Nonlinear Correlation and Dependence Using NNS

Viole, Fred
ovvo.financial.systems@gmail.com

September 10, 2019

Install NNS

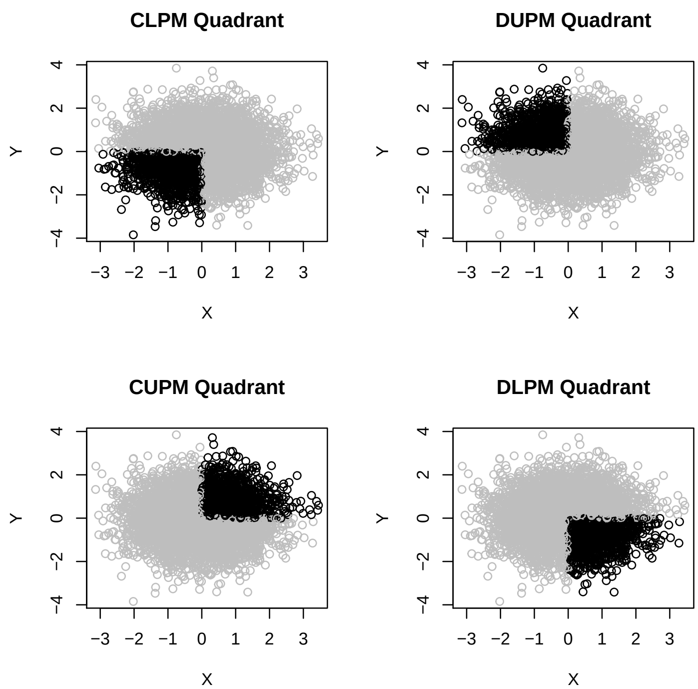
We need the latest version of NNS available on GitHub.

```
> #library(devtools); install_github('OVVO-Financial/NNS', ref = "NNS-Beta-Version")
> library(NNS)
> library(rgl)
```

Correlation

NNS nonlinear correlation is based on partial moment matrices. Partial moment matrices reflect the relationship between variables. Using these topological relations, we are able to avoid relying upon geometry for correlation. There are only 4 relationships between 2 variables and each relationship is represented by a partial moment matrix:

$X \leq target, Y \leq target \rightarrow CLPM$ matrix
$X \leq target, Y > target \rightarrow DUPM$ matrix
$X > target, Y \leq target \rightarrow DLPM$ matrix
$X > target, Y > target \rightarrow CUPM$ matrix



Linear Equivalence

First, let's demonstrate NNS equivalence to Pearson's correlation coefficient under a condition of perfect linearity.

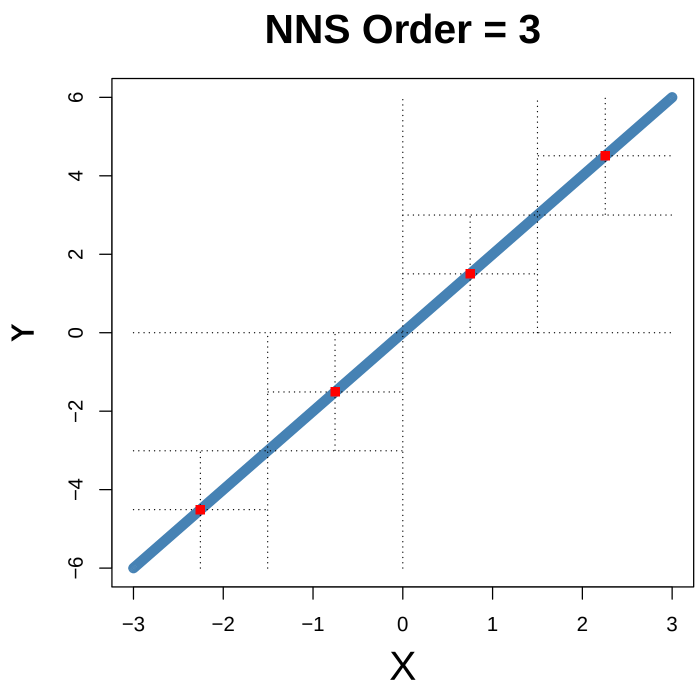
```
> x = seq(-3, 3, .01); y = 2*x
> cor(x,y)


[1] 1


> NNS.dep(x,y, print.map = TRUE, order=3)


$Correlation
[1] 1

$Dependence
[1] 1
```



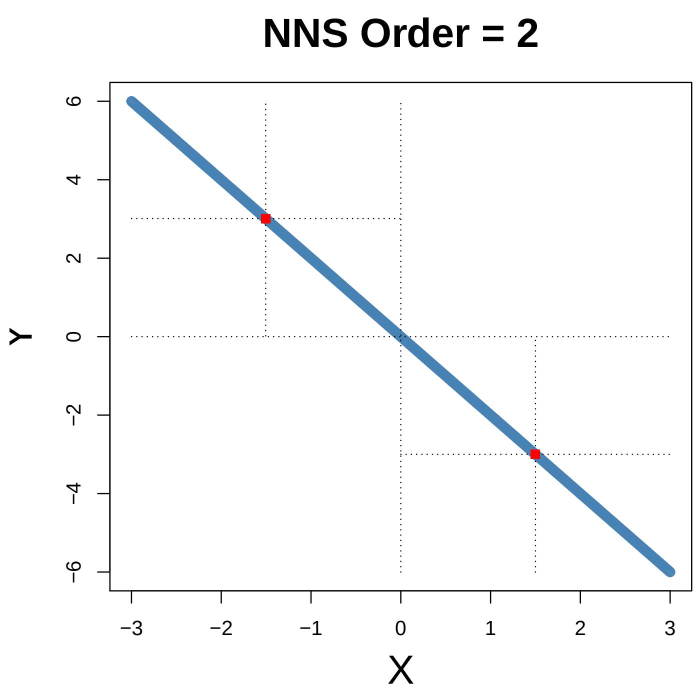
Note how under perfect correlation, all of the observations are in the similar CUPM and CLPM quadrants.

```
> x = seq(-3, 3, .01); y = -2*x
> cor(x,y); NNS.dep(x,y, print.map = TRUE)

[1] -1

$Correlation
[1] -1

$Dependence
[1] 1
```



Note how under perfect inverse correlation, all of the observations are in the divergent DUPM and DLPM quadrants.

Nonlinear Examples

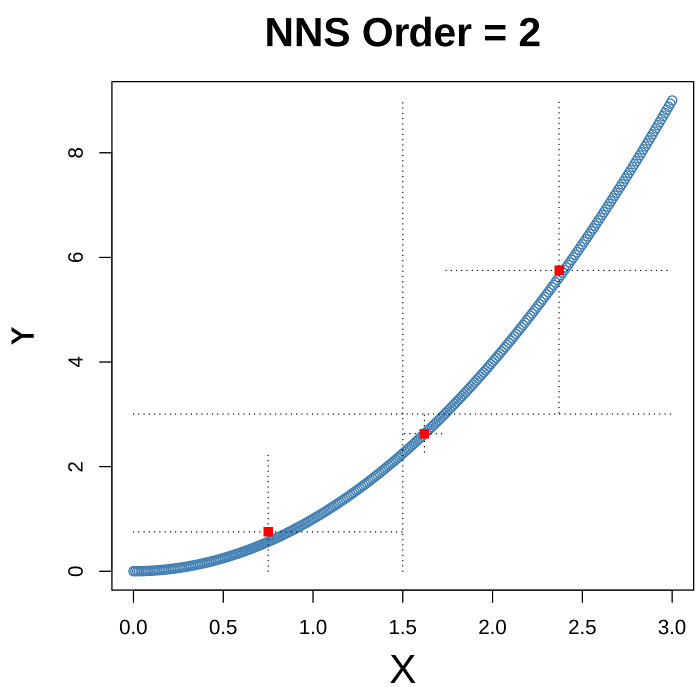
Now that you understand the relationship between the relationship matrices, we can introduce some nonlinearity to the analysis.

```
> x = seq(0, 3, .01); y = x^2
> cor(x,y); NNS.dep(x,y, print.map = TRUE)
```

```
[1] 0.9680452


$Correlation
[1] 0.9928571

$Dependence
[1] 0.9928571
```



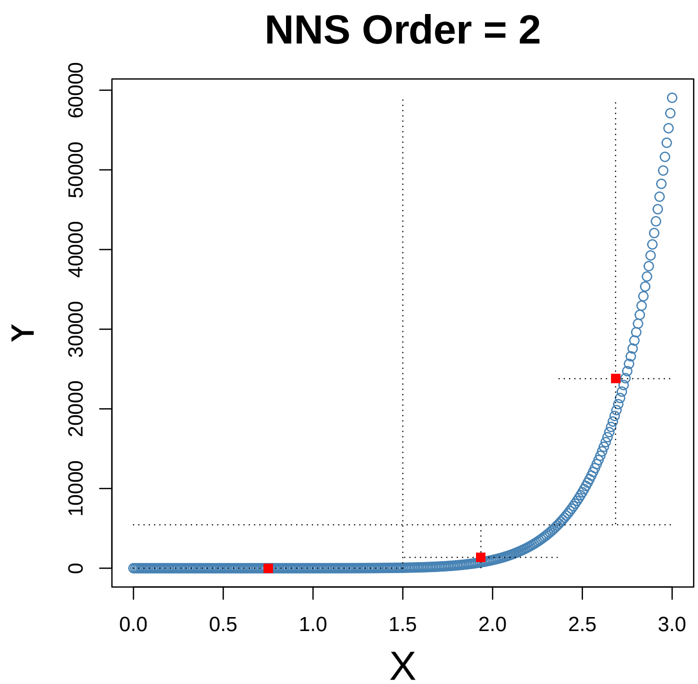
Both methods report a high correlation for the previous gentle example. Increasing the nonlinearity will reveal the deficiencies in the geometric based linear approach.

```
> x = seq(0, 3, .01); y = x^10
> cor(x,y); NNS.dep(x,y, print.map = TRUE)
```

```
[1] 0.6610183


$Correlation
[1] 0.9142776

$Dependence
[1] 0.9142776
```



Again, even though the observations do not follow a straight line, they occur primarily in the CUPM and CLPM quadrants, yielding a high NNS correlation coefficient of versus a less significant 0.6610183 for Pearson.

Clustered Examples

In highly clustered datasets, linear correlation is essentially useless. Here is an example showing 3 distinct clusters and the significant inverse correlation Pearson's inappropriately assigns (-0.6275592).

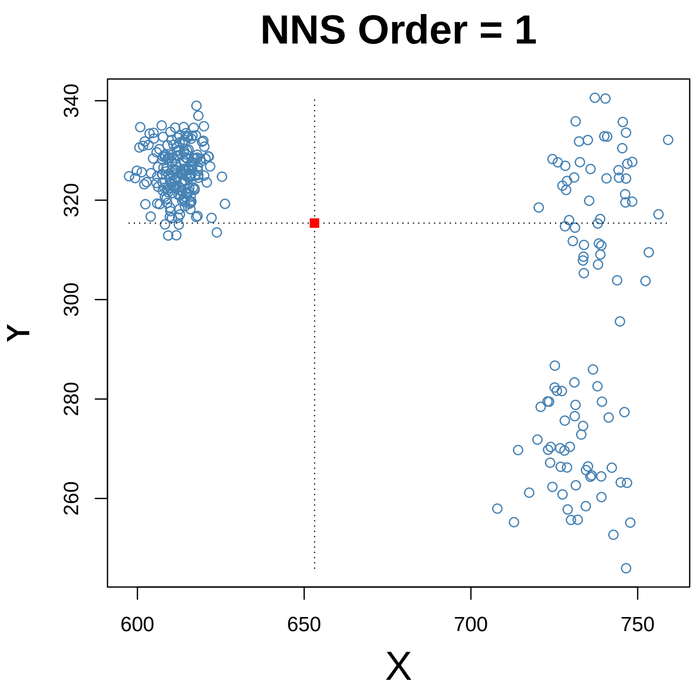
```
> cluster.df <- read.csv("https://goo.gl/AA6HMf", header=F, sep = '')
> cor(cluster.df[,3], cluster.df[,4])

[1] -0.6275592

> NNS.dep(cluster.df[,3], cluster.df[,4], print.map = TRUE)

$Correlation
[1] -0.03082504

$Dependence
[1] 0.2406689
```



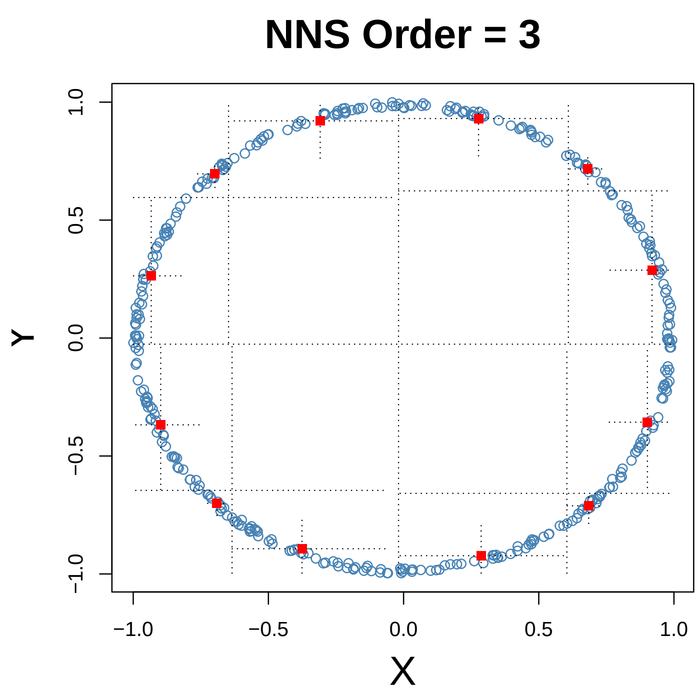
Dependence

The spherical distribution is the best example of the dependence technique. Many cite it as a counter-example to the applicability of Pearson's correlation when trying to determine the relationship between two variables.

```
> set.seed(123)
> df <- data.frame(x=runif(10000,-1,1), y=runif(10000,-1,1))
> df <- subset(df, (x^2 + y^2 <= 1 & x^2 + y^2 >= 0.95))
> NNS.dep(df$x,df$y, print.map = TRUE)
```

```
$Correlation
[1] -0.001657492

$Dependence
[1] 0.9104624
```



While there is no correlation to speak of in this spherical distribution, there is obvious dependence between X and Y. This dependence is accurately reported by NNS.

A majority of observations are in either similar (CUPM, CLPM) partial moment quadrants or divergent (DUPM,DLPM) quadrants.

Below is another example of 0 correlation, yet full dependence on a periodic sine wave.

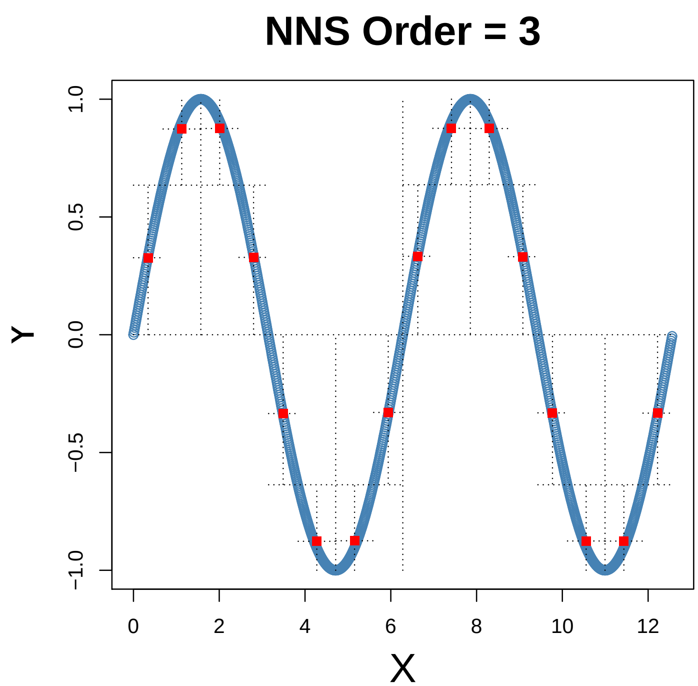
```
> x = seq(0, 4*pi, .01); y=sin(x)
> cor(x,y)


[1] -0.3896787


> NNS.dep(x,y, print.map = TRUE)


$Correlation
[1] 0.5721681

$Dependence
[1] 0.972133
```

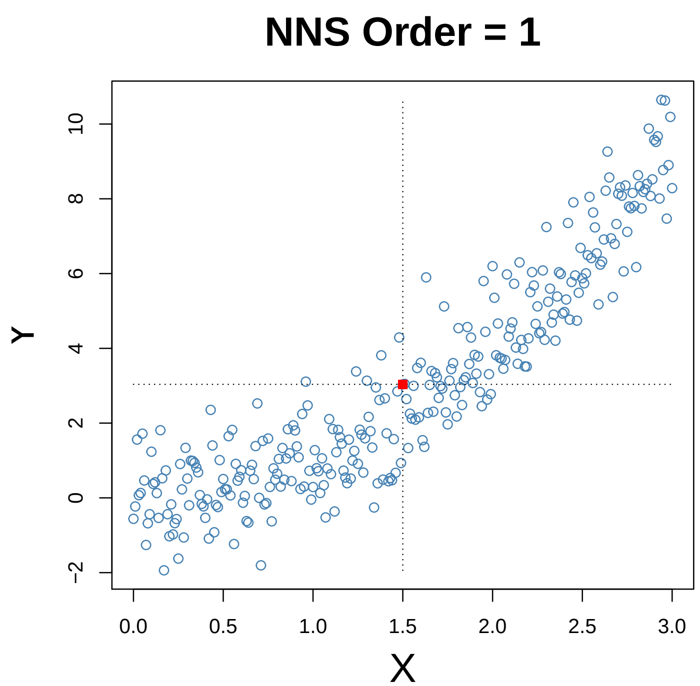


Here is our earlier $y = x^2$ example, where the dispersion around the line effects the dependence...

```
> set.seed(123)
> x = seq(0, 3, .01); y = x^2 + rnorm(length(x), sd=1)
> NNS.dep(x,y, print.map = TRUE)
```

```
$Correlation
[1] 0.3810286

$Dependence
[1] 0.4792573
```



Multivariate Dependence

These partial moment insights permit us to extend the analysis to multivariate instances. This level of analysis is simply impossible with Pearson or other rank based correlation methods, which are restricted to bi-variate cases.

Independent Case

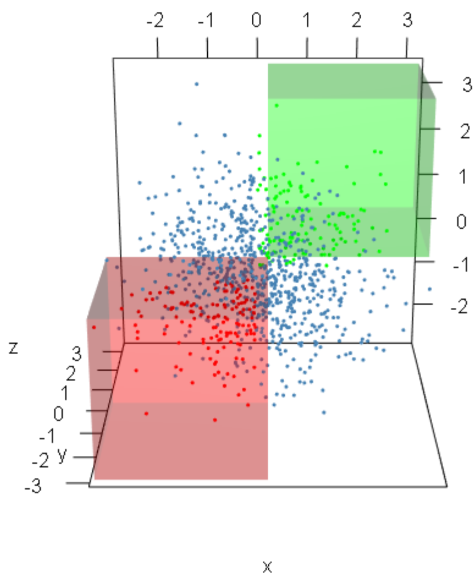
First we present the independent case. We shade the regions of co-partial moments. Green is the Co-UPM and red is the Co-LPM. Under independence, we know that $2(.5^n)$ observations should occupy these regions for n dimensions (independent.null in the output). In our 3 variable normal example, we can see that the green dots and red dots representing Co-UPM and Co-LPM observations respectively are restricted to their appropriate regions.

```
> set.seed(123)
> x = rnorm(1000); y = rnorm(1000); z = rnorm(1000)
> NNS.dep.hd(cbind(x,y,z), plot=TRUE, independence.overlay=TRUE)
```

```
$actual.observations
[1] 267

$independent.null
[1] 250

$Dependence
[1] 0.02266667
```



-2
-1
0
1
2
3
z
y
x

Dependent Case

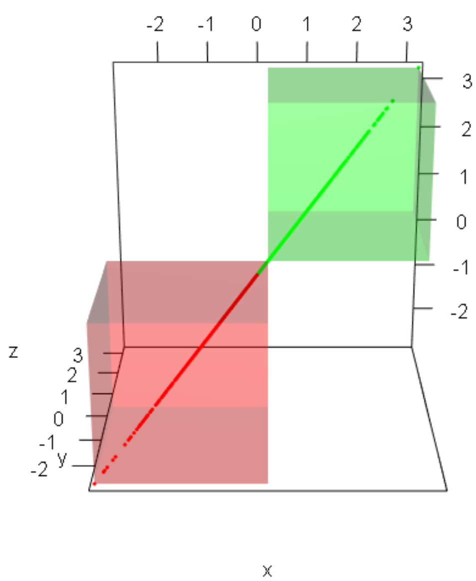
In our dependent case, where all variables are equal, we can see how all of the observations fall within the co-partial moment regions, exhibiting a co-dependence of 1.

```
> y = x; z = x
> NNS.dep.hd(cbind(x,y,z), plot=TRUE, independence.overlay=TRUE)


$actual.observations
[1] 1000

$independent.null
[1] 250

$Dependence
[1] 1
```



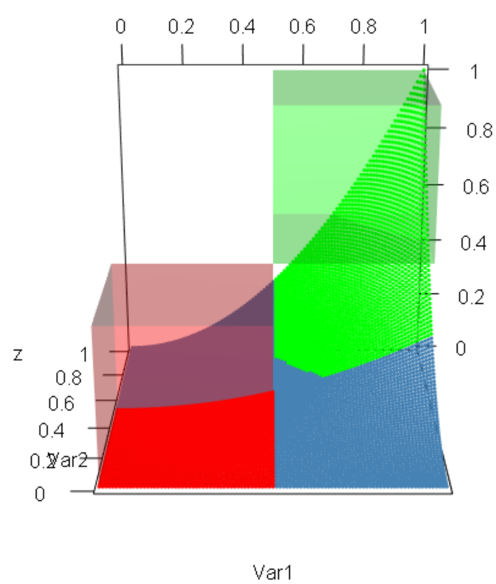
In our second dependent case, we examine at $y = x_1^2 * x_2^2$ where y is clearly dependent upon both x_1 and x_2. Note the abundance of green Co-UPM observations outside of the green Co-UPM independence shaded region.

```
> x = seq(0,1,.01); y = seq(0,1,.01); B = expand.grid(x,y)
> z = B[,1]^2*B[,2]^2
> NNS.dep.hd(cbind(B,z), plot=TRUE, independence.overlay=TRUE)
```

```
$actual.observations
[1] 4988

$independent.null
[1] 2550.25

$Dependence
[1] 0.3186289
```



Significance

One question remaining is, how significant are these results? p-values and confidence intervals can be obtained from sampling random permutations of $y->y_p$ and running NNS.dep(x,y_p) to compare against a null hypothesis of 0 correlation or independence between (x, y).

```
> x = seq(-5,5,.1); y = x^2 + rnorm(length(x))
> original_nns_cor_dep = NNS.dep(x,y)
> original_nns_cor_dep
```

```
$Correlation
[1] -0.009602208

$Dependence
[1] 0.9934588
```

```
> ## Create permutations of y
> y_p = replicate(1000, sample.int(length(y)))
> ## Generate new correlation and dependence measures on each new permutation of y
> nns.mc = apply(y_p,2, function(g) NNS.dep(x,y[g]))
> ## Store results
> nns_cors = unlist(lapply(nns.mc, "[[", 1))
> nns_deps = unlist(lapply(nns.mc, "[[", 2))
```

Correlation

Left tailed correlation p-value

```
> cor_p_value = LPM(0,original_nns_cor_dep$Correlation, nns_cors)
> cor_p_value
```

```
[1] 0.452
```

Right tailed correlation p-value

```
> cor_p_value = UPM(0,original_nns_cor_dep$Correlation, nns_cors)
> cor_p_value
```

```
[1] 0.548
```

Two sided correlation p-value

```
> cor_p_value = UPM(0,abs(original_nns_cor_dep$Correlation), abs(nns_cors))
> cor_p_value
```

```
[1] 0.951
```

We fail to reject the null hypothesis of 0 correlation between (x, y). We can see 0 within the 95% confidence interval as well.

Confidence Intervals: Correlation

For 95th percentile VaR (both-tails) see [LPM.VaR] and [UPM.VaR]

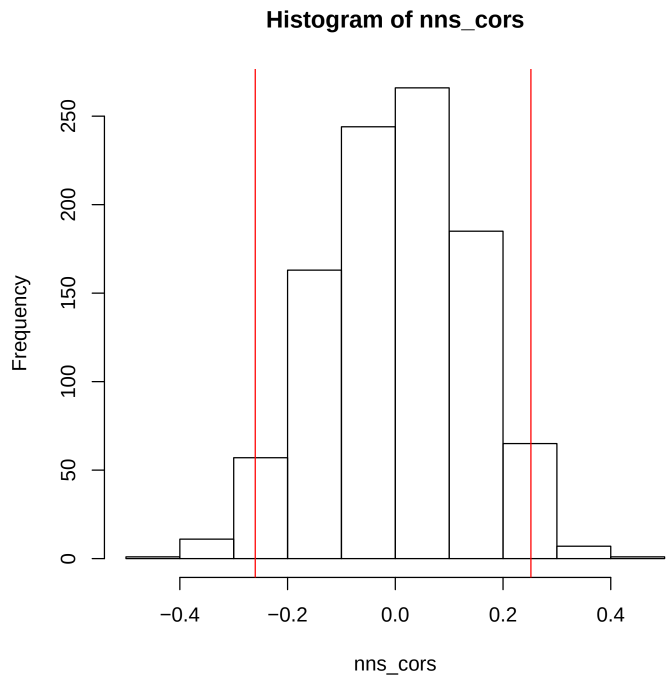
```
> ## Lower CI
> LPM.VaR(.975,0, nns_cors)
```

```
[1] -0.2600105
```

```
> ## Upper CI
> UPM.VaR(.975,0, nns_cors)
```

```
[1] 0.251751
```

```
> hist(nns_cors)
> abline(v = LPM.VaR(.975,0, nns_cors), col = 'red')
> abline(v = UPM.VaR(.975,0, nns_cors), col = 'red')
```



Dependence

Left tailed dependence p-value

```
> dep_p_value = LPM(0,original_nns_cor_dep$Dependence, nns_deps)
> dep_p_value
```

```
[1] 1
```

Right tailed dependence p-value

```
> dep_p_value = UPM(0,original_nns_cor_dep$Dependence, nns_deps)
> dep_p_value
```

```
[1] 0
```

We reject the null hypothesis of independence between (x, y). Our result was 0.99, not even on the histogram of permuted results!

Confidence Intervals: Dependence

For 95th percentile VaR (both-tails) see [LPM.VaR] and [UPM.VaR]

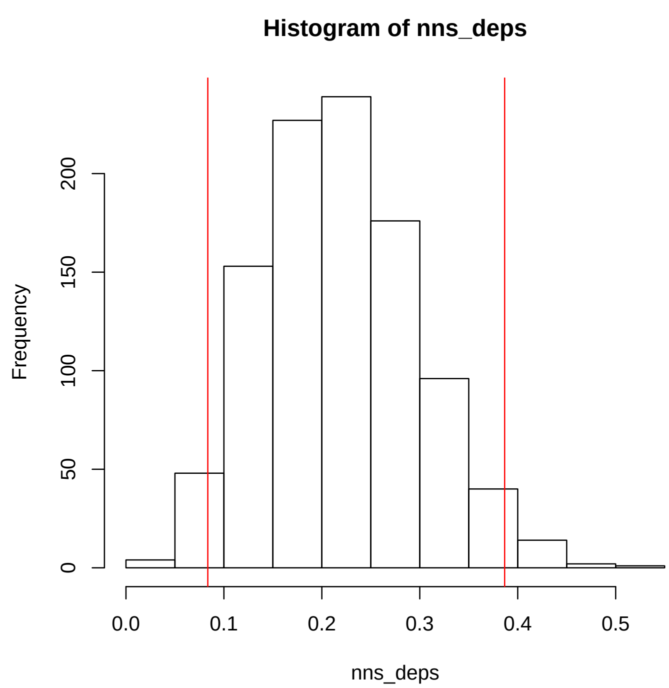
```
> ## Lower CI
> LPM.VaR(.975,0, nns_deps)
```

```
[1] 0.08355445
```

```
> ## Upper CI
> UPM.VaR(.975,0, nns_deps)
```

```
[1] 0.3866296
```

```
> hist(nns_deps)
> abline(v = LPM.VaR(.975,0, nns_deps), col = 'red')
> abline(v = UPM.VaR(.975,0, nns_deps), col = 'red')
```



Key Takeaway

These examples illustrate the main point on dependence:

If X and Y perfectly correlated or perfectly inversely correlated, they are completely dependent upon one another.

Other Considerations

Rank based correlations do not consider the area of the distribution. This potentially ignores a lot of important information when trying to determine the relationship between two variables. NNS has additional features to consider the area of the joint distribution by altering its partial moment degree, 0 for frequency and 1 for area.

Furthermore, pairwise correlation / dependence matrices are automatically generated for NNS when a matrix is entered into the 'NNS.dep()' function. We refer readers to the NNS manual available on CRAN: `https://cran.r-project.org/web/packages/NNS/NNS.pdf`

> To learn more about NNS statistics and their theoretical foundations, see *"Nonlinear Nonparametric Statistics: Using Partial Moments"* available on Amazon: `http://a.co/5bpHvUg`
> Check back to see more NNS examples posted on GitHub: `https://github.com/OVVO-Financial/NNS/tree/NNS-Beta-Version/examples`