California data

```
#################################################################
##Read GPS data
#################################################################
skipn <- 21
chil <- read.fwf(paste('chil','CleanFlt.neu',sep=''),
      widths=c(9,5,4,8,8,8,8,8,8),col.names=c('yydd','yy','dd','n','e','u',
      'sdn','sde','sdu'),skip=skipn)
lbc2 <- read.fwf(paste('lbc2','CleanFlt.neu',sep=''),
      widths=c(9,5,4,8,8,8,8,8,8),col.names=c('yydd','yy','dd','n','e','u',
      'sdn','sde','sdu'),skip=skipn)
a <- seq( 1996183, 1996366, 1 )
b <- seq( 1997001, 1997365, 1 )
c <- seq( 1998001, 1998365, 1 )
d <- seq( 1999001, 1999365, 1 )
e <- seq( 2000001, 2000366, 1 )
f <- seq( 2001001, 2001365, 1 )
g <- seq( 2002001, 2002365, 1 )
h <- seq( 2003001, 2003365, 1 )
i <- seq( 2004001, 2004366, 1 )
j <- seq( 2005001, 2005365, 1 )
k <- seq( 2006001, 2006365, 1 )
l <- seq( 2007001, 2007365, 1 )
m <- seq( 2008001, 2008366, 1 )
n <- seq( 2009001, 2009181, 1 )
Time <- c( d, e,f, g, h, i, j, k, l, m,n)
lt <- length(Time)
indt <- 1:lt

#################################################################
##Let the missing data equal to NA for chil station
#################################################################
tempn <- chil$n
tempe <- chil$e
tempu <- chil$u
lao <- length(tempn)
tempT <- chil$yy * 1000 + chil$dd
chn <- rep(NA,lt)
che <- rep(NA,lt)
chu <- rep(NA,lt)
for (i in 1:lao)
{
  chn[Time==tempT[i]] <- tempn[i]
  che[Time==tempT[i]] <- tempe[i]
  chu[Time==tempT[i]] <- tempu[i]
}

#################################################################
## Interpolation for the missing data for chil station
#################################################################
```

```
indmT <- indt[is.na(chn)]
for (i in indmT)
{
 if (i < 10){
 chn[i] <- mean( (chn[i:(i+10)])[!is.na(chn[i:(i+10)])] )
 che[i] <- mean( (che[i:(i+10)])[!is.na(che[i:(i+10)])] )
 chu[i] <- mean( (chu[i:(i+10)])[!is.na(chu[i:(i+10)])] )
 }else{
 chn[i] <- mean( (chn[(i-10):(i+10)])[!is.na(chn[(i-10):(i+10)])] )
 che[i] <- mean( (che[(i-10):(i+10)])[!is.na(che[(i-10):(i+10)])] )
 chu[i] <- mean( (chu[(i-10):(i+10)])[!is.na(chu[(i-10):(i+10)])] )
 }
}

#############################################################
##Let the missing data equal to NA for lbc2 station
#############################################################
tempn <- lbc2$n
tempe <- lbc2$e
tempu <- lbc2$u
lao <- length(tempn)
tempT <- lbc2$yy * 1000 + lbc2$dd
lb2n <- rep(NA,lt)
lb2e <- rep(NA,lt)
lb2u <- rep(NA,lt)
for (i in 1:lao)
{
 lb2n[Time==tempT[i]] <- tempn[i]
 lb2e[Time==tempT[i]] <- tempe[i]
 lb2u[Time==tempT[i]] <- tempu[i]
}

#############################################################
## Interpolation for the missing data for lbc2 station
#############################################################
indmT <- indt[is.na(lb2n)]
for (i in indmT)
{
 if (i < 10){
 lb2n[i] <- mean( (lb2n[i:(i+10)])[!is.na(lb2n[i:(i+10)])] )
 lb2e[i] <- mean( (lb2e[i:(i+10)])[!is.na(lb2e[i:(i+10)])] )
 lb2u[i] <- mean( (lb2u[i:(i+10)])[!is.na(lb2u[i:(i+10)])] )
 }else{
 lb2n[i] <- mean( (lb2n[(i-10):(i+10)])[!is.na(lb2n[(i-10):(i+10)])] )
 lb2e[i] <- mean( (lb2e[(i-10):(i+10)])[!is.na(lb2e[(i-10):(i+10)])] )
 lb2u[i] <- mean( (lb2u[(i-10):(i+10)])[!is.na(lb2u[(i-10):(i+10)])] )
 }
}

#############################################################
####### Calculate the baseline between the stations chil and lbc2
```

```
####### *1000 is to make the measurements in mm
#############################################################
#### CHIL-LBC2
#########
TdHNWn <- (chn-lb2n)*1000
TdHNWe <- (che-lb2e)*1000
TdHNWu <- (chu-lb2u)*1000

#################################################################
####### Trend of the GPS in previous lint=10 days ###########
#################################################################
ind <- 1:lt
lTd <- length(ind)
lint <- 10
dtPSn <- dtPSe <- dtPSu <- rep(0,lTd)
for (t in lint:lTd)
{
  dtPSn[t] <- as.numeric(lm(TdHNWn[(t-lint+1):t] ~ c(1:lint))$coefficients[2])
  dtPSe[t] <- as.numeric(lm(TdHNWe[(t-lint+1):t] ~ c(1:lint))$coefficients[2])
  dtPSu[t] <- as.numeric(lm(TdHNWu[(t-lint+1):t] ~ c(1:lint))$coefficients[2])
}

#################################################################
#### Maximum trend of the GPS in previous lint=10 days #####
#################################################################
lTd <- length(dtPSn)
lint <- 10
 dnma <- dtPSn
 for (t in lint:lTd)
 dnma[t] <- max(dtPSn[(t-lint+1):t])

 dema <- dtPSe
 for (t in lint:lTd)
 dema[t] <- max(dtPSe[(t-lint+1):t])

 duma <- dtPSu
 for (t in lint:lTd)
 duma[t] <- max(dtPSu[(t-lint+1):t])

#################################################################
#### Minimum trend of the GPS in previous lint=10 days #####
#################################################################
lTd <- length(dtPSn)
lint <- 10
 dnmi <- dtPSn
 for (t in lint:lTd)
 dnmi[t] <- min(dtPSn[(t-lint+1):t])

 demi <- dtPSe
 for (t in lint:lTd)
 demi[t] <- min(dtPSe[(t-lint+1):t])
```

```
 dumi <- dtPSu
 for (t in lint:lTd)
 dumi[t] <- min(dtPSu[(t-lint+1):t])

#############################################################
##### Calculate the moving rate of variation (MRV) #####
##### in Section 3 GRL 2013 paper, i.e., max - min #####
#############################################################
 dnmai <- dnma - dnmi
 demai <- dema - demi
 dumai <- duma - dumi

#############################################################
nfn <- dnmai[20:lTd]
nfe <- demai[20:lTd]
nfu <- dumai[20:lTd]
lTd1 <- length(nfn)
R <- as.matrix( cbind( nfn, nfe, nfu ) )


#################################################################
#### Read earthquake data to get the large earthquake occurrence times      #####
#################################################################
# Times is a function to convert time into days
Times <- function(Otime)
      {
        Year <- floor(Otime/100000000)
        Month <- floor(Otime/1000000)-Year*100
        Day <- floor(Otime/10000)-Year*10000-Month*100
        Days <- julian(as.Date(paste(Year,Month,Day,sep="-")))
        Min <- (Otime/100-floor(Otime/100))*100/60/24  #Minute to day
        Hour <- (floor(Otime/100)/100-floor(floor(Otime/100)/100))*100/24
        ( Days+Hour+Min-julian(as.Date("1999-01-19")) )
      }
####
eqT4 <- scan('EQ2CA1999-2009.txt',
     what=list(Year=0,Month=0,Day=0,Hour=0,
     Minute=0,Second=0,Magnitude=0,Latitude=0,Longitude=0,Depth=0),sep=',')
eqMT4 <- eqT4$Magnitude
eqtimeT4O <- eqT4$Year*100000000 + eqT4$Month*1000000 + eqT4$Day*10000 +
         eqT4$Hour*100 + eqT4$Minute
eqtimeT4 <- Times(eqtimeT4O) # earthquake occurrence times in days
                    # The first day is when the GPS data started
                    # 1999-01-20 (first 19 days of GPS data are not
                    # counted as we used them to calculate the first MRV)

Mc <- 4.649 #magnitude cutoff
eqt <- floor(eqtimeT4) # As GPS data is in days, we get rid of the decimal places

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

```
### To see the earthquakes with minimum magnitude Mc in the catalogue
#############################################################
cbind(eqT4$Year[eqt > 0 & eqMT4 > Mc],eqT4$Month[eqt > 0 & eqMT4 > Mc],
  eqT4$Day[eqt > 0 & eqMT4 > Mc],eqT4$Latitude[eqt > 0 & eqMT4 > Mc],
  eqT4$Longitude[eqt > 0 & eqMT4 > Mc],eqT4$Depth[eqt > 0 & eqMT4 > Mc],
  eqT4$Magnitude[eqt > 0 & eqMT4 > Mc])

#############################################################
tT <- eqt[eqt > 0 & eqMT4 > Mc]
temM <- eqMT4[eqt>0 & eqMT4 > Mc]
tM <- temM[1]
temt <- c(2,9,12,13,14,16:29) # When there are more than 1 earthquake in one day,
                    # we only use the largest earthquake in that day
for (i in temt){
  tM <- append(tM,temM[i])
}

#############################################################
##  Convert the earthquake data into 0-1 sequence
#############################################################
eqM <- seq(0,0,length.out=length(nfn))
eqM[unique(eqt[eqt > 0 & eqMT4 > Mc])] <- tM
eq <- seq(0,0,length.out=length(nfn))
eq[unique(eqt[eqt > 0 & eqMT4 > Mc])] <- 1

#####################################################################
#########   Molchan Diagram - GPS signals predict EQ    ############
#####################################################################
y4 <- x4 <- NULL
## next intv to intv+10 days
lTd1 <- length(R[,1])
intv <- 20
mm <- sqrt(nfn^2+nfe^2+(nfu/4)^2) ## Try different combinations
eqo <- mm1 <- NULL
for (i in 1:(floor(lTd1/20)-1))
{
  mm1[i] <- max(mm[((i-1)*intv+1):(i*intv)])
  eqo[i] <- max(eq[(i*intv+1):((i+1)*intv)])
}
ind <- 1:length(mm1)
cc <- seq(0,max(mm1),length.out=500)
mdiag <- function(ct){
c1 <- ct
mmt <- rep(0,length(mm1))
mmt[mm1 >= c1] <- 1
nopred <- 0
for (i in 1:length(mm1)){
  if (mmt[i]+eqo[i]>1.5) nopred <- nopred + 1
}
y4 <- (sum(eqo)-nopred)/sum(eqo)
x4 <- sum(mmt)/length(mm1)
```

```
c(y4,x4)
}
yx4 <- apply(t(cc),2,mdiag)
y4 <- yx4[1,]
x4 <- yx4[2,]

########################
y <- x <- NULL
## next intv to intv+10 days
lTd1 <- length(R[,1])
intv <- 20
mm <- sqrt(nfn^2+nfe^2) ## Try different combination
eqo <- mm1 <- NULL
for (i in 1:(floor(lTd1/20)-1))
{
  mm1[i] <- max(mm[((i-1)*intv+1):(i*intv)])
  eqo[i] <- max(eq[(i*intv+1):((i+1)*intv)])
}
#plot(mm1)
ind <- 1:length(mm1)
cc <- seq(0,max(mm1),length.out=500)
for (jj in 1:length(cc)){
c1 <- cc[jj]
mmt <- rep(0,length(mm1))
mmt[mm1 >= c1] <- 1
nopred <- 0
for (i in 1:length(mm1)){
  if (mmt[i]+eqo[i]>1.5) nopred <- nopred + 1
}
y <- append(y,(sum(eqo)-nopred)/sum(eqo))
x <- append(x,sum(mmt)/length(mm1))
}

Cgey <- y; Cgex <- x; Cgey4 <- y4; Cgex4 <- x4
plot(x4,y4,type='l',xlab='Proportion of alarmed cells',
    ylab='Proportion of unpredicted earthquakes',cex=2,main='Southern California',
    cex.axis=2,cex.lab=2,lwd=2,cex.main=2)
lines(c(0,1),c(1,0),lty='dashed',lwd=2)
lines(x,y,lwd=2,col='red',lty='longdash')
legend(0.39,0.95,'l=20 days, A1',lty='solid',col='black',bty='n',cex=1.7,lwd=2)
legend(0.39,0.9,'l=20 days, A2',lty='longdash',col='red',bty='n',cex=1.7,lwd=2)
text(0.98,0.98,'(b)',cex=2)


#########################################################################
#########   Molchan Diagram - EQ predict GPS signals    ############
#########################################################################
########################
lTd1 <- length(R[,1])
intv <- 10
mm <- sqrt(nfn^2+nfe^2+(nfu/4)^2) ## Try different combination
```

```
eqo <- eqmm <- mm1 <- NULL
for (i in 1:(floor(lTd1/10)-1))
{
  mm1[i] <- max(mm[((i-1)*intv+1):(i*intv)])
  eqo[i] <- max(eq[((i-1)*intv+1):(i*intv)])
  eqmm[i] <- max(eqM[((i-1)*intv+1):(i*intv)])
}
cgps <- quantile(mm1,0.5)
mgps <- rep(0,length(mm1))
mgps[mm1 > cgps] <- 1
lend <- (floor(lTd1/10)-1)
ind <- 1:lend
inde <- ind[eqo>0.5]
fxt <- rep(0,length(eqo))
for (i in inde)
  fxt <- fxt + c(rep(0,i), 10^(0.75*(eqmm[i]-Mc))/(ind[(i+1):lend]-i) )
eqoo <- eqo
gpss <- mgps
ww <- c(0,unique(sort(fxt)))
y4 <- x4 <- NULL
for (jj in 1:length(ww)){
c1 <- ww[jj]
inds <- (1:length(fxt))[fxt>=c1]
mmt <- rep(0,length(fxt))
mmt[inds] <- 1
predeq <- 0
for (ki in 1:length(fxt)){
  predeq <- predeq + sum(mmt[ki]==1&gpss[ki]==1)
}
y4 <- append(y4,(sum(gpss)-predeq)/sum(gpss))
x4 <- append(x4,sum(mmt)/(lend))
}


lTd1 <- length(R[,1])
intv <- 10
mm <- sqrt(nfn^2+nfe^2+(nfu/4)^2) ## Try different combination
eqo <- eqmm <- mm1 <- NULL
for (i in 1:(floor(lTd1/10)-1))
{
  mm1[i] <- max(mm[((i-1)*intv+1):(i*intv)])
  eqo[i] <- max(eq[((i-1)*intv+1):(i*intv)])
  eqmm[i] <- max(eqM[((i-1)*intv+1):(i*intv)])
}
cgps <- quantile(mm1,0.75)
mgps <- rep(0,length(mm1))
mgps[mm1 > cgps] <- 1
lend <- (floor(lTd1/10)-1)
ind <- 1:lend
inde <- ind[eqo>0.5]
fxt <- rep(0,length(eqo))
```

```
for (i in inde)
  fxt <- fxt + c(rep(0,i), 10^(0.75*(eqmm[i]-Mc))/(ind[(i+1):lend]-i) )
eqoo <- eqo
gpss <- mgps
ww <- c(0,unique(sort(fxt)))
y <- x <- NULL
for (jj in 1:length(ww)){
c1 <- ww[jj]
inds <- (1:length(fxt))[fxt>=c1]
mmt <- rep(0,length(fxt))
mmt[inds] <- 1
predeq <- 0
for (ki in 1:length(fxt)){
  predeq <- predeq + sum(mmt[ki]==1&gpss[ki]==1)
}
y <- append(y,(sum(gpss)-predeq)/sum(gpss))
x <- append(x,sum(mmt)/(lend))
}


########################
lTd1 <- length(R[,1])
intv <- 10
mm <- sqrt(nfn^2+nfe^2+(nfu/4)^2) ## Try different combination
eqo <- eqmm <- mm1 <- NULL
for (i in 1:(floor(lTd1/10)-1))
{
  mm1[i] <- max(mm[((i-1)*intv+1):(i*intv)])
  eqo[i] <- max(eq[((i-1)*intv+1):(i*intv)])
  eqmm[i] <- max(eqM[((i-1)*intv+1):(i*intv)])
}
cgps <- quantile(mm1,0.85)
mgps <- rep(0,length(mm1))
mgps[mm1 > cgps] <- 1
lend <- (floor(lTd1/10)-1)
ind <- 1:lend
inde <- ind[eqo>0.5]
fxt <- rep(0,length(eqo))
for (i in inde)
  fxt <- fxt + c(rep(0,i), 10^(0.75*(eqmm[i]-Mc))/(ind[(i+1):lend]-i) )
eqoo <- eqo
gpss <- mgps
ww <- c(0,unique(sort(fxt)))
y1 <- x1 <- NULL
for (jj in 1:length(ww)){
c1 <- ww[jj]
inds <- (1:length(fxt))[fxt>=c1]
mmt <- rep(0,length(fxt))
mmt[inds] <- 1
predeq <- 0
for (ki in 1:length(fxt)){
```

```
  predeq <- predeq + sum(mmt[ki]==1&gpss[ki]==1)
}
y1 <- append(y1,(sum(gpss)-predeq)/sum(gpss))
x1 <- append(x1,sum(mmt)/(lend))
}


########################
lTd1 <- length(R[,1])
intv <- 10
mm <- sqrt(nfn^2+nfe^2+(nfu/4)^2) ## Try different combination
eqo <- eqmm <- mm1 <- NULL
for (i in 1:(floor(lTd1/10)-1))
{
  mm1[i] <- max(mm[((i-1)*intv+1):(i*intv)])
  eqo[i] <- max(eq[((i-1)*intv+1):(i*intv)])
  eqmm[i] <- max(eqM[((i-1)*intv+1):(i*intv)])
}
cgps <- quantile(mm1,0.95)
mgps <- rep(0,length(mm1))
mgps[mm1 > cgps] <- 1
lend <- (floor(lTd1/10)-1)
ind <- 1:lend
inde <- ind[eqo>0.5]
fxt <- rep(0,length(eqo))
for (i in inde)
  fxt <- fxt + c(rep(0,i), 10^(0.75*(eqmm[i]-Mc))/(ind[(i+1):lend]-i) )
eqoo <- eqo
gpss <- mgps
ww <- c(0,unique(sort(fxt)))
y2 <- x2 <- NULL
for (jj in 1:length(ww)){
c1 <- ww[jj]
inds <- (1:length(fxt))[fxt>=c1]
mmt <- rep(0,length(fxt))
mmt[inds] <- 1
predeq <- 0
for (ki in 1:length(fxt)){
  predeq <- predeq + sum(mmt[ki]==1&gpss[ki]==1)
}
y2 <- append(y2,(sum(gpss)-predeq)/sum(gpss))
x2 <- append(x2,sum(mmt)/(lend))
}

Cegy4 <- y4; Cegx4 <- x4; Cegy <- y; Cegx <- x; Cegy1 <- y1; Cegx1 <- x1; Cegy2 <- y2; Cegx2 <- x2

#postscript('MolchanDiagCAEQpredictGPS.eps',paper="special",width=8, height=8)
#par(mar=c(5,5,2,2))
plot(x4,y4,type='l',xlab='Proportion of alarmed cells',
    ylab='Proportion of unpredicted GPS signals',cex=2,
    cex.axis=2,cex.lab=2,lwd=2)
```

```
lines(c(0,1),c(1,0),lty='dashed',lwd=2)
lines(x,y,col='purple',lwd=2.5,lty='dotted')
lines(x1,y1,col='red',lwd=2,lty='twodash')
lines(x2,y2,col='blue',lwd=2,lty='dotdash')
legend(0.51,0.95,'0.5 quantile',lty='solid',bty='n',cex=1.7,lwd=2)
legend(0.51,0.9,'0.75 quantile',lty='dotted',col='purple',bty='n',cex=1.7,lwd=2.5)
legend(0.51,0.85,'0.85 quantile',lty='twodash',col='red',bty='n',cex=1.7,lwd=2)
legend(0.51,0.8,'0.95 quantile',lty='dotdash',col='blue',bty='n',cex=1.7,lwd=2)
text(0.98,0.98,'(e)',cex=2)
#dev.off()


#########################################################################
#########      Molchan Diagram - EQ predict EQs       ###########
#########################################################################
########################
lTd1 <- length(R[,1])
ind <- 1:lTd1
inde <- ind[eq>0.5]
fxt <- rep(0,length(eq))
for (i in inde)
  fxt <- fxt + c(rep(0,i), 10^(0.75*(eqM[i]-Mc))/(ind[(i+1):lTd1]-i) )
fxt <- fxt[(inde[1]+1):lTd1]
eqo <- eq[(inde[1]+1):lTd1]
ww <- c(0,unique(sort(fxt)))
y4 <- x4 <- NULL
for (jj in 1:length(ww)){
c1 <- ww[jj]
inds <- (1:length(fxt))[fxt>c1]
mmt <- rep(0,length(fxt))
mmt[inds] <- 1
predeq <- 0
for (ki in 1:length(fxt)){
  predeq <- predeq + sum(mmt[ki]==1&eqo[ki]==1)
}
y4 <- append(y4,(sum(eqo)-predeq)/sum(eqo))
x4 <- append(x4,sum(mmt)/(lTd1-inde[1]))
}

Ceey4 <- y4; Ceex4 <- x4
#postscript('MolchanDiagCAEQpredictEQ.eps',paper="special",width=8, height=8)
#par(mar=c(5,5,2,2))
plot(c(0,x4),c(1,y4),type='l',xlab='Proportion of alarmed cells',
    ylab='Proportion of unpredicted earthquakes',cex=2,
    cex.axis=2,cex.lab=2,lwd=2)
lines(c(0,1),c(1,0),lty='dashed',lwd=2)
text(0.98,0.98,'(h)',cex=2)
#dev.off()
```