

Is Cycle Threshold A Worthwhile Clinical Metric for SARS-CoV2?

Andrew D. Johnston

08/31/2020

Contents

Goals of Analysis

- 1) Calculate the Ct value difference between Expired and Survivors for each Machine - Cepheid, Hologic, and Abbott
- 2) Calculate the Youden index for each Machine - Cepheid, Hologic, and Abbott
- 3) Examine the variability of samples taken within 48 hours of each other.
- 4) Observe how many specimen pairs cross the Youden Index for their respective machine

```
# load in libraries
library(data.table)
library(ggplot2)
library(ggthemes)
library(reshape2)
library(ggpubr)
library(RColorBrewer)
library(MASS)
library(AER)
library(scales)
library(Hmisc)
library("flexplot")
library(overlapping)
library(lattice)
library(lubridate)
library(ROCit)
library(pROC)
library(plotROC)
library(OptimalCutpoints)
library(cutpointr)
library(pastecs)

# set options
options(scipen = 999, stringsAsFactors = FALSE)
```

Loading in the mortatlity data for each COVID specimen

```
mort_dat <- fread("Data_used/COVID_POS_SURVIVAL_short.csv", header = T)
mort_dat <- as.data.frame(mort_dat)
dim(mort_dat)
```

```
## [1] 9646    5
```

```
mort_dat$Alive <- "Survivor"
mort_dat$Alive[mort_dat$Patient_Age == "Deceased"] <- "Expired"
mort_dat$Alive <- factor(mort_dat$Alive, levels = c("Survivor", "Expired"))
```

1a) Obtaining the CT differences between expired and survivor for Cephied Machine

```
cephied_dat <- fread("Data_used/sarsCoV2-cephied_ct.csv", header = T)
cephied_dat <- as.data.frame(cephied_dat)
cephied_dat <- cephied_dat[1:11490, ] # R read in all the empty rows
cephied_dat <- cephied_dat[, -c(4, 6, 8)] # R read in all the empty rows
colnames(cephied_dat) <- c("MRN", "Spec_ID", "Result", "Ct_E", "Ct_n", "Ct_Control",
  "start_time")
dim(cephied_dat)
```

```
## [1] 11490      7
```

```
# any with same spec ID - QC
any(duplicated(cephied_dat$Spec_ID))
```

```
## [1] TRUE
```

```
# clear all negative and error and
filter_cephied <- c("NEGATIVE", "ERROR", "INVALID", "NO RESULT")
idx_cephied_filter <- grep(paste0(filter_cephied, collapse = "|"), x = cephied_dat$Result,
  value = FALSE)
length(idx_cephied_filter) #9589
```

```
## [1] 9589
```

```
cephied_dat <- cephied_dat[-idx_cephied_filter, ]
table(cephied_dat$Result)
```

```
##
##          SARS-CoV-2 POSITIVE SARS-CoV-2 PRESUMPTIVE POS
##                1828                73
```

```
# there are 11 samples in which the control probe did not result but they were
# deemed positive anyways, so I'll keep them
# cephied_dat[which(cephied_dat$Ct_Control==0),]
# head(cephied_dat[cephied_dat$Result=='SARS-CoV-2 PRESUMPTIVE POS'],50)
# head(cephied_dat[which(cephied_dat$Ct_E==0),]) the presumptive positives were
# when E probe was positive but N probe was undetected removing Ct_N value of 0
cephied_dat <- cephied_dat[-which(cephied_dat$Ct_n == 0), ]
dim(cephied_dat) # 1828
```

```
## [1] 1828      7
```

```
cephied_dat_mort <- merge(x = cephied_dat, y = mort_dat, by.x = "Spec_ID", by.y = "Specimen ID")
dim(cephied_dat_mort) # lost two in the merge; 1826
```

```
## [1] 1826     12
```

```
wilcox.test(cephied_dat_mort$Ct_n[cephied_dat_mort$Alive == "Survivor"], cephied_dat_mort$Ct_n[cephied_dat_mort$Alive == "Expired"], conf.int = TRUE)
```

```
##
```

```
## Wilcoxon rank sum test with continuity correction
```

```
##
```

```
## data: cephied_dat_mort$Ct_n[cephied_dat_mort$Alive == "Survivor"] and cephied_dat_mort$Ct_n[cephied_dat_mort$Alive == "Expired"]
```

```
## W = 341890, p-value < 2.2e-16
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
## 4.300053 6.299982
## sample estimates:
## difference in location
## 5.299954

# 95 percent confidence interval: 4.300053 6.299982 difference in location
# 5.299954

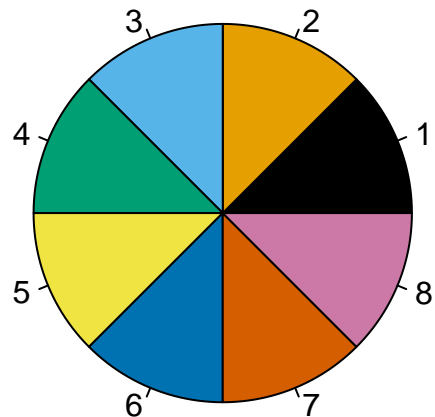
median(cephied_dat_mort$Ct_n[cephied_dat_mort$Alive == "Survivor"]) - median(cephied_dat_mort$Ct_n[cephied_dat_mort$Alive == "Expired"])

## [1] 6.8

mean(cephied_dat_mort$Ct_n[cephied_dat_mort$Alive == "Survivor"]) - mean(cephied_dat_mort$Ct_n[cephied_dat_mort$Alive == "Expired"])

## [1] 4.938142

colorBlindBlack8 <- c("#000000", "#E69F00", "#56B4E9", "#009E73", "#F0E442", "#0072B2", "#D55E00", "#CC79A7")
pie(rep(1, 8), col = colorBlindBlack8)
```

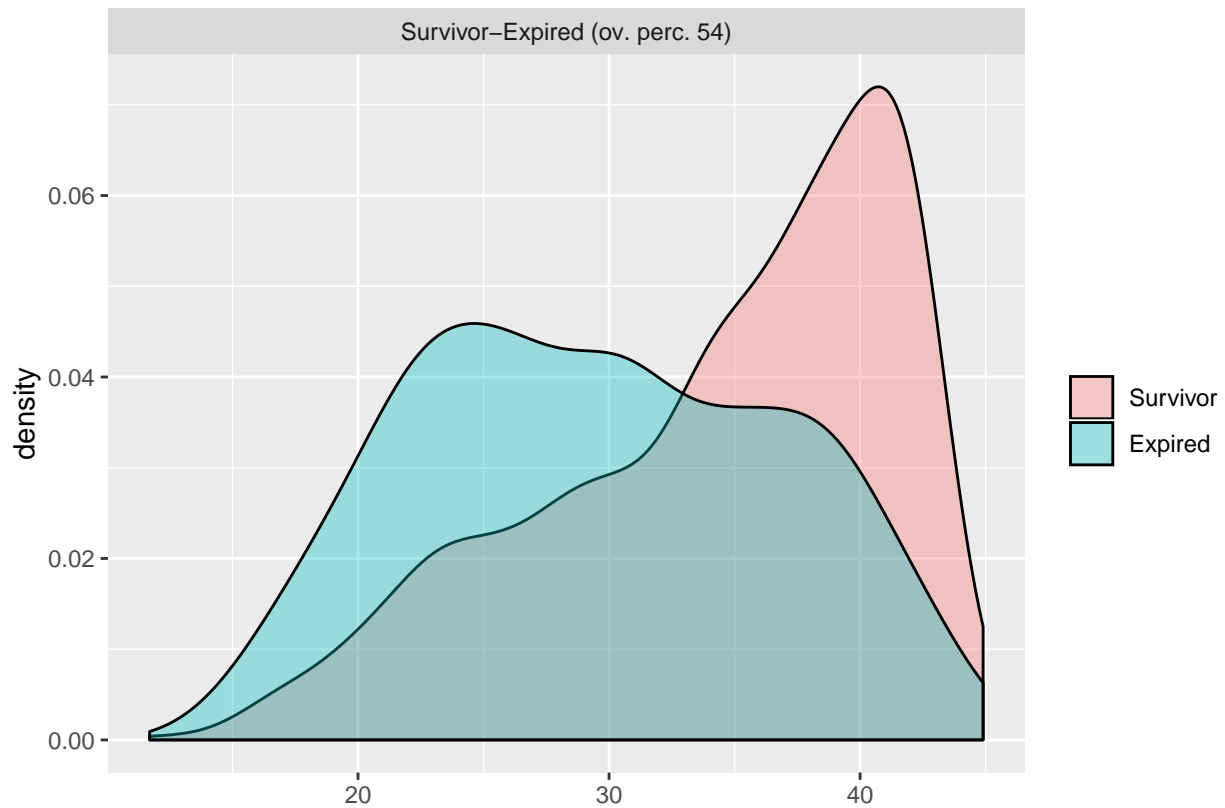


```
dark_col <- brewer.pal(6, "Dark2")
fig_ct_method_cephied <- ggplot(cephied_dat_mort, aes(x = Alive, y = Ct_n, fill = Alive)) +
  geom_dotplot(binaxis = "y", stackdir = "center", stackratio = 0.5, dotsize = 0.25,
    col = NA, position = position_jitterd(width = 0.05, height = 1, quad.points = 1,
      seed = 7)) + theme_minimal() + scale_fill_manual(values = colorBlindBlack8[c(4,
6)]) + stat_summary(data = cephied_dat_mort, mapping = aes(x = as.numeric(as.factor((Alive))),
  y = Ct_n), fun.y = "median", geom = "point", color = colorBlindBlack8[7], inherit.aes = FALSE,
  shape = 95, size = 20) + xlab("") + ylab("Ct Value") + ylim(c(0, 50))
fig_ct_method_cephied
```



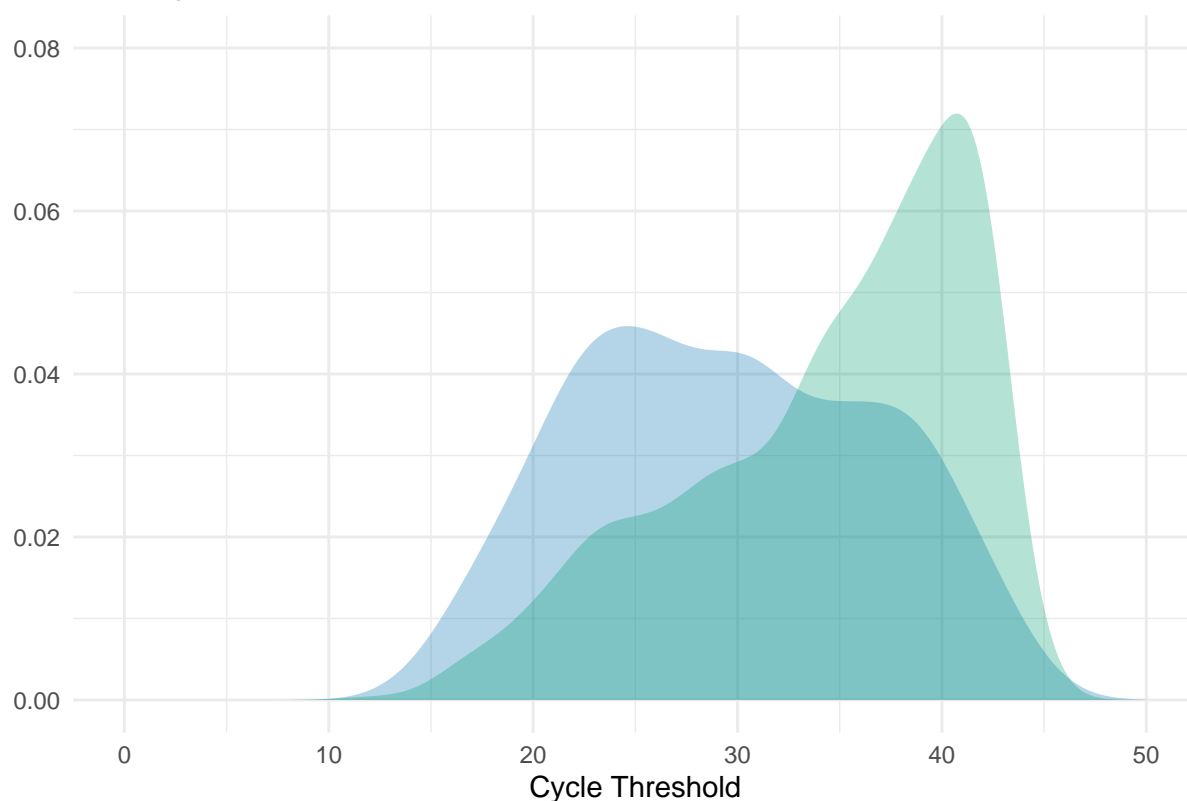
```
# ggsave(plot = fig_ct_method_cephied, filename = 'Fig1_cephied.pdf', width=4.5,
# height=3, useDingbats=FALSE)

# plot AUC overlap between survivor and expired
AUC_overlap_cephied <- list(Survivor = cephied_dat_mort$Ct_n[cephied_dat_mort$Alive ==
  "Survivor"], Expired = cephied_dat_mort$Ct_n[cephied_dat_mort$Alive == "Expired"])
AUC_overlap_plot_cephied <- overlap(AUC_overlap_cephied, plot = TRUE, theme = theme_minimal)
```



```
fig_ct_overlap_cephied <- ggplot(AUC_overlap_plot_cephied$DD, aes(x, y1)) + geom_ribbon(aes(ymin = 0,
ymax = y2), alpha = 0.3, fill = colorBlindBlack8[c(6)])) + geom_ribbon(aes(ymin = 0,
ymax = y1), alpha = 0.3, fill = colorBlindBlack8[c(4)])) + xlab("Cycle Threshold") +
xlim(c(0, 50)) + ylab("") + theme_minimal() + ylim(c(0, 0.08)) + ggtitle(paste0("Overlap = ",
round(AUC_overlap_plot_cephied$OV * 100, 2), "%"))
fig_ct_overlap_cephied
```

Overlap = 53.99%



```
# ggsave(plot = fig_ct_overlap_cephied, filename = 'Fig1_overlap_cephied.pdf',
# width=2.5, height=2.5, useDingbats=FALSE)
```

2a) Obtaining the Youden Index for the Cepheid machine

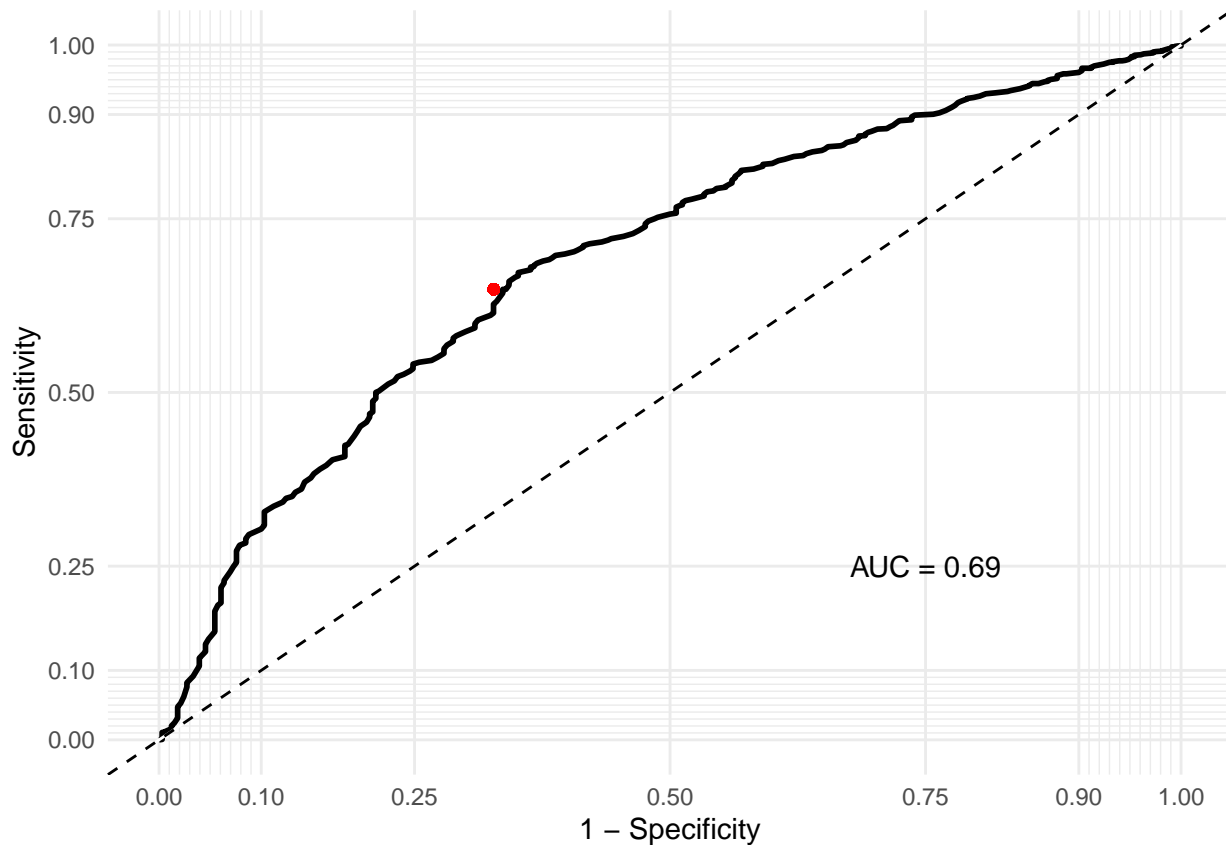
```
rocfit_cepheid <- pROC::roc(cephied_dat_mort$Alive, cepheid_dat_mort$Ct_n)
pROC::auc(rocfit_cepheid)
```

```
## Area under the curve: 0.6925
```

```
youden_coord_cepheid <- pROC::coords(rocfit_cepheid, x = "b")
summary(cephied_dat_mort$Ct_n)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      11.70  28.10   35.10   33.55  39.70   44.90
```

```
rocplot_cepheid <- ggplot(cephied_dat_mort, aes(m = Ct_n, d = Alive)) + geom_roc(n.cuts = 0)
rocplot_cepheid <- rocplot_cepheid + style_roc(theme = theme_minimal, xlab = "1 - Specificity",
ylab = "Sensitivity") + annotate("text", x = 0.75, y = 0.25, label = paste("AUC =",
round(calc_auc(rocplot_cepheid)$AUC, 2))) + geom_point(aes(x = 1 - youden_coord_cepheid$specificity,
y = youden_coord_cepheid$sensitivity), colour = "red") + geom_abline(slope = 1,
intercept = 0, lty = "dashed")
# ggtitle('Cepheid ROC')
rocplot_cepheid
```



```
# ggsave(plot = rocplot_cepheid, filename = 'Fig1_roc_cepheid.pdf', width=2.5,
# height=2.5, useDingbats=FALSE)
```

1b) Obtaining the CT differences between expired and survivor for Hologic Machine

```
hologic_dat <- fread("Copy of pcrbase.csv", header = T)
hologic_dat <- as.data.frame(hologic_dat)
dim(hologic_dat) #16732
```

```
## [1] 16732      9
```

```
# sanity check- duplicated specimen IDs?
any(duplicated(hologic_dat$Spec_name))
```

```
## [1] TRUE
```

```
sum(duplicated(hologic_dat$Spec_name)) #412
```

```
## [1] 412
```

```
# removing duplicates
hologic_dat <- hologic_dat[!duplicated(hologic_dat$Spec_name), ]

# sort based on MRN then time collected
hologic_dat <- hologic_dat[order(hologic_dat$mrn), ]

# making MRN numeric
hologic_dat$mrn <- as.numeric(hologic_dat$mrn)

# remove the hologi as well as NA MRN patients
```

```

hologic_dat <- hologic_dat[-c(1:2), ]
sum(is.na(hologic_dat$mrn)) #130

## [1] 130

hologic_dat <- hologic_dat[!is.na(hologic_dat$mrn), ]

# removing negative tests
table(hologic_dat$Result)

##
##      neg      POS
## 13331  2857

hologic_dat <- hologic_dat[hologic_dat$Result == "POS", ]

# QC
table(hologic_dat$IC_valid)

##
## Valid
##  2857

summary(hologic_dat$Control_Ct)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.     NA's
##    29.30  30.70   31.00   31.12  31.40   35.80         4

dim(hologic_dat)

## [1] 2857      9

hologic_dat_mort <- merge(x = hologic_dat, y = mort_dat, by.x = "Spec_name", by.y = "Specimen ID")
dim(hologic_dat_mort)

## [1] 2857     14

wilcox.test(hologic_dat_mort$Ct[hologic_dat_mort$Alive == "Survivor"], hologic_dat_mort$Ct[hologic_dat_mort$Alive == "Expired"], conf.int = TRUE)

##
## Wilcoxon rank sum test with continuity correction
##
## data:  hologic_dat_mort$Ct[hologic_dat_mort$Alive == "Survivor"] and hologic_dat_mort$Ct[hologic_dat_mort$Alive == "Expired"]
## W = 690220, p-value < 0.00000000000000022
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  3.499936 5.000038
## sample estimates:
## difference in location
##                4.299973
##
## 95 percent confidence interval: 3.499936 5.000038 sample estimates: difference
## in location 4.299973

fig_ct_method_hologic <- ggplot(hologic_dat_mort, aes(x = Alive, y = Ct, fill = Alive)) +
  geom_dotplot(binaxis = "y", stackdir = "center", stackratio = 0.5, dotsize = 0.17,
    col = NA, position = position_jitterd(width = 0.05, height = 1, quad.points = 1,
      seed = 7)) + theme_minimal() + scale_fill_manual(values = colorBlindBlack8[c(4,

```

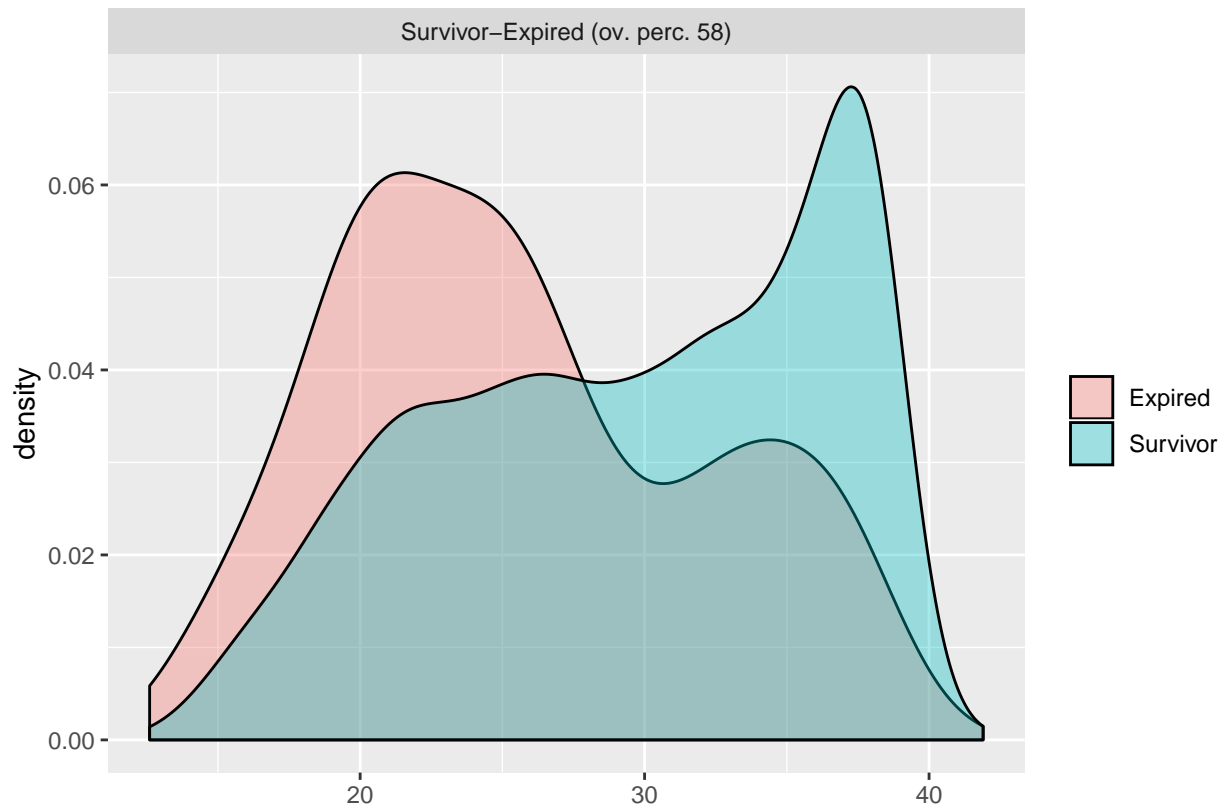


```
6)]) + stat_summary(data = hologic_dat_mort, mapping = aes(x = as.numeric(as.factor((Alive))),
y = Ct), fun.y = "median", geom = "point", color = colorBlindBlack8[7], inherit.aes = FALSE,
shape = 95, size = 20) + xlab("") + ylab("Ct Value") + ylim(c(0, 50))
fig_ct_method_hologic
```



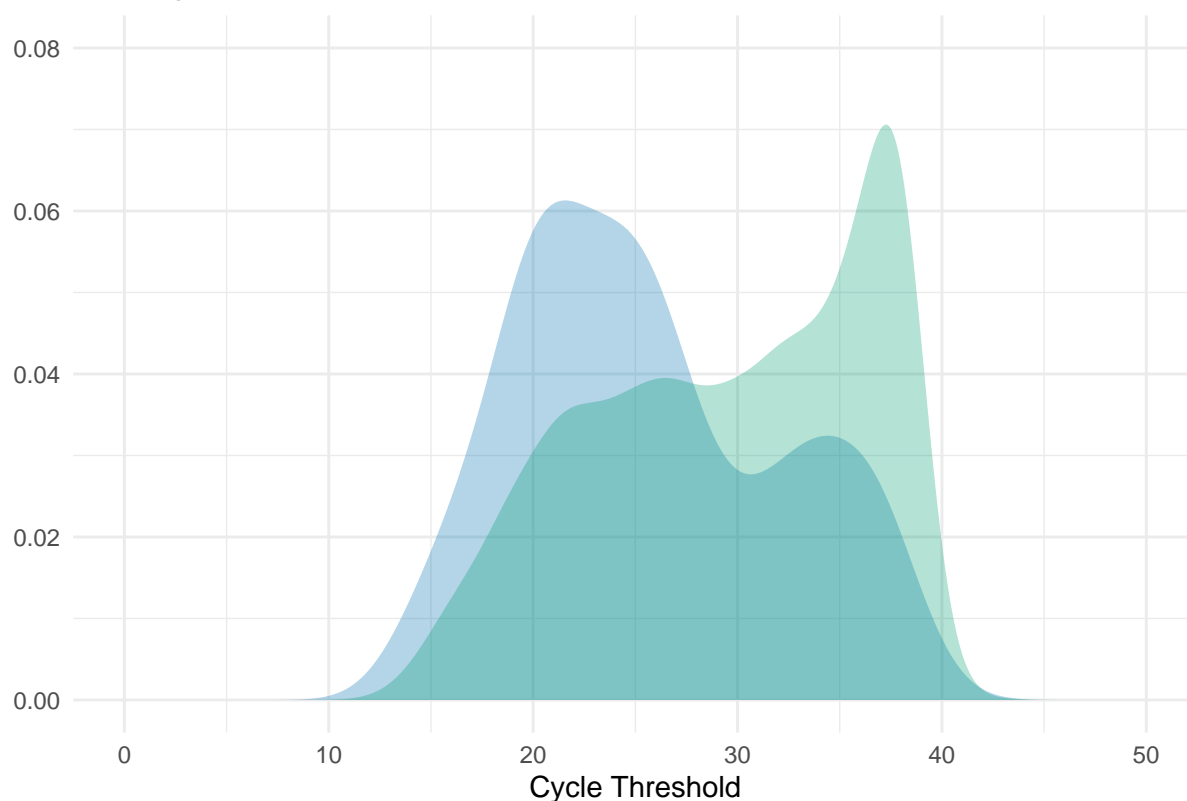
```
# ggsave(plot = fig_ct_method_hologic, filename = 'Fig1_hologic.pdf', width=4.5,
# height=3, useDingbats=FALSE)

# plot AUC overlap between survivor and expired
AUC_overlap_hologic <- list(Survivor = hologic_dat_mort$Ct[hologic_dat_mort$Alive ==
"Survivor"], Expired = hologic_dat_mort$Ct[hologic_dat_mort$Alive == "Expired"])
AUC_overlap_plot_hologic <- overlap(AUC_overlap_hologic, plot = TRUE)
```



```
fig_ct_overlap_hologic <- ggplot(AUC_overlap_plot_hologic$DD, aes(x, y1)) + geom_ribbon(aes(ymin = 0,
ymax = y2), alpha = 0.3, fill = colorBlindBlack8[c(6)]) + geom_ribbon(aes(ymin = 0,
ymax = y1), alpha = 0.3, fill = colorBlindBlack8[c(4)]) + xlab("Cycle Threshold") +
xlim(c(0, 50)) + ylab("") + ylim(c(0, 0.08)) + theme_minimal() + ggtitle(paste0("Overlap = ",
round(AUC_overlap_plot_hologic$OV * 100, 2), "%"))
fig_ct_overlap_hologic
```

Overlap = 58.44%



```
# ggsave(plot = fig_ct_overlap_hologic, filename = 'Fig1_overlap_hologic.pdf',
# width=2.5, height=2.5, useDingbats=FALSE)
```

2b) Obtaining the Youden Index for the Hologic machine

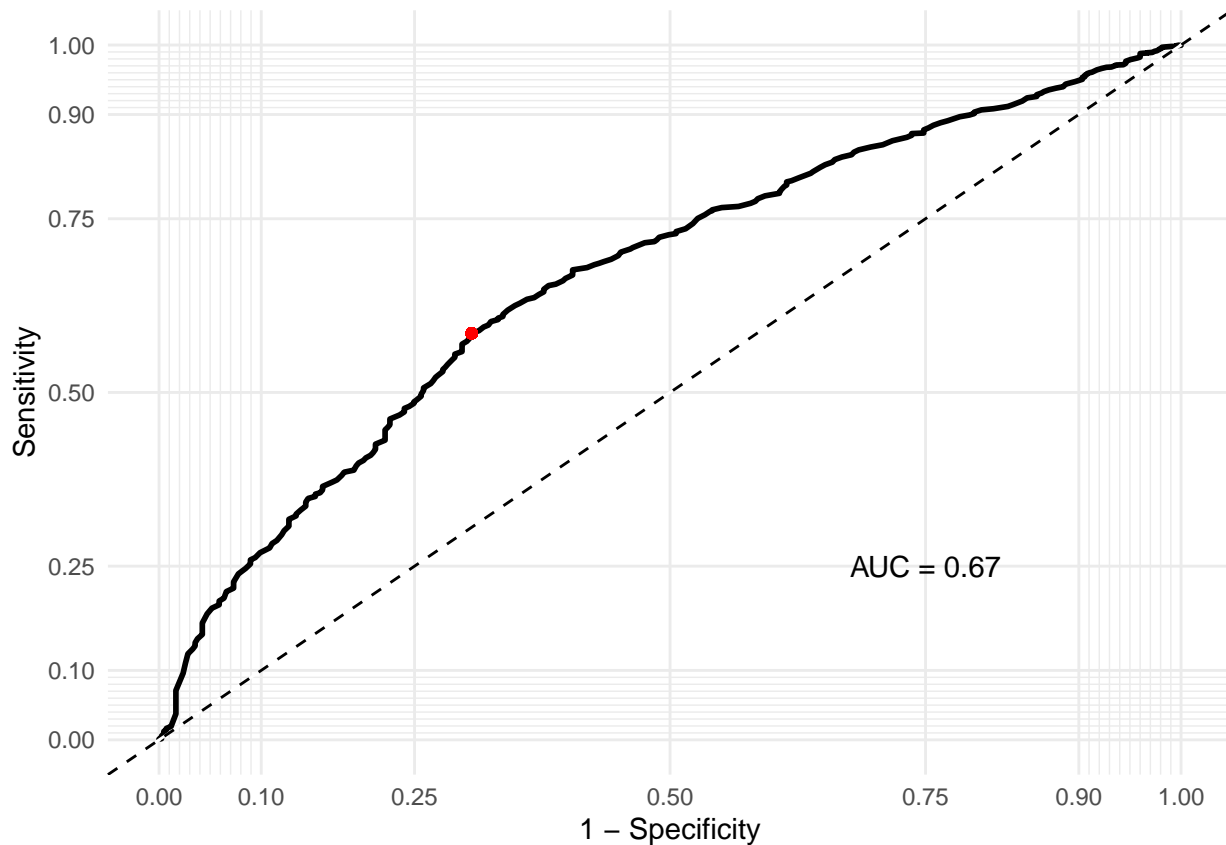
```
rocfit_hologic <- pROC::roc(as.character(hologic_dat_mort$Alive), hologic_dat_mort$Ct)
pROC::auc(rocfit_hologic)
```

```
## Area under the curve: 0.6678
```

```
youden_coord_hologic <- pROC::coords(rocfit_hologic, x = "b")
youden_coord_hologic
```

```
## threshold specificity sensitivity
## 1 28.25 0.6941176 0.5851151
```

```
rocplot_hologic <- ggplot(hologic_dat_mort, aes(m = Ct, d = Alive)) + geom_roc(n.cuts = 0)
rocplot_hologic <- rocplot_hologic + style_roc(theme = theme_minimal, xlab = "1 - Specificity",
ylab = "Sensitivity") + annotate("text", x = 0.75, y = 0.25, label = paste("AUC =",
round(calc_auc(rocplot_hologic)$AUC, 2))) + geom_point(aes(x = 1 - youden_coord_hologic$specificity,
y = youden_coord_hologic$sensitivity), colour = "red") + geom_abline(slope = 1,
intercept = 0, lty = "dashed")
# ggtitle('Hologic ROC')
rocplot_hologic
```



```
# ggsave(plot = rocplot_hologic, filename = 'Fig1_roc_hologic.pdf', width=2.5,
# height=2.5, useDingbats=FALSE)
```

1c) Obtaining the CT differences between expired and survivor for Hologic Machine

```
abbott_dat <- fread("Ct_study_race.csv", header = T)
abbott_dat <- as.data.frame(abbott_dat)
dim(abbott_dat)
```

```
## [1] 2038 27
```

```
# sanity checks are there duplicates? No
any(duplicated(abbott_dat$MR)) # FALSE
```

```
## [1] FALSE
```

```
any(duplicated(abbott_dat$`Specimen ID`)) # FALSE
```

```
## [1] FALSE
```

```
# how many samples don't have a CT value
sum(is.na(abbott_dat$Ct)) # 2
```

```
## [1] 2
```

```
# How many samples have ct of 0?
sum(abbott_dat$Ct == 0, na.rm = TRUE) # 1
```

```
## [1] 1
```

```
# I'm removing these four samples
abbott_dat <- abbot_dat[!(is.na(abbott_dat$Ct)) & !(abbott_dat$Ct == 0), ]
```

```

dim(abbott_dat)

## [1] 2035    27

# remove unnecessary columns (Patient Name, Component, Column 3, method)
colnames(abbott_dat)

## [1] "Specimen ID"      "MR"      "Patient Name"
## [4] "Component"        "Race"     "Ethnicity"
## [7] "Method"           "Method Abbr" "Value"
## [10] "Resulted"         "Collected" "Patient Age"
## [13] "Gender"           "DOB"       "Ordering Provider"
## [16] "Age at Collection" "Collection Department" "Alive"
## [19] "Specimen Id2"     "Ct"        "Column1"
## [22] "D-Dimer"          "LDH"       "WBC"
## [25] "CRP"             "Creat"     "Column3"

abbott_dat <- abbot_dat[, -c(3:4, 7, 21:27)]

# only using Abbott data points
abbott_dat <- subset(abbott_dat, `Method Abbr` == "m2000")
dim(abbott_dat)

## [1] 1190    17

# sort based on MRN then time collected
abbott_dat <- abbot_dat[order(abbott_dat$MR), ]

# making MRN numeric
abbott_dat$MR <- as.numeric(abbott_dat$MR)
sum(is.na(abbott_dat$mrn))

## [1] 0

table(abbott_dat$Value)

##
## DETECTED
##      1190

summary(abbott_dat$Ct)

##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##      1.00   8.88   13.58   13.82   18.01   31.04

# combining with mortality data
dim(abbott_dat)

## [1] 1190    17

abbott_dat_mort <- merge(x = abbot_dat, y = mort_dat, by.x = "Specimen ID", by.y = "Specimen ID")
dim(abbott_dat_mort) # all were kept :-)

## [1] 1190    22

colnames(abbott_dat_mort)[22] <- "Alive"

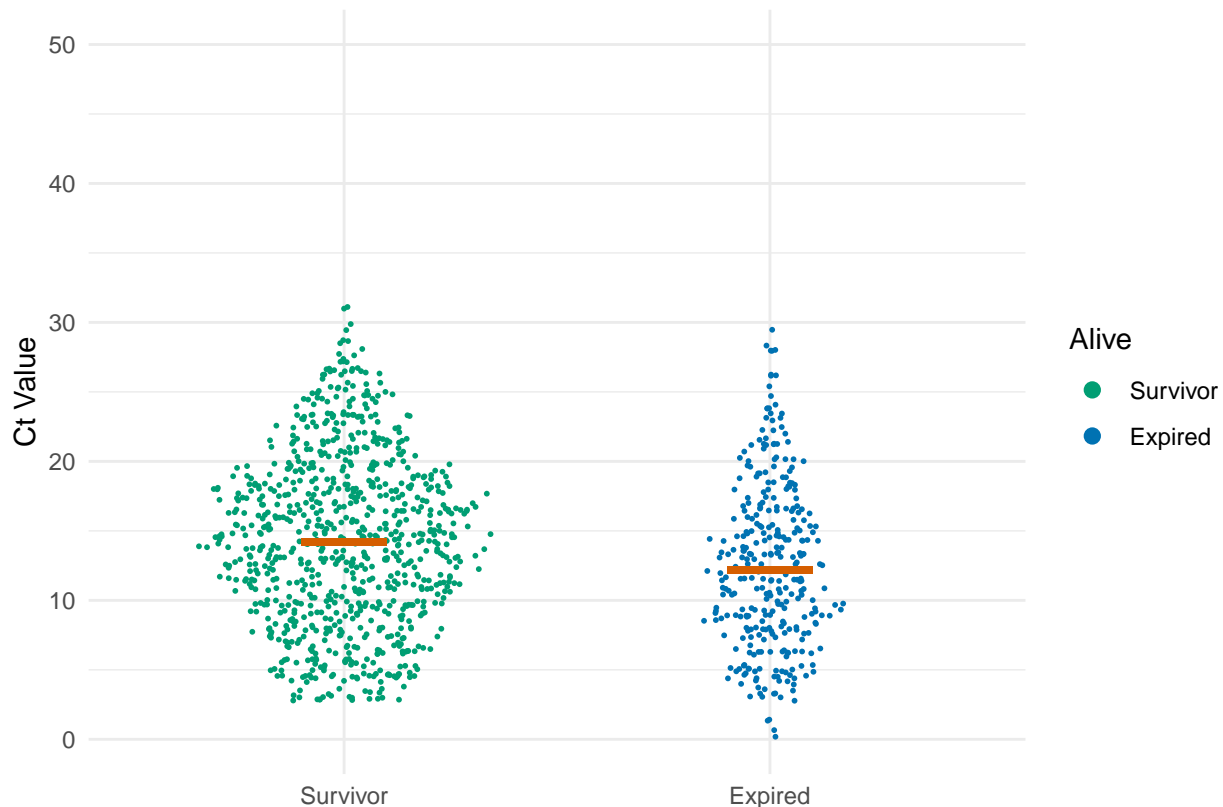
wilcox.test(abbott_dat_mort$Ct[abbott_dat_mort$Alive == "Survivor"], abbot_dat_mort$Ct[abbott_dat_mort$
  "Expired"], conf.int = TRUE)

```

```
##
## Wilcoxon rank sum test with continuity correction
##
## data:  abbottdat_mort$Ct[abbottdat_mort$Alive == "Survivor"] and abbottdat_mort$Ct[abbottdat_mort$Alive == "Expired"]
## W = 164050, p-value = 0.000006747
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  1.059992 2.669944
## sample estimates:
## difference in location
##                1.859981
```

```
# 95 percent confidence interval: 1.059992 2.669944 sample estimates: difference
# in location 1.859981
```

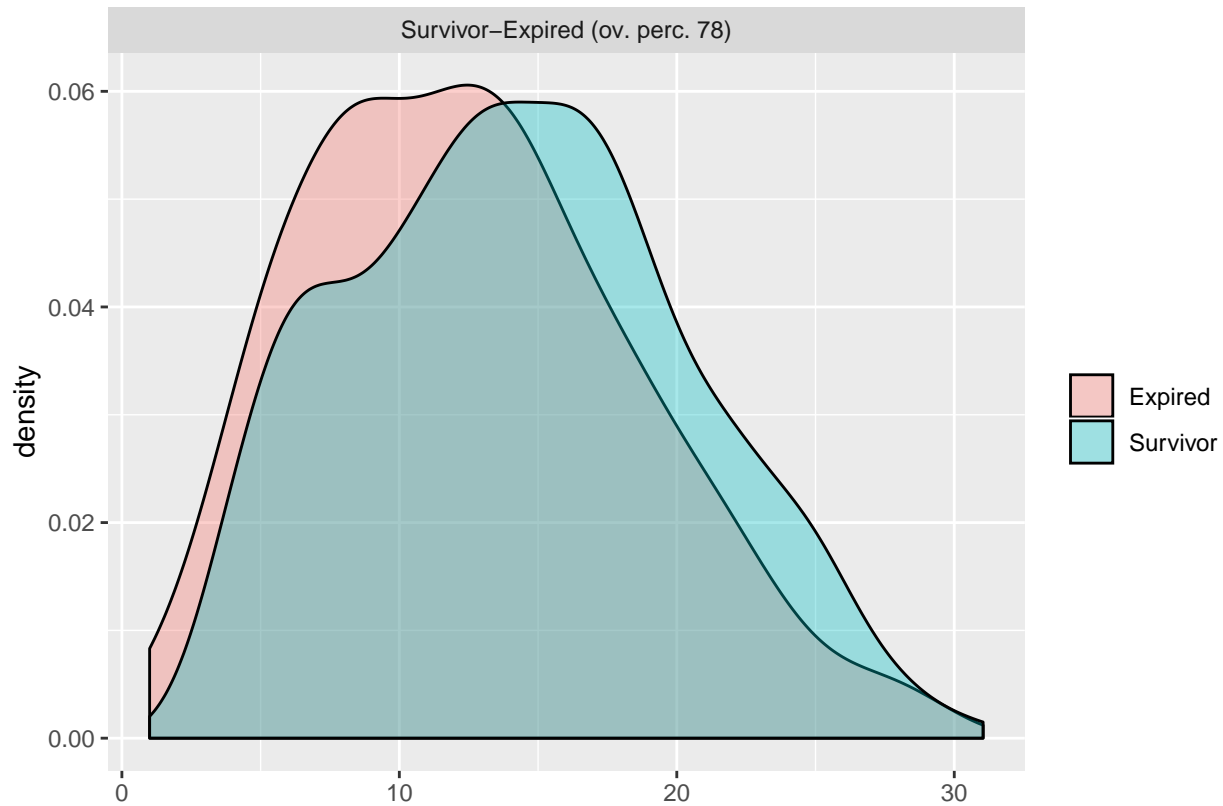
```
fig_ct_method_abbottdat_mort <- ggplot(abbottdat_mort, aes(x = Alive, y = Ct, fill = Alive)) +
  geom_dotplot(binaxis = "y", stackdir = "center", stackratio = 0.5, dotsize = 0.25,
    col = NA, position = position_jitterd(width = 0.05, height = 1, quad.points = 1,
      seed = 7)) + theme_minimal() + scale_fill_manual(values = colorBlindBlack8[c(4,
    6)]) + stat_summary(data = abbottdat_mort, mapping = aes(x = as.numeric(as.factor((Alive)))),
    y = Ct), fun.y = "median", geom = "point", color = colorBlindBlack8[7], inherit.aes = FALSE,
    shape = 95, size = 20) + xlab("") + ylab("Ct Value") + ylim(c(0, 50))
fig_ct_method_abbottdat_mort
```



```
# ggsave(plot = fig_ct_method_abbottdat_mort, filename = 'Fig1_abbottdat_mort.pdf', width=4.5,
# height=3, useDingbats=FALSE)
```

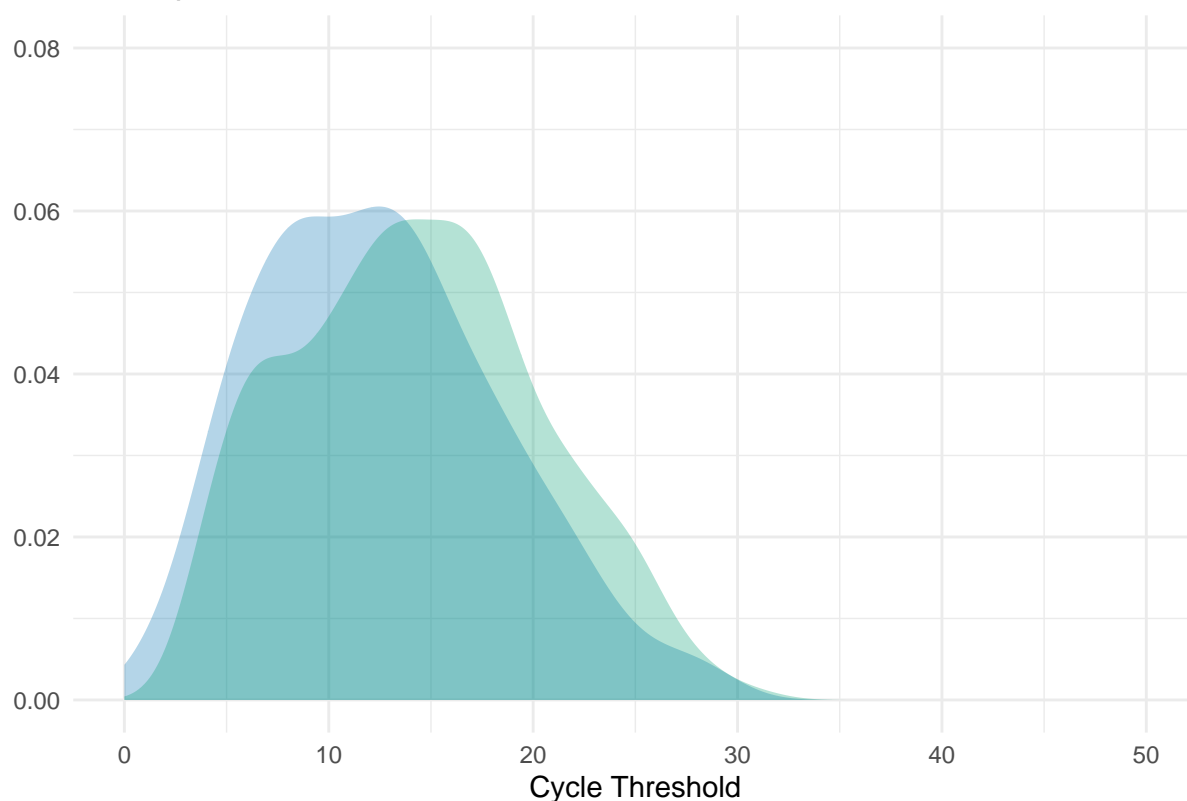
```
# plot AUC overlap between survivor and expired
```

```
AUC_overlap_abbott <- list(Survivor = abbot_dat_mort$Ct[abbott_dat_mort$Alive ==
  "Survivor"], Expired = abbot_dat_mort$Ct[abbott_dat_mort$Alive == "Expired"])
AUC_overlap_plot_abbott <- overlap(AUC_overlap_abbott, plot = TRUE, theme = theme_minimal)
```



```
fig_ct_overlap_abbott <- ggplot(AUC_overlap_plot_abbott$DD, aes(x, y1)) + geom_ribbon(aes(ymin = 0,
  ymax = y2), alpha = 0.3, fill = colorBlindBlack8[c(6)]) + geom_ribbon(aes(ymin = 0,
  ymax = y1), alpha = 0.3, fill = colorBlindBlack8[c(4)]) + xlab("Cycle Threshold") +
  xlim(c(0, 50)) + ylab("") + ylim(c(0, 0.08)) + theme_minimal() + ggtitle(paste0("Overlap = ",
  round(AUC_overlap_plot_abbott$OV * 100, 2), "%"))
fig_ct_overlap_abbott
```

Overlap = 77.68%



```
# ggsave(plot = fig_ct_overlap_abbott, filename = 'Fig1_overlap_abbott.pdf',
# width=2.5, height=2.5, useDingbats=FALSE)
```

2c) Obtaining the Youden Index for the Abbott machine

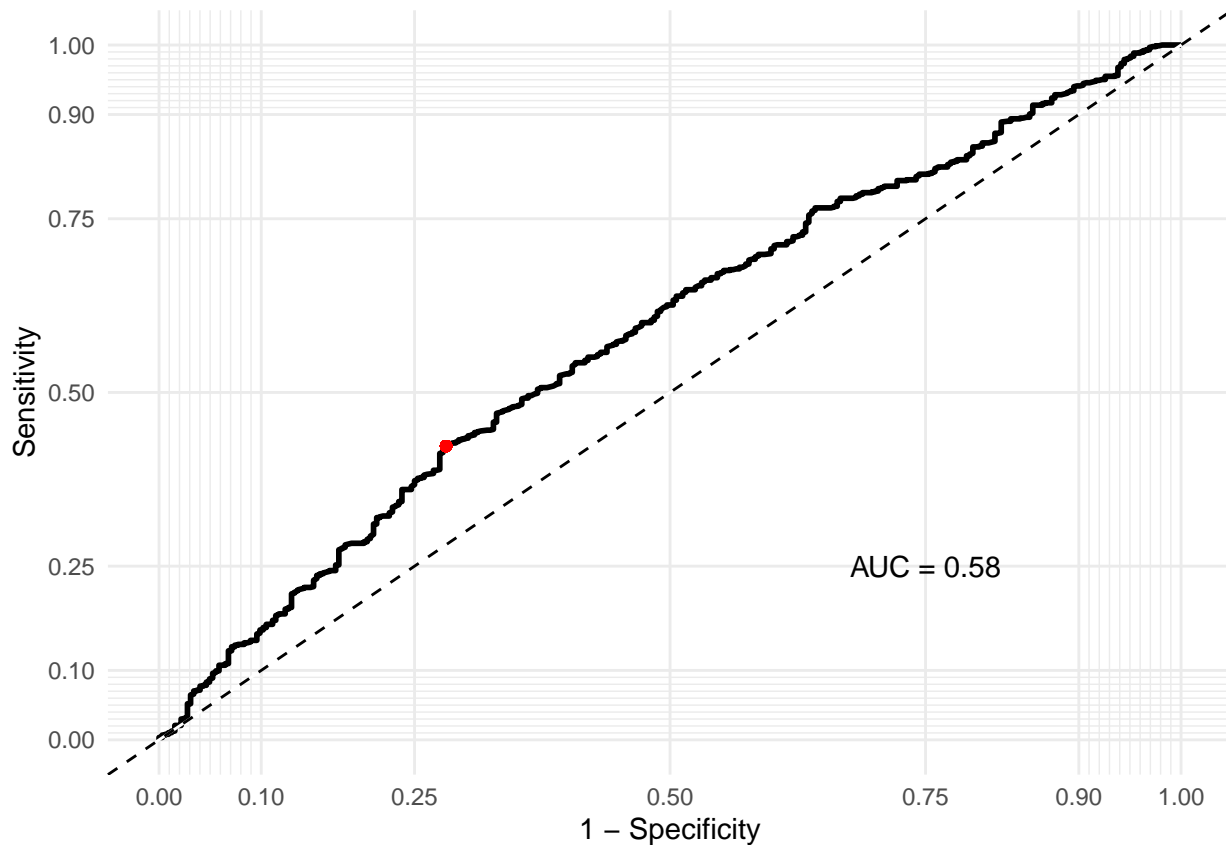
```
rocfit_abbott <- pROC::roc(as.character(abbott_dat_mort$Alive), abbott_dat_mort$Ct)
pROC::auc(rocfit_abbott)
```

```
## Area under the curve: 0.5847
```

```
youden_coord_abbott <- pROC::coords(rocfit_abbott, x = "b")
youden_coord_abbott
```

```
## threshold specificity sensitivity
## 1 15.62 0.7191358 0.4226328
```

```
rocplot_abbott <- ggplot(abbott_dat_mort, aes(m = Ct, d = Alive)) + geom_roc(n.cuts = 0)
rocplot_abbott <- rocplot_abbott + style_roc(theme = theme_minimal, xlab = "1 - Specificity",
ylab = "Sensitivity") + annotate("text", x = 0.75, y = 0.25, label = paste("AUC =",
round(calc_auc(rocplot_abbott)$AUC, 2))) + geom_point(aes(x = 1 - youden_coord_abbott$specificity,
y = youden_coord_abbott$sensitivity), colour = "red") + geom_abline(slope = 1,
intercept = 0, lty = "dashed")
# ggtitle('abbott ROC')
rocplot_abbott
```

```
# ggsave(plot = rocplot_abbott, filename = 'Fig1_roc_abbott.pdf', width=2.5,
# height=2.5, useDingbats=FALSE)
```

3) Examine the variability of samples taken within 48 hours of each other. reading and curating data from pairs of Ct value from Abbott and Cephied

```
pairs_dat <- fread("Data_used/within_48.csv", header = T)
pairs_dat <- as.data.frame(pairs_dat)
dim(pairs_dat) #93 9
```

```
## [1] 93 9
```

```
# how many different machines
table(pairs_dat$Method)
```

```
##
## Abbott Cephied Hologic
##      16      30      47
```

```
# sanity check- duplicated specimen IDs?
any(duplicated(pairs_dat$Spec_name)) # FALSE
```

```
## [1] FALSE
```

```
# parsing the date and time information
pairs_dat$collect_parse <- parse_date_time(pairs_dat$Collected, orders = "m/d/y H:M p")
```

```
# sort based on MRN then time collected
pairs_dat <- pairs_dat[order(pairs_dat$mrn, pairs_dat$collect_parse), ]
```

```

# how many patients have more than 2 samples? # two have 3, one has 5
table(table(pairs_dat$mrn))

##
## 2 3 5
## 41 2 1

# split into list
split_pairs <- split(pairs_dat, pairs_dat$mrn)

pairs_df <- data.frame()
j <- 1
for (i in 1:length(split_pairs)) {
  if (nrow(split_pairs[[i]]) < 3) {
    pairs_df[j, 1] <- split_pairs[[i]]$mrn[1]
    pairs_df[j, 2] <- diff(split_pairs[[i]]$Ct)
    pairs_df[j, 3] <- as.numeric(difftime(split_pairs[[i]]$collect_parse[2],
      split_pairs[[i]]$collect_parse[1], units = "hours"))
    pairs_df[j, 4] <- split_pairs[[i]]$Age[1]
    pairs_df[j, 5] <- split_pairs[[i]]$Sex[1]
    pairs_df[j, 6] <- split_pairs[[i]]$Method[1]
    pairs_df[j, 7] <- (split_pairs[[i]]$Method[1] == split_pairs[[i]]$Method[2])
    pairs_df[j, 8] <- split_pairs[[i]]$Ct[1]
    pairs_df[j, 9] <- split_pairs[[i]]$Ct[2]
    pairs_df[j, 10] <- mean(split_pairs[[i]]$Ct)
    pairs_df[j, 11] <- split_pairs[[i]]$collect_parse[1]
    j <- j + 1
  } else {
    for (k in 2:nrow(split_pairs[[i]])) {
      pairs_df[j, 1] <- split_pairs[[i]]$mrn[1]
      pairs_df[j, 2] <- split_pairs[[i]]$Ct[k] - split_pairs[[i]]$Ct[1]
      pairs_df[j, 3] <- as.numeric(difftime(split_pairs[[i]]$collect_parse[k],
        split_pairs[[i]]$collect_parse[1], units = "hours"))
      pairs_df[j, 4] <- split_pairs[[i]]$Age[1]
      pairs_df[j, 5] <- split_pairs[[i]]$Sex[1]
      pairs_df[j, 6] <- split_pairs[[i]]$Method[k]
      pairs_df[j, 7] <- (split_pairs[[i]]$Method[1] == split_pairs[[i]]$Method[k])
      pairs_df[j, 8] <- split_pairs[[i]]$Ct[1]
      pairs_df[j, 9] <- split_pairs[[i]]$Ct[k]
      pairs_df[j, 10] <- (split_pairs[[i]]$Ct[1] + split_pairs[[i]]$Ct[k])/2
      pairs_df[j, 11] <- split_pairs[[i]]$collect_parse[1]
      j <- j + 1
    }
  }
}
colnames(pairs_df) <- c("MR", "CT_diff", "Time_diff", "Age", "Sex", "Method", "Method_match",
  "CT1", "CT2", "CT_mean", "first_collect_date")

# QC on Method matching
any(!pairs_df$Method_match)

## [1] FALSE

# how many in first 24 hours
sum(length(pairs_df$CT_diff[pairs_df$Time_diff < 24])) #19 in first 24 hours

```

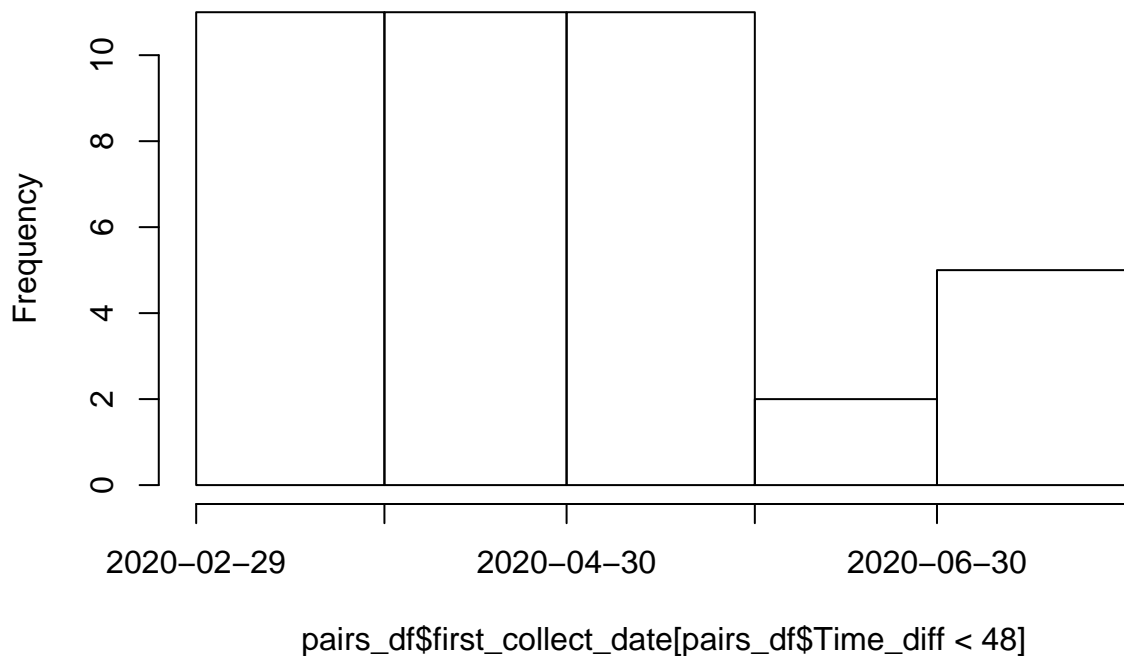
```
## [1] 19
```

```
sort(pairs_df$first_collect_date[pairs_df$Time_diff < 48]) #7/40 are after May
```

```
## [1] "2020-03-20 09:50:00 UTC" "2020-03-20 16:51:00 UTC"
## [3] "2020-03-21 22:15:00 UTC" "2020-03-26 13:16:00 UTC"
## [5] "2020-03-27 12:30:00 UTC" "2020-03-28 15:00:00 UTC"
## [7] "2020-03-28 15:45:00 UTC" "2020-03-29 03:27:00 UTC"
## [9] "2020-03-29 12:00:00 UTC" "2020-03-30 05:41:00 UTC"
## [11] "2020-03-30 20:24:00 UTC" "2020-03-31 10:09:00 UTC"
## [13] "2020-04-02 06:32:00 UTC" "2020-04-05 10:46:00 UTC"
## [15] "2020-04-06 18:28:00 UTC" "2020-04-07 05:20:00 UTC"
## [17] "2020-04-07 22:43:00 UTC" "2020-04-09 08:22:00 UTC"
## [19] "2020-04-13 16:37:00 UTC" "2020-04-17 16:05:00 UTC"
## [21] "2020-04-18 13:28:00 UTC" "2020-04-18 13:46:00 UTC"
## [23] "2020-05-08 21:50:00 UTC" "2020-05-13 10:36:00 UTC"
## [25] "2020-05-14 15:36:00 UTC" "2020-05-15 10:10:00 UTC"
## [27] "2020-05-15 11:12:00 UTC" "2020-05-15 15:35:00 UTC"
## [29] "2020-05-17 12:03:00 UTC" "2020-05-20 09:43:00 UTC"
## [31] "2020-05-20 11:37:00 UTC" "2020-05-20 16:58:00 UTC"
## [33] "2020-05-21 17:53:00 UTC" "2020-06-04 08:28:00 UTC"
## [35] "2020-06-17 14:12:00 UTC" "2020-07-05 09:58:00 UTC"
## [37] "2020-07-05 10:47:00 UTC" "2020-07-07 15:12:00 UTC"
## [39] "2020-07-07 15:33:00 UTC" "2020-07-28 13:21:00 UTC"
```

```
hist(pairs_df$first_collect_date[pairs_df$Time_diff < 48], "months", freq = TRUE)
```

Histogram of pairs_df\$first_collect_date[pairs_df\$Time_diff < 48]



```
pairs_df$first_collect_date
```

```
## [1] "2020-07-07 15:12:00 UTC" "2020-05-21 17:53:00 UTC"
## [3] "2020-05-20 16:58:00 UTC" "2020-07-05 10:47:00 UTC"
```

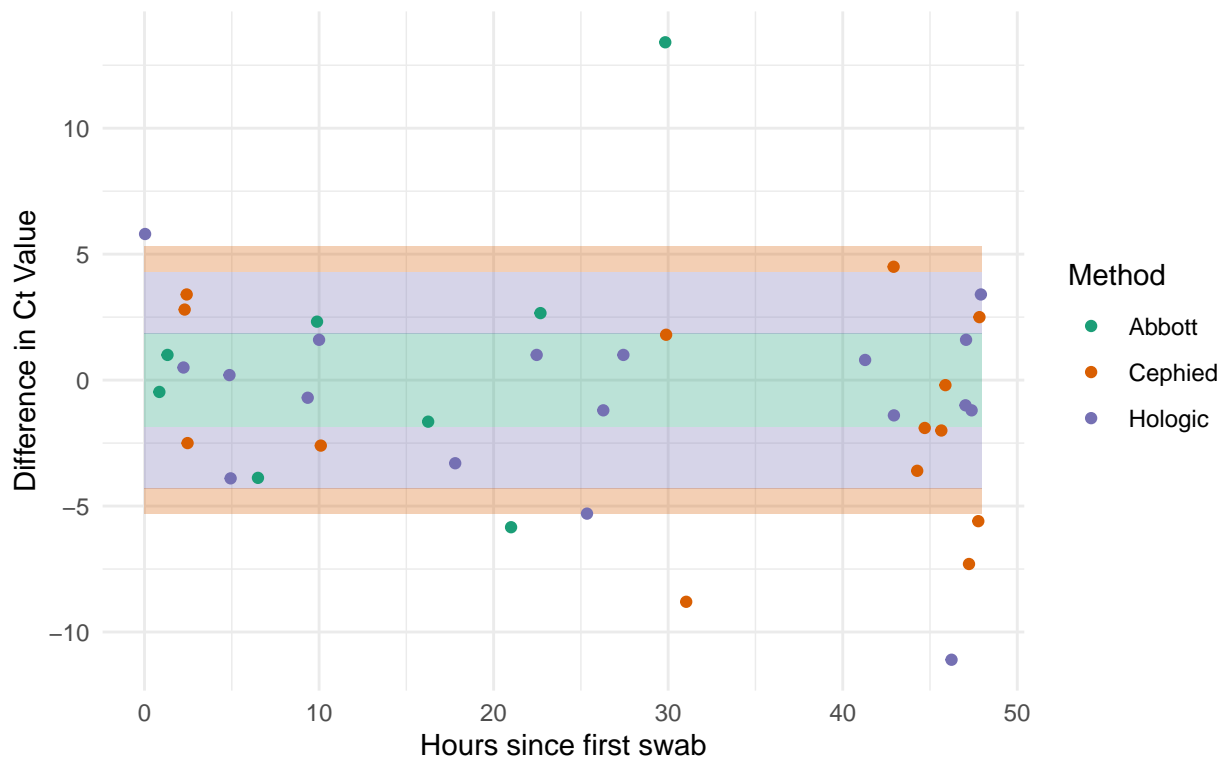
```
## [5] "2020-03-28 15:00:00 UTC" "2020-05-15 10:10:00 UTC"
## [7] "2020-08-01 10:25:00 UTC" "2020-08-01 10:25:00 UTC"
## [9] "2020-05-17 12:03:00 UTC" "2020-07-07 15:33:00 UTC"
## [11] "2020-05-08 21:50:00 UTC" "2020-03-30 20:24:00 UTC"
## [13] "2020-04-07 22:43:00 UTC" "2020-03-20 09:50:00 UTC"
## [15] "2020-05-13 10:36:00 UTC" "2020-04-05 10:46:00 UTC"
## [17] "2020-06-16 08:26:00 UTC" "2020-05-20 11:37:00 UTC"
## [19] "2020-04-07 05:20:00 UTC" "2020-04-06 18:28:00 UTC"
## [21] "2020-04-13 16:37:00 UTC" "2020-06-17 14:12:00 UTC"
## [23] "2020-05-29 13:54:00 UTC" "2020-03-27 12:30:00 UTC"
## [25] "2020-03-28 15:45:00 UTC" "2020-03-29 03:27:00 UTC"
## [27] "2020-07-28 13:21:00 UTC" "2020-05-15 11:12:00 UTC"
## [29] "2020-05-20 09:43:00 UTC" "2020-05-20 09:43:00 UTC"
## [31] "2020-05-20 09:43:00 UTC" "2020-05-20 09:43:00 UTC"
## [33] "2020-07-05 09:58:00 UTC" "2020-05-14 15:36:00 UTC"
## [35] "2020-03-30 05:41:00 UTC" "2020-05-15 15:35:00 UTC"
## [37] "2020-04-17 16:05:00 UTC" "2020-05-19 14:03:00 UTC"
## [39] "2020-05-19 14:03:00 UTC" "2020-04-18 13:28:00 UTC"
## [41] "2020-04-02 06:32:00 UTC" "2020-06-04 08:28:00 UTC"
## [43] "2020-03-26 13:16:00 UTC" "2020-03-20 16:51:00 UTC"
## [45] "2020-03-21 22:15:00 UTC" "2020-03-29 12:00:00 UTC"
## [47] "2020-03-31 10:09:00 UTC" "2020-04-18 13:46:00 UTC"
## [49] "2020-04-09 08:22:00 UTC"
```

plotting the pair with time on x, CT diff on y, fill by method

```
# color scheme
dark_col <- brewer.pal(6, "Dark2")

# layering in CT differences for each machine Cephied 5.299954 Hologic 4.299973
# Abbott 1.859981
ggplot(data = pairs_df, aes(x = Time_diff, y = CT_diff, col = Method)) + annotate("rect",
  xmin = 0, xmax = 48, ymin = -1.859981, ymax = 1.859981, alpha = 0.3, fill = dark_col[1]) +
  annotate("rect", xmin = 0, xmax = 48, ymin = 1.859981, ymax = 4.299973, alpha = 0.3,
    fill = dark_col[3]) + annotate("rect", xmin = 0, xmax = 48, ymin = -1.859981,
  ymax = -4.299973, alpha = 0.3, fill = dark_col[3]) + annotate("rect", xmin = 0,
  xmax = 48, ymin = 4.299973, ymax = 5.299954, alpha = 0.3, fill = dark_col[2]) +
  annotate("rect", xmin = 0, xmax = 48, ymin = -4.299973, ymax = -5.299954, alpha = 0.3,
    fill = dark_col[2]) + geom_point() + theme_minimal() + scale_color_manual(values = dark_col) +
  xlab("Hours since first swab") + ylab("Difference in Ct Value") + xlim(c(0, 48)) +
  ggtitle("Swab intrasample variability within first 48 hours,\nshaded areas corresponding to Ct Diff")
```

Swab intrasample variability within first 48 hours,
shaded areas corresponding to Ct Diff btwn mortality for each method



4) Observe how many specimen pairs cross the Youden Index for their respective machine

```
# grabbing only those pairs within 48 hours
```

```
pairs_df_48 <- pairs_df[pairs_df$Time_diff < 48, ]
dim(pairs_df_48)
```

```
## [1] 40 11
```

```
# Abbott number of pairs
```

```
sum(pairs_df_48$Method == "Abbott") #8
```

```
## [1] 8
```

```
## number cross threshold
```

```
sum((pairs_df_48$CT1[pairs_df_48$Method == "Abbott"] < youden_coord_abbott$threshold &
  pairs_df_48$CT2[pairs_df_48$Method == "Abbott"] > youden_coord_abbott$threshold) |
  (pairs_df_48$CT1[pairs_df_48$Method == "Abbott"] > youden_coord_abbott$threshold &
  pairs_df_48$CT2[pairs_df_48$Method == "Abbott"] < youden_coord_abbott$threshold)) #3
```

```
## [1] 3
```

```
idx_abbott_youden <- which((pairs_df_48$CT1[pairs_df_48$Method == "Abbott"] < youden_coord_abbott$threshold &
  pairs_df_48$CT2[pairs_df_48$Method == "Abbott"] > youden_coord_abbott$threshold) |
  (pairs_df_48$CT1[pairs_df_48$Method == "Abbott"] > youden_coord_abbott$threshold &
  pairs_df_48$CT2[pairs_df_48$Method == "Abbott"] < youden_coord_abbott$threshold)) #3
```

```
# Cepheid number of pairs
```

```
sum(pairs_df_48$Method == "Cepheid") #14
```

```
## [1] 14
```

```

## number cross threshold
sum((pairs_df_48$CT1[pairs_df_48$Method == "Cephied"] < youden_coord_cephied$threshold &
  pairs_df_48$CT2[pairs_df_48$Method == "Cephied"] > youden_coord_cephied$threshold) |
  (pairs_df_48$CT1[pairs_df_48$Method == "Cephied"] > youden_coord_cephied$threshold &
    pairs_df_48$CT2[pairs_df_48$Method == "Cephied"] < youden_coord_cephied$threshold)) #2

## [1] 2

idx_Cephied_youden <- which((pairs_df_48$CT1[pairs_df_48$Method == "Cephied"] < youden_coord_cephied$thres
  pairs_df_48$CT2[pairs_df_48$Method == "Cephied"] > youden_coord_cephied$threshold) |
  (pairs_df_48$CT1[pairs_df_48$Method == "Cephied"] > youden_coord_cephied$threshold &
    pairs_df_48$CT2[pairs_df_48$Method == "Cephied"] < youden_coord_cephied$threshold)) #3
# Hologic number of pairs
sum(pairs_df_48$Method == "Hologic") #18

## [1] 18

## number cross threshold
sum((pairs_df_48$CT1[pairs_df_48$Method == "Hologic"] < youden_coord_hologic$threshold &
  pairs_df_48$CT2[pairs_df_48$Method == "Hologic"] > youden_coord_hologic$threshold) |
  (pairs_df_48$CT1[pairs_df_48$Method == "Hologic"] > youden_coord_hologic$threshold &
    pairs_df_48$CT2[pairs_df_48$Method == "Hologic"] < youden_coord_hologic$threshold)) #3

## [1] 3

idx_holo_youden <- which((pairs_df_48$CT1[pairs_df_48$Method == "Hologic"] < youden_coord_hologic$thres
  pairs_df_48$CT2[pairs_df_48$Method == "Hologic"] > youden_coord_hologic$threshold) |
  (pairs_df_48$CT1[pairs_df_48$Method == "Hologic"] > youden_coord_hologic$threshold &
    pairs_df_48$CT2[pairs_df_48$Method == "Hologic"] < youden_coord_hologic$threshold)) #3

pairs_df_48$CT1[pairs_df_48$Method == "Hologic"][idx_holo_youden]

## [1] 26.9 26.6 27.3

## highlighting the pairs which crossed the Youden index
pairs_df_48$`Crossed Youden` <- "No"
pairs_df_48$`Crossed Youden`[pairs_df_48$Method == "Abbott"][idx_abbott_youden] <- "Yes"
pairs_df_48$`Crossed Youden`[pairs_df_48$Method == "Cephied"][idx_Cephied_youden] <- "Yes"
pairs_df_48$`Crossed Youden`[pairs_df_48$Method == "Hologic"][idx_holo_youden] <- "Yes"

# What is the mortality status of Crossed Youden samples?
crossed_pairs <- pairs_df_48[pairs_df_48$`Crossed Youden` == "Yes", ]
crossed_pairs$direction <- ifelse(crossed_pairs$CT_diff > 0, "up", "down")
idx_crossed_youden <- which(mort_dat$MRN %in% pairs_df_48[pairs_df_48$`Crossed Youden` ==
  "Yes", 1])

# layering in CT differences for each machine Cephied 5.299954 Hologic 4.299973
# Abbott 1.859981
Cross_youden_plot <- ggplot(data = pairs_df_48, aes(x = Time_diff, y = CT_diff, col = Method,
  size = `Crossed Youden`, shape = `Crossed Youden`)) + annotate("rect", xmin = 0,
  xmax = 48, ymin = -1.859981, ymax = 1.859981, alpha = 0.3, fill = dark_col[1]) +
  annotate("rect", xmin = 0, xmax = 48, ymin = 1.859981, ymax = 4.299973, alpha = 0.3,
    fill = dark_col[3]) + annotate("rect", xmin = 0, xmax = 48, ymin = -1.859981,
    ymax = -4.299973, alpha = 0.3, fill = dark_col[3]) + annotate("rect", xmin = 0,

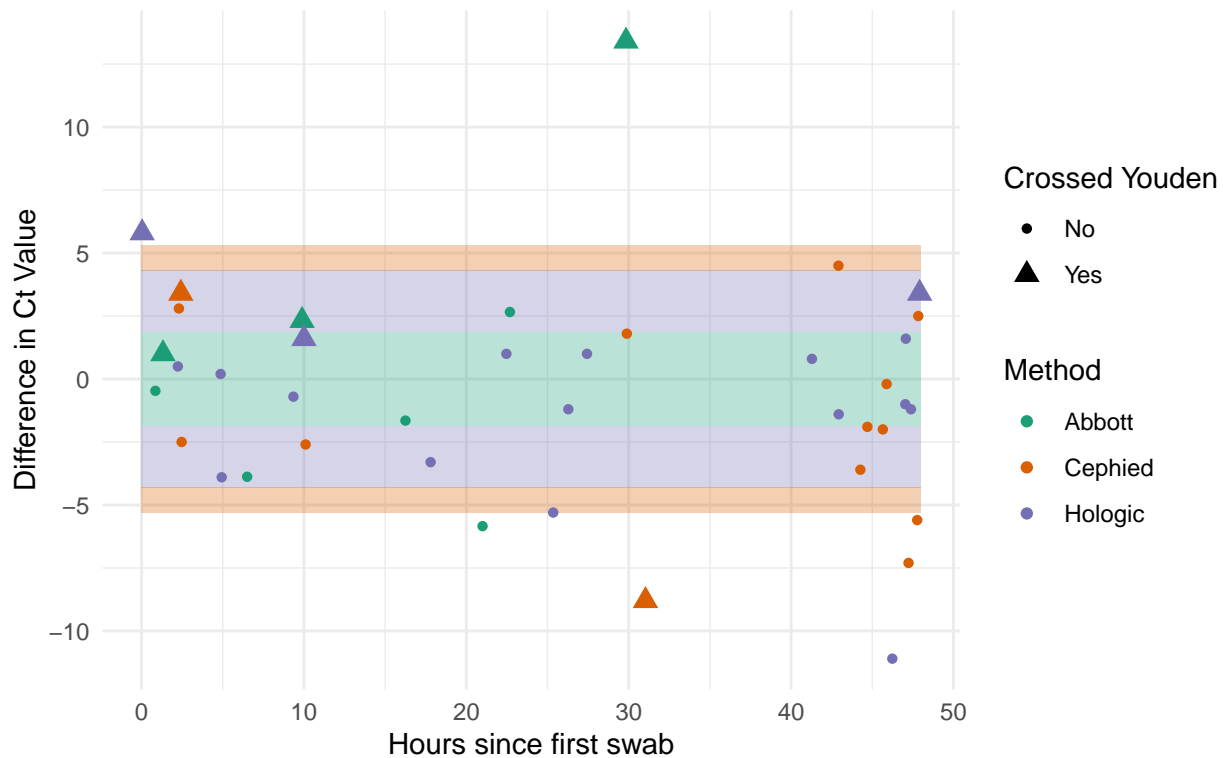
```

```

xmax = 48, ymin = 4.299973, ymax = 5.299954, alpha = 0.3, fill = dark_col[2]) +
annotate("rect", xmin = 0, xmax = 48, ymin = -4.299973, ymax = -5.299954, alpha = 0.3,
       fill = dark_col[2]) + geom_point() + theme_minimal() + scale_size_discrete(range = c(1.5,
3)) + scale_color_manual(values = dark_col) + xlab("Hours since first swab") +
ylab("Difference in Ct Value") + xlim(c(0, 48)) + ggtitle("Swab intrasample variability within first
Cross_youden_plot

```

Swab intrasample variability within first 48 hours,
shaded areas corresponding to Ct Diff btwn mortality for each method



```

# ggsave(plot = Cross_youden_plot, filename = 'Fig2_youden_cross.pdf', width=6.5,
# height=4, useDingbats=FALSE)

```

Out of the 40 pairs, 8 crossed the youden index point for predicting mortality (20%). Lending evidence to being cautionary when using Ct value to predict mortality.

Session Information:

```
sessionInfo()
```

```

## R version 3.5.1 (2018-07-02)
## Platform: x86_64-apple-darwin15.6.0 (64-bit)
## Running under: macOS High Sierra 10.13.6
##
## Matrix products: default
## BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##

```

```

## attached base packages:
## [1] stats      graphics  grDevices utils      datasets  methods  base
##
## other attached packages:
## [1] pastecs_1.3.21      cutpointnr_1.0.32      OptimalCutpoints_1.1-4
## [4] plotROC_2.2.1       pROC_1.16.1           ROCit_2.1.1
## [7] lubridate_1.7.9     overlapping_1.6        testthat_2.3.1
## [10] flexplot_0.7.5      Hmisc_4.3-0           Formula_1.2-3
## [13] lattice_0.20-38     scales_1.1.0          AER_1.2-8
## [16] survival_3.1-8      sandwich_2.5-1        lmtest_0.9-37
## [19] zoo_1.8-7           car_3.0-6             carData_3.0-3
## [22] MASS_7.3-51.5       RColorBrewer_1.1-2    ggpubr_0.2.4
## [25] magrittr_1.5        reshape2_1.4.3        ggthemes_4.2.0
## [28] ggplot2_3.2.1       data.table_1.12.8
##
## loaded via a namespace (and not attached):
## [1] foreach_1.4.7      splines_3.5.1         assertthat_0.2.1
## [4] latticeExtra_0.6-28 cellranger_1.1.0      yaml_2.2.0
## [7] pillar_1.4.3       backports_1.1.5      glue_1.3.1
## [10] digest_0.6.23      ggsignif_0.6.0       checkmate_1.9.4
## [13] colorspace_1.4-1   htmltools_0.4.0      Matrix_1.2-18
## [16] plyr_1.8.5         pkgconfig_2.0.3      haven_2.2.0
## [19] purrr_0.3.3        openxlsx_4.1.4       rio_0.5.16
## [22] tibble_3.0.3       htmlTable_1.13.3     farver_2.0.3
## [25] generics_0.0.2     ellipsis_0.3.0       withr_2.1.2
## [28] nnet_7.3-12        lazyeval_0.2.2       crayon_1.3.4
## [31] readxl_1.3.1       evaluate_0.14        forcats_0.4.0
## [34] foreign_0.8-75     tools_3.5.1          hms_0.5.3
## [37] formatR_1.7        lifecycle_0.2.0      stringr_1.4.0
## [40] munsell_0.5.0      cluster_2.1.0        zip_2.0.4
## [43] compiler_3.5.1     rlang_0.4.7          grid_3.5.1
## [46] iterators_1.0.12   rstudioapi_0.10      htmlwidgets_1.5.1
## [49] labeling_0.3       base64enc_0.1-3      rmarkdown_2.1
## [52] boot_1.3-24        codetools_0.2-16     gtable_0.3.0
## [55] abind_1.4-5        curl_4.3             R6_2.4.1
## [58] gridExtra_2.3      knitr_1.27           dplyr_0.8.3
## [61] stringi_1.4.5      Rcpp_1.0.3           vctrs_0.3.4
## [64] rpart_4.1-15       acepack_1.4.1        tidyselect_0.2.5
## [67] xfun_0.12

```