***DISTRIBUTION FUNCTION***

function(eta,lognormal){

#cauchy only is always on deltas therefore deltaeta =eta input / log normal : deltaeta=diff(eta)

#lognormal=true when you analyze individual viscosity profiles

deltaeta <- eta

meanlog <- 0

if(lognormal=="true"){ #viscosity data are following a lognormal distribution

# lognormal\_check

lognorm <- fitdistr(eta, "log-normal") #here we assume eta ~ log-normal

meanlog <- coef(lognorm)[1]

sdlog <- coef(lognorm)[2]

median <- meanlog - sdlog\*sdlog/2

mode <- meanlog - (3\*sdlog)^2/2

etaecdf <- exp(meanlog)

etaecdfhb <- exp(meanlog + sdlog)

etaecdflb <- exp(meanlog - sdlog)

#Plots

ecdf\_plot <- ecdf(eta)

min <- min(eta)

max <- max(eta)

plot(ecdf\_plot, log = "x",

xlim = c(min, max),

xlab = "Transmission",

cex.lab = 1.5)

ecdf\_rl <- rlnorm(1000, meanlog, sdlog)

plot(ecdf\_rl, cex.lab = 1.5,

col = "red",

add = TRUE,

do.points = FALSE)

text <- paste("eta lognormal", round(etaecdf, 1), "hb", round(etaecdfhb, 1), "lb", round(etaecdflb, 1))

mtext(cex = 0.75, text, 3, 0.5)

deltaeta <- diff(eta) #compute the difference between two observations of eta

}

#cauchy\_check via fitdistr

cauchy <- fitdistr(deltaeta,"cauchy") #here we assume eta ~ Cauchy

meancauchy <- coef(cauchy)[1]

scalecauchy <- coef(cauchy)[2]

mean <- round(meancauchy,2)

fwhm <- 2\*(round(scalecauchy,2))

# gauss\_check via fitdistr

normal <- fitdistr(deltaeta,"normal") #here we assume eta ~ normal

meannormal <- coef(normal)[1]

sdnormal <- coef(normal)[2]

#subtract mean (obtained from gaussian) from deltaeta, also for the master curve

delta\_eta <- deltaeta - meannormal

delta\_eta <- sort(deltaeta, na.last = NA) #Normalization and we remove NA

mykurtosis = function(x) { #Returns the flattening coefficient of the curve

mean\_x <- mean(x, na.rm=TRUE)

sd\_x <- sd(x, na.rm = TRUE)

(mean((x-mean\_x)^4)/(sd\_x^4))-3

}

myskewness = function(x) { #Returns the asymmetry coefficient of the curve

mean\_x <- mean(x, na.rm=TRUE)

sd\_x <- sd(x, na.rm = TRUE)

mean((x-mean\_x)^3)/(sd\_x^3)

}

kurtosis <- mykurtosis(delta\_eta)

skewness <- myskewness(delta\_eta)

# t-Student

lower <- c(-1, 0.02, 2)

upper <- c(1, 40, 50)

starting\_list <- list(m = mean(delta\_eta), s = sd(delta\_eta), df = 3)

Param <- fitdistr(delta\_eta, "t",

start = starting\_list,

lower = lower,

upper = upper)

#Parameters according to the law of t-Student

tm <- coef(Param)[1]

ts <- coef(Param)[2]

tdf<- coef(Param)[3]

deltaeta\_neg <- sort(abs(delta\_eta[delta\_eta<0]))

deltaeta\_pos <- sort(delta\_eta[delta\_eta>=0])

count\_neg <- length(deltaeta\_neg)

count\_pos <- length(deltaeta\_pos)

count <- count\_neg + count\_pos

xneg <- (count\_neg:1)/count

xpos <- (count\_pos:1)/count

ylim <- c(1/(2\*count), 1)

xlim\_max <- max(deltaeta\_neg, deltaeta\_pos)

xlim\_min <- min(deltaeta\_neg, deltaeta\_pos)

xlim <- c(xlim\_min, xlim\_max)

if(lognormal=="false") { #for scaling law

mean <- round(meannormal, 2)

sd <- round(sdnormal, 2)

ylim <- c(1/(2\*count), 1)

if(max(deltaeta\_pos) > 10) { #criterion for deciding on type of x axis (normalized to sigma or not)

xlab = "delta T"

}

else {

xlab = expression(paste("|", Delta, "", eta, "| / ", sigma, "(", Delta,"", eta, ")"))

xlab = "delta T/sigma(T)"

}

plot(deltaeta\_neg, xneg, cex.lab = 1.5,

pch = 20, col = "red", log = "y",

ylim = ylim, xlim = xlim,

xlab = xlab, ylab = "CDF")

}

else { #for each film

if(film=="Lehigh\_data") {

plot(deltaeta\_neg, xneg\_exp, cex.lab = 1.5, cex = 0.75,

log = "y", ylab = "CDF", xlim = xlim, ylim = ylim,

pch = 20, col = "red", xlab = expression(paste("|", Delta, "", G, "'| (Pa.s)")))

}

else {

plot(deltaeta\_neg, xneg\_exp, cex.lab = 1.5, cex = 0.75,

log = "y", ylab = "CDF", xlim = xlim, ylim = ylim,

pch = 20, col = "red",

xlab = expression(paste("|", Delta, "", eta, "| (Pa.s)")))

}

}

points(deltaeta\_pos, xpos, pch = 20)

#when Scaling function is called set "false"

#scaling is now done to sd and not the mean of the viscosity as earlier (however these are proportional to eachother)

#if statement has to be changed film="filmx" first in the series of films

Format = function(number) { #to make stardardized output

sprintf("%3.6f", number)

}

if(lognormal=="true") {

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

delta\_eta\_scaling <- delta\_eta/sd(delta\_eta)

dfscalingdeltaeta <- data.frame(delta\_eta\_scaling)

#for every frequency/film data are written to a seperate file when "lognormal=true"

WriteXLS(dfscalingdeltaeta, paste(film, "Scalingdeltaeta.xlsx", sep=""), SheetNames = "scalinglaw")

tdistribution\_fits <- data.frame(film,Format(tm),Format(ts),Format(tdf),Format(skewness),Format(kurtosis))

print(film)

if(film=="film10"){write.table(tdistribution\_fits,paste("tdistribution\_fits.txt",sep=""),col.names=TRUE, append=TRUE)

}else{

write.table(tdistribution\_fits,paste("tdistribution\_fits.txt",sep="",col.names=FALSE,append=TRUE))}

}

# plot with distributions

# cauchy

lorentz=NULL

Cauchy=function(x,meancauchy,scalecauchy){1/2-atan((x-meancauchy)/scalecauchy)/pi}

# delta\_eta with mean = 0

lines(deltaeta\_neg,Cauchy(deltaeta\_neg,0,scalecauchy),lwd=2,col="green")

#if(lognormal=="false"){

y=(1:count)/count

# gaussian

deltaeta\_gauss=qnorm(y,0,sd(delta\_eta))

deltaeta\_gauss\_neg=abs(deltaeta\_gauss[deltaeta\_gauss<0])

negcdf=y[deltaeta\_gauss<0]

deltaeta\_gauss\_pos=deltaeta\_gauss[deltaeta\_gauss>=0]

poscdf=1-y[deltaeta\_gauss>=0]

count\_gauss\_neg =length(deltaeta\_gauss\_neg)

count\_gauss\_pos =length(deltaeta\_gauss\_pos)

lines(deltaeta\_gauss\_neg,negcdf,lwd=2,col='blue')

# t student - need to develop code as qt is with scale (sigma)=1 only

# - fitdistr uses more general t student: with mean (mu), scale (sigma) and

# degrees of freedaom (nu) as output parameters)

t\_general=function(x,mu,sigma,nu){

1/sigma\*

gamma((nu+1)/2)/

sqrt(nu\*pi)/gamma(nu/2)\*

(1+((x-mu)^2)/nu/sigma/sigma)^(-(1+nu)/2)

}

min\_eta=min(delta\_eta)

max\_eta=max(delta\_eta)

if(abs(min\_eta)>max\_eta){

max\_eta=abs(min\_eta)

}else{

min\_eta=-max\_eta

}

DELTA=max\_eta-min\_eta

deltaeta\_tstudent=NULL

cfdt=NULL

tdist=NULL

cfdt[1]=t\_general(min\_eta,tm,ts,tdf)

tdist[1]=cfdt[1]

deltaeta\_tstudent[1]=min\_eta

for(i in 2:count){

deltaeta\_tstudent[i]=min\_eta+(i-1)\*DELTA/count

if(abs(deltaeta\_tstudent[i])<0.0001){deltaeta\_tstudent[i]=0.0001}

tdist[i]=t\_general(deltaeta\_tstudent[i],tm,ts,tdf)

cfdt[i]=cfdt[i-1]+t\_general(deltaeta\_tstudent[i],tm,ts,tdf)

}

cfdt=cfdt/cfdt[count]

deltaeta\_tstudent\_neg=abs(deltaeta\_tstudent[deltaeta\_tstudent<0])

negcdft=cfdt[deltaeta\_tstudent<0]

deltaeta\_tstudent\_pos=deltaeta\_tstudent[deltaeta\_tstudent>=0]

poscdft=1-cfdt[deltaeta\_tstudent>=0]

count\_tstudent\_neg =length(deltaeta\_tstudent\_neg)

count\_tstudent\_pos =length(deltaeta\_tstudent\_pos)

#points(log="xy",xneg,yneg,pch=17,cex=1.5,col="red")

#points(xpos,ypos,pch=20)

lines(deltaeta\_tstudent\_neg,negcdft,lwd=2,lty=2,col='red')

lines(deltaeta\_tstudent\_pos,poscdft,lwd=2)

mtext(cex=0.75,paste("df = ",round(tdf,2)," scale = ",round(ts,2)),3,2)

mtext(cex=0.75,paste("Skewn = ",round(skewness,2)," Kurt = ",round(kurtosis,2)),3,1)

#

#

if(film=="film9" | film=="film15" | film=="film10" | film=="film21"){#series 1

#if(film=="film12" | film=="film13" | film=="film17" | film=="film10"){#series 2

delta\_eta\_all =c(deltaeta\_neg,deltaeta\_pos,deltaeta\_gauss\_neg,deltaeta\_gauss\_pos,deltaeta\_tstudent\_neg,deltaeta\_tstudent\_pos)

ecdf\_exp\_neg =c(xneg\_exp,rep(NA,(count\_pos+count+count)))

ecdf\_exp\_pos =c(rep(NA,(count\_neg)),xpos\_exp,rep(NA,count+count))

ecdf\_cauchy\_neg =c(Cauchy(deltaeta\_neg,0,scalecauchy),rep(NA,(count\_pos+count+count)))

ecdf\_cauchy\_pos =c(rep(NA,count\_neg),Cauchy(deltaeta\_pos,0,scalecauchy), rep(NA,count+count))

ecdf\_gauss\_neg =c(rep(NA,count),negcdf,rep(NA,count\_gauss\_pos+count))

ecdf\_gauss\_pos =c(rep(NA,count+count\_gauss\_neg),poscdf,rep(NA,count))

ecdf\_tstudent\_neg =c(rep(NA,count+count),negcdft,rep(NA,count\_tstudent\_pos))

ecdf\_tstudent\_pos =c(rep(NA,count+count+count\_tstudent\_neg),poscdft)

# plot for publication non-scaled

plot\_cdf\_delta\_eta=data.frame(Format(delta\_eta\_all),

Format(ecdf\_exp\_neg),Format(ecdf\_exp\_pos),

Format(ecdf\_cauchy\_neg),Format(ecdf\_cauchy\_pos),

Format(ecdf\_gauss\_neg),Format(ecdf\_gauss\_pos),

Format(ecdf\_tstudent\_neg),Format(ecdf\_tstudent\_pos))

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/Dataprep/")

write.table(plot\_cdf\_delta\_eta,quote=FALSE,file=paste(film,"plot\_cdf\_delta\_eta.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

# plots for publication - plots for order of magnitude difference

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results")

#series 1: 9, 21; series 2: 10, 17

if(film=="film9"){

write.table(plot\_cdf\_delta\_eta,quote=FALSE,file=paste(film,"plot\_cdf\_delta\_eta.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

if(film=="film21"){

write.table(plot\_cdf\_delta\_eta,quote=FALSE,file=paste(film,"plot\_cdf\_delta\_eta.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

# plot for publication scaled data for each frequency with the respective distributions for films 9,10,15,21

delta\_eta\_scaled = delta\_eta\_all/sd(delta\_eta)

plot\_cdf\_delta\_eta\_scaling = data.frame(Format(delta\_eta\_scaled),

Format(ecdf\_exp\_neg),Format(ecdf\_exp\_pos),

Format(ecdf\_cauchy\_neg),Format(ecdf\_cauchy\_pos),

Format(ecdf\_gauss\_neg),Format(ecdf\_gauss\_pos),

Format(ecdf\_tstudent\_neg),Format(ecdf\_tstudent\_pos))

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

write.table(plot\_cdf\_delta\_eta\_scaling,quote=FALSE,file=paste(film,"plot\_correct\_scaling.txt",sep=""),na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

if(lognormal=="false"){

# plot for publication - old master curve for cdf -

# data have been scaled on individual basis - see above - therefore deltaeta\_neg/pos are scaled

# arrays of x (xneg and pos) and y (deltaeta\_pos and neg) values

delta\_eta\_vs\_eta=c(deltaeta\_neg,deltaeta\_pos,deltaeta\_gauss\_neg,deltaeta\_gauss\_pos,deltaeta\_tstudent\_neg,deltaeta\_tstudent\_pos)

ecdf\_exp\_neg =c(xneg\_exp,rep(NA,(count\_pos+count+count)))

ecdf\_exp\_pos =c(rep(NA,(count\_neg)),xpos\_exp,rep(NA,count+count))

ecdf\_cauchy\_neg =c(Cauchy(deltaeta\_neg,0,scalecauchy),rep(NA,(count\_pos+count+count)))

ecdf\_cauchy\_pos =c(rep(NA,count\_neg),Cauchy(deltaeta\_pos,0,scalecauchy), rep(NA,count+count))

ecdf\_gauss\_neg =c(rep(NA,count),negcdf,rep(NA,count\_gauss\_pos+count))

ecdf\_gauss\_pos =c(rep(NA,count+count\_gauss\_neg),poscdf,rep(NA,count))

ecdf\_tstudent\_neg =c(rep(NA,count+count),negcdft,rep(NA,count\_tstudent\_pos))

ecdf\_tstudent\_pos =c(rep(NA,count+count+count\_tstudent\_neg),poscdft)

# dataframe for plotting sprintf("%.6s",t)

#Format=function(number){sprintf("%.6s",number)}

plot\_mastercurve\_delta\_eta=data.frame(Format(delta\_eta\_vs\_eta),

Format(ecdf\_exp\_neg),Format(ecdf\_exp\_pos),

Format(ecdf\_cauchy\_neg),Format(ecdf\_cauchy\_pos),

Format(ecdf\_gauss\_neg),Format(ecdf\_gauss\_pos),

Format(ecdf\_tstudent\_neg),Format(ecdf\_tstudent\_pos))

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results/")

write.table(plot\_mastercurve\_delta\_eta,quote=FALSE,file="plot\_mastercurve\_delta\_eta\_scaled.txt",na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

}

***Final version***

function(eta,lognormal){

deltaeta <- eta

meanlog <- 0

if(lognormal=="true") {

lognorm <- fitdistr(eta, "log-normal")

meanlog <- coef(lognorm)[1]

sdlog <- coef(lognorm)[2]

median <- meanlog - sdlog\*sdlog/2

mode <- meanlog - (3\*sdlog)^2/2

etaecdf <- exp(meanlog)

etaecdfhb <- exp(meanlog + sdlog)

etaecdflb <- exp(meanlog - sdlog)

ecdf\_plot <- ecdf(eta)

min <- min(eta)

max <- max(eta)

plot(ecdf\_plot, log = "x",

xlim = c(min, max),

xlab = "Transmission",

cex.lab = 1.5)

ecdf\_rl <- rlnorm(1000, meanlog, sdlog)

plot(ecdf\_rl, cex.lab = 1.5,

col = "red",

add = TRUE,

do.points = FALSE)

text <- paste("eta lognormal", round(etaecdf, 1), "hb", round(etaecdfhb, 1), "lb", round(etaecdflb, 1))

mtext(cex = 0.75, text, 3, 0.5)

deltaeta <- diff(eta)

}

cauchy <- fitdistr(deltaeta,"cauchy")

meancauchy <- coef(cauchy)[1]

scalecauchy <- coef(cauchy)[2]

mean <- round(meancauchy,2)

fwhm <- 2\*(round(scalecauchy,2))

normal <- fitdistr(deltaeta,"normal")

meannormal <- coef(normal)[1]

sdnormal <- coef(normal)[2]

delta\_eta <- deltaeta - meannormal

delta\_eta <- sort(deltaeta, na.last = NA)

mykurtosis = function(x) {

mean\_x <- mean(x, na.rm=TRUE)

sd\_x <- sd(x, na.rm = TRUE)

(mean((x-mean\_x)^4)/(sd\_x^4))-3

}

myskewness = function(x) {

mean\_x <- mean(x, na.rm=TRUE)

sd\_x <- sd(x, na.rm = TRUE)

mean((x-mean\_x)^3)/(sd\_x^3)

}

kurtosis <- mykurtosis(delta\_eta)

skewness <- myskewness(delta\_eta)

lower <- c(-1, 0.02, 2)

upper <- c(1, 40, 50)

starting\_list <- list(m = mean(delta\_eta), s = sd(delta\_eta), df = 3)

Param <- fitdistr(delta\_eta, "t",

start = starting\_list,

lower = lower,

upper = upper)

tm <- coef(Param)[1]

ts <- coef(Param)[2]

tdf<- coef(Param)[3]

deltaeta\_neg <- sort(abs(delta\_eta[delta\_eta<0]))

deltaeta\_pos <- sort(delta\_eta[delta\_eta>=0])

count\_neg <- length(deltaeta\_neg)

count\_pos <- length(deltaeta\_pos)

count <- count\_neg + count\_pos

xneg\_exp <- (count\_neg:1)/count

xpos\_exp <- (count\_pos:1)/count

ylim <- c(1/(2\*count), 1)

xlim\_max <- max(deltaeta\_neg, deltaeta\_pos)

xlim\_min <- min(deltaeta\_neg, deltaeta\_pos)

xlim <- c(xlim\_min, xlim\_max)

if(lognormal=="false") {

mean <- round(meannormal, 2)

sd <- round(sdnormal, 2)

ylim <- c(1/(2\*count), 1)

if(max(deltaeta\_pos) > 10) {

xlab = "delta T"

}

else {

xlab = expression(paste("|", Delta, "", eta, "| / ", sigma, "(", Delta,"", eta, ")"))

xlab = "delta T/sigma(T)"

}

plot(deltaeta\_neg, xneg\_exp, cex.lab = 1.5,

pch = 20, col = "red", log = "y",

ylim = ylim, xlim = xlim,

xlab = xlab, ylab = "CDF")

}

if(film=="Lehigh\_data") {

plot(deltaeta\_neg, xneg\_exp, cex.lab = 1.5, cex = 0.75,

log = "y", ylab = "CDF", xlim = xlim, ylim = ylim,

pch = 20, col = "red", xlab = expression(paste("|", Delta, "", G, "'| (Pa.s)")))

}

else {

plot(deltaeta\_neg, xneg\_exp, cex.lab = 1.5, cex = 0.75,

log = "y", ylab = "CDF", xlim = xlim, ylim = ylim,

pch = 20, col = "red",

xlab = expression(paste("|", Delta, "", eta, "| (Pa.s)")))

}

points(deltaeta\_pos, xpos\_exp, pch = 20)

Format = function(number) {

sprintf("%3.6f", number)

}

if(lognormal=="true") {

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

delta\_eta\_scaling <- delta\_eta/sd(delta\_eta)

dfscalingdeltaeta <- data.frame(delta\_eta\_scaling)

WriteXLS(dfscalingdeltaeta, paste(film, "Scalingdeltaeta.xlsx", sep=""), SheetNames = "scalinglaw")

tdistribution\_fits <- data.frame(film, Format(tm),

Format(ts), Format(tdf),

Format(skewness), Format(kurtosis))

print(film)

if(film=="film10") {

write.table(tdistribution\_fits, paste("tdistribution\_fits.txt", sep= ""), col.names = TRUE, append = TRUE)

}

else {

write.table(tdistribution\_fits, paste("tdistribution\_fits.txt", sep = "", col.names = FALSE,append = TRUE))}

}

lorentz <- NULL

Cauchy = function(x, meancauchy, scalecauchy) {

1/2-atan((x-meancauchy)/scalecauchy)/pi

}

lines(deltaeta\_neg, Cauchy(deltaeta\_neg, 0, scalecauchy), lwd = 2, col = "green")

y <- (1:count)/count

sd\_deltaeta <- sd(delta\_eta)

deltaeta\_gauss <- qnorm(y, 0, sd\_deltaeta)

deltaeta\_gauss\_neg <- abs(deltaeta\_gauss[deltaeta\_gauss < 0])

negcdf <- y[deltaeta\_gauss < 0]

deltaeta\_gauss\_pos <- deltaeta\_gauss[deltaeta\_gauss>=0]

poscdf <- 1 - y[deltaeta\_gauss >= 0]

count\_gauss\_neg <- length(deltaeta\_gauss\_neg)

count\_gauss\_pos <- length(deltaeta\_gauss\_pos)

lines(deltaeta\_gauss\_neg, negcdf, lwd = 2, col = 'blue')

t\_general = function(x, mu, sigma, nu) {

1/sigma\*gamma((nu+1)/2)/sqrt(nu\*pi)/

gamma(nu/2)\*(1+((x-mu)^2)/nu/sigma/sigma)^(-(1+nu)/2)

}

min\_eta <- min(delta\_eta)

max\_eta <- max(delta\_eta)

if(abs(min\_eta) > max\_eta) {

max\_eta <- abs(min\_eta)

}

else {

min\_eta <- -max\_eta

}

DELTA <- max\_eta-min\_eta

deltaeta\_tstudent <- NULL

cfdt <- NULL

tdist <- NULL

cfdt[1] <- t\_general(min\_eta, tm, ts, tdf)

tdist[1] <- cfdt[1]

deltaeta\_tstudent[1] <- min\_eta

for(i in 2:count) {

deltaeta\_tstudent[i] <- min\_eta + (i-1) \* DELTA/count

if(abs(deltaeta\_tstudent[i]) < 0.0001) {

deltaeta\_tstudent[i] <- 0.0001

}

tdist[i] <- t\_general(deltaeta\_tstudent[i], tm, ts, tdf)

cfdt[i] <- cfdt[i-1] + t\_general(deltaeta\_tstudent[i], tm, ts, tdf)

}

cfdt <- cfdt/cfdt[count]

deltaeta\_tstudent\_neg <- abs(deltaeta\_tstudent[deltaeta\_tstudent < 0])

negcdft <- cfdt[deltaeta\_tstudent < 0]

deltaeta\_tstudent\_pos <- deltaeta\_tstudent[deltaeta\_tstudent >= 0]

poscdft <- 1 - cfdt[deltaeta\_tstudent >= 0]

count\_tstudent\_neg <- length(deltaeta\_tstudent\_neg)

count\_tstudent\_pos <- length(deltaeta\_tstudent\_pos)

lines(deltaeta\_tstudent\_neg, negcdft, lwd = 2, lty = 2, col = 'red')

lines(deltaeta\_tstudent\_pos, poscdft, lwd = 2)

mtext(cex = 0.75, paste("df = ", round(tdf, 2), " scale = ", round(ts, 2)), 3, 2)

mtext(cex = 0.75, paste("Skewn = ", round(skewness, 2), " Kurt = ", round(kurtosis, 2)), 3, 1)

if(film=="film9" | film=="film15" | film=="film10" | film=="film21") {

delta\_eta\_all <- c(deltaeta\_neg, deltaeta\_pos, deltaeta\_gauss\_neg,

deltaeta\_gauss\_pos, deltaeta\_tstudent\_neg, deltaeta\_tstudent\_pos)

ecdf\_exp\_neg <- c(xneg\_exp, rep(NA,(count\_pos + 2\*count)))

ecdf\_exp\_pos <- c(rep(NA, (count\_neg)), xpos\_exp, rep(NA, 2\*count))

ecdf\_cauchy\_neg <- c(Cauchy(deltaeta\_neg, 0, scalecauchy), rep(NA,(count\_pos + 2\*count)))

ecdf\_cauchy\_pos <- c(rep(NA,count\_neg), Cauchy(deltaeta\_pos, 0, scalecauchy), rep(NA, 2\*count))

ecdf\_gauss\_neg <- c(rep(NA, count), negcdf, rep(NA, count\_gauss\_pos + count))

ecdf\_gauss\_pos <- c(rep(NA, count + count\_gauss\_neg), poscdf, rep(NA, count))

ecdf\_tstudent\_neg <- c(rep(NA, 2\*count), negcdft, rep(NA, count\_tstudent\_pos))

ecdf\_tstudent\_pos <- c(rep(NA, 2\*count + count\_tstudent\_neg), poscdft)

df\_cdf\_delta\_eta <- data.frame(Format(delta\_eta\_all),

Format(ecdf\_exp\_neg),

Format(ecdf\_exp\_pos),

Format(ecdf\_cauchy\_neg),

Format(ecdf\_cauchy\_pos),

Format(ecdf\_gauss\_neg),

Format(ecdf\_gauss\_pos),

Format(ecdf\_tstudent\_neg),

Format(ecdf\_tstudent\_pos))

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/DataAnalysisMethod/Dataprep/")

write.table(df\_cdf\_delta\_eta, file = paste(film, "plot\_cdf\_delta\_eta.txt", sep = ""),

quote = FALSE, na = "", sep = " ", col.names = TRUE, row.names = FALSE, dec = ".")

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results")

if(film=="film9") {

write.table(df\_cdf\_delta\_eta, file = paste(film, "plot\_cdf\_delta\_eta.txt", sep = ""),

quote = FALSE, na = "", sep = " ", col.names = TRUE, row.names = FALSE, dec = ".")

}

if(film=="film21"){

write.table(df\_cdf\_delta\_eta, file = paste(film, "plot\_cdf\_delta\_eta.txt", sep = ""),

quote = FALSE, na = "", sep = " ", col.names = TRUE, row.names = FALSE, dec = ".")

}

delta\_eta\_scaled <- delta\_eta\_all/sd(delta\_eta)

df\_cdf\_delta\_eta\_scaling <- data.frame(Format(delta\_eta\_scaled),

Format(ecdf\_exp\_neg), Format(ecdf\_exp\_pos),

Format(ecdf\_cauchy\_neg), Format(ecdf\_cauchy\_pos),

Format(ecdf\_gauss\_neg), Format(ecdf\_gauss\_pos),

Format(ecdf\_tstudent\_neg), Format(ecdf\_tstudent\_pos))

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

write.table(df\_cdf\_delta\_eta\_scaling, file = paste(film, "plot\_correct\_scaling.txt", sep = ""),

quote = FALSE, na = "", sep = " ", col.names = TRUE, row.names = FALSE, dec = ".")

}

if(lognormal=="false") {

delta\_eta\_vs\_eta <- c(deltaeta\_neg, deltaeta\_pos, deltaeta\_gauss\_neg,

deltaeta\_gauss\_pos, deltaeta\_tstudent\_neg, deltaeta\_tstudent\_pos)

ecdf\_exp\_neg <- c(xneg\_exp, rep(NA, (count\_pos + 2\*count)))

ecdf\_exp\_pos <- c(rep(NA, (count\_neg)), xpos\_exp, rep(NA, 2\*count))

ecdf\_cauchy\_neg <- c(Cauchy(deltaeta\_neg, 0, scalecauchy), rep(NA, (count\_pos + 2\*count)))

ecdf\_cauchy\_pos <- c(rep(NA, count\_neg), Cauchy(deltaeta\_pos, 0, scalecauchy), rep(NA, 2\*count))

ecdf\_gauss\_neg <- c(rep(NA, count), negcdf, rep(NA, count\_gauss\_pos + count))

ecdf\_gauss\_pos <- c(rep(NA, count + count\_gauss\_neg), poscdf, rep(NA, count))

ecdf\_tstudent\_neg <- c(rep(NA, 2\*count), negcdft, rep(NA, count\_tstudent\_pos))

ecdf\_tstudent\_pos <- c(rep(NA, 2\*count + count\_tstudent\_neg), poscdft)

df\_mastercurve\_delta\_eta <- data.frame(Format(delta\_eta\_vs\_eta),

Format(ecdf\_exp\_neg), Format(ecdf\_exp\_pos),

Format(ecdf\_cauchy\_neg), Format(ecdf\_cauchy\_pos),

Format(ecdf\_gauss\_neg), Format(ecdf\_gauss\_pos),

Format(ecdf\_tstudent\_neg), Format(ecdf\_tstudent\_pos))

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results/")

write.table(df\_mastercurve\_delta\_eta, file = "plot\_mastercurve\_delta\_eta\_scaled.txt",

quote = FALSE, na = "", sep = " ", col.names = TRUE, row.names = FALSE, dec = ".")

}

}

***SCALING DELTA AETA***

Scalingdelta\_aeta = function() {

# for Series/File selection put on TRUE

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

# All FALSE for series 3

if(FALSE){

# Film Series 1 for scaling of the selected data for publication

File=c("film10Scalingdeltaeta.xlsx",

"film15Scalingdeltaeta.xlsx",

"film21Scalingdeltaeta.xlsx" )

}

if(TRUE){

# Film Series 1

File=c("film1Scalingdeltaeta.xlsx",

"film8Scalingdeltaeta.xlsx",

"film9Scalingdeltaeta.xlsx",

"film10Scalingdeltaeta.xlsx",

"film11Scalingdeltaeta.xlsx",

"film12Scalingdeltaeta.xlsx",

"film13Scalingdeltaeta.xlsx",

"film14Scalingdeltaeta.xlsx",

"film15Scalingdeltaeta.xlsx",

"film17Scalingdeltaeta.xlsx",

"film18Scalingdeltaeta.xlsx",

"film19Scalingdeltaeta.xlsx",

"film20Scalingdeltaeta.xlsx",

"film21Scalingdeltaeta.xlsx" )

}

if(FALSE){

# Film Series Lehigh

File=c( "SheetScalingdeltaeta.xlsx")

}

if(FALSE){

# Film Series 2 for scaling

File=c( "film12Scalingdeltaeta.xlsx",

"film13Scalingdeltaeta.xlsx",

"film17Scalingdeltaeta.xlsx")

}

if(FALSE){

# Film Series 2

File=c( "film10Scalingdeltaeta.xlsx",

"film12Scalingdeltaeta.xlsx",

"film13Scalingdeltaeta.xlsx",

"film14Scalingdeltaeta.xlsx",

"film16Scalingdeltaeta.xlsx",

"film17Scalingdeltaeta.xlsx",

"film2Scalingdeltaeta.xlsx",

"film9Scalingdeltaeta.xlsx" )

}

dim=length(File)

combined\_deltaeta=NULL

for (i\_delta\_eta in 1:dim){

delta\_eta = read\_xlsx(File[i\_delta\_eta],"scalinglaw")

#check for end of file / determination of Dimension of data - Dimdata

for (i in 1:1000){

if (is.na(delta\_eta[i,1])){

Dimdata=i-1

break}

}

deltaeta=head(unlist((delta\_eta[,1])),Dimdata)

combined\_deltaeta=c(combined\_deltaeta,deltaeta)

#Distribution\_check(deltaeta,lognormal=FALSE)

}

#### scaling law all data irrespective of frequency - one does not see this in litarature

#### this part can be used to look at limited number files for the plot distribution data

#### (FALSE) leads to analysis of limited number of data

#### provided these have been set accordingly

df=data.frame(combined\_deltaeta)

if(FALSE){WriteXLS(df,"ScalingLawDeltaEta", SheetNames ="Alldata")}else{

WriteXLS(df,"ScalingLawDeltaEta", SheetNames ="Alldata")

}

lognormal="false"

count=length(combined\_deltaeta)

Distribution\_check(combined\_deltaeta,lognormal)

#read.table(paste(film,"Fit\_",FIT,"\_variogram.csv",sep=""),header=TRUE)

ylim=c(1/(2\*count),1)

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results")

####scaling law 1 order of magnitude apart: 0.089 Hz vs 0.847 Hz (Bouchaud et al.) for delta eta

if(TRUE){scaling\_nu8=read.table(paste("film9","plot\_cdf\_delta\_eta.txt",sep=""),header=TRUE)

scaling\_nu80=read.table(paste("film21","plot\_cdf\_delta\_eta.txt",sep=""),header=TRUE)

}

if(FALSE){scaling\_nu8=read.table(paste("film10","plot\_cdf\_delta\_eta.txt",sep=""),header=TRUE)

scaling\_nu80=read.table(paste("film17","plot\_cdf\_delta\_eta.txt",sep=""),header=TRUE)

}

deltaeta\_ecdfneg8=subset(scaling\_nu8[,c(1,2)],(!is.na(scaling\_nu8[,1]) & !is.na(scaling\_nu8[,2])))

deltaeta\_ecdfpos8=subset(scaling\_nu8[,c(1,3)],(!is.na(scaling\_nu8[,1]) & !is.na(scaling\_nu8[,3])))

deltaeta\_tdfneg8=subset(scaling\_nu8[,c(1,8)],(!is.na(scaling\_nu8[,1]) & !is.na(scaling\_nu8[,8])))

deltaeta\_tdfpos8=subset(scaling\_nu8[,c(1,9)],(!is.na(scaling\_nu8[,1]) & !is.na(scaling\_nu8[,9])))

#

deltaeta\_ecdfneg80=subset(scaling\_nu80[,c(1,2)],(!is.na(scaling\_nu80[,1]) & !is.na(scaling\_nu80[,2])))

deltaeta\_ecdfpos80=subset(scaling\_nu80[,c(1,3)],(!is.na(scaling\_nu80[,1]) & !is.na(scaling\_nu80[,3])))

deltaeta\_tdfneg80=subset(scaling\_nu80[,c(1,8)], (!is.na(scaling\_nu80[,1]) & !is.na(scaling\_nu80[,8])))

deltaeta\_tdfpos80=subset(scaling\_nu80[,c(1,9)],(!is.na(scaling\_nu80[,1]) & !is.na(scaling\_nu80[,9])))

#

ylim=c(1/(2\*count),1)

ylab="CDF"

xmax=max(deltaeta\_ecdfpos80,deltaeta\_ecdfneg80,deltaeta\_tdfpos80,

deltaeta\_ecdfpos8,deltaeta\_ecdfneg8,deltaeta\_tdfpos8)

logmax=trunc(log10(xmax))+1

logmax=10^logmax

linmax=(logmax/10)\*(trunc(xmax/(logmax/10))+1)

xlab=expression(paste("|",Delta,"",eta,"| (Pa.s)"))

plot(deltaeta\_ecdfneg80,cex.lab=1.5,col="red",log="xy",ylim=ylim,xlim=c(0.0001,logmax),

ylab=ylab,

xlab=xlab)

points(deltaeta\_ecdfpos80,pch=20,cex=0.5)

#lines(deltaeta\_tdfneg80,col="red")

lines(deltaeta\_tdfpos80,lty=3)

points(deltaeta\_ecdfneg8,pch=20,col="red")

points(deltaeta\_ecdfpos8)

#lines(deltaeta\_tdfneg8,col="red")

lines(deltaeta\_tdfpos8,lty=3)

legend("bottomleft",cex=1,

c(expression(paste("Negative, ",omega,"=5.3 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=0.55 ",rad.s^-1)),

expression(paste("Positive"))),

pch=c(1,20,20,1),

col=c("red","black","red","black"))

#

plot(deltaeta\_ecdfneg80,cex.lab=1.5,col="red",log="y",ylim=ylim,xlim=c(0,linmax),

ylab=ylab,

xlab=xlab)

points(deltaeta\_ecdfpos80,pch=20,cex=0.5)

#lines(deltaeta\_tdfneg80,col="red")

lines(deltaeta\_tdfpos80,lty=3)

points(deltaeta\_ecdfneg8,pch=20,col="red")

points(deltaeta\_ecdfpos8)

#lines(deltaeta\_tdfneg8,col="red")

lines(deltaeta\_tdfpos8,lty=3)

####end of scaling law 1 order of magnitude apart

####0###49#6######17################23#########28

####scaling law 0.089 Hz, 0.104 Hz, 0.207 Hz and 0.847 Hz Mantegna and Stanley (scaled to sd)

#in case of series 3 all FALSE

#series 1

if(TRUE){

films=c("film10","film15","film21")

nus=c(0.65,1.3,5.3)

}

#series 2

if(FALSE){

films=c("film12","film13","film17")

nus=c(0.76,0.94,5.7)

}

#correct\_scaling\_film=NULL

setwd("/Users/macbookair/Desktop/Series/OutputAll/GeneratedFiles/")

all\_scaled\_datax=NULL#input data of negative and positive values for fitdistr

all\_scaled\_datay=NULL

plot\_datax=NULL

#plotfile=TRUE for making one file else 4 files txt with a cdf version

plotfile=FALSE

if(plotfile){

plot\_index=list()

for (i in 1:(length(films)\*2)){plot\_index[i]=NULL}

#plot\_index=1>8 expcdfng\_film9 > \_film21

cdf\_=NULL

}

# to get the required data from the 4 film txt files and plotting (for publication)

# for series of frequencies/films make a plot

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

file=paste(films[i],"plot\_correct\_scaling.txt",sep="")

correct\_scaling=read.table(file=file,header=TRUE)

#going from 3 columns - x y1 (>0) y2 (<0/made abs) - to 2, x y for fitdistr function

#uneven nr are negative deltas

nr=2\*i-1

cdf=subset(correct\_scaling[,c(1,2)],(!is.na(correct\_scaling[,1]) & !is.na(correct\_scaling[,2])))

cdf\_neg=-cdf[,1]#all absolute negative values > negative for fitdistr for uneven nr

all\_scaled\_datax=c(all\_scaled\_datax,cdf\_neg)#for fitdistr

plot\_datax=c(plot\_datax,cdf[,1])

#plot for Rstudio

col="red"

log="y"

if(nr==1){plot(cdf,cex.lab=1.5,log=log,type="p",pch=1,col=col,cex=2/nr,

ylim=c(0.001,1),

xlim=c(0.0001,10),

ylab="CDF",

#xlab=expression(paste("|",Delta,"",eta,"(",omega,")| / ",sigma,"(",omega,")"))

xlab=expression(paste("|",Delta,"",eta,"| / ",sigma,"(",Delta,"",eta,")"))

)

}else{points(cdf,col=col,pch=1,cex=2/nr)}

#

if(plotfile){

plot\_index[[nr]]=c(plot\_index[[nr]],cdf[,2])

for (index in (nr+1):8){plot\_index[[index]]=c(plot\_index[[index]],rep(NA,length(cdf\_neg)))}

if(nr>1){for (index in 1:(nr-1)){plot\_index[[index]]=c(plot\_index[[index]],rep(NA,length(cdf\_neg)))}}

}

#

#even nr are positive deltas

nr=2\*i

#all positive values remain positive to calculate fitdistr

cdf=subset(correct\_scaling[,c(1,3)],(!is.na(correct\_scaling[,1]) & !is.na(correct\_scaling[,3])))

cdf\_=cdf[,1]

all\_scaled\_datax=c(all\_scaled\_datax,cdf\_)

plot\_datax=c(plot\_datax,cdf[,1])

#plot for Rstudio

col="black"

points(cdf,col=col,pch=1,cex=2/(nr-1))

if(plotfile){

plot\_index[[nr]]=c(plot\_index[[nr]],cdf[,3])

for (index in 1:(nr-1)){plot\_index[[index]]=c(plot\_index[[index]],rep(NA,length(cdf\_)))}

for (index in (nr+1):8){plot\_index[[index]]=c(plot\_index[[index]],rep(NA,length(cdf\_)))}

}

}

#print(all\_data)

#lognormal="false"

#Distribution\_check(all\_scaled\_datax,lognormal)

Param=fitdistr(all\_scaled\_datax,"t",start = list(m=mean(all\_scaled\_datax),s=sd(all\_scaled\_datax), df=3), lower=c(-0.05,0.5,1),upper=c(0.05,50,50))

tm =coef(Param)[1]

ts =coef(Param)[2]

tdf=coef(Param)[3]

t\_general=function(x,mu,sigma,nu){

1/sigma\*

gamma((nu+1)/2)/

sqrt(nu\*pi)/gamma(nu/2)\*

(1+((x-mu)^2)/nu/sigma/sigma)^(-(1+nu)/2)

}

#delta\_eta is the key variable and equal to all\_scaled\_datax

delta\_eta=all\_scaled\_datax

count=length(delta\_eta)

min\_eta=min(delta\_eta)

max\_eta=max(delta\_eta)

if(abs(min\_eta)>max\_eta){

max\_eta=abs(min\_eta)

}else{

min\_eta=-max\_eta

}

DELTA=max\_eta-min\_eta

deltaeta\_tstudent=NULL

cdft=NULL

tdist=NULL

#negcfdt=NULL

#poscfdt=NULL

cdft[1]=t\_general(min\_eta,tm,ts,tdf)

tdist[1]=cdft[1]

deltaeta\_tstudent[1]=min\_eta

for(i in 2:count){

deltaeta\_tstudent[i]=min\_eta+(i-1)\*DELTA/count

if(abs(deltaeta\_tstudent[i])<0.0001){

deltaeta\_tstudent[i]=0.0001}

tdist[i]=t\_general(deltaeta\_tstudent[i],tm,ts,tdf)

cdft[i]=cdft[i-1]+t\_general(deltaeta\_tstudent[i],tm,ts,tdf)

}

cdft=cdft/cdft[count]

deltaeta\_tstudent\_neg=abs(deltaeta\_tstudent[deltaeta\_tstudent<0])

negcdft=cdft[deltaeta\_tstudent<0]

deltaeta\_tstudent\_pos=deltaeta\_tstudent[deltaeta\_tstudent>=0]

poscdft=1-cdft[deltaeta\_tstudent>=0]

count\_tstudent\_neg =length(deltaeta\_tstudent\_neg)

count\_tstudent\_pos =length(deltaeta\_tstudent\_pos)

#points(log="xy",xneg,yneg,pch=17,cex=1.5,col="red")

#points(xpos,ypos,pch=20)

lines(deltaeta\_tstudent\_neg,negcdft,lwd=2,col='red')

lines(deltaeta\_tstudent\_pos,poscdft,lwd=2,lty=2)

mtext(cex=0.75,paste("mean= ",round(tm,2)," df= ",round(tdf,2)," scale= ",round(ts,2)),3,1)

if(log=="xy"){position="bottomleft"}else{position="topright"}

#in case of series 3 all FALSE

if(TRUE){#Series 1

legend(position,cex=1,

#change frequencies when necessary series 1 0.104, 0.207, 0.847

#nus=c(0.65,1.3,5.3)

c(expression(paste("Negative, ",omega,"=0.65 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=1.3 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=5.3 ",rad.s^-1)),

expression(paste("Positive"))),

pt.cex=c(1.5,1.5,1,1,0.5,0.5),

pch=c(1,1,1,1,1,1),col=c("red","black","red","black","red","black"))

}

if(FALSE){#Series 2

legend(position,cex=1,

#change frequencies when necessary series 1 0.104, 0.207, 0.847

#nus=c(0.76,0.94,5.7)

c(expression(paste("Negative, ",omega,"=0.76 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=0.94 ",rad.s^-1)),

expression(paste("Positive")),

expression(paste("Negative, ",omega,"=5.7 ",rad.s^-1)),

expression(paste("Positive"))),

pt.cex=c(1.5,1.5,1,1,0.5,0.5),

pch=c(1,1,1,1,1,1),col=c("red","black","red","black","red","black"))

}

Format=function(number){sprintf("%3.6f",number)}#to make stardardized Format output

plot\_cdfmasterstudent=data.frame(Format(c(deltaeta\_tstudent\_neg,deltaeta\_tstudent\_pos)),

Format(c(negcdft,rep(NA,count\_tstudent\_pos))),

Format(c(rep(NA,count\_tstudent\_neg),poscdft)))

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results/")

write.table(plot\_cdfmasterstudent,quote=FALSE,file="plot\_mastercurve\_Student.txt",na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

#make a dataframe

#series 1 films 9, 10, 15, 21

#series 2 films 12, 13, 17

if(plotfile){

all\_scaled\_datax=c(all\_scaled\_datax,deltaeta\_tstudent\_neg,deltaeta\_tstudent\_pos)

negcdft=c(rep(NA,length(all\_scaled\_datax)),negcdft,rep(NA,count\_tstudent\_pos))

poscfdt=c(rep(NA,length(all\_scaled\_datax)),rep(NA,count\_tstudent\_neg),poscdft)

#assigning the right plot\_index[[i]] (a list) to the right film

if(TRUE){

plot\_expcdfneg\_film9=plot\_index[[1]]

plot\_expcdfpos\_film9=plot\_index[[2]]

plot\_expcdfneg\_film10=plot\_index[[3]]

plot\_expcdfpos\_film10=plot\_index[[4]]

plot\_expcdfneg\_film15=plot\_index[[5]]

plot\_expcdfpos\_film15=plot\_index[[6]]

plot\_expcdfneg\_film21=plot\_index[[7]]

plot\_expcdfpos\_film21=plot\_index[[8]]

plot\_all\_data=data.frame(Format(plot\_datax),

Format(plot\_expcdfneg\_film9),

Format(plot\_expcdfpos\_film9),

Format(plot\_expcdfneg\_film10),

Format(plot\_expcdfpos\_film10),

Format(plot\_expcdfneg\_film15),

Format(plot\_expcdfpos\_film15),

Format(plot\_expcdfneg\_film21),

Format(plot\_expcdfpos\_film21),

Format(negcfdt),

Format(poscfdt))

}

if(FALSE){

plot\_expcdfneg\_film12=plot\_index[[1]]

plot\_expcdfpos\_film12=plot\_index[[2]]

plot\_expcdfneg\_film13=plot\_index[[3]]

plot\_expcdfpos\_film13=plot\_index[[4]]

plot\_expcdfneg\_film17=plot\_index[[5]]

plot\_expcdfpos\_film17=plot\_index[[6]]

plot\_all\_data=data.frame(Format(plot\_datax),

Format(plot\_expcdfneg\_film12),

Format(plot\_expcdfpos\_film12),

Format(plot\_expcdfneg\_film13),

Format(plot\_expcdfpos\_film13),

Format(plot\_expcdfneg\_film17),

Format(plot\_expcdfpos\_film17),

Format(negcfdt),

Format(poscfdt))

}

setwd("/Users/macbookair/Desktop/Series/Plots4Pub/Results/")

write.table(plot\_all\_data,quote=FALSE,file="plot\_mastercurve\_delta\_eta\_scaled.txt",na="",sep=" ",col.names=TRUE,row.names=FALSE,dec=".")

}

}# End of scaling deltas

#

```

***RAWDATA ANALYSIS***

Rawdata\_RDT=function(analyserawdata){

setwd("/Users/macbookair/Desktop/Series/InputData")

#Extensive Testing (time (etamin) etamin time (etamax) etamax)

#setwd("/Users/macbookair/Desktop/BerretMicroRheology/Berret/Data/Raw data txt files")

#if KO go to line 169

### AnalyseRawData must still be written

if(analyserawdata=="OK"){

dataset = read.delim("Recol\_film1 57mHz.txt", sep = "\t", nrows=500, header = TRUE, stringsAsFactors = FALSE)

dim(dataset)

#

# skip empty files / test the size of each file， and skip the file size of 0

#for (file in list.files(,"\*.txt")){

# if (file.size(file) == 0) next

# print(file)

#}

# In case header=true: first row (number 1 = first row of matrix ) is data

# check below with Pil=OK and row=1 / print individual lines Pil=OK and row=number

Pil="OK"

if (Pil=="OK")

{

#give row number

row=1800

if (is.na(dataset[row,3])){

print("stop due to NA")

print(row)}

firstline=c(dataset[row,1],dataset[row,2],dataset[row,3])

print (firstline)

}

for (i in 1000:3000){

if (is.na(dataset[i,3])){

print("stop due to NA")

print(i-1)

iend=i-1

print(dataset[i-1,3])

print(dataset[i,3])

break}

}

quit(save="ask")

}

dim=length(File)

#combined\_eta\_all=NULL

for (i\_t\_eta in 1:dim){

#print(File[i\_t\_eta])

t\_eta = read.delim(File[i\_t\_eta], sep = "\t", nrows=500, header = TRUE, stringsAsFactors = FALSE)

print(t\_eta)

#check for end of file / determination of Dimension of data - Dimdata

for (i in 1:500){

if (is.na(t\_eta[i,5])){

Dimdata=i-1

break}

}

t=head(unlist((t\_eta[,1])),Dimdata)

eta=head(unlist((t\_eta[,5])),Dimdata)

df\_t\_eta=data.frame(t,eta)

WriteXLS(df\_t\_eta, paste((File[i\_t\_eta]),".xlsx",sep=""), SheetNames = "Checkdata")

#combined\_eta\_all=c(combined\_eta\_all,eta)

}

}# End of Rawdata reading etc.