdf(abalone.fo$len)
u1<-temp(abalone.fo$len)*fak
temp<-ecdf(abalone.fo$dia)
u2<-temp(abalone.fo$dia)*fak
temp<-ecdf(abalone.fo$h)
u3<-temp(abalone.fo$h)*fak
temp<-ecdf(abalone.fo$whole)
u4<-temp(abalone.fo$whole)*fak
temp<-ecdf(abalone.fo$shuck)
u5<-temp(abalone.fo$shuck)*fak
temp<-ecdf(abalone.fo$vis)
u6<-temp(abalone.fo$vis)*fak
temp<-ecdf(abalone.fo$shell)
u7<-temp(abalone.fo$shell)*fak
temp<-ecdf(abalone.fo$rings)
u8<-temp(abalone.fo$rings)*fak
udata.fo<-cbind(u1,u2,u3,u4,u5,u6,u7,u8)
colnames(udata.fo)<-c("len","dia","h","whole","shuck","vis","shell","rings")
udata.fo<-as.copuladata(udata.fo)
```

4. Parametric vine analysis for four variables

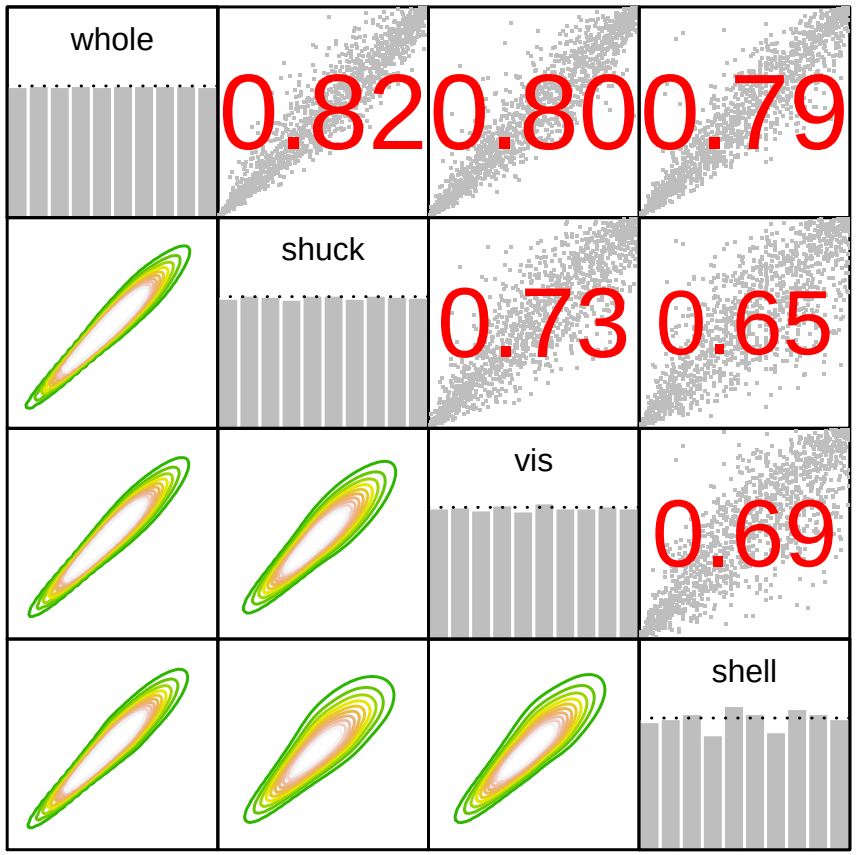
EDA for whole, shuck, vis and shell (x-level)

```
abalone.f4<-abalone.fo[,4:7]
pairs(abalone.f4)
```



Empirical normalized contour plots (z-level)

```
udata.f4<-udata.fo[,4:7]
pairs(udata.f4)
```



Pairwise Kendalls'tau

```
round(cor(udata.f4,method="kendall"),digits=2)
```

```
##        whole shuck  vis shell
## whole   1.00  0.82 0.80  0.79
## shuck   0.82  1.00 0.73  0.65
## vis     0.80  0.73 1.00  0.69
## shell   0.79  0.65 0.69  1.00
```

Fit parametric R vine using RVineStructureSelect(all families, no check for independence copula)

```
f.rv.all=RVineStructureSelect(udata.f4, familyset=NA,
selectioncrit="BIC",indeptest=FALSE, level=0.05)
summary(f.rv.all,detail=T)
```

```
## tree     edge | family        cop    par  par2 |   tau   utd   ltd
## ------------------------------------------------------------------
##    1      1,2 |    214  Tawn2_180   5.94  0.98 |  0.82     -  0.87
##           1,3 |    214  Tawn2_180   5.10  0.99 |  0.80     -  0.85
##           4,1 |    114    Tawn180   5.01  0.99 |  0.79     -  0.85
##    2    4,2;1 |      2          t  -0.65  5.06 | -0.45  0.00  0.00
##         4,3;1 |    134    Tawn270  -1.38  0.34 | -0.14     -     -
##    3  3,2;4,1 |      2          t  -0.40  4.50 | -0.26  0.01  0.01
## ---
## type: C-vine    logLik: 5348.73    AIC: -10673.45    BIC: -10611.36
## ---
## 1 <-> whole,   2 <-> shuck,   3 <-> vis,   4 <-> shell
```

Fit other parametric R vine using RVineStructureSelect

```
f.rv.all.ind=RVineStructureSelect(udata.f4, familyset=NA,
selectioncrit="BIC",indeptest=TRUE, level=0.05)
f.rv=RVineStructureSelect(udata.f4, familyset=1:6,
selectioncrit="BIC",indeptest=FALSE, level=0.05)
f.rv.ind=RVineStructureSelect(udata.f4, familyset=1:6,
selectioncrit="AIC", indeptest=TRUE, level=0.05)
f.Gauss=RVineStructureSelect(udata.f4, familyset=1, selectioncrit="AIC",
indeptest=FALSE, level=0.05)
f.Gauss.ind=RVineStructureSelect(udata.f4, familyset=1, selectioncrit="AIC",
indeptest=TRUE, level=0.05)
```

Output function for comparing model fits

```
vine.out<-function(fit=fit.rv,data=ured,digits=2){
df<-sum(abs(fit$par)>0)+sum(fit$par2>0)
out<-round(c(RVineLogLik(data, fit)$loglik,df,
    RVineAIC(data,fit)$AIC,RVineBIC(data,fit)$BIC),digits)
names(out)<-c("loglik","par","AIC","BIC")
out
}
```

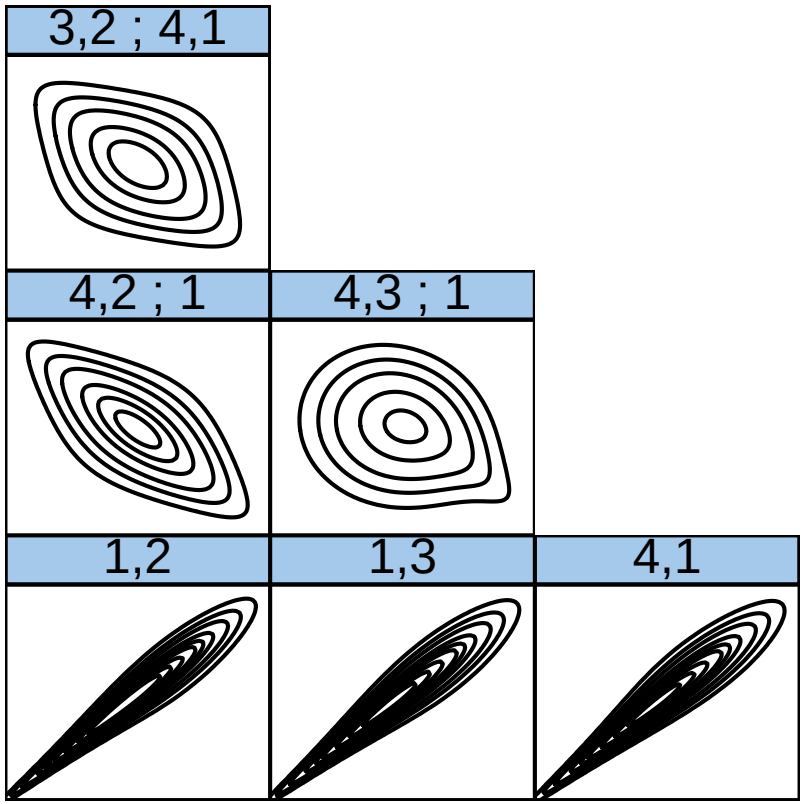
Model selection based on AIC and BIC

```
out.table<-rbind(
vine.out(fit=f.rv.all,data=udata.f4),
vine.out(fit=f.rv.all.ind,data=udata.f4),
vine.out(fit=f.rv,data=udata.f4),
vine.out(fit=f.rv.ind,data=udata.f4),
vine.out(fit=f.Gauss,data=udata.f4),
vine.out(fit=f.Gauss.ind,data=udata.f4))
row.names(out.table)<-c("R-vine-all-seq","R-vine-all-seq",
        "R-vine-seq","R-vine-ind-seq","Gauss-seq","Gauss-ind-seq")
out.table
```

```
##                  loglik par       AIC       BIC
## R-vine-all-seq 5348.73  12 -10673.45 -10611.36
## R-vine-all-seq 5348.73  12 -10673.45 -10611.36
## R-vine-seq     5287.68   8 -10559.35 -10517.95
## R-vine-ind-seq 5287.68   8 -10559.35 -10517.95
## Gauss-seq      4784.00   6  -9556.01  -9524.96
## Gauss-ind-seq  4784.00   6  -9556.01  -9524.96
```

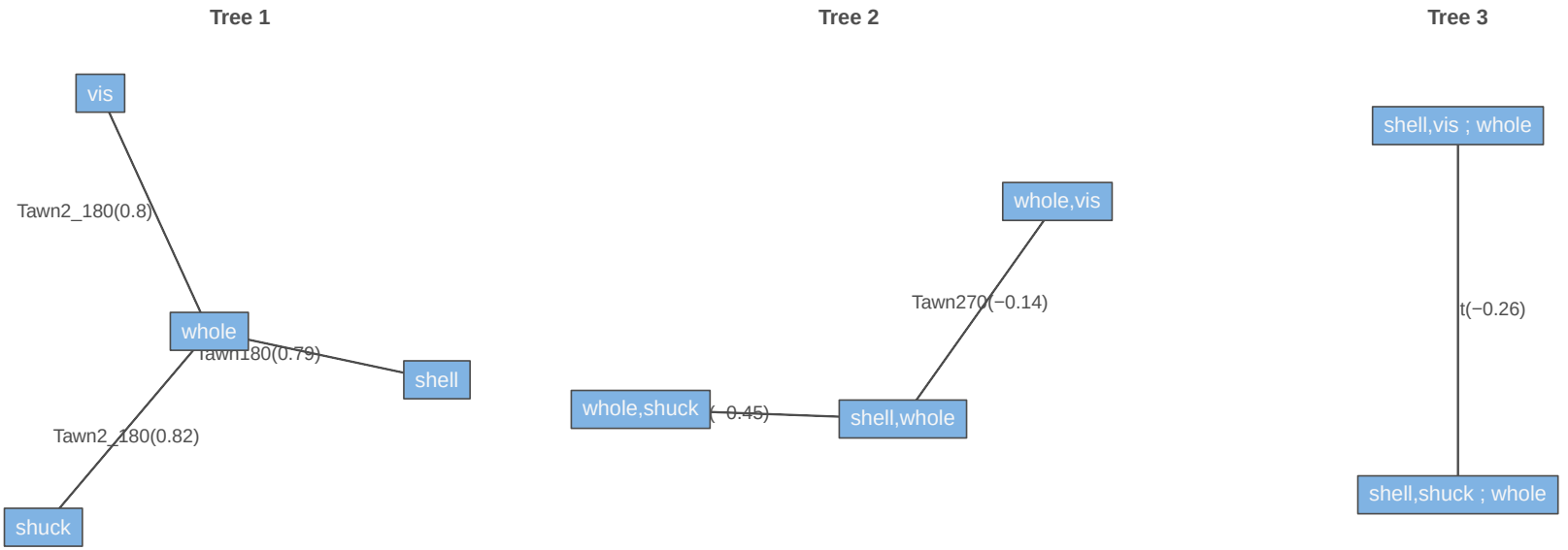
Fitted contour plot of parametric fits

```
contour(f.rv.all)
```



Tree plots of parametric fits

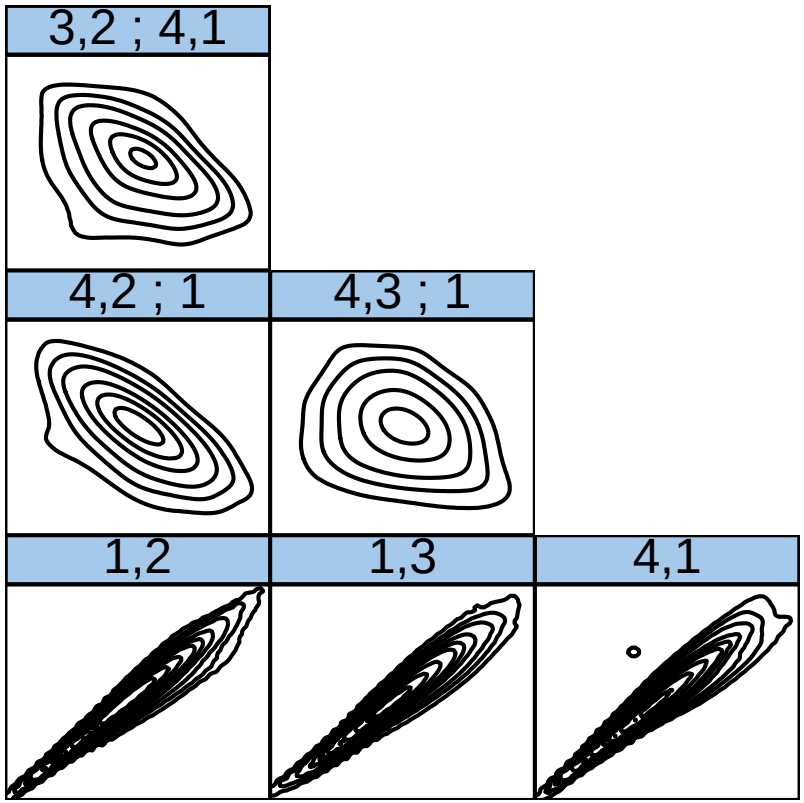
```
par(mfrow=c(1,3))
plot(f.rv.all,edge.labels = "family-tau",type=1)
```



5. Non parametric vine analysis for four variables

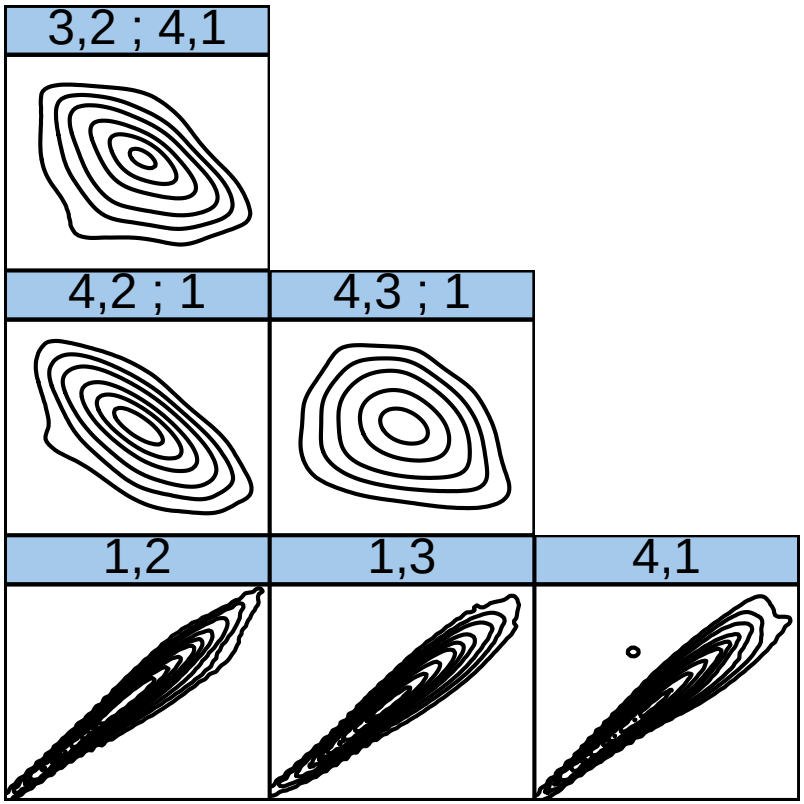
Fit non parametric vine

```
f.np <- kdevinecop(udata.f4)
contour(f.np)
```



Comparison of fitted contours from non parametric and parametric fit

```
contour(f.np)
```




```
contour(f.rv.all)
```