Analysis for the manuscript 'Hepcidin levels can distinguish anemia of chronic disease from iron deficiency anemia in a cross-sectional study of hidradenitis suppurativa patients'

Andrew D. Johnston

12/17/2020

Contents

This document details the analysis and generation of figures found in . The dataset related to this article can be found at https://data.mendeley.com/datasets/9x4n84cxxw/1, an open-source online data repository hosted at Mendeley Data Ghias, Mondana; Johnston, Andrew; Babbush, Kayla; Cohen, Steven (2020), "Elevated hepcidin in hidradenitis suppurativa", Mendeley Data, v1 http://dx.doi.org/10.17632/9x4n84cxxw.1.

Overview of Analysis

1. Examine the relationship among acute phase reactants, hepcidin, and disease severity (HS-PGA) 1a. Univariate ordinal regression Hepcidin and HS-PGA
2. Examine the relationship between hepcidin and different anemia classifications.
3. Using Hepcidin to distinguish among IDA, ACD, and ACD/IDA
4. Making Table 1
5. Making Figure 1

Curate Data

Load in packages

```
# load in libraries
library(data.table)
library(ggplot2)
library(ggthemes)
library(reshape2)
library(ggpubr)
library(RColorBrewer)
library(MASS)
library(AER)
library(scales)
library("flexplot")
library(overlapping)
library(lattice)
library(lubridate)
library(ROCit)
library(pROC)
library(plotROC)
library(OptimalCutpoints)
library(cutpointr)
library(pastecs)
library(Hmisc)
library(mblm)
library(polycor)
library(psych)
library(table1)
library(dunn.test)

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

download the data.

```
hep_dat <- read.csv("Hepcidin Data Reformated_7.31.19.csv", header = T)
tail(hep_dat[1:132, ])
```

```
##     Sample Sample.Date Gender Age Race   BMI Hurley   Hb Anemic  MCV PLT Fe
## 127     29     12/4/18      0  22    2    NA      3 15.2      0 91.8 276 74
## 128     52    12/17/18      0  27    1 31.14      3 15.3      0 90.9 193 74
## 129    150     2/11/19      1  37    3 34.33      2 15.5      0 94.9 285 50
## 130     59    12/18/18      1  24    3    NA      1 15.7      0 90.0 303 89
## 131    124     1/28/19      0  48    0 36.34      3 16.4      0 81.0 224 81
## 132     NA                NA  NA   NA    NA     NA   NA     NA   NA  NA NA
##     Hepcidin Ferritin Ferritin.class CRP CRP.class ESR PGA PGA.class Tsat
## 127    43.00      212              2 3.3         1  83   5         1   31
## 128    80.10      102              1  NA         2  30   3         1   30
## 129    12.50       37              1 0.7         0   9   3         1   12
## 130    34.33       88              1 0.7         0  15   2         0   19
## 131    35.17      270              2 0.5         0  17   3         1   25
## 132       NA       NA             NA  NA        NA  NA  NA        NA   NA
##     Transferrin Visit Paired.ID
## 127         188     1
## 128         195     1      11-1
## 129         344     1
## 130         368     1
## 131         256     1
## 132          NA    NA
```

```
hep_dat <- hep_dat[1:131, ]  # remove blanks rows
hep_dat <- hep_dat[hep_dat$Visit == 1, ]  # use only the 1st visit data
# removing 1 individual without sufficient data, anemic but no ferritin and tsat
# data
hep_dat <- hep_dat[!(hep_dat$Anemic == 1 & (is.na(hep_dat$Tsat) | is.na(hep_dat$Ferritin))),
    ]
hep_dat$PGA <- as.factor(hep_dat$PGA)
hep_dat$Gender <- as.factor(hep_dat$Gender)
hep_dat_3 <- hep_dat
hep_dat_3$Anemic <- as.factor(hep_dat_3$Anemic)
seq_col <- brewer.pal(6, "YlOrRd")
hep_dat_4 <- hep_dat_3
hep_dat_4$PGA <- as.numeric(hep_dat_4$PGA)
```

1 Relationship among hepcidin, acute phase reactants, and disease severity

1a. Univariate ordinal regression Hepcidin and HS-PGA

Hepcidin levels positively predict HS-PGA in univariate ordinal regression

```
ord_uni_PGA_hep <- polr(PGA ~ Hepcidin, data = hep_dat, Hess = TRUE)
coeftest(ord_uni_PGA_hep)
```

```
##
## z test of coefficients:
##
##            Estimate Std. Error z value           Pr(>|z|)
## Hepcidin  0.0324693  0.0067663  4.7987 0.0000015971625323 ***
## 0|1      -3.9637873  1.0145094 -3.9071 0.0000934113601393 ***
## 1|2      -0.7079215  0.2973038 -2.3811           0.017259 *
## 2|3       0.7937646  0.2800878  2.8340           0.004597 **
```

```
## 3|4       1.6491474  0.3092996  5.3319 0.0000000972030791 ***
## 4|5       2.8089882  0.3826608  7.3407 0.0000000000002125 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
summary(ord_uni_PGA_hep)
```

```
## Call:
## polr(formula = PGA ~ Hepcidin, data = hep_dat, Hess = TRUE)
##
## Coefficients:
##            Value Std. Error t value
## Hepcidin 0.03247   0.006766   4.799
##
## Intercepts:
##     Value   Std. Error t value
## 0|1 -3.9638  1.0145    -3.9071
## 1|2 -0.7079  0.2973    -2.3811
## 2|3  0.7938  0.2801     2.8340
## 3|4  1.6491  0.3093     5.3319
## 4|5  2.8090  0.3827     7.3407
##
## Residual Deviance: 338.2186
## AIC: 350.2186
```

```
## store table
(tab_ord_PGA_hep <- coef(summary(ord_uni_PGA_hep)))
```

```
##                Value  Std. Error   t value
## Hepcidin  0.03246928 0.006766297  4.798678
## 0|1      -3.96378728 1.014509364 -3.907098
## 1|2      -0.70792154 0.297303846 -2.381138
## 2|3       0.79376463 0.280087819  2.833985
## 3|4       1.64914740 0.309299628  5.331876
## 4|5       2.80898823 0.382660842  7.340673
```

```
## calculate and store p values
p_ord_PGA_hep <- pnorm(abs(tab_ord_PGA_hep[, "t value"]), lower.tail = FALSE) * 2
## combined table
(tab_ord_PGA_hep <- cbind(tab_ord_PGA_hep, `p value` = p_ord_PGA_hep))
```

```
##                Value  Std. Error   t value              p value
## Hepcidin  0.03246928 0.006766297  4.798678 0.000001597162532254
## 0|1      -3.96378728 1.014509364 -3.907098 0.000093411360139315
## 1|2      -0.70792154 0.297303846 -2.381138 0.017259236927527309
## 2|3       0.79376463 0.280087819  2.833985 0.004597151568501760
## 3|4       1.64914740 0.309299628  5.331876 0.000000097203079144
## 4|5       2.80898823 0.382660842  7.340673 0.000000000000212522
```

```
## odds ratio
exp(coef(ord_uni_PGA_hep))
```

```
## Hepcidin
## 1.033002
```

```
## odds ratio with 95% CI
c(exp(coef(ord_uni_PGA_hep)), exp(confint(ord_uni_PGA_hep)))
```

```
## Hepcidin    2.5 %   97.5 %
## 1.033002 1.020202 1.047762
```

```
# Hepcidin and Hurley
table(hep_dat$Hurley)
```

```
##
##  1  2  3
## 21 11 79
```

1b. Correlation among quantitive factors

Age, BMI, Hb, MCV, PLT,Fe, Hepcidin, Ferritin, CRP, ESR, Tsat, Transferrin are the factors for which we have data points. First, we will assess the distribution of the variables; Only MCV and Transferrin are normally distributed; therefore, will use pearson correlation test (http://www.sthda.com/english/wiki/normality-test-in-r).

```
colnames(hep_dat)
```

```
##  [1] "Sample"         "Sample.Date"    "Gender"         "Age"
##  [5] "Race"           "BMI"            "Hurley"         "Hb"
##  [9] "Anemic"         "MCV"            "PLT"            "Fe"
## [13] "Hepcidin"       "Ferritin"       "Ferritin.class" "CRP"
## [17] "CRP.class"      "ESR"            "PGA"            "PGA.class"
## [21] "Tsat"           "Transferrin"    "Visit"          "Paired.ID"
```

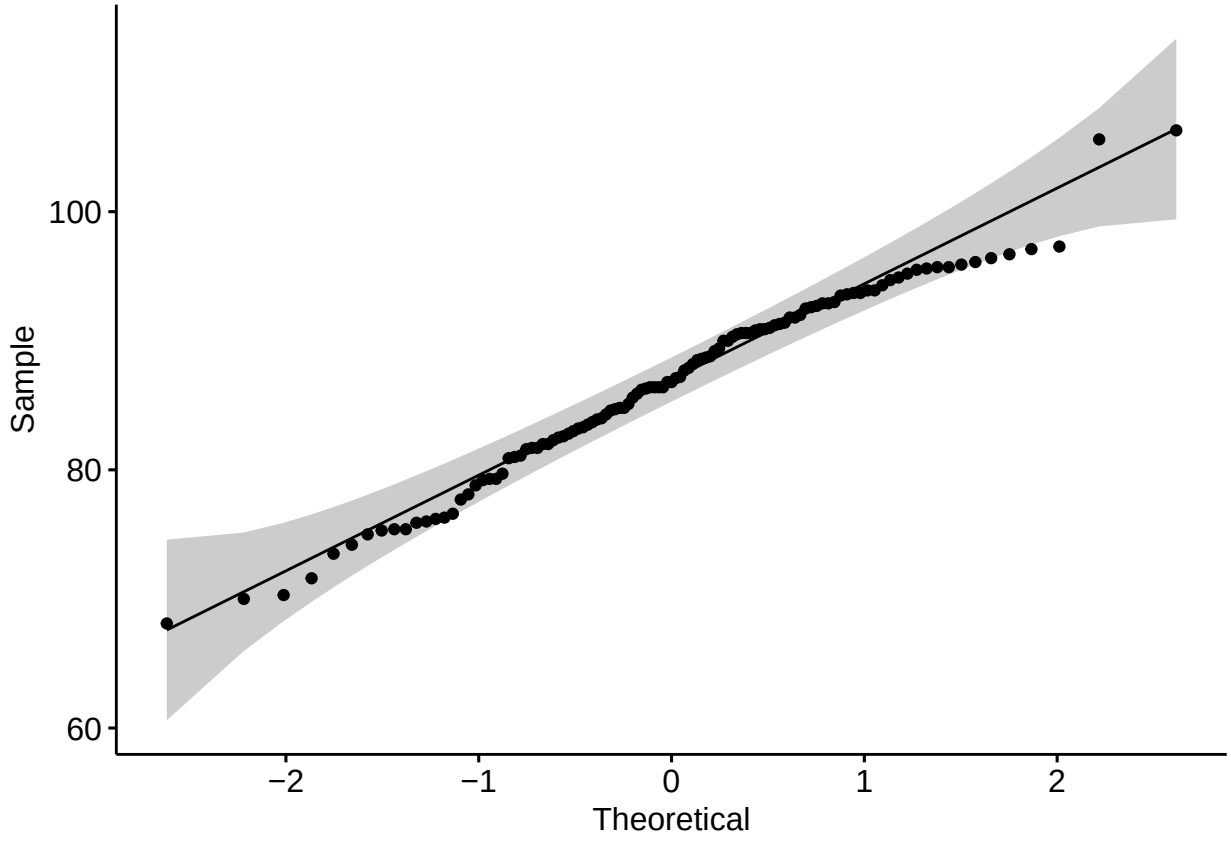
```
# Age, BMI, Hb, MCV, PLT,Fe, Hepcidin, Ferritin,CRP, ESR, Tsat, Transferrin
hep_dat_quant <- hep_dat[, c(4, 6, 8, 10:14, 16, 18, 21:22)]
tail(hep_dat_quant)
```

```
##     Age   BMI   Hb  MCV PLT  Fe Hepcidin Ferritin  CRP ESR Tsat Transferrin
## 125  30    NA 15.1 92.6 229 144    34.74       71 0.25  15   42         273
## 127  22    NA 15.2 91.8 276  74    43.00      212 3.30  83   31         188
## 128  27 31.14 15.3 90.9 193  74    80.10      102   NA  30   30         195
## 129  37 34.33 15.5 94.9 285  50    12.50       37 0.70   9   12         344
## 130  24    NA 15.7 90.0 303  89    34.33       88 0.70  15   19         368
## 131  48 36.34 16.4 81.0 224  81    35.17      270 0.50  17   25         256
```

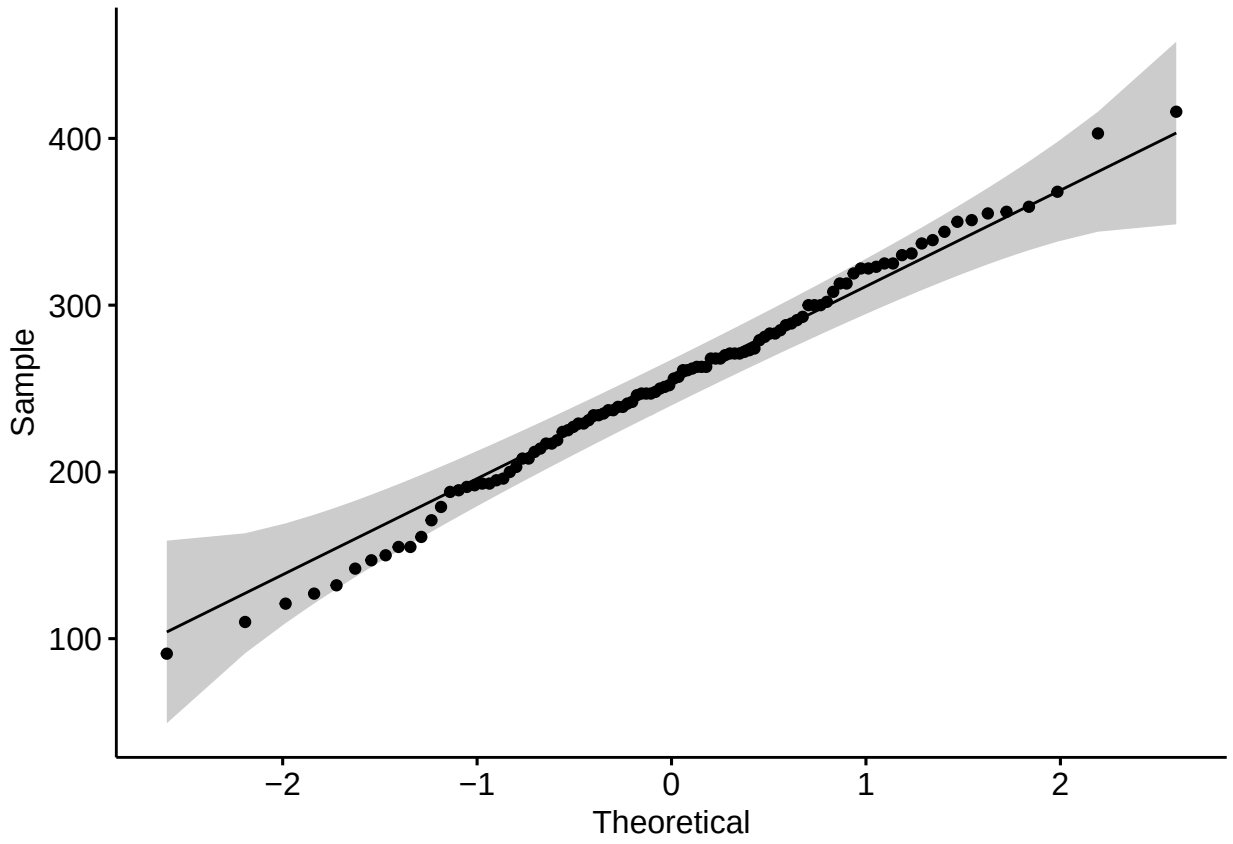
```
dim(hep_dat_quant)
```

```
## [1] 113  12
```

```
l_qqplot <- list()
for (i in 1:ncol(hep_dat_quant)) {
    l_qqplot[[i]] <- ggqqplot(hep_dat_quant[, i])
    names(l_qqplot)[i] <- colnames(hep_dat_quant)[i]
}
l_qqplot[[4]]  # MCV
```




```
l_qqplot[[12]]  # transferrin
```




```
shapiro_tests <- NULL
for (i in 1:ncol(hep_dat_quant)) {
    shapiro_tests[i] <- unlist(shapiro.test(hep_dat_quant[, i])[2])
    names(shapiro_tests)[i] <- colnames(hep_dat_quant)[i]
}
which(shapiro_tests > 0.05)
```

```
##         MCV Transferrin
##           4          12
```

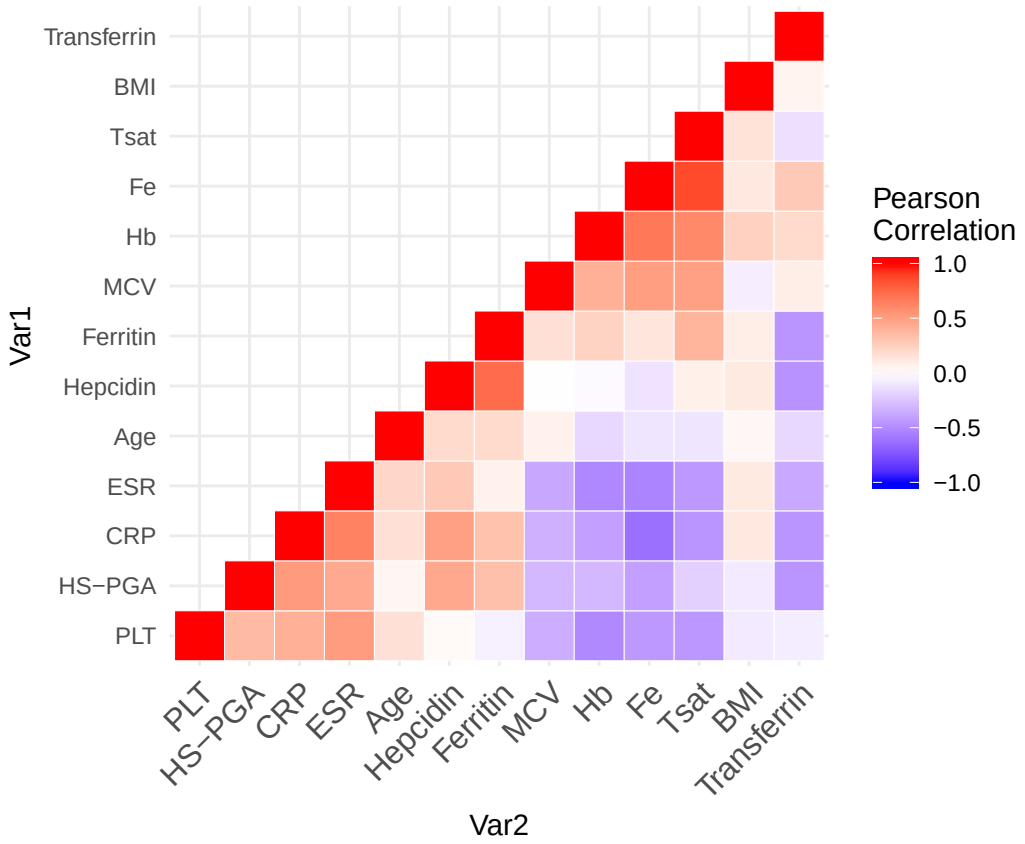
Will graphically represent the correlation among factors

```
cormat_quant <- rcorr(x = as.matrix(hep_dat_quant), type="spearman")
cormat_quant <- round(cormat_quant$r,2)

# make corelations between HS-PGA and other variables
cor_hspga <- NULL
j<-1
for (i in c(4,6,8,10:14,16,18,21:22)) {
  temp <- rcorr(hep_dat$PGA, hep_dat[,i], type = "spearman") # 0.7226236
  cor_hspga <- c(cor_hspga,temp$r[1,2])
  names(cor_hspga)[j] <- colnames(hep_dat)[i]
  j<-j+1
}
cor_hspga
```

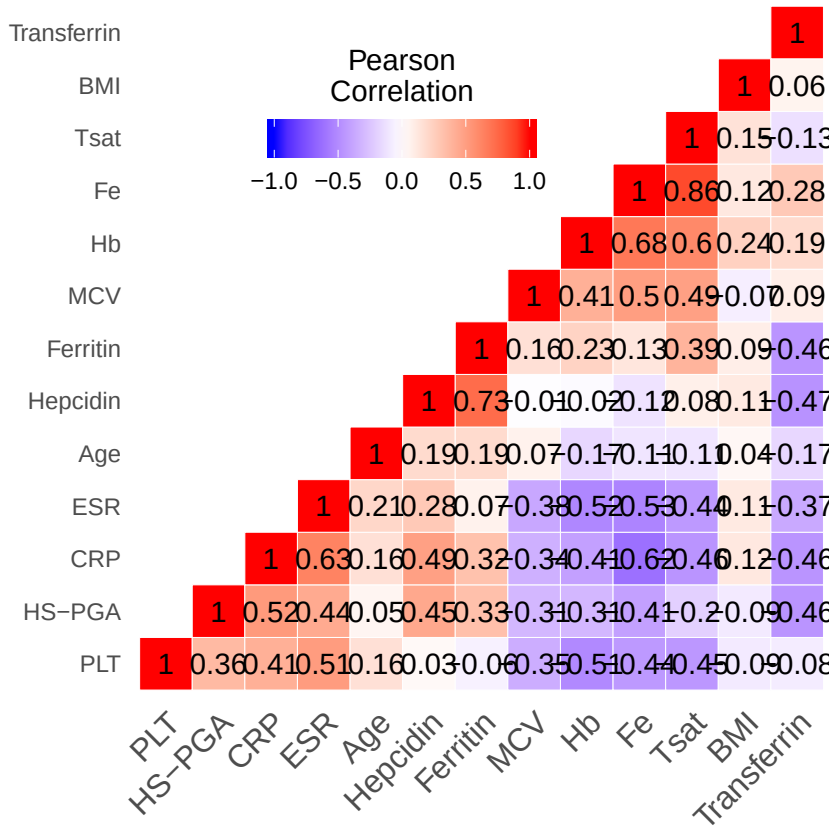
```
##         Age         BMI          Hb         MCV         PLT          Fe
##  0.04830706 -0.09423165 -0.31165635 -0.30534103  0.35934420 -0.40514345
##    Hepcidin    Ferritin         CRP         ESR        Tsat Transferrin
```

```
##  0.45255453  0.32809537  0.52079243  0.43722563 -0.20070701 -0.45999236
```

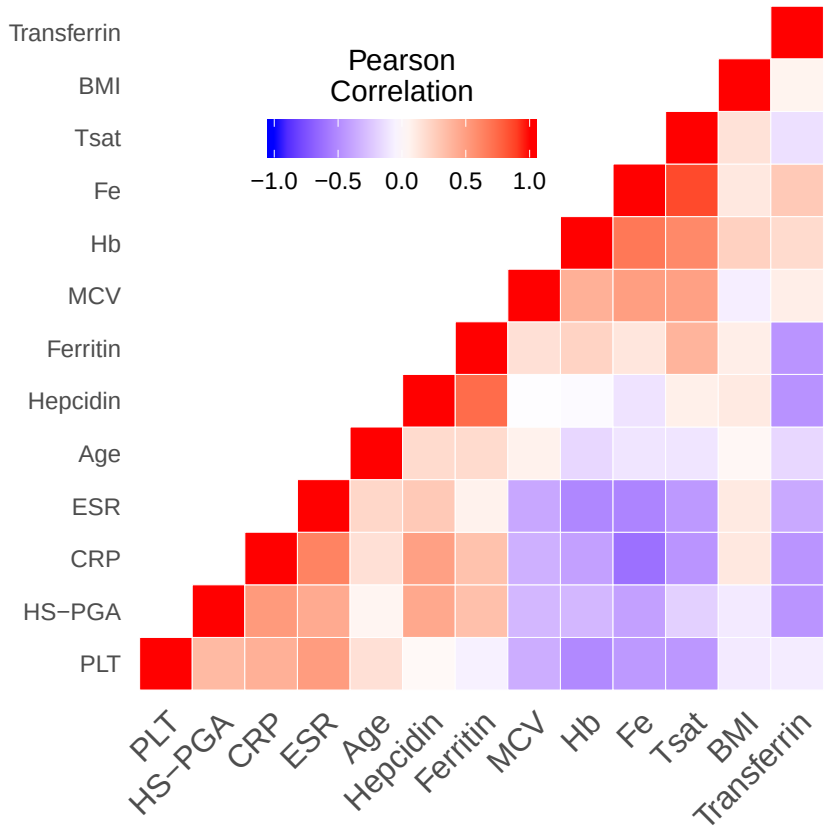
```
cormat_all <- rbind(cormat_quant, cor_hspga)
cormat_all <- cbind(cormat_all, c(cor_hspga,1))

#round
cormat_all <- round(cormat_all,2)

#fix names
rownames(cormat_all)[13] <- "HS-PGA"
colnames(cormat_all)[13] <- "HS-PGA"


# Reorder the correlation matrix
cormat <- reorder_cormat(cormat_all)
# make lower triangle
upper_tri <- get_upper_tri(cormat)

# Melt the correlation matrix
melted_cormat <- melt(upper_tri, na.rm = TRUE)
# Create a ggheatmap
ggheatmap <- ggplot(melted_cormat, aes(Var2, Var1, fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
    name="Pearson\nCorrelation") +
  theme_minimal()+ # minimal theme
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 12, hjust = 1))+
 coord_fixed()
# Print the heatmap
print(ggheatmap)
```




```
# labeled heatmap
ggheatmap_labeled <- ggheatmap +
geom_text(aes(Var2, Var1, label = value), color = "black", size = 4) +
theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
  legend.position = c(0.6, 0.7),
  legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                title.position = "top", title.hjust = 0.5))
print(ggheatmap_labeled)
```



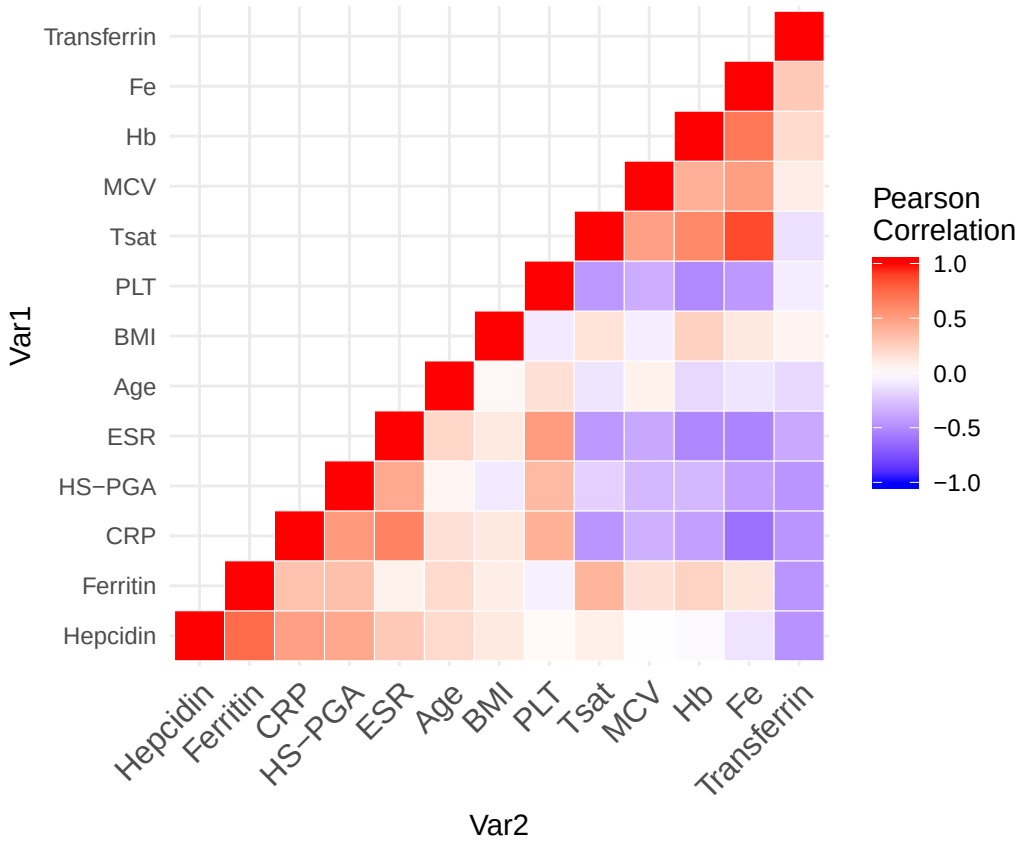

```
ggheatmap_ordered_notext <- ggheatmap +
theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
  legend.position = c(0.6, 0.7),
  legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                title.position = "top", title.hjust = 0.5))
print(ggheatmap_ordered_notext)
```



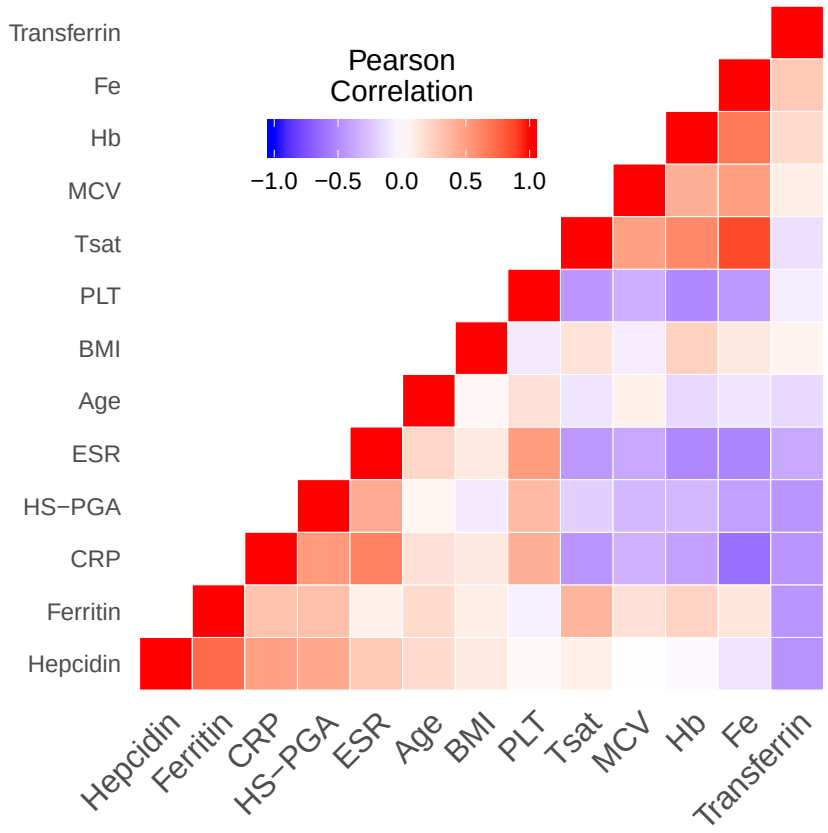

```
# supervised clustering to more easily see hepcidin's relationships with other variables
cormat <- reorder_cormat(cormat_all)
rownames(cormat)
```

```
##  [1] "PLT"         "HS-PGA"      "CRP"         "ESR"         "Age"
##  [6] "Hepcidin"    "Ferritin"    "MCV"         "Hb"          "Fe"
## [11] "Tsat"        "BMI"         "Transferrin"
```

```
# Hepcidin, ferritin, CRP, HS-PGA, ESR, Age, BMI, PLT, Tsat MCV, Hb, Fe, Transferrin
cormat_hep <- cormat[c(6:7,3,2,4,5,12,1,11,8:10,13),c(6:7,3,2,4,5,12,1,11,8:10,13)]
upper_tri_cormat_hep <- get_upper_tri(cormat_hep)
melted_cormat_hep <- melt(upper_tri_cormat_hep, na.rm = TRUE)


# Create a ggheatmap
ggheatmap_cormat <- ggplot(melted_cormat_hep, aes(Var2, Var1, fill = value))+
 geom_tile(color = "white")+
 scale_fill_gradient2(low = "blue", high = "red", mid = "white",
   midpoint = 0, limit = c(-1,1), space = "Lab",
    name="Pearson\nCorrelation") +
  theme_minimal()+ # minimal theme
 theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 12, hjust = 1))+
 coord_fixed()
# Print the heatmap
print(ggheatmap_cormat)
```




```
ggheatmap_supervised_notext <- ggheatmap_cormat +
theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.grid.major = element_blank(),
  panel.border = element_blank(),
  panel.background = element_blank(),
  axis.ticks = element_blank(),
  legend.justification = c(1, 0),
  legend.position = c(0.6, 0.7),
  legend.direction = "horizontal")+
  guides(fill = guide_colorbar(barwidth = 7, barheight = 1,
                title.position = "top", title.hjust = 0.5))
print(ggheatmap_supervised_notext)
```




```
#Correlations with Acute phase reactants:
  # Positive - CRP, ferritin, Hepcidin
  # Negative - Transferrin
```

Ferritin and hepcidin levels are strongly correlated (.73). Hepcidin levels also correlate with markers of inflammation (CRP, r=0.49) and disease-severity (HS-PGA, r=0.45). Of note, hepcidin levels do not reach the r=0.30 cutoff for ESR (r=0.28); perhaps showing that hepcidin is more affected by acute rather chronic inflammation.

1c. Correlation among quantitive factors

Using univariate analysis, do any variables seem to influence hepcidin levels? Checked with non-parametric linear regression using Siegel method (as hepcidin distribution contains outliers) as follows:

```
results_hep_mblm <- NULL
j <- 1
for (i in c(1:6, 8:ncol(hep_dat_quant))) {
    col_nam <- colnames(hep_dat_quant)[i]
    hep_no_na <- hep_dat[!is.na(hep_dat[, i]), ]
    m2 <- mblm(as.formula(paste0("Hepcidin", "~", col_nam)), data = hep_no_na, repeated = TRUE)
    results_hep_mblm <- rbind(results_hep_mblm, summary(m2)$coefficients[2, ])
    rownames(results_hep_mblm)[j] <- col_nam
    j <- j + 1
}
results_hep_mblm
```

```
##               Estimate        MAD V value                    Pr(>|V|)
## Age          0.31708333 0.82373531    4860 0.00000265978211406526013
## BMI          0.20996710 0.85633036    1752 0.40195633244911327386362
```

```
## Hb          -0.17285714 4.50971057    3048 0.6221869515986857557265 9
## MCV          0.17765568 1.15689552    3468 0.4791833959425902955686 6
## PLT         -0.01231481 0.09643465    3107 0.7461444083855086617518 3
## Fe          -0.14071429 0.28354648     505 0.0000033936088165136973 2
## Ferritin     0.39451229 0.19033319    5550 0.0000000000000000118654 3
## CRP          7.98275410 7.53082897    4343 0.0000000000109789023025 7
## ESR          0.17329425 0.24120162    4780 0.0000000042107083941919 1
## Tsat         0.37972222 1.06868985    3342 0.0473697366013779916360 5
## Transferrin -0.23193936 0.13194403     357 0.0000000000000090264955 5
```

```
# Hepcidin not affected by BMI

# Obesity how many patients were obese?
mean(!is.na(hep_dat$BMI))
```

```
## [1] 0.699115
```

```
mean(hep_dat$BMI > 30, na.rm = TRUE)  #65%
```

```
## [1] 0.6455696
```

```
wilcox.test(hep_dat$Hepcidin[hep_dat$BMI >= 30], hep_dat$Hepcidin[hep_dat$BMI < 30])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  hep_dat$Hepcidin[hep_dat$BMI >= 30] and hep_dat$Hepcidin[hep_dat$BMI < 30]
## W = 860, p-value = 0.1359
## alternative hypothesis: true location shift is not equal to 0
```

```
# No statistically significant difference


# what about non-continuous value, such as sex of individual
biserial(hep_dat$Hepcidin, hep_dat$Gender)
```

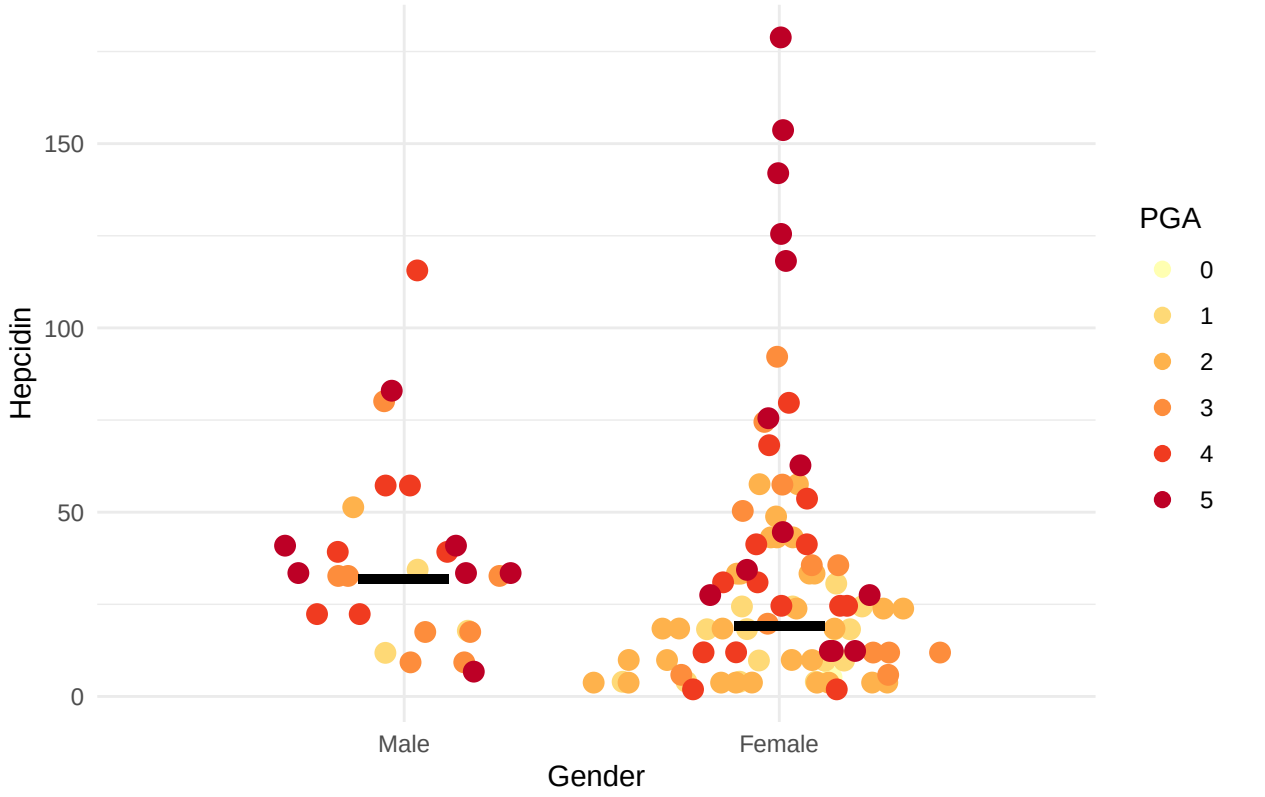
```
##            [,1]
## [1,] -0.1007311
```

```
# -0.1007311 no correlation.

wilcox.test(hep_dat$Hepcidin[hep_dat$Gender == 0], hep_dat$Hepcidin[hep_dat$Gender ==
    1])
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  hep_dat$Hepcidin[hep_dat$Gender == 0] and hep_dat$Hepcidin[hep_dat$Gender == 1]
## W = 1430, p-value = 0.04172
## alternative hypothesis: true location shift is not equal to 0
```

```
# statistically significant difference
fig_hep_sex <- ggplot(hep_dat, aes(x = Gender, y = Hepcidin, fill = PGA)) + geom_dotplot(binaxis = "y",
    stackdir = "center", stackratio = 1, dotsize = 1, col = NA, position = position_jitterd(width = NULL
        height = NULL, quad.points = 100, seed = NA)) + scale_fill_manual(values = seq_col) +
    theme_minimal() + stat_summary(data = hep_dat_3, mapping = aes(x = as.numeric(Gender),
    y = Hepcidin), fun.y = "median", geom = "point", color = "black", inherit.aes = FALSE,
    shape = 95, size = 20) + ggtitle("") + scale_x_discrete(labels = c(`0` = "Male",
```

```
    `1` = "Female"))
fig_hep_sex
```



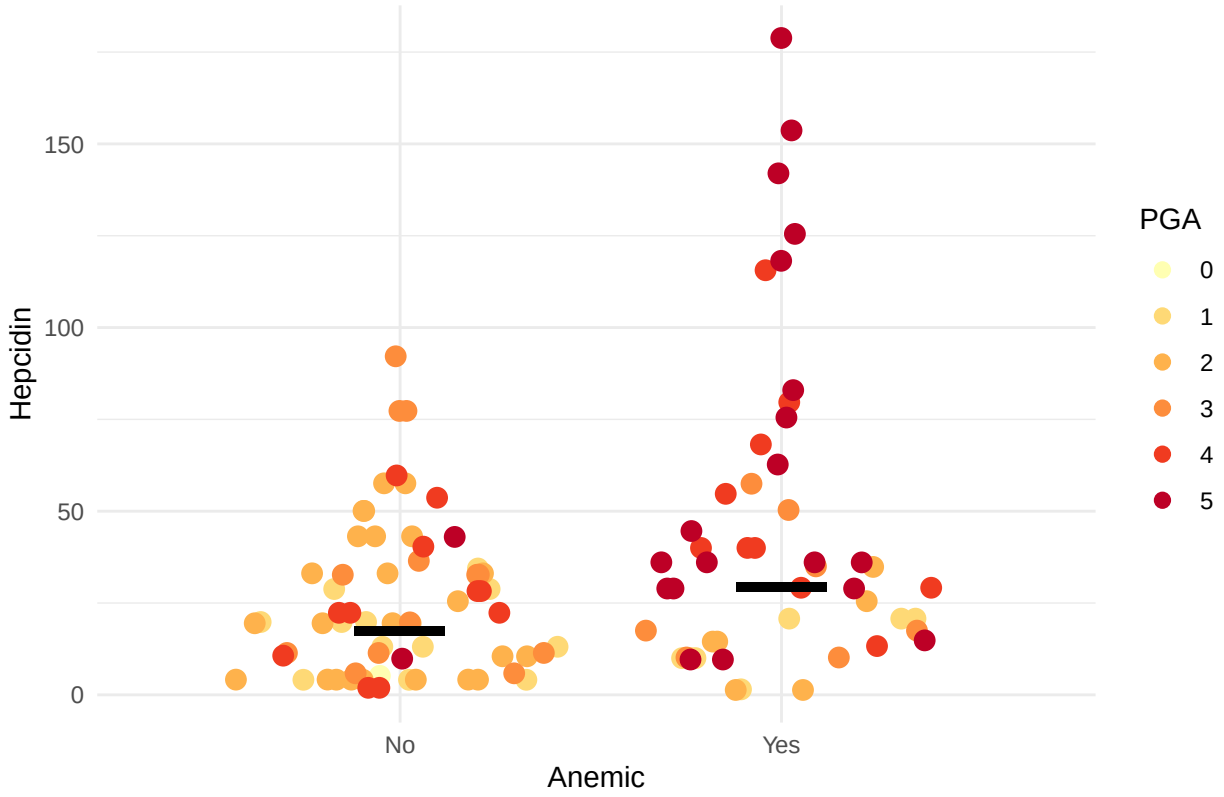

```
# Though the dataset has a majority females.
table(hep_dat$Gender)
```

```
##
##  0  1
## 26 87
```

2 Relationship between Hepcidin and Anemia

2a. Hepcidin levels in anemia vs. non-anemic

```
hep_dat_3 <- hep_dat
hep_dat_3$Anemic <- as.factor(hep_dat_3$Anemic)

seq_col <- brewer.pal(6, "YlOrRd")

fig_hep_anemia <- ggplot(hep_dat_3, aes(x = Anemic, y = Hepcidin, fill = PGA)) +
    geom_dotplot(binaxis = "y", stackdir = "center", stackratio = 1, dotsize = 1,
        col = NA, position = position_jitterd(width = NULL, height = NULL, quad.points = 100,
            seed = NA)) + scale_fill_manual(values = seq_col) + theme_minimal() +
    stat_summary(data = hep_dat_3, mapping = aes(x = as.numeric(Anemic), y = Hepcidin),
        fun.y = "median", geom = "point", color = "black", inherit.aes = FALSE, shape = 95,
        size = 20) + ggtitle("") + scale_x_discrete(labels = c(`0` = "No", `1` = "Yes"))
fig_hep_anemia
```




```
# is the hepcidin level significantly different
wilcox.test(hep_dat_3$Hepcidin[hep_dat_3$Anemic == "0"], hep_dat_3$Hepcidin[hep_dat_3$Anemic ==
    "1"], conf.int = TRUE)
```

```
##
##  Wilcoxon rank sum test with continuity correction
##
## data:  hep_dat_3$Hepcidin[hep_dat_3$Anemic == "0"] and hep_dat_3$Hepcidin[hep_dat_3$Anemic == "1"]
## W = 1143, p-value = 0.01392
## alternative hypothesis: true location shift is not equal to 0
## 95 percent confidence interval:
##  -19.120038  -1.840049
## sample estimates:
## difference in location
##              -9.634167
```

2b. Hepcidin levels predicting anemia

Likely driven by the high number of ACD individuals (who would have elevated hepcidin)

```
binom_anemia_hep <- glm(Anemic ~ Hepcidin, family = binomial(link = "logit"), data = hep_dat)

summary(binom_anemia_hep)
```

```
##
## Call:
## glm(formula = Anemic ~ Hepcidin, family = binomial(link = "logit"),
##     data = hep_dat)
##
## Deviance Residuals:
```

```
##     Min       1Q   Median       3Q      Max
## -1.5930  -0.9855  -0.8421   1.2711   1.5791
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.924625   0.301774  -3.064  0.00218 **
## Hepcidin     0.020216   0.007394   2.734  0.00625 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 154.65  on 112  degrees of freedom
## Residual deviance: 145.05  on 111  degrees of freedom
## AIC: 149.05
##
## Number of Fisher Scoring iterations: 4
```

```
anova(binom_anemia_hep, test = "Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Anemic
##
## Terms added sequentially (first to last)
##
##
##          Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                       112     154.65
## Hepcidin  1   9.6074       111     145.05 0.001938 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
# Hepcidin 1 9.6192 112 146.69 0.001926 **
confint(binom_anemia_hep)
```

```
##                    2.5 %      97.5 %
## (Intercept) -1.542973218 -0.35321635
## Hepcidin     0.006946112  0.03619856
```

```
exp(coef(binom_anemia_hep))  # exponentiated coefficients
```

```
## (Intercept)    Hepcidin
##   0.3966801   1.0204219
```

```
exp(confint(binom_anemia_hep))  # 95% CI for exponentiated coefficients
```

```
##                 2.5 %    97.5 %
## (Intercept) 0.2137446 0.7024252
## Hepcidin    1.0069703 1.0368617
```

```
predict(binom_anemia_hep, type = "response")  # predicted values
```

```
##         1         2         3         5         6         7         8         9
## 0.3268508 0.3540769 0.4154722 0.9364716 0.4942543 0.6797250 0.5590729 0.3124758
##        10        11        12        13        14        15        16        17
```

```
## 0.4023777 0.2915993 0.4450983 0.4315117 0.6114738 0.3416040 0.3486862 0.4376217
##        19        20        22        24        25        26        28        29
## 0.4296778 0.2874409 0.6650759 0.8337588 0.5453267 0.3380664 0.4026694 0.8749734
##        30        31        32        33        34        35        36        38
## 0.4427029 0.5849509 0.2902231 0.4744140 0.4806179 0.3365751 0.5231971 0.8986209
##        39        40        41        42        43        44        47        48
## 0.8121925 0.6460442 0.3187638 0.3991249 0.3548633 0.4470964 0.3767133 0.3187638
##        50        51        52        53        54        55        56        57
## 0.3902865 0.3395609 0.3814718 0.3122152 0.2910984 0.3846728 0.3579238 0.3830949
##        58        59        60        61        62        64        65        66
## 0.3190272 0.4048107 0.4064192 0.3113910 0.3680671 0.3057409 0.6414078 0.3057838
##        67        68        69        70        71        73        74        75
## 0.2930212 0.3092277 0.3672681 0.3134755 0.2985373 0.3246307 0.2895157 0.4533513
##        76        77        78        79        80        81        82        83
## 0.4458474 0.3335574 0.3042838 0.2902231 0.3625830 0.2953719 0.8041386 0.4614793
##        84        85        86        87        89        90        92        93
## 0.5400085 0.3715063 0.3525526 0.3350421 0.3679731 0.2937756 0.5550824 0.4651488
##        95        96        97        98        99       100       101       102
## 0.3261839 0.5282377 0.7188369 0.4954166 0.4301733 0.2911818 0.3327939 0.3967032
##       103       104       105       106       107       109       110       111
## 0.4785996 0.3298379 0.3625830 0.3283203 0.4429523 0.3327041 0.3759541 0.4222631
##       112       113       114       115       116       117       118       119
## 0.4202916 0.5156276 0.3991249 0.4241877 0.4842523 0.5641497 0.5702538 0.4301733
##       122       123       124       125       127       128       129       130
## 0.4728012 0.3290786 0.3712232 0.4446490 0.4861711 0.6670095 0.3380664 0.4426032
##       131
## 0.4467966
```

```
residuals(binom_anemia_hep, type = "deviance")  # residuals
```

```
##          1          2          3          5          6          7          8
##  1.4954943  1.4410004  1.3253977  0.3623149  1.1871858  0.8787114  1.0784019
##          9         10         11         12         13         14         15
##  1.5252727  1.3493436  1.5699521  1.2723680  1.2965034  0.9918500  1.4656760
##         16         17         19         20         22         24         25
##  1.4516080  1.2856131  1.2997844  1.5790744  0.9031656  0.6030110  1.1012449
##         26         28         29         30         31         32         33
##  1.4727613  1.3488064  0.5168400  1.2766020  1.0355939  1.5729624  1.2212083
##         34         35         36         38         39         40         41
##  1.2105229  1.4757601  1.1382417  0.4623723  0.6450084  0.9347592  1.5121541
##         42         43         44         47         48         50         51
##  1.3553456  1.4394600  1.2688428  1.3973338  1.5121541  1.3717684  1.4697632
##         52         53         54         55         56         57         58
## -0.9802167 -0.8651928 -0.8295041 -0.9854959 -0.9413270 -0.9828937 -0.8766218
##         59         60         61         62         64         65         66
## -1.0187010 -1.0213539 -0.8638075 -0.9580941 -0.8542950 -1.4321798 -0.8543674
##         67         68         69         70         71         73         74
## -0.8327719 -0.8601686 -0.9567743 -0.8673101 -0.8421254 -0.8859973 -0.8268112
##         75         76         77         78         79         80         81
## -1.0990440  1.2710457 -0.9008898 -0.8518374 -0.8280152  1.4244310 -0.8367618
##         82         83         84         85         86         87         89
##  0.6602782  1.2436384 -1.2462321 -0.9637732 -0.9324353  1.4788504  1.4140335
##         90         92         93         95         96         97         98
## -0.8340530 -1.2726871  1.2372533 -0.8885923 -1.2257895 -1.5929974 -1.1696343
##         99        100        101        102        103        104        105
```

```
## -1.0605875 -0.8296460 -0.8996179 -1.0053318 -1.1412598 -0.8946906 -0.9490323
##        106        107        109        110        111        112        113
## -0.8921588 -1.0817619 -0.8994684 -0.9711142 -1.0475081 -1.0442509 -1.2040774
##        114        115        116        117        118        119        122
## -1.0093247 -1.0506888 -1.1507715 -1.2887642 -1.2996619 -1.0605875 -1.1315279
##        123        124        125        127        128        129        130
## -0.8934241 -0.9633057 -1.0845782 -1.1540060 -1.4829978 -0.9083942 -1.0811825
##        131
## -1.0881448
```

2c. Prevalence of anemia and iron deficiency

As previously found in Tennant et al. (1968), there is a high proportion of anemia in our cohort (43%). Tennant et al. (1968) examined 42 patients, finding 10 had "marked" anemia (Hb <10). Additionally, the found ESR to elevated but only had results for 13 patients (11/13 had elevated ESR). Though a large cross-sectional study in Denmark (Miller et al. 2016) demonstrated no difference in Hb level after age-sex-smoking-adjusted analyses between HS patients and 20,780 of the general population. Though it should be noted that they only found 4.65% of their HS patients to have anemia. Another European (Poland) study, Ponikowska et al 2020 only found 3 HS patients (3/74; 4%) to have anemia.

Our high prevalence may be present in the Bronx

```
table(hep_dat$Anemic)
```

```
##
##  0  1
## 64 49
```

```
prop.table(table(hep_dat$Anemic))  #43% of our population
```

```
##
##         0         1
## 0.5663717 0.4336283
```

A recent (2020) study, Ponikowska et al 2020 "Deranged Iron Status Evidenced by Iron Deficiency Characterizes Patients with Hidradenitis Suppurativa" defined iron deficiency as ferrtin <100ug/L or ferritin 100-299 with TSat <20%. They found that 75% of their population met iron deficient criteria.

```
# unfortunately, missing ferritin and Tsat information for 9 non-anemic
# individuals
sum(is.na(hep_dat$Ferritin))  #7
```

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

```
sum(is.na(hep_dat$Tsat))  #9
```

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

```
hep_dat_compare <- hep_dat[which(!(is.na(hep_dat$Tsat) | is.na(hep_dat$Ferritin))),
    ]
nrow(hep_dat_compare)  #104
```

```
## [1] 104
```

```
# Ponikowska study
hep_dat_compare$Ponikowska <- 0
hep_dat_compare$Ponikowska[which(hep_dat_compare$Ferritin < 100)] <- 1
hep_dat_compare$Ponikowska[which(hep_dat_compare$Ferritin < 299 & hep_dat_compare$Tsat <
    20)] <- 1
```

```r
sum(hep_dat_compare$Ponikowska == 1)  # 89 individuals
```

```
## [1] 89
```

```r
prop.table(table(hep_dat_compare$Ponikowska == 1))
```

```
##
##     FALSE      TRUE
## 0.1442308 0.8557692
```

```r
# 85.57692 % of patient are iron deficient as defined by parameters in the
# Ponikowska study
```

2d. Classifying anemia in our cohort

Prior studies classified IDA, IDA/ACD, ACD by the following:

IDA 1) absence of inflammation AND 2) i. T-sat <20% and ferritin <30 ug/l (van Santen et al, 2011; Scholz et al, 2019) OR ii. sTfR-index >=1 mg/ug (van Santen et al, 2011) OR iii. T-sat levels <15% together with ferritin <50 ug/l and MCH in the lowest quintile (Scholz et al, 2019 as Thurnnham et al. 2010 showed that mean ferritin concentrations that were 50% (P , 0.001) and 38% (P , 0.002) higher when CRP and AGP were elevated by inflammation, respectively)

Inflammation was defined in prior studies as: 1) RA study, inflammation solely by CRP >10 mg/ml (1mg/L) OR DAS28-ESR (Scholz et al, 2019) 2) RA study, active inflammation (defined as a CRP level of >= 10 mg/ml or an ESR of >= 30 mm/hour) (van Santen et al, 2011; Khalaf et al.[30996848]) 3) IBD study, only using CRP > 5 mg/ml AND clinical disease activity indices, CDAI (Crohn's disease activity index) for CD and MTWAI (Modified Truelove and Witts activity index) for UC (Mecklenburg et al.)

ACD was defined as the following in van Santen et al, 2011: 1) Presence of inflammation 2) i. transferrin saturation <20% and ferritin level >=100 g/ml OR ii. sTfR index < 1 mg/micro-g and ferritin level >=30 micro-g/ml

IDA/ACD 1) Presence of inflammation 2) i. ferritin < 100 ng/mL and Tsat < 20% OR ii. 2) sTfR index 1 mg/micro-g.

Notes: ESR 30 is the upper limit of normal for women.

Our CRP classifiation was as follows: CRP 0 <1 mg/dL CRP 1 1-5 mg/dL CRP 2 >5 mg/dL

Total body iron stores were classified by ferritin class (0 for ferritin<20; 1 for 20==200).

Our criteria: IDA 1) absence of inflammation (CRP <10 mg/mL or ESR <50)) AND 2) i. T-sat <20% and ferritin <30 ug/l OR ii. T-sat levels <15% together with ferritin <50 ug/l

ACD 1) Presence of inflammation (CRP >=10 mg/mL or ESR >=50) 2) i. transferrin saturation <20% and ferritin level >=100 g/ml OR ii. Ferritin level >= 200 g/mL

IDA/ACD 1) Presence of inflammation (CRP >=10 mg/mL or ESR >=50) 2) ferritin < 100 ng/mL and Tsat < 20%

```r
# Determining anemic status
anemia_dat <- hep_dat
anemia_dat$anemia_type <- "Not anemic"

# IDA ferritin <30 ug/l
anemia_dat$anemia_type[which(anemia_dat$Anemic == 1 & ((anemia_dat$Ferritin < 30)))] <- "IDA"
nrow(anemia_dat[anemia_dat$anemia_type == "IDA", ])  #10
```

```
## [1] 10
```

```r
## defining inflammation ACD
anemia_dat$anemia_type[which(anemia_dat$Anemic == 1 & (anemia_dat$CRP.class >= 1 |
    anemia_dat$ESR >= 50) & ((anemia_dat$Tsat < 20 & anemia_dat$Ferritin >= 100) |
    anemia_dat$Ferritin >= 200))] <- "ACD"
nrow(anemia_dat[anemia_dat$anemia_type == "ACD", ])  #7
```

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

```r
# ACD/IDA
anemia_dat$anemia_type[which(anemia_dat$Anemic == 1 & (anemia_dat$CRP.class >= 1 |
    anemia_dat$ESR >= 50) & anemia_dat$Tsat < 20 & anemia_dat$Ferritin < 100)] <- "ACD/IDA"
nrow(anemia_dat[anemia_dat$anemia_type == "ACD/IDA", ])  # 26
```

```
## [1] 26
```

```r
# Other anemia
anemia_dat$anemia_type[which(anemia_dat$anemia_type == "Not anemic" & anemia_dat$Anemic ==
    1)] <- "Other anemia"

table(anemia_dat$anemia_type)
```

```
##
##          ACD      ACD/IDA          IDA   Not anemic Other anemia
##            7           26            4           64           12
```

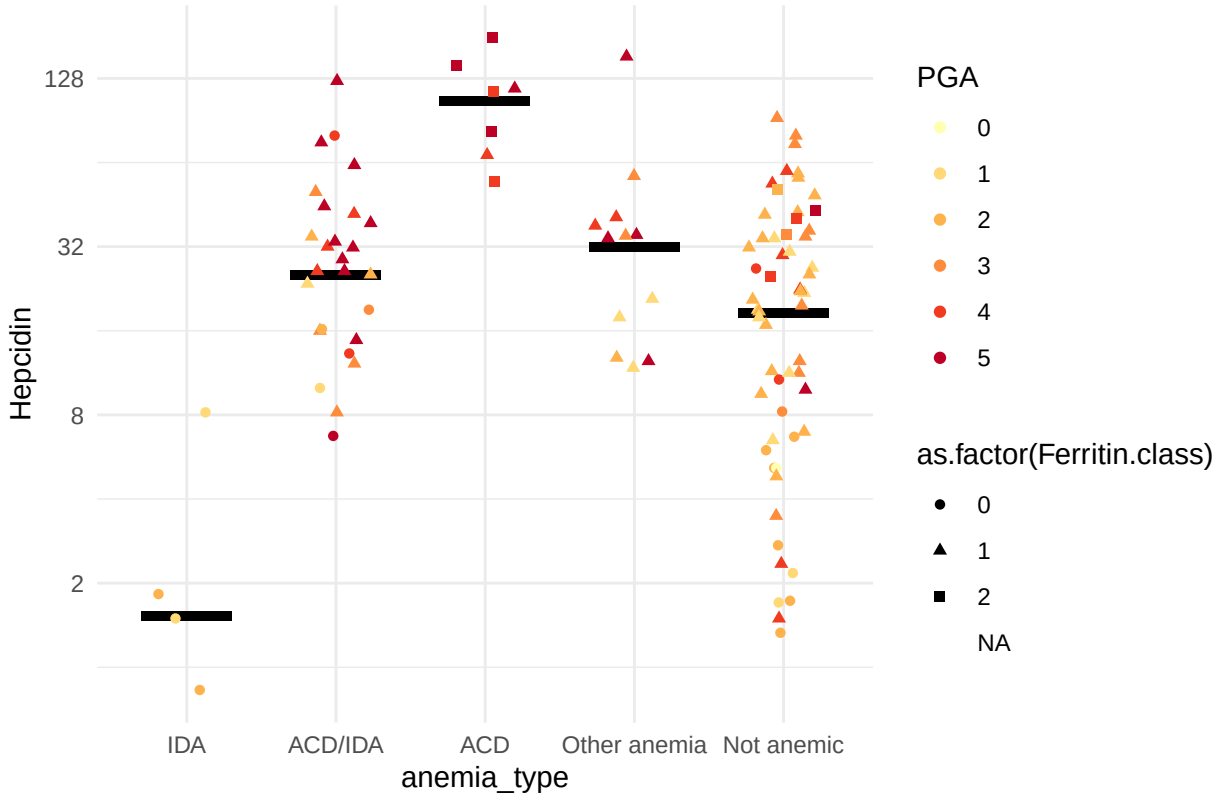
```r
prop.table(table(anemia_dat$anemia_type[anemia_dat$Anemic == 1]))
```

```
##
##          ACD      ACD/IDA          IDA Other anemia
##   0.14285714   0.53061224   0.08163265   0.24489796
```

```r
anemia_dat$anemia_type <- factor(anemia_dat$anemia_type, levels = c("IDA", "ACD/IDA",
    "ACD", "Other anemia", "Not anemic"))
table(anemia_dat$Ferritin.class)
```

```
##
##  0  1  2
## 22 74 10
```

```r
anemia_dat$CRP_class <- anemia_dat$CRP.class
fig_hep_anemia_type <- ggplot(anemia_dat, aes(x = anemia_type, y = Hepcidin, col = PGA,
    shape = as.factor(Ferritin.class))) + # geom_dotplot(binaxis='y', stackdir='center', stackratio=1,
# position = position_jitterd(width = .2, height = NULL, quad.points = 100, #seed
# = NA)) +
stat_summary(data = anemia_dat, mapping = aes(x = anemia_type, y = Hepcidin), fun.y = "median",
    geom = "point", color = "black", inherit.aes = FALSE, shape = 95, size = 20) +
    geom_point(position = position_jitterd(width = 0.3, height = NULL, quad.points = 100,
        seed = NA)) + scale_color_manual(values = seq_col) + theme_minimal() + ggtitle("") +
    scale_y_continuous(trans = "log2")
fig_hep_anemia_type
```



3 Using anemia to distinguish anemia type

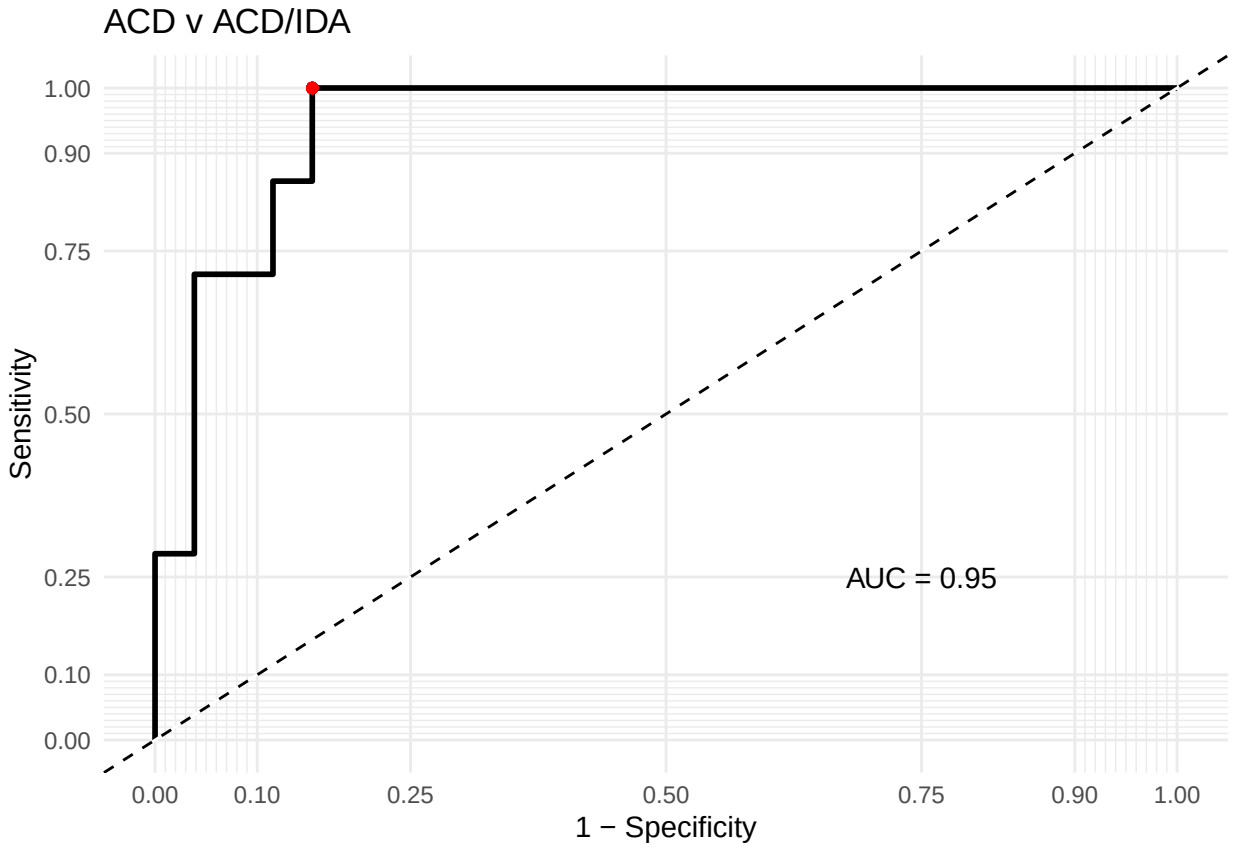
3a Hepcidin distinguishes ACD/IDA and ACD

First, we will examine the ability of hepcidin to discern between ACD/IDA and ACD

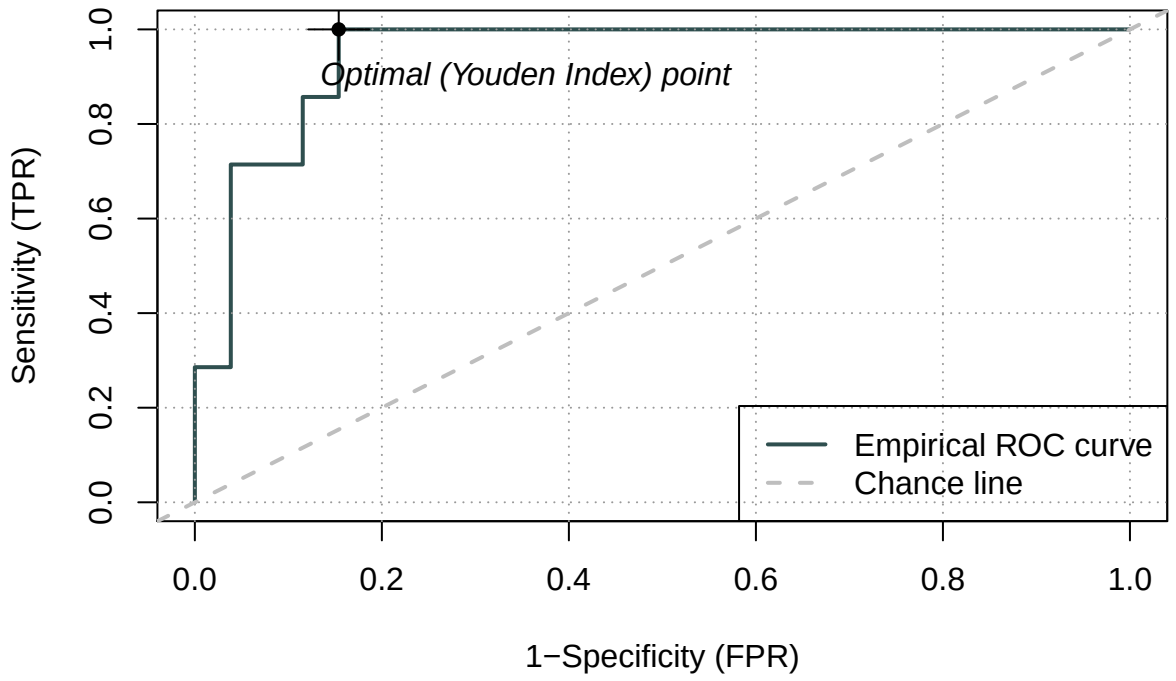
```
# ROC of IDA/ACD vs ACD
anemia_dat_roc_1 <- anemia_dat[(anemia_dat$anemia_type == "ACD/IDA" | anemia_dat$anemia_type ==
    "ACD"), ]
anemia_dat_roc_1$anemia_type <- factor(anemia_dat_roc_1$anemia_type, levels = c("ACD/IDA",
    "ACD"))

rocfit_1 <- pROC::roc(anemia_dat_roc_1$anemia_type, anemia_dat_roc_1$Hepcidin)
pROC::auc(rocfit_1)
```

```
## Area under the curve: 0.9451
```

```
youden_coord_1 <- pROC::coords(rocfit_1, x = "b")

rocplot_1 <- ggplot(anemia_dat_roc_1, aes(m = Hepcidin, d = as.numeric(anemia_type))) +
    geom_roc(n.cuts = 0)
rocplot_1 <- rocplot_1 + 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_1)$AUC, 2))) + geom_point(aes(x = 1 - youden_coord_1$specificity,
    y = youden_coord_1$sensitivity), colour = "red") + geom_abline(slope = 1, intercept = 0,
    lty = "dashed") + ggtitle("ACD v ACD/IDA")
rocplot_1
```




```
# examining other ROC package visulaizations and computing bootstrapped
# calculations of the youden index.
roc_1 <- rocit(score = anemia_dat_roc_1$Hepcidin, class = anemia_dat_roc_1$anemia_type,
    negref = "ACD/IDA")
plot(roc_1, values = TRUE)
```



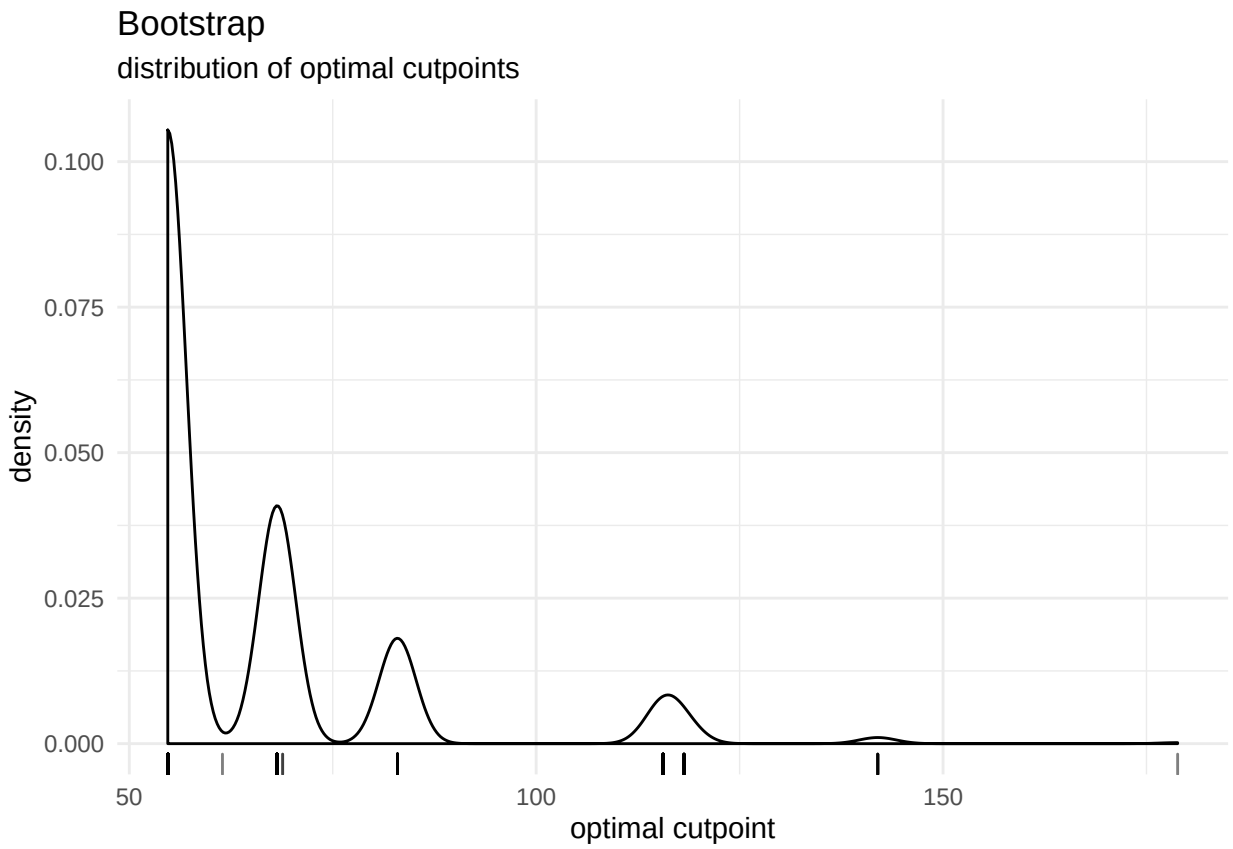
```r
set.seed(200)
ciAUC_boot_1 <- ciAUC(roc_1, level = 0.9, nboot = 1000)
print(ciAUC_boot_1)
```

```
##
##    estimated AUC : 0.945054945054945
##    AUC estimation method : empirical
##
##    bootstrap CI of AUC with 1000 boot samples
##    confidence level = 90%
##    lower = 0.868131868131868     upper = 1
```

```r
youden_coord_1
```

```
##   threshold specificity sensitivity
## 1     52.53   0.8461538           1
```

```r
opt_cut_1 <- cutpointr(anemia_dat_roc_1, Hepcidin, anemia_type, boot_runs = 1000)
plot_cut_boot(opt_cut_1) + theme_minimal()
```



3a Hepcidin distinguishes IDA and ACD/IDA

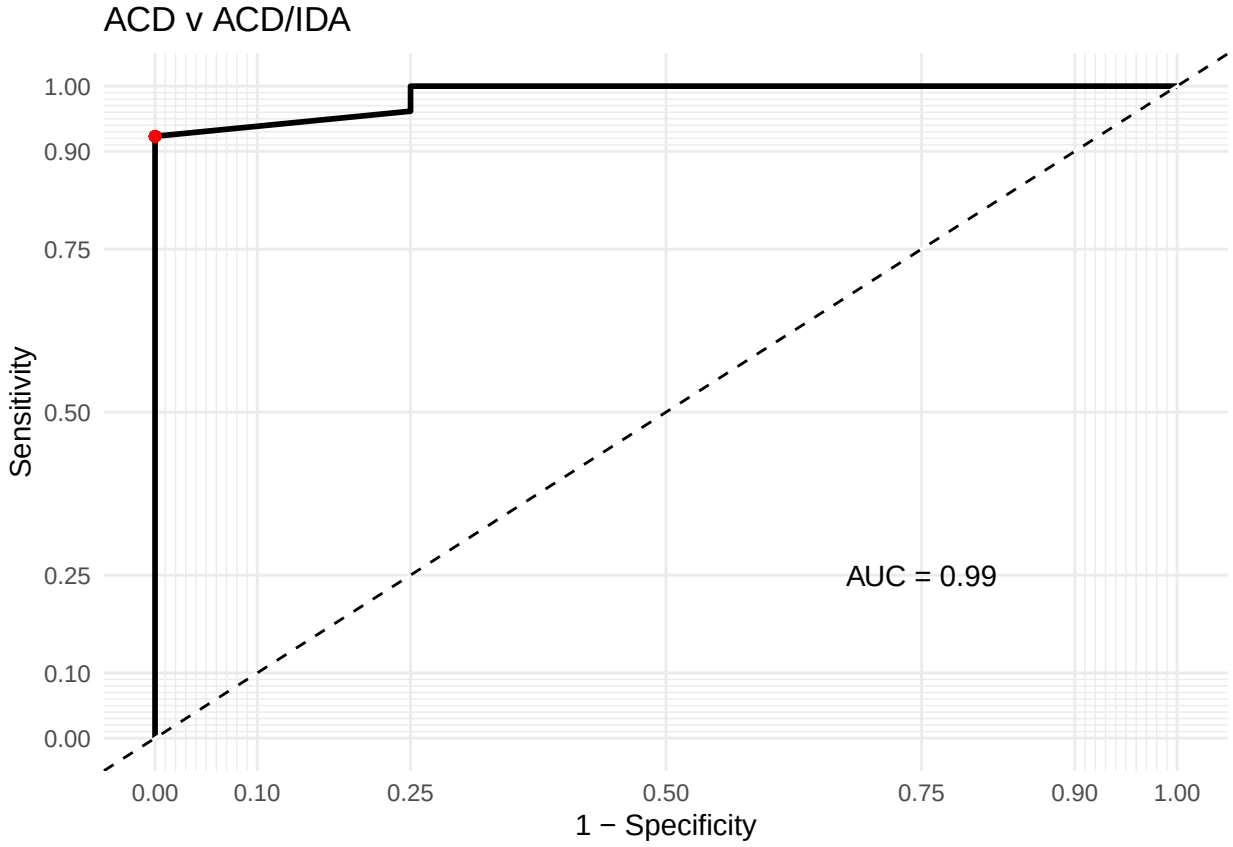
Second, we will examine the ability of hepcidin to discern between IDA and ACD/IDA

```r
# ROC of IDA/ACD vs IDA
anemia_dat_roc_4 <- anemia_dat[(anemia_dat$anemia_type == "ACD/IDA" | anemia_dat$anemia_type ==
    "IDA"), ]
anemia_dat_roc_4$anemia_type <- factor(anemia_dat_roc_4$anemia_type, levels = c("IDA",
```

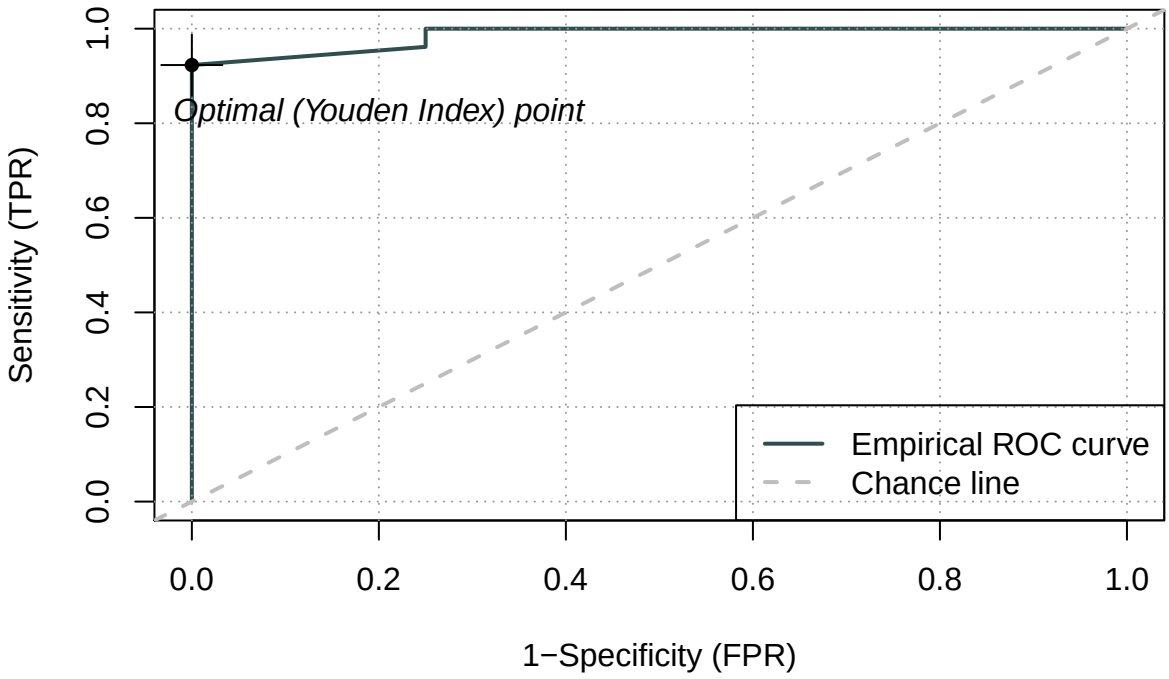
```
    "ACD/IDA"))

rocfit_4 <- pROC::roc(anemia_dat_roc_4$anemia_type, anemia_dat_roc_4$Hepcidin)
pROC::auc(rocfit_4)
```

```
## Area under the curve: 0.9856
```

```
youden_coord_4 <- pROC::coords(rocfit_4, x = "b")

rocplot_4 <- ggplot(anemia_dat_roc_4, aes(m = Hepcidin, d = as.numeric(anemia_type))) +
    geom_roc(n.cuts = 0)
rocplot_4 <- rocplot_4 + 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_4)$AUC, 2))) + geom_point(aes(x = 1 - youden_coord_4$specificity,
    y = youden_coord_4$sensitivity), colour = "red") + geom_abline(slope = 1, intercept = 0,
    lty = "dashed") + ggtitle("ACD v ACD/IDA")
rocplot_4
```




```
# examining other ROC package visulaizations and computing bootstrapped
# calculations of the youden index.
roc_4 <- rocit(score = anemia_dat_roc_4$Hepcidin, class = anemia_dat_roc_4$anemia_type,
    negref = "IDA")
plot(roc_4, values = TRUE)
```



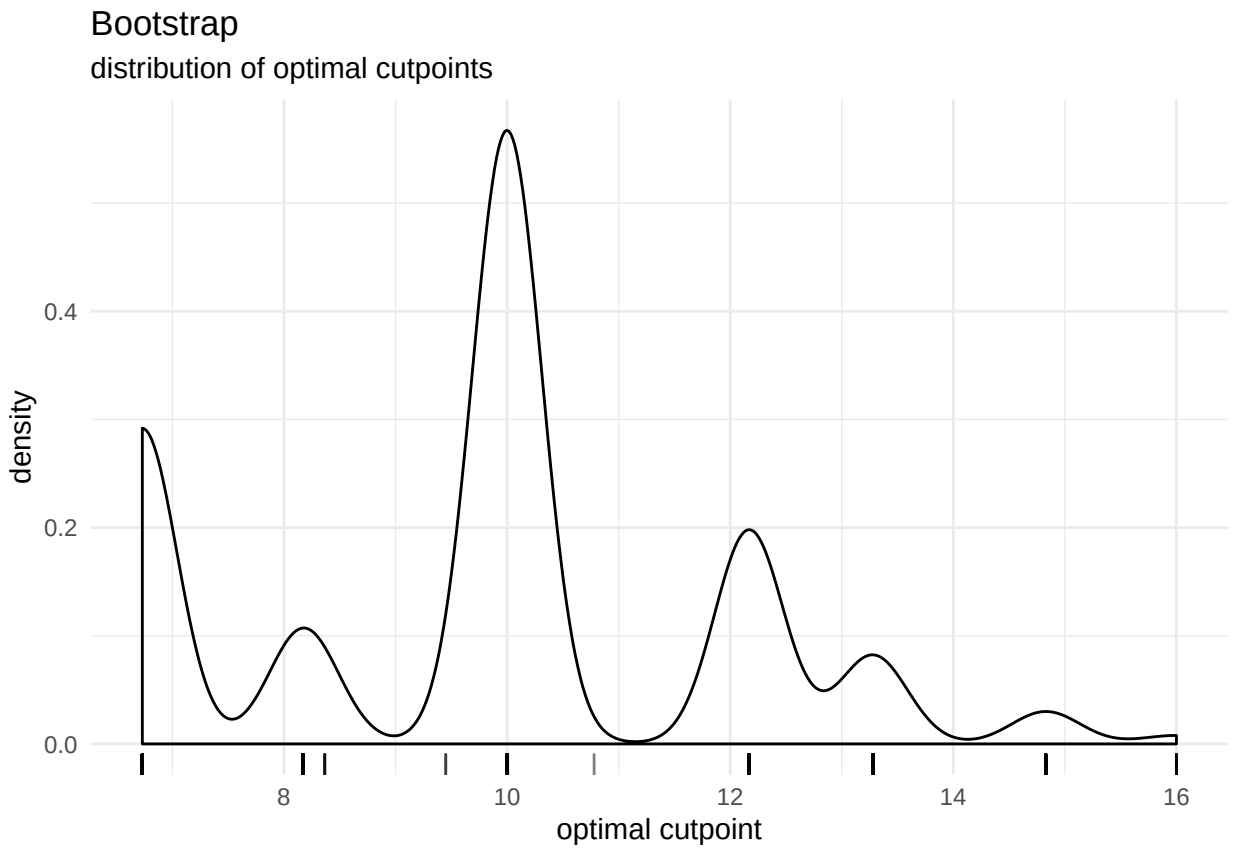

```
set.seed(200)
ciAUC_boot_4 <- ciAUC(roc_4, level = 0.9, nboot = 1000)
print(ciAUC_boot_4)
```

```
##
##    estimated AUC : 0.985576923076923
##    AUC estimation method : empirical
##
##    bootstrap CI of AUC with 1000 boot samples
##    confidence level = 90%
##    lower = 0.942307692307692     upper = 1
```

```
youden_coord_4
```

```
##   threshold specificity sensitivity
## 1     9.085           1   0.9230769
```

```
opt_cut_4 <- cutpointr(anemia_dat_roc_4, Hepcidin, anemia_type, boot_runs = 1000)
plot_cut_boot(opt_cut_4) + theme_minimal()
```



4 Making Demographic/blood parameter panel table

This code chunk is not evaluated but kept for reference for how Table1 was generated.

```
tab1_dat <- anemia_dat

tab1_dat$Gender <- factor(tab1_dat$Gender, levels = c(0, 1), labels = c("Male", "Female"))
label(tab1_dat$Gender) <- "Sex"
units(tab1_dat$Age) <- "Years"
tab1_dat$Race <- factor(tab1_dat$Race, levels = c(1, 0, 2, 3), labels = c("African-American",
    "Caucasian", "Other", "Unavailable/Declined"))
label(tab1_dat$Race) <- "Race"
tab1_dat$PGA <- factor(tab1_dat$PGA, levels = c(0, 1, 2, 3, 4, 5), labels = c("Clear (0)",
    "Minimal (1)", "Mild (2)", "Moderate (3)", "Severe (4)", "Very Severe (5)"))
label(tab1_dat$PGA) <- "HS-PGA score"
units(tab1_dat$Hepcidin) <- "ng/mL"

units(tab1_dat$Hb) <- "g/dL"
units(tab1_dat$MCV) <- "fL"
units(tab1_dat$PLT) <- "K/uL"
units(tab1_dat$Fe) <- "ug/dL"
units(tab1_dat$Transferrin) <- "mg/dL"
units(tab1_dat$Tsat) <- "%"
units(tab1_dat$Ferritin) <- "ng/mL"
units(tab1_dat$CRP) <- "mg/dL"
units(tab1_dat$ESR) <- "mm/h"
```

```r
table1(~Gender + Age + Race + PGA + Hb + MCV + PLT + Fe + Transferrin + Tsat + Ferritin +
    CRP + ESR | anemia_type, data = tab1_dat, overall = "Overall", cont.rmstat = list(c("miss")))

rndr <- function(x, name, ...) {
    if (!is.numeric(x))
        return(render.categorical.default(x))
    what <- "Median [Min, Max]"
    parse.abbrev.render.code(c("", what))(x)
}

table1(strata, labels, groupspan = c(1, 3, 1), render.continuous = c(. = "Mean (CV%)",
    . = "Median [Min, Max]", `Geo. mean (Geo. CV%)` = "GMEAN (GCV%)"))

table1(~Gender + Age + Race + PGA + Hb + MCV + PLT + Fe + Transferrin + Tsat + Ferritin +
    CRP + ESR | anemia_type, data = tab1_dat, overall = "Overall", render = rndr)

table1(~Gender + Age + BMI + Race + PGA + Hb + MCV + PLT + Fe + Transferrin + Tsat +
    Ferritin + CRP + ESR | anemia_type, data = tab1_dat, overall = "Overall", render.continuous = c(. =
```

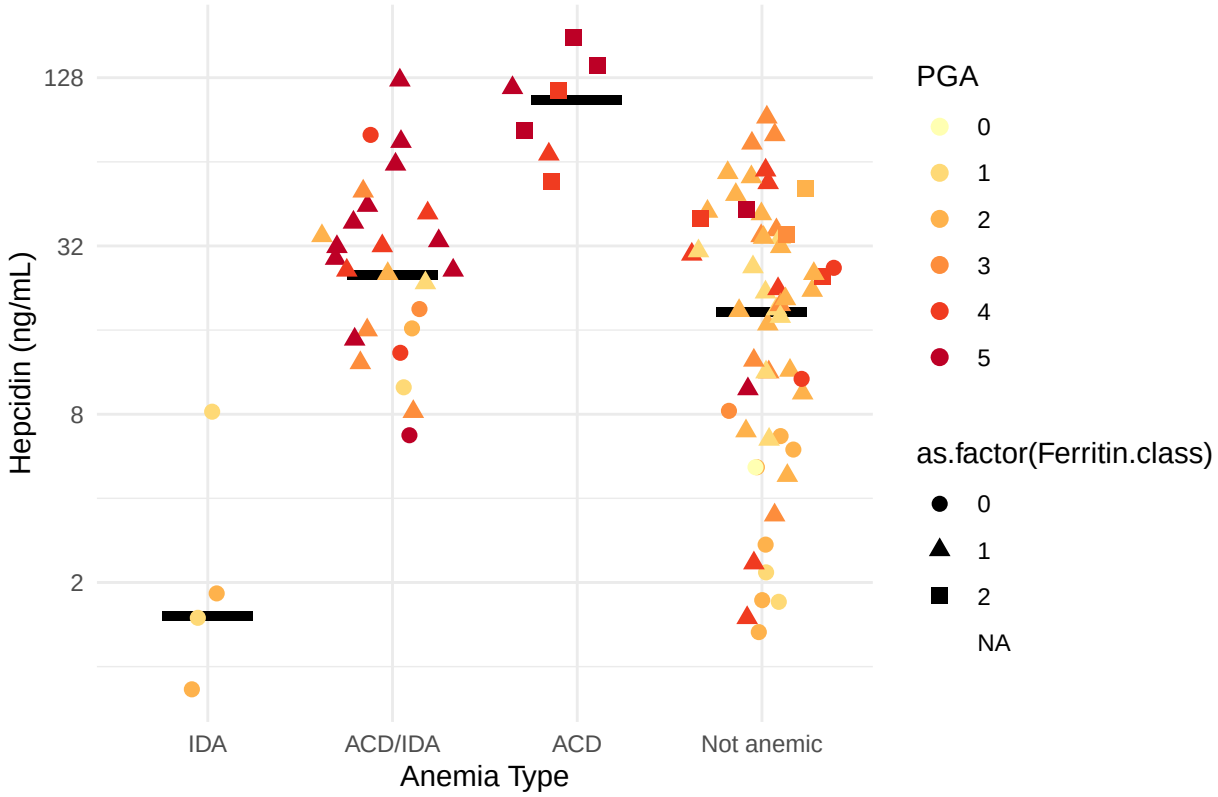
5 Figure 1A

This code chunk creates figure 1 Hepcidin levels by anemia type and performs Kruskal-Wallis ANOVA with Dunn's multiple comparisons test.

```r
anemia_dat_fig1 <- anemia_dat[!(anemia_dat$anemia_type == "Other anemia"), ]

# Obtain signficiance
dunn.test(anemia_dat_fig1$Hepcidin, g = anemia_dat_fig1$anemia_type, method = "bonferroni",
    kw = TRUE, label = TRUE, wrap = FALSE, table = TRUE, list = FALSE, rmc = FALSE,
    alpha = 0.05, altp = FALSE)
```

```
##   Kruskal-Wallis rank sum test
##
## data: x and group
## Kruskal-Wallis chi-squared = 27.5406, df = 3, p-value = 0
##
##
##                            Comparison of x by group
##                                  (Bonferroni)
## Col Mean-|
## Row Mean |        ACD    ACD/IDA        IDA
## ---------+---------------------------------
##  ACD/IDA |   3.010979
##          |    0.0078*
##          |
##      IDA |   4.745143   3.150434
##          |    0.0000*    0.0049*
##          |
## Not anem |   4.197057   1.671444  -2.528841
##          |    0.0001*     0.2839     0.0343
##
## alpha = 0.05
## Reject Ho if p <= alpha/2
```

```
fig_1 <- ggplot(anemia_dat_fig1, aes(x = anemia_type, y = Hepcidin, col = PGA, shape = as.factor(Ferriti
    # geom_dotplot(binaxis='y', stackdir='center', stackratio=1, dotsize=.75, col=NA,
# position = position_jitterd(width = .2, height = NULL, quad.points = 100, #seed
# = NA)) +
stat_summary(data = anemia_dat_fig1, mapping = aes(x = anemia_type, y = Hepcidin),
    fun.y = "median", geom = "point", color = "black", inherit.aes = FALSE, shape = 95,
    size = 20) + geom_point(size = 2.5, position = position_jitterd(width = 0.4,
    height = NULL, quad.points = 100, seed = NA)) + scale_color_manual(values = seq_col) +
    ylab("Hepcidin (ng/mL)") + xlab("Anemia Type") + theme_minimal() + ggtitle("") +
    scale_y_continuous(trans = "log2")
fig_1
```



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] dunn.test_1.3.5        table1_1.2             psych_2.0.12
##  [4] polycor_0.7-10         mblm_0.12.1            Hmisc_4.3-0
##  [7] Formula_1.2-3          pastecs_1.3.21         cutpointr_1.0.32
## [10] OptimalCutpoints_1.1-4 plotROC_2.2.1          pROC_1.16.1
## [13] ROCit_2.1.1            lubridate_1.7.9        lattice_0.20-38
## [16] overlapping_1.6        testthat_2.3.1         flexplot_0.7.5
## [19] scales_1.1.0           AER_1.2-8              survival_3.1-8
## [22] sandwich_2.5-1         lmtest_0.9-37          zoo_1.8-7
## [25] car_3.0-6              carData_3.0-3          MASS_7.3-51.5
## [28] RColorBrewer_1.1-2     ggpubr_0.2.4           magrittr_1.5
## [31] reshape2_1.4.3         ggthemes_4.2.0         ggplot2_3.2.1
## [34] data.table_1.12.8
##
## loaded via a namespace (and not attached):
##  [1] nlme_3.1-143        tools_3.5.1         backports_1.1.5
##  [4] R6_2.4.1            rpart_4.1-15        lazyeval_0.2.2
##  [7] colorspace_1.4-1    nnet_7.3-12         withr_2.1.2
## [10] tidyselect_0.2.5    gridExtra_2.3       mnormt_1.5-5
## [13] curl_4.3            compiler_3.5.1      formatR_1.7
## [16] htmlTable_1.13.3    labeling_0.3        checkmate_1.9.4
## [19] stringr_1.4.0       digest_0.6.23       foreign_0.8-75
## [22] rmarkdown_2.1       rio_0.5.16          base64enc_0.1-3
## [25] pkgconfig_2.0.3     htmltools_0.4.0     htmlwidgets_1.5.1
## [28] rlang_0.4.7         readxl_1.3.1        rstudioapi_0.10
## [31] farver_2.0.3        generics_0.0.2      acepack_1.4.1
## [34] dplyr_0.8.3         zip_2.0.4           Matrix_1.2-18
## [37] Rcpp_1.0.3          munsell_0.5.0       abind_1.4-5
## [40] lifecycle_0.2.0     stringi_1.4.5       yaml_2.2.0
## [43] plyr_1.8.5          grid_3.5.1          parallel_3.5.1
## [46] forcats_0.4.0       crayon_1.3.4        haven_2.2.0
## [49] splines_3.5.1       hms_0.5.3           knitr_1.27
## [52] pillar_1.4.3        boot_1.3-24         ggsignif_0.6.0
## [55] codetools_0.2-16    glue_1.3.1          evaluate_0.14
## [58] latticeExtra_0.6-28 vctrs_0.3.4         foreach_1.4.7
## [61] cellranger_1.1.0    tidyr_1.0.0         gtable_0.3.0
## [64] purrr_0.3.3         assertthat_0.2.1    xfun_0.12
## [67] openxlsx_4.1.4      tibble_3.0.3        iterators_1.0.12
## [70] cluster_2.1.0       ellipsis_0.3.0
```