Charactersing effect of anaemia on mortality in severe malaria

Contents

Background

This looks at the severe malaria legacy dataset from MORU

Imputation of missing variables

Quite a lot of the important covariates are missing in the older studies. We use linear regression to estimate these unknown variables:

- Mising base deficit is imputed using bicarbonate (if available) else using respiratory rate
- Missing Blood urea nitrogen is imputed using creatinine

Impute base deficit from bicarbonate

```
BD_and_bicarbonate = !is.na(Leg_data$BD) & !is.na(Leg_data$bicarbonate)
print(paste('We have ', sum(BD_and_bicarbonate), 'observations for both bicarbonate and base deficit'))
```

```
## [1] "We have  5048 observations for both bicarbonate and base deficit"
```

```
mod_impute1 = lmer(BD ~ bicarbonate + (1 | studyID), data= Leg_data[BD_and_bicarbonate,])
missing_BD = is.na(Leg_data$BD)
Available_Bicarbonate = !is.na(Leg_data$bicarbonate)
print(paste(sum(missing_BD & Available_Bicarbonate), 'observations will now be imputed'))
```

```
## [1] "309 observations will now be imputed"
```

```
# impute with model
Leg_data$BD[missing_BD & Available_Bicarbonate] = predict(mod_impute1,newdata=Leg_data[missing_BD & Avai
```

Impute base deficit from respiratory rate

```
BD_and_rr = !is.na(Leg_data$BD) & !is.na(Leg_data$rr)
print(paste('We have ', sum(BD_and_rr), 'observations for both resp rate and base deficit'))
```

```
## [1] "We have  6560 observations for both resp rate and base deficit"
```

```
mod_impute2 = lmer(BD ~ rr + (1 | studyID), data= Leg_data[BD_and_rr,])
missing_BD = is.na(Leg_data$BD)
```

```
Available_rr = !is.na(Leg_data$rr)
print(paste(sum(missing_BD & Available_rr), 'observations will now be imputed'))
```

```
## [1] "2662 observations will now be imputed"
```

```
Leg_data$BD[missing_BD & Available_rr] = predict(mod_impute2,newdata=Leg_data[missing_BD & Available_rr
```

Impute blood urea nitrogen from creatinine:

```
BUN_and_cr = !is.na(Leg_data$BUN) & !is.na(Leg_data$creatinine)
print(paste('We have ', sum(BUN_and_cr), 'observations for both blood urea nitrogen and creatinine'))
```

```
## [1] "We have  1433 observations for both blood urea nitrogen and creatinine"
```

```
mod_impute3 = lmer(BUN ~ creatinine + (1 | studyID), data= Leg_data[BUN_and_cr,])
missing_BUN = is.na(Leg_data$BUN)
Available_cr = !is.na(Leg_data$creatinine)
print(paste(sum(missing_BUN & Available_cr), 'observations will now be imputed'))
```

```
## [1] "679 observations will now be imputed"
```

```
Leg_data$BUN[missing_BUN & Available_cr] = predict(mod_impute3,newdata=Leg_data[missing_BUN & Available_
```

Resulting data we can now use: The contributions of the different studies:

```
vars_interest = c('outcome','HCT','LPAR_pct','BD','BUN','poedema','convulsions','coma','AgeInYear','drug
complete_cases = apply(Leg_data[,vars_interest], 1, function(x) sum(is.na(x))) == 0
Complete_Leg_data = Leg_data[complete_cases,] # for the model fitting
Complete_Leg_data$studyID = as.factor(as.character(Complete_Leg_data$studyID))
# Whole dataset
table(Leg_data$studyID)
```

```
## 
##          AAV           AQ     AQGambia      AQUAMAT Core Malaria 
##          370          560          579         5494         1121 
##    SEAQUAMAT 
##         1461 
```

```
# in the complete dataset (all variables recorded)
table(Complete_Leg_data$studyID)
```

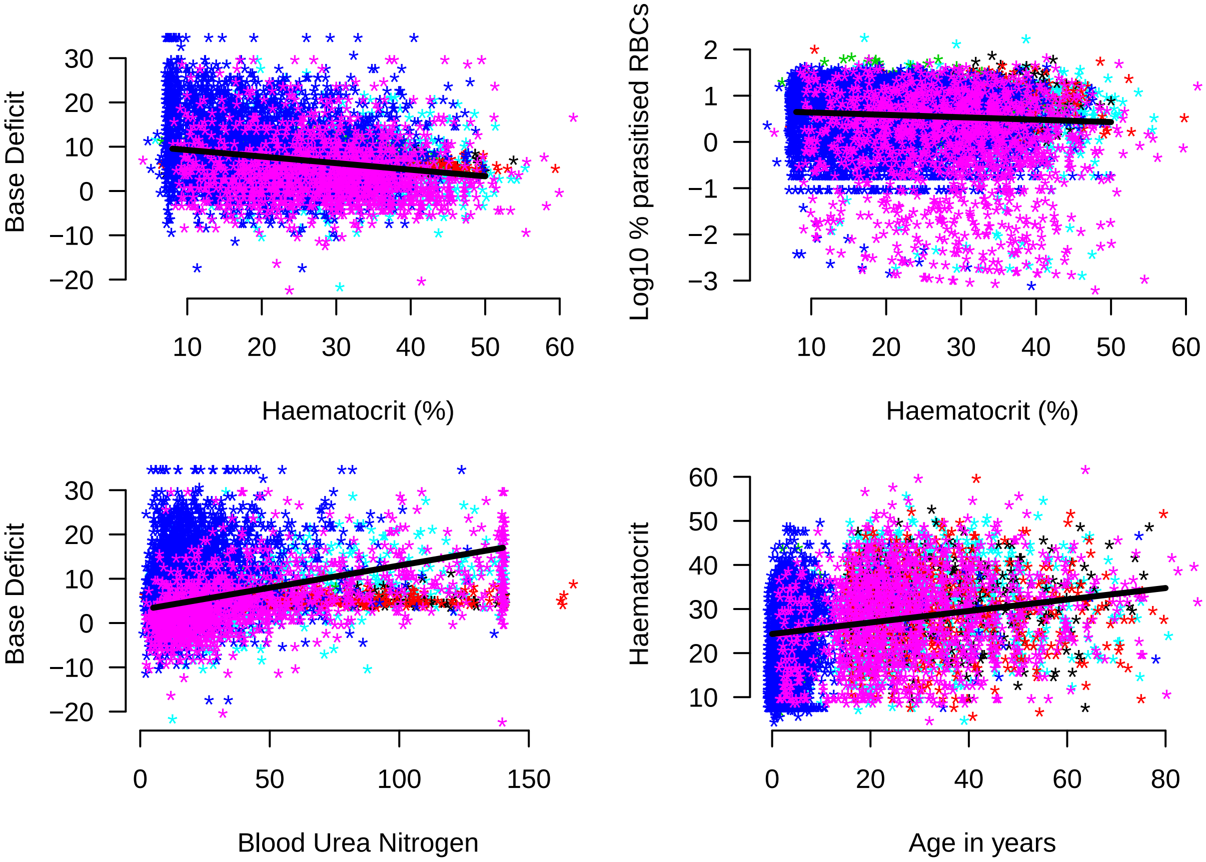
```
## 
##          AAV           AQ     AQGambia      AQUAMAT Core Malaria 
##          213          150          168         3666          639 
##    SEAQUAMAT 
##         1333 
```

```
Complete_Leg_data$drug_AS = 0
Complete_Leg_data$drug_AS[Complete_Leg_data$drug_class=='artemisinin']=1

# remove infinite log parasitaemias
ind_keep = !(is.infinite(Complete_Leg_data$LPAR_pct) | is.nan(Complete_Leg_data$LPAR_pct))
Complete_Leg_data = Complete_Leg_data[ind_keep,]
```

Exploratory analysis

Let's look at the key predictive variables. We use a random effects term to model differences between studies.



Predictive value of anaemia on death adjusting for confounders

Before fitting the more complex GAM models we explore the standard glm (logistic regression) models.

```
mod_full = glmer(outcome ~ HCT + LPAR_pct + AgeInYear + coma + convulsions +
                   poedema + BUN + BD + drug_AS + (1 | studyID),
               data=Complete_Leg_data, family=binomial)
```

```
## Warning in checkConv(attr(opt, "derivs"), opt$par, ctrl = control$checkConv, : Model is nearly unider
##  - Rescale variables?
```

```
summary(mod_full)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula:
## outcome ~ HCT + LPAR_pct + AgeInYear + coma + convulsions + poedema +
##     BUN + BD + drug_AS + (1 | studyID)
##    Data: Complete_Leg_data
##
##      AIC      BIC   logLik deviance df.resid
##   3544.7   3618.6  -1761.4   3522.7     6086
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -5.9680 -0.3455 -0.2028 -0.1256 11.1231
##
## Random effects:
##  Groups  Name        Variance Std.Dev.
##  studyID (Intercept) 0.01816  0.1347
## Number of obs: 6097, groups:  studyID, 6
##
## Fixed effects:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.974463   0.212231 -23.439  < 2e-16 ***
## HCT           0.015164   0.005220   2.905 0.003673 **
## LPAR_pct      0.050297   0.059974   0.839 0.401669
## AgeInYear     0.018937   0.003815   4.963 6.92e-07 ***
## coma          1.413238   0.097151  14.547  < 2e-16 ***
## convulsions1  0.469218   0.111356   4.214 2.51e-05 ***
## poedema1      0.771794   0.371709   2.076 0.037863 *
## BUN           0.014147   0.001574   8.985  < 2e-16 ***
## BD            0.129146   0.007000  18.449  < 2e-16 ***
## drug_AS      -0.324270   0.089117  -3.639 0.000274 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) HCT    LPAR_p AgInYr coma   cnvls1 poedm1 BUN    BD
## HCT         -0.667
## LPAR_pct    -0.096  0.008
## AgeInYear   -0.246 -0.114 -0.060
## coma        -0.244 -0.066  0.062 -0.073
## convulsins1 -0.054 -0.075  0.036  0.102 -0.243
## poedema1    -0.010 -0.005 -0.005 -0.089  0.012  0.006
## BUN         -0.238  0.116 -0.059 -0.155  0.006  0.048 -0.021
## BD          -0.435  0.223 -0.170  0.164  0.001  0.062  0.022 -0.204
## drug_AS     -0.165 -0.011 -0.017 -0.072 -0.005  0.001 -0.024 -0.052 -0.013
## convergence code: 0
## Model is nearly unidentifiable: very large eigenvalue
##  - Rescale variables?
```

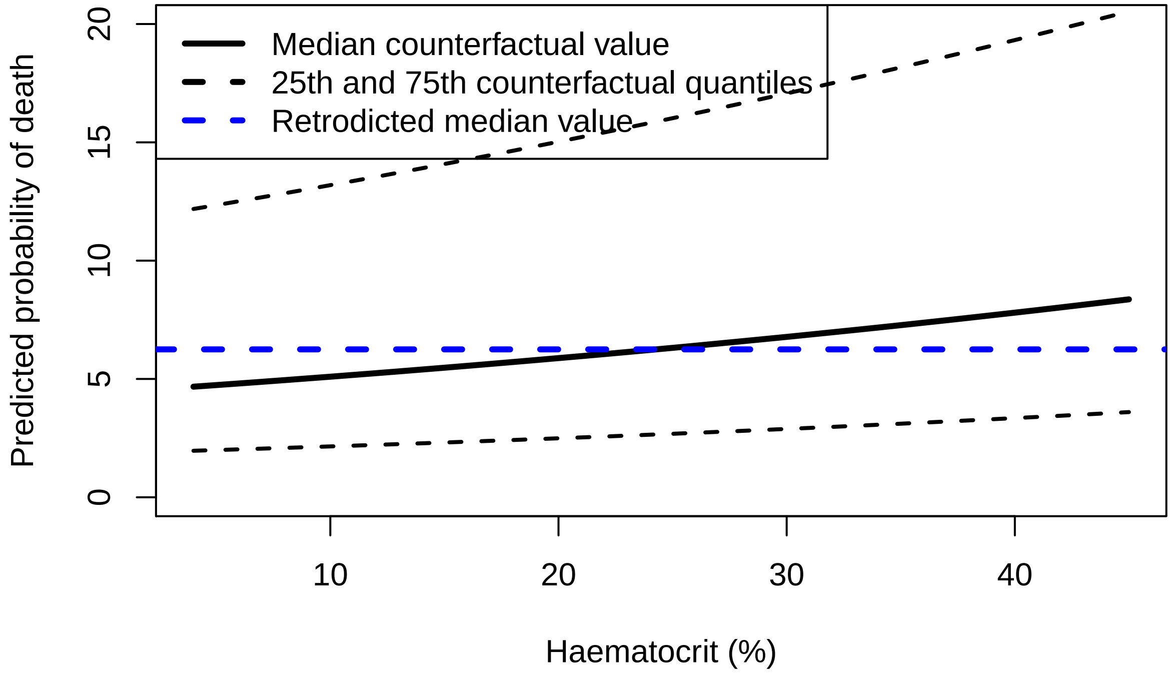
Now let's make counterfactual predictions of anaemia on death for the patients in the database.

```
myquantiles = c(0.25,0.5,0.75) # this is 50% predictive interval

overall_median_mortality = median(100*predict(mod_full, type='response'))
par(las=1, bty='n')
x_hcts = seq(4,45, by=1)
probs_lin = array(dim = c(3, length(x_hcts)))
for(i in 1:length(x_hcts)){
  mydata = Complete_Leg_data
  mydata$HCT=x_hcts[i]
  ys = 100*predict(mod_full, newdata = mydata, re.form=NA, type='response')
  probs_lin[,i] = quantile(ys, probs=myquantiles)
}
```

The way to interpret this ‘counterfactual’ plot is as follows: suppose that every individual in the dataset was assigned (as in a intervention) a specific haematocrit X, what would the resulting per patient probability of death be. Here we summarise these probabilities by the predicted mean probability of death and 80% predictive intervals.

```
plot(x_hcts,probs_lin[2,], xlim=c(4,45), ylab='Predicted probability of death',
     xlab='Haematocrit (%)', ylim=c(0,20), lty=1, lwd=3, type='l')
lines(x_hcts, probs_lin[1,], lty=2, lwd=2)
lines(x_hcts, probs_lin[3,], lty=2, lwd=2)
abline(h=overall_median_mortality, lwd=3, col='blue',lty=2)
legend('topleft', col=c('black','black','blue'), lwd=3, lty=c(1,2,2),
       legend = c('Median counterfactual value', '25th and 75th counterfactual quantiles','Retrodicted
```



More complex GAM model

The GAM model allows for non-linear relationships between certain variables and the outcome.

Here we fit as non-linear the effect of age and haematocrit on mortality. We add a random effect term for the studyID We should also be doing this for the study site…

```
mod_full_GAM = gam(outcome ~ s(HCT,AgeInYear) + LPAR_pct  + coma + convulsions +
                     poedema + BUN + BD + drug_AS + s(studyID, bs='re'),
                 data=Complete_Leg_data, family=binomial)
summary(mod_full_GAM)
```

```
##
## Family: binomial
## Link function: logit
##
## Formula:
## outcome ~ s(HCT, AgeInYear) + LPAR_pct + coma + convulsions +
##     poedema + BUN + BD + drug_AS + s(studyID, bs = "re")
##
```

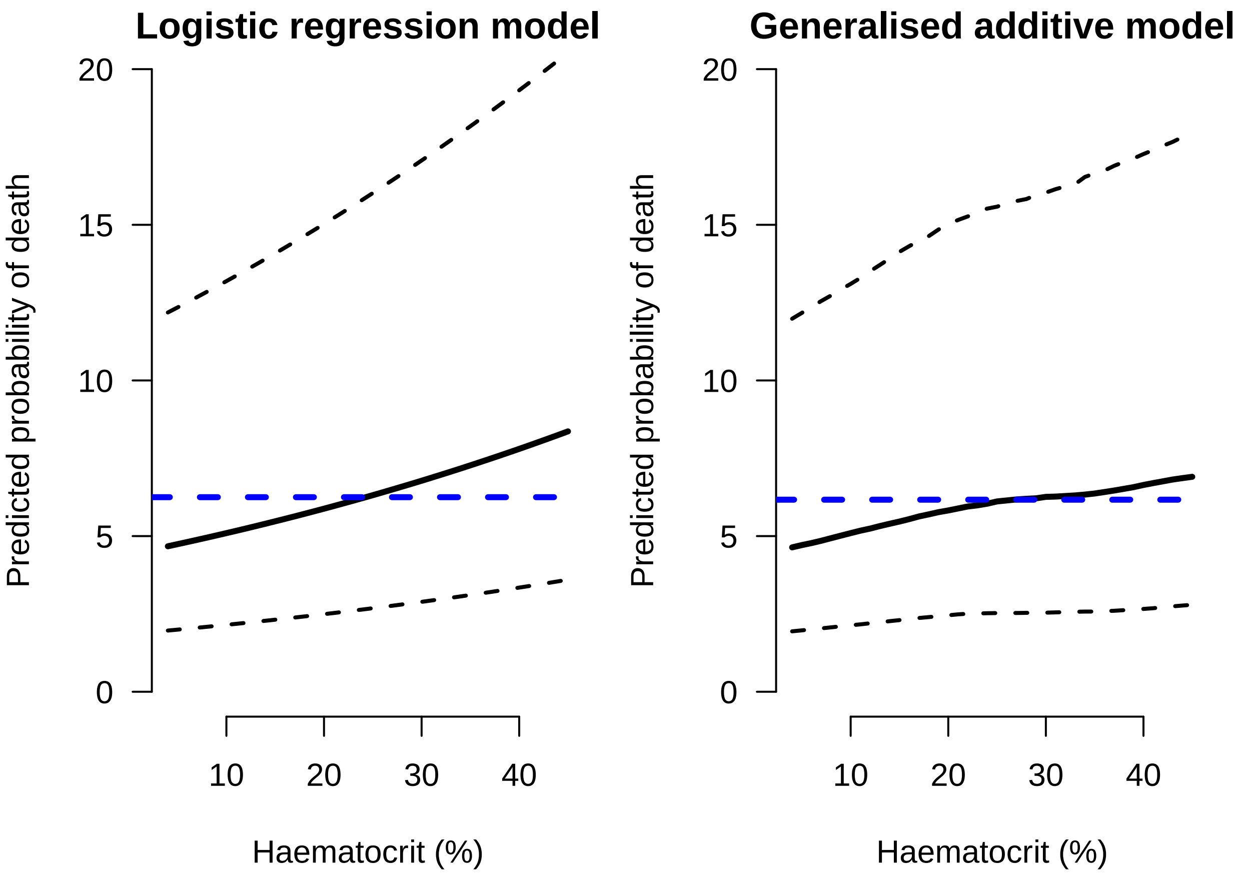
```
## Parametric coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  -4.397435   0.135968 -32.342  < 2e-16 ***
## LPAR_pct      0.050612   0.059250   0.854 0.392986
## coma          1.400706   0.096829  14.466  < 2e-16 ***
## convulsions1  0.510489   0.111814   4.566 4.98e-06 ***
## poedema1      0.748149   0.366194   2.043 0.041048 *
## BUN           0.013114   0.001601   8.192 2.57e-16 ***
## BD            0.132736   0.007276  18.243  < 2e-16 ***
## drug_AS      -0.332430   0.088940  -3.738 0.000186 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Approximate significance of smooth terms:
##                     edf Ref.df Chi.sq  p-value
## s(HCT,AgeInYear) 6.846  9.561 73.442 8.24e-12 ***
## s(studyID)       1.172  5.000  1.505    0.252
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## R-sq.(adj) =  0.251   Deviance explained = 26.7%
## UBRE = -0.42123  Scale est. = 1         n = 6097
```

Now we compute the corresponding counterfactual probabilities of death for the dataset for all values of the haematocrit:

```
overall_median_mortalityGAM = median(100*predict(mod_full_GAM, type='response'))
par(las=1, bty='n')
probs_gam = array(dim = c(3, length(x_hcts)))
for(i in 1:length(x_hcts)){
  mydata = Complete_Leg_data
  mydata$HCT=x_hcts[i]
  ys = 100*predict(mod_full_GAM, newdata = mydata, type='response')
  probs_gam[,i] = quantile(ys, probs=myquantiles)
}
```

We see that the effect of haematocrit on mortality is non-linear under this model: below 20 is protective, above 20 plateaus out:

```
#
par(las=1, mfrow=c(1,2), bty='n', mar=c(4,4,1,1))
### Plot the standard logistic regression model
plot(x_hcts,probs_lin[2,], xlim=c(4,45), ylab='Predicted probability of death',
     xlab='Haematocrit (%)', ylim=c(0,20), lty=1, lwd=3, type='l')
lines(x_hcts, probs_lin[1,], lty=2, lwd=2)
lines(x_hcts, probs_lin[3,], lty=2, lwd=2)
abline(h=overall_median_mortality, lwd=3, col='blue',lty=2)
title('Logistic regression model')
### And now the GAM model
plot(x_hcts,probs_gam[2,], xlim=c(4,45), ylab='Predicted probability of death',
     xlab='Haematocrit (%)', ylim=c(0,20), lty=1, lwd=3, type='l')
lines(x_hcts, probs_gam[1,], lty=2, lwd=2)
lines(x_hcts, probs_gam[3,], lty=2, lwd=2)
abline(h=overall_median_mortalityGAM, lwd=3, col='blue',lty=2)
title('Generalised additive model')
```


Logistic regression model
Generalised additive model
Predicted probability of death
Predicted probability of death
0
5
10
15
20
0
5
10
15
20
10
20
30
40
10
20
30
40
Haematocrit (%)
Haematocrit (%)