In [ ]:
```
STAT 420 HW 4  Donghan Liu  Netid Donghan2
```

In [ ]:
```
Question 1
a)
```

In [1]:
```
install.packages("faraway",repos = "http://cran.us.r-project.org")
library(faraway)
```

```
package 'faraway' successfully unpacked and MD5 sums checked

The downloaded binary packages are in
        C:\Users\Hans\AppData\Local\Temp\RtmpoLgAUp\downloaded_packages
```

In [2]:
```
data(gala)
area = gala $ Area
elevation = gala $ Elevation
nearest = gala $ Nearest
scruz = gala $ Scruz
adjacent = gala $ Adjacent
X = cbind("Intercept" = 1,area, elevation, nearest, scruz, adjacent)
XX = t(X) %*% X
eigen(t(X) %*% X)$values
# As the following graph shows, X is the column combination of area,
# elevation, nearest, scruz and adjacent, and XX is the product of
# inverse of X and X. For the positive definite,eigen(t(X) %*% X)$values
# implies that it is positive definite.
```

36598009.6931402 17873972.6150697 2243835.6754843 167928.143419169 3293.72967739198 10.1337091994749

In [3]:
```
#b)
y = gala $ Species
beta = solve(t(X) %*% X) %*% t(X) %*% y
beta
# In this comnined prediction, the intercept is 7.068220709, and the
# regression coefficients for area, elevation, nearest, scruz and adjacent
# is -0.023938338,0.319464761,0.009143961,-0.240524230 amd -0.074804832
# respectively.
```

Intercept	7.068220709
area	-0.023938338
elevation	0.319464761
nearest	0.009143961
scruz	-0.240524230
adjacent	-0.074804832

In [4]:
```
#c)
H = X %*% solve(t(X) %*% X) %*% t(X)
sigma2 = (t(y) %*% (diag(length(y))-H) %*% y)/(nrow(gala)-5-1)
SSE = t(y) %*% (diag(length(y)) - H) %*% y
sigma2
SSE
e = (diag(length(y)) - H) %*% y
SSE_V = sum(e^2)
sigma2_V = sum(e^2) / (length(y)-6)
sigma2_V
SSE_V
# The formula is coming from lecture note, and the result appears in the
# result area, SSE = 89231.37, and sigma^2 = 3717.974, plus, they are
# verfied by another formula.
```

3717.974

89231.37

3717.97359708546

89231.3663300511

In [5]:
```
#d)
mean_y = matrix(c(mean(y)),30,1)
SST = sum((diag(length(y))%*%y-mean_y)^2)
SST
SSR = sum((H%*%y-mean_y)^2)
SSR
R2 = SSR/SST
R2

# As the formula indicated above, SST = 381081.366666667, SSR =
# 291850.000336614, and the R^2 = 0.765846944681231
```

381081.366666667

291850.000336614

0.765846944681231

In [17]:
```
#e)
The new variable is not able to add in the existing model, since the new variable is
dependent from Nearest+Scruz. When we want to get H, we must be able to inverse X, namely,
X must be invertible, so their column have to be indenpendent, in other words,
those varible should be independent.
```

In [ ]:
```
Question 2
a)
```

In [7]:
```
data(prostate)
head(prostate)
```

lcavol	lweight	age	lbph	svi	lcp	gleason	pgg45	lpsa
-0.5798185	2.7695	50	-1.386294	0	-1.38629	6	0	-0.43078
-0.9942523	3.3196	58	-1.386294	0	-1.38629	6	0	-0.16252
-0.5108256	2.6912	74	-1.386294	0	-1.38629	7	20	-0.16252
-1.2039728	3.2828	58	-1.386294	0	-1.38629	6	0	-0.16252
0.7514161	3.4324	62	-1.386294	0	-1.38629	6	0	0.37156
-1.0498221	3.2288	50	-1.386294	0	-1.38629	6	0	0.76547

In [8]:
```
y1 = prostate $ lpsa
x1 = prostate $ lcavol
mean_x = mean(x1)
mean_y = mean(y1)
beta1 = sum(((y1-mean_y))*(x1-mean_x))/sum((x1-mean_x)^2)
beta0 = mean_y - beta1 * mean_x
SSR = sum((beta0+beta1*x1 - mean_y)^2)
SST = sum((y1-mean_y)^2)
SSE = sum((y1 - beta0 - beta1*x1)^2)
variance = SSE / (nrow(prostate)-2)
s = sqrt(variance)
s
R2_1 = SSR/SST
R2_1

# Becuase this is a simple linear regression, so we would like to use
# beta1 and beta0 to calcuate SSR, SST, and SSE. The residual standard
# error is the sqrt of variance, which is 0.787499423513712, and the
# R^2 is 0.539431908779019
SSR
```

0.787499423513712

0.539431908779019

69.0028264997623

In [9]:
```
lweight = prostate$lweight
svi = prostate$ svi
lbph = prostate$lbph
age = prostate$age
lcp = prostate$lcp
pgg45 = prostate$pgg45
gleason = prostate$gleas

# lpsa~lcavol 's r-square
summary(lm(y1~x1))$r.squared
# lpsa~lcavol 's sigma
summary(lm(y1~x1))$sigma

# lpsa~lcavol+lweight r-square
```

```
summary(lm(y1~x1+lweight))$r.squared
# lpsa~lcavol+lweight sigma
summary(lm(y1~x1+lweight))$sigma

# lpsa~lcavol+lweight+svi r-square
summary(lm(y1~x1+lweight+svi))$r.squared
# lpsa~lcavol+lweight+svi sigma
summary(lm(y1~x1+lweight+svi))$sigma

# lpsa~lcavol+lweight+svi+lbph r-square
summary(lm(y1~x1+lweight+svi+lbph))$r.squared
# lpsa~lcavol+lweight+svi+lbph sigma
summary(lm(y1~x1+lweight+svi+lbph))$sigma

# lpsa~lcavol+lweight+svi+lbph+age r-square
summary(lm(y1~x1+lweight+svi+lbph+age))$r.squared
# lpsa~lcavol+lweight+svi+lbph+age sigma
summary(lm(y1~x1+lweight+svi+lbph+age))$sigma

# lpsa~lcavol+lweight+svi+lbph+age+lcp r-square
summary(lm(y1~x1+lweight+svi+lbph+age+lcp))$r.squared
# lpsa~lcavol+lweight+svi+lbph+age+lcp sigma
summary(lm(y1~x1+lweight+svi+lbph+age+lcp))$sigma

# lpsa~lcavol+lweight+svi+lbph+age+lcp+pgg45 r-square
summary(lm(y1~x1+lweight+svi+lbph+age+lcp+pgg45))$r.squared
# lpsa~lcavol+lweight+svi+lbph+age+lcp+pgg45 sigma
summary(lm(y1~x1+lweight+svi+lbph+age+lcp+pgg45))$sigma

# lpsa~lcavol+lweight+svi+lbph+age+lcp+pgg45+gleason r-square
summary(lm(y1~x1+lweight+svi+lbph+age+lcp+pgg45+gleason))$r.squared
# lpsa~lcavol+lweight+svi+lbph+age+lcp+pgg45+gleason sigma
summary(lm(y1~x1+lweight+svi+lbph+age+lcp+pgg45+gleason))$sigma

r_squared = vector()
sigma = vector()
r_squared [1] = summary(lm(y1~x1))$r.squared
sigma[1] = summary(lm(y1~x1))$sigma
r_squared[2] = summary(lm(y1~x1+lweight))$r.squared
sigma[2] = summary(lm(y1~x1+lweight))$sigma

r_squared[3] = summary(lm(y1~x1+lweight+svi))$r.squared
```

```
sigma[3] = summary(lm(y1~x1+lweight+svi))$sigma

r_squared[4] = summary(lm(y1~x1+lweight+svi+lbph))$r.squared
sigma[4] = summary(lm(y1~x1+lweight+svi+lbph))$sigma

r_squared[5] = summary(lm(y1~x1+lweight+svi+lbph+age))$r.squared
sigma[5] = summary(lm(y1~x1+lweight+svi+lbph+age))$sigma

r_squared[6]= summary(lm(y1~x1+lweight+svi+lbph+age+lcp))$r.squared
sigma[6]=summary(lm(y1~x1+lweight+svi+lbph+age+lcp))$sigma

r_squared[7] = summary(lm(y1~x1+lweight+svi+lbph+age+lcp+pgg45))$r.squared
sigma[7] = summary(lm(y1~x1+lweight+svi+lbph+age+lcp+pgg45))$sigma

r_squared[8] = summary(lm(y1~x1+lweight+svi+lbph+age+lcp+pgg45+gleason))$r.squared
sigma[8] = summary(lm(y1~x1+lweight+svi+lbph+age+lcp+pgg45+gleason))$sigma
```

0.539431908779019

0.787499423513711

0.585934512070213

0.750646932552003

0.626440253553244

0.71680938995835

0.636603479801418

0.710823197727069

0.644102401261455

0.707305372441944

0.645112974108872

0.710213512046953

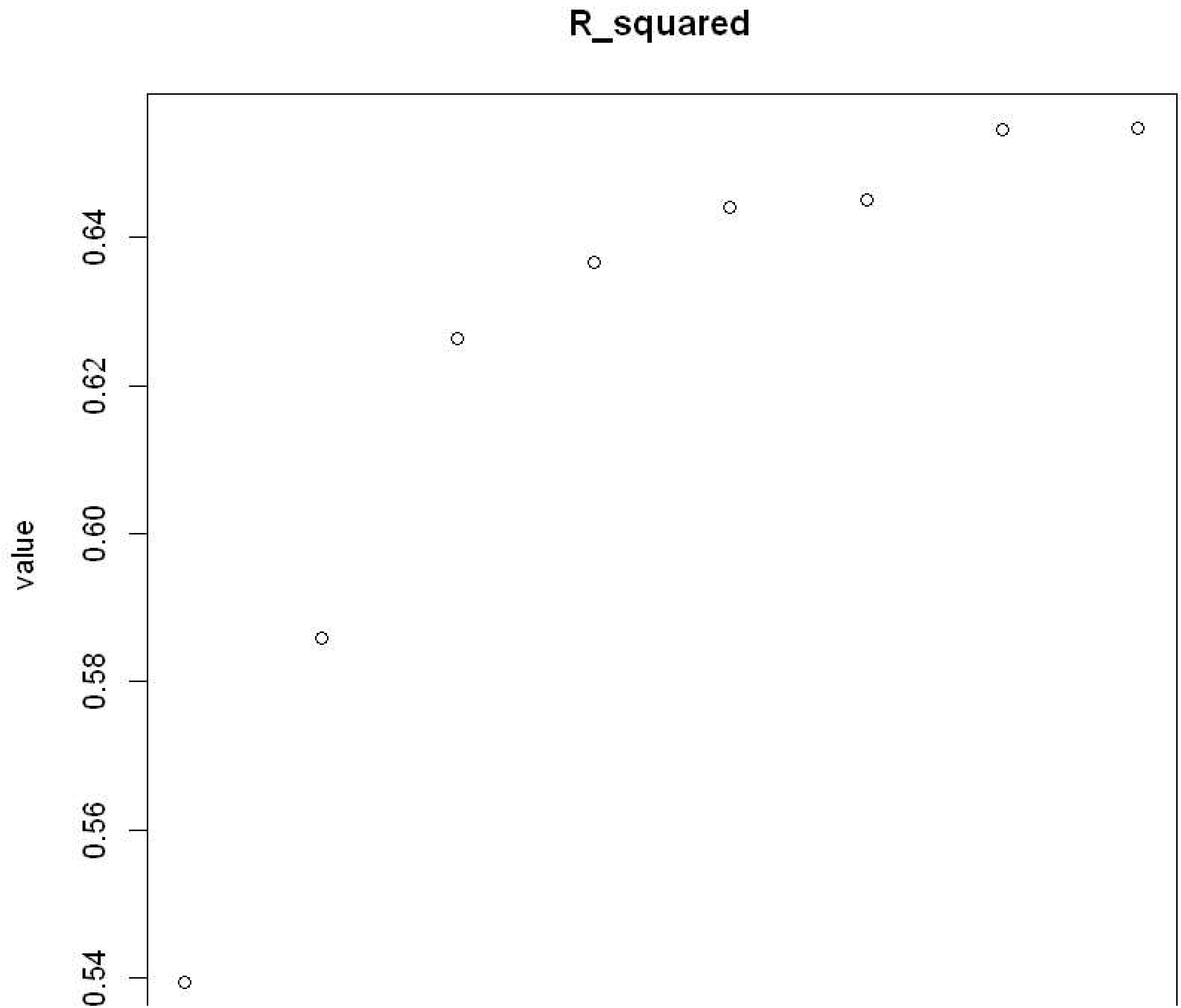
0.65443165616093

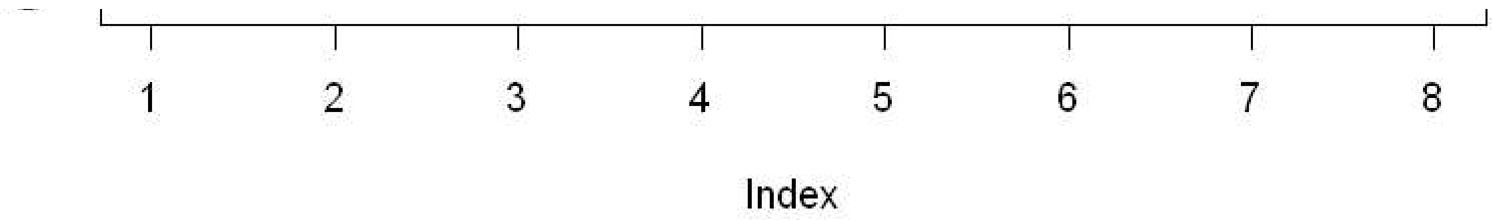
0.704753265042738

0.654754085299708

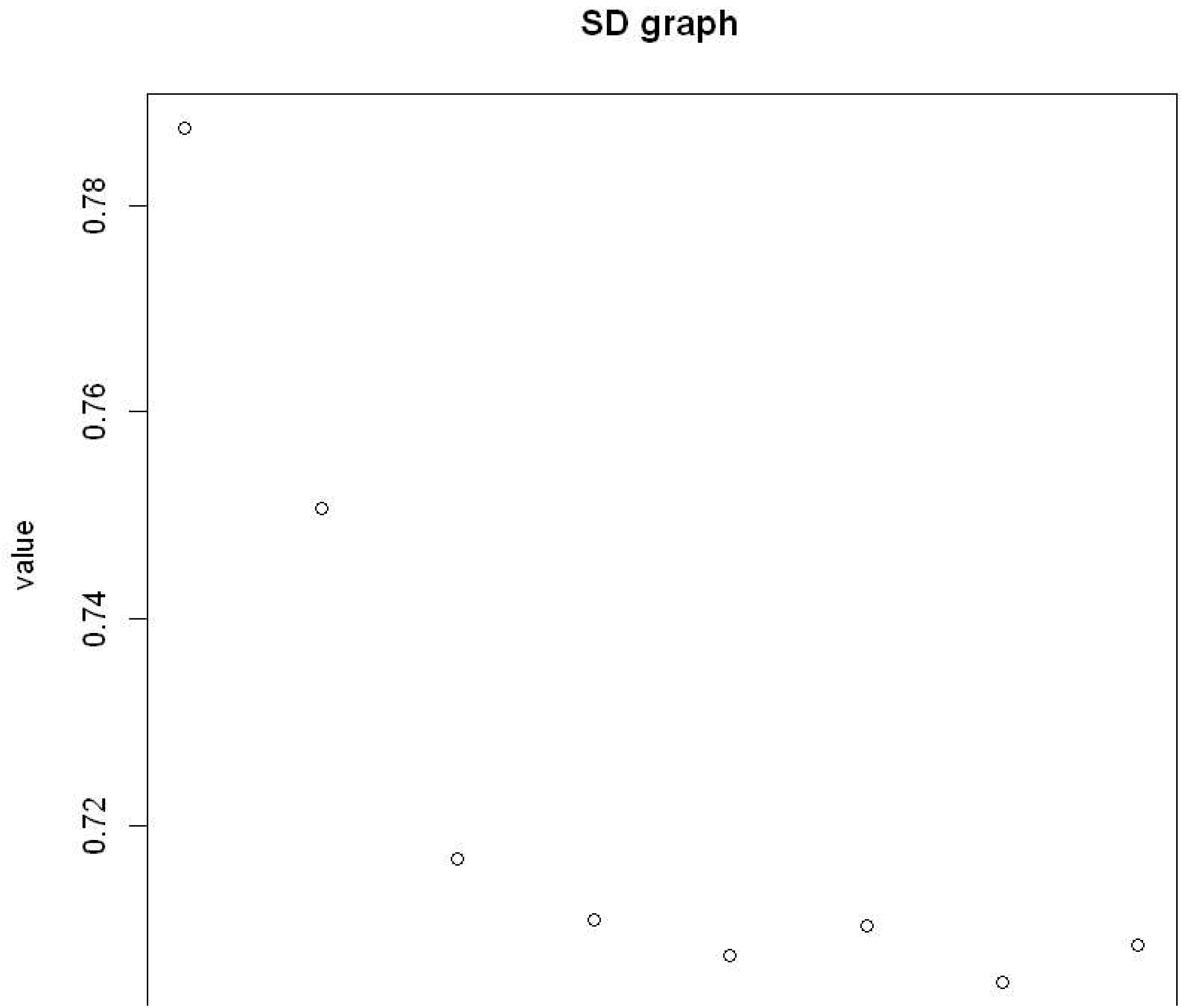
0.708415511834863

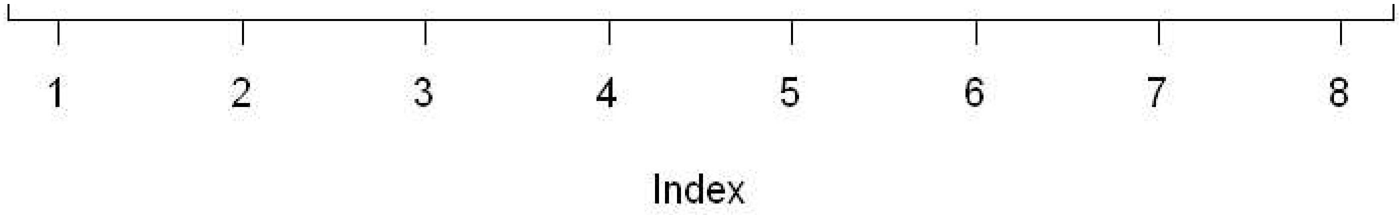
In [11]:
```
p = r_squared
plot(p,main = "R_squared",ylab = "value")
# As the function states, this r_squared graph generate four
# plots
```


R_squared
value
0.54
0.56
0.58
0.60
0.62
0.64


1
2
3
4
5
6
7
8
Index

In [13]:
```
s = sigma
plot(s,main = "SD graph",ylab = "value")
# This is the plot for sigma trend
```


SD graph
value
0.78
0.76
0.74
0.72



In [ ]:
```
Question 3
a)
```

In [14]:
```
data(truck)
truck$B= sapply(truck$B, function(x) ifelse(x == "-" , -1, 1))
truck$C =sapply(truck$C, function(x) ifelse(x == "-" , -1, 1))
truck$D =sapply(truck$D, function(x) ifelse(x == "-" , -1, 1))
truck$E = sapply(truck$E, function(x) ifelse(x == "-" , -1, 1))
truck$O =sapply(truck$O, function(x) ifelse(x == "-" , -1, 1))
```

In [15]:
```
y = truck$height
B = truck$B
C = truck$C
D = truck$D
E = truck$E
O = truck$O
X = cbind("Intercept" = 1,B, C, D, E, O)
beta = solve(t(X) %*% X) %*% t(X) %*% y
beta
# Getting the result by utiliting the formula in the lecture note.
# The result appears in the follow table, for height~B, C, D, E, O, respectively,
# their regression coefficients are 0.1106250,-0.0881250,-0.0143750,
# 0.0518750 and -0.1297917
```

Intercept	7.6360417
B	0.1106250
C	-0.0881250
D	-0.0143750
E	0.0518750
O	-0.1297917