Baruch College, STA-CIS 3920, Exercise#3 Anil Poonai 37

16 July 2020

**Exercise 3.1**

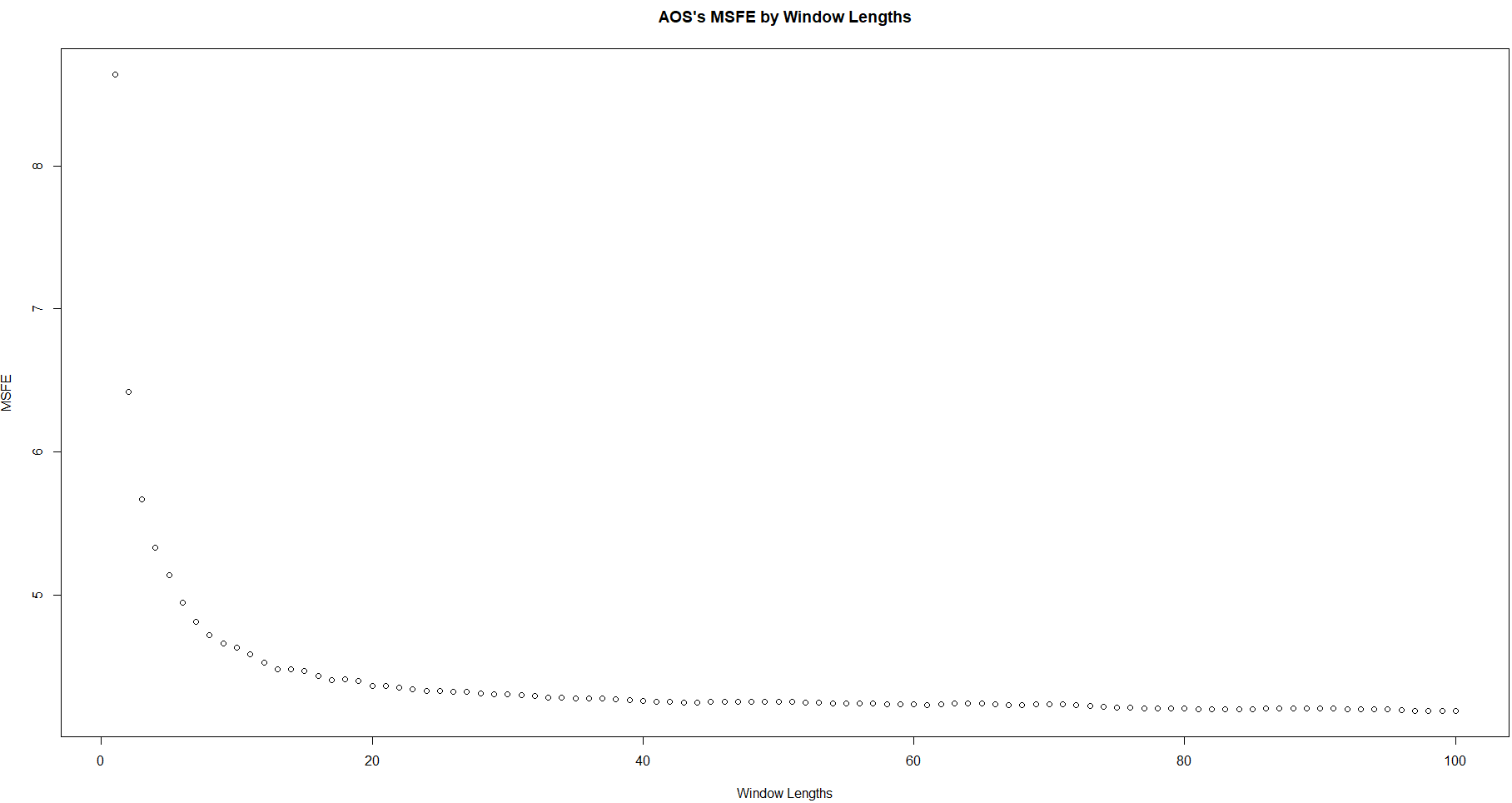


a.

A screenshot of a cell phone

Description automatically generated

b.



The point highlighted in blue on the graph shown is around a 50 window length, which explains its horizontal location as it incorporates its past 50 MSFE values and its vertical axis is explained by the stability of having more points. As its close to the horizonal axis and closer to zero on the range of points in the graph as well as the graph as a whole having a descending slope, means that as the stock returns are happening its becoming much more stable and continuous which also means that it would get smaller as the difference between the two days used to calculate the return gets smaller.

c.

I would use a 60 window length as I can see the difference in slope on the graph up to that point. The points after look fairly close to a straight line.

d.



A close up of a logo

Description automatically generatedCompared to the IBM plot, the slope looks the exact opposite after the first 50 window length. This means that the data for the IBM plot isn’t accurate as the MSFE is suppose to go down the longer it is around.

e.

Larger as it has much more of a value based on data from a longer time ago.

**Exercise 3.2**

A picture containing boat, water, person, sitting

Description automatically generated

Forecasting stock price for big companies will usually have a higher stock price as time goes by and a lower MSFE as well.

**Exercise 3.3**

A close up of a map

Description automatically generatedDaily range should have a higher length as price changes much more than the returns do after a length of 50. The magnitude of the price graph slope is way higher and would be similar to the magnitude of the daily range slope so a 100 moving average isn’t going to be as practical or useful but if we were to use it, the higher the window the better as it would have much newer information.

**Exercise 3.4**

A screenshot of a cell phone

Description automatically generated

The function calculates the factorial of a value, x. This is done through recursion by first having the function check to see if the value is greater than 1. After that if it is greater than 1 it will calculate the answer by taking x and multiplying it by the same function called on the value but for one value less than the value, x. After that it returns that calculation. This works because in the bottom I set the return value to 1 when x is less than or equal to 1. This makes it so that it will always go use the function to call itself until it gets to 1, at that point it will calculate everything and return the factorial of it. I used an if else statement and recursion as an alternative to loops as they act similar and are not conventional in R due to how it vectorizes everything.

**APPENDIX**

R version 4.0.0 (2020-04-24) -- "Arbor Day"

Copyright (C) 2020 The R Foundation for Statistical Computing

Platform: x86\_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.

You are welcome to redistribute it under certain conditions.

Type 'license()' or 'licence()' for distribution details.

Natural language support but running in an English locale

R is a collaborative project with many contributors.

Type 'contributors()' for more information and

'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or

'help.start()' for an HTML browser interface to help.

Type 'q()' to quit R.

[Previously saved workspace restored]

> getwd()

[1] "C:/Users/poona/Documents"

> setwd("C:/Users/poona/Desktop/School"

+ )

> ls()

[1] "a1" "a2" "data\_5\_2\_2020\_6\_08\_PM"

> data=read.csv("AOS.csv")

> head(data)

Date Open High Low Close Adj.Close Volume LGFet Daily.Range

1 1/3/2006 5.870000 6.146667 5.863333 6.141667 2.620321 1702800 NA 0.283334

2 1/4/2006 6.141667 6.230000 6.120000 6.175000 2.634542 1293600 0.5427198 0.110000

3 1/5/2006 6.258333 6.280000 6.085000 6.116667 2.609655 1296400 -0.9446424 0.195000

4 1/6/2006 6.283333 6.375000 6.266667 6.326667 2.699250 2917200 3.4332124 0.108333

5 1/9/2006 6.308333 6.601666 6.308333 6.541667 2.790979 2010400 3.3983143 0.293333

6 1/10/2006 6.541667 6.780000 6.490000 6.698333 2.857820 2080800 2.3948944 0.290000

> data=data[-1,]

> head(data)

Date Open High Low Close Adj.Close Volume LGFet Daily.Range

2 1/4/2006 6.141667 6.230000 6.120000 6.175000 2.634542 1293600 0.5427198 0.110000

3 1/5/2006 6.258333 6.280000 6.085000 6.116667 2.609655 1296400 -0.9446424 0.195000

4 1/6/2006 6.283333 6.375000 6.266667 6.326667 2.699250 2917200 3.4332124 0.108333

5 1/9/2006 6.308333 6.601666 6.308333 6.541667 2.790979 2010400 3.3983143 0.293333

6 1/10/2006 6.541667 6.780000 6.490000 6.698333 2.857820 2080800 2.3948944 0.290000

7 1/11/2006 6.680000 6.708333 6.565000 6.598333 2.815156 1474800 -1.4928862 0.143333

> VSFE <-

+ function (x,d) # x is data, d is max window length

+ {

+ z <- vector(length=d)

+ for(j in 1:d) {

+

+ z[j] <- MVG1(x,j)

+

+ }

+ return(z)

+ }

>

> MVG1 =

+ function (x,m)

+ {

+ n <- length(x)

+ y <- vector(length=(n-m))

+

+ for(k in 1:(n-m)) {

+ y[k] <- ( x[m+k] - mean(x[k:(k+m-1) ]) )\*\*2

+ }

+ msfe <- mean(y)

+ return(msfe)

+ }

> MVG1(data$LGFet,500)

[1] 4.101693

> VSFE(data$LGFet,500)

[1] 8.642273 6.419967 5.665387 5.331114 5.136138 4.943792 4.810732 4.718492 4.656597 4.630768 4.582227 4.522368 4.479254 4.478821 4.464090 4.432369 4.401120 4.406489 4.394435 4.362339 4.362660 4.350993 4.335509 4.326331

[25] 4.325416 4.321473 4.321059 4.310997 4.305926 4.301074 4.297112 4.291755 4.281851 4.279629 4.275495 4.273106 4.273075 4.269378 4.261338 4.255967 4.253481 4.251398 4.247426 4.245556 4.251975 4.253764 4.250772 4.250540

[49] 4.251093 4.248470 4.248100 4.244423 4.243681 4.239920 4.238370 4.239187 4.238465 4.232568 4.233799 4.231948 4.229091 4.233600 4.240850 4.238292 4.239159 4.234541 4.230475 4.227776 4.233151 4.235036 4.234926 4.227016

[73] 4.218918 4.215382 4.212690 4.210423 4.205550 4.205975 4.203440 4.201680 4.199106 4.199802 4.199057 4.197968 4.200349 4.204412 4.204861 4.205687 4.205077 4.203070 4.202483 4.200869 4.201226 4.199435 4.198962 4.191474

[97] 4.189570 4.189319 4.186710 4.186946 4.188385 4.188523 4.189286 4.188377 4.186106 4.185667 4.183917 4.183970 4.185834 4.185390 4.186215 4.187430 4.185420 4.186877 4.190849 4.190253 4.192304 4.192660 4.190017 4.190239

[121] 4.193250 4.193162 4.187625 4.185787 4.185397 4.184329 4.184714 4.186086 4.187063 4.189086 4.186666 4.186774 4.188040 4.190215 4.191099 4.184066 4.184735 4.175618 4.171125 4.169899 4.170746 4.171542 4.172869 4.171887

[145] 4.172448 4.172950 4.172020 4.172752 4.170985 4.172011 4.172039 4.172591 4.173599 4.175748 4.170791 4.167900 4.169087 4.169342 4.168332 4.166903 4.169772 4.169474 4.171249 4.171840 4.173476 4.175432 4.176088 4.175634

[169] 4.178656 4.177036 4.175250 4.176808 4.179265 4.176272 4.176162 4.176834 4.177520 4.178698 4.178776 4.176651 4.177539 4.179029 4.178943 4.178747 4.181310 4.181955 4.178998 4.180142 4.180624 4.181940 4.181827 4.183043

[193] 4.182292 4.182749 4.182982 4.179990 4.177528 4.177065 4.140203 4.137264 4.138450 4.137654 4.137331 4.137520 4.138825 4.138156 4.139621 4.140962 4.143028 4.143003 4.144706 4.145149 4.144369 4.144968 4.145949 4.147123

[217] 4.148030 4.149686 4.146280 4.146143 4.146777 4.147513 4.148022 4.150095 4.151768 4.151274 4.149515 4.150664 4.150851 4.151966 4.153382 4.154404 4.155747 4.156533 4.155537 4.156272 4.156818 4.156657 4.157404 4.158366

[241] 4.159019 4.156750 4.157691 4.156258 4.156486 4.157578 4.157175 4.158323 4.158573 4.159109 4.158919 4.158977 4.158514 4.160531 4.162483 4.163688 4.164564 4.166137 4.166706 4.168084 4.169547 4.171802 4.161400 4.163263

[265] 4.164811 4.163444 4.163687 4.162585 4.163785 4.164913 4.165653 4.166477 4.166171 4.167363 4.168202 4.169134 4.170439 4.170796 4.172189 4.174014 4.175624 4.177967 4.174106 4.175305 4.176704 4.177844 4.178267 4.174143

[289] 4.174376 4.175667 4.176431 4.177660 4.177896 4.176987 4.177491 4.177442 4.179729 4.178577 4.178178 4.179376 4.180225 4.180944 4.182189 4.181478 4.182383 4.183734 4.185560 4.186867 4.188126 4.189134 4.189121 4.190644

[313] 4.190807 4.192144 4.193173 4.193804 4.194178 4.195377 4.196001 4.197013 4.196445 4.196994 4.197847 4.195392 4.195493 4.195778 4.196583 4.197559 4.197039 4.198674 4.200358 4.201508 4.201746 4.202852 4.203762 4.204163

[337] 4.205143 4.206466 4.205327 4.206345 4.207465 4.208504 4.208918 4.209968 4.210755 4.212267 4.213468 4.214277 4.214595 4.215573 4.216844 4.217734 4.218990 4.214832 4.215760 4.217403 4.217655 4.218788 4.219215 4.220580

[361] 4.222204 4.223246 4.223651 4.224761 4.225915 4.226861 4.228545 4.229522 4.231705 4.232933 4.233692 4.234502 4.236063 4.236546 4.236189 4.237382 4.238493 4.238968 4.238860 4.240401 4.242276 4.242303 4.242626 4.243482

[385] 4.244539 4.205271 4.201853 4.202305 4.203170 4.202989 4.204248 4.202504 4.203483 4.202708 4.203858 4.202659 4.203022 4.197931 4.198988 4.200492 4.201639 4.202836 4.198027 4.199807 4.198383 4.198084 4.195177 4.191551

[409] 4.191566 4.193090 4.189303 4.189784 4.191233 4.191378 4.191379 4.192260 4.192098 4.192486 4.192281 4.193945 4.195398 4.195188 4.196407 4.198101 4.198757 4.199924 4.201313 4.202112 4.195953 4.191513 4.192528 4.193130

[433] 4.191351 4.192407 4.192927 4.194162 4.196179 4.196292 4.197043 4.198292 4.199811 4.200856 4.202205 4.203448 4.204674 4.205492 4.206104 4.200574 4.174706 4.166575 4.167744 4.167219 4.168648 4.169216 4.170290 4.169946

[457] 4.170864 4.172394 4.173891 4.174610 4.167243 4.168303 4.169599 4.170676 4.167357 4.160448 4.160512 4.162093 4.156565 4.151606 4.149327 4.150196 4.148404 4.149551 4.150368 4.152231 4.154312 4.153490 4.151056 4.151846

[481] 4.152846 4.148866 4.149012 4.149271 4.134889 4.131473 4.129143 4.125306 4.125960 4.127327 4.128513 4.127854 4.128708 4.127474 4.127793 4.110924 4.109537 4.106475 4.100002 4.101693

> VSFE(data$LGFet,100)

[1] 8.642273 6.419967 5.665387 5.331114 5.136138 4.943792 4.810732 4.718492 4.656597 4.630768 4.582227 4.522368 4.479254 4.478821 4.464090 4.432369 4.401120 4.406489 4.394435 4.362339 4.362660 4.350993 4.335509 4.326331

[25] 4.325416 4.321473 4.321059 4.310997 4.305926 4.301074 4.297112 4.291755 4.281851 4.279629 4.275495 4.273106 4.273075 4.269378 4.261338 4.255967 4.253481 4.251398 4.247426 4.245556 4.251975 4.253764 4.250772 4.250540

[49] 4.251093 4.248470 4.248100 4.244423 4.243681 4.239920 4.238370 4.239187 4.238465 4.232568 4.233799 4.231948 4.229091 4.233600 4.240850 4.238292 4.239159 4.234541 4.230475 4.227776 4.233151 4.235036 4.234926 4.227016

[73] 4.218918 4.215382 4.212690 4.210423 4.205550 4.205975 4.203440 4.201680 4.199106 4.199802 4.199057 4.197968 4.200349 4.204412 4.204861 4.205687 4.205077 4.203070 4.202483 4.200869 4.201226 4.199435 4.198962 4.191474

[97] 4.189570 4.189319 4.186710 4.186946

> plot(VSFE(data$LGFet,500))

> plot(VSFE(data$LGFet,100))

> plot(VSFE(data$LGFet,100))

> plot(VSFE(data$LGFet,100),xlab="Window Lengths",ylab="MSFE",main="AOS's MSFE by Window Lengths")

> point()

Error in point() : could not find function "point"

> identify()

Error in xy.coords(x, y, setLab = FALSE) :

argument "x" is missing, with no default

> plot(VSFE(data$LGFet,500),xlab="Window Lengths",ylab="MSFE",main="AOS's MSFE by Window Lengths")

> MVG1(data$LGFet,1)

[1] 8.642273

> MVG1(data$LGFet,500)

[1] 4.101693

> plot(VSFE(data$Adj.CLose,100),xlab="Window Lengths",ylab="MSFE",main="AOS's MSFE by Window Lengths")

Error in vector(length = (n - m)) : invalid 'length' argument

> MVG1(data$Adj.Close,500)

[1] 45.43626

> plot(VSFE(data$Adj.CLose,500),xlab="Window Lengths",ylab="MSFE",main="AOS's MSFE by Window Lengths")

Error in vector(length = (n - m)) : invalid 'length' argument

> VSFE(data$Adj.Close,500)

[1] 0.2194586 0.2740086 0.3437931 0.4191146 0.4923735 0.5668724 0.6431022 0.7207734 0.7988422 0.8770239 0.9539930 1.0286774 1.1015692 1.1738155 1.2452709 1.3149987 1.3844367 1.4539733 1.5238889 1.5939274

[21] 1.6637282 1.7342729 1.8049620 1.8753503 1.9446094 2.0126392 2.0797500 2.1459771 2.2115749 2.2764667 2.3407263 2.4038447 2.4657545 2.5270756 2.5882378 2.6496041 2.7114127 2.7738686 2.8369299 2.9010901

[41] 2.9661187 3.0317797 3.0981286 3.1655707 3.2338989 3.3031287 3.3729012 3.4429708 3.5135015 3.5840215 3.6543359 3.7246530 3.7947691 3.8648485 3.9350239 4.0052149 4.0756505 4.1464448 4.2175549 4.2887766

[61] 4.3600150 4.4314518 4.5028315 4.5740467 4.6450887 4.7156386 4.7858216 4.8556868 4.9250147 4.9938632 5.0621557 5.1298934 5.1969659 5.2634869 5.3294715 5.3950254 5.4602093 5.5249848 5.5892511 5.6528812

[81] 5.7160976 5.7787347 5.8409413 5.9028810 5.9646641 6.0260450 6.0868767 6.1472495 6.2070099 6.2662059 6.3247955 6.3827814 6.4401567 6.4970997 6.5537578 6.6101894 6.6665184 6.7226725 6.7786977 6.8346844

[101] 6.8907973 6.9469748 7.0032092 7.0593727 7.1154915 7.1716138 7.2277155 7.2836861 7.3394823 7.3951571 7.4506951 7.5061222 7.5614930 7.6168819 7.6722897 7.7275791 7.7828244 7.8379986 7.8931134 7.9481748

[121] 8.0031615 8.0581128 8.1129804 8.1677620 8.2224014 8.2769096 8.3313336 8.3856633 8.4398718 8.4939574 8.5479670 8.6018894 8.6555785 8.7090328 8.7622973 8.8154418 8.8685220 8.9215450 8.9745766 9.0276109

[141] 9.0807296 9.1339041 9.1871257 9.2404068 9.2937589 9.3472859 9.4010936 9.4553249 9.5099031 9.5648871 9.6203713 9.6764492 9.7332141 9.7906522 9.8487370 9.9073945 9.9666352 10.0264374 10.0867737 10.1476925

[161] 10.2092316 10.2712515 10.3337603 10.3967552 10.4602292 10.5242045 10.5886263 10.6535323 10.7189893 10.7849365 10.8513229 10.9181548 10.9853775 11.0529409 11.1208716 11.1891685 11.2577740 11.3265988 11.3957531 11.4653204

[181] 11.5353536 11.6057985 11.6767312 11.7481172 11.8201238 11.8927221 11.9658092 12.0394555 12.1135573 12.1880374 12.2628693 12.3380193 12.4134878 12.4892458 12.5652684 12.6415999 12.7183999 12.7956203 12.8732392 12.9512416

[201] 13.0296497 13.1084837 13.1877149 13.2673248 13.3473115 13.4276836 13.5084666 13.5896931 13.6713289 13.7533161 13.8357448 13.9185778 14.0019405 14.0858608 14.1703562 14.2553947 14.3409017 14.4268180 14.5131330 14.5998444

[221] 14.6869648 14.7744999 14.8623945 14.9505637 15.0389653 15.1275713 15.2164459 15.3056061 15.3949942 15.4845782 15.5743083 15.6641704 15.7541186 15.8441136 15.9341485 16.0243006 16.1146022 16.2049828 16.2954897 16.3861127

[241] 16.4768739 16.5678390 16.6589447 16.7502125 16.8416180 16.9331565 17.0248080 17.1166032 17.2085775 17.3007665 17.3931498 17.4857238 17.5784316 17.6713215 17.7644077 17.8576100 17.9508866 18.0441870 18.1374955 18.2308046

[261] 18.3240724 18.4172955 18.5105093 18.6036814 18.6967491 18.7896089 18.8823063 18.9748663 19.0673037 19.1595939 19.2517227 19.3437029 19.4355475 19.5272134 19.6187189 19.7100608 19.8012651 19.8923011 19.9831915 20.0739891

[281] 20.1647069 20.2553122 20.3457650 20.4361095 20.5263931 20.6166150 20.7067474 20.7967900 20.8867764 20.9767971 21.0668080 21.1568401 21.2469094 21.3370244 21.4271678 21.5173271 21.6076326 21.6980691 21.7886160 21.8792925

[301] 21.9700892 22.0610588 22.1521622 22.2433597 22.3346507 22.4261347 22.5178136 22.6096554 22.7016271 22.7936828 22.8858033 22.9780162 23.0703304 23.1627038 23.2552214 23.3478761 23.4406487 23.5336050 23.6267864 23.7202046

[321] 23.8139268 23.9079338 24.0022612 24.0969488 24.1919981 24.2874006 24.3831384 24.4792074 24.5756123 24.6723827 24.7695020 24.8669881 24.9648741 25.0631232 25.1617511 25.2607386 25.3601326 25.4599105 25.5600918 25.6606409

[341] 25.7615539 25.8627985 25.9643601 26.0662351 26.1683723 26.2707529 26.3733597 26.4762246 26.5793576 26.6827738 26.7865247 26.8905161 26.9947905 27.0993749 27.2043345 27.3096909 27.4154411 27.5215956 27.6281828 27.7352122

[361] 27.8426291 27.9504372 28.0586425 28.1672442 28.2762383 28.3856331 28.4954478 28.6056105 28.7161002 28.8268397 28.9378238 29.0491148 29.1607157 29.2726334 29.3848179 29.4972921 29.6100344 29.7229995 29.8362620 29.9498383

[381] 30.0637642 30.1780007 30.2925441 30.4074367 30.5226507 30.6380048 30.7535484 30.8694267 30.9856107 31.1021950 31.2191574 31.3365695 31.4543360 31.5724257 31.6909026 31.8096544 31.9287623 32.0483286 32.1682721 32.2885614

[401] 32.4091876 32.5301470 32.6512905 32.7726655 32.8943865 33.0164668 33.1389182 33.2616255 33.3845811 33.5078474 33.6312969 33.7550779 33.8790933 34.0033633 34.1278862 34.2525620 34.3774708 34.5025930 34.6279280 34.7535555

[421] 34.8794166 35.0055205 35.1318342 35.2583243 35.3849995 35.5118689 35.6389221 35.7661727 35.8934848 36.0209755 36.1485858 36.2763105 36.4041765 36.5322005 36.6603592 36.7886470 36.9170636 37.0455757 37.1741926 37.3029100

[441] 37.4317149 37.5606385 37.6896726 37.8188529 37.9481734 38.0776250 38.2072348 38.3371449 38.4673950 38.5978987 38.7286821 38.8597463 38.9911032 39.1227494 39.2546922 39.3868776 39.5193116 39.6519782 39.7848919 39.9180746

[461] 40.0515216 40.1852631 40.3192884 40.4536136 40.5882351 40.7231639 40.8583467 40.9938135 41.1295323 41.2654697 41.4016214 41.5379723 41.6744699 41.8111425 41.9480084 42.0850575 42.2222926 42.3596798 42.4972505 42.6350103

[481] 42.7729821 42.9111463 43.0495368 43.1881592 43.3270313 43.4661308 43.6054569 43.7449813 43.8847367 44.0247355 44.1649577 44.3053638 44.4459896 44.5867954 44.7277985 44.8690215 45.0104575 45.1521384 45.2940596 45.4362630

> plot(VSFE(data$Adj.Close,500),main="AOS Closing Price MSFE")

> data$Range=data$High=data$Low

> head(data)

Date Open High Low Close Adj.Close Volume LGFet Daily.Range Range

2 1/4/2006 6.141667 6.120000 6.120000 6.175000 2.634542 1293600 0.5427198 0.110000 6.120000

3 1/5/2006 6.258333 6.085000 6.085000 6.116667 2.609655 1296400 -0.9446424 0.195000 6.085000

4 1/6/2006 6.283333 6.266667 6.266667 6.326667 2.699250 2917200 3.4332124 0.108333 6.266667

5 1/9/2006 6.308333 6.308333 6.308333 6.541667 2.790979 2010400 3.3983143 0.293333 6.308333

6 1/10/2006 6.541667 6.490000 6.490000 6.698333 2.857820 2080800 2.3948944 0.290000 6.490000

7 1/11/2006 6.680000 6.565000 6.565000 6.598333 2.815156 1474800 -1.4928862 0.143333 6.565000

> data=read.csv("AOS.csv")

> data$Range=data$High-data$Low

> head(data)

Date Open High Low Close Adj.Close Volume LGFet Daily.Range Range

1 1/3/2006 5.870000 6.146667 5.863333 6.141667 2.620321 1702800 NA 0.283334 0.283334

2 1/4/2006 6.141667 6.230000 6.120000 6.175000 2.634542 1293600 0.5427198 0.110000 0.110000

3 1/5/2006 6.258333 6.280000 6.085000 6.116667 2.609655 1296400 -0.9446424 0.195000 0.195000

4 1/6/2006 6.283333 6.375000 6.266667 6.326667 2.699250 2917200 3.4332124 0.108333 0.108333

5 1/9/2006 6.308333 6.601666 6.308333 6.541667 2.790979 2010400 3.3983143 0.293333 0.293333

6 1/10/2006 6.541667 6.780000 6.490000 6.698333 2.857820 2080800 2.3948944 0.290000 0.290000

> plot(VSFE(data$Range,100),main="AOS Daily Range MSFE")

> plot(VSFE(data$Range,1000),main="AOS Daily Range MSFE")

> plot(VSFE(data$Range,3000),main="AOS Daily Range MSFE")

> plot(VSFE(data$Range,100),main="AOS Daily Range MSFE")

> ap <- function(x) {

+ if (x==1) {

+ )

Error: unexpected ')' in:

"if (x==1) {

)"

> ap <- function(x) {

+ ifelse(x==1,1,)

+ )

Error: unexpected ')' in:

"ifelse(x==1,1,)

)"

> ap <- function(x) {

+ if(x==1) {

+ return(1)

+ } else{

+ )

Error: unexpected ')' in:

"} else{

)"

> ap <- function(x) {

+ base=1

+ if(x>1){

+ y = x\*ap(x-1)

+ y

+ {

+ }}

+ else {

+ y=1

+ y)

Error: unexpected ')' in:

"y=1

y)"

> ap <- function(x) {

+ y=1

+ if (x>1){

+ y=x\*ap(x-1)

+ }}

> }

Error: unexpected '}' in "}"

> ap <- function(x) {

+ y=1

+ if (x>1) {

+ answer=x\*ap(x-1)

+ } else {

+ y}}

> ap 5

Error: unexpected numeric constant in "ap 5"

> ap(5)'

+ )

+ ap(5)

+

+ )

+

+

+ )

+

+

+

+

+ 6=6

+ )))

+ > head(data)

Date Open High Low Close Adj.Close Volume LGFet Daily.Range Range

1 1/3/2006 5.870000 6.146667 5.863333 6.141667 2.620321 1702800 NA 0.283334 0.283334

2 1/4/2006 6.141667 6.230000 6.120000 6.175000 2.634542 1293600 0.5427198 0.110000 0.110000

3 1/5/2006 6.258333 6.280000 6.085000 6.116667 2.609655 1296400 -0.9446424 0.195000 0.195000

4 1/6/2006 6.283333 6.375000 6.266667 6.326667 2.699250 2917200 3.4332124 0.108333 0.108333

5 1/9/2006 6.308333 6.601666 6.308333 6.541667 2.790979 2010400 3.3983143 0.293333 0.293333

6 1/10/2006 6.541667 6.780000 6.490000 6.698333 2.857820 2080800 2.3948944 0.290000 0.290000

> ap <- function(x) {

+

+ if(x>1) {

+ answer = x\*ap(x-1)

+ ) else (

Error: unexpected ')' in:

"answer = x\*ap(x-1)

)"

> return (1)

Error: no function to return from, jumping to top level

> ap <- function(x) {

+ if(x>1) {

+ answer = x\*ap(x-1)

+ ) else (

Error: unexpected ')' in:

"answer = x\*ap(x-1)

)"

> ap <- function(x) {

+ if(x>1) {

+ answer = x\*ap(x-1)

+ ) else {

Error: unexpected ')' in:

"answer = x\*ap(x-1)

)"

> ap <- function(x) {

+ if(x>1) {

+ answer = x\*ap(x-1)

+ return(answer)

+ } else {

+ return (1)

+ }

+ }

> ap(5)

[1] 120

> ap(1)

[1] 1

> ap(9)

[1] 362880

>