

Parametric and Monte Carlo Value-at-Risk for an equally weighted index portfolio

Contents

| | |
|--|-----------|
| Parametric approach | 1 |
| Advantages and Disadvantages | 3 |
| Monte Carlo Simulation | 3 |
| Advantages and Disadvantages | 17 |
| Conclusion | 18 |

Parametric approach

```
# Load all the necessary libraries

library("VineCopula")
library("goftest")
library("KSCorrect")
library("fGarch")
library("ADGofTest")
library("tseries")
library("timeSeries")
library("zoo")
```

I start by importing the weekly prices for the Nikkei 225 and CAC 40 stock indices.

```
# Get weekly quotes for the Nikkei 225 index and the CAC 40 stock index

price <- get.hist.quote(instrument = "^n225", start = "2000-01-01",
                        end = "2018-09-21", quote="AdjClose",
                        compression = "w")
price2 <- get.hist.quote(instrument = "^fchi", start = "2000-01-01",
                        end = "2018-09-21", quote="AdjClose",
                        compression = "w")

# Convert the prices into log returns

y <- diff(log(price))
x <- diff(log(price2))
```

```
# Isolate the data from the dates
```

```
y <- coredata(y)
x <- coredata(x)
```

Now I move onto the implementation of the parametric approach. Assume that each of the weekly log returns, independently and identically (i.i.d) follow a normal distribution, in relation to their corresponding stock index. Meaning, the Nikkei 225 weekly log returns follow one normal distribution i.i.d and the CAC 40 weekly log returns follow another normal distribution i.i.d. We can estimate the two pairs of the mean and variance parameters of the normal distributions through the sample means and sample variances, as these estimators are consistent and the sample at hand is rather large. To calculate the expected return of an equally weighted portfolio containing these two assets, we assign a weight of 0.5 to each asset and sum the multiples of the weights and the estimated mean parameters, giving us the portfolio weekly expected return. The portfolio variance can be calculated by multiplying the covariance matrix obtained from the two samples by the weight vectors in the way seen below in the code, which in this case, as the portfolio is equally weighted, is just the sum of all of the elements in the covariance matrix divided by 4.

```
# Calculate mean and weight vectors
```

```
mean_vector <- c(mean(y), mean(x))
weight_vector <- c(0.5, 0.5)
```

```
log_return_df <- data.frame(y,x)
cov_matrix <- cov(log_return_df) # Covariance matrix retrieved from the log returns data
```

```
# Calculate the portfolio expected return and variance
```

```
combined_var <- t(weight_vector) %*% cov_matrix %*% weight_vector
combined_mean <- t(weight_vector) %*% mean_vector
```

```
# Table containing sample means and standard deviations of index weekly log returns
```

```
mean_sd_table <- data.frame(c("Mean", "Standard Deviation"),
                             c(mean(y), sd(y)), c(mean(x), sd(x)))
knitr::kable(mean_sd_table, "pipe", col.name = c("", "N225", "FCHI"), align = c("c", "c"),
              caption = '\\label{tab:tab2}Sample means and standard deviations
of weekly log returns (used in later discussion)')
```

Table 1: Sample means and standard deviations of weekly log returns (used in later discussion)

| | N225 | FCHI |
|--------------------|-----------|------------|
| Mean | 0.0002782 | -0.0000084 |
| Standard Deviation | 0.0302379 | 0.0295819 |

Finally, I find the 99% and 95% Value-at-Risk, which is calculated as the product of the quantile of the normal distribution at the corresponding confidence level with the portfolio standard deviation, and subtracted by the portfolio expected return, as done in the code below.

```
# VAR 99%

var_par_99 <- qnorm(0.99)*sqrt(combined_var) - combined_mean

# VAR 95%

var_par_95 <- qnorm(0.95)*sqrt(combined_var) - combined_mean
```

Below is the table containing the 95% and 99% Value-at-Risk, reported as a positive number.

```
# Create the table with the results

var_par <- data.frame(c("95%", "99%"), c(var_par_95, var_par_99))
knitr::kable(var_par, "pipe", col.name = c("VAR", "log return"), align = c("c", "c"),
  caption = "\\label{tab:tab3}VAR from the parametric approach")
```

Table 2: VAR from the parametric approach

| VAR | log return |
|-----|------------|
| 95% | 0.0438963 |
| 99% | 0.0621392 |

Since small log return values are approximately equal to the net returns, we can interpret the VARs above as being 95% confident that our losses from investing in the aforementioned portfolio won't exceed 4.39% over the next week and being 99% confident that our losses from investing in the aforementioned portfolio won't exceed 6.21% over the next week.

Before moving on, it is worth mentioning the advantages and disadvantages of this method.

Advantages and Disadvantages

An advantage of the parametric approach is its easy implementation. Only two parameters need to be estimated - mean and standard deviation - which can be consistently estimated from the obtained large sample of data. This can then easily be extended into the portfolio variance and expected return, as seen from the basic lines of code above.

A disadvantage of this approach is the i.i.d normality assumption for the distribution of log returns. This assumption is dangerous as it assumes weekly log returns of the same index occur independently of the previous ones, which isn't the case as later exposed through volatility clustering in the time series plots (Figures 2 & 6). Furthermore, clearly a more appropriate distribution of the weekly log returns has fatter tails than a normal distribution, as in the same plots the log returns regularly exceed multiple standard deviations from the means (relevant values in Table 1). Finally, the use of the linear correlation coefficient to describe dependency structure of two sets of weekly log returns may result in unreliable and underestimated Value-at-Risk estimates, as joint normality is unlikely since log returns of two major stock indices likely possess some degree of tail dependence, which is supported by similar time series plot shapes.

Monte Carlo Simulation

I begin by checking whether the samples are normally distributed using the Jarque-Bera test, which tests the null hypothesis of normality.

[Note: I set-up a timer in the code below, using the result in the end discussion]

```
# Set-up a timer to see how long it takes to run - used in end discussion

start <- Sys.time()

# Create a table with the p-values from the Jarque-Bera test to check for
# normality of weekly log returns

log_return <- c("Nikkei 225", "CAC 40")
test_results <-c(jarque.bera.test(y)$p.value, jarque.bera.test(x)$p.value)
table_1 <- data.frame(log_return, test_results)
knitr::kable(table_1, "pipe", col.name = c("Index", "p-value"), align = c("c", "c"),
              caption = "\\label{tab:tab4}Jarque-Bera test p-values for the weekly
log returns")
```

Table 3: Jarque-Bera test p-values for the weekly log returns

| Index | p-value |
|------------|---------|
| Nikkei 225 | 0 |
| CAC 40 | 0 |

As the p-values received above are extremely small (essentially 0), there's evidence to reject the null hypothesis, implying there is evidence that the samples are not normally distributed. This naturally warrants further investigation into the structure of log returns, where I will employ the Box-Jenkins approach to decide on the best model that describes the log returns' evolution over time.

I will identify the models for each return separately, beginning with the Nikkei 225 weekly log returns. Below I have constructed the ACF and PACF plots from the Nikkei 225 weekly log returns.

```
# ACF and PACF plots from the Nikkei 225 weekly log returns

par(mfrow=c(1,2))
acf(y, col="green", lwd=2, main = "N225 Weekly Log Returns ACF Plot")
pacf(y, col="green", lwd=2, main = "N225 Weekly Log Returns PACF Plot")
```

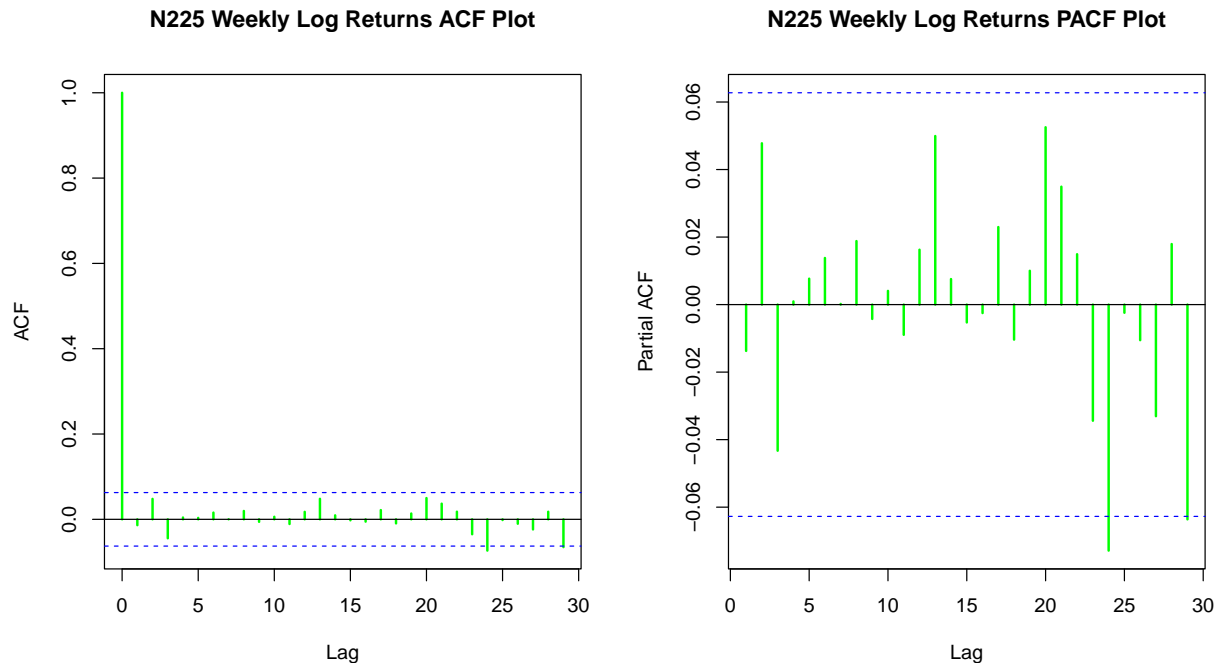


Figure 1: ACF and PACF plots from the N225 weekly log returns

Before I begin my analysis of any ACF/PACF plots it is worth mentioning that I will ignore lags of above 10, as this is the log return corresponding to 2.5 months ago and it would be sensible to assume that what happened at that time would mostly have an extremely marginal effect on the log return being inspected. What is more, I would potentially be excluding a few too many observations from the analysis, further supporting the ignorance of past 10 lags. Now, from the ACF plot above, it appears that there is no significant autocorrelation before the 10th lag, suggesting that the use of an auto-regressive (AR) model is not necessary. This is also supported by the PACF plot above which normally exposes the order of an AR model, however evidently in this case, tells us that no lags before 10 are statistically significant, while itself not exhibiting geometric decay characteristic of an MA model. Therefore, I am inclined to exclude an ARMA model from the structure of Nikkei 225 weekly log returns, as that is what the evidence points to.

I now move on to checking whether a GARCH model would be necessary. I plot the corresponding weekly log returns against time below, through which we can clearly observe volatility clustering - large changes in log returns are followed by large changes while small changes are followed by small ones. This indicates that variance is not constant and is likely auto-correlated, however I still construct an ACF plot of squared log returns below to confirm the suspicions.

```
# Construct N225 time series plot and ACF plot for squared weekly log returns of N225

par(mfrow=c(1,2))
plot(y, type = "l", main = "N225 Time Series", ylab = "N225 weekly log returns")
acf(y^2, col="red", lwd=2, main = "N225 Squared weekly Log Returns ACF Plot")
```

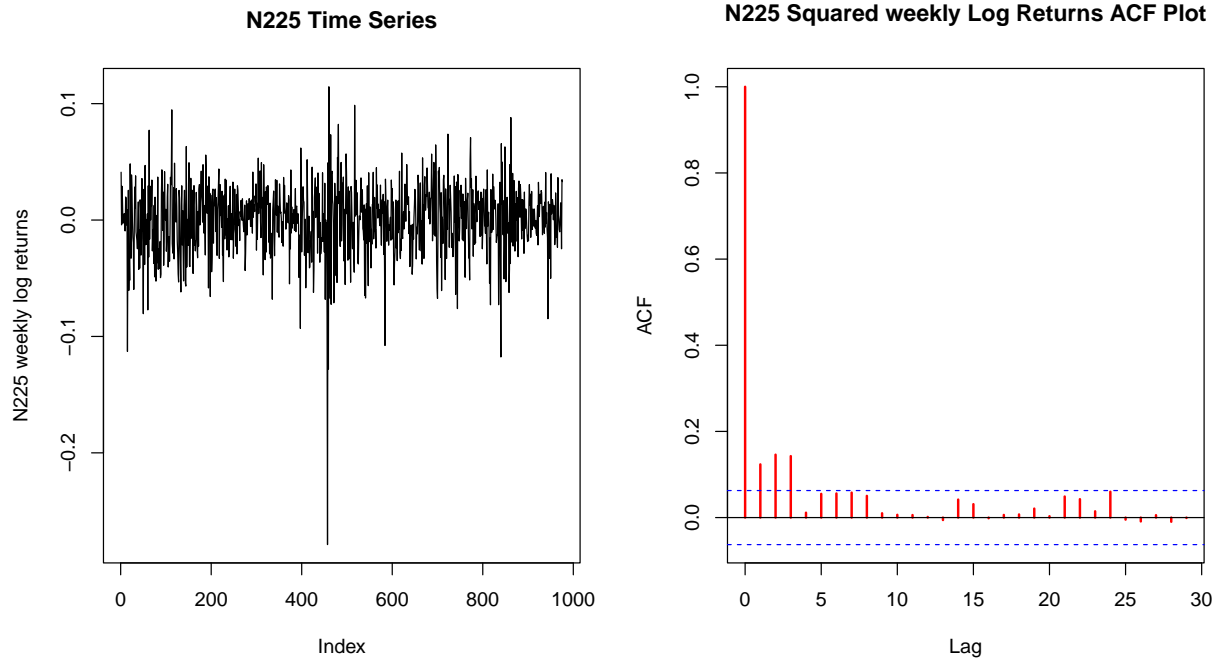


Figure 2: N225 weekly log return time series plot and ACF plot for N225 squared weekly log returns

As expected, volatility is auto-correlated (indicated by the significant correlations above).

Since it has been deduced a GARCH model is needed to most accurately describe the evolution of Nikkei 225 weekly log returns, I test the various orders of GARCH(p,q) models, up to GARCH(3,3), against different conditional distributions. Although in practice the GARCH(1,1) model is much more common than any other GARCH model, I still decided to test up to GARCH(3,3), choosing to stop there arbitrarily as, on top of there being an extremely large number of possible combinations, choosing a GARCH model of higher order would involve estimating too many parameters, and even if it results in a slightly better fit, perhaps the simpler but slightly worse model would still be preferred in practice, via the principle of parsimony.

Below is the table of the AIC/BIC criteria for each tested model.

Table 4: AIC/BIC values for different GARCH models and conditional distributions

| GARCH | Conditional distribution | AIC | BIC |
|------------|--------------------------|-----------|-----------|
| GARCH(1,1) | Normal (norm) | -4.253660 | -4.233645 |
| GARCH(1,2) | Normal (norm) | -4.261155 | -4.236137 |
| GARCH(1,3) | Normal (norm) | -4.259084 | -4.229063 |
| GARCH(2,1) | Normal (norm) | -4.251902 | -4.226884 |
| GARCH(2,2) | Normal (norm) | -4.259106 | -4.229085 |
| GARCH(2,3) | Normal (norm) | -4.257035 | -4.222010 |
| GARCH(3,1) | Normal (norm) | -4.250775 | -4.220753 |
| GARCH(3,2) | Normal (norm) | -4.256829 | -4.221804 |
| GARCH(3,3) | Normal (norm) | -4.254985 | -4.214957 |
| GARCH(1,1) | Skew Normal (snorm) | -4.295029 | -4.270011 |
| GARCH(1,2) | Skew Normal (snorm) | -4.301048 | -4.271027 |
| GARCH(1,3) | Skew Normal (snorm) | -4.299137 | -4.264112 |
| GARCH(2,1) | Skew Normal (snorm) | -4.293269 | -4.263248 |

| GARCH | Conditional distribution | AIC | BIC |
|------------|-------------------------------|-----------|-----------|
| GARCH(2,2) | Skew Normal (snorm) | -4.298999 | -4.263974 |
| GARCH(2,3) | Skew Normal (snorm) | -4.297087 | -4.257059 |
| GARCH(3,1) | Skew Normal (snorm) | -4.291218 | -4.256193 |
| GARCH(3,2) | Skew Normal (snorm) | -4.296689 | -4.256660 |
| GARCH(3,3) | Skew Normal (snorm) | -4.295038 | -4.250006 |
| GARCH(1,1) | Generalised Errorr (ged) | -4.279041 | -4.254023 |
| GARCH(1,2) | Generalised Errorr (ged) | -4.283421 | -4.253400 |
| GARCH(1,3) | Generalised Errorr (ged) | -4.281404 | -4.246379 |
| GARCH(2,1) | Generalised Errorr (ged) | -4.277220 | -4.247198 |
| GARCH(2,2) | Generalised Errorr (ged) | -4.281372 | -4.246347 |
| GARCH(2,3) | Generalised Errorr (ged) | -4.279355 | -4.239327 |
| GARCH(3,1) | Generalised Errorr (ged) | -4.274888 | -4.239864 |
| GARCH(3,2) | Generalised Errorr (ged) | -4.279236 | -4.239208 |
| GARCH(3,3) | Generalised Errorr (ged) | -4.277306 | -4.232274 |
| GARCH(1,1) | Skew Generalised Error (sged) | -4.310302 | -4.280280 |
| GARCH(1,2) | Skew Generalised Error (sged) | -4.314914 | -4.279889 |
| GARCH(1,3) | Skew Generalised Error (sged) | -4.312915 | -4.272886 |
| GARCH(2,1) | Skew Generalised Error (sged) | -4.308510 | -4.273485 |
| GARCH(2,2) | Skew Generalised Error (sged) | -4.312865 | -4.272836 |
| GARCH(2,3) | Skew Generalised Error (sged) | -4.310866 | -4.265834 |
| GARCH(3,1) | Skew Generalised Error (sged) | -4.306192 | -4.266164 |
| GARCH(3,2) | Skew Generalised Error (sged) | -4.310702 | -4.265670 |
| GARCH(3,3) | Skew Generalised Error (sged) | -4.308816 | -4.258781 |
| GARCH(1,1) | Student-t (std) | -4.304864 | -4.279847 |
| GARCH(1,2) | Student-t (std) | -4.306336 | -4.276315 |
| GARCH(1,3) | Student-t (std) | -4.304148 | -4.269123 |
| GARCH(2,1) | Student-t (std) | -4.302981 | -4.272960 |
| GARCH(2,2) | Student-t (std) | -4.304287 | -4.269262 |
| GARCH(2,3) | Student-t (std) | -4.302123 | -4.262094 |
| GARCH(3,1) | Student-t (std) | -4.300731 | -4.265706 |
| GARCH(3,2) | Student-t (std) | -4.302074 | -4.262046 |
| GARCH(3,3) | Student-t (std) | -4.300073 | -4.255042 |
| GARCH(1,1) | Skew Student-t (sstd) | -4.323743 | -4.293722 |
| GARCH(1,2) | Skew Student-t (sstd) | -4.327195 | -4.292170 |
| GARCH(1,3) | Skew Student-t (sstd) | -4.325073 | -4.285044 |
| GARCH(2,1) | Skew Student-t (sstd) | -4.321909 | -4.286884 |
| GARCH(2,2) | Skew Student-t (sstd) | -4.325146 | -4.285117 |
| GARCH(2,3) | Skew Student-t (sstd) | -4.323023 | -4.277992 |
| GARCH(3,1) | Skew Student-t (sstd) | -4.319605 | -4.279576 |
| GARCH(3,2) | Skew Student-t (sstd) | -4.322929 | -4.277897 |
| GARCH(3,3) | Skew Student-t (sstd) | -4.320974 | -4.270939 |

[**Note:** I have excluded the ‘snig’ conditional distributions as it cannot be fit to the data. I have also excluded the code from the report as it was too repetitive and long.]

From the above tables we see that the model with the smallest BIC value is a GARCH(1,1) model with a Skew Student-t conditional distribution. Although possessing the lowest BIC values, the GARCH(1,2) model with a Skew Student-t conditional distribution possesses a smaller AIC value. In this instance that the BIC and AIC values do not agree, I lean towards the model with the smaller BIC value, as the model with the smallest AIC value tends to overfit the data, while the BIC penalises higher parameter models if they do not improve the model fit substantially. Moreover, the GARCH(1,1) model is also overwhelmingly used in practice, is a simpler model than its previously mentioned counterpart and ultimately the difference in the

AIC value between it and the GARCH(1,2) model is relatively negligible (although the BIC difference is also marginal). Therefore I choose the GARCH(1,1) model and continue with the tests to check the model's adequacy using residual diagnostics.

I begin by calculating the standardised residuals from the aforementioned model and constructing two ACF plots, one of the standardised residuals and the other of the squared standardised residuals.

```
# Obtain the standardised residuals and construct corresponding ACF plots

res1 <- residuals(model11_sstd, standardize=TRUE)
par(mfrow=c(1,2))
acf(res1, col="green", lwd=2, main = "N225 Standardised Residuals ACF Plot")
acf(res1^2, col="red", lwd=2, main = "N225 Squared Standardised Residuals ACF Plot")
```

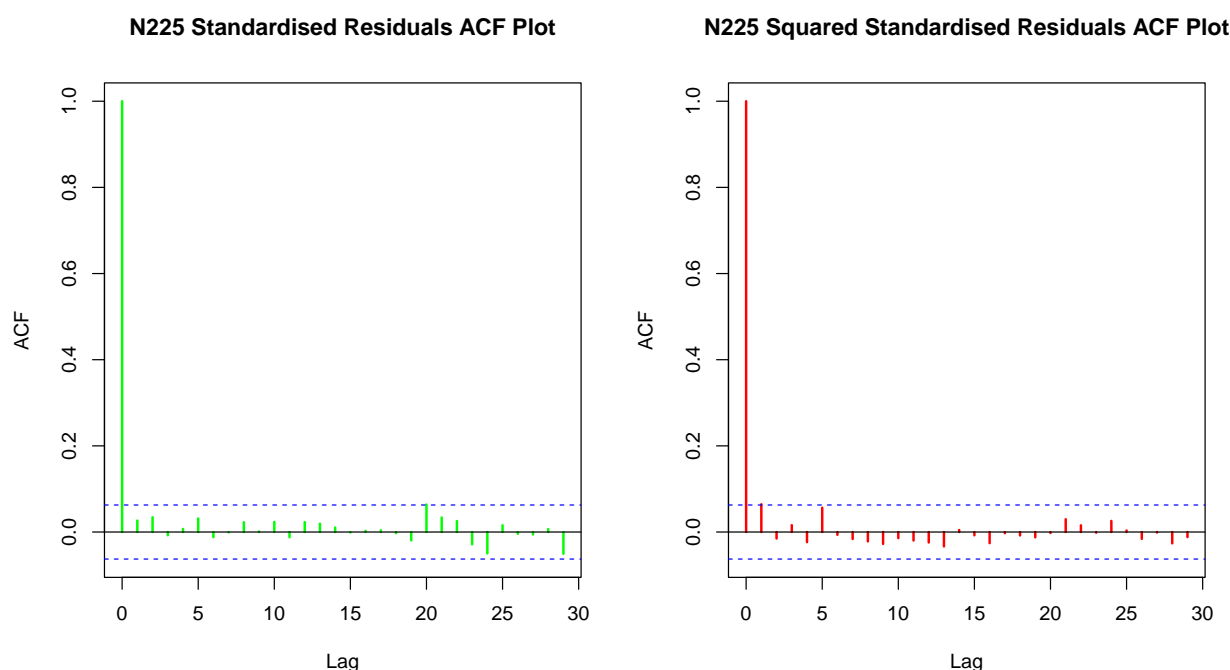


Figure 3: N225 ACF plots for standardised residuals and squared standardised residuals

It appears that the model provides a relatively good explanation for the data collected, as none of the lags are statistically significant for the residual ACF plot and most are not significant for the squared residual ACF plot, with all the lags drastically decreasing in significance from before fitting the model. I now proceed with a formal Ljung-Box test, which tests the null that autocorrelation for the first specified number of lags is zero. There are a lot of different opinions about the optimal number of lags that should be tested in the Ljung-Box test, however in this case, a sensible number to choose would be 10, as it is consistent with the previous decision to ignore statistically significant auto-correlation past the 10th lag. The degrees of freedom chosen is the order of the fitted AR model (in this case 0), as this function subtracts “fitdf” from the number of components.

```
# Apply the Ljung-Box tests on the corresponding values

box_test_1 <- Box.test(res1, lag = 10, type = c("Ljung-Box"), fitdf = 0)
box_test_2 <- Box.test(res1^2, lag = 10, type = c("Ljung-Box"), fitdf = 0)
```



```

# Construct a table containing the p-values from the Box tests

res_table <- data.frame(c("Residuals", "Squared Residuals"),
                        c(box_test_1$p.value, box_test_2$p.value))
knitr::kable(res_table, "pipe", col.name = c("Ljung-Box test", "p-value"),
              align = c("c", "c"), caption = "\\label{tab:tab6}Box test
              p-values for N225")

```

Table 5: Box test p-values for N225

| Ljung-Box test | p-value |
|-------------------|-----------|
| Residuals | 0.9414364 |
| Squared Residuals | 0.4452290 |

All the p-values above are rather large and pass the tests at the 5% significance level, possessing values of 0.94 and 0.45 for the standardised residuals and squared standardised residuals, respectively.

Finally, I use the estimates of the degrees of freedom and scale parameter of the Skew Student-t distribution obtained from the ‘garchFit’ function, which were estimated using a max log-likelihood approach, to describe the marginal distribution of the N225 standardised residuals. These estimates are used when transforming the standardised residuals with the corresponding Skew Student-t c.d.f (i.e apply the Probability Integral Transform to the N225 standardised residuals). As the residuals are standardised, I set the mean to 0 and standard deviation to 1. I then construct a histogram of the transformed residuals, noticing that it supports our choice of the marginal distribution for the corresponding residuals, since the histogram looks like one you could obtain from a sample following a standard uniform distribution. This is confirmed through the Kolmogorov-Smirnov test and the Anderson-Darling test, which exhibit relatively large p-values at 5%. These tests essentially test whether the transformed residuals come from a standard uniform distribution, and hence confirm that there is not enough evidence to reject the marginal distribution obtained.

```

# PIT with the Skew Student-t distribution cdf obtained

u_1 <- psstd(res1, mean = 0,
             sd = 1, nu = model11_sstd@fit$par[["shape"]],
             xi = model11_sstd@fit$par[["skew"]])[2:length(y)]

# Construct the histogram of the transformed residuals

hist(u_1, main = "N225 (PIT)", xlab = "")

```

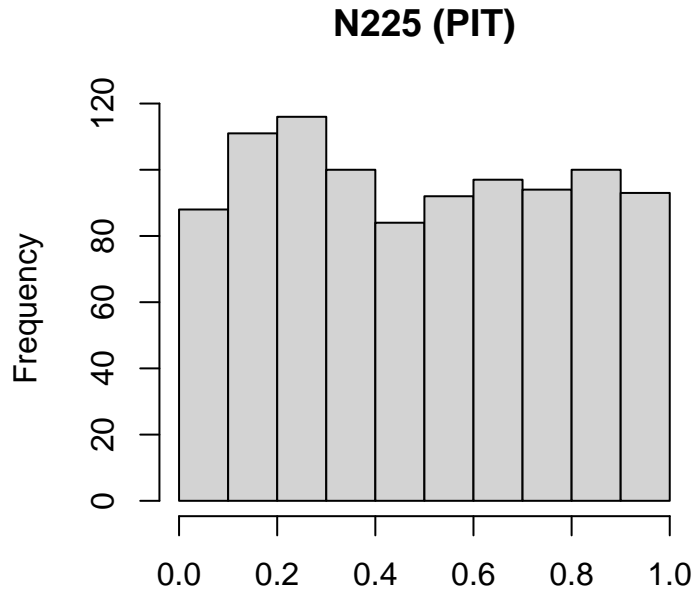


Figure 4: Histogram of transformed N225 standardised residuals

```
# Perform corresponding gof tests

KStest1 <- LcKS(u_1, cdf = "punif")
ADtest1 <- ad.test(u_1, null="punif")

# Form table with p-values from the above tests

gof_table <- data.frame(c("Kolmogorov-Smirnov", "Anderson-Darling "),
                        c(KStest1$p.value, as.vector(ADtest1$p.value)))
knitr::kable(gof_table, "pipe", col.name = c("Test", "p-value"), align = c("c", "c"),
             caption = "\\label{tab:tab7}P-values for tests testing standard uniformity")
```

Table 6: P-values for tests testing standard uniformity

| Test | p-value |
|--------------------|-----------|
| Kolmogorov-Smirnov | 0.2230000 |
| Anderson-Darling | 0.4564144 |

Moving on to the CAC 40 weekly log returns, I again construct ACF and PACF plots for the corresponding returns.

```
# ACF and PACF plots from the CAC 40 weekly log returns

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

```
acf(x, col="green", lwd=2, main = "FCHI Weekly Log Returns ACF Plot")
pacf(x, col="green", lwd=2, main = "FCHI Weekly Log Returns PACF Plot")
```

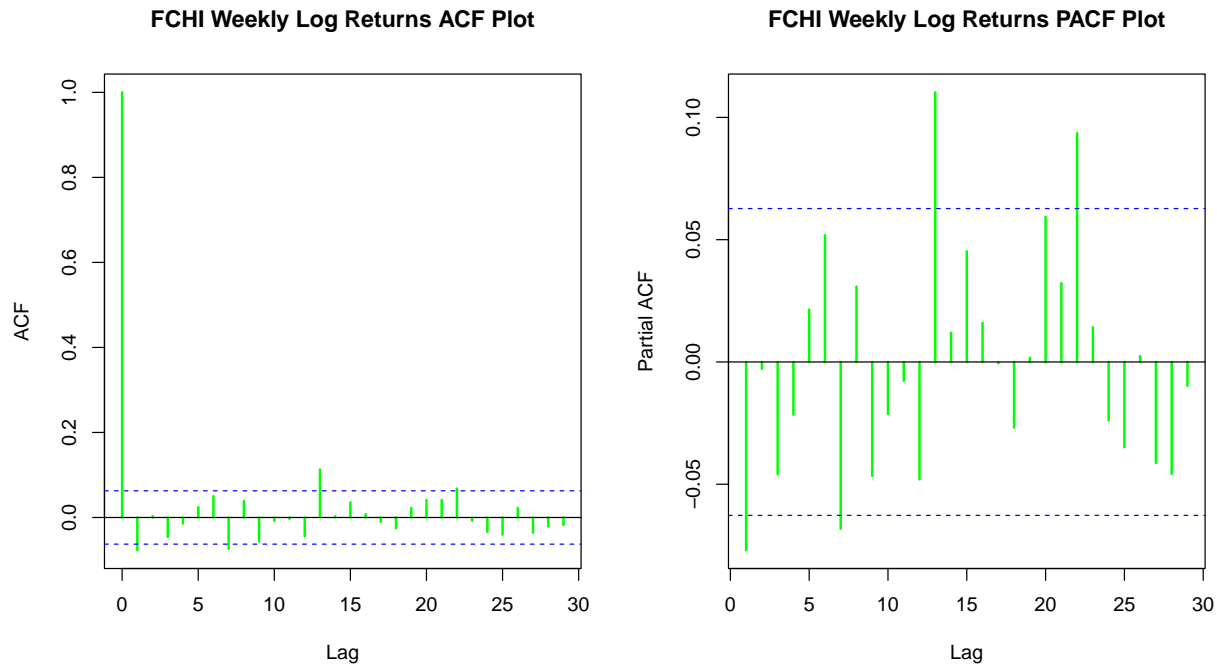


Figure 5: ACF and PACF plots from the FCHI weekly log returns

From the ACF plot above, we observe some statistically significant auto-correlation before lag 10 and maybe some geometric decay resemblance, suggesting that now in this case, an AR model could be used. Observing the PACF plot, it appears that there are 2 statistically significant lags before 10 - 1 and 7. It seems reasonable to assume that an AR(1) model should be preferred to describe the structure of CAC 40 weekly log returns, as it is the more statistically significant lag. Therefore, I will proceed to the Box-Jenkins approach with an AR(1) model. One could also continue as I do with an AR(7) model as well and then compare the AIC/BIC values collectively, but I choose not to here as the results will be the same anyways.

Similarly to the previous log returns, there's reason to believe that volatility is yet again auto-correlated, as exposed by the clear volatility clustering below from the plot of CAC 40 log returns against time. However, I still look at the ACF plot of squared log returns and formally deduce that volatility is auto-correlated through the statistically significant lags exposed below.

```
# Construct FCHI time series plot and ACF plot for squared weekly log returns of FCHI

par(mfrow=c(1,2))
plot(x, type = "l", main = 'FCHI Time Series', ylab = 'FCHI weekly log returns')
acf(x^2, col="red", lwd=2, main = "FCHI Squared Weekly Log Returns ACF Plot")
```

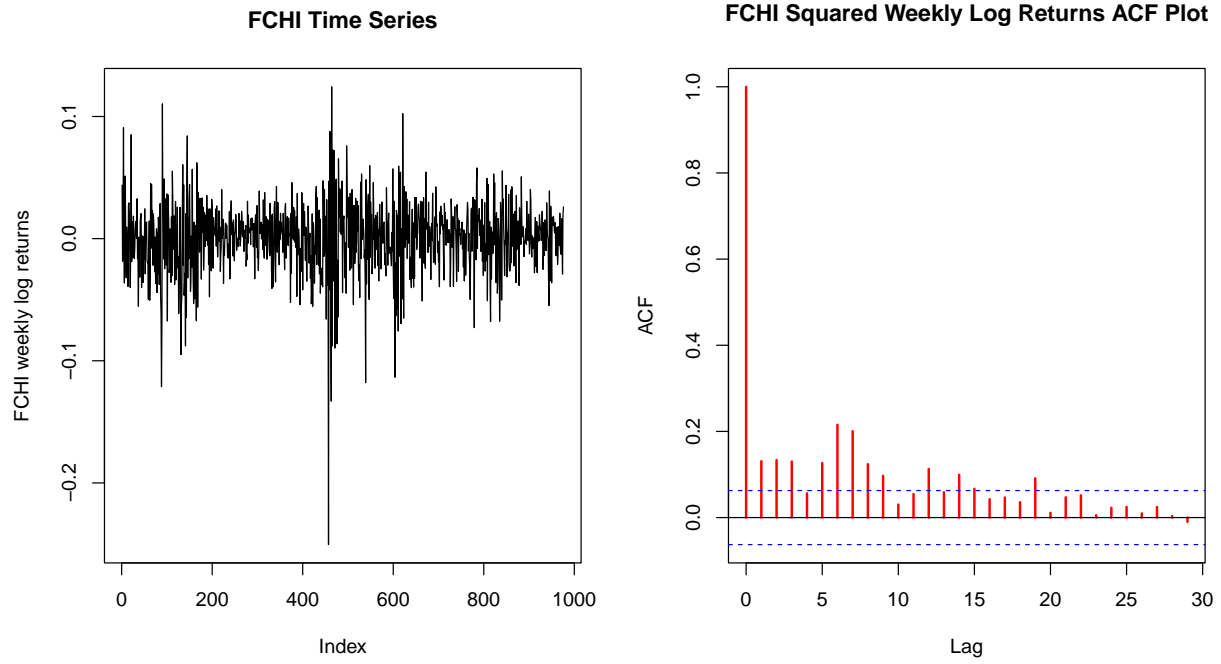


Figure 6: FCHI weekly log return time series plot and ACF plot for FCHI squared weekly log returns

I again test all GARCH models up to GARCH(3,3), for the same reasons. Below is the table of the AIC and BIC values for the corresponding GARCH models with an AR(1) model.

Table 7: AIC/BIC values for an AR(1) model combined with different GARCH models and conditional distributions

| GARCH | Conditional distribution | AIC | BIC |
|------------|--------------------------|-----------|-----------|
| GARCH(1,1) | Normal (norm) | -4.408811 | -4.383793 |
| GARCH(1,2) | Normal (norm) | -4.407679 | -4.377658 |
| GARCH(1,3) | Normal (norm) | -4.406438 | -4.371414 |
| GARCH(2,1) | Normal (norm) | -4.407684 | -4.377662 |
| GARCH(2,2) | Normal (norm) | -4.405635 | -4.370610 |
| GARCH(2,3) | Normal (norm) | -4.412513 | -4.372484 |
| GARCH(3,1) | Normal (norm) | -4.406273 | -4.371248 |
| GARCH(3,2) | Normal (norm) | -4.404224 | -4.364196 |
| GARCH(3,3) | Normal (norm) | -4.410464 | -4.365432 |
| GARCH(1,1) | Skew Normal (snorm) | -4.454377 | -4.424356 |
| GARCH(1,2) | Skew Normal (snorm) | -4.454118 | -4.419093 |
| GARCH(1,3) | Skew Normal (snorm) | -4.453869 | -4.413841 |
| GARCH(2,1) | Skew Normal (snorm) | -4.454118 | -4.419093 |
| GARCH(2,2) | Skew Normal (snorm) | -4.456207 | -4.416179 |
| GARCH(2,3) | Skew Normal (snorm) | -4.456772 | -4.411740 |
| GARCH(3,1) | Skew Normal (snorm) | -4.453481 | -4.413453 |
| GARCH(3,2) | Skew Normal (snorm) | -4.451432 | -4.406400 |
| GARCH(3,3) | Skew Normal (snorm) | -4.454723 | -4.404688 |
| GARCH(1,1) | Generalised Errorr (ged) | -4.438471 | -4.408450 |
| GARCH(1,2) | Generalised Errorr (ged) | -4.436901 | -4.401876 |
| GARCH(1,3) | Generalised Errorr (ged) | -4.435296 | -4.395267 |

| GARCH | Conditional distribution | AIC | BIC |
|------------|-------------------------------|-----------|-----------|
| GARCH(2,1) | Generalised Errorr (ged) | -4.436901 | -4.401876 |
| GARCH(2,2) | Generalised Errorr (ged) | -4.438350 | -4.398321 |
| GARCH(2,3) | Generalised Errorr (ged) | -4.436240 | -4.391208 |
| GARCH(3,1) | Generalised Errorr (ged) | -4.435201 | -4.395172 |
| GARCH(3,2) | Generalised Errorr (ged) | -4.433152 | -4.388120 |
| GARCH(3,3) | Generalised Errorr (ged) | -4.434245 | -4.384210 |
| GARCH(1,1) | Skew Generalised Error (sged) | -4.483146 | -4.448122 |
| GARCH(1,2) | Skew Generalised Error (sged) | -4.481969 | -4.441941 |
| GARCH(1,3) | Skew Generalised Error (sged) | -4.480835 | -4.435803 |
| GARCH(2,1) | Skew Generalised Error (sged) | -4.481969 | -4.441941 |
| GARCH(2,2) | Skew Generalised Error (sged) | -4.481368 | -4.436336 |
| GARCH(2,3) | Skew Generalised Error (sged) | -4.481093 | -4.431057 |
| GARCH(3,1) | Skew Generalised Error (sged) | -4.480659 | -4.435627 |
| GARCH(3,2) | Skew Generalised Error (sged) | -4.479035 | -4.429000 |
| GARCH(3,3) | Skew Generalised Error (sged) | -4.479126 | -4.424087 |
| GARCH(1,1) | Student-t (std) | -4.461573 | -4.431552 |
| GARCH(1,2) | Student-t (std) | -4.459824 | -4.424799 |
| GARCH(1,3) | Student-t (std) | -4.458143 | -4.418115 |
| GARCH(2,1) | Student-t (std) | -4.459824 | -4.424799 |
| GARCH(2,2) | Student-t (std) | -4.459289 | -4.419261 |
| GARCH(2,3) | Student-t (std) | -4.456806 | -4.411774 |
| GARCH(3,1) | Student-t (std) | -4.458028 | -4.417999 |
| GARCH(3,2) | Student-t (std) | -4.455979 | -4.410947 |
| GARCH(3,3) | Student-t (std) | -4.454822 | -4.404787 |
| GARCH(1,1) | Skew Student-t (sstd) | -4.494588 | -4.459564 |
| GARCH(1,2) | Skew Student-t (sstd) | -4.493006 | -4.452978 |
| GARCH(1,3) | Skew Student-t (sstd) | -4.491620 | -4.446588 |
| GARCH(2,1) | Skew Student-t (sstd) | -4.493004 | -4.452975 |
| GARCH(2,2) | Skew Student-t (sstd) | -4.491648 | -4.446616 |
| GARCH(2,3) | Skew Student-t (sstd) | -4.490541 | -4.440506 |
| GARCH(3,1) | Skew Student-t (sstd) | -4.491384 | -4.446352 |
| GARCH(3,2) | Skew Student-t (sstd) | -4.489335 | -4.439299 |
| GARCH(3,3) | Skew Student-t (sstd) | -4.482968 | -4.427929 |

[**Note:** Just like previously I have excluded the ‘snig’ conditional distributions as it cannot be fit to the data. I have also again excluded the code from the report as it was too repetitive and long.]

Similarly to the Nikkei 225 weekly log returns, a GARCH(1,1) model with a Skew Student-t conditional distribution possesses the smallest BIC value, however this time, it also possesses the smallest AIC value from all the other above models. So I continue to the residual diagnostics with a combination of an AR(1) model and a GARCH(1,1) model, following a Skew Student-t conditional distribution.

Again, I construct two ACF plots, one with the standardised residuals and the other with the squared standardised residuals, both obtained from the corresponding aforementioned model.

Obtain the standardised residuals and construct corresponding ACF plots

```
res2 <- residuals(model111_sstd, standardize=TRUE)
par(mfrow=c(1,2))
acf(res2, col="green", lwd=2, main = "FCHI Standardised Residuals ACF Plot")
acf(res2^2, col="red", lwd=2, main = "FCHI Squared Standardised Residuals ACF Plot")
```

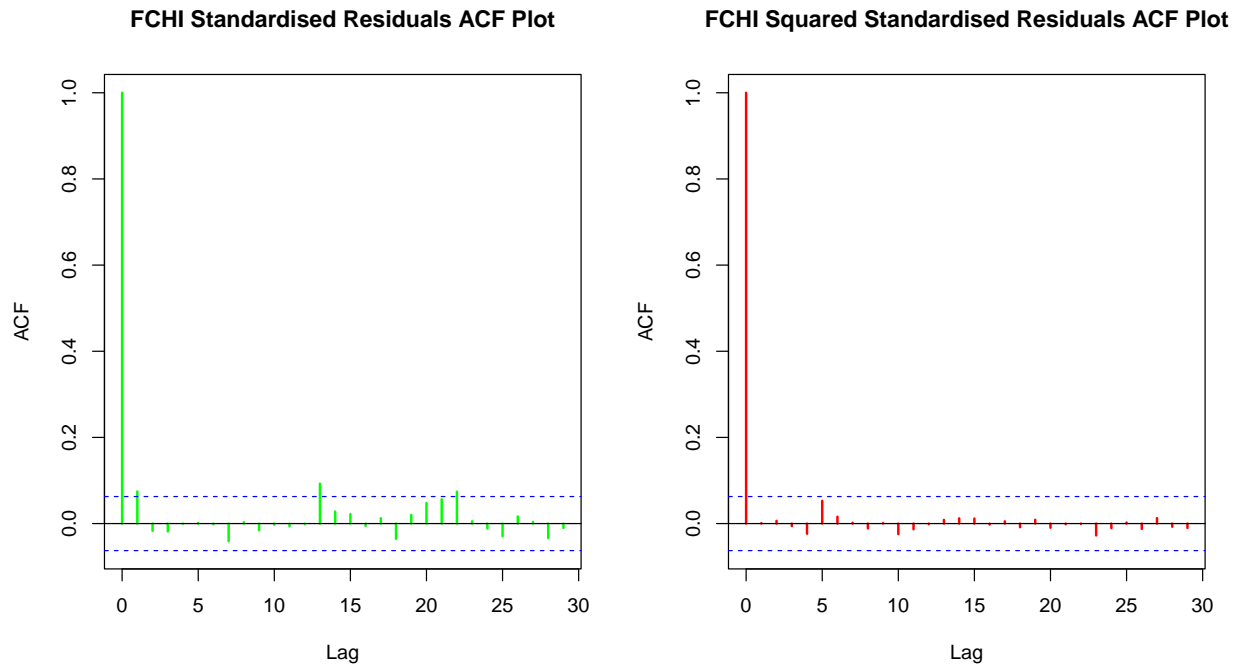


Figure 7: FCHI ACF plots for standardised residuals and squared standardised residuals

Although the ACF plot of the squared residuals looks as though the volatility is no longer auto-correlated, the ACF plot of the residuals appears slightly concerning, as it exposes a convincingly significant lag before the 10th one. I proceed with the formal Ljung-Box test, testing the same number of lags as before for the same reasons, however now using 1 degree of freedom, as I have fit an AR(1) model.

```
# Apply the Ljung-Box tests on the corresponding values

box_test_3 <- Box.test(res2, lag = 10, type = c("Ljung-Box"), fitdf = 1)
box_test_4 <- Box.test(res2^2, lag = 10, type = c("Ljung-Box"), fitdf = 1)

# Construct a table containing the p-values from the Box-tests

res_tabl2 <- data.frame(c("Residuals", "Squared Residuals"),
                        c(box_test_3$p.value, box_test_4$p.value))
knitr::kable(res_tabl2, "pipe", col.name = c("Ljung-Box test", "p-value"),
              align = c("c", "c"), caption = "\\label{tab:tab9}Box test
              p-values for FCHI")
```

Table 8: Box test p-values for FCHI

| Ljung-Box test | p-value |
|-------------------|-----------|
| Residuals | 0.5381180 |
| Squared Residuals | 0.8870647 |

The concerns are alleviated by the formal tests above, which show two large p-values (0.54, 0.89), essentially stating there is not enough evidence to reject the null of no autocorrelation between the first 10 lags.

Finally, I use the estimates of the degrees of freedom and the scale parameter of the Skew Student-t distribution, obtained from the 'garchFit' function again, and apply the Probability Integral Transform in the same way as previously, but now to the second set of residuals. I again construct a histogram and proceed with the same formal tests. The first observations in the Probability Integral Transforms are lost due to the AR(1) model.

```
# PIT with the Skew Student-t distribution cdf obtained

u_2 <- psstd(res2, mean = 0,
             sd = 1, nu = model111_sstd@fit$par[["shape"]],
             xi = model111_sstd@fit$par[["skew"]])[2:length(x)]

# Construct the histogram of the transformed residuals

hist(u_2, main = 'FCHI (PIT)', xlab = "")
```

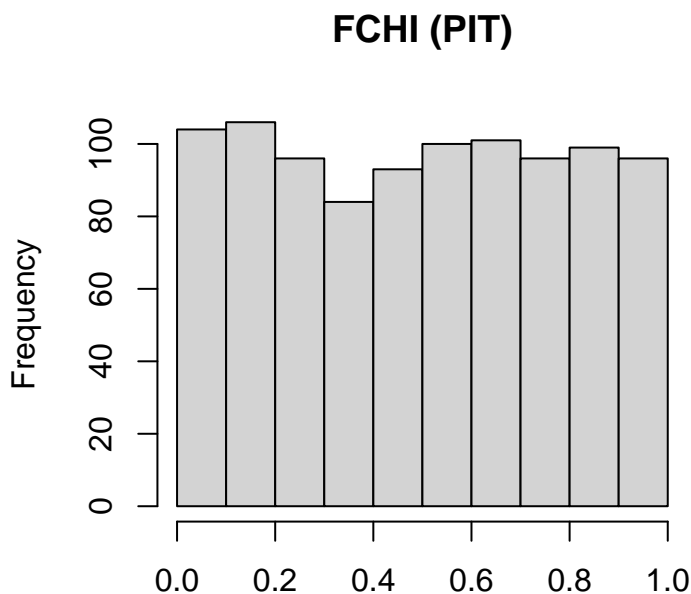


Figure 8: Histogram of transformed FCHI standardised residuals

```
# Perform corresponding gof tests

KStest2 <- LcKS(u_2, cdf = "punif")
ADtest2 <- ad.test(u_2, null="punif")

# Form table with p-values from the above tests

gof_table2 <- data.frame(c("Kolmogorov-Smirnov", "Anderson-Darling "),
                        c(KStest2$p.value, as.vector(ADtest2$p.value)))
```

```
knitr::kable(gof_table2, "pipe", col.name = c("Test", "p-value"), align = c("c", "c"),
  caption = "\\label{tab:tab10}P-values for tests testing standard uniformity")
```

Table 9: P-values for tests testing standard uniformity

| Test | p-value |
|--------------------|-----------|
| Kolmogorov-Smirnov | 0.8066000 |
| Anderson-Darling | 0.8757707 |

The histogram above looks very convincing and the standard uniformity of the transformed residuals is confirmed by the very large p-values of the corresponding tests. So the marginal distribution of the standardised FCHI residuals is adequate. Therefore, I have found relatively suitable models to describe the structure of weekly log returns for the indices chosen and so can proceed with the copula modelling.

We know from Sklar's theorem that there exists an appropriate copula that when applied to the two previously obtained residual Probability Integral Transforms, is equal to the joint distribution of the two residuals. Therefore, I find this most appropriate copula using the 'BiCopSelect' function, setting the selection criteria to BIC, to be consistent with previous choices.

```
# Implement the BiCopSelect function to find the appropriate copula

model <- BiCopSelect(u_1, u_2, familyset=NA, selectioncrit="BIC",
  indeptest=TRUE, level=0.05, se = TRUE)

model
```

```
## Bivariate copula: Survival Gumbel (par = 1.61, tau = 0.38)
```

I obtain the Survival Gumbel copula with the above parameter. Therefore, a Survival Gumbel copula with a parameter of 1.61 is the copula that best fits the data.

Now that I have everything I need to calculate the VAR using Monte Carlo simulation, I will create 10000 pairs of observations for the next time step using the dependency described by the obtained copula.

```
# Simulate 10000 pairs of observations with the appropriate parameters

N <- 10000
u_sim <- BiCopSim(N, family=model$family, model$par)
```

I now apply the inverse Probability Integral Transform to transform the values from a standard uniform back into the values following their corresponding marginal distributions.

```
# Apply the inverse probability integral transform

res1_sim <- qsst(u_sim[,1], mean = 0, sd = 1, nu = model11_sstd@fit$par[["shape"]],
  xi = model11_sstd@fit$par[["skew"]])
res2_sim <- qsst(u_sim[,2], mean = 0, sd = 1, nu = model111_sstd@fit$par[["shape"]],
  xi = model111_sstd@fit$par[["skew"]])
```

Having obtained the residuals, I reintroduce autocorrelation and GARCH effects observed in the data, doing so in the code below. I find the non-standardised residuals from the two fitted models, and then substitute the necessary values into the corresponding formulas.


```

# Find the non-standardised residuals

res1_non_standard <- residuals(model11_sstd, standardize=FALSE)
res2_non_standard <- residuals(model111_sstd, standardize=FALSE)

# Reintroduce auto-correlation and GARCH effects

sim_1 <- model11_sstd@fit$par[["mu"]] + sqrt(model11_sstd@fit$par[["omega"]])
+ model11_sstd@fit$par[["alpha1"]] * tail(res1_non_standard, 1) ^ 2
+ model11_sstd@fit$par[["beta1"]] * tail(model11_sstd@fit$series$h, 1)) * res1_sim

sim_2 <- model111_sstd@fit$par[["mu"]] + model111_sstd@fit$par[["ar1"]] *
as.vector(tail(x, 1)) + sqrt(model111_sstd@fit$par[["omega"]])
+ model111_sstd@fit$par[["alpha1"]] * tail(res2_non_standard, 1) ^ 2
+ model111_sstd@fit$par[["beta1"]] * tail(model111_sstd@fit$series$h, 1)) * res2_sim

```

Finally, I combine the two separate log returns into one log return and use that to calculate the 95% and 99% VAR, reported in the below table as positive values.

```

# Combine returns into log return

portsim <- log(1+((exp(sim_1)-1)+(exp(sim_2)-1))*(1/2))

# Calculate and form a table with the 95% and 99% VARs

varsim <- quantile(portsim,c(0.05,0.01))
var_mc <- data.frame(c("95%", "99%"), as.vector(-varsim))
knitr::kable(var_mc, "pipe", col.name = c("VAR", "log return"), align = c("c", "c"),
caption = "\\label{tab:tab11}VAR Monte Carlo simulation approach")

```

Table 10: VAR Monte Carlo simulation approach

| VAR | log return |
|-----|------------|
| 95% | 0.0357845 |
| 99% | 0.0601875 |

We know that small log return values are approximately equal to the net returns, therefore, we can interpret the Value-at-risks above as being 95% confident that losses from investing in the aforementioned portfolio won't exceed 3.58% over the next week and being 99% confident that losses from investing in the aforementioned portfolio won't exceed 6.02% over the next week.

```

# Time it took to run in seconds

time_to_run <- as.numeric(round(Sys.time() - start, 2), units = 'secs')

```

It is again worth discussing the advantages and disadvantages of this method.

Advantages and Disadvantages

An advantage of the Monte Carlo simulation approach is its flexibility. The log returns don't have to be normal, hence we can make far more marginal distribution assumptions by testing, in this case, 6 different

conditional distributions and potentially assume a marginal distribution with a more fitting skew and kurtosis for index log returns, as I did with the Skew Student-t. This allows for an appropriate dependency structure for the two weekly log returns, regardless of the separate data structures, hence capturing any tail dependence that is almost definitely present in two major stock indices (exposed through similarity in Figures **2** & **6**). As we can proceed even with the non-linear relationship of two stock indices and capture the dependencies within the corresponding index (as seen with the GARCH(1,1) and AR(1) models), we get the most reliable Value-at-Risk estimates of any approach.

A disadvantage of the Monte Carlo simulation approach is its computation time. It took 110.4 seconds to run, clearly indicating the issue with time. Moreover, the procedure is complicated and time-consuming, requiring to fit and compare 2 sets of 54 models, along with creating and storing the plethora of variables observed in the code above.

Conclusion

The 95% and 99% Value-at-Risk obtained from the parametric approach are 4.39% and 6.21%, respectively and from the Monte Carlo simulation approach - 3.58%, 6.02%. The difference between the 95% and 99% VARs for the Monte Carlo approach is greater than in the parametric approach. The extremely small p-values from the Jarque-Bera test from Table **3**, suggest there is enough evidence to reject the null of normality for the weekly log returns. This indicates the flawed nature of the normality assumption, suggesting the parametric approach is not a sensible choice in this context. In contrast, the Monte Carlo approach makes no such assumption, allowing for far more freedom in conditional distribution assumptions, capturing the appropriate joint dependencies of the log returns regardless of their varying structure. This approach however is more computationally intensive, but this isn't an issue in this case as I do not have any time constraints. Furthermore, simulating 10000 observations means we should largely expect to see similar estimates, meaning the corresponding Value-at-Risk estimates above are reliable. Therefore, I have little reason to choose the parametric approach and hence will measure the riskiness of this portfolio using the Monte Carlo simulation approach.