INFO304 Assignment 3 2022

Ethan Smith - 5652106

QUESTION ONE (15 MARKS)

Assume that you do not know the K-means clustering algorithm, but want to build a clustering tool that finds K cluster centres given a table of numeric data with F features (i.e. F explanatories, but no response). You decide to use a genetic algorithm (GA) to evolve the solution for the cluster centres.

1. Given some K value (number of clusters to evolve) and number of features F of the data, describe an appropriate representation for an individual.

I believe the most appropriate representation of an individual in this scenario would be a fixed length vector of length F*K, this vector could also be interpreted as a nested list with K lists involved each initialized with F values, which mark a point within the axes of the data to be clustered. Because clustering is typically a distance based algorithm, both the points to be clustered and the chromosome vector should be standardized values (Mean = 0, SD = 1). This is important as it reduces the risk of a difference in scale between features biasing the location of the cluster centers.

(F1, . . . , FN) * K -> all values with Mean = 0, SD = 1

2. What is the fitness function for this problem? That is, what makes a good solution to a clustering algorithm, and how is this evaluated? Assume that you have a data table of N rows and F numeric feature variables.

For this problem I would be most likely to use the total euclidean distance of all points in the cluster to the center as a fitness function as we want all observations within a cluster to be close to the set cluster point. This would be the sum of the individual euclidean distances for the subset of N observations closest to a given point K. Any good clustering algorithm should be able to separate observations into clearly defined groups based on their similarity. A indicator of good clustering will be if the clusters have a low variance within the cluster but the variation between clusters is high. This means that the clustering algorithm has been successful in finding separation in the features. Depending on the data this could possibly be a more reliable fitness function, this would most likely happen in cases when the data cannot be linearly separated.

QUESTION TWO (25 MARKS)

You are tasked with using a k-nearest neighbour (kNN) model to do prediction, but the dataset you are using has a large number of explanatories. Assume you have N explanatory variables and a response Y.

Rather than doing more traditional feature reduction/selection methods, you decide to use a Genetic Algorithm (GA) to search for the best M features (M <= N) to optimise the predictive behaviour of the kNN model for a FIXED value of k.

Describe how you would set up this combined GA and kNN model to search for the "best" kNN model for a specific dataset and set value of k. Ensure you include:

1. A representation (and how you would use the representation) for the GA and kNN;
2. The fitness function and example search operators (crossover, mutation);

To perform feature selection for a model using a genetic algorithm I would represent each individual in the population as a fixed length binary vector. Each index in this vector will correspond to an explanatory variable where a one represents the predictor being included in the model and a zero represents the predictor being excluded from the model. The data being modeled should be split into training and testing data. The model can then be built from all predictors the chromosome set to a value of one. I also believe the train test splits should be randomly generated for each individual model to ensure a significant predictor remains significant across different training sets. The fitness function for each individual will be the testing set error where a smaller value is considered a higher fitness.

For crossover and mutation I would be most likely to use a uniform crossover as each index of the chromosome only has two possible values. This works by generating each index of the child's chromosome by passing the value at that index of one parent by chance (generally 50/50 chance for each parent is used). In this case the uniform crossover could also contain a slight bias involved to give significant predictors in a model a higher chance of being kept if the other parent's model does not contain that predictor. I also believe in this case for mutation it makes sense to select a random index from the chromosome vector and flip the value. This would allow mutated chromosomes to remain similar to the parents but still retain some diversity in the population as the generations progress. In this case I also believe it would be appropriate to factor in some elitism by keeping the best performing model from one generation in the next generation to make sure a potentially optimal model is not lost.

Now assume that you also want to optimise the number of neighbours k.

3. Define a representation that would allow you to optimise both the value of "k" and the M features during the evolution. Ensure you explain how the representation is interpreted and used within the model.

To represent K in the chromosome I established above I would add an extra index in the vector to store a value of k. For the predictors themselves I would opt to keep the same methods for crossover and mutation, but for the value of k I believe it would be best to form a new value of K from the parents, this value would lie within the range of the two parent's value for K and could be calculated based on the fitness scores of the parents to correspond closer to the higher performing parent. To main a stochastic element I would introduce a form of mutation where the value of K would be modified to become higher or lower by 10% of it's current value. This would allow for some diversity of K to remain so that different models so that potentially optimal values for K that have been excluded from the range of the population can continue to be trialed within a small subset of the population, but this should result in some value of K that is optimal given the models generated through the generations. To maintain validity and a reasonable search space I would include constraints to ensure the value of K is initialized and always lies within one and the square root of N, where N is the number of observations in the training data set.

QUESTION THREE (60 MARKS) - Investment Portfolio Management

This section deals with modelling the selection of stocks, bonds and cash to make up an investment portfolio. Load the data as follows: invest <- read.table("invest.tab")

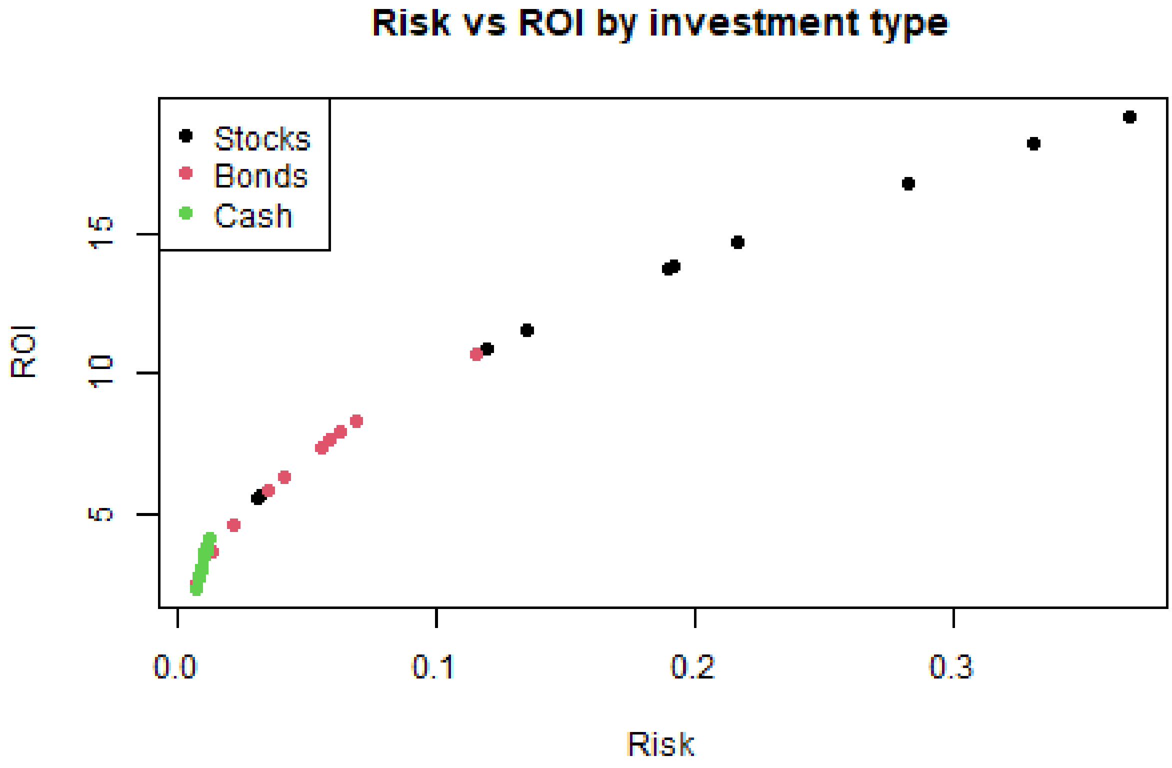
The ROI column is the percentage predicted return on investment, the Risk column is a measure of the risk associated with this particular investment, and the Type indicates the type of investment. Note that each row is labelled with the type of investment and a number, so that you can (if needed) refer to individual investments.

1. Visualise and discuss the different ROI and Risk associated with Stocks, Bonds and Cash. In addition, load, visualise and give a simple interpretation for the correlation table (HINT: See part 5 for details of "corr.tab").

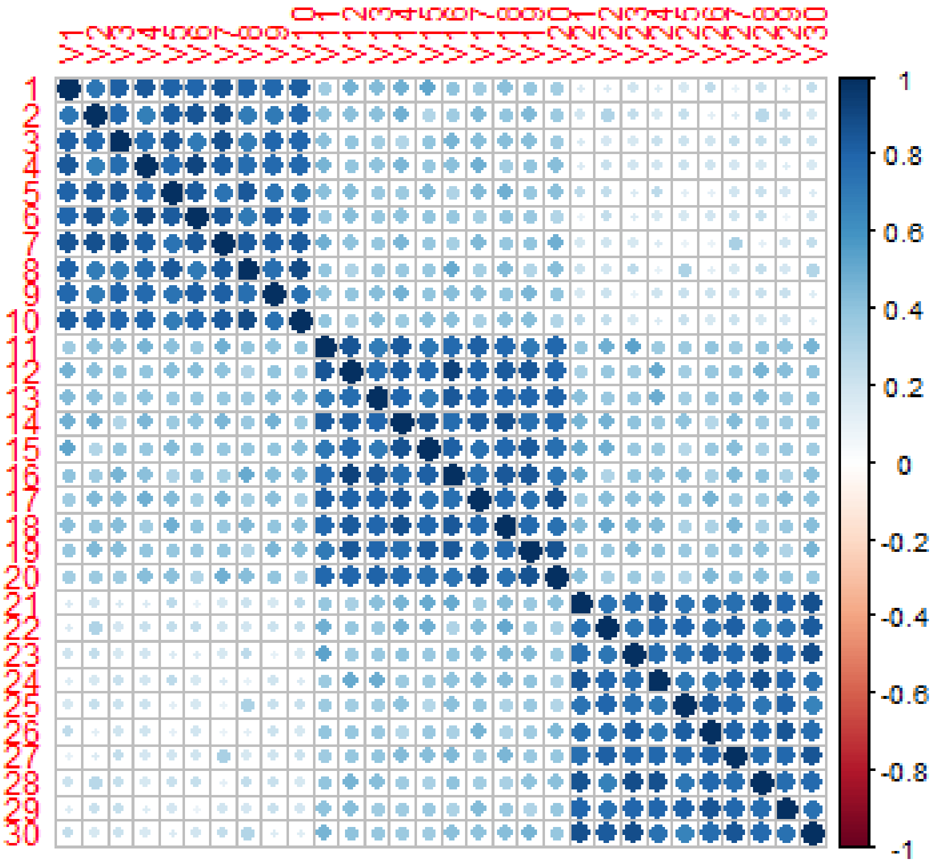
```
library(corrplot)
```

```
## corrplot 0.92 loaded
```

```
invest <- read.table("invest.tab")
invest.corr <- read.table("corr.tab")
plot(invest$Risk, invest$ROI, col = invest$Type, main = "Risk vs ROI by investment type",
     xlab = "Risk", ylab = "ROI", pch = 16)
legend("topleft", legend=c("Stocks", "Bonds", "Cash"),
       col=c(1, 2, 3), pch = 16)
```



```
corrplot(as.matrix(invest.corr))
```

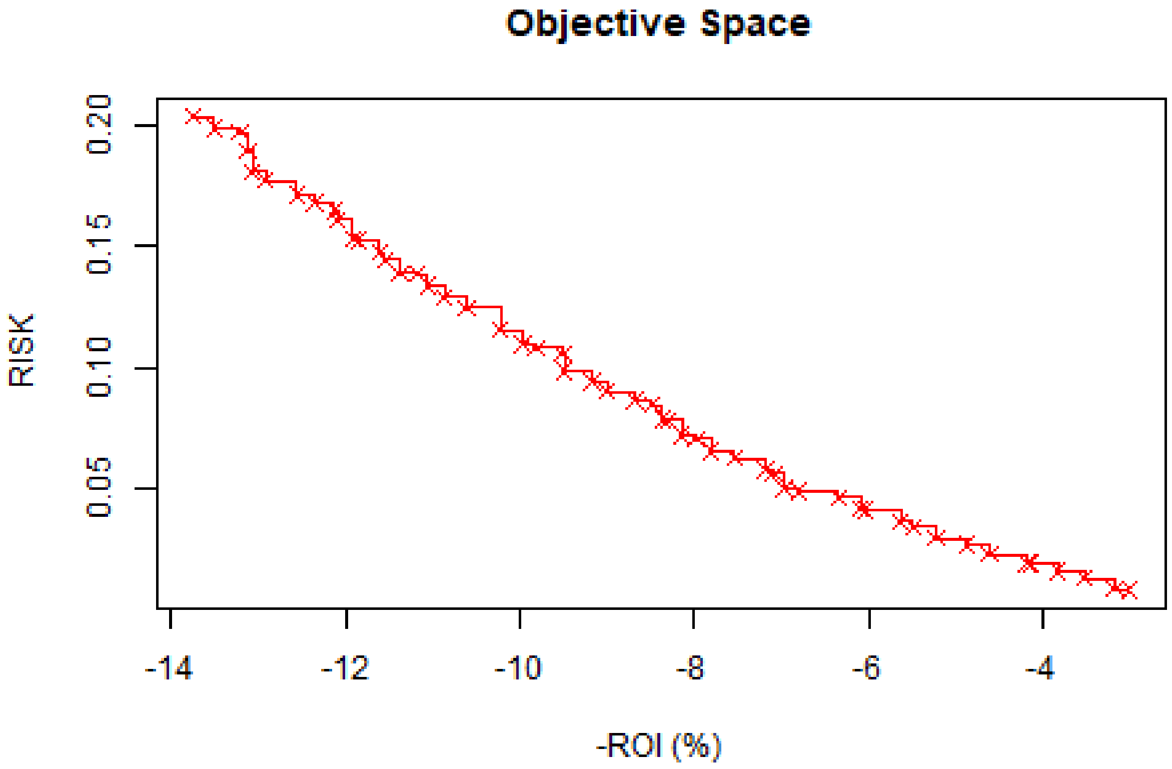


From these plots a clear relationship can be seen between the type of investment made to the resulting risk and return. There is a clearly defined relationship between risk and return as well, the larger the risk is on an investment the larger the potential return is as well. This also results in a higher risk investment having a larger potential loss as well. From the scatter plot it can be shown that cash investments are the safest to make but provide a very low return while investments made through stocks can lead to a potentially much higher return at a greater risk of losing money. Investments made through bonds appear to achieve a fair balance between the two.

The correlation plot of each investment option validates what the scatter plot is showing. Colinearity is very high between investments of the same. Because of this if multiple investments of the same type of made there will be a lack of diversity within that investment portfolio which could potentially lead to a large loss if the investments begin to lose money. The plot also shows that there is little to no relationship between the behaviour of cash and stock investments, while bonds once again appear to have a fairly balenced relationship with both stocks and cash.

2. Run the "invest.R" script. This script does a multi-objective criteria analysis to determine the best mix of stocks, bonds and cash over a range of tradeoffs. Describe in words what the "invest.R" script is doing. Ensure that you address how a solution is represented and interpreted, and the relationship between the solution space, the objective space and the constraints.

```
set.seed(5)
source("invest.R")
```



```
par <- as.data.frame(head(portfolio$par, 2))
```

The invest.R script is where a multi objective optimisation of investment portfolios is performed by a genetic algorithm using the mco and nsga2 libraries. In this implementation 52 different investment portfolios are developed for 500 generations of the genetic algorithm. The feature space is represented by the chromosomes of 52 individuals in the model. This chromosome is a vector of length 30 as we have 30 possible investments to select (input dimensions = 30), where each index holds a value between 0 and 0.2 representing the proportion of the investment portfolio that singular investment makes up.

To find the solution space for a generation, the business rule constraints must be applied to each indivdual to ensure each possible portfolio is valid. For this implementation three business rules have been applied through model constraints. The number of investments made in each portfolio must lie between 8-12 and each investment must make up 5-20% of the overall portfolio, the maximum value of a single investments weight (20%) is set in the upper bounds of each generated individual to ensure no total investment proportion should exceed 1. Along with these the sum of all contributions must make up at least 95% of the total amount that could possibly be invested. A check is made to ensure the sum of proportions does not exceed 1 but this should not occur as it is only possible for an indvidual investment to be excluded for being too small after each individual is generated.

These constraints allow us to validate that a given portfolio is valid and is also diverse to reduce the risk of a loss if some of the indiviudal investments begin to lose money. Any portfolio that meets these criteria is said to be within the solution space as it meets the criteria for a valid portfolio so it can be considered when optimising the search space for the best portfolios. From here dominance tests can be perfomed across all individuals that reside within the search space. A possible solution A is said to dominate another possible solution B, if for any given objective A is a more optimal solution than B and for any other objective it cannot be proven that A is worse than B. For example in this scenario A may dominate B if A provides a greater return on investment than B but it cannot be proved that A is any worse than B on risk. Any solution which is not dominated by any other solution is said to lie in the objective space (plotted above). All points that lie in the objective space form the pareto front. Any point on this front is considered an equally optimal solution, they only differ on their levels of risk and return based on the investments made. The plot shows the investments representation of the two output dimensions of risk and return (the values we wish to optimise).

The script returns a portfolio object which contains three components: par, value and pareto.optimal. Par shows the investment proportions for each indvidual (Chromosome values / solution space), value is the value for each output dimension (roi, risk) for each individual and pareto.optimal is a vector of boolean values stating whether each solution returned lies within the objective space / pareto front, values that are true are plotted in output.

3. Table 1 (below) shows the recommended blend of stocks, bonds and cash for a number of different brokerage houses (i.e. businesses that take your money and invest it to give you a return).

Using the result of the nsga2 model you have previously run, examine and present the blend of stocks, bonds and cash for a low risk, moderate risk and high risk investment blend (just pick one from each general category). Discuss, in relation to Table 1, the level of risk that seems to be taken by the brokerage houses and whether the one year return performance is related to the associated risk of the brokerage house.

```
results <- as.data.frame(portfolio$value)
colnames(results) <- c("roi", "risk")
blends <- portfolio$par
blends[which(blends < 0.05)] = 0
blends[which(blends > 0.20)] = 0
blends <- as.data.frame(blends)
results <- cbind(results, blends)
results <- results %>% arrange(risk)
results <- results[c(5, 25, 45),]
results <- results %>% rowwise() %>% mutate(Stocks= sum(c_across(3:12))) %>%
  rowwise() %>% mutate(Bonds= sum(c_across(13:22))) %>%
  rowwise() %>% mutate(Cash= sum(c_across(23:32))) %>%
  rowwise() %>% mutate(Total= sum(c_across(3:32))) %>%
  mutate(roi = -roi) %>%
  select(-c(3:32))
results[,c(3:6)] <- results[,c(3:6)] * 100
results[,c(3:6)] <- round(results[,c(3:6)],2)
results
```

```
## # A tibble: 3 x 6
## # Rowwise:
##     roi   risk Stocks Bonds  Cash Total
##   <dbl>  <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1  4.15 0.0194   10.1  23.4 62.4   95.9
## 2  8.49 0.0847   42.2  48.3  5.45  96.0
## 3 12.4  0.168    78.2  19.6  0     97.7
```

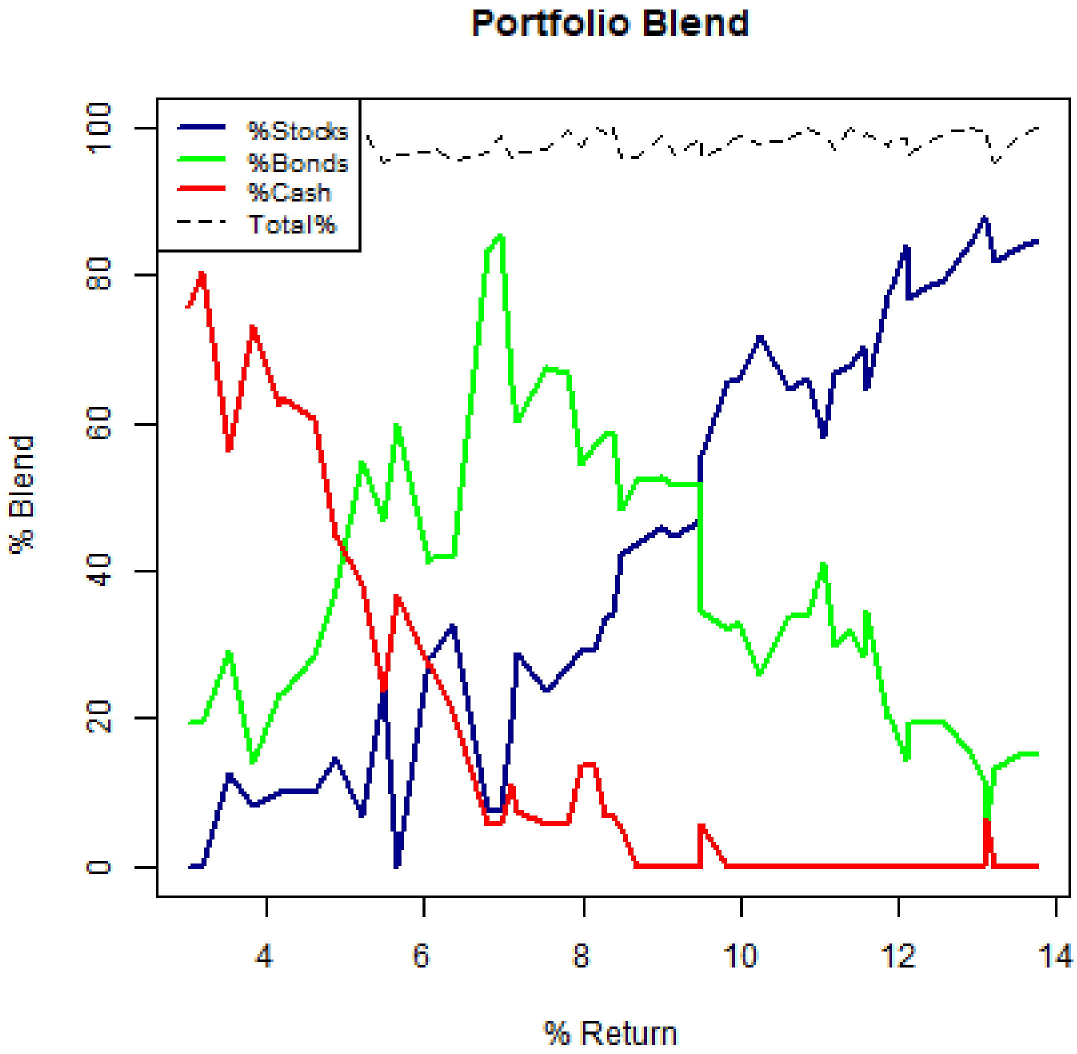
Here I have presented examples of a low, medium and high risk portfolio ordered by the level of risk. Comparing these examples to those used by different brokerage houses in Table 1 I believe brokerage houses tend to opt for high risk portfolios due to the stocks typically making up the largest proportion of the portfolios out of the three investment types available. The one year return on these portfolios is almost most similar to the return on the high risk portfolios generated from invest.R through nsga2. The only brokerage house I believe could possibly be associated with having a medium risk portfolio is Merill Lynch due to bonds which have been previously defined as achieving a good balance between risk and return having the largest share of the portfolio. This portfolio still borders with being considered a high risk portfolio due to the large amount of investment of stocks being made.

The high one year returns on these investment portfolios seem to align with a high risk when compared with the portfolios generated through nsga2, as across both tables a high investment in the stock market appears to give much higher potential returns so I believe it is appropriate to assume that the brokerage house portfolios will also share the high risk associated with similiar portfolios from nsga2.

4. Examine the plot shown in Figure 1. This shows how the percentage of bonds, stocks and cash vary as you move along the pareto front from the least to greatest percentage return. Write an R script to produce this figure given the output from nsga2. Submit your R code and the associated figure.

```r
tab1 <- as.data.frame(portfolio$value)
colnames(tab1) <- c("roi", "risk")
blends <- portfolio$par
blends[which(blends < 0.05)] = 0
blends[which(blends > 0.20)] = 0
blends <- as.data.frame(blends)
tab1 <- cbind(tab1, blends)
tab1 <- tab1[which(portfolio$pareto.optimal),]
tab1 <- tab1 %>% arrange(desc(roi)) %>%
  rowwise() %>% mutate(Stocks= sum(c_across(3:12))) %>%
  rowwise() %>% mutate(Bonds= sum(c_across(13:22))) %>%
  rowwise() %>% mutate(Cash= sum(c_across(23:32))) %>%
  rowwise() %>% mutate(Total= sum(c_across(3:32))) %>%
  mutate(roi = -roi) %>%
  select(-c(3:32))
tab1[,c(3:6)] <- tab1[,c(3:6)] * 100
tab1[,c(3:6)] <- round(tab1[,c(3:6)],2)

plot(tab1$roi, tab1$Stocks, type = "l", ylim = c(0,100), col = "darkblue",
     main = "Portfolio Blend", ylab = "% Blend", xlab = "% Return", lwd=2.0)
lines(tab1$roi, tab1$Bonds, col = "green", lwd=2.0)
lines(tab1$roi, tab1$Cash, col = "red", lwd=2.0)
lines(tab1$roi, tab1$Total, lty = "dashed")
legend("topleft", legend=c("%Stocks", "%Bonds", "%Cash", "Total%"),
       col=c("darkblue", "green", "red", 1), lwd = c(2,2,2,1),
       lty = c("solid", "solid", "solid", "dashed"), cex = 0.8)
```


Portfolio Blend
%Stocks
%Bonds
%Cash
Total%
% Blend
0
20
40
60
80
100
4
6
8
10
12
14
% Return

5. A matrix representing the estimated correlation between any two investments is given in the table corr.tab. For our example with 30 investments this is a 30x30 table. A good portfolio should aim for investments where correlation of behaviour is minimised, so that if some of the investments are decreasing, others in the portfolio may behave differently. This reduces the risk of the portfolio as a whole losing money and should result in more stable returns.

Extend the model in invest.R to have an additional objective which is to minimise the correlation for the investment blend.

a) Describe in words the approach you have chosen and explain the rational for the model.

Because the weights of each investment in a portfolio are not equal, to achieve an accurate representation of the overall correlation of a portfolio these weights must be taken into account. To achieve this I have wrote a function which builds a vector of correlation scores between every unique combination of investments in a given portfolio which are adjusted for the contribution of both of the investments to the portfolio and also their combined contribution.

Each value is calculated using the formula:
(1 + 2(1-total_prop-a)(1-total_prop-b)c) / (1+2(1-total_prop-a)(1-total_prop-b))

The final value for the correlation of a portofolio is the average of all pairwise correlation values calculated from this formula. This formula works by taking the weights of each variable being compared relative to their total contribution to the portfolio and also the combined contribution the two variables being measured provide to the overall portfolio. These values are multiplied by the measured correlation of the variables and divided by the same value but without the inclusion of the correlation measure. The code submitted in part B shows a more accurate representation of the values used (multiplication symbols removed for markdown purposes). This method appears to optimise correlation very well as from the pairplot below (part c) it can be shown that this formula provides a very smooth fit in it's relationship to both risk and return.

b) Include your R code for the function that calculates this objective.

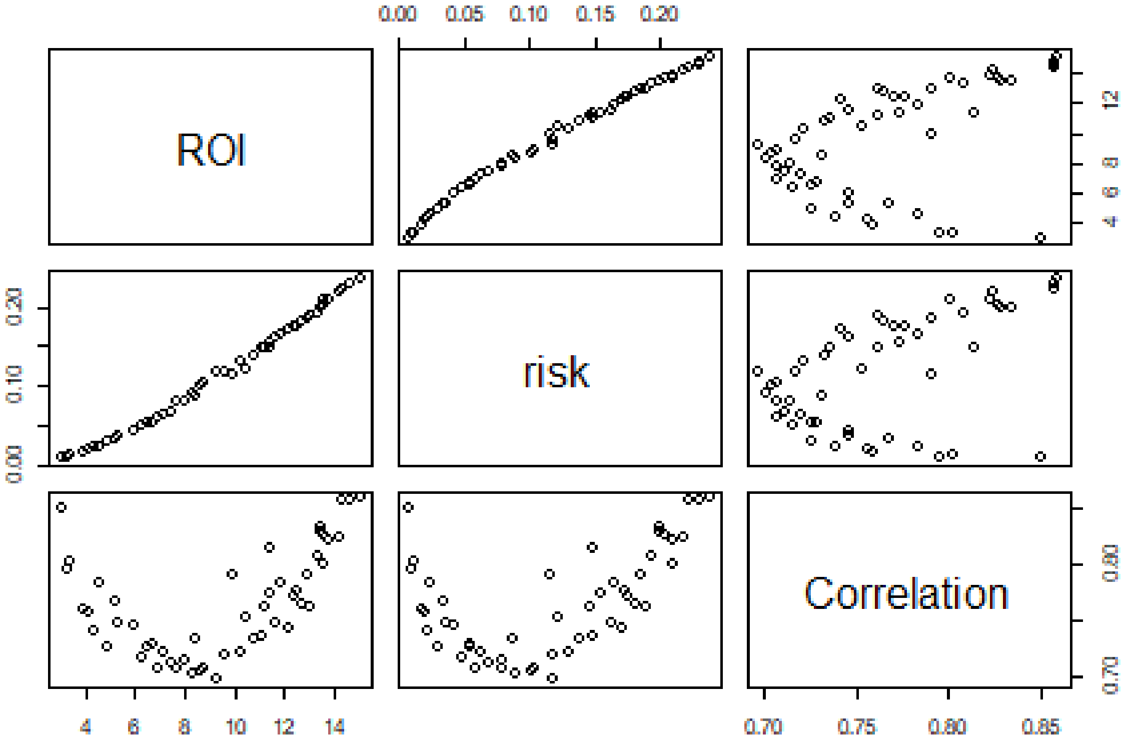
```
mycorr <- function(x) {
selected <- which(x >= minAMOUNT)
comb <- as.data.frame(combn(unique(selected), 2))
values <- rep(0, ncol(comb))
for (i in 1:length(values)) {
aidx <- comb[,i][1]
bidx <- comb[,i][2]
c <- corr[aidx,bidx]
total_prop <- x[aidx] + x[bidx]
v <- 1 /total_prop
a <- x[aidx] * v
b <- x[bidx] * v
values[i] <- (1 + 2a(1-a)c) / (1+2a(1-a))
}
return(mean(values))

}
```

*corr is previously defined in the function as the correlation matrix

c) Produce pairwise plots for the 3 objectives after a successful optimisation.

```
set.seed(5)
source("invest2.0.R")
port <- as.data.frame(portfolio2$value)
colnames(port) <- c("ROI", "risk", "Correlation")
port <- port %>% arrange(desc(ROI))
port$ROI <- -port$ROI
pairs(port)
```



d) Discuss and justify what type of tradeoff of risk/return/correlation you would choose for a portfolio based on the results from part (c).

```
results <- as.data.frame(portfolio2$value)
colnames(results) <- c("roi", "risk", "correlation")
blends <- portfolio2$par
blends[which(blends < 0.05)] = 0
blends[which(blends > 0.20)] = 0
blends <- as.data.frame(blends)
results <- cbind(results, blends)
results <- results %>% arrange(risk)
results <- results[c(5, 25, 45),]
results <- results %>% rowwise() %>% mutate(Stocks= sum(c_across(4:13))) %>%
  rowwise() %>% mutate(Bonds= sum(c_across(14:23))) %>%
  rowwise() %>% mutate(Cash= sum(c_across(24:33))) %>%
  rowwise() %>% mutate(Total= sum(c_across(4:33))) %>%
  mutate(roi = -roi) %>%
  select(-c(4:33))
results[,c(4:7)] <- results[,c(4:7)] * 100
results[,c(4:7)] <- round(results[,c(4:7)],2)
results[,3] <- results[,3]*100
results
```

```
## # A tibble: 3 x 7
## # Rowwise:
##     roi   risk correlation Stocks Bonds  Cash Total
##   <dbl>  <dbl>       <dbl>  <dbl> <dbl> <dbl> <dbl>
## 1  4.14 0.0193        75.7   10.5  29.5  55.8  95.9
## 2  9.24 0.118         69.8   46.2  12.5  36.3  95
## 3 13.6  0.204         82.6   84.0  14.2   0    98.2
```

Based on the pairplot I believe the most appropriate portfolio to choose would be one with medium level risk and return. In the plots and example portfolios provided, it appears that a portfolio with a medium risk for both risk and return also provides the lowest correlation within the portfolio, this means that these portfolios are also the most diverse and therefore could theortically be less risky than they appear as the investments are more likely to differ in behaviour so if one loses money the rest of the investments could still be stable. The values in the table align with this as for the example portfolios it can be seen that the lower correlation values are associated with a more balenced blend of Stocks, Bonds and cash, this also means that the portofolio can still make a worthwhile return as it has a fair proportion of investments that could potentially give a fair return but still holds enough lower risk investments to maintain overall stability. So overall I believe that the portfolios with medium level risk and return with minimised correlation are the safest option when considering all potential outcomes without sacrificing a return on investment.