NNS Multivariate Dependence vs. multivariance package

Intro

There's a new companion package `multivariance` in R to research demonstrating multivariate dependence,[1]

Distance multivariance is a measure of dependence which can be used to detect and quantify dependence.

`NNS` has had this capability for years![2][3]

We will show some known cases in 3 dimensions to illustrate the difference in capabilities between the two methods, then use the example found within the `multivariance` package.

Load Packages `NNS (>= 4.5)`

```
require(devtools); install_github('OVVO-Financial/NNS', ref = "NNS-Beta-Version")
library(NNS)
library(data.table)
library(rgl)
library(multivariance)
```

[1] Dependence and Dependence Structures: Estimation and Visualization Using Distance Multivariance. <arXiv:1712.06532>.

[2] Deriving Nonlinear Correlation Coefficients from Partial Moments https://ssrn.com/abstract=2148522

[3] Beyond Correlation: Using the Elements of Variance for Conditional Means and Probabilities https://ssrn.com/abstract=2745308

Case 1: Total Dependence and Positive Correlation

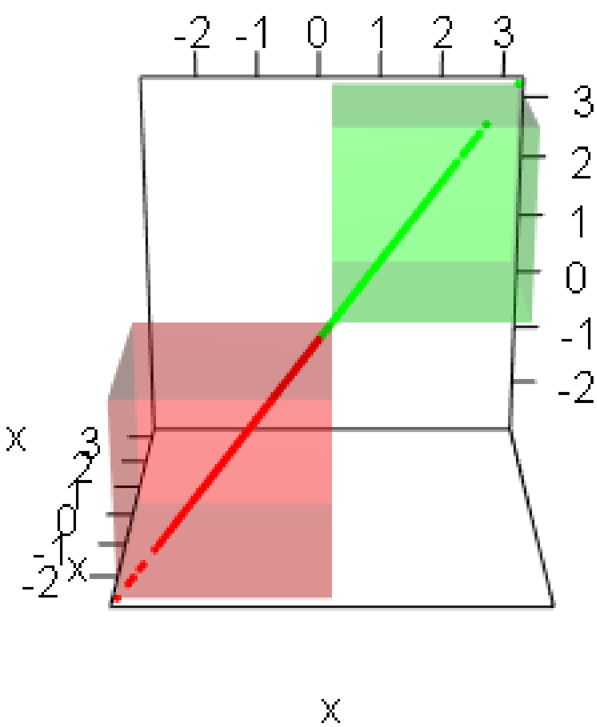
The easiest way to demonstrate this is to show a straight line in 3 dimensions, where the variables would axiomatically be dependent upon one another. All observations would occupy the Co-Upper Partial Moment quadrant (green) or Co-Lower Partial Moment quadrant (red).

```r
set.seed(123)

x <- rnorm(1000)

x3 <- cbind(x,x,x)

multicorrelation(x3)
```

```
## multicorrelation.2
##                  1
```

```r
NNS.dep.hd(x3, plot = TRUE, independence.overlay = TRUE)
```

```
## $actual.observations
## [1] 1000
##
## $independent.null
## [1] 250
##
## $Dependence
## [1] 1
```



Case 2: Total Dependence and Negative Correlation

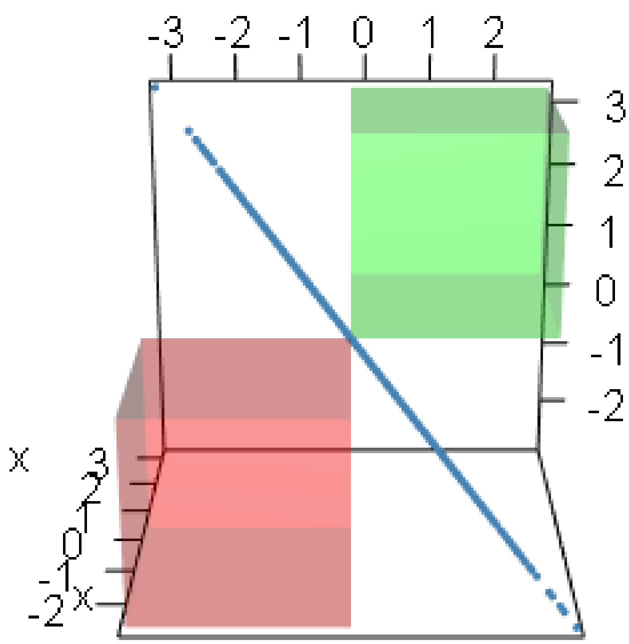
This case too would be a straight line in 3 dimensions, however, it would span the divergent partial moment quadrants (non-highlighted ones).

```
set.seed(123)

x <- rnorm(1000)

x3 <- cbind(-x,x,x)

multicorrelation(x3)
```

```
## multicorrelation.2
##                  1
```

```
NNS.dep.hd(x3, plot = TRUE, independence.overlay = TRUE)
```

```
## $actual.observations
## [1] 0
##
## $independent.null
## [1] 250
##
## $Dependence
## [1] 1
```



Case 3: Independence

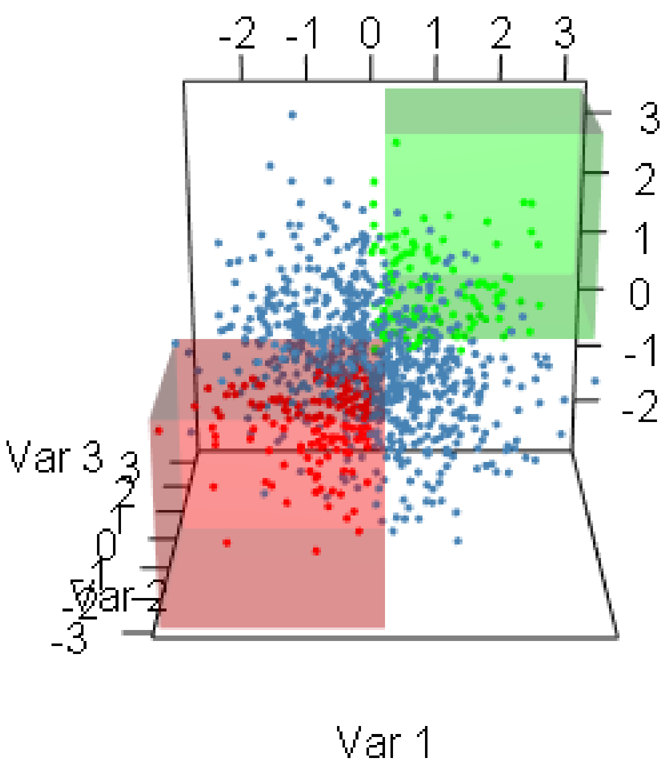
3 Normal random variables…

```
set.seed(123)

x3 <- cbind(rnorm(1000),rnorm(1000),rnorm(1000))

multicorrelation(x3)
```

```
## multicorrelation.2
##         0.00407423
```

```
NNS.dep.hd(x3, plot = TRUE, independence.overlay = TRUE)
```

```
## $actual.observations
## [1] 267
##
## $independent.null
## [1] 250
##
## $Dependence
## [1] 0.02266667
```



So far so good!

Case 4: `multivariance` Example

```r
y = rnorm(100)
x = cbind(y,y*2,(y-2)/3,y+1,y*5)

multicorrelation(x)
```

```
## multicorrelation.2
##                  1
```

```r
NNS.dep.hd(x, plot = TRUE, independence.overlay = TRUE)
```

```
## $actual.observations
## [1] 100
##
## $independent.null
## [1] 6.25
##
## $Dependence
## [1] 1
```

Still good! That's all of the observations within the Co-Partial Moment quadrants per Case 1.

Case 5: `multivariance` Example Expanded

Let's add some more terms to the matrix, and add more observations to eliminate any small sample concerns.

```r
y = rnorm(1000)
x = cbind(y,y*2,(y-2)/3,y+1,y*5,y^2,-y^3,-3*y^4,(y^5-4)/3)

NNS.dep.hd(x, plot = TRUE, independence.overlay = TRUE)
```

```
## $actual.observations
## [1] 0
##
## $independent.null
## [1] 3.90625
##
## $Dependence
## [1] 1
```

```r
multicorrelation(x)
```

```
## multicorrelation.2
##          0.6288725
```

Wait…what happened? All the variables are still very much related according to the pairwise `NNS.dep` measures.[4]

[4] Nonlinear Correlation and Dependence Using NNS https://ssrn.com/abstract=3010414

```r
round(NNS.dep(x)$Dependence, 3)
```

```
##       y
## y 1.000 1.000 1.000 1.000 1.000 0.971 0.917 0.920 0.885
##   1.000 1.000 1.000 1.000 1.000 0.971 0.917 0.920 0.885
##   1.000 1.000 1.000 1.000 1.000 0.971 0.917 0.920 0.885
##   1.000 1.000 1.000 1.000 1.000 0.971 0.917 0.920 0.885
##   1.000 1.000 1.000 1.000 1.000 0.971 0.917 0.920 0.885
##   0.971 0.971 0.971 0.971 0.971 1.000 0.987 0.977 0.830
##   0.917 0.917 0.917 0.917 0.917 0.987 1.000 0.988 0.974
##   0.920 0.920 0.920 0.920 0.920 0.977 0.988 1.000 0.988
##   0.885 0.885 0.885 0.885 0.885 0.830 0.974 0.988 1.000
```

There was ***not a single observation*** in the Co-Partial Moment quadrants versus the expectation of 3.90625 total observations in those quadrants, per Case 2.

Let's increase the number of observations...

```r
y = rnorm(10000)
x = cbind(y,y*2,(y-2)/3,y+1,y*5,y^2,-y^3,-3*y^4,(y^5-4)/3)

NNS.dep.hd(x, plot = TRUE, independence.overlay = TRUE)
```

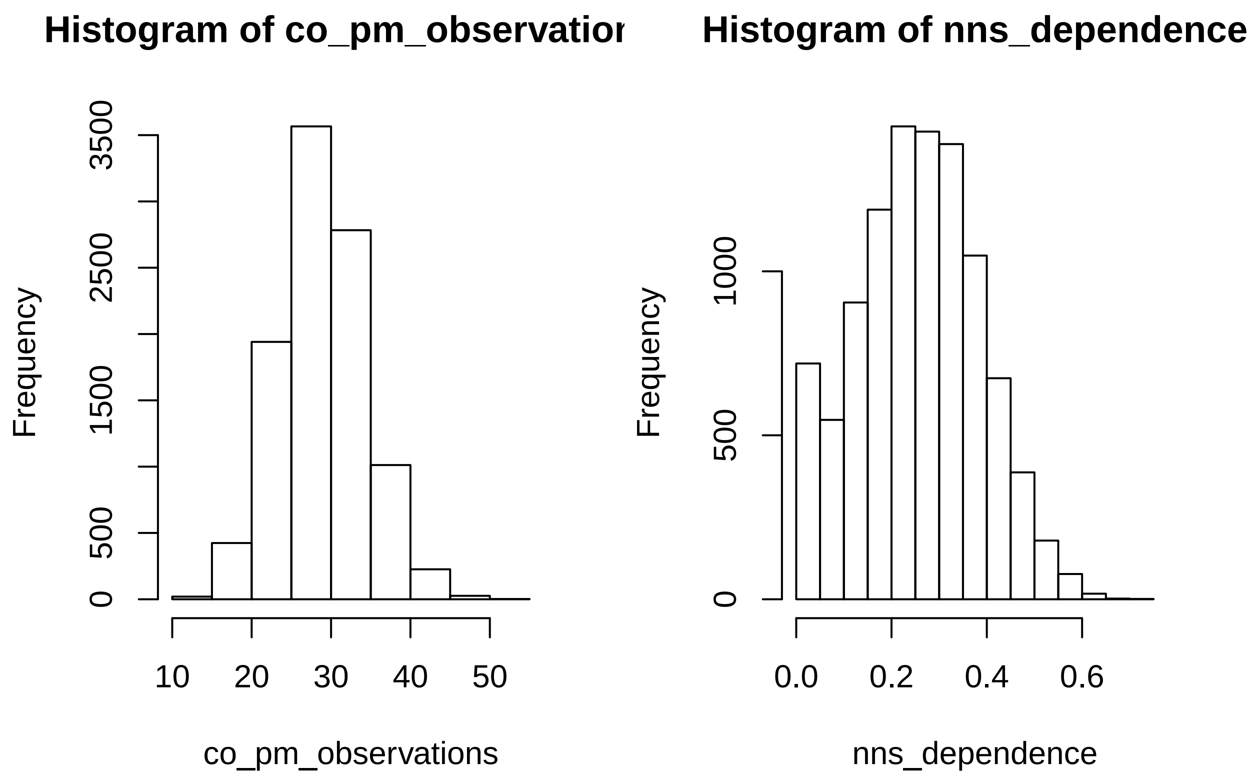
```
## $actual.observations
## [1] 0
##
## $independent.null
## [1] 39.0625
##
## $Dependence
## [1] 1
```

How Likely is This Result?

Like other frequency based metrics (mutual information, etc.) in order to ascertain a p-value, we need to resample the statistic over random permutations. Let's try it and see how often we observe 0 observations in both quadrants under independence...

```r
samples <- list()


for(j in 1:10000){
    x2 <- list()

    for (i in 1:ncol(x)) x2[[i]] <- x[order(runif(length(x[,i]))),i]

    x2 <- do.call(cbind,x2)
    samples[[j]] <- NNS.dep.hd(x2,plot=FALSE)
}

co_pm_observations <- unlist(lapply(samples, `[[`, 1))
nns_dependence <- unlist(lapply(samples, `[[`, 3))

par(mfrow=c(1,2))
hist(co_pm_observations)
```

```r
hist(nns_dependence)
```




```r
# Percentage of observations = 0
LPM(0,0,co_pm_observations)
```

```
## [1] 0
```

```r
# Maximum nns_dependence
max(nns_dependence)
```

```
## [1] 0.744
```

Again, in our initial result there was ***not a single observation*** in those quadrants. The least amount of observations in the Co-Partial Moment quadrants from our resampling was 10. That translates to a p-value of 0 for 0 observations. It also translates to a p-value equal to 0 for the `NNS` multivariate dependence measure.

In the absence of independence, there is dependence.

Timing

Using the prior example settings of $n = 1,000$, let's compare the timing of the results for each.

```r
library(microbenchmark)
y = rnorm(1000)
x = cbind(y,y*2,(y-2)/3,y+1,y*5,y^2,-y^3,-3*y^4,(y^5-4)/3)

microbenchmark(NNS=NNS.dep.hd(x),multivariance=multicorrelation(x),times = 100)
```

```
## Unit: milliseconds
##           expr      min       lq      mean   median        uq      max
##            NNS   2.0731   2.1710   2.35335   2.2711   2.49265   3.3437
##  multivariance 258.5635 263.2559 270.44039 264.8380 268.10390 438.0802
##  neval
##    100
##    100
```

Comments

I look forward to further discussions and collaboration with those equally as passionate about these issues, and open to embracing alternative solutions. If you found this presentation interesting or useful, please feel free to reach out via e-mail: ovvo.financial.systems@gmail.com

Thanks for your interest!