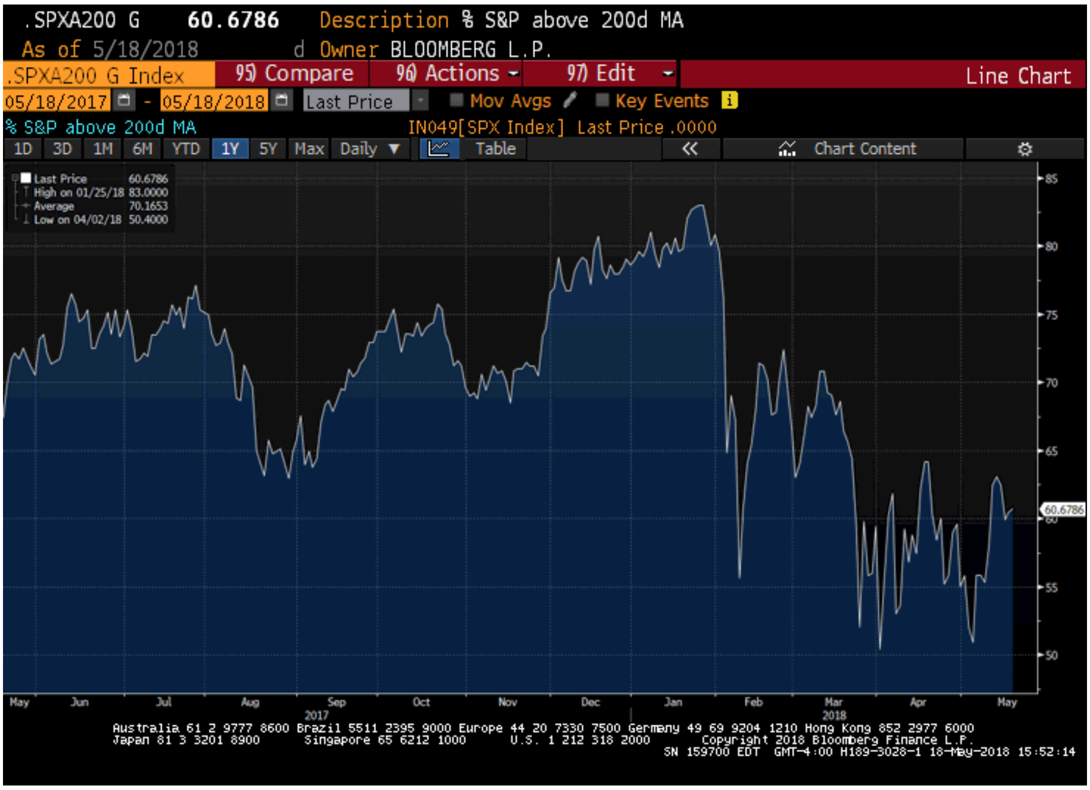
Plus Plus Capital Model Analysis

Data
The given data needed to be refined a bit in two places.

1) The 10 Day Rank was incomplete for first few entries for S&P data
2) An extra date was present in S&P data on line 291 where extra date is present on 22/02/91, NHMNL data doesn't have any such date

I downloaded the above 200d Moving average from Bloomberg terminal and data was available only after the 10/15/1990.
Symbol Used - .SPXA200



To run the code, please use the two csv files I have attached within the code folder, they are complete & data syncs with dates

Analysis -

Step 1

```r
# Step 1
calculate.return = function(days){
  # To Store the positions where we entered the trade
  position = c()
  # To Calculate profit and loss
  profit.loss=c()
  # To keep track of out vectors length
  t=1
  for(i in 1:l){
    if(five.SP[i] < 0.10 || ten.SP[i]< 0.10){
        if(five.NHMNL[i] > 0.95 || ten.NHMNL[i]>0.95){
        position[t]=i+1
        profit.loss[t]=(Price.SP[i+days]/Price.SP[i+1]-1)*100
        t=t+1
      }
    }
  }
  position
  return(profit.loss)
}

returns.data=NULL
for(i in 2:101){
  # calling the calculate.return function to find the returns for days given
  returns.data=cbind(returns.data,calculate.return(i))
  #colnames(returns.data)=i
}
# This vector has 1 to 100 days of returns as asked for in vector 1
returns.data
```

So I built a function called calculate.returns and called it 100 times from a for loop. Return Matrix holds the 100 vectors who are called as “ 1 Day exits”,”2 Day exits” etc…

```
           [,56]      [,57]       [,58]      [,59]      [,60]
 [1,]  -9.3764878 -9.3943699 -8.49427782 -7.0547162 -6.2380763
 [2,] -10.2821380 -9.3908651 -7.96540855 -7.1567702 -8.0332904
 [3,]  -1.3443726 -1.3132511 -0.62239373 -1.8765165 -1.6524544
 [4,]  -2.1808220 -1.4960381 -2.73913568 -2.5170434 -2.8039100
 [5,]   6.9384422  6.8574521  6.60219759  6.6463742  5.7161813
 [6,]   1.0188646  1.0947585  1.28105040  1.2695536  2.9369870
 [7,]  -0.2971391 -0.1134121 -0.12475061  1.5197251  2.2047318
 [8,]   3.5690463  3.2918575  0.94973739  1.3687434  1.2161784
 [9,]  -0.1651579 -0.4419107 -0.09373575 -0.8436556 -0.4128950
[10,]  -1.0668022 -0.7208126 -1.46602553 -1.0379686 -1.0002684
            [,61]      [,62]       [,63]      [,64]        [,65]
 [1,] -7.12326979 -8.7952968 -8.31843256 -6.4914162 -4.780632608
 [2,] -9.68893471 -9.2167428 -7.40762788 -5.7136068 -6.247781623
 [3,] -1.94186534 -1.5030765 -1.00827474 -1.0549524 -1.531083295
 [4,] -2.36897867 -1.8785268 -1.92479409 -2.3967392 -0.607668647
 [5,]  5.30384496  5.7701772  5.79226558  6.2806786  6.994894920
 [6,]  3.63155668  2.5391003  3.03588141  3.2313750  4.287031857
 [7,]  1.12731668  1.6172579  1.81005990  2.8511822  2.617553730
```

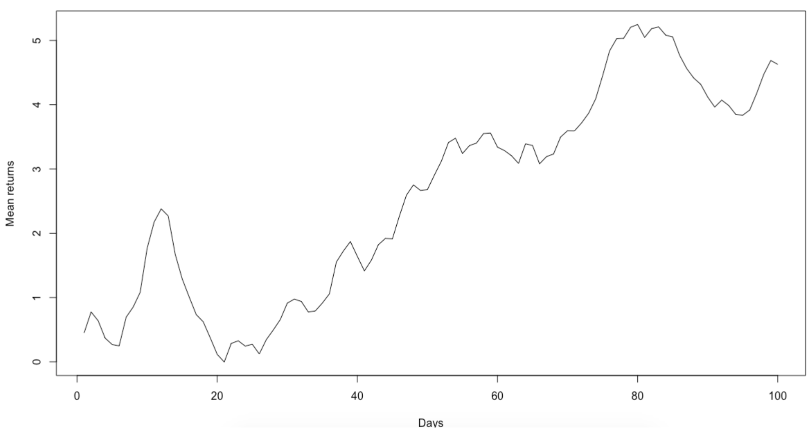
Step 2 Summary Statistics

```r
# Step 2

# To find the occurence
Mode <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

mean.returns=c()
sd.returns=c()
skew.returns=c()
ts.test=c()
occurence.returns=c()
for(i in 1:100){
  mean.returns[i]=mean(returns.data[,i])
  sd.returns[i]=sd(returns.data[,i])
  occurence.returns[i]=Mode(returns.data[,i])
  ts.test[i]=t.test(returns.data[,i])$statistic
  skew.returns[i]=skewness(returns.data[,i])
}
mean.returns
sd.returns
occurence.returns
skew.returns
ts.test
```

All the functions used are already defined in R except Mode<- which calculates the statistical mode of the data, I just ran a for loop through the 100 vectors

Step 3

```r
# Step 3
plot(c(1:100),mean.returns,type="l",xlab="Days",ylab="Mean returns")
```



With the help of plot I can interpret that by increasing the holding period time, the profits earned were gradually increasing. So I will prefer holding an S&P 500 ETF for longer duration than shorter durations

Step 4 I wrote a new function for it, I could have used the function built in Step 1 but, few things were getting added I want to keep code clean for understanding purpose.

```r
# Step 4
calculate.return2 = function(days,check){
  # To Store the positions where we entered the trade
  position = c()
  # To Calculate profit and loss
  profit.loss=c()
  # To keep track of out vectors length
  t=1
  for(i in 1:l){
    if(five.SP[i] < 0.10 || ten.SP[i]< 0.10){
      if(five.NHMNL[i] > 0.95 || ten.NHMNL[i]>0.95){
        position[t]=i+1
        profit.loss[t]=(Price.SP[i+days]/Price.SP[i+1]-1)*100
        t=t+1
      }
    }
  }
  complete.data=NULL
  # entry day
  if(check==0){
    temp=as.character(data$X31.05.90)
    complete.data=cbind(complete.data,temp[position])
  }else{
    complete.data=cbind(complete.data,dates.SP.NHMNL[position])
  }
  # entry price of day
  complete.data=cbind(complete.data,as.integer(Price.SP[position]))
  complete.data=cbind(complete.data,Price.SP[position+days])
  complete.data=cbind(complete.data,dates.SP.NHMNL[position+days])
  complete.data=cbind(complete.data,profit.loss)
  complete.data=cbind(complete.data,five.SP[position-1])
  complete.data=cbind(complete.data,ten.SP[position-1])
  complete.data=cbind(complete.data,five.NHMNL[position-1])
  complete.data=cbind(complete.data,ten.NHMNL[position-1])

  colnames(complete.data)=c("Date","Entry.price","Price.45.days.later","Date.Trade.45.later","Return","
  complete.data=as.data.frame(complete.data)
  return(complete.data)
}

days=45
value=calculate.return2(days,1)
write.csv(calculate.return2(days,0),"NewHighMinusNewLowIndicators.csv")
```

The xls file has been attached with final result, a sip shot from the result

	Date	Entry.price	Price.45.days.later	Date.Trade.45.later	Return	SP.5.ranks	SP.10.rank	NHMNL.5d.rank	NHMNL.10d.rank
1	2154	335	303.23	3340	-10.5776082	0.23	0.07	1.00	0.25
2	2849	338	298.92	3565	-10.5093807	0.52	0.02	1.00	0.25
3	6203	321	304.00	6807	-5.3774812	0.44	0.06	1.00	0.70
4	6411	324	307.02	183	-6.2278299	0.78	0.06	0.99	0.63
5	2657	407	432.57	3379	6.2340452	0.17	0.01	0.99	0.47
6	4991	434	438.02	6130	0.7382755	0.50	0.05	0.98	0.45
7	5221	440	438.89	6337	-0.6464504	0.50	0.08	0.99	0.84
8	2678	465	474.25	3849	1.6996439	0.72	0.01	1.00	0.34
9	1160	448	458.67	2117	2.1894872	0.16	0.03	0.96	0.07
10	1389	450	459.10	2814	1.7277342	0.71	0.04	1.00	0.09
11	1617	447	462.37	3042	2.6839633	0.91	0.06	1.00	0.81
12	2313	449	460.61	3266	2.7785805	0.98	0.06	1.00	0.51
13	6261	454	470.40	17	3.5802380	0.08	0.03	0.98	0.95
14	6677	453	478.65	465	4.2099246	0.80	0.07	0.98	0.83
15	204	448	481.14	1124	6.6225564	0.72	0.02	1.00	0.87

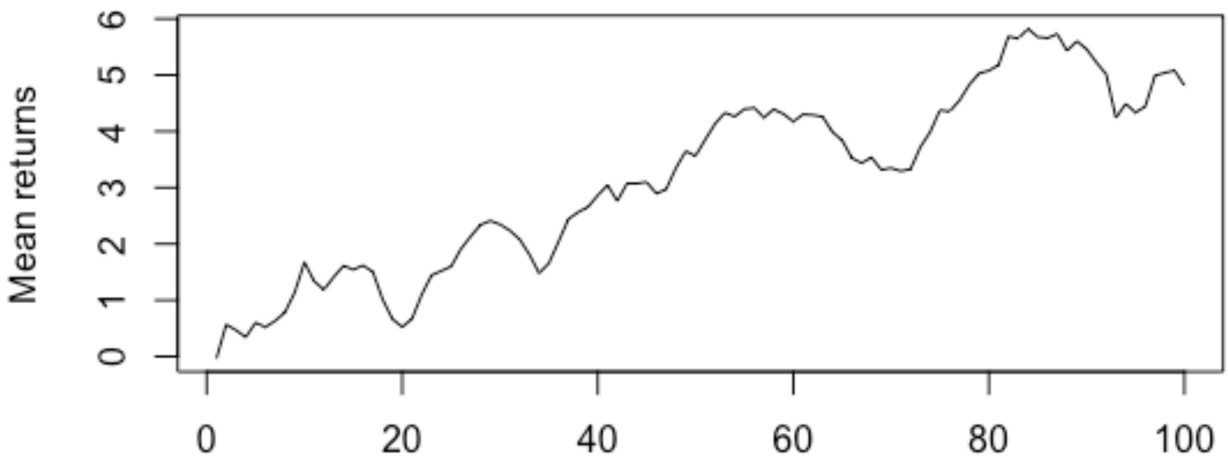
Step 5

```
# Step 5
t.test(value$Return)$statistic
# We are trying to find evidence of a significant between the population mean and a hypothesized value
# The greater the value of t which is 1.7049 in our case the greater evidnece against null hypothesis, that
# there is no significant differnece
```

We are trying to find evidence of a significant between the population mean(mean return) and a hypothesised value(0). The greater the value of t which is 1.7049 in our case the greater evidnece against null hypothesis, that there is no significant difference.

Step 6 & 7 The given tasks were performed for 200 D above Moving average and the xls file name “% of stocks above 200 D MA was attached.”. If we plot the chart for mean returns, it’s very clear that 200 D Moving average performed much better than the NHMNL signal.

Fig 200 D Moving average



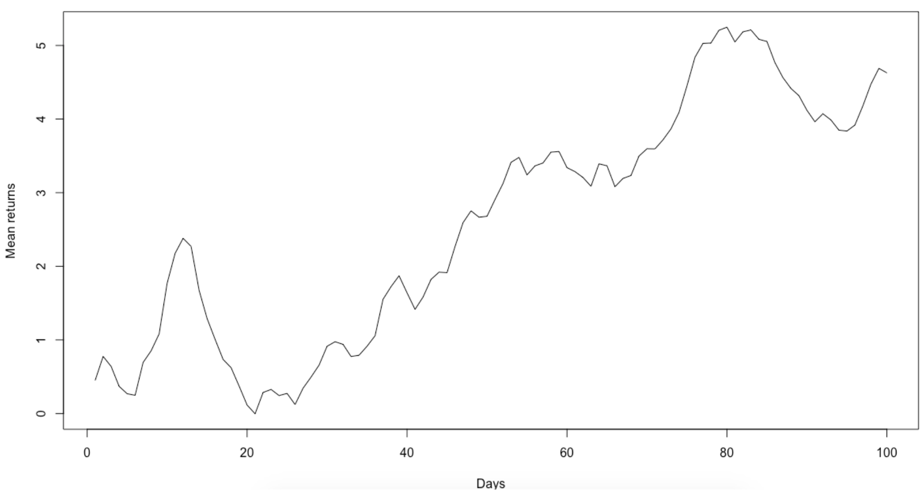


Fig NHMNL signal

Comparison of other parameters for both signals

200 Day MA

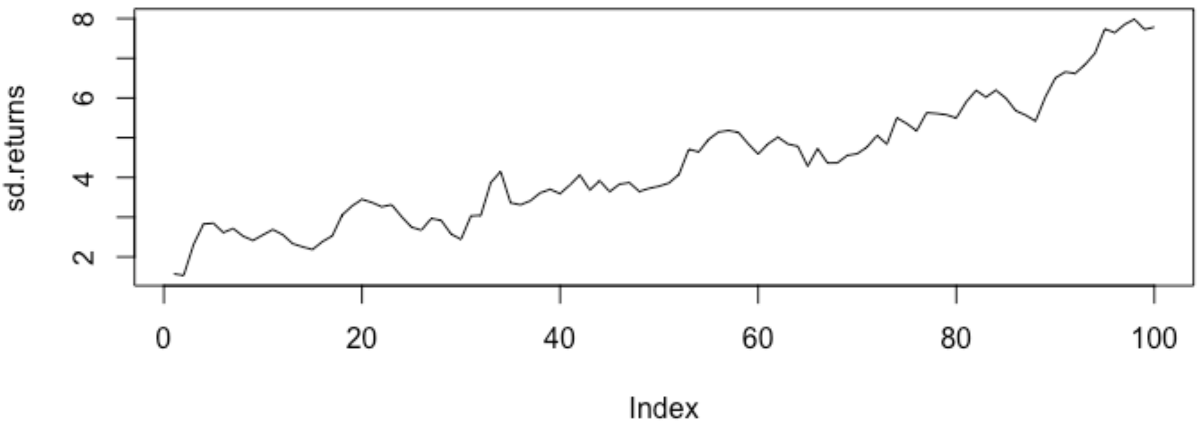
T Stat for 45 days
3.419668

NHMNL

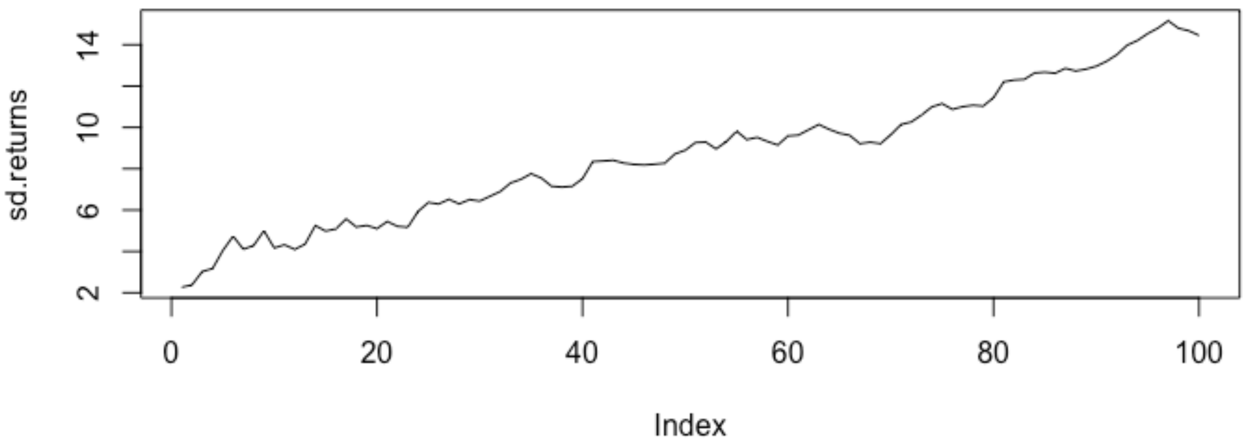
T Stat for 45 days
1.7049

Standard Deviation for Returns

200 Day MA



NHMNL



Clearly NHMNL shows more have returns with higher SD and 200 DAY MA have better returns. Again 200 MA is better indicator than NHMNL, however 200 Day MA have a lot high T Stat which means the chances of this event happening are very less, which was derived from conditioning in first part.