Suffolk Scarp Analysis R Code:

```
##need package:gdata
##setting Perl PATH to read excel files
perl <- "C:\\Perl64\\bin\\perl.exe"

##test
#df <- read.xls("001.xls", sheet=1, perl=perl)



### feet to meter conversion

cov = 0.3048


#############

L = list.files(path = ".", pattern = 'xls')


MYLIST = vector(mode='list')


for(i in 1:length(L))
{
  MYLIST[[i]] = read.xls(L[i], sheet=1, perl=perl)

}



baserow = vector(mode='numeric')


for(i in 1:length(MYLIST))
{
  G = MYLIST[[i]]
  x = G[,3]
  y = G[,4]

  plot(x,y, type = 'l', xlab = "M", ylab= "Z", main = i)
  axis(side=3, at=c(0,4000), labels=c('W', 'E'))

  b = identify(x, y, n=1, labels = i, plot = T)
```

```
  baserow[[i]] = b


}


write(baserow, file = "base_row_suff")


baserow = scan("base_row_suff")



basesx = vector(mode='numeric')
basesy = vector(mode='numeric')

for(i in 1:length(MYLIST))
{

  G = MYLIST[[i]]
  B = baserow[[i]]
  x = G[B,3]
  y = G[B,4]
  basesx[[i]] = x
  basesy[[i]] = y

  ## x values are along each cross section --> not significant to plot
  ## just plot y values (elevation) vs. index of ys

}

write(basesx, file = "base_points_x_suff")
write(basesy, file = "base_points_y_suff")

A = scan("base_points_x_suff")

Bft = scan("base_points_y_suff")

B = Bft*cov


###Raw data plot
xleft = c(24,55,117,141,145,185)
ybottom = rep(-1, times=length(xleft))
xright = c(31,85,135,142,148,230)
ytop = rep(10, times=length(xleft))
```

```
plot(B, type = 'l', ylab = 'Elevation (m)',
     main = 'Long Wavelength of Suffolk Scarp',
     xlab = "Distance along scarp (km)",
     panel.first = rect(xleft, ybottom, xright, ytop,
                        col='gray', border=NA))

lines(lowess(B, f = .5, iter = 10), col = "red", lwd=2)
axis(side=3, at=c(0,230), labels=c('N', 'S'))



###Raw data in reverse order (S-N) to match Oran
C = rev(B)

plot(C, type = 'l', ylab = 'Elevation (m)',
     main = 'Long Wavelength of Suffolk Scarp',
     xlab = "Distance along scarp (km)")

lines(lowess(C, f = .5, iter = 10), col = "red", lwd=2)
axis(side=3, at=c(0,230), labels=c('S', 'N'))


###Removing Values without definite scarp position

bad = c(24:31, 55:85, 117:135, 141:142, 145:148,
        185:230)

B[bad] = NA

exall = c(1:length(B))
exall[bad] = NA

ex = na.omit(exall)
why = na.omit(B)


#New plot
plot(ex, why, type = 'l', ylab = 'Elevation (m)',
     main = 'Long Wavelength of Suffolk Scarp',
     xlab = "Distance along scarp (km)",
     panel.first = rect(xleft, ybottom, xright, ytop,
                        col='gray', border=NA))
lines(lowess(ex, why, f = .5, iter = 10), col = 'red', lwd=2)
axis(side=3, at=c(0,185), labels=c('N', 'S'))

#no gray
plot(ex, why, type = 'l', ylab = 'Elevation (m)',
```

```
     main = 'Long Wavelength of Suffolk Scarp',
     xlab = "Distance along scarp (km)")
lines(lowess(ex, why, f = .5, iter = 10), col = 'red', lwd=2)
axis(side=3, at=c(0,185), labels=c('N', 'S'))


#New plot, reverse
plot(ex, why, type = 'l', ylab = 'Elevation (m)',
     xlim=rev(range(ex)),
     main = 'Long Wavelength of Suffolk Scarp',
     xlab = "Distance along scarp (km)")
lines(lowess(ex, why, f = .5, iter = 10), col = 'red', lwd=2)
axis(side=3, at=c(0,185), labels=c('N', 'S'))



##########################remove last trough###################
###doesnt do anything###

xleft = c(14,34,42,55,117,141,145,185)
ybottom = rep(-1, times=length(xleft))
xright = c(31,35,43,85,135,142,148,230)
ytop = rep(10, times=length(xleft))


bad = c(14:31, 34:35, 42:43, 55:85, 117:135, 141:142, 145:148,
        185:230)

B[bad] = NA

exall = c(1:length(B))
exall[bad] = NA

ex = na.omit(exall)
why = na.omit(B)


#New plot
plot(ex, why, type = 'l', ylab = 'Elevation (m)',
     main = 'Long Wavelength of Suffolk Scarp',
     xlab = "Distance along scarp (km)",
     panel.first = rect(xleft, ybottom, xright, ytop,
                        col='gray', border=NA))
lines(lowess(ex, why, f = .5, iter = 10), col = 'red', lwd=2)
axis(side=3, at=c(0,185), labels=c('N', 'S'))

#no gray
plot(ex, why, type = 'l', ylab = 'Elevation (m)',
```

```
     main = 'Long Wavelength of Suffolk Scarp',
     xlab = "Distance along scarp (km)")
lines(lowess(ex, why, f = .5, iter = 10), col = 'red', lwd=2)
axis(side=3, at=c(0,185), labels=c('N', 'S'))


#New plot, reverse
plot(ex, why, type = 'l', ylab = 'Elevation (m)',
     xlim=rev(range(ex)),
     main = 'Long Wavelength of Suffolk Scarp',
     xlab = "Distance along scarp (km)")
lines(lowess(ex, why, f = .5, iter = 10), col = 'red', lwd=2)
axis(side=3, at=c(0,185), labels=c('N', 'S'))


#####################################################################


Orangeburg Scarp Analysis R Code:

##need package:gdata
##setting Perl PATH to read excel files
perl <- "C:\\Perl64\\bin\\perl.exe"

##test
#df <- read.xls("001.xls", sheet=1, perl=perl)



### feet to meter conversion

cov = 0.3048


############

###changed list name from capital L to capital O
###add _O to all vectors and saved things


O = list.files(path = ".", pattern = 'xls')


MYLIST_O = vector(mode='list')


for(i in 1:length(O))
{
```

```
  MYLIST_O[[i]] = read.xls(O[i], sheet=1, perl=perl)

}



baserow_O = vector(mode='numeric')


for(i in 1:length(MYLIST_O))
{
  G = MYLIST_O[[i]]
  x = G[,3]
  y = G[,4]

  plot(x,y, type = 'l', xlab = "M", ylab= "Z", main = i)
  axis(side=3, at=c(0,14000), labels=c('NW', 'SE'))

  b = identify(x, y, n=1, labels = i, plot = T)

  baserow_O[[i]] = b


}



write(baserow_O, file = "base_row_oran")


baserow_O = scan("base_row_oran")



basesx_O = vector(mode='numeric')
basesy_O = vector(mode='numeric')

for(i in 1:length(MYLIST_O))
{

  G = MYLIST_O[[i]]
  B = baserow_O[[i]]
  x = G[B,3]
  y = G[B,4]
  basesx_O[[i]] = x
  basesy_O[[i]] = y

  ## x values are along each cross section --> not significant to plot
```

```
  ## just plot y values (elevation) vs. index of ys

}

write(basesx_O, file = "base_points_x_oran")
write(basesy_O, file = "base_points_y_oran")


Bft = scan("base_points_y_oran")

B = Bft*cov

A = seq(from = 1, to = 3*length(B), by = 3)


###Raw data plot
xleft = c(1,12,17,22,38,59,75,95,110,145,153,173,177)
ybottom = rep(20, times=length(xleft))
xright = c(2,14,19,24,45,61,91,96,143,149,160,175,185)
ytop = rep(100, times=length(xleft))
singles = c(4,8,29,33,35,66,70,93,102,107,163,166,169,171)


plot(x = A, y = B, type = 'l', ylab = 'Elevation (m)',
     main = 'Long Wavelength of Orangeburg Scarp',
     xlab = "Distance along scarp (km)",
     panel.first = rect(A[xleft], ybottom, A[xright], ytop,
                        col='gray', border=NA))
abline(v=A[singles], col='gray')
lines(lowess(A, B, f = .5, iter = 10), col = 2, lwd=2)
axis(side=3, at=c(0,553), labels=c('SW', 'NE'))
abline(v=A[88], col='blue', lwd=1, lty=2)
axis(side=3, pos=90, at=A[c(80,97)], labels=c('SC','NC'), tick=F)

#No gray bars
plot(x = A, y = B, type = 'l', ylab = 'Elevation (m)',
     main = 'Long Wavelength of Orangeburg Scarp',
     xlab = "Distance along scarp (km)")

lines(lowess(A, B, f = .5, iter = 10), col = 2, lwd=2)
axis(side=3, at=c(0,553), labels=c('SW', 'NE'))
abline(v=A[88], col='blue', lwd=1, lty=2)
axis(side=3, pos=90, at=A[c(80,97)], labels=c('SC','NC'), tick=F)

###Removing Values without definite scarp position


bad = c(1:2, 4, 8, 12:14, 17:19, 22:24, 29, 33, 35, 38:45, 59:61,
```

```
        66, 70, 75:91, 93, 95:96, 102, 107, 110:143, 145:149,
        153:160, 163, 166, 169, 171, 173:175, 177:185)

B[bad] = NA

exall = seq(from = 1, to = 3*length(B), by = 3)
exall[bad] = NA

ex = na.omit(exall)
why = na.omit(B)

#New plot
plot(ex, why, type = 'l', ylab = 'Elevation (m)',
     main = 'Long Wavelength of Orangeburg Scarp',
     xlab = "Distance along (km)",
     panel.first = rect(A[xleft], ybottom, A[xright], ytop,
                        col='gray', border=NA))
abline(v=A[singles], col='gray')
lines(lowess(ex, why, f = .5, iter = 10), col = 2, lwd=2)
axis(side=3, at=c(0,500), labels=c('SW', 'NE'))
abline(v=A[88], col='blue', lwd=1, lty=2)
axis(side=3, pos=90, at=A[c(80,97)], labels=c('SC','NC'), tick=F)

#no gray
plot(ex, why, type = 'l', ylab = 'Elevation (m)',
     main = 'Long Wavelength of Orangeburg Scarp',
     xlab = "Distance along (km)")

lines(lowess(ex, why, f = .5, iter = 10), col = 2, lwd=2)
axis(side=3, at=c(0,500), labels=c('SW', 'NE'))
abline(v=A[88], col='blue', lwd=1, lty=2)
axis(side=3, pos=90, at=A[c(80,97)], labels=c('SC','NC'), tick=F)

#################################################################
```