dglo=function(x,loc1,sc1,sh1) ((sc1^-1)\*exp(-(1-sh1)\*((-sh1^-1)\*log(1-sh1\*(x-loc1)/sc1))))/(1+exp(-(((-sh1^-1)\*log(1-sh1\*(x-loc1)/sc1)))))^2

pglo=function(q,loc1,sc1,sh1){1/(1+exp(-(((-sh1^-1)\*log(1-sh1\*(q-loc1)/sc1)))))}

qglo=function(p,loc1,sc1,sh1) {loc1+sc1\*(1-((1-p)/p)^sh1)/sh1}

library(SPEI) # package SPEI is required.

library(zoo) # package zoo is required.

library(fitdistrplus) # package fitdistrplus is required.

latitude= -21.1 # - for South hemisphere + for North hemisphere

First.Year=1960 # Enter the first year

AWC=100 # Enter the Avaliable Water Capacity (millimeters)

setwd("C:/Users/User/Desktop") # Setting the working directory

##############Climatic Water Balance##############

dataset= as.matrix(read.table("myfile.txt", head=T))

#loading the txt file: 1st column Year, 2nd Month, 3rd Precipitation and 4th monthly average temperature (ºC)

#see exemple at the end of the code

pre=as.matrix(dataset[,3])

temp=as.matrix(dataset[,4])

EP1=t(as.matrix(thornthwaite(temp,latitude, na.rm = TRUE)))

EP=t(EP1)

period=12 #monthly data

timescale=12 #12 time scales ranging from 1 to 12 months

if (is.numeric(AWC)==FALSE) {print('Available Water Capacity (AWC) required')}

PEP=pre-EP

N=length(PEP)

NegAcum=matrix(NA,N,1)

Arm=matrix(NA,N,1)

ETR=matrix(NA,N,1)

Alt=matrix(NA,N,1)

Def=matrix(NA,N,1)

Exd=matrix(NA,N,1)

Ss=matrix(NA,N,1)

Su=matrix(NA,N,1)

S=matrix(NA,N,1)

Ls=matrix(NA,N,1)

Lu=matrix(NA,N,1)

L=matrix(NA,N,1)

R=matrix(NA,N,1)

PR=matrix(NA,N,1)

PRO=matrix(NA,N,1)

Pls=matrix(NA,N,1)

Plu=matrix(NA,N,1)

Pl=matrix(NA,N,1)

RO=matrix(NA,N,1)

CAFEC.EP=matrix(NA,N,1)

CAFEC.R=matrix(NA,N,1)

CAFEC.RO=matrix(NA,N,1)

CAFEC.L=matrix(NA,N,1)

CAFEC.P=matrix(NA,N,1)

d=matrix(NA,N,1)

NegAcum[1]=0

Alt[1]=0

Def[1]=0

Exd[1]=0

PR[1]=0

Arm[1]=AWC

ETR[1]=EP[1]

Ss[1]=0.25\*AWC

Su[1]=0.75\*AWC

R[1]=0

RO[1]=0

PRO[1]=AWC

if (Ss[1]<abs(PEP[1])){Ls[1]=Ss[1]}else{Ls[1]=abs(PEP[1])}

if (EP[1]<Ss[1]){Pls[1]=EP[1]}else{Pls[1]=Ss[1]}

Lu[1]=(EP[1]-pre[1]-Ls[1])\*Su[1]/AWC

L[1]=Ls[1]+Lu[1]

Lu[1]=(EP[1]-pre[1]-Ls[1])\*Su[1]/AWC

L[1]=Ls[1]+Lu[1]

Plu[1]=(EP[1]-Pls[1])\*Su[1]/AWC

Pl[1]=Pls[1]+Plu[1]

for (i in 2:N){

 if (PEP[i]<0){NegAcum[i]=NegAcum[i-1]+PEP[i];Arm[i]=AWC\*exp(NegAcum[i]/AWC)}

 if (PEP[i]>=0){Arm[i]=Arm[i-1]+PEP[i];if(Arm[i]>AWC){Arm[i]=AWC};NegAcum[i]=AWC\*log(Arm[i]/AWC)}

 Alt[i]=Arm[i]-Arm[i-1]

 if (PEP[i]<0){ETR[i]=pre[i]+abs(Alt[i])}else{ETR[i]=EP[i]}

 Def[i]=EP[i]-ETR[i]

 if (Arm[i]<AWC){Exd[i]=0}else{Exd[i]=PEP[i]-Alt[i]}

 ####################Palmer Coefficientes ####################

 Ss[i]=0.25\*Arm[i-1]

 Su[i]=0.75\*Arm[i-1]

 S[i]=Ss[i]+Su[i]

 if (Ss[i]<abs(PEP[i])){Ls[i]=Ss[i]}else{Ls[i]=abs(PEP[i])}

 Lu[i]=(EP[i]-pre[i]-Ls[i])\*Su[i]/AWC

 L[i]=Ls[i]+Lu[i]

 if (Alt[i]>0){R[i]=Alt[i]}else{R[i]=0}

 RO[i]=Exd[i]

 ####################Palmer Coefficientes: Potential Values ####################

 PR[i]=AWC-Arm[i-1]

 PRO[i]=S[i]

 if (EP[i]<Ss[i]){Pls[i]=EP[i]}else{Pls[i]=Ss[i]}

 Plu[i]=(EP[i]-Pls[i])\*Su[i]/AWC

 Pl[i]=Pls[i]+Plu[i]

}

dates1 = seq(from=First.Year+1/(period+1), by = 1/period,length.out=N)

meses=as.matrix(dates1-floor(dates1))

BH=format(cbind(pre,temp,EP,PEP,ETR,Arm,Exd,Def,Alt),digits=2,scientific = NA)

########d factor#############

BHC=as.matrix(cbind(dates1,meses,pre,temp,EP,PEP,ETR,Arm,Exd,Def,Alt,R,RO,L,PR,PRO,Plu,Pl))

BHC=BHC[order(BHC[,2]),]

means=aggregate(BHC, by=list(BHC[,2]), FUN='mean')

alfa=matrix(NA,period,1)

beta=matrix(NA,period,1)

gamma=matrix(NA,period,1)

delta=matrix(NA,period,1)

denominator.monthly=matrix(NA,period,1)

for (i in 1:period) {

 alfa[i]=means[i,8]/means[i,6]

 beta[i]=means[i,13]/means[i,16]

 gamma[i]=means[i,14]/means[i,17]

 delta[i]=means[i,15]/means[i,18]}

##############CAFEC Precipitation and d factor#################

CAFEC.EP[1]=alfa[1]\*EP[1]

CAFEC.R[1]=beta[1]\*PR[1]

CAFEC.RO[1]=gamma[1]\*PRO[1]

CAFEC.L[1]=delta[1]\*Pl[1]

j=1

limit=N-1

for (i in 2:limit){

 if (BHC[i,2]==BHC[i-1,2]){

 j=j

 CAFEC.EP[i]=alfa[j]\*EP[i]

 CAFEC.R[i]=beta[j]\*PR[i]

 CAFEC.RO[i]=gamma[j]\*PRO[i]

 CAFEC.L[i]=delta[j]\*Pl[i];i=i+1}else{

 j=j+1

 CAFEC.EP[i]=alfa[j]\*EP[i]

 CAFEC.R[i]=beta[j]\*PR[i]

 CAFEC.RO[i]=gamma[j]\*PRO[i]

 CAFEC.L[i]=delta[j]\*Pl[i]

 }}

CAFEC.EP[i]=alfa[j]\*EP[i]

CAFEC.R[i]=beta[j]\*PR[i]

CAFEC.RO[i]=gamma[j]\*PRO[i]

CAFEC.L[i]=delta[j]\*Pl[i]

CAFEC.P=CAFEC.EP+CAFEC.R+CAFEC.RO-CAFEC.L

d=(BHC[,3]-CAFEC.P)

d.abs=abs(d)

d1=as.matrix(cbind(meses,d.abs))

means.d=aggregate(d1, by=list(d1[,1]), FUN='mean',na.rm=TRUE)

Events=as.matrix(cbind(BHC[,1],d))

Events=Events[order(Events[,1]),]

Prec.Exd.Def=as.data.frame(cbind(dataset[,1],dataset[,2],Events[,2]))

colnames(Prec.Exd.Def)=c("Year","Period","Standar.PalmerAWC=150")

# The Multi-Scalar Standardized Palmer Index #####################

beginning=N-period

multi.scalar.d=matrix(NA,beginning,timescale+2)

for (TS in 1:timescale){

 col=TS+2

 TS.d=as.matrix(rollapply(Prec.Exd.Def[,3],TS,sum))

 begin=period+1

 aj1=begin-(TS-1)

 aj2=N-(TS-1)

 multi.scalar.d[,col]=TS.d[aj1:aj2,]}

multi.scalar.d[,1]=dataset[begin:N,1]

multi.scalar.d[,2]=dataset[begin:N,2]

StdPDSIf=list()

col=2

for (coluna in 1:timescale){

 a=0

 ci=0

 col=col+1

 t=as.matrix(multi.scalar.d[order(multi.scalar.d[,2]),decreasing=T])

 NN=(length(t))/(timescale+2)

 jj=2

 inicio=1

 fim=1

 for (tempo in 1:period) {

 while (t[jj-1,2]==t[jj,2]){if(jj<NN){fim=fim+1;jj=jj+1}else{fim=fim+1;break}}

 x=matrix(NA,fim,1)

 x=t[inicio:fim,col]

 inicio=fim+1

 fim=fim+1

 jj=jj+1

 N1=length(x)

 y=matrix(NA, N1, 1)

 y[,1]=(sort(x))

 probacum= matrix(NA, N1, 1)

 z=matrix(1: N1,nrow=N1,ncol=1)

 lmr = lmom.ub(y);parm=parglo(lmr)

 if (tempo==1){StdPDSI=matrix(NA,N1,period)}

 parmwei=parm[1]

 y1=as.vector(y)

 lmr = lmom.ub(y1)

 parm1=parglo(lmr)

 parmglo=parm1[2]

 dfprov=unique(as.numeric(unlist(parmglo)))

 loc=dfprov[1]

 sc=dfprov[2]

 sh=dfprov[3]

 parm=fitdist(y1, "glo", start=list(loc1=loc, sc1=sc, sh1=sh),optim.method = "Nelder-Mead")

 probacum1=as.matrix(pglo(x,parm$estimate[1],parm$estimate[2],parm$estimate[3]))

 for (isi in 1:length(x)){

 if (probacum1[isi,1]<=0.5){sinal=sqrt(log(1/probacum1[isi,1]^2))

 StdPDSI[isi,tempo]=-1\*(sinal-(2.515517+0.802853\*sinal+0.010328\*sinal^2)/(1+1.432788\*sinal+0.189269\*sinal^2+0.001308\*sinal^3))}

 else{sinal=sqrt(log(1/(1-probacum1[isi,1])^2))

 StdPDSI[isi,tempo]=sinal-(2.515517+0.802853\*sinal+0.010328\*sinal^2)/(1+1.432788\*sinal+0.189269\*sinal^2+0.001308\*sinal^3)}}

 }

 StdPDSIf[[coluna]]=StdPDSI

}

StdPDSIf1=as.data.frame(StdPDSIf)

year=as.vector(seq(from=(First.Year+1), by = 1,length.out=N1))

SPDI=cbind(year,StdPDSIf1)

write.csv(SPDI, "SPDI.csv")

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

#myfile.txt exemple

Year Month prec tmed

1960 1 277.7 23.6

1960 2 224.1 23.6

1960 3 83.2 23.05

1960 4 81.1 21.4

1960 5 98 18.5

1960 6 51.7 18.1

1960 7 0.8 16.9

1960 8 8 18.95

1960 9 5 19.35

1960 10 91.7 22.3

1960 11 230 22.1

1960 12 267.8 23.1

[....................]

2015 1 203.7 26.6

2015 2 266.5 24.7

2015 3 279.2 23.5

2015 4 29.2 22.9

2015 5 82.9 20.1

2015 6 16.2 19.2

2015 7 31.7 18.9

2015 8 6.7 21.4

2015 9 127 23.9

2015 10 107.7 26

2015 11 238.4 24.8

2015 12 194.4 25.4