In [ ]:

```
#data wrangling
#prop3 from biolinc
prop3 <-  readRDS("...")
#there are 835 patients
#first step is to make all of the weeks subsetted from Week1 all the way to ~ We
ek17
w1 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 1")
w2 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 2")
w3 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 3")
w4 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 4")
w5 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 5")
w6 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 6")
w7 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 7")
w8 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 8")
w9 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 9")
w10 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 10")
w11 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 11")
w12 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 12")
w13 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 13")
w14 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 14")
w15 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 15")
w16 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 16")
w17 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 17")
#...do up to w17
w17 <- subset(prop3, prop3$VISIT_TYPE == "WEEK 17")

# combine the prop4 data by week
#... bring the splitto the highest week

w1_split <- split(w1, w1$NEWPT)
w2_split <- split(w2, w2$NEWPT)

w1_split <- split(w1, w1$NEWPT)
w2_split <- split(w2, w2$NEWPT)
w3_split <- split(w3, w3$NEWPT)
w4_split <- split(w4, w4$NEWPT)
w5_split <- split(w5, w5$NEWPT)
w6_split <- split(w6, w6$NEWPT)
w7_split <- split(w7, w7$NEWPT)
w8_split <- split(w8, w8$NEWPT)
w9_split <- split(w9, w9$NEWPT)
w10_split <- split(w10, w10$NEWPT)
w11_split <- split(w11, w11$NEWPT)
w12_split <- split(w12, w12$NEWPT)
w13_split <- split(w13, w13$NEWPT)
w14_split <- split(w14, w14$NEWPT)
```

```
w15_split <- split(w15, w15$NEWPT)
w16_split <- split(w16, w16$NEWPT)
w17_split <- split(w17, w17$NEWPT)
#..do up to w17_split
w17_split <- split(w17, w17$NEWPT)



get_number <- function(x)  {
  as.numeric(gsub("[^[:digit:].]", "",  as.character(x)))
}


func_pnd <- function(x)  {
  #x is going to be a data frame
  x$day_of_week <- 1:nrow(x)
  x$week_number <- get_number(x$VISIT_TYPE)
  x$pnd <- x$day_of_week + x$week_number*7-7
  return(x)
}


#### now we have a way to do that week by week,
#next step is how to rebuild the list
w1_s2 <- lapply(w1_split, func_pnd)
w2_s2 <- lapply(w2_split, func_pnd)
w17_s2 <- lapply(w17_split, func_pnd)

#next step is how to rebuild the list


w1_s2 <- lapply(w1_split, func_pnd)
w2_s2 <- lapply(w2_split, func_pnd)

w3_s2 <- lapply(w3_split, func_pnd)

w4_s2 <- lapply(w4_split, func_pnd)

w5_s2 <- lapply(w5_split, func_pnd)
w6_s2 <- lapply(w6_split, func_pnd)
w7_s2 <- lapply(w7_split, func_pnd)
w8_s2 <- lapply(w8_split, func_pnd)
w9_s2 <- lapply(w9_split, func_pnd)
w10_s2 <- lapply(w10_split, func_pnd)
w11_s2 <- lapply(w11_split, func_pnd)
w12_s2 <- lapply(w12_split, func_pnd)
w13_s2 <- lapply(w13_split, func_pnd)
w14_s2 <- lapply(w14_split, func_pnd)
w15_s2 <- lapply(w15_split, func_pnd)
w16_s2 <- lapply(w16_split, func_pnd)
```

```
w17_s2 <- lapply(w17_split, func_pnd)


s <- names(w1_s2)
my_names <- names(w1_s2)
#prop1
prop1 <-readRDS("...")
prop1$NEWPT <- as.factor(prop1$NEWPT)
prop1_s <- split(prop1, prop1$NEWPT)
names(prop1_s) ->b
names(prop1_s) <- paste("pt", b, sep = "_")
setwd("...")
for(i in 1:length(my_names))  {
  da_name <- my_names[i]
  df1 <- as.data.frame(w1_s2[[da_name]])
  df2 <- as.data.frame(w2_s2[[da_name]])
  df3 <- as.data.frame(w3_s2[[da_name]])
  df4 <- as.data.frame(w4_s2[[da_name]])
  df5 <- as.data.frame(w5_s2[[da_name]])
  df6 <- as.data.frame(w6_s2[[da_name]])
  df7 <- as.data.frame(w7_s2[[da_name]])
  df8 <- as.data.frame(w8_s2[[da_name]])
  df9 <- as.data.frame(w9_s2[[da_name]])
  df10 <- as.data.frame(w10_s2[[da_name]])
  df11 <- as.data.frame(w11_s2[[da_name]])
  df12 <- as.data.frame(w12_s2[[da_name]])
  df13 <- as.data.frame(w13_s2[[da_name]])
  df14 <- as.data.frame(w14_s2[[da_name]])
  df15 <- as.data.frame(w15_s2[[da_name]])
  df16 <- as.data.frame(w16_s2[[da_name]])
  df17 <- as.data.frame(w17_s2[[da_name]])
  my_data <- rbind(df1, df2, df3, df4, df5, df6,
                   df7, df8, df9, df10, df11,
                   df12, df13, df14, df15, df16, df17)
  print('my_data_done')
  weight= prop1_s[[da_name]]$BBAS_WEIGHT[1]
  sex= prop1_s[[da_name]]$BBAS_SEX[1]
  my_data$birth_weight= rep(weight,times=nrow(my_data))
  my_data$birth_sex= rep(sex,times=nrow(my_data))
  saveRDS(my_data, file = paste("pt", my_names[i], ".R", sep = ""))
  print(i)
}
dim(my_data)
setwd("...")
#this is where I made all the files together.
my_files <- list.files()
final_df <- data.frame()
for(i in 1:length(my_files)){
  new_df <- readRDS(my_files[i])
  final_df <- rbind(final_df, new_df)
  print(i)
}
dim(final_df)
names(final_df)
plot(final_df$GNMD_WEIGHT, y = final_df$pnd, col= final_df$NEWPT)

saveRDS(final_df, file = "final_df.R")
final_df$NEWPT <- as.factor(final_df$NEWPT)
final_df$age_in_days_from_birth <- final_df$day_of_week*7 + final_df$day_of_week
saveRDS(final_df, file = "final_df_age_in_days.R")
final_split <- split(final_df, final_df$NEWPT)
#this is the procedure for linking the data from the prior dataframe that has
#one patient per line to this final_df which has ~66 lines per patient
#because of the daily wait log.



saveRDS(final_split, file = "final_split.R")

final_split <- readRDS("...")
info <- readRDS("...")
names(final_split)
dim(info)

info$Patient <- as.factor(info$Patient)
levels(info$Patient)
bb <- final_split$`1`

info_vector <- function(x){
  rep(x, times = nrow(a))

}
cc <- as.data.frame(apply(bb, 2, info_vector))

######now we will add all of the things here for each member of the list.

#split info into patients.
info_split <- split(info, info$Patient)
names(info_split)
names(final_split)

setwd("...")
for(i in 1:length(info_split)){
  #  for(i in 1:3){
  df_info <- info_split[[i]]
  df_final <- final_split[[i]]
  print(i)

  info_vector <- function(x){
    rep(x, times = nrow(df_final))
  }
  cc <- as.data.frame(apply(df_info, 2, info_vector))
  dd <- cbind(df_final, cc)
  saveRDS(dd, file = paste("pt_", names(final_split)[i], ".R", sep = ""))
  print(i)
}

list.files()

test <- readRDS(
  "pt_1.R"
)
test$pnd
dim(test)
test$NEWPT
dim(final_split$`1`)


setwd("...")
my_files <- list.files()
df_moving <- data.frame()
for(i in 1:length(my_files))  {
  new_df <- readRDS(my_files[i])
  df_moving <- rbind(df_moving, new_df)
  rm(new_df)
  print(i)
}
#saveRDS(df_moving, file = "df_moving.R")
df_moving <- readRDS("...")
dim(df_moving)
dim(train_df)
dim(df_moving)
names(df_moving)
plot(as.numeric(as.character(df_moving$NEWPT)), as.numeric(as.character(df_movin
g$Patient)))
summary(df$NEWPT)

all.equal(as.numeric(as.character(df_moving$NEWPT)), as.numeric(as.character(df_
moving$Patient)))

df_moving$Num_of_completed_weeks[1:10]
tail(df_moving$gestational_age)
df_moving$corrected_pma <- as.numeric(as.character(df_moving$Num_of_completed_we
eks)) +
  as.numeric(as.character(df_moving$Num_of_completed_days))/7 + df_moving$pnd/7

pdf(file = "test.pdf")
plot(y = df_moving$GNMD_WEIGHT, x = df_moving$corrected_pma, col = df_moving$Pat
ient)
dev.off()
```

```
make_num <- function(x) {
  new_var <- as.numeric(as.character(x))
  return(new_var)
}
df_moving$Diag_pda
info$ROP
df_moving$rac_1st_wt <- make_num(df_moving$rac_1st_wt)
df_moving$birth_weight <- make_num(df_moving$birth_weight)
df_moving$Head_circumference <- make_num(df_moving$Head_circumference)
df_moving$GNMD_CMMULTIV04
df_moving$Head_circumference
df_moving$Sex
class(df_moving$Sex)
summary(df_moving$Sex)

####df_moving is now used to generate some data for the figures.

info$real_gestational_age <- info$Num_of_completed_weeks + info$Num_of_completed_days/7
summary(info)
my_col <- c("Patient", "real_gestational_age", "sepsis_presumed",
            "sepsis_culture_proven", "ROP", "Clinical_Chorioamnionitis",
            "Mother_Diabetes", "Apgar_5min", "Apgar_1min",
            "birth_weight", "Sex", "Head_circumference",
            "pulm_hem",  "Diag_pda",
            "Pulmonary_Hypertension")
summary(info$rac1_baseline_rr)

library(caret)

data <- info[, my_col]
dim(na.omit(data))

data <- subset(data, data$Sex > 1)
data$Sex <- as.factor(data$Sex  )
?dummyVars
sex_vars <- dummyVars(~ Sex, data = data)
data2 <- data
names(data2)
data2$Sex <- NULL
sex_p <- predict(sex_vars, data)
data3 <- cbind(data2, sex_p)
head(data3)
head(data2)
data2$Patient

dim(na.omit(data2[,2:length(names(data2))]))
pca <- prcomp(na.omit(data2[,2:length(names(data2))]), scale = TRUE)
summary(pca)
my_cor <- cor(na.omit(data2[,2:length(names(data2))]))
```

```r
dim(info)
library(corrplot)
pdf(file = "corrplot.pdf")
corrplot(my_cor)
dev.off()
names(data2[,2:12])
d_o <- na.omit(data2[,2:length(names(data2))])
d_o1 <- na.omit(data2)
dim(d_o)
dim(d_o1)
d_o$patient <- d_o1$Patient
names(d_o)
summary(pca)
my_clustersx <- kmeans(pca$x, centers = 2)
d_o$cluster <- my_clustersx$cluster %>% as.factor
install.packages("plot3D")
library("plot3D")


#FIGURE 1B IN MANUSCRIPT
pdf(file = "optimal clusters by silhuette of patients.pdf")

optimal_kmeans_silhuette <- function(x) {

  library(tidyverse)
  library(factoextra)
  library(cluster)

  fviz_nbclust(as.matrix(x), kmeans, method = "silhouette")

}
optimal_kmeans_silhuette(pca$x)
dev.off()
getwd()
#FIGURE 1 A IN THE MANUSCRIPT
library(factoextra)
pdf(file = "eigenvector plot of patient data2.pdf")
fviz_pca_var(pca,
             col.var = "contrib", # Color by contributions to the PC
             gradient.cols = c("#00AFBB", "#E7B800", "#FC4E07"),
             repel = TRUE     # Avoid text overlapping
)
dev.off()


#FIGURE 1C

my_dist <- dist(pca$x)
summary(my_dist)
my_stats <- cluster.stats(my_dist, as.numeric(d_o$cluster))
```

```
summary(d_o$cluster)
my_stats$average.between

my_random <- sample(c(1,2), size = length(d_o$cluster),
                    prob = c(433, 384), replace = TRUE)
summary(as.factor(my_random))
#THE RANDOM_STATS IS A RANDOM SAMPLE TO EVALUATE THE DISTANCE OF CLUSTERS BASED ON RANDOM DISTRIBUTION
random_stats <- list()
for(i in 1:1000)  {
  my_random <- sample(c(1,2), size = length(d_o$cluster),
                      prob = c(433, 384), replace = TRUE)
  new_stats <- cluster.stats(my_dist, my_random)
  random_stats[[i]] <- new_stats
  print(i)
}
my_stats$entropy
random_stats[[5]]$average.within
my_dunn <- function(x)  {
  my_var <- x$average.within
}
saveRDS(random_stats, file = "random_cluster_stats.R")
saveRDS(my_stats, file = "my_stats_stats.R")
my_stats$within.cluster.ss
pdf(file = "clusterstatsbarplot.pdf")
barplot(c(my_stats$average.within, my_stats$average.between, my_stats$dunn2),
        names.arg=c("Ave. w/in Cluster Dist.","Ave btw, Cluster Dist.",
                    "Dunn2"),
        horiz = TRUE, las = 1, col = "snow2",
        main = "Cluster Statistics of PROP Dataset",
        font.lab = 2, font.axis = 2)
dev.off()


#FIGURE 1D

library(Boruta)
str(d_o)
df4 <- d_o
str(df4)
df4 <- as.data.frame(apply(df4, 2, as.factor))
df4$real_gestational_age <- as.numeric(as.character(df4$real_gestational_age))
df4$birth_weight <- as.numeric(as.character(df4$birth_weight))
df4$Head_circumference <- as.numeric(as.character(df4$Head_circumference))
df4$sepsis_presumed <- as.factor(df4$sepsis_presumed)
df4$sepsis_culture_proven <- as.factor(df4$sepsis_culture_proven)
df4$ROP <- as.factor(df4$ROP)
df4$Clinical_Chorioamnionitis <- as.factor(df4$Clinical_Chorioamnionitis)
df4$Mother_Diabetes<- as.factor(df4$Mother_Diabetes)
df4$Apgar_1min <- as.factor(df4$Apgar_1min)
```

```
df4$Apgar_5min <- as.factor(df4$Apgar_5min)
df4$pulm_hem <- as.factor(df4$pulm_hem)
df4$cluster <- as.factor(df4$cluster)
df4$Diag_pda <- as.factor(df4$Diag_pda)
df4$Pulmonary_Hypertension <- as.factor(df4$Pulmonary_Hypertension)
str(df4)
my_boruta <- Boruta(cluster~., data = df4)
my_rows <- sample(1:nrow(df4), size = 0.7*nrow(df4))
names(df4)
train = df4[ my_rows,]
test = df4[-my_rows,]
rf <- randomForest(cluster~., data = train)
rf_p <- predict(rf, test)
setwd("...")
sink(file = "rf_p of clusters in prop.txt")
rf
confusionMatrix(rf_p, test$cluster)
sink()
saveRDS(pca, file = "prop_pca_limieted.R")
rf$importance[,order(1)]
rf$importance[order(rf$importance[,1],decreasing=FALSE),]

rfb <- Boruta(cluster~., data = train)
pdf(file = "rfb_boruta.pdf")
plot(rfb, las = 2, font.axis = 2, font.lab  = 2)
dev.off()
#rf test set accuracy of  Accuracy : 0.9106
pdf(file = "barplot_RFMODEL.pdf")
barplot(rf$importance[order(rf$importance[,1],decreasing=FALSE),], col = "green4
",
        horiz = FALSE, font.axis = 2, cex.axis = 2,
        las = 2)
dev.off()





# FIGURE 2B IN THE MANUSCRIPT
pdf(file = "pca_clusters.pdf")
plot(pca$x[,1], pca$x[,2],
     col = ifelse(d_o$cluster == 1, "orange4", "blue4" ),
     pch = ifelse(d_o$sepsis_presumed == 1, 1, 16 ), cex = {2.5+scale(d_o$real_g
estational_age)},
     las = 1, xlab = "PC1 (20.96%)", ylab = "PC 2 (10.01%)", ylim = c(-5, 5),
     font.axis = 2, font.lab = 2, cex.axis = 1.8, cex.lab  = 1.5,
     main = "col = cluster, size = weight, aes = sepsis_presumed")
dev.off()
#FIGURE 2B
pdf(file = "pca_clusterstesxt.pdf")
```

```
#min(scale(d_o$real_gestational_age))
plot(pca$x[,1], pca$x[,2],
     col = ifelse(d_o$cluster == 1, "orange4", "blue4" ),
     pch = ifelse(d_o$sepsis_presumed == 1, 1, 16 ), cex = {2.5 + scale(d_o$real_gestational_age)},
     las = 1, xlab = "PC1 (20.96%)", ylab = "PC 2 (10.01%)", ylim = c(-5, 5),
     font.axis = 2, font.lab = 2, cex.axis = 1.8, cex.lab  = 1.5,
     main = "col = cluster, size = weight, aes = sepsis_presumed")
text(x = pca$x[,1], y = pca$x[,2], labels = round(d_o$real_gestational_age, digits = 1), cex = 0.5)
dev.off()
max(d_o$real_gestational_age)

#FIGURE 2 A
pdf(file = "pca_clustersbirthweight.pdf")
#min(scale(d_o$real_gestational_age))
plot(pca$x[,1], pca$x[,2],
     col = ifelse(d_o$cluster == 1, "orange4", "blue4" ),
     pch = ifelse(d_o$sepsis_presumed == 1, 1, 16 ), cex = {2.2 + scale(d_o$birth_weight)},
     las = 1, xlab = "PC1 (20.96%)", ylab = "PC 2 (10.01%)", ylim = c(-5, 5),
     font.axis = 2, font.lab = 2, cex.axis = 1.8, cex.lab  = 1.5,
     main = "col = cluster, size = weight, aes = sepsis_presumed")
#text(x = pca$x[,1], y = pca$x[,2], labels = round(d_o$birth_weight, digits = 1), cex = 0.5)
dev.off()


#FIGURE 2A2
pdf(file = "pca_clustersbirthweighttext.pdf")
#min(scale(d_o$real_gestational_age))
plot(pca$x[,1], pca$x[,2],
     col = ifelse(d_o$cluster == 1, "orange4", "blue4" ),
     pch = ifelse(d_o$sepsis_presumed == 1, 1, 16 ), cex = {2.2 + scale(d_o$birth_weight)},
     las = 1, xlab = "PC1 (20.96%)", ylab = "PC 2 (10.01%)", ylim = c(-5, 5),
     font.axis = 2, font.lab = 2, cex.axis = 1.8, cex.lab  = 1.5,
     main = "col = cluster, size = weight, aes = sepsis_presumed")
text(x = pca$x[,1], y = pca$x[,2], labels = round(d_o$birth_weight, digits = 1), cex = 0.5)
dev.off()

#FIGURE 2B2

pdf(file = "gestationalage.pdf")
p <- ggplot(aes(y = real_gestational_age, x = cluster,
                fill = cluster), data = d_o)
p+ geom_violin( aes(), col = "black")+ scale_fill_manual(values=c("orange4", "blue3"))+
  ylim(20, 30)+ ggtitle("Cluster Analysis, Gestational Age")+
  ylab("Gestational Age (weeks)")
dev.off()

names(d_o)
#FIGURE 2C2
pdf(file = "head_circ.pdf")
p <- ggplot(aes(y = Head_circumference, x = cluster,
                fill = cluster), data = d_o)
p+ geom_violin( aes(), col = "black")+ scale_fill_manual(values=c("orange4", "blue3"))+

  ylim(20, 30)+ ggtitle("Cluster Analysis, Head Circumference")+
  ylab("Head Circumference (cm)")
dev.off()
sink(file = "ttest of the features.txt")
print("t.test(Head_circumference ~ cluster, data = d_o)")
t.test(Head_circumference ~ cluster, data = d_o)
print("t.test(real_gestational_age ~ cluster, data = d_o)")
t.test(real_gestational_age ~ cluster, data = d_o)
print("t.test(birth_weight ~ cluster, data = d_o)")
t.test(birth_weight ~ cluster, data = d_o)
sink()
#######
#######
########
#data for Figure 3

setwd("...")
prop_jt <- readRDS("...")
str(prop_jt)
prop_jt$Patient_ID
prop_jt$Presumed_Sepsis
install.packages("semTools")

rac <- subset(prop_jt, prop_jt$RAC_Performed == 1)
hct <- subset(prop_jt, prop_jt$HCT_Performed == 1)

rac <- subset(prop_jt, prop_jt$RAC_Performed == 1)
bpd <- subset(rac, rac$RAC_Result == "2")
bpd

bpd$diagnosis <- rep("BPD", times  = nrow(bpd))
bpd$diagnosis

hct_pass <- subset(hct, hct$HCT_Result == "1")
hct_pass
hct_pass$diagnosis <- rep("hypox_pass", times = nrow(hct_pass))
hct_pass$diagnosis
```

```r
hct_2 <- subset(hct, hct$HCT_Reason_Failure == "2")
hct_2$diagnosis <- rep("desat_less85_65", times = nrow(hct_2))
hct_3 <- subset(hct, hct$HCT_Reason_Failure == "3")
hct_3$diagnosis <- rep("major_desat_15sec", times = nrow(hct_3))
hct_3$diagnosis

df_hct <- rbind(hct_pass, hct_2, hct_3, bpd)
names(df_hct)

df <- df_hct

test <- subset(df, df$Presumed_Sepsis == 0)
df$Number_Episodes_Presumed_Sepsis
df$Presumed_Sepsis
test$Presumed_Sepsis
test$Number_Episodes_Presumed_Sepsis <- rep(0, times = nrow(test))
test2 <- subset(df, df$Presumed_Sepsis != 0)
test2$Number_Episodes_Presumed_Sepsis
df <- rbind(test,test2)


rac$RAC_Result <- as.factor(rac$RAC_Result)
hct$HCT_Result <- as.factor(hct$HCT_Result)
df$Patient_ID <- NULL
names(df)
df2 <- df[, c(1:7, 11:16,  21:22, 24:25, 48:49, 67:74,79:81,
              89:92, 96:97, 101:105)]

names(df2)
summary(df2)
df2$diagnosis <- as.factor(df2$diagnosis)
summary(df2)

str(df2)
saveRDS(df2, file = "df2.R")
df2 <- readRDS("df2.R")


df2$surfactant_administration <- as.factor(df2$surfactant_administration)
df2$prophylactic_indomethacin <- as.factor(df2$prophylactic_indomethacin)
df2$Mother_Diabetes <- as.factor(df2$Mother_Diabetes)
df2$Mother_Asthma <- as.factor(df2$Mother_Asthma)
df2$Smoking_Home_Preg <- as.factor(df2$Smoking_Home_Preg)
df2$Mother_Medications_Preg <- as.factor(df2$Mother_Medications_Preg)
df2$Placental_Abruption <- as.factor(df2$Placental_Abruption)
df2$Clinical_Chorio <- as.factor(df2$Clinical_Chorio)
df2$Delivery_Membrance_Rupture <- as.factor(df2$Delivery_Membrance_Rupture)
df2$PDA <- as.factor(df2$PDA)
df2$Pulmonary_Hypertension <- as.factor(df2$Pulmonary_Hypertension)
df2$Tracheotomy <- as.factor(df2$Tracheotomy)
```

```
df2$Presumed_Sepsis <- as.factor(df2$Presumed_Sepsis)
df2$Retinopathy_of_Prematurity <- as.factor(df2$Retinopathy_of_Prematurity)
df2$Ventricular_Shunt <- as.factor(df2$Ventricular_Shunt)
df2$Intubation <- as.factor(df2$Intubation)
df2$Sex <- as.factor(df2$Sex)
df2$diagnosis <- as.factor(df2$diagnosis)
#df2$ <- as.factor(df2$)
str(df2)
df2$diagnosis
df2[321,]
##### picking up where I left off.

df_impute <- rfImpute(diagnosis ~., data  = df2, ntree = 1000)
install.packages("missForest")
library(missForest)
test <- missForest(df2)
testzz <- missForest(df2222)
testzz$OOBerror
summary(df2222)
summary(testzz$ximp)
class(test)
test
test$ximp
nrow(test$ximp)
nrow(df2)
testzz$OOBerror
saveRDS(test, "miss_forestimputation.R")
test <- readRDS("...")
test$ximp
dfz <- test$ximp#####

str(df3)
dfz$gestational_age <- dfz$Completed_Weeks + dfz$Completed_Days/3
levels(dfz$diagnosis)


pp <- boxplot(dfz$gestational_age ~dfz$diagnosis, horizontal = TRUE, las = 1,
              col =        c("red4", "blue3", "orange2", "green4"),
              xlab = "Gestational Age (weeks)", main = "Respiratory Dysfunction
Associated with Decreased Gestational Age",
              font.axis = 2, font.lab = 2, font.main = 2
)
dfz$diagnosis <- relevel(dfz$diagnosis,"hypox_pass"  )

setwd("...")
pdf(file = "gestational age link boplot.pdf")
boxplot(dfz$gestational_age ~dfz$diagnosis, horizontal = TRUE, las = 1,
        col =        c("green4", "blue3", "orange2", "red4"),
        xlab = "Gestational Age (weeks)",
        font.axis = 2, font.lab = 2, font.main = 2, ylab = ""
)
dev.off()
pdf(file = "birthwegiht link boplot.pdf")
boxplot(dfz$Patient_wt ~dfz$diagnosis, horizontal = TRUE, las = 1,
        col =        c("green4", "blue3", "orange2", "red4"),
        xlab = "Birth Weight (grams)",
        font.axis = 2, font.lab = 2, font.main = 2, ylab = ""
)
dev.off()
pdf(file = "headcirk link boplot.pdf")
boxplot(dfz$head_circ ~dfz$diagnosis, horizontal = TRUE, las = 1,
        col =        c("green4", "blue3", "orange2", "red4"),
        xlab = "Head Circumference (cm)",
        font.axis = 2, font.lab = 2, font.main = 2, ylab = ""
)
dev.off()

pdf(file = "headcirk link boplot.pdf")
boxplot(dfz$RAC_Weight ~dfz$diagnosis, horizontal = TRUE, las = 1,
        col =        c("green4", "blue3", "orange2", "red4"),
        xlab = "Head Circumference (cm)",
        font.axis = 2, font.lab = 2, font.main = 2, ylab = ""
)
dev.off()

sink(file = "tukeyhsd of gestational age with diagnosis.txt")
print("this is looking at the gestational age anova of the imputed data")
TukeyHSD(aov(dfz$gestational_age ~ dfz$diagnosis))
print("this is looking at the rac weight anova of the imputed data")
TukeyHSD(aov(dfz$RAC_Weight ~ dfz$diagnosis))
print("this is looking at the head circ age anova of the imputed data")

TukeyHSD(aov(dfz$head_circ ~ dfz$diagnosis))
print("this is looking at the birth age anova of the imputed data")

TukeyHSD(aov(dfz$Patient_wt ~ dfz$diagnosis))



########
######
#### FIgure 4 data.
saveRDS(df3, file = "df3 used for the logistict regression.R")
df3 <- readRDS("df3 used for the logistict regression.R")
head(prop_jt)
names(prop_jt)
min(scale(df3$gestation_at_birth))
library(ggplot2)

gestation_at_birth +head_circ+Patient_wt
```

```
summary(my_model)
summary(my_model2)

1-pchisq(577.89-543.88, 416-411)
mmm <- predict(my_model, type = "response")
mmm2 <- predict(my_model2, type = "response")

#trying to see which model is the best.
library(randomForest)
rf_model <- randomForest(Presumed_Sepsis ~ diagnosis +
                           gestation_at_birth +head_circ+Patient_wt+
                           RAC_Baseline_Resp_Rate + RAC_Weight,
                         data = df3z, ntree = 1000)
rf_model$importance
rf_predict <- predict(rf_model, type = "prob")
head(rf_predict)
library(e1071)
svm_model <- svm(Presumed_Sepsis ~ diagnosis +
                   gestation_at_birth +head_circ+Patient_wt+
                   RAC_Baseline_Resp_Rate + RAC_Weight,
                 data = df3z, kernel = "radial")
sv_predict <- predict(svm_model, type = "prob")
confusionMatrix(sv_predict, df3z$Presumed_Sepsis)
confusionMatrix(predict(rf_model), df3z$Presumed_Sepsis)
zzz <- predict(my_model, type = "response")
for(i in 1:length(zzz)) {
  my_point <- ifelse(zzz[i] > 0.5,  1, 0)
  zzz[i] <- my_point
}
zzz <-as.factor(zzz)
levels(zzz)
levels(df3$Presumed_Sepsis)
confusionMatrix(zzz, df3$Presumed_Sepsis)
rf_votes <- predict(rf_model, type = "prob")
head(rf_votes)
rf_1 <- rf_predict[,2]


df3z$rf_model_sepsis_predict <- rf_1
#the glm is ~ 61 % accurate, and has a p over the nir of 2.2
plot(x = df3z$rf_model_sepsis_predict, y = df3z$sepsis_predict, col = ifelse(df3
z$Presumed_Sepsis == 0, "blue", "red4"))
df3z$sepsis_predict <- mmm
df3z$sepsis_predict2 <- mmm2

df3z$rf_model_sepsis_predict <- rf_1

plot(df3$rf_model_sepsis_predict, ylim = c(0, 1), col = ifelse(df3$Presumed_Seps
is == 0, "blue", "red4"),
     pch = 16)
```

```
abline(h = 0.5)
names(df3)
plot(df3z$Number_Episodes_Presumed_Sepsis, col = df3z$diagnosis)
str(df3)
summary(df3z$Number_Episodes_Presumed_Sepsis)
df3z$head_circ




##### These are the four models that will be used to calculate this effect in th
e
#####experimets.
#gestation_at_birth +head_circ+Patient_wt
g_model <- ggplot(data = df3z, aes(x = RAC_Weight, y = sepsis_predict,
                                   color = diagnosis,
                                   size = Number_Episodes_Presumed_Sepsis))
g_model2 <- ggplot(data = df3z, aes(x = gestation_at_birth, y = sepsis_predict,
                                    color = diagnosis,
                                    size = Number_Episodes_Presumed_Sepsis))
g_model3 <- ggplot(data = df3z, aes(x = head_circ, y = sepsis_predict,
                                    color = diagnosis,
                                    size = Number_Episodes_Presumed_Sepsis))
g_model4<- ggplot(data = df3z, aes(x = Patient_wt, y = sepsis_predict,
                                   color = diagnosis,
                                   size = Number_Episodes_Presumed_Sepsis))
#####glm model
g_model_2 <- ggplot(data = df3z, aes(x = RAC_Weight, y = sepsis_predict2,
                                     color = diagnosis,
                                     size = Number_Episodes_Presumed_Sepsis))
g_model2_2 <- ggplot(data = df3z, aes(x = gestation_at_birth, y = sepsis_predict
2,
                                      color = diagnosis,
                                      size = Number_Episodes_Presumed_Sepsis))
g_model3_2 <- ggplot(data = df3z, aes(x = head_circ, y = sepsis_predict2,
                                      color = diagnosis,
                                      size = Number_Episodes_Presumed_Sepsis))
g_model4_2<- ggplot(data = df3z, aes(x = Patient_wt, y = sepsis_predict2,
                                     color = diagnosis,
                                     size = Number_Episodes_Presumed_Sepsis))
###$##rf model
df3z$rf_model_sepsis_predict
g_model_2rf <- ggplot(data = df3z, aes(x = RAC_Weight, y = rf_model_sepsis_predi
ct,
                                       color = diagnosis,
                                       size = Number_Episodes_Presumed_Sepsis))
g_model2_2rf <- ggplot(data = df3z, aes(x = gestation_at_birth, y = rf_model_sep
sis_predict,
                                        color = diagnosis,
                                        size = Number_Episodes_Presumed_Sepsis))
```

```
g_model3_2rf <- ggplot(data = df3z, aes(x = head_circ, y = rf_model_sepsis_predi
ct,
                                        color = diagnosis,
                                        size = Number_Episodes_Presumed_Sepsis))
g_model4_2rf <- ggplot(data = df3z, aes(x = Patient_wt, y = rf_model_sepsis_pred
ict,
                                        color = diagnosis,
                                        size = Number_Episodes_Presumed_Sepsis))



g_model4 + geom_point(pch = 1) + geom_smooth(method = "glm")+
  guides(fill = FALSE)
#pdf(file = "test.pdf")
plot0 <-  g_model + geom_point(pch = 1) + geom_smooth(method = "glm")+
  guides(fill = FALSE)  +
  geom_point(pch = 1, size = df3z$Number_Episodes_Presumed_Sepsis)+
  scale_color_manual(values = c("green4", "red4", "orange2", "blue4"))+
  geom_smooth(method = "glm")
plot2 <-  g_model2 + geom_point(pch = 1) + geom_smooth(method = "glm")+
  guides(fill = FALSE)  +
  geom_point(pch = 1, size = df3z$Number_Episodes_Presumed_Sepsis)+
  scale_color_manual(values = c("green4", "red4", "orange2", "blue4"))+
  geom_smooth(method = "glm")
plot3 <-  g_model3 + geom_point(pch = 1) + geom_smooth(method = "glm")+
  guides(fill = FALSE)  +
  geom_point(pch = 1, size = df3z$Number_Episodes_Presumed_Sepsis)+
  scale_color_manual(values = c("green4", "red4", "orange2", "blue4"))+
  geom_smooth(method = "glm")+ xlim(20, 30)
plot4 <-  g_model4 + geom_point(pch = 1) + geom_smooth(method = "glm")+
  guides(fill = FALSE)  +
  geom_point(pch = 1, size = df3z$Number_Episodes_Presumed_Sepsis)+
  scale_color_manual(values = c("green4", "red4", "orange2", "blue4"))+
  geom_smooth(method = "glm")
setwd("...")
plot0_2rf <-  g_model_2rf + geom_point(pch = 1) + geom_smooth(method = "glm")+
  guides(fill = FALSE)  +
  geom_point(pch = 1, size = df3z$Number_Episodes_Presumed_Sepsis)+
  scale_color_manual(values = c("green4", "red4", "orange2", "blue4"))+
  geom_smooth(method = "glm")

pdf(file = "respiratory_plot0_2.pdf")
plot0_2
dev.off()
plot2_2 <-  g_model2_2rf + geom_point(pch = 1) + geom_smooth(method = "glm")+
  guides(fill = FALSE)  +
  geom_point(pch = 1, size = df3z$Number_Episodes_Presumed_Sepsis)+
  scale_color_manual(values = c("green4", "red4", "orange2", "blue4"))+
  geom_smooth(method = "glm")
pdf(file = "respiratory_plot2_2.pdf")
```

```
plot2_2
dev.off()
plot3_2 <-  g_model3_2 + geom_point(pch = 1) + geom_smooth(method = "glm")+
  guides(fill = FALSE)  +
  geom_point(pch = 1, size = df3z$Number_Episodes_Presumed_Sepsis)+
  scale_color_manual(values = c("green4", "red4", "orange2", "blue4"))+
  geom_smooth(method = "glm")+ xlim(20, 30)
pdf(file = "respiratory_plot3_2.pdf")

plot3_2
dev.off()
plot4_2 <-  g_model4_2 + geom_point(pch = 1) + geom_smooth(method = "glm")+
  guides(fill = FALSE)  +
  geom_point(pch = 1, size = df3z$Number_Episodes_Presumed_Sepsis)+
  scale_color_manual(values = c("green4", "red4", "orange2", "blue4"))+
  geom_smooth(method = "glm")

pdf(file = "respiratory_plot4_2.pdf")
plot4_2

dev.off()

saveRDS(df3z, file = "df3z.R")
######now the ggplot with the rf model
plot0_2 <-  g_model_2 + geom_point(pch = 1) + geom_smooth(method = "glm")+
  guides(fill = FALSE)  +
  geom_point(pch = 1, size = df3z$Number_Episodes_Presumed_Sepsis)+
  scale_color_manual(values = c("green4", "red4", "orange2", "blue4"))+
  geom_smooth(method = "glm")

pdf(file = "respiratory_plot0_2.pdf")
plot0_2
dev.off()
plot2_2 <-  g_model2_2 + geom_point(pch = 1) + geom_smooth(method = "glm")+
  guides(fill = FALSE)  +
  geom_point(pch = 1, size = df3z$Number_Episodes_Presumed_Sepsis)+
  scale_color_manual(values = c("green4", "red4", "orange2", "blue4"))+
  geom_smooth(method = "glm")
pdf(file = "respiratory_plot2_2.pdf")

plot2_2
dev.off()
plot3_2 <-  g_model3_2 + geom_point(pch = 1) + geom_smooth(method = "glm")+
  guides(fill = FALSE)  +
  geom_point(pch = 1, size = df3z$Number_Episodes_Presumed_Sepsis)+
  scale_color_manual(values = c("green4", "red4", "orange2", "blue4"))+
  geom_smooth(method = "glm")+ xlim(20, 30)
pdf(file = "respiratory_plot3_2.pdf")
```

```
plot3_2
dev.off()
plot4_2 <-  g_model4_2 + geom_point(pch = 1) + geom_smooth(method = "glm")+
  guides(fill = FALSE)  +
  geom_point(pch = 1, size = df3z$Number_Episodes_Presumed_Sepsis)+
  scale_color_manual(values = c("green4", "red4", "orange2", "blue4"))+
  geom_smooth(method = "glm")

pdf(file = "respiratory_plot4_2.pdf")
plot4_2

dev.off()




########
#data for figure 5

#data obtained from daily measurements.

#
dddd <- readRDS(file = "...")
class(dddd)
length(dddd)
dddd[[1]]$STUDY.x
names(dddd)
View(dddd[[24]])
length(dddd)
plot(dddd[[234]]$GNMD_WEIGHT, x = dddd[[234]]$corrected_pma,
     las = 2, font.axis = 2, xlab = "Corrected PMA (weeks)", ylab = "Weight (g)"
,
     font.axis = 2, font.lab = 2, pch = 16, cex  = 0.75)
library(RColorBrewer)
display.brewer.all()

my_rainbow <- rainbow( n= length(dddd))
my_pallette <- sample(my_rainbow, size = length(my_rainbow), replace = FALSE)
palette(my_pallette)
display.brewer.all(n = NULL, type = "all", select = NULL,
                   colorblindFriendly = TRUE)
display.brewer.pal(n = 835, name = 'Dark2')

palette()
fenton <- read.csv("...")
setwd("...")
pdf(file = "birthweights_lp3.pdf")
plot(dddd[[1]]$GNMD_WEIGHT, x = dddd[[1]]$corrected_pma,
     las = 1, font.axis = 2, xlab = "Corrected PMA (weeks)", ylab = "Weight (g)"
,
     font.axis = 2, font.lab = 2, pch = 16, cex  = 0.5, xlim = c(24, 40),
     ylim = c(250, 4000), col = my_pallette[1])
for(i in 2:length(dddd)){
  lines(dddd[[i]]$GNMD_WEIGHT, x = dddd[[i]]$corrected_pma,
        pch = 16, cex  = 0.75,col = rgb(red = col2rgb(my_pallette[i])[1,1],
                                        green = col2rgb(my_pallette[i])[2,1],
                                        blue = col2rgb(my_pallette[i])[3,1],
                                        max = 255, alpha = 125))
  print(i)
}
for(i in 2:length(dddd)){
  points(dddd[[i]]$GNMD_WEIGHT, x = dddd[[i]]$corrected_pma,
         pch = 16, cex  = 0.75,col = rgb(red = col2rgb(my_pallette[i])[1,1],
                                         green = col2rgb(my_pallette[i])[2,1],
                                         blue = col2rgb(my_pallette[i])[3,1],
                                         max = 255, alpha = 95))
  print(i)
}
points(fenton$Week, fenton$g10, col = "pink2", type = "l", lwd =2)
points(fenton$Week, fenton$g90, col = "pink2", type = "l", lwd =2)
points(fenton$Week, fenton$b90, col = "blue2", type = "l", lwd =2)
points(fenton$Week, fenton$b10, col = "blue2", type = "l", lwd =2)

dev.off()

pdf(file = "headcirc_lp3.pdf")
plot(dddd[[1]]$GNMD_VSHEAD, x = dddd[[1]]$corrected_pma,
     las = 1, font.axis = 2, xlab = "Corrected PMA (weeks)",
     ylab = "Head Circumference (cm)",
     font.axis = 2, font.lab = 2, pch = 16, cex  = 0.5, xlim = c(24, 40),
     ylim = c(10, 40), col = my_pallette[1])
for(i in 2:length(dddd)){
  lines(dddd[[i]]$GNMD_VSHEAD, x = dddd[[i]]$corrected_pma,
        pch = 16, cex  = 0.75,col = rgb(red = col2rgb(my_pallette[i])[1,1],
                                        green = col2rgb(my_pallette[i])[2,1],
                                        blue = col2rgb(my_pallette[i])[3,1],
                                        max = 255, alpha = 125))
  print(i)
}
for(i in 2:length(dddd)){
  points(dddd[[i]]$GNMD_VSHEAD, x = dddd[[i]]$corrected_pma,
         pch = 16, cex  = 0.75,col = rgb(red = col2rgb(my_pallette[i])[1,1],
                                         green = col2rgb(my_pallette[i])[2,1],
                                         blue = col2rgb(my_pallette[i])[3,1],
                                         max = 255, alpha = 95))
  print(i)
}
#points(fenton$Week, fenton$g10, col = "pink2", type = "l", lwd =2)
#points(fenton$Week, fenton$g90, col = "pink2", type = "l", lwd =2)
#points(fenton$Week, fenton$b90, col = "blue2", type = "l", lwd =2)
```

```
#points(fenton$Week, fenton$b10, col = "blue2", type = "l", lwd =2)

dev.off()

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

#code for Figure 6.

df4merge <- readRDS("/...")
#rf_model3 <- readRDS("...")
#df4merge has all of the training set in it.
df4merge <- readRDS("...")
str(df4merge)
cor(df4merge$GNMD_VSHEAD, df4merge$GNMD_WEIGHT, na.action = na.omit)
df <- data.frame("head" = df4merge$GNMD_VSHEAD, "daily_wt" = df4merge$GNMD_WEIGH
T )
df <- na.omit(df)

my_features <- c("GNMD_WEIGHT",  "corrected_pma", "GNMD_VSHEAD", "birth_weight",
                 "Sex" , "Head_circumference")
df4merge1 <- df4merge[,my_features]
df4merge2 <-na.omit(df4merge1)
tune.radial <- tune.svm(GNMD_WEIGHT~  corrected_pma + GNMD_VSHEAD + birth_weight
+
                          Sex + Head_circumference, data = df4merge2,
                        gamma =seq(.01, 0.1, by = .02),kernel = 'radial',
                        cost = seq(0.1,1, by = 0.2))

hc_svm_total <- svm(GNMD_VSHEAD~  corrected_pma +  birth_weight +
                      Sex + Head_circumference, data = df4merge)
hc_svm_total
RMSE(predict(hc_svm_radial$best.model, df4merge3), df4merge3$GNMD_VSHEAD)
print("mae")
MAE(predict(hc_svm_radial$best.model, df4merge3), df4merge3$GNMD_VSHEAD)

print("into on the test set")
print("rmse")
RMSE(predict(hc_svm_radial$best.model, testmerge2), testmerge2$GNMD_VSHEAD)
print("mae")

MAE(predict(hc_svm_radial$best.model, testmerge2), testmerge2$GNMD_VSHEAD)
sink()

library(readxl)
uruguay <- read_excel("...")
str(uruguay)
#svm_model <- readRDS("...")
library(e1071)
svm_df4$call
```

```
uruguay$GNMD_WEIGHT <- uruguay$`Peso al egreso`
uruguay$corrected_pma <- uruguay$`Edad gestacional al EGRESO`
uruguay$GNMD_VSHEAD <- (uruguay$`Perímetro cefálico al egreso`)/10
uruguay$birth_weight <- uruguay$`Peso en g al nacer`
uruguay$Head_circumference <- (uruguay$`Perímetro cefálico en mm`)/10
uruguay$Sexo <- as.factor(uruguay$Sexo)
uruguay$gestation_at_birth <- uruguay$`Edad gestacional al NACER`
levels(uruguay$Sexo)
uruguay$pt_id <- as.factor(uruguay$`número de cédula materna`)
levels(uruguay$pt_id) <- paste("pt_", levels(uruguay$pt_id), sep = "")

library(plyr)
Sex <- uruguay$Sexo
class(Sex)
levels(Sex)
Sex <- revalue(Sex, c("femenino" = "2", "masculino" = "3", "no definido" = "1"))
svm_df4$call
uruguay$Sex <- Sex
str(uruguay)
setwd("...")
uruguay <- readRDS( file = "uruguay_3_9_2021.R")
##make the subset

summary(premie$`Edad gestacional al NACER`)
class(uruguay$`Fecha de nacimiento`)

#write.csv(premie, file = "uruguay_23_30.csv")

class(uruguay$`Fecha egreso recién nacido`)
uruguay <- subset(uruguay, is.na(uruguay$`Peso al egreso`) == FALSE)
uruguay$`Peso al egreso`


b <- difftime(uruguay$`Fecha egreso recién nacido`,uruguay$`Fecha de nacimiento
`,
              units = "weeks")
uruguay$diff_time <- b
View(uruguay)
uruguay <- subset(uruguay, is.na(uruguay$diff_time) == FALSE)
dim(uruguay)
uruguay$corrected_pma <- NULL
class(uruguay$diff_time)
uruguay$corrected_pma <- uruguay$`Edad gestacional al NACER` + as.numeric(urugua
y$diff_time)
class(uruguay$corrected_pma)
uruguay <- subset(uruguay, uruguay$corrected_pma > 10)
dim(uruguay)
uruguay$corrected_pma

dim(uruguay)
```

```
u1 <- subset(uruguay, uruguay$`Edad gestacional al NACER` < 30)
u2 <- subset(uruguay, uruguay$`Edad gestacional al NACER` > 30)
summary(u1$`Edad gestacional al NACER`)
dim(u1)
dim(u2)
head(u1)
names(u1)
uruguay$`Edad gestacional al NACER`
uruguay$gestation_at_birth <- uruguay$`Edad gestacional al NACER`

my_names <- c("GNMD_WEIGHT" , "pt_id","gestation_at_birth",
              "corrected_pma"  ,  "GNMD_VSHEAD",
              "birth_weight"  ,"Head_circumference" ,
              "Sex")
u1 <-u1[,my_names]
u2 <-u2[,my_names]

dim(u1)
names(u1)
u1 <- na.omit(u1)
u2 <- na.omit(u2)
dim(u1)
dim(u2)
######making the data for Figure 6.

setwd("...")
tune.radial <- readRDS(file = "tune.radial.R")
my_svm <- tune.radial$best.model
tune.radial$best.model
hc_svm_radial <- readRDS("hc_svm_radial_tune.R")
hc_svm <- hc_svm_radial$best.model


#########
my_names
names(u1)
u1$gestation_at_birth
u1$pt_id
names(u1)
dim(u1)
hc_svm
uruguay_list <- list()

for(i in 1:nrow(u1)) {
  library(ggplot2)
  library(e1071)
  #  i = 1
  ptx <- u1[i,]

  my_days <- seq(from = ptx$gestation_at_birth[1], to =  ptx$corrected_pma[1] ,
by = 0.035)
  new_df <- data.frame(
    "corrected_pma" = my_days,
    "birth_weight"  = rep(ptx$birth_weight[1], times = length(my_days)),
    "Sex"  = rep(ptx$Sex[1], times = length(my_days)),
    "Head_circumference"= rep(ptx$Head_circumference[1], times = length(my_days)
)
  )
  str(ptx)
  str(new_df)
  predicted_hc <- predict(hc_svm, new_df)
  new_df$GNMD_VSHEAD <- predicted_hc
  predicted_wt <- predict(my_svm, new_df)
  new_df$wt_predicted <- predicted_wt

  discharge_predict <- predict(my_svm, ptx)
  ptx$discharge_predict <- discharge_predict
  fenton <- read.csv("...")
  p <- ggplot(data = new_df, aes(x = corrected_pma, y = wt_predicted))
  my_plot <- p + geom_point() + geom_point(data = ptx, aes(x=corrected_pma, y= G
NMD_WEIGHT), colour="blue")+
    ggtitle(paste("Predicted growth of ", ptx$pt_id, "red = discharge_predict, b
lue = actual wt",sep = "")) + ylab("Weight (grams)") +
    xlab("Post-menstrual age (weeks")+ geom_line(data = fenton, aes( y = g10, x
= Week), col =
                                                   "pink")+
    geom_line(data = fenton, aes( y = g90, x = Week), col =
                "pink")+
    geom_line(data = fenton, aes( y = b90, x = Week), col =
                "blue2")+
    geom_line(data = fenton, aes( y = b10, x = Week), col =
                "blue2")+
    geom_point(data = ptx, aes(x=corrected_pma, y= discharge_predict), colour="r
ed4")

  max_new_df <- subset(new_df, new_df$corrected_pma == max(new_df$corrected_pma)
)

  delta_wt <- ptx$GNMD_WEIGHT - discharge_predict
  delta_wt_hc_model <- ptx$GNMD_WEIGHT  - max_new_df$wt_predicted
  my_list <- list("plot" = my_plot, "new_data" = new_df, "pt_data" = ptx, "delta
_wt" = delta_wt,
                  "delta_wt_hc_model"  = delta_wt_hc_model)

  uruguay_list[[i]] = my_list
  names(uruguay_list)[i] = as.character(ptx$pt_id[1])
}
head(u1)
a <- subset(u1, u1$`número de cédula materna` == "55747178")
######
```

```
a$`número de cédula materna`
a$corrected_pma
a$`Edad gestacional al NACER`
a$corrected_pma
a$birth_weight
my_days <- seq(from = a$`Edad gestacional al NACER`, to =  a$corrected_pma , by
= 0.001)
new_df <- data.frame(
  "corrected_pma" = my_days,
  "birth_weight" = rep(a$birth_weight, times = length(my_days)),
  "Sex" = rep(a$Sex, times = length(my_days)),
  "Head_circumference"= rep(a$Head_circumference, times = length(my_days))
)

predicted_hc <- predict(hc_svm, new_df)
new_df$GNMD_VSHEAD <- predicted_hc
predicted_wt <- predict(my_svm, new_df)
new_df$wt_predicted <- predicted_wt
#fenton <- read.csv("...")
setwd("...")
p <- ggplot(data = new_df, aes(x = corrected_pma, y = wt_predicted))
my_plot <- p + geom_point() +
  geom_point(data = a, cex = 1.5,aes(x=a$`Edad gestacional al NACER`, y= a$birth
_weight), colour="blue")+
  ylim(200, 1050) + xlim(24, 28)+
  ggtitle(paste("Patient", a$`número de cédula materna`,  " black = model, red =
birth wt,blue = death wt",sep = "")) + ylab("Weight (grams)") +
  xlab("Post-menstrual age (weeks)")+ geom_line(data = fenton, aes( y = g10, x =
Week), col =
                                                    "pink")+
  geom_line(data = fenton, aes( y = g90, x = Week), col =
              "pink")+
  geom_line(data = fenton, aes( y = b90, x = Week), col =
              "blue2")+
  geom_line(data = fenton, aes( y = b10, x = Week), col =
              "blue2")+ geom_point(data = a, cex = 1.5,aes(x=corrected_pma, y= G
NMD_WEIGHT), colour="red2")
my_plot
pdf(file = paste(a$`número de cédula materna`, ".pdf", sep = ""))
print(my_plot)
dev.off()
#####

a <- subset(u1, u1$`número de cédula materna` == "53532533")
a
a$`número de cédula materna`
######

a$corrected_pma
a$`Edad gestacional al NACER`
```

```
a$corrected_pma
a$birth_weight
my_days <- seq(from = a$`Edad gestacional al NACER`, to =  a$corrected_pma , by
= 0.001)
new_df <- data.frame(
  "corrected_pma" = my_days,
  "birth_weight" = rep(a$birth_weight, times = length(my_days)),
  "Sex" = rep(a$Sex, times = length(my_days)),
  "Head_circumference"= rep(a$Head_circumference, times = length(my_days))
)

predicted_hc <- predict(hc_svm, new_df)
new_df$GNMD_VSHEAD <- predicted_hc
predicted_wt <- predict(my_svm, new_df)
new_df$wt_predicted <- predicted_wt
#fenton <- read.csv("...")
setwd("...")
p <- ggplot(data = new_df, aes(x = corrected_pma, y = wt_predicted))
my_plot <- p + geom_point() +
  geom_point(data = a, cex = 1.5,aes(x=a$`Edad gestacional al NACER`, y= a$birth
_weight), colour="blue")+
  ylim(200, 1050) + xlim(24, 28)+
  ggtitle(paste("Patient", a$`número de cédula materna`,  " black = model, red =
birth wt,blue = death wt",sep = "")) + ylab("Weight (grams)") +
  xlab("Post-menstrual age (weeks)")+ geom_line(data = fenton, aes( y = g10, x =
Week), col =
                                                 "pink")+
  geom_line(data = fenton, aes( y = g90, x = Week), col =
              "pink")+
  geom_line(data = fenton, aes( y = b90, x = Week), col =
              "blue2")+
  geom_line(data = fenton, aes( y = b10, x = Week), col =
              "blue2")+ geom_point(data = a, cex = 1.5,aes(x=corrected_pma, y= G
NMD_WEIGHT), colour="red2")
my_plot
pdf(file = paste(a$`número de cédula materna`, ".pdf", sep = ""))
print(my_plot)
dev.off()

#####


a <- subset(u1, u1$`número de cédula materna` == "48042787")
a
a$`número de cédula materna`
######

a$corrected_pma
a$`Edad gestacional al NACER`
a$corrected_pma
```

```
a$birth_weight
my_days <- seq(from = a$`Edad gestacional al NACER`, to =  a$corrected_pma , by
= 0.001)
new_df <- data.frame(
  "corrected_pma" = my_days,
  "birth_weight" = rep(a$birth_weight, times = length(my_days)),
  "Sex" = rep(a$Sex, times = length(my_days)),
  "Head_circumference"= rep(a$Head_circumference, times = length(my_days))
)

predicted_hc <- predict(hc_svm, new_df)
new_df$GNMD_VSHEAD <- predicted_hc
predicted_wt <- predict(my_svm, new_df)
new_df$wt_predicted <- predicted_wt
#fenton <- read.csv("...")
setwd("...")
p <- ggplot(data = new_df, aes(x = corrected_pma, y = wt_predicted))
my_plot <- p + geom_point() +
  geom_point(data = a, cex = 1.5,aes(x=a$`Edad gestacional al NACER`, y= a$birth
_weight), colour="blue")+
  ylim(200, 1050) + xlim(24, 28)+
  ggtitle(paste("Patient", a$`número de cédula materna`,  " black = model, red =
birth wt,blue = death wt",sep = "")) + ylab("Weight (grams)") +
  xlab("Post-menstrual age (weeks)")+ geom_line(data = fenton, aes( y = g10, x =
Week), col =
                                                  "pink")+
  geom_line(data = fenton, aes( y = g90, x = Week), col =
              "pink")+
  geom_line(data = fenton, aes( y = b90, x = Week), col =
              "blue2")+
  geom_line(data = fenton, aes( y = b10, x = Week), col =
              "blue2")+ geom_point(data = a, cex = 1.5,aes(x=corrected_pma, y= G
NMD_WEIGHT), colour="red2")
my_plot
pdf(file = paste(a$`número de cédula materna`, ".pdf", sep = ""))
print(my_plot)
dev.off()

####

a <- subset(u1, u1$`número de cédula materna` == "46941109")
a
a$`número de cédula materna`
######

a$corrected_pma
a$`Edad gestacional al NACER`
a$corrected_pma
a$birth_weight
my_days <- seq(from = a$`Edad gestacional al NACER`, to =  a$corrected_pma , by
= 0.001)
new_df <- data.frame(
  "corrected_pma" = my_days,
  "birth_weight" = rep(a$birth_weight, times = length(my_days)),
  "Sex" = rep(a$Sex, times = length(my_days)),
  "Head_circumference"= rep(a$Head_circumference, times = length(my_days))
)

predicted_hc <- predict(hc_svm, new_df)
new_df$GNMD_VSHEAD <- predicted_hc
predicted_wt <- predict(my_svm, new_df)
new_df$wt_predicted <- predicted_wt
#fenton <- read.csv("/Users/oter04/Documents/Projects/prop/Fenton_etal.csv")
setwd("/Users/oter04/Documents/projects/prop/Manuscript/figure6 files")
p <- ggplot(data = new_df, aes(x = corrected_pma, y = wt_predicted))
my_plot <- p + geom_point() +
  geom_point(data = a, cex = 1.5,aes(x=a$`Edad gestacional al NACER`, y= a$birth
_weight), colour="blue")+
  ylim(200, 1050) + xlim(24, 28)+
  ggtitle(paste("Patient", a$`número de cédula materna`,  " black = model, red =
birth wt,blue = death wt",sep = "")) + ylab("Weight (grams)") +
  xlab("Post-menstrual age (weeks)")+ geom_line(data = fenton, aes( y = g10, x =
Week), col =
                                                  "pink")+
  geom_line(data = fenton, aes( y = g90, x = Week), col =
            "pink")+
  geom_line(data = fenton, aes( y = b90, x = Week), col =
            "blue2")+
  geom_line(data = fenton, aes( y = b10, x = Week), col =
            "blue2")+ geom_point(data = a, cex = 1.5,aes(x=corrected_pma, y= G
NMD_WEIGHT), colour="red2")
my_plot
pdf(file = paste(a$`número de cédula materna`, ".pdf", sep = ""))
print(my_plot)
dev.off()




a <- subset(u1, u1$`número de cédula materna` == "49650462")
a
a$`número de cédula materna`
######

a$corrected_pma
a$`Edad gestacional al NACER`
a$corrected_pma
a$birth_weight
my_days <- seq(from = a$`Edad gestacional al NACER`, to =  a$corrected_pma , by
= 0.001)
```

```
new_df <- data.frame(
  "corrected_pma" = my_days,
  "birth_weight" = rep(a$birth_weight, times = length(my_days)),
  "Sex" = rep(a$Sex, times = length(my_days)),
  "Head_circumference"= rep(a$Head_circumference, times = length(my_days))
)

predicted_hc <- predict(hc_svm, new_df)
new_df$GNMD_VSHEAD <- predicted_hc
predicted_wt <- predict(my_svm, new_df)
new_df$wt_predicted <- predicted_wt

p <- ggplot(data = new_df, aes(x = corrected_pma, y = wt_predicted))
my_plot <- p + geom_point() +
  geom_point(data = a, cex = 1.5,aes(x=a$`Edad gestacional al NACER`, y= a$birth
_weight), colour="blue")+
  ylim(200, 1050) + xlim(24, 28)+
  ggtitle(paste("Patient", a$`número de cédula materna`,  " black = model, red =
birth wt,blue = death wt",sep = "")) + ylab("Weight (grams)") +
  xlab("Post-menstrual age (weeks)")+ geom_line(data = fenton, aes( y = g10, x =
Week), col =
                                                    "pink")+
  geom_line(data = fenton, aes( y = g90, x = Week), col =
              "pink")+
  geom_line(data = fenton, aes( y = b90, x = Week), col =
              "blue2")+
  geom_line(data = fenton, aes( y = b10, x = Week), col =
              "blue2")+ geom_point(data = a, cex = 1.5,aes(x=corrected_pma, y= G
NMD_WEIGHT), colour="red2")
my_plot
pdf(file = paste(a$`número de cédula materna`, ".pdf", sep = ""))
print(my_plot)
dev.off()

fallece$observed
fallece_fenton

####df_m_split...
df_m_split <- split(df_moving, df_moving$NEWPT)
names(df_m_split)


#figue 7

dddd <- readRDS(file = "....")
prop <- dddd
class(prop)
df4merge <- readRDS("/...")
class(df4merge)
dim(df4merge)
```

```
levels(df4merge$NEWPT.x)
library(e1071)

eg <- c("corrected_pma", "birth_weight",
        "GNMD_VSHEAD", "Sex", "Head_circumference","GNMD_WEIGHT" )
svm_tune <- tune.svm(GNMD_WEIGHT ~ corrected_pma + GNMD_VSHEAD + birth_weight +
                       Sex + Head_circumference, data = df4merge, na.action = na
.omit,
                     kernal = 'radialr', gamma = c(0.1, 0.5, 1,2,3,4))
svm_tune$best.model
a <- Sys.time()
my_svm <- svm(GNMD_WEIGHT ~ corrected_pma + GNMD_VSHEAD + birth_weight +
                Sex + Head_circumference, data = df4merge, na.action = na.omit)
#my_svm$residuals
b <- Sys.time()
b - a

library(caret)
df3merge <- na.omit(df4merge[, eg])
RMSE(df3merge$GNMD_WEIGHT, predict(my_svm, df3merge))
RMSE(df3merge$GNMD_WEIGHT, predict(svm_tune$best.model, df3merge))

length(prop)
lapply(prop, dim)
svm_list <- list()
hc_svm_list <- list()
rmse_test <- c()
disharge_predict <- list()
hc_wt_predict <- list()
actal_weight_ptx <-list()
setwd("/Users/oter04/Documents/projects/prop/svm_permutations")
for(i in 1:1000)  {
  #  for( i in 1:2){
  # i = 2
  patients <- 1:length(prop)
  train_patients <- sample(patients, size = .7*length(patients), replace = FALSE
)
  test_patients <- patients[-train_patients]

  train_prop <- data.frame()
  for(j in 1:length(train_patients))  {
    #j = 2
    dfx_train <- prop[[train_patients[j]]]
    train_prop <- rbind(train_prop, dfx_train[,eg])
  }
  print("train_done")
  test_prop <- data.frame()
  for(k in 1:length(train_patients))  {
    #k = 2
    dfx_test <- prop[[train_patients[k]]]
    test_prop <- rbind(test_prop, dfx_test[, eg])
  }
  print("test_done")
  my_svm <- svm(GNMD_WEIGHT ~ corrected_pma + GNMD_VSHEAD + birth_weight +
                  Sex + Head_circumference, data = train_prop, na.action = na.om
it)
  svm_list[[i]] <- my_svm
  hc_svm_total <- svm(GNMD_VSHEAD~  corrected_pma +  birth_weight +
                        Sex + Head_circumference, data = train_prop, na.action =
na.omit)
  hc_svm_list[[i]] <- hc_svm_total
  testx <- na.omit(test_prop)
  rmse1 <- RMSE(testx$GNMD_WEIGHT, predict(my_svm, testx))
  rmse_test[i] <- rmse1

  predicted_wt_x <- c()
  discharge_predict_x <- c()
  actual_wt_xxx <- c()
  for(l in 1:nrow(u1)) {

    #  l = 1
    ptx <- u1[l,]
    my_days <- seq(from = ptx$gestation_at_birth[1], to =  ptx$corrected_pma[1]
, by = 0.035)
    new_df <- data.frame(
      "corrected_pma" = my_days,
      "birth_weight" = rep(ptx$birth_weight[1], times = length(my_days)),
      "Sex" = rep(ptx$Sex[1], times = length(my_days)),
      "Head_circumference"= rep(ptx$Head_circumference[1], times = length(my_day
s))
    )

    predicted_hc <- predict(hc_svm_total, new_df)
    new_df$GNMD_VSHEAD <- predicted_hc
    das_predict <- predict(my_svm, new_df)
    predicted_wt_x[l] <- das_predict[length(das_predict)]
    new_df$wt_predicted <- das_predict
    discharge_predict_x[l] <- predict(my_svm, ptx)
    actual_wt_xxx[l] <- ptx$GNMD_WEIGHT
    print(l)
  }



  print(i)

  disharge_predict[[i]] <- discharge_predict_x
  hc_wt_predict[[i]] <-   predicted_wt_x
  actal_weight_ptx[[i]] <-actual_wt_xxx
  saveRDS(svm_list, file  = "final_svm_list.R")
  saveRDS(hc_svm_list, file  = "hc_svm_list_list.R")
  saveRDS(rmse_test, file  = "rmse_test.R")
  saveRDS(disharge_predict, file  = "disharge_predict.R")
  saveRDS(hc_wt_predict, file  = "hc_wt_predict.R")
  saveRDS(actal_weight_ptx, file  = "actal_weight_ptx")
}
class(rmse_test)
mean(rmse_test)
sd(rmse_test)

final_df <- data.frame()
df_hc_p <- data.frame("hc_p"= rep(NA, times = length(hc_wt_predict[[1]])))
df_dc_p <- data.frame("dc_p"= rep(NA, times = length(hc_wt_predict[[1]])) )
dim(u1)
rownames(df_hc_p) <- u1$pt_id
rownames(df_dc_p) <- u1$pt_id
for(i in 1:length(hc_wt_predict)) {
  #i = 2
  hc_p <- hc_wt_predict[[i]]
  dc_p <- disharge_predict[[i]]
  df_hc_p <- cbind(df_hc_p, hc_p)
  df_dc_p <- cbind(df_dc_p, dc_p)
  print(i)

}
head(df_hc_p)
hcx <- apply(df_hc_p[,2:ncol(df_hc_p)], 1, mean)
dcx <- apply(df_dc_p[,2:ncol(df_dc_p)], 1, mean)
final_df <- data.frame("Actual_dc_wt" = actal_weight_ptx[[1]],
                       "Discharge_predict" = dcx,
                       "HC_predict" = hcx)



View(final_df)
getwd()
write.csv(final_df, "df_prop_over800Permutatins.csv")
dim(df_hc_p)
all.equal(names(hcx), names(dcx))




#####looking at the Uruguay data
rmse_test <- readRDS("...")
summary(rmse_test)
urug1 <- read.csv("...")
ur800 <- read.csv("...")
names(urug1)
names(ur800)
urug1$pt_id
ur800$pt_id <- ur800$X
```

```
library(dplyr)
dfx <- left_join(ur800, urug1,  by = "pt_id")
names(dfx)

plot(dfx$Discharge_predict, dfx$HC_predict)
plot(dfx$Actual_dc_wt, dfx$Discharge_predict, col= "red")
points( dfx$HC_predict, dfx$Discharge_predict)

summary(as.factor(dfx$SEPSIS))
(names(dfx))
dfx2 <- dfx[,c("BDP"  , "Actual_dc_wt", "Discharge_predict",
               "HC_predict"  , "pt_id"      )] %>% na.omit
names(dfx2)
summary(dfx2)
dfx2$BDP <- as.factor(dfx2$BDP)
levels(dfx2$BPD)
plot(dfx$Actual_dc_wt, dfx$HC_predict,ylim = c(500,5000),
     xlim = c(500, 5000),
     col = ifelse(dfx2$BDP == 1, "red2", "blue"))
lines(dfx$HC_predict, dfx$HC_predict)
?text
text(labels = dfx$pt_id, y = dfx$HC_predict,
     x = dfx$Actual_dc_wt,     cex = 0.3)

dfx2$diff <- dfx2$Actual_dc_wt - dfx2$HC_predict
dfx2$SEPSIS <- as.factor(dfx2$SEPSIS)
summary(dfx2$SEPSIS)
randomForest(SEPSIS ~ Actual_dc_wt+diff, data = dfx2)

library(caret)
RMSE(pred = dfx2$Discharge_predict, obs = dfx2$Actual_dc_wt)
RMSE(pred = dfx2$HC_predict, obs = dfx2$Actual_dc_wt)
rmse_uruguay <- c()
for(i in 1:nrow(dfx)) {
  rmse_uruguay[i] <- RMSE(pred = dfx2$Discharge_predict[i], obs = dfx2$Actual_dc_wt[i])
  print(i)
}
plot(density(rmse_uruguay))
lines(density(rmse_test))
library(ggplot2)
rmse_df <- data.frame("prop_rmse"  = rmse_test,
                      "uruguay_rmse"  = append(
                        rep(NA, times = 831-44) ,rmse_uruguay))

density(rmse_test) -> b
as.vector(b)
b$x
density(rmse_uruguay) -> u
all.equal(b$x, rmse_test)
```

```
?density
summary(u$y)
summary(b$x)
summary(u$x)
plot(b, xlim = c(-250, 1500))
dev.off()
par(mar= (4,4,4,4))
hist(rmse_uruguay, freq = FALSE)
lines(u)
prop_hist <- hist(rmse_test, freq = FALSE)
u_hist <- hist(rmse_uruguay, freq = FALSE)
plot(prop_hist)
plot(u_hist)
plot(u_hist, add = T)
?scale_y_continuous
?ggbreak
length(b$y)
summary(rmse_uruguay)
plot(y = rmse_uruguay, x = dfx2$Actual_dc_wt)
points(y = rmse_test, x = )
new_df <- data.frame("Y" = b$y, "X" = b$x,
                     "Study_Site" = rep("PROP", times = length(b$x)))
new_df <- rbind(new_df, data.frame("Y" = u$y,
                                   "X" = u$x,
                                   "Study_Site" =
                                     rep("Uruguay", times = length(u$x))))
new_df$Study_Site <- as.factor(new_df$Study_Site)
summary(new_df)
names(dfx2)
head(rmse2)
tail(rmse2)
names(rmse2)
?geom_jitter
pp<-ggplot(data = rmse2, aes(y = values,x =  ind, fill = ind))
my_pp <- pp+geom_violin() +geom_jitter(fill = "blue", alpha = 0.1)+theme_bw()+
  ylab("RMSE")

getwd()
setwd("/Users/oter04/Documents/projects/prop/Manuscript")
pdf(file = "rmse_isse.pdf")
print(my_pp)
dev.off()
length(rmse_uruguay)
boxplot(values ~ ind, data = rmse2)

?hist
bb <- (dfx$Actual_dc_wt-  dfx2$Discharge_predict) %>% hist
pdf(file = "hist_of_diff.pdf")
hist((dfx$Actual_dc_wt-  dfx2$Discharge_predict),
     col = "green4", xlab= "Actual Discharge Weight - Predicted Weight",
     main = "PROP-based Models Overpredict Weight Gain",
     las = 1, font.axis= 2, font.lab  = 2)
dev.off()
```

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]: