Table Of Contents

1. Introduction

The study of copulas and their applications is a relatively new concept in the field of statistics. The word "copula" is a Latin word that means "a link, tie, bond" (Nelson, 2006). So, in the statistical sense, what are copulas? A copula is a function that joins a multivariate distribution function to their one-dimensional marginal distribution functions (Nelson, 2006). Regression analysis is typically the most popular method one uses to explain the relationship amongst variables, however it does have its limitations. Regression mainly requires one variable to be the primary variable of interest; the dependent variable, while all other variables are used in a supporting role (Frees & Valdez, 1998). However, when a relationship for several outcomes is required, as in a multivariate distribution, things become complex and simple regression tends to be less effective. Copulas are a way of describing and understanding relationships amongst variables through their joint multivariate distribution.

The first mention of copulas in Mathematics & Statistics literature came from Abe Sklar in 1959, with his theorem, which is now named after him. It essentially states there are functions that join one-dimensional distributions to form multivariate distributions. A more in depth look at this theorem is found in the following section. Actuarial and finance science is the field in mathematics where copulas are the most popular. Their systems are generally complex with several outcomes (Frees & Valdez). Copulas have the ability to characterize the relationship between several outcomes using multivariate data, which is why they are useful in

these fields. A more in depth introduction to copulas, their functional form and their properties are found in the succeeding sections.

Often, one is interested in the intersection of two or more random variables at the same time when analyzing data. For example, how they relate to one another, and more specifically, how they depend on one another. It is these intersections that create a multivariate distribution. For the bivariate case for two continuous random variables, Y_1 and Y_2, the joint distribution function is defined by

$$F(y_1, y_2) = \int_{-\infty}^{y_1} \int_{-\infty}^{y_2} f(t_1, t_2)\, dt_2\, dt_1, \qquad -\infty \le y_k \le \infty$$

where the function $f(y_1, y_2)$ is the joint density function (Lebo, 2005). The joint density function has the following properties;

$$f(y_1, y_2) \ge 0, \quad \forall y_1, y_2$$

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(y_1, y_2) dy_1 dy_2 = 1.$$

These definitions can be easily extended to more than two dimensions. It is important to note that with these joint distribution functions comes marginal and conditional probabilities as well. It is in these marginal probabilities that one is able to independently define the copula using the marginal distributions of the data.

Obviously when dealing with these distributions, the dependence between variables is one of major concern and intrigue. Dependence explains the relationship between two or more variables. Each multivariate distribution comes with its own unique dependence structure. These varying structures can be difficult to model, but the goal of using copulas is to have the ability to model more complex

distributions and be able to properly explain the relationship among the variables. Measures of dependence are common tools to explain a complex dependence structure (Schmidt, 2006). There are several different ways to describe the dependencies, with the most prominent being the correlation. Some common ways to describe the correlation include the Pearson's correlation coefficient, Spearman's rho, and Kendall's tau. All give an estimate of the correlation between variables, however the Pearson's correlation coefficient is only useful in explaining linear relationships between variables. The other two are based on the ranks of the data. Another type of dependence is tail dependence. Tail dependence is more concerned with what is happening at the extreme values of the distribution (Schmidt, 2006). Different copulas have different dependent structures, and a lot happens in the tails of these distributions so they cannot be forgotten. If U_1 and U_2 are marginal distributions with copula C, the upper tail dependence means that for larger values of U_1, large values of U_2 will also occur. It is in these dependence relationships that the most interesting information is found.

1.1 Benthic Data

As a motivating application, the data that is being used in this analysis is from a tidal region in St Anns Bay, NS, Canada, from Dowd et al. 2013 (submitted manuscript). The information in the data consists of coastal marine benthic macrofaunal data and its associated environmental data. Macrofaunal data is information that was collected from benthic, or soil organisms, that are found in the ocean. Typically data of this sort is collected on a 0.5mm sieve. It was collected on a sampling grid of 48 stations. This data was collected to assess the ecological health

of the environment. Variables such as abundance, species number, richness, and diversity were calculated from the macrofaunal data and categorized as faunal indices. Environmental variables such as porosity, organic matter, water depth, and many others were also collected at each of the sites. In total, 10 environmental variables were collected, as well as five faunal indices.

The data has already been analyzed with the goal of modeling the faunal data as the response variables and the environmental data as the explanatory variables. The rationale for modeling the data in this way is because as the faunal data provides more direct information about the health and diversity of the ecosystem, it can be quite difficult and costly to collect that information, compared to the environmental data. Therefore it may be more effective to make predictions about the health of the ecosystem, based on the environmental data collected. The analysis was based on multivariate generalized least squares regression. The data showed some high correlation and strong relationships amongst one another, especially spatially. It implies that variables close to one another are not independent, so neither are their errors (Dowd et al., 2013). This creates a complicated dependence structure, in particular, the dependence in the residuals, which is the motivation behind extending the approach to make use of copulas. That is, what can't be described by the regression, might be able to be explained through copulas.

2. Basic Theory

2.1 Definitions, Theorems and Families of Copulas

Copulas, simply, are functions that are used to describe the dependence between variables. They are a distribution function such that the cumulative density function (cdf) can be written in terms of the marginal distribution of the variables and the copula itself. A copula is a multivariate distribution whose marginals are all uniform over [0,1] (Yan, 2006). Any continuous random variable can be transformed, by its probability integral transform, to be uniform over [0,1], and so copulas can be used to give dependence information separate from that of the marginal distributions (Yan, 2006). Let the random vector $(X_1,...,X_d)$ exist with marginal cdfs $F_i(x) = P(X_i \le x)$. By the probability integral transform, any continuous random variable can be transformed to be uniform on [0,1]. Using this, the vector $(U_1,...,U_d) = (F_1(x_1),...,F_d(x_d))$ has uniform margins. Then the copula of $(X_1,...,X_d)$ is the joint cdf of $(U_1,...,U_d)$. Namely, it is

$$C(u_1,...,u_d) = P[U_1 \le u_1,...,U_d \le u_d].$$

This is the same as saying a copula is any function $C:[0,1]^n \to [0,1]$ with the following properties (de Matteis, 2001):

1. $C(x_i,...,x_n)$ is increasing in each component x_i.
2. $C(1,...1,x_i,1,...,1) = x_i$ for all $i \in \{1,...,n\}, x_i \in [0,1]$

3. For $a_i \leq b_i$, the probability $P(U_1 \in [a_1, b_1], ..., U_d \in [a_d, b_d])$ must be non-negative. This leads to the rectangle inequality

$$\sum_{i_1=1}^{2} \cdots \sum_{i_d=1}^{2} (-1)^{i_1+\cdots+i_d} C(u_{1,i_1}, \ldots, u_{d,i_d}) \geq 0,$$

These few properties are what help define copulas as the functions they are. It is these properties, as well as the following theorem, that is making the field of copulas more and more popular.

Copulas are useful in describing the full multivariate distribution of data or random variables and linking this full distribution to the univariate marginals of each variable (Frees & Valdez, 1998). The popularity of copulas stems from the fact that entire multivariate distributions can be decomposed from the joint distribution function of random variables to individual marginal distributions of these variables and the copula linking them together (Kang, 2007). In the past, multivariate distributions have been developed as extensions of the univariate distributions. However, as Frees and Valdez point out, there are several problems with this approach. The first is that a different family is needed for each marginal distribution. Another is that anything beyond a bivariate model quickly becomes complicated, and the last is that measures of association often show up in the marginal distributions. These issues can all be forgotten when dealing with copulas. As will be shown in this thesis, copulas allow for variables to have different marginals while having the same copula. This means that the choice of copula model selected to model the dependence between the variables can be independent of the marginal

distributions for those variables (Genest & Favre, 2007). This powerful feature is what makes copulas so useful.

One very important theorem in the copula world is a theorem by Sklar (1959). In terms of the bivariate case (examined here for simplicity), it states that the multivariate cumulative distribution function for the random vector (X,Y)

$$H(x,y) = P[X \leq x, Y \leq y]$$

can be written as

$$H(x,y) = C(F(x), G(x)), \quad x, y \in R$$

where $F(x)$ is the marginal of X, $G(y)$ is the marginal for Y and C is the copula function. Conversely, if $F(x)$ and $G(y)$ are continuous marginal distribution functions and C is a copula, then the function H defined by the above equation is a joint distribution function with marginal distributions $F(x)$ and $G(y)$. Sklar's theorem means that a multivariate distribution function can be broken up into marginal distributions of each variable and the copula form linking the margins together (Kang, 2007). This idea of breaking down the joint distribution is a primary motivation to a lot of the work done.

As the field of copulas becomes increasingly popular, more research is being done on them and more information is being discovered. There are several different types of copulas and they can be divided into two different families: Elliptical copulas and Archimedean copulas. An elliptical copula corresponds to an elliptical distribution by Sklar's theorem (Yan, 2006). In general, an elliptical distribution is a probability distribution that generalizes and inherits some properties of the multivariate normal distribution. Examples of elliptical copulas include the Gaussian

and Student's t-copula. Archimedean copulas are popular because they allow for the reduction of a multivariate copula to a single univariate function (Frees & Valdez) and also support specific features such as skewness. This family is constructed through generator functions. If φ is the generator function, then an Archimedean copula has the form

$$C(u_1, \ldots, u_p) = \varphi^{-1}\{\varphi(u_1) + \cdots + \varphi(u_p)\}$$

where φ^{-1} is the inverse of the generator. In order for the above function to be a copula, the generator function needs to be a complete monotonic function (Yan, 2006). Table 1 below gives the generator functions for different types of Archimedean copulas. As shown in this table, different choice of generator functions yields different Archimedean copulas (Frees & Valdez).

Family	Generator $\phi(t)$	Dependence Parameter (α) Space	Bivariate Copula $C_\phi(u,v)$
Independence	$-\ln t$	Not applicable	uv
Clayton (1978), Cook-Johnson (1981), Oakes (1982)	$t^{-\alpha} - 1$	$\alpha > 1$	$\left(u^{-\alpha} + v^{-\alpha} - 1\right)^{-1/\alpha}$
Gumbel (1960), Hougaard (1986)	$\left(-\ln t\right)^{\alpha}$	$\alpha \geq 1$	$\exp\left\{-\left[\left(-\ln u\right)^{\alpha} + \left(-\ln v\right)^{\alpha}\right]^{1/\alpha}\right\}$
Frank (1979)	$\ln\frac{e^{\alpha t} - 1}{e^{\alpha} - 1}$	$-\infty < \alpha < \infty$	$\frac{1}{\alpha}\ln\left(1 + \frac{(e^{\alpha u} - 1)(e^{\alpha v} - 1)}{e^{\alpha} - 1}\right)$

Table 1*: Archimedean Copulas and their generators*

Some advantages to working with Archimedean copulas include the ease in which they can be constructed, the large variety of dependence structures that can be constructed with this family and the nice properties that come along with the members of this family (de Matteis, 2001). As observed in the above table, three of the more popular Archimedean copulas are the Clayton, the Gumbel and the Frank

copula. Their form and generator function can been seen above, and for the bivariate case, they are shown in Figure 1 (from Yan, 2006),

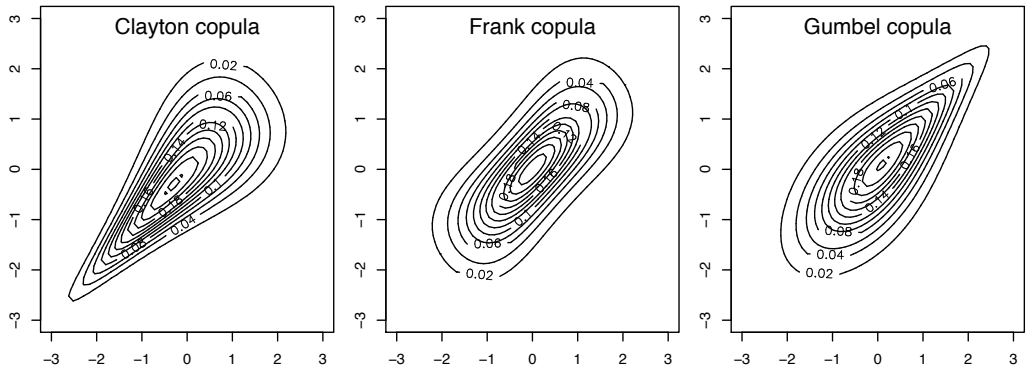


Figure 1*: Contour plots of Archimedean Copulas*

In these contour plots, the differences amongst the dependence structure can be clearly seen. Archimedean copulas are a good choice for modeling bivariate distributions, but in higher dimensions, they tend to be too restrictive. This is because they assume the exact same dependence structure between all the pairs of variables in the data (Kang, 2007). Much of this thesis, however, will be spent examining the elliptical family of copulas, mainly the Gaussian and Student t-copula.

The Gaussian copula is a part of the elliptical family of copulas. It is related to the normal distribution. The normal distribution is, of course, one of the most popular distributions in the statistical world. The Gaussian copula, in one of its simplest forms, is given by

$$C_\rho(u,v) = \Phi_\rho(\Phi^{-1}(u) + \Phi^{-1}(v))$$

where Φ_ρ and Φ are the bivariate and univariate standard normal cumulative distribution functions respectively, and ρ is the coefficient of correlation between the random variables X and Y[2] (Arnold, 2006). To get an understanding of what a

Gaussian copula could look like, a sample of size 1000 was generated from a Gaussian copula, with ρ=0.75, in R. It is shown in Figure 2a below.

(a)

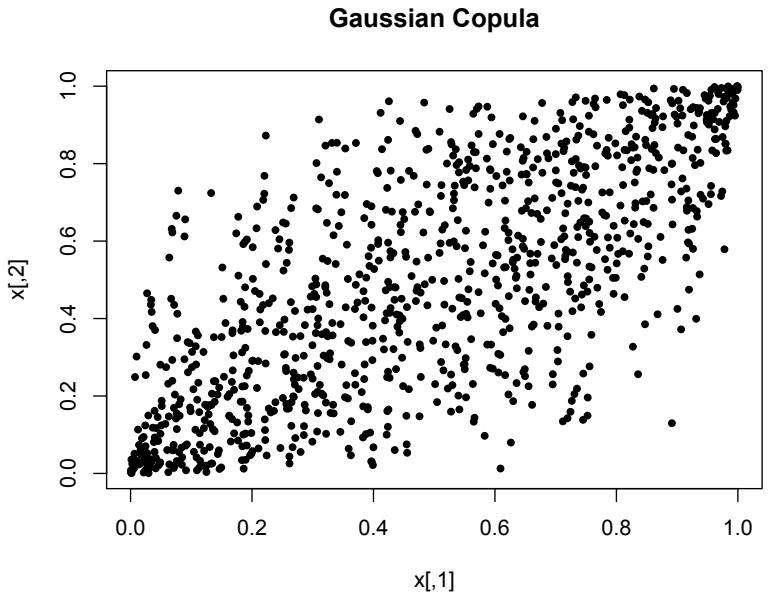


Figure 2a: *Data generated from a Gaussian copula with* $\rho = 0.75$

In this figure, the dependence with a correlation of 0.75 is quite visible. The uniform marginals can also be picked out. There is a clear cyclical distribution in the contours of this plot as well.

Another copula from the elliptical family is the Student t-copula. Like the t-distribution is similar to the normal distribution, the t-copula is similar to the Gaussian copula, but it has an extra parameter. This parameter is similar to degrees of freedom of a t-distributions and the purpose of this parameter is to control the tail dependence (Arnold, 2006). Small values of this parameter ν correspond to increasing the amount of probability in the tails of the copula, meaning an increase in the probability of joint extreme events. Continuing to parallel with the univariate

distribution, with higher values of ν, the Student's t-copula approaches the Gaussian copula. The Student's t-copula has the form

$$C_{\rho,\nu}(u,v) = t_{\rho,\nu}(t_\nu^{-1}(u) + t_\nu^{-1}(v))$$

where t_ν and $t_{\rho,\nu}$ are the univariate and bivariate standard Student's t cdfs with ν degrees of freedom, and ρ is the correlation coefficient between the random variables X and Y (Arnold, 2006). As with the Gaussian copula, a sample of size 1000 was generated from the Student's t-Copula with ρ=0.75 and four degrees of freedom. It is shown in Figure 2b.

(b)

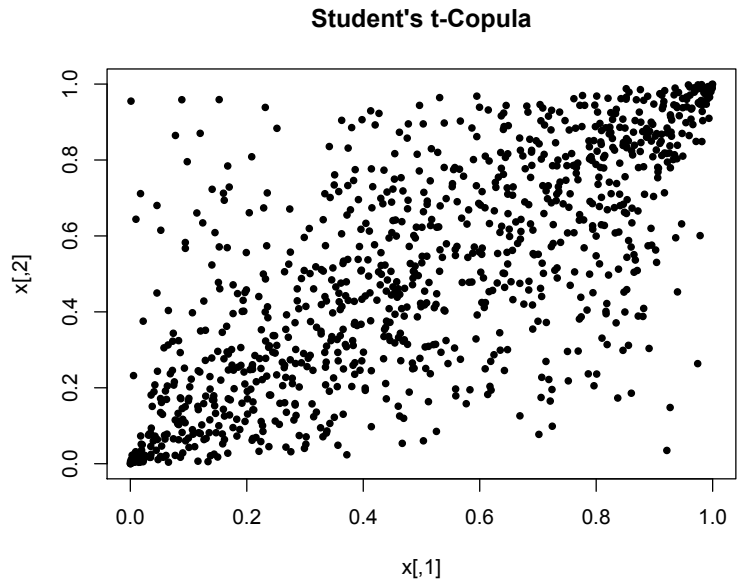


Figure 2b: *Data generated from a Student's t-copula with ρ =0.75 and four degrees of freedom*

The similarities between the Gaussian and Student's t-Copula are evident in the two corresponding plots, however, there is more of a dependence in the tails of the Student's t-Copula. This is illustrated by a higher concentration of points in the tails of the Student's t-Copula compared to the Gaussian.

2.2 Measures of Dependence

Since copulas are a way of defining multivariate distributions, meaning there are multiple variables being used at the same time, dependence and correlation amongst those variables is something that must be discussed. In the grand scheme of things, dependence is any statistical relationship between two random variables. There is a lot more to it than just that simple definition, however that is the basis of understanding dependence - it is a relationship between variables.

Correlation includes any of a wide range of statistical relationships involving dependence. Correlation is often used to describe or quantify the dependence among random variables. For example, correlation can easily be used to describe the demand for a product and its price. The higher the price of a product, the lower the demand for that product, and vice versa, implying a negative correlation between demand and price. Correlation can thus indicate a predictive relationship.
It is a very specialized type of relationship between variables. There are several different coefficients that are used to measure the degree of correlation. The more common ones include Pearson's correlations coefficient, Spearman's rho, and Kendall's tau. The latter two use the ranks of the data instead of the data itself and have the advantage of being more robust, i.e. less sensitive to departures from normality in the data. The rank coefficients are also invariant under monotonic transformations and are more robust against outliers (de Matteis, 2001). However, these rank correlations are not moment-based correlations and therefore they cannot be subject to the same variance-covariance manipulations such as linear correlation (de Matteis, 2001).

The Pearson's correlation coefficient is a very common measure of dependence. It captures only linear relationships between two variables. It is defined as:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}.$$

A Pearson value of +1 indicates a perfect positive linear relationship between the two variables, a value of 0 indicates no linear relationship at all, and a value of -1 indicates a perfect negative linear relationship.

Dependence structures are much more than just correlations. In fact, outside of the elliptical world, linear correlations must be used with relative caution (de Matteis, 2001). To see how dependence structure can vary, de Matteis (2001) shows that even though two copulas can be generated from the same distribution, with the same correlation, their dependence structure can be quite different.

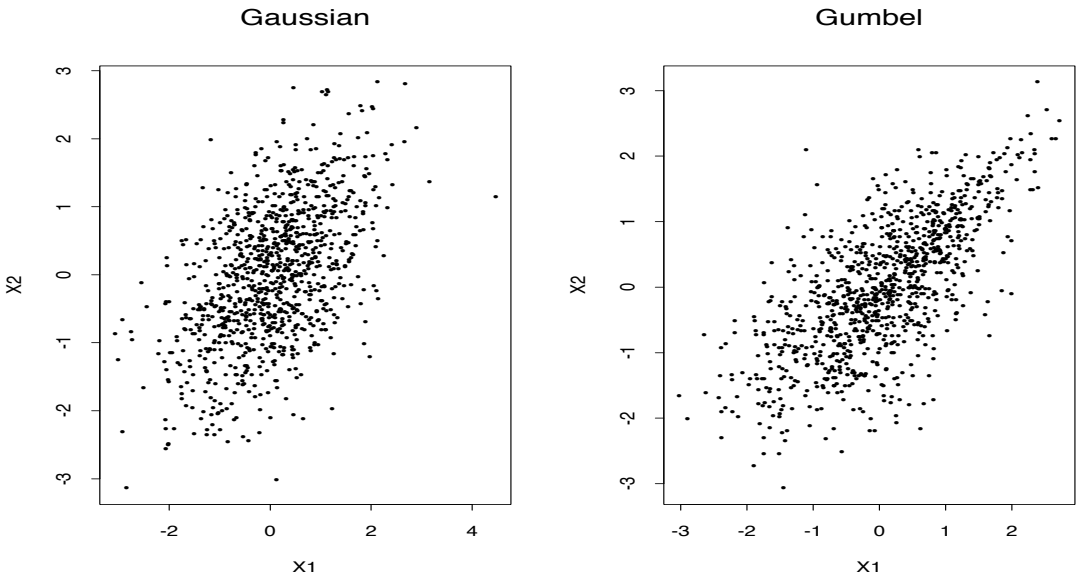


Figure 3: *Example of a differing dependence structure from two different copulas from de Matteis (2001).*

Figure 3 above shows 1000 random variates from two distributions with identical standard normal distributions and identical correlation of $\rho=0.5$, but with notably different dependence structures; one using an elliptical Gaussian copula, the other using the Archimedean Gumbel copula. These differences can be seen in the relationship between the points. The Gaussian copula is more circular in its points and more concentrated to lower values of X1, where as the Gumbel is more elongated and shares the concentration among higher and lower values of X1. Seeing the varying dependence structures shows the importance of defining the proper structure when attempting to use copulas to model data.

With a little bit of knowledge about the basic theory of copulas, one way to get a better understanding of the power of these functions is to apply this knowledge to simulated and real data in R and work with the results. The following sections have several exercises dealing with different components of the copula theory.

3. Empirical Copula

This section is motivated by the work and research done by Genest and Favre (2007). This work is driven by the use of dependence ranks to define an empirical copula. Genest and Favre showed that the ranks of a data set can be used to define an empirical copula, which in turn, can then be used to calculate dependence empirically.

Using R statistical software, a very small data set (n=10) was generated from a bivariate normal distribution with zero mean and zero correlation. The data can be found in Table 2. The pairs are listed so that $X_1 < \ldots < X_{10}$ for simplicity.

i	1	2	3	4	5	6	7	8	9	10
Xi	-0.835	-0.82	-0.626	-0.305	-0.184	0.329	0.487	0.576	0.738	1.595
Yi	-0.621	-0.045	1.511	0.594	0.39	1.125	-0.016	0.821	0.944	-2.215

Table 2*: Data set from a bivariate normal distribution*

A scatterplot of this data set (Figure 4) was constructed and observed. The goal of this plot is to get a feeling for the dependence between the two generated variables.

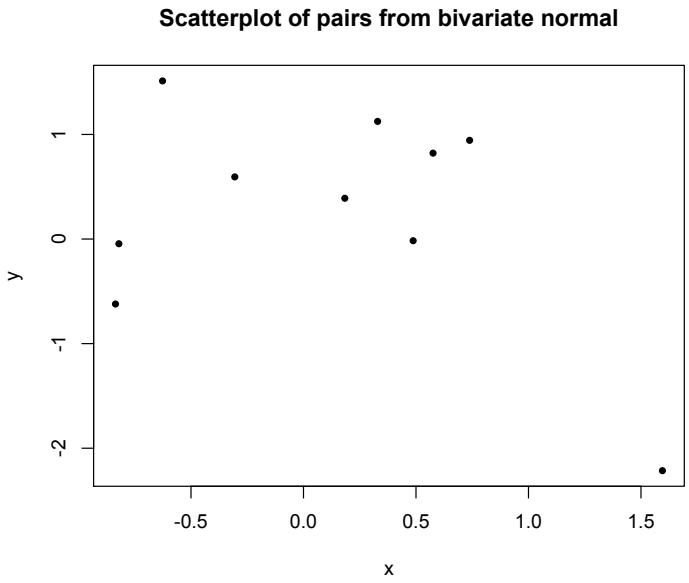


Figure 4*: Scatterplot of the data set*

Something as simple as a scatterplot can prove to be very important. It is an easy way of visualizing and getting a good feeling for the dependence between two variables (Genest & Favre, 2007). If something has a linear relationship or a high correlation, this will be very evident in the scatterplot itself. On the other end of the spectrum, if there is no relationship at all, this will also be clearly shown. The scatter

plot also gives information about the marginal behavior of the variables is included as well (Genest & Favre, 2007). This can be seen with the scatterplot of the variables Z and T, which are transformations of X and Y according to the following,

$$Z_i = \exp(X_i), \qquad T_i = \exp(3Y_i), \quad i = 1,...,10.$$

The scatterplot of these variables is shown in Figure 5, and it is clear that this plot is exceedingly different than that in Figure 4. In Figure 4, all the points are spread about about the X variable, but more concentrated to the higher values of Y. The last point also seems to be an outlier. The in scatterplot of Z and T, the points are more concentrated to the lower values of both variables, and there appears to be more than one outlier (the third and last).

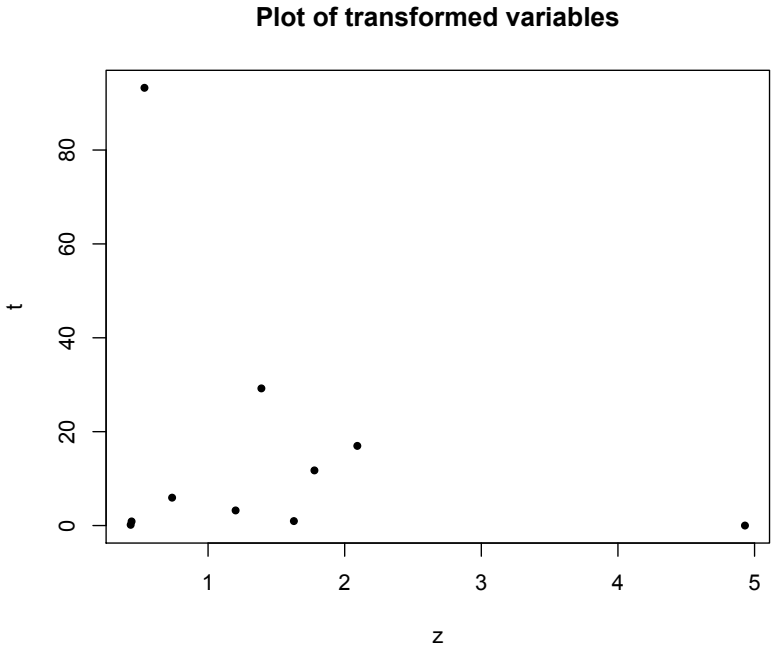


Figure 5*: Plot of Z, T*

The next step of this exercise was to use the ranks of the data to help define the dependence between X and Y, as well as to define the empirical copula. As previously mentioned, the ranks of data are invariant under monotonic transformations and are more robust to outliers. Because of this, ranks of data can be very important to statistics. For this data set, the ranks were found and recorded. These ranks were used to calculate a value for Spearman's rho by the formula

$$\rho_n = \frac{12}{n(n+1)(n-1)} \sum_{i=1}^{n} R_i S_i - 3\frac{n+1}{n-1}$$

where R_i is the rank of X_i and S_i is the corresponding rank of Y_i (Genest & Favre, 2007). The use of these ranks leads into the concept of the empirical formula for the copula. The empirical copula is a sample-based version of the copula for the data. It is defined by

$$C_n(u,v) = \frac{1}{n} \sum_{i=1}^{n} 1(\frac{R_i}{n+1} \le u, \frac{S_i}{n+1} \le v)$$

where 1(A) is the indicator function for a set A, meaning if , for $1(x)$, $x \in A$, the function takes on a value of one, if not, the value of the function is 0. Genest and Favre (2007) showed that the empirical copula can be used to calculate empirical versions of both Spearman's rho and Kendall's tau, both which are measures of dependence. Spearman's rho, as a function of the empirical copula is defined by Genest & Favre as

$$12 \int_{[0,1]^2} uv dC_n(u,v) - 3 = \frac{12}{n} \sum_{i=1}^{n} \frac{R_i}{n+1} \frac{S_i}{n+1} - 3 = \frac{n-1}{n+1} \rho_n$$

.

The empirical formula for Kendall's tau is

$$\tau_n = 4\frac{n}{n-1}\bar{W} - \frac{n+3}{n-1}$$

$$\text{where } \overline{W} = \int_{[0,1]^2} C_n(u,v)\,dC_n(u,v).$$

These empirical copulas are just sample based estimations of the true copula, but can be used to estimate values of dependence measures. All the R code for this example can be found in Appendix I.

This example was also repeated with a larger data set (n=1000), but is not shown here . It was very clear with this data, by plotting the histograms of each of the variables that their marginals are of a normal distribution. Being able to change the marginal and having different marginal with a copula function will be discussed later on.

4. Creating Multivariate distributions from the R copula library

Copulas are a convenient way of defining more complex multivariate distributions. It has been mentioned that with a marginal distribution for each variable and the proper copula, the entire joint distribution function can be described. As multivariate distributions can quickly become very complex this can prove to be a very useful tool. Like most things in statistics, to do all the calculations by hand or from scratch can be a daunting and challenging task. The R statistical software has a *copula* library with plenty of functions and commands to define not only just copula's, but entire multivariate distributions themselves. Within the library, there is a *copula* class that can be used to define copula objects, and there is

a *mvdc* class that uses copulas to define entire multivariate distributions (Yan, 2006). An example of a copula created using the copula command is the Gaussian copula that was shown in the "Basic Theory" section. It was created using the following code:

```
>library(copula)
set.seed(1)

norm.cop=normalCopula(.75, dim=2)
norm.cop
x=rcopula(norm.cop, 1000)
plot(x, pch=20)
title("Gaussian Copula")
```

The variable x was sampled from the copula created and was previously shown in Figure 2a.

The *mvdc* command has three major arguments: first is the copula, which specifies the copula C. Second is the *margins* arguments, where each individual margin is named. The third is the *paramMargins* argument. This argument is a set of a list, where each list specifies the values of the parameters for each of the marginal distributions (Yan, 2006). With these three simple components, multivariate distributions can be constructed, with one having full control over the margins and the copula separately (as is suggested by Sklar's Theorem).

Each different multivariate distribution comes with its own set of distinct properties. Some of these properties can be seen in the contour plots of these distributions (that is, contours of probability density). Contours for a few of the Archimedean copulas have already been shown. The Frank, Gumbel, and Clayton copulas each have their own specific form (Figure 1). Plotting these contours can be a powerful method in defining particular distributions. A wide range of distributions

can be created by simply changing the marginal distributions. For instance, Figure 6 shows the contour, scatterplot, and the two marginals of a multivariate distribution with two standard normal marginals and a Gaussian copula with ρ=-0.5.

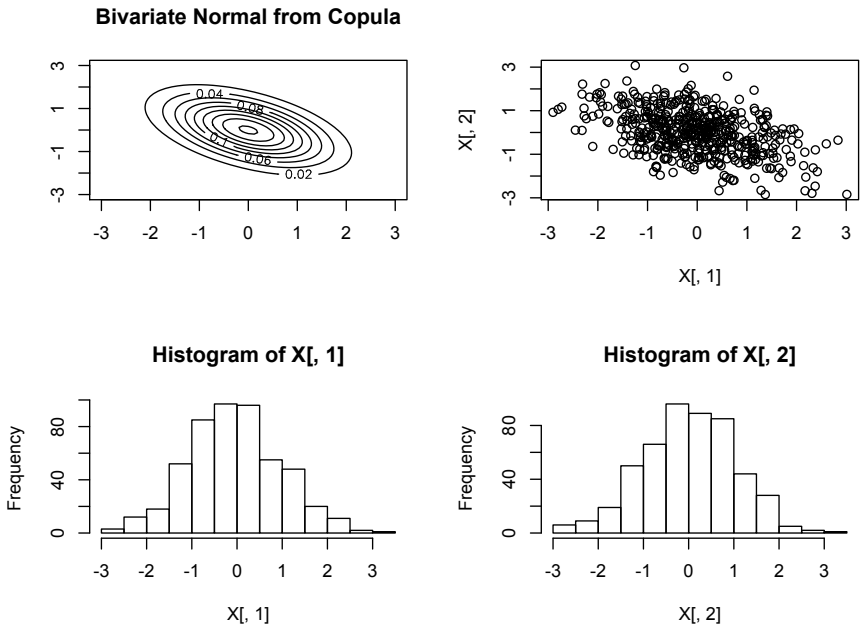


Figure 6*: A multivariate distribution with two standard normal marginals and a Gaussian copula.*

The *mvdc* command used to create this particular distribution was

```
> norm.cop <- normalCopula(-0.5, dim =2)
MVN <- mvdc(norm.cop, margins = c("norm", "norm"), paramMargins =
list(list(mean = 0, sd = 1), list(mean = 0, sd = 1)))
```

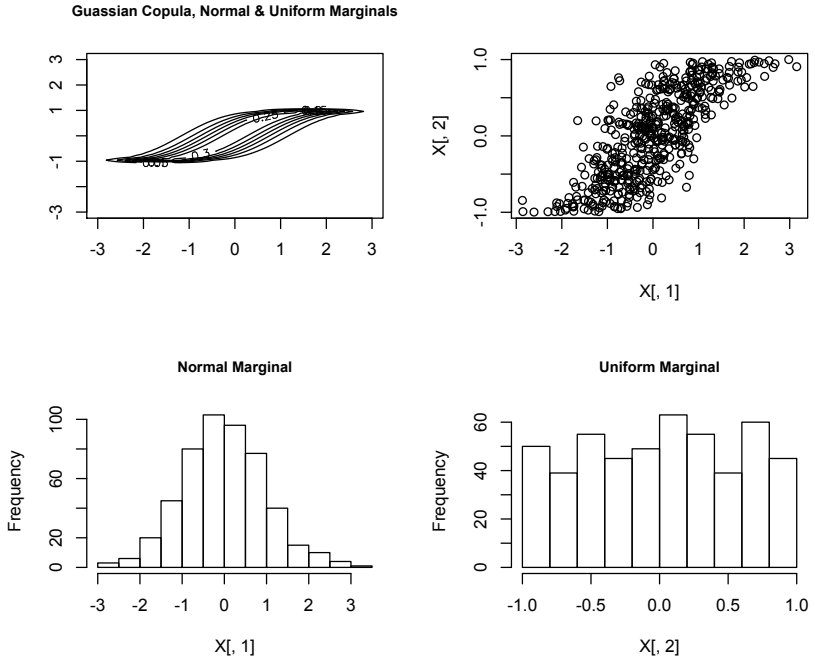
The 'norm.cop' is the copula object created using the *copula* class, and the MVN object is the multivariate distribution itself. The rest of the R code used to make the plot is as follows;

```
>par(mfrow=c(2,2))
contour(MVN, dmvdc, xlim = c(-3, 3), ylim = c(-3, 3))
title("Bivariate Normal from Copula")
X=rmvdc(MVN,500)   ##sampling from the MV normal created from the
 copula
plot(X[,1], X[,2])
```

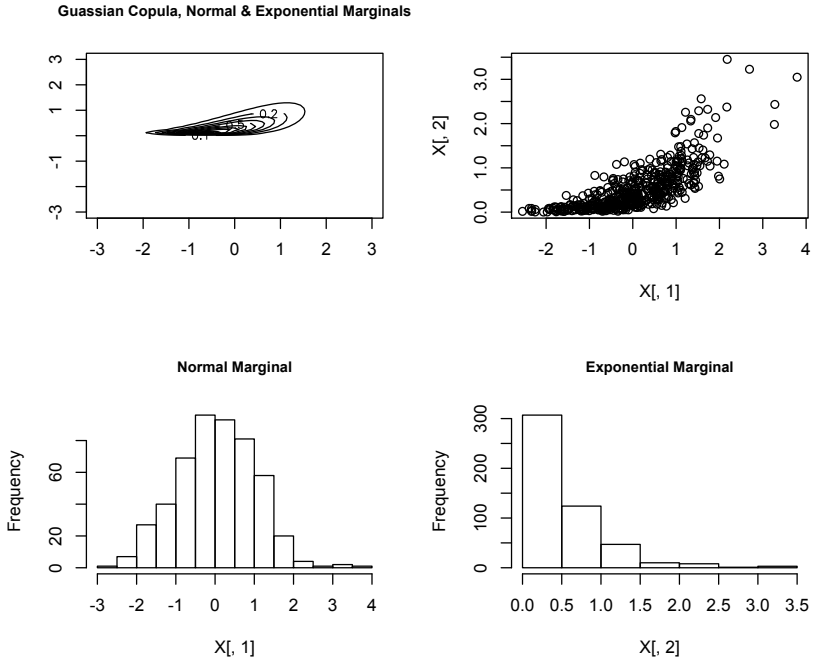
```
hist(X[,1])
hist(X[,2])
```

As previously mentioned, having complete control of the marginal distributions gives one the ability to make numerous distributions which are completely different from one another. For example, simply by changing one of the standard normal marginals from the distribution above to a uniform marginal with parameter values [-1,1], the new distribution looks like the one in Figure 7a. The contour in Figure 7a is much more skewed than that in Figure 6. With two normal margins, there is a certain amount of cyclical symmetry within the points. By changing one marginal, there becomes an "S-shaped" distribution of points, where the concentration of points tapers off at the bottom and top of the graph. The *mvdc* command has many built in distributions such as normal, uniform, exponential, alpha and beta. Any combination of these marginal distribution, with given parameter values allows for the construction of many "creative looking" distributions. The rest of the distributions in Figure 7 were all created by using a combination of these marginal distributions and the *mvdc* command.

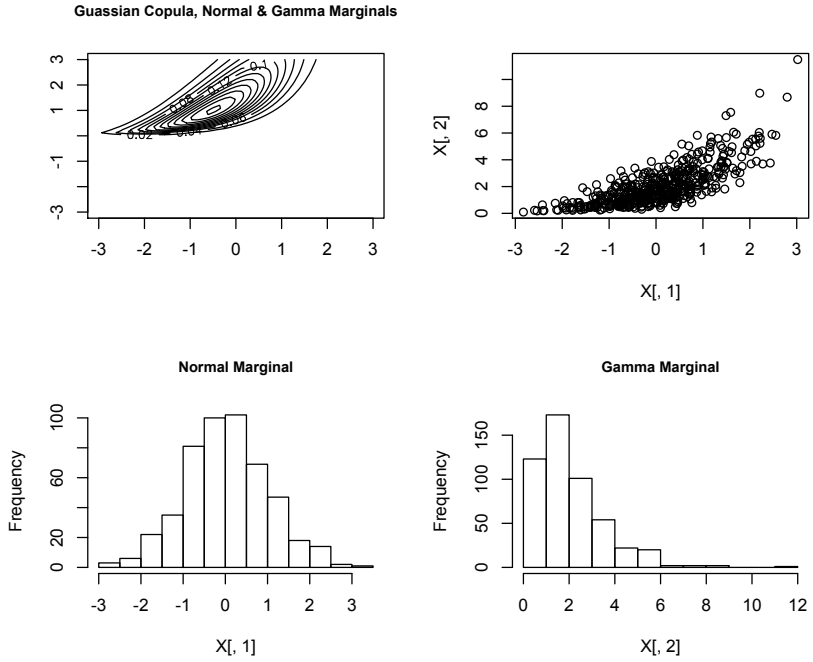
(a)


Guassian Copula, Normal & Uniform Marginals
X[, 1]
X[, 2]
Normal Marginal
Uniform Marginal
Frequency
X[, 1]
X[, 2]

(b)


Guassian Copula, Normal & Exponential Marginals
X[, 1]
X[, 2]
Normal Marginal
Exponential Marginal
Frequency
X[, 1]
X[, 2]

(c)



(d)

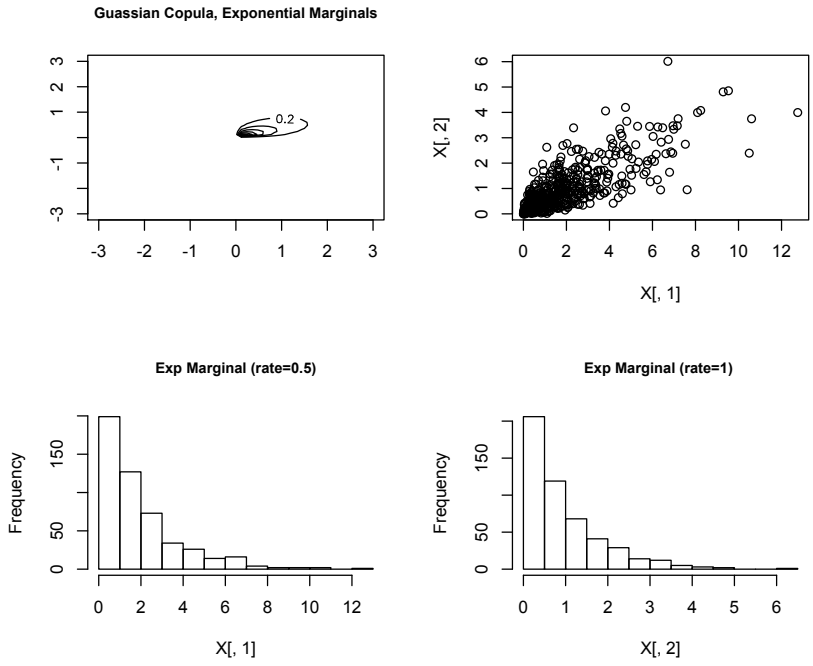


Figure 7: *(a) Distribution created from a normal and a uniform marginal distribution. (b) Distribution created from a normal and an exponential marginal. (c) Distribution*

created from a normal and a gamma marginal. (d) Distribution created from two exponential distribution with different rates.

One interesting thing to note from this exercise is that all of these distributions were created with only one type of copula; the Gaussian. All the *mvdc* objects were created with a Gaussian copula with a correlation value of 0.8. Many more of these "creative" distributions could have been created by also changing the dependence structure, which is controlled by the copula.

Another use for copulas in the *mvdc* object is the random number generator. In the code, we have already seen the *rcopula* command, which is the method for random number generation for the copula objects created in R (Yan, 2006). This links back to Sklar's theorem again, as it suggests that random numbers from a copula can be produced by using the probability integral transform to transform the margin of random numbers from its multivariate distribution (Yan, 2006).

5. Copulas from the Empirical Moments

Sklar's theorem for copulas states that multivariate distributions can be fully defined by the marginal distributions for each of the variables and the proper copula that describes the dependence between those random variables and that links the marginals together (Kang, 2007). This concept was explored in another exercise completed in R. The idea was that, given a data set, the marginal distributions could be estimated, along with their parameter values. Once the distributions for each random variable were fitted to data, the copula would hopefully also be able to be fitted, thus giving a multivariate distribution to model the data. This was first done

via simulation with R using synthetic or generated data, with the intention of extending the work to two chosen variables from the real benthic data from St Ann's Bay.

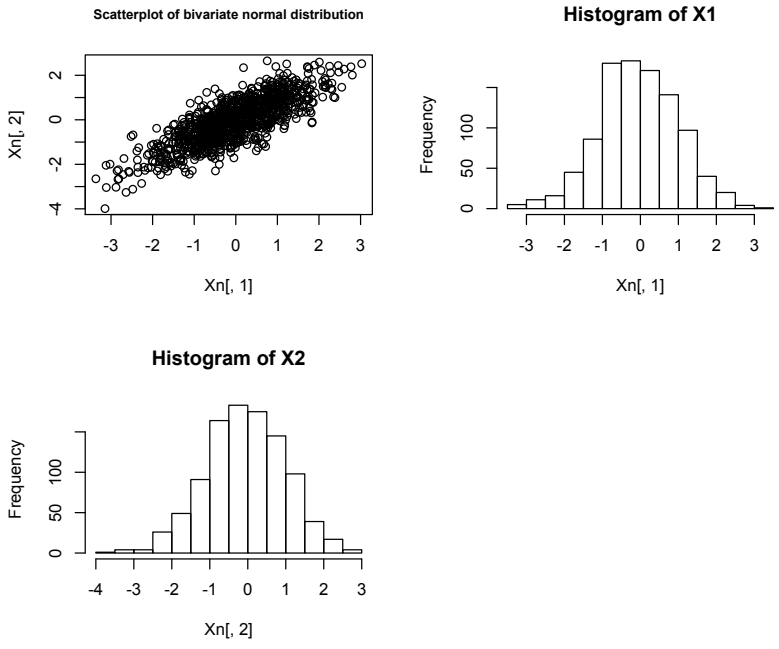
The basic idea, when applying it to real data, is to look at the marginals and essentially "guess" the type of distribution. From there, the parameter values for each distribution can be estimated using a log-likelihood method (the *fitdistr* command in R proves to be helpful here). For the time being, only two variables are being analyzed, so the correlation between these two variables can also be determined rather simply. It has been said that it is important to define the marginal distributions properly, and once this is done, the task of defining the complete multivariate distribution becomes that of properly choosing a copula that best describes the dependence structure between the variables (Kang, 2007).

Before attempting to apply this idea to real data, simulated data was created in R. In this particular example, a bivariate normal distribution was generated using the *rmvnorm* command, and a correlation of 0.8 was chosen for the two variables. Figure 8a shows the scatterplot of the data as well as a histogram of each of the marginal distributions. Then, using the following code, the parameter values of the marginals were estimated. Since the true values of the parameters were known (because they were chosen and simulated), the estimated values could be compared to the true values. As expected, the estimate was very similar to the true value. These values were then used to overlay the estimated normal curve on the histogram to assure the fit was acceptable, shown in Figure 8b.

```
>library(MASS)
library(mvtnorm)
```

```
# First generate from a bivariate normal
C <- matrix(c(1,.8,.8,1),2,2)
Xn <- rmvnorm(mean=c(0,0),sig=C,n=1000)

par(mfrow=c(2,2))
plot(Xn[,1], Xn[,2])
title("Scatterplot of bivariate normal distribution", cex.main=0.8)
hist(Xn[,1], main="Histogram of X1")
hist(Xn[,2], main="Histogram of X2")

#emperical moments of marginals and correlation of 2 variables
a=fitdistr(Xn[,1],"normal")
b=fitdistr(Xn[,2], "normal")
cor(Xn[,1],Xn[,2])

##fitting the parameter estimates the data (histogram)
par(mfrow=c(1,1))
hist(Xn[,1], prob=TRUE)
mu1=a$estimate[1]
sd1=a$estimate[2]
x=seq(min(Xn[,1]),max(Xn[,1]),length=50)
y=dnorm(x,mu1,sd1)
lines(x,y, col="red")
```

(a)



(b)

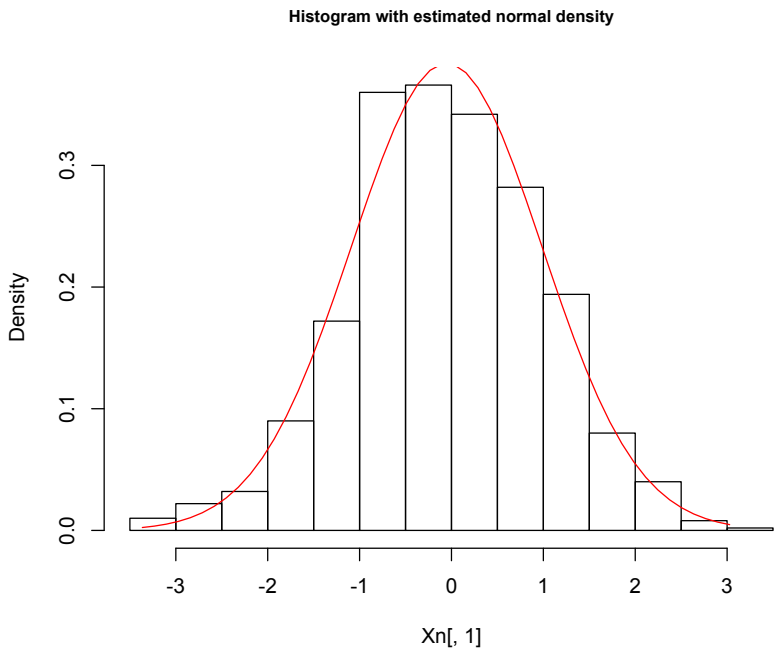


Figure 8: *(a) Scatterplot and marginals of a bivariate standard normal distribution. (b)histogram of X1 with the estimated normal density curve over top*

This process was repeated for several other distributions, such as the exponential and gamma, all of which were successful in fitting the empirical moments.

Since the topic of this thesis is copulas, the more illustrative approach would be to generate the data, not from a bivariate normal distribution strictly, but from a multivariate distribution that was defined by a given copula. With the *mvdc* command, the marginals are controlled independent of the copula, so the first attempt was the very basic Gaussian copula with standard normal marginals. Data was generated from this multivariate distribution, and plotted. The same process was carried out, fitting the distributions and getting values for the empirical moments. The fitted values were approximated well ($\mu_1 = -0.038, \sigma_1 = 1.04$, $\mu_2 = -0.033, \sigma_2 = 0.983$, and $\rho = 0.804$). The sample size was changed on different occasions, and as expected, the smaller the sample size, the less accurate the estimate for the true parameter values. Different multivariate distributions with marginals different from a standard normal were also simulated and fitted with success. For example, the results of a multivariate distribution with a standard normal and exponential marginal with rate=2 are shown below in Figure 9:

Standard Normal:		Exponential
mean	sd	rate
-0.02561710	1.01949637	2.04645669

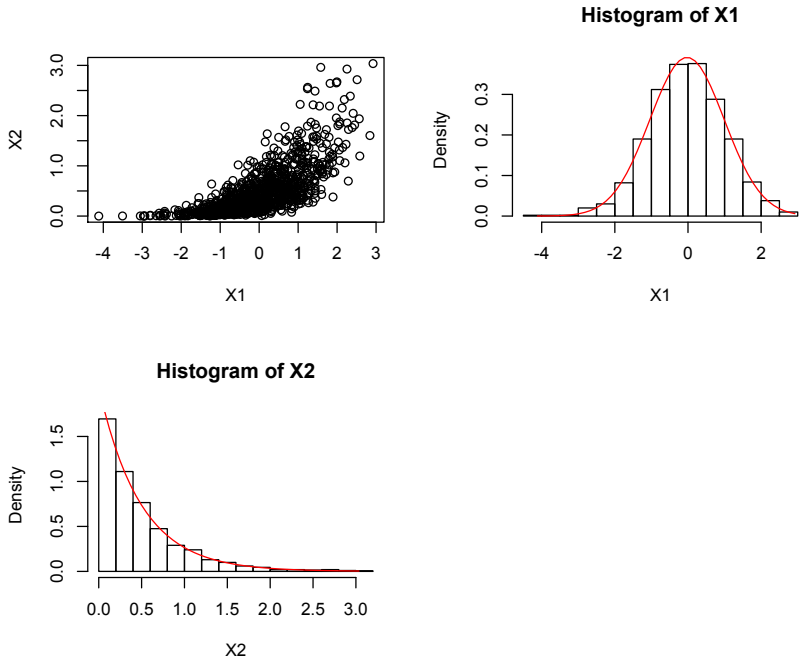


Figure 9: *Multivariate distribution with a normal and an exponential marginal are shown with the overlay of the fitted marginal distributions shown in red.*

The calculated parameter values are a good estimate for the true values and fit each respective distribution well. The sample value for the correlation can also be calculated using numerous methods (Spearman's rho, or the most popular, Pearson's correlation coefficient). With this information, a proper copula can be selected that best represents the link between the two variables. In this case, a Gaussian copula with the sample correlation is appropriate. With this copula and marginal distributions, the multivariate distribution is complete.

Moving forward with this work, the next step is to work with real data instead of simulated data from R. This will show the practicality of this method and how its ability to apply it to the real world. In the benthic data, two response variables were chosen. For the purpose of this exercise, the abundance and species number variables were selected. Figure 10 shows the scatterplot of the multivariate

distribution created by these two variables, as well as their individual marginals. By visually assessing the marginal, a normal distribution was fitted to the abundance variable, and both an exponential and gamma distribution were fitted to the species number. Of the two distributions fitted to the species number variable, both were visually deemed an appropriate fit, however the gamma distribution was chosen as the value for the maximum log-likelihood was slightly larger. The estimated mean and standard deviation for the normal distribution were found to be 2.67 and 1.16 respectively. The shape parameter estimate for the gamma distribution was estimated to be 1.46 and the rate was estimated to be 0.25. These fitted curves were overlaid on the original data in red and are seen in Figure 8.

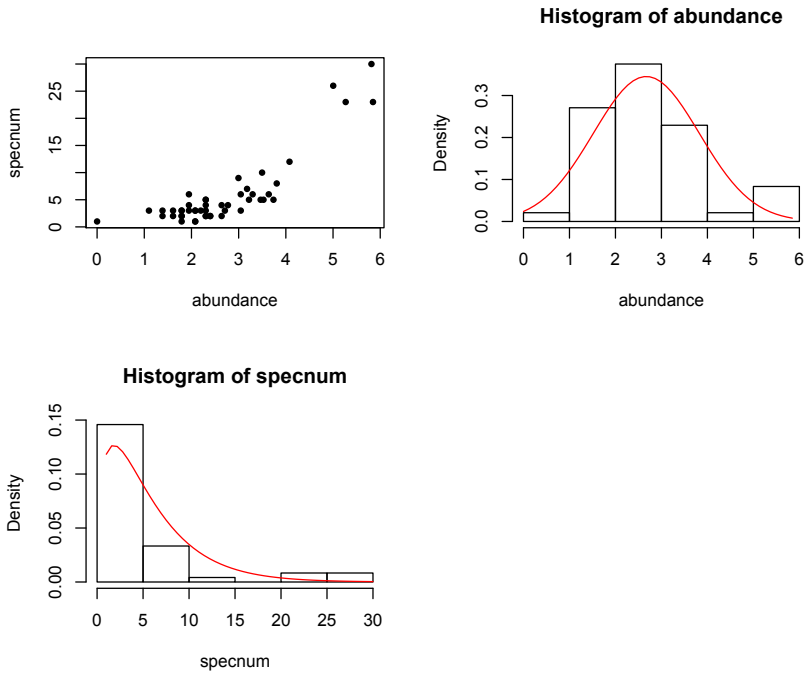


Figure 10: *A multivariate distribution for two response variables from the benthic data, as well as the marginal fitted curves.*

Comparing the scatterplot of these two variables to the multivariate distribution created with a normal and gamma marginal and a Gaussian copula, as seen in Figure

5c, it is appropriate to join the two marginals with a Gaussian copula, where the correlation was found to be 0.853. This correlation is the estimated value for the parameter of the Gaussian copula. The R code used to fit these marginal distributions is as follows:

```
X=cbind(abundance,specnum)
par(mfrow=c(2,2))
plot(abundance,specnum, pch=20)

fit1=fitdistr(abundance,"normal")
hist(abundance, prob=TRUE)
mu1=fit1$estimate[1]
sd1=fit1$estimate[2]
x1=seq(min(abundance),max(abundance),length=50)
y1=dnorm(x1,mu1,sd1)
lines(x1,y1,col="red")

fit2=fitdistr(specnum,"gamma")
hist(specnum, prob=TRUE)
shape=fit2$estimate[1]
rate=fit2$estimate[2]
x1=seq(min(specnum),max(specnum),length=50)
y1=dgamma(x1,shape,rate)
lines(x1,y1,col="red")

cor=cor(abundance, specnum)
```

Comparing the scatterplot of the data in Figure 10 with that of the multivariate distribution created in Figure 7c, there are definite similarities between the real data and the simulated data. The real data takes the same shape as the simulated data (which has a normal marginal and a gamma marginal), indicating the dependence structure is likely appropriate for these two variables. Fitting these two variables using fitting formulas in R is demonstrated in the next section.

6. Fitting Copulas with built-in R Functions

Using the empirical moments, and log-likelihood methods to estimate parameter values, in theory this should be enough to define a multivariate distribution for the data. However, fitting the proper marginal distributions is very important to this process, and when the data is not simulated and the marginals are not exactly known there is a lot of possibility for error. Also, with all the available copulas, selecting the most appropriate one to model the dependence between the variables can be tricky. Likewise, with more than two variables, the dependence structure can quickly become much more complicated. Other methods to fitting copulas and multivariate distributions will be explored in the next section, as well as looking at copulas in higher dimensions.

6.1 Fitting copulas with *fitMvdc*

In the section above, fitting copulas and entire multivariate distributions to data by estimating their empirical moments and sample correlation was introduced. In the *copula* library in R there is a *fit* command that can be used to fit the marginals and copulas to the given data set called *fitMvdc*. Since the ability to fit copulas to describe the complex dependence between variables is one of the goals when dealing with multivariable data, the ability to model these copulas in statistical software programs like R will be a necessity in the future. In this section, this *fitMvdc* command will be explored.

There are density functions for *copula* and *mvdc* objects available, and because of their availability, it has made it much easier to fit copula models using the maximum likelihood method (Yan, 2006). The *fitMvdc* command is used to carry out estimation and it also reports on the results of the fitted model for *mvdc* objects.

There is a corresponding *fitCopula* command that reports the results for *copula* objects. The main three arguments (also the first three arguments) of the *fitMvdc* command are the data, the *mvdc* object, and the starting values (Yan, 2006). After those arguments, the remaining arguments are mostly control parameters that are specific to the situation. For the starting values, it is convenient to use the estimated marginal parameters chosen by fitting each marginal separately, as was discussed in the previous section. When specifying the starting values, parameter estimates for the individual marginals are first and then the copula parameters follow. Simulated data from R was selected and the *fitMvdc* command was used to see how accurate the fit as compared to the true data. The R code used, and the output can be summarized as;

```
>#  generate "data" from a distribution with one normal marginals,
 plus correlation
norm.cop <- normalCopula(0.8, dim =2)
myD <- mvdc(norm.cop, margins = c("norm", "norm"), paramMargins =
 list(list(mean = 0, sd = 1), list(0,1)))
```

The simulated data was a *mvdc* object with a Gaussian copula and two standard normal marginals, and the correlation between the two variables was set to 0.8.

```
par(mfrow=c(2,2))
contour(myD, dmvdc, xlim = c(-3, 3), ylim = c(-3, 3))
X=rmvdc(myD,500)
plot(X[,1], X[,2])
hist(X[,1])
hist(X[,2])


# Now see if you can recover the parameters
start <- c(0, 1, 0, 1, 0.5) # starting value (note: only
 correlation mis-specified)
```

For the starting values, only the correlation was misspecified. This was done to see if

the fit would recover the proper parameter values.

```
fit <- fitMvdc(X, myD, start = start, optim.control = list(trace =
 TRUE, maxit = 2000))
initial  value 1268.038801
iter  10 value 1179.849215
final  value 1179.676608
converged
initial  value 1179.676608
final  value 1179.676602
    stopped after 1 iterations

fit
The Maximum Likelihood estimation is based on  500  observations.
Margin  1 :
        Estimate Std. Error
m1.mean  0.01447      0.047
m1.sd    1.04385      0.033
Margin  2 :
        Estimate Std. Error
m2.mean  0.00154      0.046
m2.sd    1.01813      0.032
Copula:
      Estimate Std. Error
rho.1   0.8124      0.015
The maximized loglikelihood is  -1179.677
Optimization converged
Number of loglikelihood evaluations:
function gradient
      74       15
```

As shown above, the fit command had no issue recovering parameter values close to the true value. The mean and standard deviations for the marginals are approximately 0 and 1 respectively, and the correlation recovered was 0.812, after it had a starting value 0.5. The maximum log likelihood is also available with this fit output. It can be read directly from the output, or the *loglikMvdc* command in R gives the maximum log likelihood as its output. The maximum log likelihood was found to be 1179.6. For this example, since all the true distributions and parameter values are known, the real maximum log likelihood can be found and compared to

the fitted one. In this case, the following command calculates the true log likelihood value;

```
>maxL <- loglikMvdc(c(0, 1, 0, 1, 0.8), X, myD); maxL
[1] -1180.845
```

The log likelihood from the fitted model is very close to that of the true value.

Using the *fitMvdc* in this example was pretty straight forward, as all the information about the data was known before trying to fit the model. In the real world, this is obviously not the case. This begs the question as to what the limitations are for successful application of this fitting procedure.. That is, what happens when more than one parameter estimate is misspecified? What happens when the dependence structure becomes more complex than a simple correlation between two variables? How are the results affected by changing the sample size? And most importantly, how is one able to apply this fit command to real data?

The next steps for this exercise was to make the R program work a little harder by misspecifying multiple parameters and observe its ability to output something close to the true value. For this example, a multivariable distribution with a Gaussian copula with correlation of 0.8, a standard normal marginal, and an exponential marginal with a rate of 2 was simulated. See Appendix II for the R code used. When all parameter values were given starting values, which were not their true values, the fit was able to recover estimates approximately equal to the true values. However, when it was assumed that no information about the parameters was known (ie, all starting values of 0), an error message was returned indicating the maximization of the likelihood failed. This is an extreme case however, because

as long as the marginals are chosen correctly, the data can be fitted to these marginals, and some information can be retrieved.

The effect of changing the sample size was explored next. In the original exercise, a sample size of 500 was used to fit the distribution. The general rule in statistics is that the bigger the sample size, the better the estimation Since the real data has 48 sites, the sample size in the synthetic example was changed to n=50. The results were similar to that when n=500. The estimates were not quite as accurate as when n=500 and the standard errors of the estimated parameters were larger with n=50. The log likelihood was also much smaller. When n=500, the maximum log likelihood was 614.5, and with n=50 it was 61.4. These are both compared to the true value of 615.7 and 64.7 respectively. For the actual results, see Appendix II. The *fitMvdc* command seems to handle small sample sizes well.

The question about how the fitting method handles more complex dependence structure will be explored in the final section about copulas in higher dimensions. As more variables are added to the data set, the dependence among the variables quickly becomes complicated and the number of parameters increases, and this might prove too much for *fitMvdc* to handle with ease.

Using the maximum log likelihood is a good way to determine which value is the best estimator for the parameter. Another tool that can be useful for understanding the estimation procedure better is graphing the likelihood profile. Since the example of misspecifying the correlation in the bivariate model was used above, the likelihood profiles for the correlation will also be shown. These plots give the likelihood value for each estimate of ρ. This is a good way to check that the

estimated value of ρ given by the fitted model is indeed the best estimate. It is also a way of visualizing the effect that miscalculating the estimate will have, based on how steep the curve is around the best estimate, which provides a measure of uncertainty. The usual data was simulated (Gaussian copula, ρ=0.8, two standard normal marginals) with a sample size of 500. Plotting all the log-likelihood values against possible values for ρ created the log-likelihood profile for ρ. The R code used can be found in Appendix II. Figure 11 shows the resulting plot. In this case, the maximum log-likelihood does in fact occur at ρ =0.8. By the nature of the graph, it seems like underestimating the correlation has less of an effect than overestimating ρ with major changes as it approaches 1. Some of the other trials did however produce maximums that were just off from the 0.8; 0.79 and 0.81 for instance.

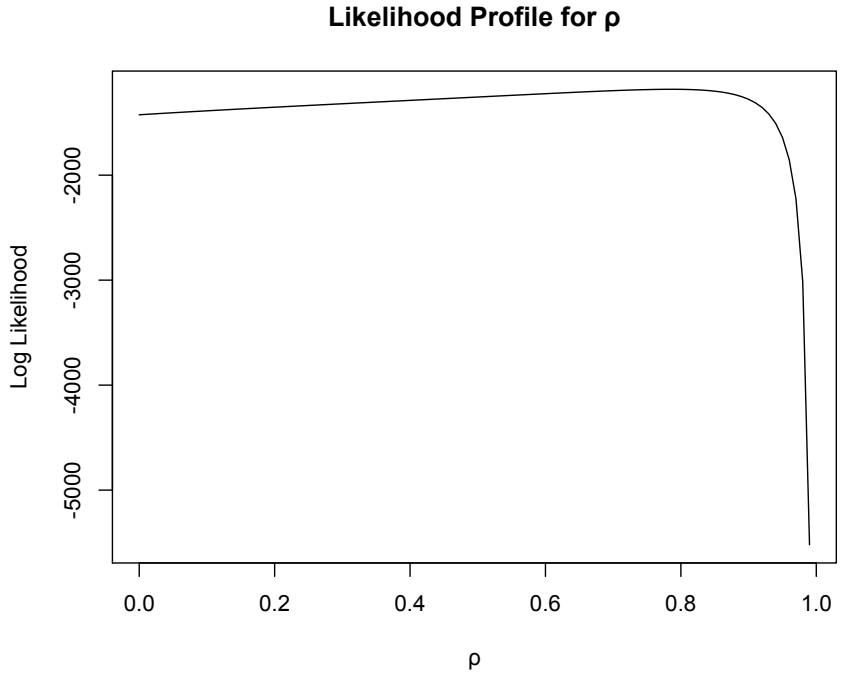


Figure 11: *Log Likelihood Profile for ρ for the bivariate distribution*

With real data, the hope is that we are able to apply this *fitMvdc* command on two variables at least, and extend it to three or more variables, with the desire to fit appropriate distributions to the data. This model fitting procedure was applied to two chosen variables of the benthic data, which was introduced at the beginning. The species number and abundance variables were used as their marginals were already fitted in the section above. The *fitMvdc* command was used to fit a multivariate distribution to this data. First a Gaussian copula was constructed using the correlation between the two variables as the parameter estimate. Then a multivariate distribution with a normal marginal and a gamma marginal was constructed. The maximum likelihood estimates for the marginals that were found using the fitting techniques in the previous section were used as the starting values. The code used for this was straightforward.

```
norm.cop=normalCopula(cor, dim=2)
MVN=mvdc(norm.cop, margins=c("norm","gamma"),
 paramMargins=list(list(mean=0, sd=1), list(shape=2, rate=1)))
start=c(mu1,sd1,shape,rate,cor)
```

The fit command was then used with the above copula and *mvdc* object to fit an appropriate distribution. The code and its output are shown below.

```
fit=fitMvdc(X,MVN,start=start, optim.control = list(trace = TRUE,
maxit = 2000))
 initial  value 170.234910
final  value 169.760046
converged
initial  value 169.760046
final  value 169.760046
stopped after 1 iterations
> fit
The Maximum Likelihood estimation is based on  48  observations.
Margin  1 :
        Estimate Std. Error
m1.mean    2.692      0.167
m1.sd      1.162      0.120
```

```
Margin  2 :
         Estimate Std. Error
m2.shape   1.4618      0.271
m2.rate    0.2553      0.056
Copula:
      Estimate Std. Error
rho.1   0.8794      0.033
The maximized loglikelihood is  -169.76
Optimization converged
Number of loglikelihood evaluations:
function gradient
35        9
```

From this output it can be seen that the mean for the normal distribution was estimated as 2.69 and the standard deviation was 1.16. The shape and rate were found to be 1.46 and 0.26 respectively. The value of the maximum log-likelihood was 169.76 also. This is comparable to the multivariate distribution found using the empirical moments. This is an example of how the *fitMvdc* command can be used with real data to model a multivariate distribution.

7. Higher Dimensions

The majority of the work done here with copulas so far has focused on the bivariate case. This is mainly due to our basic knowledge of copulas and the simplicity that goes along with bivariate distributions. However, it isn't very often multivariate data can be focused in on just two variables. The ability to model copulas with three or more variables is very important. With more variables, the dependence among them can quickly become very complex, and it is with these complex dependencies that copulas are applied.

A small amount of work with copulas in higher dimensions was done in R. The goal of this exercise was to explore copulas in more than two dimensions.

Sticking with the basic normal copula seemed to be the most logical continuation of this introduction into copulas. The copula function was used to generate data from three dimensions with a certain variance-covariance structure. There are several built in correlation matrices in the elliptical copula family in R (Yan, 2006). Some of the more commonly used structures include autoregressive of order 1 (ar1), exchangeable (ex), Toelpitz (toep) and unstructured (un). For the case when p=3, each of the correlation matrices have the corresponding form (Figure 12):

$$\begin{pmatrix} 1 & \rho_1 & \rho_1^2 \\ \rho_1 & 1 & \rho_1 \\ \rho_1^2 & \rho_1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & \rho_1 & \rho_1 \\ \rho_1 & 1 & \rho_1 \\ \rho_1 & \rho_1 & 1 \end{pmatrix}, \begin{pmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_1 \\ \rho_2 & \rho_1 & 1 \end{pmatrix}, \text{ and } \begin{pmatrix} 1 & \rho_1 & \rho_2 \\ \rho_1 & 1 & \rho_3 \\ \rho_2 & \rho_3 & 1 \end{pmatrix},$$

Figure 12: *Built-in dispersion structures in R, retrieved from Yan (2006).*

The choice of correlation structure quite obviously depends on the data you are working with. It is noted that the sample generated from the trivariate normal copula produces, not surprisingly, uniform marginals and a sample correlation structure similar to the dispersion parameter values chosen to produce the copula. In this exercise, an exchangeable correlation structure was chosen with ρ=0.8. The 'pairs' command was also used to plot every variable against all the others (as seen in Figure 13) to visually see the dependence between specific pairs of variables (the importance and effectiveness of scatterplot was previously mentioned). All three variables had similar scatterplots with one another, as was to be expected with the chosen structure.

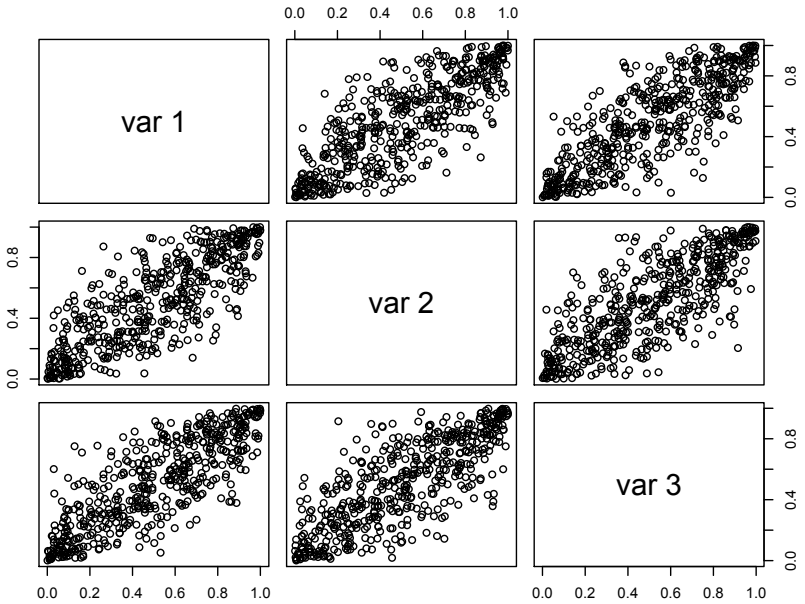


Figure 13: *pairs plot of the three variables from the trivariate copula created in R. With the exchangeable dependence structure, all variables are correlated with a chosen correlation of 0.8.*

A multivariate distribution was also created with these higher dimensions, in the very same way the bivariate distributions were generated. The copula and marginals were manually controlled, each individual pairs plot and histogram for the data was plotted. The pair plots were used to observe the dependence among the variables, and the histograms gave information about the marginals. Figure 14 shows the scatterplots and individual histograms for this multivariate distribution.

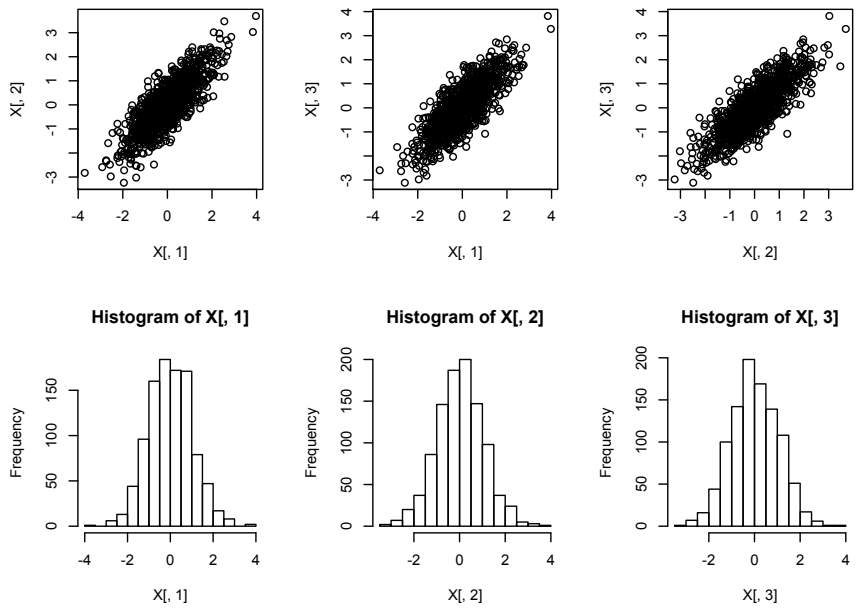


Figure 14: *Marginal distribution of the simulated trivariate distribution*

Defining all normal marginals in the multivariate distributions command, all the histograms resembled normal distribution curves. With more than three variables, the ability to visualize the dependence between each variable becomes increasingly difficult since with every variable added, the number of comparisons amongst variables increases quickly. The number of comparisons is $\binom{n}{2}$, where n is the number of variables, so as n increases, so do the number of comparisons. The plotting power beyond something that is 3D is difficult. To effectively visualize dependence relationships that are four or five dimensions is almost impossible. Normal copulas with all the different correlation structures mentioned were also generated. When all the pair plots are compared to one another, the differing

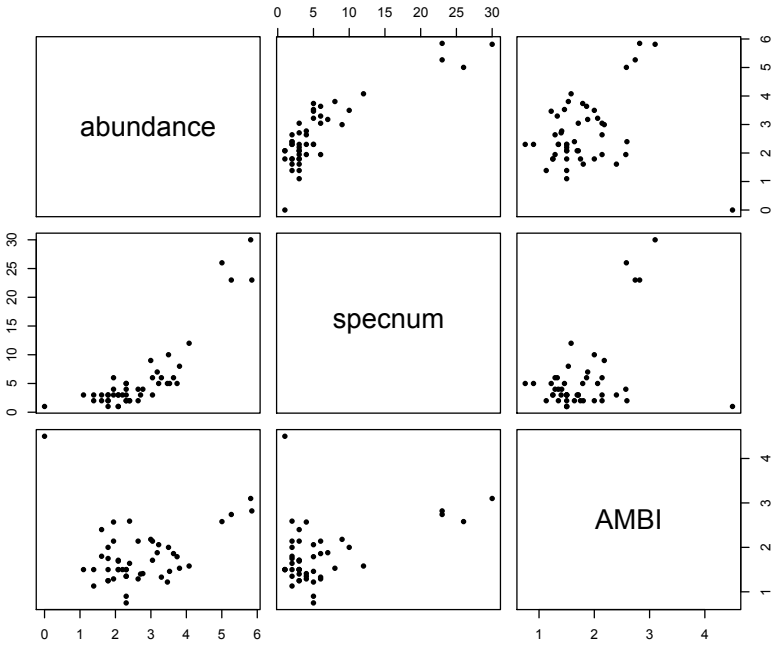
dependence structures can be recognized. For example, the correlation between variables one and three in the ar1 structure is noticeably different than the correlation between the same variables in 'unstructured' dependence structure. The other three pairs plots can be found in Appendix III. One can see how quickly things can become complex. All four of these different dependence structures were created without even changing the copula (all distributions contained a Gaussian copula). The dependence structure cannot only vary within the copulas but between different copulas and different families of copulas as well. The dependence structure for a trivariate distribution with three normal standard marginals but a Frank Archimedean copula joining the marginals together would look different still. The ultimate goal would be to define the proper copula to data with several variates in order to model the distribution properly.

7.1 Fitting a trivariate distribution with Benthic data

Just as the fit procedure was applied to the benthic data in the previous section, three selected variables were chosen from the benthic data with the ultimate goal of modeling the dependence properly. With real data, it seems as though the 'unstructured' dependence structure would be the most commonly used, as it required one to define all possible values for the correlation among the variables. Unless it is glaringly obvious the dependence structure follows one of the given forms, all of the sample correlation estimated would have to be specified individually in order to create the copula. These values are easy to estimate as they are retrieved from the sample variance-covariance matrix. Along with the abundance and species number variable used in the previous sections, the AMBI

variable was also chosen to apply this fitting method to higher dimensions. Figure 15a shows the pairs plot for the three chosen variables.

(a)



(b)

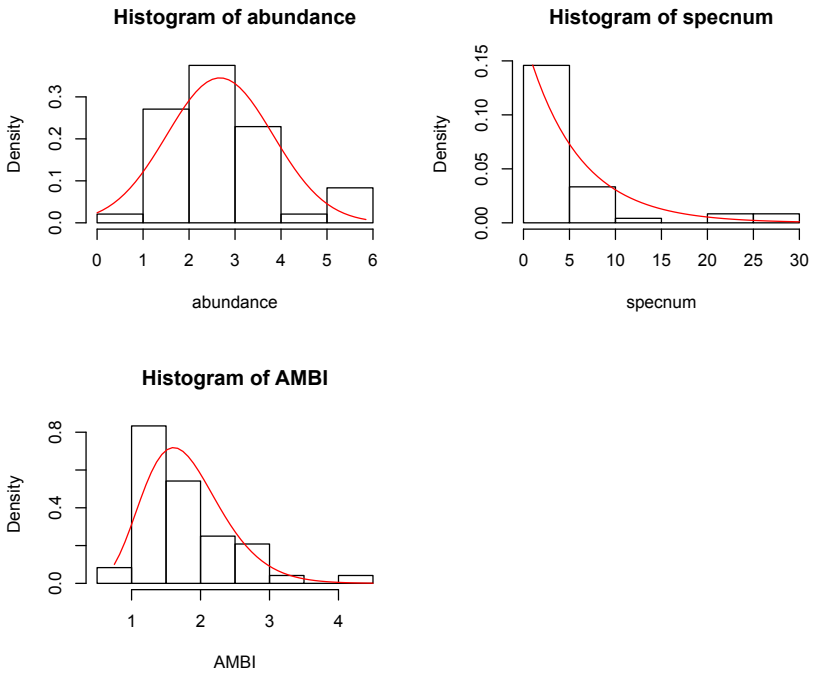


Figure 15: *(a)Pairwise scatterplot for the variables abundance, species number, and AMBI and (b) All three marginals with fitted curves*

For this example, the species number marginal was actually modeled as an exponential distribution and the AMBI variate was modeled as a gamma distribution. This gives three variables with three different marginals. Figure 10b shows the fitted curved overlaid on the marginal histograms for all three marginal distributions. For this example, again a Gaussian copula was used to connect the three variables together. As mentioned above, the unstructured dispersion structure was used in the *copula* object, with the three parameters being the values of the sample correlations between each of the variables. The following code shows how this was accomplished.

```
X=cbind(abundance,specnum,AMBI)
pairs(X, pch=20)
cor=cor(X)
rho1=cor[1,2]
rho2=cor[1,3]
rho3=cor[2,3]

norm.cop=normalCopula(param=c(rho1,rho2,rho3), dim=3, dispstr="un")
MVN=mvdc(norm.cop, margins=c("norm","exp","gamma"),
paramMargins=list(list(mean=0,sd=1),list(0.25),list(shape=10,rate=5
)))
```

The *fitMvdc* command was then used to fit the multivariate distribution with three dimensions. The maximum likelihood parameter estimates were again used as starting values. The fit had the following output;

```
start=c(mu1,sd1,lambda,shape,rate,rho1,rho2,rho3)
> fit=fitMvdc(X,MVN, start=start, optim.control = list(trace =
TRUE, maxit = 2000))
initial  value 215.684492
iter  10 value 209.373219
final  value 209.332519
converged
```

```
initial  value 209.332519
final  value 209.332519
stopped after 1 iterations
> fit
The Maximum Likelihood estimation is based on  48  observations.
Margin  1 :
        Estimate Std. Error
m1.mean    2.548      0.177
m1.sd      1.358      0.107
Margin  2 :
        Estimate Std. Error
m2.rate   0.1762      0.025
Margin  3 :
         Estimate Std. Error
m3.shape    9.073      1.921
m3.rate     5.101      1.100
Copula:
      Estimate Std. Error
rho.1   0.9132      0.021
rho.2   0.3105      0.146
rho.3   0.3740      0.145
The maximized loglikelihood is  -209.3325
Optimization converged
Number of loglikelihood evaluations:
function gradient
      56       15
```

All parameter estimates correspond to those found using the moments method, as can be seen in the output above. There was a desire to create a contour plot of the fitted multivariate distribution to compare to the scatterplot of the real data to ensure the dependence was modeled properly. As contours become more difficult to plot with increasing dimensions, 500 data points were instead generated from the fitted multivariate distribution and its pairwise scatterplot is shown in Figure 16. The scatterplot from the generated data has definite similarities in each of the pairwise comparisons to the scatterplot of the real data. For instance, for abundance and species number, both the generated data and real data steeply increase at the

beginning and then taper off near the higher limits. The shape for all three comparisons are relatively consistent.

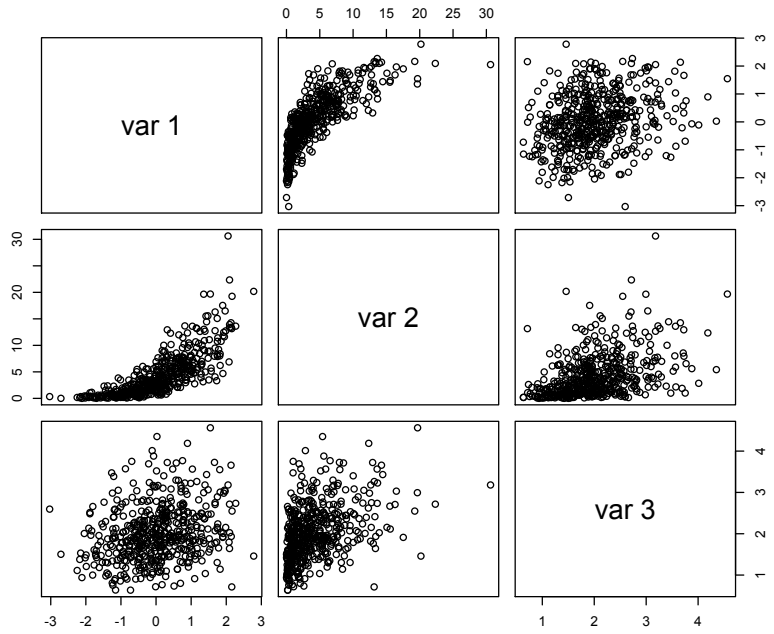


Figure 16: *Pairwaise scatterplot of the 500 data points generated from the fitted trivariate distribution*

As more dimensions are added, the complexity of the problem and the model drastically increases. Much more work would have to be done to fit even higher dimension models.

8. Conclusion

Copula theory is relatively new to the field of statistics, but thanks to economics and finance, they are quickly growing in popularity. In fact, copulas were a huge success on Wall Street, and at the same time are also being blamed for a part of the collapse (Salmon, 2009). Although the theory is relatively new, it is quite complex and in depth. There are two different families of copulas, each with their

own variety of functions and dependence structures. Elliptical copulas are used where elliptical distributions are involved, while Archimedean copulas are popular because they support much more complex forms and skewness. Between these two families, numerous multivariate distributions can be defined to model data.

Sklar's theory is instrumental in describing and understanding the basis of these functions. It explains that multivariate distributions can be completely explained with their individual marginals and the copulas that joins the random variables together. This idea was the basis for being able to fit multivariate distributions through empirical moments of the marginals and correlation. This was supported with exercises in R where distributions were simulated and fitted with both the moment method and the built-in R functions, with comparative results. Moving on to higher dimensions, the copulas and the multivariate distributions become much more complex. Even though trivariate distributions were experimented with this is an area that has been left for further discovery in this thesis.

Researching the theory on copulas is a difficult but yet rewarding task. The theory and computations quickly become very detailed. With the desire to model complex multivariate distributions, it is no surprise there is a growing interest in this topic. This thesis is a good stepping stone to the much larger world of copulas.

9. References

Arnold, H. (2006). *Dependence Modelling via the Copula Method* [PDF document]. Retrieved from ethernet-switch.googlecode.com/files/copula_project.pdf

de Matteis, R. (2001). *Fitting Copulas to Data* (Diploma Thesis). Institute of Mathematics of the University of Zurich, Zurich, CH.

Dowd, M. et al. (2013). *Spatial Aspects of Assessment and Sampling Design for Coastal Benthic Monitoring* [submitted manuscript]. Dalhousie University, Halifax, NS.

Frees, E.W., Valdez, E.A. (1998). Understanding Relationships using Copuals. *North American Actuarial Journal 2(1).* 1-25. doi: 10.1080/10920277.1998.10595667

Genest, C., Favre, A. (2007). Everything You Always Wanted to Know About Copula Modelling but were Afraid to Ask. *Journal of Hydrologic Engineering 12(4).* 347-368. doi: 10.1061/(ASCE)1084-0699(2007)12:4(347)

Genest, C., MacKay, J. (1986). The Joy of Copulas: Bivariate Distributions with Uniform Marginals. *The American Statistician 40(4).* 280-283.

Kang, L. (2007). *Modeling the Dependence Structure between Bonds and Stocks: A Multidimensional Copula Approach*. Indiana University Bloomington, Indiana, US.

Lebo, M. (2005) *Multivariate Probability Distributions* [PDF document]. Retrieved from http://ms.cc.sunysb.edu/~mlebo/Multivariate_Probability_Distributions.pdf

Nelson, R.B. (2006). *An Introduction to Copulas.* New York, NY: Springer.

Salmon, F. (2009) Recipe for Disaster; The Formula that Killed Wall Street [news article]. Retrieved from http://www.wired.com/techbiz/it/magazine/17-03/wp_quant?currentPage=all

Schmidt, T. (2006). Coping With Copulas In J. Rank (Ed.), *Copulas: From Theory to Application in Finance* (pp. 3-34). London: Donaldson.

Yan, J. (2006). Enjoy the Joys of Copulas. *Journal of Statistical Software 21(4).*

Appendix I – R code from "Empirical Copula"

```
>library(copula)
library(mvtnorm)
library(sn)
library(scatterplot3d)
library(MASS)

set.seed(1)  #to replicate the results

sigma=matrix(c(1,0,0,1), byrow=T, nrow=2)  #covariance matrix with 0
 correlation between X and Y
mu=c(0,0)  #zero mean for X and Y

data=mvrnorm(n=10, mu, sigma)
x=data[,1]
y=data[,2]
n=length(x)
o=order(x)
data1=rbind(x[o], y[o])
x=data1[1,]
y=data1[2,]
plot(x,y, pch=20)  #scatterplot of the pairs (Xi,Yi)

z=exp(x)
t=exp(3*y)
plot(z,t, pch=20) #scatterplot of the pairs (Zi,Ti)

par(mfrow=c(2,1))
plot(x,y, pch=20)
plot(z,t, pch=20)

##Ranks of X and Y
R=rank(x)
S=rank(y)

rho=((12/(n*(n+1)*(n-1)))*(sum(R*S)))-(3*(n+1)/(n-1))
rho; cor(x,y)
sqrt(n-1)*abs(rho)

############################################################

set.seed(1)

par(mfrow=c(1,1))

sigma=matrix(c(1,0,0,1), byrow=T, nrow=2)
mu=c(0,0)
```

```
data=mvrnorm(n=1000, mu, sigma)
x=data[,1]
y=data[,2]
n=length(x)
o=order(x)
data1=rbind(x[o], y[o])
x=data1[1,]
y=data1[2,]
plot(x,y, pch=19)

##Ranks of X and Y
R=rank(x)
S=rank(y)

rho=((12/(n*(n+1)*(n-1)))*(sum(R*S)))-(3*(n+1)/(n-1))
rho; cor(x,y)
sqrt(n-1)*abs(rho)
```

Appendix II

Misspecifying multiple parameter estimates

```
norm.cop <- normalCopula(0.8, dim =2)
myD <- mvdc(norm.cop, margins = c("norm", "exp"), paramMargins =
 list(list(mean = 0, sd = 1), list(rate=2)))
contour(myD, dmvdc, xlim = c(-3, 3), ylim = c(-3, 3))
X=rmvdc(myD,500)
plot(X[,1], X[,2])
hist(X[,1])
hist(X[,2])

start=c(0.1,0.89,1.74,0.5)

fit=fitMvdc(X,myD, start=start, optim.control = list(trace = TRUE,
 maxit = 2000))
   initial  value 657.100272
   iter  10 value 578.531779
   final  value 578.531616
   converged
   initial  value 578.531616
   final  value 578.531607
   stopped after 2 iterations
fit

    The Maximum Likelihood estimation is based on  500  observations.
```

```
Margin  1 :
        Estimate Std. Error
m1.mean -0.07074      0.041
m1.sd    0.96803      0.027
Margin  2 :
        Estimate Std. Error
m2.rate    2.162      0.097
Copula:
      Estimate Std. Error
rho.1    0.774      0.016
The maximized loglikelihood is  -578.5316
Optimization converged
Number of loglikelihood evaluations:
function gradient
      52       12
```

Changing the sample size

For n=50, the fitted model is as follows:

```
> fit=fitMvdc(X,myD, start=start, optim.control = list(trace =
TRUE, maxit = 2000))
initial  value 75.574811
iter  10 value 61.406876
final  value 61.406866
converged
initial  value 61.406866
final  value 61.406866
stopped after 1 iterations
> fit
The Maximum Likelihood estimation is based on  50  observations.
Margin  1 :
        Estimate Std. Error
m1.mean   0.1889      0.120
m1.sd     0.9127      0.076
Margin  2 :
        Estimate Std. Error
m2.rate    1.628       0.23
Copula:
      Estimate Std. Error
rho.1   0.8404      0.037
The maximized loglikelihood is  -61.40687
Optimization converged
Number of loglikelihood evaluations:
function gradient
      41       12
```

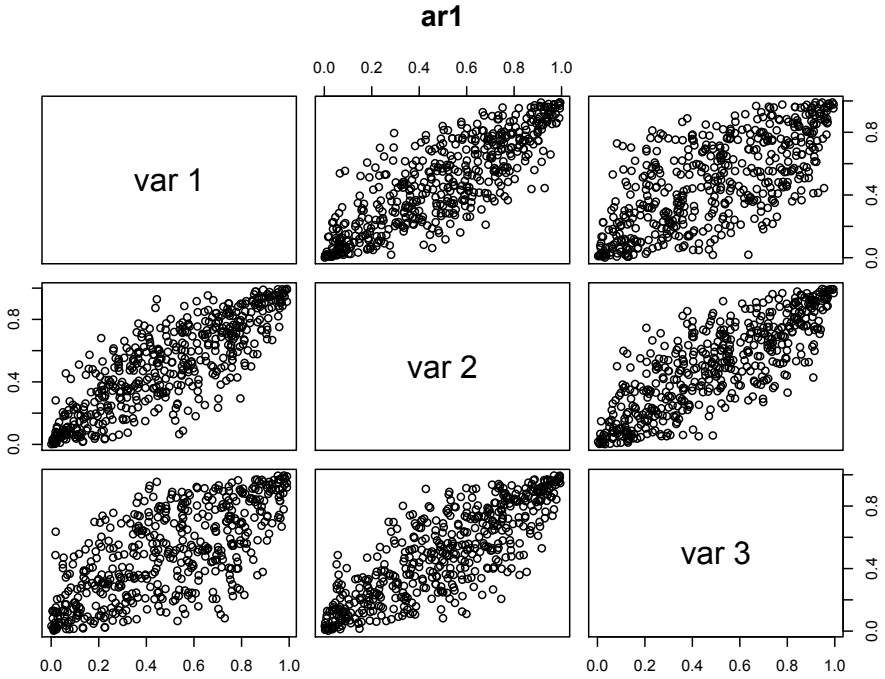
For n=500, the fitted model returned

```
> fit=fitMvdc(X,myD, start=start, optim.control = list(trace =
TRUE, maxit = 2000))
initial  value 708.936880
iter  10 value 614.516442
final  value 614.514495
converged
initial  value 614.514495
final  value 614.514495
stopped after 1 iterations
> fit
The Maximum Likelihood estimation is based on  500  observations.
Margin  1 :
        Estimate Std. Error
m1.mean 0.004734      0.043
m1.sd   1.022908      0.026
Margin  2 :
        Estimate Std. Error
m2.rate     1.94      0.086
Copula:
      Estimate Std. Error
rho.1   0.8185      0.013
The maximized loglikelihood is  -614.5145
Optimization converged
Number of loglikelihood evaluations:
function gradient
      64       15
```

Appendix III – differing dependence structures

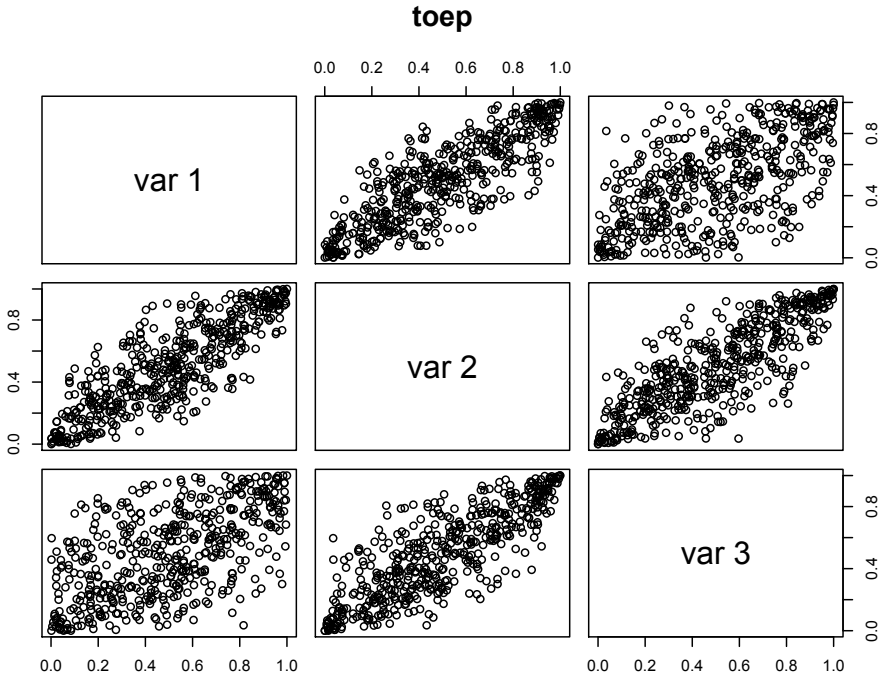
Auto-regressive 1:

```
norm.cop=normalCopula(0.85, dim=3, dispstr="ar1")
Xn=rcopula(norm.cop, 500)
pairs(Xn, main="ar1")
cor(Xn)

[1,] 1.0000000 0.8393743 0.6957665
[2,] 0.8393743 1.0000000 0.8245148
[3,] 0.6957665 0.8245148 1.0000000
```



Toelpitz:

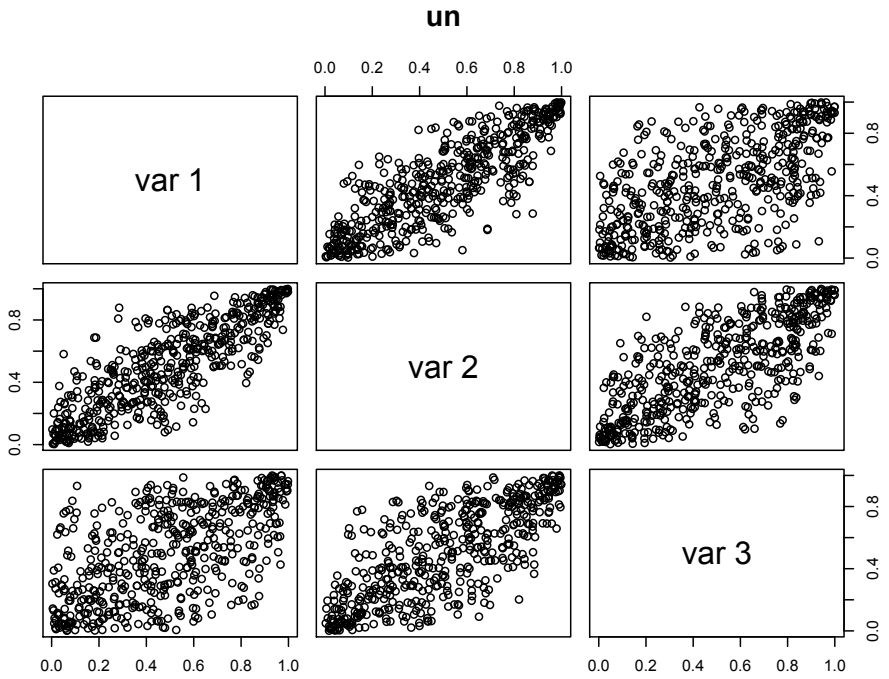
```
norm.cop=normalCopula(param=c(0.85,0.6), dim=3, dispstr="toep")
Xn=rcopula(norm.cop, 500)
pairs(Xn, main="toep")
cor(Xn)

          [,1]      [,2]      [,3]
[1,] 1.0000000 0.8432308 0.5863669
[2,] 0.8432308 1.0000000 0.8261980
[3,] 0.5863669 0.8261980 1.0000000
```



Unstructured:

```
norm.cop=normalCopula(param=c(0.85,0.6,0.75), dim=3, dispstr="un")
Xn=rcopula(norm.cop, 500)
pairs(Xn, main="un")
cor(Xn)
```

```
          [,1]      [,2]      [,3]
[1,] 1.0000000 0.8395342 0.5998914
[2,] 0.8395342 1.0000000 0.7502231
[3,] 0.5998914 0.7502231 1.0000000
```


un
var 1
var 2
var 3
0.0
0.2
0.4
0.6
0.8
1.0