```
############################
#### Packages required ####
############################
install.packages("mailR", repos="http://cran.rstudio.com/")
library(mailR)  ## need to install the most updated version of JAVA
install.packages("plyr", repos="http://cran.rstudio.com/")
library(plyr)


########################
#### Paths required #####
########################
UPASfiles=list.files(path="C:/Research/Remote/Uploads",full.names=T)



#############################################################################
#### Read in previously checked log files and determine newly uploaded files ####
#############################################################################


ACread=readRDS("C:/Research/Remote/Uploads/Files/AlreadyChecked.rds")


Errors=data.frame(FileName=as.character(),
                  TotalVolume=as.numeric(),
                  FlowRate=as.numeric(),
                  SampleLength=as.character(),
                  Problem=as.character(), stringsAsFactors = F)



Info=data.frame(file.info(UPASfiles))
Info$Name=row.names(Info)
Info15=Info[,c("Name","size", "mtime")]

Info15$Letters=nchar(as.character(Info15$Name))
Info16=subset(Info15, Letters>80)


Info16$FinalLetters=substr(Info16$Name,nchar(Info16$Name)-6,nchar(Info16$Name)-4)

Info2=subset(Info16, FinalLetters!="___" & FinalLetters!="hhh" &
FinalLetters!="HHH")


Info2$NoData=0
```

```
TodayDate=Sys.Date()

Info3=as.data.frame(Info2[,1])

names(Info3)[1]="AlreadyChecked"


CheckedInfo=rbind(Info3,ACread)

CheckedInfo2=CheckedInfo

DuplicateFiles2=duplicated(CheckedInfo2)

Duplicate3=cbind(CheckedInfo2,DuplicateFiles2)
Duplicate4=subset(Duplicate3, DuplicateFiles2=="TRUE")

NonDuplicate=subset(Duplicate3, DuplicateFiles2=="FALSE")
NonDuplicate1=subset(NonDuplicate, select=-DuplicateFiles2)

Duplicate5=merge(Duplicate4,NonDuplicate1,by="AlreadyChecked",all.y=T)
NonDuplicate2=subset(Duplicate5, is.na(DuplicateFiles2))


NonDuplicate2$SubName1=substr(NonDuplicate2$AlreadyChecked,58,63)
NonDuplicate2$SubName2=substr(NonDuplicate2$AlreadyChecked,65,70)

NonDuplicate2$Filter=substr(NonDuplicate2$AlreadyChecked,79,84)

##########################################################################################
########################
#### account for some field staff not correctly entering the date/other data entry
errors with the UPAS ####
##########################################################################################
########################

numbers_only <- function(x) !grepl("\\D", x)

NonDuplicate2$CheckNumber=numbers_only(NonDuplicate2$SubName2)

NonDuplicate3=subset(NonDuplicate2, SubName1!="000000" | (SubName1=="000000" &
CheckNumber=="TRUE"))

NonDuplicate4=subset(NonDuplicate3, SubName2!="upasvs" & nchar(SubName2)>5 &
Filter!="filter")

NewFiles15=as.vector(NonDuplicate4$AlreadyChecked)

NewFiles2=data.frame(file.info(NewFiles15))
```

```
NewFiles2$Name=row.names(NewFiles2)
NewFiles3=NewFiles2[,c("Name","size", "mtime")]

NewFiles3$Letters=nchar(as.character(NewFiles3$Name))
NewFiles4=subset(NewFiles3, Letters>80)

#################################################################
#### Triggers an email if no new files were uploaded today ####
#################################################################

if (nrow(NewFiles4)==0){


  CountProblems=paste("Dear Matt, The code ran at", Sys.time(),
                      "You have 0 runs with no errors and 0
                      runs with at least one error")



send.mail(from = "pureair.upas@gmail.com",
            to = "mshupler@gmail.com",
            subject = "New PURE AIR data",
            body = CountProblems,
            smtp = list(host.name = "smtp.gmail.com", port = 465, user.name =,
passwd = , ssl = TRUE),
            authenticate = TRUE,
            send = TRUE)

  stop("the script must end here")


#########################################################
#### Code continues if new files were uploaded today ####
#########################################################

} else if (nrow(NewFiles4)>0) {NewFiles4$NoData=0}


Errors=data.frame(FileName=as.character(),
                  TotalVolume=as.numeric(),
                  FlowRate=as.numeric(),
                  SampleLength=as.numeric(),
                  Problem=as.character(), stringsAsFactors = F)

for (i in 1:nrow(NewFiles4)){

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

```
#### Examines if there was an upload problem due to low file size ####
#########################################################################

  if ((NewFiles4$size[i] <1000 | is.na(NewFiles4$size[i])) &
nchar(as.character(NewFiles4$Name[i]))>85){

    NewFiles4$NoData[i]=1



    Errors[i, "FileName"]=NewFiles4$Name[i]
    Errors[i,"SampleLength"]="No Data"
    Errors[i, "Problem"]="No Data Uploaded"

  }

  else {}

}

    ErrorTime=subset(Errors, !(is.na(SampleLength)))

     write.table(ErrorTime, file =
paste(paste("C:/Research/Remote/Uploads/Files/ErrorFiles/ErrorFiles-",TodayDate,sep=
""),".csv",sep=""),append=T, col.names=F)


NewFiles5=subset(NewFiles4, NoData==0)

NewFiles=as.vector(NewFiles5$Name)

Errors=data.frame(FileName=as.character(),
                  TotalVolume=as.numeric(),
                  FlowRate=as.numeric(),
                  SampleLength=as.numeric(),
                  Problem=as.character(), stringsAsFactors = F)


#########################################################################################
#### This loops through all newly uploaded files to check for UPAS errors ####
#########################################################################################


j=0
k=0
l=0

if (length(NewFiles)!=0) {
```

```
  TodayDate=Sys.Date()

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

  CheckFileHeader=read.table(NewFiles[i], header=F,quote = "",sep=",",
fill=T,na.string=c("","null","NaN"))[1,c(1:5)]


    CheckOne=substr(CheckFileHeader$V1,18,19)
    CheckOne=as.numeric(as.character(CheckOne))


################################################################################
#####
#### Different versions of UPAS firmware (e.g. v84) are read in slightly differently
####
################################################################################
#####


        if (CheckOne>=84) {

       Test=read.table(NewFiles[i], header=T,quote =
"",sep=",",skip=7,na.string=c("","null","NaN"))[,c(1:19)]}




        else { CheckFile=read.table(NewFiles[i], header=F,quote =
"",sep=",",skip=4,na.string=c("","null","NaN"))[c(1:5),]



               if (CheckFile$V1[1] ==as.character(as.factor("YYMMDDHHMMSS"))) {


                  Test=read.table(NewFiles[i], header=T,quote =
"",sep=",",skip=5,na.string=c("","null","NaN"))[,c(1:19)]
                    }

               else if (CheckFile$V1[1] ==as.character(as.factor("timestr")))

               {Test=read.table(NewFiles[i], header=T,quote =
"",sep=",",skip=4,na.string=c("","null","NaN"))[,c(1:19)]}

               else {WrongFirmware=read.table(NewFiles[i], header=T,quote =
"",sep=",",skip=4,na.string=c("","null","NaN"))[,c(1:19)]
```

```
                write.table(WrongFirmware, file =
paste(paste("C:/Research/Remote/Uploads/Files/ErrorFiles/WrongFirmware",NewFiles[i],
sep=""),".csv",sep=""),append=T, col.names=F)
                  }

          }

###############################################################################
##############
#### Check for problems with the total air sampled volume being less than
pre-defined cut off ####
###############################################################################
##############

    TotalVolumeSampled=Test$sampledVol[nrow(Test)]

    TotalVol= ifelse(abs(1440-TotalVolumeSampled) < 500, "Correct", "Total Volume
Sampled Error")

   ShortName=substr(NewFiles[i],64,119)


    if (TotalVol!="Correct") {


          file.copy(from=NewFiles[i],
to=paste(paste("C:/Research/Remote/Uploads/Files/ErrorFiles-TotalVolume",ShortName,s
ep="/"),"txt",sep="."))

          Errors[i, "FileName"]=ShortName

          Errors[i,"TotalVolume"]=TotalVolumeSampled
          Errors[i, "Problem"]=TotalVol

          ErrorComplete=subset(Errors, !(is.na(TotalVolume)))

          j=j+1

          write.table(ErrorComplete, file =
paste(paste("C:/Research/Remote/Uploads/Files/ErrorFiles/ErrorFiles-",TodayDate,sep=
""),".csv",sep=""),append=T, col.names=F)


          }
```

```
 }


################################################################################
#### Check for problems with the flow rate being within +/- 0.5 of 1.0 LPM ####
################################################################################

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

      Errors=data.frame(FileName=as.character(),
                        TotalVolume=as.numeric(),
                        FlowRate=as.numeric(),
                        SampleLength=as.numeric(),
                        Problem=as.character(), stringsAsFactors = F)

      CheckFileHeader=read.table(NewFiles[i], header=F,quote = "",sep=",",
fill=T,na.string=c("","null","NaN"))[1,c(1:5)]


      CheckOne=substr(CheckFileHeader$V1,18,19)
      CheckOne=as.numeric(as.character(CheckOne))


      if (CheckOne>=84) {


        Test=read.table(NewFiles[i], header=T,quote =
"",sep=",",skip=7,na.string=c("","null","NaN"))[,c(1:19)]}

      else {CheckFile=read.table(NewFiles[i], header=F,quote =
"",sep=",",skip=4,na.string=c("","null","NaN"))[c(1:5),]



      if (CheckFile$V1[1] ==as.character(as.factor("YYMMDDHHMMSS"))) {



        Test=read.table(NewFiles[i], header=T,quote =
"",sep=",",skip=5,na.string=c("","null","NaN"))[,c(1:19)]



      } else if (CheckFile$V1[1] ==as.character(as.factor("timestr")))

      {Test=read.table(NewFiles[i], header=T,quote =
"",sep=",",skip=4,na.string=c("","null","NaN"))[,c(1:19)]}
```

```
      }



      ShortName=substr(NewFiles[i],64,119)

        Test$RateCheck=NULL
        for (i in (1:(nrow(Test)-1))){
               Test$RateCheck[i]=if(abs(Test$volflow[i]-1) > 0.5 &
abs(Test$volflow[i+1]-1) >0.5) "Flow Rate Error" else NA
                  }

        RateError=subset(Test, !(is.na(RateCheck)))

      if (nrow(RateError)>1) {

        RateError1=RateError[1,]

        write.table(Test,
file=paste(paste("C:/Research/Remote/Uploads/Files/ErrorFiles-FlowRate",ShortName,se
p="/"),"txt",sep="."), col.names=T,quote=F,sep=',')



                Errors[i, "FileName"]=ShortName

                Errors[i,"FlowRate"]=RateError1$volflow

                Errors[i, "Problem"]=RateError1$RateCheck



                ErrorComplete2=subset(Errors, !(is.na(FlowRate)))


                k=k+1

    write.table(ErrorComplete2, file =
paste(paste("C:/Research/Remote/Uploads/Files/ErrorFiles/ErrorFiles-",TodayDate,sep=
""),".csv",sep=""),append=T, col.names=F)

    }

}
```

```
#####################################################################
#### Check for problems with the sample time being <47 hours ####
#####################################################################

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

    Errors=data.frame(FileName=as.character(),
                      TotalVolume=as.numeric(),
                      FlowRate=as.numeric(),
                      SampleLength=as.numeric(),
                      Problem=as.character(), stringsAsFactors = F)


    CheckFileHeader=read.table(NewFiles[i], header=F,quote = "",sep=",",
fill=T,na.string=c("","null","NaN"))[1,c(1:5)]


    CheckOne=substr(CheckFileHeader$V1,18,19)
    CheckOne=as.numeric(as.character(CheckOne))


    if (CheckOne>=84) {

      #need to change UPASfiles to NewFiles

      Test=read.table(NewFiles[i], header=T,quote =
"",sep=",",skip=7,na.string=c("","null","NaN"))[,c(1:19)]}

    else {



    CheckFile=read.table(NewFiles[i], header=F,quote =
"",sep=",",skip=4,na.string=c("","null","NaN"))[c(1:5),]


    if (CheckFile$V1[1] ==as.character(as.factor("YYMMDDHHMMSS"))) {


     Test=read.table(NewFiles[i], header=T,quote =
"",sep=",",skip=5,na.string=c("","null","NaN"))[,c(1:19)]
    } else if (CheckFile$V1[1] ==as.character(as.factor("timestr")))

    {Test=read.table(NewFiles[i], header=T,quote =
"",sep=",",skip=4,na.string=c("","null","NaN"))[,c(1:19)]}
```

```
    ShortName=substr(NewFiles[i],64,119)

    Test$timestr=as.factor(as.character(as.numeric(Test$timestr)))

    Test$Minsec=paste(substr(Test$timestr,9,10),substr(Test$timestr,11,12),sep=':')
    Test$Time=paste(substr(Test$timestr,7,8),Test$Minsec,sep=":")


Test$Date=paste(paste(paste("20",substr(Test$timestr,1,2),sep=""),substr(Test$timest
r,3,4),sep="/"),substr(Test$timestr,5,6),sep="/")

    Test$FullDate=as.POSIXct(strptime(paste(Test$Date, Test$Time, sep=" "),
format="%Y/%m/%d %H:%M:%S"))

    TimeDiff=difftime(Test$FullDate[nrow(Test)], Test$FullDate[1],units="hours")


    if (TimeDiff < 47){

      file.copy(from=NewFiles[i],
to=paste(paste("C:/Research/Remote/Uploads/Files/ErrorFiles-SampleTime",ShortName,se
p="/"),"txt",sep="."))


    Errors[i, "FileName"]=ShortName
    Errors[i,"SampleLength"]=TimeDiff
    Errors[i, "Problem"]="Time Error"

    ErrorTime=subset(Errors, !(is.na(SampleLength)))

    l=l+1

     write.table(ErrorTime, file =
paste(paste("C:/Research/Remote/Uploads/Files/ErrorFiles/ErrorFiles-",TodayDate,sep=
""),".csv",sep=""),append=T, col.names=F) }

    }
  }

}


#######################################################################
#### Determine total number of error free and erroneous files ####
#######################################################################
```

```
  Good=(length(NewFiles)*3)-j-k-l

  Bad=j+k+l


  NewFilesNew=as.data.frame(cbind(NewFiles3$Name,NewFiles3$size))

  NewFiles2=as.data.frame(NewFiles)

  NewFilesNew2=as.data.frame(NewFilesNew[,1])
  names(NewFilesNew2)="AlreadyChecked"

  AlreadyChecked=rbind(ACread,NewFilesNew2)

################################################################################
################
#### Add all the currently assessed files to .rds file so they are not examined on
the next run ####
################################################################################
################


  saveRDS(AlreadyChecked,
file="C:/Research/Remote/Uploads/Files/AlreadyChecked.rds")

  CountProblems=paste("Dear Matt, The code ran at", Sys.time(),
                      "You have", Good, "runs with no errors and",
                      Bad, "runs with at least one error")


#####################################################
#### Send an email with the details of the run  ####
#### first need to enable IMAP in gmail account ####
#####################################################



  send.mail(from = "pureair.upas@gmail.com",
            to = "mshupler@gmail.com",
            subject = "New PURE AIR data",
            body = CountProblems,
            #attach.files = "update_R.R",
            smtp = list(host.name = "smtp.gmail.com", port = 465, user.name = ,
passwd = , ssl = TRUE),
            authenticate = TRUE,
            send = TRUE)
```