FE 582 Assignment Problem2

Mugdha

10/17/2020

Problem 2

The data provided in the files contains several quantitative and categorical variables associatewith each ticker. Please select a subset of 100 tickers from each file and use data for a specific year (ex: 2013). Use a small number of quantitative variables (10 or 12) out of ~76 columns available (example: After Tax ROE, Cash Ratio, Current Ratio, Operating Margin, Pre-Tax Margin, Pre-Tax ROE, Profit Margin, Quick Ratio, Total Assets, Total Liabilities, Earnings Per Share, etc. . .).The categorical variables available are GICS Sector, GICS Sub Industry, and possibly HQ Address (although this is sparse data for the 100 tickers subset selected).Next, you have to apply several distance and similarity functions to find the extreme values for distance and similarities between the subset of tickers that you chose. For each of the following cases, please define the function that allows you to calculate the quantity required, calculate the values for all ticker pairs, and rank the pairs by calculated value of distance or similarity, and report the top and bottom 10 values for each case:

Solution:

```
tickers<- read.csv("securities.csv")
fundamentals<- read.csv("fundamentals.csv")
#Taking only 2 categories from securities
tickers$GICS.Sector<- as.numeric(as.factor(tickers$GICS.Sector))
tickers$GICS.Sub.Industry<- as.numeric(as.factor(tickers$GICS.Sub.Industry))

#Data frame of ticker and 2 categories : GICS Sector and GICS Sub Industry
tickersfinal<-data.frame(tickers$Ticker.symbol,tickers$GICS.Sector,tickers$GICS.Sub.Industry)
names(tickersfinal)<-c("Ticker.symbol","GICS.Sector",
                       "GICS.Sub.Industry")

#Selecting year 2012
Dataset<- subset(fundamentals,format(as.Date(fundamentals$Period.Ending),"%Y")==2012)

#Creating a data frame of 9 quantitative features from fundamentals:Ticker symbol,Cash Ratio,Current Ra
FDataset<- data.frame(Dataset$Ticker.Symbol,Dataset$Cash.Ratio,Dataset$Current.Ratio,Dataset$After.Tax.

#Taking subset of 100 from data of 2012 year
Finaldataset<- head(FDataset,100)

names(Finaldataset)<- c("Ticker.symbol","Cash.Ratio","Current.Ratio","After.Tax.ROE","Gross.Margin","Pr

#Merging data set of both quantitative and categorical data (11 features)to form a Final Data which I w
Finaldata<-merge(Finaldataset,tickersfinal,by="Ticker.symbol",all.x=TRUE)
```

```
#Removing NA values from dataset
Finaldata<- na.omit(Finaldata)
```

a,b,c,d)Calculating Lp norms for p=1,2,3,10

Solution:

```
lpnorm <- function(feature_data,rownum,p)
{
  #Selecting only 1 to 7 features for now as they are quantitative and 10 and 11 are categorical.Not ta
  feature_data<- feature_data[1:7]
  ldist =c()
  for(j in 1:nrow(feature_data))
  {
    result<-(sum(abs(feature_data[rownum,2:7]-
            feature_data[j,2:7])^p))^(1/p)
    templ1dist = c(feature_data[rownum,1],feature_data[j,1],result)
    ldist<-append(ldist, templ1dist)
  }
  #Output in the form of Ticker1, Ticker 2 and Distance
  matrix1 <- matrix(ldist, ncol=3, byrow=TRUE)
  df <- as.data.frame(matrix1, stringsAsFactors=FALSE)
  df <- na.omit(df)
  df$V3 <- as.numeric(as.character(df$V3))
  dfSorted <-df[order(-df$V3),]
  names(dfSorted)<- c("Ticker1","Ticker2","Distance")
  #Top 10
  print(head(dfSorted, 10))
  #Bottom 10
  print(tail(dfSorted, 10))
}
#Distance of ticker 1 (Row=1) with other tickers, p=1
lpnorm(Finaldata,1,1)
```

```
##    Ticker1 Ticker2 Distance
## 3      AAL    ABBV     3341
## 60     AAL    FLIR      520
## 9      AAL    AKAM      485
## 27     AAL      CF      459
## 35     AAL    CTSH      452
## 53     AAL      EW      422
## 10     AAL     ALB      402
## 54     AAL    EXPD      339
## 66     AAL    GRMN      338
## 17     AAL    ATVI      326
##    Ticker1 Ticker2 Distance
## 41     AAL     DNB      107
## 59     AAL    FISV      106
## 23     AAL     BMY       98
## 34     AAL     CTL       97
## 8      AAL     AEP       89
## 11     AAL     ALK       87
## 28     AAL     CHD       86
```

```
## 32     AAL     CNP       72
## 45     AAL     EFX       63
## 1      AAL     AAL        0
```

```
#Row=1 ,p=2
lpnorm(Finaldata,1,2)
```

```
##    Ticker1 Ticker2  Distance
## 3      AAL    ABBV 2192.1307
## 60     AAL    FLIR  412.2693
## 9      AAL    AKAM  325.6885
## 35     AAL    CTSH  313.5379
## 53     AAL      EW  310.1355
## 10     AAL     ALB  297.7281
## 27     AAL      CF  287.4596
## 56     AAL     FCX  238.4492
## 15     AAL     APH  234.7190
## 54     AAL    EXPD  225.8561
##    Ticker1 Ticker2 Distance
## 42     AAL     DUK 56.88585
## 49     AAL    EQIX 54.15718
## 28     AAL     CHD 53.94442
## 52     AAL     ETR 52.78257
## 11     AAL     ALK 51.87485
## 8      AAL     AEP 50.56679
## 23     AAL     BMY 47.66550
## 32     AAL     CNP 35.07136
## 45     AAL     EFX 33.71943
## 1      AAL     AAL  0.00000
```

```
#Row=1 ,p=3
lpnorm(Finaldata,1,3)
```

```
##    Ticker1 Ticker2  Distance
## 3      AAL    ABBV 1949.2449
## 60     AAL    FLIR  405.0504
## 9      AAL    AKAM  298.2371
## 53     AAL      EW  297.4956
## 10     AAL     ALB  289.4800
## 35     AAL    CTSH  287.8551
## 27     AAL      CF  255.4603
## 56     AAL     FCX  231.2947
## 15     AAL     APH  228.0860
## 54     AAL    EXPD  209.3493
##    Ticker1 Ticker2 Distance
## 33     AAL     COG 49.46374
## 11     AAL     ALK 45.59939
## 8      AAL     AEP 44.99144
## 49     AAL    EQIX 44.66066
## 42     AAL     DUK 44.60494
## 52     AAL     ETR 43.74299
## 23     AAL     BMY 40.22951
## 45     AAL     EFX 30.66408
```

```
## 32     AAL     CNP 28.46255
## 1      AAL     AAL  0.00000
```

```
#Row=1 ,p=10
lpnorm(Finaldata,1,10)
```

```
##    Ticker1 Ticker2  Distance
## 3      AAL    ABBV 1666.5207
## 60     AAL    FLIR  404.0000
## 53     AAL      EW  294.0004
## 10     AAL     ALB  288.0000
## 9      AAL    AKAM  281.1210
## 35     AAL    CTSH  272.0981
## 56     AAL     FCX  230.0000
## 15     AAL     APH  227.0000
## 27     AAL      CF  221.2502
## 54     AAL    EXPD  202.0132
##    Ticker1 Ticker2 Distance
## 41     AAL     DNB 42.82262
## 21     AAL     BBY 41.90277
## 11     AAL     ALK 39.09580
## 49     AAL    EQIX 39.04047
## 52     AAL     ETR 39.02083
## 23     AAL     BMY 37.00578
## 42     AAL     DUK 35.25227
## 45     AAL     EFX 30.00004
## 32     AAL     CNP 22.74620
## 1      AAL     AAL  0.00000
```

e)Minkovski distance (assign different weights for the feature components in the Lp-norm based on your assessment on the importance of the features)

Solution:

```
minkovskiDist <- function(feature_data,rownum,p)
{
  #Selecting only 1 to 9 features as they are quantitative    and 10 and 11 are categorical
  feature_data<- feature_data[1:9]
  l1dist =c()
  for(j in 1:nrow(feature_data))
  {

    result<-(sum(abs(feature_data[rownum,2:7]-feature_data[j,2:7])^p))

    #Assigning weight to Assets and Liabilities features
    resultTotalAssetLiab<-(sum((1/1000000000) * (abs(feature_data[rownum,8:9]-feature_data[j,8:9])^p)))
    finalResult<-(sum(result, resultTotalAssetLiab))^(1/p)

    templ1dist = c(feature_data[rownum,1], feature_data[j,1], finalResult)
    l1dist<-append(l1dist, templ1dist)
  }
  #Output in the form of Ticker1, Ticker 2 and Distance
  m1 <- matrix(l1dist, ncol=3, byrow=TRUE)
```

```
  df <- as.data.frame(m1, stringsAsFactors=FALSE)
  #print(df)
  df <- na.omit(df)
  df$V3 <- as.numeric(as.character(df$V3))
  dfSorted <-df[order(-df$V3),]
  #print(dfSorted)
  names(dfSorted)<- c("Ticker1","Ticker2","Distance")
  print(head(dfSorted, 10))
  print(tail(dfSorted, 10))
}
#Minkovski distance for row=1(First row) with other tickers
minkovskiDist(Finaldata,1,1)
```

```
##    Ticker1 Ticker2  Distance
## 3      AAL    ABBV 3346.9650
## 60     AAL    FLIR  565.6196
## 9      AAL    AKAM  530.5455
## 27     AAL      CF  492.9694
## 35     AAL    CTSH  492.2122
## 53     AAL      EW  467.4373
## 10     AAL     ALB  445.3600
## 36     AAL     CVX  414.0390
## 54     AAL    EXPD  383.5204
## 66     AAL    GRMN  380.2945
##    Ticker1 Ticker2 Distance
## 34     AAL     CTL 137.1900
## 52     AAL     ETR 136.7129
## 50     AAL      ES 134.6181
## 8      AAL     AEP 134.0960
## 28     AAL     CHD 128.2657
## 11     AAL     ALK 125.8120
## 23     AAL     BMY 113.0040
## 45     AAL     EFX 104.2940
## 32     AAL     CNP  78.9600
## 1      AAL     AAL   0.0000
```

Assets and Liabilities in the data set had values in billions which resulted in large distances. Hence assigned weights to these features

f)Match-Based Similarity Computation

Solution:

```
matchSim <- function(feature_data,rownum,p)
{
  #Selecting 1 to 9 as those are the 9 quantitative features of the data
  feature_data<- feature_data[1:9]
  matchdist =c()
  for(j in 1:nrow(feature_data))
  {
    tempMatchDist = c()
    for(k in 2:ncol(feature_data)) {
      max = max(feature_data[,k])
      min = min(feature_data[,k])
```

```
  # calculate the bucket ranges using bucket size as 3
     bucketRange = round(((max-min)/3),0)
  # create buckets
     feature_data$buck= cut(feature_data[,k],c(min,bucketRange,bucketRange*2,max),
                  labels=c("Bucket1","Bucket2","Bucket3"),
                  include.lowest = TRUE)
  # check if the feature belongs to same bucket
      if(feature_data[rownum,10] == feature_data[j,10]){
        # find min and max of the bucket
        minBuck=c()
        maxBuck=c()
        if(feature_data[rownum,10] == "Bucket1") {
          minBuck = min
          maxBuck = bucketRange
        } else if(feature_data[rownum,10] == "Bucket2") {
          minBuck = bucketRange+1
          maxBuck = bucketRange*2
        } else {
          minBuck = bucketRange*2+1
          maxBuck = max
        }
        # compute the expression
        result = (1-abs(feature_data[rownum,k]-feature_data[j,k])/(maxBuck-minBuck))^p

        # add to tempMatchDist
        tempMatchDist = append(tempMatchDist,result)
      }
      # removing temporary bucket column
      feature_data$buck <- NULL
    }
    # add tickers and tempMatchDist to matchDist
    finalResult = c(feature_data[rownum,1], feature_data[j,1], sum(tempMatchDist)^(1/p))
    matchdist = append(matchdist, finalResult)
  }
  # sorting and printing
   #Output in the form of Ticker1, Ticker 2 and Distance
  m1 <- matrix(matchdist, ncol=3, byrow=TRUE)
  df <- as.data.frame(m1, stringsAsFactors=FALSE)
  df <- na.omit(df)
  df$V3 <- as.numeric(as.character(df$V3))
  dfSorted <-df[order(-df$V3),]
  names(dfSorted)<- c("Ticker1","Ticker2","Match Based Similarity")
  #Top 10
  print(head(dfSorted, 10))
  #Bottom 10
  print(tail(dfSorted, 10))
}
#Match based similarity for row=1 (First row) and p=2 with other tickers
matchSim(Finaldata,1,2)
```

```
##    Ticker1 Ticker2 Match Based Similarity
## 1      AAL     AAL               2.828427
## 32     AAL     CNP               2.431381
```

```
## 50     AAL      ES               2.331827
## 8      AAL     AEP               2.270985
## 23     AAL     BMY               2.258286
## 45     AAL     EFX               2.252010
## 29     AAL     CHK               2.241684
## 40     AAL     DHR               2.224828
## 34     AAL     CTL               2.216629
## 44     AAL     ECL               2.214318
##    Ticker1 Ticker2 Match Based Similarity
## 27     AAL      CF               1.817072
## 66     AAL    GRMN               1.804113
## 10     AAL     ALB               1.792602
## 15     AAL     APH               1.777464
## 24     AAL     BSX               1.767921
## 62     AAL    FSLR               1.731998
## 53     AAL      EW               1.699036
## 4      AAL     ABT               1.663164
## 17     AAL    ATVI               1.657058
## 3      AAL    ABBV               1.326582
```

g)Mahalanobis distance

Solution:

```
mahalanoDist <- function(feature_data,rownum){
  mahaDist <- c()
  for(j in 1:nrow(feature_data)){
    feature_data<- na.omit(feature_data)
    #Calculating Covariance of Feature Data
    #Selecting only first 7 quantitative features
    CovMat <- cov(feature_data[2:7])
    matrix_S <-as.matrix(feature_data[rownum,2:7]-
                         feature_data[j,2:7])[1,]
    vector1<-(as.vector(matrix_S))
    #Calculating the expression
    dmahal <- vector1 %*% solve(as.matrix(CovMat))
    finalmaha <- dmahal * t(matrix_S)
    mahafinal<-(sum(finalmaha))
   tempmahafinal<-c(feature_data[rownum,1], feature_data[j,1], mahafinal)
    mahaDist <- append(mahaDist,tempmahafinal)
  }
  #Output in the form of Ticker1, Ticker 2 and Distance
  m1 <- matrix(mahaDist, ncol=3, byrow=TRUE)
  df <- as.data.frame(m1, stringsAsFactors=FALSE)
  df <- na.omit(df)
  df$V3 <- as.numeric(as.character(df$V3))
  dfSorted <-df[order(-df$V3),]
  names(dfSorted)<- c("Ticker1","Ticker2","Distance")
  #Top 10
  print(head(dfSorted, 10))
  #Bottom 10
  print(tail(dfSorted, 10))

}
```

```
#Mahalanolobis Distance for row=1 (First row) with other tickers
mahalanoDist(Finaldata,1)
```

```
##    Ticker1 Ticker2 Distance
## 3      AAL    ABBV 70.17360
## 24     AAL     BSX 43.70043
## 60     AAL    FLIR 33.42771
## 31     AAL      CL 28.49432
## 6      AAL     ADS 20.14839
## 27     AAL      CF 19.43168
## 4      AAL     ABT 16.16001
## 53     AAL      EW 15.14914
## 10     AAL     ALB 14.44476
## 67     AAL     GWW 14.38654
##    Ticker1 Ticker2  Distance
## 29     AAL     CHK 1.8806659
## 26     AAL     CCI 1.7813290
## 28     AAL     CHD 1.7760207
## 45     AAL     EFX 1.5942758
## 57     AAL      FE 1.5834305
## 8      AAL     AEP 1.5047047
## 32     AAL     CNP 1.2946144
## 11     AAL     ALK 1.0867975
## 49     AAL    EQIX 0.5949362
## 1      AAL     AAL 0.0000000
```

h)Similarity: overlap measure

Solution:

```
olap<- function(feature_data,rownum)
{
  #Taking categorical features Sector and Sub sector from the data set
  feature_data<- feature_data[c(1,10:11)]
  feature_data<- na.omit(feature_data)
  opdist <- c()
  for(j in 1:nrow(feature_data))
  {
    result = c()
    #Checking if same category for both the features
    if(feature_data[rownum,2]==feature_data[j,2])
    {
       result = append(result,1)
    }
    if(feature_data[rownum,3]==feature_data[j,3])
    {
       result = append(result,1)
    }
    tempOpDist =  c(feature_data[rownum,1],feature_data[j,1],sum(result))
    opdist<- append(opdist, tempOpDist)
  }
   #Output in the form of Ticker1, Ticker 2 and Overlap
  m1 <- matrix(opdist, ncol=3, byrow=TRUE)
```

```
  df <- as.data.frame(m1, stringsAsFactors=FALSE)
  df <- na.omit(df)
  df$V3 <- as.numeric(as.character(df$V3))
  dfSorted <-df[order(-df$V3),]
  names(dfSorted)<- c("Ticker1","Ticker2","Overlap Measure")
  #Top 10
  print(head(dfSorted, 10))
  #Bottom 10
  print(tail(dfSorted, 10))
}

#Overlap measure for row=7 with other tickers
olap(Finaldata,7)
```

```
##    Ticker1 Ticker2 Overlap Measure
## 7      AEE     AEE               2
## 32     AEE     CNP               2
## 50     AEE      ES               2
## 8      AEE     AEP               1
## 19     AEE     AWK               1
## 38     AEE       D               1
## 42     AEE     DUK               1
## 46     AEE     EIX               1
## 52     AEE     ETR               1
## 57     AEE      FE               1
##    Ticker1 Ticker2 Overlap Measure
## 60     AEE    FLIR               0
## 61     AEE     FMC               0
## 62     AEE    FSLR               0
## 63     AEE     FTR               0
## 64     AEE    GILD               0
## 65     AEE     GPC               0
## 66     AEE    GRMN               0
## 67     AEE     GWW               0
## 68     AEE     HAS               0
## 69     AEE     HCA               0
```

i)Similarity: inverse frequency

Solution:

```
invf<- function(feature_data,rownum)
{   #Taking categorical features Sector and Sub sector from     the data set
  feature_data<- feature_data[c(1,10:11)]
  iofDist <- c()
  for(j in 1:nrow(feature_data))
  {
    result = c()
    #Checking if same category
    if(feature_data[rownum,2]==feature_data[j,2])
    {
    #Calculating the fraction of records for same category feature
      fractionDF = feature_data[feature_data$GICS.Sector == feature_data[rownum,2],]
```

```
      p = length(fractionDF$GICS.Sector) / 100
      # Calculating the expression
      result = append(result,(1/p)^2)
    }
    if(feature_data[rownum,3]==feature_data[j,3])
    {
    #Calculating the fraction of records for same categoryfeature
      fractionDF = feature_data[feature_data$GICS.Sector == feature_data[rownum,3],]
      p = length(fractionDF$GICS.Sector) / 100
       # Calculating the expression
      result = append(result,(1/p)^2)
    }
    tempIofDist = c(feature_data[rownum,1],feature_data[j,1],sum(result))
    iofDist<- append(iofDist, tempIofDist)
  }
   #Output in the form of Ticker1, Ticker 2 and Similarity
  m1 <- matrix(iofDist, ncol=3, byrow=TRUE)
  df <- as.data.frame(m1, stringsAsFactors=FALSE)
  df <- na.omit(df)
  df$V3 <- as.numeric(as.character(df$V3))
  dfSorted <-df[order(-df$V3),]
  names(dfSorted)<- c("Ticker1","Ticker2","Similarity")
  #Top 10
  print(head(dfSorted, 10))
  #Bottom 10
  print(tail(dfSorted, 10))
}
#Inverse occurrence frequency for row=1 (First row) with other tickers
invf(Finaldata,1)
```

```
##    Ticker1 Ticker2 Similarity
## 1      AAL     AAL  169.44444
## 11     AAL     ALK  169.44444
## 12     AAL     AME   69.44444
## 16     AAL    ARNC   69.44444
## 30     AAL    CHRW   69.44444
## 40     AAL     DHR   69.44444
## 41     AAL     DNB   69.44444
## 45     AAL     EFX   69.44444
## 51     AAL     ETN   69.44444
## 54     AAL    EXPD   69.44444
##    Ticker1 Ticker2 Similarity
## 59     AAL    FISV          0
## 60     AAL    FLIR          0
## 61     AAL     FMC          0
## 62     AAL    FSLR          0
## 63     AAL     FTR          0
## 64     AAL    GILD          0
## 65     AAL     GPC          0
## 66     AAL    GRMN          0
## 68     AAL     HAS          0
## 69     AAL     HCA          0
```

j)Similarity: Goodall

Solution:

```
gooDallSim<- function(feature_data,rownum)
{
  feature_data<- feature_data[c(1,10:11)]
  iofDist <- c()
  for(j in 1:nrow(feature_data))
  {
    result = c()
    if(feature_data[rownum,2]==feature_data[j,2])
    {
      fractionDF = feature_data[feature_data$GICS.Sector == feature_data[rownum,2],]
    #Calculating the fraction of records for same category feature
      p = length(fractionDF$GICS.Sector) / 100
       # Calculating the expression
      result = append(result,(1-p^2))
    }
    if(feature_data[rownum,3]==feature_data[j,3])
    {
      fractionDF = feature_data[feature_data$GICS.Sector == feature_data[rownum,3],]
    #Calculating the fraction of records for same category feature
      p = length(fractionDF$GICS.Sector) / 100
        #Calculating the expression
      result = append(result,(1-p^2))
    }
    tempIofDist = c(feature_data[rownum,1],feature_data[j,1],sum(result))
    iofDist<- append(iofDist, tempIofDist)

  }
  #Output in the form of Ticker1, Ticker 2 and Similarity
  m1 <- matrix(iofDist, ncol=3, byrow=TRUE)
  df <- as.data.frame(m1, stringsAsFactors=FALSE)
  df <- na.omit(df)
  df$V3 <- as.numeric(as.character(df$V3))
  dfSorted <-df[order(-df$V3),]
  names(dfSorted)<- c("Ticker1","Ticker2","Similarity")
  #Top 10
  print(head(dfSorted, 10))
  #Bottom 10
  print(tail(dfSorted, 10))
}

#Goodall Measure for row=1 (First row) with other tickers
gooDallSim(Finaldata,2)
```

```
##    Ticker1 Ticker2 Similarity
## 2      AAP     AAP     1.9975
## 21     AAP     BBY     0.9975
## 65     AAP     GPC     0.9975
## 66     AAP    GRMN     0.9975
## 68     AAP     HAS     0.9975
## 1      AAP     AAL     0.0000
## 3      AAP    ABBV     0.0000
## 4      AAP     ABT     0.0000
```

```
## 5      AAP     ADM     0.0000
## 6      AAP     ADS     0.0000
##    Ticker1 Ticker2 Similarity
## 57     AAP      FE          0
## 58     AAP     FIS          0
## 59     AAP    FISV          0
## 60     AAP    FLIR          0
## 61     AAP     FMC          0
## 62     AAP    FSLR          0
## 63     AAP     FTR          0
## 64     AAP    GILD          0
## 67     AAP     GWW          0
## 69     AAP     HCA          0
```

k)Overall similarity between tickers by using mixed type data (choose a lambda value for calculation)

Solution:

```
overallsim <- function(feature_data,rownum,p,lmda)
{
  #calculating match based similarity for NumSim
  overallSimilarities = c()
  for(j in 1:nrow(feature_data))
  {
    tempMatchDist = c()
    #Taking columns 2 to 9 as they are quantitative features
    for(k in 2:(ncol(feature_data)-2)) {
      max = max(feature_data[,k])
      min = min(feature_data[,k])
      # calculate the bucket ranges
      bucketRange = round(((max-min)/3),0)
      # create buckets
      feature_data$buck= cut(feature_data[,k], c(min,bucketRange,bucketRange*2,max),
                             labels = c("Bucket1","Bucket2","Bucket3"),
                             include.lowest = TRUE)

      # check if features belongs to same bucket
      if(feature_data[rownum,12] == feature_data[j,12]){
        # find min and max of the bucket
        minBuck=c()
        maxBuck=c()
        if(feature_data[rownum,12] == "Bucket1") {
          minBuck = min
          maxBuck = bucketRange
        } else if(feature_data[rownum,12] == "Bucket2") {
          minBuck = bucketRange+1
          maxBuck = bucketRange*2
        } else {
          minBuck = bucketRange*2+1
          maxBuck = max
        }
        # compute the expression
        result = (1-abs(feature_data[rownum,k]-feature_data[j,k])/(maxBuck-minBuck))^p
```

```
        # add to tempMatchDist
        tempMatchDist = append(tempMatchDist,result)
      }
      # removing temporary bucket column
      feature_data$buck <- NULL
    }
     # compute the expression
    numSim = sum(tempMatchDist)^(1/p) * lmda

    # Calculate CatSim using overlap measure
    opdist <- c()
    #Taking features 10 and 11 as they are categorical features
    if(feature_data[rownum,10]==feature_data[j,10])
    {
      opdist = append(opdist,1)
    }
    if(feature_data[rownum,11]==feature_data[j,11])
    {
      opdist = append(opdist,1)
    }
     # compute the expression
    catSim <- sum(opdist) * (1-lmda)

    overallSimilarity = c(feature_data[rownum,1], feature_data[j,1], sum(numSim,catSim))
    overallSimilarities = append(overallSimilarities, overallSimilarity)
  }
  # sorting and printing
   #Output in the form of Ticker1, Ticker 2 and Similarity
  m1 <- matrix(overallSimilarities, ncol=3, byrow=TRUE)
  df <- as.data.frame(m1, stringsAsFactors=FALSE)
  df <- na.omit(df)
  df$V3 <- as.numeric(as.character(df$V3))
  dfSorted <-df[order(-df$V3),]
  names(dfSorted)<- c("Ticker1","Ticker2","Similarity")
  #Top 10
  print(head(dfSorted, 10))
  #Bottom 10
  print(tail(dfSorted, 10))
}
#Calculating overall similarity by taking row=1, p=2 and lambda value as 0.25
overallsim(Finaldata,1,2,0.25)
```

```
##    Ticker1 Ticker2 Similarity
## 1      AAL     AAL   2.207107
## 11     AAL     ALK   2.029507
## 45     AAL     EFX   1.313002
## 40     AAL     DHR   1.306207
## 67     AAL     GWW   1.290186
## 51     AAL     ETN   1.289194
## 16     AAL    ARNC   1.273223
## 55     AAL    FBHS   1.254776
## 41     AAL     DNB   1.243364
## 12     AAL     AME   1.226110
```

```
##    Ticker1 Ticker2 Similarity
## 27     AAL      CF  0.4542679
## 66     AAL    GRMN  0.4510283
## 10     AAL     ALB  0.4481506
## 15     AAL     APH  0.4443659
## 24     AAL     BSX  0.4419802
## 62     AAL    FSLR  0.4329994
## 53     AAL      EW  0.4247590
## 4      AAL     ABT  0.4157910
## 17     AAL    ATVI  0.4142645
## 3      AAL    ABBV  0.3316455
```

l)Overall normalized similarity between tickers by using mixed type data (choose a lambda value for calculation)

Solution:

```
overallSimNormalized <- function(feature_data,rownum,p,lmda)
{
  #calculating match based similarity for NumSim
  overallSimilarities = c()
  for(j in 1:nrow(feature_data))
  {
    tempMatchDist = c()
    for(k in 2:(ncol(feature_data)-2)) {
      max = max(feature_data[,k])
      min = min(feature_data[,k])
      # calculate the bucket ranges
      bucketRange = round(((max-min)/3),0)
      # create buckets
      feature_data$buck= cut(feature_data[,k], c(min,bucketRange,bucketRange*2,max),
                             labels = c("Bucket1","Bucket2","Bucket3"),
                             include.lowest = TRUE)

      # check if features belongs to same bucket
      if(feature_data[rownum,12] == feature_data[j,12]){
        # find min and max of the bucket
        minBuck=c()
        maxBuck=c()
        if(feature_data[rownum,12] == "Bucket1") {
          minBuck = min
          maxBuck = bucketRange
        } else if(feature_data[rownum,12] == "Bucket2") {
          minBuck = bucketRange+1
          maxBuck = bucketRange*2
        } else {
          minBuck = bucketRange*2+1
          maxBuck = max
        }
        # compute the expression
        result = (1-abs(feature_data[rownum,k]-feature_data[j,k])/(maxBuck-minBuck))^p

        # add to tempMatchDist
        tempMatchDist = append(tempMatchDist,result)
```

```
      }
      # removing temporary bucket column
      feature_data$buck <- NULL
    }
    # compute the expression
    overallNumSim = sum(tempMatchDist)^(1/p)
    # Calculate CatSim using overlap measure
    opdist <- c()
    if(feature_data[rownum,10]==feature_data[j,10])
    {
      opdist = append(opdist,1)
    }
    if(feature_data[rownum,11]==feature_data[j,11])
    {
      opdist = append(opdist,1)
    }
    overallCatSim = sum(opdist)
    overallSimilarity = c(feature_data[rownum,1], feature_data[j,1], overallNumSim, overallCatSim)
    overallSimilarities = append(overallSimilarities, overallSimilarity)
  }
  # sorting and printing
   #Output in the form of Ticker1, Ticker 2 and Similarity
  m1 <- matrix(overallSimilarities, ncol=4, byrow=TRUE)
  df <- as.data.frame(m1, stringsAsFactors=FALSE)
  #print(df)
  df <- na.omit(df)
  df$V3 <- as.numeric(as.character(df$V3))
  df$V4 <- as.numeric(as.character(df$V4))

  sigmaNum = sd(df$V3)
  sigmaCat = sd(df$V4)
  #calculating the sigmaNum and sigmaCat values for quantitative and categorical features
  df$V5 <- (df$V3 * (1/sigmaNum) * lmda) + (df$V4 * (1/sigmaCat) * (1-lmda))
  df$V3<-NULL
  df$V4<-NULL
  df$V5 <- as.numeric(as.character(df$V5))
  dfSorted <-df[order(-df$V5),]
  # print(dfSorted)
  names(dfSorted)<- c("Ticker1","Ticker2","Similarity")
  print(head(dfSorted, 10))
  print(tail(dfSorted, 10))
}
#Calculating overall normalized similarity by taking row=1, p=2 and lambda value as 0.25
overallSimNormalized(Finaldata,1,2,0.25)
```

```
##    Ticker1 Ticker2 Similarity
## 1      AAL     AAL   6.512960
## 11     AAL     ALK   5.675065
## 45     AAL     EFX   4.244638
## 40     AAL     DHR   4.212578
## 67     AAL     GWW   4.136992
## 51     AAL     ETN   4.132314
## 16     AAL    ARNC   4.056962
```

```
## 55     AAL    FBHS   3.969931
## 41     AAL     DNB   3.916091
## 12     AAL     AME   3.834689
##    Ticker1 Ticker2 Similarity
## 27     AAL      CF   2.143187
## 66     AAL    GRMN   2.127903
## 10     AAL     ALB   2.114326
## 15     AAL     APH   2.096471
## 24     AAL     BSX   2.085215
## 62     AAL    FSLR   2.042845
## 53     AAL      EW   2.003967
## 4      AAL     ABT   1.961657
## 17     AAL    ATVI   1.954456
## 3      AAL    ABBV   1.564668
```