DS-670 Assignment 9: Mohamed Mohamar

My chunk of code has about 120 lines

The R loops take more time to run. To optimize them, I will use vectorization instead R loops.

Codes	Elapsed Time (in Seconds)

# 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))
	27.83 sec elapsed

# selecting my 20 factors market_data.factors <- market_data[c(1,3,12,17,18,22,26,30,38,40,43,65,67,68,69,70,72,73,74,77,89,92,94)] head(market_data.factors, 2) t(names(market_data.factors))	0.01 sec elapsed

# 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))
	22.325 sec elapsed

# 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)	2.287 sec elapsed

```
# Normalizing all 20 factors

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

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

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

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

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

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

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

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

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

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

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

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

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

```
0.04 sec elapsed
```

```
# creating a data frame of all the normalized 20 factors
market_data.factors.Norm <- data.frame(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)

# 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, logreturns)
head(market_data.factors.Norm_complete)
```

```
0.013 sec elapsed
```

```
# finding out how many different date are there in the data
date <- split(market_data.factors.Norm_complete, as.factor(market_data.factors.Norm_complete$calendardate))
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$calendardate)
factor_names <- c("n.market_data.factors_2011_03_31",
"n.market_data.factors_2011_06_30",
"n.market_data.factors_2011_09_30",
"n.market_data.factors_2011_12_31",
"n.market_data.factors_2012_03_31",
"n.market_data.factors_2012_06_30",
"n.market_data.factors_2012_09_30",
"n.market_data.factors_2012_12_31",
"n.market_data.factors_2013_03_31",
"n.market_data.factors_2013_06_30",
"n.market_data.factors_2013_09_30",
"n.market_data.factors_2013_12_31",
"n.market_data.factors_2014_03_31",
"n.market_data.factors_2014_06_30",
"n.market_data.factors_2014_09_30",
"n.market_data.factors_2014_12_31",
"n.market_data.factors_2015_03_31",
"n.market_data.factors_2015_06_30",
"n.market_data.factors_2015_09_30",
"n.market_data.factors_2015_12_31")

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)
```

```
0.101 sec elapsed
```

names(n.market_data.factors_2011_03_31) nn.formula <- as.formula(paste("logreturns ~", paste(colnames[!colnames %in% c("calendardate", "returns","logreturns")], collapse = " + "))) nn.formula	0.008 sec elapsed

# For 2011 	
nn.model_2011_03_31 <- neuralnet(nn.formula, data=n.market_data.factors_2011_03_31, hidden=c(7,7), linear.output = T)
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)
	6.605 sec elapsed