1. Data Preparation

FINISHED

```
# Here I have calculated returns from price and normalization of selected indicator data is done.
```

Took 0 sec. Last updated by anonymous at March 25 2017, 8:38:20 AM. (outdated)

FINISHED

```
%spark.r

small_cap_stock <- read.csv("/home/scarface/Desktop/sem-3/capstone/madhu/stockPerform/iwm.csv
large_cap_stock <- read.csv("/home/scarface/Desktop/sem-3/capstone/madhu/stockPerform/spy.csv

View(small_cap_stock)
View(large_cap_stock)
typeof(large_cap_stock$Date)
```

```
[1] “integer”
```

Took 5 sec. Last updated by anonymous at April 15 2017, 10:11:00 PM.

FINISHED

```
%spark.r
large_cap_stock2000 <- subset(large_cap_stock, large_cap_stock$Date >= "2000-01-01")
large_cap_stock2000
```

```
[1] Date      Open      High      Low       Close     Volume    Adj.Close
&lt;0 rows&gt; (or 0-length row.names)
```

Took 0 sec. Last updated by anonymous at April 15 2017, 10:13:48 PM.

FINISHED

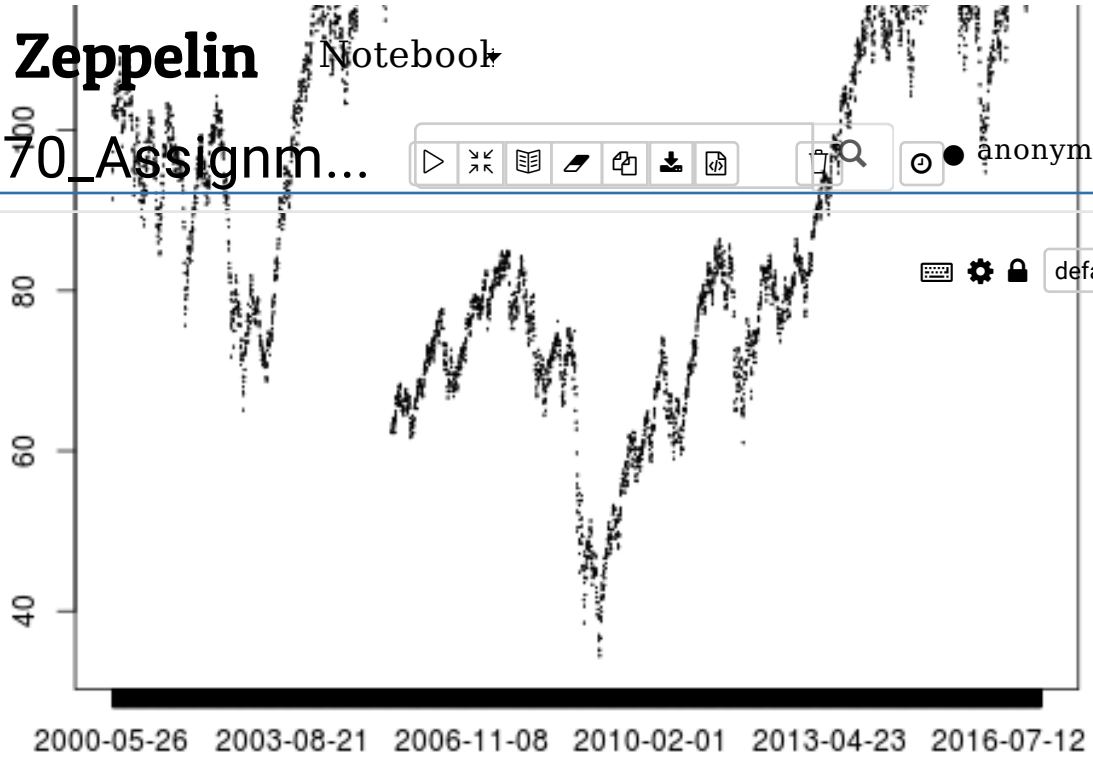
```
%spark.r
plot(small_cap_stock$Date, small_cap_stock$'Close')
```

Zeppelin Notebook

DS-670_Assignm...

anonymou

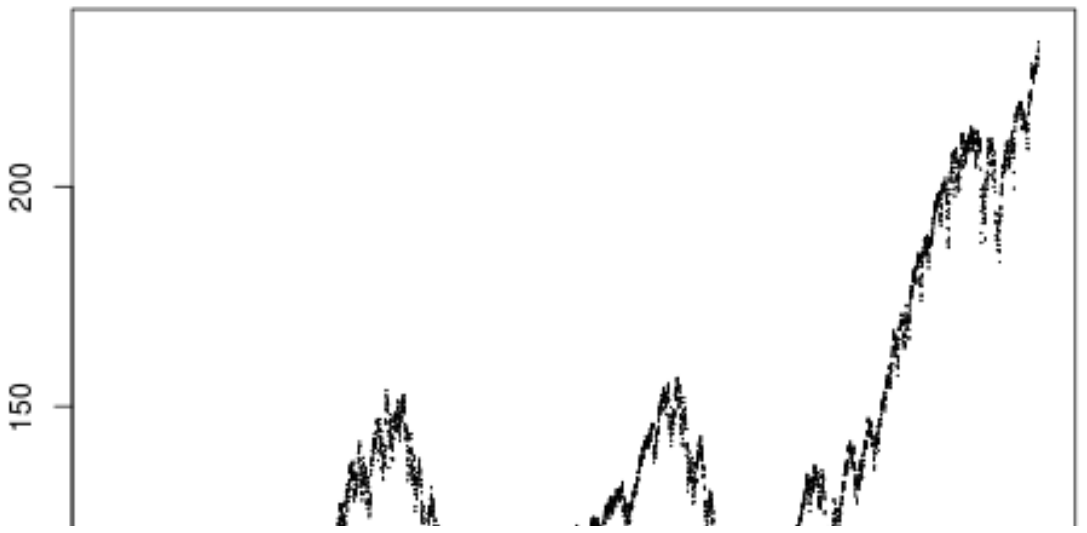
default



Took 3 sec. Last updated by anonymous at April 15 2017, 10:17:04 PM.

FINISHED

```
%spark.r
plot(large_cap_stock$Date, large_cap_stock$'Close')
```

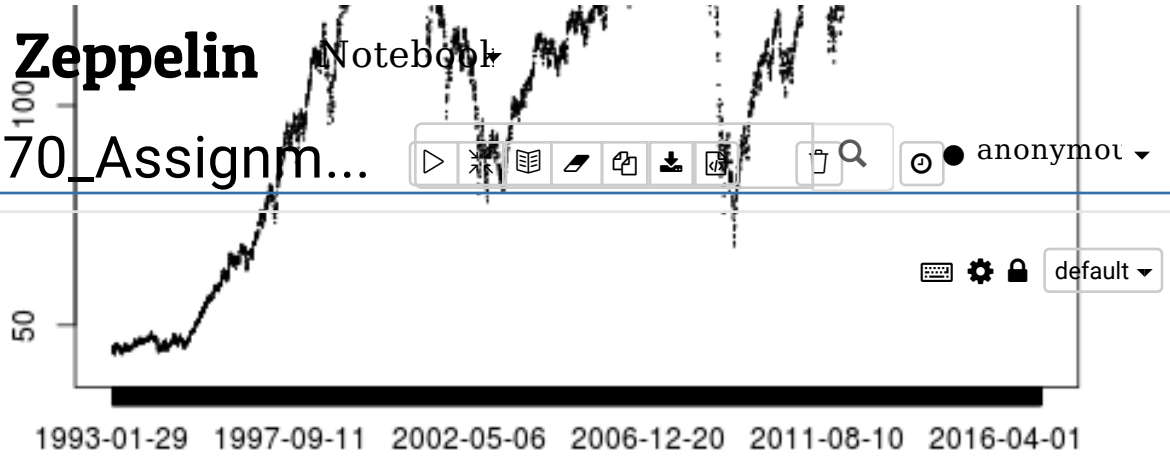


Zeppelin Notebook

DS-670_Assignm...

anonymou

default



Took 5 sec. Last updated by anonymous at April 15 2017, 10:17:51 PM.

FINISHED

```
1 %spark.r
2 arqdata <- read.csv(file = "/home/scarface/Desktop/sem-3/capstone/madhu/ARQ_Data.csv", header=
3 save(arqdata, file="arq.rdata")
4 load("arq.rdata")
5 nrow(arqdata)
```

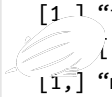
```
[1] 89206
```

Took 18 sec. Last updated by anonymous at April 15 2017, 10:22:40 PM.

FINISHED

```
1 %spark.r
2 #View(arqdata)
3 t(names(arqdata))
4 arq[72]
5 #Considering the column price with data
6 arqdata_na <- subset(arqdata, arqdata$price != "NA")
7 #View(arqdata_na)
```

```
[1,] “ticker” “dimension” “calendardate” “datekey” “reportperiod” “accoci”
     [,7]     [,8]        [,9]      [,10]      [,11]           [,12]
[1,] “assets” “assetsavg” “assetsc” “assetsnc” “assetturnover” “bvps”
     [,13]   [,14]     [,15]        [,16] [,17]          [,18] [,19]
[1,] “capex” “cashneq” “cashnequsd” “cor” “currentratio” “de”  “debt”
     [,20]     [,21]     [,22]      [,23] [,24]  [,25]    [,26]<br />
[1,] “debtusd” “depamor” “divyield” “dps” “ebit” “ebitda” “ebitdamargin”
     [,27]       [,28]     [,29] [,30] [,31]    [,32]    [,33]<br />
[1,] “ebitdausd” “ebitusd” “ebt” “eps” “epsdil” “epsusd” “equity”
     [,34]       [,35]       [,36] [,37]    [,38]      [,39] [,40]<br />
[1,] “equityavg” “equityusd” “ev”  “evebit” “evebitda” “fcf” “fcfps”
     [,41]   [,42] [,43]         [,44]         [,45]    [,46]<br />
[1,] “fxusd” “gp”  “grossmargin” “intangibles” “intexp” “invcap”
     [,47]       [,48]       [,49]         [,50]          [,51]<br />
```



Zeppelin Notebook

DS-670_Assignm...

anonymous

default

```
[1,] “invcapavg” “inventory” “liabilities” “liabilitiesc” “liabilitiesnc”
     [,5]  [,6]  [,7]  [,8]  [,9]  [,10] [,11] [,12] [,13] [,14] [,15]
[1,] “marketcap” “ncf” “ncfcommon” “ncfdebt” “ncfdiv” “ncff” “ncfi” “ncfo”
     [,60]  [,61]   [,62]      [,63]      [,64]      [,65]<br />
[1,] “ncfx” “netinc” “netinccmn” “netinccmnusd” “netincnci” “netmargin”
     [,66]     [,67]      [,68] [,69] [,70] [,71]      [,72]   [,73]
[1,] “payables” “payoutratio” “pb”  “pe”  “pe1” “prefdivis” “price” “ps”
     [,74] [,75]        [,76]     [,77]     [,78]       [,79] [,80]
[1,] “ps1” “receivables” “retearn” “revenue” “revenueusd” “rnd” “roa”
     [,81] [,82]  [,83] [,84]  [,85]        [,86]       [,87]<br />
[1,] “roe” “roic” “ros” “sgna” “sharefactor” “sharesbas” “shareswa”
     [,88]        [,89] [,90]       [,91]    [,92]   [,93]<br />
[1,] “shareswadil” “sps” “tangibles” “taxexp” “tbvps” “workingcapital”
```

Took 1 sec. Last updated by anonymous at April 15 2017, 10:24:19 PM.

FINISHED

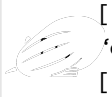
```
%spark.r
###################################Data_File - with ARQ listings#############################
return <- vector();
for (i in 2:length(arqdata_na[,1]))
{
  if (identical(arqdata_na[i,1],arqdata_na[i-1,1]))
  {
    #    return[i] = ((arqdata[i,72] / arqdata[i-1,72]) - 1);
    return[i] = (arqdata_na[i,72] / arqdata_na[i-1,72]);
  }
  else
  {
    return[i] = 0;
  }
}

return[1]=0;

```

Took 17 sec. Last updated by anonymous at April 15 2017, 10:24:42 PM.

FINISHED

```
%spark.r
#adding return to arqdata dataset
arqdata_returns <- cbind(arqdata_na,return)
#View(arqdata_returns)
t(names(arqdata_returns))
#View(arqdata_returns)
```

```
[1,] “ticker” “dimension” “calendardate” “datekey” “reportperiod” “accoci”
     [,7]     [,8]        [,9]      [,10]      [,11]           [,12]
[1,] “assets” “assetsavg” “assetsc” “assetsnc” “assetturnover” “bvps”
     [,13]   [,14]     [,15]        [,16] [,17]          [,18] [,19]
[1,] “capex” “cashneq” “cashnequsd” “cor” “currentratio” “de”  “debt”
     [,20]     [,21]     [,22]      [,23] [,24]  [,25]    [,26]<br />
[1,] “debtusd” “depamor” “divyield” “dps” “ebit” “ebitda” “ebitdamargin”
     [,27]       [,28]     [,29] [,30] [,31]    [,32]    [,33]<br />
[1,] “ebitdausd” “ebitusd” “ebt” “eps” “epsdil” “epsusd” “equity”
```



Zeppelin Notebook

DS-670_Assignm...

anonymous

default

```
     [,34]       [,35]       [,36] [,37]    [,38]      [,39] [,40]<br />
[1,] “equityavg” “equityusd” “ev”  “evebit” “evebitda” “fcf” “fcfps”
     [,41]   [,42] [,43]        [,44]        [,45]    [,46]<br />
[1,] “fxusd” “gp”  “grossmargin” “intangibles” “intexp” “invcap”
     [,47]       [,48]       [,49]        [,50]        [,51]<br />
[1,] “invcapavg” “inventory” “liabilities” “liabilitiesc” “liabilitiesnc”
     [,52]       [,53] [,54]       [,55]     [,56]    [,57]  [,58]  [,59]
[1,] “marketcap” “ncf” “ncfcommon” “ncfdebt” “ncfdiv” “ncff” “ncfi” “ncfo”
     [,60]  [,61]    [,62]       [,63]          [,64]       [,65]<br />
[1,] “ncfx” “netinc” “netinccmn” “netinccmnusd” “netincdis” “netmargin”
     [,66]      [,67]         [,68] [,69] [,70] [,71]       [,72]   [,73]
[1,] “payables” “payoutratio” “pb”  “pe”  “pe1” “prefdivis” “price” “ps”
     [,74] [,75]         [,76]     [,77]     [,78]        [,79] [,80]
[1,] “ps1” “receivables” “retearn” “revenue” “revenueusd” “rnd” “roa”
     [,81] [,82]  [,83] [,84]  [,85]         [,86]       [,87]<br />
[1,] “roe” “roic” “ros” “sgna” “sharefactor” “sharesbas” “shareswa”
     [,88]         [,89] [,90]       [,91]    [,92]   [,93]<br />
```

Took 0 sec. Last updated by anonymous at April 15 2017, 10:25:11 PM.

FINISHED

```
1  %spark.r
2  #Indicators listed.
3  #Calculating ratios by choice 1.SGNA/REVENUE (sgnamargin) and 2.ebitmargin
4  sgnamargin = arqdata_returns$sgna / arqdata_returns$revenue
5  ebitmargin = arqdata_returns$ebit / arqdata_returns$revenue
6
7  #adding ratios by choice to dataset - arqdata_returns
8  arqdata_returns_ratios <- cbind(arqdata_returns, sgnamargin,ebitmargin)
9  #View(arqdata_returns_ratios)
10 t(names(arqdata_returns_ratios))
11
```

```
[1,] “ticker” “dimension” “calendardate” “datekey” “reportperiod” “accoci”
     [,7]     [,8]        [,9]      [,10]      [,11]           [,12]
[1,] “assets” “assetsavg” “assetsc” “assetsnc” “assetturnover” “bvps”
     [,13]   [,14]     [,15]        [,16] [,17]          [,18] [,19]
[1,] “capex” “cashneq” “cashnequsd” “cor” “currentratio” “de”  “debt”
     [,20]     [,21]     [,22]      [,23] [,24]  [,25]    [,26]<br />
[1,] “debtusd” “depamor” “divyield” “dps” “ebit” “ebitda” “ebitdamargin”
     [,27]       [,28]     [,29] [,30] [,31]    [,32]    [,33]<br />
[1,] “ebitdausd” “ebitusd” “ebt” “eps” “epsdil” “epsusd” “equity”
     [,34]       [,35]       [,36] [,37]    [,38]      [,39] [,40]<br />
[1,] “equityavg” “equityusd” “ev”  “evebit” “evebitda” “fcf” “fcfps”
     [,41]   [,42] [,43]        [,44]        [,45]    [,46]<br />
[1,] “fxusd” “gp”  “grossmargin” “intangibles” “intexp” “invcap”
     [,47]       [,48]       [,49]        [,50]          [,51]<br />
[1,] “invcapavg” “inventory” “liabilities” “liabilitiesc” “liabilitiesnc”
     [,52]       [,53] [,54]       [,55]     [,56]    [,57]  [,58]  [,59]
[1,] “marketcap” “ncf” “ncfcommon” “ncfdebt” “ncfdiv” “ncff” “ncfi” “ncfo”
     [,60]  [,61]    [,62]       [,63]          [,64]       [,65]<br />
[1,] “ncfx” “netinc” “netinccmn” “netinccmnusd” “netincdis” “netmargin”
```

```
     [,66]      [,67]          [,68] [,69] [,70] [,71]       [,72]   [,73]
[1,] "pa"  "payables" "payoutratio" "pb" "pe" "peg" "prefdivis" "price" "ps"
     [,74] [,75]          [,76]     [,77]     [,78]        [,79] [,80]
[1,] "ps1" "receivables" "retearn" "revenue" "revenueusd" "rnd" "roa"
     [,81] [,82]          [,83]     [,84]     [,85]
[1,] "roe" "roic" "ros" "sgna" "sharefactor" "sharesbas" "shareswa"
     [,88]         [,89] [,90]       [,91]    [,92]   [,93]<br />
[1,] "shareswadil" "sps" "tangibles" "taxexp" "tbvps" "workingcapital"
     [,94]    [,95]        [,96]<br />
[1,] "return" "sgnamargin" "ebitmargin"
```

Took 0 sec. Last updated by anonymous at April 15 2017, 10:25:18 PM.

FINISHED

```
1  %spark.r
2  #Consider the 20 indicators chosen
3  #factors required in the dataset
4  arq_data_factors <- arqdata_returns_ratios[c(1,3,77,16,42,84,24,43,95,96,45,91,61,29,30,65,7,9,
5  head(arq_data_factors)
6
7  #17 lines
```

```
ticker calendardate   revenue      cor       gp     sgna     ebit
1      A   31-03-2011 1.519e+09 7.03e+08 8.16e+08 4.46e+08 2.21e+08
2      A   30-06-2011 1.677e+09 7.77e+08 9.00e+08 4.69e+08 2.80e+08
3      A   30-09-2011 1.691e+09 7.99e+08 8.92e+08 4.49e+08 3.01e+08
4      A   31-12-2011 1.728e+09 8.07e+08 9.21e+08 4.45e+08 3.16e+08
5      A   31-03-2012 1.635e+09 7.61e+08 8.74e+08 4.41e+08 2.82e+08
6      A   30-06-2012 1.733e+09 8.15e+08 9.18e+08 4.52e+08 3.18e+08
  grossmargin sgnamargin ebitmargin  intexp   taxexp   netinc      ebt
1       0.537  0.2936142  0.1454905 2.3e+07  5.0e+06 1.93e+08 1.98e+08
2       0.537  0.2796661  0.1669648 2.0e+07  6.0e+07 2.00e+08 2.60e+08
3       0.527  0.2655234  0.1780012 2.0e+07 -4.9e+07 3.30e+08 2.81e+08
4       0.533  0.2575231  0.1828704 2.3e+07  4.0e+06 2.89e+08 2.93e+08
5       0.535  0.2697248  0.1724771 2.6e+07  2.6e+07 2.30e+08 2.56e+08
6       0.530  0.2608194  0.1834968 2.5e+07  3.8e+07 2.55e+08 2.93e+08
   eps netmargin    assets   assetsc liabilitiesc currentratio
1 0.56     0.127 8.044e+09 4.598e+09    1.406e+09        3.270
2 0.58     0.119 8.649e+09 5.096e+09    1.592e+09        3.201
```

Took 0 sec. Last updated by anonymous at April 15 2017, 10:25:23 PM.

FINISHED

```
1  %spark.r
2  caldate = unique(arq_data_factors$calendardate)
3  length(caldate)
4  prj2_arq_date = vector();
5  prj2_arq_nn = vector();
6  prj2_date_replace = vector();
7  factors1 <- NULL
8  factors2 <- NULL
9  factors3 <- NULL
10 #35 lines
```

[1] 20

Zeppelin Notebook

took 0 sec. Last updated by anonymous at April 15 2017, 10:25:32 PM.

DS-670_Assignm...

FINISHED

anonymou

default

```
%spark.r
# Loop for dates - each date we get a dataset prj2_arq_'date'
for (i in 1:length(caldate)){
  #if (i < 16)
  #{
    #print(paste0("calendar date: ", caldate[i]))
    factors1 <- subset(arq_data_factors, calendardate == caldate[i])

    ### Calculating log(returns) ###
    factors1 <- subset(factors1, factors1$return != 0)
    return_log <- log(factors1$return)
    factors1 <- cbind(factors1, return_log)

    ### Remove all NAs in our dataset before normalizing
    factors1 <- na.omit(factors1)

    ### Normalizing all Indicators
    revenue_nor <- (factors1[,3] - mean(factors1[,3])) / sd(factors1[,3])
    cor_nor <- (factors1[,4] - mean(factors1[,4])) / sd(factors1[,4])
    gp_nor <- (factors1[,5] - mean(factors1[,5])) / sd(factors1[,5])
    sgna_nor <- (factors1[,6] - mean(factors1[,6])) / sd(factors1[,6])
    ebit_nor <- (factors1[,7] - mean(factors1[,7])) / sd(factors1[,7])
    gm_nor <- (factors1[,8] - mean(factors1[,8])) / sd(factors1[,8])
    sgna_mg_nor <- (factors1[,9] - mean(factors1[,9])) / sd(factors1[,9])
    ebit_mg_nor <- (factors1[,10] - mean(factors1[,10])) / sd(factors1[,10])
    intexp_nor <- (factors1[,11] - mean(factors1[,11])) / sd(factors1[,11])
    taxexp_nor <- (factors1[,12] - mean(factors1[,12])) / sd(factors1[,12])
    netinc_nor <- (factors1[,13] - mean(factors1[,13])) / sd(factors1[,13])
    ebt_nor <- (factors1[,14] - mean(factors1[,14])) / sd(factors1[,14])
    eps_nor <- (factors1[,15] - mean(factors1[,15])) / sd(factors1[,15])
    netmargin_nor <- (factors1[,16] - mean(factors1[,16])) / sd(factors1[,16])
    assets_nor <- (factors1[,17] - mean(factors1[,17])) / sd(factors1[,17])
    assetsc_nor <- (factors1[,18] - mean(factors1[,18])) / sd(factors1[,18])
    liabc_nor <- (factors1[,19] - mean(factors1[,19])) / sd(factors1[,19])
    cur_ratio_nor <- (factors1[,20] - mean(factors1[,20])) / sd(factors1[,20])
    wc_nor <- (factors1[,21] - mean(factors1[,21])) / sd(factors1[,21])
    capex_nor <- (factors1[,22] - mean(factors1[,22])) / sd(factors1[,22])

    ### Appending normalized columns to new factors2

    factors1 <- cbind(factors1, revenue_nor,cor_nor,gp_nor,sgna_nor,ebit_nor,gm_nor,sgna_mg_
                      intexp_nor,taxexp_nor,netinc_nor,ebt_nor,eps_nor,netmargin_nor,assets_n
                      liabc_nor,cur_ratio_nor,wc_nor,capex_nor)
    factors2 <- factors1[c(1,2,25:44,23,24)]
    factors3 <- factors1[c(25:44,24)]

    prj2_date_replace[i] <- gsub("-", "_", caldate[i])
    prj2_arq_date[i] <- paste("prj2_arq_", prj2_date_replace[i], sep = "")
    assign(prj2_arq_date[i], factors2)

    prj2_date_replace[i] <- gsub("-", "_", caldate[i])
    prj2_arq_nn[i] <- paste("prj2_arqnn_", prj2_date_replace[i], sep = "")
    assign(prj2_arq_nn[i], factors3)

    factors1 <- NULL
    factors2 <- NULL
```

Zeppelin Notebook

DS-670_Assignm...

anonymou

Took 1 sec. Last updated by anonymous at April 15 2017, 10:25:46 PM.

default

FINISHED

```
1 %spark.r
2 # Datasets names
3 head(prj2_arq_date)
4 head(prj2_arq_nn)
5
6 #head(get(prj2_arq_nn[1])
7 #get(prj2_arq_nn[1])
8
```

```
[1] “prj2_arq_31_03_2011” “prj2_arq_30_06_2011” “prj2_arq_30_09_2011”
[4] “prj2_arq_31_12_2011” “prj2_arq_31_03_2012” “prj2_arq_30_06_2012”
[1] “prj2_arqnn_31_03_2011” “prj2_arqnn_30_06_2011” “prj2_arqnn_30_09_2011”
[4] “prj2_arqnn_31_12_2011” “prj2_arqnn_31_03_2012” “prj2_arqnn_30_06_2012”
```

Took 0 sec. Last updated by anonymous at April 15 2017, 10:25:49 PM.

2. Neural Network modeling

FINISHED

```
# For every quarter neural network model was run and weights from each model were gathered in a matr
```

Took 36 sec. Last updated by anonymous at March 26 2017, 8:02:42 AM. (outdated)

FINISHED

```
1 %spark.r
2 install.packages("neuralnet", repos = "http://cran.us.r-project.org")
3 install.packages("MASS", repos = "http://cran.us.r-project.org")
4
```

```
The downloaded source packages are in
    '/tmp/Rtmpbds3G9/downloaded_packages'
```

Took 18 sec. Last updated by anonymous at April 15 2017, 10:26:12 PM.

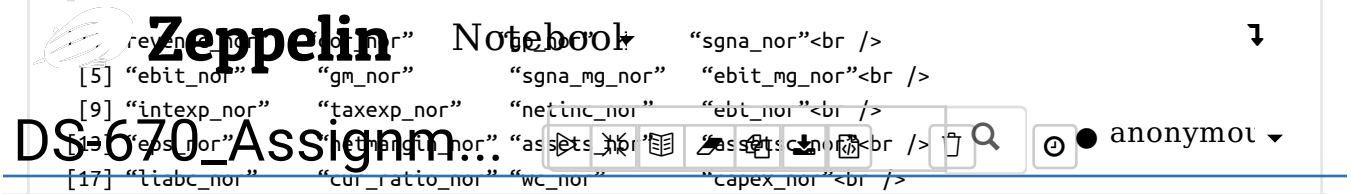
FINISHED

```
1 %spark.r
2 library("MASS")
3 library("neuralnet")
```

Took 0 sec. Last updated by anonymous at April 15 2017, 10:26:23 PM.

FINISHED

```
1 %spark.r
2 ###################### neural network - package does not take Strings!!
3 names_date <- names(get(prj2_arq_nn[1]))
4 names_date
```




```
 [1] “revenue_nor”   “cor_nor”       “gp_nor”        “sgna_nor”<br />
 [5] “ebit_nor”      “gm_nor”        “sgna_mg_nor”   “ebit_mg_nor”<br />
 [9] “intexp_nor”    “taxexp_nor”    “netinc_nor”    “ebt_nor”<br />
[13] “eps_nor”       “netmargin_nor” “assets_nor”    “assetsc_nor”<br />
[17] “liabc_nor”     “cur_ratio_nor” “wc_nor”        “capex_nor”<br />
[21] “return_log”
```

Took 0 sec. Last updated by anonymous at April 15 2017, 10:26:26 PM.

default

FINISHED

```
%spark.r
### log(returns) as y .... we need to get the formula to use in nueral networks as it does not
names_date %in% "return_log"
!names_date %in% "return_log"
paste(names_date[!names_date %in% "return_log"])
paste(names_date[!names_date %in% "return_log"], collapse = "+")
paste("return_log ~ ", paste(names_date[!names_date %in% "return_log"], collapse = "+"))
formula1 <- as.formula(paste("return_log ~ ", paste(names_date[!names_date %in% "return_log"],
formula1

```

```
[1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
 [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
 [1] “revenue_nor”   “cor_nor”       “gp_nor”        “sgna_nor”<br />
 [5] “ebit_nor”      “gm_nor”        “sgna_mg_nor”   “ebit_mg_nor”<br />
 [9] “intexp_nor”    “taxexp_nor”    “netinc_nor”    “ebt_nor”<br />
[13] “eps_nor”       “netmargin_nor” “assets_nor”    “assetsc_nor”<br />
[17] “liabc_nor”     “cur_ratio_nor” “wc_nor”        “capex_nor”<br />
[1] “revenue_nor+cor_nor+gp_nor+sgna_nor+ebit_nor+gm_nor+sgna_mg_nor+ebit_mg_nor+intexp_nor+taxexp_no
r+netinc_nor+ebt_nor+eps_nor+netmargin_nor+assets_nor+assetsc_nor+liabc_nor+cur_ratio_nor+wc_nor+cape
x_nor”
[1] “return_log ~  revenue_nor+cor_nor+gp_nor+sgna_nor+ebit_nor+gm_nor+sgna_mg_nor+ebit_mg_nor+intexp
_nor+taxexp_nor+netinc_nor+ebt_nor+eps_nor+netmargin_nor+assets_nor+assetsc_nor+liabc_nor+cur_ratio_n
or+wc_nor+capex_nor”
return_log ~ revenue_nor + cor_nor + gp_nor + sgna_nor + ebit_nor +
    gm_nor + sgna_mg_nor + ebit_mg_nor + intexp_nor + taxexp_nor +
```

Took 0 sec. Last updated by anonymous at March 25 2017, 5:07:11 AM. (outdated)

FINISHED

```
%spark.r

names_date1 <- names(get(prj2_arq_nn[1])[,1:4])
names_date1
formula2 <- as.formula(paste("return_log ~ ", paste(names_date1[!names_date1 %in% "return_log"], cc
formula2
```

```
[1] “revenue_nor” “cor_nor”     “gp_nor”      “sgna_nor”<br />
return_log ~ revenue_nor + cor_nor + gp_nor + sgna_nor
```

Took 0 sec. Last updated by anonymous at April 15 2017, 10:28:30 PM.

Zeppelin Notebook

DS-670_Assignm...

anonymou

default

```
%spark.r

############## Assigning datasets - from date 15 to date 20
uvw1 <- get(prj2_arq_nn[15])
uvw2 <- get(prj2_arq_nn[16])
uvw3 <- get(prj2_arq_nn[17])
uvw4 <- get(prj2_arq_nn[18])
uvw5 <- get(prj2_arq_nn[19])
uvw6 <- get(prj2_arq_nn[20])
head(get(prj2_arq_date[17]))
#94 lines
```

```
17       A   31-03-2015  0.01439033 -0.04207989  0.1390798  0.20840226
38      AA   31-03-2015  1.20665857  1.26814971  0.7906741  0.09887883
44     AAC   31-03-2015 -0.23017689 -0.21310986 -0.2159198 -0.19253635
83     AAL   31-03-2015  2.20365650  1.19547031  3.9814477  3.57694991
146   AAOI   31-03-2015 -0.23330843 -0.20638101 -0.2406638 -0.21766917
167   AAON   31-03-2015 -0.22173300 -0.19478331 -0.2317944 -0.21520489
        ebit_nor      gm_nor sgna_mg_nor ebit_mg_nor   intexp_nor
17   0.005610863  0.05788391 -0.04398179  0.04982287 -0.005237661
38   0.743086384 -0.07349110 -0.04627883  0.04981099  1.976632243
44  -0.150845233  0.30670023 -0.04162526  0.04988165 -0.290533443
83   1.756284735  0.09271820 -0.04421367  0.04990082  3.621958201
146 -0.159739786 -0.02571837 -0.04472714  0.04939392 -0.302050725
167 -0.136494358 -0.04960474 -0.04567916  0.05011704 -0.304387835
     taxexp_nor netinc_nor      ebt_nor    eps_nor netmargin_nor
17  -0.07104492  0.0357906  0.006291527 0.01802950    0.05025438
38   1.22135355  0.3202280  0.587601802 0.01791388    0.05011986
44  -0.12908262 -0.1244217 -0.129466780 0.01781754    0.05017217
```

Took 1 sec. Last updated by anonymous at March 25 2017, 9:05:34 AM.

FINISHED

```
%spark.r

############## Assigning datasets - from date 1 to date 15
mar2011 <- get(prj2_arq_nn[1])
jun2011 <- get(prj2_arq_nn[2])
sep2011 <- get(prj2_arq_nn[3])
dec2011 <- get(prj2_arq_nn[4])
mar2012 <- get(prj2_arq_nn[5])
jun2012 <- get(prj2_arq_nn[6])
sep2012 <- get(prj2_arq_nn[7])
dec2012 <- get(prj2_arq_nn[8])
mar2013 <- get(prj2_arq_nn[9])
jun2013 <- get(prj2_arq_nn[10])
sep2013 <- get(prj2_arq_nn[11])
dec2013 <- get(prj2_arq_nn[12])
mar2014 <- get(prj2_arq_nn[13])
jun2014 <- get(prj2_arq_nn[14])
sep2014 <- get(prj2_arq_nn[15])
dec2014 <- get(prj2_arq_nn[16])
mar2015 <- get(prj2_arq_nn[17])
jun2015 <- get(prj2_arq_nn[18])
sep2015 <- get(prj2_arq_nn[19])
dec2015 <- get(prj2_arq_nn[20])

```

Zeppelin Notebook

DS-670_Assignm...

Took 1 sec. Last updated by anonymous at April 15 2017, 10:33:03 PM. (outdated)

anonymou

FINISHED

default

```
%spark.r
nn2_2011_mar <- neuralnet(formula1, data=mar2011, hidden = c(8,7), linear.output = T, stepmax = 1e6
plot(nn2_2011_mar)
```

Took 0 sec. Last updated by anonymous at April 15 2017, 10:34:05 PM.

FINISHED

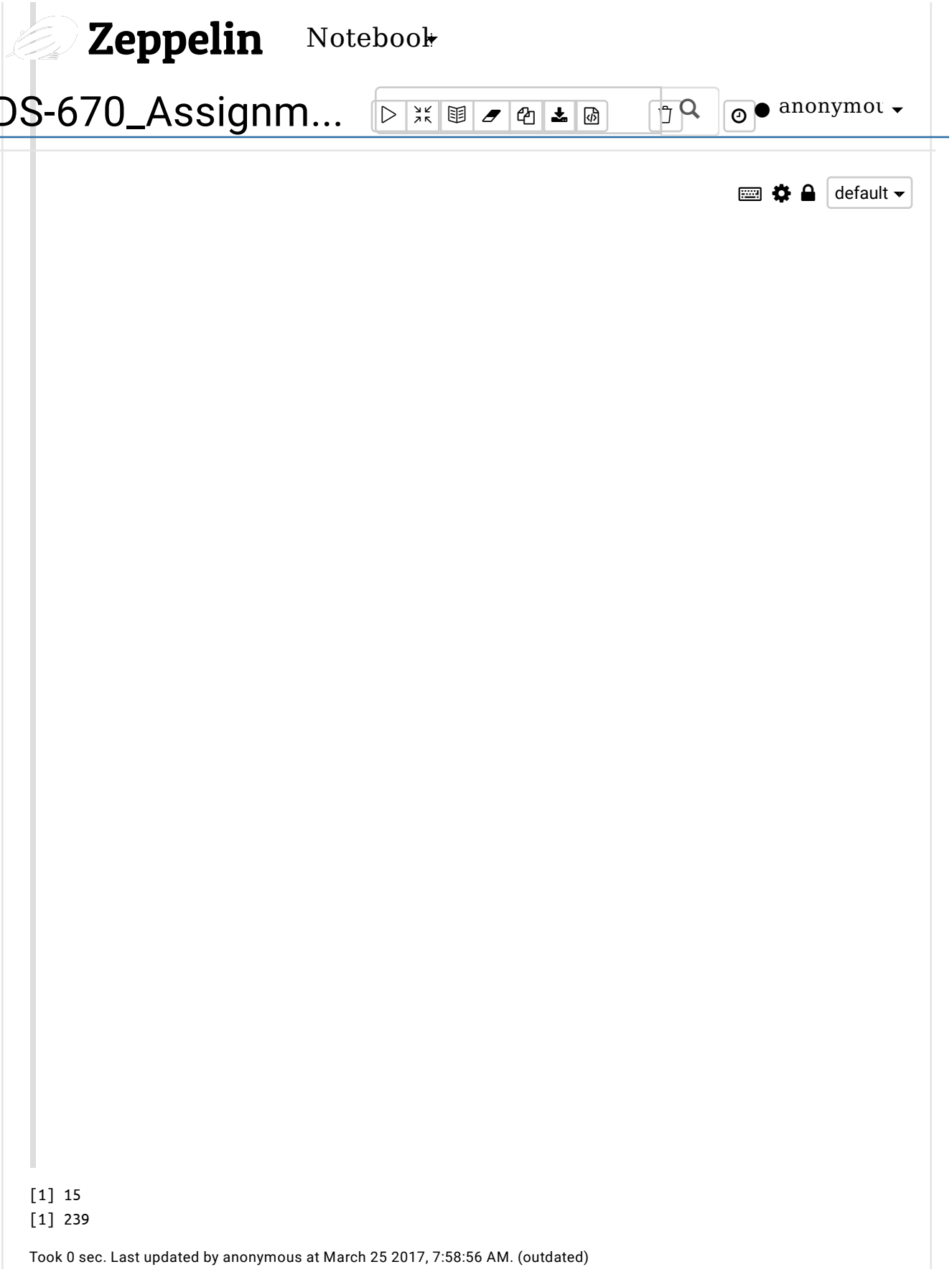
```
1  %spark.r
2  ###################### neural network - package does not
3  names_date <- names(get(prj2_arq_nn[1]))
4  names_date
5  ### log(returns) as y .... we need to get the formula to use in nueral networks as it does n
6  names_date %in% "return_log"
7  !names_date %in% "return_log"
8  paste(names_date[!names_date %in% "return_log"])
9  paste(names_date[!names_date %in% "return_log"], collapse = "+")
10 paste("return_log ~ ", paste(names_date[!names_date %in% "return_log"], collapse = "+"))
11 formula1 <- as.formula(paste("return_log ~ ", paste(names_date[!names_date %in% "return_log"
12 formula1
13 #124 lines
```

```
[1] “revenue_nor”   “cor_nor”       “gp_nor”        “sgna_nor”<br />
 [5] “ebit_nor”      “gm_nor”        “sgna_mg_nor”   “ebit_mg_nor”<br />
 [9] “intexp_nor”    “taxexp_nor”    “netinc_nor”    “ebt_nor”<br />
[13] “eps_nor”       “netmargin_nor” “assets_nor”    “assetsc_nor”<br />
[17] “liabc_nor”     “cur_ratio_nor” “wc_nor”        “capex_nor”<br />
[21] “return_log”<br />
 [1] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE
[12] FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE FALSE  TRUE
 [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
[12]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE FALSE
 [1] “revenue_nor”   “cor_nor”       “gp_nor”        “sgna_nor”<br />
 [5] “ebit_nor”      “gm_nor”        “sgna_mg_nor”   “ebit_mg_nor”<br />
 [9] “intexp_nor”    “taxexp_nor”    “netinc_nor”    “ebt_nor”<br />
[13] “eps_nor”       “netmargin_nor” “assets_nor”    “assetsc_nor”<br />
[17] “liabc_nor”     “cur_ratio_nor” “wc_nor”        “capex_nor”<br />
[1] “revenue_nor+cor_nor+gp_nor+sgna_nor+ebit_nor+gm_nor+sgna_mg_nor+ebit_mg_nor+intexp_nor+taxexp_no
r+netinc_nor+ebt_nor+eps_nor+netmargin_nor+assets_nor+assetsc_nor+liabc_nor+cur_ratio_nor+wc_nor+cape
x_nor”
```

Took 0 sec. Last updated by anonymous at April 15 2017, 10:34:01 PM.

FINISHED

```
1  %spark.r
2
3  ## 2011 ##
4  nn1_2011_03_31 <- neuralnet(formula1, data=mar2011, hidden = c(8,7), linear.output = T)
5  nn1_2011_06_30 <- neuralnet(formula1, data=jun2011, hidden = c(8,7), linear.output = T, thresh
6  nn1_2011_09_30 <- neuralnet(formula1, data=sep2011, hidden = c(8,7), linear.output = T, thresh
```


Zeppelin
Notebook
DS-670_Assignm...
anonymou
default
[1] 15
[1] 239
Took 0 sec. Last updated by anonymous at March 25 2017, 7:58:56 AM. (outdated)

Zeppelin Notebook

DS-670_Assignm...

anonymou

FINISHED

```
%spark.r
matrix_weight <- read.csv(file = "/home/scarface/Desktop/sem-3/capstone/madhu/Matrix_Weight.csv", he
save(matrix_weight, file="matWeights.rdata")
load("matWeights.rdata")
nrow(matrix_weight)
ncol(matrix_weight)
```

default

```
[1] 15
[1] 239
```

Took 1 sec. Last updated by anonymous at April 15 2017, 10:37:13 PM.

3. Time Series Analysis

FINISHED

```
# Time series modeling for gathered weights is done using the Arima function to predict the weights
```

Took 0 sec. Last updated by anonymous at March 26 2017, 8:03:11 AM. (outdated)

FINISHED

```
%spark.r
####### Smoothing the weights from NN model
####### Time series Analysis - MA model timeseries_nn.R
#######
################## Use p=0, d=2, q=1 as model ########################
p_w <- vector();
predict_weights = vector();
abc=vector();
for(j in 1:5)
{
  print(paste0("Date: ", j))
  for(i in 1:ncol(matrix_weight))
  {
    test1 <- arima(matrix_weight[,i], order = c(0,2,1))
    test2 <- predict(test1, n.ahead = 1)
    p_w = test2$pred[1]
    predict_weights[i] <- p_w
  }
  predict_weight <- t(predict_weights)
  #print(paste0(nrow(matrix_weight),":",nrow))
  matrix_weight = rbind(matrix_weight, predict_weight)
  #print(rbind(matrix_weight, predict_weight))
  print (paste0("Row count: ", nrow(matrix_weight)))

  p_w = NULL;
  predict_weights = NULL;
  predict_weight = NULL;
  #test1 <- NULL;
  #test2 <- NULL;
}

#View(matrix_weight)

#177 lines
```

Zeppelin Notebook

DS-670_Assignm...

anonymou

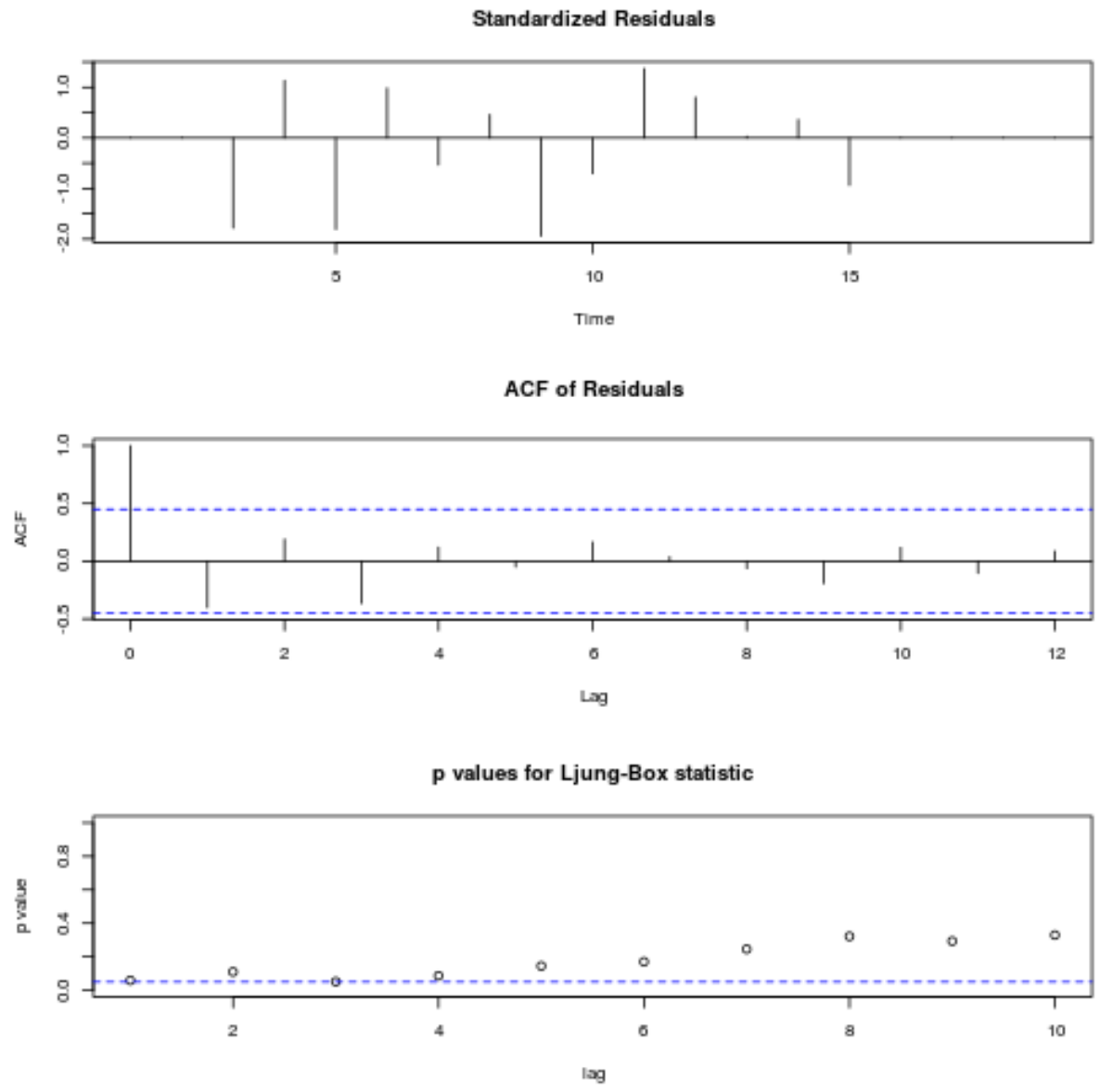
```
[1] "Date: 1"
[1] "Row count: 16"
[1] "Date: 2"
[1] "Row count: 17"
[1] "Date: 3"
[1] "Row count: 18"
[1] "Date: 4"
[1] "Row count: 19"
[1] "Date: 5"
[1] "Row count: 20"
```

default

Took 3 sec. Last updated by anonymous at April 15 2017, 10:37:54 PM.

FINISHED

```
%spark.r
tsdiag(test1)
```



Zeppelin Notebook

DS-670_Assignm...

FINISHED anonymous

```
1 %spark.r
2 nrow(matrix_weight)
3 ncol(matrix_weight)
4 #head(matrix_weight)
```

default

```
[1] 20
[1] 239
```

Took 0 sec. Last updated by anonymous at March 25 2017, 9:07:08 AM.

FINISHED

```
1  %spark.r
2  ###################### Neural Network Equation ######################
3  #View(dec2014)
4  for (a in 16:20)
5  {
6    if (a==16){
7      for(b in 1:ncol(get(prj2_arq_nn[a]))-1)
8      {
9        if(b == 1)
10       {
11         #print(paste0("Column no:", b, get(prj2_arq_nn[a])[b]))
12       }
13     }
14   }
15
16 }
17 #185 lines
```

Took 0 sec. Last updated by anonymous at March 25 2017, 9:07:44 AM.

FINISHED

```
val act_returns = sqlContext.read.format("com.databricks.spark.csv").option("header", "true").load("
val exp_returns = sqlContext.read.format("com.databricks.spark.csv").option("header", "true").load("
```

```
act_returns: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [Quarter_Date: string, Group1:
string ... 4 more fields]
exp_returns: org.apache.spark.sql.Dataset[org.apache.spark.sql.Row] = [Quarter_Date: string, Group1:
string ... 4 more fields]
```

Took 5 sec. Last updated by anonymous at April 15 2017, 10:47:32 PM.

FINISHED

```
act_returns.toDF().registerTempTable("actReturns")
exp_returns.toDF().registerTempTable("expReturns")
```

```
warning: there was one deprecation warning; re-run with -deprecation for details
warning: there was one deprecation warning; re-run with -deprecation for details
```

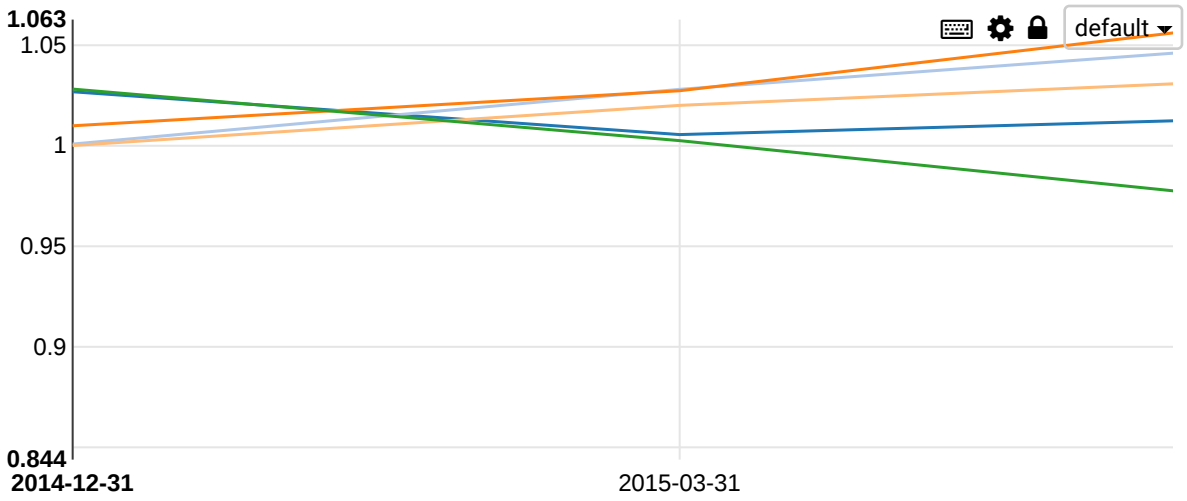
Took 1 sec. Last updated by anonymous at April 15 2017, 10:49:06 PM.

Zeppelin Notebook

DS-670_Assignm...

```
select * from actReturns
```

settings anonymous

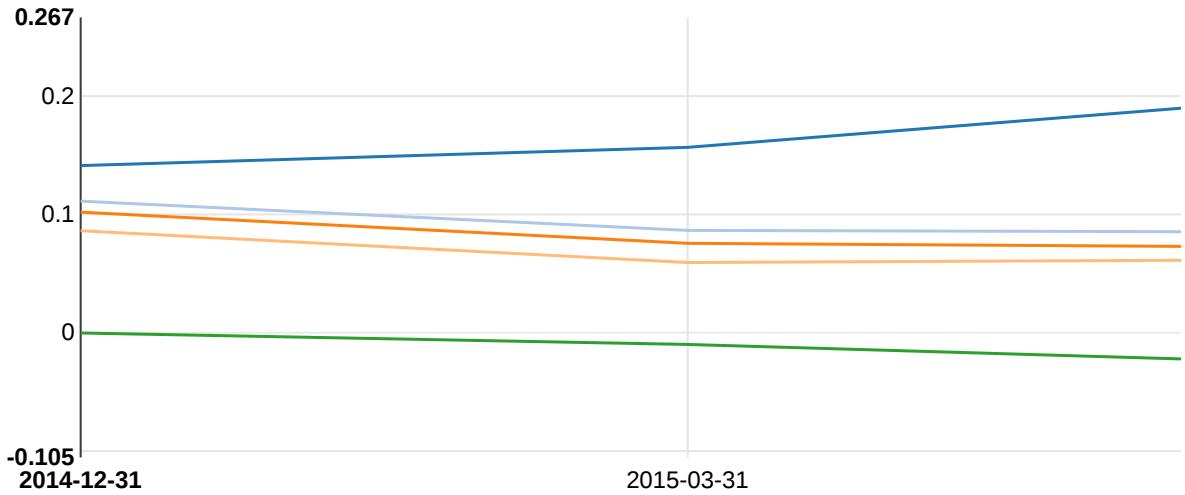


Took 1 sec. Last updated by anonymous at April 15 2017, 10:51:02 PM. (outdated)

FINISHED

```
%sql
select * from expReturns
```

settings

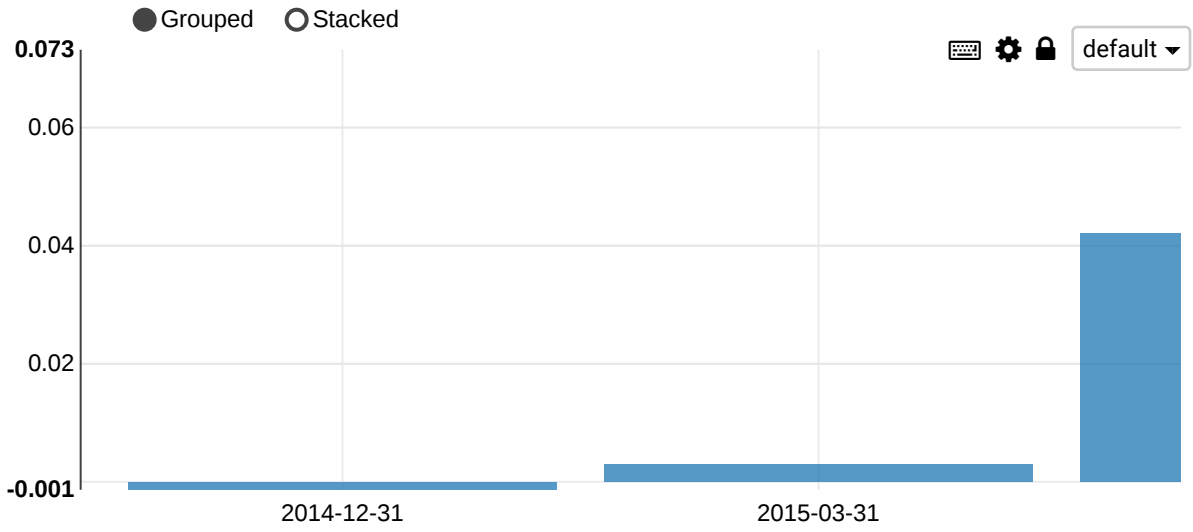


Took 0 sec. Last updated by anonymous at April 15 2017, 10:51:04 PM. (outdated)

Zeppelin Notebook

```
select Group1-Group5 as Depreciation, Quarter_Date from actReturns
```

DS670_Assignm...



Took 0 sec. Last updated by anonymous at April 15 2017, 11:01:35 PM. (outdated)

FINISHED

Zeppelin Notebook

```
select Group1-Group5 as Depreciation, Quarter_Date from expReturns
```

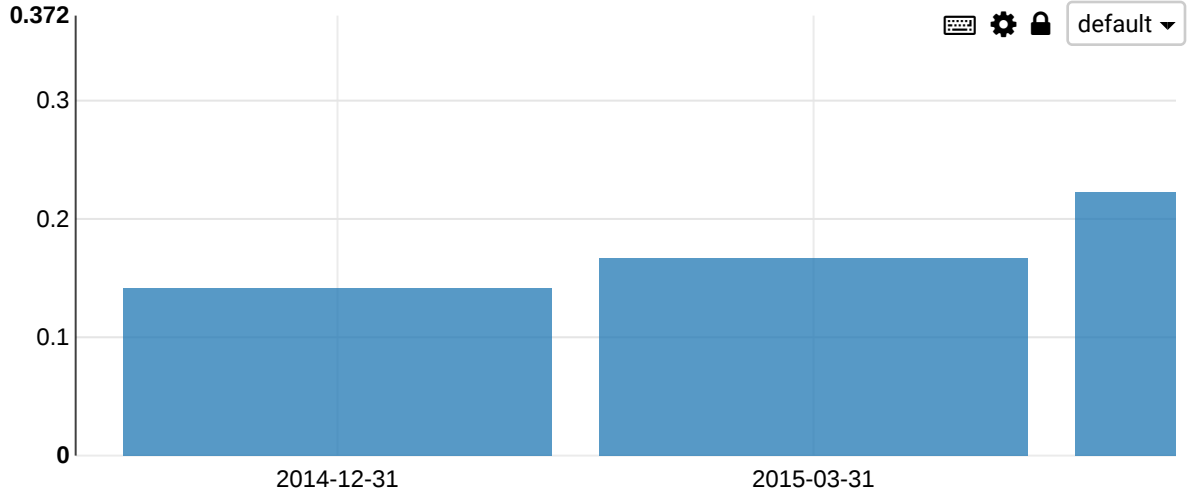
DS-670_Assignm...

settings

anonymou

Grouped Stacked

default



Took 0 sec. Last updated by anonymous at April 15 2017, 11:00:39 PM. (outdated)

READY