
STATISTICAL METHODS IN RESEARCH

Assignment 1

MARCH 4, 2017
MOHIT KADUSKAR (1544187)
NAFISA JASSANI (1543203)

1. Introduction:

The dataset is acquired by carrying out a controlled experiment on a driving simulator where n=68 volunteers drove under four different conditions: No distraction, cognitive distraction, emotional distraction, and sensorimotor distraction. Different response variables were recorded including speed, acceleration, brake force, steering, and lane position signals, and different explanatory variables including perinasal EDA, palm EDA, heart rate, breathing rate, and facial expression signals; biographical and psychometric covariates were also obtained.

The dataset is organized into various folders T001, T002 and so on which indicate the different volunteers on which the experiment was carried. Each of those folders include sub-folders (Sessions) like, BL for the Baseline, PD for the Practice Drive, RD for the Relaxing Drive, ND for the Normal Drive, CD for the Cognitive Drive, ED for the Emotional Drive, MD for the Sensorimotor Drive, and FDL or FDN for the Failure Drive. Each of the sub-folder includes various data channels like Heart Rate(HR), Breathing Rate (BR), Perinasal Perspiration (pp), Performance Response (res) and Palm EDA(peda) signal, FACS Signals, thermal data, stm, facial (avi), ROI and OT.

The project aims at performing quality control on the following data channels:

BR: Breathing Rate Signal. This Excel file contains three synced columns: Frame #; Time; and, Breathing Rate signal.

HR: Heart Rate Signal. This Excel file contains three synced columns: Frame #; Time; and Heart Rate signal.

peda: Palm EDA Signal. This Excel file contains three synced columns: Frame #; Time; and, Palm EDA signal.

pp: Perinasal EDA Signal. This Excel file contains four synced columns: Frame #; Time; Perinasal EDA signal; and, Noise Reduction (NR) Perinasal EDA signal.

res: Performance Response Variables. This Excel file contains 7 synced columns: Frame #; Time; Speed signal; Acceleration signal; Brake Force signal; Steering signal; and, Lane Position signal.

Below are the ranges for performing quality control on above mentioned data channels:

Palm EDA Sensor: 10 - 4,700 k Ohm

Breathing Rate Signals: 4 - 70 bpm

Heart Rate Signals: 40 - 140 bpm

Acceleration Signals: > 0 km/hr

PP Signal: Already pre-cleaned.

Once the data channels are processed by above mentioned ranges next step involves plotting the unclean as well as processed data for visualization. More specifically we plot the signals for various data channels, for each data channel we group the various signals for each session. For all data channels besides PP and acceleration, we need to plot both the raw and cleaned signals. Since PP signal is pre-cleaned only 1 graph needs to be plot. For acceleration signals, we replace all negative values by NA (not to be plotted).

2. Methodology

• Part 1: Generation of Index File:

- As we are given the file named Dataset-Table-Index.xlsx which has status of the data sessions with respect to data channels for each subject. We have replaced all 0, 1, N

and -1 with blank values in a file named 'Dataset-Table-Index_output' and reused this for our processing.

2. We traverse the file and for each subject and session we check if the value is not 'NA', and assign value for the following things:
 - '0' if the file is not present
 - '-1' if the values are not in valid range as described above (for BR, HR and peda signals). The values which are outside the range are printed in 'outputFinal.txt' with the respective file name and their location. The file is created in the output folder of working directory.
 - '1' if the values are valid
 3. The new values analyzed are saved in output file named 'UpdatedIndex' created in output folder of working directory.
 4. For res files, we create a new column NR Acceleration where all values less than 0 for acceleration are replaced by NA and new files are generated in UpdatedRESFiles folder. (<https://drive.google.com/open?id=0B6HyjgeSQWUzODZGZ3Irc2xSaFE>)
 5. All values for speed between -0.1 and 0.1 is replaced by 0 and new column is created named 'NR Speed'. And all values for Braking greater than 300 is replaced by 300 and new column 'NR Braking' is created. Copies of all new files are created in UpdatedRESFiles folder present in working directory.
- Part 2: Generation of Output File:
 1. The index file created in Part 1 is used as reference (UpdatedIndex)
 2. We know the various Sessions to be used for processing are as follows: BL, CD , ED , MD, ND , PD , RD , FD, where FD=FDN+FDL and the various data channels used are: BR, HR, peda, pp and res.
 3. Single pdf file is created for each data channel - BR, HR and peda. For each session two graphs are plotted on each page. The purpose for using one page per Session is that the data is clearly visible and the cumulative data for each data channel is available in single pdf file.
 4. Similarly for pp data channel, for each session one graph is plotted on each page since pre cleaned data is available.
 5. The output files are created in output folder of working directory and the naming convention used is OUTPUT (Data channel).pdf, for example: OUTPUTHR.pdf where HR is the data channel.

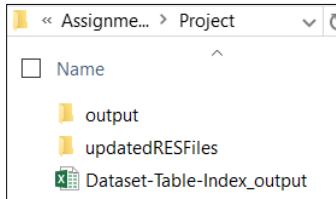
6. Since the file generated for res with 8 graphs on each page is too heavy and takes too long to open, we have created one png image for each session. The graphs present in each image are as follows: Speed, Acceleration, Braking, Steering, Lane Position, NR Speed, NR Acceleration and NR Braking, where NR plots are the cleaned data for Speed, Acceleration and Braking respectively. The graph images are generated in the output folder of working directory and the naming convention used is as follows: OUTPUTGraphs_res_(Session_Name).png; for example: OUTPUTGraphs_res_CD.png where CD is the Session.
7. To plot the raw data, we collect the data in the following way: for each data channel, we append the raw data (with values -1 or 1 in the index file created) belonging to same Session for all Subjects in one data frame. We then plot this data using ggplot, grouped by the various subjects.
8. Same process is repeated for all clean data (with value = 1 in the index file created).

3. Assumptions:

1. We assume that in all res files, only Frame, Time, Speed, Acceleration, Braking, Steering and Lane Position are relevant columns. We have observed that in some files extra column called 'Lane offset' is present as the second last column. We observed for few files that this location is fixed for Lane offset column, and hence if more than 7 columns are present we ignore the second last column.

4. Instructions for execution:

1. The folder structure for working directory is as follows:



2. The graph files generated (all pdf and png images) and the updated index is generated in the output folder of the working directory.
3. The Res files that get updated are saved in updatedRESFiles folder.
4. The index Dataset-Table-Index_output is a file we have used where we replaced all 0, 1, -1 and N values by blanks and used that as reference when validating if the data in the file was in range.
5. Set the values of the variables 'workingDir' to this working directory (- Project folder in zip file) and 'dataPath' to the folder where the data is present.
6. UpdatedRES files can be downloaded from:
<https://drive.google.com/open?id=0B6HyjgeSQWUzODZGZ3Irc2xSaFE>
7. Output files generated can be downloaded from:
<https://drive.google.com/open?id=0B6HyjgeSQWUzbGk3SGRIN0daVXM>

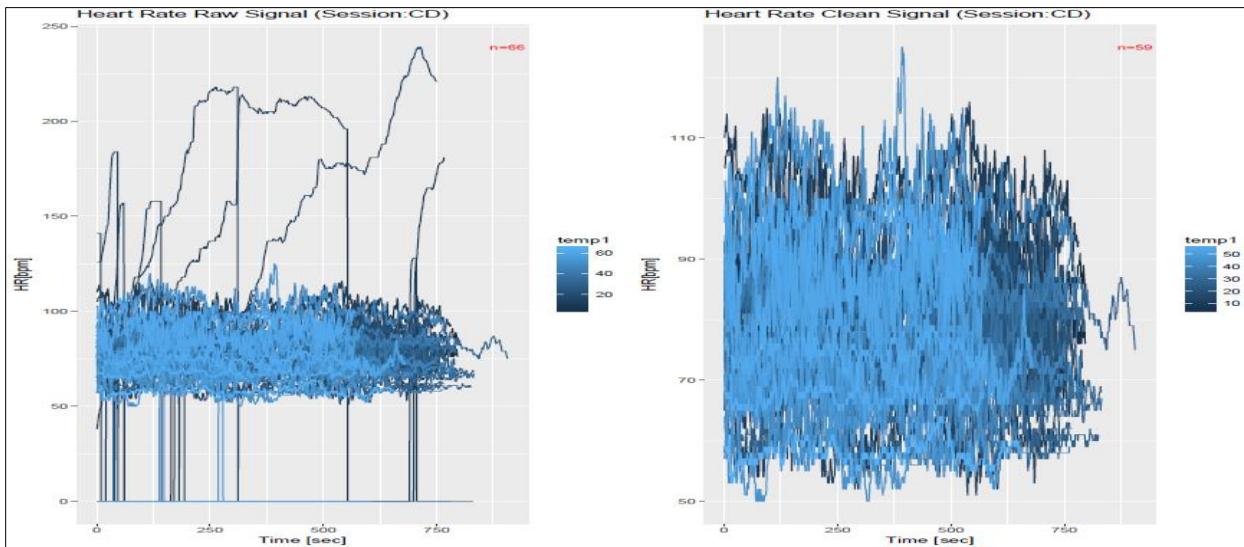
Assignment 1

5. Results and Observations:

For each data channel, for each Session we get the plot of raw and clean data. The Subjects eliminated for clean data are described below for each session:

Data Channel: HR (OUTPUTHR.pdf created in output folder of Working Directory)

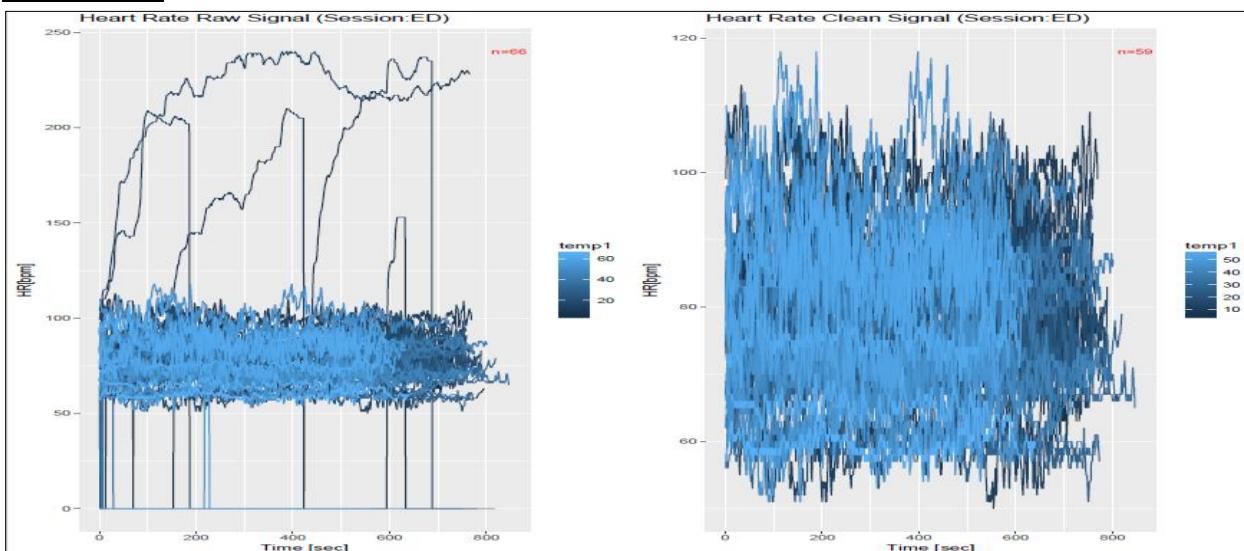
Session: CD

**Observations:**

Data of 7 Subjects are eliminated for the following files:

- T001-005.HR: 164 data entries out of range
- T009-005.HR: 505 data entries out of range
- T012-005.HR: 703 data entries out of range
- T015-005.HR: 541 data entries out of range
- T041-005.HR: 744 data entries out of range
- T064-005.HR: 561 data entries out of range
- T074-005.HR: 586 data entries out of range

Session: ED

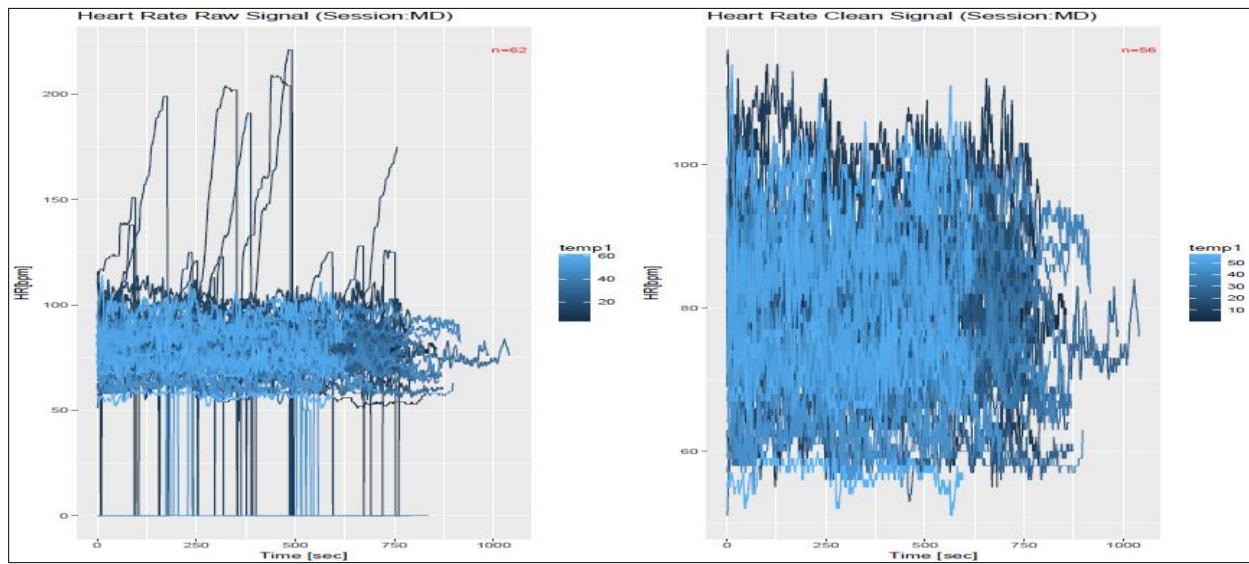


Assignment 1

Observations:

Data of 7 Subjects are eliminated for the following files:

- T001-006.HR: 153 data entries out of range
- T009-006.HR: 565 data entries out of range
- T012-006.HR: 732 data entries out of range
- T015-006.HR: 721 data entries out of range
- T041-006.HR: 793 data entries out of range
- T064-006.HR: 559 data entries out of range
- T074-006.HR: 551 data entries out of range

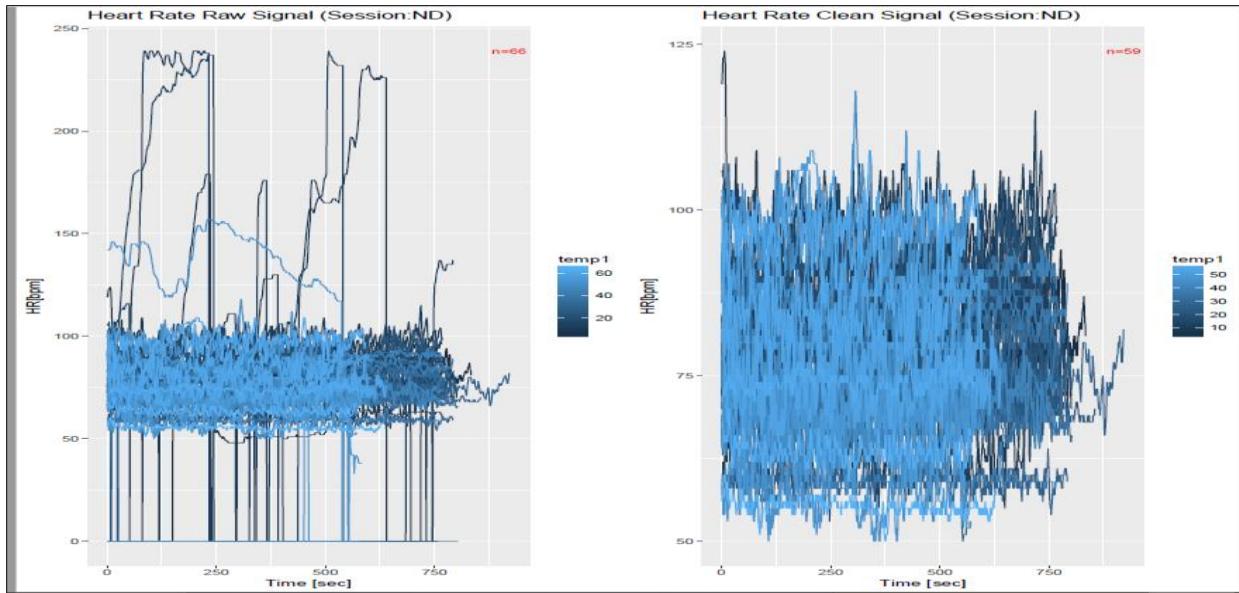
Session: MD**Observations:**

Data of 6 Subjects are eliminated for the following files:

- T001-007.HR: 178 data entries out of range
- T009-007.HR: 222 data entries out of range
- T012-007.HR: 455 data entries out of range
- T015-007.HR: 258 data entries out of range
- T041-007.HR: 834 data entries out of range
- T074-007.HR: 545 data entries out of range

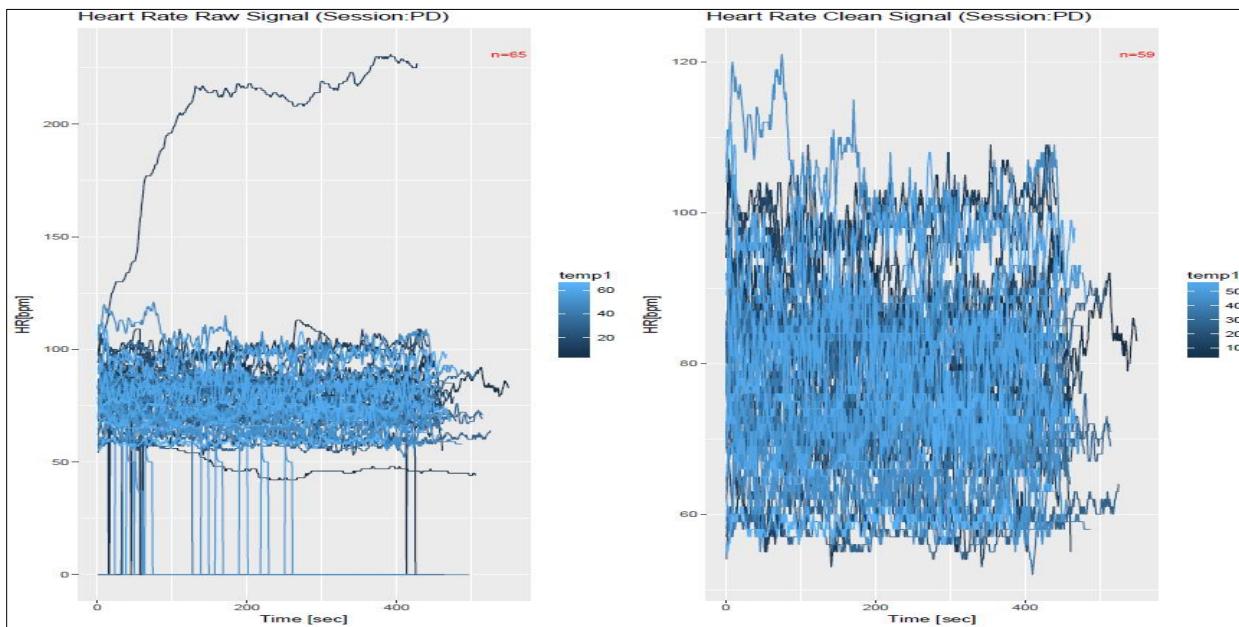
Session: ND

Assignment 1

**Observations:**

Data of 7 Subjects are eliminated for the following files:

- T001-004.HR: 150 data entries out of range
- T009-004.HR: 332 data entries out of range
- T012-004.HR: 625 data entries out of range
- T015-004.HR: 717 data entries out of range
- T041-004.HR: 780 data entries out of range
- T064-004.HR: 280 data entries out of range
- T074-004.HR: 570 data entries out of range

Session: PD

Assignment 1

Observations:

Data of 6 Subjects are eliminated for the following files:

T001-002.HR: 60 data entries out of range

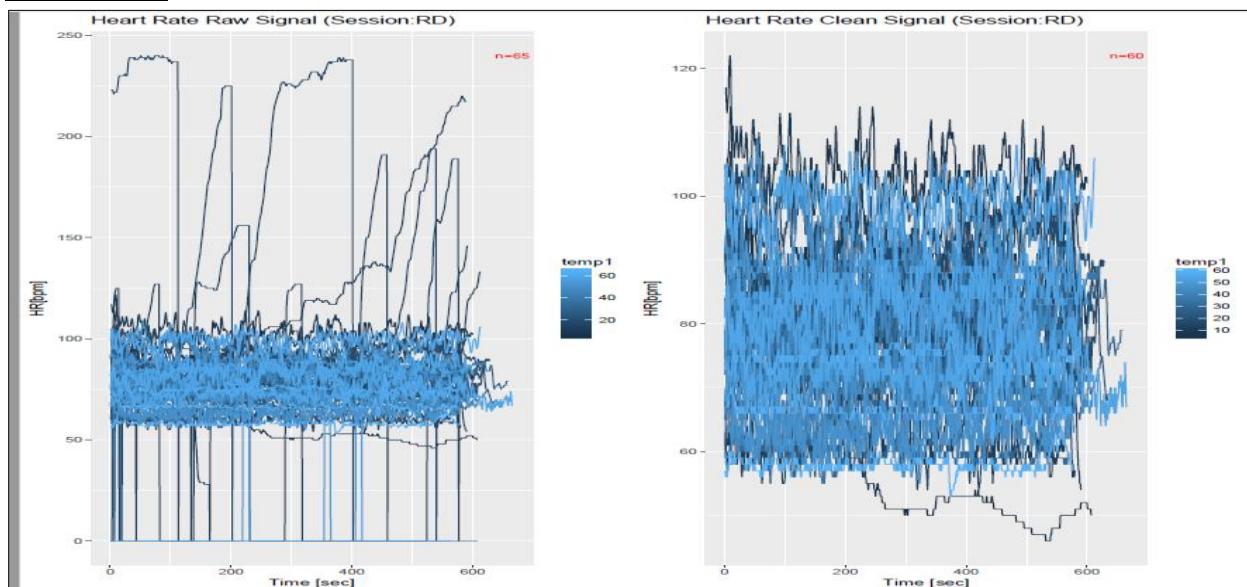
T009-002.HR: 8 data entries out of range

T012-002.HR: 439 data entries out of range

T015-002.HR: 376 data entries out of range

T041-002.HR: 371 data entries out of range

T074-002.HR: 333 data entries out of range

Session: RD**Observations:**

Data of 5 Subjects are eliminated for the following files:

T009-003.HR: 152 data entries out of range

T012-003.HR: 412 data entries out of range

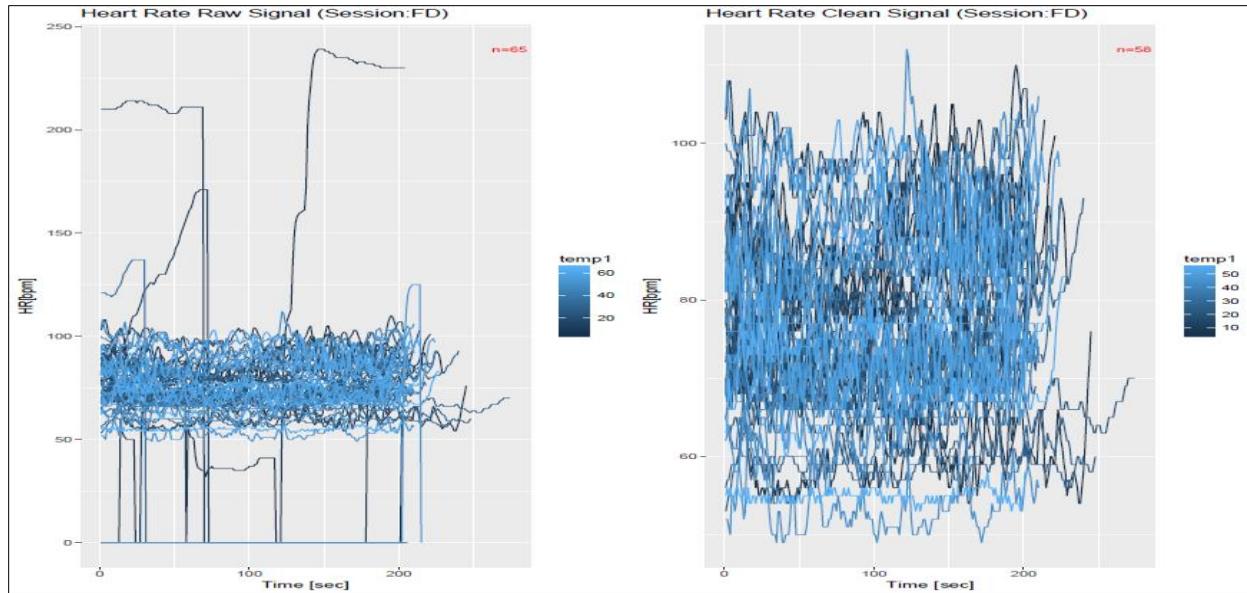
T015-003.HR: 415 data entries out of range

T041-003.HR: 592 data entries out of range

T074-003.HR: 533 data entries out of range

Session: FD=FDN+FDL

Assignment 1

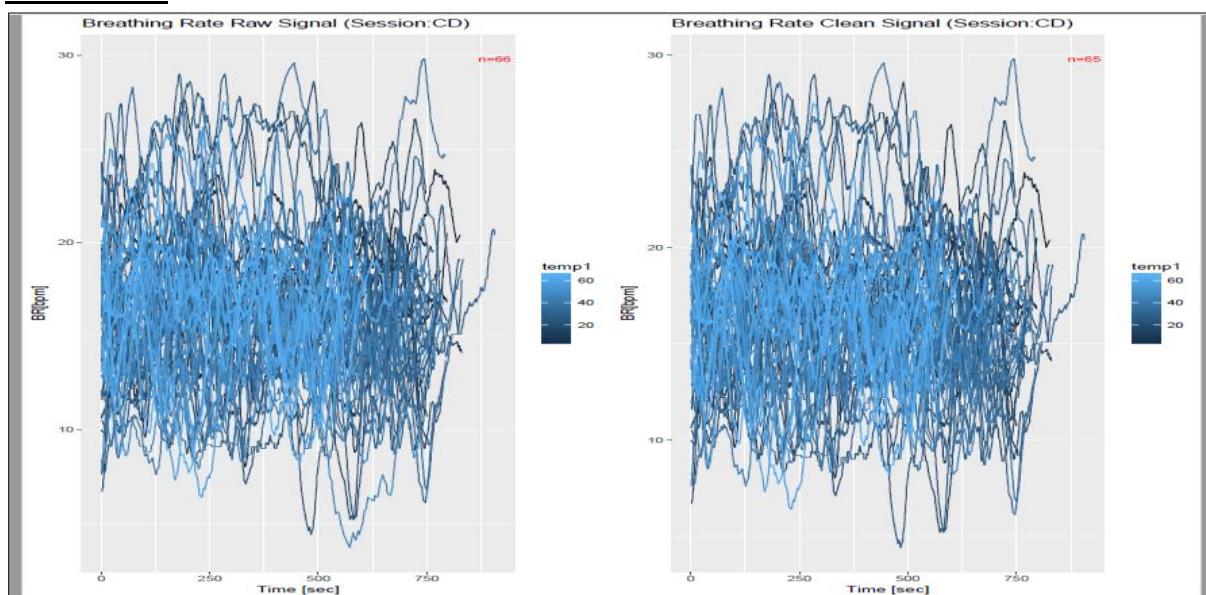
**Observations:**

Data of 7 Subjects are eliminated for the following files:

- T001-008.HR: 201 data entries out of range
- T009-008.HR: 138 data entries out of range
- T012-008.HR: 206 data entries out of range
- T015-008.HR: 128 data entries out of range
- T041-008.HR: 174 data entries out of range
- T064-008.HR: 200 data entries out of range
- T074-008.HR: 203 data entries out of range

Data Channel: BR (OUTPUTBR.pdf created in output folder of Working Directory)

Session: CD



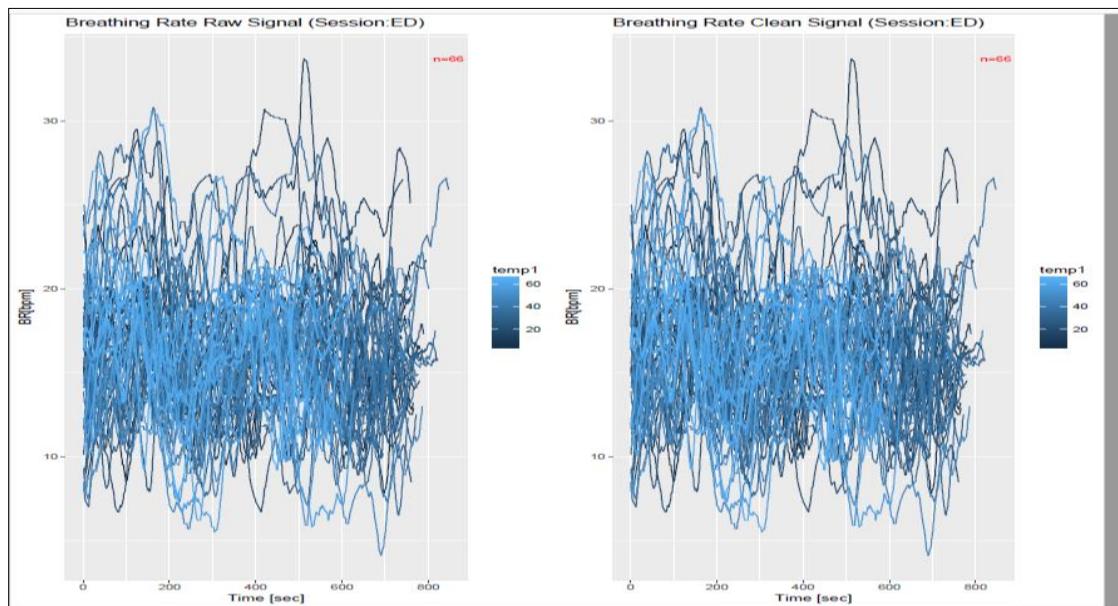
Assignment 1

Observations:

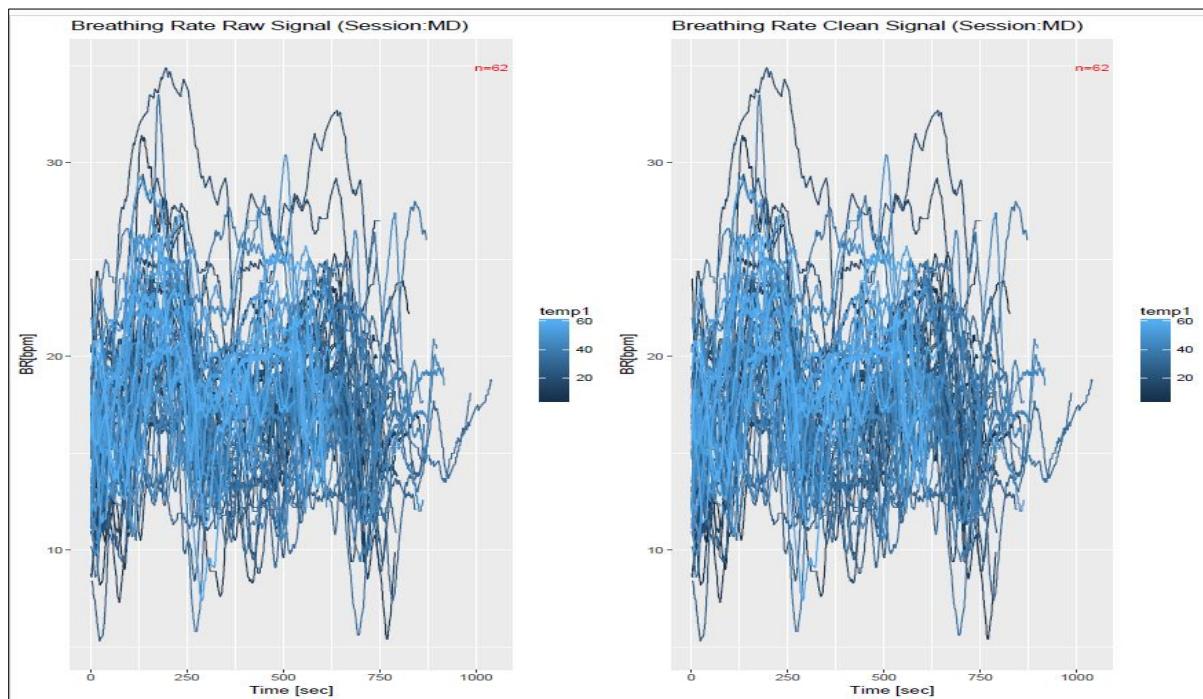
Data of 1 Subjects are eliminated for the following files:

T054-005.BR: 8 data entries out of range

Session: ED

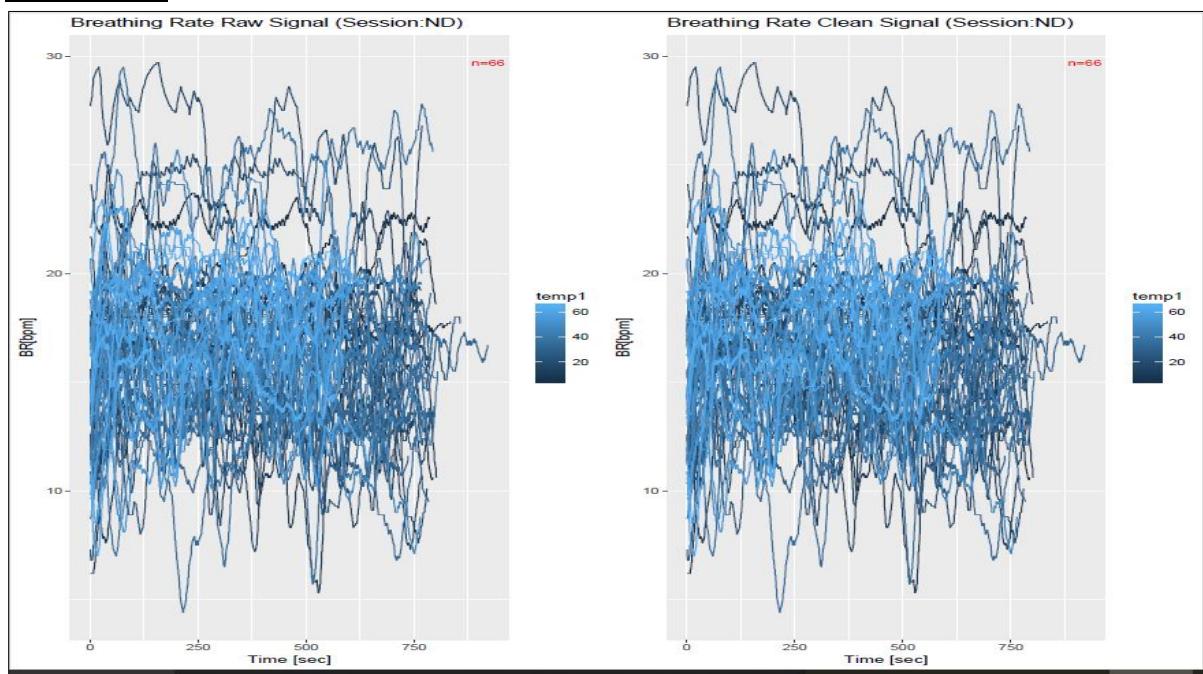


Session: MD

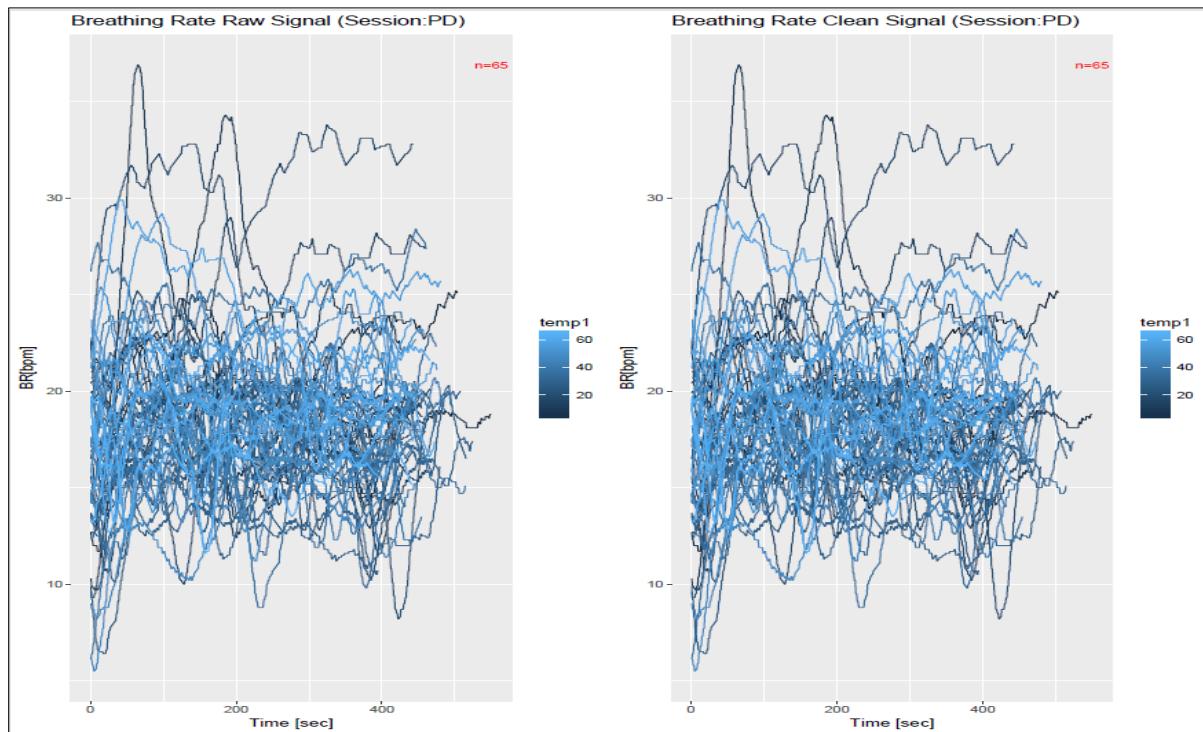


Assignment 1

Session: ND

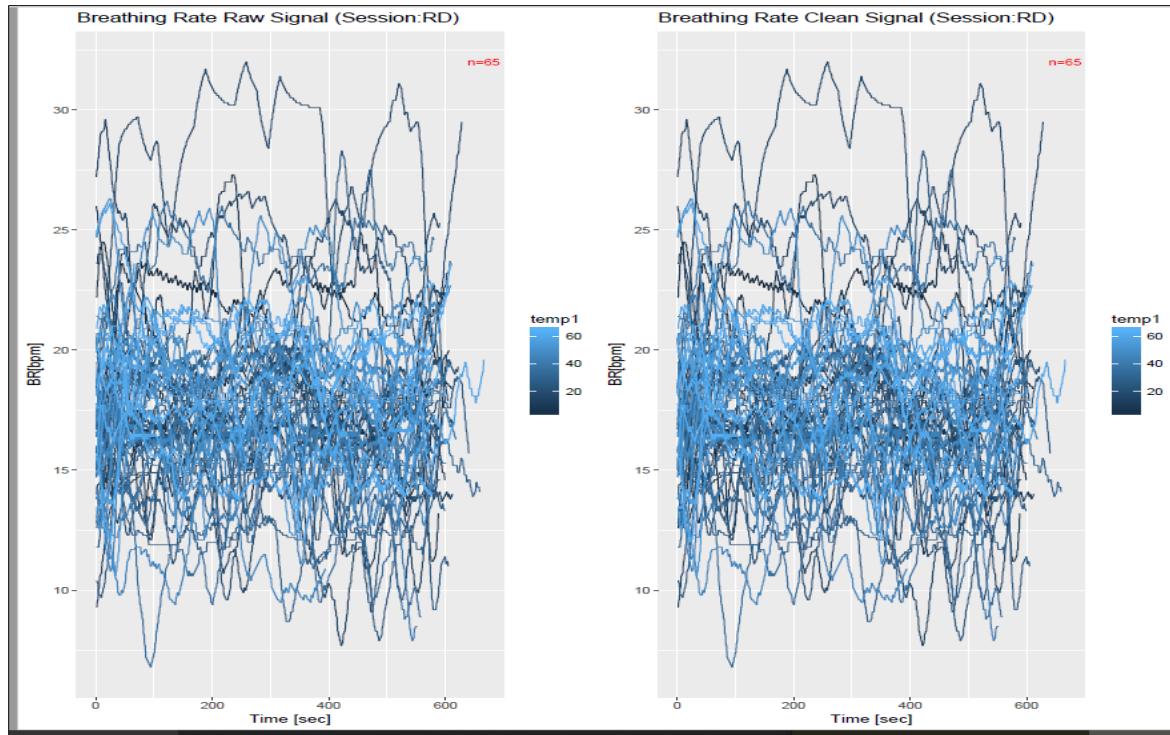


Session: PD

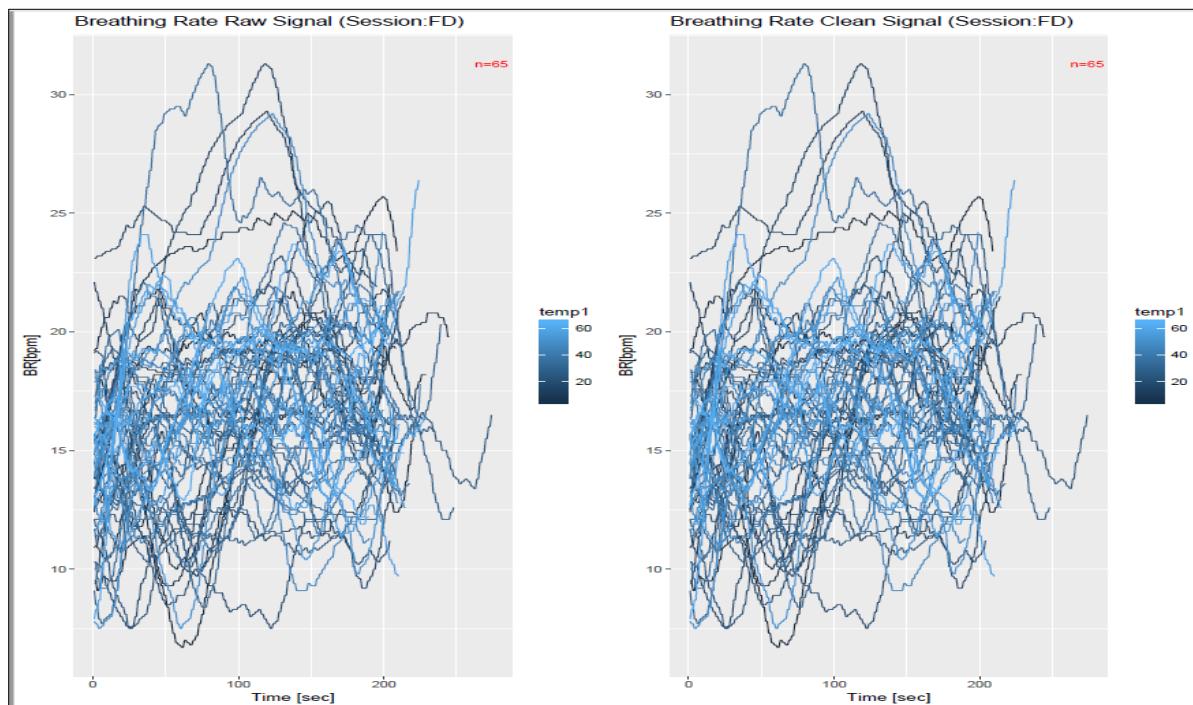


Session: RD

Assignment 1



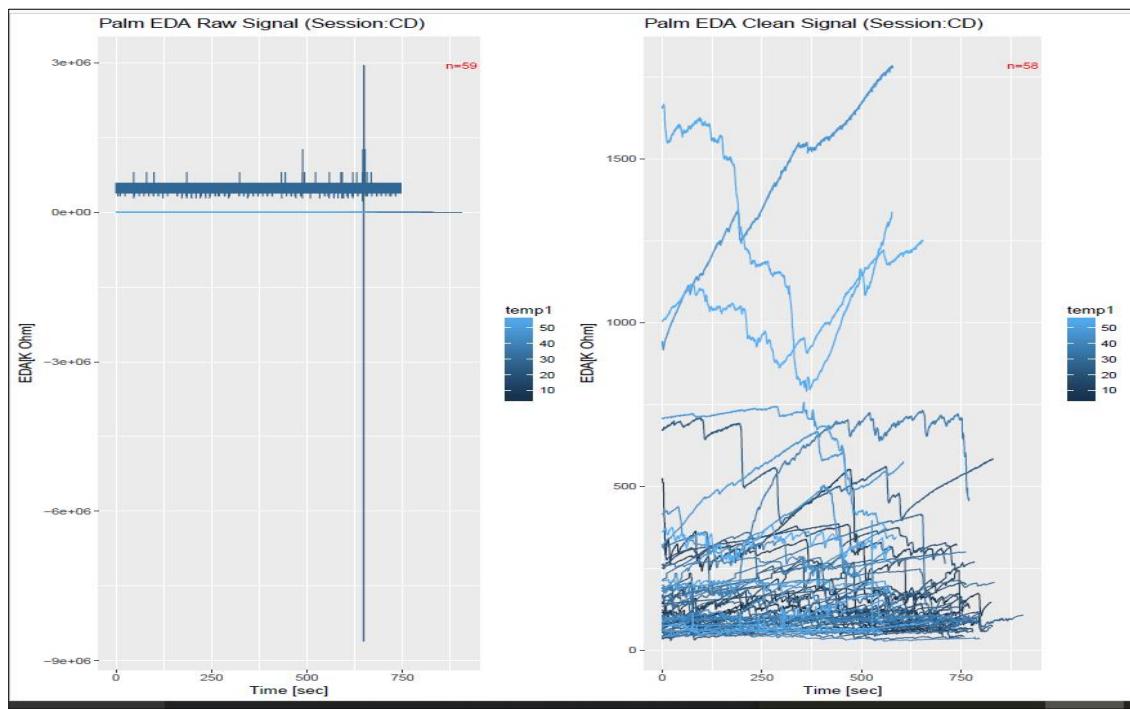
Session: FD



Assignment 1

Data channel: peda(OUTPUTpeda.pdf created in output folder of Working Directory)

Session: CD

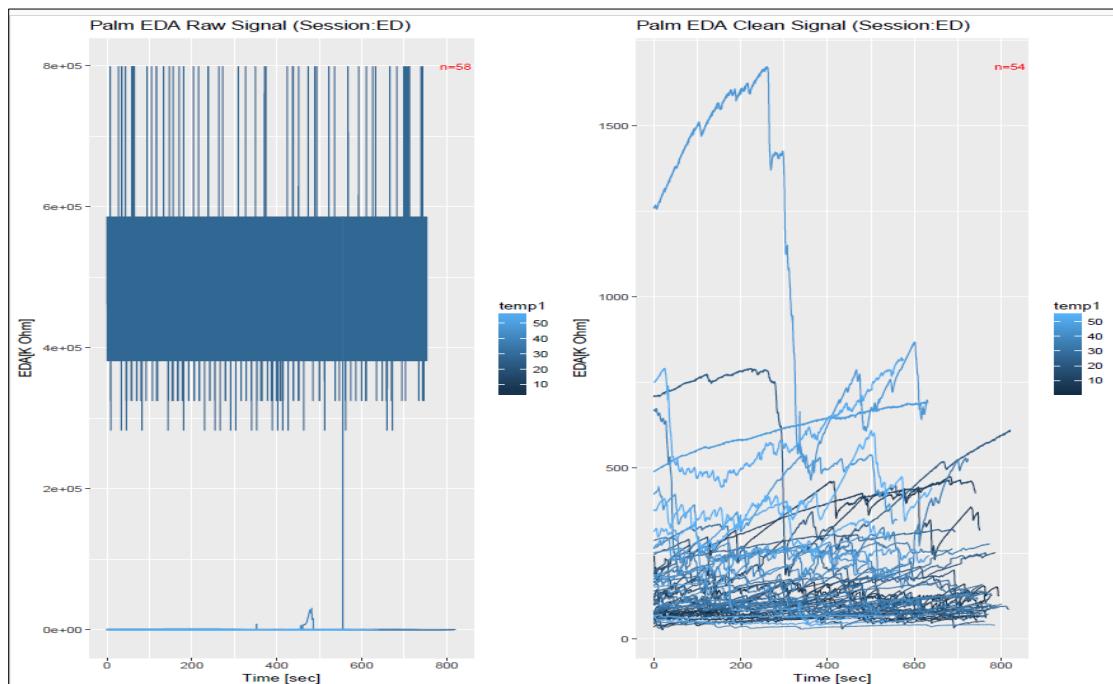


Observations:

Data of 1 Subjects are eliminated for the following files:

T035-005.peda: 18651 data entries out of range

Session: ED



Assignment 1

Observations:

Data of 4 Subjects are eliminated for the following files:

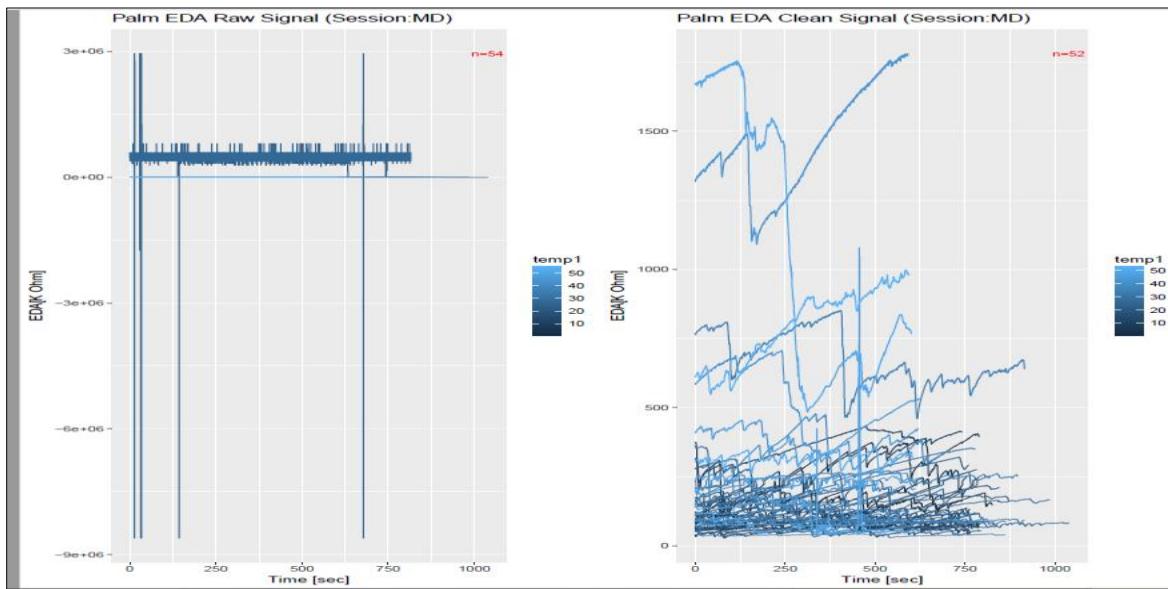
T035-005.peda: 37480 data entries out of range

T041-006.peda: 13 data entries out of range

T055-006.peda: 738 data entries out of range

T075-006.peda: 4 data entries out of range

Session: MD



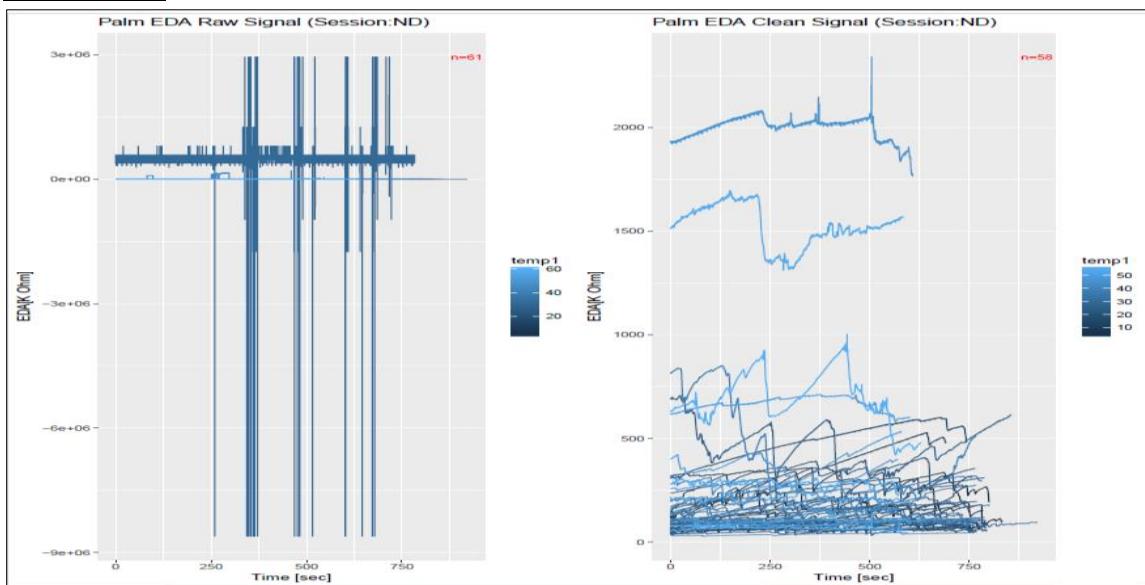
Observations:

Data of 2 Subjects are eliminated for the following files:

T025-008.peda: 12491 data entries out of range

T035-007.peda: 20401 data entries out of range

Session: ND



Assignment 1

Observations:

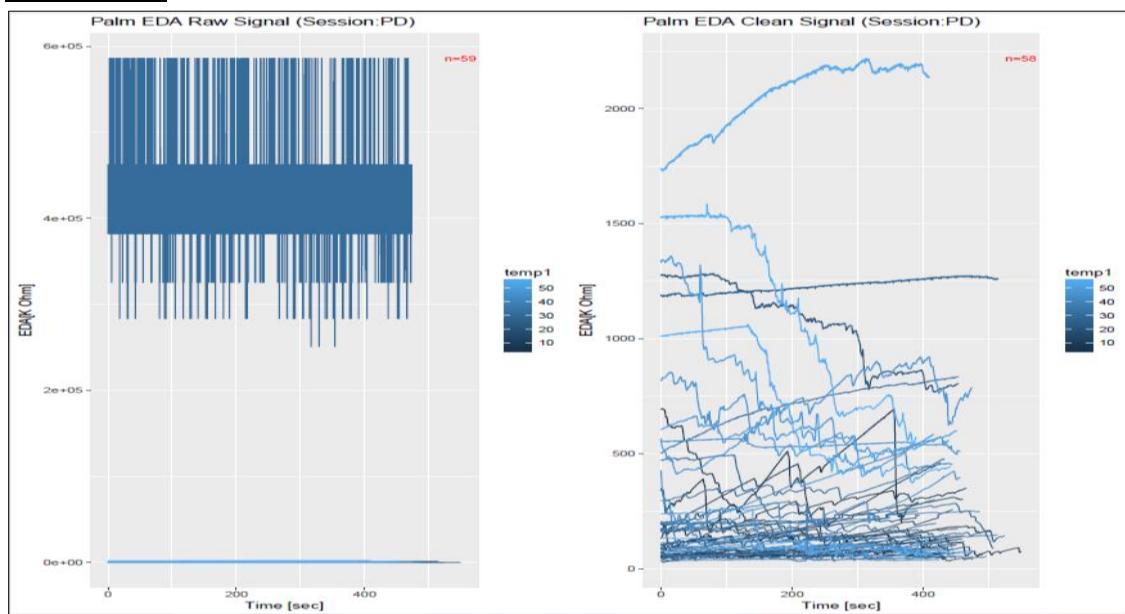
Data of 3 Subjects are eliminated for the following files:

T035-004.peda: 19576 data entries out of range

T041-004.peda: 121 data entries out of range

T055-004.peda: 6192 data entries out of range

Session: PD

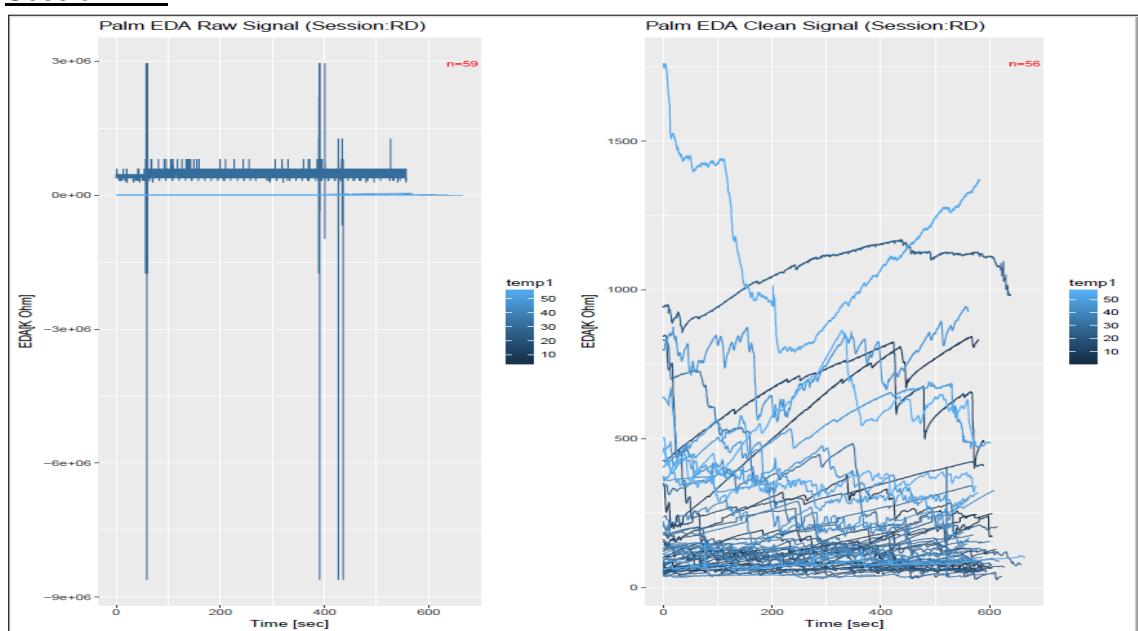


Observations:

Data of 1 Subjects are eliminated for the following files:

T035-002.peda: 11826 data entries out of range

Session: RD



Assignment 1

Observations:

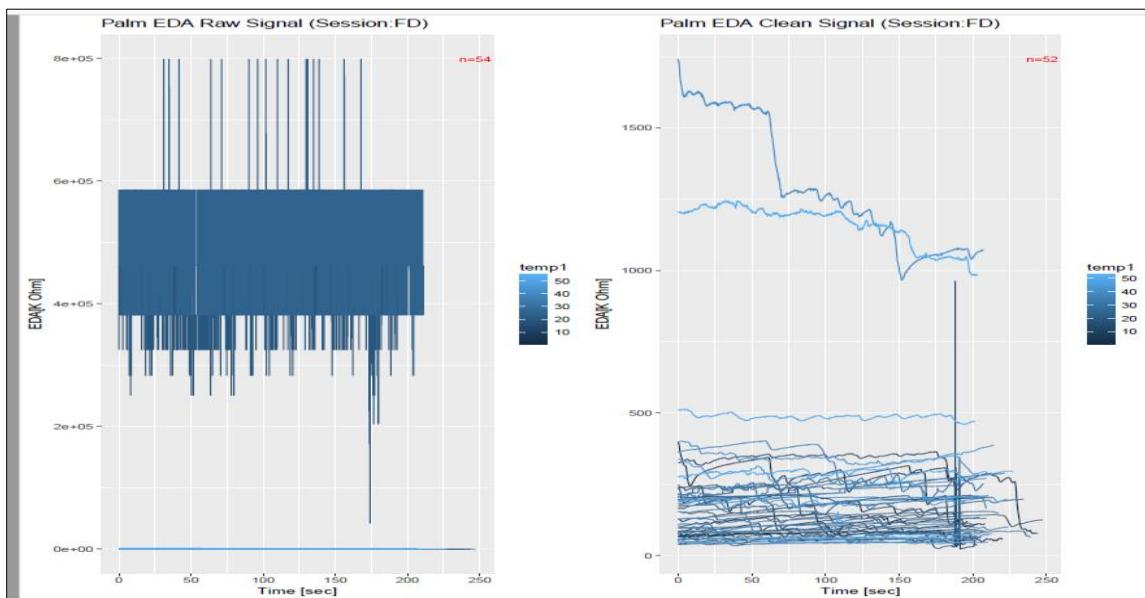
Data of 3 Subjects are eliminated for the following files:

T035-003.peda: 13926 data entries out of range

T055-003.peda: 3650 data entries out of range

T079-003.peda: 4 data entries out of range

Session: FD



Observations:

Data of 2 Subjects are eliminated for the following files:

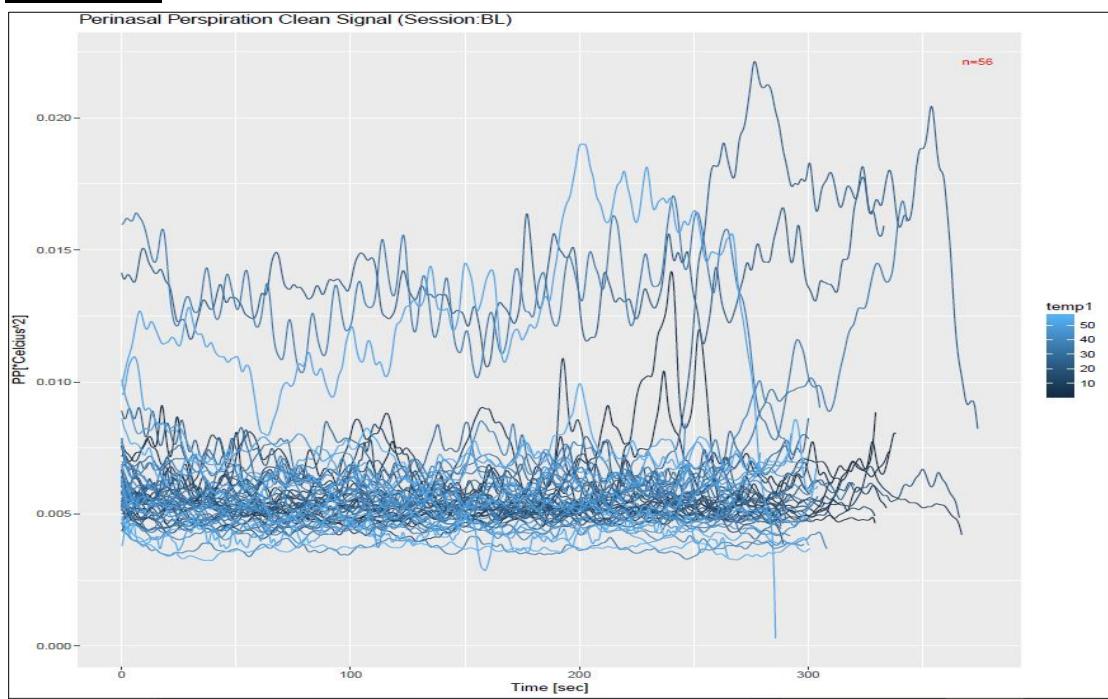
T025-008.peda: 13926 data entries out of range

T035-008.peda: 5276 data entries out of range

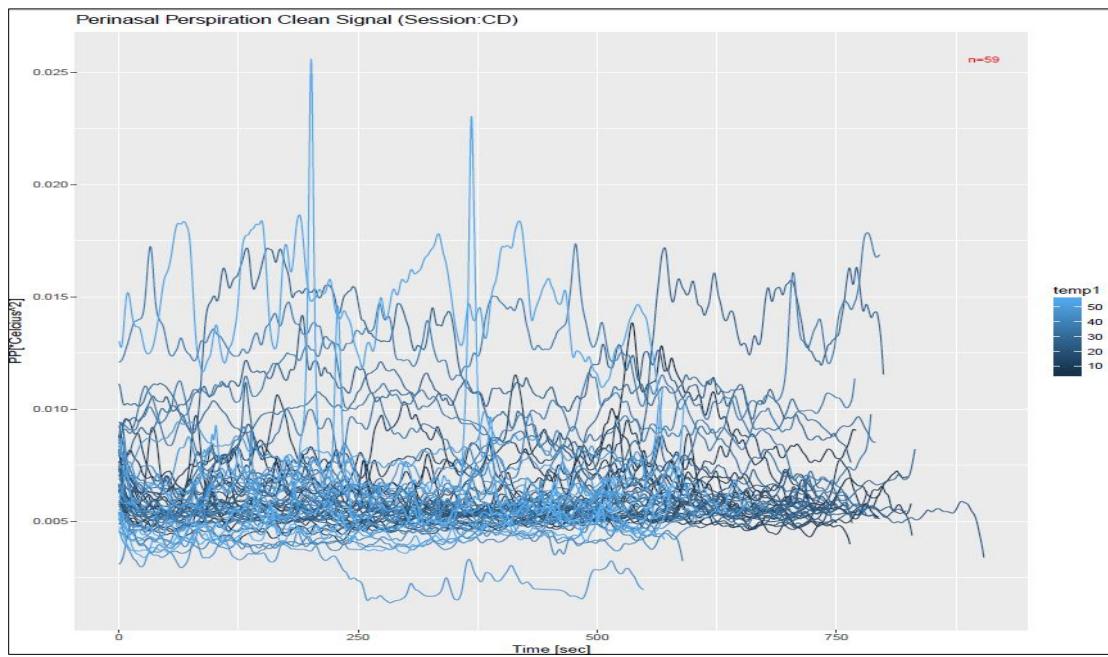
Assignment 1

Data Channel: PP(OUTPUTpp.pdf created in output folder of Working Directory)

Session: BL

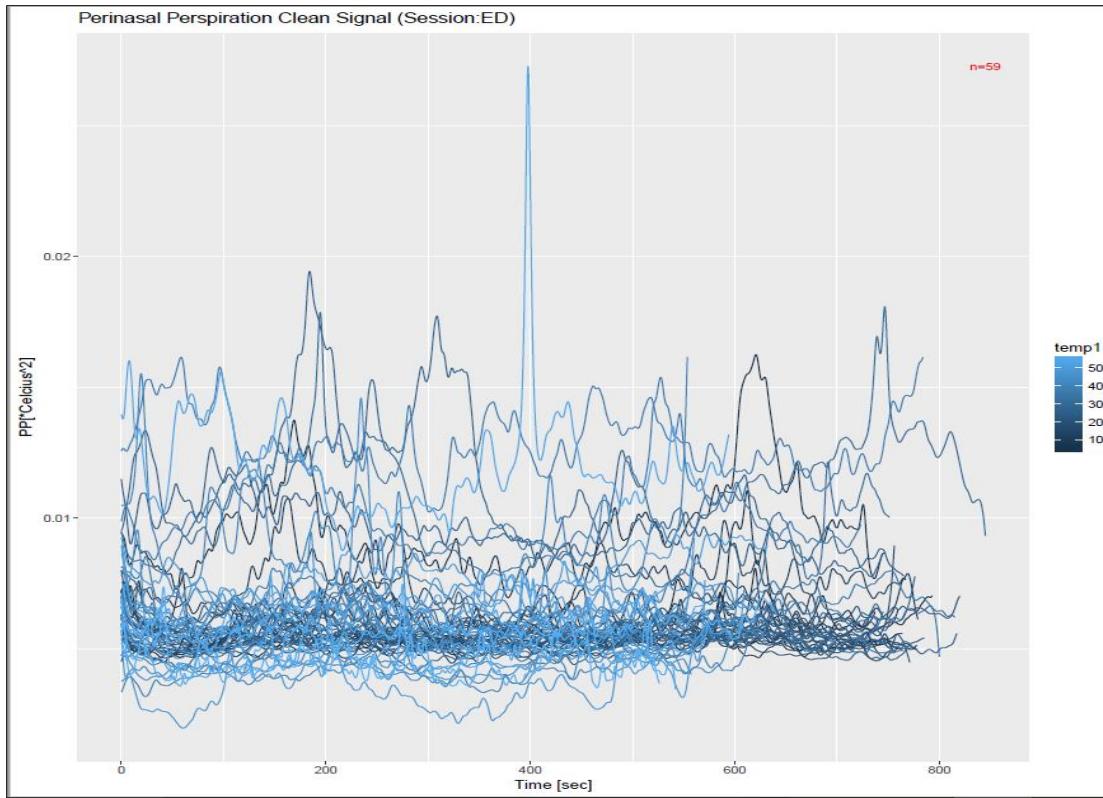
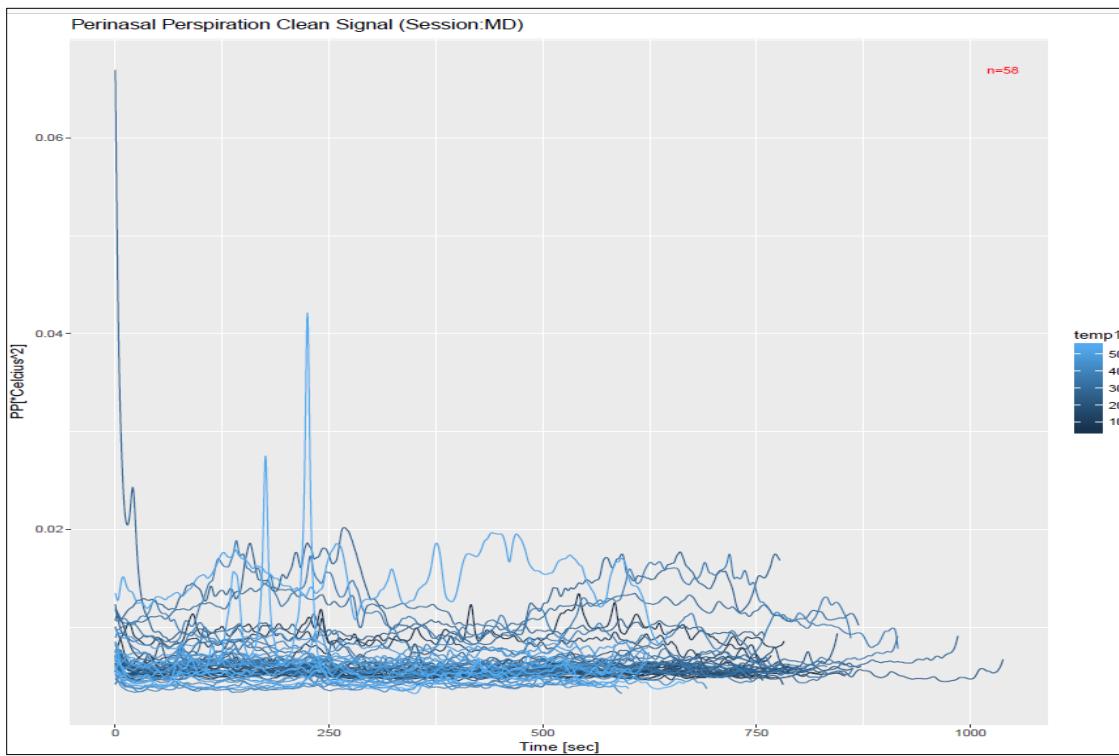


Session: CD

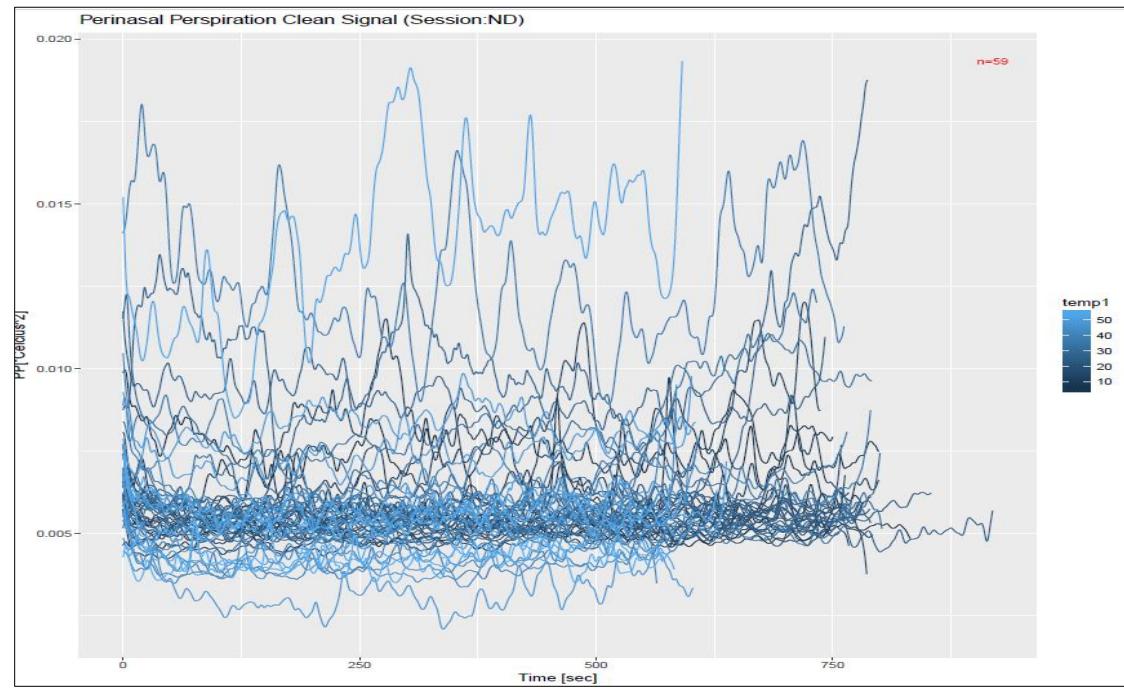
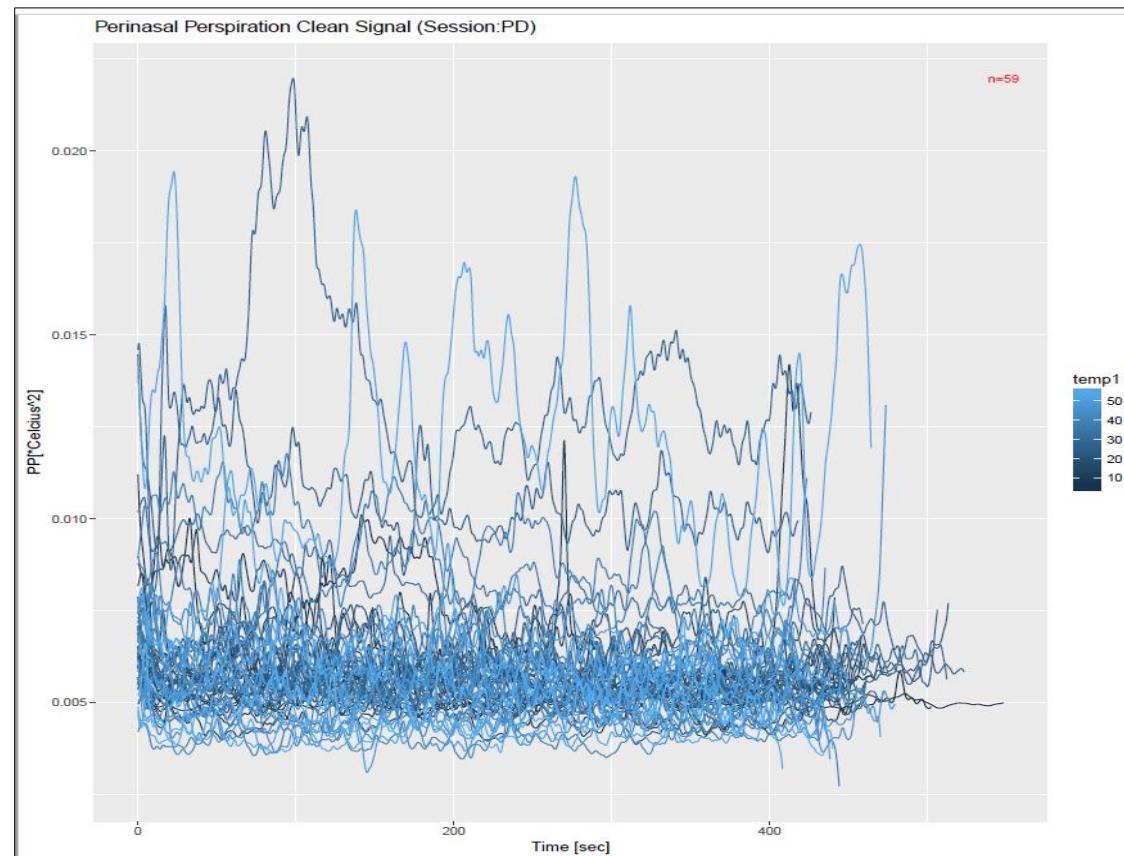


Session: ED

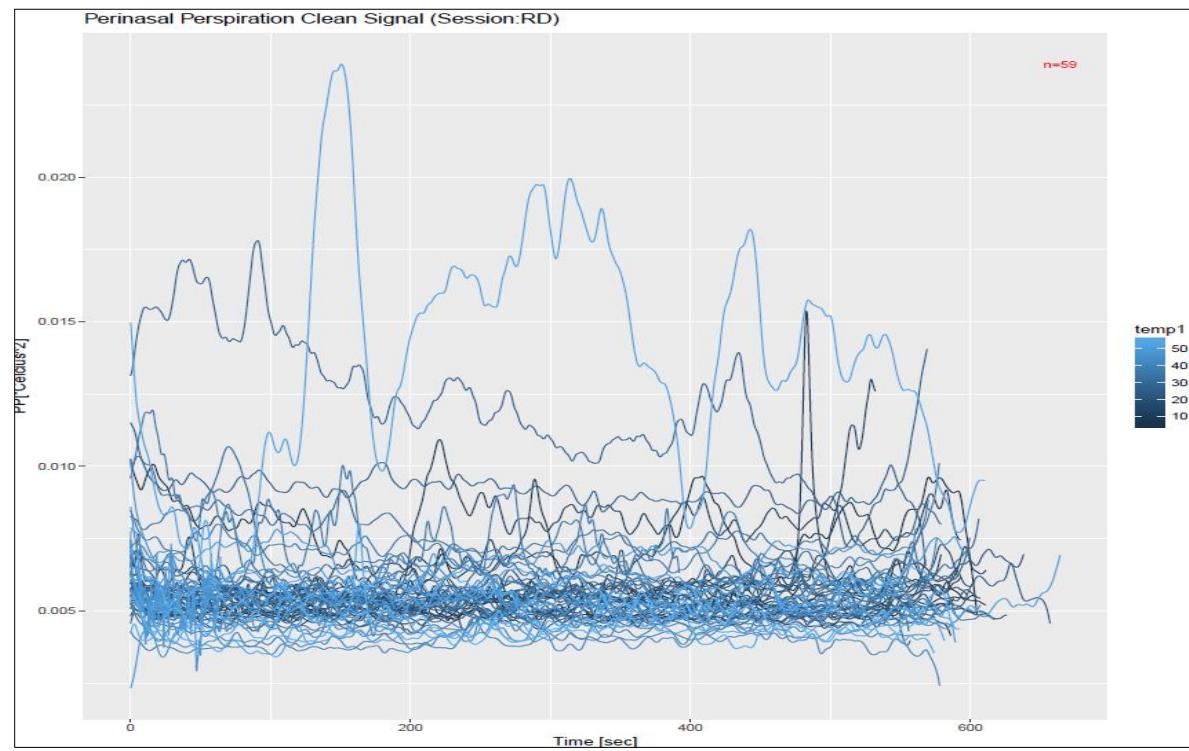
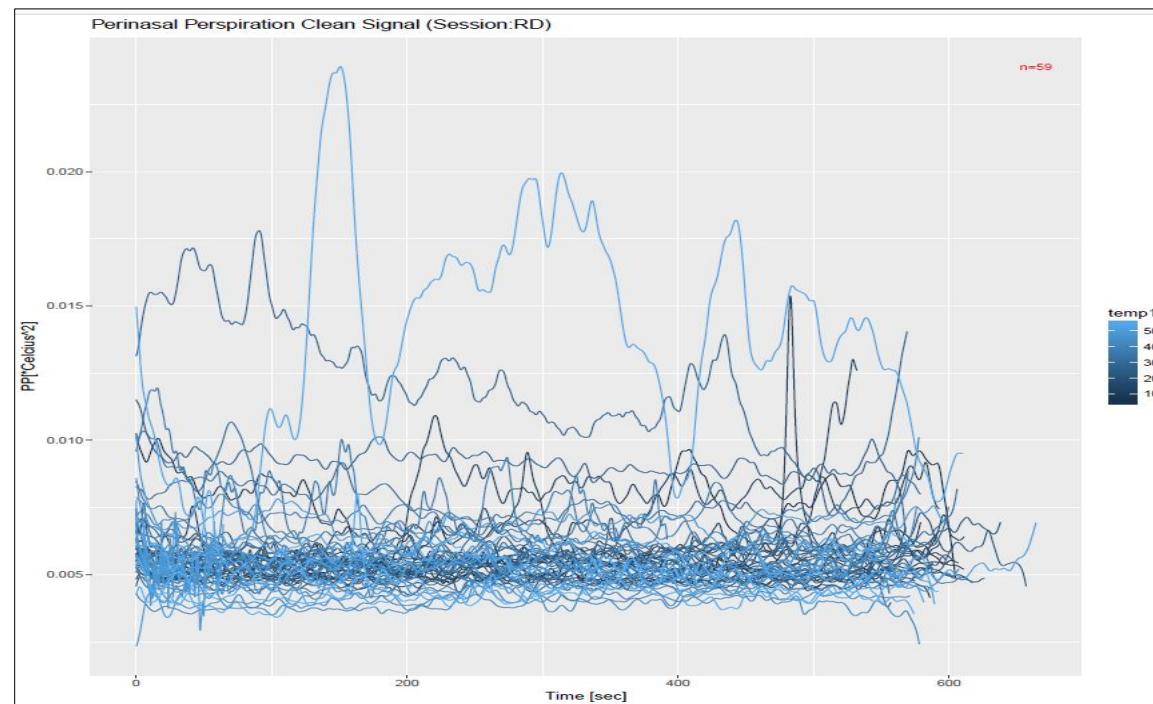
Assignment 1

Session: MD

Assignment 1

Session: NDSession: PD

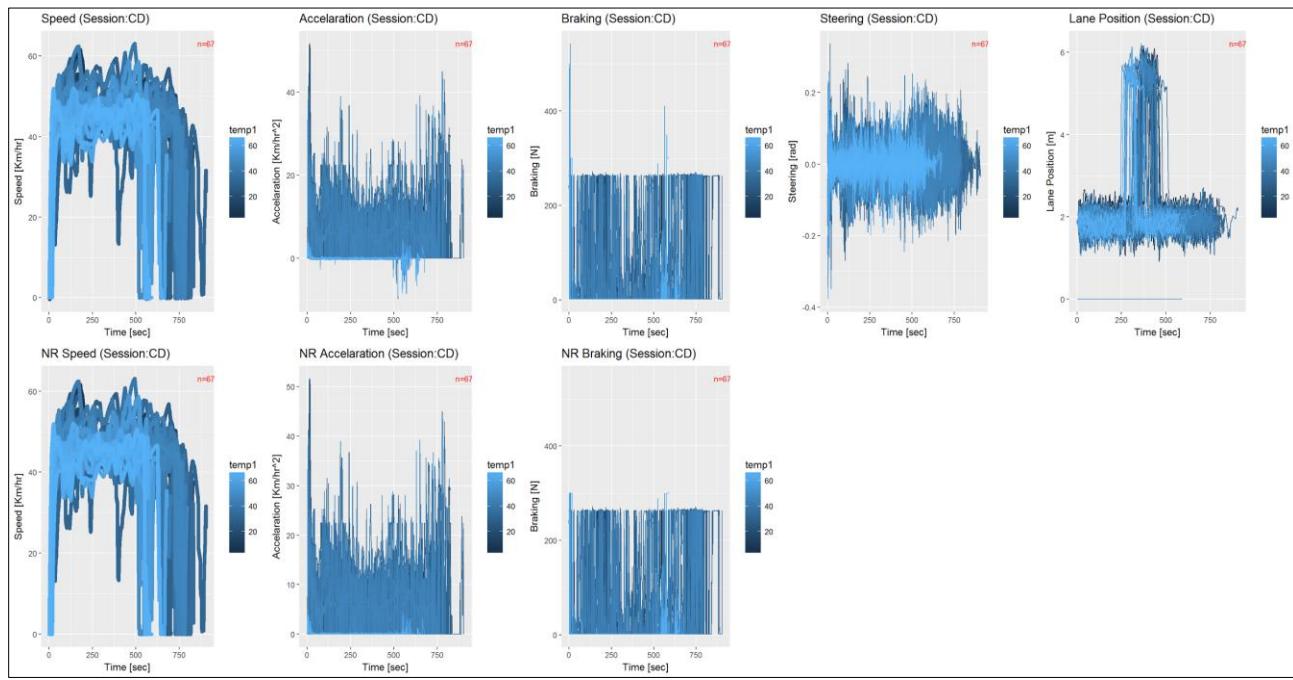
Assignment 1

Session: RDSession: FD

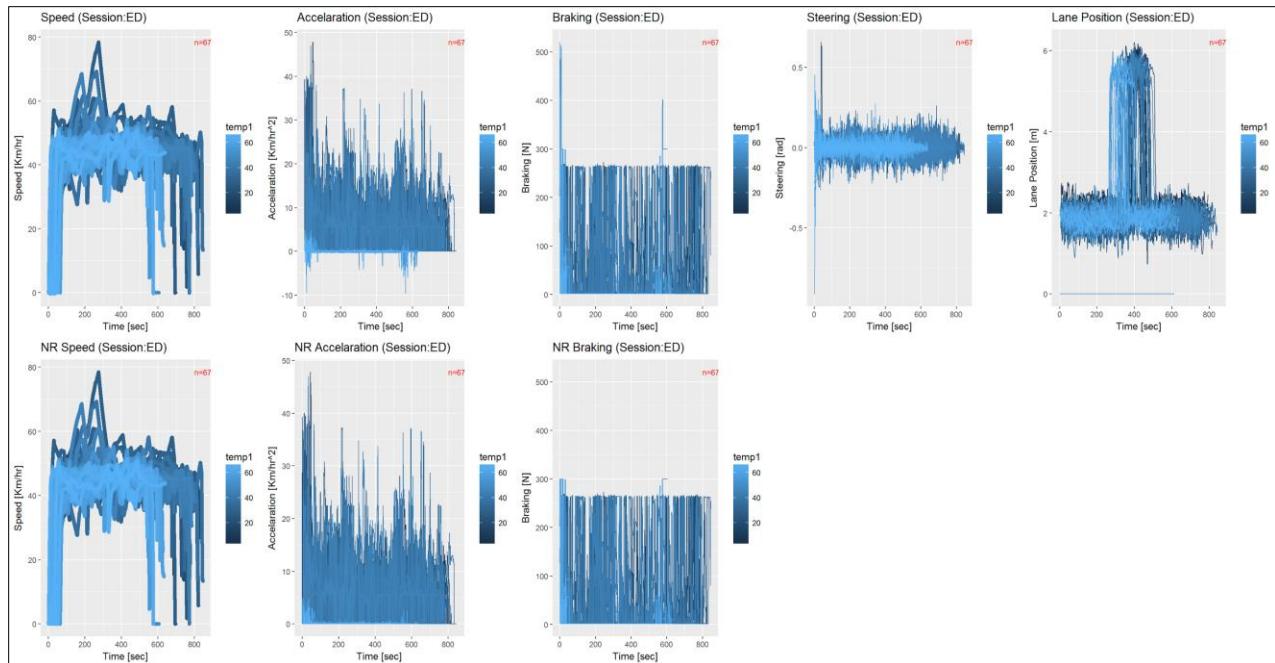
Assignment 1

Data Channel: Res

Session: CD

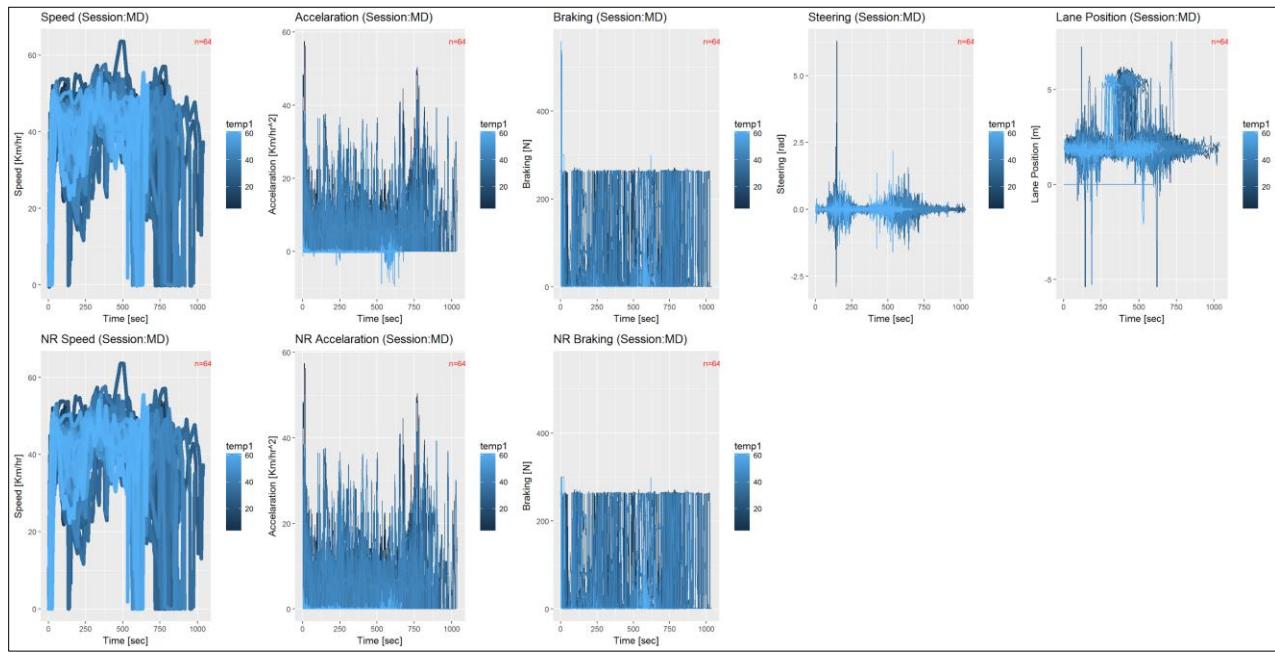


Session: ED

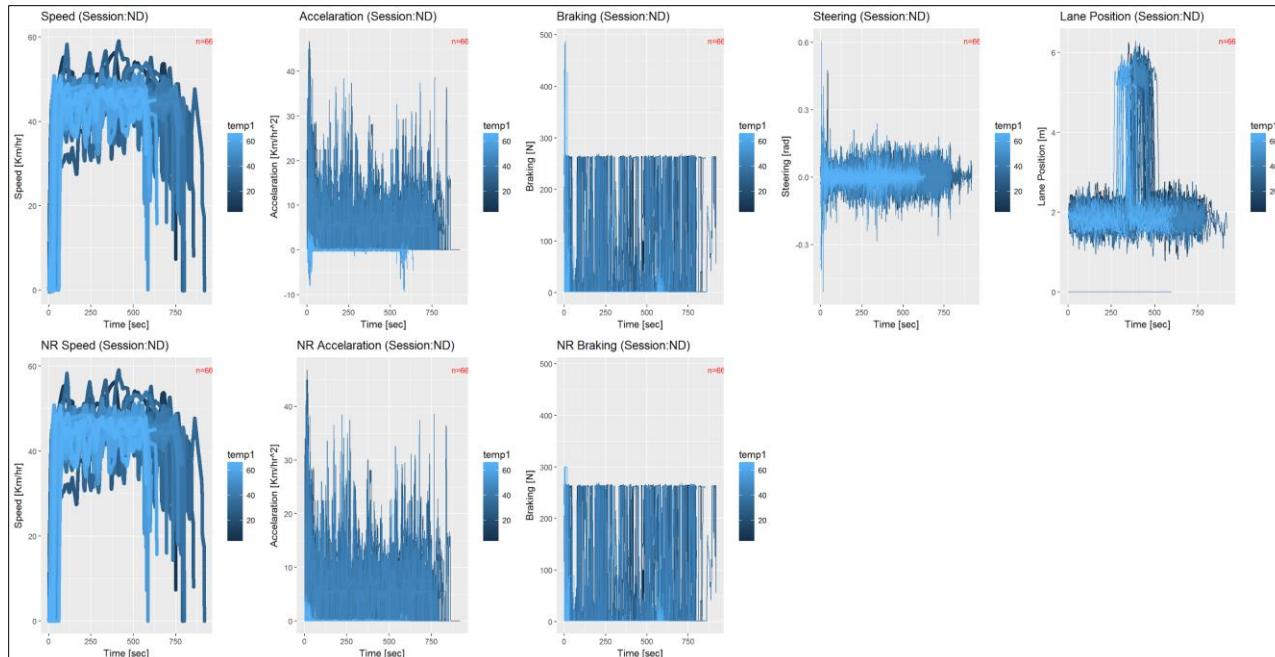


Assignment 1

Session: MD

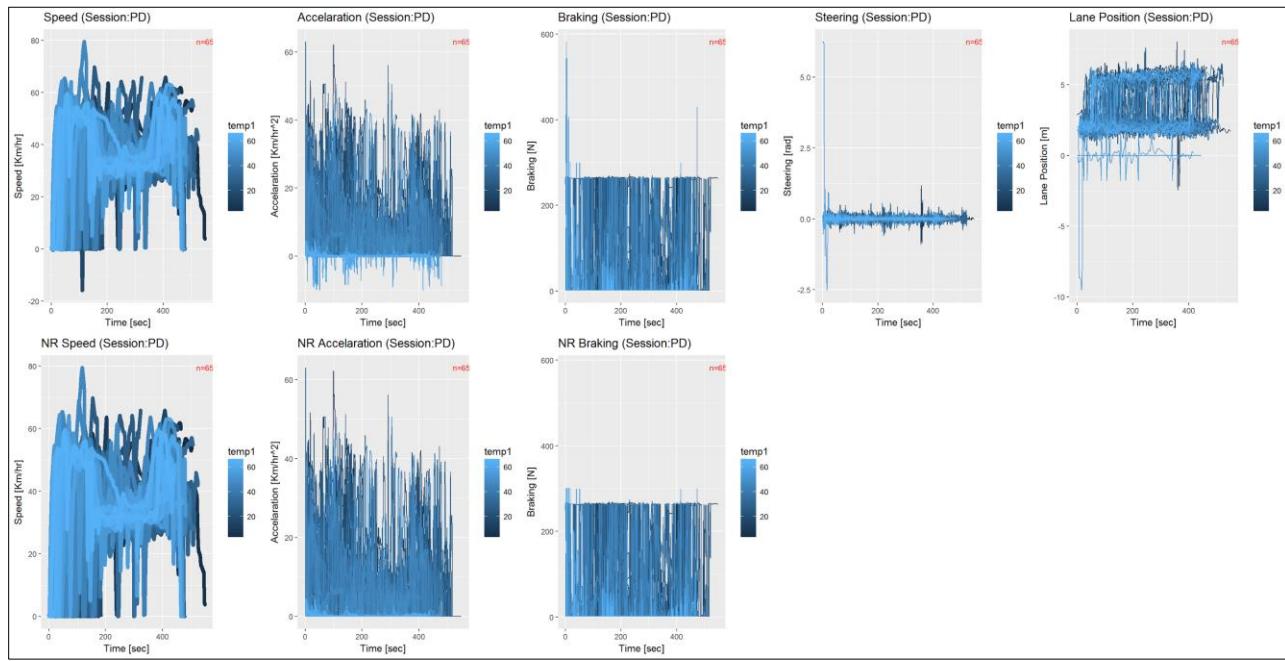


Session: ND

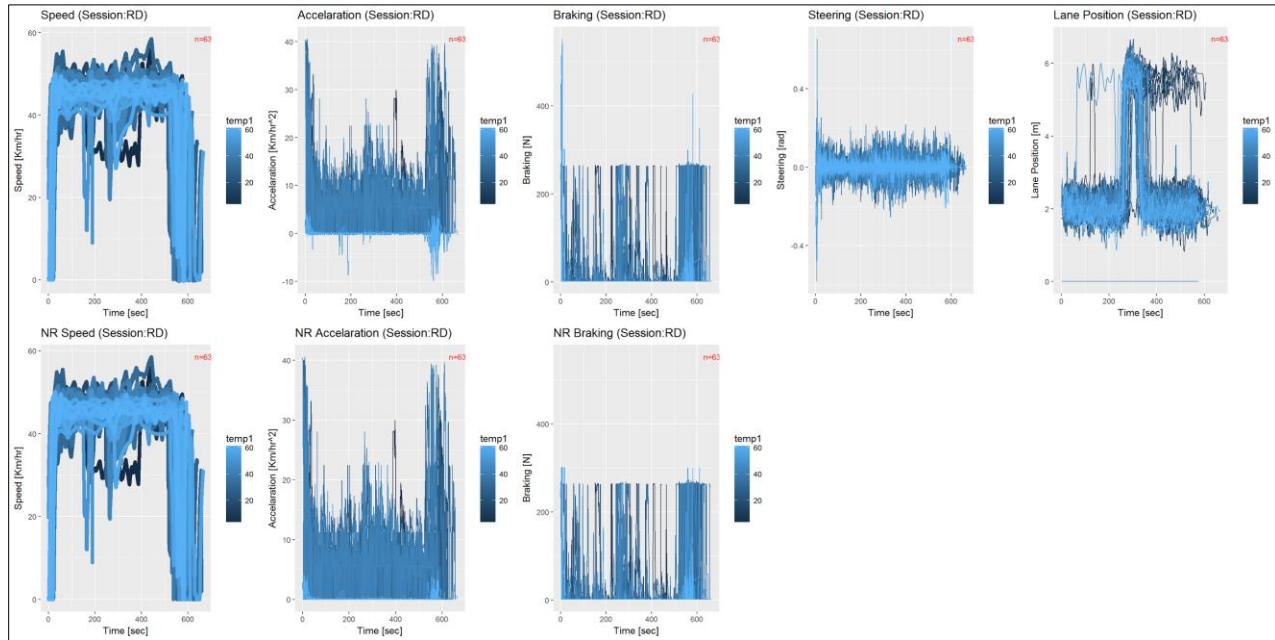


Assignment 1

Session: PD

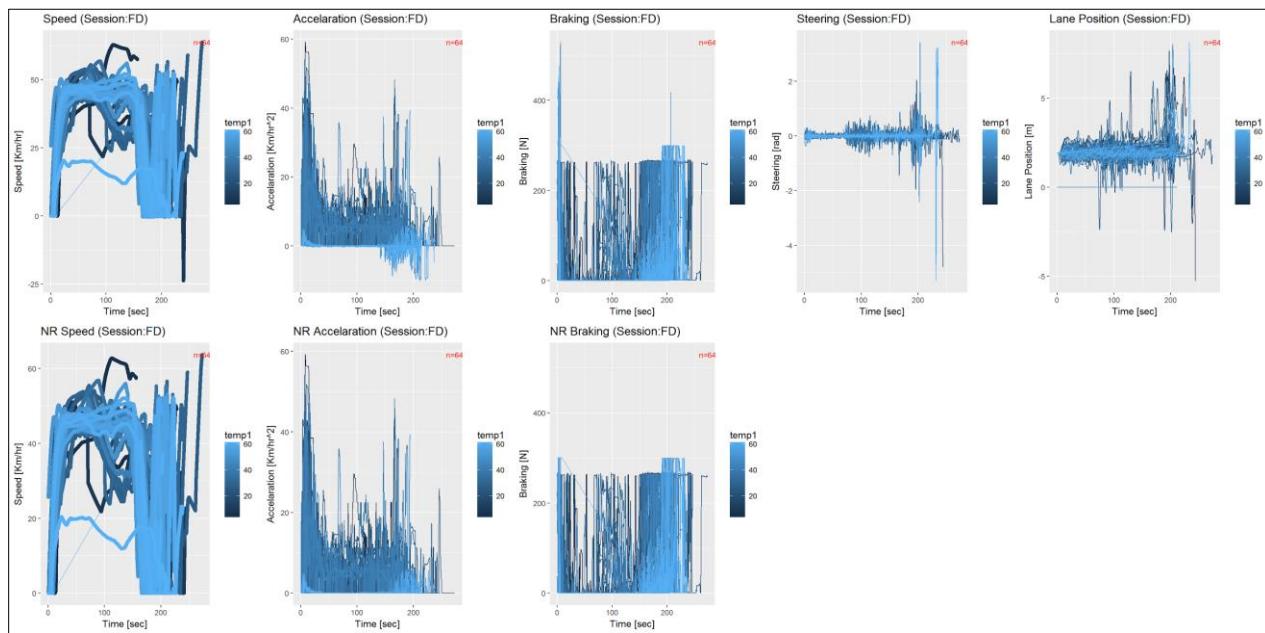


Session: RD



Assignment 1

Session: FD



6. Conclusion

1. During practical collection of data, we observe that many errors may arise. These errors may be due to following reasons:
 - a. Device malfunction
 - b. Errors due to improper positioning of the device
2. There are different ways to handle these errors:
 - a. To ignore the data outside the valid range.
 - b. To ignore the entire signal if sufficient data is available.

We have used the second method since we had sufficient data.
3. The entire process of cleaning the data can be divided into two steps:
 - a. Generation of updated Index files with 0 when the file is not present, 1 for clean data and -1 for files containing data not within the valid range. RES files with updated values are also generated in this phase.
 - b. Generation of output file graphs with both clean and raw data.

7. Appendix:

The code used for generation of index file and output files is as follows:

```
library(gdata)
library(ggplot2)
library(gridExtra)
library(grid)
```

Assignment 1

```
options("max.print" = 999999)
dataPath="C:/Users/nafis/Desktop/data"
workingDir="C:/Users/nafis/Desktop/Project"
setwd(workingDir)
outputFile = read.xls(paste(workingDir,"/Dataset-Table-
Index_output.xlsx",sep=""),header=TRUE, stringsAsFactors=FALSE)
setwd("output")
dataChannels=c("BR","HR","peda","pp","res")
sink("outputFinal.txt")
setwd(dataPath)

for(i in 1:nrow(outputFile))
{
  if(!substr(outputFile[i,1],0,1)=="T"){
    break
  }
  setwd(as.character(outputFile$Subject[i]))
  if(as.character(outputFile$Subject[i])==list.files())
    setwd(as.character(outputFile$Subject[i]))
  finalSession=list.files(pattern = paste(as.character(outputFile$Session[i])))
  if(length(finalSession)==0)
  {
    for(j in 1:length(dataChannels) )
    {
      if(!dataChannels[j]=="res")
      {
        if(!is.na(outputFile[dataChannels[j]][i,]))
        {
          outputFile[dataChannels[j]][i,]="0"
        }
      }
      else
      {
        if(!is.na(outputFile["performance..res. "][i,]))
        {
          outputFile["performance..res. "][i,]="0"
        }
      }
    }
  }
  else
  {
    setwd(as.character(finalSession))
  }
}
```

Assignment 1

```

for(j in 1:length(dataChannels) )
{
  if(!dataChannels[j]=="res")
  {
    if(!is.na(outputFile[dataChannels[j]][i,]))
    {
      fileFound = list.files(pattern = paste(as.character(dataChannels[j])))
      if(length(fileFound)==0)
        outputFile[dataChannels[j]][i,]="0"
      else{
        outputFile[dataChannels[j]][i,]=checkIfInRange(as.character(dataChannels[j]),fileFound )
      }
    }
  }
  else
  {
    if(!is.na(outputFile["performance..res. "][i,]))
    {
      fileFound = list.files(pattern = paste(as.character(dataChannels[j])))
      if(length(fileFound)==0)
        outputFile["performance..res. "][i,]="0"
      else
      {

outputFile["performance..res. "][i,]=checkIfInRange(as.character(dataChannels[j]),fileFound )
      }
    }
  }
}
setwd(dataPath)
}

setwd(workingDir)
setwd("output")
sink()
write.csv(outputFile, file = "UpdatedIndex.csv")

checkIfInRange <- function(FilenameFormat,FinalfileName) {
  FinalfileData=read.xls(FinalfileName, header=TRUE, stringsAsFactors=FALSE)
  if(FilenameFormat=="res"){
    if(ncol(FinalfileData)>8)
      FinalfileData=FinalfileData[, 1:8]
  }
}

```

Assignment 1

```
if(ncol(FinalfileData)==8)
  FinalfileData=FinalfileData[,-7]
  colnames(FinalfileData)=c("Frame","Time","Speed",
    "Acceleration","Braking","Steering","Lane Position")

}
FinalfileData = as.data.frame(na.omit(sapply(FinalfileData, as.numeric) )

switch(FilenameFormat,
  "BR"={
    errorResults=FinalfileData[FinalfileData[, "X.1"] < 4 | FinalfileData[, "X.1"] > 70,]
    if(nrow(errorResults)>0){
      print(getwd())
      print(FinalfileName)
      print(errorResults)
      return("-1")
    }
  },
  "HR"={
    errorResults=FinalfileData[FinalfileData[, "X.1"] < 40 | FinalfileData[, "X.1"] > 140,]
    if(nrow(errorResults)>0){
      print(getwd())
      print(FinalfileName)
      print(errorResults)
      return("-1")
    }
  },
  "peda"={
    errorResults=FinalfileData[FinalfileData[, "X.1"] < 10 | FinalfileData[, "X.1"] > 4700,]
    if(nrow(errorResults)>0){
      print(getwd())
      print(FinalfileName)
      print(errorResults)
      return("-1")
    }
  },
  "res"={
    errorResults=FinalfileData[FinalfileData[, "Acceleration"] < 0 | (FinalfileData[, "Speed"] >=
-0.1 & FinalfileData[, "Speed"] <= 0.1 ) | FinalfileData[, "Braking"] >300,]

    temp=FinalfileData[ "Acceleration"]
    if(nrow(FinalfileData[FinalfileData[, "Acceleration"] < 0,]["Acceleration"])>0){
```

Assignment 1

```

temp[temp["Acceleration"]<0]=NA
}
colnames(temp)="NR Acceleration"
FinalfileData=cbind(FinalfileData,temp)

temp=FinalfileData[ "Speed"]
if((nrow(FinalfileData[FinalfileData[, "Speed"] >= -0.1 & FinalfileData[, "Speed"] <= 0.1
,]["Speed"])>0)
    temp[temp[ "Speed"] >= -0.1 & temp[ "Speed"] <= 0.1 ,]=0

if(nrow(FinalfileData[FinalfileData[, "Speed"] < -0.1,]["Speed"])>0)
    temp[temp[ "Speed"] < -0.1,]=NA
colnames(temp)="NR Speed"
FinalfileData=cbind(FinalfileData,temp)
temp=FinalfileData[ "Braking"]
if(nrow(FinalfileData[FinalfileData[, "Braking"] >300,]["Braking"])>0)
    temp[temp["Braking"]>300,]=300
colnames(temp)="NR Braking"
FinalfileData=cbind(FinalfileData,temp)
write.csv(FinalfileData, file = FinalfileName, row.names=FALSE, sep=",")
if(nrow(errorResults)>0){
    setwd(workingDir)
    setwd("updatedRESFiles")
    write.csv(FinalfileData, file = FinalfileName, row.names=FALSE)
}
return("1")
}

)
return("1")
}

metaDataSet=outputFile
str=dataPath
folderNames = c("BL" , "CD" , "ED" , "MD" , "ND" , "PD" , "RD" , "FD")
setwd(dataPath)

for(chan in 1:length(dataChannels))
{
#chan=1
print(paste("chan: ",chan))
if(!dataChannels[chan]=="res")

```

Assignment 1

```

pdf( paste(workingDir,"/output/OUTPUT",dataChannels[chan],".pdf",sep=""),width = 10,height
= 10)
for(folder in folderNames)
{
  #folder="MD"
  if(!is.na(folder) & !folder=="")
  {
    sessionPlot<- getGGPlot(chan,folder)
    if(!dataChannels[chan]=="res" & !dataChannels[chan]=="pp"){
      if(!length(sessionPlot)==0)
        do.call(grid.arrange, c(sessionPlot, list(ncol=2)))
    }
    else if(dataChannels[chan]=="pp"){
      if(!length(sessionPlot)==0)
        do.call(grid.arrange, c(sessionPlot, list(ncol=1)))
    }
    else{
      if(!length(sessionPlot)==0){
        png(
          paste(workingDir,"/output/OUTPUTGraphs_",dataChannels[chan],"_",folder,".png",sep=""),
          width = 20,height = 10, units="in", res=200)

        do.call(grid.arrange, c(sessionPlot, list(ncol=5, nrow=2)))
        dev.off()
      }
    }
  }
  dev.off()
}

getGGPlot <- function(chan,folder) {
  sessionPlot=list()
  if(!is.na(folder) & !folder=="")
  {
    print(folder)
    title=""
    lab_y=""
    lab_x="Time [sec]"
    switch(as.character(dataChannels[chan]),
      "BR"={
        title="Breathing Rate"
      }
    )
  }
}

```

Assignment 1

```

lab_y="BR[bpm]"
},
"HR"={
  title="Heart Rate"
  lab_y="HR[bpm]"
},
"peda"={
  title="Palm EDA"
  lab_y="EDA[K Ohm]"
},
"res"={
  title="Accelaration Signals"
},
"pp"={
  title="Perinasal Perspiration"
  lab_y="PP[*Celcius^2]"
},
{
  #print('default')
}
)
numOfRows=0;

#code for Raw Data
metaDataSetrelVal=metaDataSet[is.na(metaDataSet$dataChannels$chan)]==FALSE &
!metaDataSet[dataChannels$chan]==0 ,]
metaDataSetrelVal=metaDataSetrelVal[grep(as.character(folder),metaDataSetrelVal$Session),]
}else{
  metaDataSetrelVal=metaDataSet[is.na(metaDataSet["performance..res."])]==FALSE &
!metaDataSet["performance..res."]==0 ,]
metaDataSetrelVal=metaDataSetrelVal[grep(as.character(folder),metaDataSetrelVal$Session),]
}

numOfRows=length(unique(metaDataSetrelVal$Subject))
if(!nrow(metaDataSetrelVal)==0)
{
  drawPath=cbind(rep(str,each=nrow(metaDataSetrelVal)),'/',metaDataSetrelVal$Subject,'/')
  colnames(drawPath) <- c("ParentDir","FstSlash","Subject","SecSlash")

drawPath=paste(drawPath$ParentDir,drawPath$FstSlash,drawPath$Subject,drawPath$SecSlash,sep = "")
for(i in 1:length(drawPath))
{
  print(drawPath[i])
}

```

```

setwd(drawPath[i])
tempArr = strsplit(drawPath[i], "/")
folderT = lapply(tempArr, tail, 1)
if(folderT == list.files())
  setwd(as.character(folderT))
FinalFolder=list.files(pattern = paste(as.character(folder)))
setwd(as.character(FinalFolder))
FinalfileName=list.files(pattern = paste(dataChannels[chan]))
print(FinalfileName)
if(!length(FinalfileName)==0)
{
  if(!dataChannels[chan]=="res")
    FinalfileData=read.xls(FinalfileName, stringsAsFactors=FALSE)
  else
    FinalfileData=read.csv(FinalfileName, stringsAsFactors=FALSE)
  FinalfileData = sapply(FinalfileData, as.numeric)
  FinalfileData = data.frame(FinalfileData)
  FinalfileData=FinalfileData[colSums(!is.na(FinalfileData)) > 0]
  if(as.character(dataChannels[chan])=="pp"){
    FinalfileData=FinalfileData[, 1:4]
    colnames(FinalfileData) <- c("Perinasal.Perspiration", "X", "X.1", "X.2")
  }
}

temp1 = rep(as.character(folderT), times = nrow(FinalfileData))
FinalfileData=cbind(FinalfileData, temp1 )
if(i==1)
{
  combinedData = FinalfileData
}
if(i!=1)
{
  combinedData = rbind(combinedData,FinalfileData)
}
}
combinedData = sapply(combinedData, as.numeric)
combinedData=data.frame(combinedData)
if(as.character(dataChannels[chan])=="res" ||
as.character(dataChannels[chan])=="performance..res ."){
  #cleanData=combinedData[combinedData[, "Acceleration"] > 0,]
  p1 <- ggplot(combinedData, aes((combinedData)[ "Time"], (combinedData)[ "Speed"],
group=temp1, color=temp1)) + geom_line() + geom_point() + xlab("lab_x") + ylab("Speed [Km/hr]") +
  ggtitle(paste("Speed (Session:", folder, ")", sep = "")) + annotate("text", label =

```

Assignment 1

```

paste("n=",numOfRows,sep=""), x = max(combinedData[2][!is.na(combinedData[2]),]) , y =
max(combinedData[3][!is.na(combinedData[3]),]) , size = 3, colour = "red")
sessionPlot[[length(sessionPlot)+1]] <- p1
p1 <- ggplot(combinedData, aes((combinedData)[ "Time"], (combinedData)[ "Acceleration"],
group=temp1, color=temp1)) + geom_line() + xlab(lab_x) + ylab("Accelaration [Km/hr^2]") +
ggtitle(paste("Accelaration (Session:", folder, ")", sep = "")) + annotate("text", label =
paste("n=", numOfRows, sep = ""), x = max(combinedData[2][!is.na(combinedData[2]),]) , y =
max(combinedData[4][!is.na(combinedData[4]),]) , size = 3, colour = "red")
sessionPlot[[length(sessionPlot)+1]] <- p1
p1 <- ggplot(combinedData, aes((combinedData)[ "Time"], (combinedData)[ "Braking"],
group=temp1, color=temp1)) + geom_line() + xlab(lab_x) + ylab("Braking [N]") +
ggtitle(paste("Braking (Session:", folder, ")", sep = "")) + annotate("text", label =
paste("n=", numOfRows, sep = ""), x = max(combinedData[2][!is.na(combinedData[2]),]) , y =
max(combinedData[5][!is.na(combinedData[5]),]) , size = 3, colour = "red")
sessionPlot[[length(sessionPlot)+1]] <- p1
p1 <- ggplot(combinedData, aes((combinedData)[ "Time"], (combinedData)[ "Steering"],
group=temp1, color=temp1)) + geom_line() + xlab(lab_x) + ylab("Steering [rad]") +
ggtitle(paste("Steering (Session:", folder, ")", sep = "")) + annotate("text", label =
paste("n=", numOfRows, sep = ""), x = max(combinedData[2][!is.na(combinedData[2]),]) , y =
max(combinedData[6][!is.na(combinedData[6]),]) , size = 3, colour = "red")
sessionPlot[[length(sessionPlot)+1]] <- p1
p1 <- ggplot(combinedData, aes((combinedData)[ "Time"],
(combinedData)[ "Lane.Position"], group=temp1, color=temp1)) +
geom_line() + xlab(lab_x) + ylab("Lane Position [m]") + ggtitle(paste("Lane Position
(Session:", folder, ")", sep = "")) + annotate("text", label = paste("n=", numOfRows, sep = ""),
x = max(combinedData[2][!is.na(combinedData[2]),]) , y =
max(combinedData[7][!is.na(combinedData[7]),]) , size = 3, colour = "red")
sessionPlot[[length(sessionPlot)+1]] <- p1

p1 <- ggplot(combinedData, aes((combinedData)[ "Time"], (combinedData)[ "NR.Speed"],
group=temp1, color=temp1)) + geom_line() + geom_point() + xlab(lab_x) + ylab("Speed [Km/hr]") +
ggtitle(paste("NR Speed (Session:", folder, ")", sep = "")) + annotate("text", label =
paste("n=", numOfRows, sep = ""), x = max(combinedData[2][!is.na(combinedData[2]),]) , y =
max(combinedData[3][!is.na(combinedData[3]),]) , size = 3, colour = "red")
sessionPlot[[length(sessionPlot)+1]] <- p1
p1 <- ggplot(combinedData, aes((combinedData)[ "Time"],
(combinedData)[ "NR.Accelaration"], group=temp1, color=temp1)) +
geom_line() + xlab(lab_x) + ylab("Accelaration [Km/hr^2]") + ggtitle(paste("NR Accelaration
(Session:", folder, ")", sep = "")) + annotate("text", label = paste("n=", numOfRows, sep = ""),
x = max(combinedData[2][!is.na(combinedData[2]),]) , y =
max(combinedData[4][!is.na(combinedData[4]),]) , size = 3, colour = "red")
sessionPlot[[length(sessionPlot)+1]] <- p1
p1 <- ggplot(combinedData, aes((combinedData)[ "Time"], (combinedData)[ "NR.Braking"],
group=temp1, color=temp1)) + geom_line() + xlab(lab_x) + ylab("Braking [N]") + ggtitle(paste("NR

```

Assignment 1

```

Braking (Session:",folder,"),sep="))+annotate("text", label = paste("n=",numOfRows,sep=""), x
= max(combinedData[2][!is.na(combinedData[2])],) , y =
max(combinedData[5][!is.na(combinedData[5])],) , size = 3, colour = "red")
sessionPlot[[length(sessionPlot)+1]] <- p1

}

else if(as.character(dataChannels[chan])=="pp"){

  p1 <- ggplot(combinedData, aes((combinedData)[2], (combinedData)[4], group=temp1,
color=temp1)) + geom_line() + xlab(lab_x)+ylab(lab_y)+ ggtitle(paste(title, " Clean Signal
(Session:",folder,"),sep="))+annotate("text", label = paste("n=",numOfRows,sep=""), x =
max(combinedData[2][!is.na(combinedData[2])],) , y =
max(combinedData[4][!is.na(combinedData[4])],) , size = 3, colour = "red")
sessionPlot[[length(sessionPlot)+1]] <- p1

}

else{

  p1 <- ggplot(combinedData, aes((combinedData)[2], (combinedData)[3], group=temp1,
color=temp1)) + geom_line() + xlab(lab_x)+ylab(lab_y)+ ggtitle(paste(title, " Raw Signal
(Session:",folder,"),sep="))+annotate("text", label = paste("n=",numOfRows,sep=""), x =
max(combinedData[2][!is.na(combinedData[2])],) , y =
max(combinedData[3][!is.na(combinedData[3])],) , size = 3, colour = "red")
sessionPlot[[length(sessionPlot)+1]] <- p1

}

setwd("C:/Users/nafis/Desktop/ANA Files/output")
if(!as.character(dataChannels[chan])=="res" &
!as.character(dataChannels[chan])=="performance..res." &
!as.character(dataChannels[chan])=="pp"){

# find signals which have all values in range
metaDataSetrelVal2=metaDataSet[is.na(metaDataSet[dataChannels[chan]])==FALSE &
metaDataSet[dataChannels[chan]]==1 ,]

metaDataSetrelVal2=metaDataSetrelVal2[grep(as.character(folder),metaDataSetrelVal2$Session),]

metaDataSetrelVal2=metaDataSetrelVal2[unique(metaDataSetrelVal2$Subject)]
if(!nrow(metaDataSetrelVal2)==0)
{
drawPath=cbind(rep(str,each=nrow(metaDataSetrelVal2)),'/',metaDataSetrelVal2["Subject"],'/')
colnames(drawPath) <- c("ParentDir","FstSlash","Subject","SecSlash")
drawPath=paste(drawPath$ParentDir,drawPath$FstSlash,drawPath$Subject,drawPath$SecSlash,sep = "")

for(i in 1:length(drawPath))
{
  print(drawPath[i])
}
}
}

```

Assignment 1

```
setwd(drawPath[i])
tempArr = strsplit(drawPath[i], "/")
folderT = lapply(tempArr, tail, 1)
if(folderT == list.files())
  setwd(as.character(folderT))
FinalFolder=list.files(pattern = paste(as.character(folder)))
setwd(as.character(FinalFolder))
FinalfileName=list.files(pattern = paste(dataChannels[chan]))
print(FinalfileName)
if(!length(FinalfileName)==0)
{
  FinalfileData=read.xls(FinalfileName, stringsAsFactors=FALSE)
  FinalFileHeaders=FinalfileData[1,]
  FinalfileData=FinalfileData[c(2:nrow(FinalfileData)),]
  FinalfileData = sapply(FinalfileData, as.numeric)
  FinalfileData = data.frame(FinalfileData)
  FinalfileData=FinalfileData[colSums(!is.na(FinalfileData)) > 0]
  temp1 = rep(as.character(folderT), times = nrow(FinalfileData))
  FinalfileData=cbind(FinalfileData, temp1 )
  if(i==1)
  {
    cleanData = FinalfileData
  }
  if(i!=1)
  {
    cleanData = rbind(cleanData,FinalfileData)
  }
}
cleanData = sapply(cleanData, as.numeric)
cleanData=data.frame(cleanData)

p1 <- ggplot(cleanData, aes((cleanData)[2], (cleanData)[3], group=temp1, color=temp1))
+ geom_line() + xlab(lab_x) + ylab(lab_y) + ggtitle(paste(title, " Clean Signal
(Session:", folder, ")", sep=""))
+ annotate("text", label = paste("n=", numRows, sep=""), x =
max(cleanData[2][!is.na(cleanData[2]),]) , y = max(cleanData[3][!is.na(cleanData[3]),]) , size = 3,
colour = "red")
  sessionPlot[[length(sessionPlot)+1]] <- p1
}
}
}
}
return(sessionPlot)
}
```