

Supplementary File 1

```
#-----  
--  
# TITLE: Create oximetry database from raw data files  
# Author: Clarence Wong  
# Last updated: 2/7/2017  
# RStudio v. 1.0.136  
#-----  
--  
  
library(readr)  
require(lubridate)  
require(TTR)  
require(xts)  
require(zoo)  
library(reshape2)  
  
#-----  
--  
# Read all data files and save as R object  
#-----  
--  
  
master<-0  
for (i in 1:8)  
{  
  file <-  
  read.csv(paste("D:/SS/R_data/FINAL_oximetry_data/",as.character(i),".csv",sep=""))  
  master <- rbind(master,file)  
}  
master$date_time <- paste(master$Date, master$Time..GMT.)  
master$date_time <- mdy_hms(master$date_time)  
converted_master <- master[,c(58,3:57)]  
  
save(converted_master,file = "converted_master.RData")  
  
database_times <- read_csv("D:/SS/R_data/database_times.csv")  
date_vector <- database_times[,c(1,5,6,7,11,12)]  
  
date_vector$start_date_time <- mdy_hms(paste(date_vector$`Date of  
surgery`,date_vector$`Monitoring Start`))  
date_vector$end_date_time <- mdy_hms(paste(date_vector$`Date of  
surgery`,date_vector$`Monitoring End`))
```

```
date_vector$surg_start_date_time <- mdy_hms(paste(date_vector$`Date of
surgery`,date_vector$`Start Time`))
date_vector$surg_end_date_time <- mdy_hms(paste(date_vector$`Date of
surgery`,date_vector$`Finish Time`))

converted_date_vector <- date_vector[,c(1,7,8,9,10)]

save(converted_date_vector,file = "converted_date_vector.RData")

rm(master,date_vector,file)

#-----
--
# 1. Convert data types and locate monitoring periods
# 2. Identify oximetry values at various time points
# 3. Compute percentage change from baseline
# 4. Identify and locate problematic data
#-----
--

minutes_taken_as_baseline <- 2.5
minutes_interval <- 5

secs_taken_as_baseline <- minutes_taken_as_baseline*60
secs_interval <- minutes_interval*60

load("converted_master.RData")
load("converted_date_vector.RData")
print("data loaded. check data version")

oximetry_L <-
as.numeric(levels(converted_master$RSO2_A1)[converted_master$RSO2_A1])
oximetry_R <-
as.numeric(levels(converted_master$RSO2_A2)[converted_master$RSO2_A2])
PSI <- as.numeric(levels(converted_master$PSI)[converted_master$PSI])

# monitoring duration
duration_mins <-
difftime(converted_date_vector$end_date_time,converted_date_vector$start_date_time,uni
ts = "mins")
duration_secs <-
difftime(converted_date_vector$end_date_time,converted_date_vector$start_date_time,uni
ts = "secs")

locate_start = seq(-1,-1,length.out = dim(converted_date_vector)[1])
```

```

for (i in 1:dim(converted_date_vector)[1]){
if(length(which(converted_date_vector$start_date_time[i]==converted_master$date_time))
==1)
{
locate_start[i] <-
which(converted_date_vector$start_date_time[i]==converted_master$date_time)
}
}

# create final_oximetry data frame
final_oximetry <- data.frame()
baseline_L_mu<-baseline_L_std<-baseline_L_N<-baseline_R_mu<-baseline_R_std<-
baseline_R_N<-rep(9999,dim(converted_date_vector)[1])
num_time_pts <- rep(1,40)

for(j in 1:dim(converted_date_vector)[1])
{
# for each patient
if(locate_start[j]==-1)
{
p_id <- j
time_id<-minute_from_baseline<-percentage_total_monitoring_period<-L_delta<-
L_mu<-L_sig<-L_N<-R_delta<-R_mu<-R_sig<-R_N<-PSI_mu<-9999

} else{

locate_baseline <- locate_start[j]+secs_taken_as_baseline/2
locate_times <- seq(0,0)
num_measurements <- (as.numeric(duration_secs)[j]-
secs_taken_as_baseline)/%/%secs_interval +1
num_time_pts[j] <- num_measurements
locate_times[1] <- locate_baseline
locate_times[2] <- locate_times[1] + secs_interval/2
locate_times[2:num_measurements]<-
seq(locate_times[2],locate_start[j]+as.numeric(duration_secs[j])/2,by=secs_interval/2)
locate_times[num_measurements+1]<-locate_start[j]+as.numeric(duration_secs[j])/2

baseline_L_mu[j] <- mean(oximetry_L[locate_start[j]:(locate_baseline-1)],na.rm =
TRUE)
baseline_L_std[j] <- sd(oximetry_L[locate_start[j]:(locate_baseline-1)],na.rm = TRUE)
baseline_L_N[j] <- length(oximetry_L[locate_start[j]:(locate_baseline-1)])-
sum(is.na(oximetry_L[locate_start[j]:(locate_baseline-1)]))

```

```

baseline_R_mu[j] <- mean(oximetry_R[locate_start[j]:(locate_baseline-1)],na.rm =
TRUE)
baseline_R_std[j] <- sd(oximetry_R[locate_start[j]:(locate_baseline-1)],na.rm = TRUE)
baseline_R_N[j] <- length(oximetry_R[locate_start[j]:(locate_baseline-1)])-
sum(is.na(oximetry_R[locate_start[j]:(locate_baseline-1)]))

L_delta <- L_mu <- L_sig <- L_N <- R_delta <- R_mu <- R_sig <- R_N <- PSI_mu <-
seq(0,0)

for (k in 1:num_measurements)
{
  L_mu[k] <- mean(oximetry_L[locate_times[k]:(locate_times[k+1]-1)],na.rm = TRUE)
  L_sig[k] <- sd(oximetry_L[locate_times[k]:(locate_times[k+1]-1)],na.rm = TRUE)
  L_N[k] <- length(oximetry_L[locate_times[k]:(locate_times[k+1]-1)])-
sum(is.na(oximetry_L[locate_times[k]:(locate_times[k+1]-1)]))

  R_mu[k] <- mean(oximetry_R[locate_times[k]:(locate_times[k+1]-1)],na.rm = TRUE)
  R_sig[k] <- sd(oximetry_R[locate_times[k]:(locate_times[k+1]-1)],na.rm = TRUE)
  R_N[k] <- length(oximetry_R[locate_times[k]:(locate_times[k+1]-1)])-
sum(is.na(oximetry_R[locate_times[k]:(locate_times[k+1]-1)]))

  PSI_mu[k] <- mean(PSI[locate_times[k]:(locate_times[k+1]-1)],na.rm = TRUE)
}

L_delta <- (L_mu/baseline_L_mu[j] -1)*100
R_delta <- (R_mu/baseline_R_mu[j] -1)*100

time_id <- 1:num_measurements
minute_from_baseline <- c(seq(minutes_interval,minutes_interval*(num_measurements-
1), by = minutes_interval),as.numeric(duration_mins[j]-minutes_taken_as_baseline))
p_id <- rep(j,num_measurements)
percentage_total_monitoring_period <-
((minute_from_baseline*60+secs_taken_as_baseline)/as.numeric(duration_secs[j]))*100

}

temp_df <-
data.frame(p_id,time_id,minute_from_baseline,percentage_total_monitoring_period,L_delt
a,L_mu,L_sig,L_N,R_delta,R_mu,R_sig,R_N,PSI_mu)
final_oximetry <- rbind(final_oximetry,temp_df)
rm(temp_df)

}

missing_L <- unique(final_oximetry$p_id[is.na(final_oximetry$L_delta)])

```

```

missing_R <- unique(final_oximetry$p_id[is.na(final_oximetry$R_delta)])
percentage_total_missing_L <-
100*(rle(final_oximetry$p_id[is.na(final_oximetry$L_delta)])$lengths) /
(num_time_pts[unique(final_oximetry$p_id[is.na(final_oximetry$L_delta)])])
percentage_total_missing_R <-
100*(rle(final_oximetry$p_id[is.na(final_oximetry$R_delta)])$lengths) /
(num_time_pts[unique(final_oximetry$p_id[is.na(final_oximetry$R_delta)])])
missing_data <- unique(final_oximetry$p_id[(final_oximetry$L_delta==9999)])
missing_data <- missing_data[!is.na(missing_data)]
missing_PSI <- unique(final_oximetry$p_id[is.na(final_oximetry$PSI_mu)])
percentage_total_missing_PSI <-
100*(rle(final_oximetry$p_id[is.na(final_oximetry$PSI_mu)])$lengths) /
(num_time_pts[unique(final_oximetry$p_id[is.na(final_oximetry$PSI_mu)])])

print("there are missing delta oximetry values in the following patients")
print(missing_L)
print(percentage_total_missing_L)

print(missing_R)
print(percentage_total_missing_R)

print(missing_data)

print("there are missing PSI values in the following patients")
print(missing_PSI)
print(percentage_total_missing_PSI)

other_data <-
data.frame(num_time_pts,baseline_L_mu,baseline_L_std,baseline_L_N,baseline_R_mu,
baseline_R_std,baseline_R_N)
other_data[is.na(other_data)]<-9999
save(other_data, file="other_data.RData")

final_oximetry[is.na(final_oximetry)]<-9999
save(final_oximetry,file = "final_oximetry.RData")

#-----
--
# 1. Convert baseline characteristic database from wide to long format
# 2. Incorporating oximetry data in the database with time as a nested data in the hierarchy
# 3. Create final database
#-----
--

```

```
load("final_oximetry.RData")
load("other_data.RData")
print("check if final oximetry is latest")

baseline_results <- read.csv("D:/SS/R_data/FINAL_oximetry_data/all_baseline.csv",
sep=",", stringsAsFactors=FALSE)

baseline_results$baseline_L_mu <- other_data$baseline_L_mu
baseline_results$baseline_L_std <- other_data$baseline_L_std
baseline_results$baseline_L_N <- other_data$baseline_L_N
baseline_results$baseline_R_mu <- other_data$baseline_R_mu
baseline_results$baseline_R_std <- other_data$baseline_R_std
baseline_results$baseline_R_N <- other_data$baseline_R_N

baseline_results$P_id <- index(baseline_results)

baseline_results[baseline_results == "#N/A"]<-9999

#generate baseline_results with the same number of rows as final oximetry
baseline_results <- baseline_results[rep(seq_len((40)),num_time_pts),]

all_results <- cbind(baseline_results,final_oximetry)
if (sum(1*(all_results$P_id != all_results$p_id))==0)
{
  all_results <- all_results[,c(which(colnames(all_results)==="p_id"),1:109,112:122)]
}

save(all_results,file = "all_results.RData")

#UNCOMMENT TO WRITE CSV
#-----
write.csv(all_results, file="all_results.csv")
```