DS-670 Assignme…

```
%r
# loading the data into RStudio. All the data with the dimension = ARQ.
market_data <- read.csv("/Users/Mohamar/Documents/ARQ_Data_file.csv", header=TRUE, sep=",")
head(market_data, 2)
t(names(market_data))
```

FINISHED

```
ticker dimension calendardate    datekey reportperiod    accoci
1      A       ARQ   2011-03-31 2011-03-09   2011-01-31 -63000000
2      A       ARQ   2011-06-30 2011-06-07   2011-04-30 278000000
     assets assetsavg   assetsc  assetsnc assetturnover   bvps    capex
1 8.044e+09        NA 4.598e+09 3.446e+09            NA  9.602 -3.8e+07
2 8.649e+09        NA 5.096e+09 3.553e+09            NA 11.392 -5.1e+07
    cashneq cashnequsd      cor currentratio    de      debt   debtusd
1 2.655e+09  2.655e+09 7.03e+08        3.270 1.412 2.139e+09 2.139e+09
2 2.975e+09  2.975e+09 7.77e+08        3.201 1.186 2.144e+09 2.144e+09
  depamor divyield dps     ebit   ebitda ebitdamargin ebitdausd  ebitusd
1 6.3e+07        0   0 2.21e+08 2.84e+08        0.187  2.84e+08 2.21e+08
2 6.4e+07        0   0 2.80e+08 3.44e+08        0.205  3.44e+08 2.80e+08
       ebt  eps epsdil epsusd    equity equityavg equityusd          ev
1 1.98e+08 0.56   0.54   0.56 3.332e+09        NA 3.332e+09 10852293091
2 2.60e+08 0.58   0.56   0.58 3.953e+09        NA 3.953e+09 11044359780
  evebit evebitda      fcf fcfps fxusd       gp grossmargin intangibles
1     12    9.612 8.20e+07 0.236     1 8.16e+08       0.537   1.915e+09
```

```
%r
# for loop to estimate returns (as a single column) based on price: P(i)/P(i-1)
returns <- vector();
for(i in 2:length(market_data[,1]))
{
  if(identical(market_data[i,1], market_data[i-1,1]))
  {
    returns[i]=market_data[i,72]/market_data[i-1,72];
  }
  else {
    returns[i]=0;
  }
}

returns[1]=0;

# adding the returns column to the market_data
market_data <- cbind(market_data,returns)

head(market_data, 2)
t(names(market_data))
```

FINISHED

```
ticker dimension calendardate    datekey reportperiod    accoci
1      A       ARQ   2011-03-31 2011-03-09   2011-01-31 -63000000
2      A       ARQ   2011-06-30 2011-06-07   2011-04-30 278000000
     assets assetsavg   assetsc  assetsnc assetturnover   bvps    capex
1 8.044e+09        NA 4.598e+09 3.446e+09            NA  9.602 -3.8e+07
2 8.649e+09        NA 5.096e+09 3.553e+09            NA 11.392 -5.1e+07
    cashneq cashnequsd      cor currentratio    de      debt   debtusd
1 2.655e+09  2.655e+09 7.03e+08        3.270 1.412 2.139e+09 2.139e+09
2 2.975e+09  2.975e+09 7.77e+08        3.201 1.186 2.144e+09 2.144e+09
  depamor divyield dps     ebit   ebitda ebitdamargin ebitdausd  ebitusd
1 6.3e+07        0   0 2.21e+08 2.84e+08        0.187  2.84e+08 2.21e+08
2 6.4e+07        0   0 2.80e+08 3.44e+08        0.205  3.44e+08 2.80e+08
       ebt  eps epsdil epsusd    equity equityavg equityusd          ev
1 1.98e+08 0.56   0.54   0.56 3.332e+09        NA 3.332e+09 10852293091
2 2.60e+08 0.58   0.56   0.58 3.953e+09        NA 3.953e+09 11044359780
  evebit evebitda      fcf fcfps fxusd       gp grossmargin intangibles
1     12    9.612 8.20e+07 0.236     1 8.16e+08       0.537   1.915e+09
2     10    8.655 3.27e+08 0.942     1 9.00e+08       0.537   2.048e+09
```

```
%r                                                                                    FINISHED
# selecting my 20 factors
market_data.factors <- market_data[c(1,3,12,17,18,22,26,30,38,40,43,65,67,68,69,70,72,73,74
head(market_data.factors, 2)
t(names(market_data.factors))
```

```
ticker calendardate   bvps currentratio    de divyield ebitdamargin  eps
1      A   2011-03-31  9.602        3.270 1.412       0        0.187 0.56
2      A   2011-06-30 11.392        3.201 1.186       0        0.205 0.58
  evebitda fcfps grossmargin netmargin payoutratio    pb     pe    pe1
1    9.612 0.236       0.537     0.127           0 3.412 14.246 14.322
2    8.655 0.942       0.537     0.119           0 3.004 13.343 13.276
  price    ps   ps1   revenue    sps  tbvps  returns
1 32.94 1.977 1.988 1.519e+09 16.571 17.663 0.000000
2 34.12 1.929 1.923 1.677e+09 17.741 19.023 1.035823
     [,1]     [,2]           [,3]   [,4]           [,5] [,6]
[1,] “ticker” “calendardate” “bvps” “currentratio” “de” “divyield”
     [,7]           [,8]  [,9]       [,10]   [,11]         [,12]
[1,] “ebitdamargin” “eps” “evebitda” “fcfps” “grossmargin” “netmargin”
     [,13]         [,14] [,15] [,16] [,17]   [,18] [,19] [,20]     [,21]
[1,] “payoutratio” “pb”  “pe”  “pe1” “price” “ps”  “ps1” “revenue” “sps”
     [,22]   [,23]
[1,] “tbvps” “returns”
```

```
%r                                                                                    FINISHED
# for loop to estimate log of returns (as a single column) based on price: Pt/P(t-1)
logreturns <- vector();
for(i in 2:length(market_data.factors[,1]))
  {
    logreturns[i]=log(market_data.factors[i,23]);
}
```

```
head(logreturns)

# adding the logreturns column to the market_data
market_data.factors <- cbind(market_data.factors,logreturns)

head(market_data.factors, 2)
t(names(market_data.factors))
```

```
[1]          NA  0.03519600 -0.29006866 -0.06451502  0.24573265 -0.12271741
  ticker calendardate   bvps currentratio    de divyield ebitdamargin  eps
1      A   2011-03-31  9.602        3.270 1.412        0        0.187 0.56
2      A   2011-06-30 11.392        3.201 1.186        0        0.205 0.58
  evebitda fcfps grossmargin netmargin payoutratio    pb     pe    pe1
1    9.612 0.236       0.537     0.127           0 3.412 14.246 14.322
2    8.655 0.942       0.537     0.119           0 3.004 13.343 13.276
  price    ps   ps1   revenue    sps  tbvps  returns logreturns
1 32.94 1.977 1.988 1.519e+09 16.571 17.663 0.000000         NA
2 34.12 1.929 1.923 1.677e+09 17.741 19.023 1.035823   0.035196
     [,1]     [,2]           [,3]   [,4]           [,5] [,6]
[1,] “ticker” “calendardate” “bvps” “currentratio” “de” “divyield”
     [,7]           [,8]  [,9]       [,10]   [,11]         [,12]
[1,] “ebitdamargin” “eps” “evebitda” “fcfps” “grossmargin” “netmargin”
     [,13]         [,14] [,15] [,16] [,17]   [,18] [,19] [,20]     [,21]
[1,] “payoutratio” “pb”  “pe”  “pe1” “price” “ps”  “ps1” “revenue” “sps”
     [,22]   [,23]     [,24]
[1,] “tbvps” “returns” “logreturns”
```

FINISHED

```
%r
# replacing all 'Inf' in the data with NA
market_data.factors[mapply(is.infinite, market_data.factors)] <- NA
head(market_data.factors)

# removing all NAs
market_data.factors.noNA <- na.omit(market_data.factors)
head(market_data.factors.noNA, 2)
```

```
ticker calendardate   bvps currentratio    de divyield ebitdamargin  eps
1      A   2011-03-31  9.602        3.270 1.412    0.000        0.187 0.56
2      A   2011-06-30 11.392        3.201 1.186    0.000        0.205 0.58
3      A   2011-09-30 12.046        3.470 1.086    0.000        0.215 0.95
4      A   2011-12-31 12.415        3.032 1.101    0.000        0.219 0.84
5      A   2012-03-31 12.920        3.352 1.023    0.000        0.210 0.66
6      A   2012-06-30 13.569        3.275 0.993    0.003        0.219 0.73
  evebitda fcfps grossmargin netmargin payoutratio    pb     pe    pe1
1    9.612 0.236       0.537     0.127       0.000 3.412 14.246 14.322
2    8.655 0.942       0.537     0.119       0.000 3.004 13.343 13.276
3    5.968 0.632       0.527     0.195       0.000 2.115  8.733  8.713
4    5.099 1.329       0.533     0.167       0.000 1.934  8.233  8.197
5    6.399 0.299       0.535     0.141       0.000 2.366 10.141 10.099
6    5.243 0.908       0.530     0.147       0.096 1.992  8.520  8.512
   price    ps   ps1   revenue    sps  tbvps   returns  logreturns
1 32.940 1.977 1.988 1.519e+09 16.571 17.663 0.0000000          NA
2 34.120 1.929 1.923 1.677e+09 17.741 19.023 1.0358227  0.03519600
3 25.529 1.372 1.375 1.691e+09 18.572 19.284 0.7482122 -0.29006866
```

%r FINISHED

```
# Normalizing all 20 factors

bvps_N <- (market_data.factors.noNA[,3] - mean(market_data.factors.noNA[,3]))/sd(market_da

currentratio_N <- (market_data.factors.noNA[,4] - mean(market_data.factors.noNA[,4]))/sd(m

de_N <- (market_data.factors.noNA[,5] - mean(market_data.factors.noNA[,5]))/sd(market_data

divyield_N <- (market_data.factors.noNA[,6] - mean(market_data.factors.noNA[,6]))/sd(marke

ebitdamargin_N <- (market_data.factors.noNA[,7] - mean(market_data.factors.noNA[,7]))/sd(m

eps_N <- (market_data.factors.noNA[,8] - mean(market_data.factors.noNA[,8]))/sd(market_dat

evebitda_N <- (market_data.factors.noNA[,9] - mean(market_data.factors.noNA[,9]))/sd(marke

fcfps_N <- (market_data.factors.noNA[,10] - mean(market_data.factors.noNA[,10]))/sd(market_

grossmargin_N <- (market_data.factors.noNA[,11] - mean(market_data.factors.noNA[,11]))/sd(

netmargin_N <- (market_data.factors.noNA[,12] - mean(market_data.factors.noNA[,12]))/sd(ma

payoutratio_N <- (market_data.factors.noNA[,13] - mean(market_data.factors.noNA[,13]))/sd(

pb_N <- (market_data.factors.noNA[,14] - mean(market_data.factors.noNA[,14]))/sd(market_da

pe_N <- (market_data.factors.noNA[,15] - mean(market_data.factors.noNA[,15]))/sd(market_da

pe1_N <- (market_data.factors.noNA[,16] - mean(market_data.factors.noNA[,16]))/sd(market_d

price_N <- (market_data.factors.noNA[,17] - mean(market_data.factors.noNA[,17]))/sd(market_

ps_N <- (market_data.factors.noNA[,18] - mean(market_data.factors.noNA[,18]))/sd(market_da

ps1_N <- (market_data.factors.noNA[,19] - mean(market_data.factors.noNA[,19]))/sd(market_d

revenue_N <- (market_data.factors.noNA[,20] - mean(market_data.factors.noNA[,20]))/sd(mark

sps_N <- (market_data.factors.noNA[,21] - mean(market_data.factors.noNA[,21]))/sd(market_d

tbvps_N <- (market_data.factors.noNA[,22] - mean(market_data.factors.noNA[,22]))/sd(market_
```

%r FINISHED

```
# creating a data frame of all the normalized 20 factors
market_data.factors.Norm <- data.frame(bvps_N, currentratio_N, de_N, divyield_N, ebitdamarg
    payoutratio_N, pb_N, pe_N, pe1_N, price_N, ps_N, ps1_N, revenue_N, sps_N, tbvps_N)

# cbinding columns 'calendardate', 'returns', and 'logreturns' to the normalized factors.
calendardate <- market_data.factors.noNA$calendardate
returns <- market_data.factors.noNA$returns
logreturns <- market_data.factors.noNA$logreturns
```

```
ticker <- market_data.factors.noNA$ticker

market_data.factors.Norm_complete <- cbind(calendardate, market_data.factors.Norm, returns
head(market_data.factors.Norm_complete)
```

```
  calendardate       bvps_N currentratio_N         de_N  divyield_N
1   2011-06-30 -0.002459223     0.05235115 0.0017163731 -0.09739857
2   2011-09-30  0.006320483     0.11527403 0.0009796611 -0.09739857
3   2011-12-31  0.011274170     0.01281968 0.0010901679 -0.09739857
4   2012-03-31  0.018053607     0.08767217 0.0005155326 -0.09739857
5   2012-06-30  0.026766189     0.06966079 0.0002945190 -0.07909305
6   2012-09-30  0.032471656    -0.22507090 0.0003681902 -0.06688936
  ebitdamargin_N        eps_N  evebitda_N     fcfps_N grossmargin_N
1    0.008747055 -0.003881792 0.004304101 0.018521000    0.03571064
2    0.008748785 -0.003630736 0.003810672 0.011472763    0.03331195
3    0.008749477 -0.003705374 0.003651093 0.027319929    0.03475116
4    0.008747920 -0.003827510 0.003889819 0.003901592    0.03523090
5    0.008749477 -0.003780013 0.003677537 0.017747968    0.03403156
6    0.008746363 -0.003800369 0.003909835 0.009585654    0.03091326
  netmargin_N payoutratio_N          pb_N          pe_N         pe1_N
1 0.008862848   -0.04068744 -0.0009749304  0.0039505328  6.077121e-03
2 0.008875600   -0.04068744 -0.0041137844  0.0001323263  3.477839e-04
3 0.008870902   -0.04068744 -0.0047528537 -0.0002817959 -3.001097e-04
```

```
%r                                                                    FINISHED
# finding out how many different date are there in the data
date <- split(market_data.factors.Norm_complete, as.factor(market_data.factors.Norm_comple
length(date) # there are 20 different calendar dates

# extracting all data for a particular calendar date and assign each to a dataframe
funct3 <- split(market_data.factors.Norm_complete, market_data.factors.Norm_complete$calen
factor_names <- c("n.market_data.factors_2011_03_31", "n.market_data.factors_2011_06_30", "
    .market_data.factors_2012_03_31", "n.market_data.factors_2012_06_30", "n.market_data.fa
    .factors_2013_03_31", "n.market_data.factors_2013_06_30", "n.market_data.factors_2013_0
    .factors_2014_03_31", "n.market_data.factors_2014_06_30", "n.market_data.factors_2014_0
    .factors_2015_03_31", "n.market_data.factors_2015_06_30", "n.market_data.factors_2015_0

for (i in 1:length(funct3)) {
  assign(factor_names[i], funct3[[i]])
}

# Verifying data for few calendar dates
head(n.market_data.factors_2011_09_30, 2)
head(n.market_data.factors_2013_03_31, 2)
head(n.market_data.factors_2011_12_31, 2)
head(n.market_data.factors_2014_06_30, 2)
head(n.market_data.factors_2015_09_30, 2)

head(n.market_data.factors_2011_06_30)

head(n.market_data.factors_2011_03_31)
```

```
[1] 20
   calendardate      bvps_N currentratio_N         de_N  divyield_N
2    2011-09-30 0.006320483      0.1152740 0.0009796611 -0.09739857
21   2011-09-30 0.032793847     -0.3773489 0.0035065832 -0.02417646
   ebitdamargin_N        eps_N  evebitda_N      fcfps_N grossmargin_N
2     0.008748785 -0.003630736 0.003810672 0.0114727627    0.03331195
21    0.008731313 -0.004166774 0.003704347 0.0006048354   -0.05088193
   netmargin_N payoutratio_N         pb_N         pe_N        pe1_N
2  0.008875600  -0.040687439 -0.004113784 0.0001323263 0.0003477839
21 0.008847411  -0.001323075 -0.009074516 0.0011734294 0.0017766653
       price_N        ps_N       ps1_N revenue_N       sps_N     tbvps_N
2  -0.06561558 -0.01136768 -0.01112355  0.144882 -0.03191092 -0.04735782
21 -0.26310794 -0.01146580 -0.01122471  1.218632 -0.02520986  0.04456954
     returns logreturns
2  0.7482122 -0.2900687
21 0.6309448 -0.4605368
   calendardate     bvps_N currentratio_N         de_N   divyield_N
8    2013-03-31 0.05150778    -0.09922515 0.0002797848 -0.042481986
```

```
%r                                                                    FINISHED
# Using Neuralnet on logreturns on the 20 factors, for 75% of the dates.
# I am using the first three quaters of years 2011 to 2015.

install.packages('neuralnet', repos = 'http://cran.us.r-project.org')
library(neuralnet)
```

```
The downloaded binary packages are in
    /var/folders/7b/jk0vn7m91g599255yn29jd500000gn/T//Rtmp8EWqL7/downloaded_packages
```

```
%r                                                                    FINISHED
names(n.market_data.factors_2011_03_31)

# creating a formula to be used with neuralnet
colnames <- names(n.market_data.factors_2011_03_31)
# colnames %in% c("calendardate", "returns","logreturns")
# !colnames %in% c("calendardate", "returns","logreturns")
# paste(colnames[!colnames %in% c("calendardate", "returns","logreturns")])
# paste(colnames[!colnames %in% c("calendardate", "returns","logreturns")], collapse = " +
# paste("logreturns ~", paste(colnames[!colnames %in% c("calendardate", "returns","logretu

nn.formula <- as.formula(paste("logreturns ~", paste(colnames[!colnames %in% c("calendarda
nn.formula
```

```
[1] “calendardate”   “bvps_N”         “currentratio_N” “de_N”
 [5] “divyield_N”     “ebitdamargin_N” “eps_N”          “evebitda_N”
 [9] “fcfps_N”        “grossmargin_N”  “netmargin_N”    “payoutratio_N”
[13] “pb_N”           “pe_N”           “pe1_N”          “price_N”
[17] “ps_N”           “ps1_N”          “revenue_N”      “sps_N”
[21] “tbvps_N”        “returns”        “logreturns”
logreturns ~ bvps_N + currentratio_N + de_N + divyield_N + ebitdamargin_N +
    eps_N + evebitda_N + fcfps_N + grossmargin_N + netmargin_N +
    payoutratio_N + pb_N + pe_N + pe1_N + price_N + ps_N + ps1_N +
    revenue_N + sps_N + tbvps_N
```

```
%r                                                                    FINISHED
# For 2011
nn.model_2011_03_31 <- neuralnet(nn.formula, data=n.market_data.factors_2011_03_31, hidden=
plot(nn.model_2011_03_31)

weights_nn.model_2011_03_31 <- nn.model_2011_03_31$result.matrix

View(weights_nn.model_2011_03_31)
```

```
%r                                                                    READY
```