CW2

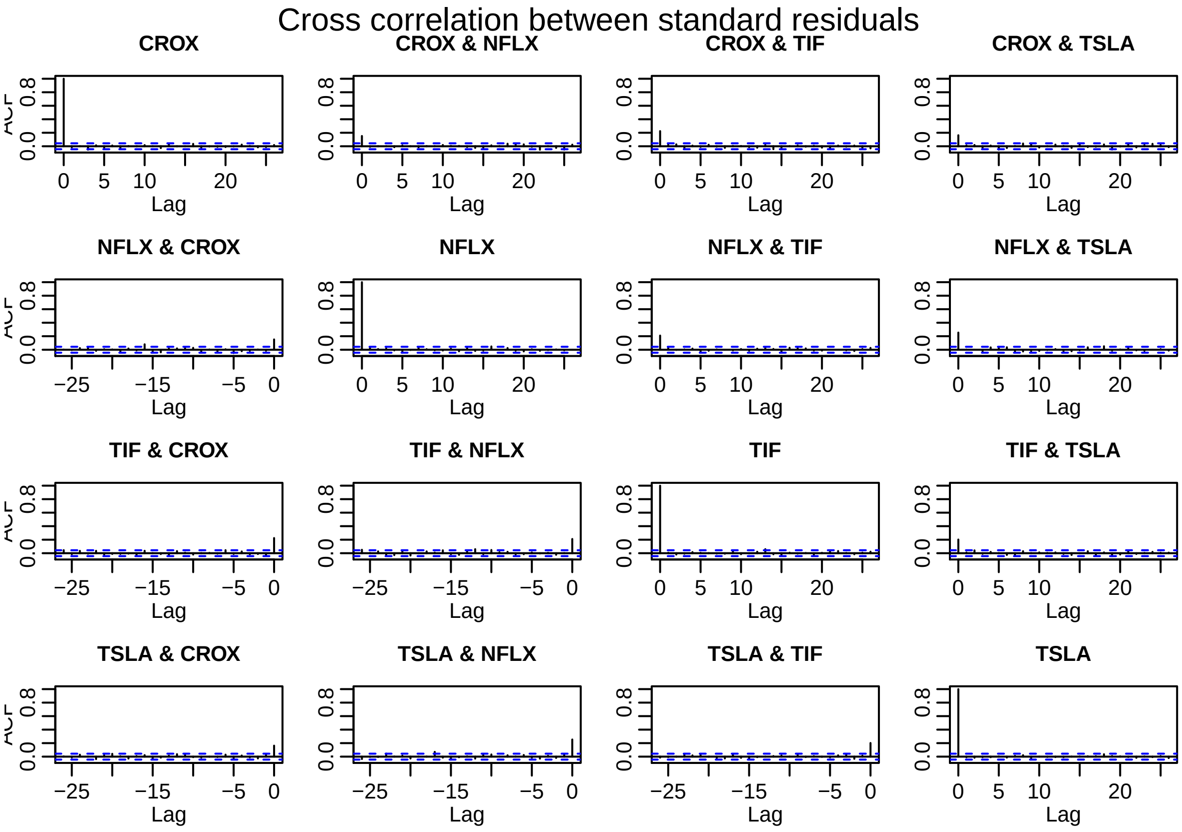
R Markdown

```
std <- read.csv('/Users/pan/Documents/Imperial/Autumn Term/Quantitative Risk Management (MATH97108)/CW2/
head(std)
```

```
##          CROX       NFLX         TIF       TSLA
## 1  0.64468337  0.9539073  0.14936185  0.2407160
## 2 -1.17308305 -1.6052789 -0.56116298 -0.1146408
## 3 -0.17578630  0.1699062 -1.25810787  0.3311566
## 4  0.05726676 -1.2291360  0.08315469 -0.1702118
## 5 -1.81978352  0.5293322 -0.50734052 -0.1048913
## 6  0.07732944  1.0028072 -1.95306753 -0.6457587
```

Cross Correlation

```
acf(std, type = "correlation")
mtext("Cross correlation between standard residuals", line=-1, side=3, outer=TRUE)
```

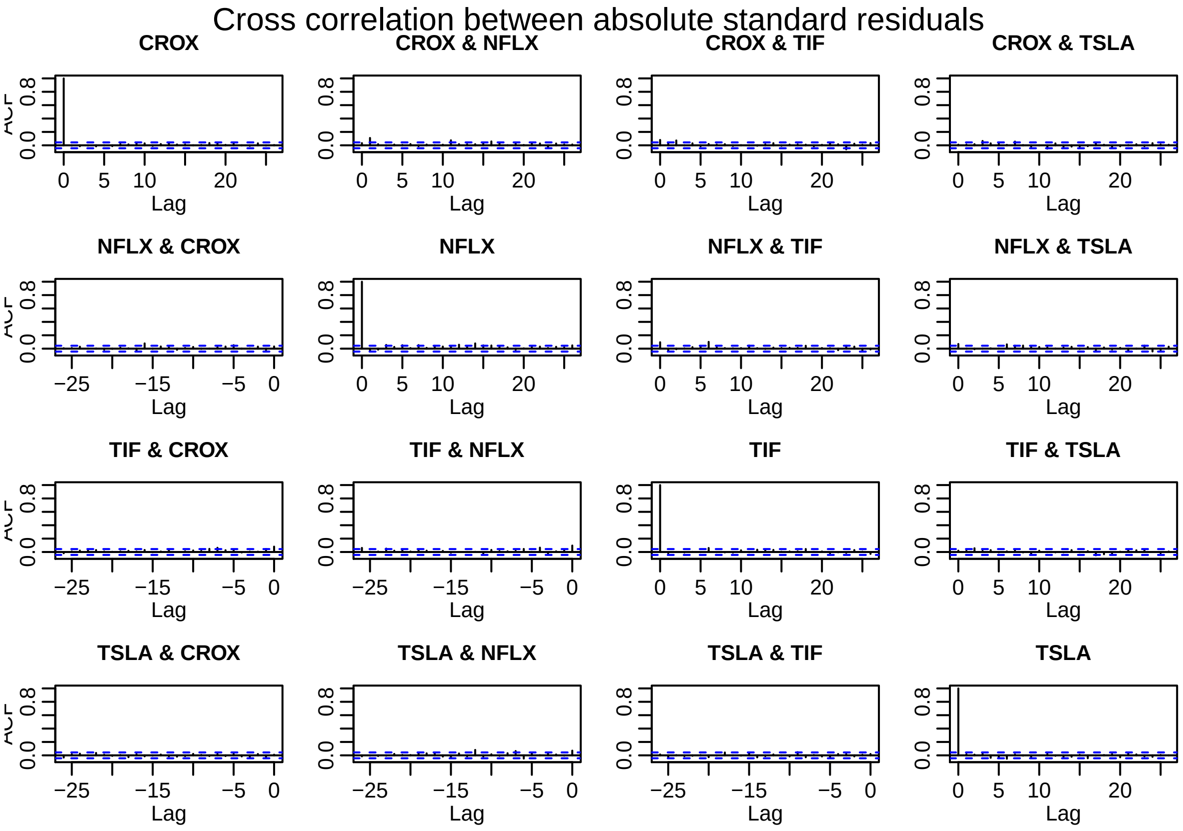


Absolute Cross Correlation

```
ab_std <- read.csv('/Users/pan/Documents/Imperial/Autumn Term/Quantitative Risk Management (MATH97108)/C
head(ab_std)
```

```
##          CROX      NFLX        TIF      TSLA
## 1 0.64468337 0.9539073 0.14936185 0.2407160
## 2 1.17308305 1.6052789 0.56116298 0.1146408
## 3 0.17578630 0.1699062 1.25810787 0.3311566
## 4 0.05726676 1.2291360 0.08315469 0.1702118
## 5 1.81978352 0.5293322 0.50734052 0.1048913
## 6 0.07732944 1.0028072 1.95306753 0.6457587
```

```
acf(ab_std, type = "correlation")
mtext("Cross correlation between absolute standard residuals", line=-1, side=3, outer=TRUE)
```



(iii) fit a Gauss Copula

```
# Spearman's rho
rho <- cor(std, method = "spearman")
rho
```

```
##           CROX      NFLX       TIF      TSLA
## CROX 1.0000000 0.2229531 0.3617210 0.2359355
## NFLX 0.2229531 1.0000000 0.2903952 0.3205309
## TIF  0.3617210 0.2903952 1.0000000 0.2739437
## TSLA 0.2359355 0.3205309 0.2739437 1.0000000
```

```
# Compute Real P
#####
reverse <- function(x) {
  return(2*sin(pi*x/6))
}
```

```
P <- apply(rho, 2,reverse)
P
```

```
##           CROX      NFLX       TIF      TSLA
## CROX 1.0000000 0.2329460 0.3765328 0.2464431
## NFLX 0.2329460 1.0000000 0.3029308 0.3340856
## TIF  0.3765328 0.3029308 1.0000000 0.2858904
## TSLA 0.2464431 0.3340856 0.2858904 1.0000000
```

Fit a Gauss Copula We estimate P by the matrix of pairwise Spearman's rank correlation coefficients

```
library(QRM)
```

```
## Loading required package: gsl

## Loading required package: Matrix

## Loading required package: mvtnorm

## Loading required package: numDeriv

## Loading required package: timeSeries

## Loading required package: timeDate

##
## Attaching package: 'QRM'

## The following object is masked from 'package:base':
##
##     lbeta
```

```
copgauss <- fit.gausscopula(std)
```

```
## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in nlminb(theta, negloglik, data = Udata, ...): NA/NaN function
## evaluation

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in nlminb(theta, negloglik, data = Udata, ...): NA/NaN function
## evaluation

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced
## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in nlminb(theta, negloglik, data = Udata, ...): NA/NaN function
## evaluation

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in nlminb(theta, negloglik, data = Udata, ...): NA/NaN function
## evaluation

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in nlminb(theta, negloglik, data = Udata, ...): NA/NaN function
## evaluation

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in nlminb(theta, negloglik, data = Udata, ...): NA/NaN function
## evaluation

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in nlminb(theta, negloglik, data = Udata, ...): NA/NaN function
## evaluation

## Warning in cov2cor(Q): diag(.) had 0 or NA entries; non-finite result is
## doubtful

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced
## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in nlminb(theta, negloglik, data = Udata, ...): NA/NaN function
## evaluation

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in FUN(newX[, i], ...): NaNs produced

## Warning in nlminb(theta, negloglik, data = Udata, ...): NA/NaN function
## evaluation
```

```
copgauss$P  #see this P
```

```
##           [,1]      [,2]      [,3]      [,4]
## [1,] 1.0000000 0.2229531 0.3617210 0.2359355
## [2,] 0.2229531 1.0000000 0.2903952 0.3205309
## [3,] 0.3617210 0.2903952 1.0000000 0.2739437
## [4,] 0.2359355 0.3205309 0.2739437 1.0000000
```

(iv) Monte Carlo Stimulation for Gauss Copula

```
N <- 100000
data <- rcopula.gauss(N,rho)
write.csv(data, "/Users/pan/Documents/Imperial/Autumn Term/Quantitative Risk Management (MATH97108)/CW2/
```